Locality Preserving Matching

Jiayi Ma*, Ji Zhao*, Hanqi Guo*, Junjun Jiang†, Huabing Zhou‡, and **Yuan Gao§**
*Electronic Information School, Wuhan University, Wuhan 430072, China
†School of Computer Science, China University of Geosciences, Wuhan 430074, China
‡School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430073, China
§Tencent AI Laboratory, Shenzhen 518057, China
{jyma2010, zhaoji84, guohanqi.alpha, jjjiang1986, zhouhuabing, ethan.y.gao}@gmail.com

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. More specifically, our method can accomplish the mismatch removal from thousands of putative correspondences in only a few milliseconds. Experiments on various real image pairs for general feature matching, as well as for visual homing and image retrieval demonstrate the generality of our method for handling different types of image deformations, and it is more than two orders of magnitude faster than state-of-the-art methods in the same range of or better accuracy.

1 Introduction

This study focuses on the problem of establishing reliable point correspondences between two images of the same scene. Many of the computer vision tasks such as 3D reconstruction, content-based image retrieval, visual homing, object detection and recognition start by assuming that the point correspondences have been successfully recovered [Ma *et al.*, 2013; Bai *et al.*, 2017]. 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 of all the possible matches huge. Even without considering the 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 address this issue, a popular strategy is to construct a group of putative correspondences by imposing a similarity constraint to reduce the set of possible matches. It requires that points can only match points with similar descriptors. Thus the matching task boils down to determine the correctness of each match in the putative set. This paper intends to conquer the 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 a challenging task to customize a practical algorithm when dealing with many real-world problems. 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 frequently required in many computer vision tasks such as deformable object recognition where the transformation models are often unknown in advance. Thirdly, the high computational load, especially of complex non-rigid transformation models, limits its applicability in real-time tasks.

To address the above three challenges, in this paper we propose 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 change or non-rigid deformation, but the spatial neighborhood relationship among feature points representing the topological structures of an image scene are generally well preserved due to physical constraints. Based on this fact, we introduce a mathematical model that constrains the unknown inlier correspondences to have similar local neighborhood structures. This formulation is general, and it can handle both rigid and non-rigid deformations related between two images. 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. Experiments on various image data demonstrate that the proposed method can produce more accurate matching results with much less computation time (e.g., more than two orders of magnitude faster) compared with other state-of-the-art methods.

Our contribution in this paper is two-fold. On the one hand, we propose a simple yet effective approach for robust fea-

ture matching. Unlike most existing methods which 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. On the other hand, we derive a closed-form solution with linearithmic complexity which can solve a typical matching problem with over $1,000$ putative correspondences in only a few milliseconds, therefore, it is beneficial for many real-time applications and can quickly provide a good initialization for more complicated problem-specific matching algorithms. We validate this solution on real-word tasks such as image retrieval and visual homing, and obtain better results than other state-of-the-arts in terms of both accuracy and efficiency.

2 Related Work

Numerous mismatch removal methods have been proposed 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.* [Liu *et al.*, 2015] proposed a regression method based on adaptive boosting learning for 3D rigid matching. Maier *et al.* [Maier *et al.*, 2016] introduced a guided matching scheme based on statistical optical flow. The most popular resampling method is 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; they also tend to severely degrade if the outlier ratio becomes large [Li and Hu, 2010]. Several non-parametric interpolation methods [Ma *et al.*, 2014; Li and Hu, 2010; Wang *et al.*, 2017] have recently been introduced to address these issues. 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 set, which limits their uses in real-time applications such as object tracking, visual odometry, SLAM, etc. 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], and graph shift (GS) [Liu and Yan, 2010]. In addition, Lee *et al.* [Lee *et al.*, 2015] proposed to use local neighborhoods for feature description to alleviate false matches. Graph matching provides considerable flexibility to the object model and delivers robust matching and recognition. Nevertheless, it suffers from similar drawbacks of its non-polynomial-hard nature.

In addition to the mismatch removal, some efforts on generating better putative correspondences have also been carried. Guo and Cao [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]. Hu *et al.* [Hu *et al.*, 2015] proposed 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].

3 Method

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]) which can efficiently establish putative correspondence between feature sets. Therefore, in the following, we will focus on the mismatch removal problem and introduce a simple yet efficient strategy by preserving local neighborhood structure.

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.

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 discourage the outliers, and the parameter $\lambda > 0$ controls the tradeoff between these two terms. Ideally, the optimal solution should achieve zero penalty, i.e., the first term of C should be zero.

However, if the image pair undergoes a relatively complex non-rigid transformation, the above distance relationship, in

general, will not hold, especially for matches that are far from each other. Fortunately, the local neighborhood structure among feature points may not change freely due to the physical constraints in the small region around a point, which means that the distribution of neighboring point pairs after transformation should be preserved [Zheng and Doermann, 2006]. Therefore, by preserving only local neighborhood structures, the cost function in Eq. (2) becomes:

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

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 ($K=4$ in default) nearest neighbors for each point in the corresponding feature set under the Euclidean distance.

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

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

and the same as $d(\mathbf{y}_i,\mathbf{y}_j)$. In this case, the cost function in Eq. (3) is converted to:

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

With the distance definition in Eq. (4), the objective function in Eq. (5) is translation, rotation, and scale invariant. The problem of removing outliers and establishing accurate feature correspondences could be solved by searching an optimal solution $\mathbf{p}$ that minimizes the cost function (5).

3.2 Solution

To optimize the objective function (5), we reorganize its form by merging the terms related to p_i and obtain:

$$C(\mathbf{p};S,\lambda)=\sum_{i=1}^{N}p_i(c_i-\lambda)+\lambda N, \quad (6)$$

where

$$c_i=\sum_{j|\mathbf{x}_j\in\mathcal{N}_{\mathbf{x}_i}}d(\mathbf{y}_i,\mathbf{y}_j)+\sum_{j|\mathbf{y}_j\in\mathcal{N}_{\mathbf{y}_i}}d(\mathbf{x}_i,\mathbf{x}_j) \quad (7)$$

is a cost that measures if the i-th correspondence $(\mathbf{x}_i,\mathbf{y}_i)$ meets the geometric constraint of preserving 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. (6) 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. (6) is determined by the following simple criterion:

$$p_i=\begin{cases}1 & c_i\le\lambda\\ 0 & c_i>\lambda\end{cases},\ i=1,\cdots,N. \quad (8)$$

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

$$\mathcal{I}^*=\{i\mid p_i=1,i=1,\cdots,N\}. \quad (9)$$

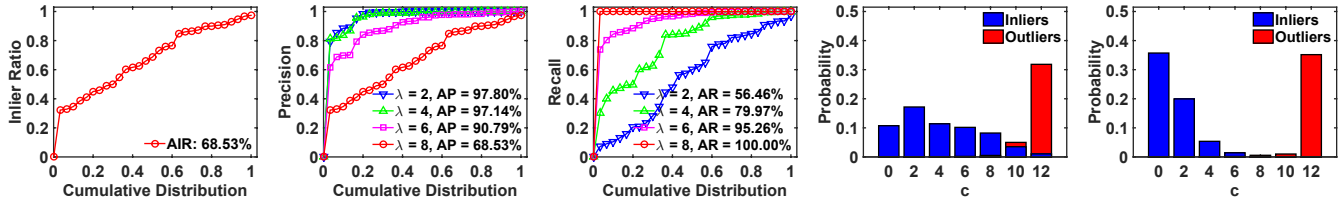
From Eq. (8), we see that parameter λ also plays a role of threshold that judges the match correctness of a putative correspondence. Note that the setting of p_i can be arbitrary when $c_i=\lambda$. Besides, the value of c_i is always an integer, and hence we do not need to tune λ among non-integer values.

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 which also involves outliers. This strategy in general 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 typically inliers which conform to the geometric constraint, and hence its cost c_j will not be large.

To verify how good it works, we collect in total 30 image pairs involving natural images, remote sensing images, medical images, as well as infrared images. The precision and recall are used 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 initial inlier percentages of the SIFT matching on the test data are summarized in Fig. 1a, where in average only 68.53% putative correspondences are inliers. Figures 1b and 1c demonstrate the precision and recall curves with respect to different λ. We see that with a proper value of λ (e.g., 6), our method can preserve about 95.26% of the true correspondences, and the precision can also reach up to 90.79%.

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



(a) (b) (c) (d) (e)

Figure 1: The influence of neighborhood construction to the matching performance. (a)-(c): Initial inlier ratio, precision and recall w.r.t. the cumulative distribution by using the whole feature set to construct neighborhood. A point on the curve with coordinate (x, y) denotes that there are $100 * x$ percents of image pairs which have inlier ratios, precisions or recalls no more than y. (d) Distribution of the cost c_i in Eq. (7) by using the whole feature set to construct neighborhood. (e) Distribution of the cost c_i in Eq. (7) by using $\mathcal{I}_0$ to construct neighborhood.

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. To solve this dilemma, here we seek an approximation $\mathcal{I}_0$ of it. As shown in Fig. 1, 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., $\mathcal{I}_0 = \arg\min_{\mathcal{I}} C(\mathcal{I}; S, \lambda)$ with the neighborhood constructed based on the whole set S.

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

$$\mathcal{I}^* = \arg\min_{\mathcal{I}} C(\mathcal{I}; \mathcal{I}_0, S, \lambda). \tag{10}$$

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 (90.79%, 95.26%) to (96.17%, 97.21%). The distributions of the cost c_i by using the whole feature set and using $\mathcal{I}_0$ to construct neighborhood are reported in Figs. 1d and 1e, respectively. 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 correspondence set generated in the previous iteration for neighborhood construction until convergence, and the average precision-recall pair is then further increased to (96.33%, 98.06%). Note that such progressive strategy can only slightly improve the matching performance, which means that $\mathcal{I}_0$ is good enough to approximate the true inlier set for neighborhood construction. Therefore, we just use Eq. (10) to determine the optimal inlier set for simplicity.

Parameter settings. There are two parameters in our method: K and λ. The former determines the number of nearest neighbors for neighborhood construction, while the latter controls the threshold for judging the correctness of a putative correspondence. Clearly, large value of K or small value of λ will increase the precision and simultaneously decrease the recall, and vice versa. In our evaluate, we set the default values as $K = 4$, and $\lambda = 6$.

Since our matching strategy is to preserve local neighborhood structures, we name our method locality preserving matching (LPM). We summarize our LPM in Alg. 1.

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 Alg. 1 is about $O((K + N) \log N)$. According to Eq. (7), calculating the cost $\{c_i\}_{i=1}^N$ in Lines 2 and 5 only involves some addition operation, and its time complexity is $O(KN)$. Moreover, determining $\mathbf{p}$ and $\mathcal{I}$ using Eqs. (8) and (9) in Lines 3 and 6 cost $O(N)$ complexity. Therefore, the total time complexity of our LPM is about $O(KN + (K + N) \log N)$. The space complexity of our LPM is $O(KN)$ due to the memory requirements for storing the neighborhoods $\mathcal{N}_{\mathbf{x}}$ and $\mathcal{N}_{\mathbf{y}}$. Generally, $K \ll N$, 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.

Algorithm 1: The LPM Algorithm

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

1 Construct neighborhood $\{\mathcal{N}_{\mathbf{x}_i}, \mathcal{N}_{\mathbf{y}_i}\}_{i=1}^N$ based on S;
2 Calculate cost $\{c_i\}_{i=1}^N$ using Eq. (7);
3 Determine $\mathcal{I}_0$ using Eqs. (8) and (9);
4 Construct neighborhood $\{\mathcal{N}_{\mathbf{x}_i}, \mathcal{N}_{\mathbf{y}_i}\}_{i=1}^N$ based on $\mathcal{I}_0$;
5 Calculate cost $\{c_i\}_{i=1}^N$ using Eq. (7);
6 Determine $\mathcal{I}^*$ using Eqs. (8), (9) and (10).

4 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 two real-world tasks such as visual homing and 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 realized without special optimization such as parallel computing or streaming SIMD extensions (SSE).

4.1 Results on Feature Matching

In this section, we focus on establishing feature correspondences for real images. To this end, we first test the perfor-

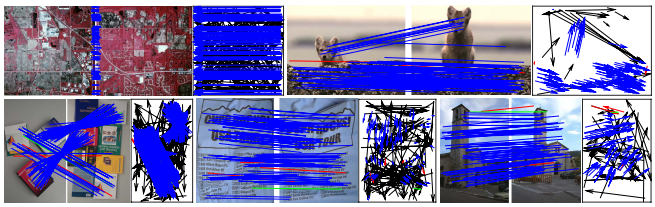


Figure 2: Feature matching results of our LPM on *Land* (top left), *Fox* (top right), *Book* (bottom left), *T-shirt* (bottom middle) and *Church* (bottom right). The ratio of outliers in the 5 image pairs are 40.81%, 85.93%, 76.14%, 43.81%, and 57.26%. The head and tail of each arrow in the motion 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.

Table 1: Results of RANSAC, ICF, GS, MR-RPM, and our LPM on the five image pairs in Fig. 2. For each result in the bracket, the left is the precision and the right is the recall.

	RANSAC	ICF	GS	MR-RPM	LPM
Land	(100.0, 100.0)	(98.23, 100.0)	(100.0, 90.09)	(97.37, 100.0)	(100.0, 100.0)
Fox	(100.0, 87.93)	(85.93, 100.0)	(100.0, 89.66)	(100.0, 89.66)	(98.31, 100.0)
Book	(100.0, 44.19)	(82.62, 91.20)	(100.0, 82.22)	(99.79, 82.57)	(98.23, 97.89)
T-shirt	(97.44, 38.34)	(43.81, 100.0)	(91.49, 86.87)	(98.99, 98.99)	(96.07, 98.99)
Church	(94.52, 100.0)	(91.67, 63.77)	(91.78, 97.10)	(98.33, 85.51)	(95.89, 98.59)

mance of our method on five representative image pairs undergoing different types of image transformations, as shown in Fig. 2. 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*" pairs undergoes piecewise linear transformation, which is typically arisen in image/vedio retrieval. The "*T-shirt*" pair involves a deformable object with non-rigid motion, which is typically arisen in medical image registration. The "*Church*" pair is a wide baseline image pair, 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. The ground-truth is established by manual checking, and we made the benchmark before performing experiments to ensure objectivity. 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 these five 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], and MR-RPM [Ma *et al.*, 2017]. The performance is characterized by precision and recall, as shown in Table 1. From the results, we see that for rigid matching such as in the *Land* pair, all methods perform quite well. 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 *Church* pairs. GS usually has high precision and low recall due to it cannot automatically estimate the factor for affinity matrix and it is not affine-

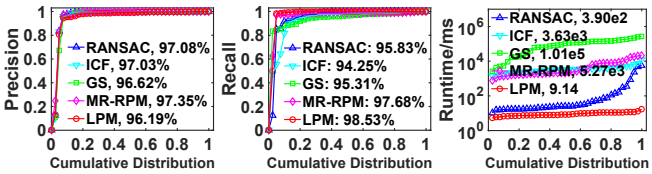


Figure 3: Precisions (left), recalls (middle) and Runtimes (right) of RANSAC, ICF, GS, MR-RPM and LPM on an image dataset.

invariant. In comparison, our LPM does not suffer from all these problems, which demonstrates its generality and its ability to handle various matching problems.

To test the computational efficiency of our LPM, we next conduct experiments on a publicly available feature matching dataset [Mikolajczyk *et al.*, 2005], which contains 40 image pairs either of planar scenes or captured by camera in a fixed position during acquisition. Therefore, these images always obey homography. The ground truth homographies are supplied by the dataset. The average number of putative SIFT correspondences is about 1347.2 on the dataset. The precision, recall and runtime statistics of the five algorithms are reported in Fig. 3. From the results, we see that our LPM does not have obvious advantage in terms of matching accuracy compared with other methods, especially the MR-RPM. This is because the image transformations here are all linear homography, which is relatively simple and easy to handle. However, our LPM is surprisingly effective, which is more than two orders of magnitude faster than state-of-the-arts. More specifically, our average runtime is merely 9.14 ms, making it ideal for real-time applications.

4.2 Results on Visual Homing

Visual homing is the ability of an mobile robot to navigate to a goal position using visual information. The robustness of visual homing methods is dominated by the feature matching results, where recent visual homing methods typically use some heuristic methods to remove mismatches. To validate the effectiveness of our LPM on this problem, we test several state-of-the-art visual homing methods, where their feature matching scenarios are replaced with our LPM. These comparison methods include homing in scale-space (HiSS), bearing-only visual servoing (BOVS), scale-only visual servoing (SOVS), scale and bearing visual servoing (SBVS), and simplified scale-based visual servoing (SSVS), where the first comes from [Churchill and Vardy, 2013] and the rest four come from [Liu *et al.*, 2013].

We conduct experiments on the *A1originalH* and *CHall1H* which are two scenes from a widely used panoramic image database for visual homing[1]. The two scenes contain 170 and 200 images, respectively, which are omnidirectional and unwrapped images of size 561×81 in an indoor environment, plus ground truth for positions where the images are collected. 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.

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

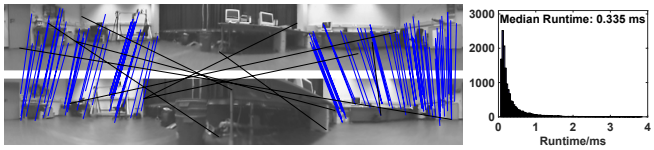


Figure 4: Left: The matching result of a typical image pair from the *A1originalH* scene. Blue and black lines indicate the preserved and removed putative matches. Right: Runtime statistics of LPM for visual homing over $14,365$ trials.

Table 2: Visual homing error statistics of different algorithms on the *A1originalH* dataset. Bold indicates better result (unit: degree).

	TAAE	Min	Max	StdVar
HISS	14.67	8.05	36.40	5.42
HISS+Ours	**14.28**	**7.85**	**36.23**	**5.19**
BOVS	27.41	10.24	70.69	11.91
BOVS+Ours	**14.50**	**4.05**	**44.37**	**9.33**
SOVS	18.75	10.34	37.81	5.81
SOVS+Ours	**16.76**	**9.04**	**33.70**	**4.71**
SBVS	15.90	8.57	34.43	5.86
SBVS+Ours	**13.52**	**7.03**	**31.23**	**5.00**
SSVS	12.59	**6.50**	28.36	4.17
SSVS+Ours	**11.44**	6.53	**25.89**	**3.89**

We schematically show our matching result for a typical image pair on the left of Fig. 4. Clearly, all the inliers and outliers in the putative set are correctly distinguished. To test the efficiency of our LPM for visual homing, we repeat the experiment $14,365$ times on different image pairs from the *A1originalH* scene and report the runtime statistics on the right of Fig. 4, where the median runtime is merely about 0.335 ms. The homing vector errors of all methods on the two scenes are shown in Tables 2 and 3. We can see that our LPM is able to consistently improve the state-of-the-art visual homing methods, due to that our LPM produces more accurate matching results.

4.3 Results on 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 $7,140$ 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. 5. 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, MR-RPM and our LPM for $RN = 10$ are approximately 7.45, 5.18, 7.13, 7.13 and 8.69, respectively. The runtime statistics of our LPM on all the $7,140$ image pairs is provided on the right of Fig. 5, where the median runtime is about 0.579 ms.

Table 3: Visual homing error statistics of different algorithms on the *CHall1H* dataset. Bold indicates better result (unit: degree).

	TAAE	Min	Max	StdVar
HISS	11.69	8.05	18.84	1.81
HISS+Ours	**10.89**	**7.46**	**17.50**	**1.65**
BOVS	45.25	22.56	81.80	11.00
BOVS+Ours	**14.20**	**4.77**	**49.71**	**8.23**
SOVS	28.29	19.04	42.52	4.68
SOVS+Ours	**24.21**	**15.01**	**38.83**	**4.63**
SBVS	23.94	15.50	36.07	**4.04**
SBVS+Ours	**18.04**	**10.98**	**35.56**	4.29
SSVS	15.94	10.79	28.84	3.32
SSVS+Ours	**11.50**	**7.79**	**20.63**	**2.08**

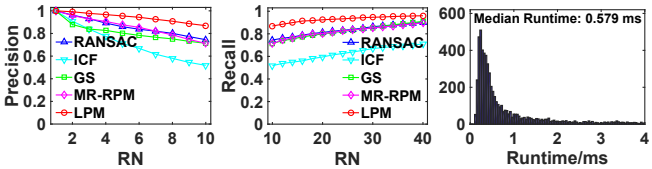


Figure 5: Precisions (left) and recalls (middle) of RANSAC, ICF, GS, MR-RPM and LPM with respect to RN, i.e., the required number of images to be retrieved for a given image. Right: Runtime statistics of LPM over $7,140$ trials.

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, MR-RPM and our LPM are approximately 81.25%, 59.50%, 80.00%, 80.25% and 92.42%, respectively. Our method again evidently showcases the best performance.

5 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 formulate this idea into a mathematic model and derive a closed-form solution with linearithmic time complexity. The qualitative and quantitative results on feature matching as well as other real-world tasks demonstrate 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, it can provide a quick initialization for RANSAC to estimate the epipolar geometry between wide baseline image pairs.

Acknowledgments

We are very grateful to Alan L. Yuille for motivating us to consider the approach discussed here. This paper was supported by the National Natural Science Foundation of China under Grant nos. 61503288, 61501413 and 41501505.

References

[Bai *et al.*, 2010] Xiang Bai, Xingwei Yang, Longin Jan Latecki, Wenyu Liu, and Zhuowen Tu. Learning context-sensitive shape similarity by graph transduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):861–874, 2010.

[Bai *et al.*, 2017] Song Bai, Xiang Bai, and Qi Tian. Scalable person re-identification on supervised smoothed manifold. In *CVPR*, 2017.

[Bentley, 1975] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

[Cho and Lee, 2012] Minsu Cho and Kyoung Mu Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *CVPR*, pages 398–405, 2012.

[Chum and Matas, 2005] Ondrej Chum and Jiri Matas. Matching with PROSAC - progressive sample consensus. In *CVPR*, pages 220–226, 2005.

[Churchill and Vardy, 2013] David Churchill and Andrew Vardy. An orientation invariant visual homing algorithm. *J. Intell. Robot. Syst.*, 71(1):3–29, 2013.

[Fischler and Bolles, 1981] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[Guo and Cao, 2012] Xiaojie Guo and Xiaochun Cao. Good match exploration using triangle constraint. *Pattern Recognit. Lett.*, 33(7):872–881, 2012.

[Hu *et al.*, 2015] Yuan-Ting Hu, Yen-Yu Lin, Hsin-Yi Chen, Kuang-Jui Hsu, and Bing-Yu Chen. Matching images with multiple descriptors: An unsupervised approach for locally adaptive descriptor selection. *IEEE Trans. Image Process.*, 24(12):5995–6010, 2015.

[Huber, 1981] Peter J Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.

[Jinda-Apiraksa *et al.*, 2013] Amornched Jinda-Apiraksa, Vassilios Vonikakis, and Stefan Winkler. California-ND: An annotated dataset for near-duplicate detection in personal photo collections. In *QoMEX*, pages 142–147, 2013.

[Lee *et al.*, 2015] Man Hee Lee, Minsu Cho, and In Kyu Park. Feature description using local neighborhoods. *Pattern Recognit. Lett.*, 68:76–82, 2015.

[Leordeanu and Hebert, 2005] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pages 1482–1489, 2005.

[Li and Hu, 2010] Xiangru Li and Zhanyi Hu. Rejecting mismatches by correspondence function. *Int. J. Comput. Vis.*, 89(1):1–17, 2010.

[Liu and Yan, 2010] Hairong Liu and Shuicheng Yan. Common visual pattern discovery via spatially coherent correspondence. In *CVPR*, pages 1609–1616, 2010.

[Liu *et al.*, 2013] Ming Liu, Cedric Pradalier, and Roland Siegwart. Visual homing from scale with an uncalibrated omnidirectional camera. *IEEE Trans. Robotics*, 29(6):1353–1365, 2013.

[Liu *et al.*, 2015] Yonghuai Liu, Luigi De Dominicis, Baogang Wei, Liang Chen, and Ralph R Martin. Regularization based iterative point match weighting for accurate rigid transformation estimation. *IEEE Trans. Vis. Comput. Graph.*, 21(9):1058–1071, 2015.

[Lowe, 2004] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[Ma *et al.*, 2013] Jiayi Ma, Ji Zhao, Jinwen Tian, Zhuowen Tu, and Alan L Yuille. Robust estimation of nonrigid transformation for point set registration. In *CVPR*, pages 2147–2154, 2013.

[Ma *et al.*, 2014] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE Trans. Image Process.*, 23(4):1706–1721, 2014.

[Ma *et al.*, 2017] Jiayi Ma, Ji Zhao, Junjun Jiang, and Huabing Zhou. Non-rigid point set registration with robust transformation estimation under manifold regularization. In *AAAI*, pages 4218–4224, 2017.

[Maier *et al.*, 2016] Josef Maier, Martin Humenberger, Markus Murschitz, Oliver Zendel, and Markus Vincze. Guided matching based on statistical optical flow for fast and robust correspondence analysis. In *ECCV*, pages 101–117, 2016.

[Mikolajczyk *et al.*, 2005] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vis.*, 65(1):43–72, 2005.

[Torr and Zisserman, 2000] Philip HS Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Comput. Vis. Image Understand.*, 78(1):138–156, 2000.

[Torresani *et al.*, 2008] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, pages 596–609, 2008.

[Vedaldi and Fulkerson, 2010] Andrea Vedaldi and Brian Fulkerson. VLFeat - An open and portable library of computer vision algorithms. In *MM*, pages 1469–1472, 2010.

[Wang *et al.*, 2014] Chao Wang, Lei Wang, and Lingqiao Liu. Progressive mode-seeking on graphs for sparse feature matching. In *ECCV*, pages 788–802, 2014.

[Wang *et al.*, 2017] Gang Wang, Qiangqiang Zhou, and Yufei Chen. Robust non-rigid point set registration using spatially constrained gaussian fields. *IEEE Trans. Image Process.*, 26(4):1759–1769, 2017.

[Zheng and Doermann, 2006] Yefeng Zheng and David Doermann. Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):643–649, 2006.