IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 September 2023, accepted 22 September 2023, date of publication 29 September 2023,
date of current version 4 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3320682

TOPICAL REVIEW

A Survey on the Deep Learning-Based Mismatch Removal: Principles and Methods

SHIYU CHEN[1], CAILONG DENG[2,3], YONG ZHANG[4,5], YONG WANG[2,3], QIXIN ZHANG[1], AND ZHIMIN ZHOU[2,3]

[1]School of Geographic Sciences, Xinyang Normal University, Xinyang 464000, China
[2]Sichuan Institute of Land Science and Technology (Sichuan Center of Satellite Application Technology), Chengdu, Sichuan 610041, China
[3]Key Laboratory of Investigation, Monitoring, Protection and Utilization for Cultivated Land Resources, MNR, Chengdu 610041, China
[4]Visiontek Inc., Wuhan 430205, China
[5]School of Electronics and Information Engineering, Wuzhou University, Wuzhou 543003, China

Corresponding author: Shiyu Chen (csy_hy@xynu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 41901402, in part by the Nanhu Scholars Program for Young Scholars of XYNU under Grant 2017037, in part by the Science and Technology Project from the Department of Natural Resources of Sichuan Province: Research on Knowledge-Based Service of the Investigation and Monitoring Results of Non-Grain Cultivated Land under Grant 2023JDKY0017-3, and in part by the Research on Monitoring Technology of Internal Land Class Changes of Agricultural Land and Cultivated Land Planting Attributes in Sichuan Province under Grant RMTL2023011.

ABSTRACT Due to the inherent limitations of matching algorithms and the complexities associated with image contents, mismatches are inevitable and can have detrimental effects on downstream tasks in computer vision and remote sensing. Researchers have published numerous reviews on mismatch removal, which may suffer from two primary deficiencies. Firstly, these reviews are often embedded within studies that primarily focus on image matching, thereby limiting the detailed and comprehensive analysis of mismatch removal methods. Secondly, reviews of deep learning (DL)-based methods, despite their numerous existence and interconnection, tend to be fragmentary and lack a systematic approach. To address these two shortcomings, this paper presents a comprehensive survey of DL-based mismatch removal principles and methods. We provide a summary of network architectures, techniques for extracting geometrical information, and various training modes. Specifically, we highlight the importance of permutation invariance in mining operations, enumerate a majority of existing mining methods, and provide an explanation of their permutation invariant properties. Furthermore, we present both the intuitive motivation and mathematical analysis of commonly used methods, elucidating their underlying principles and efficacy. In the conclusion, we predict upcoming trends based on the findings of our review, aiming to provide valuable insights into mismatch removal techniques and guide their practical applications.

INDEX TERMS Image matching, mismatch removal, deep learning, geometrical information mining, permutation invariant.

I. INTRODUCTION

Image matching lies the core of fundamental computer and remote sensing vision tasks [1], [2], such as instance retrieval [3], [4], 3D scene reconstruction [5], [6], automatic positioning of sensors [7], [8], and image classification [9], [10]. Due to inevitable radiometric and geometric distortions between images, considerable mismatches are mixed with correct matches which will impair the subsequent applications. Thus, a mismatch removal process is necessary to retain as more correct matches while maintain high matching precision. The key challenge in solving the mismatch removal problem lies in modeling the geometric invariance among correct matches. However, two major challenges still exist: firstly, the invariance cannot be accurately represented by a single model, as is the case in nonrigid matching [11], [12] and multi-model image matching [13]; Secondly, there is a lack of robust estimators that can accurately estimate the model without being influenced by mismatches.

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

Researchers have made extensive efforts to address the first challenge by accurately describing the local or global geometric transformations between image pairs. Global transformations are commonly used when image pairs exhibit rigid geometric relationships. For instance, the fundamental or essential matrix is suitable for image pairs captured by pin-hole cameras [14], while the homography matrix is appropriate for images captured by satellite cameras [15], [16], [17]. On the other hand, local transformations provide a more generalized approach and can describe complex geometric relationships between image pairs. These complex relationships can arise due to various factors, including non-rigid transformations [18], [19], [20], [21], [22], [23], [24], [25], [26], occlusion and repetitive patterns [27], [28], [29], [30], [31], and sensor distortions [32], [33], [34], [35], [36], [37]. Both global and local transformations offer unique advantages in handling image geometry. Global transformations can be easily and accurately estimated, resulting in efficient algorithms. However, their application is often limited to specific scenarios. In contrast, local transformations can be applied to arbitrary image pairs. The local transformations are often estimated by non-rigid transformation estimators. For example, Grid-based Motion Statistics (GMS) [25] proposes a real-time and robust estimator by encapsulating motion smoothness as statistical likelihood of a certain number of neighboring matches; Locality Preserving Matching (LPM) [11] proposes an algorithm of linear time and space complexity by mining neighboring true matches; Locality Affine-invariant Matching (LAM) [38] is also an linear time and space complexity algorithm, while the estimator is implemented a local barycentric coordinate and matching coordinate matrices. These non-rigid transformation estimators are still prevalent because of their effectiveness and efficiency, and they have great impacts on the designs of neural networks of mismatch removal. While their calculating processes can be progressive and involve numerous user-defined parameters that may not be easily tuned.

To tackle the second challenge, various robust estimators have been introduced. In the early years, techniques such as data-snooping [39] and iteratively reweighted least squares (IRLS) [40] were employed to estimate models with low outlier rates in matching. Since its introduction in 1981, RANdom Sample Consensus (RANSAC) [41] and its variants have remained prevalent for over four decades [41]. Generally, these resample-based methods follow a process of sampling, hypothesis formation, and verification. Firstly, samples are drawn from a probability distribution established under mild assumptions [41], [43], [44] or estimated based on spatial coherence or radiometric similarity [45], [46], [47], [48], [49], [50], [51]; Subsequently, geometric models (rigid or non-rigid) are estimated during the hypothesis formation stage. Finally, the estimated models are verified using putative matches, and the optimal model is selected based on the maximum consensus set. Resample-based methods, such as RANSAC and its variants, are typically more robust to higher outlier rates compared to data-snooping and IRLS. For example, the most recent Quadratic-time Guaranteed Outlier Removal (QGORE) [52] certifies the geometric consistency via resampling processes for upper bound estimation, and it is claimed that the method is efficient and can cope with outlier rate higher than 95%, while the algorithm is designed specially for point cloud registration.

In addition, non-resample-based robust estimators can also be immune to high outlier rate. For example, Adaptive M-estimators (AM-estimators) [53] uses shape-control parameters to replace the original constant parameter in the weight function, and it applies a coarse-to-fine process to purify the matches, experiments show that AM-estimators can deal with matches with outlier rate as high as 80%; Scaled Welsch q-Norm [54] likewise uses a coarse-to-fine process but changes the weight function by decreasing its scale parameter, the proposed method is still robust even if the outlier rate is up to 90%. However, non-resample-based robust estimators are analogous to the non-rigid transformation estimators, they both have many scenario-specified parameters which are not easily tuned.

Deep learning (DL)-based methods have the ability to address both challenges (comparisons of representative methods are shown in Figure 1). The problem of mismatch removal can be seen as a binary classification problem that has been extensively studied and successfully addressed using DL. DL-based methods employ deep neural networks (DNNs) to generate a matching probability or weight that signifies whether a match is an outlier. Theoretically, an appropriate DNN can accurately approximate the geometric model among inliers [55], [56]; moreover, the generated weights can be leveraged to further refine the geometric model; additionally, the DNN is trained iteratively to enhance prediction precision and model accuracy. Thus, given sufficient training data, DL-based methods usually outperform traditional handcrafted methods. Nonetheless, when employing DNNs for mismatch removal, three issues need to be resolved.

Firstly, DNNs applied in image classification cannot be directly migrated to the classification of matches since inputted training data in these two tasks have noteworthy differences. Images are composed of pixels arranged in a specific order, and altering the order of pixels generates new images. While matches can be viewed as point clouds composed of unordered pairs of 2D coordinates. Despite the disordering of these coordinate pairs, they remain the same point clouds. Therefore, changes in the order of matches do not affect the features of the matches (note, the features will be used to separate inliers from outliers). Mathematically, if a DNN g follows the formula:

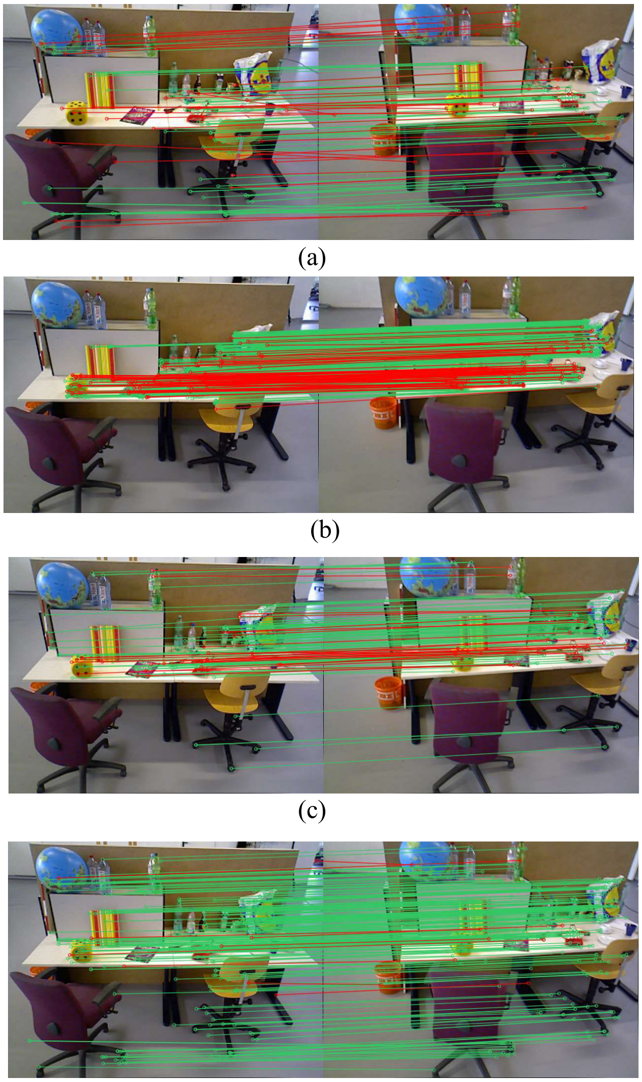
$$\mathrm{g}(P \times F) = \mathrm{g}(F) \tag{1}$$

where $F \in R^{N\times C}$ is a feature map, its every row vector is a feature vector of a match which is embedded with geometrical information, and $P \in \{0, 1\}^{N\times N}$ is a permutation matrix, then the network is called a permutation-invariant network (PIN) [57]. To ensure permutation-invariant, some operations typically employed in processing images, e.g., convolutions with kernel size greater than one, cannot be utilized to extract geometrical information of matches. Alternatively, numerous permutation-invariant operations, such as points-wise multilayer perceptron (point-wise MLP [57], i.e., convolution with kernel size equal to one), pooling [58], k-nearest neighbor (KNN) [59], [60] aggregation, normalization [61], attention [62], [63], as well as some operations based on graphs (e.g., graph convolution [64], graph pooling [65], and graph attention [66]) are widely employed to extract local-global information, thereby aiding in the removal of mismatches.

Secondly, PINs should integrate the local with global geometric information to enhance the separability of matches; meanwhile, the information should be specifically gathered from matching inliers while excluding any contamination from matching outliers. Local information gathered from neighboring matches can be effectively modeled, as local geometric variations between images exhibit smoothness. Global information is generally constructed based on local information, which is concatenated with local information to enhance the distinguishability. The main concern is how to aggregate information from matching inliers while neglect noise from outliers. Currently, attention [62], [63] and weighting are widely used solutions due to their simplicity and interpretability.

Thirdly, most PINs are trained in a supervised manner, remanding matching labels or ground truth geometrical constraints, such as epipolar geometrical (EG) constraints or homography transformation between matches. Whereas, labeling data is cumbersome and time-consuming, and inevitable wrong labels may pose negative effects on the performances of the PINs [67]. Therefore, an unsupervised learning approach is crucial and urgent in order to improve both training efficiency and the generalization of PINs. Currently, most unsupervised methods are built upon the assumption that the optimal geometrical model is the one with the maximal consensus set. That is, the maximal cluster of matches in high dimensional space form an inlier set under the constraint of the correct geometrical model. Thus, to construct an unsupervised learning framework, the first step is applying PINs to project coordinates of matches to high dimensional space (i.e., extracting features of matches); then modulating these features to output matching probabilities (weights); finally, utilizing resampling methods [68] or weighted regression methods [69] to obtain a maximal consensus matching set.

Since the first DL-based mismatch removal paper, "Learning to Find Good Correspondences" (LFGC) [70] was introduced in 2018, extensive research has been conducted to address and resolve the aforementioned issues. However, to our knowledge, no existing paper thoroughly summarizes the similarities and differences between these algorithms, the theoretical foundations behind their operations, and provides systematic classifications of the proposed methods. In this regard, we provide a comprehensive survey of the DL-based methods, explain the permutation-invariance of operations, and analyze the effectiveness of mathematical models. The paper is organized as follows. Section II formulates the mismatch removal problem, Section III gives the commonly-used network architectures, Section IV presents the permutation-invariant operations, Section V shows how to build a supervised or an unsupervised training framework, and Section VI concludes representative DL-based methods (shown in Table 1) and points out the future trends.


(a)

(b)

(c)

(d)

FIGURE 1. Exampled representative methods, remaining correct and false matches are linked by green and red lines, respectively. (a) RANSAC [40], remaining 73 correct and 33 false, (b) GMS [25], remaining 231 correct and 85 false, (c) LFGC [70], remaining 125 correct and 30 false, and (d) ULMR [68], remaining 258 correct and 9 false.

II. PROBLEM FORMULATION

Given an image pair (I, I'), a putative matching set $\mathcal{X} = \{x_i\}_{i=1}^{N}$ can be extracted by handcrafted methods (e.g., SIFT [71]) or learning-based methods (e.g., SuperGlue [72]), where N is the element number of the matching set, $x_i = (u_i, v_i, u'_i, v'_i)$ is a match, (u_i, v_i) and (u'_i, v'_i) are the 2D coordinate of two matched keypoints in image I and I', respectively. The match x_i representing by a 4D coordinate are sometimes normalized by the image size [72] or camera intrinsic parameters [70] to alleviate the negative effect of scale changes:

$$(u, v) = \left((u^r - a_1)\big/a_2, (v^r - b_1)\big/b_2\right) \tag{2}$$

where (u, v) is the normalized 2D coordinate, (u^r, v^r) is the original pixel coordinate; if the camera intrinsic parameters are given, $a_2 = f_x$ and $b_2 = f_y$ are the focal lengths in x and y direction, respectively; $(a_1, b_1) = (c_x, c_y)$ is the principal point offset; if the camera intrinsic parameters are not available, $(a_2, b_2) = (l_1/2, l_2/2)$ where l_1 and l_2 are the image width and height, $a_1 = b_1 = max(l_1, l_2)$.

As stated in [1], mismatch removal is a process of removing false matches from putative matching sets by using extra geometric constraint. Specifically, from the perspective of DL, this process can be formally described as:

$$w = g_{\omega}(X), \quad E = \varphi(w, X) \tag{3}$$

where w is a vector of weights and w_i is the matching probability of match x_i, X is the matrix representation of the set $\mathcal{X}$, $\varphi(\cdot, \cdot)$ is a weighted regression model and E is the regressed geometrical model between matches, $g_{\omega}(\cdot)$ is a PIN with a learnable parameter ω. Note, some radiometric information, such as nearest neighboring distance ratios (NNDR) [71] generated from descriptor matching, can be included in x_i, while Equation (3) does not loss the generality since increasing the depths of the first convolutional layer of g can still accommodate these types of data.

The objective of DL-based methods is to design a PIN and learn its optimal parameters. The optimal PIN assigns higher weights for matching inliers and lower weights for outliers (ideally, 0 for outliers and 1 for inliers). Consequently, though mixed with considerable mismatches, correct matches are retained as more as possible and matching precision are also maintained.

III. NETWORK ARCHITECTURE

The network architectures of PINs for mismatch removal mainly contain two types of blocks: (1) information mining blocks (IMBs), aiming to mine geometrical information to build features of matches; (2) weight generating blocks (WGBs), aiming to modulate these features to output matching probabilities. Furthermore, IMBs are constructed by a series of permutation-invariant operations, and WGBs are typically located at the end of DL networks and use activation functions to normalize the upstream outputted digits to a range of 0 to 1.

Many activation functions, such as Sigmoid function [73], [74] and binary step function, have the ability of normalizing a digit number to 0 to 1, and debates regarding the performances of activation functions are still ongoing [75]. While in mismatch, the choice of activation functions somewhat have clues: if there needs to regress a geometric model between matches (e.g., E in Equation (2)), then the activation function could be the combination of rectified linear unit (ReLU) [76] and hyperbolic tangent (Tanh) function [77]; if the regression is not needed, Sigmoid and Softmax function are suitable alternatives. This is because outliers can be perceived as chaotic noises that cannot contribute to regular and powerful positive signals or features. The combination of the activation functions effectively suppresses weak features that are generated from outliers, and it can be illustrated in Figure 2. Comparing to Sigmoid function, the combination functions yield greater response if the signal intensity is greater than 6.000, and they exhibit weaker responses if the input is smaller than 0.881.

WGBs and IMBs are combined to form a linear structure and a "T" structure. The specific details of these two structures will be presented in the following sections.

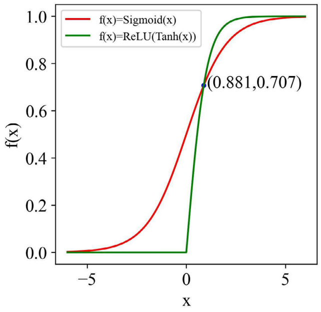


FIGURE 2. Plots of f(x) = Sigmoid(x) and f(x) = ReLU(Tanh(x)).

A. LINEAR STRUCTURE

The linear structure is constructed by a cascade of IMBs, and a residual connection [78] may be included between two adjacent IMBs to avoid the vanishing gradient problem [79]. At the end of the linear structure, a WGB is used to produce the weights (as shown in Figure 3).

The majority of existing PINs are linear, with LFGC [70] is of the typical. The IMBs of LFGC is consist of 12 Resnet blocks (the block is also called as PointCN, where CN is short for context normalization and it will be detailed in the following), and WGB is at the end of the network constructed by the combination of ReLU and Tanh. Attentive Context Normalization Network (ACNe) [80] uses a weighted CN on the consideration of CN may be afflicted by outliers; to compensate for the missing local information of CN, Neighbor Mining Network (NM-Net) [81] employs K nearest neighbor (KNN) searching to aggregate local information; Order-Aware Network (OA-Net) [82] utilizes pooling algorithm to integrate local information. Henceforth, CN, KNN mining, and pooling are the standard operations for constructing a linear structure.

The IMBs in the linear structures can be organized in two modes [83]: one-shot and progressive. The former operates

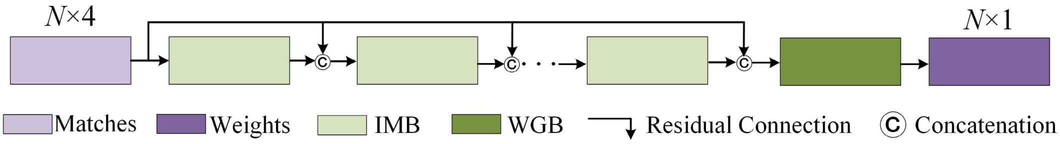


FIGURE 3. Linear structure of PIN architecture.

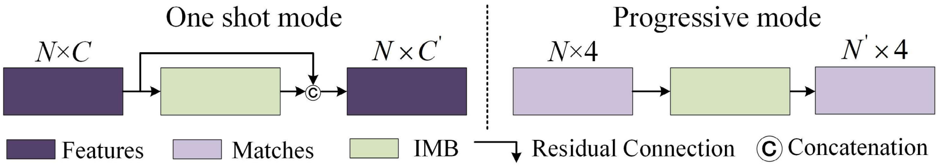


FIGURE 4. Organization modes of IMBs.

with features and fine-tuned features as input and output, while for the latter, the input and output are correspondences and pruned correspondences, respectively (schematic illustrations of the two modes are shown in Figure 4). For instance, in order to address the imbalance between inliers and outliers, Guided Loss and Hybrid Attention (GLHA) [84] constructs a coarse-to-fine cascade network by using attention mechanism; Consensus Learning Network (CLNet) [83] learns a network by progressively pruning the correspondences; Interactive Generative Structure Network (IGS-Net) [85] captures the coarse-to-fine transformations of matches through progressive representation learning. In summary, for the purpose of designing a deeper network, most of the IMBs are organized in the one-shot mode. While input and output features in the progressive mode have different feature sizes which lead to networks cannot be skip-connected, therefore progressive- mode networks cannot go deeper.

B. "T" STRUCTURE

Drawing inspiration from the "squeeze-and-extraction" operation [86], [87], "T" structure is proposed aiming to extract channel-wise geometric information. It has two mainstreams: a "–" mainstream and a "|" mainstream. The "–" stream is organized as the linear structure, and the "|" stream receives the outputs of every IMB in the "–" stream and finally outputs the features of matches (the schematic illustration of the "T" structure is shown in Figure 5).

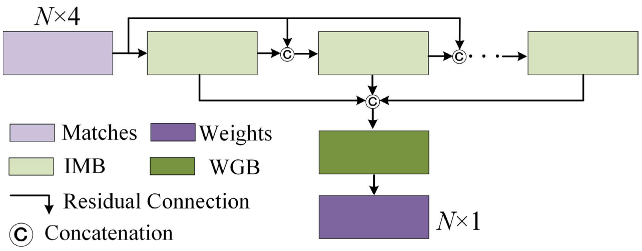


FIGURE 5. "T" structure of PIN architecture.

"T" structure network was firstly introduced by T-Net [84]. It adopts "–" stream to iteratively learn features of matches and another "|" stream to integrate the features and generate matching probabilities. "T" structure has enlightened the design of sub-networks. For example, Point2CN [89] leverages PointCN [70] to extract the hierarchical features from subsets of feature maps and fuses the learned results by weighted addition; similarly, Permutation-Equivariant Split Attention Network (PESA-Net) [90] begins by splitting features into paths and learning multiple geometric information by MLPs, it then applies a union operation to aggregate information and ultimately generates the matching probabilities; additionally, Preference-Guided Filtering Network (PGFNet) [91] employs the same splitting operation as PESA-Net and introduces a grouped residual attention mechanism to recalibrate and fuse the input features.

C. COMPARISON OF NETWORK STRUCTURES

Compared to the linear structure, "T" structure theoretically has two main merits. Firstly, networks of the linear structure cannot go deeper although residual connections are often used. In contrast, the "T" structure has a network depth that is approximately half of the linear structure, theoretically enabling the networks to go greater depths. Secondly, in the linear structure, as shown in Figure 3, only the outputs of the last IMB are used for predicating the matching probabilities, resulting in the loss of valuable information generated by the previous IMBs [88], while for "T" structure, all the outputted results are treated equally to predict the probabilities and thus can retain as much information as possible. While it is a common sense that network should be deeper, thus most mismatch removal networks are linear, and "T" structure networks are sometimes function as Resnet aiming to avoid vanishing gradient problem.

IV. GEOMETRICAL INFORMATION MINING

To construct stable and separable features of matches, extensive algorithms are proposed to construct IMBs trying to mining more geometric information while keeping IMBs invariant to permutation of matches. There exist some operations making up the foundational basis of IMBs, to sum up, the foundation operations consist of point-wise MLP (Section IV-A), normalization (Section IV-B), attention mechanism (Section IV-C), and some operations on the graph (Section IV-D). This section will provide formal descriptions of the aforementioned operations along with proofs or explanations to demonstrate their invariance. Furthermore, some technical details and variants of these operations are also presented.

A. POINT-WISE MLP

Point-wise MLP [57] is a prerequisite operation for projecting coordinates of correspondences, and it also a common operation for altering the dimensions of learned features. To simplify and without losing generality, let's consider a feature f_i and assume the MLP is a one-layer network without activation, then point-wise MLP can be written as:

$$\mathrm{MLP}(f_i) = W \times f_i \tag{4}$$

where W is a learned projecting matrix. The function of the point-wise MLP, as demonstrated in Equation (4), is to project the feature f_i to other space. If f_i is a match, it is a 4D coordinate and may not be separable due to its low dimensional [92]. By using point-wise MLP, the 4D point is re-projected to a high dimensional space in which the separability of the match will be increased [92]; if f_i is a feature, the point-wise MLP is utilized to learn distinctive features in different dimensional spaces, these features are subsequently aggregated by the downstream IMBs to facilitate the separation of inliers from outliers.

Since point-wise MLP is operating on a signal feature and the learnable parameters are shared (as shown in Figure 6), permutation of a feature is itself. Therefore, the point-wise MLP is permutation invariant.

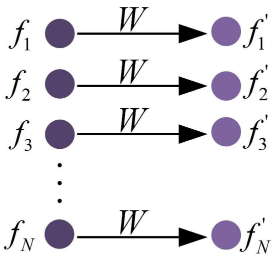


FIGURE 6. Schematic illustration of point-wise MLP. *W* is the shared learnable parameters, f_i is the input feature, and f_i' is the corresponding learned feature.

B. NORMALIZATION

Commonly used normalizations [61] in DL, such as batch normalization (BN) [93], instance normalization (IN) [94], layer normalization (LN) [95], and et. al., can benefit for the performances of PINs if they are appropriately used. Additionally, some specialized normalization methods exist for PINs that aim to improve the separability, such as CN [70] and Attentive Context Normalization (ACN) [80].

1) CONTEXT NORMALIZATION

CN, Similar to MLP, is another standard operation used in the construction of a PIN. Given a set of features $\mathcal{F} = \{f_i\}_1^N$, CN normalize the features by the mean and standard deviation:

$$\mathrm{CN}(f_i) = \frac{f_i - \mu}{\sigma},$$
$$\mu = \frac{1}{N}\sum_{i=1}^{N} f_i, \quad \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - \mu)^2} \tag{5}$$

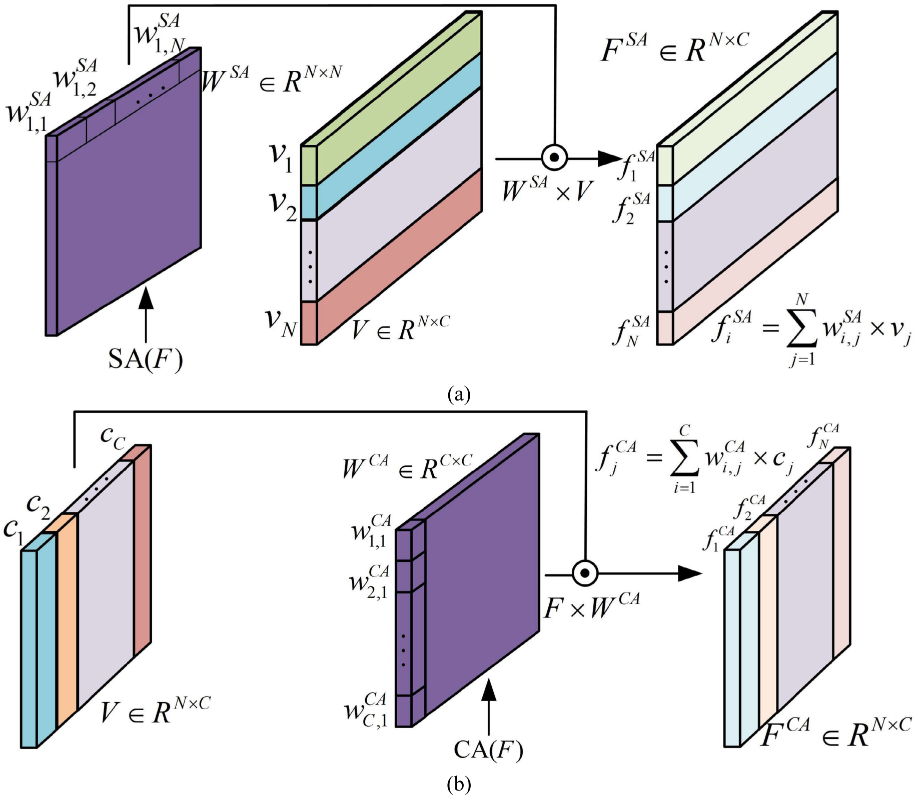
where μ is the mean and σ is the standard deviation. Since the mean and standard deviation are invariant to permutation of features, and CN is operating on a single feature (i.e., CN is point-wise). Therefore, CN is permutation invariant. It is important to note that although the mathematical form of CN is very similar to BN, the two have distinct meanings. In BN, $\mathcal{F}$ represents a batch of features, and N is the batch size which can be various; While for CN, $\mathcal{F}$ denotes a feature set generated from putative matches, N is the feature number and is fixed if matches are established.

The effectiveness of CN is very similar to that of data-snooping [38] which is referred as an ancient outlier removal method. Given l_i is a measurement of l, then N measurements can generate a set of residual errors, each consisting of N elements. In Ordinary Least Square (OLS) [96], the set of residual errors can be denoted as $\mathcal{R} = \{r_i\}_1^N = \{l_i - \mu_l\}_1^N$ where μ_l is the exception of l. Additionally, we can calculate the normalized residual error of r_i: $\bar{r}_i = r_i / \sigma_{r_i} = (l_i - \mu_l) / \sigma_{r_i}$, where σ_{r_i} is the standard deviation of r_i. It has been proven that if the measurements do not contain outliers, then $\bar{r}_i$ admits a standard normal distribution [39]. If the absolute value of calculated $\bar{r}_i$ is significantly large, it indicates a deviation from the assumption of a standard normal distribution. Consequently, the corresponding measurement l_i may contain gross error and can be classified as an outlier. In mismatch removal networks, CN generally functions as a denoiser that enhances the responses of inliers and suppresses the responses of outliers.

2) ATTENTIVE CONTEXT NORMALIZATION

Equation (5) demonstrates that the mean and standard deviation can be obtained through the calculation of the first and second moments of the features. This holds true in the context of OLS, where measurements only contain random errors. However, when a significant number of outliers are present, the estimations of mean and standard deviation become imprecise, resulting in negative impacts on the performance of CN. To address this issue, ACN [80] is proposed:

$$\mathrm{ACN}(f_i) = \frac{f_i - \mu_w}{\sigma_w},$$



(a)

(b)

FIGURE 7. Illustration of the two attention modes. (a) SA, and (b) CA.

$$\mu_w = \sum_{i=1}^{N} w_i \times f_i,$$

$$\sigma_w = \sqrt{\sum_{i=1}^{N} w_i(f_i - \mu_w)^2}, \quad w = g_\omega(F) \tag{6}$$

where σ_w and μ_w are the weighted mean and standard deviation, $w \in R^N$ is the learned weights ranging from 0 to 1 and w_i is the weight corresponding to feature f_i, g is a neural network. ACN is permutation invariant since g is a PIN.

If g is capable of learning appropriate weights for the features, then it will be unaffected by outliers and can make precise estimations of the mean and standard deviation. Consequently, the performance of CN will be improved. While in shallow networks, the weights cannot be effectively learned due to the less distinguishable output features of the networks. To address this limitation, the improved version of ACN, Bayesian Attentive Context Normalization (BACN) [84], was introduced. Whereas, BACN employs NNDR [71] to compensate for the dilemma of weight learning in shallow networks, which are not available if matches are not generated from feature matching. As CN is generally followed by other normalizations which will cause disturbing information from other image pairs [97], Two-step Sparse Switchable Normalization (TSSN-Net) [98] learns a switchable normalizer among BN, IN, and LN, and therefore avoids the disturbance of the defective information. In summary, normalization methods can be served as denoisers that enhance signals from inliers and suppress that from outlier, therefore they are commonly used as a preprocessing step to filter features of matches.

C. ATTENTION

Attention is one of the widely employed techniques in the field of natural language processing (NLP) [62], it is firstly introduced to eschew recurrent networks (such as long short-term memory [99], and gated recurrent [100]). Subsequently, attention is introduced to the field of computer vision [101], and later, it gains prominence in mismatch removal for its effectiveness and ease of implementation. Formally, attention defines a function on a feature set $\mathcal{F}$:

$$W = \text{ATN}(F) \tag{7}$$

where $F \in R^{N\times C}$ is the matrix representation of $\mathcal{F}$, ATN(F) is the attention from F, W is the attention map. There are massive variants of attentions, and these variants can be classified into two primary types: spatial attention (SA) [62] and channel attention (CA) [86], [102], the former generates weights from the spatial compatibility and reweights features, the latter generates weights from feature channels and recalibrates feature channels (the two modes are depicted in Figure 7).

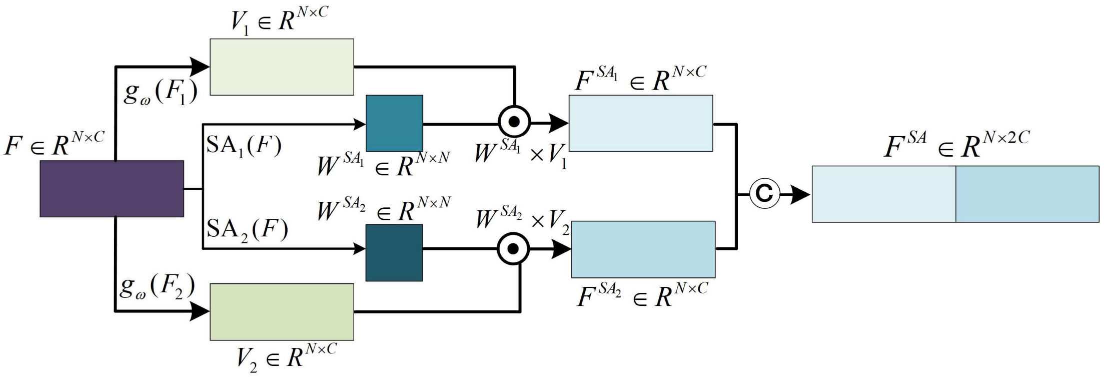


FIGURE 8. Illustration of two-head attention. SA_1 and SA_2 are the two-head attentions which produce the weights W^{SA_1} and W^{SA_2}, then weighting on the V_1 and V_2 respectively to generate two features F^{SA_1} and F^{SA_2}, and concatenating the two to output the final features.

1) SPATIAL ATTENTION

SA leverages spatially (geometrically) related information among features and assigns weights to these features based on their spatial compatibility. Commonly, SA has the following form:

$$W^{SA} = \mathrm{SA}(F) = \mathrm{Softmax}(Q \times K^T) \tag{8}$$

where $K = g_{\omega_1}(F) \in R^{N\times C}$ and $Q = g_{\omega_2}(F) \in R^{N\times C}$ are referred to as key and query, respectively, C is the number of channels, $g_{\omega_1}(\cdot)$ and $g_{\omega_1}(\cdot)$ are PINs with learnable parameters ω_1 and ω_2, $W^{SA} \in R^{N\times N}$ with its elements ranging from 0 to 1 is the weight generated by SA. W^{SA} is generally called self-attention since it is essentially originated from the feature set $\mathcal{F}$ and itself. By using W^{SA}, the features can be reconstructed:

$$F^{SA} = W^{SA} \times V \tag{9}$$

where $V = g_{\omega_3}(F) \in R^{N\times C}$ is called value.

Equation (8) can be rewritten as an element-wise form:

$$w^{SA}_{i,j} = \mathrm{Softmax}_i(q_i \times k^T_j) \tag{10}$$

where $w^{SA}_{i,j}$ is the element of W^{SA} in i-th row j-th column, q_i and k_j are the i-th and j-th row vector of Q and K, respectively. Since K and Q are the outputs of PINs, they can be regarded as the features of F in new spaces. In addition, the dot product represents the spatial correlation between two vectors. Thus, Equation (10) provides the spatial compatibilities of two feature vectors. As $w^{SA}_{i,j}$ is generated from feature pairs and it remains unchanged when the order of features is exchanged, thus SA is permutation invariant.

By using Equation (10), Equation (9) can be rewritten as:

$$f^{SA}_i = \sum_{j=1}^{N} w^{SA}_{i,j} \times v_j \tag{11}$$

where v_j is the j-th feature vector in V, and $f^{SA}_i \in R^{1\times C}$ the new generated i-th feature in F^{SA}. Equation (11) demonstrates that the newly generated feature is a weighted summation of multiple features: if the j-th feature is highly correlated with i-th feature, then j-th feature will contribute more in generating f^{SA}_i. As $w^{SA}_{i,j}$ represents the spatial consistency, f^{SA}_i is essentially embedded with the motion coherence of features [103].

SA can be extended to multiple heads [62]. Multiple head attentions can be constructed by multiple projections of F, and the final features are produced by concatenating the generated features of each head (an example is shown in Figure 8). Multi-head attention is analogous to multi-filter used in Convolutional Neural Networks (CNNs), this technique can enhance the separability of matching features by capturing multiple distinctive patterns [104].

Also, SA can be cross [72] in which attentions are originated from different feature sets (schematic illustration is shown in Figure 9). Cross SA can be beneficial since correct matched point pairs in the left and right image usually exhibit similar geometric layouts, which can be attributed to their spatial compatibility. Similar to self SA, cross SA has the following form:

$$W^{SA} = \mathrm{SA}(F_1, F_2) = \mathrm{Softmax}(Q \times K^T) \tag{12}$$

where $K = g_{\omega_1}(F_1) \in R^{N\times C}$ and $Q = g_{\omega_2}(F_2) \in R^{N\times C}$, and they have same meanings as defined in Equation (8). By using cross SA, the mined features are cross validated and the distinctiveness and separability are further improved [104].

Besides, SA can be local (LSA) and global (GSA). Equation (9) actually is a GSA since the weights are generated for all the feature pairs regardless of their distances. Actually, feature pairs that are far apart in distances are very likely irrelevant, and calculating the attentions of these feature pairs is futile. Moreover, extensive researches demonstrate that spatially close matches in images exhibit motion coherence (i.e., local matches move together) [11], [25], [105], and mining information from neighboring features improves the precision of match representations [81], [106]. Therefore, to increase efficiency and precision, LSA only use

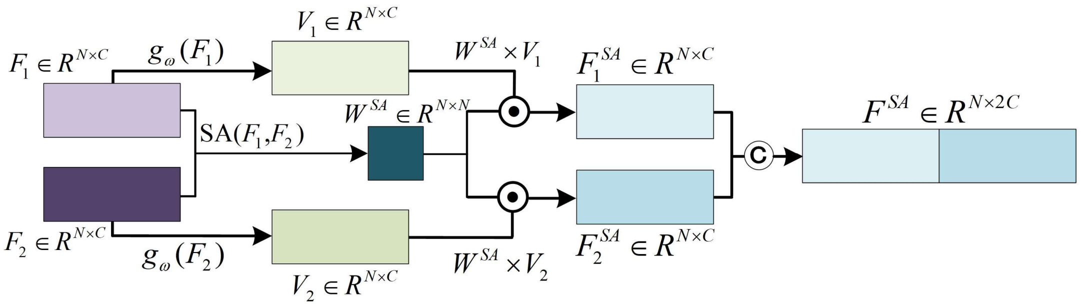


FIGURE 9. Schematic illustration of cross attention.

the k nearest neighbors (KNNs) to compute attentions. Note each feature can be extended to have multi-KNNs [96], [107], [108] in which multiple KNN sets are contained.

Extensive works apply SA to construct features of matches that incorporate both local and global information. NM-Net [81] searches consistent KNNs and aggregate information from these neighbors, while only geometrical information of single KNN is utilized; Local Neighborhood Correlation Network (LNCNet) [109] divides matches into sets of KNNs and calculates the correlation matrix among these features, the resulting correlation matrix is subsequently used to weight the features of matches, while the correlation matrix only uses local information. Local-Global Self-Attention (LAGA)-LFGC++ [110] enriches the representations of matches by combining global-local information obtained through self-attention, and similar to NM-Net, the local information is mined from KNNs, while it ignores multi-scale information; Multi-Scale Attention Network (MANet) extends the mine scopes to multiple scales, and it [111] respectively employs an attentive PointCN [70] block and an attentive pooling layer to capture both global and local information in a discriminative manner; Context Structure Representation Network (CSR-Net) [112] formulates the representations of matches as a whole-part consensus learning (similar to local-global information mining), and it also uses attention mechanism to recalibrate the mined features; IGS-Net [85] performs a search and visualization of KNNs to create a representation [113] that contains local context information of potential correspondences; Hierarchical Consensus Attention Network (HCA-Net) [114] incorporates a consensus attention mechanism to regularize sparse matches, the consensus attention is essentially a local attention as it primarily focuses on the nearest neighbors in Euclidean space.

In addition, several works have focused on mining information from both SA and CA, such as PESA-Net [90], Spatial-Channel Self-Attention Network (SCSA-Net) [115], Complex Information Extraction (CIE-Net) [116], Channel-Spatial Difference Augment Network (CSDA-Net) [117], and et. al. For the works concentrating on the integration of spatial-channel attention, we will give a detailed list in the following.

2) CHANNEL ATTENTION

SA estimates weights based on the similarity between feature pairs, while CA considers that not all the feature channels are essential for separating inliers and outliers, and some channels may be affected by noise, leading to a decrease in classification precision. The objective of CA is to assign lower weights on the classification-irrelative channels and suppress them to recalibrate the features [86], [101], [118]. The general form of CA is illustrated in Figure 7(b), and the recalibrated features can be represented by

$$F^{CA} = V \times W^{CA} \tag{13}$$

where $V = g_\omega(F) \in R^{N\times C}$ is the projected features of F, and g is the network and ω is its parameters. Analogous to spatial attention, Equation (13) can be rewritten as

$$f_j^{CA} = \sum_{i=1}^{C} w_{i,j}^{CA} \times c_i \tag{14}$$

where $f_j^{CA} \in R^{N\times 1}$ is the j-th recalibrated channel (column) vector, $w_{i,j}^{CA}$ is the i-th row and j-th column element of W^{CA}, and $c_i \in R^{N\times 1}$ is i-th channel (column) vector in V. It can be seen clearly that the channel attention is performed on the channels: if the weight is higher, the corresponding channel will be augmented; conversely, the channel is suppressed and nonsignificant in classification. By the ways of constructing the attention, CA that applies in mismatch removal can be classified into three types: global average pooling (GAP), covariance, and normalized covariance.

The strategy of using GAP to generate CA is proposed by Squeeze-and-Excitation Network (SE-Net) [86]. SE-Net squeezes the feature map F in channel dimension to calculate GAP, then it applies a simple gating mechanism [86] to capture channel-wise dependencies:

$$W^{CA} = \mathrm{Sigmoid}(W_2 \mathrm{ReLU}(W_1 \mathrm{GAP}(F))) \tag{15}$$

where $W_1 \in R^{{}^{C}/_{r} \times C}$ and $W_2 \in R^{C \times {}^{C}/_{r}}$ are the two learnable matrices and r is the reduction ratio. SE-Net is invariant to permutation since GAP is element-wise.

T-Net [88] utilizes GAP strategy to generate CA, while it replaces linear projection of SE-Net by a 1×1 convolutional kernel that captures more valuable channel-wise geometrical information and simultaneously maintain permutation invariance; CIE-Net [116] also employs GAP strategy to construct CA, which is used as a preprocessing process of SA; CSDA-Net [117] uses a max pooling (MP) operation, which is effective and efficient, to compensate for GAP, and it designs an overlay attention mechanism to integrate CA with SA; Relation-Aware Network (RANet) [119] also adopts the approach of simultaneously utilizing GAP and MP to generate CA, the features are synchronously weighted by SA and CA, and concatenating the weighted features as the final output; Multi-Scale Attention Network (MSA-Net) [120] also applies GAP to generate CA, aiming to exploit global information embedding in the channel dimensions, and it simultaneously mines local information by SA, both these two attentions are summed up to form a multi-scale attention to recalibrate the features; Unlike RANet, Joint Representation Attention Network (JRA-Net) [121] employs GAP to capture global information and SA to capture local information; Local Structure Visualization-Attention network (LSV-ANet) [113] incorporates SE-Net as a post-processing step to dynamically recalibrate the features, aiming to enhance useful feature channels and suppress needless channels; Representation-recalibration Network (R-Net) [122] applies the same gating mechanism previously employed in SE-Net to learn prominent channel, it combines with multiple KNNs to increase the versatility of learned features; Graph Context Attention Network (GCA-Net) [123] also follows the diagram of R-Net to process channel information, while it utilizes a graph context attention block to capture the local context exchanges. In summary, GAP is a rather simple method since it only considers the first-order channel correlation (concretely, mean values of features). Though it is effective, while some complex information (e.g., second-order information) can not be utilized, while leads to unsatisfied results in specific scenarios.

GAP is proposed based on the first-order statistics characteristic of features, although it is intuitional and effective in exploiting the significant channels of features, it is too simplistic for capturing complex global information, especially high order statistics [101]. To address this issue, many researchers propose to use correlations, such as covariance matrices, of features to model second-order statistics [124], [125], [126], [127], [128], [129], [130], [131]. Generally, covariance matrix of features can be expressed in the following form:

$$\tilde{Z} = \mathrm{Cov}(\tilde{F}) = \tilde{F}^T \times \tilde{F}, \quad \tilde{F} = F - \frac{1}{N}\sum_{i=1}^{N} f_i \tag{16}$$

where $\mathrm{Cov}(\cdot)$ computes the pairwise channel correlations, f_i is the i-th row of F, $\tilde{Z} \in R^{C\times C}$ is the covariance matrix. And CA generated from the covariance matrix is

$$W^{CA} = \mathrm{Softmax}(\tilde{Z}){=}\mathrm{Softmax}(\tilde{F}^T \times \tilde{F}) \tag{17}$$

Generating CA from covariance matrix is permutation invariant since

$$\begin{aligned}\mathrm{Softmax}(\mathrm{Cov}(P\tilde{F})) &= \mathrm{Softmax}(\tilde{F}^T P^T \times P\tilde{F}) \\ &= \mathrm{Softmax}(\tilde{F}^T \tilde{F})\end{aligned} \tag{18}$$

where $P^T \times P = I$ holds since P is a permutation matrix.

SCSA-Net [115] is a typical network that applies covariance matrix to extract CA, it also utilizes SA to mine the geometric information, both channel and spatial information is fused together to enhance the network's representative capability; Attention in Attention Network (ANA-Net) [132] takes into account the computational complexity associated with the covariance matrix, and it employs a linear approximation of the covariance matrix, which significantly reduces the complexity from $O(N^3)$ to $O(N)$.

The covariance presented in Equation (15) is essentially derived under Maximum Likelihood Estimation (MLE) of normally distributed features [131], and it is well known that MLE is not robust to features contaminated by outliers, or features of large dimensions with small size [133]. To address this issue, regularized MLE [134] and shrinkage principle (i.e., shrinking the largest eigenvalues and stretching the smallest ones) are proposed to robustly estimate the covariance, a notable method in this regard is Matrix Power Normalized COVariance (MPN-COV) [125], [128], [131], [134]. The normalized covariance estimated by MPN-COV is

$$\bar{Z} = \mathrm{NCov}(F) = \tilde{Z}^{1/2} = B \times \mathrm{DIAG}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_C}) \times B^T \tag{19}$$

where $\mathrm{NCov}(\cdot)$ is the normalized covariance of features, $\mathrm{DIAG}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_C})$ is a diagonal matrix and its diagonal elements are non-increasing, λ_i is the eigenvalue of the covariance matrix $\tilde{Z}$, $B = \left(b_1, \ldots, b_C\right)$ is an orthogonal matrix and its column vector b_i is the eigenvector corresponding to the eigenvalue λ_i. The adherence of the solution to the shrinkage principle is clearly evidenced, as the square roots of eigenvalues are enlarged if they are smaller than 1.0, and lowered if they are greater than 1.0. Normalized covariance is permutation invariant since

$$\mathrm{NCov}(PF)){=}(\tilde{F}^T P^T \times P\tilde{F})^{1/2} = \tilde{Z}^{1/2} \tag{20}$$

Correspondence Attention Transformer Network (CAT-Net) [136] proposes to use normalized covariance to compute CA since it argues that normalized covariance has two distinctive merits. Firstly, normalizing covariance is a precise way to calculate CA, leading to a more salient feature representation by performing information exchanges in channel dimension. In addition, CA generated by normalizing covariance requires only O(N) complexity in memory which is a much less memory request.

Similar to multi-head attention in SA, split attention in CA tries to mine cross-channel interactions among features [137], [138]. Split attention typically employs different sizes of convolutional kernels to partition a feature map into multiple

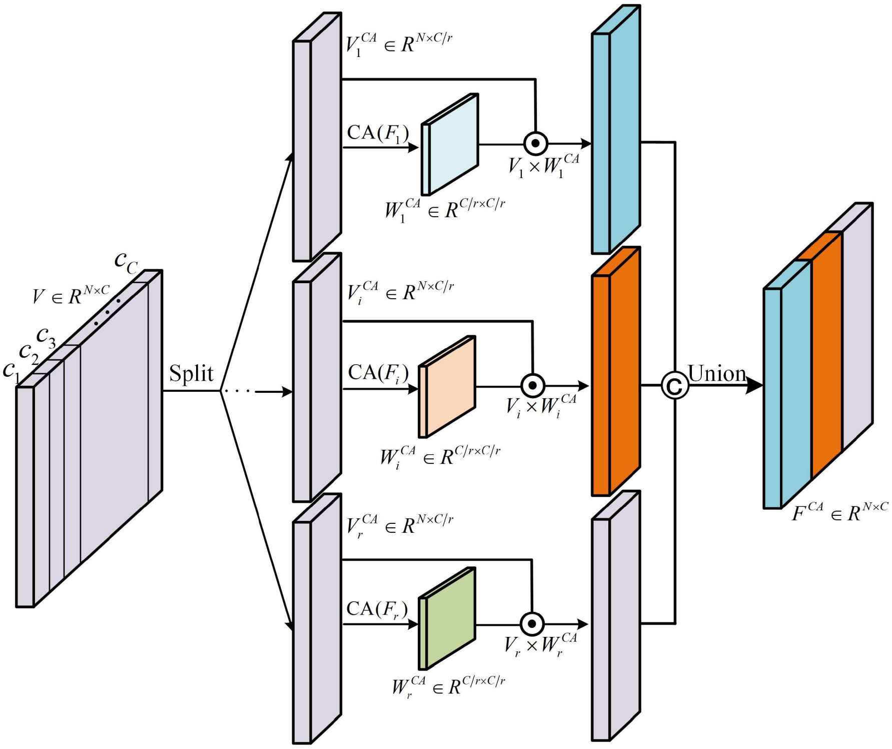

$V \in R^{N\times C}$
c_C
c_3
c_2
c_1
Split
$V_1^{CA} \in R^{N\times C/r}$
CA(F_1)
$V_1 \times W_1^{CA}$
$W_1^{CA} \in R^{C/r\times C/r}$
$V_i^{CA} \in R^{N\times C/r}$
CA(F_i)
$V_i \times W_i^{CA}$
$W_i^{CA} \in R^{C/r\times C/r}$
$V_r^{CA} \in R^{N\times C/r}$
CA(F_r)
$V_r \times W_r^{CA}$
$W_r^{CA} \in R^{C/r\times C/r}$
Union
$F^{CA} \in R^{N\times C}$

FIGURE 10. Illustration of network architecture of split attention.

branches and then utilizes CA to capture cross-channel information. While for the networks used in mismatch removal, every operation in the networks must be permutation invariant, making convolutional kernels with sizes greater than one unsuitable. To apply split attention in mismatch removal, the feature map is firstly divided into equal parts along the channels and then CA is used to extract information among channels. Splitting the feature map along channels reduces the dimensions of the feature vectors, thereby mitigating the problem of MLE in covariance estimation [133]. Figure 10 illustrates the schematic diagram of split attention.

The network architecture of split attention is divided into three parts (as exampled in Figure 10): splitting, extraction, and union. The process of splitting involves dividing the feature map F into r equal sub-maps along channels. In the extraction, each sub-map is used to form a branch in which CA is extracted and used to recalibrate the feature channels. In union, the outputs of CA branches are concatenated along channels to form a comprehensive feature map. Since splitting and union is permutation invariant, and CA branches are also permutation invariant, thus the split attention is permutation invariant.

PESA-Net [90] applies the same network architecture as presented in Figure 10, it divides the feature map into four parts that form four CA branches in attempt to model the interdependencies among the feature channels; Similarly, the network architecture of splitting operation is also adopted by PGFNet [91], while PGFNet employs a hierarchical residual-like manner to union the outputs of CA branches.

D. GRAPH NEURAL NETWORK

Graph Neural Networks (GNNs) are well-suited for processing putative matches for two main reasons. Firstly, matches can naturally be deemed as graph nodes, and relationships between matches can be encoded in edge weights. This allows for obtaining features of the matches by leveraging the node representations in the graph, which is the fundamental concept of GNNs. Secondly, GNNs are designed to process unorder data points though structed data, such images, can also be processed by GNNs. Furthermore, the sub-layers in GNNs are invariant to permutations of data points [139], making them applicable for remove mismatches without needing to explicitly account for permutation invariance. A GNN is operating on a graph $\mathcal{G}$ that gradually aggregates node features from neighboring nodes using stacked GNN layers (the example of feature updating through layers is illustrated in Figure 11). After obtaining the node features, the nodes corresponding to matches can be classified by a binary classifier and ultimately fulfill the mismatch removal.

The commonly used GNN structures in mismatch removal, such as graph convolution network, graph attention network, and graph pooling network, are the specialized forms of GNNs, which will be detailly described in the following.

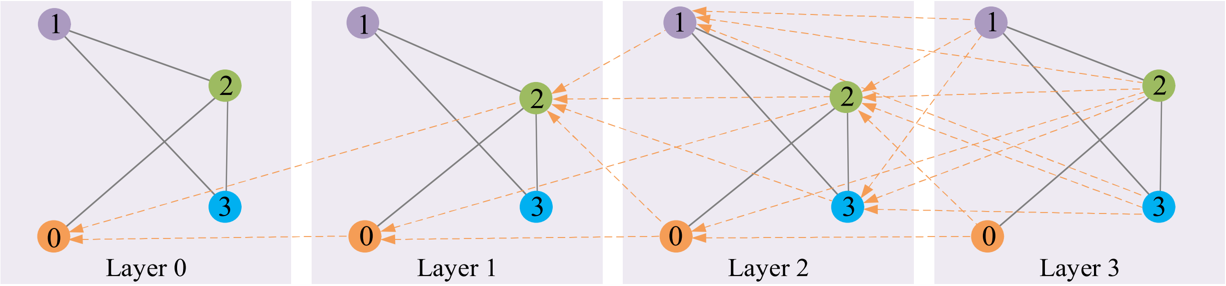


FIGURE 11. Example illustration of feature updating through GNN layers. As example showing by graph node 0, at the shallow layers, it accumulates information from neighboring nodes, as the layers going deeper, it can aggravate information from global graph nodes.

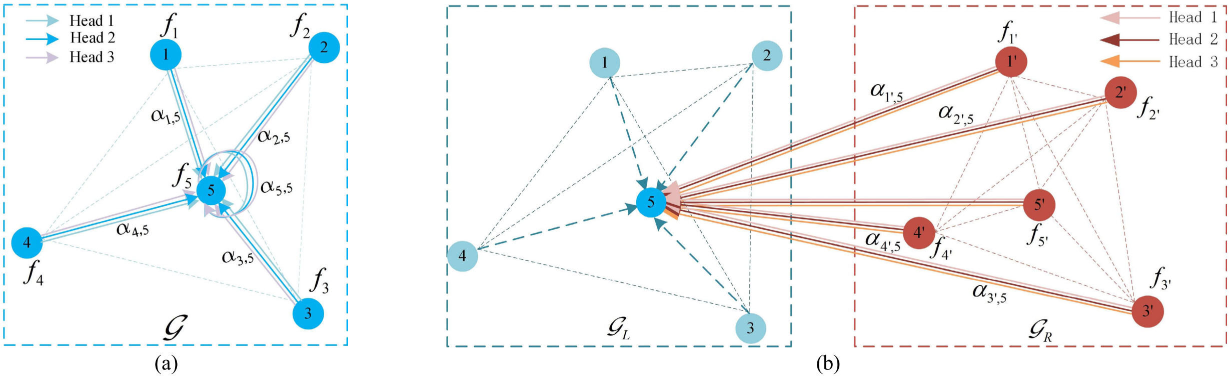


FIGURE 12. Schematic illustration of the GAT. (a) multiple head GAT, graph $\mathcal{G}$ is constructed by the putative correspondences, (b) multiple head cross GAT, $\mathcal{G}_L$ and $\mathcal{G}_R$ are respectively constructed by matched points in the left and right images.

1) GRAPH CONVOLUTION NETWORK

Graph convolution utilizes polynomial filters over node neighbors [140] to construct node features, which is much like the way that localized convolutional filters are computed over neighboring pixels of images. An effective approach to implement graph convolution is through spectral graph convolution [141] which is introduced in Graph Convolution Network (GCN) [142]. GCN considers spectral convolutions on graphs defined as the multiplication a graph signal $F \in R^{N\times C}$ (where N is the number of graph nodes, i.e., every node is assigned a C dimensional vector) and a filter g_w with w is the learnable parameter

$$g_w * F = Ug_wU^TF \tag{21}$$

where the operator "$*$" represents convolution, U is the matrix of eigenvectors of the normalized graph Laplacian $\tilde{L}$. Graph Laplacian $\tilde{L}$ has many useful properties of its own and can be directly used to mine geometrical information of matches. For example, Laplacian Motion Coherence Network (LMCNet) [105] firstly deduces smooth motions of matches from graph Laplacian, and the smooth motions are subsequently used as training signals under the consideration that coherence residuals of inliers are much smaller than those of outliers; ANA-Net [132] considers the high computational consumption of graph Laplacian and uses an attention-consistent context to exploit the motion consistency, the complexity is therefore decreased since the attention-consistent context can be approximated to a linear form.

In Equation (21), since the multiplication of U and the eigen decomposition of $\tilde{L}$ is time consuming, GCN approximates it by truncating the Chebyshev polynomial to first-order [143], [144], [145] and uses a renormalization trick to increase the numerical stabilities, the final output of GCN goes to

$$f_i' = \delta(\sum_{j\in\mathcal{N}_i}\frac{\sqrt{\tilde{D}_{i,i}\tilde{D}_{j,j}}}{\tilde{A}_{i,j}} \times f_jW) \tag{22}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of graph $\mathcal{G}$ with self-connections, $\tilde{D}$ is a diagonal matrix with $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$ and $W \in R^{C\times C'}$ is a learnable matrix; $\delta(\cdot)$ is an activation function, $f_j \in R^{1\times C}$ is the feature of node j (i.e., the j-th row of F), $f_i' \in R^{1\times C'}$ is the updated feature of node i.

Since GCN is essentially an operation of aggregating information from neighboring nodes in the graph, it is frequently used to mine the global information of matches. CLNet [83] introduces a two-step process involving KNN searching to construct local graphs and annular convolution to exploit local information, the local graphs are subsequently connected to form a global graph and GCN is applied to mine the global information, both local and global information are combined to separate inliers from outliers; the annular convolution proposed in CLNet is also applied by

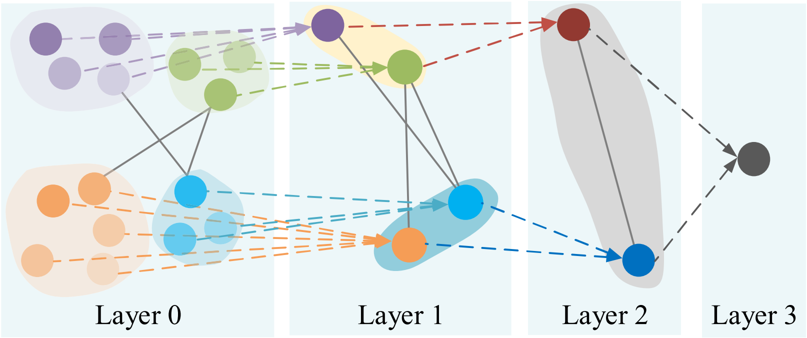


FIGURE 13. Conceptual diagram of graph pooling.

GCA-Net [123], but in this case, global information is captured by CA.

2) GRAPH ATTENTION NETWORK

In contrast to GCN, which performs graph convolutions in the spectral domain, Graph Attention Networks (GATs) [64] are spatial-based that operate convolutions on spatially close neighbors, the schematic illustration of the GAT is presented in Figure 12.

The input to a GAT layer is a set of node features $\mathcal{F} = \{f_1, f_2, \ldots, f_N\}$ where N is the number of graph nodes, $f_i \in R^{1\times C}$ is the feature vector of node i and C is number of feature vector channels. An attention between node i and j can be formulated as:

$$e_{i,j} = \mathrm{ATN}(f_i \times W, f_j \times W) \tag{23}$$

where $e_{i,j} \in R$ is the attention, W is a learnable matrix, $\mathrm{ATN}(\cdot,\cdot)$ is an attentional function (generally, an activation function) that maps two C-dimensional vectors to a scalar.

To make the attentions comparable across nodes, the attentions between node i and its neighbors are generally normalized by Softmax function

$$\alpha_{i,j} = \mathrm{Softmax}_j(e_{i,j}) = \frac{\exp(e_{i,j})}{\sum_{k\in\mathcal{N}_i}\exp(e_{i,k})} \tag{24}$$

where $\alpha_{i,j}$ is the normalized attention between node i and j, $\mathcal{N}_i$ is the set of neighboring nodes of node i. Finally, the output feature of node i can be expressed as

$$f_i' = \delta(\sum_{j\in\mathcal{N}_i} \alpha_{i,j} \times f_j W) \tag{25}$$

where f_i' is the output feature of node i by a GAT. And analogous to traditional multi-head attention mechanism [62], [63], GATs can also be multiple heads which are used to learn more expressive features (the schematic illustration of the multiple head attention is presented in Figure 11(a)). For the mismatch removal problem, the matched points in the left and right images can form two graphs, each graph's nodes have their own features, which can then be used to form an inter-graph attention known as cross GAT (its schematic illustration is presented in Figure 12(b)) [70], [104]. The motivation for using cross GAT is that attention essentially expresses the compatibility of features, since true matched points in the left and right image have similar geometrical layouts, allowing GAT to grasp the compatibility cross images.

Comparing Equation (22) and (25) makes the differences between GCN and GAT become evident. In GCN, the aggregation weight between a node and one of its neighbors is determined nonparametrically once the graph structure is established (the weight is a normalized const $\sqrt{\tilde{D}_{i,i}\tilde{D}_{j,j}/\tilde{A}_{i,j}}$). On the other hand, in GAT, as demonstrated in Equation (25), the weights are learnable and measure the compatibility of node features. Consequently, GAT can be generalized to unseen graphs with different structures, making it more suitable for tackling mismatch removal problems.

Coordinate Embedding Network (CE-Net) [104] takes into account the layout similarities of matching inliers, and it simultaneously incorporates multi-head cross and self GAT to extract geometrical information from intra and inter graphs; Graph Attention Network (GANet) [146] also employs the multiple head GAT to capture fine-grained geometric information from inliers while suppressing that from outliers, it meanwhile recognizes that only a small number of graph node features play a vital role in mismatch removal and proposes a sparse GANet to reduce computational complexity; Multiple Sparse Semantics Dynamic Graph Network (MS2DG-Net) [147] considers the local topology among matches and proposes dynamically building sparse semantic graphs to predicate matching probabilities, the core component of MS^{2}DG-Net is a self GAT layer that consolidates geometrical information.

3) GRAPH POOLING NETWORK

In contrast to GCN and GAT, which focus on node representations of graphs, graph pooling networks are designed to learn hierarchical representations of graphs. They gradually condense graph nodes and eventually predicates the entire label of a graph. DiffPool [65] is the notable example, and its conceptual diagram is depicted in Figure 13.

Given $F^{(l)} \in R^{N\times C}$ the graph node feature map inputting to layer l (i.e., the graph has N nodes and every node feature is a C-dimension vector), a DiffPool layer learns a soft assignment matrix $S_{pool} \in R^{N\times M}(M < N)$ to coarsen the graph

$$\begin{aligned} F^{(l+1)} &= S_{pool}^{T} F^{(l)} \\ S_{pool} &= \text{Softmax}(g_{\omega}(F^{(l)})) \end{aligned} \tag{26}$$

where $F^{(l+1)} \in R^{M\times C}$ is the feature map of the coarsened graph, $g_{\omega}(\cdot)$ is a GNN layer with a learnable parameter ω, which takes a feature map as the input. As demonstrated in Figure 13, through stacked multiple DiffPool layers, the global characteristic of the graph is progressively obtained and can be used to predict the label of the entire graph.

For mismatch removal problem, DiffPool layers can be useful to mine the global information of matches. However, relying solely on global information is inadequate for distinguishing inliers from outliers. To address this limitation, a DiffUnpool [82] layer is proposed to upsample and simultaneously refine the coarse representation. An intuitive way to implement a DiffUnpool layer is to learn a projection matrix from $F^{(l+1)}$ and using the matrix to unpool $F^{(l+1)}$ back to $R^{C\times N}$. While this operation is not optimal since $F^{(l+1)}$ is obtained by a permutation-invariant operation, which makes $F^{(l+1)}$ lose the original order of $F^{(l)}$. OANet [82] proposes an Order-Aware (OA) DiffUnpool to address the order disturbed problem by learning a soft assignment matrix directly from $F^{(l)}$

$$\begin{aligned} F'^{(l)} &= S_{unpool} \times F'^{(l+1)} \\ S_{unpool} &= \text{Softmax}(g'_{w}(F^{(l)})) \end{aligned} \tag{27}$$

where $g'_w(\cdot)$ is a GNN layer, $S_{unpool} \in R^{N\times M}$ is a learned projection matrix, $F'^{(l+1)} \in R^{M\times C}$ is a feature map computed from $F^{(l+1)}$, and $F'^{(l)} \in R^{N\times C}$ is the corrected feature map. From Equation (27) we can see that $F'^{(l)}$ can be viewed as a weighted average result of the features. $F'^{(l)}$ eventually can be concatenated with $F^{(l)}$ to fuse a new feature map.

The DiffPool and DiffUnpool layers serve as the foundational structure of OA-Net [82], and they act as basic functional layers to extract local and global geometrical information by various learning-based methods, including T-Net [88], RA-Net [121], GANet [146], and et. al.

E. COMPARSION OF MINING METHODS

Point-wise MLP is theoretically a feature dimension transformer, and therefore it is frequently used before a mining block to project old features and compose new features; Normalizations are almost an indispensable operation since they function as denoisers, which will increase the separability of features; SA aims at giving higher weights for features that originate from correct matches, and CA assign lower weights on the classification-irrelative channels to recalibrate the features, these two methods are sometimes alternately used to compensate for each other; GNN aims to mine global information from features of matches, GCN is a spectral domain based method and GAT is a spatial domain based method, and for GCN, the weights for integrating features are normalized const, while for GAT, the weights are learned, thus GAT are more suitable for scalable graphs; Unlike GCN and GAT which focus on node representations, graph pooling learns to coarsen graphs or extract generality from subgraphs, thus it is generally used to mine local information of matches.

V. TRAINING MODE

If we consider the mismatch removal problem as a classification problem, it can be addressed by a supervised learning with a sufficient number of labeled data. While the problem can also be viewed as a clustering problem, therefore unsupervised methods can also be employed. In the case of the supervised methods, the loss functions involve matching labels or/and geometrical constraints; while unsupervised methods try to explore inherent relationships among matches, thus prior labels are unnecessary.

A. SUPERVISED LEARNING

Supervised methods are widely utilized due to their straightforward nature. Designing a supervised method should firstly consider the training signals, then supervise the signals by loss functions involving with labels, and finally minimize the loss to get the optimal network. In the following, we will provide detailed descriptions about the training signals as well as various types of losses.

1) TRAINING SIGNALS

The straightforward training signal is matching probabilities. Bides, there are optional training signals, such as EG constraints and motion coherence residuals. If two images are stereo images and they are both captured by pin-hole cameras, then the true matched points from the two images are constraint by an EG [14]. If only Gaussian noises are present in the matches, the constraint matrix [148], [149] can be directly estimated; and if outliers are contained, IRLS [40] should be utilized to improve the accuracy. The predicted EGs can be supervised by ground truth EGs, such that the optimal DNNs can produce smaller residuals which are constrained by the estimated EGs.

EGs are limited to image pairs captured by pinhole cameras, while motion coherence residuals are universal. Correct matches will exhibit coherence residuals, since neighboring correct matches are physically constrained and cannot change freely in a small region [11]. The soft motion coherence residuals can be estimated from graph Laplacian [101]:

$$\begin{aligned} S &= U \times \text{DIAG}(1\big/(1+\eta\lambda_i)) \times U^T F^T - F^T \\ F_i &= f_i - f'_i \end{aligned} \tag{28}$$

where F_i is the i-th row of F which represents graph node features, U is the matrix of eigenvectors of the normalized graph Laplacian, λ_i is the eigenvalue of the eigenvector which is the i-th column of U (the meaning of U is presented in Equation (21)), and η a hyperparameter.

The motion coherence residuals cannot be directly supervised in training since displacements of features of matches cannot be accurately obtained. The residuals generally treated as an intermediate training signal which input to a DNN to output the probabilities of the corresponding matches, and then supervise the probabilities in training.

2) LOSS FUNCTION

In supervised learning, the loss functions measure the errors between predictions and labels. For mismatch removal, the main errors are cross entropy (i.e., the classification loss) that measures the probability distribution difference between the predicated probabilities and matching labels. Apart from cross entropy, geometrical residual errors (i.e., the regression loss) measure the difference between predicated and ground truth geometrical models. Both these two errors compose the general form of a loss function:

$$L = L_{cls} + \alpha \times L_{reg} \tag{29}$$

where L_{cls} and L_{reg} are respectively the classification loss and regression loss, and α is the balance factor between the two losses and generally set as a const. Classification loss L_{cls} usually is the binary cross entropy of labels and predications.

Directly using the binary cross entropy as the training loss will lead to the network being biased towards negative class since the number of outliers is generally much more than that of outliers [84]. Considering the bias problem, BACN [84] proposes to minimize Instance Balance Cross Entropy Loss (IB-CE-Loss)

$$L_{cls} = -\left(\lambda \frac{1}{N_{pos}} \sum_{i=1}^{N_{pos}} \log(y_i) + \mu \frac{1}{N_{neg}} \sum_{j=1}^{N_{neg}} \log(1-y_j)\right), \quad s.t.\ \lambda + \mu = 1, \quad N_{pos} + N_{neg} = N \tag{30}$$

where N_{pos} and N_{neg} are number of inliers and outliers, respectively, and λ and μ are the weights for inliers and outliers. By dynamically adjusting the two weights, BACN negatively correlates IB-CE-Loss with Fn-measure [150], i.e., IB-CE-Loss is decreasing with the increasing of Fn-measure, and ultimately the bias problem is alleviated.

The objective of minimizing L_{cls} is to increasing the recall and precision (i.e., increase the Fn-measure). While for the regression loss L_{reg}, it is sometimes combined with L_{cls} to improve the model regression accuracy simultaneously. L_{reg} has many different forms, here we review the most common EG models. If the fundamental/essential matrices of image pairs are given (note, if camera poses are given, the matrices can also be decomposed from the poses), L_{reg} has the following two forms

$$L_{reg\_se} = \sqrt{\sum_{i=1}^{N} \frac{y_i'^T E y_i}{\|S \times E y_i\|_2^2 + \|S \times E^T y_i'\|_2^2}}, \quad S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{31}$$

where L_{reg_se} is Sampson error [14], $\|\cdot\|_2$ represents L2 norm, E is the predicated EG constraint matrices, $y_i = (u_i, v_i, 1)$ and $y_i' = (u_i', v_i', 1)$ are the virtual matched point pair which are simulated by the ground truth EGs. And L_{reg} can also be

$$L_{reg\_L2} = \left\| E \pm \hat{E} \right\|_2^2 \tag{32}$$

where L_{reg_L2} is the $L2$ norm error, $\hat{E}$ and E are respectively the ground truth and predicated EG constraint matrices. Both Equation (30) and (32) have the same objective, that is, producing a precise EG constraint, while each approach adopts a different strategy. The former imposes the simulated correct matches with smaller Sampson errors under the predicated EG constraint; the latter adjusts the predicated matrix to make it numerically close to the ground truth.

B. UNSUPERVISED LEARNING

The mismatch removal problem can also be viewed as a clustering problem, utilizing the intrinsic geometric constraint to cluster the correct matches. Consequently, DNNs can be trained in an unsupervised mode. Based on the principles of RANSAC [41], the cluster with the greatest number of matches is the correct cluster, i.e., the maximal consensus set of the geometrical constraint is the outlier free matching set. By the way of clustering the maximal consensus, the unsupervised learning framework can be categorized into RESampling-based Methods (RESMs) and REGression-based Methods (REGMs).

1) RESAMPLING-BASED METHODS

RESMs follow the paradigm of RANSAC, i.e., sampling-hypothesis-verification, while the objective functions of the RESMs are slightly different from those of RANSAC. For RANSAC, the objective is to maximize the following equation

$$r_{\mathcal{M}}(\mathcal{C}) = |\mathcal{C}| \tag{33}$$

where $\mathcal{C}$ is a consensus set which is determined by the sampled minimal set $\mathcal{M}$ (the minimal set consists of minimal number of matches that can estimate a geometrical model), and $|\cdot|$ represents the element number of a set. Generally, Equation (33) is maximized by an iteratively sampling process. Whereas, using backpropagation to compute the gradient of Equation (33) is not feasible due to its non-differentiability with regards to the network parameters [151], thus Equation (33) cannot serve as a training objective function.

Alternatively, we can view the consensus set $\mathcal{C}$ as a random variable, and turn to maximize the expectation of Equation (33)

$$L = E_{\mathcal{M} \sim \Pr(\mathcal{M}|\mu)}(r_{\mathcal{M}}(\mathcal{C})) \tag{34}$$

where $\Pr(\mathcal{M}|\mu)$ is the likelihood of a minimal set $\mathcal{M}$, and $\mathcal{M} \sim \Pr(\mathcal{M}|\mu)$ means sampling a minimal set from the likelihood. Maximizing Equation (34) means maximizing the consensus set in probability. Though this slightly differs from directly maximizing Equation (33), both approaches yield similar results in statistics. Additionally, Equation (34) is differentiable, enabling estimation of the gradients using the policy gradient [152], As a result, the optimal DNN, which predicts a maximal consensus set, can be obtained through a gradient ascent algorithm.

Neural-Guided RANSAC (NG-RANSAC) [153] models the matching probabilities of putative matches by a categorical distribution and samples minimal sets from it. The optimal DNNs can be obtained by directly maximizing Equation (34). Unsupervised Learning for Mismatch Removal (ULMR) [68] analogizes the mismatch removal problem to playing games and applies reinforcement learning (RL) [154], [155] to solve it. From the perspective of RL, the putative matches can be seen as states and the sampling processes are actions. As a result, a reward can be assigned to every state-action pair, and maximizing the expected reward is equivalent to maximizing Equation (34). Therefore, the mismatch removal can be solved within the framework of deep RL, eliminating the need for matching labels or ground truth geometrical constraints.

2) REGRESSION-BASED METHODS

REGMs learn to regress a model of matching inliers, and the model has the maximal consensus set and simultaneously can minimize the proposed model fitting cost. Unsupervised Learning of Consensus Maximization (ULCM) [69] is a representative REGM, it formulates the mismatch removal problem as finding the maximal consensus set $\mathcal{S}$ that can be explained by a parametric model Φ

$$\Omega, \mathcal{S} = \underset{\Phi, \mathcal{C} \in \mathcal{X}}{\arg\max} \, |\mathcal{C}| \,, \qquad s.t. \; d(\Phi(y_i), y_i') \leq \varepsilon \tag{35}$$

where $\mathcal{C}$ is the consensus set of Φ and $\mathcal{X}$ is the putative matches, y_i and y_i' is a matched point pair in the consensus set $\mathcal{C}$, $d(\cdot,\cdot)$ is the distance of two elements.

By using ring theory [156], it can be certified that the model Φ is encoded in a weighted Vandermonde matrix $A \in R^{N\times 9}$ (N is the number of putative matches), and A can be constructed by the putative matches $\mathcal{X}$.

Given some reasonable approximations, Equation (34) equals to minimize the following loss

$$\begin{aligned} L &= \mathrm{Det}(E) + \eta \times \lambda_{\min} \\ \mathrm{SVD}(A) &= U \times S \times V \end{aligned} \tag{36}$$

where $\mathrm{SVD}(A)$ represents the singular value decomposition of A, U and V are the left and right singular matrices, S is a diagonal matrix and the singular values are arranged on its main diagonal in a descending order, $E \in R^{3\times 3}$ is a matrix representation of the last column of U and $\mathrm{Det}(E)$ is the determinant of E, $\lambda_{\min}$ is the minimum singular value of A (i.e. the last diagonal element of S), and η is a hyper-parameter.

By minimizing Equation (36), a model Ω with the maximal consensus set can be obtained, and the maximal consensus set is an outlier free set. As the loss function (i.e. Equation (36)) does not involve of matching labels or ground truth geometrical constraints, the DNNs can be trained in an unsupervised manner, and finally outputs matching probabilities to separate inliers from outliers.

C. COMPARISON OF LEARNING MODES

Most existing mismatch removal methods are supervised, since a supervised learning mode is straightforward, and additional constraints can be easily embedded in loss functions. While supervised learning methods are sometimes confronted with issues of labeling, which will reduce detection potentials of the mismatch removal methods. Unsupervised learning methods can refrain from effects of wrong labels, while because of the lack of labeling information, unsupervised learning generally needs more training data and consume more training time.

VI. CONCLUSION AND FUTURE TRENDS

Image matching is a critical component in remote sensing and computer vision tasks; however, mismatches are inevitable due to the complex image contents, which can have negative effects on downstream applications. Researchers have made impressive progresses in mismatch removal using handcrafted methods. The recent advancements in DL have further facilitated the development of mismatch removal methods; however, a comprehensive survey of DL-based mismatch removal methods is still lacking. Therefore, we provide a comprehensive review of these methods (please refer to Table 1 in Appendix A to see the representative methods and their used technologies).

The essence of DL-based methods is designing a PIN (i.e., a DNN invariant to permutation of inputted matches) to mine geometrical information among matches. A DNN with a linear or "T" structure can be formed by the combination of permutation-invariant operations, and the commonly used operations in the DL-based methods are point-wise MPL, normalization, attention, and graph-based operations. While current methods offer various improvements and advantages, the issue of mismatch removal remains an unresolved challenge that requires further attention in the following directions.

1) Multiple representations. Despite numerous proposed operations to extract geometrical information from matches, a majority of these operations operate under the assumption that true matches possess similar geometrical layouts. This assumption yields good results in narrow baseline stereo images, while it becomes less accurate for wide baseline stereo images and images with sudden depth changes, and DL-based methods will only preserve matches that exhibit geometrical similarity. Despite these matches are correct, they are poorly conditioned and result in an inaccurate model (an example can be found in Figure 1(b) of [157]). Therefore, it is crucial to consider multiple representations of matches. DL-based methods should provide both the matching probability of individual pairs and the overall quality of the matching set to distinguish inliers from outliers.

TABLE 1. Representative DL-based mismatch removal methods and their used technologies.

NO.	Reference	Abbreviation	Network structure	Main technologies used in IMB	Activation function used in WGB	Published date
0	[57]	PointNet	Linear	MLP	Softmax	2017
1	[70]	LFGC	Linear	MLP, CN	ReLU + Tanh	2018
2	[40]	DFENet	Linear	MLP, IRLS	Softmax	2018
3	[106]	N^3-Net	Linear	MLP, Differential KNN	ReLU + Tanh	2018
4	[81]	NM-Net	Linear	MLP, KNN, IN, CN	Sigmoid	2019
5	[82]	OA-Net	Linear	MLP, CN, OA, IRLS	ReLU + Tanh	2019
6	[107]	LMR	Linear	MLP, Multiple KNNs	Sigmoid	2019
7	[153]	NG-RANSAC	-	Unsupervised	-	2019
8	[69]	ULCM	-	Unsupervised	-	2019
9	[84]	GLHA	Linear/coarse to fine	MLP, CA, CN, Class imbalance	ReLU + Tanh	2020
10	[80]	ACNe	Linear	MLP, Weighted CN	ReLU + Tanh	2020
11	[105]	LMCNet	Linear	MLP, KNN, CN, Graph Laplacian	Sigmoid	2020
12	[109]	LFRC	Linear	MLP, KNN, CN	ReLU + Tanh	2020
13	[89]	Point2CN	T/coarse to fine	MLP, CN	ReLU + Tanh	2021
14	[98]	TSSN-Net	Linear	CN (followed by a switchable BN, IN, LN), Multiple KNNs	ReLU + Tanh	2021
15	[104]	CE-Net	Linear	MLP, GAT (cross, multiple heads)	ReLU + Tanh	2021
16	[83]	CLNet	Linear/coarse to fine	MLP, GCN, Graph Laplacian	ReLU + Tanh	2021
17	[88]	T-Net	T	MLP, CA (GAP), CN, OA	ReLU + Tanh	2021
18	[104]	LNCNet	Linear	MLP, KNN, SA, OA	ReLU + Tanh	2021
19	[111]	MANet	Linear	MLP, CN, SA	ReLU + Tanh	2021
20	[120]	LAGA-LFGC++	Linear	MLP, GSA, LSA, OA	ReLU + Tanh	2021
21	[115]	SCSA-Net	Linear	MLP, SA, CA (Covariance), CN, OA	ReLU + Tanh	2021
22	[116]	CIE-Net	Linear	MLP, SA, CA (GAP), CN	ReLU + Tanh	2021
23	[90]	PESA-Net	Linear/T	MLP, CN, CA (Split attention), OA	ReLU + Tanh	2022
24	[117]	CSDA-Net	Linear	MLP, SA, CA (GAP), OA, CN	ReLU + Tanh	2022
25	[136]	CAT-Net	Linear	MLP, CN, SA (multiple heads), CA (normalized covariance)	ReLU + Tanh	2022
26	[119]	RANet	Linear	MLP, CA (GAP), SA, CN, OA	ReLU + Tanh	2022
27	[140]	MSA-Net	Linear	MLP, CA (GAP), SA, OA	ReLU + Tanh	2022
28	[112]	CSR-Net	Linear	MLP, Structure Representation, SA	Softmax	2022
39	[113]	LSV-ANet	Linear	MLP, Structure Representation, SA, CA (GAP)	Softmax	2022
30	[85]	IGS-Net	Linear/T/coarse-to-fine	MLP, KNN, SA	Softmax	2022
31	[114]	HCA-Net	Linear	MLP, SA	Softmax	2022
32	[122]	R-net	Linear	MLP, SA (multiple KNNs), CA (GAP)	Softmax	2022
33	[146]	GANet	Linear	MLP, GAT (multiple heads), CN, OA	ReLU + Tanh	2022
34	[147]	MS^{2}DG-Net	Linear	MLP, CN, GAT	ReLU + Tanh	2022
35	[157]	NeFSAC	Linear/T	MLP, Class imbalance	Sigmoid	2022
36	[68]	ULMR	-	Unsupervised	-	2022
37	[91]	PGFNet	Linear/T/coarse to fine	MLP, SA, CA (split attention), OA, CN	ReLU + Tanh	2023
38	[121]	JRA-Net	Linear	MLP, SA, CA (GAP)	ReLU + Tanh	2023
49	[123]	GCA-Net	Linear/T	MLP, CA, SA, CN, OA, GCN	ReLU + Tanh	2023
40	[132]	ANA-Net	Linear	MLP, SA, CA (Covariance)	Sigmoid	2023

2) Unsupervised learning. Labeling data is time-consuming and cumbersome; meanwhile, inevitable erroneous labels will decrease the performance of DNNs, and in turn, more data is needed to compensate for the performance degradation of DNNs. Unsupervised learning takes a way out of the labeling problem, and makes the DNNs have a better generalization to unseen data. While current unsupervised methods are generally implemented within the framework of classical RANSAC, i.e., they learn to find the maximal consensus set and partly inherit the flaws of RANSAC (e.g., more models may exist in training data). Thus, multiple-model-based methods should be proposed to increase the stability and accuracy of mismatch removal.

3) Few-shot learning. As mentioned previously, labeling data between stereo images is a costly endeavor, and training DNNs with a large number of images is time-consuming. Additionally, numerous DNNs for mismatch removal have been proposed and trained using publicly available datasets. However, it remains unclear how few-shot learning is applied to generalize the DNNs to new scenarios and to speed up training. Therefore, few-shot learning-based methods should be studied to alleviate the problems of the expensive endeavor and time-consuming training.

APPENDIX A

See Table 1.

REFERENCES

[1] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, ''Image matching from handcrafted to deep features: A survey,'' *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 23–79, Jan. 2021.

[2] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls, ''Image matching across wide baselines: From paper to practice,'' *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 517–547, Feb. 2021.

[3] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew, ''Deep learning for instance retrieval: A survey,'' *Int. J. Comput. Vis.*, vol. 45, no. 6, pp. 7270–7292, Jun. 2023.

[4] L. Zheng, Y. Yang, and Q. Tian, ''SIFT meets CNN: A decade survey of instance retrieval,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1224–1244, May 2018.

[5] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, ''Building Rome in a day,'' *Commun. ACM*, vol. 54, no. 10, pp. 105–112, Oct. 2011.

[6] J. L. Schonberger and J. M. Frahm, ''Structure-from-motion revisited,'' in *Proc. IEEE-CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 4104–4113.

[7] S. Lynen, B. Zeisl, D. Aiger, M. Bosse, J. Hesch, M. Pollefeys, R. Siegwart, and T. Sattler, ''Large-scale, real-time visual-inertial localization revisited,'' *Int. J. Robot. Res.*, vol. 39, no. 9, pp. 1061–1084, Jul. 2020.

[8] C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla, F. Kahl, and T. Sattler, ''Long-term visual localization revisited,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2074–2088, Apr. 2022.

[9] D. Lu and Q. Weng, ''A survey of image classification methods and techniques for improving classification performance,'' *Int. J. Remote Sens.*, vol. 28, no. 5, pp. 823–870, Mar. 2007.

[10] S. S. Nath, G. Mishra, J. Kar, S. Chakraborty, and N. Dey, ''A survey of image classification methods and techniques,'' in *Proc. ICCICCT*, Paris, France, Jul. 2014, pp. 554–557.

[11] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, ''Locality preserving matching,'' *Int. J. Comput. Vis.*, vol. 127, no. 5, pp. 512–531, May 2019.

[12] J. Ma, X. Jiang, J. Jiang, and Y. Gao, ''Feature-guided Gaussian mixture model for image matching,'' *Pattern Recognit.*, vol. 92, pp. 231–245, Aug. 2019.

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

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Compute Vision*, 2nd ed. Cambridge, U.K.: CUP, 2004, pp. 239–257.

[15] J. Tian, Y. Wu, Y. Cai, H. Fan, and W. Yu, ''A novel mosaic method for spaceborne ScanSAR images based on homography matrix compensation,'' *Remote Sens.*, vol. 13, no. 15, p. 2866, Jul. 2021.

[16] H. Ji, Z. Gao, T. Mei, and Y. Li, ''Improved faster R-CNN with multiscale feature fusion and homography augmentation for vehicle detection in remote sensing images,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1761–1765, Nov. 2019.

[17] S. Chen, X. Yuan, W. Yuan, J. Niu, F. Xu, and Y. Zhang, ''Matching multi-sensor remote sensing images via an affinity tensor,'' *Remote Sens.*, vol. 10, no. 7, p. 1104, Jul. 2018.

[18] W. Crum, T. Hartkens, and D. Hill, ''Non-rigid image registration: Theory and practice,'' *Brit. J. Radiol.*, vol. 77, no. suppl_2, pp. 140–153, Jan. 2004.

[19] Y. Xia, J. Jiang, Y. Lu, W. Liu, and J. Ma, ''Robust feature matching via progressive smoothness consensus,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 196, pp. 502–513, Feb. 2023.

[20] J. Ma, A. Fan, X. Jiang, and G. Xiao, ''Feature matching via motion-consistency driven probabilistic graphical model,'' *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2249–2264, Sep. 2022.

[21] X. Jiang, Y. Xia, X.-P. Zhang, and J. Ma, ''Robust image matching via local graph structure consensus,'' *Pattern Recognit.*, vol. 126, Jun. 2022, Art. no. 108588.

[22] Y. Xia and J. Ma, ''Locality-guided global-preserving optimization for robust feature matching,'' *IEEE Trans. Image Process.*, vol. 31, pp. 5093–5108, 2022.

[23] A. Fan, X. Jiang, Y. Ma, X. Mei, and J. Ma, ''Smoothness-driven consensus based on compact representation for robust feature matching,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4460–4472, Oct. 2021.

[24] B. M. Dawant, ''Non-rigid registration of medical images: Purpose and methods, a short survey,'' in *Proc. IEEE-ISBI*, Washington, DC, USA, Jul. 2002, pp. 465–468.

[25] J. Bian, W. Lin, Y. Matsushita, S. Yeung, T. Nguyen, and M. Cheng, ''GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence,'' in *Proc. IEEE-CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 4181–4190.

[26] Z. Fan, C. Mao, Y. Wu, and J. Xu, ''Spectral graph matching and regularized quadratic relaxations: Algorithm and theory,'' in *Proc. ICML*, Vienna, Austria, Jul. 2020, pp. 2985–2995.

[27] B. Wu, Y. Zhang, and Q. Zhu, ''Integrated point and edge matching on poor textural images constrained by self-adaptive triangulations,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 68, pp. 40–55, Mar. 2012.

[28] X. Yuan, S. Chen, W. Yuan, and Y. Cai, ''Poor textural image tie point matching via graph theory,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 129, pp. 21–31, Jul. 2017.

[29] G. Qiao, H. Mi, T. Feng, P. Lu, and Y. Hong, ''Multiple constraints based robust matching of poor-texture close-range images for monitoring a simulated landslide,'' *Remote Sens.*, vol. 8, no. 5, p. 396, May 2016.

[30] X. Yang, J. Wang, X. Qin, J. Wang, X. Ye, and Q. Qin, ''Fast urban aerial image matching based on rectangular building extraction,'' *IEEE Geosci. Remote Sens. Mag. Replaces Newslett.*, vol. 3, no. 4, pp. 21–27, Dec. 2015.

[31] P. Men, H. Guo, J. An, and G. Li, ''An improved L2Net for repetitive texture image registration with intensity difference heterogeneous SAR images,'' *Remote Sens.*, vol. 14, no. 11, p. 2527, May 2022.

[32] G. Xu, Q. Wu, Y. Cheng, F. Yan, Z. Li, and Q. Yu, ''A robust deformed image matching method for multi-source image matching,'' *Infr. Phys. Technol.*, vol. 115, Jun. 2021, Art. no. 103691.

[33] Q. Wu, G. Xu, Y. Cheng, W. Dong, L. Ma, and Z. Li, ''Histogram of maximal point-edge orientation for multi-source image matching,'' *Int. J. Remote Sens.*, vol. 41, no. 14, pp. 5166–5185, Apr. 2020.

[34] E. Karami, S. Prasad, and M. Shehata, ''Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images,'' 2017, *arXiv:1710.02726*.

[35] L. Yue and Zheng, ''Distorted building image matching with automatic viewpoint rectification and fusion,'' *Sensors*, vol. 19, no. 23, p. 5205, Nov. 2019.

[36] M. Lourenco, J. P. Barreto, and F. Vasconcelos, ''SRD-SIFT: Keypoint detection and matching in images with radial distortion,'' *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 752–760, Jun. 2012.

[37] J. Flusser and T. Suk, ''A moment-based approach to registration of images with affine geometric distortion,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 2, pp. 382–387, Mar. 1994.

[38] J. Li, Q. Hu, and M. Ai, ''LAM: Locality affine-invariant feature matching,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 154, pp. 28–40, Aug. 2019.

[39] W. Baarda, ''A testing procedure for use in geodetic networks,'' *Neth. Geod. Commun.*, vol. 2, no. 5, pp. 2–5, 1968.

[40] R. Ranftl and V. Koltun, ''Deep fundamental matrix estimation,'' in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 284–299.

[41] M. A. Fischler and R. Bolles, ''Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,'' *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[42] *RANSAC in 2020: A CVPR Tutorial*. Accessed: Jul. 24, 2023. [Online]. Available: https://cmp.felk.cvut.cz/cvpr2020-ransac-tutorial/

[43] P. H. S. Torr and A. Zisserman, ''MLESAC: A new robust estimator with application to estimating image geometry,'' *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 138–156, Apr. 2000.

[44] P. H. S. Torr, ''Bayesian model estimation and selection for epipolar geometry and generic manifold fitting,'' *Int. J. Comput. Vis.*, vol. 50, no. 1, pp. 35–61, 2002.

[45] B. Tordoff and D. Murray, ''Guided sampling and consensus for motion estimation,'' in *Proc. IEEE-ECCV*, Berlin, Germany, May 2002, pp. 82–96.

[46] O. Chum and J. Matas, ''Matching with PROSAC-progressive sample consensus,'' in *Proc. IEEE-CVPR*, San Diego, CA, USA, Jul. 2005, pp. 220–226.

[47] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk, ''EVSAC: Accelerating hypotheses generation by modeling matching scores with extreme value theory,'' in *Proc. IEEE-ICCV*, Sydney, NSW, Australia, Dec. 2013, pp. 2472–2479.

[48] L. Haan and A. Ferreira, *Extreme Value Theory: An Introduction*. New York, NY, USA: Springer, 2006, pp. 65–126.

[49] D. R. Myatt, P. H. Torr, S. Nasuto, J. M. Bishop, and R. Craddock, "NAPSAC: High noise, high dimensional robust estimation-it's in the bag," in *Proc. IEEE-BMVC*, Cardiff, U.K., Sep. 2002, pp. 458–467.

[50] K. Ni, H. Jin, and F. Dellaert, "GroupSAC: Efficient consensus in the presence of groupings," in *Proc. IEEE-ICCV*, Kyoto, Japan, Sep. 2009, pp. 2193–2200.

[51] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: A universal framework for random sample consensus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 2022–2038, Aug. 2013.

[52] J. Li, P. Shi, Q. Hu, and Y. Zhang, "QGORE: Quadratic-time guaranteed outlier removal for point cloud registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11136–11151, Sep. 2023.

[53] J. Li, Q. Hu, M. Ai, and S. Wang, "A geometric estimation technique based on adaptive M-estimators: Algorithm and applications," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 14, pp. 5613–5626, 2021.

[54] J. Li, Q. Hu, and M. Ai, "Robust geometric model estimation based on scaled Welsch q-norm," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5908–5921, Aug. 2020.

[55] D.-X. Zhou, "Universality of deep convolutional neural networks," *Appl. Comput. Harmon. Anal.*, vol. 48, no. 2, pp. 787–794, Mar. 2020.

[56] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[57] C. R. Qi, H. Su, K. Mo, and L. J. Guiba, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE-CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 652–660.

[58] Y. L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. ICML*, Haifa, Israel, Jun. 2010, pp. 111–118.

[59] E. Fix and J. L. Hodges, "Discriminatory analysis—Nonparametric discrimination: Small sample performance," *Int. Stat. Rev.*, vol. 57, no. 3, pp. 238–247, Dec. 1989.

[60] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[61] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training DNNs: Methodology, analysis and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10173–10196, Aug. 2023.

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[63] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021.

[64] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: A comprehensive review," *Comput. Social Netw.*, vol. 6, no. 1, pp. 1–23, Nov. 2019.

[65] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. NIPS*, Montréal, QC, Canada, Dec. 2018, pp. 4805–4815.

[66] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[67] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[68] C. Deng, S. Chen, Y. Zhang, Q. Zhang, and F. Chen, "ULMR: An unsupervised learning framework for mismatch removal," *Sensors*, vol. 22, no. 16, p. 6110, Aug. 2022.

[69] T. Probst, D. P. Paudel, A. Chhatkuli, and L. V. Gool, "Unsupervised learning of consensus maximization for 3D vision problems," in *Proc. IEEE-CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 929–938.

[70] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE-CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 2666–2674.

[71] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[72] P. E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE-CVPR*, Seattle, WA, USA, Jun. 2020, pp. 4937–4946.

[73] H. R. Wilson and J. D. Cowan, "Excitatory and inhibitory interactions in localized populations of model neurons," *Biophysical J.*, vol. 12, no. 1, pp. 1–24, Jan. 1972.

[74] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[75] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, Sep. 2022.

[76] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, Jun. 2000.

[77] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE-CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[79] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[80] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, and K. M. Yi, "ACNe: Attentive context normalization for robust permutation-equivariant learning," in *Proc. IEEE-CVPR*, Seattle, WA, USA, Jun. 2020, pp. 11283–11292.

[81] C. Zhao, Z. Cao, C. Li, X. Li, and J. Yang, "NM-Net: Mining reliable neighbors for robust feature correspondences," in *Proc. IEEE-CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 215–224.

[82] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, H. Liao, and L. Quan, "Learning two-view correspondences and geometry using order-aware network," in *Proc. IEEE-ICCV*, Seoul, South Korea, Oct. 2019, pp. 5844–5853.

[83] C. Zhao, Y. Ge, F. Zhu, R. Zhao, H. Li, and M. Salzmann, "Progressive correspondence pruning by consensus learning," in *Proc. IEEE-ICCV*, Montreal, QC, Canada, Oct. 2021, pp. 6464–6473.

[84] Z. Chen, F. Yang, and W. Tao, "Cascade network with guided loss and hybrid attention for finding good correspondences," in *Proc. AAAI*, Vancouver, BC, Canada, Feb. 2021, pp. 1123–1131.

[85] J. Chen, S. Chen, Y. Liu, X. Chen, X. Fan, Y. Rao, C. Zhou, and Y. Yang, "IGS-net: Seeking good correspondences via interactive generative structure learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4705013.

[86] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE-CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[87] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE-ICCV*, Seoul, South Korea, Nov. 2019, pp. 510–519.

[88] Z. Zhong, G. Xiao, L. Zheng, Y. Lu, and J. Ma, "T-Net: Effective permutation-equivariant network for two-view correspondence learning," in *Proc. IEEE-ICCV*, Montreal, QC, Canada, Oct. 2021, pp. 1950–1959.

[89] X. Liu, G. Xiao, Z. Li, and R. Chen, "Point2CN: Progressive two-view correspondence learning via information fusion," *Signal Process.*, vol. 189, Dec. 2021, Art. no. 108304.

[90] Z. Zhong, G. Xiao, S. Wang, L. Wei, and X. Zhang, "PESA-Net: Permutation-equivariant split attention network for correspondence learning," *Inf. Fusion*, vol. 77, pp. 81–89, Jan. 2022.

[91] X. Liu, G. Xiao, R. Chen, and J. Ma, "PGFNet: Preference-guided filtering network for two-view correspondence learning," *IEEE Trans. Image Process.*, vol. 32, pp. 1367–1378, 2023.

[92] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Jul. 1995.

[93] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, Lille, France, Jul. 2015, pp. 448–456.

[94] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*.

[95] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[96] A. S. Goldberger, *Econometric Theory*. New York, NY, USA: Wiley, 1964, pp. 17–20.

[97] J. Sun, X. Cao, H. Liang, W. Huang, Z. Chen, and Z. Li, "New interpretations of normalization methods in deep learning," in *Proc. AAAI*, New York, NY, USA, Feb. 2020, pp. 5875–5882.

[98] Z. Zhong, G. Xiao, K. Zeng, and S. Wang, "TSSN-Net: Two-step sparse switchable normalization for learning correspondences with heavy outliers," *Neurocomputing*, vol. 452, pp. 159–168, Sep. 2021.

[99] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[100] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[101] M. Guo, T. Xu, J. Liu, Z. Liu, P. Jiang, T. Mu, S. Zhang, R. Martin, M. Cheng, and S. Hu, "Attention mechanisms in computer vision: A survey," *Comput. Vis. Media*, vol. 8, pp. 331–368, Mar. 2022.

[102] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. ECCV*, Munich, Germany, Sep. 2018, pp. 286–301.

[103] C. Chen, D. Gong, H. Wang, Z. Li, and K. K. Wong, "Learning spatial attention for face super-resolution," *IEEE Trans. Image Process.*, vol. 30, pp. 1219–1231, 2021.

[104] S. Chen, J. Niu, C. Deng, Y. Zhang, F. Chen, and F. Xu, "CE-net: A coordinate embedding network for mismatching removal," *IEEE Access*, vol. 9, pp. 147634–147648, 2021.

[105] Y. Liu, L. Liu, C. Lin, Z. Dong, and W. Wang, "Learnable motion coherence for correspondence pruning," in *Proc. IEEE-CVPR*, Nashville, TN, USA, Jun. 2021, pp. 3237–3246.

[106] T. Plötz and S. Roth, "Neural nearest neighbors networks," 2018, *arXiv:1810.12575*.

[107] J. Ma, X. Jiang, J. Jiang, J. Zhao, and X. Guo, "LMR: Learning a two-class classifier for mismatch removal," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4045–4059, Aug. 2019.

[108] Y. Wang, X. Mei, Y. Ma, J. Huang, F. Fan, and J. Ma, "Learning to find reliable correspondences with local neighborhood consensus," *Neurocomputing*, vol. 406, pp. 150–158, Sep. 2020.

[109] L. Dai, X. Liu, J. Wang, C. Yang, and R. Chen, "Learning two-view correspondences and geometry via local neighborhood correlation," *Entropy*, vol. 23, no. 8, p. 1024, Aug. 2021.

[110] L. Dai, X. Liu, Y. Liu, C. Yang, L. Wei, Y. Lin, and R. Chen, "Enhancing two-view correspondence learning by local–global self-attention," *Neurocomputing*, vol. 459, pp. 176–187, Oct. 2021.

[111] Y. Chen, L. Zheng, X. Liu, and G. Xiao, "MANet: Multi-scale attention network for correspondence learning," *IEEE Signal Process. Lett.*, vol. 28, pp. 1978–1982, 2021.

[112] J. Chen, S. Chen, X. Chen, Y. Dai, and Y. Yang, "CSR-net: Learning adaptive context structure representation for robust feature correspondence," *IEEE Trans. Image Process.*, vol. 31, pp. 3197–3210, 2022.

[113] J. Chen, S. Chen, X. Chen, Y. Yang, L. Xing, X. Fan, and Y. Rao, "LSV-ANet: Deep learning on local structure visualization for feature matching," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4700818.

[114] S. Chen, J. Chen, Y. Rao, X. Chen, X. Fan, H. Bai, L. Xing, C. Zhou, and Y. Yang, "A hierarchical consensus attention network for feature matching of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4706211.

[115] X. Liu, G. Xiao, L. Dai, K. Zeng, C. Yang, and R. Chen, "SCSA-net: Presentation of two-view reliable correspondence learning via spatial-channel self-attention," *Neurocomputing*, vol. 431, pp. 137–147, Mar. 2021.

[116] C. Jun, G. Yue, L. Linbo, G. Wenping, and W. Yong, "Two-view correspondence learning via complex information extraction," *Multimedia Tools Appl.*, vol. 81, no. 3, pp. 3939–3957, Jan. 2022.

[117] S. Chen, L. Zheng, G. Xiao, Z. Zhong, and J. Ma, "CSDA-net: Seeking reliable correspondences by channel-spatial difference augment network," *Pattern Recognit.*, vol. 126, Jun. 2022, Art. no. 108539.

[118] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T. S. Chua, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE-CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 5659–5667.

[119] G. Lin, X. Liu, F. Lin, G. Xiao, and J. Ma, "RANet: A relation-aware network for two-view correspondence learning," *Neurocomputing*, vol. 488, pp. 547–556, Jun. 2022.

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

[121] Z. Shi, G. Xiao, L. Zheng, J. Ma, and R. Chen, "JRA-net: Joint representation attention network for correspondence learning," *Pattern Recognit.*, vol. 135, Mar. 2023, Art. no. 109180.

[122] S. Chen, J. Chen, Z. Xiong, L. Xing, Y. Yang, F. Xiao, K. Yan, and H. Li, "Learning relaxed neighborhood consistency for feature matching," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4702913.

[123] J. Guo, G. Xiao, Z. Tang, S. Chen, S. Wang, and J. Ma, "Learning for feature matching via graph context attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5102714.

[124] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, Jun. 2013.

[125] Q. Wang, J. Xie, W. Zuo, L. Zhang, and P. Li, "Deep CNNs meet global covariance pooling: Better representation and generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2582–2597, Aug. 2021.

[126] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. ECCV*, Heraklion, Crete, Greece, Sep. 2010, pp. 143–156.

[127] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE-CVPR*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.

[128] T. Dai, J. Cai, Y. Zhang, S. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proc. IEEE-CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 11065–11074.

[129] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE-CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 3024–3033.

[130] T. Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. ECCV*, Santiago, Chile, Dec. 2015, pp. 1449–1457.

[131] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" in *Proc. ECCV*, Venice, Italy, Oct. 2017, pp. 2070–2078.

[132] X. Ye, W. Zhao, H. Lu, and Z. Cao, "Learning second-order attentive context for efficient correspondence pruning," 2023, *arXiv:2303.15761*.

[133] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006, pp. 220–222.

[134] Q. Wang, P. Li, W. Zuo, and L. Zhang, "RAID-G: Robust estimation of approximate infinite dimensional Gaussian with application to material recognition," in *Proc. IEEE-CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 4433–4441.

[135] B. Kulis, M. A. Sustik, and I. S. Dhillon, "Low-rank kernel learning with Bregman matrix divergences," *J. Mach. Learn. Res.*, vol. 10, pp. 341–376, Feb. 2009.

[136] J. Ma, Y. Wang, A. Fan, G. Xiao, and R. Chen, "Correspondence attention transformer: A context-sensitive network for two-view correspondence learning," *IEEE Trans. Multimedia*, vol. 25, pp. 3509–3524, 2023.

[137] L. Su, C. Hu, G. Li, and D. Cao, "MSAF: Multimodal split attention fusion," 2020, *arXiv:2012.07175*.

[138] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Müeller, and R. Manmatha, "ResNeSt: Split-attention networks," in *Proc. IEEE-CVPR*, New Orleans, LA, USA, Jun. 2022, pp. 2736–2746.

[139] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," 2021, *arXiv:2104.13478*.

[140] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: Algorithms, applications and open challenges," in *Proc. CSoNet*, Shanghai, China, Dec. 2018, pp. 79–91.

[141] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.

[142] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[143] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Boca Raton, FL, USA: CRC Press, 2002, pp. 3–10.

[144] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[145] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NIPS*, Barcelona, Spain, Dec. 2016, pp. 3844–3852.

[146] X. Jiang, Y. Wang, A. Fan, and J. Ma, "Learning for mismatch removal via graph attention networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 190, pp. 181–195, Aug. 2022.

[147] L. Dai, Y. Liu, J. Ma, L. Wei, T. Lai, C. Yang, and R. Chen, "MS2DG-Net: Progressive correspondence learning via multiple sparse semantics dynamic graph," in *Proc. IEEE-CVPR*, Orleans, LA, USA, Jun. 2022, pp. 8963–8972.

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

[149] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, Jun. 1997.

[150] C. J. Van Rijsbergen, "Foundation of evaluation," *J. Documentation*, vol. 30, no. 4, pp. 365–373, Apr. 1974.

[151] J. Schulman, N. Heess, T. Weber, and P. Abbeel, "Gradient estimation using stochastic computation graphs," in *Proc. NIPS*, Montreal, QC, Canada, Dec. 2015, pp. 3528–3536.

[152] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, Dec. 1999, pp. 1057–1063.

[153] E. Brachmann and C. Rother, "Neural-guided RANSAC: Learning where to sample model hypotheses," in *Proc. IEEE-CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 4322–4331.

[154] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.

[155] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," 2020, *arXiv:2006.16712v4*.

[156] C. P. Milies, S. K. Sehgal, and S. Sehgal, *An Introduction to Group Rings*. Berlin, Germany: Springer, 2002, pp. 41–50.

[157] L. Cavalli, M. Pollefeys, and D. Barath, "NeFSAC: Neurally filtered minimal samples," in *Proc. ECCV*, Tel Aviv, Israel, Oct. 2022, pp. 351–366.

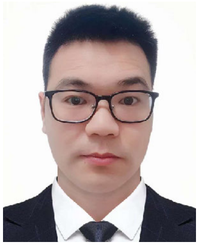
SHIYU CHEN was born in Xinyang, Henan, China, in 1986. He received the B.S. degree in remote sensing science and technology from Information Engineering University, Zhengzhou, Henan, in 2010, and the M.S. and D.E. degrees from the School of Remote Sensing and Engineering, Wuhan University, Wuhan, China, in 2014 and 2017, respectively. He is currently a Lecturer with Xinyang Normal University. His research interests include computer vision and precise agriculture, and devote himself to pest control via remote sensing methods.

CAILONG DENG received the B.S. degree in geomatics engineering from Wuhan University, Wuhan, China, in 2012, the M.S. degree in environmental engineering from The First Institute of Oceanography, Qingdao, China, in 2015, and the D.E. degree from the School of Remote Sensing and Engineering, Wuhan University in 2022. His current research interests include image processing, deep learning, and multi-sensor integration and fusion.

YONG ZHANG received the Ph.D. degree from the School of Remote Sensing and Engineering, Wuhan University, Wuhan, China, in 2017. He has been with VisionTech research, since 2000. He became a Senior Research Scientist, in 2021. His research interests include computer vision, multispectral remote sensing, and 3D reconstruction from images.

YONG WANG received the B.S. degree in cartography and GIS from the Graduate University of Chinese Academy of Sciences, China, in 2008. He is currently pursuing the Ph.D. degree in traffic and transportation with Southwest Jiaotong University, Chengdu, China. He is a Senior Engineer with the Sichuan Institute of Land Science and Technology (Sichuan Center of Satellite Application Technology). His research interest includes investigation and monitoring for natural resources.

QIXIN ZHANG was born in Luoyang, Henan, China, in 1998. She received the Bachelor of Science degree in geography from Xinyang University, Henan, in 2021. She is currently pursuing the master's degree with Xinyang Normal University. Her research interest includes computer vision.



ZHIMIN ZHOU received the B.S. degree in geomatics engineering from the China University of Petroleum (East China), Qingdao, China, in 2017. He is currently with the Sichuan Institute of Land Science and Technology (Sichuan Center of Satellite Application Technology). His research interest includes survey and monitoring of natural resources.