LMR: Learning a Two-Class Classifier for Mismatch Removal

Jiayi Ma, Xingyu Jiang, Junjun Jiang, Ji Zhao, and Xiaojie Guo

***Abstract*— Feature matching, which refers to establishing reliable correspondence between two sets of features, is a critical prerequisite in a wide spectrum of vision-based tasks. Existing attempts typically involve the mismatch removal from a set of putative matches based on estimating the underlying image transformation. However, the transformation could vary with different data. Thus, a pre-defined transformation model is often demanded, which severely limits the applicability. From a novel perspective, this paper casts the mismatch removal into a two-class classification problem, learning a general classifier to determine the correctness of an arbitrary putative match, termed as Learning for Mismatch Removal (LMR). The classifier is trained based on a general match representation associated with each putative match through exploiting the consensus of local neighborhood structures based on a multiple K-nearest neighbors strategy. With only ten training image pairs involving about 8000 putative matches, the learned classifier can generate promising matching results in linearithmic time complexity on arbitrary testing data. The generality and robustness of our approach are verified under several representative supervised learning techniques as well as on different training and testing data. Extensive experiments on feature matching, visual homing, and near-duplicate image retrieval are conducted to reveal the superiority of our LMR over the state-of-the-art competitors.**

***Index Terms*— Feature matching, supervised learning, classifier, outlier, mismatch removal.**

I. Introduction

As a fundamental problem in vision, establishing reliable feature correspondences between two images of the same or similar scenes has been at the core of many tasks including structure-from-motion, panoramic image mosaics, content-based image retrieval, simultaneous localization and mapping [1]–[5]. The problem is typically solved in a two-step manner, *i.e.* first constructing a set of putative matches and then removing false matches from them. Very often, the putative set includes, besides true positive matches (*inliers*), a number of false positives (*outliers*), due to ambiguities of the local descriptors (particularly if the images suffer from low-quality, occlusion and repetitive patterns). Therefore, it is critical to design a robust approach to remove outliers/mismatches for boosting the reliability of matches.

Existing methods usually address the mismatch removal by imposing a geometrical constraint, which restricts matches satisfying an underlying image transformation. In general, the transformation can vary with respect to different data. Thus, a pre-defined transformation model is often demanded, which can be either parametric (*e.g.*, affine, homography, epipolar geometry [7]) or non-parametric (*e.g.*, nonrigid [8]). However, this demand severely limits the applicability in many vision-based tasks such as deformable object recognition and dynamic scene matching, as the transformation models in these tasks are unknown beforehand. Moreover, the high computational complexity is another demerit of existing methods, especially when the image transformation is a complex nonrigid model, which is a further obstacle in real-time tasks.

Contributions: To address the above issues, this study proposes a learning-based approach and formulates the mismatch removal as a two-class classification problem, termed as *Learning for Mismatch Removal* (LMR). Our method aims to learn a general classifier to determine the correctness of an arbitrary putative match. More specifically, we first construct a representation/feature for each putative match through exploiting the consensus of local neighborhood structures. This match representation is general as it does not rely on any specific image transformation, and hence can be applied to different kinds of image pairs. The classifier is then trained based on a set of match representations with ground truth labels using a supervised learning technique. Experiments on various real data demonstrate that with only 10 training image pairs (as shown in Fig. 4, from which we extract approximately 8,000 SIFT putative matches as training samples), the learned classifier can generate satisfying results with only tens of milliseconds on testing data even of different types of images or with diverse transformation models. In summary, our contributions include: (i) a general yet surprisingly effective learning approach named LMR is proposed to address the mismatch removal problem from a novel classification perspective; (ii) we also apply our LMR to solving real-world tasks such as visual homing and near-duplicate image retrieval in addition to general feature matching, and obtain better results than other state-of-the-art competitors. To the best of our knowledge, the learning-based technique for addressing general mismatch removal has not yet been studied.

Manuscript received July 14, 2018; revised January 2, 2019 and March 8, 2019; accepted March 16, 2019. Date of publication March 20, 2019; date of current version June 20, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61773295 and Grant 61772512, and in part by the CCF-Tencent Open Research Fund. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hitoshi Kiya. *(Corresponding author: Junjun Jiang.)*

J. Ma, X. Jiang, and J. Zhao are with the Electronic Information School, Wuhan University, Wuhan 430072, China (e-mail: jyma2010@gmail.com; jiangx.y@whu.edu.cn; zhaoji84@gmail.com).

J. Jiang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: junjun0595@163.com).

X. Guo is with the School of Computer Software, Tianjin University, Tianjin 300350, China (e-mail: xj.max.guo@gmail.com).

Digital Object Identifier 10.1109/TIP.2019.2906490

The remainder of this paper is organized as follows. Sec. II describes background material and related work. In Sec. III, we present our learning framework for feature matching. We apply our approach to visual homing and near-duplicate image retrieval, and design corresponding methods in Sec. IV. Sec. V illustrates the performance of our method in comparison with other approaches on different kinds of vision-based tasks, followed by concluding remarks in Sec. VI.

II. Related Work

Feature matching has been widely used in many fields including computer vision [9], pattern recognition [10], medical image analysis [11], remote sensing [12], robotics [13], *etc*. Here we briefly review the background material applied as reference for the current study. This material includes three method types: the first type establishes a set of putative correspondence and then removes false matches, the second type solves a correspondence matrix between a couple of point sets, and the third type leverages the deep learning techniques.

A. Two-Step Strategy-Based Methods

A popular strategy for solving the matching problem involves two steps [14]: first computing a set of putative correspondences, 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 [6] 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 [10] proposed a triangle constraint, which can produce better putative correspondences in terms of quantity and accuracy compared with the distance ratio in [6]. Pele and Werman [15] applied the earth mover's distance to replace the Euclidean distance in [6] to measure the similarity between descriptors and improve the matching accuracy. In addition, Hu *et al.* [16] 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 [17], [18].

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 [19]. Liu *et al.* [20] proposed a regression method based on adaptive boosting learning for 3D rigid matching. Recently, Maier *et al.* [21] 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) [7], which has several variants such as maximum likelihood estimation sample consensus [9], progressive sample consensus [22], and spatially consistent random sample consensus [23]. 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 nonrigid; these methods also tend to severely degrade if the outlier proportion becomes large [24]. To mitigate these issues, several non-parametric interpolation methods [8], [14], [24]–[26] have been investigated. 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 [27], dual decomposition [28], mode-seeking [17], [29], composition based affinity optimization [30], [31], and graph shift (GS) [32]. 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.

Additionally to the methods mentioned above, the piecewise smoothness constraints have also been introduced to feature matching, such as coherence based decision boundaries [33]–[35] and grid-based motion statistics (GMS) [36]. The former 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 it is robust to a large number of outliers. While the latter 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. It can provide real-time, ultra-robust feature correspondences, and hence is beneficial to video applications.

B. Correspondence Matrix-Based Methods

Another strategy is to incorporate a correspondence matrix with a parametric or non-parametric geometrical 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) [37]. 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 [38] established a general framework for nonrigid matching called thin plate spline-based robust point matching, which replaces the nearest point strategy of ICP with soft assignments within a continuous optimization framework that involves deterministic annealing. Yang *et al.* [39] further introduced an approach termed as global and local mixture distance with thin plate spline, and has shown promising results. Boughorbel *et al.* [40] brought the Gaussian fields into rigid registration, which was later generalized to the nonrigid setting in [41] and [42]. Point set registration has commonly been solved by probabilistic methods in recent years [43]–[46]. 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 performance very likely degrades, especially when the image pair involving nonrigid deformations [46].

C. Learning-Based Methods

Recently, many deep learning-based approaches have made dramatic progress on a wide range of complex computer vision tasks, such as image classification, object detection and tracking, image segmentation [47]–[49], *etc*. Analogously, it is reasonable and meaningful to learn from raw images directly for certain matching tasks including key-point detection and feature description [50], image patch matching [51], and stereo matching [52]. For the key-point detection and feature description, different from traditional purely engineered features like SIFT, the learning-based methods aim to construct sparse point correspondences from image pairs of the same or similar scenes by leveraging deep learning architectures [50], [53]–[56]. Although these methods have been verified to be superior to those hand-crafted representations [57], [58], there are still a large number of false matches in the generated putative match set [55], and hence an efficient mismatch removal method as the post-processing is still required. Image patch matching aims to extract the latent deep features from image pathes using deep convolutional networks and compute a similarity between the extracted features to establish reliable path correspondences [51], [59]–[61]. It has been applied to wide-baseline stereo [52], object instance recognition and image registration [51], [62]–[64]. Nevertheless, this kind of methods operates on region and aims to establish region correspondences, which is different from point matching, especially when the image pair undergoes a nonrigid deformation. Learning-based stereo matching [52], [65], which aims to extract the depth information of each pixel (*i.e.*, depth map) from the rectified image pair typically obtained from a binocular camera, has occupied the top performance in the common datasets. It has achieved great success in several stereo applications such as automatic driving, robotics and 3D scene reconstruction [13], [66]. However, the success of these methods highly depends on the specific image pair that must be rectified and largely overlapped.

Another application of deep learning to the matching task is learning local and global features from two point sets to find reliable point correspondences. This inspiration has obtained great concern in 3D point cloud registration due to the dense point distribution [67]–[69], which can form obvious context structure that is similar to the image texture and easy to learn under a deep convolutional network. Feature descriptor learned from 3D point clouds can be regarded as a substitution of the existing hand-crafted 3D point feature descriptors such as fast point feature histograms [70] and spin images [71]. However, the learning strategy for 3D point clouds is in general not suitable for sparse 2D feature points due to the lack of obvious context structure. To the best of our knowledge, learning directly from 2D feature points instead of the image pixel information for establishing point correspondences has not been adequately explored. To this end and most recently, Yi *et al.* [72] presented a first attempt to introduce a deep leaning-based technique for mismatch removal from putative matches, termed as learning to find good correspondences (LFGC), which aims to train a multi-layer perceptron from a set of putative sparse matches and the camera intrinsics under a parametric geometrical constraint, to label the testing correspondences as inliers or outliers and output the camera motion simultaneously. However, the method requires ground truth camera intrinsics as input and depends on a specific parametric transformation model, failing to handle general matching problems such as deformable image matching which cannot be characterized by a parametric model. In our experiments, we have demonstrated significant superiority of our LMR over LFGC in addressing general image matching.

III. Method

As aforementioned, the first step is to construct a set of putative matches by considering all possible matches between the given two feature point sets with those having distant descriptors eliminated. Then the problem boils down to removing false matches from the putative set. Fortunately, many well-designed image descriptors (*e.g.*, SIFT) can efficiently establish putative matches. Thus, in the following, we focus on mismatch removal and formulate it as a two-class classification problem under an efficient learning framework.

A. Overview of the Framework

The proposed learning framework for feature matching is presented in Fig. 1. Without loss of generality, suppose the training set contains N image pairs and for each image pair we have extracted a set of putative matches $\mathcal{S}_n = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{L_n}$ by using an off-the-shelf image descriptor, where $\mathbf{x}_i$ and $\mathbf{y}_i$ are the spatial positions of two corresponding feature points,[1] and $L_n (n \in \mathbb{N}_N)$ is the number of putative matches in the n-th image pair. Therefore, we have $L = \sum_{n=1}^{N} L_n$ putative matches in total in the training set, and their ground truth labels (*i.e.*, being inlier or outlier) are available in advance. The goal

[1] Strictly we should use the notations $\mathbf{x}_i^n$ and $\mathbf{y}_i^n$; here we omit the superscripts to keep them uncluttered.

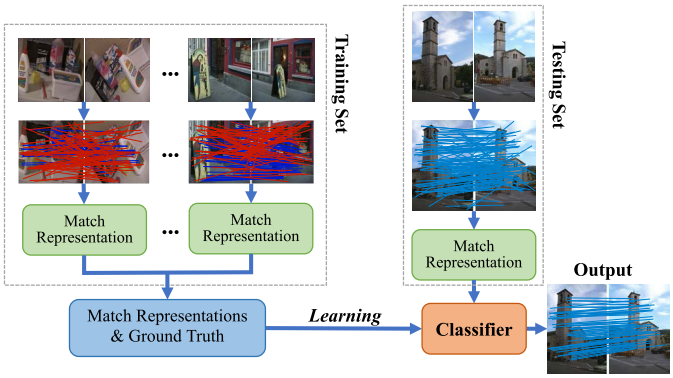


Fig. 1. The proposed learning framework for feature matching. For the putative matches in the training set, blue and red lines indicate ground truth inliers and outliers, respectively.

is to learn a classifier that can distinguish inliers from outliers in a new putative set, *e.g.* extracted from a test image pair.

Our framework involves three major steps: match representation, classifier training and testing. In the match representation step, we aim to describe each putative match by a group of properties, called a representation,[2] which could be constructed based on the geometrical relationship among the putative set that the putative match belongs to. After obtaining all the match representations in the training set together with their ground truth labels, we can then train a classifier using a supervised learning technique. In the testing step, given a new image pair, we first extract a set of putative matches and construct their representations, and then use the learned classifier to identify the outliers.

Compared with the traditional matching methods, the proposed learning framework is more general and efficient. On the one hand, unlike traditional methods (including the existing deep learning-based method [72]) which typically rely on certain special transformation models, the learned classifier in our framework can be used for feature matching on any new image pairs. By using a general match representation, the method works well even when the training and testing data undergo different types of image transformations. On the other hand, our learning framework is also quite efficient. With only 10 training image pairs involving about 8,000 putative matches and an off-the-shelf supervised learning technique, the method is able to generate promising matching performance on arbitrary testing data. While in the testing process, only tens of milliseconds are needed to identify the mismatches, which is about one order of magnitude faster than many commonly used methods.

B. Match Representation

Constructing a proper match representation is the key to the success of our learning framework. Generally, its composed properties should not only be able to distinguish the inliers and outliers efficiently, but should also be general enough to adapt to different types of image transformations. In addition, as the input in this step is just a putative match set composed of only spatial positions of feature correspondences, the match representation then cannot be constructed based on the original image content.[3] Instead, it aims to exploit the geometrical relationship among the putative set.

For an image pair of the same scene or object, the absolute distance between two feature points may change significantly under viewpoint changes (*e.g.*, stereo disparities) or nonrigid deformations (*e.g.*, dynamic scenes); however, the spatial neighborhood relationship among feature points representing the local topological structures of an image scene would be generally well preserved [4], [73]. Specifically, a variety of approaches exploiting neighborhood consistency have been investigated to improve feature matching, image retrieval, and spatial verification [74]–[78]. We take the nonrigid human face as an example. Due to the physical constraint of bones and muscles, expression and viewpoint changes cannot lead to topological structure changes of the face, such as the relative positions of eyes, nose, mouth, chin, *etc.* Based on this observation, in the following, we design two properties for constructing a general match representation.

1) Consensus of Neighborhood Elements: For a putative match $(\mathbf{x}_i, \mathbf{y}_i)$ from $\mathcal{S}_n$, if it is an inlier, then the distributions of local neighborhood elements of the two corresponding feature points should be similar. In contrast, for an outlier, the corresponding neighborhood distributions will be quite different. We call this property the consensus of neighborhood elements. To capture such a property mathematically, we use multi-neighborhood representation and define a set of neighborhoods with different sizes $\{K_j\}_{j=1}^{M}$, *e.g.* $\{\mathcal{N}_{\mathbf{x}_i}^{K_j}\}_{j=1}^{M}$ and $\{\mathcal{N}_{\mathbf{y}_i}^{K_j}\}_{j=1}^{M}$, where $\mathcal{N}_{\mathbf{x}_i}^{K_j}$ denotes the neighborhood of point $\mathbf{x}_i$ composed of its K_j nearest neighbors in the point set $\{\mathbf{x}_i\}_{i=1}^{L_n}$ under the Euclidean distance. We call this strategy multiple K-nearest neighbors (multi-KNN). Then the consensus of neighborhood elements between $\mathcal{N}_{\mathbf{x}_i}^{K_j}$ and $\mathcal{N}_{\mathbf{y}_i}^{K_j}$ (*e.g.*, the degree of local neighborhood preserving between $\mathbf{x}_i$ and $\mathbf{y}_i$) can be characterized by

$$r_i^{K_j} = \mathcal{I}_i^{K_j} / K_j, \tag{1}$$

where $\mathcal{I}_i^{K_j} \leq K_j$ is the number of common elements in the two neighborhoods $\mathcal{N}_{\mathbf{x}_i}^{K_j}$ and $\mathcal{N}_{\mathbf{y}_i}^{K_j}$, and $r_i^{K_j} \in [0, 1]$ is the ratio of common feature points in the corresponding neighborhoods. Clearly, an inlier will lead to a large value of $r_i^{K_j}$ and vice versa. Note that the common elements cannot be determined without ground truth. Nevertheless, a good approximation is to apply the number of putative matches between $\mathcal{N}_{\mathbf{x}_i}^{K_j}$ and $\mathcal{N}_{\mathbf{y}_i}^{K_j}$ contained in $\mathcal{S}_n$ as a replacement.[4]

[2]It is typically termed as *feature* in the machine learning and computer vision communities. However, to avoid confusion with the term *feature* matching, here we use the term *representation* instead.

[3]It is the major difference between our approach and the existing learning-based matching methods.

[4]This is due to that if the putative match $(\mathbf{x}_i, \mathbf{y}_i)$ is an inlier, then the putative matches simultaneously appearing in the local neighborhoods $\mathcal{N}_{\mathbf{x}_i}^{K_j}$ and $\mathcal{N}_{\mathbf{y}_i}^{K_j}$ will probably be inliers. On the contrary, if $(\mathbf{x}_i, \mathbf{y}_i)$ is an outlier, then there will be probably no putative match simultaneously appearing in $\mathcal{N}_{\mathbf{x}_i}^{K_j}$ and $\mathcal{N}_{\mathbf{y}_i}^{K_j}$.

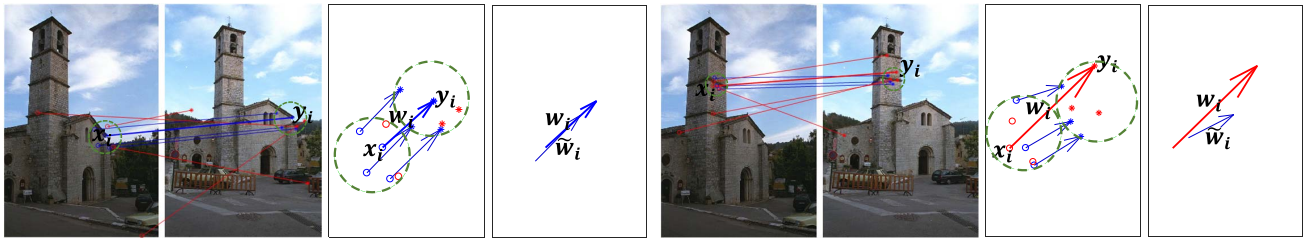


Fig. 2. Schematic illustration of the consensus of neighborhood topology. The putative match $(\mathbf{x}_i, \mathbf{y}_j)$ is an inlier in the left group and an outlier in the right group. For each group, the first plot shows a putative match $(\mathbf{x}_i, \mathbf{y}_j)$ together with its neighborhood elements, their corresponding displacement vectors are shown in the second plot with $\mathbf{w}_i$ corresponding to $(\mathbf{x}_i, \mathbf{y}_j)$, and $\widetilde{\mathbf{w}}_i$ in the third plot is the average of the three neighboring vectors.

Considering the multi-KNN, the consensus of neighborhood elements for the putative match $(\mathbf{x}_i, \mathbf{y}_i)$ is then defined as

$$R_i = \left(r_i^{K_1}, r_i^{K_2}, \cdots, r_i^{K_M}\right). \tag{2}$$

2) Consensus of Neighborhood Topology: The consensus of neighborhood elements described above essentially aims to preserve the intersection of neighbors, which ignores their topological structure. To address this issue, here we design another property to further exploit the consensus of neighborhood topology.

For a putative match $(\mathbf{x}_i, \mathbf{y}_i)$ from $\mathcal{S}_n$, as shown in Fig. 2, we first extract its $\mathcal{I}_i^{K_j}$ neighboring putative matches located in $\mathcal{N}_{\mathbf{x}_i}^{K_j}$ and $\mathcal{N}_{\mathbf{y}_i}^{K_j}$, where $K_j = 5$ and $\mathcal{I}_i^{K_j} = 3$. 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.* $\mathbf{w}_i$. Subsequently, we compute the average displacement vector of the $\mathcal{I}_i^{K_j}$ neighboring putative matches, *i.e.* $\widetilde{\mathbf{w}}_i$. The neighborhood topology can then be exploited by comparing the difference between $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_i$. More specifically, the changes of topological structures of the $\mathcal{I}_i^{K_j}$ elements with respect to $\mathbf{x}_i$ and $\mathbf{y}_i$ will lead to significant differences between $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_i$ in both lengths and directions, as demonstrated in the two examples in Fig. 2.

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

- Consensus of ratio of length between $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_i$:

$$s_i^{K_j} = \begin{cases} \exp\left\{-\dfrac{(p_i^{K_j} - 1)^2}{2\sigma_1^2}\right\}, & \mathcal{I}_i^{K_j} \geq 1, \\ 0, & \mathcal{I}_i^{K_j} = 0, \end{cases} \tag{3}$$

  where $p_i^{K_j} = \frac{\max\{|\mathbf{w}_i|, |\widetilde{\mathbf{w}}_i|\}}{\min\{|\mathbf{w}_i|, |\widetilde{\mathbf{w}}_i|\}} \geq 1$ characterizes the ratio of length between $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_i$.
- Consensus of angle between $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_i$:

$$t_i^{K_j} = \begin{cases} \exp\left\{-\dfrac{(\theta_i^{K_j})^2}{2\sigma_2^2}\right\}, & \mathcal{I}_i^{K_j} \geq 1, \\ 0, & \mathcal{I}_i^{K_j} = 0, \end{cases} \tag{4}$$

  where $\theta_i^{K_j} = \langle \mathbf{w}_i, \widetilde{\mathbf{w}}_i \rangle \in [0, \pi]$ characterizes the angle between $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_i$.

Clearly, $s_i^{K_j}, t_i^{K_j} \in [0, 1]$, and an inlier will lead to large values of $s_i^{K_j}$ and $t_i^{K_j}$ and vice versa. Note that in Eqs. (3) and (4), we choose the Gaussian penalty with σ_1 and σ_2 being the corresponding range parameters. In our experiments, we empirically fix $\sigma_1 = 0.4$ and $\sigma_2 = 0.8$. The use of Gaussian penalty can normalize the value of property into the range of $[0, 1]$. Besides, the penalty curve has a "long tail" which can prevent over-penalization on the outliers. Considering the multi-KNN, the consensus of neighborhood topology for the putative match $(\mathbf{x}_i, \mathbf{y}_i)$ is then defined as

$$ST_i = \left(s_i^{K_1}, t_i^{K_1}, s_i^{K_2}, t_i^{K_2}, \cdots, s_i^{K_M}, t_i^{K_M}\right). \tag{5}$$

3) Match Representation Construction and Analysis: Based on the two properties defined in Eqs. (2) and (5), the final match representation for a putative match $(\mathbf{x}_i, \mathbf{y}_i)$ from $\mathcal{S}_n$ is a $3M$ dimensional vector defined as

$$F_i = \left(r_i^{K_1}, s_i^{K_1}, t_i^{K_1}, r_i^{K_2}, s_i^{K_2}, t_i^{K_2}, \cdots, r_i^{K_M}, s_i^{K_M}, t_i^{K_M}\right). \tag{6}$$

Now we revisit the whole process of match representation construction in the training phase. Given N training image pairs and for the n-th image pair we have extracted L_n putative matches $\mathcal{S}_n$; for each putative match $(\mathbf{x}_i, \mathbf{y}_i)$ from $\mathcal{S}_n$ we construct its match representation F_i based on $\mathcal{S}_n$ and then L_n match representations are obtained for the n-th image pair. We repeat this procedure on the N training image pairs, and obtain L match representations for all the L putative matches.

To verify the discriminability of the proposed match representation, here we randomly choose 10 image pairs including wide baseline, remote sensing and medical images, and compute their match representations for visualization, as shown in Fig. 3. Since the dimension of the match representation $3M \gg 3$, we only choose three entries $(r_i^{K_j}, s_i^{K_j}, t_i^{K_j})$ for visualization, as shown in the left two plots. In addition, we also apply principal component analysis (PCA) to the data and extract the first two principal components to further show the discriminability. From the results in the right plot, we see that the distributions of inliers and outliers are quite different, which are almost linearly separable.

C. Classifier Training and Testing

After obtaining the L match representations $\{F_i\}$ for all the L putative matches on the training set, *i.e.* $\mathbf{F} \in \mathbb{R}^{L \times 3M}$, we can then train a classifier $\mathbf{f}$, *i.e.* $\mathbf{L} = \mathbf{f}(\mathbf{F})$, based on a supervised

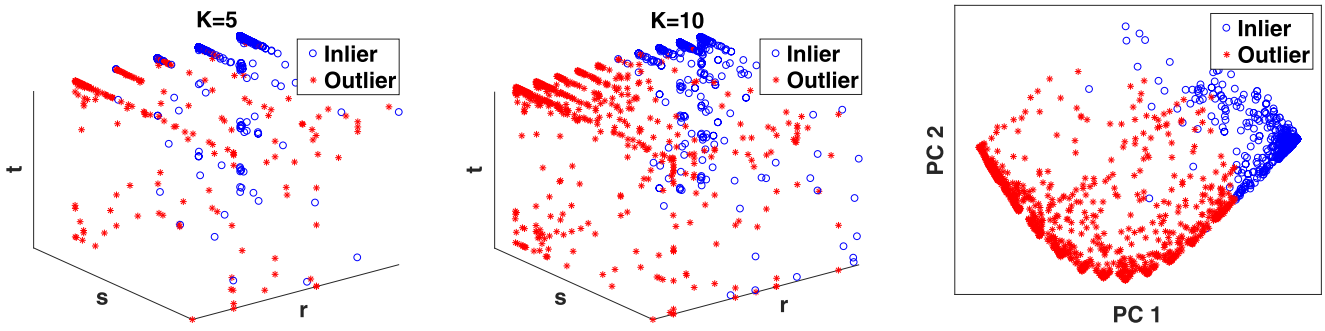


Fig. 3. Schematic illustration of the discriminability of the proposed match representation. In the left two plots, we only choose the three entries $(r_i^{K_j}, s_i^{K_j}, t_i^{K_j})$ corresponding to $K_j = 5$ and $K_j = 10$ from the $3M$ dimensional match representation for visualization. In the right plot, we apply PCA on the whole match representation set and show the first two principal components.

Algorithm 1 The LMR Algorithm

Input: Training data $\mathcal{S}$ with N image pairs, testing data $\mathcal{T}$
Output: Classifier $\mathbf{f}$, matching results on $\mathcal{T}$
1 *Training phase:*
2 Extract putative matches on $\mathcal{S}$;
3 Construct match representations $\mathbf{F}$ for putative matches;
4 Construct ground truth labels $\mathbf{L}$ for putative matches;
5 Train classifier $\mathbf{f}$ using a supervised learning technique;
6 *Testing phase:*
7 Extract putative matches on $\mathcal{T}$;
8 Construct match representations for putative matches;
9 Compute confidence for each putative match using $\mathbf{f}$;
10 Obtain final matching results from the confidences.

learning technique, where $\mathbf{L} \in \mathbb{R}^{L\times 2}$ is the ground truth labels. Specifically, the ground truth label is represented as $(1, 0)$ for an inlier and $(0, 1)$ for an outlier.

There are several widely used supervised learning techniques for classifier training, such as Bayes classifiers based on probability statistics, neural network based on multilayer perceptron, decision tree or random forest (RF) based on logical decision rules, and support vector machine (SVM) based on kernel trick [79]. In this paper, we choose the back-propagation neural network (BPNN) as an instance[5] [80]. In general, the neural network contains three kinds of layers, *i.e.* input, hidden and output layers. A number of neurons exist in the hidden layer(s), each of which is a calculation unit. These units are somehow connected and their weights and biases are gradually refined through gradient back propagation so as to minimize a certain loss between the output and ground truth.

During the testing phase, given a new image pair to be matched, we first extract a set of putative matches and construct their match representations. Then we use the trained classifier to generate a 2D output for each match representation separately. The two elements of the output can be respectively seen as the confidences of the putative match being an inlier and an outlier, and the final decision goes to the category with the larger value. The whole procedure of the proposed LMR algorithm is outlined in Alg. 1.

[5]We have also considered other learning techniques in our experiments.

D. Computational Complexity

Given a set of N putative matches extracted from a test image pair, the major computational cost focuses on the step of match representation construction.[6] It involves searching the K_j nearest neighbors for each feature point, which has about $O((K_j + N)\log N)$ complexity by using K-D tree [81], and hence the computational complexity of Line 8 in Alg. 1 is about $O((\sum_{j=1}^{M} K_j + MN)\log N)$. The computation of confidence for each putative match does not depend on the scale of putative matches, and hence the computational complexity of Line 9 is about $O(N)$. As $\sum_{j=1}^{M} K_j$ is a constant and $M \ll N$, the total computational complexity of our LMR can be simply written as $O(N \log N)$. That is to say, our LMR has linearithmic complexity which is significant for handling large-scale problems or real-time applications.

E. Implementation Details

To construct match representations, we have defined a set of neighborhoods. Clearly, it would be better if the neighborhood is defined on only inliers, which can preserve the consensus of neighborhood elements and topology better. However, the inlier set is unknown in advance; here we seek an approximation to it. More specifically, we calculate the consensus of neighborhood elements for each putative match by Eq. (1) at $K_j = 10$, and then use only those putative matches with $r > 0.2$ for neighborhood construction.[7] This operation can filter out quite a lot of outliers without sacrificing too many inliers. In addition, we choose $M = 11$ and $K = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15\}$ to construct the neighborhoods $\{\mathcal{N}_{\mathbf{x}_i}^{K_j}\}_{j=1}^{M}$ and $\{\mathcal{N}_{\mathbf{y}_i}^{K_j}\}_{j=1}^{M}$.

For the BPNN, we design a simple 3-hidden-layer network $[48, 10, 2]$ with each element denoting the number of neurons in the corresponding hidden layer. In the input and hidden layers we use the hyperbolic tangent sigmoid as activation function and log-sigmoid transfer function in the output layer. In the training phase, we set the stop condition with the max iteration at 5,000 or the min gradient value at

[6]Without ambiguity, in this paper we use the same symbol N with different meanings in different contexts.

[7]It means that the neighborhood is defined on the putative matches with $r > 0.2$. We do not remove the matches with $r \leq 0.2$, and the outlier removal is still operated on the original putative set.

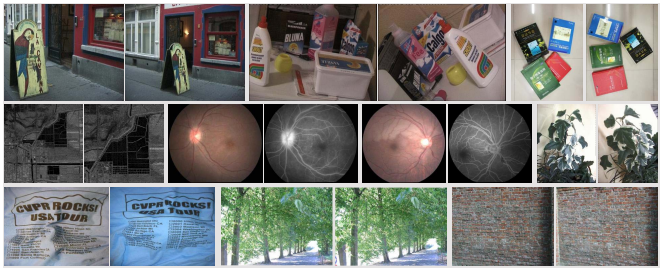


Fig. 4. The 10 image pairs used for training in our LMR, which contains in total 7,659 SIFT putative matches as the training samples with 3,858 positive samples and 3,801 negative samples.

10^{-5} with the target function being the mean square error, and the scaled conjugate gradient back-propagation is used as the gradient descent mode for efficient training. We have used two other supervised learning techniques such as RF and SVM for comparison in our experiments. For the former we set the number of trees as 20 to train a forest for classification, while for the latter we use the linear kernel and sequential minimal optimization to train our SVM with the max iteration at 50,000. In addition, all these training and testing procedures are implemented with MATLAB toolbox. Meanwhile, we found that the learned classifier in general will have better generalization ability if the image transformations involved in the training set are more complex. In our experiments, we use 10 wide baseline and deformable image pairs for training, and test the learned classifier on all the other data even of different types of images or with different transformation models.[8] The 10 pairs (as shown in Fig. 4) contain 7,659 putative matches in total (*i.e.*, $L = 7,659$) with 3,858 inliers and 3,801 outliers.

IV. APPLICATIONS

This section describes how to apply our LMR to different vision-based tasks, including visual homing and near-duplicate image retrieval, whose performance is dominated by the feature matching quality.

A. 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 [3]. We use LMR for robust feature matching and estimate the dense motion flow $\mathcal{F}$ accordingly. And then we derive the focus-of-contraction (FOC) and focus-of-expansion (FOE) based on it to determine homing directions.

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.* $\mathbf{x} = (\mathbf{x}^h, \mathbf{x}^v)^{\mathrm{T}}$, the Euclidean distance then can be modified as the following cylinder distance:

$$\mathrm{CylDist}^2(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathrm{CylDist}^h(\mathbf{x}_i^h, \mathbf{x}_j^h)\right)^2 + \left(\mathrm{CylDist}^v(\mathbf{x}_i^v, \mathbf{x}_j^v)\right)^2, \tag{7}$$

where the horizontal and vertical distances are defined as

$$\begin{aligned}\mathrm{CylDist}^h(\mathbf{x}_i^h, \mathbf{x}_j^h) = \min\big\{&|\mathbf{x}_i^h - \mathbf{x}_j^h|, |\mathbf{x}_i^h - \mathbf{x}_j^h - \mathbf{x}_{\max}^h|, \\ &|\mathbf{x}_i^h - \mathbf{x}_j^h + \mathbf{x}_{\max}^h|\big\},\end{aligned} \tag{8}$$

$$\mathrm{CylDist}^v(\mathbf{x}_i^v, \mathbf{x}_j^v) = |\mathbf{x}_i^v - \mathbf{x}_j^v|, \tag{9}$$

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

To conduct feature matching on panoramic image pairs by using our LMR in Alg. 1, the only required modification is to construct the neighborhoods $\{\mathcal{N}_{\mathbf{x}}, \mathcal{N}_{\mathbf{y}}\}$ in Lines 3 and 8 by using the cylinder distance defined in Eq. (7) 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.

2) Motion Flow Estimation: After obtaining the accurate feature match set, *e.g.* $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, we focus on recovering the dense motion flow $\mathcal{F}$ from the matches. To this end, we first convert the match $(\mathbf{x}_i, \mathbf{y}_i)$ to a motion vector $(\mathbf{u}_i, \mathbf{v}_i)$ according to the cylinder coordinate, *e.g.*,

$$\mathbf{u}_i = \mathbf{x}_i, \tag{10}$$

$$\mathbf{v}_i = (\mathbf{y}_i^h - \mathbf{x}_i^h + \alpha \mathbf{x}_{\max}^h, \mathbf{y}_i^v - \mathbf{x}_i^v), \tag{11}$$

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 $[-\mathbf{x}_{\max}^h/2, \mathbf{x}_{\max}^h/2]$.

To estimate the dense motion flow $\mathcal{F}$, *i.e.* $\mathbf{v}_i = \mathcal{F}(\mathbf{u}_i)$ for a true match $(\mathbf{x}_i, \mathbf{y}_i)$, it is natural to consider the supervised learning technique such as regression. Typically, we have a limited number of matches and the flow $\mathcal{F}$ may be relatively complex due to nonrigid transformation, and hence we cannot expect to obtain satisfactory performance by blindly choosing a function model. A highly-parameterized model will probably overfit the data, and a too simple model may not adequately describe the data. Regularization in this context provides us with one way to strike the appropriate balance in creating the model. The goal of regularization is to solve the empirical error minimization problem by controlling the complexity of the function space, for example, by introducing a penalty term into the empirical error

$$\mathrm{ERR}(\mathcal{F}) + \mu \mathrm{PEN}(\mathcal{F}), \tag{12}$$

where the first term is the empirical error measuring the fitting degree of the function and the samples, the second term is

[8] In fact, the classifier can perform better if the testing data is closely related to the training data. However, in this paper we use the fixed training data to demonstrate the generality of our approach.

a penalty item which requires the function to be not too complex, and μ is used as a regularization parameter to make a trade-off between the two items. We model the flow $\mathcal{F}$ by restricting it to lie within a specific functional space $\mathcal{H}$, namely a reproducing kernel Hilbert space (RKHS) [82], which is defined by a positive definite matrix-valued kernel Γ. In this paper we choose a diagonal decomposable Gaussian kernel $\Gamma(\mathbf{u}_i, \mathbf{u}_j) = e^{-\beta\|\mathbf{u}_i - \mathbf{u}_j\|^2} \cdot \mathbf{I}$ 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 [82]:

$$\mathcal{E}(\mathcal{F}) = \min \left\{ \sum\nolimits_{i=1}^{N} \|\mathbf{v}_i - \mathcal{F}(\mathbf{u}_i)\|^2 + \mu \|\mathcal{F}\|_{\mathcal{H}}^2 \right\}. \quad (13)$$

According to the representer theorem [82], the optimal solution of the minimization problem in Eq. (13) is given by

$$\mathcal{F}(\mathbf{u}) = \sum\nolimits_{i=1}^{N} \Gamma(\mathbf{u}, \mathbf{u}_i)\mathbf{c}_i, \quad (14)$$

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

$$(\Gamma + \mu\mathbf{I})\mathbf{C} = \mathbf{V}, \quad (15)$$

where $\Gamma \in \mathbb{R}^{N \times N}$ is the so-called Gram matrix with $\Gamma_{ij} = e^{-\beta\|\mathbf{u}_i - \mathbf{u}_j\|^2}$, $\mathbf{C} = (\mathbf{c}_1, \cdots, \mathbf{c}_N)^{\mathrm{T}}$ and $\mathbf{V} = (\mathbf{v}_1, \cdots, \mathbf{v}_N)^{\mathrm{T}}$ are matrices of size $N \times 2$.

Note that there are two parameters needing 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 dense motion flow estimation more robust, the vector field consensus [14] algorithm is preferable. It generalizes the Tikhonov regularization to handle contaminated data under a Bayesian framework, which introduces a latent variable to resist outliers.

3) Estimation of Homing Direction: It has been shown in previous work that the motion flow of a panoramic image pair has two singularities [83], 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, the FOC and FOE are used to determine the homing direction [3], [84]. 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 [84].

Next, we introduce a method that uses the dense motion flow to determine the FOC and FOE [3]. In general, the FOC and FOE should lie on the horizontal line $\mathbf{u}^v = \mathbf{u}^v_{\max}/2$ and are separated by $\mathbf{u}^h_{\max}$, with $\mathbf{u}^h_{\max}$ and $\mathbf{u}^v_{\max}$ 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 $\mathcal{F}(\mathbf{u})$ in Eq. (14), 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 $\mathbf{u}^v = \mathbf{u}^v_{\max}/2$, we define a 1D function

$$g(\mathbf{u}^h) \triangleq \mathcal{F}([\mathbf{u}^h, \mathbf{u}^v_{\max}/2]). \quad (16)$$

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

We use a coarse-to-fine grid search strategy to find the optimal solution of FOC, which is able to achieve arbitrary precision. In the 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:

$$\theta_{\mathrm{homing}} = \theta_{\mathrm{FOC}} = \frac{2\pi \cdot \mathbf{u}^h_{\mathrm{FOC}}}{\mathbf{u}^h_{\max}}. \quad (17)$$

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

B. 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 [85]. 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 characterized by the feature matching result. Therefore, our LMR is desirable to produce reliable performance.

For the near-duplicate image retrieval 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 LMR to remove false matches, the similarity $s(I_i, I_j)$ is then assigned by the number of preserved matches in 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, for example, 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 to the query I_i.

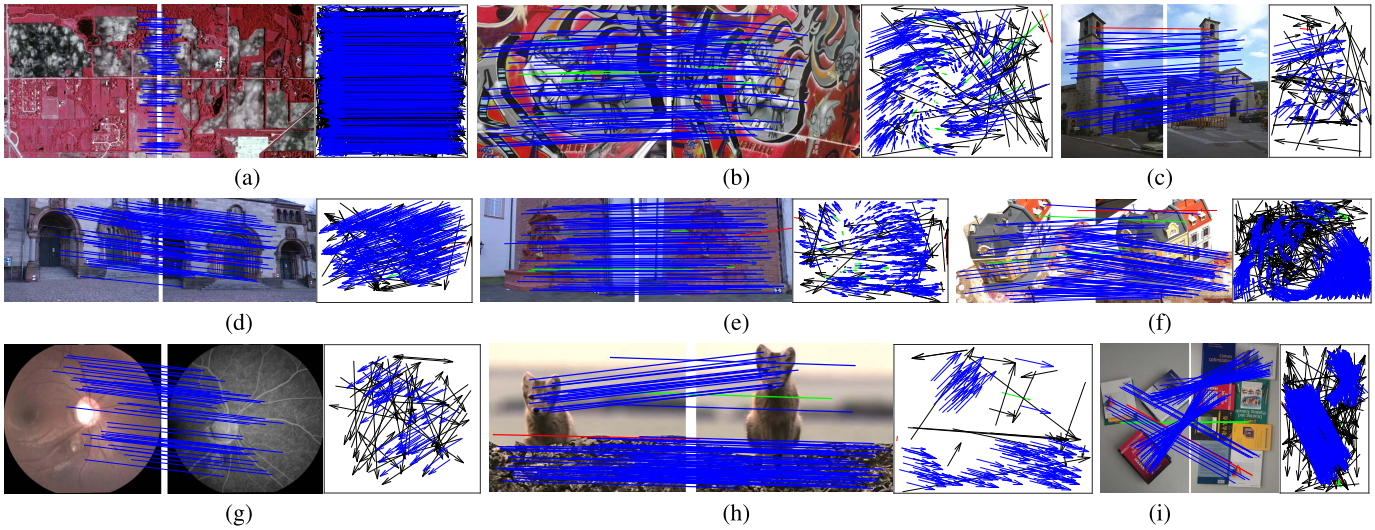


Fig. 5. Feature matching results of our LMR on 9 typical image pairs involving different types of transformations (blue = true positive, black = true negative, green = false negative, red = false positive). For each group of results, the first value is the initial inlier ratio, while the rest two values are the precision and recall after using our learned classifier to remove mismatches, *i.e.*, (Inlier Ratio, Precision, Recall). For visibility, in the image pairs, at most 100 randomly selected matches are shown, and we do not show the true negatives. (a) (13.28%, 100.0%, 100.0%). (b) (84.62%, 99.72%, 94.65%). (c) (54.76%, 98.55%, 98.55%). (d) (79.72%, 98.21%, 97.77%). (e) (87.99%, 98.85%, 95.82%). (f) (78.29%, 99.27%, 98.25%). (g) (49.50%, 100.0%, 100.0%). (h) (85.81%, 99.25%, 99.25%). (i) (75.81%, 98.21%, 97.00%).

V. Experimental Results

In this section, we first evaluate the performance of our LMR for general feature matching and test its robustness and generality, and then apply it to solve two vision-based tasks, *i.e.*, visual homing and near-duplicate image retrieval. We use the open source VLFeat toolbox [86] to extract SIFT putative matches and to search nearest neighbors with K-D tree. The experiments are conducted on a desktop with 4.0 GHz Intel Core i7-6700K CPU, 8GB memory, and MATLAB code.

A. Results on Feature Matching

1) Qualitative Illustration: We first present some intuitive results on the matching performance of our LMR. To this end, we test it on 9 representative image pairs undergoing different types of transformations including affine (*e.g.*, Fig. 5a), homography (*e.g.*, Fig. 5b), epipolar geometry (*e.g.*, Figs. 5c-5f), and nonrigid deformation (*e.g.*, Figs. 5g-5i). We use precision and recall to characterize the matching performance, where precision is defined as the percentage of true inliers among those preserved "inlier" by a matching algorithm, and recall is defined as the percentage of preserved true inliers among the whole inliers contained in the original putative set. The ground truth is established by manually checking each putative match in each image pair, and we make the benchmark before conducting experiments to ensure its objectivity. From the results, we see that our learned classifier has strong generalization ability to handle different types of transformations, where very few putative matches are misjudged on all the 9 test pairs.

2) Quantitative Comparison: To provide quantitative comparisons with state-of-the-art competitors, we conduct experiments on four datasets, say *RS* [87], *Retina* [11], *DAISY* [88], and *DTU* [89]. *RS* is a remote sensing dataset consists of 153 image pairs including color-infrared, SAR and panchromatic photographs which suffer from parametric transformation model. *Retina* is a medical image dataset consists of 52 retinal image pairs undergoing non-parametric transformation model. *DAISY* [88] consists of wide baseline image pairs and sequences with ground truth depth maps, in which we create 52 image pairs in total for evaluation. *DTU* contains a lot of different scenes taken from 49 or 64 positions with ground truth camera positions and internal camera parameters, in which we choose two scenes from the dataset (*i.e.*, *Frustum* and *House*) and create 131 image pairs involving relatively large viewpoint changes for evaluation.

For the first two publicly available datasets, the correctness of each feature correspondence in a putative set is determined based on the ground truth information supplied by the corresponding datasets. The rest 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. Seven representative matching algorithms are used for comparison including RANSAC [7], identify correspondence function (ICF) [24], GS [32], manifold regularization-based robust point matching (MR-RPM) [8], GMS [36], LFGC [72], and locality preserving matching (LPM) [4]. In particular, RANSAC is a classical resampling method, ICF and MR-RPM are non-parametric interpolation methods, GS is a graph matching method, GMS and LPM are neighborhood preserving methods, and LFGC is a deep learning method. We implement all the competitors based on publicly available codes and try our best to find optimal parameter settings. Throughout all the experiments, eight algorithms' parameters are all fixed.

The initial inlier ratio, precision, recall and runtime statistics on the four datasets are summarized in Fig. 6. We see that the

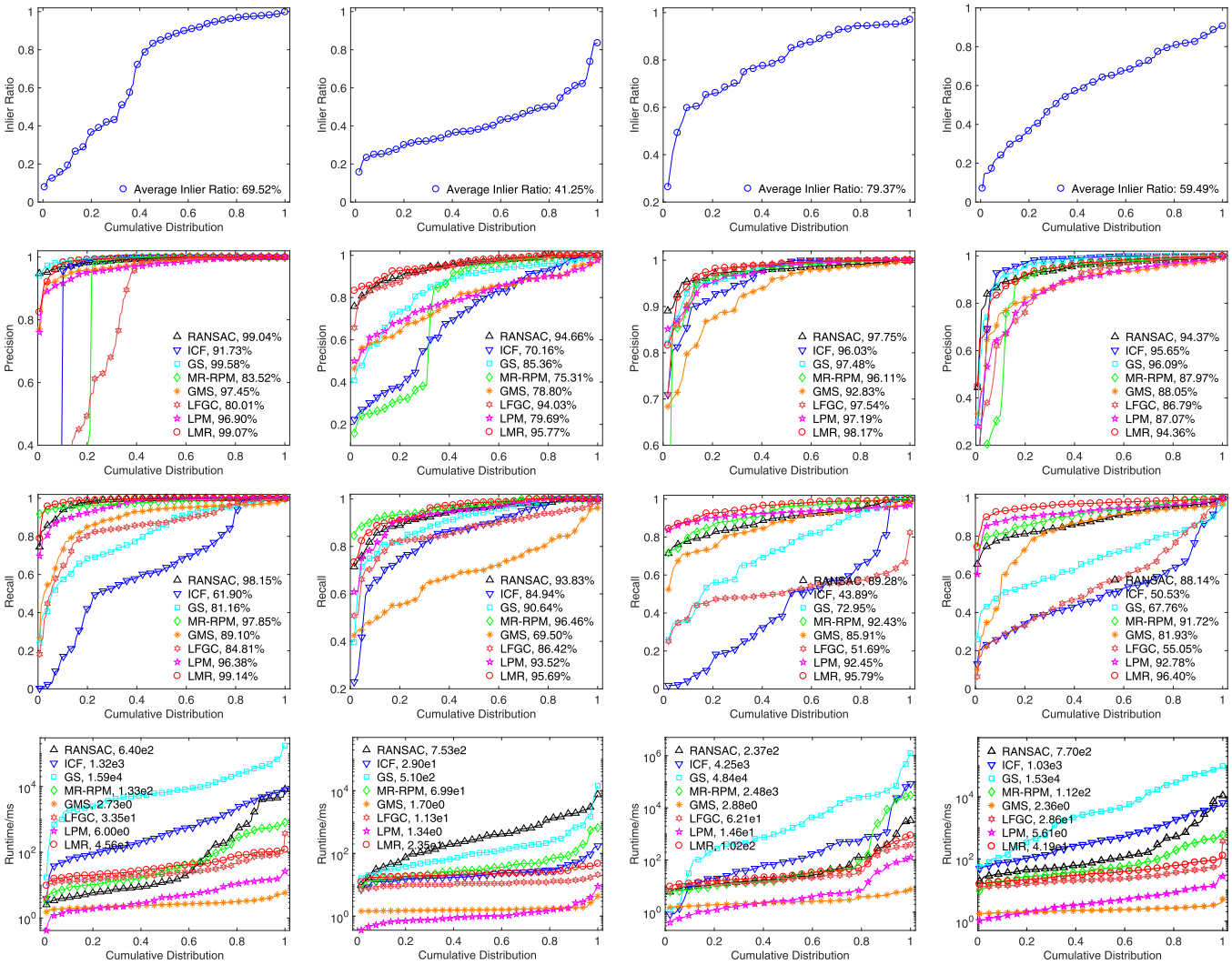


Fig. 6. Quantitative comparison on four datasets. From top to bottom: cumulative distributions of inlier ratio in the putative sets, precision, recall, and runtime. From left to right: results on the datasets of *RS* [87], *Retina* [11], *DAISY* [88], and *DTU* [89]. Seven matching algorithms such as RANSAC [7], ICF [24], GS [32], MR-RPM [8], GMS [36], LFGC [72] and LPM [4] are used for comparison. A point on the curve in the first and third rows with coordinate (x, y) denotes that there are $100 * x$ percent of image pairs which have *Inlier Ratio* or *Runtime* no more than y.

initial inlier ratios, especially in the second dataset, are quite low, which makes the feature matching task challenging. The average numbers of putative matches in the four datasets are about 445, 69, 1,476 and 546, respectively. For the precision and recall statistics, we see that RANSAC can produce satisfying results on all the four datasets. This is due to that we have used enough sampling times to obtain an outlier-free subset for transformation estimation even in case of low initial inlier ratio. ICF and GS usually have high precision or recall, but not simultaneously. MR-RPM works well on most image pairs, but may fail in case of low initial inlier ratio. The performance of GMS is not that satisfying, especially on the *Retina* dataset, due to that it usually requires a larger number of putative matches to achieve better performance, and the consensus of neighborhood topology demonstrated in Fig. 2 cannot be well addressed either. LFGC typically achieves high precision but low recall. This is due to that its main goal is to identify good matches and accurately recover the transformation matrix between two point sets, which may falsely remove a set of unstable true matches, leading to a low recall. In addition, our testing data such as *RS* and *Retina* involving low-overlapped areas or non-rigid deformations are different from the training data of LFGC typically suffer from large scale or viewpoint changes, and the LFGC requires additional ground truth camera intrinsics as input for data normalization, which are not available in our testing data. LPM has high recall, but its precision is badly degraded in case of low initial inlier ratio either. In contrast, our LMR clearly has the best precision-recall trade-off. In addition, our LMR is also quite efficient, and its average runtime on the four datasets is much less than the other state-of-the-art competitors except for GMS, LFGC and LPM.

3) Robustness and Generality Test: We further report the robustness and generality of our LMR. We consider four scenarios for evaluation: i) different degrees of deformation; ii) different types of supervised learning technique; iii) testing data with different feature descriptors for putative match construction; iv) different scales of training data.

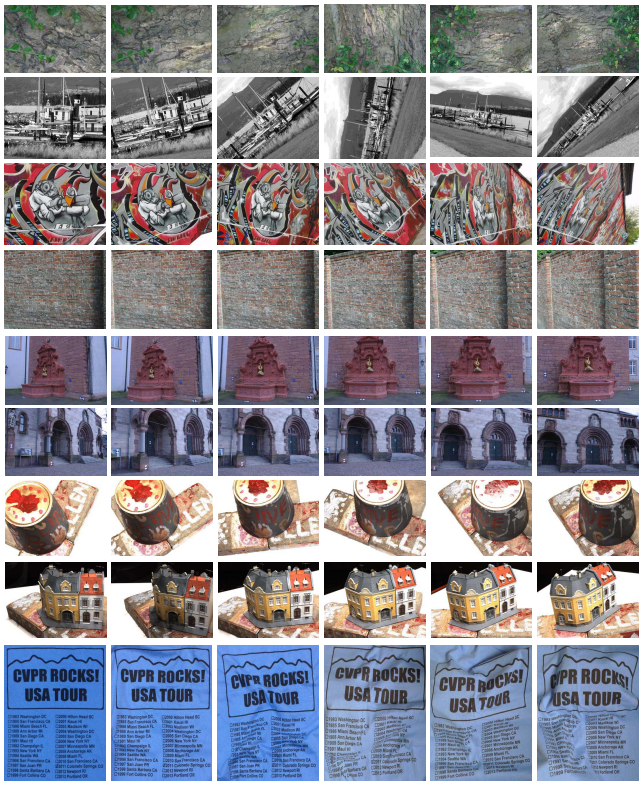


Fig. 7. Different degrees of deformation in 9 scenes. The first 4 rows are selected from the *VGG* [90] dataset, the fifth and sixth rows come from *DAISY* [88], and the next two rows are chosen from *DTU* [89], and the last row is collected by ourselves. From left to right, the deformation degree increases gradually comparing to the first column.

To this end, we first test our method on a group of images with five different degrees of deformation, which includes 8 scenes selected from *VGG* [90], *DAISY* [88] and *DTU* [89], and 1 scene collected by ourselves, as shown in Fig. 7. Note that the images from the second to the last columns are considered as an increasing degree of deformation comparing to the first column. We pairwise the first and the rest images, resulting in 5 image pairs for each scene. We adopt the F-score to characterize the matching performance defined as the harmonic mean of precision and recall [91]: $F\text{-}score = 2{\cdot}Precision{\cdot}Recall/(Precision{+}Recall)$. The mean value and standard deviation of inlier ratio and F-score of all methods with respect to deformation degree are demonstrated in Fig. 8. Clearly, we see that our LMR consistently achieves the best performance over all the other state-of-the-art competitors.

Next, we consider the rest three scenarios. The three widely used supervised learning techniques such as RF, SVM and BPNN and three widely used feature descriptors such as SIFT, speeded-up robust features (SURF) [92], and oriented FAST and rotated BRIEF (ORB) [93] are adopted for quantitative evaluation. In addition to the four datasets in Fig. 6, we use the *VGG* dataset involving all the 40 image pairs as well. For each pair, we set the SIFT distance ratio threshold as 1.5 or 1.0 to construct the *VGG-SFIT* putative set, use the top 60% or 40% similar SURF descriptors as our *VGG-SURF* putative set, and select the top 1,000 or 2,000 correspondences as the *VGG-ORB* putative set. The ground truth correspondences

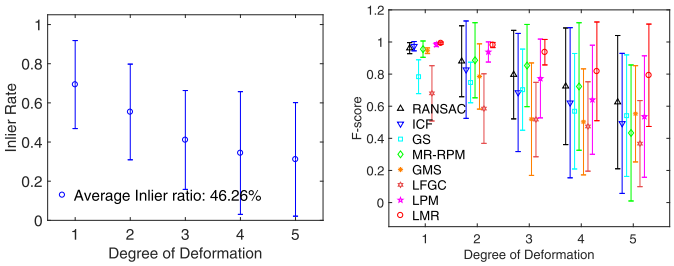


Fig. 8. Quantitative comparison on different degrees of deformation shown in Fig. 7. From left to right: mean and standard deviation of inlier ratio and the F-score of eight methods.

are established according to the ground truth homographies supplied by the dataset. Furthermore, we also change the scale of training data, *e.g.,* 5, 10 and 20 image pairs with 4,391, 7,659 and 15,883 training samples, respectively.

The average F-scores of the 7 competitors and the 3 learning techniques with 3 different scales of training sets for our LMR on each dataset are summarized in Table I. We see that our LMR can achieve the best performances on all datasets including the *VGG* dataset with different feature descriptors. By fixing the scale of training data, the performance of the three supervised learning techniques are quite similar, where none of them is obviously better than the others. In particular, with only 10 image pairs to construct the training samples, our LMR can generate a reliable classifier and achieve promising results. For different scales of training data, using 10 image pairs for training performs better than using 5 image pairs, and the performance of using 20 image pairs is not improved much, even degrades in some cases. This is because the selected 10 image pairs are representative to generate good performance for non-deep learning techniques, and adding another arbitrary 10 image pairs for training may introduce additional noise, especially when the classifier's performance becomes saturated. From the results in Table I, we can draw a conclusion that our LMR is general and does not rely on any specific learning technique, feature descriptor, or large scale training data, to improve the matching performance. Note that the training sample of our LMR is quite different from that of the deep learning-based method LFGC [72]. Specifically, a training sample is just a putative match in our LMR, but a whole putative set in LFGC.[9] This is why our LMR requires only 10 image pairs but LFGC requires thousands of image pairs for training.

B. Results on Visual Homing

It has been verified that the presence and amount of false matches injure the robustness of a visual homing method [94]. Therefore, some heuristic methods expect improvement from removing false matches. In our evaluation, we replace the feature matching strategies with our LMR in several commonly adopted homing methods, and test the difference of the homing performance. The involved methods are homing in scale-space (HiSS), bearing-only visual servoing (BOVS), and scale-only visual servoing (SOVS) from [84], as well

[9]The LFGC needs a whole putative set to estimate the camera intrinsics.

TABLE I

AVERAGE F-SCORE OF OUR LMR AND 7 COMPETITORS ON *RS* [87], *Retina* [11], *DAISY* [88], *DTU* [89] AND *VGG* DATASETS [90]. FOR *VGG*, WE USE 3 DESCRIPTORS SUCH AS SIFT, SURF, ORB TO CONSTRUCT PUTATIVE MATCHES AND TERMED AS *VGG-SIFT*, *VGG-SURF* AND *VGG-ORB*, RESPECTIVELY. FOR EACH DATASET, THE AVERAGE INLIER NUMBER (AIN) AND AVERAGE INLIER RATE (AIR) ARE SHOWN IN THE SECOND AND THIRD ROWS, FOLLOWING ARE THE RESULTS OF 7 COMPETITORS. FOR OUR LMR, WE CHANGE THE SCALE OF TRAINING DATA WITH 5, 10 AND 20 IMAGE PAIRS INVOLVING 4,391, 7,659 AND 15,883 TRAINING SAMPLES, AND USE DIFFERENT TYPES OF SUPERVISED LEARNING TECHNIQUE FOR TRAINING. IN PARTICULAR, LMR-RF-5 MEANS THE CLASSIFIER IS TRAINED USING RANDOM FOREST WITH 5 TRAINING IMAGE PAIRS. BOLD INDICATES THE BEST RESULT

	RS	*Retina*	*DAISY*	*DTU*	*VGG-SIFT*	*VGG-SURF*	*VGG-ORB*
AIN	445.34	69.029	1476.2	545.99	693.18	625.78	843.34
AIR	0.6952	0.4125	0.7937	0.5949	0.8810	0.5761	0.5224
RANSAC [7]	0.9855	0.9408	0.9317	0.9090	0.9530	0.9484	0.9489
ICF [24]	0.6504	0.7195	0.5411	0.6254	0.9603	0.8975	0.9287
GS [32]	0.8838	0.8675	0.8149	0.7808	0.9305	0.8345	0.8405
MR-RPM [8]	0.8532	0.7979	0.9360	0.8699	0.9224	0.9070	0.9273
GMS [36]	0.9255	0.7313	0.8900	0.8401	0.8558	0.8169	0.9224
LFGC [72]	0.7925	0.8953	0.6694	0.6504	0.8707	0.7685	0.8195
LPM [4]	0.9656	0.8579	0.9474	0.8920	0.9598	0.9154	0.9477
LMR-RF-5	0.9929	0.9163	0.9625	0.9434	0.9686	0.9489	0.9537
LMR-SVM-5	0.9867	0.8904	0.9372	0.9135	0.9678	0.9373	0.9386
LMR-BPNN-5	0.9842	0.9382	0.9626	0.9321	0.9509	0.9264	0.9285
LMR-RF-10	0.9932	0.9473	0.9754	0.9489	**0.9710**	0.9572	**0.9644**
LMR-SVM-10	0.9936	0.9470	0.9773	0.9497	0.9697	0.9542	0.9597
LMR-BPNN-10	0.9908	**0.9563**	0.9694	**0.9518**	0.9552	0.9354	0.9515
LMR-RF-20	**0.9937**	0.9425	**0.9780**	0.9455	0.9689	**0.9574**	0.9640
LMR-SVM-20	0.9932	0.9305	0.9772	0.9409	0.9701	0.9549	0.9643
LMR-BPNN-20	0.9895	0.9401	0.9666	0.9326	0.9612	0.9428	0.9536

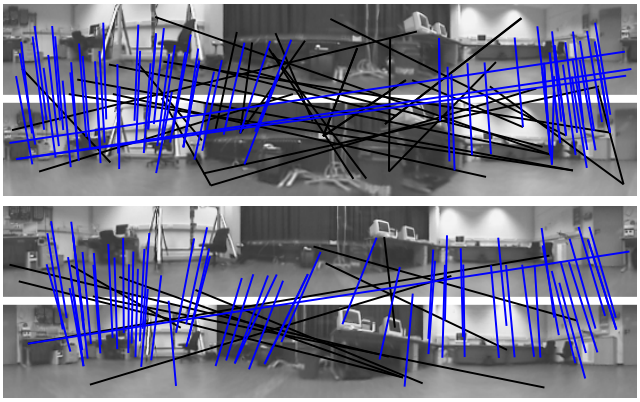


Fig. 9. Matching results of LMR on two typical panoramic pairs.

as scale and bearing visual servoing (SBVS), and simplified scale-based visual servoing (SSVS) from [13]. In addition, the LPM [4] is also used to replace the matching strategies in these methods for comparison.

We choose the *A1originalH* scene, a widely used panoramic image database for visual homing,[10] for quantitative evaluation. It consists of 170 omni-directional and unwrapped images captured in an indoor environment with ground truth capturing positions. Four metrics including total average angular error (TAAE), minimal error (Min), maximal error (Max) and standard variation of error (StdVar) as in [13] are employed to measure the performance. For all the metrics, smaller values indicate better results.

Figure 9 depicts the matching results on two typical image pairs from the dataset, from which we see that all the inliers and outliers on the two examples are correctly identified. Note that there are several inliers across the whole images and our method can also correctly identify them. This is because the scenes are panoramic images and we have replaced the Euclidean distance by the cylinder distance [3] for neighborhood construction. The homing vector errors on the whole dataset are listed in Table II. Clearly, our LMR can consistently improve the state-of-the-art, due to its ability of generating more accurate feature matches.

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

TABLE II

VISUAL HOMING ERROR STATISTICS OF DIFFERENT METHODS ON *A1originalH* DATASET. BOLD INDICATES THE BEST RESULT (UNIT: DEGREE)

	TAAE	Min	Max	StdVar
HISS	14.67	8.05	36.40	5.42
HISS+LPM	14.28	7.85	36.23	5.19
HISS+LMR	**14.14**	**7.73**	**36.09**	**4.89**
BOVS	27.41	10.24	70.69	11.91
BOVS+LPM	14.50	**4.05**	44.37	9.33
BOVS+LMR	**14.37**	4.12	**43.29**	**9.15**
SOVS	18.75	10.34	37.81	5.81
SOVS+LPM	16.76	9.04	**33.70**	4.71
SOVS+LMR	**16.53**	**8.76**	34.12	**4.66**
SBVS	15.90	8.57	34.43	5.86
SBVS+LPM	13.52	7.03	31.23	5.00
SBVS+LMR	**13.31**	**6.81**	**31.21**	**4.93**
SSVS	12.59	6.50	28.36	4.17
SSVS+LPM	11.44	6.53	25.89	**3.89**
SSVS+LMR	**11.38**	**6.47**	**24.45**	3.95

C. Results on Near-Duplicate Image Retrieval

Finally, we test our LMR for near-duplicate image retrieval on the *California-ND* dataset [95]. All the categories with 10 or more images are enlisted, and for each category 10 images are randomly selected for quantitative evaluation,

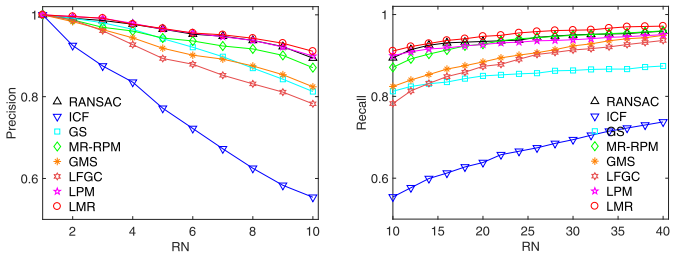


Fig. 10. Precisions (left) and recalls (right) with respect to RN, *i.e.*, the required number of images to be retrieved for a given image.

resulting in 7, 140 image pairs in total. The matching algorithms on all the 7, 140 image pairs are executed and the number of preserved matches is employed to measure the similarity between two images. A ranked list for each given image according to its similarities with every other image in the dataset is returned. The performance is characterized by precision and recall based on the ranked lists. The precision is valid only for $RN \leq 10$ and the recall is valid for $RN \geq 10$ with RN denoting the number of retrieved images.

The precision and recall curves of RANSAC, ICF, GS, MR-RPM, GMS, LFGC, LPM and our LMR are provided in Fig. 10. Our LMR evidently outperforms all the other methods in both precision and recall, followed by LPM. Specifically, the average retrieved correct image numbers of RANSAC, ICF, GS, MR-RPM, GMS, LFGC, LPM and our LMR for $RN = 10$ are approximately 8.94, 5.54, 8.13, 8.71, 8.24, 7.83, 8.99, and 9.11, respectively.

VI. Conclusion

In this paper, we proposed a novel learning-based approach to identify inlier and outlier for local feature matching. It is able to produce a general classifier to determine the correctness of an arbitrary putative match within linearithmic time complexity. The qualitative and quantitative results on general feature matching as well as two real-world tasks demonstrate the superiority of our strategy over state-of-the-art competitors in terms of both accuracy and efficiency. In addition, thanks to the generality, our method can also be used to provide a good initialization for more complicated problem-specific matching algorithms which rely on certain special transformation models.

References

[1] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *Int. J. Comput. Vis.*, vol. 59, no. 1, pp. 61–85, 2004.

[2] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, Apr. 2009.

[3] J. Zhao and J. Ma, "Visual homing by robust interpolation for sparse motion flow," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 1282–1288.

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

[5] X. Guo, Y. Li, J. Ma, and H. Ling, "Mutually guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published. doi: 10.1109/TPAMI.2018.2883553.

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

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

[8] J. Ma, J. Wu, J. Zhao, J. Jiang, H. Zhou, and Q. Z. Sheng, "Nonrigid point set registration with robust transformation learning under manifold regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published. doi: 10.1109/TNNLS.2018.2872528.

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

[10] X. Guo and X. Cao, "Good match exploration using triangle constraint," *Pattern Recognit. Lett.*, vol. 33, no. 7, pp. 872–881, 2012.

[11] J. Ma, J. Jiang, C. Liu, and Y. Li, "Feature guided Gaussian mixture model with semi-supervised EM and local geometric constraint for retinal image registration," *Inf. Sci.*, vol. 417, pp. 128–142, Nov. 2017.

[12] J. Ma, H. Zhou, J. Zhao, Y. Gao, J. Jiang, and J. Tian, "Robust feature matching for remote sensing image registration via locally linear transforming," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6469–6481, Dec. 2015.

[13] M. Liu, C. Pradalier, and R. Siegwart, "Visual homing from scale with an uncalibrated omnidirectional camera," *IEEE Trans. Robot.*, vol. 29, no. 6, pp. 1353–1365, Dec. 2013.

[14] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu, "Robust point matching via vector field consensus," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1706–1721, Apr. 2014.

[15] O. Pele and M. Werman, "A linear time histogram metric for improved SIFT matching," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 495–508.

[16] Y. T. Hu, Y. Y. Lin, H. Y. Chen, K. J. Hsu, and B. Y. Chen, "Matching images with multiple descriptors: An unsupervised approach for locally adaptive descriptor selection," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5995–6010, Dec. 2015.

[17] C. Wang, L. Wang, and L. Liu, "Progressive mode-seeking on graphs for sparse feature matching," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 788–802.

[18] M. Cho and K. M. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 398–405.

[19] P. J. Huber, *Robust Statistics*. New York, NY, USA: Wiley, 1981.

[20] Y. Liu, L. De Dominicis, B. Wei, L. Chen, and R. R. Martin, "Regularization based iterative point match weighting for accurate rigid transformation estimation," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 9, pp. 1058–1071, Sep. 2015.

[21] J. Maier, M. Humenberger, M. Murschitz, O. Zendel, and M. Vincze, "Guided matching based on statistical optical flow for fast and robust correspondence analysis," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 101–117.

[22] O. Chum and J. Matas, "Matching with PROSAC—Progressive sample consensus," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 220–226.

[23] T. Sattler, B. Leibe, and L. Kobbelt, "SCRAMSAC: Improving RANSAC's efficiency with a spatial consistency filter," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2090–2097.

[24] X. Li and Z. Hu, "Rejecting mismatches by correspondence function," *Int. J. Comput. Vis.*, vol. 89, no. 1, pp. 1–17, 2010.

[25] J. Ma, W. Qiu, J. Zhao, Y. Ma, A. L. Yuille, and Z. Tu, "Robust L_2E estimation of transformation for non-rigid registration," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1115–1129, Mar. 2015.

[26] G. Wang, Z. Wang, Y. Chen, X. Liu, Y. Ren, and L. Peng, "Learning coherent vector fields for robust point matching under manifold regularization," *Neurocomputing*, vol. 216, pp. 393–401, Dec. 2016.

[27] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1482–1489.

[28] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 596–609.

[29] M. Cho and K. M. Lee, "Mode-seeking on graphs via random walks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 606–613.

[30] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, Jun. 2016.

[31] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu, "Consistency-driven alternating optimization for multigraph matching: A unified approach," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 994–1009, Mar. 2015.

[32] H. Liu and S. Yan, "Common visual pattern discovery via spatially coherent correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1609–1616.

[33] W.-Y. Lin, M.-M. Cheng, J. Lu, H. Yang, M. N. Do, and P. Torr, "Bilateral functions for global motion modeling," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 341–356.

[34] W.-Y. Lin, M.-M. Cheng, S. Zheng, J. Lu, and N. Crook, "Robust non-parametric data fitting for correspondence modeling," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2376–2383.

[35] W.-Y. Lin *et al.*, "CODE: Coherence based decision boundaries for feature correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 34–47, Jan. 2018.

[36] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M. M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4181–4190.

[37] P. J. Besl and D. N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[38] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Understand.*, vol. 89, nos. 2–3, pp. 114–141, Feb. 2003.

[39] Y. Yang, S. H. Ong, and K. W. C. Foong, "A robust global and local mixture distance based non-rigid point set registration," *Pattern Recognit.*, vol. 48, no. 1, pp. 156–173, 2015.

[40] F. Boughorbel, A. Koschan, B. Abidi, and M. Abidi, "Gaussian fields: A new criterion for 3D rigid registration," *Pattern Recognit.*, vol. 37, no. 7, pp. 1567–1571, 2004.

[41] J. Ma, J. Zhao, Y. Ma, and J. Tian, "Non-rigid visible and infrared face registration via regularized Gaussian fields criterion," *Pattern Recognit.*, vol. 48, no. 3, pp. 772–784, 2015.

[42] G. Wang, Z. Wang, Y. Chen, Q. Zhou, and W. Zhao, "Context-aware Gaussian fields for non-rigid point set registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5811–5819.

[43] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[44] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, "Rigid and articulated point registration with expectation conditional maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 587–602, Mar. 2011.

[45] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.

[46] J. Ma, J. Zhao, and A. L. Yuille, "Non-rigid point set registration by preserving global and local structures," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 53–64, Jan. 2016.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[48] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[49] J. Redmon and A. Farhadi. (2018). "YOLOv3: An incremental improvement." [Online]. Available: https://arxiv.org/abs/1804.02767

[50] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 118–126.

[51] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3279–3286.

[52] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, nos. 1–32, p. 2, 2016.

[53] H. Altwaijry, A. Veit, S. J. Belongie, and C. Tech, "Learning to detect and match keypoints with deep architectures," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–12.

[54] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2414–2422.

[55] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 467–483.

[56] G. Georgakis, S. Karanam, Z. Wu, J. Ernst, and J. Kosecka, "End-to-end learning of keypoint detector and descriptor for pose invariant 3D matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1965–1973.

[57] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1482–1491.

[58] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 661–669.

[59] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4353–4361.

[60] H. Altwaijry, E. Trulls, J. Hays, P. Fua, and S. Belongie, "Learning to match aerial images with deep attentive architectures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3539–3547.

[61] S. En, A. Lechervy, and F. Jurie, "TS-NET: Combining modality specific and common features for multimodal patch matching," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2018, pp. 3024–3028.

[62] S. Miao, Z. J. Wang, and R. Liao, "A CNN regression approach for real-time 2D/3D registration," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1352–1363, May 2016.

[63] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9252–9260.

[64] P. Jiang and J. A. Shackleford, "CNN driven sparse multi-level B-spline image registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9281–9289.

[65] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "DeepMatching: Hierarchical deformable dense matching," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 300–323, 2016.

[66] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1851–1858.

[67] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 652–660.

[68] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 195–205.

[69] F. J. Lawin, M. Danelljan, F. S. Khan, P.-E. Forssen, and M. Felsberg, "Density adaptive point set registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3829–3837.

[70] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2009, pp. 3212–3217.

[71] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[72] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2666–2674.

[73] Y. Zheng and D. Doermann, "Robust point matching for nonrigid shapes by preserving local neighborhood structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 643–649, Apr. 2006.

[74] O. Chum, M. Perd'och, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 17–24.

[75] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling features for large scale partial-duplicate Web image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 25–32.

[76] H. J. Kim, E. Dunn, and J.-M. Frahm, "Predicting good features for image geo-localization using per-bundle VLAD," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1170–1178.

[77] X. Wu and K. Kashino, "Robust spatial matching as ensemble of weak geometric relations," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 25.1–25.12.

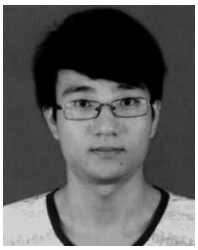
[78] D. Nasuto and J. B. R. Craddock, "NAPSAC: High noise, high dimensional robust estimation-it's in the bag," in *Proc. Brit. Mach. Vision Conf.*, 2002, pp. 458–467.

[79] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

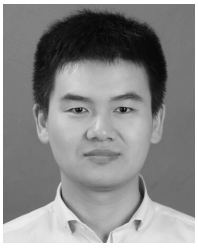
[80] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[81] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[82] C. A. Micchelli and M. A. Pontil, "On learning vector-valued functions," *Neural Comput.*, vol. 17, no. 1, pp. 177–204, 2005.
[83] R. Möller and A. Vardy, "Local visual homing by matched-filter descent in image distances," *Biol. Cybern.*, vol. 95, no. 5, pp. 413–430, 2006.
[84] D. Churchill and A. Vardy, "An orientation invariant visual homing algorithm," *J. Intell. Robotic Syst.*, vol. 71, no. 1, pp. 3–29, 2013.
[85] J. Chen, Y. Wang, L. Luo, J.-G. Yu, and J. Ma, "Image retrieval based on image-to-class similarity," *Pattern Recognit. Lett.*, vol. 83, pp. 379–387, Nov. 2016.
[86] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 1469–1472.
[87] J. Ma, J. Jiang, H. Zhou, J. Zhao, and X. Guo, "Guided locality preserving feature matching for remote sensing image registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4435–4447, Aug. 2018.
[88] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.
[89] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 153–168, 2016.
[90] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, no. 1, pp. 43–72, 2005.
[91] Y. Lin, Z. Lin, and H. Zha, "The shape interaction matrix-based affine invariant mismatch removal for partial-duplicate image search," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 561–573, Feb. 2017.
[92] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
[93] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
[94] D. Schroeter and P. Newman, "On the robustness of visual homing under landmark uncertainty," in *Intelligent Autonomous Systems*, vol. 10. Clifton, VA, USA: IOS Press, 2008, pp. 278–287.
[95] A. Jinda-Apiraksa, V. Vonikakis, and S. Winkler, "California-ND: An annotated dataset for near-duplicate detection in personal photo collections," in *Proc. QoMEX*, Jul. 2013, pp. 142–147.

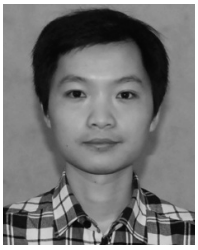
Jiayi Ma received the B.S. degree in information and computing science and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively. From 2012 to 2013, he was an Exchange Student with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA, USA. From 2014 to 2015, he was a Post-Doctoral Researcher with the Electronic Information School, Wuhan University, Wuhan, where he is currently an Associate Professor. He has authored or co-authored over 100 refereed journal and conference papers, including IEEE TPAMI/TIP/TSP/TNNLS/TGRS/TCYB/TMM/TCSVT, IJCV, CVPR, IJCAI, AAAI, ICRA, IROS, and ACM MM. His current research interests include the areas of computer vision, machine learning, and pattern recognition. He received the Natural Science Award of Hubei Province (First Class) as the first author. He also received the Chinese Association for Artificial Intelligence (CAAI) Excellent Doctoral Dissertation Award (a total of eight winners in China) and the Chinese Association of Automation (CAA) Excellent Doctoral Dissertation Award (a total of ten winners in China). He has been an Associate Editor of IEEE ACCESS, since 2017, and *Neurocomputing*, since 2019, and a Guest Editor of *Remote Sensing*.



Xingyu Jiang received the B.E. degree from the Department of Mechanical and Electronic Engineering, Huazhong Agricultural University, Wuhan, China, in 2017. He is currently pursuing the Ph.D. degree with the Electronic Information School, Wuhan University. His research interests include computer vision, machine learning, and pattern recognition.



Junjun Jiang received the B.S. degree from the Department of Mathematics, Huaqiao University, Quanzhou, China, in 2009, and the Ph.D. degree from the School of Computer, Wuhan University, Wuhan, China, in 2014. From 2015 to 2018, he was an Associate Professor with the China University of Geosciences, Wuhan. Since 2016, he has been a Project Researcher with the National Institute of Informatics, Tokyo, Japan. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. His research interests include image processing and computer vision. He received the Best Student Paper Runner-Up Award at the MMM 2015. He also received the 2015 ACM Wuhan Doctoral Dissertation Award and the 2016 China Computer Federation (CCF) Outstanding Doctoral Dissertation Award. He was the Finalist of the World's FIRST 10K Best Paper Award at the ICME 2017.



Ji Zhao received the B.S. degree in automation from the Nanjing University of Posts and Telecommunication in 2005 and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology in 2012. From 2012 to 2014, he was a Post-Doctoral Research Associate with the Robotics Institute, Carnegie Mellon University. He is currently a Research Scientist with the Electronic Information School, Wuhan University, Wuhan, China.

Xiaojie Guo received the B.E. degree in software engineering from the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China, in 2008, and the M.S. and Ph.D. degrees in computer science from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2010 and 2013, respectively. He is currently an Associate Professor with the School of Computer Software, Tianjin University. He was a recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition (International Association on Pattern Recognition) in 2010.