ISPRS Journal of Photogrammetry and Remote Sensing 154 (2019) 28–40

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

LAM: Locality affine-invariant feature matching

Jiayuan Li[a,*], Qingwu Hu[a], Mingyao Ai[a,b]

[a] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[b] State Key Laboratory of Information Engineering, Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

ARTICLE INFO

Keywords:
Feature matching
Mismatch removal
Image registration
Nonrigid
Locality invariant matching
Local barycentric coordinate (LBC)

ABSTRACT

False match removal is a crucial and fundamental task in photogrammetry and computer vision. This paper proposes a robust and efficient mismatch-removal algorithm based on the concepts of local barycentric coordinate (LBC) and matching coordinate matrices (MCMs), called locality affine-invariant matching (LAM). LAM is suitable for both rigid and nonrigid image matching problems. We define a novel LBC system based on area ratios, which is invariant to local affine transformations. We also present the MCMs based on the coordinates of matches, whose degeneracy is able to indicate the correctness of correspondences. Our LAM method first builds a mathematical model based on the LBCs to extract good matches that preserve local neighborhood structures. Then, LAM constructs local MCMs using the extracted reliable correspondences and identifies the correctness for the remaining matches via minimizing the rank of the MCMs. LAM has linear space and linearithmic time complexities. Extensive experiments on both rigid and nonrigid real datasets demonstrate the power of the proposed method; i.e., LAM is more robust to complex transformations compared to other methods and is two orders of magnitude faster than RANSAC under low inlier rates. The source code of the proposed LAM method will be publicly available in http://www.escience.cn/people/lijiayuan/index.html.

1. Introduction

Mismatch removal has many applications in photogrammetry and computer vision, such as structure-from-motion (Wu, 2013), simultaneous localization and mapping (SLAM) (Mur-Artal et al., 2015), image retrieval (Murala et al., 2012), and panoramic image registration (Brown and Lowe, 2007). Its goal is to distinguish inliers from outliers in the initial correspondence set which is obtained by feature detection and description methods, e.g., the scale-invariant feature transform (SIFT) (Lowe, 2004).

Mismatch removal is still a very challenging problem, although a large number of methods have been proposed in the past few decades (Li et al., 2017a; Ma et al., 2015). It still suffers several difficulties. First, images captured under complex scene conditions may suffer from serious radiation and geometric distortions, which inevitably lead to a large proportion of outliers. Such high outlier ratios bring a great challenge to traditional methods, such as RANSAC-type algorithms (Chum and Matas, 2005; Fischler and Bolles, 1981; Torr and Zisserman, 2000). Second, the geometric transformations between image pairs are various. It is difficult to propose a general framework for both global transformations and complex nonrigid deformations. For example, traditional robust estimators and RANSAC-type methods are only suitable for global transformations. Third, real-time performance is rarely achieved, especially for nonrigid images. Nonrigid image matching methods, such as graph matching (Conte et al., 2004), usually have a very high computational complexity, which largely limits their usage in real-world applications.

In this paper, we propose a robust and efficient false match removal algorithm, called locality affine-invariant matching (LAM), to cope with the abovementioned difficulties. LAM is based on the observation that local structures inside an image are still preserved after nonrigid transformations. Namely, the relationships between small local regions inside an image pair can still be well modeled by affine or homography transformations. To make full use of the neighborhoods of a correspondence, we present a new local barycentric coordinate (LBC). The LBC is defined by normalized area ratios and is invariant to affine transformations. It is more suitable for feature matching tasks than Cartesian coordinates, since false matches should not have the same LBC, while correct matches may have the same LBC. In addition, the LBC is robust to nonrigid deformations. Hence, LAM first builds a mathematical model based on the LBC to extract good matches that preserve local neighborhood structures. The limitation of this stage is that it may miss some good matches with false neighborhood correspondences. Fortunately, we also define a matching coordinate matrices

* Corresponding author.
E-mail address: ljy_whu_2012@whu.edu.cn (J. Li).

https://doi.org/10.1016/j.isprsjprs.2019.05.006
Received 30 April 2019; Received in revised form 19 May 2019; Accepted 22 May 2019
Available online 04 June 2019

(MCMs) based on the coordinates of matches. LAM constructs local MCMs using the extracted reliable correspondences and identifies the correctness for the remaining matches by minimizing the rank of the MCMs. Our LAM method can be solved with linear space and linearithmic time complexities. Extensive experiments on real data demonstrate the power of the proposed LAM method, i.e., LAM is more robust than the compared state-of-the-art methods and is two orders of magnitude faster than RANSAC under low inlier rates (Fischler and Bolles, 1981).

There are two main contributions in our paper. First, we present a new LBC system and adapt it into a local structure-preserving mathematical model for robust feature matching. In contrast to traditional methods that rely on a global transformation, our LAM method is also suitable for complex nonrigid transformations. Second, we define a novel concept of the MCMs. Based on the rank deficient property of the MCMs, we can identify the correctness of the remaining matches. Hence, LAM is able to extract as many good matches as possible from the initial correspondence set.

2. Related work

We briefly review the outlier removal methods in feature matching. According to the types of deformations between image pairs, we roughly classify these methods into two groups, i.e., methods for rigid deformations (rigid methods) and methods for nonrigid deformations (nonrigid methods).

2.1. Rigid methods

Here, we regard global geometric transformations, such as similarity, affine, and perspective transformations, as rigid deformations. Traditional rigid methods include hypothesize-and-verify methods and robust estimator methods. Recently, deep learning-based methods have also shown their potential.

2.1.1. Hypothesize-and-verify techniques

The most popular hypothesize-and-verify method is the RANSAC method, which alternates between transformation estimation using minimum subset sampling and geometric model verification. The geometric model with the largest supporting correspondence set will be accepted as the optimal solution. RANSAC has many variants, such as MLESAC (Torr and Zisserman, 2000), LORANSAC (Chum et al., 2003), PROSAC (Chum and Matas, 2005), USAC (Raguram et al., 2013), and DSAC (Brachmann et al., 2017). The major limitations common to RANSAC-type methods are two-fold. On the one hand, RANSAC-type methods are sensitive to the outlier ratios as pointed out by (Li and Hu, 2010). On the other hand, RANSAC-type methods are no guarantee of the optimality of the estimated solutions (Chin and Suter, 2017).

2.1.2. Robust estimators

These methods treat a mismatch removal task as a robust regression problem. The advantages of these methods are the efficiency and a guarantee of an optimal solution. M-estimators (Huber, 1981; Maronna et al., 2006; Rousseeuw and Leroy, 1987) constitute a widely used robust regression technique. However, M-estimators inherently suffer from a breakdown point of 0.5. Namely, M-estimators will fail if the mismatch ratio is larger than 50%. Recently, an l_q estimator ($0 < q < 1$) (Li et al., 2016; Li et al., 2017c) was proposed for robust feature matching. The l_q estimator uses an l_q-norm instead of an l_2-norm in the cost function and optimizes the cost via the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). This method overcomes the limitation of M-estimators and greatly improves the breakdown point. Lin et al. (Lin et al., 2018) proposed a nonlinear regression technique called coherence-based decision boundaries (CODE), which is based on the observation that correct matches tend to be coherent while outliers are randomly scattered. As reported, CODE is still robust under 90% of outliers.

2.1.3. Deep learning methods

More recently, researchers have attempted to adapt deep learning techniques to geometric processing. Rocco et al. (2017) developed an architecture that performs in a bottom-up manner similar to the Hough voting technique. It uses early convolutional layers to generate candidate transformations, and adopts later layers to aggregate the votes. Yi et al. (2018) proposed an end-to-end architecture to label the initial match set as outliers or inliers. The loss function of the training network consists of two terms. One term is the classification loss, whose role is to reject outliers; another term is a regression loss, which can predict the essential matrix. However, this method requires knowledge of the intrinsics of images.

2.1.4. Other methods

Cai et al. (2018) proposed a novel deterministic optimization algorithm, called iterative biconvex optimization (IBCO), which performs a deterministic search on an initial consensus solution. In another work by the same research team (Chin et al., 2015), the maximum consensus problem is transformed into a tree-search problem. They integrated the A^* search algorithm into the framework of LP-type methods. The limitation common to these methods is that they are very slow.

2.2. Nonrigid methods

Unlike rigid methods, nonrigid methods are able to cope with more complex deformations, including both rigid deformations and nonrigid deformations. Graph matching methods, nonparametric interpolation methods, and local geometric prior-based methods are the three types of typical nonrigid methods.

2.2.1. Graph matching methods

These methods usually organize the two sets of feature points as graphs and minimize their structural distortions via an energy function (Conte et al., 2004). Several representative works include integer projected fixed-point (Leordeanu et al., 2009), tensor matching (Duchenne et al., 2011), reweighted random walk (Cho et al., 2010), minimum spanning tree induced triangulation (Lian et al., 2012), and max-pooling matching (Cho et al., 2014). Graph matching methods do not rely on the assumption that the nonrigid transformation obeys smooth and slow motion. Thus, these methods can achieve good performance even if an image pair undergoes multiple geometric modeling. However, graph matching is an NP-hard problem. Its time and space complexities are very large, which largely limits the permissible size of input graphs (Li et al., 2017a, 2017b).

2.2.2. Nonparametric interpolation methods

In many cases, the motion field of feature matches is smooth and slow. Hence, the nonrigid transformation can be approximately interpolated by a nonparametric function. Based on this observation, many effective methods have been developed, including vector field consensus (VFC) (Ma et al., 2013), Gaussian mixture models (GMM) (Jian and Vemuri, 2011), spatially constrained Gaussian fields (Wang et al., 2017), identifying correspondence function (ICF) (Li and Hu, 2010), and coherence point drift (CPD) (Myronenko and Song, 2010). However, if the motion field of point correspondences does not obey the “slow and smooth” principle, the performance of these methods will drop rapidly. In addition, they suffer from a similar limitation with graph matching, i.e., being time consuming. Hence, they are not suitable for real-time applications such as SLAM.

2.2.3. Local geometric prior-based methods

Li et al. (2017b) proposed a support-line voting strategy based on the neighborhoods of correspondences and used affine-invariant ratios to filter outliers. They also proposed a local region descriptor based on a

4-point local structure (Li et al., 2017a). These methods achieve very good performance since they consider both photometric and geometric properties inside a small local region. However, the consideration of photometric constraints inevitably increases the computational complexity. Ma et al. (2017) proposed a locality preserving matching (LPM) method based on the observation that the spatial distribution of the neighborhoods of a correct correspondence should be preserved. They gave a mathematical model and derived a closed-form solution. Bian et al. (2017) developed a grid-based motion statistics (GMS) method based on the piecewise smoothness assumption. The GMS method first divides images into small grids and calculates the number of neighborhood matches. Then, it uses the statistical likelihoods to distinguish inliers from outliers. Both the LPM and GMS methods are very efficient and suitable for real-time tasks. However, they only exploit a weak local geometric constraint, which makes it difficult to separate the true inliers from relatively low-precision noisy matches.

3. Methodology

3.1. Local barycentric coordinate (LBC)

Here, we first give the definition and properties of LBC. Suppose we are given a feature point p_1 and its neighborhood points p_2, p_3, p_4. Any three points of p_1, p_2, p_3, p_4 are not collinear (see Fig. 1). Thus, point p_1 can form three triangles $(S_{\Delta p_1p_2p_3}, S_{\Delta p_1p_2p_4}, S_{\Delta p_1p_3p_4})$ with p_2, p_3, p_4.

Definition 1 (*Local barycentric coordinate (LBC)*).

$$(\lambda_1, \lambda_2, \lambda_3) = \frac{1}{s_{sum}} \cdot \left(S_{\Delta p_1p_2p_3}, S_{\Delta p_1p_2p_4}, S_{\Delta p_1p_3p_4}\right) \tag{1}$$

are called the LBC of point p_1, where $s_{sum} = S_{\Delta p_1p_2p_3} + S_{\Delta p_1p_2p_4} + S_{\Delta p_1p_3p_4}$.

Theorem 1. *LBC is invariant under affine transformation. If points p_1, p_2, p_3, p_4 are transformed to q_1, q_2, q_3, q_4 by an affine transformation* **A**. *Points q_1, q_2, q_3, q_4 form triangles $(S_{\Delta q_1q_2q_3}, S_{\Delta q_1q_2q_4}, S_{\Delta q_1q_3q_4})$. Then, we have,*

$$(\lambda_1, \lambda_2, \lambda_3) = (\widetilde{\lambda}_1, \widetilde{\lambda}_2, \widetilde{\lambda}_3) \tag{2}$$

where $(\widetilde{\lambda}_1, \widetilde{\lambda}_2, \widetilde{\lambda}_3)$ are the LBC of point q_1.

Proof. As known, affine transformation has three invariants, i.e., parallel lines, ratios of lengths of parallel line segments, and ratios of areas. After affine transformation, areas are scaled by a factor of $\det(\boldsymbol{A}_{2\times2})$, thus,

$$\det(\boldsymbol{A}_{2\times2}) = \frac{S_{\Delta q_1q_2q_3}}{S_{\Delta p_1p_2p_3}} = \frac{S_{\Delta q_1q_2q_4}}{S_{\Delta p_1p_2p_4}} = \frac{S_{\Delta q_1q_3q_4}}{S_{\Delta p_1p_3p_4}} \tag{3}$$

According to the definition of LBC and Eq. (3), we have,

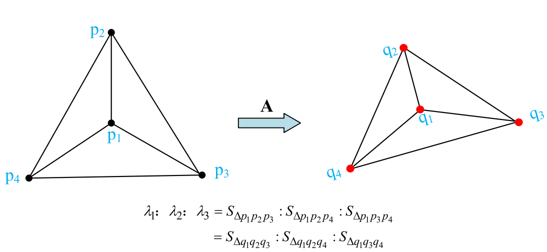


Fig. 1. The invariant of LBCs. Given a feature point p_1 and its neighborhood points p_2, p_3, p_4, p_1 forms three triangles with p_2, p_3, p_4, and we define the LBC of p_1 as the area ratios of these triangles. q_1, q_2, q_3, q_4 are the correspondences of p_1, p_2, p_3, p_4 after affine transformation $\boldsymbol{A}$. Therefore, the LBC of p_1 is equal to the LBC of q_1.

$$\begin{aligned}(\widetilde{\lambda}_1, \widetilde{\lambda}_2, \widetilde{\lambda}_3) &= \frac{1}{\widetilde{s}_{sum}} \cdot (S_{\Delta q_1q_2q_3}, S_{\Delta q_1q_2q_4}, S_{\Delta q_1q_3q_4})\\ &= \frac{\det(\mathbf{A}_{2\times2})}{\widetilde{s}_{sum}} \cdot (S_{\Delta p_1p_2p_3}, S_{\Delta p_1p_2p_4}, S_{\Delta p_1p_3p_4})\\ &= \frac{\det(\mathbf{A}_{2\times2})}{\det(\mathbf{A}_{2\times2}) \cdot s_{sum}} \cdot (S_{\Delta p_1p_2p_3}, S_{\Delta p_1p_2p_4}, S_{\Delta p_1p_3p_4})\\ &= (\lambda_1, \lambda_2, \lambda_3)\end{aligned} \tag{4}$$

$$\widetilde{s}_{sum} = S_{\Delta q_1q_2q_3} + S_{\Delta q_1q_2q_4} + S_{\Delta q_1q_3q_4}$$

LBC is very suitable for robust feature matching tasks, which is much superior to Cartesian coordinates. Because LBC is invariant to affine transformation, the two feature points of a correct match should have the same LBC. In contrast, the two features of a mismatch usually have different LBCs. In addition, LBC is a local coordinate, which is not sensitive to nonrigid transformations. Hence, LBC is suitable for both rigid and nonrigid image matching problems. Based on such properties, we can easily distinguish outliers from inliers.

3.2. Matching coordinate matrices (MCMs)

Given K point correspondences $\{(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)\}_1^K$, where $\boldsymbol{x}_i = [\mathrm{x_i}, \mathrm{y_i}]^{\mathrm{T}}$ and $\widetilde{\boldsymbol{x}}_i = [\widetilde{\mathrm{x}}_i, \widetilde{\mathrm{y}}_i]^{\mathrm{T}}$. The geometric relationship between these two point sets $\{\boldsymbol{x}_i\}_1^K$ and $\{\widetilde{\boldsymbol{x}}_i\}_1^K$ can be exactly modelled by an affine transformation **A**.

Definition 2 (*matching coordinate matrices (MCMs)*).

$$\mathbf{M}^{\mathrm{x}} = \begin{bmatrix}\widetilde{\mathbf{X}}^{\mathrm{x}}\\ \mathbf{X}^{\mathrm{x}}\\ \mathbf{X}^{\mathrm{y}}\\ \mathbf{1}\end{bmatrix} = \begin{bmatrix}\widetilde{\mathrm{x}}_1, \widetilde{\mathrm{x}}_2 \cdots, \widetilde{\mathrm{x}}_{\mathrm{k}}\\ \mathrm{x}_1, \mathrm{x}_2 \cdots, \mathrm{x}_{\mathrm{k}}\\ \mathrm{y}_1, \mathrm{y}_2 \cdots, \mathrm{y}_{\mathrm{k}}\\ 1, 1 \cdots, 1\end{bmatrix}, \quad \mathbf{M}^{\mathrm{y}} = \begin{bmatrix}\widetilde{\mathbf{X}}^{\mathrm{y}}\\ \mathbf{X}^{\mathrm{x}}\\ \mathbf{X}^{\mathrm{y}}\\ \mathbf{1}\end{bmatrix} = \begin{bmatrix}\widetilde{\mathrm{y}}_1, \widetilde{\mathrm{y}}_2 \cdots, \widetilde{\mathrm{y}}_{\mathrm{k}}\\ \mathrm{x}_1, \mathrm{x}_2 \cdots, \mathrm{x}_{\mathrm{k}}\\ \mathrm{y}_1, \mathrm{y}_2 \cdots, \mathrm{y}_{\mathrm{k}}\\ 1, 1 \cdots, 1\end{bmatrix} \tag{5}$$

$\mathbf{M}^{\mathrm{x}}$ and $\mathbf{M}^{\mathrm{y}}$ are called the MCMs.

Theorem 2. $\mathrm{rank}(\mathbf{M}^{\mathrm{x}}) \leqslant 3$, $\mathrm{rank}(\mathbf{M}^{\mathrm{y}}) \leqslant 3$; *and* $\det(\mathbf{M}^{\mathrm{x}}\mathbf{M}^{\mathrm{x,T}}) = 0$, $\det(\mathbf{M}^{\mathrm{y}}\mathbf{M}^{\mathrm{y,T}}) = 0$.

Proof. There exists an affine transformation $\mathbf{A} = (\mathrm{a_{ij}})_{2\times3}$ between $\{\boldsymbol{x}_i\}_1^K$ and $\{\widetilde{\boldsymbol{x}}_i\}_1^K$, namely,

$$\begin{cases}\widetilde{\mathrm{x}}_{\mathrm{i}} = \mathrm{a}_{11}\mathrm{x}_{\mathrm{i}} + \mathrm{a}_{12}\mathrm{y}_{\mathrm{i}} + \mathrm{a}_{13}\\ \widetilde{\mathrm{y}}_{\mathrm{i}} = \mathrm{a}_{21}\mathrm{x}_{\mathrm{i}} + \mathrm{a}_{22}\mathrm{y}_{\mathrm{i}} + \mathrm{a}_{23}\end{cases} \tag{6}$$

Then, writing the above 2 k equations in a vector form, we have,

$$\begin{cases}\mathbf{M}_1^{\mathrm{x}} = \mathrm{a}_{11}\mathbf{M}_2^{\mathrm{x}} + \mathrm{a}_{12}\mathbf{M}_3^{\mathrm{x}} + \mathrm{a}_{13}\mathbf{M}_4^{\mathrm{x}}\\ \mathbf{M}_1^{\mathrm{y}} = \mathrm{a}_{21}\mathbf{M}_2^{\mathrm{y}} + \mathrm{a}_{22}\mathbf{M}_3^{\mathrm{y}} + \mathrm{a}_{23}\mathbf{M}_4^{\mathrm{y}}\end{cases} \tag{7}$$

where $\mathbf{M}_i^{\mathrm{x}}$, $\mathbf{M}_i^{\mathrm{y}} (\mathrm{i} = 1, 2, 3, 4)$ are the i-th row of $\mathbf{M}^{\mathrm{x}}$ and $\mathbf{M}^{\mathrm{y}}$, respectively. As shown, the rows of $\mathbf{M}^{\mathrm{x}}$ and $\mathbf{M}^{\mathrm{y}}$ are linearly dependent. Thus, $\mathrm{rank}(\mathbf{M}^{\mathrm{x}}) \leqslant 3$, $\mathrm{rank}(\mathbf{M}^{\mathrm{y}}) \leqslant 3$; and $\det(\mathbf{M}^{\mathrm{x}}\mathbf{M}^{\mathrm{x,T}}) = 0$, $\det(\mathbf{M}^{\mathrm{y}}\mathbf{M}^{\mathrm{y,T}}) = 0$.

Suppose we have obtained the MCMs $\mathbf{M}^{\mathrm{x}}$, $\mathbf{M}^{\mathrm{y}}$ and a point $\boldsymbol{x}_j = [\mathrm{x_j}, \mathrm{y_j}]^{\mathrm{T}}$; now we show how to exactly predict the corresponding point $\widetilde{\boldsymbol{x}}_j = [\widetilde{\mathrm{x}}_{\mathrm{j}}, \widetilde{\mathrm{y}}_{\mathrm{j}}]^{\mathrm{T}}$ of $\boldsymbol{x}_j$. For $\mathbf{M}^{\mathrm{x}}$, it is easy to append columns, since each column is formed by the coordinates of one correspondence. Thus, appending $\boldsymbol{m}_j = [\widetilde{\mathrm{x}}_{\mathrm{j}}; \mathbf{x}_{\mathrm{j}}; 1]$ to $\mathbf{M}^{\mathrm{x}}$, we get a new MCM $\mathbf{M}_j^{\mathrm{x}} = [\mathbf{m}_{\mathrm{j}}, \mathbf{M}^{\mathrm{x}}]$. The rank of $\mathbf{M}_j^{\mathrm{x}}$ should also be smaller than 4 and the determinant of $\mathbf{M}_{\mathrm{j}}^{\mathrm{x}}\mathbf{M}_{\mathrm{j}}^{\mathrm{x,T}}$ should be zero, because there are no noises and outliers in the observations. In $\mathbf{M}_j^{\mathrm{x}}$, $\widetilde{\mathrm{x}}_{\mathrm{j}}$ is an unknown variable. If the value of $\widetilde{\mathrm{x}}_{\mathrm{j}}$ deviates from its ground truth value, matrix $\mathbf{M}_j^{\mathrm{x}}$ becomes full rank, which is reflected sensitively by $\det(\mathbf{M}_{\mathrm{j}}^{\mathrm{x}}\mathbf{M}_{\mathrm{j}}^{\mathrm{x,T}})$. However, real observation data usually contain noises, the determinant of $\mathbf{M}_j^{\mathrm{x}}\mathbf{M}_{\mathrm{j}}^{\mathrm{x,T}}$ cannot be zero. Fortunately, matrix $\mathbf{M}_{\mathrm{j}}^{\mathrm{x}}\mathbf{M}_{\mathrm{j}}^{\mathrm{x,T}}$ is a positive definite matrix, namely, $\det(\mathbf{M}_{\mathrm{j}}^{\mathrm{x}}\mathbf{M}_{\mathrm{j}}^{\mathrm{x,T}}) > 0$. Thus, we can estimate $\widetilde{\mathrm{x}}_{\mathrm{j}}$ by minimizing the following cost,

$$\widetilde{\mathrm{x}}_{\mathrm{j}}^{*} = \underset{\widetilde{\mathrm{x}}_{\mathrm{j}}}{\mathrm{argmin}} \det(\mathbf{M}_{\mathrm{j}}^{\mathrm{x}}\mathbf{M}_{\mathrm{j}}^{\mathrm{x,T}}) \tag{8}$$

where $\tilde{x}_j^*$ represents the optimal solution of $\tilde{x}_j$.

Let us first expand the term $\det(\mathbf{M}_j^x\mathbf{M}_j^{x,T})$ and obtain,

$$\det(\mathbf{M}_j^x\mathbf{M}_j^{x,T}) = \det(\mathbf{m}_j\mathbf{m}_j^T + \mathbf{M}^x\mathbf{M}^{x,T}) \tag{9}$$

Due to the noises in real data, $\mathbf{M}^x$ is also a full rank matrix and $\mathbf{M}^x\mathbf{M}^{x,T}$ is also positive definite. Therefore, we can find a 4×4 invertible matrix $\mathbf{V}^x$, which satisfies,

$$\mathbf{V}^x(\mathbf{M}^x\mathbf{M}^{x,T})\mathbf{V}^{x,T} = \mathbf{I} \tag{10}$$

Then, Eq. (9) becomes,

$$\det(\mathbf{M}_j^x\mathbf{M}_j^{x,T}) = \frac{\det((\mathbf{V}^x\mathbf{m}_j)(\mathbf{V}^x\mathbf{m}_j)^T + \mathbf{I})}{(\det(\mathbf{V}^x))^2} \tag{11}$$

Based on the theory of Sylvester's determinant identity, Eq. (11) can be reformulated as,

$$\begin{aligned}\det(\mathbf{M}_j^x\mathbf{M}_j^{x,T}) &= \frac{1}{(\det(\mathbf{V}^x))^2}\cdot[\det((\mathbf{V}^x\mathbf{m}_j)(\mathbf{V}^x\mathbf{m}_j)^T + \mathbf{I})]\\ &= \frac{1}{(\det(\mathbf{V}^x))^2}\cdot[\det((\mathbf{V}^x\mathbf{m}_j)^T(\mathbf{V}^x\mathbf{m}_j) + 1)]\\ &= \frac{1}{(\det(\mathbf{V}^x))^2}\cdot[\boldsymbol{m}_j^T\mathbf{V}^{x,T}\mathbf{V}^x\mathbf{m}_j + 1]\end{aligned} \tag{12}$$

From Eq. (10), we get $\mathbf{V}^{x,T}\mathbf{V}^x = (\mathbf{M}^x\mathbf{M}^{x,T})^{-1}$. Let,

$$\mathbf{V}^{x,T}\mathbf{V}^x = \mathbf{U}^x = (u_{ij}^x)_{4\times4} \tag{13}$$

Hence, minimization of $\det(\mathbf{M}_j^x\mathbf{M}_j^{x,T})$ is equal to minimize,

$$\begin{aligned}\min_{\tilde{x}_j}(\mathbf{m}_j^T\mathbf{V}^{x,T}\mathbf{V}^x\mathbf{m}_j) &= \min_{\tilde{x}_j}([\tilde{x}_j;\ \mathbf{x}_j;\ 1]^T\mathbf{U}^x[\tilde{x}_j;\ \mathbf{x}_j;\ 1])\\ &= \min_{\tilde{x}_j}(u_{11}^x\tilde{x}_j^2 + 2(u_{12}^x x_j + u_{13}^x y_j + u_{14}^x)\tilde{x}_j)\end{aligned} \tag{14}$$

It is a quadratic function of $\tilde{x}_j$ and its minimizer is,

$$\tilde{x}_j^* = \frac{-(u_{12}^x x_j + u_{13}^x y_j + u_{14}^x)}{u_{11}^x} \tag{15}$$

Similarly, the $\tilde{y}_j$ can be estimated by,

$$\tilde{y}_j^* = \frac{-(u_{12}^y x_j + u_{13}^y y_j + u_{14}^y)}{u_{11}^y} \tag{16}$$

Then, we can identify whether the correspondence is an inlier or an outlier by comparing $\tilde{\boldsymbol{x}}_j = [\tilde{x}_j,\tilde{y}_j]^T$ with $\tilde{\boldsymbol{x}}_j^* = [\tilde{x}_j^*,\tilde{y}_j^*]^T$.

3.3. Locality invariant matching

In this section, we develop a robust locality invariant matching method based on LBC and MCMs. Suppose we are given a set of initial feature correspondences $M = \{(\mathbf{x}_i,\tilde{\mathbf{x}}_i)\}_1^N$, where $\boldsymbol{x}_i = [x_i,y_i]^T$ and $\tilde{\boldsymbol{x}}_i = [\tilde{x}_i,\tilde{y}_i]^T$ are image coordinates in the reference image I_R and the target image I_T, respectively. The initial feature correspondences are usually extracted by the similarity of feature descriptors, such as the SIFT descriptor. Due to the influence of geometric and radiation distortions, the initial feature correspondence set M inevitably suffers from noises and outliers. Hence, our goal is to distinguish the inliers from the outliers in M and extract the inlier set I.

Traditional outlier removal methods, such as RANSAC-type methods and M-estimators, rely heavily on the global geometric model between the image pair (I_R, I_T). These methods may be effective for satellite images, whose elevation ranges are very small compared with the flight altitudes of the sensors. Thus, a global affine or homography transformation can well model the geometric relationship. However, with the development of sensors, many other kinds of images have become increasingly popular, such as unmanned aerial vehicle (UAV) images, panorama images, and oblique images. The viewpoint changes and scene elevation ranges of UAV images and oblique images are usually very large; panorama images usually contain serious geometric distortions. The geometric relationships between these types of images are more complex, which generally undergo nonrigid transformations and cannot be well modeled by global transformations. In these cases, traditional methods may get low *Precision* or *Recall* accuracies. Fortunately, the local structures inside an image are still preserved after non-rigid transformations. Namely, the relationship between small local regions inside the image pair (I_R, I_T) can still be well modeled by an affine or homography transformation. In the above, we have shown that the LBC is invariant to local affine transformations. In other words, the LBC is preserved under both rigid transformations and nonrigid transformations. Thus, the robust feature matching problem can be formulated as,

$$I^* = \arg\min_I C(I;\ M,\ \tau) = \arg\min_I\Big[\sum_{i\in I}\|\boldsymbol{x}_i^{lbc} - \tilde{\boldsymbol{x}}_i^{lbc}\|_2^2 + \tau(N-K)\Big] \tag{17}$$

where $C(I;\ M,\ \tau)$ is a cost function; $(\mathbf{x}_i^{lbc},\ \tilde{\mathbf{x}}_i^{lbc})$ is the LBC of $(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_i)$; $\|\cdot\|_2$ is the l_2-*norm* operator; τ is a balance parameter; K stands for the number of inliers in I; and I^* is the optimal solution of I. In this cost function $C(I;\ M,\ \tau)$, $\sum_{i\in I}\|\boldsymbol{x}_i^{nbc} - \tilde{\boldsymbol{x}}_i^{nbc}\|_2^2$ is a data term (called the LBC distance), which penalizes any point correspondences with large LBC distances; the term $(N-K)$ minimizes the number of outliers; and parameter τ balances these two terms. If the correspondences in I can be perfectly matched, the optimal solution will obtain zero data cost, namely, the data term will be zero.

In the above cost function, we only consider the inlier set I. To extend to the total initial correspondence set M, we introduce an $N \times 1$ binary vector $\boldsymbol{b} = \{b_i\}_1^N$ to assign a flag for each correspondence in M, where $b_i \in \{0,\ 1\}$. Specifically, if a correspondence $(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_i)$ is an inlier, its flag $b_i = 1$; otherwise, $b_i = 0$. Hence, the feature matching problem is finally converted to,

$$\begin{aligned}\boldsymbol{b}^* &= \arg\min_{\boldsymbol{b}} C(\boldsymbol{b};\ M,\ \tau)\\ &= \arg\min_{\boldsymbol{b}}\Big[\sum_{i=1}^N b_i(\|\boldsymbol{x}_i^{lbc} - \boldsymbol{x}_i^{lbc}\|_2^2) + \tau\Big(N - \sum_{i=1}^N b_i\Big)\Big]\\ &= \arg\min_{\boldsymbol{b}}\Big[\sum_{i=1}^N b_i(\|\boldsymbol{x}_i^{lbc} - \boldsymbol{x}_i^{lbc}\|_2^2 - \tau) + \tau N\Big]\\ &= \arg\min_{\boldsymbol{b}} \sum_{i=1}^N b_i(\|\boldsymbol{x}_i^{lbc} - \boldsymbol{x}_i^{lbc}\|_2^2 - \tau)\end{aligned} \tag{18}$$

Our goal is to find the optimal flag vector $\boldsymbol{b}^*$. In this cost function, $\|\boldsymbol{x}_i^{lbc} - \boldsymbol{x}_i^{lbc}\|_2^2$ is the LBC distance of the i-th correspondence $(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_i)$. Specifically, if $(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_i)$ is an inlier, the data term will be zero or a very small value; otherwise, the outlier will lead to a large cost.

As mentioned earlier, initial feature correspondences are given in advance by descriptor-based methods, which means that the neighborhood relationship of each point is fixed. Thus, for each correspondence $(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_i)$, we search its three neighboring correspondences and compute the LBC of $(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_i)$. Then, the LBC distance of each correspondence can be calculated. That is, in Eq. (18), the correspondence number N, the balance parameter τ, and the LBC distance $\|\boldsymbol{x}_i^{lbc} - \boldsymbol{x}_i^{lbc}\|_2^2$ are known values. The only unknown variable is $\boldsymbol{b} = \{b_i\}_1^N$. Clearly, correspondences with LBC distances less than τ lead to negative costs, which decrease the total energy function; in contrast, correspondences whose LBC distances are larger than τ will increase the total energy. Thus, the optimal solution $\boldsymbol{b}^*$ can be simply obtained by,

$$b_i = \begin{cases}1 & \|\boldsymbol{x}_i^{lbc} - \boldsymbol{x}_i^{lbc}\|_2^2 \leqslant \tau\\ 0 & \|\boldsymbol{x}_i^{lbc} - \boldsymbol{x}_i^{lbc}\|_2^2 > \tau,\end{cases} \quad i = 1,\ 2,\ \ldots,N \tag{19}$$

Once the binary vector $\boldsymbol{b}^*$ is determined, the optimal inlier set I^* is obtained simultaneously,

$$I^* = \{i|b_i = 1,\ i = 1,\ 2,\ \ldots,N\} \tag{20}$$

The LBC-based feature matching method is summarized in

Algorithm 1.

Algorithm 1: Feature matching based on the LBC

Input: Initial correspondences $M = \{(\mathbf{x}_i,\tilde{\mathbf{x}}_i)\}_1^N$ and parameter τ
Output: Optimal inlier set I^*

1 Search 3 neighborhoods for each correspondence $(\mathbf{x}_i,\tilde{\mathbf{x}}_i) \in \mathrm{M}$;
2 Convert Cartesian coordinates $\{(\mathbf{x}_i,\tilde{\mathbf{x}}_i)\}_1^N$ to LBCs $\{(\mathbf{x}_i^{lbc},\tilde{\mathbf{x}}_i^{lbc})\}_1^N$;
3 Calculate b_i based on Eq. (19);
4 Determine the inlier set I^* via Eq. (20).

Generally, **Algorithm 1** can achieve sufficiently good results. However, the algorithm still suffers two drawbacks: First, some inliers may be treated as outliers, which will decrease the number of inliers and the *Recall* performance. In our method, the neighborhood relationship of each point is fixed. If the neighborhoods of an inlier contain outliers, the calculated LBC will be not affine-invariant and will lead to a very large LBC distance. The large LBC distance, then, will classify the inlier as an outlier according to Eq. (19). Second, the obtained inliers are not quantitatively evaluated; therefore, it may be difficult to distinguish ground truth inliers from noises with relatively low position accuracy. The proposed method performs feature matching in the LBC system instead of the traditional Cartesian coordinate system. In the Cartesian coordinate system, the residual of a correspondence usually means its reprojection error, which measures how accurate the correspondence is. Specifically, correspondences with smaller residuals are closer to the ground truth. In contrast, correspondences with larger residuals deviate from their ground truth. However, the residual in the LBC system $\|\boldsymbol{x}_i^{lbc} - \boldsymbol{x}_i^{lbc}\|_2$ loses its physical meaning. Reprojection error metric is more straightforward to measure the matching position accuracy of a correspondence.

Fortunately, we have proposed the concept of MCM, which is suitable for dealing with such problems. Specifically, we first treat the matches I^* obtained by **Algorithm 1** as an inlier set and build a k-d tree (Indyk and Motwani, 1998) for efficient searching. Then, for each feature match $(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) \in (M - I^*)$, that is not extracted by **Algorithm 1**, we search its k nearest neighbors in the k-d tree and construct local MCMs $\mathbf{M}^x$ and $\mathbf{M}^y$. They are not full-rank matrices, and they approximately satisfy **Theorem 2**. Next, we can obtain the predicted correct matches $\tilde{\boldsymbol{x}}_i^*$ of $\boldsymbol{x}_i$ according to Eqs. (15) and (16). Finally, we calculate the residuals between $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{x}}_i^*$. Matches whose residuals are smaller than a predefined threshold ε are regarded as inliers. We add these reidentified inliers into I^* and obtain the final inlier set I_f^*. The proposed MCM-based correct match identification method is summarized in **Algorithm 2**. However, if the outlier rate of initial matches is extremely high (more than 80%), the performance of **Algorithm 1** and **Algorithm 2** may largely decrease. Fortunately, we can easily address this problem via a correspondence sampling strategy with matching scores (Li et al., 2016) or a guided strategy with smaller NNDR.

Algorithm 2: Match identification based on MCMs

Input: Optimal inlier set I^* and $M - I^*$
Output: Final inlier set I_f^*

1 Build a k-d tree for I^*;
2 For each match $(\boldsymbol{x}_i,\tilde{\boldsymbol{x}}_i) \in (M - I^*)$, search its k neighbors;
3 Construct the local MCMs based on these neighbors;
4 Predict the correct match $\tilde{\boldsymbol{x}}_i^*$ of $\boldsymbol{x}_i$ according to Eqs. (15) and (16);
5 Calculate the residuals between $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{x}}_i^*$, and identify the inlier.
6 Add inliers into I^* to get the final inlier set I_f^*.

3.4. Computational complexity

The proposed method consists of two main stages, which are summarized in **Algorithm 1** and **Algorithm 2**. In both algorithms, we use a k-d tree for efficient neighbor searching. Generally, the time complexity of the k-d tree search is linearithmic. In **Algorithm 1**, the time complexity of searching 3 neighbors is close to $\mathrm{O}((3 + \mathrm{N})\log \mathrm{N})$. Lines 2 and 3 have linear time complexity, which are close to $\mathrm{O}(\mathrm{N})$. The final line only involves a comparison operation, whose complexity is $\mathrm{O}(1)$. The time complexity of **Algorithm 1** is approximately $\mathrm{O}((3 + \mathrm{N})\log \mathrm{N} + \mathrm{N} + 1)$. In **Algorithm 2**, lines 1 and 2 search the k nearest neighbors and cost $\mathrm{O}((\mathrm{k} + (\mathrm{N} - |\mathrm{I}^*|))\log(\mathrm{N} - |\mathrm{I}^*|))$ complexity, where $|I^*|$ represents the number of matches in the inlier set I^*. Lines 3, 4, and 5 only involve some simple operations, such as addition, subtraction and multiplication. Their time cost is only $\mathrm{O}(\mathrm{N})$. The complexity of line 6 is $\mathrm{O}(1)$. The time complexity of **Algorithm 2** is approximately $\mathrm{O}((\mathrm{k} + (\mathrm{N} - |\mathrm{I}^*|))\log(\mathrm{N} - |\mathrm{I}^*|) + \mathrm{N} + 1)$. Thus, the total time complexity of the proposed method can be simplified as $\mathrm{O}(\mathrm{N}\log \mathrm{N})$. Both stages cost linear space complexity. In **Algorithm 1**, the storage of neighborhoods is $\mathrm{O}(3\mathrm{N})$; the storage of $\boldsymbol{b}^*$ is $\mathrm{O}(\mathrm{N})$; and the storage of I^* is $\mathrm{O}(|\mathrm{I}^*|)$. In **Algorithm 2**, the storage of neighborhoods is $\mathrm{O}(\mathrm{k}(\mathrm{N} - |\mathrm{I}^*|))$; the storage of I_f^* is $\mathrm{O}(|\mathrm{I}_f^*|)$. The total space complexity of LAM can be simplified as $\mathrm{O}(\mathrm{N})$. Hence, the proposed LAM algorithm has linear space and linearithmic time complexities, which are very suitable for real-time and large-scale applications compared with traditional methods.

4. Experiments and evaluations

In this section, we comprehensively study the performance of the proposed LAM algorithm on image datasets with both rigid and non-rigid transformations. We compare our LAM algorithm with eight other state-of-the-art approaches, i.e., RANSAC (Fischler and Bolles, 1981), MLESAC (Torr and Zisserman, 2000), LORANSAC (Chum et al., 2003), FastVFC (Ma et al., 2013), LLT (Ma et al., 2015), LPM (Ma et al., 2017), GMS (GMS + R&S represents the GMS that considers rotation and scaling, which increases the robustness to rotation changes while also largely increasing the computational complexity) (Bian et al., 2017), and IBCO (Cai et al., 2018). MLESAC uses an affine model while RANSAC, LORANSAC, and IBCO use a homography model. To make fair comparisons, we use the implementations of these algorithms from open source websites. There are three main parameters in the proposed LAM method: the LBC distance threshold τ, the number of nearest neighbors k, and the inlier threshold ε. We set $\tau = 0.05$, $k = 6$, and $\varepsilon = 3$. All the parameters are fixed throughout the following experiments. We use five metrics for quantitative evaluation, i.e., *Precision*, *Recall*, *Fscore*, mean absolute error (*MAE*), and root-mean-square error (*RMSE*). *Precision* reflects the proportion of inliers in the whole detected matches. *Recall* is the ratio of the detected inlier number and the ground truth inlier number. *Fscore* describes the overall accuracy that combines the metrics of *Precision* and *Recall*. The *Fscore* is computed as follows,

$$Fscore = \frac{2\mathrm{Precision}\cdot\mathrm{Recall}}{Precision + Recall} \tag{21}$$

The formulas of MAE and *RMSE* are,

$$\begin{cases} MAE = \frac{1}{N}\sum_{i=1}^{N}|v_i| \\ RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}v_i^2} \end{cases} \tag{22}$$

where v_i is the residual error of the i-th correspondence. The experiment settings, including parameters, datasets, initial feature matcher, methods for comparison, and evaluation metrics, are briefly summarized in Table 1. All the reported running time is calculated on a laptop with an Intel Core i7-8550U @ 1.8 GHz CPU, 8 GB of RAM.

4.1. Rigid feature matching

We use the Oxford dataset (Mikolajczyk and Schmid, 2005) and a satellite image dataset (shorted by satellite dataset) for this experiment.

Table 1
The details of experimental settings.

Settings	Information
Parameters (LAM)	LBC distance threshold: $\tau = 0.05$; Number of neighbors: $k = 6$; Inlier threshold: $\varepsilon = 3$.
Datasets	Rigid: (1) Oxford dataset, 40 image pairs; url: http://www.robots.ox.ac.uk/~vgg/research/affine/ (2) Satellite dataset, 10 image pairs; url: http://www.escience.cn/people/lijiayuan Nonrigid Nonrigid dataset, 10 image pairs. url: http://www.escience.cn/people/lijiayuan
Initial matcher	SIFT, Code: http://www.vlfeat.org/ ASIFT, Code: http://www.cmap.polytechnique.fr/~yu/research/ASIFT/
Methods	RANSAC, MATLAB: https://www.peterkovesi.com/matlabfns/index.html#robust MLESAC, MATLAB: http://www.math.unipa.it/fbellavia/htm/research.html LORANSAC, MATLAB: https://zhipengcai.github.io/ FastVFC, MATLAB: http://www.escience.cn/people/jiayima/index.html LLT, MATLAB: http://www.escience.cn/people/jiayima/index.html LPM, MATLAB&C++: http://www.escience.cn/people/jiayima/index.html GMS, C++: (without GPU setting) https://github.com/JiawangBian/GMS-Feature-Matcher IBCO, MATLAB: https://zhipengcai.github.io/ LAM, MATLAB: http://www.escience.cn/people/lijiayuan
Evaluation metrics	*Precision; Recall; Fscore; MAE; RMSE.*

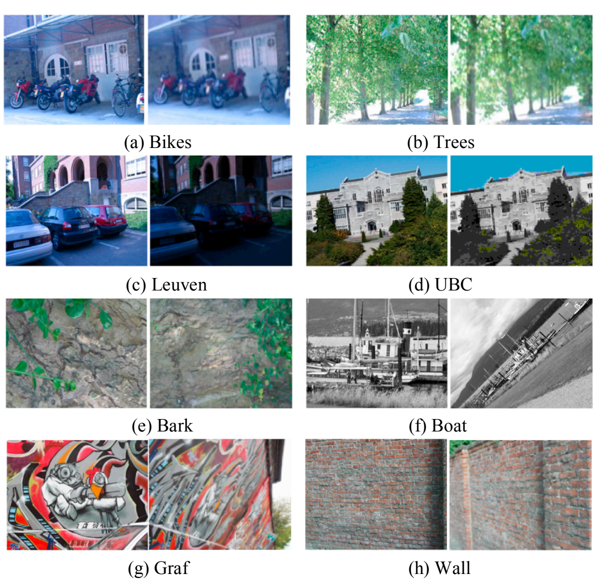


Fig. 2. Example images of Oxford dataset. Bikes and Trees, blur; Graf and Wall, viewpoint change; Bark and Boat, zoom and rotation; Leuven, illumination change; UBC, JPEG compression.

4.1.1. Comparisons on the Oxford dataset

The Oxford dataset contains eight categories, including Bikes, Trees, Leuven, UBC, Bark, Boat, Graf, and Wall (as shown in Fig. 2). Bikes and Trees suffer from blur changes; Leuven contains illumination variations; UBC suffers from JPEG compression artifacts; Bark and Boat contain zoom and rotation variations; Graf and Wall suffer from viewpoint changes. Each category consists of six images with increasing variations. The first image can form five image pairs with other images. Therefore, the Oxford dataset consists of a total of 40 image pairs. The ground truth homography transformations of these image pairs are also provided. We use the SIFT algorithm implemented by the VLFeat (Vedaldi and Fulkerson, 2010) toolbox to generate initial feature matches, where the nearest-neighbor distanceratio (NNDR) is set to 0.83. SIFT algorithm is sensitive to large viewpoint variations. It may fail to extract any correct correspondences on some image pairs, such as the fifth image pair of Graf. In these cases, we use the affine-SIFT (ASIFT) (Morel and Yu, 2009) algorithm to generate putative correspondences. For each image pair, we regard matches whose reprojection errors are smaller than $\varepsilon = 3$ pixels under the ground truth transformation as inliers.

Figs. 3-5 plot the *Precision*, *Recall*, and *Fscore* results on the Oxford dataset, respectively. From Fig. 3, we can see that: MLESAC, LORANSAC, and IBCO achieve the best *Precision* performance. Our LAM is comparable with them on the most of image pairs. The *Precision* of LAM is much higher than the other five methods. In the cases with large viewpoint changes (such as Graf and Wall categories), LAM is worse than MLESAC, LORANSAC, and IBCO. This indicates that the proposed LAM may be slightly sensitive to large viewpoint variations, because large viewpoint changes lead to serious projective distortions rather than affine distortions. Fortunately, the proposed LAM still achieves sufficiently good results in such cases. RANSAC, FastVFC, LLT, and GMS + R&S get less satisfactory *Precision* results than LAM. Their performance generally lies in the middle level among the nine compared methods. FastVFC and LLT are very sensitive to large projective distortions. For example, they obtain the lowest *Precision* accuracy on the Graf category. LPM performs the worst on the first four categories. GMS gets the lowest *Precisions* on the Bark category. According to the *Recall* comparison (Fig. 4), LLT is the best, with results close to 100%. FastVFC and LPM perform similarly to LLT. The proposed LAM ranks next. The worst result from LAM among these 40 pairs is still close to 90%. MLESAC is very sensitive to large viewpoint changes since it uses a global affine transformation model. Its *Recall* accuracy is only 20% on the Wall category. Compared with RANSAC, we find that affine transformations are more sensitive to complex geometric distortions than homography transformations. GMS is very sensitive to rotation and zoom changes. Its *Recall* accuracy is the lowest on the Bark and Boat categories. Fig. 5 gives the overall performance of each method. As can be seen, IBCO ranks best in terms of *Fscore*. Only several results do not attain the highest *Fscore.* The proposed LAM is comparable with IBCO. RANSAC, LORANSAC, FastVFC, and LLT achieve similar results, whose performances rank in the second group. MLESAC is comparable to the proposed LAM in most cases. However, MLESAC performs too poorly on the last two categories. Similar to the *Precision* performance comparison, the LPM and GMS perform the worst in all categories except for the Graf and Wall categories.

Table 2 reports the average *Precision*, *Recall*, *Fscore*, *MAE*, *RMSE*, and running time results. The *Precision* of the proposed LAM ranks second among all nine compared methods. The *Recall* of our method is higher than 95%, which is sufficient for photogrammetric applications. The average *Fscore* accuracies of RANSAC, MLESAC, LORANSAC, FastVFC, LLT, LPM, GMS, GMS + R&S, IBCO, and LAM are 90.63%, 81.15%, 93.09%, 91.97%, 91.48%, 87.49%, 69.52%, 76.30%, 94.25% and 93.75%, respectively. Our method ranks second, which achieves an 0.66% growth rate compared with LORANSAC which ranks third. MLESAC achieves comparable performance with the proposed method in most cases. However, the *Recall* of MLESAC is very sensitive to large viewpoint changes, which significantly decreases its overall performance. LORANSAC has the best *MAE* and *RMSE.* The proposed method is slightly worse than LORANSAC, which is much better than RANSAC, FastVFC, and GMS. Both the *MAE* and *RMSE* of LPM are the worst. LPM

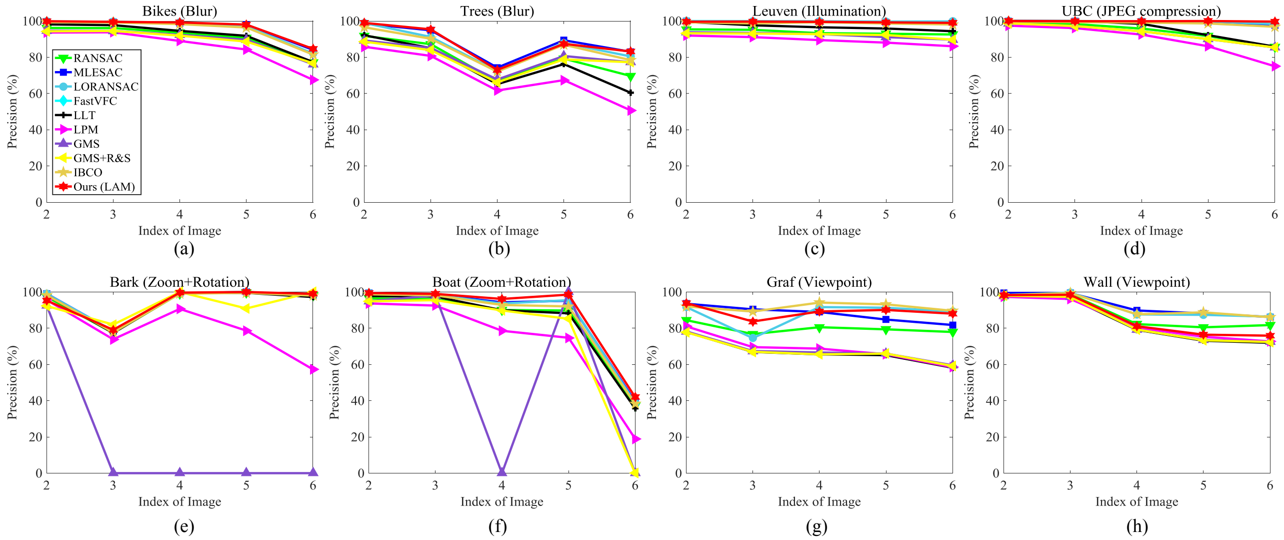


Fig. 3. Comparison of *Precision* on the Oxford dataset.

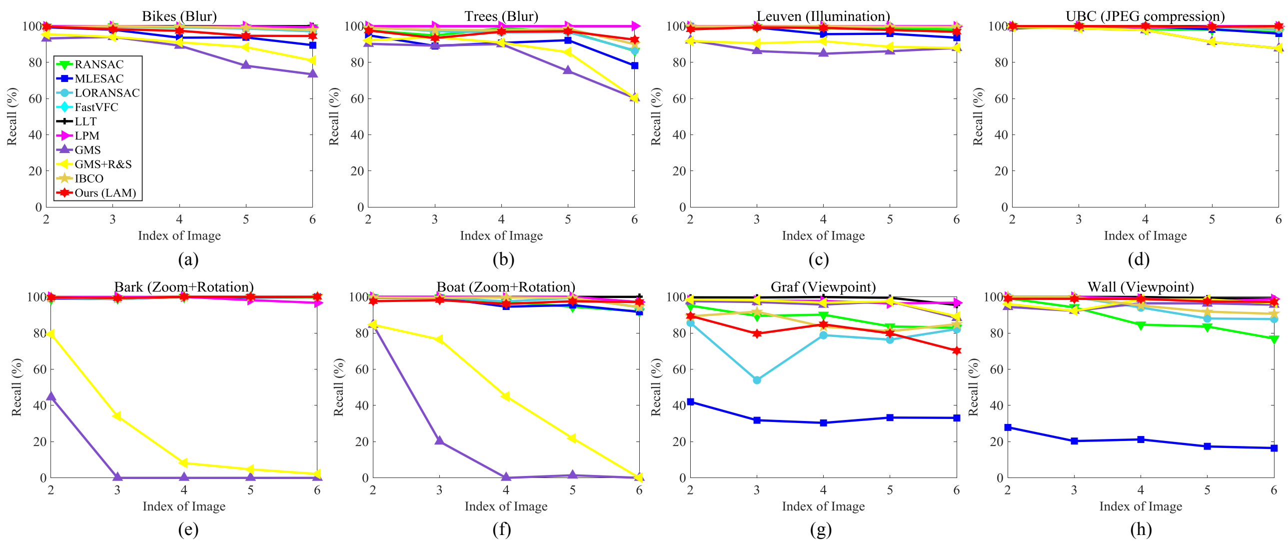


Fig. 4. Comparison of *Recall* on the Oxford dataset.

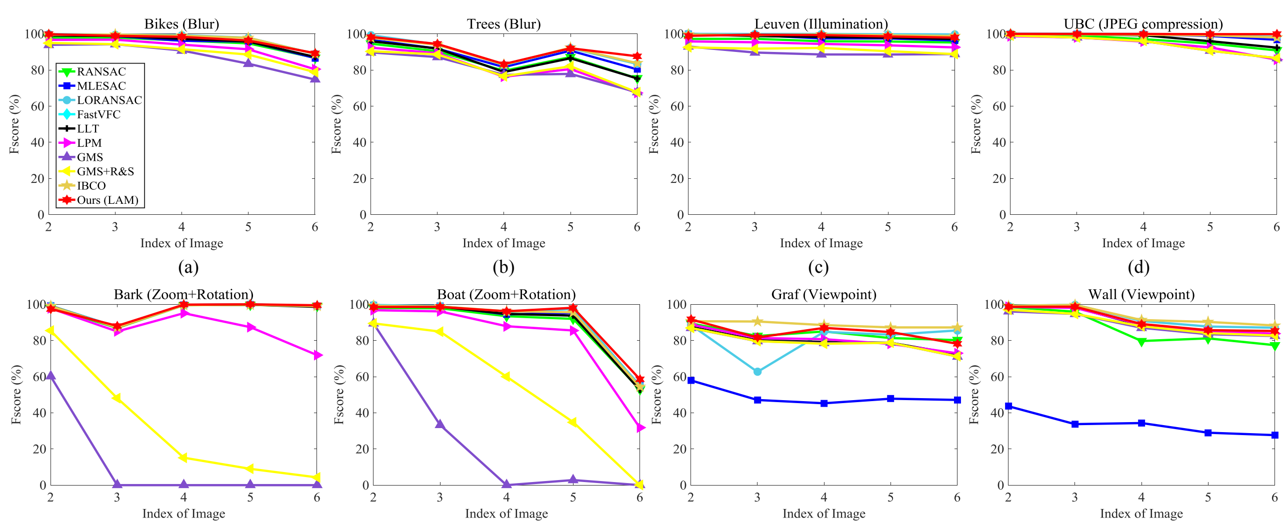


Fig. 5. Comparison of *Fscore* on the Oxford dataset.

Table 2
Performance comparison on the Oxford dataset.

Method	*Precision/%*	*Recall/%*	*Fscore/%*	*MAE/pixels*	*RMSE/pixels*	*Time/s*
RANSAC	87.67±0.39	95.25±1.21	90.63±0.74	1.68±0.02	2.15±0.03	0.79±0.41
MLESAC	92.93±0.26	78.76±0.33	81.15±0.25	1.45±0.01	1.73±0.02	0.60±0.04
LORANSAC	92.61±0.20	94.11±0.25	93.09±0.23	1.44±0.01	1.71±0.01	25.60±4.75
FastVFC	86.84±0.00	99.57±0.00	91.97±0.00	1.72±0.00	2.10±0.00	23.17±1.73
LLT	85.81±0.00	99.77±0.00	91.48±0.00	1.81±0.00	2.36±0.00	2.29±0.20
LPM	79.69±0.00	99.34±0.00	87.49±0.00	5.37±0.00	12.18±0.00	0.02±0.00
GMS	72.95±0.00	71.20±0.00	69.52±0.00	4.27±0.00	5.51±0.00	0.001±0.00
GMS+R&S	83.86±0.00	77.95±0.00	76.30±0.00	2.11±0.00	3.61±0.00	0.05±0.02
IBCO	92.45±0.22	96.94±0.23	94.25±0.19	1.45±0.01	1.73±0.02	120.67±23.16
Ours (LAM)	92.84±0.00	95.61±0.00	93.75±0.00	1.48±0.00	1.74±0.00	0.27±0.15

Numbers in red and blue represent the best and the second. We run the complete test 40 times, and each cell contains an empirical mean and a standard deviation of the results.

is only based on the observation that the spatial distribution of the neighborhoods of a correct correspondence should be preserved. This property is a local topological constraint that may be sensitive to noisy matches. For instance, matches with residuals larger than 3 pixels while smaller than 10 pixels may also obey the local topological constraint. These matches will be accepted as inliers by LPM. The basic idea of LPM is the closest to that of our LAM among the compared methods. However, our method performs much better than LPM. The reason may be that the proposed LBC is a more exact local geometric constraint compared with the spatial distribution of the neighborhoods. In terms of *Time*, GMS is the fastest in the table. However, its performance is the worst since it does not address rotation and scaling changes. GMS + R&S is much slower than GMS because GMS + R&S performs the standard GMS in 9 directions and 5 scales. As a result, the computational complexity of GMS + R&S is almost 45 times that of GMS. LPM ranks second. It is almost 10+ times faster than GMS + R&S, 30+ times faster than RANSAC and MLESAC, 100+ times faster than LLT, 1000+ times faster than FastVFC and LORANSAC, and 5000+ times faster than IBCO. First, the core algorithm of LPM is implemented by C++, while other methods (except for GMS and GMS + R&S) are all implemented by MATLAB. Second, the time complexity of LPM is linearithmic. In fact, the proposed LAM has the same time complexity as LPM. Both of them have $O(N \log N)$ complexity, which means that LAM will be as fast as LPM if we rewrite LAM by C++. Thus, our method is very suitable for real-time and large scale feature matching problems. The time complexity of FastVFC is $O(N^3)$. The time complexity of IBCO is polynomial to the size of initial matches. It is very time-consuming on the Graf and Wall categories, since the numbers of ASIFT initial matches are generally larger than 10000. The inlier rates of this dataset are very high (generally higher than 50%). Thus, RANSAC and MLESAC are not very time-consuming on this dataset. Even so, the proposed LAM is still two times faster than them. Although RANSAC-type methods and deterministic methods are slow in our comparison, they will regain a part of their investment later. They establish reliable correspondences and fit geometric models simultaneously, which can provide accurate initial values and let costly global bundle adjustment converge faster.

4.1.2. Comparisons on the Satellite dataset

The Satellite dataset contains 15 remote sensing image pairs, which are formed by multi-sensor, multi-temporal, or multi-spectral satellite/aerial images. The image pairs 8–15 are selected from the Erdas sample data[1]. Their image sizes are normalized to 1450 × 1380 pixels. The spectral mode, image size, ground sample distance (GSD), acquisition date, location, and description information of each image pair in this dataset are summarized in Table 3. As shown, the GSD of this dataset ranges from 0.5 m to 30 m, namely, the dataset contains both low- and high- resolution remote sensing images. Matching on the Satellite dataset is very challenging due to serious geometric and radiation distortions. For instance, multi-sensor and multi-spectral images suffer from significant radiation differences; the geometric distortions may be large in multi-temporal image pairs; and the overlapping regions of image pairs 8–15 are extremely small (smaller than 5% of the image width). For each image pair, an approximate ground-truth affine transformation is established. Specifically, we manually select six evenly distributed image matches with a location accuracy of 0.2 pixels; then, we treat these matches as control points and estimate an accurate affine transformation by least squares. The estimated transformation is accepted as the approximate ground-truth transformation. Again, we use SIFT to extract initial correspondences, and correspondences with residual errors smaller than 3 pixels are accepted as ground truth inliers.

First, the proposed LAM algorithm is qualitatively evaluated on three image pairs 1, 9, and 14, where image pair 1 has a large land-use difference, such as buildings; image pair 9 and image pair 14 suffer from extremely small overlapping regions along the horizontal and vertical directions, respectively. The outlier rates of the initial matches extracted from these three image pairs are 81.15%, 92.37%, and 94.42%, respectively. Due to such high outlier rates, matching these image pairs is very challenging. The results are given in Figs. 6–8.

From the results, we know that RANSAC achieves sufficient good performance on these images. However, it still preserves some noisy matches with low location precisions since it only uses a minimal subset instead of the whole matching set to estimate the geometric transformation. Its *Recall* is also not very high. For example, its *Recall* on image pair 9 is lower than 85%. MLESAC, GMS, and GMS + R&S achieve very good results on the *Precision* metric. Their *Precision* accuracies are even better than the proposed LAM on image pair 9. However, their *Recall* accuracies are very poor. For instance, the *Recall* accuracies of MLESAC, GMS, and GMS + R&S on image pair 14 are only approximately 18%, 22% and 24%, respectively. FastVFC obtains good results on image pair 1. However, it does not detect any matches on image pair 9, which means that the method is completely ineffective. The results of FastVFC may still preserve many false matches with very large projection residuals, such as the result in Fig. 8. LLT performs similarly to FastVFC. It fails to detect any correspondences in the Fig. 8. LLT uses a closed-form solution to solve the affine transformation. Thus, it is also sensitive to noise, which can be clearly seen in Fig. 6 and Fig. 7. LORANSAC and IBCO achieve similar results. Both methods yield poor results on image pair 14. The reason is that IBCO takes the output of LORANSAC as input. Thus, IBCO relies heavily on the initial solution provided by LORANSAC and is sensitive to very high outlier rates. LPM has the worst *Precision* among all compared methods. It is not suitable for cases with high outlier rates, especially those higher than 80%. In contrast,

[1] http://download.intergraph.com/downloads/erdas-imagine-2013-2014-example-data.

Table 3
The information about the Satellite dataset.

No.	Image pair	Spectral mode	Image size	GSD(m)	Acquisition date	Location	Description
1	World View 2	Pan	405 × 350	0.5	2011	USA-	Multi-temporal
	World View 2	Pan	405 × 350	0.5	2014	California	
2	TM	Band 5	512 × 512	30	1992	Brazil-	Multi-temporal
	TM	Band 5	512 × 512	30	1994	Amazon	
3	JERS-1	Radar	256 × 256	18	1995	Brazil-	Multi-temporal
	JERS-1	Radar	256 × 256	18	1996	Amazon	
4	TM	Band 5	512 × 512	30	1990	USA-	Multi-temporal
	TM	Band 5	512 × 512	30	1994	Iowa	
5	SPOT 5	True color	800 × 800	2.5	2002	China-	Multi-temporal
	SPOT 6	True color	800 × 800	1.5	2012	Beijing	Multi-sensors
6	TM	Band 1	1450 × 1480	30	2000	Unknown	Multi-
	TM	Band 4	1450 × 1480	30			bands
7	Radarsat-2	Radar	800 × 800	3	2013	China-	Multi-sensors
	Airborne SAR	Radar	800 × 800	3		Jiangsu	
8–15	Aerial	Color-infrared	1450 × 1380	0.2	2011	USA-	Small overlaps
	Aerial	Color-infrared	1450 × 1380	0.2		Illinois	

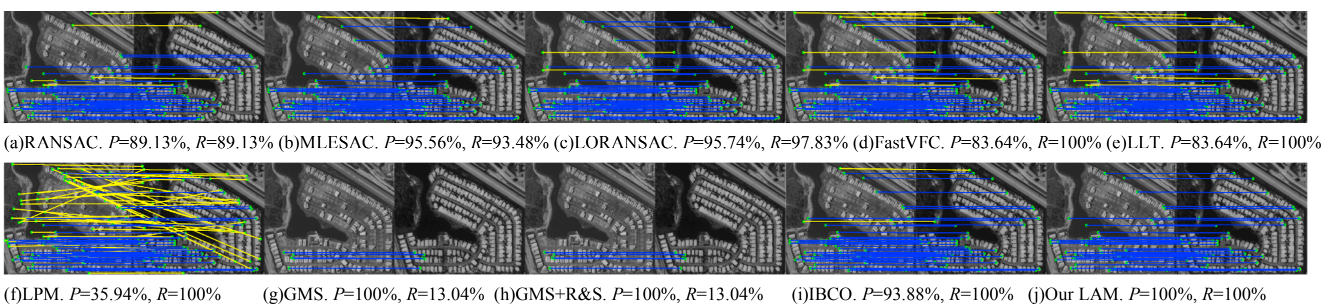


Fig. 6. Results on image pair 1 of the Satellite dataset. Green dots are keypoints, yellow lines are false correspondences, and blue lines are inliers. No more than 100 correspondences are displayed for good visualization. (P and R stand for *Precision* and *Recall*, respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

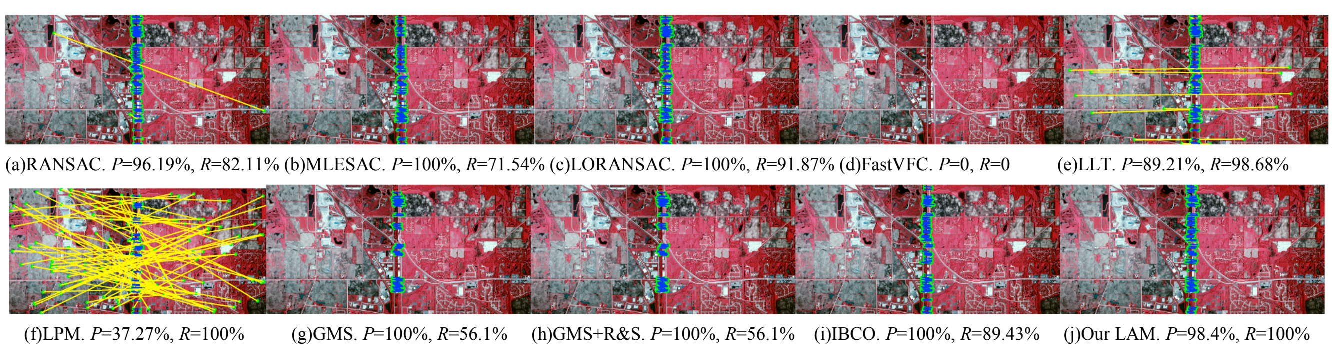
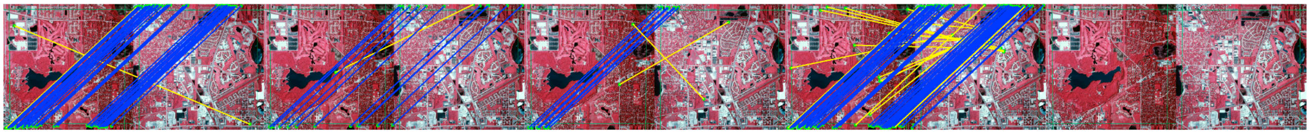


Fig. 7. Results on image pair 9 of the Satellite dataset. Green dots are keypoints, yellow lines are false correspondences, and blue lines are inliers. No more than 100 correspondences are displayed for good visualization. (P and R stand for *Precision* and *Recall*, respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

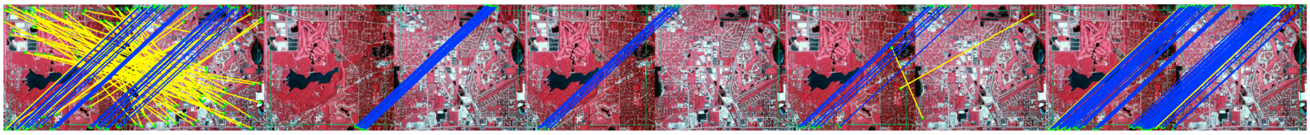
the proposed LAM algorithm achieves the best results. Both the *Precision* and *Recall* are very high, i.e., close to 100%. Only several matches with residuals slightly higher than 3 pixels are preserved.

Then, we quantitatively evaluate our method on the whole Satellite dataset. Fig. 9 plots the *Fscore* results. Table 4 summarizes the average *Precision*, *Recall*, *Fscore*, *MAE*, *RMSE*, and running time results. From Fig. 9, we can draw similar conclusions with the qualitative evaluation. RANSAC obtains acceptable results on all the image pairs. The performance is neither too good nor bad. MLESAC obtains comparable *Precision* accuracy with the proposed LAM. However, MLESAC gets very low *Recall* on some image pairs, which significantly decreases its total performance *Fscore*. FastVFC and LLT achieve similar results. Both of them may fail in some cases. Thus, they get zero *Fscores* on several image pairs. LORANSAC and IBCO are worse than FastVFC and LLT. LPM, GMS, and GMS + R&S obtain the lowest *Fscores* on most of the image pairs. The *Precision* performance of LPM is very poor. Its results are just better than the initial matches. In contrast, the proposed LAM method achieves the best overall accuracy.

As reported in Table 4, the proposed LAM achieves the best average performance in terms of *Precision*, *Fscore*, *MAE*, and *RMSE* and achieves the second best in terms of *Recall*. The average *Fscore* accuracies of RANSAC, MLESAC, LORANSAC, FastVFC, LLT, LPM, GMS, GMS + R&S, IBCO, and LAM are 87.34%, 89.01%, 75.22%, 73.14%, 74.41%, 60.45%, 45.95%, 52.47%, 73.45%, and 98.59%, respectively. Our method achieves an 9.58% growth rate compared with MLESAC, which ranks second. Since our method achieves the best *Precision* accuracy (close to 100%), the *MAE* and *RMSE* of our LAM are much smaller than those of other compared methods. Our average *MAE* and *RMSE* are 1.12 pixels and 1.34 pixels, which means that there are no outliers with large projection residuals in our results. The *MAE* and *RMSE* of the second-



(a)RANSAC. P=93.33%, R=71.79% (b)MLESAC. P=91.33%, R=17.94% (c)LORANSAC. P=77.78%, R=8.97% (d)FastVFC. P=78.79%, R=100% (e)LLT. P=0, R=0



(f)LPM. P=21.58%, R=100% (g)GMS. P=100%, R=21.79% (h)GMS+R&S. P=90.48%, R=24.36% (i)IBCO. P=75%, R=11.38% (j)Our LAM. P=92.86%, R=100%

Fig. 8. Results on image pair 14 of the Satellite dataset. Green dots are keypoints, yellow lines are false correspondences, and blue lines are inliers. No more than 100 correspondences are displayed for good visualization. (P and R stand for *Precision* and *Recall*, respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

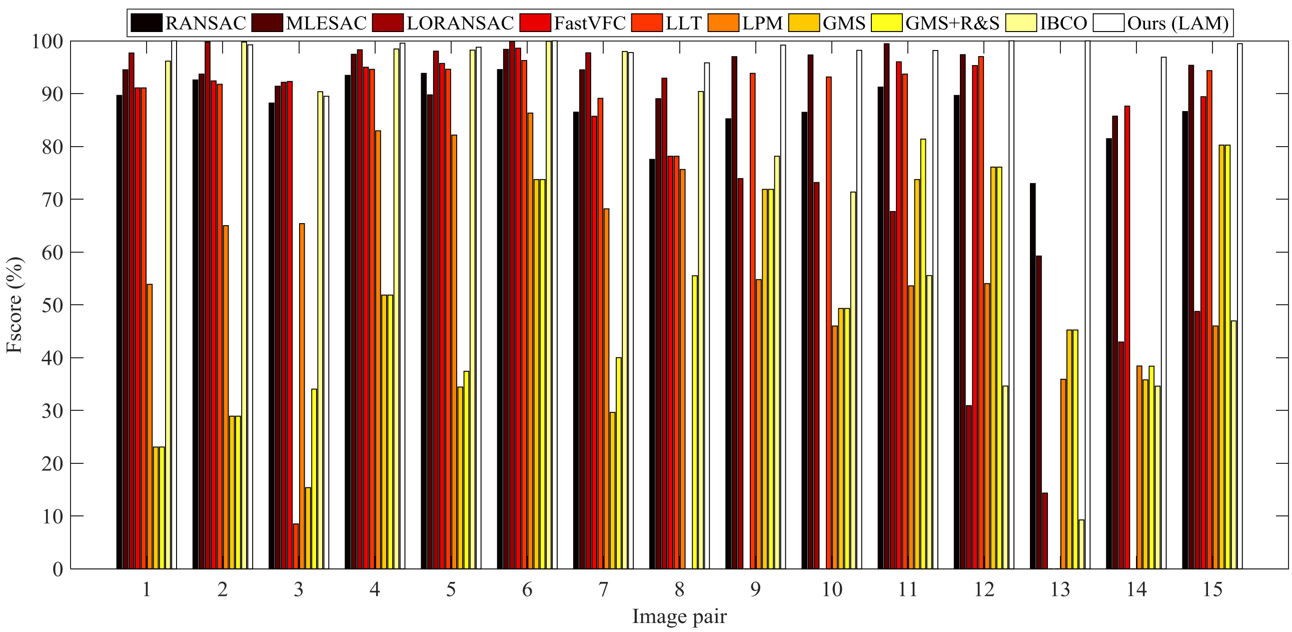


Fig. 9. Comparison of *Fscore* on the Satellite dataset.

Table 4
Performance comparison on the Satellite dataset.

Method	*Precision/%*	*Recall/%*	*Fscore/%*	*MAE/pixels*	*RMSE/pixels*	*Time/s*
RANSAC	90.74±0.37	86.17±2.27	87.34±1.43	10.02±0.62	10.29±0.61	36.19±12.65
MLESAC	97.54±0.81	85.22±4.98	89.01±3.98	2.06±0.89	2.31±0.88	5.43±0.27
LORANSAC	85.72±7.34	73.35±6.10	75.22±6.04	4.65±1.28	4.84±1.27	80.26±3.45
FastVFC	67.89±0.00	79.83±0.00	73.14±0.00	10.55±0.00	11.71±0.00	0.93±0.05
LLT	70.15±0.00	80.24±0.00	74.41±0.00	11.50±0.00	11.75±0.00	2.76±0.97
LPM	45.29±0.00	99.32±0.00	60.45±0.00	19.02±0.00	19.73±0.00	0.02±0.01
GMS	88.59±0.00	33.69±0.00	45.95±0.00	2.35±0.00	2.67±0.00	0.001±0.00
GMS+R&S	89.68±0.00	39.51±0.00	52.47±0.00	1.56±0.00	2.27±0.00	0.06±0.02
IBCO	87.09±4.98	71.93±5.37	73.45±5.33	5.82±1.43	5.99±1.41	97.06±5.42
Ours (LAM)	98.31±0.00	98.92±0.00	98.59±0.00	1.12±0.00	1.34±0.00	0.10±0.01

Note that if the MAE or RMSE of a method on an image pair is higher than 20 pixels, we regard it as 20 pixels. Namely, the maximum of MAE or RMSE is 20 pixels. Numbers in red and blue represent the best and the second. We run the complete test 40 times, and each cell contains an empirical mean and a standard deviation of the results.

best method are only 1.56 pixels and 2.27 pixels. In terms of *Time*, GMS only takes 0.001 s, which is much faster than other methods. RANSAC, MLESAC, and LORANSAC are time consuming due to the high outlier rates. They require a large number of sampling trials to obtain acceptable results. As mentioned earlier, the running time of FastVFC is related to the size of the data. In this dataset, the numbers of initial matches are relatively small (the maximum number is 1600). Thus, the running time of FastVFC is much shorter than that on the Oxford dataset. Our LAM is almost 10 times faster than FastVFC and LLT, 50 + times faster than MLESAC, 300 + times faster than RANSAC, and 900 + times faster than IBCO.

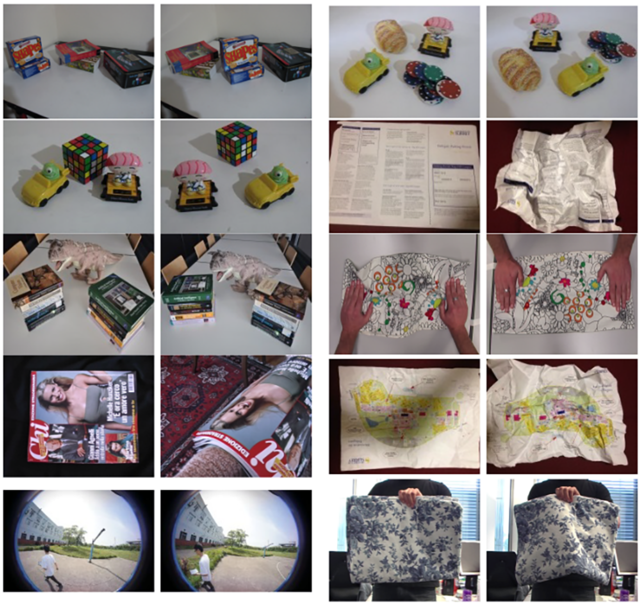


Fig. 10. Nonrigid dataset.

4.2. Nonrigid feature matching

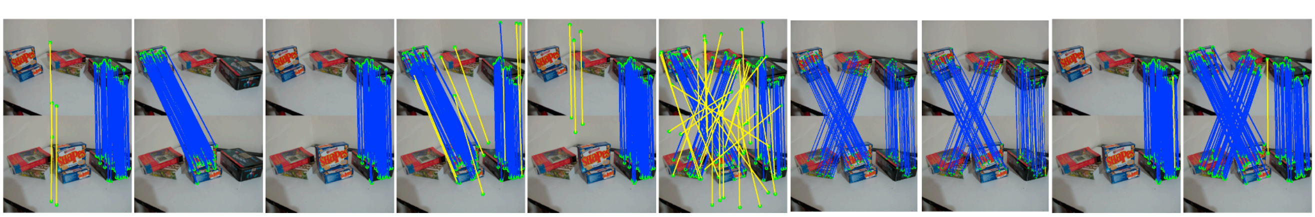
As mentioned above, the proposed LAM is also suitable for nonrigid feature matching problems. We collect 10 image pairs with nonrigid transformations for evaluation, as shown in Fig. 10. The image pairs do not contain ground truth transformations. We also use the SIFT algorithm to provide initial matches. We set the NNDR to 0.67 because smaller values of NNDR generate initial matches with higher inlier rates. Hence, the power of LPM and GMS (GMS + R&S) can be demonstrated. To establish the ground truth inlier correspondences, we first use the 4FP-Structure method (Li et al., 2017a) (without match expansion stage) to extract a reliable match set. Then, we artificially confirm the correctness of each match in the reliable set and check the matches in the remaining set. Calculating *MAE* and *RMSE* on image pairs with complex nonrigid transformations is difficult. Therefore, we only use *Precision*, *Recall*, and *Fscore* metrics for quantitative evaluations.

Figs. 11–13 give the qualitative comparisons on the image pairs 1, 7, and 8 of the Nonrigid dataset, respectively. There are three rigid geometric models in image pair 1. Thus, the matches should be three groups. Image pair 2 suffers from serious "slow and smooth" geometric distortions and a large rotation (large than 45°). Image pair 14 also suffers from large distortions and a 180° rotation. Unfortunately, the distortions are not "slow and smooth". As shown, if an image pair contains multiple rigid geometric models such as image pair 1, RANSAC, MLESAC, LORANSAC, LLT and IBCO are only able to estimate one of them. Thus, their results only contain one group of matches, which is why they have very attractive *Precision* results while very poor *Recall* performances. FastVFC performs better than these methods. It can extract two groups of matches. However, there is a group of correct matches that is discarded by FastVFC. Only LPM, GMS, GMS + R&S, and the proposed LAM are able to extract all the three groups of correct matches. For image pair 7, RANSAC, MLESAC, LORANSAC, and IBCO only preserve the matches that obey the rigid geometric transformation. Thus, correct matches with large local geometric distortions are classified as outliers. GMS failed completely. The reason may be that GMS are sensitive to large rotations. FastVFC obtains a comparable result with our LAM. FastVFC assumes that the vector field is "continuous and smooth". Thus, it is very suitable for "slow and smooth" distortions. From Fig. 13, both FastVFC and LLT are completely failed. The reasons are two-fold: First, the distortions in this image pair conflict with the principle of "continuous and smooth". Second, the image pair suffers from an extremely large rotation. Again, the *Recall* accuracies of MLESAC, LORANSAC and IBCO are very low. Only RANSAC, GMS + R&S, and the proposed LAM achieve good results. Comparing MLESAC and RANSAC, we can see that homography model is more robust than the affine model under complex geometric distortions. Although LORANSAC and IBCO also use a homography model, they generate a quasiconvex matrix to compute residuals. This process is sensitive to large geometric distortions and is time consuming. Although LPM obtains acceptable results on all three image pairs, LPM still preserves many false matches, which largely decreases its *Precision* performance. In contrast, the *Precision* performance of our LAM is much higher than that of LPM.Fig. 12.

Fig. 14 gives the quantitative comparisons. LORANSAC and IBCO achieve the best *Precision* accuracies. However, their *Recall* accuracies are low, which result in poor overall accuracy, i.e., *Fscore*. MLESAC gets comparable results with RANSAC except for image pair 4. FastVFC achieves attractive results on most of the image pairs. However, it failed on image pair 8. LLT performs the worst. It failed on half of the image pairs. LPM achieves the best *Recall* accuracy, and its *Precision* fluctuates by approximately 80%. In contrast, the *Precision* results of our LAM are higher than 95% on most of the image pairs. The average *Precision*, *Recall*, and *Fscore* results on the Nonrigid dataset are reported in Table 5. As shown, the proposed LAM achieves the best average *Fscore* performance and gets the second best in terms of *Recall*. Our *Precision* is slightly lower than LORANSAC and IBCO. The average *Fscore* accuracies of RANSAC, MLESAC, LORANSAC, FastVFC, LLT, LPM, GMS, GMS + R&S, IBCO, and LAM are 60.97%, 38.53%, 47.45%, 77.95%, 39.26%, 88.19%, 51.01%, 78.96%, 47.93%, and 96.33%, respectively. Our method achieves an 8.14% growth rate compared with LPM which ranks second.

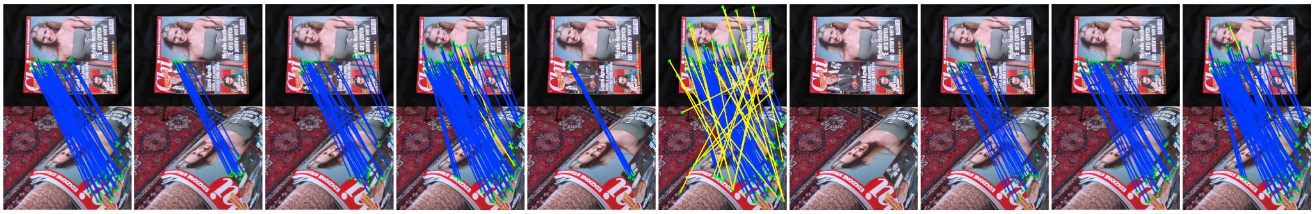
5. Conclusions

This paper proposes a new robust feature matching algorithm called LAM. In contrast to RANSAC-type methods, LAM is suitable for both



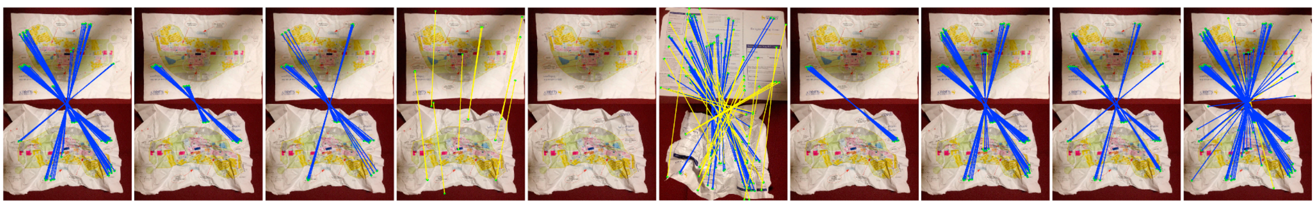
(a) RANSAC	(b) MLESAC	(c) LORANSAC	(d) FastVFC	(e) LLT	(f) LPM	(g) GMS	(h) GMS+R&S	(i) IBCO	(j) Our LAM
P=92.75%	*P*=100%	*P*=100%	*P*=92.91%	*P*=93.06%	*P*=75.20%	*P*=95.8%	*P*=95.24%	*P*=98.46%	*P*=95.19%
R=34.41%	*R*=30.11%	*R*=33.87%	*R*=70.43%	*R*=36.02%	*R*=99.54%	*R*=73.66%	*R*=86.02%	*R*=34.41%	*R*=95.7%

Fig. 11. Results on image pair 1 of the Nonrigid dataset. Green dots are keypoints, yellow lines are false correspondences, and blue lines are inliers. No more than 100 correspondences are displayed for good visualization. (*P* and *R* stand for *Precision* and *Recall*, respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



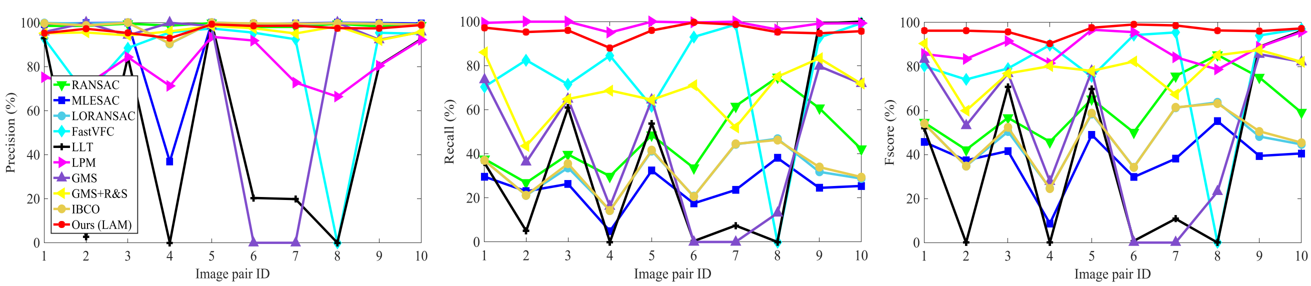
(a) RANSAC	(b) MLESAC	(c) LORANSAC	(d) FastVFC	(e) LLT	(f) LPM	(g) GMS	(h) GMS+R&S	(h) IBCO	(j) Our LAM
P=98.08%	*P*=100%	*P*=100%	*P*=92.5%	*P*=100%	*P*=72.82%	*P*=0	*P*=95.12%	*P*=100%	*P*=97.37%
R=68%	*R*=22.67%	*R*=42.67%	*R*=98.67%	*R*=8%	*R*=100%	*R*=0	*R*=52%	*R*=48%	*R*=98.67%

Fig. 12. Results on image pair 7 of the Nonrigid dataset. Green dots are keypoints, yellow lines are false correspondences, and blue lines are inliers. No more than 100 correspondences are displayed for good visualization. (*P* and *R* stand for *Precision* and *Recall*, respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) RANSAC	(b) MLESAC	(c) LORANSAC	(d) FastVFC	(e) LLT	(f) LPM	(g) GMS	(h) GMS+R&S	(h) IBCO	(j) Our LAM
P=100%	*P*=100%	*P*=100%	*P*=0	*P*=0	*P*=66.39%	*P*=100%	*P*=98.44%	*P*=100%	*P*=97.59%
R=78.57%	*R*=36.9%	*R*=43.05%	*R*=0	*R*=0	*R*=96.71%	*R*=13.1%	*R*=75%	*R*=44.08%	*R*=96.43%

Fig. 13. Results on image pair 8 of the Nonrigid dataset. Green dots are keypoints, yellow lines are false correspondences, and blue lines are inliers. No more than 100 correspondences are displayed for good visualization. (*P* and *R* stand for *Precision* and *Recall*, respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) *Precision* (b) *Recall* (c) *Fscore*

Fig. 14. Quantitative comparisons on the Nonrigid dataset.

Table 5
Performance comparison on the Non-rigid dataset.

Method	*Precision*/%	*Recall*/%	*Fscore*/%
RANSAC	98.83±0.45	45.57±0.87	60.97±0.84
MLESAC	93.49±4.79	24.56±0.89	38.53±1.44
LORANSAC	98.76±2.74	32.02±0.88	47.45±1.21
FastVFC	82.01±0.00	75.44±0.00	77.95±0.00
LLT	49.34±6.29	36.27±3.23	39.26±3.26
LPM	79.99±0.00	98.91±0.00	88.19±0.00
GMS	77.67±0.00	42.02±0.00	51.01±0.00
GMS+R&S	95.86±0.00	68.13±0.00	78.96±0.00
IBCO	98.69±2.74	32.46±0.87	47.93±1.14
Ours (LAM)	96.97±0.00	95.82±0.00	96.33±0.00

The numbers in red and blue represent the best and the second best. We run the complete test 40 times, and each cell contains an empirical mean and a standard deviation of the results.

rigid and nonrigid image matching problems. In addition, LAM is very efficient since it only has linear space complexity and linearithmic time complexity.

We define a new concept called MCM and present a new LBC system in the proposed LAM. The LBC is invariant to local affine transformations. This property is still preserved under complex non-rigid transformations, which makes the LBC more suitable for feature matching tasks than Cartesian coordinates. Thus, we adapt the LBC into a mathematical model to extract potential reliable matches that preserve local geometric constraints. To find other correct matches with false neighborhood correspondences, LAM constructs local MCMs to identify the correctness for the remaining matches by minimizing the rank of MCMs. These two stages not only guarantee the *Precision* accuracy but also the *Recall* performance of the proposed LAM. Extensive experiments on both rigid and nonrigid datasets demonstrate the power of the proposed LAM method. The limitation of LAM is that it only considers geometric constraints. Thus, false matches that satisfy the local affine constraints will be accepted as inliers by our LAM. In addition, RANSAC-type methods and deterministic methods are robust estimation techniques, which are more generic and applicable to any model fitting problem of not too high dimension, whereas the proposed method is tailored specifically to matching problem but not straightforward to adapt to other tasks.

Acknowledgements

The authors would like to express their gratitude to the reviewers for their constructive and helpful comments for substantial improvement of this paper. This work was supported by Project funded by China Postdoctoral Science Foundation (Grant No. 2018M640734) and Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University (Grant No. 18E02).

References

Bian, J., Lin, W.-Y., Matsushita, Y., Yeung, S.-K., Nguyen, T.-D., Cheng, M.-M., 2017. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2828–2837.
Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. 3, 1–122.
Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C., 2017. DSAC-differentiable RANSAC for camera localization. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Brown, M., Lowe, D.G., 2007. Automatic panoramic image stitching using invariant features. Int. J. Comput. Vision 74, 59–73.
Cai, Z., Chin, T.-J., Le, H., Suter, D., 2018. Deterministic consensus maximization with biconvex programming. In: 15th European Conference on Computer Vision (ECCV). Springer International Publishing, Cham, pp. 699–714.
Chin, T.-J., Purkait, P., Eriksson, A., Suter, D., 2015. Efficient globally optimal consensus maximisation with tree search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2413–2421.
Chin, T.-J., Suter, D., 2017. The maximum consensus problem: recent algorithmic advances. Synth. Lect. Comput. Vision 7, 1–194.
Cho, M., Lee, J., Lee, K.M., 2010. Reweighted random walks for graph matching. In: 11th European Conference on Computer Vision (ECCV). Springer, pp. 492–505.
Cho, M., Sun, J., Duchenne, O., Ponce, J., 2014. Finding matches in a haystack: a max-pooling strategy for graph matching in the presence of outliers. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2083–2090.
Chum, O., Matas, J., 2005. Matching with PROSAC-progressive sample consensus. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 220–226.
Chum, O., Matas, J., Kittler, J., 2003. Locally optimized RANSAC. In: Joint Pattern Recognition Symposium. Springer, pp. 236–243.
Conte, D., Foggia, P., Sansone, C., Vento, M., 2004. Thirty years of graph matching in pattern recognition. Int. J. Pattern Recogn. Artif. Intell. 18, 265–298.
Duchenne, O., Bach, F., Kweon, I.-S., Ponce, J., 2011. A tensor-based algorithm for high-order graph matching. IEEE Trans. Pattern Anal. Mach. Intell. 33, 2383–2395.
Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 381–395.
Huber, P.J., 1981. Robust Statistics. Wiley.
Indyk, P., Motwani, R., 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, pp. 604–613.
Jian, B., Vemuri, B.C., 2011. Robust point set registration using gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 33, 1633–1645.
Leordeanu, M., Hebert, M., Sukthankar, R., 2009. An integer projected fixed point method for graph matching and map inference. Adv. Neural Inf. Process. Syst. (NIPS) 1114–1122.
Li, J., Hu, Q., Ai, M., 2016. Robust feature matching for remote sensing image registration based on $ L_ q $-estimator. IEEE Geosci. Remote Sens. Lett. 13, 1989–1993.
Li, J., Hu, Q., Ai, M., 2017a. 4FP-structure: a robust local region feature descriptor. Photogramm. Eng. Remote Sens. 83, 813–826.
Li, J., Hu, Q., Ai, M., Zhong, R., 2017b. Robust feature matching via support-line voting and affine-invariant ratios. ISPRS J. Photogramm. Remote Sens. 132, 61–76.
Li, J., Hu, Q., Zhong, R., Ai, M., 2017c. Exterior orientation revisited: a robust method based on lq-norm. Photogramm. Eng. Remote Sens. 83, 47–56.
Li, X., Hu, Z., 2010. Rejecting mismatches by correspondence function. Int. J. Comput. Vision 89, 1–17.
Lian, W., Zhang, L., Zhang, D., 2012. Rotation-invariant nonrigid point set matching in cluttered scenes. IEEE Trans. Image Process. 21, 2786–2797.
Lin, W.-Y., Wang, F., Cheng, M.-M., Yeung, S.-K., Torr, P.H., Do, M.N., Lu, J., 2018. Code: Coherence based decision boundaries for feature correspondence. IEEE Trans. Pattern Anal. Mach. Intell. 40, 34–47.
Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60, 91–110.
Ma, J., Zhao, J., Jiang, J., Zhou, H., Guo, X., 2017. Locality preserving matching. Int. J. Comput. Vision 1–20.
Ma, J., Zhao, J., Tian, J., Bai, X., Tu, Z., 2013. Regularized vector field learning with sparse approximation for mismatch removal. Pattern Recogn. 46, 3519–3532.
Ma, J., Zhou, H., Zhao, J., Gao, Y., Jiang, J., Tian, J., 2015. Robust feature matching for remote sensing image registration via locally linear transforming. IEEE Trans. Geosci. Remote Sens. 53, 6469–6481.
Maronna, R., Martin, R.D., Yohai, V., 2006. Robust Statistics. John Wiley & Sons, Chichester.
Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1615–1630.
Morel, J.-M., Yu, G., 2009. ASIFT: A new framework for fully affine invariant image comparison. SIAM J. Imag. Sci. 2, 438–469.
Mur-Artal, R., Montiel, J.M.M., Tardos, J.D., 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Rob. 31, 1147–1163.
Murala, S., Maheshwari, R., Balasubramanian, R., 2012. Local tetra patterns: a new feature descriptor for content-based image retrieval. IEEE Trans. Image Process. 21, 2874–2886.
Myronenko, A., Song, X., 2010. Point set registration: coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. 32, 2262–2275.
Raguram, R., Chum, O., Pollefeys, M., Matas, J., Frahm, J.-M., 2013. USAC: a universal framework for random sample consensus. IEEE Trans. Pattern Anal. Mach. Intell. 35, 2022–2038.
Rocco, I., Arandjelovic, R., Sivic, J., 2017. Convolutional neural network architecture for geometric matching. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR.
Rousseeuw, P.J., Leroy, A.M., 1987. Robust Regression and Outlier Detection. John Wiley & Sons.
Torr, P.H., Zisserman, A., 2000. MLESAC: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. 78, 138–156.
Vedaldi, A., Fulkerson, B., 2010. VLFeat: An open and portable library of computer vision algorithms. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 1469–1472.
Wang, G., Zhou, Q., Chen, Y., 2017. Robust non-rigid point set registration using spatially constrained gaussian fields. IEEE Trans. Image Process. 26, 1759–1769.
Wu, C., 2013. Towards linear-time incremental structure from motion. In: International Conference on 3D Vision-3DV, pp. 127–134.
Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P., 2018. Learning to find good correspondences. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).