Neurocomputing 452 (2021) 159–168

Contents lists available at ScienceDirect

Neurocomputing

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

TSSN-Net: Two-step Sparse Switchable Normalization for learning correspondences with heavy outliers

Zhen Zhong [a,b], Guobao Xiao [a,*], Kun Zeng [a], Shiping Wang [c]

[a] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, College of Computer and Control Engineering, Minjiang University, Fuzhou 350108, China
[b] Research Base of Traditional Chinese Medicine Syndrome, Fujian University of Traditional Chinese Medicine, Fuzhou 350108, China
[c] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

ARTICLE INFO

Article history:
Received 2 August 2020
Revised 21 March 2021
Accepted 22 April 2021
Available online 29 April 2021
Communicated by Zidong Wang

Keywords:
Deep learning
Feature matching
Adaptive normalization
Multi-scale neighborhood representation

ABSTRACT

In this paper, we solve the problem of feature matching by designing an end-to-end network (called TSSN-Net). Given putative correspondences of feature points in two views, existing deep learning based methods formulate the feature matching problem as a binary classification problem. In these methods, a normalizer plays an important role in the networks. However, they adopt the same normalizer in all normalization layers of the entire networks, which will result in suboptimal performance. To address this problem, we propose a Two-step Sparse Switchable Normalization Block, which involves the advantage of adaptive normalization for different convolution layers from Sparse Switchable Normalization and robust global context information from Context Normalization. Moreover, to capture local information of correspondences, we propose a Multi-Scale Correspondence Grouping algorithm, by defining a multi-scale neighborhood representation, to search for consistent neighbors of each correspondence. Finally, with a series of convolution layers, the end-to-end TSSN-Net is proposed to learn correspondences with heavy outliers for feature matching. Our experimental results have shown that our network achieves the state-of-the-art performance on benchmark datasets.

1. Introduction

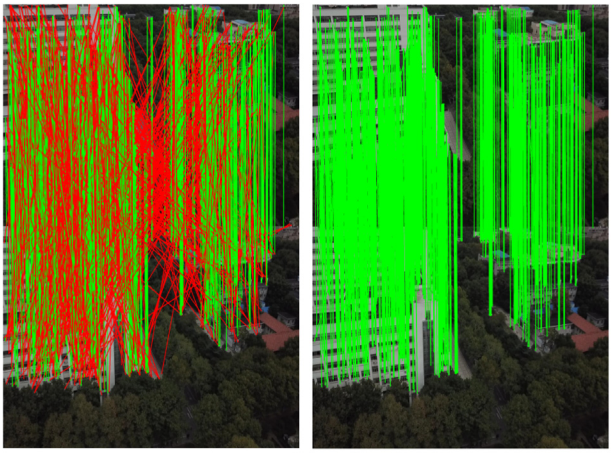
Establishing reliable feature correspondences is a fundamental problem in computer vision [1–3], e.g., Structure from Motion (SfM) [4,5], Simultaneous Localization and Mapping (SLAM) [6,7], Image Fusion [8] and geometric model fitting [9–12]. This problem relies on two steps, i.e., correspondence generation and correspondence selection [13]. The first step can be dealt with matching local key-point features (e.g., SIFT [14], Hessian-Affine Detector [15]). However, the initial correspondences are often inevitable to be contaminated by outliers (see Fig. 1) due to various problems, e.g., local key-point localization errors and ambiguities of the local descriptors. Thus, many methods (e.g., [13,16–23]) have been proposed to remove outliers.

Among current existing feature matching methods, deep learning based methods are the most popular technique due to its effectiveness, e.g., LGC-Net [20] and NM-Net [13]. LGC-Net introduces Multi-Layer Perceptrons (MLPs) and Context Normalization (CN) for feature matching. NM-Net includes a compatibility-specific mining algorithm to successively extract and aggregate local features for a hierarchical deep learning network. Despite their great successes, they adopt the same normalizer in all normalization layers of the entire networks, which will result in suboptimal performance.

Recently, Sparse Switchable Normalization (SSN) [24], which formulates the optimization problem into the differentiable feed-forward computation problem by a sparse learning algorithm, is proposed to adaptively select normalizers for the tasks of image classification, semantic segmentation and action recognition. Specifically, SSN employs a sparse learning algorithm to adaptively select the most suitable normalization, from Batch Normalization (BN) [25], Instance Normalization (IN) [26], and Layer Normalization (LN) [27], for each normalization layer of a deep network. As we know, different normalizers have their own advantages and disadvantages. For example, BN acts as a feature regularizer and improves generalization of the network, but it is sensitive to the value of batch size (i.e., the number of samples per GPU) since the mean and standard deviation of BN are derived from the batch size. In contrast, IN and LN are not sensitive to the value of batch size but they have a limited generalization ability of the network for feature matching. Thus, through adaptively select normalizers,

[*] Corresponding author.
E-mail address: gbx@mju.edu.cn (G. Xiao).

https://doi.org/10.1016/j.neucom.2021.04.093
0925-2312/© 2021 Elsevier B.V. All rights reserved.



(a) Initial correspondences (b) Results of our network

Fig. 1. Illustration of (a) initial correspondences and (b) correspondence selection results of our network. If they conform to the ground-truth epipolar geometry we draw them in green, and otherwise in red.

SSN can inherit all benefits of normalizers (i.e., BN, IN, LN) and also has good generalization ability.

In this paper, we propose a hierarchical deep learning network (called TSSN-Net) to learn correspondences with heavy outliers for feature matching. The architecture of the proposed TSSN-Net is shown in Fig. 2. Specifically, we propose to add SSN into the deep network, to improve the performance of current existing deep networks for feature matching. Note that the performance of SSN depends on the effectiveness of the sparse learning algorithm, which trains deep models with sparse constraints. However, the input data for feature matching problem has not obvious sparse space. To address this issue, we propose a novel Two-step Sparse Switchable Normalization block (TSSN Block), which introduces CN to construct the sparse space for SSN. Thus, TSSN Block is able to inherit all benefits from different normalizers for feature matching.

In addition, to extract local features, we search for consistent neighbors of each correspondence according to the compatibility-specific distance, which is able to find more reliable neighbors than spatial distance for feature matching. To further improve the generalization ability of the proposed network, we propose a Multi-Scale Correspondence Grouping algorithm, which searches for neighbors of each correspondence and integrates correspondence and neighbors to a graph, by a novel multi-scale neighborhood representation.

The contributions are summarized as follows:

- We develop an end-to-end network to handle the feature matching problem with heavy outliers. The proposed network is able to achieve the state-of-the-art performance on three benchmark datasets with various proportions of outliers.

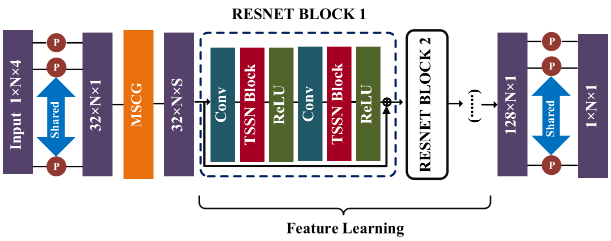


Fig. 2. The proposed TSSN-Net architecture.

- We propose a novel normalization block, which effectively involves the global context information and adaptively switches normalizers in different convolution layers, for feature matching. The proposed block is able to significantly improve the performance of our network since it inherits all benefits of from different normalizers.
- We propose a correspondence grouping algorithm to capture local information of correspondences. The proposed algorithm searches for neighbors based on compatibility-specific distance and integrate correspondences with neighbors to a graph, which is permutation-equivariant to initial correspondences. Therefore, our network is insensitive to the order of correspondences and has good generalization ability for feature matching.

The rest of the paper is organized as follows: We briefly review the related work and background material in Section 2. We present our network for feature matching in Section 3. In Section 4, we describe and analyze the experimental results on three benchmark datasets. At last, we draw conclusions in Section 6.

2. Related work

In this section, we briefly review the related feature matching methods, including parametric methods, non-parametric methods and learning based methods, and we also review some normalization techniques that the proposed method is based on.

2.1. Parametric methods

Parametric methods often use some predefined parametric models to solve the matching problem. For example, RANSAC [16] and its variations (e.g., PROSAC [28], SCRAMSAC [29] and USAC [30]) predefine a model of homography or essential matrices, and then use a generation-verification strategy for feature matching. FG-GMM [31] formulates the match problem as an estimation of a feature-guided mixture of densities, and then exploits the unified maximum-likelihood framework to handle the matching problem. SDF [9] also predefines a model of homography, and then uses superpixels to get the feature appearances.

These parametric methods can obtain good results with proper parameters for most of rigid matching, but they suffer from some weaknesses: 1) they cannot effectively work well when the ratio of inliers is low. 2) they often use a single geometric model, which cannot express the complex models, such as non-rigid matching and multi-consistency matching.

2.2. Non-parametric methods

Non-parametric methods mine the local information for correspondence selection. For example, LPM [19] uses spatial neighbor relationships to remove outliers and retain inliers. GLPM [32] formulates the neighborhood structures of accurate potential matches between two images into a mathematical model. GMS [33] utilizes the spatially local information to exploit the statistical method. SM [34] establishes the adjacency matrix of the graph, where the nodes represent the potential correspondences, and the weights on the links represent pairwise agreements between potential correspondences. These methods involve the compatibility information among correspondences. However, they have not mined local information from compatible correspondences.

2.3. Learning based methods

Learning based methods have made a huge success in a wide range of computer vision tasks. Many researchers attempted to employ learning-based methods to solve matching tasks. Existing learning based feature matching methods can be roughly classified into two categories, according to the two tasks, i.e., correspondence generation and outlier rejection. For correspondence generation, the feature matching methods [35–38] construct sparse point correspondences from image pairs of the same or similar scene by using deep learning architecture. Outlier rejection methods [13,20,39] employ Point-Net-like architecture [40,41] to remove outliers. In addition, these methods have proven to be better than hand-made representations [42,43]. LGC-Net [20] captures the global context information through CN and embeds the context information in the node, but its correspondence data is easily affected by other image pair data. NM-Net [13] defines the neighbors by the compatibility-specific distance of the corresponding relationship, and mines local information according to the defined neighbor combination. Nevertheless, NM-Net neglects the information between neighbors. In contrast, the proposed TSSN-Block in this paper use a learnable manner to switch normalizers adaptively in different convolution layers, to reduce the impact of other matching pairs.

2.4. Normalization techniques

The normalization technique is originally proposed to make deep neural network (DNN) training faster and the initialization works adopt the assumption of feature distributions to normalize hidden features. However, with the training evolves, these works become invalid. BN [25] is a milestone technique in the Normalization field and it normalizes data along the batch dimension. However, it is limited by batch size (i.e., a small batch leads to poor performance when adopting BN). LN [27] normalizes data along the channel dimension and can accelerate the training of recurrent neural networks (RNNs). In addition, LN performs well in the small batch. IN [26] performs along with each sample and it can filter out complex appearance variances. Nevertheless, they perform poorer than BN in the big batch because of losing global information. In addition, these normalizers are used in the entire network, which results in suboptimal performance. Consequently, Switchable Normalization (SN) [44] learns importance ratios to compute the weighted average statistics of BN, LN and IN. However, SN suffers from the slow speed and heavy computation cost. SSN employs SparsestMax to adaptively select the most suitable normalizer from BN, LN and IN, for speeding up the network training process. Nevertheless, SSN cannot capture enough context information for correspondence data due to rare sparse information.

To address this issue, we introduce CN to capture context information for the sparse information, and then adopt SSN to adaptively normalize different convolution layers.

3. Proposed method

3.1. Overview

As shown in Fig. 2, our network consists of two important modules, i.e., TSSN Block and Multi-Scale Correspondence Grouping algorithm (called MSCG). We use a hierarchically network with TSSN Block, which can capture global context information and adaptively select normalizers in different convolution layers for optimal performance. In addition, our MSCG can exploit multi-scale local information.

The details of our network are described as follows: The feature learning and aggregation layers are composed of a series of ResNet blocks. TSSN block embeds global information to each correspondence in CN, and uses Sparse Switchable Normalization to effectively improve the normalization performance in SSN. Note that convolution operation is sensitive to the order of the data and cannot effectively deal with the sparse and unordered correspondences. To address this problem, we propose MSCG to extract the local information for every correspondence, and group unordered correspondences into a graph. Then, the correspondences with neighbors will be integrated as a set of regular organizations, and the convolution can further perform the operating of the feature extraction and aggregation based on the regular organizations.

3.2. Problem formulation

Formally, given an image pairs (I, I'), we respectively extract keypoint k_i and k'_i from two images by using handcrafted features [15] for each correspondence. Then, we construct the input correspondence data D as:

$$D = [d_1; d_2; \ldots d_i \ldots ; d_N], d_i = [x_i, y_i, x'_i, y'_i] \tag{1}$$

where d_i represents correspondence, and (x_i, y_i) and (x'_i, y'_i) represent the coordinates of k_i and k'_i, respectively. N is the number of input correspondences. Then, we formulate the feature matching problem as a binary classification problem: we input the putative correspondence set (i.e., a 4D dataset), and output the probability of each correspondence to be an inlier $\{0, 1\}$. Specifically, we firstly train the neural network to get the logit values. Then we compute the predicted value of range $[0, 1)$ from the operation of $tanh(ReLU(output))$ [20]. Here, the predicted value is 0 if the corresponding correspondence is an outlier; otherwise, it is an inlier.

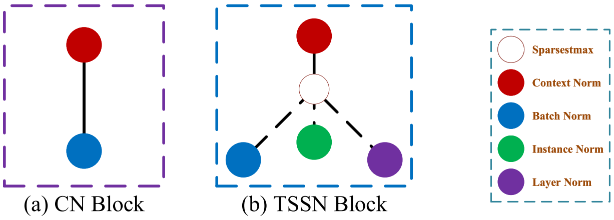
3.3. TSSN block

In LGC-Net [20], CN and BN, (called CN Block) are used to extract global context feature and speed up network training. Although BN can improve the generalization of the network, it is limited by the batch size. To address this problem, we propose the Two-step Switchable Normalization block (TSSN Block). As shown in Fig. 3, our block can be divided into two steps. In the first step, we use a simple but effective strategy (i.e., CN) to normalize correspondences of each image pair:

$$TSSN_1(d_i^l) = \frac{(d_i^l - u^l)}{o^l} \tag{2}$$

where $TSSN_1(d_i^l)$ denotes the output correspondence d_i of layer l. u^l and o^l represent mean and standard deviation, respectively.

Global information embeds every correspondence through the operation of the first normalization step. However, the correspondence information of CN is easily disturbed by the information from other image pairs. The main reason is that the following BN normalizes the features by a batch of correspondences from different image pairs. This operation will mix the context information of different image pairs and disturb the correspondence information from CN. In addition, BN is sensitive to batch size. To solve above problems, we adopt a sparse learning algorithm (i.e., SparsestMax)



(a) CN Block (b) TSSN Block

Fig. 3. The visualization of CN Block and TSSN Block. The solid line indicates the connection for which data transmission is necessary, and the dotted line indicates the connection for which data transmission is possible.

to select a proper normalizer (i.e., BN, IN, LN) in the second step. Specifically, the second normalization step (i.e., SSN) adopted a learnable strategy to select a proper normalization that operation can remain the advantage and avoid the disadvantage of each normalizer. Then we propose the second normalization step to enhance the global feature information of each correspondence and speed up the training process.

We formulate the second step $TSSN_2(d_i^l)$ as:

$$TSSN_2(d_i^l) = \lambda \frac{TSSN_1(d_i^l) - \sum_{n=1}^{|\psi|} r_n u_n}{\sqrt{\sum_{n=1}^{|\psi|} r'_n o_n^2 + \epsilon}} + \beta$$
$$s.t. \sum_{n=1}^{|\psi|} r_n = 1, \sum_{n=1}^{|\psi|} r'_n = 1, \forall r_n, r'_n \in \{0,1\} \tag{3}$$

where λ and β represent a scale and a shift parameter, respectively. $|\psi|$ denotes a set composition by $\{LN, BN, IN\}$. u_n and o_n^2 are their means and variances, and $n = 1,2,3$ corresponds to different normalizers. r_n and r'_n are importance ratios of mean and variance, respectively.

3.4. MSCG

LGC-Net [20] is a pioneer in deep learning on feature matching, and MLP plays a core role in processing unordered correspondences on this network. However, LGC-Net fails to utilize the local information of each correspondence. To deal with this problem, we propose MSCG to mine the local information and we also employ ResNet to extract and aggregate the local information. The Schematic illustration of MSCG is shown in Fig. 4.

Then we use a Hessian-affine-based method [15] to detect keypoints, which involves the local affine information required by the further compatibility metric. The keypoint is described as a 3 × 3 matrix A_i:

$$A_i = \begin{bmatrix} v_i & u_i \\ 0 & 1 \end{bmatrix} \tag{4}$$

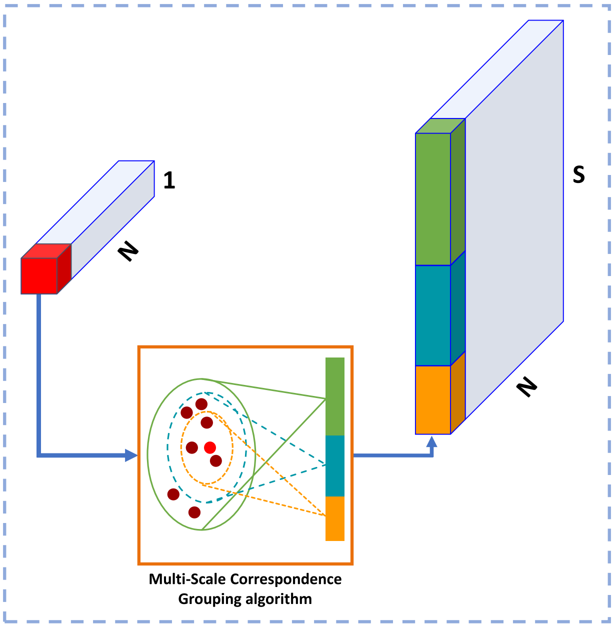


Fig. 4. Schematic illustration of MSCG. The closed curves of different colors represent different scales..

where v_i is a 2 × 2 matrix that denotes the local affine information extracted by the Hessian-affine detector. u_i represents the position of the keypoint.

The transformation J_j of a correspondence d_i is calculated as [13]:

$$J_i = A'_i A_i^{-1} \tag{5}$$

Note that if J_i and J_j are more consistent, then d_i and d_j will be more similar (i.e., the correspondences are more adjacent). After that, we compute the similarity of two correspondences (d_i, d_j) by:

$$s_j(c_i) = \left| \alpha\left(J_j \cdot \begin{bmatrix} u_i \\ 1 \end{bmatrix}\right) - \alpha\left(J_i \cdot \begin{bmatrix} u_i \\ 1 \end{bmatrix}\right) \right| \tag{6}$$

where $\alpha([abc]^T) = [a/cb/c]^T$.

To make the final result in the range of (0,1], we use a Gaussian kernel to evaluate the similarity:

$$S(d_j, d_i) = e^{-\lambda(s_j(d_i) + s_i(d_j))} \tag{7}$$

where λ is a parameter and it is experimentally 10^{-3}.

After that, we construct graphs to group correspondences by preserving the local neighborhood structure. Unlike the normal KNN-based method, which roughly combines the neighbors (i.e., the top rank of similarity with the correspondence). In order to capture the information of neighbors accurately, we use a multi-scale neighborhood representation to address the general feature matching problem.

Specifically, given a correspondence d_i, we first compute the similarity between d_i and the other correspondences by Eq. (7), and then, we define a set of neighbors with multiple scales $\mathcal{N}_i = \{K_i^m\}_{m=1}^M$, where K_i^m is the m nearest neighbors of d_i and M is the number of scales. After that, we sequentially link c_i with all its neighbors to construct a graph G_i, which is used to represent d_i.

It is worth pointing out that, NM-Net also uses the local neighborhood information to group correspondences as the proposed method. However, NM-Net requires searching for a fixed number of nearest neighbors, which is problematic for addressing the general feature matching problem, due to some reasons, e.g., the distribution of the initial correspondences is often nonuniform, and the inlier ratio also varies with different input datasets. In contrast, the proposed MSCG is much more general for addressing the feature matching problem than NM-Net.

3.5. Loss function

Recall that, the feature matching problem is formulated as the binary classification problem in our network. Given the initial correspondence data $D = [d_1; d_2; \ldots\ldots; d_N]$, our network outputs the predicted probability $W = [w_1; w_2; \ldots\ldots w_N]$. Thus, we employ a simple but effective loss function, i.e. cross entropy loss:

$$L = \frac{1}{N}\sum_{i=1}^{N} \ddot{\alpha}_i F(y_i, Logic(p_i)) \tag{8}$$

where $\ddot{\alpha}_i$ denotes a self-adaptive weight of balancing the positive and negative samples, $Logic(\cdot)$ represents the logistic function, $F(\cdot)$ indicates the binary cross entropy function, y_i and p_i represent the ground truth and predicted value, respectively. Of course, we can adopt other loss functions to replace cross entropy loss, but the experimental results show our simple loss function can achieve good performance on three benchmark datasets.

Table 1
Detail of the experimental dataset. The inlier ratio denotes the average proportion of inliers in a whole dataset and the VP means viewpoint.

Dataset	Image pairs	Inlier ratio (%)	Challenges
NARROW	20209	41.57	VP changes
WIDE	12760	32.57	VP changes
COLMAP	21560	7.55	VP changes, rotation.

4. Experimental results

In the section, we evaluate the performance of our TSSN-Net on three benchmark datasets: NARROW, WIDE [13] and COLMAP [45]. All experiments are performed on Linux 3.10.0 with NVIDIA TESLA $P100$ GPUs.

4.1. Experimental setup

4.1.1. Architecture details

The details of our architecture are shown in Fig. 2. Specifically, the architecture of TSSN-Net is $C(1,1,4) - MSCG - C(32,1,3) - R(32,1,3) - R(32,1,3) - R(32,1,3) - R(64,1,3) - R(64,1,3) - R(64,1,3) - R(128,1,3) - R(128,1,3) - R(128,1,3) - R(256,1,3) - C(256,1,3) - C(128,1,3) - O$. $C(c,h,w)$ represents the convolution layer has c input channels. In addition, the convolution kernel is $h \times w$. $R(c,h,w)$ denotes a ResNet block, which contains two convolutional layers and TSSN blocks. Specifically, it consists of a two-layer structure $C(c,h,w)$ - TSSN Block - ReLU activation.

4.1.2. Benchmark datasets

We evaluate the performance of eight competing methods on three benchmark datasets, i.e., COLMAP [45], WIDE [13] and NARROW [13] (see Table 1). COLMAP is a public dataset, which contains four sequences (i.e., south, person, graham and gerrard). Our evaluating image pairs have the same structures as NM-Net, and the WIDE and NARROW datasets are made by drone photography. These two datasets contain four sequences that get from the interval 10 and 20 samples, respectively. We use VisualSFM [4] to obtain the ground-truth camera parameters and ground-truth

Table 2
Quantitative comparison results on three benchmark datasets from top to bottom (NARROW, WIDE, and COLMAP datasets). The best result in each column is boldfaced.

Dataset	Method	F-measure	Precision	Recall
NARROW	RANSAC	0.8003	0.8189	0.8003
	LPM	0.7867	0.8765	0.7328
	Point-Net	0.9143	0.8937	0.9402
	Point-Net++	0.9683	0.9634	0.9741
	LGC-Net	0.9690	0.9690	0.9700
	NM-Net	0.9543	0.9611	0.9494
	TSSN-Net1	**0.9751**	**0.9761**	**0.9745**
	TSSN-Net2	0.9718	0.9726	0.9714
WIDE	RANSAC	0.7002	0.7508	0.6812
	LPM	0.7020	0.7837	0.6552
	Point-Net	0.7994	0.7468	0.8716
	Point-Net++	0.9254	0.9156	0.9382
	LGC-Net	0.9286	0.9242	0.9346
	NM-Net	0.9330	0.9182	**0.9498**
	TSSN-Net1	0.9341	0.9318	0.9373
	TSSN-Net2	**0.9352**	**0.9320**	0.9355
COLMAP	RANSAC	0.1914	0.2222	0.1879
	LPM	0.2213	0.2415	0.2579
	Point-Net	0.1493	0.0911	0.3847
	Point-Net++	0.3298	0.2545	0.5668
	LGC-Net	0.3953	0.3063	**0.6839**
	NM-Net	0.3804	0.3029	0.5784
	TSSN-Net1	0.3820	0.3289	0.4907
	TSSN-Net2	**0.4357**	**0.3733**	0.5518

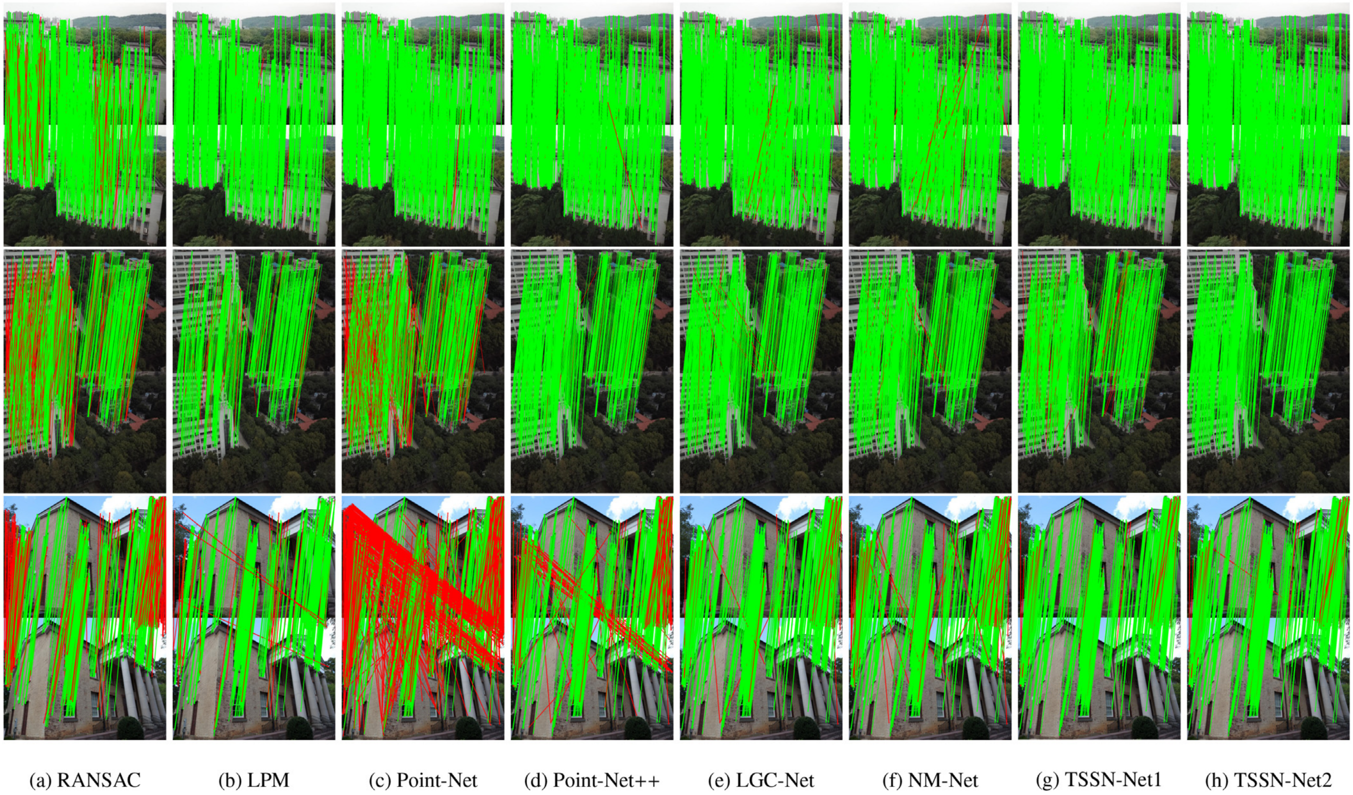


Fig. 5. Visual of (a) RANSAC, (b) LPM, (c) Point-Net, (d) Point-Net++, (e) LGC-Net, (f) NM-Net, (g) TSSN-Net1 and (e) TSSN-Net2 on three datasets (top to bottom: NARROW, WIDE, and COLMAP dataset). We draw the correspondences in green if they conform to the ground truth epipolar geometry, and in red otherwise.

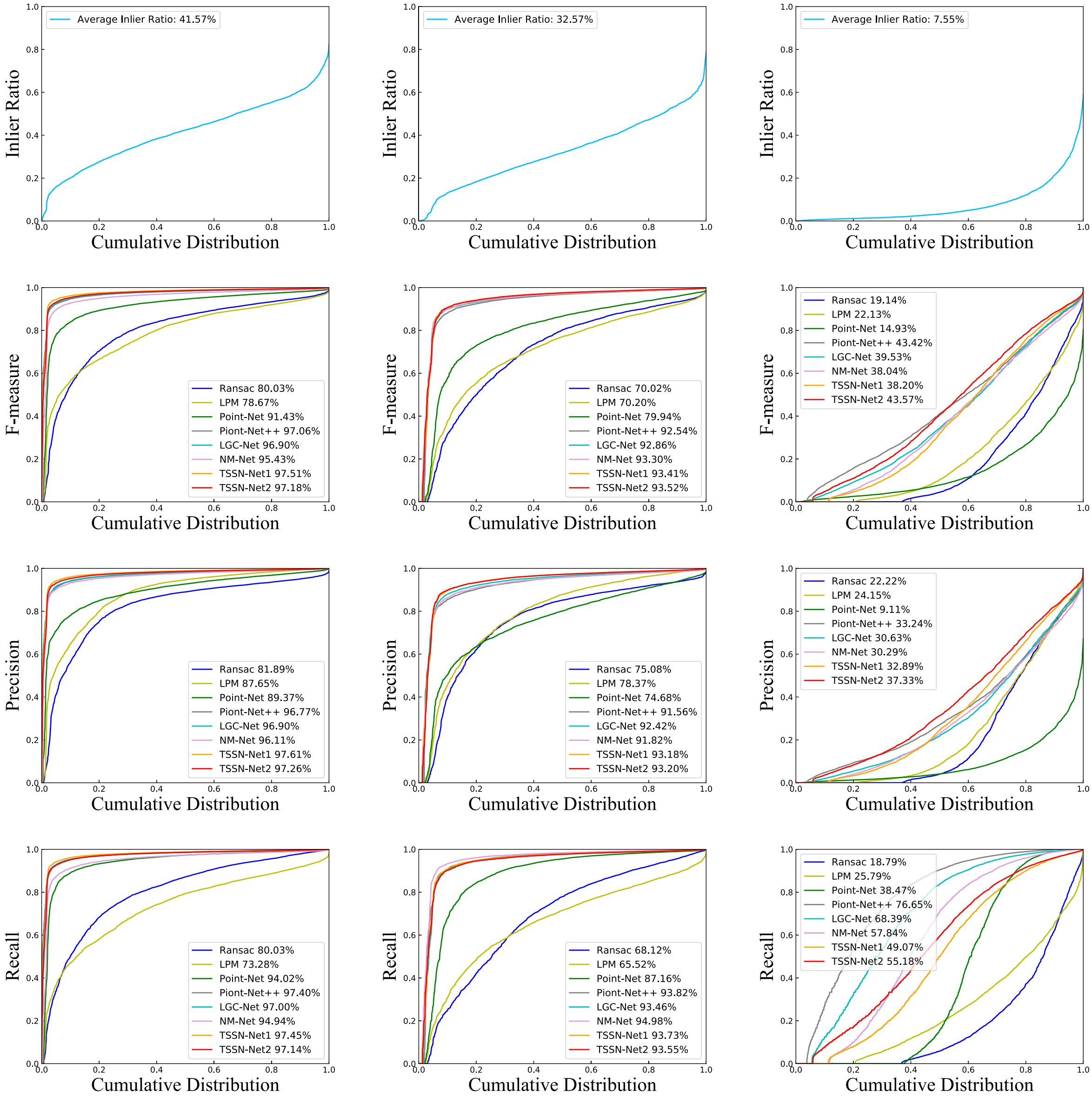


Fig. 6. Quantitative comparison obtained by eight competing methods on three datasets. From top to bottom: results on the datasets of NARROW, WIDE, and COLMAP. From left to right: Inlier Ratio, F-measure, Precision and Recall. A point on the curve with coordinate (x, y) denotes that there are 100 × x percents of image pairs which have values no more than y.

labels by the Epipolar distance. The Epipolar distance is the distance between a correspondence and a Epipolar line [20]:

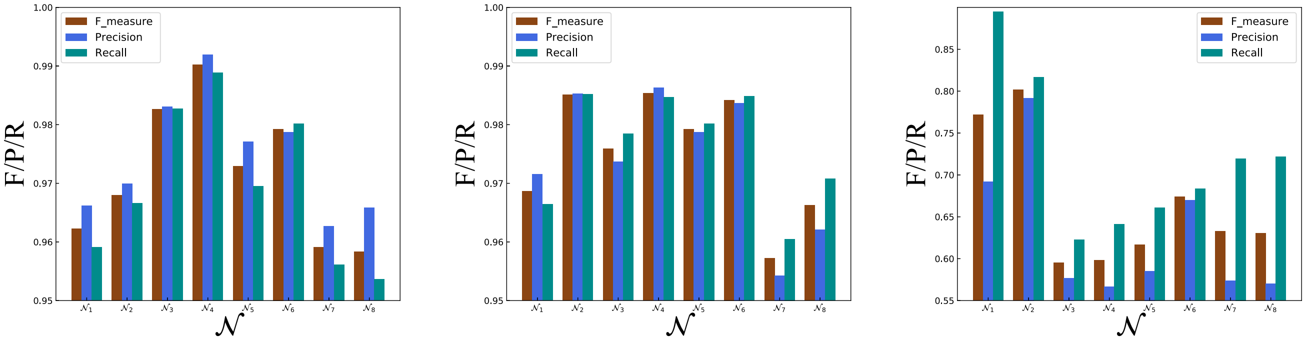
$$dist(p_i, Ep_i') = \frac{p_i'^T E p_i}{\sqrt{(Ep_i)_{[1]}^2 + (Ep_i)_{[2]}^2}}, \tag{9}$$

where p_i and p_i' indicate two keypoint positions from the putative correspondence d_i. E is the essential matrix derived from the camera parameters. $A_{[j]}$ represents the j-th element of vector A. A correspondence is labeled an inlier if its corresponding Epipolar distance is less the threshold (10^{-4}); Otherwise, it is labeled an outlier. It is worth pointing out that, the labels are weakly supervised. Thus, a few outliers may still have a small epipolar distance when they are close to the Epipolar line, and then they will be wrongly labeled as inliers. Of course, we can adopt a fully supervised labeling strategy to avoid this problem, but it would require annotating correspondences for every point in both images, which is extremely expensive and not available in most scenarios. Even so, a few outliers will not affect the accuracy of performance evaluation of matching methods [20].

Finally, we adopt the top 70% image pairs as the training dataset, the following 15% image pairs as the validation dataset, and the last 15% image pairs as the test dataset.

4.1.3. Competing methods

To evaluate the performance of our method, we compare it with six state-of-the-art feature matching methods, which contain a classic method (i.e., RANSAC), a hand-crafted method (i.e., LPM) and four learning-based methods (i.e., Point-Net [40], Point-Net++ [41], LGC-Net [20] and NM-Net [13]). For LGC-Net, we use an adapted version to the author, which is a PyTorch version. For Point-Net and Point-Net++, we employ the official implementation. However, Point-Net and Point-Net++ are suitable for processing 3D data. Thus, we implement the 4D version of Point-Net and Point-Net++. For NM-Net, we adopt the official implementation. Each learning-based method is trained by Adam optimizer with the



(a) NARROW-main (b) WIDE-science (c) COLMAP-south

Fig. 7. Analysis of multi-scale neighborhood representation $\mathcal{N}$.

learning rate of 10^{-3} and batch size of 16. We use a simple yet effective cross-entropy loss to calculate the deviation between the predicted values and labels. For the proposed network, we test two versions: TSSN-Net1, which does not use MSCG in the network, and TSSN-Net2, which uses the MSCG in the network.

4.1.4. Evaluation criteria

To measure the correspondence selection performance, we employ three evaluation criteria: Precision, Recall, and F-measure. The three evaluation criteria are computed as follows:

$$Precision = \frac{T_P}{T_P + F_P} \tag{10}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{11}$$

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{12}$$

where T_P is True Positive. F_P represents False Positive and F_N denotes False Negative. We can see that, Precision is the proportion of predicted positives that are correctly true positives, and it is used to measure the accuracy of predicted positives; Recall is the proportion of true positives that are correctly predicted positives, and it is used to measure the coverage of true positives by predicted positives; F-measure references the true positives to the Arithmetic Mean of predicted positives and true positives, and it is the harmonic average of recall and precision. For the three evaluation criteria, larger values indicate better classification accuracy.

4.2. Comparative results

4.2.1. Qualitative illustration

We first present some visualization results obtained from the eight competing methods in Fig. 5, where we test all eight methods on three representative image pairs from NARROW, WIDE, and COLMAP, respectively. We can see that, all competing methods perform well in the two datasets (i.e., NARROW and WIDE). However, the COLMAP dataset has some challenges for all methods because of its extremely high percentage of outliers. Our TSSN-Net1/TSSN-Net2 are able to achieve better results than the other competing methods on three challenging scenes such as textureless objects, wide baselines, and large illumination changes. In contrast, RANSAC/LPM remove outliers according to the geometric constraint, but the no-rigid matching often has not obvious geometric information. Thus, they often treat the false correspondences as right correspondences (especially in the COLMAP dataset). Point-Net and Point-Net++ are limited by local information getting from spatially KNN search, and LGC-Net neglects the local information extracting, while NM-Net is limited by its rough correspondence integration.

4.2.2. Quantitative comparison

We report quantitative comparison results in Table 2 and Fig. 6. We can see that our method remarkably outperforms the state-of-the-art methods. The hand-crafted methods (i.e., RANSAC and LPM) have an unsatisfactory result, and our method gets almost 25% to improve, especially in heavy outliers dataset (i.e., COLMAP). This is because RANSAC may falsely remove the fall short of geometric constraints but true matches, and remain the meet geometric constraints but false matches. LPM focuses on computing speed but accuracy. Therefore, these two hand-crafted methods achieve unsatisfactory performance.

Point-Net++ has a low F-measure and Precision since it often considers a lot of uncertain matches as correct correspondences. In addition, Point-Net++ suffers from the irregularity of the spatial distribution of correspondences. At last, comparing with TSSN-Net1, TSSN-Net2 achieves a better performance in COLMAP and NARROW dataset. Particularly in the dataset (i.e., COLMAP) with heavy outliers, TSSN-Net2 improves 5% over TSSN-Net1 on all three evaluation criteria (i.e., F-measure, Precision and Recall). This is due to that the Normal Grouping is badly degraded in case of heavy outlier dataset. NM-Net roughly integrates correspondences with neighbors and neglects the information about itself. In contrast, the proposed MSCG is much more general for addressing the feature matching problem. Therefore, the performance of TSSN-Net2 is slightly different on the NARROW dataset, and has significant advantages on WIDE and COLMAP, especially for COLMAP.

4.3. Ablation studies

4.3.1. The multi-scale neighborhood representation $\mathcal{N}$

The multi-scale neighborhood representation $\mathcal{N}$ is use to construct graphs for correspondence representation. To evaluate the impact of different $\mathcal{N}$, we test TSSN-Net with eight different groups (i.e., $\mathcal{N}_1 = \{2,4\}$, $\mathcal{N}_2 = \{2,3,4\}$, $\mathcal{N}_3 = \{4,8\}$, $\mathcal{N}_4 = \{4,6,8\}$, $\mathcal{N}_5 = \{8,16\}$, $\mathcal{N}_6 = \{6,10,16\}$, $\mathcal{N}_7 = \{16,32\}$, $\mathcal{N}_8 = \{11,21,32\}$) on three sequences, i.e., NARROW-main, WIDE-science, and COLMAP-south. We show the Precision, Recall, and F-measure

Table 3
Quantitative comparison results obtained by the proposed method with different batch sizes on three sequences from top to bottom (NARROW-main, WIDE-science, and COLMAP-south). The best result in each column is boldfaced.

Dataset	Batch Size	2	4	8	16	32
Main	F-measure	0.9554	0.9593	0.9781	**0.9902**	0.9531
	Precision	0.9614	0.9657	0.9734	**0.9919**	0.9597
	Recall	0.9490	0.952	0.9725	**0.9846**	0.9465
Science	F-measure	0.9545	0.9598	0.9762	**0.9853**	0.9492
	Precision	0.9602	0.9613	0.9756	**0.9862**	0.9543
	Recall	0.9486	0.9584	0.9769	**0.9846**	0.9465
South	F-measure	0.5761	0.5931	0.5965	**0.5979**	0.5945
	Precision	0.5391	0.5636	**0.5673**	0.5664	0.5572
	Recall	0.6178	0.6257	0.6322	**0.6413**	0.6372

derived from the matching results obtained by the eight networks in Fig. 7.

We can see that, when we consider the same neighbors but with different sizes of elements, such as, $\mathcal{N}_1 = \{2,4\}$ and $\mathcal{N}_2 = \{2,3,4\}$, TSSN-Net with $\mathcal{N}_2$ achieves better performance than that with $\mathcal{N}_1$. The reason behind this is that, the neighborhood representation with more elements will involve more local information to further increase the separability of a correspondence. We also find that the results vary with the inlier ratio. That is, if the sequence has a high inlier ratio (i.e, NARROW-main and WIDE-science), then TSSN-Net with more neighbors can achieve better results. In contrast, if the sequence has a low inlier ratio (i.e., COLMAP-south), then TSSN-Net with less neighbors can achieve better results. This is because, the local information with more neighbors will not be preserved if the neighbors consist of more outliers for the sequence with a low inlier ratio. Thus, if we focus on the dataset with heavy outliers, we can set the groups with small values, e.g., $\mathcal{N}_1 = \{2,4\}$ and $\mathcal{N}_2 = \{2,3,4\}$; Otherwise, we can set the groups with large values, e.g., $\mathcal{N}_3 = \{4,8\}$ and $\mathcal{N}_4 = \{4,6,8\}$. However, to compare with other state-of-the-art methods fairly, we fix $\mathcal{N} = \{4,6,8\}$ for our TSSN-Net in the experiment.

4.4. Batch size analysis

The batch size influences the performance of the network by the sensitivity of normalization. Recall that, BN is sensitive to the value of batch size. In contrast, IN and LN are not sensitive to the value of batch size but they will limit the generalization of our network. Thus, the second normalization step of TSSN Block adopts a switchable strategy to select a suitable normalizer. To analyze the impact of batch size on our network, we test TSSN-Net2 with different batch sizes on three sequences, i.e., NARROW-main, WIDE-science, and COLMAP-south, and we report the quantitative comparison results in Table 3. We can see that, TSSN increases from 95.54% to 99.02% in the term of F-measure in NARROW-main sequences when the batch size increases from 2 to 16, because of the increase in the normalization sample. However, when the batch size increases from 16 to 32, F-measure decreases from 99.02% to 95.31%. The reason is that the increasing batch size will lead to fewer parameter updates in a fixed number of epochs. Thus, we train our network with batch size as 16.

5. Discussion

Our network includes two main important parts, i.e., TSSN Block and MSCG. Recall that, TSSN-Net1 includes TSSN Block, and it gets the global context information from the first normalization step and adaptively selects normalizers for different convolution layers, while NM-Net includes CN Block. Thus, we can compare TSSN-Net1 and NM-Net to show the effectiveness. As shown in Table 2 and Fig. 6, the proposed TSSN block (CN+SSN) can significantly improve the performance of plain CN Block (CN + BN). Especially for Precision, TSSN-Net1 can get an improvement of 2% over NM-Net on all three benchmark dataset. Thus, the proposed TSSN Block is significantly effective for feature matching.

For MSCG, we can directly compare TSSN-Net1 and TSSN-Net2. From Table 2 and Fig. 6, we can see that, TSSN-Net2 obtains significant improvement over TSSN-Net1 on the COLMAP dataset which has the extremely lower inlier ratio (only contains average inlier ratio 7.55%). That is, TSSN-Net2 achieves the improvements of the F-measure, Precision, Recall, 5.37%, 4.44%, and 6.11%, respectively, over TSSN-Net1. Therefore, this can show the effectiveness of the proposed MSCG.

6. Conclusion

In this paper, we propose an end-to-end network (called TSSN-Net) to identify inliers and outliers for two-view feature matching problems. The proposed TSSN-Net includes two innovations, i.e., TSSN Block and MSCG. Specifically, the proposed TSSN Block can help to extract global information and adaptively select normalizers for different convolution layers, which can improve the performance on algorithm accuracy. Also, we propose MSCG to search for correspondence neighbors and integrate correspondence with different scale neighbors to a graph, which can capture local information and improve the generalization ability. The experimental results have shown that TSSN-Net can significantly improve the match performance of accuracy over the state-of-the-art methods, on publicly available datasets, especially for datasets with heavy outliers.

CRediT authorship contribution statement

Zhen Zhong: Conceptualization, Methodology, Software, Writing - original draft. **Guobao Xiao:** Methodology, Writing - review & editing, Supervision. **Kun Zeng:** Visualization, Investigation. **Shiping Wang:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 62072223, by the Natural Science Foundation of Fujian Province under Grants 2020J01829, and by the Project of Central Leading Local under Grants 2020L3024.

References

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

[2] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K.M. Yi, E. Trulls, Image matching across wide baselines: From paper to practice, International Journal of Computer Vision (2020) 1–31.

[3] X. Jiang, J. Ma, G. Xiao, Z. Shao, X. Guo, A review of multimodal image matching: Methods and applications, Information Fusion 73 (2021) 22–71.

[4] C. Wu, Towards linear-time incremental structure from motion, in: 3DV, 2013, pp. 127–134..

[5] J.L. Schonberger, J.-M. Frahm, Structure-from-motion revisited, in: CVPR, 2016, pp. 4104–4113.

[6] R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, Orb-slam: a versatile and accurate monocular slam system, IEEE Transactions on Robotics 31 (5) (2015) 1147–1163.

[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J.J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, IEEE Transactions on Robotics 32 (6) (2016) 1309–1332.

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

[9] G. Xiao, H. Wang, Y. Yan, D. Suter, Superpixel-guided two-view deterministic geometric model fitting, International Journal of Computer Vision 127 (4) (2019) 323–339.

[10] G. Xiao, H. Wang, Y. Yan, L. Zhang, Robust geometric model fitting based on iterative hypergraph construction and partition, Neurocomputing 336 (2019) 56–66.

[11] F. Kluger, E. Brachmann, H. Ackermann, C. Rother, M.Y. Yang, B. Rosenhahn, Consac: Robust multi-model fitting by conditional sample consensus, in: CVPR, 2020, pp. 4634–4643.

[12] G. Xiao, H. Luo, K. Zeng, L. Wei, J. Ma, Robust feature matching for remote sensing image registration via guided hyperplane fitting, IEEE Transactions on Geoscience and Remote Sensing..

[13] C. Zhao, Z. Cao, C. Li, X. Li, J. Yang, Nm-net: Mining reliable neighbors for robust feature correspondences, in: CVPR, 2019, pp. 215–224.

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

[15] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, International Journal of Computer Vision 60 (1) (2004) 63–86.

[16] M.A. Fischler, R.C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications of The ACM 24 (6) (1981) 381–395.

[17] D. Zeng, T. Zhang, R. Fang, W. Shen, Q. Tian, Neighborhood geometry based feature matching for geostationary satellite remote sensing image, Neurocomputing 236 (2017) 65–72.

[18] S.M.M. Kahaki, M.J. Nordin, A.H. Ashtari, S.J. Zahra, Deformation invariant image matching based on dissimilarity of spatial features, Neurocomputing 175 (2016) 1009–1018.

[19] J. Ma, J. Zhao, J. Jiang, H. Zhou, X. Guo, Locality preserving matching, International Journal of Computer Vision 127 (5) (2019) 512–531.

[20] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, P. Fua, Learning to find good correspondences, in: CVPR, 2018, pp. 2666–2674.

[21] D. Barath, J. Noskova, M. Ivashechkin, J. Matas, Magsac++, a fast, reliable and accurate robust estimator, in: CVPR, 2020, pp. 1304–1312.

[22] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, K.M. Yi, Acne: Attentive context normalization for robust permutation-equivariant learning, in: CVPR, 2020, pp. 11286–11295.

[23] J. Ma, X. Jiang, J. Jiang, J. Zhao, X. Guo, Lmr: Learning a two-class classifier for mismatch removal, Transactions on Image Processing..

[24] W. Shao, T. Meng, J. Li, R. Zhang, Y. Li, X. Wang, P. Luo, Ssn: Learning sparse switchable normalization via sparsestmax, in: CVPR, 2019, pp. 443–451.

[25] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: ICML, 2015, pp. 448–456..

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

[27] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450..

[28] O. Chum, J. Matas, Matching with prosac-progressive sample consensus, in: CVPR, 2005, pp. 220–226.

[29] T. Sattler, B. Leibe, L. Kobbelt, Scramsac: Improving ransac's efficiency with a spatial consistency filter, in: ICCV, 2009, pp. 2090–2097..

[30] R. Raguram, O. Chum, M. Pollefeys, J. Matas, J.-M. Frahm, Usac: A universal framework for random sample consensus, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (35) (2013) 2022–2038.

[31] J. Ma, J. Jiang, Y. Gao, J. Chen, C. Liu, Robust image matching via feature guided gaussian mixture model, in: ICME, 2016, pp. 1–6.

[32] J. Ma, J. Jiang, H. Zhou, J. Zhao, X. Guo, Guided locality preserving feature matching for remote sensing image registration, IEEE Transactions on Geoscience and Remote Sensing 56 (8) (2018) 4435–4447.

[33] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, M.-M. Cheng, Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence, in: CVPR, 2017, pp. 4181–4190.

[34] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: ICCV, 2005, pp. 1482–1489.

[35] A. Bhowmik, S. Gumhold, C. Rother, E. Brachmann, Reinforced feature points: Optimizing feature detection and description for a high-level task, in: CVPR, 2020, pp. 4948–4957.

[36] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, L. Quan, Aslfeat: Learning local features of accurate shape and localization, in: CVPR, 2020, pp. 6589–6598.

[37] G. Georgakis, S. Karanam, Z. Wu, J. Ernst, J. Košecká, End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching, in: CVPR, 2018, pp. 1965–1973.

[38] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, L. Quan, Contextdesc: Local descriptor augmentation with cross-modality context, in: CVPR, 2019, pp. 2527–2536.

[39] 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: ICCV, 2019, pp. 5845–5854..

[40] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: CVPR, 2017, pp. 652–660.

[41] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: NIPS, 2017, pp. 5099–5108.

[42] J.L. Schonberger, H. Hardmeier, T. Sattler, M. Pollefeys, Comparative evaluation of hand-crafted and learned local features, in: CVPR, 2017, pp. 1482–1491.

[43] Y. Tian, B. Fan, F. Wu, L2-net: Deep learning of discriminative patch descriptor in euclidean space, in: CVPR, 2017, pp. 661–669.

[44] P. Luo, J. Ren, Z. Peng, R. Zhang, J. Li, Differentiable learning-to-normalize via switchable normalization, arXiv preprint arXiv:1806.10779..

[45] J.L. Schonberger, J.-M. Frahm, Structure-from-motion revisited, in: CVPR, 2016, pp. 4104–4113.

Zhen Zhong received the bachelor's degree in traditional Chinese medicine from the Hunan University of Chinese Medicine, Hunan, China, in 2018. He is currently pursuing the M.S. degree with the Department of Traditional Chinese Medicine, Fujian University of Traditional Chinese Medicine, where he is also attached to the Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University. His research interests include computer vision, machine learning, and pattern recognition.

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 Minjiang University, China. He has published over 30 papers in the international journals and conferences including IEEE TPAMI/TIP/TITS/TIE, IJCV, PR, ICCV, ECCV, ACCV, AAAI, etc. His research interests include machine learning, computer vision and pattern recognition. He has been awarded the best PhD thesis in Fujian Province and 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, AAAI, WACV, etc. He was the General Chair for IEEE BDCLOUD 2019.

Kun Zeng received Ph. D degrees from Department of Computer Science, Xiamen University, Xiamen, China, in 2015, where he was a Postdoctoral Fellow with the Department of Electronic Science, from 2016 to 2019. He is currently a Lecturer with Minjiang University, China. His current research interests include image processing, machine learning and medical image reconstruction.

Shiping Wang received the Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2014.

He was a Research Fellow at Nanyang Technological University, Singapore, from 2015 to 2016. He is currently a Full Professor and Qishan Scholar with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His research interests include machine learning, computer vision, and granular computing.