The Shape Interaction Matrix-Based Affine Invariant Mismatch Removal for Partial-Duplicate Image Search

Yang Lin, Zhouchen Lin, *Senior Member, IEEE*, and Hongbin Zha, *Member, IEEE*

***Abstract*— Mismatch removal is a key step in many computer vision problems. In this paper, we handle the mismatch removal problem by adopting shape interaction matrix (SIM). Given the homogeneous coordinates of the two corresponding point sets, we first compute the SIMs of the two point sets. Then, we detect the mismatches by picking out the most different entries between the two SIMs. Even under strong affine transformations, outliers, noises, and burstiness, our method can still work well. Actually, this paper is the first non-iterative mismatch removal method that achieves affine invariance. Extensive results on synthetic 2D points matching data sets and real image matching data sets verify the effectiveness, efficiency, and robustness of our method in removing mismatches. Moreover, when applied to partial-duplicate image search, our method reaches higher retrieval precisions with shorter time cost compared with the state-of-the-art geometric verification methods.**

***Index Terms*— Image matching, mismatch removal, shape interaction matrix, affine invariance, image retrieval.**

I. Introduction

REMOVING the mismatches from two given corresponding point sets is a fundamental problem in computer vision. For many applications in this field, such as structure-from-motion recovery [1], [2], registration [3], [4], stereo matching [5], object recognition [6], tracking [7], and partial-duplicate image search [8]–[13], the first step is to compute the point correspondences of two point sets, which can also be regarded as a matching problem. The more accurate matching result we obtain, the better performance of the subsequent computer vision task we will achieve.

The point sets could be extracted from two dimensional images (e.g., SIFT [14], SURF [15], ASIFT [16]), or three dimensional depth surfaces (e.g., MeshDoG [17]). After detecting the feature points, the next step is to determine the putative matches between the two point sets. There are two constraints that putative matches should satisfy. One is the descriptor consistency, which enforces that only the features having similar descriptors are matched; the other is called the geometric invariance, which assumes that the true matches share a unified spatial transformation. The above two constraints are usually combined to filter the mismatches. More specifically, by utilizing the descriptor consistency, we can acquire enough putative matches first. However, the descriptor consistency can only be treated as a sufficient constraint, because a point could be matched with more than one points in different locations that share similar descriptors, which leads to 1-vs-N mismatches. To remove the ambiguities of the descriptors, the second step is to filter mismatches from the putative matches which do not satisfy the geometric invariance, and keep the remaining true matches for further processing. There are mainly three difficulties in mismatch removal, including the deviation of the feature location caused by the detector, a large percentage of unmatched features introduced by partial occlusions or the limitation of the detectors, and unknown type of geometric relation between two images. A good mismatch removal approach should overcome noises and outliers under different geometric transformations.

A typical application of mismatch removal is partial-duplicate image search based on Bag-of-Features (a.k.a., BoF) [19], [20]. The idea is that although using BoF to index the SIFT descriptors can reduce the matching cost dramatically, it will certainly introduce repetitive visual regions, i.e., the *"burstiness"* phenomenon observed in [18]. The *"burstiness"* means some visual features appear many times in both two matched images, which will cause 1-vs-N mismatches. Such 1-vs-N mismatches happen prevalently in man-made and natural scenes (see Figure 1). For many large scale image search systems, the total number of the matches is commonly used as a similarity for re-ranking. If an irrelevant image shares many 1-vs-N mismatches with the query image, it will be unsatisfactorily ranked at the top of the retrieval result. Hence, mismatch removal can be utilized to filter the 1-vs-N mismatches from the putative matches and re-rank the coarse retrieval result by the number of remaining true matches. The above process is also called the geometric verification in the image search area.

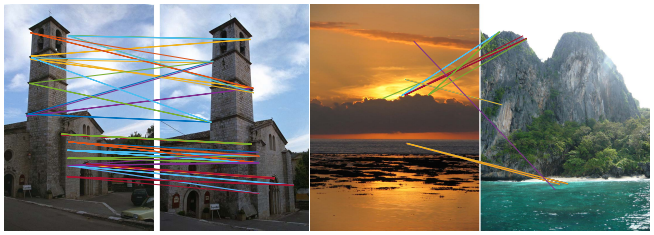
There are mainly two categories of mismatch removal methods. One is iterative fitting methods, and the other is non-iterative filtering methods. The iterative fitting methods are able to estimate complicated transformations, but have to spend much time on iterative fitting. Different from the

Manuscript received February 5, 2016; revised July 25, 2016; accepted November 1, 2016. Date of publication November 16, 2016; date of current version December 2, 2016. The work of Z. Lin was supported in part by the National Basic Research Program of China (973 Program) under Grant 2015CB352502, in part by the National Natural Science Foundation of China under Grant 61625301 and Grant 61231002, and in part by the Qualcomm. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ling Shao. *(Corresponding author: Zhouchen Lin.)*

The authors are with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China, and also with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: linyang@cis.pku.edu.cn; zlin@pku.edu.cn; zha@cis.pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2629452



(a) (b)

Fig. 1. Examples of the **burstiness** [18], i.e., 1-vs-N visual word mismatches. Lines of the same color represent 1-vs-N mismatches that share the same visual word (best viewed in color). The images on the left column are the query images, and the compared images are on the right column. One can see that 1-vs-N mismatches exist in both **(a)** relevant and **(b)** irrelevant images, and in both (a) man-made and (b) natural scenes.

iterative fitting methods, the non-iterative filtering methods try to directly and efficiently filter the mismatches without estimating a unified geometric transformation model. However, such methods may fail when affine transformations exist.

Motivated by the above state-of-the-arts, our goal in this paper is to make a trade-off between geometric invariance and time cost. More specifically, we try to find a global spatial representation of a point set with affine transformation invariance that can be compared easily without expensive iterations. In the following we review some previous works about affine invariant spatial representation.

Many works have been done to define an affine invariant expression of a set of points. Werman and Weinshall [21] give a distance up to 2D affine transformations between two point sets. Zhang et al. [22] propose a genetic algorithm based method under general affine transformations by using partial Hausdorff distance. Begelfor and Werman [23] propose a Riemannian geometric framework to compute mean and distributions of point sets so that different configurations up to affine transformations are considered to be the same. The above mentioned methods give us a way to compute an affine invariant global similarity between two point sets, but none of them are able to distinguish the mismatches. Based on complex numbers, Ha and Moura [24] and Ho et al. [25] propose two affine matching algorithms that can handle mismatches, but they are only applicable to 2D point sets. Recently, Guo and Cao [26] propose a method to find more true matches which is robust to affine changes. They first select some features by Bi-matching as seed points, and organize them by adopting the Delaunay triangulation algorithm. Then the true matches are explored by utilizing a triangle-constraint. Here the triangle-constraint means that two matched points are true match if only they share the same linear combination parameters when adopting the vertexes of their neighbor triangle as the linear bases. Motivated by their idea, instead of utilizing only the three vertexes as bases, we try to find a more global way to represent the feature by considering the whole point set under any affine transformations.

A. Our Contributions and Advantages

In this paper, we address the mismatch removal problem by using the shape interaction matrix (SIM), which was first introduced by Costeira and Kanade [27] for multi-body segmentations. The SIM has been widely used for subspace segmentation to characterize the geometric relationship between data points. Given putative matches, we first compute SIMs of the two point sets to characterize the affine invariance of each point set. Then we compare the obtained two SIMs. The mismatches can be easily determined by picking out the most inconsistent entries between the two SIMs. We illustrate our work-flow in Figure 2. The main contributions of our method can be summarized as follows:

- We discover that the shape interaction matrix is affine invariant.
- We apply the shape interaction matrix to mismatch removal to achieve affine invariance. To the best of our knowledge, we are the first to achieve affine invariance **without iterations** for the mismatch removal problem.
- We design an effective and simple algorithm to detect the mismatches robustly when outliers and burstiness phenomenon exist.

Compared with the state-of-the-arts, our method has three advantages:

- Our method is simple. We only utilize the location of features as input to filter the mismatches, while most of the traditional geometric verification methods need extra spatial prior (e.g., scale and orientation of SIFT features).
- Our algorithm avoids expensive iterative fitting. The SIMs of the two point sets can be independently computed by a closed-form solution. The following matching step is also quite simple and effective, which is suitable for partial-duplicate image search as a geometric verification step. On the contrary, most of the state-of-the-arts need iterative matching steps, which are time-consuming.
- Our model is robust to affine changes. The SIM model we adopt provides a theoretical guarantee to exactly obtain the same representations from two corresponding point sets under any affine transformations (e.g., rotation, scale, and skew changes).

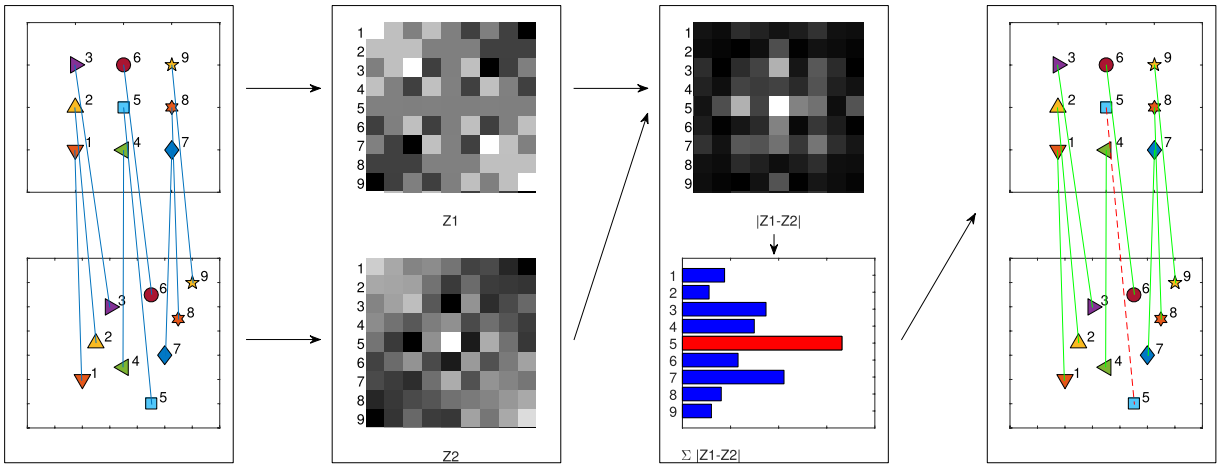
II. Related Work

Before introducing our model, we first review the recent development of mismatch removal. We classify the mismatch removal methods into two categories. One is iterative fitting methods, the other is non-iterative filtering methods.

A. Iterative Fitting Methods

To handle more complex transformations, such as affine, perspective, and non-rigid transformations, the iterative fitting methods alternately estimating the geometric transformation model and the true matches by several iterations. The putative matches can be first utilized to compute a hypothetical geometric transformation model. Once the model is acquired, the fitting error of each correspondence is computed to determine whether it belongs to true matches or mismatches. Some iterative fitting methods are introduced as follows.

RANSAC As a classical method, RANSAC [28] and its variants (e.g., MLESAC) [29]) tried to iteratively estimate the perspective transformations relationship between the true matches. Instead of maximizing the number of true matches



(a) Putative matches. (b) SIMs. (c) Corruptions. (d) Detected mismatch.

Fig. 2. The work-flow of our mismatch removal method. (a) Putative local feature matches between two images with affine changes. (b) Computed shape interaction matrices (SIM) of the two feature point sets. (c) Corruptions between the two SIM. (d) Detected mismatches. Compared with the other feature pairs, the locations of the feature pair No. 5 in two images are inconsistent with each other, which do not satisfy the geometric invariance constraint. With our method, the inconsistently matched feature pair No.5 is correctly detected as a mismatch, represented by the red dashed line.

adopted by RANSAC, MLESAC estimated the inliers by maximizing a likelihood function, which is more general. However, the above two methods are time-consuming, non-deterministic, and sensitive to the percentage of outliers.

ICF Li and Hu [30] introduced support vector machine to learn a pair of correspondence functions that mutually map one point set to the other, then rejected the mismatches by checking whether they are consistent with the two estimated correspondence functions. Since they use a radial basis kernel, it is applicable to non-rigid deformations.

VFC Zhao et al. [31] and Ma et al. [32] investigated a vector field learning method. It learns an interpolated vector motion field that fits the putative matches based on the Tiknonov regularization in a vector-valued reproducing kernel Hilbert space, and simultaneously estimate the true matches by the EM algorithm. They also provided a linear time complexity version called FastVFC and a sparse approximation one named SparseVFC, which speeds up significantly without large performance degradation.

Compared with RANSAC and MLESAC, ICF and VFC can handle affine, perspective, and even non-rigid deformations when large percentage of outliers exist. However, they are still not very efficient when applied to partial-duplicate image search as a geometric verification step. In particular, ICF estimates two non-parametric model correspondence functions, and VFC learns a vector field mapping, whose time cost is unacceptable when applied to partial-duplicate image search.

B. Non-Iterative Filtering Methods

As a key step to achieving high precision, in the partial-duplicate image search area, a number of non-iterative filtering methods are proposed to efficiently verify the geometric invariance after BoF matching. Such methods are based on the following similarity transformation model:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = s \cdot \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \quad (1)$$

where (x_1, y_1) and (x_2, y_2) are the locations of matched feature points, s is a scaling parameter, θ is a rotation parameter, and (t_x, t_y) is the translation vector.

WGC Jegou et al. [8] proposed a simple way to estimate the two parameters:

$$s = \hat{s}_2/\hat{s}_1, \theta = \hat{\theta}_2 - \hat{\theta}_1, \quad (2)$$

where $\hat{s}_1$, $\hat{s}_2$, $\hat{\theta}_1$, and $\hat{\theta}_2$ are the scales and dominant orientations of two matched SIFT features. The idea of WGC is that most of the true matches share unified similarity transformation, so the scale parameter s and the rotation parameter θ of the true matches should be close to each other. In detail, the peak value of the histograms of scale parameter and rotation parameter are used to measure the similarity between two images, and the matches within this range are true matches.

EWGC Instead of using s and θ, Zhao et al. [9] utilized the l_2 norm of the translation vector (t_x, t_y) as the statistic to compute the peak value of the histograms. Given the locations, scales, and dominant orientations, the translation vector (t_x, t_y) can be computed by (1) and (2).

SGC Unlike WGC and EWGC, Wang et al. [12] proposed to group the matches by rotation first. Then for each group, they computed the peak values in the histogram of the translation vectors (t_x, t_y) as the similarity to filter mismatches.

With scales and dominant orientations in SIFT, the above three approaches try to estimate a similarity transformation model from the aspect of isolated feature patches, hence they cannot make use of the geometric prior beyond the isolated features. Another set of methods, instead, focus on how to build a relatively strong similarity transformation model from the aspect of the global spatial relationship. Such global methods are described as follows.

GC Zhou et al. [10], [13] designed a similarity invariant geometric coding map to encode the spatial relationship between features. The coding map consists of geometric square coding and fan coding, which can achieve scale and rotation

invariance. The mismatches can be identified and removed by comparing the inconsistency between two coding maps.

LRGGC Inspired by GC [10], [13], Yang et al. [33] utilized the squared distance matrix as a coding map, which is robust to rotation and translation changes. Based on robust principal component analysis [34], the coding map can be decomposed into two parts. One is a low rank matrix representing the true matches, and the other is a sparse matrix representing the mismatches. The scale invariance can be naturally achieved based on the property of low-rankness.

L1GGC Lin et al. [35] proposed an accelerated version of LRGGC. In order to achieve scale invariance, after computing the two squared distance matrices, they solved a one-variable ℓ_1-norm error minimization problem by adopting the golden section search method. L1GGC is simpler than LRGGC [33].

PGM Compared with WGC, EWGC, and SGC, which focus on individual correspondences, Li et al. [36] considered the pairwise geometric relations between correspondences. By proposing a strategy to combine the geometric information from both the individual matches and pairs of correspondences, they further improved the verification accuracy without high computational cost.

Although the above four methods are efficient on filtering mismatches under similarity transformations when applied to partial-duplicate image search, they may fail when affine or perspective transformations exist, which is more common in retrieving photos taken for the same place at different views.

To conclude, RANSAC, MLESAC, ICF, and VFC are time consuming for partial-duplicate image search. WGC, EWGC, SGC, GC, and PGM need extra prior about the scales and dominant orientations of SIFT features, which limit their applications. And LRGGC and L1GGC cannot handle more complex geometric changes very well.

III. Our Approach

In the following subsections, we will introduce our SIM based mismatch removal method in four parts: the first part is about how to represent the features by SIM; the second part focuses on the proof about the affine invariance of our method; the third part discusses the geometric interpretation of SIM; and the last part introduces our mismatch detection approach.

A. Feature Spatial Representation by SIM

Note that our method can handle any dimensional point matching problems. Without loss of generality, we take the two-dimensional image features as an example. Given two matched feature point sets extracted from two images, let $\mathbf{X} \in \mathbb{R}^{3\times n}$ be one of the homogeneous coordinate of the two point sets ($\forall i,\ [\mathbf{X}]_{3,i} \equiv 1$). The Shape Interaction Matrix (SIM) [27] is defined as follows:

$$\mathbf{Z} = \mathbf{X}^{\dagger}\mathbf{X}, \tag{3}$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudo-inverse [37]. When $\mathbf{X}$ has linearly independent rows, $\mathbf{Z}$ can be computed as:

$$\mathbf{Z} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}. \tag{4}$$

When the number of points is relatively small, $\mathbf{Z}$ can be efficiently computed by using economic-size QR factorization:

$$\mathbf{X}^T = \mathbf{Q}\mathbf{R},\ \mathbf{Z} = \mathbf{Q}\mathbf{Q}^T, \tag{5}$$

where $\mathbf{Q} \in \mathbb{R}^{n\times 3}$ is a matrix with orthonormal columns, and $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is an upper triangular matrix.

Mathematically, the SIM is the orthogonal projection matrix that projects n dimensional vectors to the subspace spanned by the rows of $\mathbf{X}$ (i.e., $ran(\mathbf{X}^T)$).

B. Affine Transformation Invariance of SIM

Given the closed-form solution in (3), we can derive a theorem that the SIM remains invariant under affine transformations. The theorem and its proof are described as follows:

Theorem 1: Let $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^{3\times n}$ be two point sets with homogeneous coordinates ($\forall i, [\mathbf{X}]_{3,i} \equiv 1$). Assuming that there exists an affine transformation $\mathbf{T} \in \mathbb{R}^{3\times 3}$ such that

$$\mathbf{X}_2 = \mathbf{T}\mathbf{X}_1, \tag{6}$$

where $\mathbf{T}$ is of full rank, then the shape interaction matrix of $\mathbf{X}_1$ and $\mathbf{X}_2$ satisfies

$$\mathbf{Z}_1 = \mathbf{Z}_2. \tag{7}$$

Proof: For self completeness, we first introduce some properties of pseudo inverse [38].

Let $\mathbf{C} \in \mathbb{R}^{m\times t}$ and $\mathbf{D} \in \mathbb{R}^{t\times n}$ be two matrices of rank t and $\mathbf{B} = \mathbf{C}\mathbf{D}$. We have

$$\mathbf{B}^{\dagger} = \mathbf{D}^{\dagger}\mathbf{C}^{\dagger}. \tag{8}$$

Furthermore,

$$\mathbf{C}^{\dagger}\mathbf{C} = \mathbf{I},\ \mathbf{D}\mathbf{D}^{\dagger} = \mathbf{I}. \tag{9}$$

For the point sets $\mathbf{X}_1$, there always exists a decomposition $\mathbf{X}_1 = \mathbf{M}\mathbf{N}$, where $\mathbf{M} \in \mathbb{R}^{3\times r}$ and $\mathbf{N} \in \mathbb{R}^{r\times n}$ is of rank r. Then

$$\begin{aligned}\mathbf{Z}_1 &= {\mathbf{X}_1}^{\dagger}\mathbf{X}_1\\ &= (\mathbf{M}\mathbf{N})^{\dagger}\mathbf{M}\mathbf{N}\\ &= \mathbf{N}^{\dagger}\mathbf{M}^{\dagger}\mathbf{M}\mathbf{N}\\ &= \mathbf{N}^{\dagger}\mathbf{N},\end{aligned} \tag{10}$$

where the third equality is derived from (8) and the last equality is derived from (9).

As $\mathbf{T}$ is of full rank, $\mathbf{T}\mathbf{M}$ remains the same rank as $\mathbf{M}$. Following the same deduction of (10), we have

$$\begin{aligned}\mathbf{Z}_2 &= {\mathbf{X}_2}^{\dagger}\mathbf{X}_2\\ &= (\mathbf{T}\mathbf{X}_1)^{\dagger}\mathbf{T}\mathbf{X}_1\\ &= (\mathbf{T}\mathbf{M}\mathbf{N})^{\dagger}\mathbf{T}\mathbf{M}\mathbf{N}\\ &= \mathbf{N}^{\dagger}(\mathbf{T}\mathbf{M})^{\dagger}(\mathbf{T}\mathbf{M})\mathbf{N}\\ &= \mathbf{N}^{\dagger}\mathbf{N}.\end{aligned} \tag{11}$$

Thus (7) holds. □

Denote the two subspaces $S_1 = ran(\mathbf{X}_1^T)$ and $S_2 = ran(\mathbf{X}_2^T)$. Theorem 1 indicates that if all the correspondences are true matches under an affine transformation, the orthogonal projection matrices onto the subspace S_1 and S_2 (i.e., the SIMs $\mathbf{Z}_1$ and $\mathbf{Z}_2$) remain the same. Since the orthogonal projection matrix for a subspace is unique [39], the subspace S_1 and S_2 are also the same in this case.

C. Geometric Interpretation of SIM

The SIM has been widely used for subspace segmentation, especially on solving the low rank representation problem [40]:

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_*, \ \text{s.t.} \ \mathbf{X} = \mathbf{X}\mathbf{Z}, \tag{12}$$

where $\|\cdot\|_*$ denotes the nuclear norm (i.e., the sum of the singular values of a matrix, which is also the convex surrogate of the rank function), $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the data matrix, and $\mathbf{Z} \in \mathbb{R}^{n\times n}$ is the coefficient matrix. Liu et al. [41] prove that the unique solution to (12) is exactly the SIM:

$$\mathbf{Z} = \mathbf{X}^{\dagger}\mathbf{X}. \tag{13}$$

Moreover, according to the constraint of (12) and the fact that $\forall i, \ [\mathbf{X}]_{3,i} \equiv 1$, the sum of each column of $\mathbf{Z}$ satisfies:

$$\sum_i [\mathbf{Z}]_{i,j} = 1. \tag{14}$$

where $[\cdot]_{i,j}$ means the element in the i^{th} row and the j^{th} column.

According to the above analysis, each column of the SIM can be seen as the linear combination coefficients of the corresponding point when taking account of all the points in this set as linear bases. Unlike the SIM, Guo and Cao [26], [42] only involve the three vertexes of the neighboring triangle around the point as the linear bases to compute the linear combination coefficients, whose triangles are determined by Delaunay triangulation on some sampled seed points. By comparing the above two methods, we find out that both of the linear combination coefficients in the two methods are actually the **barycentric coordinates** of the data points obtained from the Cartesian coordinates. Specifically, [26], [42] computes the coefficients with respect to a triangle, which is also known as the *area coordinates*, while the coefficients of SIM can be regarded as the *normalized generalized barycentric coordinates*, which are determined with respect to a polytope.

D. Mismatch Detection Approach Based on SIM

According to the result in Subsection III-B, when all the matches are true, the SIMs and the two corresponding subspaces S_1 and S_2 remain the same. If there exist mismatches, they will be different from each other. Based on the one-to-one correspondence between the orthogonal projection matrix (i.e., the SIM) and their subspace, Golub and Loan [39] and Werman and Weinshall [21] define the distance between the two subspaces:

$$dist(S_1, S_2) = \|\mathbf{Z}_1 - \mathbf{Z}_2\|_F, \tag{15}$$

where $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are the orthogonal projection matrices (i.e., the SIMs) onto to S_1 and S_2, $\|\cdot\|_F$ is the Frobenius norm.

Based on the analysis in Subsection III-C, we further infer that if the two given corresponding points are true match, their barycentric coordinates (i.e., the corresponding column of $\mathbf{Z}$), should be close to each other, whose difference gives small contribution to $dist(S_1, S_2)$. On the contrary, the barycentric coordinates of two mismatched points cannot be very similar, whose difference gives major contribution to $dist(S_1, S_2)$.

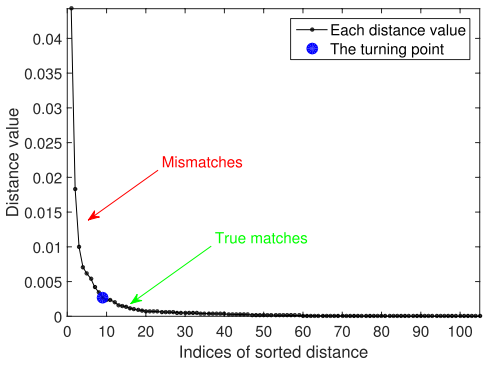


Fig. 3. Mismatch detection. The big dot indicates a "turning point" of the curve, which is the closest one to the origin. We take the distance value of this "turning point" (i.e., $\{\mathbf{d}_{sort}\}_{i_t}$) as a threshold to pick out the mismatches whose distance value is larger than it.

Therefore, we may detect the mismatches by finding the difference between the two barycentric coordinates of the matched points that account for the majority of $dist(S_1, S_2)$.

We adopt the Euclidean distance between the two barycentric coordinates as a similarity measure to detect mismatches:

$$d_i = \|[\mathbf{Z}_1]_{:,i} - [\mathbf{Z}_2]_{:,i}\|_2, \tag{16}$$

where d_i is the Euclidean distance between the i^{th} corresponding point pair, and $[\cdot]_{:,i}$ means the i^{th} column.

After computing Euclidean distances of all the matching pairs, we sort it in a descending order:

$$\mathbf{d}_{sort} = sort(\mathbf{d}), \tag{17}$$

where $\mathbf{d} = [d_1 \ d_2 \cdots d_n]$. Then the mismatches can be easily detected. As shown in Figure 3, there always exist a turning point in $\mathbf{d}_{sort}$ whose distance value can be set as a dynamic threshold to pick out the mismatches whose corresponding distances in (16) are larger than this threshold. Since the ideal turning point is usually very close to the origin, based on this observation, we determine the index of the turning point as follows:

$$i_t = \arg\min_i \sqrt{\left(\frac{i-1}{n}\right)^2 + \left(\frac{\{\mathbf{d}_{sort}\}_i - \{\mathbf{d}_{sort}\}_{min}}{\{\mathbf{d}_{sort}\}_{max} - \{\mathbf{d}_{sort}\}_{min}}\right)^2}, \tag{18}$$

where $i, j \in \{1, 2, 3, \cdots, n\}$ are the indices of the sorted distances, n is the total number of the matches, $\{\mathbf{d}_{sort}\}_{min} = \min_i\{\mathbf{d}_{sort}\}_i$, $\{\mathbf{d}_{sort}\}_{max} = \max_i\{\mathbf{d}_{sort}\}_i$, and i_t denotes the index of the turning point we find. The meaning of (18) is to find the turning point which is the closest to the origin of the coordinate system in Figure 3 under Euclidean distance. More specifically, $\left(\frac{i-1}{n}\right)$ means the normalized distance between the putative turning point and the origin on the x-axis in Figure 3, and $\left(\frac{\{\mathbf{d}_{sort}\}_i - \{\mathbf{d}_{sort}\}_{min}}{\{\mathbf{d}_{sort}\}_{max} - \{\mathbf{d}_{sort}\}_{min}}\right)$ means the normalized distance on the y-axis in Figure 3.

1) An Extra Step to Deal With Outliers: we design an extra step to refine the sorted Euclidean distance to weaken the effect of outliers. When the percentage of outliers or the level of outliers becomes large, the barycentric coordinates of the true matches will be disturbed by the large number

Algorithm 1 Process of Computing the Refined Sorted Euclidean Distance Between the Two Barycentric Coordinates

Input:
$\mathbf{Z}_1, \mathbf{Z}_2, \mathbb{S} = \{1, 2, 3, \cdots, n\}$
Initialization:
$\mathbf{E} \leftarrow (\mathbf{Z}_1 - \mathbf{Z}_2) \odot (\mathbf{Z}_1 - \mathbf{Z}_2)$;
$\mathbf{e}_i \leftarrow \sum_i [\mathbf{E}]_{:,i},\ i \in \mathbb{S}$;
for all $t \in \{1, 2, 3, \cdots, n\}$ **do**
  $\{\mathbf{d}'_{sort}\}_t \leftarrow \max_{i \in \mathbb{S}} \sqrt{\mathbf{e}_i}$
  $i_{max} \leftarrow \arg\max_{i \in \mathbb{S}} \sqrt{\mathbf{e}_i}$
  $\mathbf{e} \leftarrow \mathbf{e} - [\mathbf{E}]_{:,i_{max}}$
  $\mathbb{S} \leftarrow \{j | j \in \mathbb{S}, j \neq i_{max}\}$
end for
Output: $\mathbf{d}'_{sort}$

of mismatches. So that their corresponding Euclidean distances will be as large as the mismatches'. In this case, the curve of the sorted distance value $\mathbf{d}_{sort}$ will be flat, which is hard to determine the turning point. Obviously, we cannot distinguish the true matches from the false matches in such cases, unless we weaken the effect of mismatches on the above defined Euclidean distances. Based on this idea, our solution is to cut off the component of the mismatches from the above defined Euclidean distances. More specifically, the Euclidean distances could be decomposed by two components. One is d_{iT} from the true matches, and the other is d_{iM} from the mismatches:

$$\begin{aligned} d_i &= \sqrt{\|[\mathbf{Z}_1]_{\mathbb{J}_T,i} - [\mathbf{Z}_2]_{\mathbb{J}_T,i}\|_2^2 + \|[\mathbf{Z}_1]_{\mathbb{J}_M,i} - [\mathbf{Z}_2]_{\mathbb{J}_M,i}\|_2^2} \\ &= \sqrt{d_{iT}^2 + d_{iM}^2}, \end{aligned} \tag{19}$$

where $\mathbb{J}_T$ is the set of indices from the true matches, and $\mathbb{J}_M$ is the set of indices from the mismatches. Since the true $\mathbb{J}_T$ and $\mathbb{J}_M$ are unknown, for the i^{th} matches, its $\mathbb{J}_M$ are determined by those whose original Euclidean distances are larger than the current one, which is more likely to be the mismatches:

$$\mathbb{J}_M = \{j | d_j >= d_i\}, \tag{20}$$

and its $\mathbb{J}_T$ are determined as follows:

$$\mathbb{J}_T = \{j | d_j < d_i\}. \tag{21}$$

Following the above idea, to weaken the effect of outliers, we recompute a $\{\mathbf{d}'_{sort}\}_i$ by only considering $\{\mathbf{d}_{sort}\}_{iT}$, which is the true matches' components of $\{\mathbf{d}_{sort}\}_i$:

$$\begin{aligned} \{\mathbf{d}'_{sort}\}_i &= \{\mathbf{d}_{sort}\}_{iT} \\ &= \sqrt{\sum_{t \in \{j | \{\mathbf{d}_{sort}\}_j < \{\mathbf{d}_{sort}\}_i\}} \left([\mathbf{Z}_1]_{t,i} - [\mathbf{Z}_2]_{t,i}\right)^2}, \end{aligned} \tag{22}$$

where $[\cdot]_{t,i}$ means the element in the t^{th} row and the i^{th} column. We further design an algorithm to compute $\{\mathbf{d}'_{sort}\}$ with a complexity of $O(n)$, as described in Algorithm 1.

2) An Optional Step to Deal With Burstiness: when the strong burstiness phenomenon exists, we further design an optional step to completely remove 1-vs-N mismatches. Assume that there exist m corresponding point pairs that are 1-vs-N mismatches, and the true matches are unique among these mismatches. Since the barycentric coordinates of the true match should be close to each other, the index of the true match can be determined by finding the minimum distances:

$$i_r = \arg\min_i \{\mathbf{d}_{sort}\}_i, \tag{23}$$

where i_r is the index of the true match among the 1-vs-N matches. And all the other matches will be mismatches.

3) An Alternative Way for Real Time Image Search Tasks: we also design an even simpler and faster technique called cosine similarity based technique, which is more suitable for real time image search without large performance drop. The cosine similarity is defined as:

$$s_i = \frac{-[\mathbf{Z}_1]_{:,i}[\mathbf{Z}_2]_{:,i}}{\|[\mathbf{Z}_1]_{:,i}\|_2 \|[\mathbf{Z}_2]_{:,i}\|_2}. \tag{24}$$

Note that $s_i \in [-1, 1]$. If s_i is close to 1, the barycentric coordinates of the two matched points will be similar to each other, which indicates a true match between these two points. We simply choose a fixed threshold τ (the empirical value of τ is around 0.6), and pick out the mismatches whose $s_i < \tau$.

IV. Experiments

In this section, we compare our method with thirteen state-of-the-arts which also focus on removing mismatches from putative matches, including RANSAC, MLESAC, ICF, VFC, FastVFC, SparseVFC, WGC, EWGC, SGC, GC, LRGGC, L1GGC, and PGM. We first use both synthetic and real datasets to verify the validity and robustness of all the methods in removing mismatches, then we apply all the methods to image search to see how they improve the retrieval performance. All the experiments are tested on a server with an Intel Xeon E7-4820 CPU at 2.00GHz and with 64GB of memory, running Windows Server 2008 and Matlab version R2012a. For all the state-of-the-arts, we implement them based on the publicly available codes or algorithms. All the parameters are optimized and fixed during the whole experiments.

Precision and recall are commonly used measurement to evaluate the performance of all the methods on detecting mismatches. Precision is defined as the number of true positive matches divided by the number of all positive matches detected, and recall is defined as the number of true positive matches divided by the total number of true matches. To combine precision and recall with an equal weight, we adopt F-score to evaluate the performance [43] defined as the harmonic mean of precision and recall:

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{25}$$

A. Mismatch Removal on Synthetic Dataset

In this subsection, we compared our method with the state-of-the-arts on the synthetic data in four aspects: robustness to noises, robustness to outliers, robustness to burstiness, and affine invariance. Note that we cannot compare with WGC, EWGC, SGC, GC, and PGM on synthetic data, because they need scale and orientation of SIFT features as the input. We will compare with them later on real image datasets.

We totally generate 500 samples for testing. For each example, we first randomly generate a point set including 100 points within a range of 800×600 pixels. Second, we make a duplication of the first point set to simulate the true matches. Next, for different examples, we add Gaussian noises with different levels, outliers with different percentages, and multiple mappings with different burstiness degrees on the second point set. Then we transform the second point set with different types of affine transformations.

Now we analyze the performance on synthetic data.

1) Comparison at Different Levels of Noises: to test the robustness to noises, we add different levels of noise to the data. The standard deviation of Gaussian noises are set to be 1%, 3%, 5%, 7%, and 9% of the image size, respectively. As shown in Figure 5(a), our method always achieves the highest F-score among all the state-of-the-arts when the level of Gaussian noises increases. One can see that only our method can achieve an average F-score higher than 0.65 when the level of Gaussian noises is 9% of image size.

2) Comparison at Different Percentages of Outliers: to test the robustness to the number of mismatches, we set the percentage of outliers to be 5%, 25%, 45%, 65%, and 85%, respectively. As shown in Figure 5(b), our method achieves the highest F-score among all the methods at different percentages of mismatches. The performances of L1GGC and LRGGC drop quickly with the percentage of outliers increases. Please notice that when the percentage of outliers increases to 85%, only our method can keep an average F-score larger than 0.5.

3) Comparison at Different Burstiness Degrees: as we mentioned in the Introduction, the widely existing "burstiness" phenomenon, namely, the 1-vs-N mismatches, will greatly reduce the retrieval precision when applied to large scale image search. Therefore, we need to test the robustness of all the methods in filtering this kind of specific 1-vs-N mismatches. To this end, we must define a variable to define the difficulty of the tasks on 1-vs-N mismatches removal. For two extreme examples, if all the matches are 1-vs-1 matches (Figure 4(a)), it's relatively simple to solve, because no burstiness phenomenon occurs. On the other hand, if all the features in one set are fully matched with the other set (Figure 4(c)), the removal task becomes extremely difficult. Following the above examples, we define a novel variable called *"burstiness degree"*, which measures the difficulty of removing 1-vs-N mismatches by the percentage of 1-vs-N mismatches that occur:

$$\eta = \frac{p - q}{n_1 n_2 - \min(n_1, n_2)} \in [0, 1], \tag{26}$$

where p is the total number of putative matches, q is the total number of 1-vs-1 true matches in the putative matches, n_1 and n_2 are the number of points in the two sets. The numerator of (26) means the total number of 1-vs-N mismatches in the current putative matches, and the denominator of (26) means the maximum number of 1-vs-N mismatches that the current two point sets could possibly have, and finally η means the percentage of 1-vs-N mismatches that occur. An illustration of computing burstiness degree can be found in Figure 4.

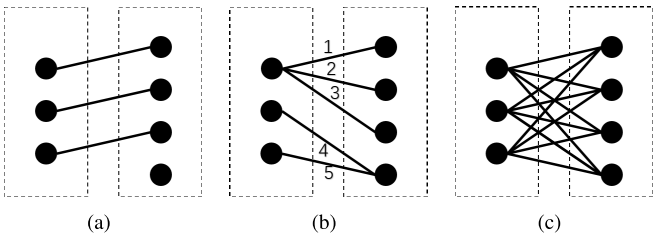


Fig. 4. An illustration of the burstiness degree of the correspondence between the two point sets. The burstiness degree of (a)-(c) is: $\eta_a = \frac{3-3}{12-3} = 0$, $\eta_b = \frac{5-2}{12-3} \approx 0.33$ (here the total number of possible 1-vs-1 true matches is 2, one from the matches No. 1-3, the other from the matches No. 4-5), and $\eta_c = \frac{12-3}{12-3} = 1$, respectively. With the burstiness degree increases from 0 to 1, the task of filtering the 1-vs-N mismatches changes from the easiest case to the most difficult one, which indicates that the burstiness degree measures the difficulty of 1-vs-N mismatches removal task.

To test the robustness to burstiness, we add different numbers of the 1-vs-N mismatches to the original 1-vs-1 true matches, the corresponding burstiness degrees are 1.68×10^{-4}, 3.37×10^{-4}, 5.05×10^{-4}, 6.73×10^{-4}, and 8.42×10^{-4}, respectively. As shown in Figure 5(c), our method achieves the highest F-score among most of the state-of-the-arts when the burstiness degree increases.

Besides, with increasing burstiness degree, the average F-scores of ICF and LRGGC go down and up. One possible reason is that, most of the methods are designed for the spatial consistency satisfied mismatch removal task, but not for the particular 1-vs-N mismatch removal. Especially for these two methods, when 1-vs-N mismatches exist, the set of feasible true correspondences under their own geometric prior is enlarged. In this case, the probability of finding the correspondence that is close to the ground truth is lower than that of the case when all the putative matches are 1-vs-1 matches. As a result, these two methods become unstable, and their average F-scores are likely to oscillate.

4) Comparison Under Different Affine Transformations: to make the test more challenging, we use three types of affine transformations to test the affine invariance of all the methods, including rotations, scale changes, and skew changes, respectively. The settings and results are specified as follows.

The rotation angle θ is set to be $\frac{\pi}{18}$, $\frac{\pi}{6}$, $\frac{5\pi}{18}$, $\frac{7\pi}{18}$, and $\frac{\pi}{2}$, respectively. As shown in Figure 5(d), the F-score we obtained is still higher than other methods under different degrees of rotation, while VFC achieves lower F-scores when the rotation angle increases to $\frac{\pi}{2}$.

The scale factor on x-axis and y-axis (sc_x and sc_y) are simultaneously set to be 0.56, 1.67, 2.78, 3.89, and 5.00, respectively. In Figure 5(e), the F-score of VFC and our method is the highest among all the methods under different degrees of scale change. On the other hand, MLESAC, RANSAC, and ICF are obviously not robust to scale changes.

The skew transformation matrix is defined as follows:

$$T_{skew} = \begin{bmatrix} 1 & sk_x & 0 \\ sk_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{27}$$

where sk_x and sk_y are simultaneously set to be 0.56, 1.67, 2.78, 3.89, and 5.00, respectively. Please notice that the robustness to skew changes play a key role in achieving

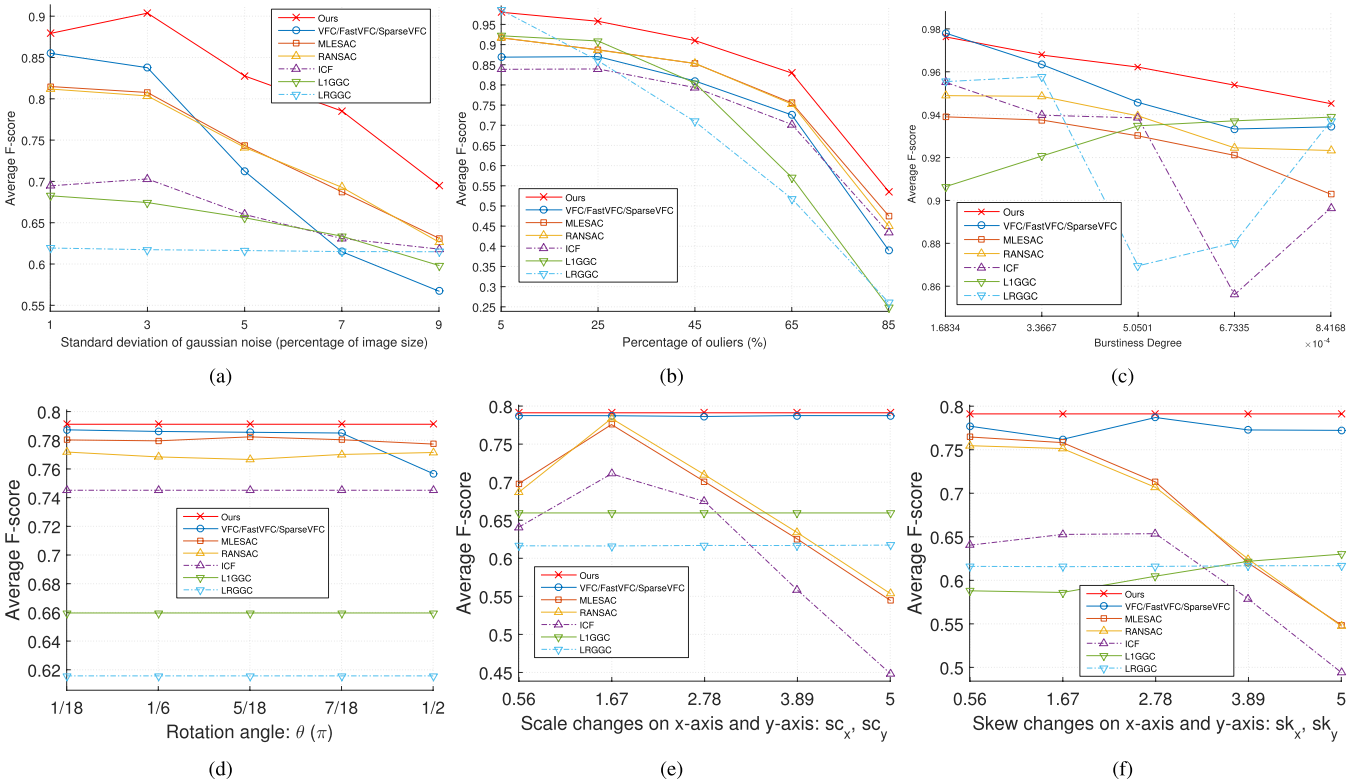


Fig. 5. The average F-score statistics of all the methods on synthetic dataset under different (a) standard deviations of Gaussian noises, (b) percentage of outliers, (c) burstiness degrees, (d) degrees of rotations, (e) degrees of scale changes, and (f) degrees of skew changes.

TABLE I

AVERAGE F-SCORE AND AVERAGE TIME COST COMPARISON OF ALL THE METHODS ON THE SYNTHETIC DATASET

Category	Method	Average F-score	Average time cost
Non-iterative	Ours	**88.42%**	**0.006s**
	Ours-Ori	70.70%	0.001s
	L1GGC	69.30%	0.019s
	LRGGC	66.73%	0.352s
Iterative	SparseVFC	85.43%	0.009s
	FastVFC	85.48%	0.029s
	VFC	85.08%	0.213s
	ICF	72.35%	0.204s
	MLESAC	78.54%	0.654s
	RANSAC	78.60%	0.654s

affine invariance. In Figure 5(f), our method is very robust and achieves the highest F-score under skew changes, while MLESAC, RANSAC, and ICF are sensitive to skew changes.

In summary, when the degree of affine changes increases, our method is very robust and achieves higher F-scores than other methods do, while all the state-of-the-arts are sensitive to some types of affine transformations.

5) Comparison on Time Efficiency: Table I is the comparison on the average time cost and the average F-score on synthetic dataset under all kinds of distortions. As shown in the table, our method achieves the best performance on detecting true matches with the lowest time cost.

Besides, in Table I, we give a comparison between our original Euclidean based method called "Ours-Ori" and its robust version named "Ours" which we used in the above six comparisons. Compared with "Ours-Ori", the average F-score of "Ours" is improved by 17.72% with approximately the same time cost, which can be attributed to its robustness in dealing with large percentage of outliers.

B. Mismatch Removal on Real Image Dataset

In this subsection, we test our method on a standard dataset named Mikolajczyk and Schmid [44] to evaluate the robustness and invariance of our method to several types of distortions. The distortions in this dataset include near-affine viewpoint changes, similarity transformations, blurs, lighting conditions, and JPEG compressions. There are totally eight groups in the dataset. Each group includes five image pairs with increasing levels of different types of distortions. The dataset also provides the ground truth perspective transformation matrix between each image pair to help determining the true feature matches.

We use ASIFT [16] as the feature detector and descriptor. It performs well when detecting features under large affine transformation distortions, while most of the state-of-the-arts fail (e.g., SIFT [14], Harris-Affine [45], Hessian-Affine [45], and MSER [46]). We use the executable code of ASIFT on Yu's website [47] to detect the features and obtain putative matches. All the input images are resized to 800×600 pixels to limit the time cost. We must mention that, by adopting ASIFT uniformly, all the state-of-the-arts and our method use exactly the same location and descriptor of each detected feature as input.

After detecting the features, the ground truth transformation matrix is used to compute the true matches. A match is regarded as true if only the reprojection error is withi 5 pixels.

TABLE II
AVERAGE F-SCORE AND TIME COST COMPARISON OF ALL THE METHODS ON THE MIKOLAJCZYK DATASET

Category	Method	Average F-score	Average time cost
Non-iterative	Ours	**91.38%**	0.147s
	Ours-Ori	90.46%	0.099s
	PGM	84.50%	1.290s
	L1GGC	87.58%	1.656s
	LRGGC	87.33%	2.515s
	GC	89.44%	1.901s
	SGC	79.27%	0.246s
	EWGC	76.17%	0.025s
	WGC	69.37%	**0.001s**
Iterative	SparseVFC	90.37%	0.076s
	FastVFC	90.38%	1.049s
	VFC	90.38%	5.510s
	ICF	86.38%	2.516s
	MLESAC	88.49%	1.878s
	RANSAC	87.24%	1.878s

The average percentage of the true matches is 85.86%. And the average number of matches is about 2600.

We compare our method with thirteen state-of-the-arts. Table II is the average F-score and average time cost comparison of all the methods. According to Table II, WGC and EWGC are the two fastest methods, but their performance is much inferior. The average F-scores of FastVFC, VFC, GC, MLESAC, L1GGC, LRGGC, and RANSAC are acceptable, but they are not very efficient. Our method is a good trade-off between time cost and performance. It is highly competitive with SparseVFC on speed, and obtains the highest F-score among all the methods.

Here we also provide our original Euclidean based method called "Ours-Ori" in Table II. Compared with "Ours-Ori", the average F-score of "Ours" is improved by 0.92% with approximately the same time cost, possibly due to its robustness to outliers.

C. Mismatch Removal for Partial Duplicated Image Search

In this subsection, to test the effectiveness and efficiency of our mismatch removal approach, we apply it to partial-duplicate image search as a geometric verification step. In detail, we utilize BoF [20] to obtain the coarse retrieval results, and the putative visual word matches are also available. After applying all the methods to remove the mismatches, the remaining true matches can help to refine the retrieval results.

We adopt three popular benchmark datasets for evaluation, including the GCDup dataset [11], the Holiday dataset [8], and the Oxford5k dataset [6]. The GCDup dataset has 1104 partial-duplicate images in 33 groups, which are collected from the Internet. Most of the images are man-made composite images, whose resolution is relatively low. The Holiday dataset are mainly personal photos taken on a large number of scenes (natural, man-made, water, fire effects, etc.). The images are in high resolution. It has 1491 near-duplicated images in 500 groups. The Oxford5k dataset consists of 5062 high resolution photos collected from Flickr by searching for some famous landmarks in Oxford. There are totally 55 groups in the dataset (eleven landmarks, each having five queries), which is manually annotated by the author.

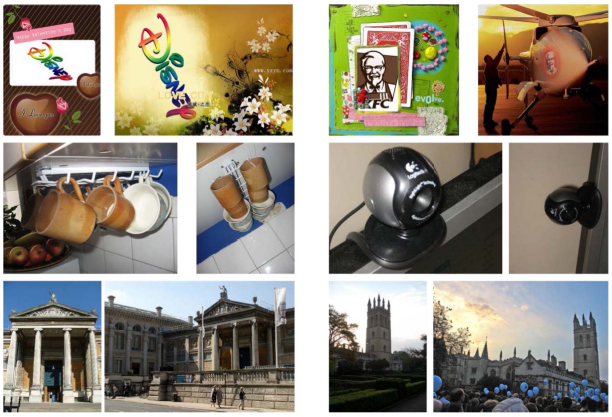


Fig. 6. Examples of the images and their relevant ones with strong affine (or perspective) transformations from the three benchmark datasets.

To make the experiment more challenging and realistic, we also use the MIRFlickr1M [48] dataset as a distractor dataset. It contains one million unrelated images from Flickr. It is often utilized as a distractor dataset. By adding increasing numbers of images from this dataset to the benchmark datasets, we can examine the robustness and scalability of a method.

The first five images of each group on the three benchmark datasets are used as the query images, and the expected retrieval results are all the remaining ones in the same group.

We use ASIFT [16] as the feature detector and descriptor. Here we must mention that for all the state-of-the-arts and our method, we utilize exactly that same location and descriptor of each detected feature as input by adopting ASIFT uniformly. After feature extraction, we train a codebook with one million visual words on the three benchmark datasets by using the hierarchical k-means clustering method [20]. With the trained codebook, we quantize each 128-dimension feature descriptor into a visual word. We follow the stop list technique from Sivic and Zisserman [19] to avoid frequent and uncommon visual words. The top 5% and bottom 10% visual words are stopped. So finally there are 850, 000 visual words remaining. All the features pairs that belong to the same visual word are determined as putative matches. Then, we applying all the mismatch removal approaches to filter the mismatches, and use the number of remaining true matches as the similarity measurement to re-rank the baseline retrieval results. For the re-ranking step, the re-ranking range is set to be the top 5000 images of the coarse retrieval results.

Mean average precision (mAP) [11] and average time cost are adopted to evaluate the accuracy and speed of all the methods. The average precision is defined as the area under the precision-recall curve:

$$AP = \sum_{i \in R} \frac{N_R / i}{N_A}, \tag{28}$$

where i means the i^{th} ranked images, R is the set of all the true relevant images, N_R is the total number of R, and N_A is the number of all the images. The mAP is computed as:

$$mAP = \sum_{q=1}^{N_Q} \frac{AP_q}{N_Q}, \tag{29}$$

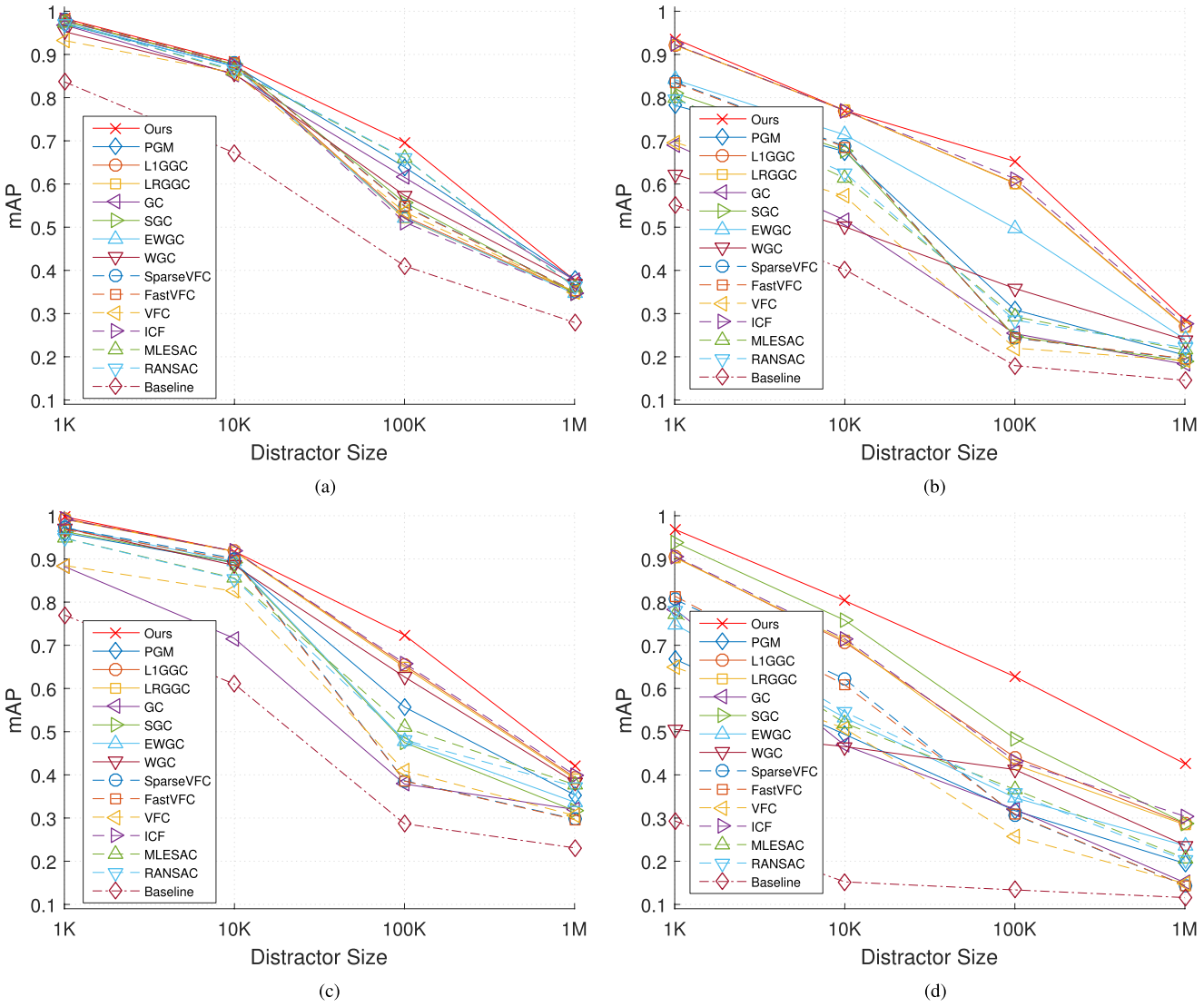


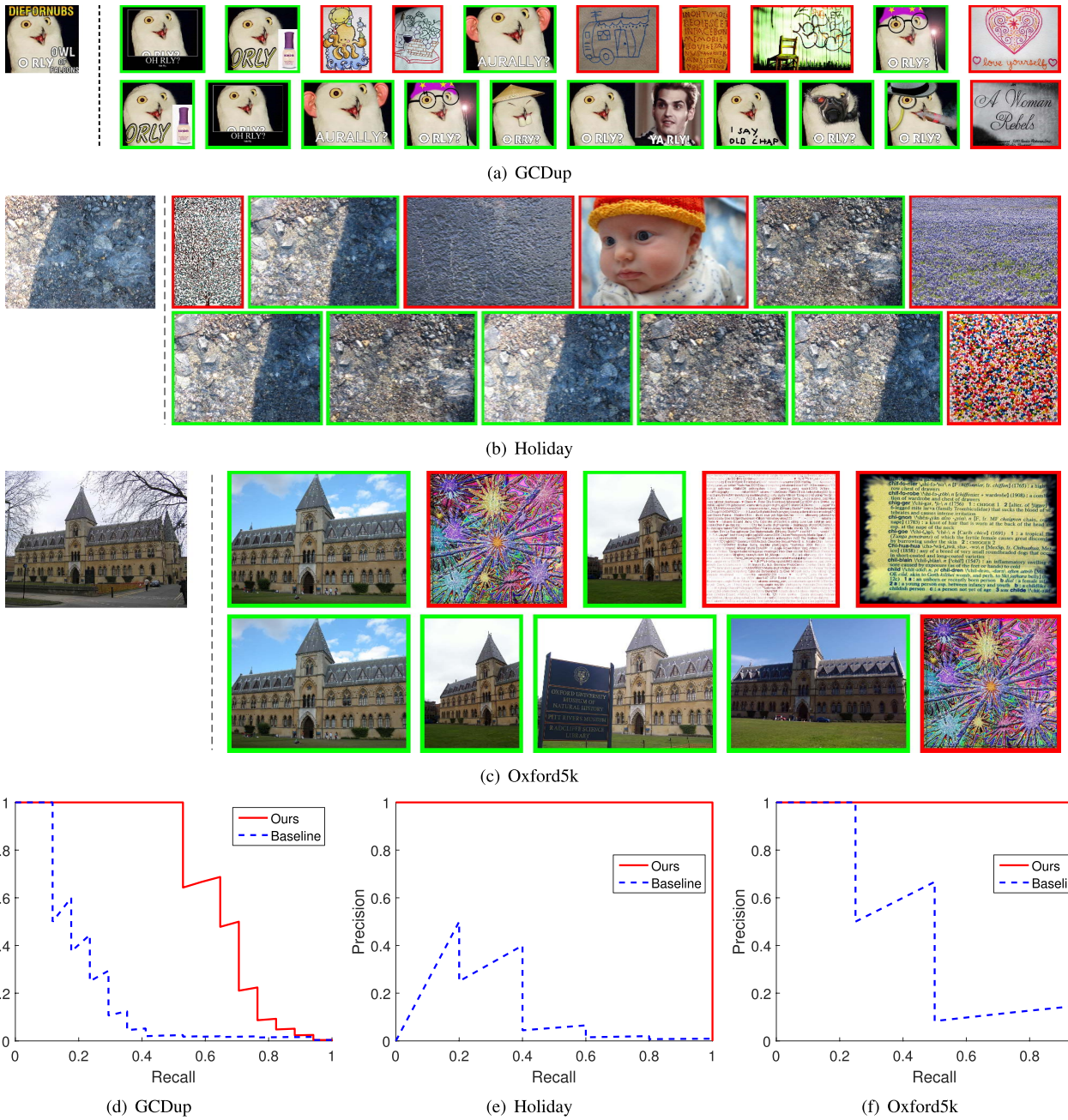
Fig. 7. The mAP benefit after applying all the geometric verification methods with different number of distractor images on the three benchmark datasets and sample images with strong affine or perspective transformations. The dash line indicates the iterative fitting methods, and the solid line represents the non-iterative filtering methods. Our method achieves higher mAP on most of the three datasets with different number of distractor images. Moreover, our method also achieves the highest mAP with large performance gap on the sample images with strong affine or perspective transformations. (a) GCDup dataset. (b) Holiday dataset. (c) Oxford5k dataset. (d) Sample images with strong affine or perspective transformations.

where N_Q is the number of queries, and AP_q is average precision of the q^{th} query.

As described in Subsection III-D, compared with cosine similarity based technique, our Euclidean based technique can achieve more accurate correspondence, which is more suitable for precise image matching tasks. However, for large scale real time image search tasks, the retrieval time cost is more important than the matching accuracy, because merely improving the matching accuracy between the query image and each retrieved image will not improve the retrieval mAP a lot, but costs more time. For this reason, we choose to use cosine similarity based technique for the following large scale image search experiments to make a trade-off between retrieval time and mean average precision.

Figure 7 shows the mAP benefits after using geometric verification methods under different numbers of distractor images. Compared with the baseline, it is obvious that the mAPs on the three benchmark datasets are improved greatly by applying mismatch removal approaches, as shown in Figure 7(a), 7(b), and 7(c). In particular, our method achieves the highest mAPs on most of the three datasets with different numbers of distractor images. When there exist strong affine (or perspective) transformations between the query image and its relevant ones, geometric verification becomes tougher. To compare the performance of all the methods on such a case, as shown in Figure 6, we take images from the three benchmark datasets that have strong affine or perspective transformations between them. The mAP results on the selected images are shown in Figure 7(d). Our method achieves higher mAP than other methods with a large performance gap.

In Figure 8, we provide some retrieval results on the three benchmark datasets with one million distractor images, our method greatly enhances the retrieval accuracy. On the GCDup dataset, our method improves AP by



(a) GCDup (b) Holiday (c) Oxford5k

(d) GCDup (e) Holiday (f) Oxford5k

Fig. 8. Sample retrieval results of **(a)** GCDup dataset, **(b)** Holiday dataset, and **(c)** Oxford5k dataset with one million distractor images. The query images are on the left of the dash line and the top-ranked images are on the right of the dash line. For each query, the first row is the original retrieval results of the baseline method, and the second row is the re-ranking results after applying our method. The green and red box indicate the relevant and irrelevant images, respectively. One can see that all the relevant images are ranked at the very top after applying our method on Holiday and Oxford5k. The Precision-Recall curves for the three queries are shown in **(d)**-**(e)**. Among the three examples, AP increase from 0.21 to 0.65 (+0.44) on GCDup, AP from 0.13 to 1 (+0.87) on Holiday, and AP from 0.46 to 1 (+0.54) on Oxford5k, respectively.

+0.44(from 0.21 to 0.65). And on Holiday and Oxford5k dataset, APs are both increased to 1, which means that all the relevant images are ranked at the top.

Table III is the average mAP and average time cost comparisons of all the methods on the three datasets with 1K distractors. Note that we only count the on-line query time, while all the common steps, such as feature extraction, codebook training, feature matching, and other off-line procedures, are not included. As shown in the table, our method is the third fastest method. Although WGC and EWGC are slightly faster than ours, they both achieve lower mAPs. Moreover, we give a comparison between our Euclidean distance based robust method called "Ours-Euc" and the cosine similarity based method named "Ours". Compared with "Ours-Euc", "Ours" achieves the same average mAP with lower time cost, which verifies that the cosine similarity based method is more suitable for large scale image search system.

TABLE III

THE AVERAGE MAP AND AVERAGE TIME COST COMPARISON OF ALL THE METHODS ON THE THREE DATASETS WITH 1K DISTRACTORS

Category	Method	Average mAP	Average time cost
Non-iterative	Ours	0.960	0.358s
	Ours-Euc	0.960	0.395s
	PGM	0.891	1.485s
	L1GGC	0.958	1.717s
	LRGGC	0.957	4.637s
	GC	0.841	5.353s
	SGC	0.962	3.467s
	EWGC	0.919	0.265s
	WGC	0.819	0.084s
Iterative	SparseVFC	0.920	1.105s
	FastVFC	0.919	4.380s
	VFC	0.828	5.306s
	ICF	0.957	10.96s
	MLESAC	0.903	2.119s
	RANSAC	0.900	2.013s

To sum up, our method is a good trade-off between effectiveness and efficiency on mismatch removal when applied to partial-duplicate image search as a geometric verification step.

V. CONCLUSIONS

We propose a novel mismatch removal method based on the shape interaction matrix (SIM). Given two corresponding point sets, we only use the coordinates of the feature points to compute the SIMs of the two point sets. The underlying SIM model provides a theoretical base to ensure the affine invariance of our method, the provided geometric interpretation further helps picking out the mismatches by finding the most inconsistent entries of the two SIMs. Compared with other state-of-the-arts, our method is simple, fast, and robust to affine changes, outliers, noises, and burstiness. Experiments on synthetic 2D points matching datasets and real images matching datasets show that our method is effective, efficient, and robust in mismatch removal. The experiment on partial-duplicate image search further verifies that our method, as a geometric verification step, gets better performance than state-of-the-arts on the three benchmark datasets with a large number of distractor images. In the future, we will target on improving our method to handle more complex transformations, such as perspective transformation, articulated motion, and non-rigid deformation.

REFERENCES

[1] R. Hartley and A. Zisserman, *Multiple View Geometry Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[2] S. Agarwal *et al.*, "Building Rome a day," *Commun. ACM*, vol. 62, no. 10, pp. 105–112, 2011.

[3] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image Vis. Comput.*, vol. 21, no. 13, pp. 1145–1153, Dec. 2003.

[4] J. Ma, J. Zhao, J. Tian, Z. Tu, and A. Yuille, "Robust estimation of nonrigid transformation for point set registration," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2147–2154.

[5] M. Yang, Y. Liu, Z. You, X. Li, and Y. Zhang, "A homography transform based higher-order MRF model for stereo matching," *Pattern Recognit. Lett.*, vol. 40, pp. 66–71, Apr. 2014.

[6] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[7] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.

[8] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.

[9] W.-L. Zhao, X. Wu, and C.-W. Ngo, "On the annotation of Web videos by efficient near-duplicate search," *IEEE Trans. Multimedia*, vol. 12, no. 5, pp. 448–461, Aug. 2010.

[10] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, "Spatial coding for large scale partial-duplicate Web image search," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 511–520.

[11] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Large scale image search with geometric coding," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1349–1352.

[12] J. Wang, J. Tang, and Y.-G. Jiang, "Strong geometrical consistency large scale partial-duplicate image search," in *Proc. ACM Int. Conf. Multimedia*, 2013, pp. 633–636.

[13] W. Zhou, H. Li, Y. Lu, and Q. Tian, "SIFT match verification by geometric coding for large-scale partial-duplicate Web image search," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, p. 4, 2013.

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

[15] 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.

[16] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 438–469, 2009.

[17] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 373–380.

[18] H. Jégou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1169–1176.

[19] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. Int. Conf. Comput. Vis.*, vol. 2l. Oct. 2003, pp. 1470–1477.

[20] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2006, pp. 2161–2168.

[21] M. Werman and D. Weinshall, "Similarity and affine invariant distances between 2D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 810–814, Aug. 1995.

[22] L. Zhang, W. Xu, and C. Chang, "Genetic algorithm for affine point pattern matching," *Pattern Recognit. Lett.*, vol. 24, nos. 1–3, pp. 9–19, 2003.

[23] E. Begelfor and M. Werman, "Affine invariance revisited," in *Proc. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2006, pp. 2087–2094.

[24] V. H. S. Ha and J. M. F. Moura, "Affine-permutation invariance of 2-D shapes," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1687–1700, Nov. 2005.

[25] J. Ho, M.-H. Yang, A. Rangarajan, and B. Vemuri, "A new affine registration algorithm for matching 2d point sets," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Feb. 2007, p. 25.

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

[27] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, 1998.

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

[29] 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.

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

[31] J. Zhao, J. Ma, J. Tian, J. Ma, and D. Zhang, "A robust method for vector field learning with application to mismatch removing," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 2977–2984.

[32] 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.

[33] L. Yang, Y. Lin, Z. Lin, and H. Zha, "Low rank global geometric consistency for partial-duplicate image search," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 3939–3944.

[34] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, May 2011.

[35] Y. Lin, C. Xu, L. Yang, Z. Lin, and H. Zha, "ℓ_1-norm global geometric consistency for partial-duplicate image retrieval," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 3033–3037.
[36] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5153–5161.
[37] R. Penrose, "A generalized inverse for matrices," in *Math. Proc. Cambridge Philos. Soc.*, vol. 51, no. 3, pp. 406–413, Jul. 1955.
[38] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," *Tech. Univ. Denmark*, vol. 7, p. 15, Nov. 2008.
[39] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, Maryland, USA: JHU Press, 2012, vol. 3.
[40] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 663–670.
[41] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[42] X. Guo and X. Cao, "Triangle-constraint for finding more good features," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1393–1396.
[43] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Newton, MA, USA: Butterworth, 1979.
[44] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[45] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, no. 1, pp. 43–72, 2005.
[46] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.
[47] G. Yu and J.-M. Morel, "ASIFT: An algorithm for fully affine invariant comparison," *Image Process. Line*, vol. 1, no. 1, pp. 1–28, Feb. 2011. [Online]. Available: http://dx.doi.org/10.5201/ipol.2011.my-asift
[48] M. Huiskes, B. Thomee, and S. Michael, "New trends and ideas visual concept detection: The MIR Flickr retrieval evaluation initiative," in *Proc. ACM Int. Conf. Multimedia Inf. Retr.*, 2010, pp. 527–536.

Yang Lin received the B.E. degree in telecommunications engineering from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, in 2012. He is currently pursuing the Ph.D. degree with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interest is computer vision and image processing.

Zhouchen Lin (M'00–SM'08) received the Ph.D. degree in applied mathematics from Peking University in 2000. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. He is also a Chair Professor with Northeast Normal University. His research areas include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an Area Chair of the CVPR 2014/2016, the ICCV 2015, and the NIPS 2015, and a Senior Program Committee Member of the AAAI 2016/2017 and the IJCAI 2016. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*. He is an IAPR Fellow.

Hongbin Zha (M'06) received the M.S. and Ph.D. degrees in electrical engineering from Kyushu University, Fukuoka, Japan, in 1987 and 1990, respectively. He joined Kyushu University in 1991 as an associate professor. He was a Research Associate with the Kyushu Institute of Technology. He was also a Visiting Professor with the Center for Vision, Speech, and Signal Processing, Surrey University, U.K., in 1999. Since 2000, he has been a Professor with the Key Laboratory of Machine Perception, Peking University, Beijing, China. He has authored more than 300 technical publications in journals, books, and international conference proceedings. His research interests include computer vision, digital geometry processing, and robotics. He received the Franklin V. Taylor Award from the IEEE Systems, Man, and Cybernetics Society in 1999. He is a member of the IEEE Computer Society.