International Journal of Computer Vision (2019) 127:512–531
https://doi.org/10.1007/s11263-018-1117-z

Locality Preserving Matching

Jiayi Ma[1] · Ji Zhao[2] · Junjun Jiang[3] · Huabing Zhou[4] · Xiaojie Guo[5]

Received: 25 January 2018 / Accepted: 24 August 2018 / Published online: 22 September 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract
Seeking reliable correspondences between two feature sets is a fundamental and important task in computer vision. This paper attempts to remove mismatches from given putative image feature correspondences. To achieve the goal, an efficient approach, termed as locality preserving matching (LPM), is designed, the principle of which is to maintain the local neighborhood structures of those potential true matches. We formulate the problem into a mathematical model, and derive a closed-form solution with linearithmic time and linear space complexities. Our method can accomplish the mismatch removal from thousands of putative correspondences in only a few milliseconds. To demonstrate the generality of our strategy for handling image matching problems, extensive experiments on various real image pairs for general feature matching, as well as for point set registration, visual homing and near-duplicate image retrieval are conducted. Compared with other state-of-the-art alternatives, our LPM achieves better or favorably competitive performance in accuracy while intensively cutting time cost by more than two orders of magnitude.

Keywords Feature matching · Image registration · Locality preservation · Rigid and non-rigid transformations · Outlier removal

1 Introduction

This study focuses on the problem of establishing reliable point correspondences between two images of the same scene. Many computer vision tasks, such as 3D reconstruction, content-based image retrieval, visual homing, image mosaic, image registration and fusion, start by assuming that the point correspondences have been successfully recovered (Bian et al. 2017; Lin et al. 2018; Ma et al. 2013). In this paper, we treat the target task as a matching problem between two sets of discrete points, where each point is an image feature extracted by a feature detector and has a local image descriptor, e.g. the scale invariant feature transform (SIFT) (Lowe 2004).

The matching problem possesses a combinatorial nature, making the matching space huge. Even without considering outliers, a simple problem of matching N points to another N points would lead to a total of $N!$ permutations (Wang et al. 2014). To relieve the computational pressure, a popular strategy is to construct a group of putative correspondences by imposing a similarity constraint to reduce the amount of possible matches. It requires that points can only match points with similar descriptors. Under the circumstances, the matching task boils down to determining the correctness of each match in the putative set. This paper intends to conquer the

Communicated by V. Lepetit.

✉ Xiaojie Guo
xguo@tju.edu.cn

Jiayi Ma
jyma2010@gmail.com

Ji Zhao
zhaoji84@gmail.com

Junjun Jiang
junjun0595@163.com

Huabing Zhou
zhouhuabing@gmail.com

[1] Electronic Information School, Wuhan University, Wuhan 430072, China

[2] ReadSense Ltd., Shanghai 200040, China

[3] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

[4] Hubei Provincial Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan, 430073, China

[5] School of Computer Software, Tianjin University, Tianjin 300350, China

mismatch removal from some given putative point correspondences.

During the past few decades, a variety of robust estimators have been developed to address the mismatch removal problem. Nevertheless, it is still challenging to customize an effective and efficient algorithm for practical use. The challenges mainly come from three aspects. Firstly, the use of only local descriptor information will inevitably lead to a number of false matches in the putative set, and this problem is typically even worse if the image pairs suffer from low-quality, occlusion, repeated structures, etc. Secondly, the transformation models between two images are various, making it difficult to design a general algorithm. However, such a general algorithm is often required in many computer vision tasks such as deformable object recognition where the transformation models are unknown in advance. Thirdly, the high computational load, especially when the geometric transformation between two images is an unknown complex non-rigid model, limits its applicability in real-time tasks.

To address the above three challenges, this paper proposes a simple yet surprisingly effective feature matching approach, which is able to accurately remove the outliers from a putative correspondence set in only a few milliseconds. We observe that for an image pair of the same scene or object, the absolute distance between two feature points may change significantly under viewpoint changes or non-rigid deformations, but the spatial neighborhood relationship among feature points representing the topological structures of an image scene is generally well preserved due to physical constraints. Based on this observation, we introduce a mathematical model that aims to constrain the unknown inlier correspondences to have similar local neighborhood structures. The model is general, and it can embrace both rigid and non-rigid deformations. We further derive a simple closed-form solution, which has linearithmic time complexity and linear space complexity with respect to the scale of the given putative set. The qualitative and quantitative experiments on various image data demonstrate that the proposed method can produce more accurate matching results with much less computation time (more than two orders of magnitude faster) in comparison with other state-of-the-art methods.

More concretely, the contributions of this paper can be summarized as follows:

- We propose a simple yet effective approach for robust feature matching. Unlike most existing methods that require a special parametric or non-parametric model to characterize the global image transformation, our method merely aims to preserve local neighborhood structures of feature points and hence, it is more general.
- We derive a closed-form solution with linearithmic complexity, which can solve a typical matching problem with like 1000 putative correspondences in only a few milliseconds. This is beneficial for many real-time applications and can quickly provide a good initialization for complicated problem-specific matching algorithms.
- We apply our approach to several visual tasks, including point set registration, visual homing and near-duplicate image retrieval, and design corresponding methods. We validate the proposed methods on publicly available datasets, and obtain better results than other state-of-the-art methods in terms of both accuracy and efficiency.

A preliminary version of this manuscript appeared in Ma et al. (2017). The primary new contributions include the following four aspects. First, we provide an expanded derivation of the proposed method with more details. Second, we generalize the formulation and give a comprehensive definition on the spatial neighborhood relationship which can further promote the matching performance. Third, we apply the proposed method to several visual tasks and design the corresponding algorithms in detail. Last, we conduct extensive experiments on more challenging datasets with comparisons to more state-of-the-art methods. To allow more comparisons from the community and encourage future work, we have released our code.[1]

The remainder of this paper is organized as follows. Section 2 describes background material and related work. In Sect. 3, we present our locality preserving matching for robust feature matching. We apply our approach to several visual tasks and design corresponding methods in Sect. 4. Section 5 illustrates the performance of our method in comparison with other approaches on different visual tasks, followed by some concluding remarks in Sect. 6.

2 Related Work

Feature matching has been widely used in many fields including computer vision (Torr and Zisserman 2000; Jiang et al. 2017), pattern recognition (Gao et al. 2017; Guo and Cao 2012), medical image analysis (Ma et al. 2017; Wang et al. 2016), remote sensing (Ma et al. 2015; Yang et al. 2017), robotics (Liu et al. 2013; Zhao and Ma 2017), etc. Here we briefly review the background material applied as reference for the current study. This material includes two method types: the first type establishes a set of putative correspondence and then removes false matches, whereas the second type solves a correspondence matrix between a couple of point sets.

2.1 Two-Step Strategy Based Methods

A popular strategy for solving the matching problem involves two steps (Ma et al. 2014): first computing a set of putative

[1] https://sites.google.com/site/jiayima2013/home.

correspondence, and then removing the outliers via geometrical constraints. Putative correspondence instances are obtained in the first step by pruning the set of all possible point matches. This scenario is achieved by computing feature descriptors at the points and eliminating the matches between points whose descriptors are excessively dissimilar. Lowe (2004) proposed the SIFT descriptor with a distance ratio method that compares the ratio between the nearest and next-nearest neighbors against a predefined threshold to filter out unstable matches. Guo and Cao (2012) proposed a triangle constraint, which can produce better putative correspondences in terms of quantity and accuracy compared with the distance ratio in Lowe (2004). Pele and Werman (2008) applied the earth mover's distance to replace the Euclidean distance in Lowe (2004) to measure the similarity between descriptors and improve the matching accuracy. In addition, Hu et al. 2015 adopted the local selection of a suitable descriptor for each feature point instead of employing a global descriptor during putative correspondence construction. A cascade scheme has been suggested to prevent the loss of true matches, which can significantly enhance the correspondence number (Wang et al. 2014; Cho and Lee 2012). Although there have been various sophisticated approaches for putative match construction, the use of only local appearance features will inevitably result in a lot of false matches. In the second step, robust estimators based on some geometrical constraints are used to detect and remove the outliers.

To remove false matches from putative sets, numerous methods have been developed over the last decades, which can be roughly divided into four categories, say statistical regression methods, resampling methods, non-parametric interpolation methods, and graph matching methods. Statistics literature shows that the methods that minimize the L_1 norm are more robust and can resist a larger proportion of outliers compared with quadratic L_2 norms (Huber 1981). Liu et al. 2015 proposed a regression method based on adaptive boosting learning for 3D rigid matching. Recently, Maier et al. 2016 introduced a guided matching scheme based on statistical optical flow, and promising results have been demonstrated in terms of both accuracy and efficiency. The most popular resampling method is random sample consensus (RANSAC), which has several variants such as MLESAC (Torr and Zisserman 2000) and PROSAC (Chum and Matas 2005). These methods adopt a hypothesize-and-verify approach and attempt to obtain the smallest possible outlier-free subset to estimate a provided parametric model by resampling. The statistical regression and resampling methods rely on a predefined parametric model, which become less efficient when the underlying image transformation is non-rigid; these methods also tend to severely degrade if the outlier proportion becomes large (Li and Hu 2010). Several non-parametric interpolation methods have recently been introduced to address these issues, including identifying correspondence function (ICF) (Li and Hu 2010), bounded distortion (BD) (Lipman et al. 2014), vector field consensus (VFC) (Ma et al. 2014), and robust point matching with manifold regularization (MR-RPM) (Ma et al. 2017; Wang et al. 2016). These methods commonly interpolate a non-parametric function by applying the prior condition, in which the motion field associated with the feature correspondence is slow-and-smooth. However, they typically have cubic complexities and the computational costs are huge for large putative sets, which limits their applicability on real-time tasks. Graph matching is another technique to solve the matching problem; several representative studies include spectral matching (Leordeanu and Hebert 2005), dual decomposition (Torresani et al. 2008), mode-seeking (Wang et al. 2014; Cho and Lee 2012), graph shift (GS) (Liu and Yan 2010), and discrete tabu search (Adamczewski et al. 2015). Graph matching provides considerable flexibility to the transformation model and delivers robust matching and recognition. Nevertheless, it suffers from similar drawbacks of its non-polynomial-hard nature. Technically, our work belongs to this category.

Additionally to the methods above, we want to highlight two recently proposed important algorithms which incorporate piecewise-smoothness constraints into matching. The first one is a non-linear regression technique called coherence based decision boundaries (Lin et al. 2014, 2013, 2018). This algorithm aims to discover a coherence based separability constraint from highly noisy matches and embed it into a correspondence likelihood model, and the accurate matches are then obtained by varying affine motion model. It is able to yield high quality matches at wide baselines and robust to a large number of outliers (even up to 90%). The second one is grid-based motion statistics (GMS) (Bian et al. 2017), which removes outliers by converting the motion smoothness constraints into statistical measures based on the number of neighboring matches. A major advantage of this algorithm is that it develops an efficient grid-based score estimator which can provide real-time, ultra-robust feature correspondences, and hence is beneficial to video applications.

2.2 Correspondence Matrix based Methods

Another strategy is to incorporate a correspondence matrix with a parametric, or non-parametric, geometric constraint. In this situation, the feature points usually do not have information of local image descriptors. One of the best-known point matching approaches is iterative closest point (ICP) (Besl and McKay 1992). ICP alternatively assigns a binary correspondence utilizing nearest-neighbor relationships; it then performs least squares transformation estimation via the estimated correspondence until a local minimum is reached. Chui and Rangarajan (Chui and Rangarajan 2003) established a general framework for non-rigid matching called

robust point matching with thin plate spline (TPS-RPM), which replaces the nearest point strategy of ICP with soft assignments within a continuous optimization framework that involves deterministic annealing. Yang *et al.* (Yang et al. 2015) further introduced an approach termed as global and local mixture distance with thin plate spline, and has shown promising results. Zheng and Doermann proposed a graph based method for robust point matching based on preserving local neighborhood structures (RPM-LNS) (Zheng and Doermann 2006). Boughorbel et al. (Boughorbel et al. 2004) brought the Gaussian fields into rigid registration, which was later generalized to the non-rigid setting in Ma et al. (2015) and (Wang et al. 2016). Point set registration has commonly been solved by probabilistic methods in recent years, such as Gaussian mixture model based registration (GMMREG) (Jian and Vemuri 2011), coherent point drift (CPD) (Myronenko and Song 2010) and its variants (Horaud et al. 2011; Ma et al. 2016). These methods formulate the matching problem as the estimation of a mixture of densities utilizing Gaussian mixture models, which is solved within the maximum-likelihood framework and expectation-maximization algorithm. However, since these methods completely discard the abundant information of local image descriptors, their matching performance very likely degrades, especially when the image pair involves non-rigid deformations (Ma et al. 2016).

3 Methodology

This section describes our method for establishing accurate correspondences between two feature sets extracted respectively from two images of the same or similar scenes. To this end, we first construct a set of putative matches by considering all possible matches between two feature sets and filtering out matches whose feature descriptor vectors are sufficiently different. We then use a geometric constraint to remove the false matches contained in the putative set, which further filters out those matches with different spatial neighborhood structures among feature points. Fortunately, there are several well-designed feature descriptors (e.g., SIFT Lowe 2004) can efficiently establish putative correspondence between feature sets, therefore, we consider this component as an easy mission. In the following, we concentrate on the mismatch removal problem.

3.1 Problem Formulation

Suppose we have obtained a set of N putative feature correspondences $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ extracted from two given images, where $\mathbf{x}_i$ and $\mathbf{y}_i$ are 2D column vectors denoting the spatial positions of feature points (our approach is not limited by the dimension of the input data, which can be directly applied to 3D matching problems). Our goal is to remove the outliers contained in S to establish accurate correspondences.

3.1.1 Formulation for Ideal Rigid Transformation

If the spatial relationship between the image pair is a simple rigid transformation, then the distance between any feature correspondence will be preserved. In other words, denoting $\mathcal{I}$ the unknown inlier set, its optimal solution is

$$\mathcal{I}^* = \arg\min_{\mathcal{I}} C(\mathcal{I}; S, \lambda), \tag{1}$$

with the cost function C defined as:

$$C(\mathcal{I}; S, \lambda) = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{I}} \big(d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{y}_i, \mathbf{y}_j)\big)^2 + \lambda(N - |\mathcal{I}|), \tag{2}$$

where d is a certain distance metric such as Euclidean distance, and $|\cdot|$ denotes the cardinality of a set. In this cost function, the first term penalizes any match which does not preserve the distance of a point pair, the second term discourages the outliers, and the parameter $\lambda > 0$ balances the two terms. Ideally, the optimal solution should achieve zero penalty, i.e, the first term of C should be zero.

3.1.2 Formulation for General Feature Matching

In real-world scenarios, however, the rigid transformation is barely the case. For example, if the image pair undergoes a relatively complex non-rigid transformation, the above distance relationship will no longer hold, especially for matches far from each other. Nevertheless, the local neighborhood structure among feature points may not change freely due to the physical constraints in a small region around a point, which means that the distribution of neighboring point pairs after transformation should be preserved (Zheng and Doermann 2006). In the sequel, by preserving only local structures, the cost function in Eq. (2) becomes:

$$C(\mathcal{I}; S, \lambda) = \sum_{i\in\mathcal{I}} \frac{1}{2K} \Bigg(\sum_{j|\mathbf{x}_j\in\mathcal{N}_{\mathbf{x}_i}} \big(d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{y}_i, \mathbf{y}_j)\big)^2 + \sum_{j|\mathbf{y}_j\in\mathcal{N}_{\mathbf{y}_i}} \big(d(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{y}_i, \mathbf{y}_j)\big)^2 \Bigg) + \lambda(N - |\mathcal{I}|), \tag{3}$$

where $\mathcal{N}_{\mathbf{x}}$ denotes the neighborhood of point $\mathbf{x}$. There is no obvious neighborhood definition for a point set. In our evaluation, we adopt a simple strategy that searches the K nearest neighbors for each point in the corresponding feature set under the Euclidean distance. Note that we use $1/2K$ in the first term of Eq. (3) to normalize the contribution of each element in the neighborhood.

We associate the putative set S with an $N \times 1$ binary vector $\mathbf{p}$, where $p_i \in \{0, 1\}$ represents the match correctness of the i-th correspondence $(\mathbf{x}_i, \mathbf{y}_i)$. Specifically, $p_i = 1$ indicates an inlier, and an outlier otherwise. Note that the absolute distance of a point pair is not well maintained under non-rigid deformations such as scale changes. To address this issue, we quantize the distance into two levels as:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 0, & \mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i} \\ 1, & \mathbf{x}_j \notin \mathcal{N}_{\mathbf{x}_i} \end{cases}, \quad d(\mathbf{y}_i, \mathbf{y}_j) = \begin{cases} 0, & \mathbf{y}_j \in \mathcal{N}_{\mathbf{y}_i} \\ 1, & \mathbf{y}_j \notin \mathcal{N}_{\mathbf{y}_i} \end{cases}. \tag{4}$$

Proposition 1 *With the distance definition in Eq. (4), the cost function in Eq. (3) is equivalent to the following minimization problem:*

$$C(\mathbf{p}; S, \lambda) = \sum_{i=1}^{N} \frac{p_i}{K} \sum_{j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} d(\mathbf{y}_i, \mathbf{y}_j) + \lambda \left(N - \sum_{i=1}^{N} p_i \right). \tag{5}$$

Proof By using the distance defined in Eq. (4) and a binary vector $\mathbf{p}$, the cost function in Eq. (3) turns out to be:

$$C(\mathbf{p}; S, \lambda) = \sum_{i=1}^{N} \frac{p_i}{2K} \Bigg(\sum_{j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} d(\mathbf{y}_i, \mathbf{y}_j) + \sum_{j|\mathbf{y}_j \in \mathcal{N}_{\mathbf{y}_i}} d(\mathbf{x}_i, \mathbf{x}_j) \Bigg) + \lambda \left(N - \sum_{i=1}^{N} p_i \right). \tag{6}$$

We consider the item $\sum_{j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} d(\mathbf{y}_i, \mathbf{y}_j)$:

$$\begin{aligned} \sum_{j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} d(\mathbf{y}_i, \mathbf{y}_j) &= \sum_{j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}, \mathbf{y}_j \in \mathcal{N}_{\mathbf{y}_i}} d(\mathbf{y}_i, \mathbf{y}_j) + \sum_{j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}, \mathbf{y}_j \notin \mathcal{N}_{\mathbf{y}_i}} d(\mathbf{y}_i, \mathbf{y}_j) \\ &= 0 + count(j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}, \mathbf{y}_j \notin \mathcal{N}_{\mathbf{y}_i}) \\ &= K - count(j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}, \mathbf{y}_j \in \mathcal{N}_{\mathbf{y}_i}) \\ &= K - n_i, \end{aligned} \tag{7}$$

where $count(\cdot)$ counts the number of elements in a set, and n_i denotes the number of common elements in the two neighborhoods $\mathcal{N}_{\mathbf{x}_i}$ and $\mathcal{N}_{\mathbf{y}_i}$. Similarly, we also have

$$\sum_{j|\mathbf{y}_j \in \mathcal{N}_{\mathbf{y}_i}} d(\mathbf{x}_i, \mathbf{x}_j) = K - n_i = \sum_{j|\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} d(\mathbf{y}_i, \mathbf{y}_j). \tag{8}$$

By substituting Eq. (8) into Eq. (6), we obtain the minimization problem in Eq. (5). □

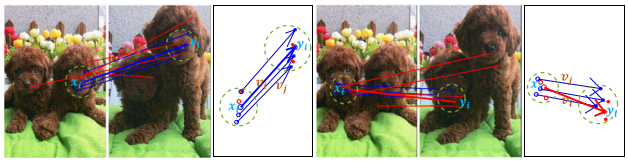

Fig. 1 Schematic illustration of the consensus of neighborhood topology. The putative match $(\mathbf{x}_i, \mathbf{y}_i)$ (highlighted with bold) is an inlier in the left group and an outlier in the right group. For each group, the left figure shows a putative match $(\mathbf{x}_i, \mathbf{y}_i)$ together with its neighborhood elements, their corresponding displacement vectors are shown in the right figure with $\mathbf{v}_i$ corresponding to $(\mathbf{x}_i, \mathbf{y}_i)$

3.1.3 Consensus of Neighborhood Topology

The minimization problem described above essentially aims to preserve the intersection of neighbors (e.g., the consensus of neighborhood elements), which ignores their topological structure. To address this issue, here we design a cost to further exploit the consensus of neighborhood topology.

For a putative match $(\mathbf{x}_i, \mathbf{y}_i)$, as shown in Fig. 1, we first extract its n_i neighboring putative matches located in $\mathcal{N}_{\mathbf{x}_i}$ and $\mathcal{N}_{\mathbf{y}_i}$, where $K = 5$ and $n_i = 3$. Next, we convert the putative matches into displacement vectors, where the head and tail of each vector correspond to the spatial positions of two corresponding feature points in the two images, and the vector associated with $(\mathbf{x}_i, \mathbf{y}_i)$ is highlighted with bold, i.e $\mathbf{v}_i$. The neighborhood topology can then be exploited by comparing the difference between $\mathbf{v}_i$ and $\mathbf{v}_j$ associated with the n_i neighboring putative matches. More specifically, the changes of topological structures of the n_i elements with respect to $\mathbf{x}_i$ and $\mathbf{y}_i$ will lead to significant differences between $\mathbf{v}_i$ and $\mathbf{v}_j$ in both lengths and directions, as demonstrated in the two examples in Fig. 1.

According to the analysis above, we define the consensus of neighborhood topology based on the ratio of length and the angle between $\mathbf{v}_i$ and $\mathbf{v}_j$:

$$s(\mathbf{v}_i, \mathbf{v}_j) = \frac{\min\{|\mathbf{v}_i|, |\mathbf{v}_j|\}}{\max\{|\mathbf{v}_i|, |\mathbf{v}_j|\}} \cdot \frac{(\mathbf{v}_i, \mathbf{v}_j)}{|\mathbf{v}_i| \cdot |\mathbf{v}_j|}, \tag{9}$$

where $s(\mathbf{v}_i, \mathbf{v}_j) \in [-1, 1]$ and a larger value indicates higher consensus, and the cosine similarity is used to characterize the consensus of angle with $(\cdot, \cdot)$ denoting the inner product.

According to Eq. (9) and considering the issue of non-rigid deformations, we define a quantized distance between $\mathbf{v}_i$ and $\mathbf{v}_j$ with a predefined threshold τ as follows:

$$d(\mathbf{v}_i, \mathbf{v}_j) = \begin{cases} 0, & s(\mathbf{v}_i, \mathbf{v}_j) \geq \tau \\ 1, & s(\mathbf{v}_i, \mathbf{v}_j) < \tau \end{cases}. \tag{10}$$

With the above distance and considering the minimization problem in Eq. (5), we obtain a new objective function:

$$C(\mathbf{p};S,\lambda,\tau)=\sum_{i=1}^{N}\frac{p_i}{K}\Bigg(\sum_{j|\mathbf{x}_j\in\mathcal{N}_{\mathbf{x}_i}} d(\mathbf{y}_i,\mathbf{y}_j) + \sum_{j|\mathbf{x}_j\in\mathcal{N}_{\mathbf{x}_i},\mathbf{y}_j\in\mathcal{N}_{\mathbf{y}_i}} d(\mathbf{v}_i,\mathbf{v}_j)\Bigg)+\lambda\Bigg(N-\sum_{i=1}^{N}p_i\Bigg), \tag{11}$$

where the value inside the bracket of the first term is an integer ranging from 0 to K.

3.1.4 Multi-Scale Neighborhood Representation

In our formulation, we propose to search the K nearest neighbors for each point $\mathbf{x}$ to construct its neighborhood $\mathcal{N}_{\mathbf{x}}$. However, the optimal value of K may change due to the following two reasons: i) the putative matches are usually not uniformly distributed across the image domain, and ii) the proportion of outliers changes along with different putative sets. Therefore, using a fixed K will be problematic for addressing the general feature matching problem.

To address this issue, we use a multi-scale neighborhood representation and define a set of neighborhoods with sizes $\mathbf{K}=\{K_m\}_{m=1}^{M}$, e.g. $\{\mathcal{N}_{\mathbf{x}_i}^{K_m}\}_{m=1}^{M}$ and $\{\mathcal{N}_{\mathbf{y}_i}^{K_m}\}_{m=1}^{M}$, where $\mathcal{N}_{\mathbf{x}_i}^{K_m}$ denotes the neighborhood of point $\mathbf{x}_i$ composed of its K_m nearest neighbors under Euclidean distance. In this case, the objective function in Eq. (11) becomes

$$C(\mathbf{p};S,\lambda,\tau)=\sum_{i=1}^{N}\frac{p_i}{M}\sum_{m=1}^{M}\frac{1}{K_m}\Bigg(\sum_{j|\mathbf{x}_j\in\mathcal{N}_{\mathbf{x}_i}^{K_m}} d(\mathbf{y}_i,\mathbf{y}_j) + \sum_{j|\mathbf{x}_j\in\mathcal{N}_{\mathbf{x}_i}^{K_m},\mathbf{y}_j\in\mathcal{N}_{\mathbf{y}_i}^{K_m}} d(\mathbf{v}_i,\mathbf{v}_j)\Bigg)+\lambda\Bigg(N-\sum_{i=1}^{N}p_i\Bigg), \tag{12}$$

where $1/M$ is used to normalize the contribution of each level of neighborhood. Clearly, the final objective function in Eq. (12) is translation, rotation, and scale invariant. The problem of removing outliers and establishing accurate feature matches can then be solved by minimizing Eq. (12).

3.2 Solution

To optimize the objective function (12), we reorganize its form by merging the terms related to p_i and obtain:

$$C(\mathbf{p};S,\lambda,\tau)=\sum_{i=1}^{N}p_i(c_i-\lambda)+\lambda N, \tag{13}$$

where

$$c_i=\sum_{m=1}^{M}\frac{1}{MK_m}\Big(\sum\nolimits_{j|\mathbf{x}_j\in\mathcal{N}_{\mathbf{x}_i}^{K_m}} d(\mathbf{y}_i,\mathbf{y}_j) + \sum\nolimits_{j|\mathbf{x}_j\in\mathcal{N}_{\mathbf{x}_i}^{K_m},\mathbf{y}_j\in\mathcal{N}_{\mathbf{y}_i}^{K_m}} d(\mathbf{v}_i,\mathbf{v}_j)\Big) \tag{14}$$

measures if the i-th correspondence $(\mathbf{x}_i,\mathbf{y}_i)$ meets the geometric constraint of preserving the local neighborhood structure. Clearly, a correct match will bring zero cost or a small cost while a false match will increase the cost largely.

For a given putative set, the neighborhood relationship between the feature points is fixed, and hence all the cost values $\{c_i\}_{i=1}^{N}$ can be calculated in advance. That is to say, the only unknown variable in Eq. (13) is p_i, and its solution is obvious: any correspondence with a cost smaller than λ will lead to a negative term and decrease the objective function, while any correspondence with a cost larger than λ will result in an positive term and increase the objective function. Therefore, the optimal solution of $\mathbf{p}$ that minimizes Eq. (13) is determined by the following simple criterion:

$$p_i=\begin{cases}1, & c_i\le\lambda\\ 0, & c_i>\lambda\end{cases},\ i=1,\ldots,N. \tag{15}$$

And hence, the optimal inlier set $\mathcal{I}^*$ is determined by:

$$\mathcal{I}^*=\{i\mid p_i=1,i=1,\ldots,N\}. \tag{16}$$

From Eq. (15), we see that parameter λ also plays a role of threshold for judging the match correctness of each putative correspondence. Note that the setting of p_i can be arbitrary when $c_i=\lambda$.

3.3 Neighborhood Construction

The neighborhood $\mathcal{N}_{\mathbf{x}}$ of each point $\mathbf{x}$ in Eq. (3) is constructed based on the whole feature set, probably involving outliers. This strategy works well due to the following reasons. On the one hand, for an outlier $(\mathbf{x}_i,\mathbf{y}_i)$, its local neighborhood structures cannot be preserved between two images, leading to a large cost c_i, and hence it will be easily identified as an outlier. On the other hand, for an inlier $(\mathbf{x}_j,\mathbf{y}_j)$, even if its neighborhood $\mathcal{N}_{\mathbf{x}_j}$ or $\mathcal{N}_{\mathbf{y}_j}$ contains some outliers, the major components are inliers, which is still consistent with the geometric constraint. Therefore, its cost c_j will not be large.

To verify how well it works, we collect in total 30 image pairs with different types of transformations including piecewise linear transformation, non-rigid deformation, wide baseline image pair, etc. The average initial inlier percentage of SIFT matching on the whole test data is only 51.19%,

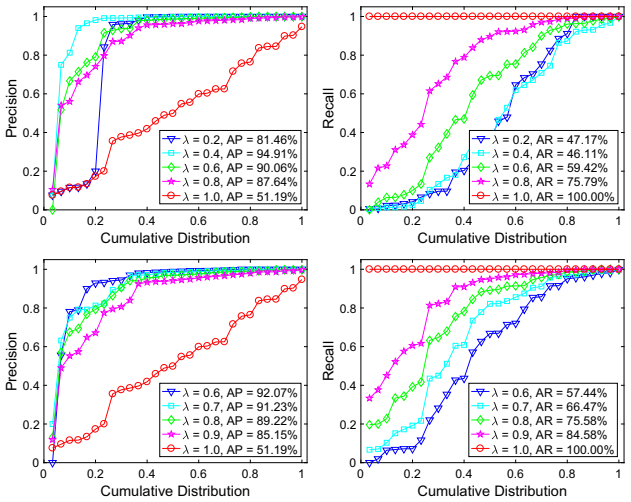

Fig. 2 Precision and recall with respect to the cumulative distribution by using the whole feature set to construct the neighborhood. Top row: result without using multi-scale neighborhood representation, e.g., $K = 6$; bottom row: result using multi-scale neighborhood representation, e.g., $\mathbf{K} = [4, 6, 8]$. A point on the curve with coordinate (x, y) denotes that there are $100 * x$ percents of image pairs which have precision or recall no more than y

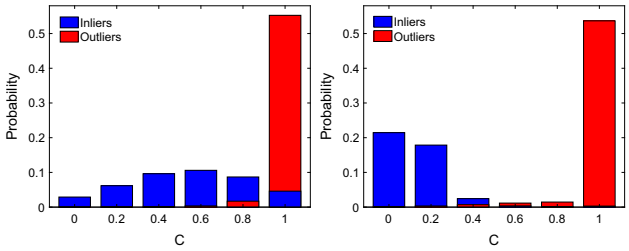

Fig. 3 Distribution of the cost c_i in Eq. (14) by using the whole feature set (left) and by using $\mathcal{I}_0$ (right) to construct the neighborhood. For each bin, we overlap the inlier and outlier probabilities, where the one with smaller probability is shown in the outer layer

and hence the outlier removal task is quite challenging.[2] The precision and recall are used as our metrics to evaluate the matching performance, where the precision is defined as the ratio of the identified correct match number and the preserved match number, and the recall is defined as the ratio of the identified correct match number and the correct match number contained in the putative set. The precision and recall curves with respect to different λ are summarized in Fig. 2. We see that with a proper value of λ (e.g., 0.9 in the bottom row), our method is able to preserve about 84.58% of the true matches, and the precision can also reach up to 85.15%. In addition, the effectiveness of multi-scale neighborhood representation is also validated in Fig. 2, where the top and bottom rows are respectively the results without and with using the multi-scale neighborhood representation. Clearly, the multi-scale neighborhood representation is able to largely promote the matching performance.

Nevertheless, it will be more desirable if the neighborhood $\mathcal{N}_\mathbf{x}$ can be constructed based on only the inlier set $\mathcal{I}$. In this case, the calculation of the cost c_j for an inlier will be more accurate and is not influenced by the outlier, therefore, the margin between inlier and outlier will be distinctly enlarged. This is helpful for accurate classification of the putative correspondences, especially when the putative set S contains a large number of outliers. However, the true inlier set $\mathcal{I}$ cannot be known in advance and it is to be solved in our problem.

[2] The distribution of initial inlier percentages on the test data can be seen from the precision curve at $\lambda = 1$ in Fig. 2 as in this case all putative matches are considered as inliers.

To solve this dilemma, here we seek an approximation $\mathcal{I}_0$ of it. As shown in Fig. 2, our method is able to generate a correspondence set which can remove most of the outliers and simultaneously keep most of the inliers just by using S for neighborhood construction. Clearly, this set is a good approximation of the true inlier set, i.e, $\mathcal{I}_0 = \arg\min_{\mathcal{I}} C(\mathcal{I}; S, \lambda, \tau)$ with the neighborhood constructed based on the whole set S.

Subsequently, we use $\mathcal{I}_0$ to construct the neighborhood for each correspondence in S, and solve the optimal $\mathcal{I}^*$ as:

$$\mathcal{I}^* = \arg\min_{\mathcal{I}} C(\mathcal{I}; \mathcal{I}_0, S, \lambda, \tau). \tag{17}$$

By using $\mathcal{I}_0$ instead of S for neighborhood construction, the average precision-recall pair on the 30 test pairs can be largely increased from (84.58%, 85.15%) to (91.28%, 94.49%). The distributions of the cost c_i by using the whole feature set and using $\mathcal{I}_0$ to construct the neighborhood are reported in Fig. 3. We see that the margin between inlier and outlier has been distinctly enlarged.

In fact, we could use a progressive strategy to construct the neighborhood, i.e, iteratively using the match set generated in the previous iteration for neighborhood construction until convergence, and the average precision-recall pair is then further increased to (92.26%, 94.26%). Note that such progressive strategy can only slightly improve the performance, which means that $\mathcal{I}_0$ is good enough to approximate the true inlier set for neighborhood construction. Therefore, we just use Eq. (17) to determine the optimal inlier set for simplicity. Since our matching strategy is to preserve local neighborhood structures, we name our method locality preserving matching (LPM). The whole procedure of our LPM has been outlined in Algorithm 1.

Parameter settings There are three parameters in our method: $\mathbf{K}$, λ, and τ. Parameter $\mathbf{K}$ determines the number of nearest neighbors for multi-scale neighborhood construction. Parameter λ controls the threshold for judging the correctness of a putative correspondence. Parameter τ determines whether a neighboring putative match preserves the consensus of neighborhood topology. Clearly, a large value of $\mathbf{K}$, a small value of λ, or a large value of τ will increase the

Algorithm 1: The LPM Algorithm

Input: putative set $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, parameters $\mathbf{K}$, λ, τ
Output: inlier set $\mathcal{I}^*$

1 Construct neighborhood $\{\mathcal{N}_{\mathbf{x}_i}^{K_m}, \mathcal{N}_{\mathbf{y}_i}^{K_m}\}_{m=1,i=1}^{M,N}$ using S;
2 Calculate cost $\{c_i\}_{i=1}^{N}$ using Eq. (14);
3 Determine $\mathcal{I}_0$ using Eqs. (15) and (16);
4 Construct neighborhood $\{\mathcal{N}_{\mathbf{x}_i}^{K_m}, \mathcal{N}_{\mathbf{y}_i}^{K_m}\}_{m=1,i=1}^{M,N}$ using $\mathcal{I}_0$;
5 Calculate cost $\{c_i\}_{i=1}^{N}$ using Eq. (14);
6 Determine $\mathcal{I}^*$ using Eqs. (15), (16) and (17).

precision and simultaneously decrease the recall, and vice versa. In our evaluate, we empirically set the default values as $\mathbf{K} = [4, 6, 8]$, $\tau = 0.2$, $\lambda = 0.9$ and 0.5 in the two iterations, respectively.

3.4 Computational Complexity

To search the K nearest neighbors for each feature point in S, the time complexity is close to $O((K+N)\log N)$ by using K-D tree (Bentley 1975). Thus the time complexity of Lines 1 and 4 in Algorithm 1 is about $O((K_M+N)\log N)$. This is the most time consuming step of our LPM. After obtaining the K_M neighborhood $\mathcal{N}_{\mathbf{x}_i}^{K_M}$, its corresponding K_m $(m < M)$ neighborhood $\mathcal{N}_{\mathbf{x}_i}^{K_m}$ can be directly obtained from $\mathcal{N}_{\mathbf{x}_i}^{K_M}$.

According to Eq. (14), the major cost of calculating $\{c_i\}_{i=1}^{N}$ in Lines 2 and 5 only involves some addition operation, and its time complexity is less than $O(MK_MN)$. Moreover, determining $\mathbf{p}$ and $\mathcal{I}$ using Eqs. (15) and (16) in Lines 3 and 6 cost $O(N)$ complexity. Therefore, the total time complexity of our LPM is about $O(MK_MN + (K_M+N)\log N)$. The space complexity of our LPM is $O(MK_MN)$ due to the memory requirement for storing the neighborhoods $\{\mathcal{N}_{\mathbf{x}_i}^{K_m}\}$ and $\{\mathcal{N}_{\mathbf{y}_i}^{K_m}\}$. Generally, $MK_M \ll N$, thus the time and space complexities of our method can be simply written as $O(N\log N)$ and $O(N)$, respectively. That is to say, our LPM has linearithmic time complexity and linear space complexity with respect to the scale of the given putative set. This is significant for large-scale problems or real-time applications.

4 Applications

This section describes how we can apply the locality preserving matching algorithm to several different visual tasks, including point set registration, visual homing and near-duplicate image retrieval, whose performance is in general dominated by the feature matching quality.

4.1 Non-rigid Point Set Registration

Point set registration aims to determine the right correspondences and/or to recover the spatial transformation between two sets of discrete points, e.g., $\{\mathbf{x}_i\}_{i=1}^{M_x}$ and $\{\mathbf{y}_j\}_{j=1}^{M_y}$. The registration problem is typically solved by using an iterative framework, where point correspondences are established to estimate the transformation, and vice versa (Ma et al. 2017). Here we use the LPM algorithm to establish reliable correspondences between two point sets and the transformation is estimated accordingly based on Tikhonov regularization (Micchelli and Pontil 2005).

4.1.1 Correspondence Construction

In the registration problem, the points are usually just spatial coordinates and extracted from shape contours. Therefore, they are not associated with local image descriptors such as SIFT. However, there are several descriptors capturing geometrical structures of shapes or point clouds can be made use of to establish putative correspondences, both in 2D and in 3D cases (Belongie et al. 2002; Rusu et al. 2009).

For 2D cases, the shape context (SC) (Belongie et al. 2002), which captures the distribution of neighboring points, has been widely used for shape matching. Consider two points $\mathbf{x}_i$ and $\mathbf{y}_j$, their SCs are histograms $\{p_i(l)\}_{l=1}^{L}$ and $\{q_j(l)\}_{l=1}^{L}$, with L being the dimension of the feature. The χ^2 distance is used to measure their difference $D(\mathbf{x}_i, \mathbf{y}_j)$:

$$D(\mathbf{x}_i, \mathbf{y}_j) = \frac{1}{2}\sum_{k=1}^{K}\frac{\left[p_i(k) - q_j(k)\right]^2}{p_i(k) + q_j(k)}. \tag{18}$$

After the distances of all point pairs, i.e. $\{D(\mathbf{x}_i, \mathbf{y}_j)\}_{i,j=1}^{M_x,M_y}$, have been computed, the Hungarian method (Papadimitriou and Steiglitz 1982) is applied to seek the putative correspondences between two point sets.

For 3D cases, the fast point feature histograms (FPFH) (Rusu et al. 2009) can be used as the feature descriptor. It is a histogram representing the underlying surface model properties that collects the pairwise pan, tilt and yaw angles between every point and its k-nearest neighbors, followed by a reweighting of the resultant histogram of a point with the neighboring histograms. The computation of the histogram is quite efficient, which has linear complexity with respect to the number of surface normals. The matching of FPFH descriptors is performed by a sample consensus initial alignment method.

After using some local feature descriptors to find correspondences, we obtain a putative set $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$. Next, our LPM algorithm is used to remove the false matches and establish reliable correspondences.

4.1.2 Transformation Estimation

The transformation $\mathbf{f}$, i.e. $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$ for a true correspondence $(\mathbf{x}_i, \mathbf{y}_i)$, can be characterized by a rigid or non-rigid model. The rigid model only involves a small number of parameters, and it is relatively easy and has been widely studied. Here we consider the more complex and general non-rigid model, which is required for many real world tasks. To estimate $\mathbf{f}$, it is natural to consider the supervised learning technique such as regression.

We model the transformation $\mathbf{f}$ by restricting it to lie within a specific functional space $\mathcal{H}$, namely a reproducing kernel Hilbert space (RKHS) (Micchelli and Pontil 2005), which is defined by a positive definite matrix-valued kernel Γ. In this paper we choose a diagonal decomposable Gaussian kernel $\Gamma(\mathbf{x}_i, \mathbf{x}_j) = e^{-\beta\|\mathbf{x}_i - \mathbf{x}_j\|^2} \cdot \mathbf{I}$ with β being a spread parameter and $\mathbf{I}$ being a 2×2 identity matrix. By using the L_2 loss on the data fitting and L_2 functional norm on the model complexity, the Tikhonov regularization minimizes the following regularized risk functional (Micchelli and Pontil 2005):

$$\mathcal{E}(\mathbf{f}) = \min \left\{ \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i)\|^2 + \mu \|\mathbf{f}\|_{\mathcal{H}}^2 \right\}. \tag{19}$$

According to the representer theorem (Micchelli and Pontil 2005), the optimal solution of the minimization problem in Eq. (19) is given by

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} \Gamma(\mathbf{x}, \mathbf{x}_i)\mathbf{w}_i, \tag{20}$$

with the coefficients $\{\mathbf{w}_i\}_{i=1}^{N}$ determined by a linear system:

$$(\mathbf{\Gamma} + \mu\mathbf{I})\mathbf{W} = \mathbf{Y}, \tag{21}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{N \times N}$ is the so-called Gram matrix with $\mathbf{\Gamma}_{ij} = e^{-\beta\|\mathbf{x}_i - \mathbf{x}_j\|^2}$, $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_N)^{\mathrm{T}}$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^{\mathrm{T}}$ are matrices of size $N \times 2$.

Note that there are two parameters need to be set, i.e. μ and β, where we fix them as $\mu = 3$ and $\beta = 0.8$ throughout this paper. In addition, to make the transformation estimation more robust, the VFC (Ma et al. 2014) algorithm is preferable. It generalizes the Tikhonov regularization to handle contaminated data under a Bayesian framework, which introduces a latent variable to resist outliers. Specifically, it assumes the noise of inlier to be Gaussian with zero mean and uniform standard deviation σ, and the outlier to be uniform distributed $1/a$ with a being the area of input image. Thus the likelihood is a mixture model:

$$p(\mathbf{X}, \mathbf{Y}|\theta) = \prod_{i=1}^{N} \left(\frac{\gamma}{2\pi\sigma^2} e^{-\frac{\|\mathbf{y}_i - \mathbf{x}_i - \mathbf{f}(\mathbf{x}_i)\|^2}{2\sigma^2}} + \frac{1-\gamma}{a} \right), \tag{22}$$

where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^{\mathrm{T}}$, $\theta = \{\mathbf{f}, \sigma^2, \gamma\}$ includes a set of unknown parameters to be solved, and γ is the mixing coefficient. By imposing a slow-and-smooth prior on the transformation: $p(\mathbf{f}) \propto e^{-\frac{\mu}{2}\|\mathbf{f}\|_{\mathcal{H}}^2}$, a MAP solution of θ can then be estimated, which is solved by using an iterative expectation-maximization approach. In particular, in the maximization step, the transformation is updated according to a regularized risk functional as:

$$\mathcal{E}(\mathbf{f}) = \min \left\{ \sum_{i=1}^{N} p_i \|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i)\|^2 + \mu\sigma^2 \|\mathbf{f}\|_{\mathcal{H}}^2 \right\}, \tag{23}$$

where p_i is posterior probability estimated in the expectation step, which indicates to what degree $(\mathbf{x}_i, \mathbf{y}_i)$ being an inlier. We refer to Ma et al. (2014) for more details on the VFC algorithm.

The above two steps of correspondence constructions and transformation estimation are iterated to obtain a reliable result. The iteration number is fixed to 10, and larger value is preferable if the input data is badly degraded.

4.2 Visual Homing

Visual homing aims to navigate a robot from an arbitrary starting position to some goal or home position solely based on visual information. It is usually solved by first matching local features in two panoramic images captured respectively at the current position and home position, and then transforming the correspondences into motion flows which are finally used to determine the homing vector (Zhao and Ma 2017). It has been verified that the robustness of visual homing methods is dominated by the presence and amount of false correspondences (Schroeter and Newman 2008). To remedy the degradation caused by mismatches, usually some heuristic methods are adopted to remove them. As in the non-rigid point set registration problem, we use LPM for robust feature matching and estimate the transformation $\mathbf{f}$ accordingly. The dense motion flow can then be directly obtained from $\mathbf{f}$, and we subsequently derive the focus-of-contraction (FOC) and focus-of-expansion (FOE) based on it to determine homing directions.

4.2.1 Feature Matching for Panoramic Image Pairs

In the visual homing problem, the panoramic image usually has reached 360° field of view horizontally, which is typically called "360 cylindrical panorama". The image plane of this type of image could be seen as a cylinder unrolled along with a certain vertical cutting line. Therefore, it is not appropriate to define the distance between pixels on the image plane by directly using the Euclidean distance, as in this case the distance will depend on the cutting line. For example, two nearby pixels on the cylinder will have large distance on

the image plane if they are located on the two sides of the cutting line. To address this issue, we define the two dimensional pixel position as a horizontal coordinate and a vertical coordinate, i.e $\mathbf{x} = (\mathbf{x}^h, \mathbf{x}^v)^{\mathrm{T}}$, where $\mathbf{x}^h$ and $\mathbf{x}^v$ are scalars. The Euclidean distance then can be modified as the following cylinder distance:

$$\begin{aligned}\mathrm{CylDist}^2(\mathbf{x}_i, \mathbf{x}_j) = &\left(\mathrm{CylDist}^h(\mathbf{x}_i^h, \mathbf{x}_j^h)\right)^2 \\ &+ \left(\mathrm{CylDist}^v(\mathbf{x}_i^v, \mathbf{x}_j^v)\right)^2, \end{aligned} \tag{24}$$

where the horizontal and vertical distances are defined as

$$\begin{aligned}\mathrm{CylDist}^h(\mathbf{x}_i^h, \mathbf{x}_j^h) = \min\big\{&|\mathbf{x}_i^h - \mathbf{x}_j^h|, |\mathbf{x}_i^h - \mathbf{x}_j^h - \mathbf{x}_{\max}^h|, \\ &|\mathbf{x}_i^h - \mathbf{x}_j^h + \mathbf{x}_{\max}^h|\big\}, \end{aligned} \tag{25}$$

$$\mathrm{CylDist}^v(\mathbf{x}_i^v, \mathbf{x}_j^v) = |\mathbf{x}_i^v - \mathbf{x}_j^v|, \tag{26}$$

with $\mathbf{x}_{\max}^h$ being the horizontal width of the image plane.

To conduct feature matching on panoramic image pairs by using our LPM approach in Algorithm 1, the only required modification is to construct the neighborhoods $\{\mathcal{N}_{\mathbf{x}}, \mathcal{N}_{\mathbf{y}}\}$ in Lines 1 and 4 by using the cylinder distance defined in Eq. (24) rather than the original Euclidean distance. This strategy enables our method to identify those true matches located on the two sides of the cutting line.

4.2.2 Motion Flow Estimation

After obtaining the feature correspondences, we focus on recovering the dense motion flow by estimating the transformation $\mathbf{f}$ from the matches. This can be achieved by using the regularization technique as described in the last section. The major difference is that the match $(\mathbf{x}_i, \mathbf{y}_i)$ should be converted to a motion vector $(\mathbf{u}_i, \mathbf{v}_i)$ according to the cylinder coordinate, e.g.,

$$\mathbf{u}_i = \mathbf{x}_i, \tag{27}$$

$$\mathbf{v}_i = (\mathbf{y}_i^h - \mathbf{x}_i^h + \alpha \mathbf{x}_{\max}^h, \mathbf{y}_i^v - \mathbf{x}_i^v), \tag{28}$$

where $\mathbf{u}_i$ is a position on an image plane, $\mathbf{v}_i$ is its associated motion vector, and parameter $\alpha \in \{0, \pm 1\}$ is used to wrap the horizontal displacement to $[-\mathbf{x}_{\max}^h/2, \mathbf{x}_{\max}^h/2]$. Then we could interpolate a motion field $\mathbf{f}$: $\mathbf{v}_i = \mathbf{f}(\mathbf{u}_i)$ for an inlier sample $(\mathbf{u}_i, \mathbf{v}_i)$ by using regularization technique.

4.2.3 Estimation of Homing Direction

It has been shown in previous work that the motion flow of a panoramic image pair has two singularities (Möller and Vardy 2006), which correspond to the FOC and FOE, respectively. In addition, these two singularities are separated by half horizontal width of the panoramic image.

The FOC and FOE have been used in many applications, including 3D environment reconstruction and estimation of time-to-contact in visual navigation. Specifically, in the visual homing literature, FOE corresponds to the homing direction, and FOC corresponds to the opposition of homing direction (Churchill and Vardy 2013; Zhao and Ma 2017). To localize the two singularities, a heuristic strategy has been proposed by detecting whether the SIFT features have grown or shrunk with respect to their sizes in the reference home image (Churchill and Vardy 2013).

Next, we introduce a method that uses the dense motion flow to determine the FOC and FOE. In general, the FOC and FOE should lie on the horizontal line $\mathbf{u}^v = \mathbf{u}_{\max}^v/2$ and are separated by $\mathbf{u}_{\max}^h$, with $\mathbf{u}_{\max}^h$ and $\mathbf{u}_{\max}^v$ being the horizontal width and vertical width of the panoramic image. Therefore, there is no significant difference about the estimation of these two singularities. In the following, we will only focus on the estimation of FOC, and the generalization to FOE is straightforward.

After obtaining the motion flow $\mathbf{f}(\mathbf{u})$ in Eq. (20), finding out the analytical solution of its singularities is impossible or very difficult. Instead, some numerical method can be adopted to seek an approximate solution. Formally, since FOC lies on the horizontal line $\mathbf{u}^v = \mathbf{u}_{\max}^v/2$, we define a 1D function

$$g(\mathbf{u}^h) \triangleq \mathbf{f}([\mathbf{u}^h, \mathbf{u}_{\max}^v/2]). \tag{29}$$

Clearly, $g(\theta)$ is continuous and differentiable, and the singularities correspond to the points whose left and right local neighborhoods have different signs. We give the formal definition of the FOC as below.

Definition: Focus of contraction (FOC) Focus of contraction $\mathbf{u}_{\mathrm{FOC}}^h$ is the point satisfying that: (i) $g(\mathbf{u}_{\mathrm{FOC}}^h) = 0$; and (ii) $\exists\ \epsilon > 0$ satisfies that $g(\mathbf{u}^h) > 0$ for any $\mathbf{u}^h$ in the left ϵ-neighborhood of $\mathbf{u}_{\mathrm{FOC}}^h$ and $g(\mathbf{u}^h) < 0$ for any $\mathbf{u}^h$ in the right ϵ-neighborhood of $\mathbf{u}_{\mathrm{FOC}}^h$.

We use a coarse-to-fine grid search strategy to find the optimal solution of FOC, which is able to achieve arbitrary precision. In visual homing literature, usually all panoramic images have identical compass orientation by preprocessing. By converting the coordinate to angle, the homing direction can then be obtained as follows:

$$\theta_{\mathrm{homing}} = \theta_{\mathrm{FOC}} = \frac{2\pi \cdot \mathbf{u}_{\mathrm{FOC}}^h}{\mathbf{u}_{\max}^h}. \tag{30}$$

With this direction, we can fulfill the visual homing task and navigate a robot back to its reference home position.

4.3 Near-Duplicate Image Retrieval

Given a query image, the goal of near-duplicate image retrieval is to retrieve the images of the same object or scene from a large database and return a ranked list. It is typically solved by first calculating the similarities between the query image and all the images in the database, and then sorting the similarities to return a ranked list (Chen et al. 2016). In this procedure, the similarity between two images could be determined by the similarity of features contained in them, while the feature similarity is usually measured by feature matching result. Thus LPM is desirable to produce reliable performance.

For the image registration problem, we are given an image database $\mathcal{S} = \{I_i\}_{i=1}^{N}$ together with a similarity function s: $I \times I \rightarrow \mathbb{R}^+$ that assigns each pair of images with a positive similarity value. In this paper, the similarity function s is defined as follows: we first establish SIFT putative feature correspondences and subsequently use our LPM to remove false matches, the similarity $s(I_i, I_j)$ is then assigned by the number of preserved matches on the two given images I_i and I_j. Therefore, we obtain an $N \times N$ similarity matrix $\mathbf{S}$ related to the whole image database, where $\mathbf{S}_{ij} = s(I_i, I_j)$.

Given a query image I_i, we aim to search the most similar images from a set of known database images $\mathcal{S}$. By sorting the values $\{\mathbf{S}_{in}\}_{n=1}^{N}$ in decreasing order, we obtain a ranking of database images according to their similarities to the query, e.g., the most similar database image has the highest value and is listed first. Usually, the first M ($M \ll N$) images are returned as the most similar ones to the query.

5 Experimental Results

In order to evaluate the performance of our LPM, we first conduct experiments on feature matching for various real image pairs, and then apply it to the visual tasks, say non-rigid point set registration, visual homing and near-duplicate image retrieval. The open source VLFeat toolbox (Vedaldi and Fulkerson 2010) is employed to determine the putative correspondence of SIFT (Lowe 2004) and to search the K nearest neighbors using K-D tree. The experiments are performed on a desktop with 3.0 GHz Intel Core CPU, 8 GB memory, and C++ code. Besides, all the codes were implemented without special optimization such as parallel computing or streaming SIMD extensions.

5.1 Results on Feature Matching

In this section, we focus on establishing feature correspondences for real images. To this end, we first test the performance of our LPM on several representative image pairs undergoing different types of image transformations, and then provide quantitative results on five datasets as follows:

- *VGG* (Mikolajczyk et al. 2005). The dataset contains 40 image pairs either of planar scenes or captured by a camera in a fixed position during acquisition. Therefore, the image pairs in this dataset always obey homography. The ground truth homographies are supplied by the dataset.
- *DAISY* (Tola et al. 2010). The dataset consists of wide baseline image pairs with ground truth depth maps, including two short image sequences and several individual image pairs. We create 52 image pairs in total for evaluation, including all the individual pairs, and for the two sequences we create all possible image pairs from them.
- *DTU* (Aanæs et al. 2016). The dataset is originally designed for multiple view stereo evaluation, which involves a lot of different scenes with a wide range of objects. Each scene has been taken from 49 or 64 positions, and the ground truth camera positions and internal camera parameters have been found with high accuracy. We choose two scenes from the dataset (i.e., *Frustum* and *House*) and create 131 image pairs in total for evaluation, which consist of those pairs with large viewpoint changes in the scenes.
- *RS*. The dataset consists of 156 remote sensing image pairs including color-infrared, SAR and panchromatic photographs. The feature matching task for such image pairs typically arises in image mosaic, positioning and navigating, change detection, etc.
- *Retina*. The dataset consists of 65 retinal image pairs undergoing non-rigid transformations. The feature matching task aims to align multiple retina images together and integrate information from them for comprehensive understanding and better diagnoses of retinal diseases.

For the first three publicly available datasets, the correctness of each feature correspondence in a putative set is determined based on the ground truth information supplied by the datasets. The other two datasets are collected by ourselves, where the ground truth correspondence is established with respect to a benchmark prepared in advance, before conducting any experiments, to ensure objectivity; in particular, the correctness of each putative correspondence in each image pair is checked manually.

5.1.1 Results on Representative Image Pairs

Ten representative image pairs undergoing different types of transformations are used for test, as shown in Fig. 4. The "*Land*" pair is an aerial photograph pair involving only linear (e.g., rigid or affine) transformation, which is typically arisen in image stitching. The "*Fox*" and "*Book*"

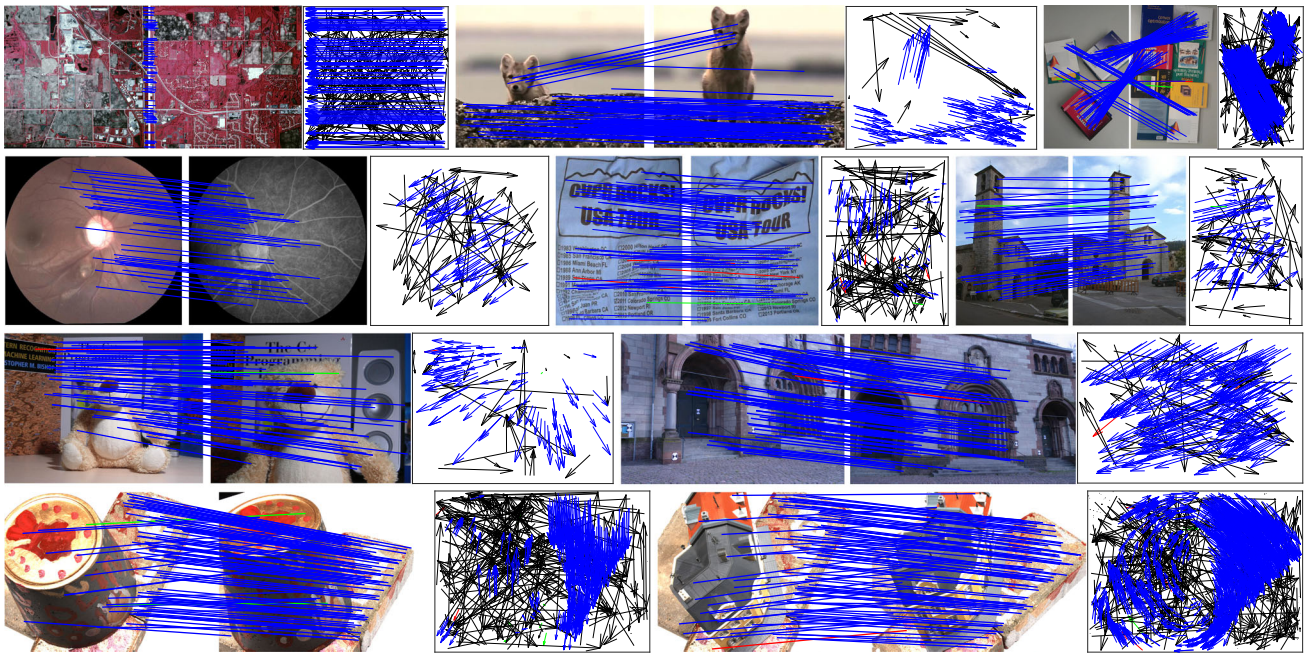

Fig. 4 Feature matching results of our LPM on 10 representative image pairs. From top to bottom and left to right: *Land*, *Fox*, *Book*, *Retina*, *T-shirt*, *Church*, *Bear*, *Herzjesu*, *Frustum* and *House*. The ratio of outliers in the 10 image pairs are 40.81%, 85.93%, 76.14%, 49.50%, 43.81%, 57.26%, 65.94%, 78.71%, 65.96% and 78.49%. The head and tail of each arrow in the motion field correspond to the positions of feature points in two images (blue = true positive, black = true negative, green = false negative, red = false positive). For visibility, in the image pairs, at most 100 randomly selected matches are presented, and the true negatives are not shown. Best viewed in color (Color figure online)

pairs undergo piecewise linear transformation, which is often arisen in image/video retrieval. The "*Retina*" and "*T-shirt*" pairs involve non-rigid motions, which frequently happens in medical image registration. The rest five pairs are wide baseline image pairs, which is typically arisen in structure-from-motion. For each group of results, the left image pair schematically shows the matching result, and the right motion field provides the decision correctness of each correspondence in the putative set. From the results, we see that our LPM can always produce satisfying results and very few putative matches are misjudged.

We also provide quantitative comparison on the 10 image pairs with four state-of-the-art matching methods such as RANSAC (Fischler and Bolles 1981), ICF (Li and Hu 2010), GS (Liu and Yan 2010), GMS (Bian et al. 2017), BD (Lipman et al. 2014) and MR-RPM Ma et al. (2017). All these algorithms are implemented based on publicly available codes, and we have tried our best to tune their parameters to achieve their best performance. The matching performance is characterized by precision and recall,[3] as shown in Table 1. From the results, we see that for rigid matching such as in the *Land* pair, all methods except BD perform quite well. BD uses a piecewise deformation model with relatively weak global constraints which is sensitive to outliers, leading to its inferior performance. RANSAC cannot work well when the image transformation does not satisfy a parametric model, such as in the *Fox*, *Book* and *T-shirt* pairs. ICF and MR-RPM use a slow-and-smooth prior, which will probably fail if the motion field involves large depth discontinuity or motion inconsistency, such as in the *Fox*, *Book* and wide baseline image pairs. GS often has high precision and low recall, because it cannot automatically estimate the factor for affinity matrix and it is not affine-invariant. GMS does not achieve the best performance, due to that we use it with the same input as the other methods, even though it was designed with a very large number of low-quality matches instead. In addition, the consensus of neighborhood topology demonstrated in Fig. 1 cannot be well addressed either. In comparison, our LPM does not suffer from all these problems, which demonstrates its generality and ability to handle various matching problems.

[3] For real-world tasks such as multiple view stereo and SLAM, a better metric would be to use the inliers to retrieve the camera pose from stereo images and evaluate their accuracy (Bian et al. 2017). However, such camera pose estimation usually relies on an additional robust estimator such as RANSAC, which may not directly characterize the matching performance. Therefore, for the purpose of general feature matching, we only use precision and recall to characterize the performance.

5.1.2 Results on Image Datasets

To provide a comprehensive quantitative evaluation of our LPM, we next conduct experiments on five feature matching datasets, such as *VGG*, *DAISY*, *DTU*, *RS* and *Retina*.

Table 1 Precision and recall pair (%) of RANSAC, ICF, GS, GMS, BD, MR-RPM and LPM on the 10 image pairs shown in Fig. 4

	RANSAC Fischler and Bolles (1981)	ICF Li and Hu (2010)	GS Liu and Yan (2010)	GMS Bian et al. (2017)	BD Lipman et al. (2014)	MR-RPM Ma et al. (2017)	LPM
Land	(100.0, 100.0)	(98.23, 100.0)	(100.0, 90.09)	(86.78, 94.59)	(51.81, 38.74)	(97.37, 100.0)	(100.0, 100.0)
Fox	(100.0, 87.93)	(85.93, 100.0)	(100.0, 89.66)	(97.37, 97.37)	(86.02, 70.18)	(100.0, 89.66)	(100.0, 100.0)
Book	(100.0, 44.19)	(82.62, 91.20)	(100.0, 82.22)	(92.62, 60.00)	(77.60, 26.37)	(99.79, 82.57)	(98.76, 98.94)
Retina	(100.0, 98.00)	(74.63, 100.0)	(96.15, 100.0)	(83.33, 70.00)	(58.70, 54.00)	(100.0, 90.00)	(100.0, 100.0)
T-shirt	(97.44, 38.34)	(43.81, 100.0)	(91.49, 86.87)	(79.21, 80.81)	(38.55, 32.32)	(98.99, 98.99)	(96.07, 98.99)
Church	(94.52, 100.0)	(91.67, 63.77)	(91.78, 97.10)	(86.76, 83.10)	(53.70, 40.85)	(98.33, 85.51)	(100.0, 98.59)
Bear	(95.88, 93.00)	(91.23, 52.00)	(95.96, 95.00)	(88.75, 71.00)	(76.92, 96.00)	(96.08, 98.00)	(99.00, 99.00)
Herzjesu	(98.11, 80.83)	(93.59, 37.82)	(98.73, 80.31)	(92.02, 89.60)	(79.27, 67.36)	(97.42, 97.93)	(98.99, 100.0)
Frustum	(98.43, 86.01)	(99.11, 25.46)	(99.04, 76.04)	(97.31, 91.28)	(70.92, 63.76)	(99.20, 85.09)	(99.54, 98.40)
House	(98.77, 82.51)	(100.0, 60.90)	(100.0, 58.93)	(95.62, 93.92)	(78.47, 70.91)	(97.16, 96.07)	(99.26, 99.72)
Average	(98.31, 81.08)	(86.08, 73.11)	(97.32, 85.62)	(89.98, 83.17)	(67.20, 56.05)	(98.43, 92.38)	*(99.16, 99.36)*

The average numbers of putative SIFT correspondences on the five datasets are about 693.17, 1475.60, 545.99, 445.34 and 69.03, respectively. The initial inlier percentage, precision, recall and runtime statistics of the seven algorithms are reported in Fig. 5. From the results, we see that LPM does not have obvious advantage in terms of precision compared with other methods, especially for RANSAC; however, it can always produce the best recall index. We give an explanation as follows. For those scenes suffer from large depth discontinuity, motion inconsistency or non-rigid deformation, existing methods preserve only a part of the whole true correspondences that obey some specifical geometrical constraints (e.g., motion models). By contrast, our LPM does not require a motion model between image pairs, therefore it works well in presence of non-rigid deformations or multiple motion fields.

We also report the runtime statistics in the last column of Fig. 5. From the results, we see that GMS and our LPM are very effective, which are more than two orders of magnitude faster than the other state-of-the-art methods. The runtime of GMS is about one third of LPM. In particular, our average runtime on the five datasets is merely about 12.9 ms, 19.8 ms, 7.55 ms, 8.75 ms, and 2.06 ms, respectively, making it ideal for real-time applications.

5.2 Results on Point Set Registration

We next evaluate our LPM for point set registration on both 2D shape contour and 3D point cloud. For the 2D case, we use the synthesized data created in Chui and Rangarajan (2003) and Zheng and Doermann (2006), which consists of two shape patterns such as a *fish* pattern and a *Chinese character* pattern with both about 100 points. The dataset involves several different types of data degeneration, and each degeneration type involves several different degeneration levels where each level contains 100 samples. For the 3D case, we consider a surface correspondence benchmark (Kim et al. 2011) and choose a point cloud pair representing a *wolf* with about 5000 points in different poses for evaluation. To make the dataset more challenging, we add two types of degeneration including occlusion and outlier to the point cloud pair with different degeneration levels where each level contains 50 samples. The ground truth correspondences are supplied by the datasets.

5.2.1 Results on 2D Shape Contour

Some qualitative results of our method on the two shape patterns are presented in Fig. 6. Our goal is to align a model point set (blue pluses) onto a target point set (red circles). We organize the results in every two rows: the first row is the initial point sets, the second row is the corresponding registration results, and the degree of degradation increases from left to

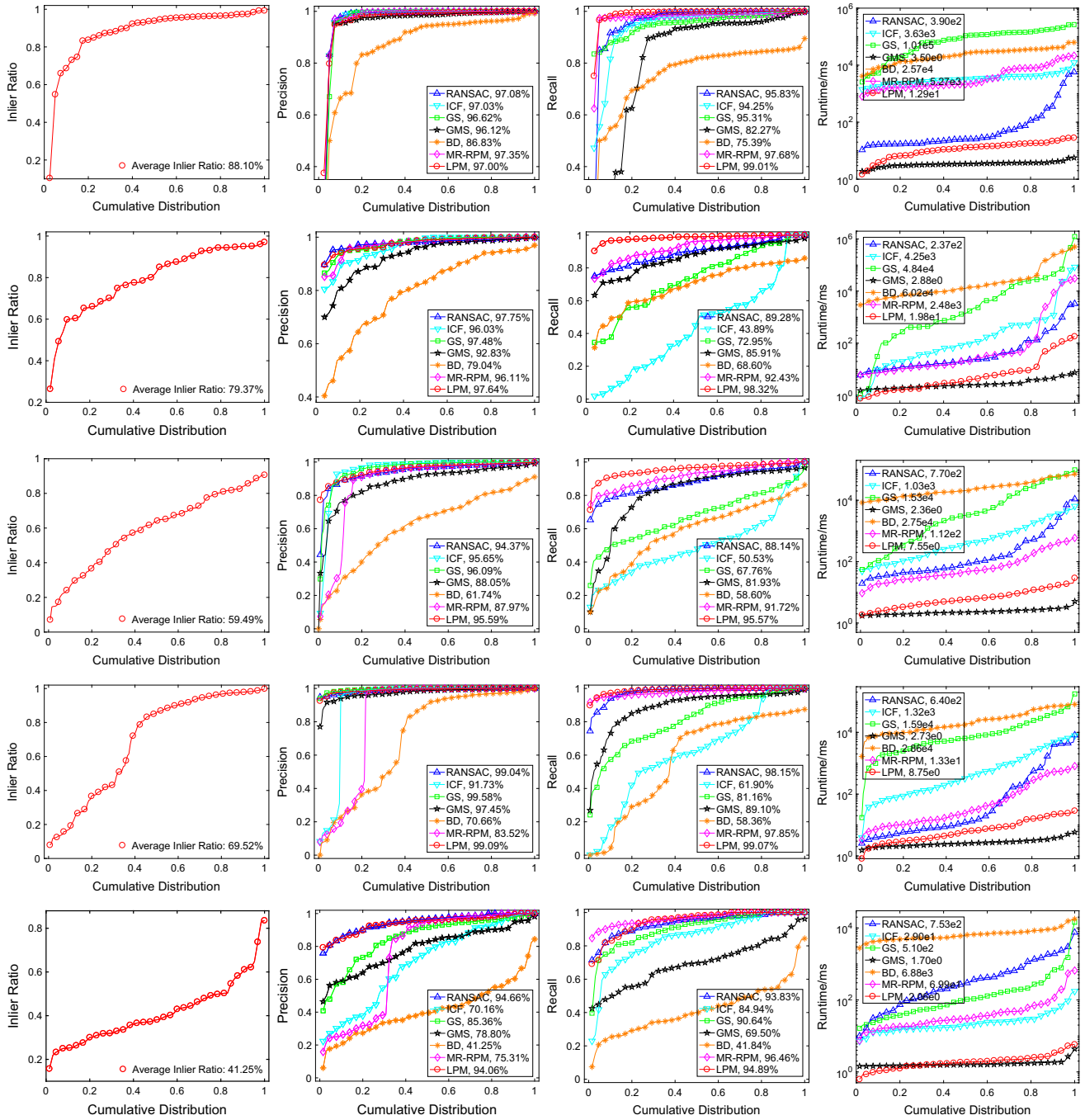

Fig. 5 Quantitative comparisons of RANSAC, ICF, GS, GMS, BD, MR-RPM and LPM on five datasets, such as (Top to Bottom) *VGG*, *DAISY*, *DTU*, *RS* and *Retina*. (Left to Right) Initial inlier ratio, precision, recall, and run time with respect to the cumulative distribution

right. From the results, we see that the *fish* pattern is relatively simple and the local neighborhood structures among contour points are preserved well even in case of large degree of deformation or occlusion, and hence our method is able to always produce almost perfect alignments. By contrast, the points of the *Chinese character* pattern are spread out on the shape, which affects the locality preserving under large degradation. The matching performance then degrades gradually, but it remains acceptable, even for large degradation. The iterative correspondence construction and transformation estimation process typically converges in about 5 iterations on this dataset.

We also provide a quantitative comparison on the dataset with six state-of-the-art registration methods, including SC (Belongie et al. 2002), TPS-RPM (Chui and Rangarajan 2003), RPM-LNS (Zheng and Doermann 2006), GMMREG

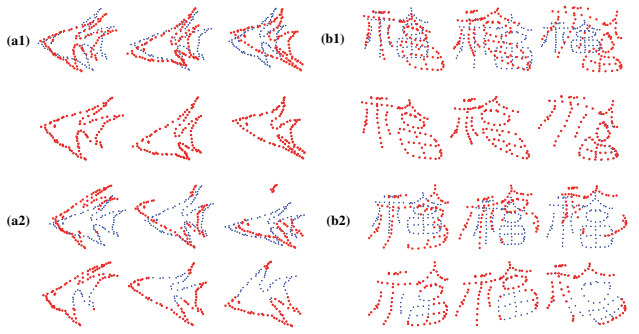

Fig. 6 Registration results of our method on the *fish* (left) and *Chinese character* (right) patterns, with (a1, b1) deformation and (a2, b2) occlusion presented in every two rows. The goal is to align the model point sets (blue pluses) onto the target point sets (red circles). For each group, the first row is the initial point sets, the second row is the corresponding registration results, and the degree of degradation increases from left to right (Color figure online)

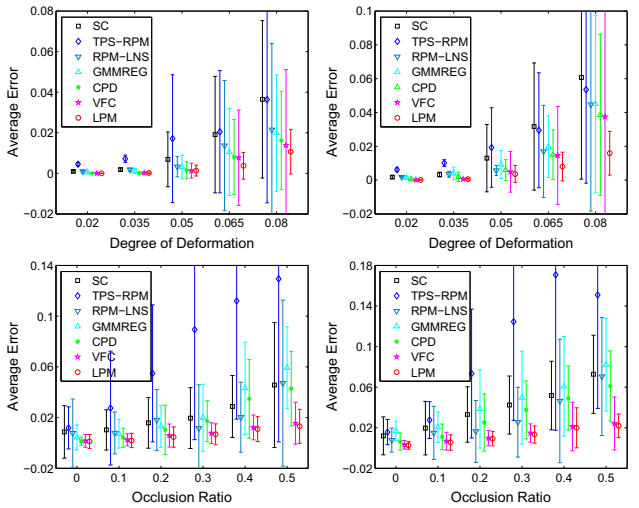

Fig. 7 Comparison of LPM with SC, TPS-RPM, RPM-LNS, GMMREG, CPD and VFC on the *fish* (left) and *Chinese character* (right) patterns. The error bars indicate the registration error means and standard deviations over 100 trials

(Jian and Vemuri 2011), CPD (Myronenko and Song 2010) and VFC (Ma et al. 2014), which are implemented based on publicly available codes. The registration error between two point sets is characterized by the average Euclidean distance of the ground truth correspondences between the warped model set and the target set. For each degradation level in a certain degradation type, we then compute the mean and standard deviation of the registration errors on all 100 samples for performance comparison. The statistic results are reported in Fig. 7. From the results, we see that all the seven algorithms perform well at low degradation levels, and the performance degrades as the degradation level increases, especially for SC and TPS-RPM. GMMREG and CPD do not consider local shape features for correspondence estimation, while RPM-LNS does not use robust estimator for transformation estimation; they all cannot achieve satisfying results in case of large degradation levels. VFC and our LPM do not suffer from such problems and hence perform better. The major difference between our LPM and VFC is that we use an additional locality preserving constraint to filter out false correspondence; our almost consistently best results demonstrate that the locality preserving does play an important role for improving the registration performance.

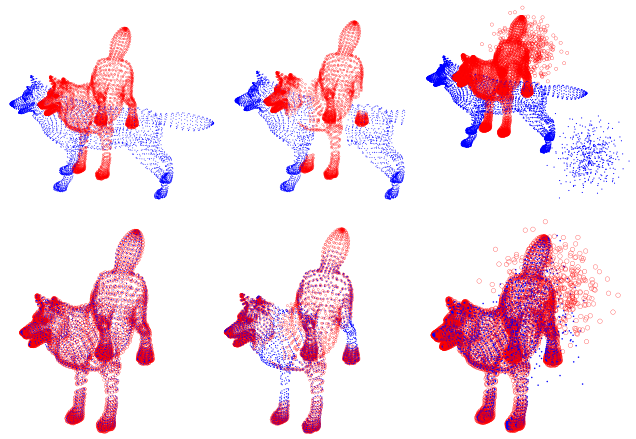

Fig. 8 Registration results of our method on the *wolf* pattern involving deformation (left column), occlusion (middle column), and outlier (right column). For each group, the top figure is the initial point sets, and the bottom is the corresponding result

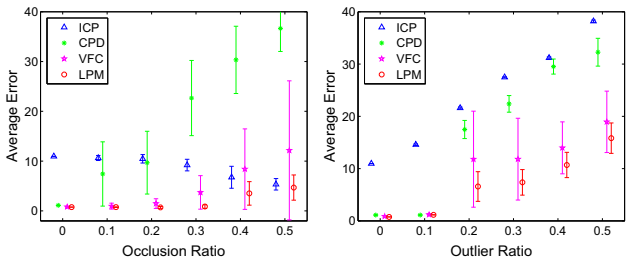

Fig. 9 Comparison of LPM with ICP, CPD and VFC on the *wolf* pattern. The error bars indicate the registration error means and standard deviations over 50 trials

5.2.2 Results on 3D Point Cloud

We further evaluate our LPM for registration of 3D point cloud pairs. The results are given in Fig. 8, where the tests on non-rigid deformation, occlusion and outlier are shown in the left, middle and right columns, respectively. In addition, to make the data more challenging, we remove a part of points on both the model and target patterns in the occlusion test, and add outliers on both the model and target patterns in the outlier test. From the results, we see that our method again is able to produce almost perfect alignments.

We also provide a quantitative comparison on the two point cloud pairs with three representative methods such as

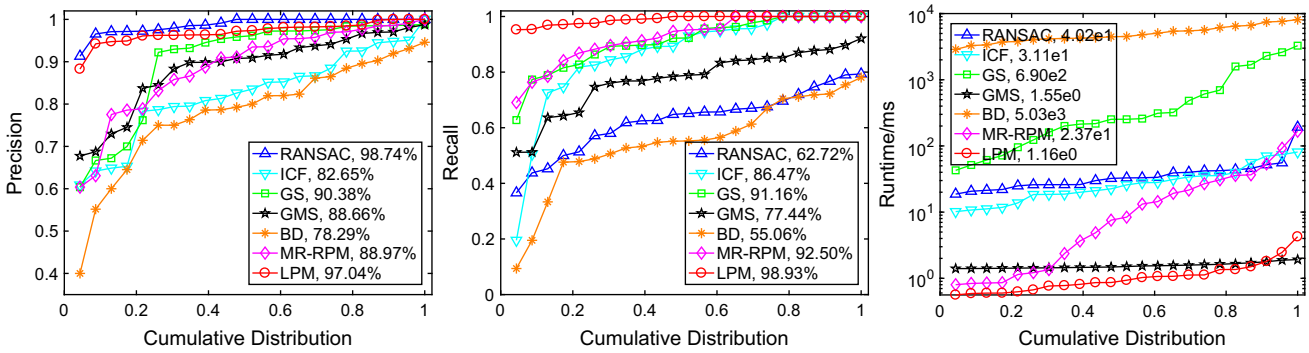

Fig. 10 Precision (left), recall (middle) and runtime (right) of RANSAC, ICF, GS, GMS, BD, MR-RPM and LPM on a panoramic dataset (Liu et al. 2013)

ICP (Besl and McKay 1992), CPD (Myronenko and Song 2010) and VFC (Ma et al. 2014). We calculate the average registration error on each degradation type and each degradation level, and report the statistic results in Fig. 9. Clearly, our method yields the best performance, which demonstrates the generality and effectiveness of our method for handling both 2D and 3D point set registration problems. Note that ICP is consistently unable to generate satisfying results as it relies on a rigid transformation model, while the testing data here involves large degree of non-rigid deformation. In addition, the average registration error of ICP in the occlusion test decreases as the occlusion ratio grows. This is because that the degree of non-rigid deformation becomes smaller as more parts are removed from the shape pattern.

5.3 Results on Visual Homing

We evaluate our LPM on a widely used panoramic image database[4] in the visual homing literature (Churchill and Vardy 2013; Liu et al. 2013). It contains a collection of omnidirectional and unwrapped images in an indoor environment, together with ground truth for positions where the images were collected. The database includes several scenes, and the collected images are of size 561×81, 583×81 or 295×41. The actual intervals between two nearest positions for image collection are 30 cm. As the image resolution is low, we modify the default parameter of SIFT to generate more features. Specifically, the number of layers in each octave is increased from default 3 to 6.

To validate the effectiveness of our LPM on visual homing, we use three types of methods for quantitative comparison including homing in scale-space (HiSS) (Churchill and Vardy 2013), visual servoing-based methods (Liu et al. 2013), and motion flow interpolation by smoothness prior (MFI-SP) (Zhao and Ma 2017). Note that in (Liu et al. 2013), it has introduced four variants of homing methods: (i) bearing-only visual servoing; (ii) scale-only visual servoing; (iii) scale and bearing visual servoing; (iv) simplified scale-based visual servoing (SSVS). For these four variants, we only report the results of SSVS due to its superior performance and efficiency compared to the other three methods, and it has also been suggested as the first choice by the original authors according to their comprehensive evaluation.[5] In addition, as in Churchill and Vardy (2013); Liu et al. (2013), we use total average angular error (TAAE), minimal error (Min), maximal error (Max) and standard variation of error (StdVar) to evaluate the homing performance. For all the metrics, smaller values indicate better results.

5.3.1 Feature Matching on Panoramic Images

We first test our method for feature matching on panoramic images. The ground truth is established by manually checking of each putative match in each image pair, and we only choose 23 image pairs with large viewpoint changes for quantitative evaluation. This can not only make the test data more challenging, but also simplify the construction of ground truth.

The matching results of different methods are reported in Fig. 10. The average inlier ratio in the putative sets is about 78.18%, and the average number of putative matches is about 113.5. From the results, we see that our LPM clearly has the best precision and recall tradeoff. We see that RANSAC has the best precision, but simultaneously has the worst recall. This is due to that the panoramic pair does not exactly satisfy a parametric model, and hence only a part of the true matches can be identified. The missing matches will inevitably affected the subsequent dense motion field interpolation. We also provide the runtime statistics of different

[4] http://www.ti.uni-bielefeld.de/html/research/avardy/index.html.

[5] As different feature extraction used in this paper, the performance of HiSS (Churchill and Vardy 2013) and SSVS (Liu et al. 2013) is not exactly the same as reported in the original papers. In addition, the reimplemented SSVS method in this paper does not contain the mismatch removal introduced in (Liu et al. 2013).

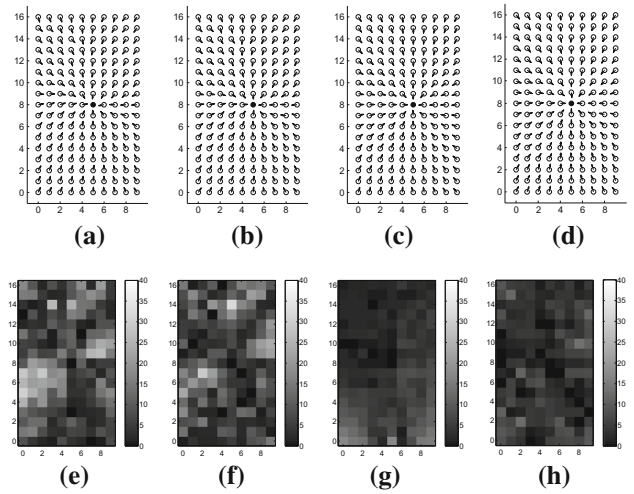

Fig. 11 Homing vectors and error analysis referring to grid position (5, 8) in dataset *A1originalH*. **a**–**d** Homing vectors. The solid circle in each figure is the homing position. **e**–**h** Angular errors for each position (unit: degree). **a** HiSS (Churchill and Vardy 2013), **b** SSVS (Liu et al. 2013), **c** MFI-SP (Zhao and Ma 2017), **d** LPM, **e** HiSS, 9.43°, **f** SSVS, 8.05°, **g** MFI-SP, 4.66°, and **h** LPM, 4.21°

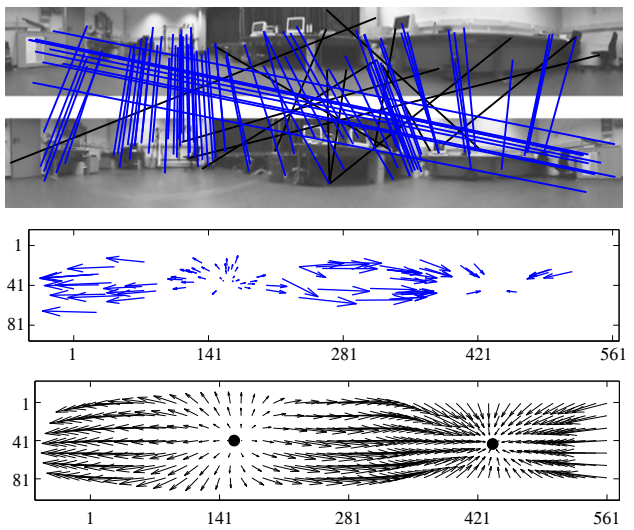

Fig. 12 Schematic illustration of feature matching and dense motion flow estimation results of our LPM. Top: the feature matching result, where blue and black lines indicate the preserved inliers and removed outliers. Middle: the corresponding sparse motion flow samples. Bottom: dense motion flow estimated based on the persevered matches by our LPM, where black dots are localized FOC and FOE (Color figure online)

methods on the rightmost figure in Fig. 10. The average runtime of GMS and our LPM is less than 2 ms, which is far less than the other methods.

5.3.2 Visual Homing on Panoramic Images

We further test our method for visual homing. Figure 11 provides some intuitive results of different methods on the

Table 2 The statistics of visual homing error by using different methods (unit: degree)

Database	HiSS Churchill and Vardy (2013)				SSVS Liu et al. (2013)				MFI-SP Zhao and Ma (2017)				LPM			
	TAAE	Min	Max	Var	TAAE	Min	Max	Var	TAAE	Min	Max	Var	TAAE	Min	Max	Var
AloriginalH	14.67	8.05	36.40	5.42	12.59	6.50	28.36	*4.17*	7.78	*3.06*	26.70	4.75	*7.61*	3.17	*25.62*	4.72
CHall1H	11.69	8.05	18.84	*1.81*	15.94	10.79	28.84	3.32	7.27	3.67	16.80	2.46	7.22	*3.53*	*16.61*	2.29
CHall2H	15.75	10.53	27.59	2.92	24.69	12.46	55.16	7.22	13.89	*7.42*	29.24	4.31	*13.55*	7.93	*27.86*	4.25
KitchenH	21.76	12.86	47.62	6.36	24.29	13.75	51.65	7.10	20.10	11.94	42.05	5.90	*19.61*	*10.78*	*39.54*	5.82
Roeben1H	28.95	10.96	61.07	12.27	26.70	8.99	64.64	12.82	24.52	8.33	60.07	12.09	*23.15*	*7.81*	*59.33*	*11.78*

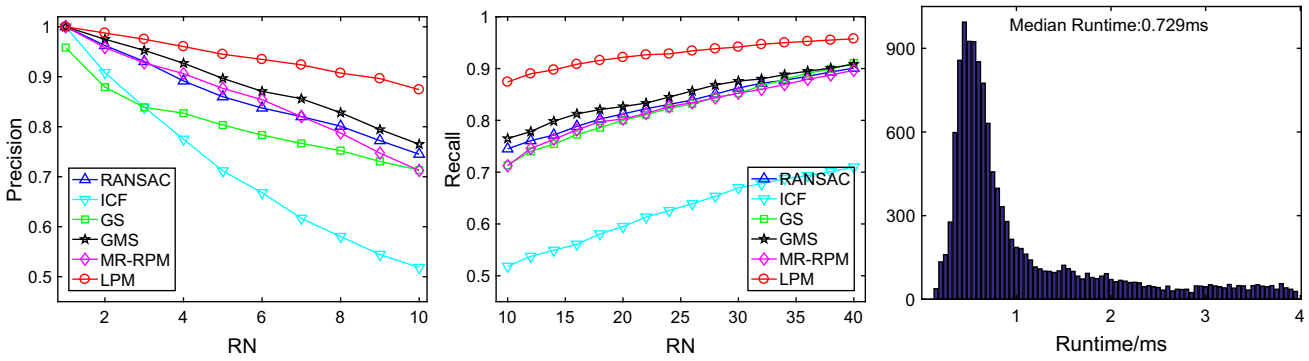

Fig. 13 Precision (left) and recall (middle) of RANSAC, ICF, GS, GMS, MR-RPM and LPM with respect to RN, i.e, the required number of images to be retrieved for a given image. Right: runtime statistics of LPM over 14, 280 trials

homing performance. We take position (5, 8) of *A1originalH* dataset as the reference home position, and the homing vectors calculated from other images by using the four methods are shown in Fig. 11a–d. The corresponding average angular errors for each position of the dataset are shown in Fig. 11e–h. From the results, it can be seen that our LPM can provide more accurate homing results.

We schematically show our feature matching result and estimated dense motion flow on a typical image pair in Fig. 12. Clearly, all the inliers and outliers in the putative set are correctly distinguished. In addition, the estimated dense motion flow, FOC and FOE are consistent with the real motion flow. In this example, the FOC and FOE are about (437, 41) and (157, 41), respectively. Usually, it takes about 10 milliseconds for our method to localize the FOE/FOC.

The statistics of the homing vector errors of all methods on the test database are reported in Table 2. We can see that our LPM in general can produce better or comparable results compared with the other state-of-the-art methods.

5.4 Results on Near-Duplicate Image Retrieval

We also test our LPM for near-duplicate image retrieval and compare it with RANSAC, ICF, GS, and MR-RPM on the California-ND dataset (Jinda-Apiraksa et al. 2013). We select all of the classes that have 10 or more images, and for each class we randomly select 10 images for evaluation which results in 14, 280 image pairs in total. The sizes of the test images are all 1024×768. We run the matching algorithms and utilize the number of preserved matches as the similarity between image pairs, and then return a ranked list for a provided image according to its similarities with every other image in the dataset. The performance is also characterized by precision and recall. We denote the required image number to be retrieved for a provided image as RN. The precision is valid for $RN \leq 10$ and the recall is valid for $RN \geq 10$, because each class contains 10 images.

The statistic retrieval results of the four methods in the dataset are presented on the left two figures of Fig. 13. Our LPM evidently outperforms all other methods and obtains the best precision and recall, followed by RANSAC and MR-RPM. Specifically, the average retrieved correct image numbers of RANSAC, ICF, GS, GMS, MR-RPM and our LPM for $RN = 10$ are approximately 7.45, 5.18, 7.13, 7.65, 7.13 and 8.74, respectively. The runtime statistics of LPM on all the 14, 280 image pairs is provided on the right of Fig. 13, where the median runtime is about 0.729 ms.

We also measure the retrieval performance of the so-called bulls-eye score (Bai et al. 2010), which is defined as the ratio of the total number of correct images among the 20 most similar images to the highest possible number (i.e, 10). The best possible rate is 100%. The bulls-eye scores of RANSAC, ICF, GS, GMS, MR-RPM and our LPM are approximately 81.25%, 59.50%, 80.00%, 82.67, 80.25% and 92.17%, respectively. Our method again evidently showcases the best performance.

6 Discussion and Conclusion

In this paper, we proposed a novel mismatch removal method for robust feature matching. It works based on a general characteristic that the neighborhood structures of feature correspondences between two images of the same scene should be similar. We formulated this idea into a mathematic model and derived a closed-form solution with linearithmic time complexity. The qualitative and quantitative results on feature matching as well as other real-world tasks demonstrated that our method can handle a variety of matching problems. More importantly, it can identify outliers from over 1, 000 putative matches in only a few milliseconds, which is more than two orders of magnitude faster than state-of-the-art methods. Since our method is very fast, it can be used to provide a quick initialization for more complicated problem-specific

matching algorithms, for instance RANSAC, to estimate the epipolar geometry between wide baseline image pairs.

For most existing feature matching methods, there is a critical prerequisite that the putative set should not contain a huge number of outliers. To ensure relatively high inlier ratio, typical strategies for putative set construction often falsely discard a part of true matches. This will be problematic if image pairs themselves contain very few true matches, for example, matching low-overlap images (*e.g.,* remote sensing images for mosaic) or low-quality images (*e.g.,* medical images for fusion). To address this issue, we have designed a guided matching strategy based on our preliminary LPM method (Ma et al. 2017) in the context of solving the remote sensing image registration (Ma et al. 2018) and visual homing (Ma et al. 2018) problems. It uses the matching result on a small putative set with a high inlier ratio to guide the matching on a large putative set with a (very) low inlier ratio. Therefore, it is able to address the matching problem when the putative set is constructed from a large number of (cheap) features (possibly with high noise) and is thus semi-dense (Bian et al. 2017; Lin et al. 2018). This guided matching strategy can be directly applied to our LPM in this work to boost the number of true matches.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant Nos. 61773295, 61503288, 61501413, 41501505 and 61772512, and the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant No. 2016IRS15.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Aanæs, H., Jensen, R. R., Vogiatzis, G., Tola, E., & Dahl, A. B. (2016). Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, *120*(2), 153–168.

Adamczewski, K., Suh, Y., Mu Lee, K.: Discrete tabu search for graph matching. In: Proceedings of the 10th European conference on computer vision, pp. 109–117 (2015)

Bai, X., Yang, X., Latecki, L. J., Liu, W., & Tu, Z. (2010). Learning context-sensitive shape similarity by graph transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(5), 861–874.

Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(24), 509–522.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*(9), 509–517.

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256.

Bian, J., Lin, W. Y., Matsushita, Y., Yeung, S. K., Nguyen, T. D., Cheng, M. M.: GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In: Proceedings of the 10th European conference on computer vision pattern Recognition, pp. 2828–2837 (2017)

Boughorbel, F., Koschan, A., Abidi, B., & Abidi, M. (2004). Gaussian fields: A new criterion for 3d rigid registration. *Pattern Recognition*, *37*(7), 1567–1571.

Chen, J., Wang, Y., Luo, L., Yu, J. G., & Ma, J. (2016). Image retrieval based on image-to-class similarity. *Pattern Recognition Letters*, *83*, 379–387.

Cho, M., Lee, K. M.: Mode-seeking on graphs via random walks. In: Proceedings of the European conference on computer vision pattern recognition, pp. 606–613 (2012)

Cho, M., Lee, K. M.: Progressive graph matching: Making a move of graphs via probabilistic voting. In: Proceedings of the European conference on computer vision pattern recognition, pp. 398–405 (2012)

Chui, H., & Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, *89*, 114–141.

Chum, O., Matas, J.: Matching with PROSAC - progressive sample consensus. In: Proceedings of the European conference on computer vision pattern recognition, pp. 220–226 (2005)

Churchill, D., & Vardy, A. (2013). An orientation invariant visual homing algorithm. *Journal of Intelligent and Robotic Systems*, *71*(1), 3–29.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Gao, Y., Ma, J., & Yuille, A. L. (2017). Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples. *IEEE Transactions on Image Processing*, *26*(5), 2545–2560.

Guo, X., & Cao, X. (2012). Good match exploration using triangle constraint. *Pattern Recognition Letters*, *33*(7), 872–881.

Horaud, R., Forbes, F., Yguel, M., Dewaele, G., & Zhang, J. (2011). Rigid and articulated point registration with expectation conditional maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(3), 587–602.

Hu, Y. T., Lin, Y. Y., Chen, H. Y., Hsu, K. J., & Chen, B. Y. (2015). Matching images with multiple descriptors: An unsupervised approach for locally adaptive descriptor selection. *IEEE Transactions on Image Processing*, *24*(12), 5995–6010.

Huber, P. J. (1981). *Robust statistics*. New York: John Wiley & Sons.

Jian, B., & Vemuri, B. C. (2011). Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(8), 1633–1645.

Jiang, J., Chen, C., Ma, J., Wang, Z., Wang, Z., & Hu, R. (2017). Srlsp: A face image super-resolution algorithm using smooth regression with local structure prior. *IEEE Transactions on Multimedia*, *19*(1), 27–40.

Jinda-Apiraksa, A., Vonikakis, V., Winkler, S.: California-ND: An annotated dataset for near-duplicate detection in personal photo collections. In: QoMEX, pp. 142–147 (2013)

Kim, V. G., Lipman, Y., & Funkhouser, T. (2011). Blended intrinsic maps. *ACM Transactions on Graphics*, *30*(4), 79.

Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: Proceedings IEEE international conference on computer vision, pp. 1482–1489 (2005)

Li, X., & Hu, Z. (2010). Rejecting mismatches by correspondence function. *International Journal of Computer Vision*, *89*(1), 1–17.

Lin, W. Y., Cheng, M. M., Lu, J., Yang, H., Do, M. N., Torr, P.: Bilateral functions for global motion modeling. In: Proceedings IEEE International Conference on Computer Vision, pp. 341–356 (2014)

Lin, W. Y., Cheng, M. M., Zheng, S., Lu, J., Crook, N.: Robust non-parametric data fitting for correspondence modeling. In: Proceedings IEEE International Conference on Computer Vision, pp. 2376–2383 (2013)

Lin, W. Y., Wang, F., Cheng, M. M., Yeung, S. K., Torr, P. H., Do, M. N., et al. (2018). CODE: Coherence based decision boundaries for feature correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(1), 34–47.

Lipman, Y., Yagev, S., Poranne, R., Jacobs, D. W., & Basri, R. (2014). Feature matching with bounded distortion. *ACM Transactions on Graphics*, *33*(3), 26.

Liu, H., Yan, S.: Common visual pattern discovery via spatially coherent correspondence. In: IEEE conference on computer vision and pattern recognition, pp. 1609–1616 (2010)

Liu, M., Pradalier, C., & Siegwart, R. (2013). Visual homing from scale with an uncalibrated omnidirectional camera. *IEEE Transactions on Robotics*, *29*(6), 1353–1365.

Liu, Y., Dominicis, L., Wei, B., Chen, L., & Martin, R. (2015). Regularization based iterative point match weighting for accurate rigid transformation estimation. *IEEE Transactions on Visualization and Computer Graphics*, *21*(9), 1058–1071.

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

Ma, J., Jiang, J., Liu, C., & Li, Y. (2017). Feature guided gaussian mixture model with semi-supervised em and local geometric constraint for retinal image registration. *Information Sciences*, *417*, 128–142.

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

Ma, J., Zhao, J., Guo, H., Jiang, J., Zhou, H., Gao, Y.: Locality preserving matching. In: Proceedings of the international joint conference on artificial intelligence, pp. 4492–4498 (2017)

Ma, J., Zhao, J., Jiang, J., Zhou, H.: Non-rigid point set registration with robust transformation estimation under manifold regularization. In: Proceedings of AAAI conference artificial intelligence, pp. 4218–4224 (2017)

Ma, J., Zhao, J., Jiang, J., Zhou, H., Zhou, Y., Wang, Z., Guo, X.: Visual homing via guided locality preserving matching. In: Proceedings of IEEE international conference on robotics and automation, pp. 7254–7261 (2018)

Ma, J., Zhao, J., Ma, Y., & Tian, J. (2015). Non-rigid visible and infrared face registration via regularized gaussian fields criterion. *Pattern Recognition*, *48*(3), 772–784.

Ma, J., Zhao, J., Tian, J., Tu, Z., Yuille, A.: Robust estimation of nonrigid transformation for point set registration. In: Proceedings of IEEE conference computer vision pattern recognition, pp. 2147–2154 (2013)

Ma, J., Zhao, J., Tian, J., Yuille, A. L., & Tu, Z. (2014). Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, *23*(4), 1706–1721.

Ma, J., Zhao, J., & Yuille, A. L. (2016). Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on Image Processing*, *25*(1), 53–64.

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 Transactions on Geoscience and Remote Sensing*, *53*(12), 6469–6481.

Maier, J., Humenberger, M., Murschitz, M., Zendel, O., Vincze, M.: Guided matching based on statistical optical flow for fast and robust correspondence analysis. In: Proceedings of European conference on computer vision, pp. 101–117 (2016)

Micchelli, C. A., & Pontil, M. (2005). On learning vector-valued functions. *Neural Computation*, *17*(1), 177–204.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., et al. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, *65*(1), 43–72.

Möller, R., & Vardy, A. (2006). Local visual homing by matched-filter descent in image distances. *Biological Cybernetics*, *95*(5), 413–430.

Myronenko, A., & Song, X. (2010). Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(12), 2262–2275.

Papadimitriou, C. H., & Steiglitz, K. (1982). *Combinatorial optimization: Algorithms and complexity*. North Chelmsford: Courier Corporation.

Pele, O., Werman, M.: A linear time histogram metric for improved SIFT matching. In: Proceedings of European conference on computer vision, pp. 495–508 (2008)

Rusu, R. B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3d registration. In: Proc. IEEE International conference on robotics and automation, pp. 3212–3217 (2009)

Schroeter, D., & Newman, P. (2008). On the robustness of visual homing under landmark uncertainty. *Intelligent Autonomous Systems*, *10*, 278–287.

Tola, E., Lepetit, V., & Fua, P. (2010). DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(5), 815–830.

Torr, P. H. S., & Zisserman, A. (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, *78*(1), 138–156.

Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: Proceedings of the European conference on computer vision, pp. 596–609 (2008)

Vedaldi, A., Fulkerson, B.: VLFeat - An open and portable library of computer vision algorithms. In: Proceedings of the ACM international conference on multimedia, pp. 1469–1472 (2010)

Wang, C., Wang, L., Liu, L.: Progressive mode-seeking on graphs for sparse feature matching. In: Proceedings of the 10th European conference on computer vision, pp. 788–802 (2014)

Wang, G., Wang, Z., Chen, Y., Liu, X., Ren, Y., & Peng, L. (2016). Learning coherent vector fields for robust point matching under manifold regularization. *Neurocomputing*, *216*, 393–401.

Wang, G., Wang, Z., Chen, Y., Zhou, Q., Zhao, W.: Context-aware gaussian fields for non-rigid point set registration. In: Proceedings of the IEEE conference on computer vision pattern recognition, pp. 5811–5819 (2016)

Wang, G., Wang, Z., Chen, Y., Zhou, Q., & Zhao, W. (2016). Removing mismatches for retinal image registration via multi-attribute-driven regularized mixture model. *Information Sciences*, *372*, 492–504.

Yang, K., Pan, A., Yang, Y., Zhang, S., Ong, S. H., & Tang, H. (2017). Remote sensing image registration using multiple image features. *Remote Sensing*, *9*(6), 581.

Yang, Y., Ong, S. H., & Foong, K. W. C. (2015). A robust global and local mixture distance based non-rigid point set registration. *Pattern Recognition*, *48*(1), 156–173.

Zhao, J., Ma, J.: Visual homing by robust interpolation for sparse motion flow. In: Proc. IEEE/RSJ International conference on intelligent robots and systems, pp. 1282–1288 (2017)

Zheng, Y., & Doermann, D. (2006). Robust point matching for non-rigid shapes by preserving local neighborhood structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(4), 643–649.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.