Machine Vision and Applications (2023) 34:130
https://doi.org/10.1007/s00138-023-01482-3

ORIGINAL PAPER

LTM: efficient learning with triangular topology constraint for feature matching with heavy outliers

Chentao Shen[1] · Zaixing He[1] · Xinyue Zhao[1] · Wenfeng Cui[2] · Huarong Shen[2]

Received: 22 April 2023 / Revised: 10 September 2023 / Accepted: 4 October 2023 / Published online: 1 November 2023

Abstract
Image feature matching, which aims to establish correspondence between two images, is an important task in computer vision. Among image feature matching, the removal of mismatches is crucial to ensure the correctness of the matches. In recent years, machine learning has become a new perspective for mismatch removal. However, existing learning-based methods require a large amount of image data for training, which shows a lack of generalizability and is hard to deal with cases with high mismatch ratio. In this paper, we induce the triangular topology constraint into machine learning, where topology constraints around the matching points are summarized; combining with the idea of sampling, we achieve the task of removing mismatches. Topology constraints are studied in spite of the image input; our LTM (learning topology for matching) just needs fewer than 20 parameters as input, so that only ten training image pairs from four image sets involving about 3,000 matches are employed to train; it still achieves promising results on various datasets with different machine learning approaches. The experimental results of this study also demonstrate the superior performance of our LTM over existing methods.

Keywords Topology constraints · Machine learning · Probability sampling · Mismatch removal

1 Introduction

Feature point matching is a fundamental and critical aspect of machine vision that plays a significant role in wide applications, including object detection, image registration and fusion, tracking, and pose estimation.

Generally, feature point matching consists of two key steps: the first step is to get initial feature matches. It first extracts feature points that describe local information, for which many methods have been proposed such as SIFT (scale-invariant feature transform) [1], SURF (speeded up robust features) [2], and ORB (oriented FAST and rotated BRIEF) [3], or deep learning-based, such as LIFT (learned invariant feature transform) [4] and SuperPoint [5]. After feature extraction, a one-to-one matching relationship between feature points in different images can be established, using methods such as brute force, constructing K-nearest tree, etc.

However, despite the optimization of establishing initial matches, it is inevitable to produce mismatches. Therefore, the second step is the removal of mismatching, which is of great importance.

The existing methods to remove mismatches can be mainly divided into resampling-based, geometry-based, and learning-based methods. Resampling-based methods continually sample points to generate a transformation matrix for two images, while geometry-based methods rely on distance or angle constraints to identify mismatches, some learning-based methods have been proposed recent years, which apply machine learning to image matching. However, as the number of mismatches increases, resampling-based methods may not converge, and geometry-based methods become more complex to solve, resulting in poor performance, and for images with non-global homographic transformations, these methods often manage to filter out a certain portion of correct matches, which reach better performance. Though learning-based methods achieve better adaptability to the situations,

✉ Zaixing He
zaixinghe@zju.edu.cn

✉ Xinyue Zhao
zhaoxinyue@zju.edu.cn

[1] School of Mechanical Engineering, the State Key Lab of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310058, People's Republic of China

[2] Zhejiang Feihang Intelligent Technology Co., LTD, Huzhou 313200, People's Republic of China

most of them have limited generalizability and need lots of training data.

To address the issues above, our paper proposes a novel approach, LTM (learning topology for matching). It first segments the matches based on the clustering relationship of the matched feature points, which results in match subsets and discrete matches. For the match subsets, we construct both a triangular topology network and a reconnection network and obtain the mismatch probability for each match by machine learning based on the distortion of reconnection network, then use the mismatching probability as a prior probability for sampling to calculate the transformation matrix for each subset, and finally remove the outliers in the subset. For the discrete points, they are judged based on their topological relationship with surrounding neighbor points.

The contributions of this paper are twofold:

(1) A novel framework for image feature matching is proposed. Despite relying on purely data-driven learning with large models, we first extract topological information and then use the processed information as input for small models to calculate the mismatching probability for each match. Based on this probability, we further remove the mismatches with sampling.
(2) A method for mismatch probability based on machine learning and triangular topology network constraints is proposed. It firstly segments the matches to some subsets and discrete matches by triangulation and then constructs reconnection network; the distortions of reconnection network of matching points and their neighboring points are inputted to learning approaches.

The remainder of this paper is organized as follows. Section 2 describes related work of mismatch removal. In Sect. 3, we introduce the principle of our method. Section 4 illustrates the performance of our method compared with other state-of-the-art methods on different experiments. Then we conclude our work in Sect. 5.

2 Related works

The current study includes three categories: resampling-based methods, geometry-based methods, and learning-based methods.

2.1 Resampling-based methods

The basis of the resampling-based methods is the principle that the correct matching points conform to a transform model, the outliers (the points do not conform to the model) are mismatches and be removed them. The resampling-based method is to find a transform model that makes the maximum number or more than a certain number of matches that meet which.

The most popular method in the area of mismatch removal is using RANSAC (random sample consensus) to calculate the transform matrix [6, 7], it estimates the optimal matrix of two images, and the outliers of the matrix model are considered the mismatches. However, the efficiency and the probability to find the solution will be reduced when there are more than 50% mismatches [40, 43].

Therefore, a lot of variations are proposed to improve RANSAC. PROSAC (progressive sampling consensus) firstly obtains the probability of each data being an inlier and then preferentially extracts the data with high probability in the random process [8]. GroupSAC firstly groups all of matches, the group with more matching points is preferred when in the sampling [9]. R-RANSAC (randomized RANSAC) and SPRT-RANSAC (randomized RANSAC with sequential probability ratio test) will judge whether it is the correct model first after finding the model and will continue to sample and iterate if not [10, 11]. DL-RANSAC (descendant likelihood-RANSAC) introduces descending likelihood to reduce the randomness so that it converges faster than the conventional RANSAC [12]. SESAC (sequential evaluation on sample consensus) sorts the matches based on the similarity of the corresponding features and then selects the samples sequentially and the get the model by least squares method, which performs better than PROSAC [13], and Wu et al. proposed fast sample consensus, improving the efficiency of RANSAC [41].

Over the past few decades, these methods had continued to be considered the effective solution for selecting accurate inliers and robustly estimating models, so that they are widely used in rigid feature matching. Generally, it performs well in ordinary scenes, where the mismatch ratio is not high. However, it is challenging for it to deal with the high mismatch cases.

2.2 Geometry-based methods

Many researchers focus on the geometry of the matching points to construct the geometric or topological constraint between the matching points to remove mismatches.

GTM (graph transformation matching) [14], proposed by Aguilar et. al., is a typical method based on geometry, which constructs the KNN (k-nearest neighbors) undirected graph based on matching points, then removes the mismatches and reconstructs the KNN graph until two images have similar graph. Zhang et al. also relied on the k-nearest neighbors with triangle-area representation to identify mismatches [39]. GMS (grid-based motion statistics) [15], based on theory of motion statistics, can separate true matches from mismatches especially at the image pairs with high-speed transform.

LPM (locality preserving matching) [16], LGSC (local graph structure consensus) [17], LOGO (locality-guided global-preserving optimization) [33], and PSC (progressive smoothness consensus) [34], all proposed by Ma, use the principle that the local neighborhood structures of true matches will maintain in different images. Meanwhile, Liu et al. combined the local neighborhood structures and global information to find out correct matches [38].

In [18], we proposed a robust method based on comparing triangular topology and distance constraint of feature points. Luo et. al. analyzed the relationship of Euclidean distance between the matching points then correct the mismatch based on angular cosine [19], Zhao et al. removed the mismatches according to the constraints that matching distances tend to be consistent [20]. Jiang et al. [22] removed the mismatches based on the descriptor similarity, which casts the feature matching into a spatial clustering problem achieved by DBSCAN. Shao et al. [23] proposed MRME which calculates the minimum relative motion entropy to improve the accuracy of matching. Recently, Cavalli et al. [24] proposed a hierarchical pipeline for effective mismatching detection based on the local affine consensus.

These methods can detect mismatches more efficiently compared with resampling-based method because it avoids the iteration of sample. However, although these methods have better results, they cannot perform well when outliers are dominate, because it will fail to construct the correct constraints.

Recently, few geometry and resampling-combining methods have been proposed, which aim to integrate sampling and geometry constraints to take both advantages, such as Zhu et al. [21] proposed improved RANSAC and Lan et al. proposed GMS-RANSAC [22]. Furthermore, Li et al. combined sampling with the affine invariance of the triangle-area representation, which enhances the robustness and accuracy in remote sensing image matching [40], while H. Zhang et al. also relied on them with a circle descriptor to improve the performance [42]; however, such a simple combination of geometry constraints and sampling shows a limited improvement.

In [23], an effective combination of geometry and resampling is proposed, which calculates the mismatch probability of each matching point through triangulation constraints and calculates the transformation model of the image pairs through probability sampling. The method shows a good performance especially in a high mismatching rate compared to the existing methods. However, its application is limited to homography or approximate homography transformations.

2.3 Learning-based methods

Learning-based matching methods leverage the advantages of learning to induce geometric features for improving the effectiveness of mismatch removal.

Yi et al. [30] first attempted to remove mismatches with the depth neural network, which is based on multilayer perception for binocular vision. However, its effect depends on the input of camera intrinsic parameters. PGFNet [35] designs a novel iterative filtering structure to remove mismatches and get the camera position. SuperGlue [31] combines the inferred matching and outlier removal based on graphic neural network (GNN), which has a good result combined with SuperPoint [5]. Recently, Zhang et al. [32] used order-aware networks (OAnet) to learn binocular stereo geometry. However, the dependence on training data greatly impacts the universality of these learning-based methods. Therefore, there is an urgent need for an efficient and more practical method learning geometric information rather than image information to obtain a more universal solution.

Ma et al. [36] proposed an innovative two-class classifier LMR for removing mismatch data with a linear time complexity, which is more universal compared to the aforementioned learning-based approaches. However, it tends to preserve outliers when images undergo structural deformations or in the presence of high mismatch rates.

Despite the effectiveness of learning-based methods, real-world visual tasks still encounter numerous challenges. Due to the typically unpredictable and complex nature of the transformation types between image pairs, more universal methods are required.

3 Methodology

Given the current challenges in the field, we propose a learning and geometry combined mismatch removal approach by employing a learning approach based on triangular topology. As shown in Fig. 1, the method segments the matches based on a triangulation topology network and applies topological constraints to machine learning to obtain the mismatch probability, calculation a transformation model through probability sampling, finally achieving the removal of mismatches. Each stage will be introduced in Sects. 3.1–3.3.

3.1 Feature match clustering.

Generally, the position and shape of objects in two images conform to a set of transformation relationships, such as affine and homographic transformations. When an object undergoes a transformation, the position of the object in the image changes, but the topological relationships of its constituent parts remain the same. Therefore, as for the points

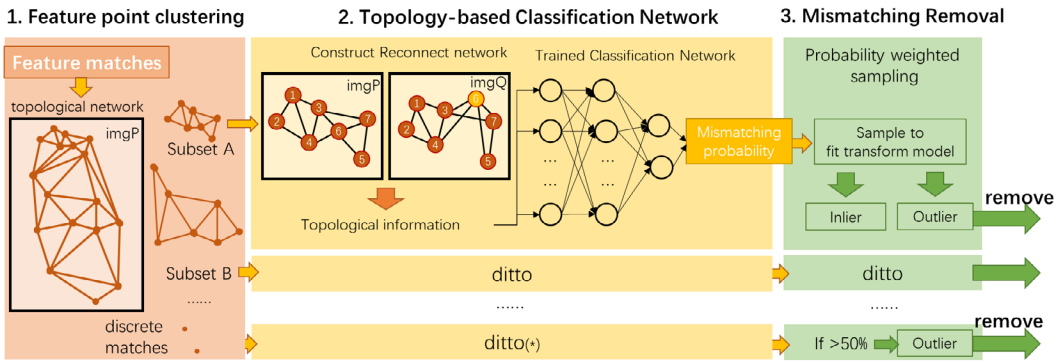

Fig. 1 Flowchart of the proposed method. * The stage 2 for discrete matches should construct network with neighbor points and then continue the process as subsets

pairs on two images of an object, the correctly matched points in the two images will have similar topological relationships, while the incorrectly matched points will not, as they do not conform to the transformation relationship.

However, in many cases, a pair of images contains multiple objects, each of which may have their own transformation different from each other. For example, in binocular vision, the objects at different depths correspond to different homography transformations, as shown in Fig. 2. The topological relationships between different objects may not remain consistent. However, though sometimes the entire image does not conform to a single transformation relationship. Locally, each object's surface conforms to a homography transformation, and the topological relationships of the points on the surface are preserved.

Therefore, in this paper, all matches are first segmented based on their neighbor relationships and divided into many "localities", which we refer to as subsets, the matching points which usually reflect the same object and may conform to the same transformation. Thus, in each local subset, mismatches can be judged and removed based on topological constraints.

The specific details are as follows:

(a) Firstly, the matching points on one image (called image *P*) is triangulated to construct a triangular topology network. As shown in Fig. 3(a), each feature point is connected to its neighboring feature points by topological edges. In regions with dense feature points, the edges are relatively short, while in regions with sparse feature points, the line segments are relatively long. Then, we calculate the lengths of the line segments, remove some longest edges, and we divide matches into some subsets according to whether their matching points in image P are connected. The red dotted lines in Fig. 3(b) indicate the removed edges.

(b) Next, we use a growth-based approach to create and merge local subsets. We select the end of the topological edges that are mutually the shortest as the starting point for the feature match subset growth, and then expand outward to its neighborhood points. If the expansion reaches the feature points of another subset, we compare the length of edge around this point with the average edge length of two subsets. If it is closer to the average edge length of one subset, this matching point will be assigned to that subset, as shown in the point in purple circle in Fig. 3(c), whose outgoing edge length of is closer to the subset below, so it is assigned to the subset below.

(c) Lastly, there may be matches that do not belong to any subset, which we refer to as discrete match. These are the three matching points in the lower right corner of Fig. 3(c–d). Unlike the matching points in subsets, these discrete matches cannot be removed using the sampling methods. Therefore, a special method for them is introduced in Sect. 3.3.

3.2 Topology constraint-based mismatching learning

As feature matches have been divided into subsets, in each subset, the feature matches will more likely present the same object, conform to the same transformation, and will preserve the topology relationship in two images.

In this subsection, we input the feature matches in a subset and construct the reconnected network in image Q, finally we calculate the mismatching probability for each feature match based on machine learning according to the principle of topology constraint, as shown in Fig. 4.

3.2.1 Topological constraint construct

According to the geometry of camera model and triangles, the two projections of an object in space satisfy the same homography transformation, where their coordinate distances and

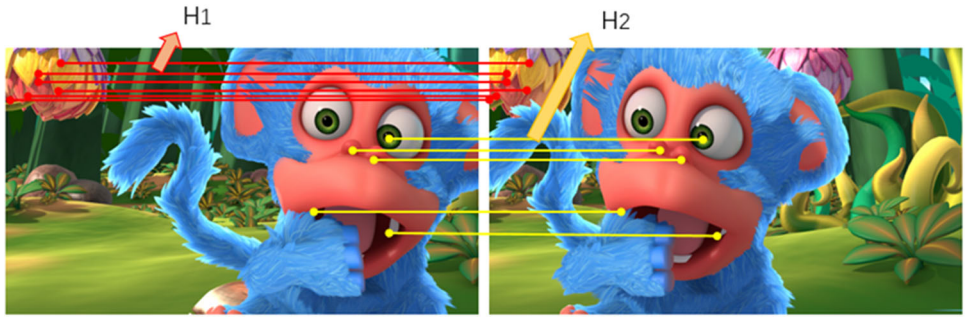

Fig. 2 An example of stereo vision, where objects at different depths correspond to different transformation models. The image is taken from the SceneFlow dataset [37]

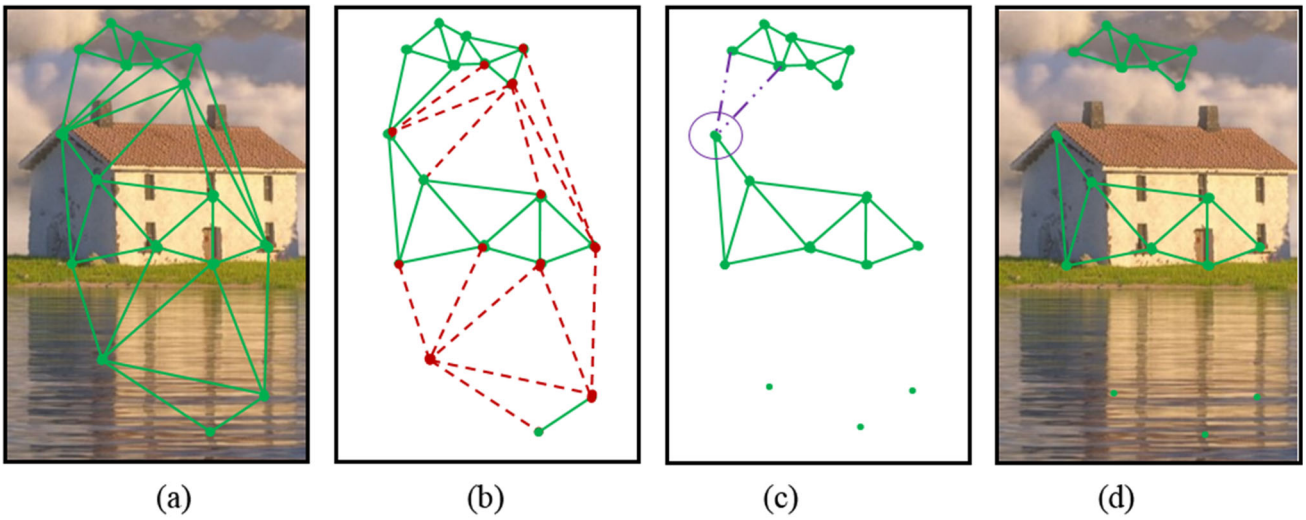

Fig. 3 An example of segmentation. **a** presents the triangular topology network in the image; **b** presents the result after removing the longer edges (colored in red); **c** and **d** shows the result of segmentation, the purple circle in **c** is a common point in two set and it assigned to the set below after judgment

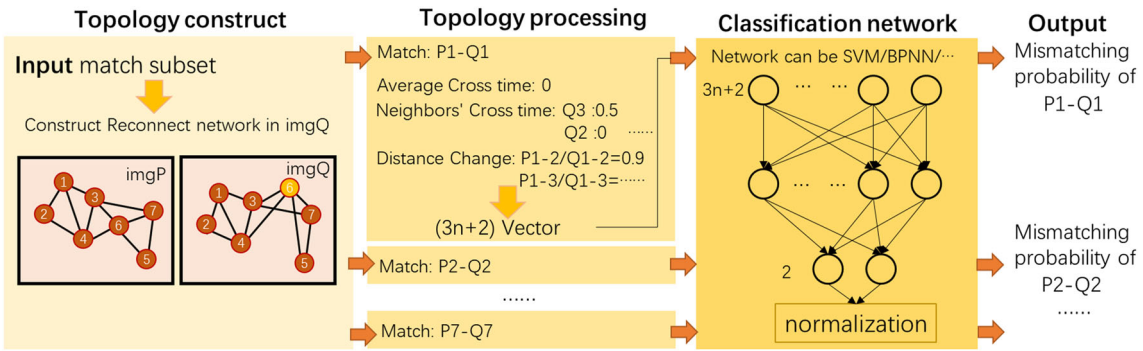

Fig. 4 Illustration of obtaining the probability of false matches

angles between the two images may change, but their topological relationships usually remain the same.

In the triangulated network, the topological relationship can be specifically expressed as follows: the inclusion relationship between a feature point and the triangle composed of the other three feature points remains unchanged. In other words, the feature point will remain inside or outside the triangle after transformation. If a point's inclusion relationship changes before and after transformation, it is an incorrect point, and its connection edge with one of the endpoints and the triangle edge will across, which cause a network distortion.

In the previous section, we have performed triangulation on the feature points in one image (image P). Then we reconnect the corresponded feature in another image (image Q)

similar to the connection in image P, which called reconnected work.

Therefore, if some feature points are not correctly matched, these points will cause distortion in the reconnected network. The orange points pair in Fig. 5 is a mismatch, it causes distortion in the reconnected network in image Q.

3.2.2 Topological constraint process for learning

The quantification of distortion in reconnected network is still a challenging task before calculate mismatching probability based on it.

In this paper, the distortion in the reconnected network is represented by the intersection times of the edges. If there is no intersection, it means that the network is essentially consistent, and all matches can be considered correct. If there are partial intersections in the network, it indicates the presence of some mismatches. Generally, if there are numerous intersections around a particular matching point, this match is more likely to be a mismatch. However, the number of intersections is not solely indicative of the probability of a mismatch. For instance, a point near the mismatched point, there are also number of intersections around which. Therefore, this paper considers more topological information, such as the intersection times of neighboring points and constraints of length, to represent distortion better. This is also the reason we do not directly rely on this for mismatch removal, but instead assigned them a mismatching probability.

For feature matches in subset input, let $P = \{P_1, P_2 \ldots\}$ be the set of feature points in image P, and let $Q = \{Q_1, Q_2 \ldots\}$ be the set of feature points in image Q, where $P_k - Q_k$ presents a feature match.

For each feature match $P_k - Q_k$, we use the following specific topological information to quantify the distortion condition of the network caused by it. Here n presents the number of neighboring points;

N: the total number of intersections on emanating edges of Q_k

A: the average number of intersections per emanating edge of Q_k

Ni: N of the i-th neighboring point

Ai: A of the i-th neighboring point

Ci: the distance change ratio, it can be calculated as follows:

$$Ci = |Q_i Q_k|/|P_i P_k| \tag{1}$$

Then, we can fit a function model f which makes values of above parameters for calculating mismatching probability for each match. It can be written as:

$$p = f(N, A, N_1, A_1, C_1 \ldots N_n, A_n, C_n) \tag{2}$$

We can easily get the dimensionality of the input parameters is $3n + 2$. For this function f, there is no common function or combination of functions that can represent it well. Therefore, we use machine learning to fit it as closely as possible to the actual mismatch situation and obtain mismatching probabilities.

3.2.3 Mismatching probability from machine learning

In this paper, we use classification network and a normalization to get mismatching probability. Here we define the labels of classification network as "correct" and "incorrect". It firstly calculates the probability of being "correct" p_c and being "incorrect" p_m, based on the topological information, then normalizes these two probabilities as follows to get mismatching probability.

$$p = p_m/(p_c + p_m) \tag{3}$$

The training process of this classification network is simple. We select various types of image pairs from dataset, perform feature extraction and match them. Then we label the each match based on the ground truth, which is created based on transform matrix provided by dataset; if the match is correct, we label this match as "correct", if it is a mismatch, it will be labeled as "incorrect". Then we calculate the topological information above for each match as in Sect. 3.2.2 and train the network.

In this paper, the classification network (we referred it as learning approach in following text) and the number of neighboring points for input (the value of n) are not specified; they have corresponding effects on the results. In this paper, we also conducted experiments on different machine learning approaches and different value of n in Sect. 4.

Compared with end-to-end networks, our method does not adopt images as inputs, but uses a more general topological information which exists in each pair of image matches instead. It greatly increases the generality of the method; moreover, the input parameters of our method are less than 20, so that it can achieve good results using simple machine learning methods.

Now, the mismatching probability of all matches in each subset has been calculated. For the matches that are not in any subset, i.e., the discrete matches, their mismatching probability is obtained after the mismatch removal of the matches in subsets, which will be explained in detail in Sect. 3.3.2.

3.3 Mismatch removal

In the previous section, feature matching has been divided into multiple subsets and a small number of discrete matches. Furthermore, each match in subset contains a mismatching probability of obtained above. Here we perform mismatch

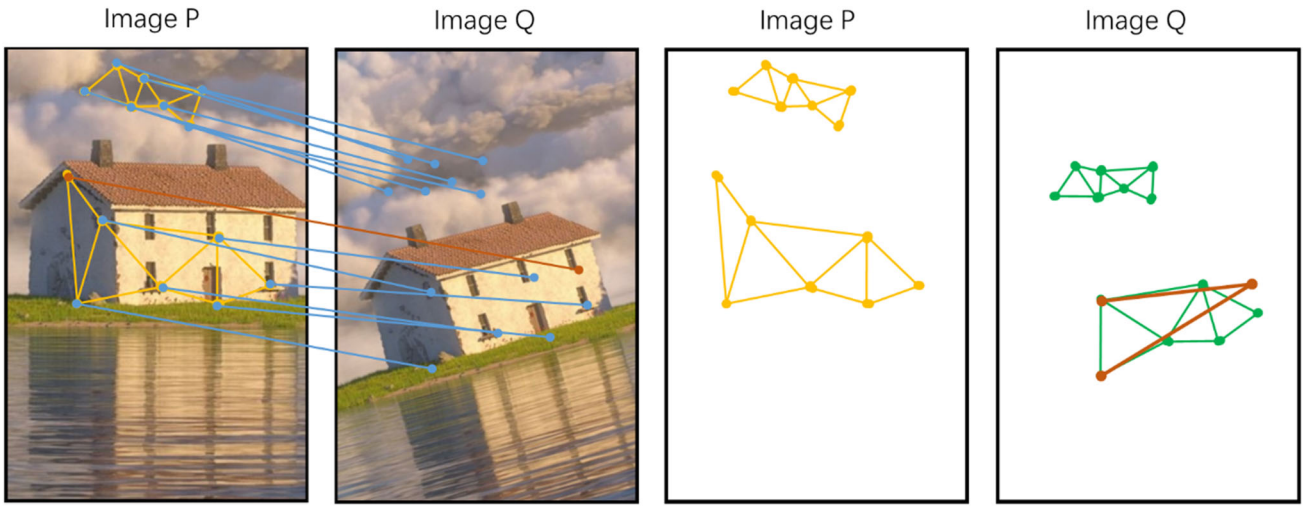

Fig. 5 A mismatch (colored in orange, the correct match colored in blue) causes some distortion in the reconnection network in image Q

removal through probability sampling for matches in subsets and use geometric topological constraints to remove discrete points.

3.3.1 Mismatch removal in subset

To improve the performance of the method, we introduce the idea of sampling, especially in case of large number of matches, sampling-based method can enhance the efficiency.

The random sample consensus algorithm (RANSAC) is a typical sampling-based approach for removing mismatches. RANSAC calculates a model by sampling a small amount of data and then fits the model to all data points, computing the number of data points that conform to the model. Finally, the model that conforms to the most data points is considered as the optimal model. In the application of mismatch removal, the model to be computed is the transformation matrix between two images. When the main content of the image lies on a plane, the homography matrix H (perspective transformation matrix) is usually used as the transformation matrix, which assumes that the points on the two images conform to the homography transformation (perspective transformation).

Specifically, the algorithm randomly samples four pairs of matched points and computes the transformation matrix. The points on the image should satisfy Eq. 4.

$$x' F x^T = 0 or x' = Hx \tag{4}$$

In practical situations, taking into account other factors such as camera distortion, rounding errors of pixels, etc., it is generally considered to meet Eq. 5.

$$x' F x^T < \text{threshold or } \left| x' - Hx \right| < \text{threshold} \tag{5}$$

After calculating the transformation matrix, if a pair of matching points satisfies the above conditions, the correspond match is considered correct. If not, it will be removed. However, due to its strong randomness, RANSAC is difficult to converge in situations with a large number of mismatches.

Therefore, in each subset, we can assign a sampling probability, *ps*, for each pair of matches based on the mismatch probability *p*, which calculated in Sect. 3.2. The relationship between the two can be expressed by Eq. 6. In other words, the higher the correct probability *p*, the higher the probability of being sampled.

$$ps_i = p_i / \sum_j^n p_j \tag{6}$$

After obtaining the sampling probabilities *ps*, we sample matches according to which. Four pairs of matches are sampled each time, and the homography matrix H, between image P and Q is calculated. Similarly, the sampling operation is performed for each subset, and each subset can obtain its own homography matrix.

When homography matrix between two subsets is fundamental similar, it can be inferred that the two subsets are likely to be matched points of a same object or that the matched points on the two subsets are essentially on a same plane. Therefore, it can be deduced that they should conform to a same homographic transformation. Consequently, we merge these two subsets and repeat the procedure above.

Then, combining the transformation matrix obtained above, we can filter the inlier of the model (the matches that have a low reprojection error under this transformation matrix), those we consider to be the correct matches.

With probability sampling, the matches with less mismatching probability will be sampled more probably, thus

transformation model can be calculated more efficiently, and we can correct the mismatches according to this transformation matrix easily.

3.3.2 Mismatch removal for discrete matches

As for the discrete matches, which refer to the matches that are not divided into subsets, we consider two possibilities caused for. The first is that there are too few feature points on an object to form a subset. The second is that an object is too large that some points are too far away from other points, causing it to become discrete during segmentation.

However, due to the sparsity of the discrete matches, it is impossible to calculate the homography matrix solely based on them. Therefore, we need to find alternative methods to judge the correctness of the matches.

We consider several matching situations here. Firstly, when two images are obtained from the perspective transformation of an object, the relative position of the object compared to the surrounding objects does not change significantly, and the topological relationship between the matching points and the topology relationship around them will not change significantly. Secondly, when two frames of images are captured by a camera to record the position changes of objects in the scene, the interval between the two frames is generally not enough to cause a significant change in the relative position between the objects. Hence, the topological relationship between the feature points on the object and their surroundings will also not change significantly.

Therefore, we focus on the topological relationship between the matching and the neighboring matching points in the two images. We connect the matching points with other matching points in surrounding subset in one image and reconnected in another image. It should be noted that, to increase the correctness of the method, the subsets need to go through the mismatch removal in step 3.3.1 before connecting with the surrounding points to ensure the correctness of the neighborhood points. Then, similar to 3.2, we count the number of crossings and calculate the mismatching probability. If the probability of error matching is low, it is considered correct; otherwise, it will be removed.

4 Experiment

To evaluate and analyze the performance of our method, we conduct extensive experiments. Firstly, we measured the effect of different machine learning networks on the results of this method. Secondly, we use our LTM in feature matching tasks and compare it with other state-of-the-art methods, under different feature descriptors. Finally, we conduct the transform matrix estimations on large datasets to reliably evaluate our method.

Environment The experiments are performed on Windows 11 operating system with an Intel Core i5-12,400 processor and 16-GB RAM. All the algorithms in this paper are written in Python. And in all experiments, the correspondences are computed from the ORB and SIFT keypoints.

Parameters The threshold of reprojection error we used is 4 pixels, the matches with error larger than 4 pixels we consider mismatches. The max iteration times of sampling process is set to 200 (for TSAC, RANSAC, LTM). The training data for LMR is about 7500 matches (data provided by author), and for our LTM is 3,000 matches.

Datasets For qualitative comparison we used the database of SceneFlow [37]. For quantitative comparison, we used the database of Mikolajczyk [24], one of the most widely used database. The dataset contains 40 image pairs, and the image pairs in the dataset always obey homography, where ground truth homography matrixes are suppled. Also, we carried out the experiment on the dataset of HPatches [25], and hannover [26], which contains several scenes of images and their groundtruth transform matrix. In order to reduce the contingency of random processes, we carried out experiments for ten times on each pair of images.

Evaluation indicators The main indicators of the experiments are precision, recall, and F-score. We define the accuracy as the proportion of correct matches in the matches extracted by mismatch removal method; And Recall is defined as the proportion of correct matches after extraction in whole correct matches. Generally, different value of thresholds will lead to an increase or decrease in the precision and recall, which are negatively correlated. Therefore, F-score is usually used to measure the whole performance of the methods for mismatch removal, which is define as Eq. 7:

$$F - \text{score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

4.1 Experiments under different machine learning approaches

Here we consider the influence of using different machine learning methods to study topology constraints and calculate mismatching probability. We chose three widely used supervised learning techniques SVM, BPNN, DenseNet and Transformer and feature descriptors SIFT. For this subexperiment, we use the Mikolajczyk dataset involving all the 40 image pairs.

For the input of machine learning, we consider the number of neighborhood points n to be 2,4,6, so when building the network, we set the corresponding input dimension to be 8,14,20. The dimension of output layers is 2, which indicates 2 class (correct match and mismatch).

Table 1 The result of different learning approaches and the number of neighborhood points n

Approach	Proposed method with SVM			Proposed method with BPNN			Proposed method with DenseNet			Proposed method with Transformer		
N	2	4	6	2	4	6	2	4	6	2	4	6
Precision	0.862	0.886	0.892	0.899	0.908	0.912	0.904	0.912	0.913	0.900	0.912	0.914
Recall	0.855	0.868	0.916	0.918	0.935	0.944	0.923	0.937	0.944	0.922	0.939	0.944
F-score	0.858	0.877	0.904	0.908	0.921	0.928	0.913	0.924	0.928	0.912	0.925	0.928

For the training samples of machine learning, we only used 2700 matches from different images in the Mikolajczyk dataset. Then, we selected 1,000 matches from four sets of images in the HPatches dataset as the test set, and the experimental results are shown in Table 1.

From the result, it can also be seen that the performance using BPNN, DenseNet and Transformer is basically similar, while the performance of SVM is slightly worse. For the number of neighboring points n, the best performance is achieved when $n = 6$, but also comparable when $n = 4$, and it requires less data and has faster training efficiency. Therefore, we selected BPNN as the machine learning method with four neighboring points as input in the subsequent experiments 4.2 and 4.3.

4.2 Qualitative comparative experiment

For this sub-experiment, we test our method on several representative image pairs in different types of transformations, including affine (e.g., Fig. 6a), homography (e.g., Fig. 6b–c), binocular stereo vision from SceneFlow (e.g., Fig. 6d–e). We present some intuitive results on the matching performance of our LTM and also compared with other state-of-art methods.

From the results above, we can see that there are very few matches are misjudged on all of test image pairs, which indicates that our method has a strong ability on mismatch removal under different types of transformations.

Here we also test our method on the images with higher ratio of mismatches. Here we exhibit some examples of this case. Figure 7(a–b) contains repetitive textures (Squares in 7(a) and circles in 7(b)), the feature descriptor of textures are nearly similar, which causes lots of mismatch between them. Figure 7c contains large viewpoints transform and Fig. 7(d) contains large illumination change, where feature descriptor of the same object will be different inevitably more or less, it is hard to match them correctly. Figure 7(e) contains similar element composition, the image pairs are composed of sketch lines, in this case, the feature descriptors may focus more on features of lines (such as crossing angles) and ignore the image features reflected by lines. From the images, we can find that the proposed method exhibits superior adaptability toward images with high mismatch ratio.

However, in some cases, our method may still produce some incorrect matches. As shown by the red matches in Fig. 8, after the mismatch removal process in this paper, some mismatches persist. It is evident that this is not caused by errors from calculating transform matrix. Instead, it is a discrete match, but due to the large distance to its neighboring points, our computed topological information cannot effectively reflect its correctness.

4.3 Quantitative comparison on dataset

To provide quantitative comparisons with state-of-the-art competitors, we conduct experiments on three datasets, Mikolajczyk [24], HPatches [25], and hannover [26].

Here we compare the method proposed with the geometry-based state-of-the-art methods such as LPM proposed by Ma [16], GTM proposed by Wendy [14], GMS proposed by Bian [15], resampling-based method RANSAC, resampling and geometry combined method TASC[23], leaning-based method OANet[32], and LMR[36].

Here we use table of average value on three indicators and cumulative distribution curves to reflect the characteristics of each method in the dataset. As for cumulative distribution curves, the faster the corresponding curve rises means that the method has better performance in the dataset. In the cumulative distribution curve graph, for a method, the smaller the horizontal ordinate (cumulative proportion) under a certain vertical ordinate (value of the indicator), the larger proportion of images have reached the value of the indicator.

In order to reduce the contingency of random processes, we carried out experiments for ten times on each pair of images. The result of each dataset is shown in Fig. 9 and Tables 2, 3 and 4.

From the results, we can observe that GMS performs not so accurate, while RANSAC has a better performance on precision but tends to have a low recall, GTM performs not so well, LPM always performs well on recall especially in change of viewpoints, but the precision is lower than the resampling-based method. For our previous research TSAC, it performs little better than LPM.

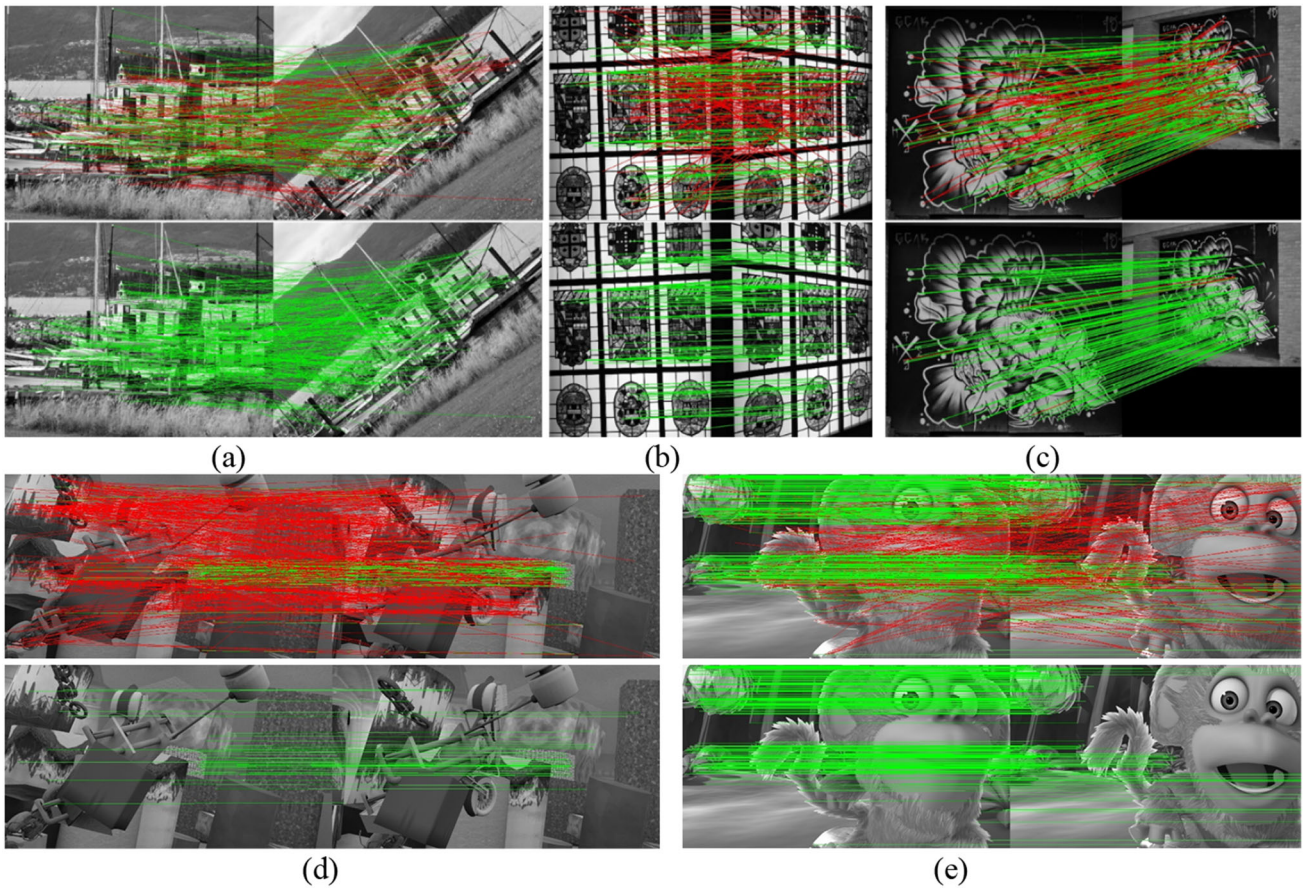

Fig. 6 The result of matching. In each image pair, the image above presents the initial matches, and the image below shows the result of our LTM. The matches colored in green presents correct matches while red for mismatches. Image pairs in **a**–**e** present different transformations, **a** for affine, **b**–**c** for homography, **d**–**e** for binocular stereo vision

As for learning-based methods, LMR, OANet, and our LTM, they improve the performance on all these datasets, especially in HPatches. It can also be found that they have great enhancement on recall. Compared with LMR and OANet, our LTM shows more accurate, and the performance of it tends to declines more slowly when outlier ratio increases in image pairs.

In order to test the adaptability of our method in high mismatching environments, we selected some challenging images from three datasets for testing. The selected images have a low inlier rate of less than 30%, with an average inlier rate of 18% and a minimum inlier rate of 6%. The results are shown in Fig. 10 and Table 5.

In the comparative analysis, LMR, TSAC, OANet, and LPM demonstrate superior performance, while the remaining methods struggle to function properly in high mismatching ratio environments. Among them, LTM and LMR exhibit higher recall rates, but LMR has lower precision compared to LTM. TSAC and LPM achieve similar results, with TSAC perform better in precision. Overall, the proposed method LTM exhibits the best performance.

5 Conclusion

In this paper, a robust method LTM for mismatch removal is proposed. It segments the matches to subsets and discrete matches, and calculates the mismatch probability by machine learning on the basis of topology constraint, and then inputs the probability into process of the sampling so that the mismatches can be detected and removed more easily and efficiently. It is proven by the experiment that it has high accuracy and a good adaptability to high mismatching ratio conditions.

The proposed method LTM has achieved good results as a geometry and learning combined method, with few parameters as input and only few images needed to train, which shows the method learning from geometry constraint has a huge potential in the fields of mismatch removal.

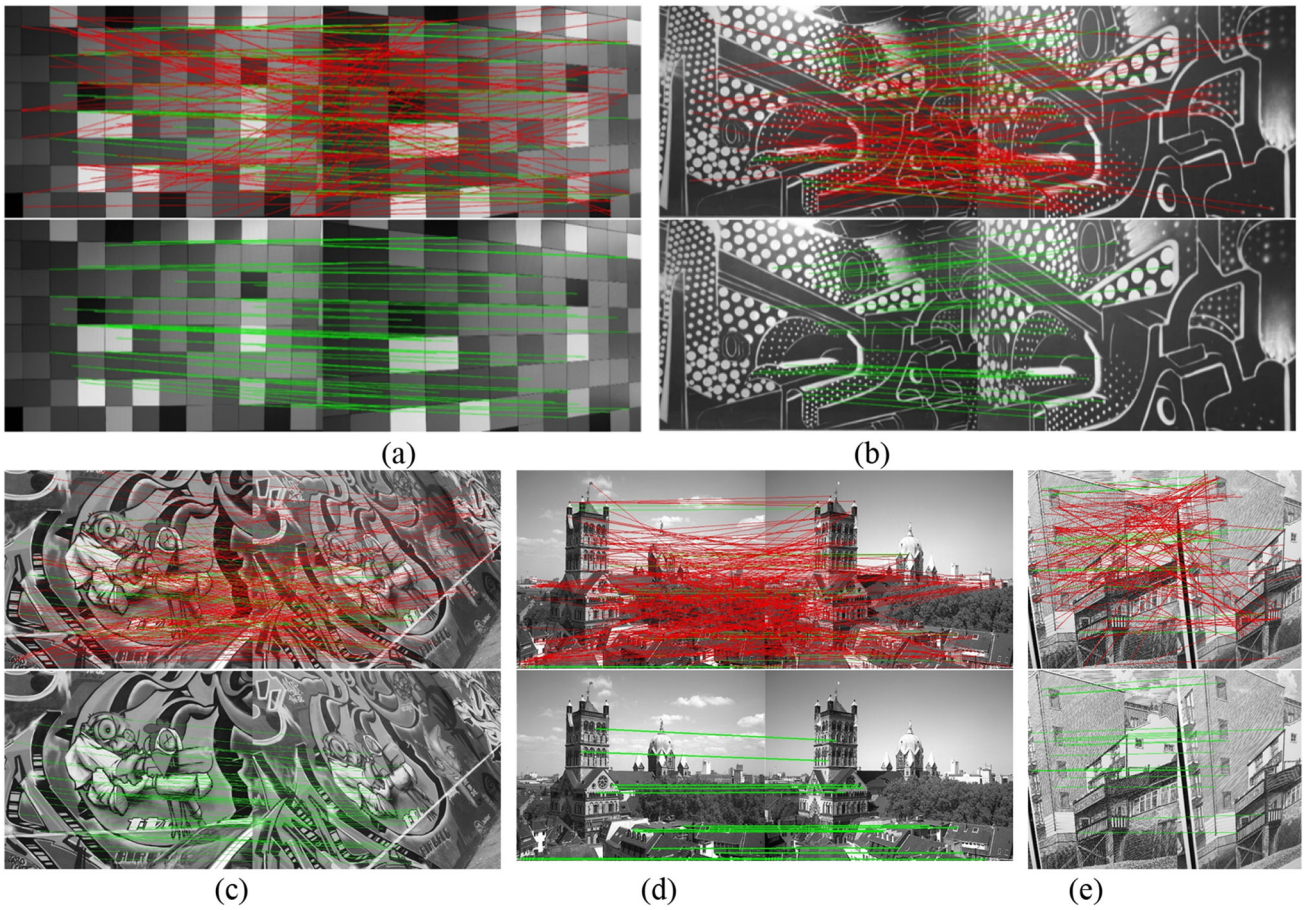

Fig. 7 The result of matching in high mismatching ratio. In each image pair, the image above presents the initial matches, and the image below shows the result of our LTM. The matches colored in green presents correct matches while red for mismatches. Image pairs in **a**–**e** presents different cases, **a**–**b** for repetitive texture, **c** for large viewpoints transform, **d** for large illumination change, **e** for similar element composition (sketch lines in the image)

Fig. 8 An example that the proposed method cannot deal with. In each image pair, the image above presents the initial matches, and the image below shows the result of our LTM

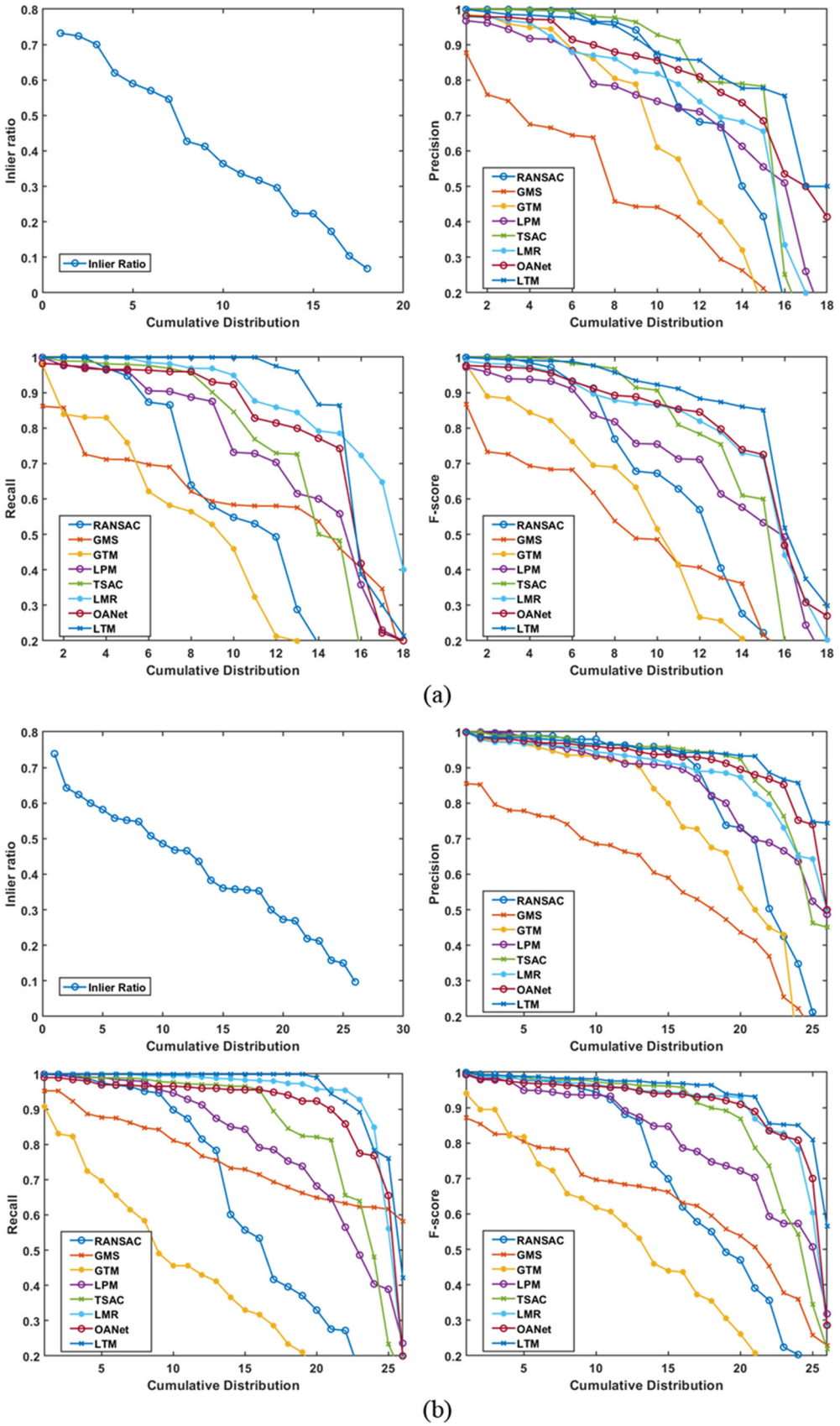

Fig. 9 The experimental results of the dataset. The subfigures **a**–**c** present results of the dataset hannover [26], Mikolajczyk [24], and Hpatches [25], each consists of the cumulative distribution graph of the inlier rate, precision, recall, and F-score in the dataset

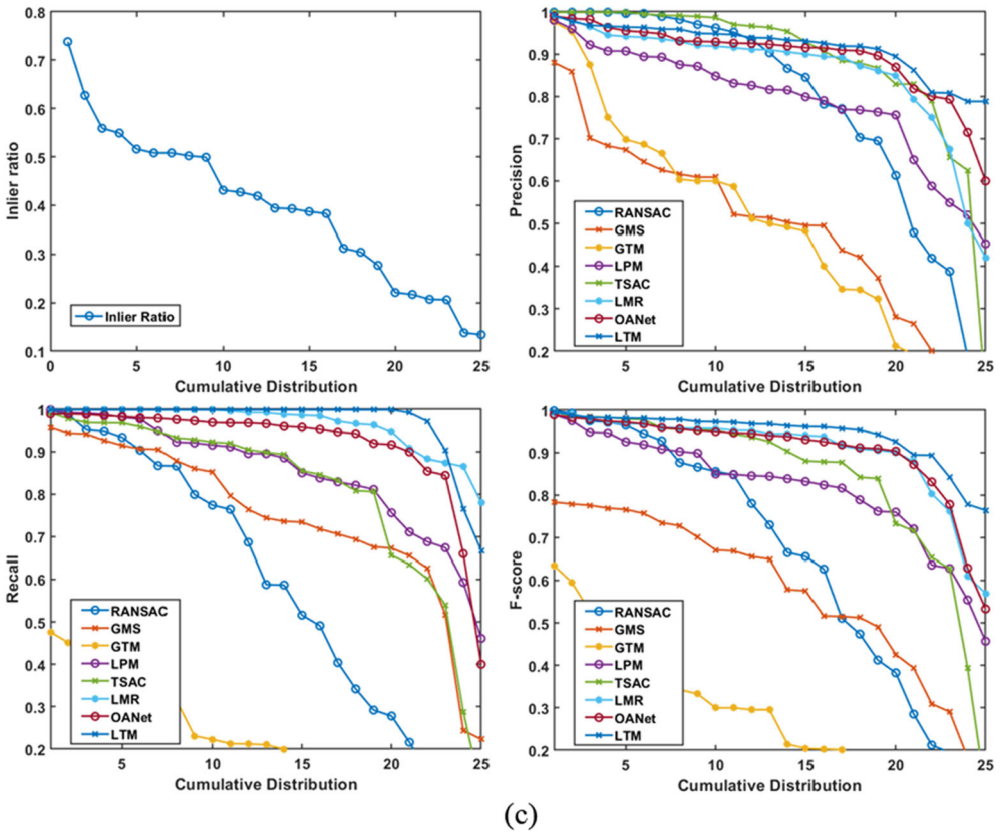

Fig. 9 continued

Table 2 The result of Mikolajczyk datasets

inlier ratio	Indicators	RANSAC	GMS	GTM	LPM	TSAC	LMR	OANet	LTM
0.412	precision	0.818	0.575	0.731	0.851	0.899	0.880	0.910	**0.936**
	recall	0.628	0.759	0.400	0.793	0.848	0.931	0.899	**0.950**
	F-score	0.692	0.633	0.489	0.806	0.863	0.901	0.901	**0.938**

Bold represents the best performance in each indicator among all of mismatch removal methods

Table 3 The result of Hpatches dataset

inlier ratio	Indicators	RANSAC	GMS	GTM	LPM	TSAC	LMR	OANet	LTM
0.394	precision	0.728	0.495	0.491	0.790	0.884	0.860	0.896	**0.918**
	recall	0.587	0.744	0.233	0.850	0.818	0.963	0.919	**0.971**
	F-score	0.656	0.576	0.296	0.817	0.845	0.905	0.903	**0.942**

Bold represents the best performance in each indicator among all of mismatch removal methods

Table 4 The result of hannover dataset

inlier ratio	Indicators	RANSAC	GMS	GTM	LPM	TSAC	LMR	OANet	LTM
0.413	precision	0.725	0.457	0.610	0.711	0.793	0.738	0.809	**0.859**
	recall	0.584	0.593	0.459	0.731	0.726	**0.876**	0.799	0.866
	F-score	0.632	0.489	0.451	0.713	0.754	0.788	0.797	**0.851**

Bold represents the best performance in each indicator among all of mismatch removal methods

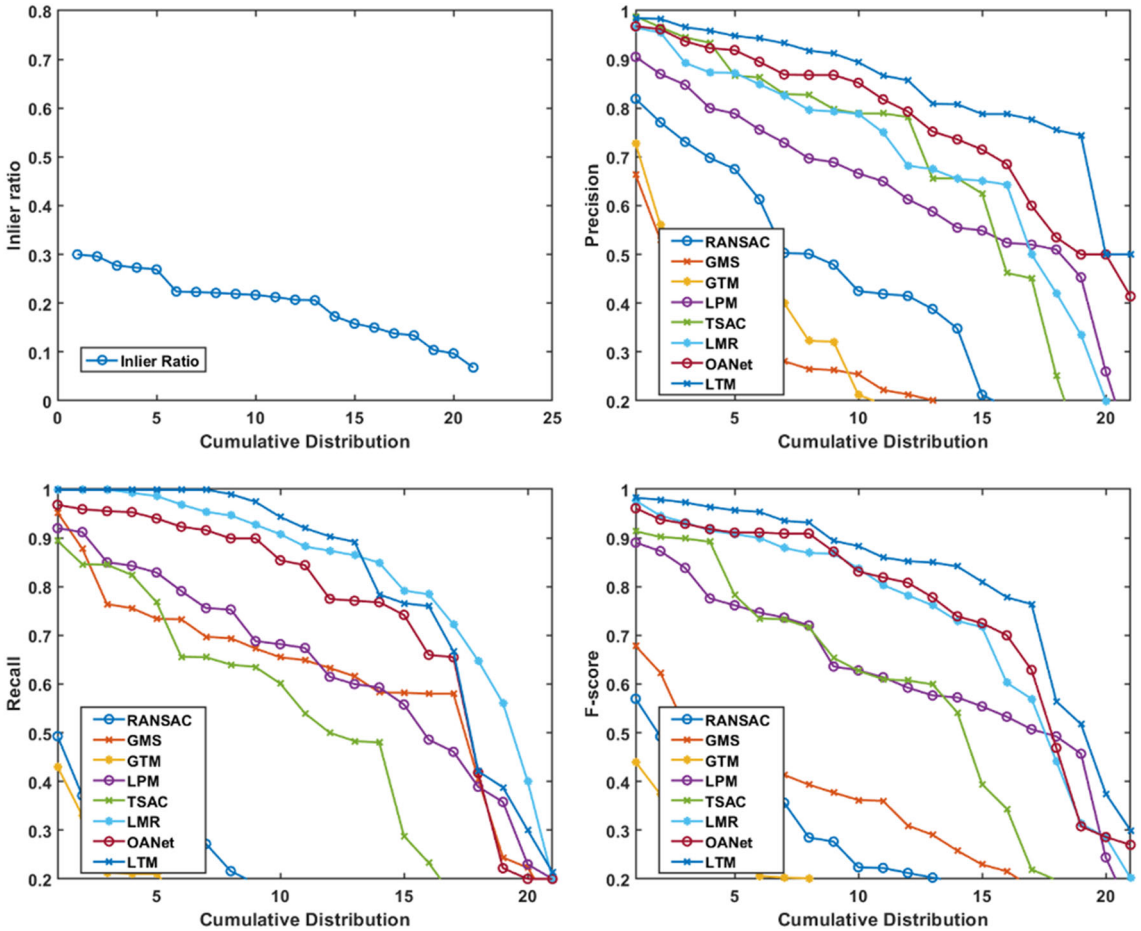

Fig. 10 The experimental results of high mismatching ratio, which consists of the cumulative distribution graph of the inlier rate, precision, recall, and F-score in the dataset

Table 5 The result of high mismatch ratio (mismatch ratio > 70%)

inlier ratio	Indicators	RANSAC	GMS	GTM	LPM	TSAC	LMR	OANet	LTM
0.176	precision	0.417	0.261	0.271	0.622	0.654	0.679	0.767	**0.840**
	recall	0.196	0.608	0.142	0.628	0.496	**0.822**	0.739	0.805
	F-score	0.260	0.346	0.172	0.614	0.552	0.726	0.744	**0.808**

Bold represents the best performance in each indicator among all of mismatch removal methods

Funding This work was supported by the National Natural Science Foundation of China (52275547 and 52275514) and the Zhejiang Provincial Natural Science Foundation of China (LY21E050021).

Data availability All datasets used in the work are publicly available and can be accessed as described in each of the referenced presentation papers.

Declarations

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

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94
2. Bay, H., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features. Springer-Verlag, Cham (2006)
3. C. Minchael, L. Vincent, S. Christoph, V. F. Pascal, BRIEF: binary robust independent elementary features. In: Proceedings of the 11th European Conference on Computer vision: Part IV Sept 2010 Pp. 778–792. https://doi.org/10.1007/978-3-642-15561-1_56.
4. Yi, K.M., Trulls, E., Lepetit, V., et al.: LIFT: learned invariant feature transform. Eur. Conf. Comput. Vis. (2016). https://doi.org/10.1007/978-3-319-46466-4_28
5. D. Detone, Malisiewicz T , Rabinovich A. SuperPoint: self-supervised interest point detection and description[C]// In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). https://doi.org/10.1109/CVPRW.2018.00060.
6. Richard, H., Andrew, Z.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981). https://doi.org/10.1145/358669.358692
8. Chum O, Matas J. Matching with PROSAC-progressive sample consensus. In: Proceeding of IEEE Computer Society Conference on Computer Society, 2005: 220–226. https://doi.org/10.1109/CVPR.2005.221
9. Ni Kai, Jin Hailing, Dellaert F. GroupSAC: efficient consensus in the presence of groupings. In: Proceeding of the 12th IEEE International Conference on Computer Vision. 2009:2193–2200. https://doi.org/10.1109/ICCV.2009.5459241
10. Chun O, Matas J. Randomized RANSAC with Td, d test. In: Proceeding of the 13th British Machine Vision Conference. Berlin: Springer, 2002: 448–457
11. Matas J, Chun O. Randomized RANSAC with sequential probability ratio test [C]. In: Proc of the 10th IEEE International Conference on Computer Vision (ICCV): IEEE Press, 2005: 1727–1732. https://doi.org/10.1109/ICCV.2005.198
12. M. Rahman, X. Li and X. Yin. DL-RANSAC: An improved RANSAC with modified sampling strategy based on the likelihood [C]. In: 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC). 2019: 463–468. https://doi.org/10.1109/ICIVC47709.2019.8981025
13. Shi C, Wang Y, Li H, Feature point matching using sequential evaluation on sample consensus. In: International Conference on Security. IEEE, 2017: 302–306. https://doi.org/10.1109/SPAC.2017.8304294
14. Aguilar, W., Fraud, Y., Escolano, F., et al.: A robust Graph Transformation Matching for non-rigid registration. Image Vis. Comput. **27**(7), 897–910 (2009)
15. Bian, J., Lin, W., Matsushita, Y., Yeung, S., Nguyen, T., Cheng, M.: GMS: grid-based motion statistics for fast, ultra-robust feature correspondence. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR). **2017**, 2828–2837 (2017). https://doi.org/10.1109/CVPR.2017.302
16. Ma, J., Zhao, J., Jiang, J., et al.: Locality preserving matching. Int. J. Comput. Vision **127**(5), 512–531 (2019). https://doi.org/10.1007/s11263-018-1117-z
17. Jiang, X., Xia, Y., Zhang, X.-P., Ma, J.: Robust image matching via local graph structure consensus. Pattern Recogn. **126**, 108588 (2022). https://doi.org/10.1016/j.patcog.2022.108588
18. Zhao, X., He, Z., et al.: Improved keypoint descriptors based on Delaunay triangulation for image matching. Optik-Int. J. Light Electron. **125**(13), 3121–3123 (2014). https://doi.org/10.1016/j.ijleo.2013.12.022
19. Luo Y, Li R, Zhang J, et al. Research on Correction Method of Local Feature Descriptor Mismatch. In: 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). https://doi.org/10.1109/IAEAC47372.2019.8997949
20. Zhao M, Chen H, Song T, et al. Research on image matching based on improved RANSAC-SIFT algorithm. International Conference on Optical Communications and Networks, pp 1–3 (2017). https://doi.org/10.1109/ICOCN.2017.8121270
21. Zhu, W., Sun, W., Wang, Y., Liu, S., Xu, K.: An improved RANSAC algorithm based on similar structure constraints. Int. Conf. Robot Intell. Syst. (ICRIS) **2016**, 94–98 (2016). https://doi.org/10.1109/ICRIS.2016.19
22. X. Lan, B. Guo, Z. Huang and S. Zhang, "An improved UAV aerial image mosaic algorithm based on GMS-RANSAC". In: 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), 2020, pp. 148–152. https://doi.org/10.1109/ICSIP49896.2020.9339283
23. He, Z., Shen, C., Wang, Q., Zhao, X., Jiang, H.: Mismatching removal for feature-point matching based on triangular topology probability sampling consensus. Remote Sens. **14**, 706 (2022). https://doi.org/10.3390/rs14030706
24. Mikolajczyk, K., Tuytelaars, T., Schmid, C., et al.: A comparison of affine region detectors. Int. J. Comput. Vision **65**(1), 43–72 (2005). https://doi.org/10.1007/s11263-005-3848-x
25. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: HPatches: a benchmark and evaluation of handcrafted and learned local descriptors. IEEE Conf. Comput. Vision Pattern Recog. (CVPR) **2017**, 3852–3861 (2017). https://doi.org/10.1109/ICSIP49896.2020.9339283
26. K. Cordes, B. Rosenhahn, and J. Ostermann. High-resolution feature evaluation benchmark. In: Proc. CAIP, pp. 327– 334, 2013
27. Jiang, X., Ma, J., Jiang, J., Guo, X.: Robust feature matching using spatial clustering with heavy outliers. IEEE Trans. Image Process. **29**, 736–746 (2019). https://doi.org/10.1109/TIP.2019.2934572
28. Shao, F., Liu, Z., An, J.: Feature matching based on minimum relative motion entropy for image registration. IEEE Trans. Geosci. Remote Sens. **60**, 1–12 (2022). https://doi.org/10.1109/TGRS.2021.3068185
29. L. Cavalli, V. Larsson, M. R. Oswald, T. Sattler and M. Pollefeys, "Handcrafted outlier detection revisited", In: Computer Vision–ECCV 2020: 16th European Conference, pp. 770–787, 2020
30. Yi KM, Trulls E, Ono Y, Lepetit V, Salzmann M, Fua P. Learning to find good correspondences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018 (pp. 2666-2674) https://doi.org/10.1109/CVPR.2018.00282
31. Sarlin P.-E., DeTone D., Malisiewicz T. and Rabinovich A., "SuperGlue: learning feature matching with graph neural networks". In:

Proceeding of IEEE/CVF Conference of Computer Vision and Pattern Recognition (CVPR), pp. 4938–4947, Jun. 2020. https://doi.org/10.1109/CVPR42600.2020.00499
32. J. Zhang et al., "OANet: Learning Two-View Correspondences and Geometry Using Order-Aware Network". In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 6, pp. 3110–3122, 1 June 2022. https://doi.org/10.1109/TPAMI.2020.3048013
33. Xia, Y., Ma, J.: Locality-guided global-preserving optimization for robust feature matching. IEEE Trans. Image Process. **31**, 5093–5108 (2022). https://doi.org/10.1109/TIP.2022.3192993
34. Xia, Y., Jiang, J., Yifan, Lu., Liu, W., Ma, J.: Robust feature matching via progressive smoothness consensus. ISPRS J. Photogr. Remote. Sens. **196**, 502–513 (2023)
35. Liu, X., Xiao, G., Chen, R., Ma, J.: PGFNet: preference-guided filtering network for two-view correspondence learning. IEEE Trans. Image Process. **32**, 1367–1378 (2023). https://doi.org/10.1109/TIP.2023.3242598
36. Ma, J., Jiang, X., Jiang, J., Zhao, J., Guo, X.: LMR: learning a two-class classifier for mismatch removal. IEEE Trans. Image Process. **28**(8), 4045–4059 (2019). https://doi.org/10.1109/TIP.2019.2906490
37. N. Mayer et al., "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 4040–4048. https://doi.org/10.1109/CVPR.2016.438
38. Liu, Z., An, J., Jing, Y.: A simple and robust feature point matching algorithm based on restricted spatial order constraints for aerial image registration. IEEE Trans. Geosci. Remote Sens. **50**(2), 514–527 (2012). https://doi.org/10.1109/TGRS.2011.2160645
39. Zhang, K., Li, X., Zhang, J.: A robust point-matching algorithm for remote sensing image registration. IEEE Geosci. Remote Sens. Lett. **11**(2), 469–473 (2014). https://doi.org/10.1109/LGRS.2013.2267771
40. Wu, Y., Ma, W., Gong, M., Su, L., Jiao, L.: A novel point-matching algorithm based on fast sample consensus for image registration. IEEE Geosci. Remote Sens. Lett. **12**(1), 43–47 (2015). https://doi.org/10.1109/LGRS.2014.2325970
41. Li, B., Ye, H.: RSCJ: robust sample consensus judging algorithm for remote sensing image registration. IEEE Geosci. Remote Sens. Lett. **9**(4), 574–578 (2012). https://doi.org/10.1109/LGRS.2011.2175434
42. H. Zhang et al., "Remote sensing image registration based on local affine constraint with circle descriptor." In: IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1–5, 2022, Art no. 8002205, https://doi.org/10.1109/LGRS.2020.3027096
43. Feng, R., Shen, H., Bai, J., Li, X.: Advances and opportunities in remote sensing image geometric registration: a systematic review of state-of-the-art approaches and future research directions. IEEE Geosci Remote Sens Magaz **9**(4), 120–142 (2021). https://doi.org/10.1109/MGRS.2021.3081763