# The Augmented Homogeneous Coordinates Matrix Based Projective Mismatch Removal for Partial-Duplicate Image Search

Yan Zheng, Zhouchen Lin, *Fellow, IEEE*

***Abstract*—Mismatch removal is a key step in many computer vision problems that involve point matching. The existing methods for checking geometric consistency mainly focus on similarity or affine transformations. In this paper, we propose a novel mismatch removal method that can cope with the projective transformation between two corresponding point sets. Our approach is based on the augmented homogeneous coordinates (AHC) matrix constructed from the coordinates of anchor matches, whose degeneracy can indicate the correctness of anchor matches. The set of anchor matches is initially all the matches and is iteratively updated by calculating the difference between the estimated matched points, which can be easily computed in closed form, and the actually matched points and removing those with large differences. Experimental results on synthetic 2D point matching data sets and real image matching data sets verify that our method achieves the highest F-score among all methods under similarity, affine and projective transformations with noises and outliers. Our method can also achieve faster speed than all other iterative method. Those non-iterative methods with slight advantage in speed are not competitive in accuracy when compared with ours. We also show that the set of anchor matches is stable through the iteration and the computation time grows very slowly with respect to the number of matched points. When applied to mismatch removal in partial-duplicate image search, our method achieves the best retrieval precision and its computing time is also highly competitive.**

***Index Terms*—point matching, image retrieval, partial-duplicate image search, mismatch removal, projective transformation, augmented homogeneous coordinates matrix.**

## I. Introduction

PARTIAL duplicate image search is an important problem in computer vision. Given a query image, the goal is to search target images in a large database, such as web images at a billion scale. The target images may contain duplicated parts, which may be part of the query image after scaling, rotation, translation, skewing and projective deformation. The technology has many applications, such as image registration [1], copyright infringement detection [2], [3], security surveillance and redundant image filtering [4], [5]. Nevertheless, it is a difficult problem. On the one hand, there are a large amount of images distracting search for target images. On the other hand, the duplicated part may not be exactly the same as that in the query image because it usually undergoes some transformations, e.g., due to view change or in order to counteract the detection.

Image-based search usually relies on the bag-of-features model (BOF) [6] in computer vision, which treats image features as words. Based on the feature points extracted from corresponding images (e.g., ASIFT [7], SIFT [8], and SURF [9] apply to two dimensional images, and MeshDoG [10] apply to three dimensional depth surfaces), the descriptor consistency is usually utilized to obtain enough putative matches. Namely, the feature points are matched if their descriptors are similar. However, reliable correspondence cannot be ensured only by the descriptor consistency because the descriptors only encode local information in an imperfect way. So the retrieval result offered by the BOF model is unsatisfactory due to the presence of a lot of mismatches.

Hence it is necessary to use another constraint, geometric consistency, that putative matches should satisfy, to refine the coarse matching result. Namely, the geometric relationship among the points in one image should be preserved in another image. The points that break the geometric consistency are considered as mismatches. This is also termed as geometric verification in computer vision. As the total number of matches is commonly used as a measurement for re-ranking the retrieval results, mismatch removal is of great importance to improve the retrieved results in large scale image search.

Therefore, how to efficiently verify the geometric consistency among the matched points is a key problem. As we have mentioned above, one of the main difficulties is handling the complex transformation between the duplicated parts. The difficulties also lie in the error of the feature location caused by the detector and a large percentage of unmatched features caused by occlusions or the limitation of detector, we call *noises* and *outliers*, respectively, in the sequel. Hence, a good mismatch removal algorithm should establish geometric correspondence between two point sets containing noises and outliers under complex geometric transformations, in order to detect as many mismatches as possible.

In the literature, the commonly considered geometric transformation include similarity transformation and affine transformation, where the similarity transformation involves trans-

Yan Zheng is with School of Mathematics and System Science, Beihang University, P. R. China. Email: yan.zheng.mat@gmail.com.

Zhouchen Lin is with Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, P. R. China. He is also with Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, P. R. China. Zhouchen Lin is the corresponding author, Email: zlin@pku.edu.cn.

We are very grateful to Jianlong Wu for valuable suggestions in our revision.

Zhouchen Lin is supported by National Basic Research Program of China (973 Program) (grant no. 2015CB352502), National Natural Science Foundation (NSF) of China (grant nos. 61625301 and 61731018), Qualcomm and Microsoft Research Asia.

lation, rotation and scaling:

$$\begin{bmatrix} \widehat{x}_i \\ \widehat{y}_i \end{bmatrix} = s \cdot \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \quad (1)$$

where $[\widehat{x}_i, \widehat{y}_i]^T$ and $[x_i, y_i]^T$ represent the coordinates of the $i^{th}$ corresponding feature points in two images, and $s$, $\theta$ and $[t_x, t_y]^T$ are the scaling factor, rotation angle and translation vector, respectively. The projective transformation is much less studied due to its difficulty. Accordingly, we can divide all the mismatch removal methods into three categories, i.e., methods for similarity transformation [3], [11], [12], [13], affine transformation [14], [15], [16], [17], [18] and projective transformation [19], [20], [21], respectively. Much work has been done on similarity and affine matching. Unfortunately, projective transformation is very common in large scale image databases. So the performance of these methods drops sharply when projective transformation exsits. Although there have been some iterative methods that can handle projective transformation [19], [20], [21], their time costs for robustly estimating the transformation are relatively high.

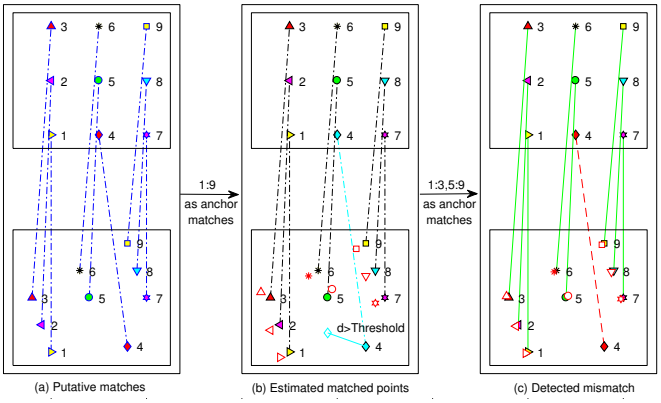


Fig. 1. Illustration of our AHC matrix based approach. (a) Putative matches given by the BOF matching between two images that undergo a projective transformation. Construct AHC matrices based on anchor matches, which is initially all the putative matches. (b) In the iteration, the distance between the fourth point in the target image, denoted by the blue diamond, and the estimated matched point, denoted by the light blue diamond, is greater than a threshold. Thus the fourth match will not be considered as an anchor match in the next iteration. (c) The procedure of (a) and (b) iterate until convergence. In this example, only the fourth match is detected as a mismatch, represented by the red dashed line.

### A. Our Contributions

In this paper, we propose a novel mismatch removal algorithm for point sets undergoing projective transformation. Besides high detection accuracy, our method is much faster than other ones for projective transformation. Actually, it is almost the fastest even if it is compared with methods for similarity transformation only.

Our method is based on a new matrix called the augmented homogeneous coordinates (AHC) matrix, constructed from the coordinates of some anchor matches which we temporarily assume to be correct matches. The rank of the AHC matrix can be an indicator of whether there exist mismatches in the chosen anchor matches. The degeneracy of the AHC matrix can be judged by the determinant of the AHC matrix multiplied with its transpose. Then given some anchor matches, we can utilize the determinant to estimate the coordinates of points in the target image that match those in the query image, which we call the *estimated matched points* and have closed-form solutions. Thus we can find corresponding points in the target image without explicitly recovering the projective transformation, which is in stark contrast to RANSAC based methods [19], [20]. Then the new anchor matches are chosen as those having relatively small distances between the estimated matched points and the actually matches point, which we call the *reprojection error*. The initial set of anchor matches is simply all the putative matches and the iteration terminates when the reprojection errors of the anchor matches are all below a threshold, namely the quality of anchor matches is good enough. We illustrate the iterative process in Figure 1.

In summary, the main contributions of our paper are as follows:

- We propose the augmented homogeneous coordinates (AHC) matrix for projective mismatch removal. Its rank indicates whether there exist mismatches in the given anchor matches. The AHC matrices only utilize the coordinates of features, making our method simple and general. In comparison, many of traditional geometric verification methods require extra spatial prior (e.g., scale and orientation of SIFT features [8]).
- Based on the AHC matrices, we provide closed-form solutions to estimate the coordinates of matched points in the target image. Thus we need not recover the underlying projective transformation.
- We choose new anchor matches by finding those with relatively small reprojection errors. We show that the set of anchor matches is stable through the iteration and our AHC matrix based mismatch removal algorithm has better performance on robustness in most cases than least squares based approaches, such as RANSAC based ones [19], [20].
- We compare our approach with the state-of-the-art geometric verification methods on both synthetic and real datasets in the tasks of mismatch removal and partial-duplicate image search. Experimental results show that our approach achieves the highest accuracy and almost the fastest speed.

## II. Related Work

We review the recent development of geometric consistency based mismatch removal methods. We classify these methods into three categories. Namely, methods for similarity transformation, affine transformation and projective transformation, respectively.

### A. Methods for Similarity Transformation

Detecting mismatches under similarity transformation (see (1)) is relatively easy. Thus a great many methods have been proposed to efficiently verify the geometric consistency maintained by similarity transformation. Representative methods include:

*Weak Geometric Consistency (WGC)* Jegou et al. [11] assumed that matched features should be consistent in orientation and scale. So they collected the differences between the orientations and the scales of the features in the two images, forming two histograms. The peak values of the two histograms indicate the most possible differences in orientation and scale, which can be used to determine the consistent matches.

*Enhanced Weak Geometric Consistency (EWGC)* Unlike [11], Zhao et al. [3] built a histogram of the $\ell_2$ norm of the translation vector derived from Eq. (1) then found its peak value. Such a one dimensional value was efficient for finding matched feature pairs but inevitably loses a lot of geometrical information.

*Strong Geometric Consistency (SGC)* On the basis of WGC and EWGC, Wang et al. [12] first grouped the matches based on their rotation angles. Then in each group, they utilized the histogram peak of the translation vector as the dominant translation of the group. After removing matched feature pairs with significant differences from the dominant translation in each group, the number of remaining pairs of the largest group was used to measure the similarity between two images.

*Pairwise Geometric Matching (PGM)* Li et al. [13] used global scaling and rotation relationship to enforce the local geometric consistency derived from the coordinates of matches. Mapping the coordinates of points to pairwise rotation and scaling made the method more tolerant to noises. Also, using filtering steps reduced the number of matches, leading to relatively high image matching reliability without high computational cost.

With scales and dominant orientations in SIFT, the above four approaches try to estimate a similarity transformation from putative matches. When the transformation between two images is affine or projective, they will be ineffective.

### B. Methods for Affine Transformation

*Geometric Coding (GC)* Zhou et al. [14], [15] proposed a spatial coding to encode relative spatial locations among features, which could encode rotation changes and discover false feature matches effectively.

*Low Rank Global Geometric Consistency (LRGGC)* Inspired by GC [14], [15], Yang et al. [16] modeled the global geometric consistency with a low rank matrix, then formulated the problem of detecting false matches as a problem of decomposing the stacked squared distance matrices into a low rank matrix representing the true matches and a sparse matrix representing the mismatches, which can be solved by the Alternating Direction Method efficiently.

*L1-norm Global Geometric Consistency (L1GGC)* Lin et al. [17] first formed the squared distance matrices from all the matched feature points, which is similarity invariant. Then L1GGC solves a one-variable $\ell_1$-norm error minimization problem by adopting the Golden Section Search method.

*Identifying point correspondences by Correspondence Function (ICF)* Li and Hu [22] proposed an iterative algorithm based on a diagnostic technique and SVM to learn correspondence functions that mutually map one point set to the other. Then mismatches are identified by checking whether they are consistent with the estimated correspondence functions.

*Shape Interaction Matrix-based affine invariant (SIM)* Lin et al. [18] proposed a non-iterative mismatch removal method that achieves affine invariance by computing the shape interaction matrices of two corresponding point sets. The method detects the mismatches by picking out the most different entries between the two shape interaction matrices.

The above five methods are efficient in filtering mismatches under similarity or affine transformations. However, they fail when severe projective transformation exists, which is quite common when retrieving natural images.

### C. Methods for Projective Transformation

Handling projective transformation is much more difficult than similarity transformation and affine transformation due to its nonlinearity. Thus work on this aspect is relatively sparse.

*Random Sample Consensus (RANSAC)* As a classic method, RANSAC [19] and its variants (e.g., MLESAC [20]) tried to estimate the projective transformation between two images directly. They repeatedly picked a random subset of the whole matches to estimate a projective transformation. The trial procedure was repeated for a fixed number of times or is terminated when the correspondence error was below a threshold, each time producing either a transformation which is rejected if too few points are classified as inliers or a refined transformation with a lower correspondence error.

*Vector Field Consensus (VFC)* Ma et al. [21] proposed a vector field learning method, which learns an interpolated vector motion field fitting the putative matches based on the Tiknonov regularization in a vector-valued reproducing kernel Hilbert space. Meanwhile, true matches were estimeted by the EM algorithm. *Sparse Vector Field Consensus (SparseVFC)* [21] was an improved version of VFC with higher speed but no performance degradation.

*Robust Point Matching (RPM)* Wang et al. [23] introduced and improved a robust registration framework based on partial intensity invariant feature descriptor, which performed well even when confronted a large number of outliers in the correspondence set.

*Non-Rigid Point Set Registration with Robust Transformation Estimation under Manifold Regularization (MR-RPM)* Ma et al. [24] proposed a robust transformation estimation method based on manifold regularization for non-rigid point set registration, which iteratively recovers the point correspondence and estimates the spatial transformation between two point sets.

*Locality Preserving Matching (LPM)* Ma et al. [25] proposed a locality preserving matching method in 2017, the principle of which is to maintain the local neighborhood structures of the potential true matches. They formulated the problem into a mathematical model and derived a closed-form solution with linearithmic time and linear space complexities.

RANSAC, MLESAC, VFC and SparseVFC can handle projective transformation even when a large percentage of outliers exist. However, RANSAC and MLESAC are based on randomness and usually need a lot of trials in order to

TABLE I
MAIN NOTATIONS USED IN THE PAPER.

| Notations | Meanings |
|---|---|
| Bold capital | a matrix. Especially, $\mathbf{I}_{(n)}$ is the $n\times n$ identity matrix |
| Bold lowercase | a vector |
| $Q$ | the query image |
| $\widetilde{Q}$ | the target image matched to the query image $Q$ |
| $n$ | the number of matches given by *BOF* matching |
| $k$ | the number of anchor matches |
| $i$ | the index of a feature point in image $Q$ |
| $t$ | the iteration number |
| $\odot$ | the Hadamard product of two matrices or vectors |
| $\mathbf{1}$ | the all-one vector |
| $\mathbf{u}_i=[x_i,y_i,1]^T$ | the homogeneous coordinate of the $i$-th point in $Q$ |
| $\widetilde{\mathbf{u}}_i=[\widetilde{x}_i,\widetilde{y}_i,1]^T$ | the homogeneous coordinate of the $i$-th point in $\widetilde{Q}$ matched to $\mathbf{u}_i=[x_i,y_i,1]^T$ in $Q$ |
| $\overline{\mathbf{u}}_i=[\overline{x}_i,\overline{y}_i,1]^T$ | the homogeneous coordinate of the estimated point in $\tilde{Q}$ that is matched to $[x_i,y_i,1]^T$ |
| $(\overline{x}_i^{(t)},\overline{y}_i^{(t)},1)^T$ | the homogeneous coordinate of the $i$-th estimated matched point computed in the $t$-th iteration |
| $\mathbf{x},\ \mathbf{y}$ | $\mathbf{x}=[x_1,x_2,\cdots,x_k]^T,\ \mathbf{y}=[y_1,y_2,\cdots,y_k]^T$ |
| $\widetilde{\mathbf{x}},\ \widetilde{\mathbf{y}}$ | $\widetilde{\mathbf{x}}=[\widetilde{x}_1,\widetilde{x}_2,\cdots,\widetilde{x}_k]^T,\ \widetilde{\mathbf{y}}=[\widetilde{y}_1,\widetilde{y}_2,\cdots,\widetilde{y}_k]^T$ |
| $\overline{\mathbf{x}}^{(t)},\ \overline{\mathbf{y}}^{(t)}$ | $\overline{\mathbf{x}}^{(t)}=[\overline{x}_1^{(t)},\overline{x}_2^{(t)},\cdots,\overline{x}_n^{(t)}]^T,\ \overline{\mathbf{y}}^{(t)}=[\overline{y}_1^{(t)},\overline{y}_2^{(t)},\cdots,\overline{y}_n^{(t)}]^T$ |
| $\mathbf{U},\widetilde{\mathbf{U}},\overline{\mathbf{U}}$ | $\mathbf{U}=[\mathbf{u}_1,\cdots,\mathbf{u}_n],\ \widetilde{\mathbf{U}}=[\widetilde{\mathbf{u}}_1,\cdots,\widetilde{\mathbf{u}}_n],\ \overline{\mathbf{U}}=[\overline{\mathbf{u}}_1,\cdots,\overline{\mathbf{u}}_n]$ |
| $\mathbf{P}=(P_{ij})$ | the projective transformation matrix from $Q$ to $\widetilde{Q}$ |
| $\mathbf{H}^x,\ \mathbf{H}^y$ | $x$ and $y$ augmented homogeneous coordinate matrices |
| $\mathbf{h}_i^x(\alpha),\ \mathbf{h}_i^y(\beta)$ | $\mathbf{h}_i^x(\alpha)=[\alpha\cdot x_i,\alpha\cdot y_i,\alpha,x_i,y_i,1]^T$, $\mathbf{h}_i^y(\beta)=[\beta\cdot x_i,\beta\cdot y_i,\beta,x_i,y_i,1]^T$ |
| $\mathbf{H}_i^x(\alpha),\ \mathbf{H}_i^y(\beta)$ | $\mathbf{H}_i^x(\alpha)=[\mathbf{h}_i^x(\alpha),\mathbf{H}^x],\ \mathbf{H}_i^y(\beta)=[\mathbf{h}_i^y(\beta),\mathbf{H}^y]$ |
| $\mathbf{V}^x,\ \mathbf{V}^y$ | the square matrices such that $\mathbf{V}^x(\mathbf{H}^x\mathbf{H}^{x,T})\mathbf{V}^{x,T}=\mathbf{I}_{(6)}$, $\mathbf{V}^y(\mathbf{H}^y\mathbf{H}^{y,T})\mathbf{V}^{y,T}=\mathbf{I}_{(6)}$ |
| $\mathscr{I}^{(t)}$ | the index set of anchor matches in the $t$-th iteration |
| $\mu_x^{(t)},\sigma_x^{(t)}$, $\mu_y^{(t)},\sigma_y^{(t)}$ | the mean values and the standard deviations of $\left\{\widetilde{x}_i-\overline{x}_i^{(t)}\right\}$ and $\left\{\widetilde{y}_i-\overline{y}_i^{(t)}\right\}(i\in\mathscr{I}^{(t)})$, respectively |

have satisfactory results. Thus RANSAC and MLESAC are quite time-consuming. VFC and SparseVFC are somewhat sensitive to noises and their performances are unsatisfactory when the initial number of inliers is relatively small. LPM can accomplish the mismatch removal from thousands of putative correspondences in a few milliseconds, but it is sensitive to noises and large percentage of outliers. RPM and MR-RPM are for non-rigid matching and are quite time-consuming, unsuitable for image retrieval.

## III. OUR APPROACH

In this section, we introduce our mismatch removal algorithm in five parts. We first introduce the augmented homogeneous coordinates (AHC) matrix, which lays the foundation of our method, and illustrate the intuition about this matrix. Then we deduce how to calculate the estimated matched points in a robust and efficient way. Next, we present the iterative procedure for choosing anchor matches. Finally, we test the robustness of our approach. The major notations used in the paper are listed in Table I.

### A. Modeling Global Geometric Consistency by the AHC Matrices

In principle, our method can be generalized to any dimension of projective matching problems. Here we only take the two-dimensional point matching as an example. Given a set of putatively matched pairs, let $\mathbf{u}_i=[x_i,y_i,1]^T\in\mathbb{R}^3,\ i\in\{1,2,\cdots,n\}$ be the feature points extracted from the query image $Q$, and their matched ones $\widetilde{\mathbf{u}}_i=[\widetilde{x}_i,\widetilde{y}_i,1]^T\in\mathbb{R}^3, i\in\{1,2,\cdots,n\}$ in the target image $\widetilde{Q}$. For notational simplicity, we assume that the first $k$ pairs of points are correct matches used for inference and we call them anchor matches. Initially, we set $k=n$. When the iteration goes on, $k$ gradually decreases. Although the first $k$ pairs of points are correct matches, there may still be noises in the coordinates of the points due to the imperfection of feature point detctor and numerical qantization.

To make the mathematical deduction simple, we first assume that there are no noises in correct matches.

**Definition 1** (Augmented Homogeneous Coordinates (AHC) Matrix)**.**

$$\mathbf{H}^x=\begin{bmatrix}\widetilde{\mathbf{x}}^T\odot\mathbf{x}^T\\ \widetilde{\mathbf{x}}^T\odot\mathbf{y}^T\\ \widetilde{\mathbf{x}}^T\odot\mathbf{1}^T\\ \mathbf{x}^T\\ \mathbf{y}^T\\ \mathbf{1}^T\end{bmatrix}\in\mathbb{R}^{6\times k}\ \textit{and}\ \ \mathbf{H}^y=\begin{bmatrix}\widetilde{\mathbf{y}}^T\odot\mathbf{x}^T\\ \widetilde{\mathbf{y}}^T\odot\mathbf{y}^T\\ \widetilde{\mathbf{y}}^T\odot\mathbf{1}^T\\ \mathbf{x}^T\\ \mathbf{y}^T\\ \mathbf{1}^T\end{bmatrix}\in\mathbb{R}^{6\times k},$$

*are called the x and y Augmented Homogeneous Coordinates (AHC) Matrices, respectively.*

**Theorem 1.** *If there exists a projective transformation between the two point sets* $\left\{\mathbf{u}_i=[x_i,y_i,1]^T\right\}_{i=1}^k$ *and* $\left\{\widetilde{\mathbf{u}}_i=[\widetilde{x}_i,\widetilde{y}_i,1]^T\right\}_{i=1}^k$, *then* $rank(\mathbf{H}^x)\leqslant 5$ *and* $rank(\mathbf{H}^y)\leqslant 5$.

**Proof.** *Assume that the projective transformation matrix is* $\mathbf{P}=(P_{ij})\in\mathbb{R}^{3\times 3}$. *Then*

$$(P_{31}x_i+P_{32}y_i+P_{33})\begin{bmatrix}\widetilde{x}_i\\ \widetilde{y}_i\\ 1\end{bmatrix}=(P_{31}x_i+P_{32}y_i+P_{33})\begin{bmatrix}\dfrac{P_{11}x_i+P_{12}y_i+P_{13}}{P_{31}x_i+P_{32}y_i+P_{33}}\\ \dfrac{P_{21}x_i+P_{22}y_i+P_{23}}{P_{31}x_i+P_{32}y_i+P_{33}}\\ \dfrac{P_{31}x_i+P_{32}y_i+P_{33}}{P_{31}x_i+P_{32}y_i+P_{33}}\end{bmatrix}$$
$$=\begin{bmatrix}P_{11}&P_{12}&P_{13}\\ P_{21}&P_{22}&P_{23}\\ P_{31}&P_{32}&P_{33}\end{bmatrix}\begin{bmatrix}x_i\\ y_i\\ 1\end{bmatrix},\quad i=1,\cdots,k.$$

*Writing the above k equations in a matrix form, we have*

$$\begin{bmatrix}P_{31}\cdot\widetilde{\mathbf{x}}^T\odot\mathbf{x}^T+P_{32}\cdot\widetilde{\mathbf{x}}^T\odot\mathbf{y}^T+P_{33}\cdot\widetilde{\mathbf{x}}^T\odot\mathbf{1}^T\\ P_{31}\cdot\widetilde{\mathbf{y}}^T\odot\mathbf{x}^T+P_{32}\cdot\widetilde{\mathbf{y}}^T\odot\mathbf{y}^T+P_{33}\cdot\widetilde{\mathbf{y}}^T\odot\mathbf{1}^T\\ P_{31}\cdot\mathbf{x}^T+P_{32}\cdot\mathbf{y}^T+P_{33}\cdot\mathbf{1}^T\end{bmatrix}$$
$$=\begin{bmatrix}P_{11}\cdot\mathbf{x}^T+P_{12}\cdot\mathbf{y}^T+P_{13}\cdot\mathbf{1}^T\\ P_{21}\cdot\mathbf{x}^T+P_{22}\cdot\mathbf{y}^T+P_{23}\cdot\mathbf{1}^T\\ P_{31}\cdot\mathbf{x}^T+P_{32}\cdot\mathbf{y}^T+P_{33}\cdot\mathbf{1}^T\end{bmatrix}.$$

*Comparing the first two rows gives*

$$P_{31}\cdot\widetilde{\mathbf{x}}^T\odot\mathbf{x}^T+P_{32}\cdot\widetilde{\mathbf{x}}^T\odot\mathbf{y}^T+P_{33}\cdot\widetilde{\mathbf{x}}^T\odot\mathbf{1}^T=P_{11}\cdot\mathbf{x}^T+P_{12}\cdot\mathbf{y}^T+P_{13}\cdot\mathbf{1}^T,$$
$$P_{31}\cdot\widetilde{\mathbf{y}}^T\odot\mathbf{y}^T+P_{32}\cdot\widetilde{\mathbf{y}}^T\odot\mathbf{y}^T+P_{33}\cdot\widetilde{\mathbf{y}}^T\odot\mathbf{1}^T=P_{21}\cdot\mathbf{x}^T+P_{22}\cdot\mathbf{y}^T+P_{23}\cdot\mathbf{1}^T,$$

*which show that the rows of* $\mathbf{H}^x$ *and* $\mathbf{H}^y$ *are linearly dependent. So* $rank(\mathbf{H}^x)\leqslant 5$ *and* $rank(\mathbf{H}^y)\leqslant 5$.

### B. Intuition about the AHC Matrix

Any two images of the same planar surface in space are related by a homography. Much work has been done in matching or estimation of affine homography, whose last row is fixed to $P_{31}=0, P_{32}=0, P_{33}=1$. That is

$$\begin{bmatrix}\widetilde{\mathbf{x}}^T\\ \widetilde{\mathbf{y}}^T\\ \widetilde{\mathbf{1}}^T\end{bmatrix}=\begin{bmatrix}P_{11} & P_{21} & P_{31}\\ P_{21} & P_{22} & P_{23}\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\\ \mathbf{1}\end{bmatrix}.$$

Therefore, $\begin{bmatrix}\mathbf{x} & \mathbf{y} & \mathbf{1}\end{bmatrix}^T$ and $\begin{bmatrix}\widetilde{\mathbf{x}} & \widetilde{\mathbf{y}} & \mathbf{1}\end{bmatrix}^T$ are linearly related. However, when the homography matrix represents the projective transformation that does not degenerate to affine transformation,

$$\lambda_i\begin{bmatrix}\widetilde{x}_i\\ \widetilde{y}_i\\ 1\end{bmatrix}=\begin{bmatrix}P_{11} & P_{12} & P_{13}\\ P_{21} & P_{22} & P_{23}\\ P_{31} & P_{32} & P_{33}\end{bmatrix}\begin{bmatrix}x_i\\ y_i\\ 1\end{bmatrix}\quad where\quad \lambda_i=P_{31}x_i+P_{32}y_i+P_{33}.$$

For each point, $\lambda_i$ is unknown, which brings great difficulty in estimating the projective transformation. But if we want to recover the linear relation between $\begin{bmatrix}\mathbf{x} & \mathbf{y} & \mathbf{1}\end{bmatrix}^T$ and $\begin{bmatrix}\widetilde{\mathbf{x}} & \widetilde{\mathbf{y}} & \mathbf{1}\end{bmatrix}^T$, we can just multiply $[\widetilde{x}_i,\widetilde{y}_i,1]^T$ by $\lambda_i$. And then, we can substitute $\lambda_i$ by its physical meaning $\lambda_i=P_{31}x_i+P_{32}y_i+P_{33}$, which derives the linear relation among the permutated coordinates in vector form presented in Theorem 1. This idea is just the basic intuition to construct the AHC matrix in Definition 1 and Theorem 1.

### C. Calculating the Estimated Matched Points Based on the AHC Matrices

Now we consider a point $\mathbf{u}_i=[x_i,y_i,1]^T$ in image $Q$. We want to estimate its matched point $\overline{\mathbf{u}}_i=[\bar{x}_i,\bar{y}_i,1]^T$ in $\widetilde{Q}$. We show how to compute $\bar{x}_i$ and $\bar{y}_i$ robustly and efficiently. For brevity, we only give the details of computing $\bar{x}_i$.

In $\mathbf{H}^x$, every column is only relavant to coordinates of one pair of matched points. So it is free to append or reduce few columns. Suppose we get the $\mathbf{H}^x$ constructed from coordinates of two point sets and we assume the data are perfect without noise and outliers. Then we get another pair of perfectly matched points, knowing their coordinates. We can construct the new column by this new pair and append it to $\mathbf{H}^x$. The new $\mathbf{H}^x$ should also have rank 5. Now we look at the added column and consider the added $x$ to be an unknown variable. When the value of this variable deviates from the value with the physical meanning (coordiate $x$)the rank of $\mathbf{H}^x$ becomes 6, which can be reflected sensitively by its determinant. By this idea, we derive a method to estimate matched coordinates with a closed-form solution.

Suppose that $[\alpha,\beta,1]^T$ exactly matches $[x_i,y_i,1]^T$. We add a column $\mathbf{h}_i^x(\alpha)=[\alpha\mathbf{u}_i;\mathbf{u}_i]=[\alpha\cdot x_i,\alpha\cdot y_i,\alpha,x_i,y_i,1]^T$ to $\mathbf{H}^x$ and obtain

$$\mathbf{H}_i^x(\alpha)=[\mathbf{h}_i^x(\alpha),\mathbf{H}^x]. \tag{2}$$

Since all the points are assumed to be matched, by Theorem 1 $\mathrm{rank}(\mathbf{H}_i^x(\alpha))\le 5$. Thus $\det(\mathbf{H}_i^x(\alpha)\mathbf{H}_i^x(\alpha)^T)$ should be zero. However, Theorem 1 is deduced under the assumption that all the points are correctly matched and there are no noises in the matched points. These may not be true. Instead, due to the positive semi-definiteness of $\mathbf{H}_i^x(\alpha)\mathbf{H}_i^x(\alpha)^T$, most likely we will have $\det(\mathbf{H}_i^x(\alpha)\mathbf{H}_i^x(\alpha)^T)>0$. Thus we can solve

$$\min_{\alpha\in\mathbb{R}}\det(\mathbf{H}_i^x(\alpha)\mathbf{H}_i^x(\alpha)^T) \tag{3}$$

instead and assign its minimizer to $\bar{x}_i$.

Next, we show that $\det(\mathbf{H}_i^x(\alpha)\mathbf{h}_i^x(\alpha)^T)$ is actually a quadratic function of $\alpha$. Hence, (3) has a simple closed form.

**Theorem 2.** *Patitioning* $(\mathbf{H}^x\mathbf{H}^{x,T})^{-1}$ *as* $\begin{bmatrix}\mathbf{Z}_{11}^x & \mathbf{Z}_{12}^x\\ \mathbf{Z}_{21}^x & \mathbf{Z}_{22}^x\end{bmatrix}$*, the solution to* (3) *is*

$$\bar{x}_i=-\frac{\mathbf{u}_i^T\mathbf{Z}_{21}^x\mathbf{u}_i}{\mathbf{u}_i^T\mathbf{Z}_{11}^x\mathbf{u}_i},\quad i=1,\cdots,n. \tag{4}$$

**Proof.** *First,*

$$\det(\mathbf{H}_i^x(\alpha)\mathbf{H}_i^x(\alpha)^T)=\det(\mathbf{h}_i^x(\alpha)\mathbf{h}_i^x(\alpha)^T+\mathbf{H}^x\mathbf{H}^{x,T}). \tag{5}$$

*Since there actually exist noises in the coordinates, it is reasonable to assume that* $\mathbf{H}^x$ *is of full rank*[1]*. Therefore* $\mathbf{H}^x\mathbf{H}^{x,T}$ *is positive definite. Hence there exists an invertible square matrix* $\mathbf{V}^x$*, such that* $\mathbf{V}^x(\mathbf{H}^x\mathbf{H}^{x,T})\mathbf{V}^{x,T}=\mathbf{I}_{(6)}$*. Then we can check that* $\mathbf{V}^{x,T}\mathbf{V}^x=(\mathbf{H}^x\mathbf{H}^{x,T})^{-1}$ *and*

$$\begin{aligned}&\det(\mathbf{H}_i^x(\alpha)\mathbf{H}_i^x(\alpha)^T)\\ =&\frac{1}{\det(\mathbf{V}^x\mathbf{V}^{x,T})}\det((\mathbf{V}^x\mathbf{h}_i^x(\alpha))(\mathbf{V}^x\mathbf{h}_i^x(\alpha))^T+\mathbf{I}_{(6)})\\ =&\frac{1}{\det(\mathbf{V}^x\mathbf{V}^{x,T})}\left[1+\mathbf{h}_i^x(\alpha)^T\mathbf{V}^{x,T}\mathbf{V}^x\mathbf{h}_i^x(\alpha)\right]\\ =&\frac{1}{\det(\mathbf{V}^x\mathbf{V}^{x,T})}\left[1+\mathbf{h}_i^x(\alpha)^T(\mathbf{H}^x\mathbf{H}^{x,T})^{-1}\mathbf{h}_i^x(\alpha)\right]\\ =&\frac{1}{\det(\mathbf{V}^x\mathbf{V}^{x,T})}\left(1+\alpha^2\mathbf{u}_i^T\mathbf{Z}_{11}^x\mathbf{u}_i+2\alpha\mathbf{u}_i^T\mathbf{Z}_{21}^x\mathbf{u}_i+\mathbf{u}_i^T\mathbf{Z}_{22}^x\mathbf{u}_i\right).\end{aligned} \tag{6}$$

*It is a quadratic function of* $\alpha$ *and its minimizer is* (4)*.*

An efficient Matlab implementation of computing all $\bar{x}_i$'s given in (4), rather than looping (4) from 1 to $n$, is as follows:

$$\begin{aligned}\overline{\mathbf{x}}^T&=-\mathrm{sum}(\mathbf{B}^x(4:6,:))./\mathrm{sum}(\mathbf{B}^x(1:3,:)),\\ &\text{where }\mathbf{B}^x=\begin{bmatrix}\mathbf{U}\\ \mathbf{U}\end{bmatrix}\odot\left(\begin{bmatrix}\mathbf{Z}_{11}^x\\ \mathbf{Z}_{21}^x\end{bmatrix}\mathbf{U}\right).\end{aligned} \tag{7}$$

Note that we do not explicitly recover the projective transformation but still obtain the estimated matched points in the target image. The algorithm to estimate the matched points in the target image is summarized in Algorithm 1, where $\mathscr{I}^{(t)}$ is the index set of anchor matches in the $t$-th iteration whose choice will be explained immediately.

### D. Selecting Anchor Matches

Estimating the matched points requires reliable anchor matches. Unfortunately, there is no guarantee that the chosen anchor matches are indeed correct matches. So we have to update the set of anchor matches iteratively so that it becomes more and more trustworthy. The initial choice of anchor matches is simply all the matches, i.e., $\mathscr{I}^{(1)}=\{1,\cdots,n\}$.

[1]Even if $\mathbf{H}^x$ is degenerate, we may still add small perturbation to $\mathbf{H}^x$ and let the perturbation approach zero. So the following arguments are still valid.

**Algorithm 1** Estimating the Matched Points Based on Existing Anchor Matches

**Input:** $\mathbf{U}, \widetilde{\mathbf{U}}, \mathscr{I}^{(t)}$.

**Process** :

1: $\mathbf{H}^x \leftarrow [(\mathbf{1}_{(3)} \otimes (\tilde{\mathbf{x}}(\mathscr{I}^{(t)}))^T) \odot \mathbf{U}(:,\mathscr{I}^{(t)}); \mathbf{U}(:,\mathscr{I}^{(t)})]$, $\mathbf{H}^y \leftarrow [(\mathbf{1}_{(3)} \otimes (\tilde{\mathbf{y}}(\mathscr{I}^{(t)}))^T) \odot \mathbf{U}(:,\mathscr{I}^{(t)}); \mathbf{U}(:,\mathscr{I}^{(t)})]$,

2: Compute the inverses of $\mathbf{H}^x\mathbf{H}^{x,T}$ and $\mathbf{H}^y\mathbf{H}^{y,T}$ and partition them as $\begin{bmatrix}\mathbf{Z}^x_{11} & \mathbf{Z}^x_{12}\\ \mathbf{Z}^x_{21} & \mathbf{Z}^x_{22}\end{bmatrix}$ and $\begin{bmatrix}\mathbf{Z}^y_{11} & \mathbf{Z}^y_{12}\\ \mathbf{Z}^y_{21} & \mathbf{Z}^y_{22}\end{bmatrix}$, respectively,

3: Compute $\mathbf{B}^x = \begin{bmatrix}\mathbf{U}\\ \mathbf{U}\end{bmatrix} \odot \left(\begin{bmatrix}\mathbf{Z}^x_{11}\\ \mathbf{Z}^x_{21}\end{bmatrix}\mathbf{U}\right)$ and $\mathbf{B}^y = \begin{bmatrix}\mathbf{U}\\ \mathbf{U}\end{bmatrix} \odot \left(\begin{bmatrix}\mathbf{Z}^y_{11}\\ \mathbf{Z}^y_{21}\end{bmatrix}\mathbf{U}\right)$,

4: Compute $\bar{\mathbf{x}}^T = -\text{sum}(\mathbf{B}^x(4:6,:))./\text{sum}(\mathbf{B}^x(1:3,:))$ and $\bar{\mathbf{y}}^T = -\text{sum}(\mathbf{B}^y(4:6,:))./\text{sum}(\mathbf{B}^y(1:3,:))$.

**Output:** $\overline{\mathbf{U}}^{(t)} = [\bar{\mathbf{x}}^T; \bar{\mathbf{y}}^T; \mathbf{1}^T]$.

A robust method should have relatively small reprojection errors for correct matches and relatively large reprojection errors for mismatches. Therefore, we may select anchor matches which correspond to points with relatively small reprojection errors. More specifically, in the $t$-th iteration we first compute the estimated matched points, stored in $\overline{\mathbf{U}}^{(t)}$, by Algorithm 1. Then we perform z-score normalization on the first two rows of $\widetilde{\mathbf{U}} - \overline{\mathbf{U}}^{(t)}$, getting the mean $\mu_x^{(t)}$ and variance $\sigma_x^{(t)}$ for $x$ coordinates of anchor and the mean $\mu_y^{(t)}$ and variance $\sigma_y^{(t)}$ for $y$ coordinates of anchor. Then indices whose corresponding matches have relatively small normalized reprojection errors are considered as in the index set of anchor matches for the next iteration:

$$\mathscr{I}^{(t+1)} = \left\{ i \left| \left| \frac{(\widetilde{x}_i - \bar{x}_i^{(t)}) - \mu_x^{(t)}}{\sigma_x^{(t)}} \right| < \delta^{(t)}, \left| \frac{(\widetilde{y}_i - \bar{y}_i^{(t)}) - \mu_y^{(t)}}{\sigma_y^{(t)}} \right| < \delta^{(t)} \right. \right\}. \tag{8}$$

During the iteration, the threshold $\delta^{(t)}$ is gradually lowered, which means a stricter threshold is set to seek more qualified anchor matches. In our implementation, we set $\delta^{(t+1)} = \rho * \delta^{(t)}$, where $\rho \in (0,1)$ is a constant factor to lower the filtering threshold $\delta^{(t)}$ and we take $\rho = 0.98$ in all experiments.

To help decide when to terminate the iteration, we use a quantity $D^{(t)}$ to measure the quality of anchor matches, defined as the maximum reprojection error among the anchor matches:

$$D^{(t)} = \max_{i \in \mathscr{I}^{(t)}} \left\| [\bar{x}_i^{(t)}, \bar{y}_i^{(t)}, 1]^T - [\widetilde{x}_i, \widetilde{y}_i, 1]^T \right\|_F. \tag{9}$$

If $D^{(t)} \leq$ *end_threshold*, where *end_threshold* is a threshold, the quality of anchor matches is considered good enough, thus the iteration should terminate. The anchor match selection process is summarized in Algorithm 2. Normally, the iteration terminates quickly.

### *E. Robustness Analysis on Our Algorithm*

In this subsection, we empirically analyze the robustness of our algorithm.

**Algorithm 2** Iterative Procedure for Selecting Anchor Matches

**Input:** $\mathbf{U}, \widetilde{\mathbf{U}}, \delta^{(1)}, \rho \in (0,1)$, end_threshold

**Initialization** : $t \leftarrow 1, D \leftarrow 1E+8,\ \mathscr{I} \leftarrow \{1, 2 \cdots, n\}$

**Process** :

1: **while** $D >$ end_threshold **do**

2: Compute $\overline{\mathbf{U}}^{(t)}$ by Algorithm 1,

3: Compute the mean $\mu_x^{(t)}$ and standard deviation $\sigma_x^{(t)}$ of anchor from the first row of $\tilde{\mathbf{U}} - \overline{\mathbf{U}}^{(t)}$, and the mean $\mu_y^{(t)}$ and standard deviation $\sigma_y^{(t)}$ of anchor from the second row of $\tilde{\mathbf{U}} - \overline{\mathbf{U}}^{(t)}$,

4: $\mathscr{I}^{(t+1)} \leftarrow \left\{ i \left| \left| \frac{(\widetilde{x}_i - \bar{x}_i^{(t)}) - \mu_x^{(t)}}{\sigma_x^{(t)}} \right| < \delta^{(t)}, \left| \frac{(\widetilde{y}_i - \bar{y}_i^{(t)}) - \mu_y^{(t)}}{\sigma_y^{(t)}} \right| < \delta^{(t)} \right. \right\}$,

5: $D \leftarrow \max_{i \in \mathscr{I}^{(t)}} \left\| [\bar{x}_i^{(t)}, \bar{y}_i^{(t)}, 1]^T - [\widetilde{x}_i, \widetilde{y}_i, 1]^T \right\|_F$,

6: $\delta^{(t+1)} = \rho * \delta^{(t)}$,

7: $t \leftarrow t+1$,

8: **end while**

9: $\Omega \leftarrow \left\{ i \left| \left\| [\widetilde{x}_i, \widetilde{y}_i, 1]^T - [\bar{x}_i^{(t)}, \bar{y}_i^{(t)}, 1]^T \right\|_F > \textit{end\_threshold} \right. \right\}$,

**Output:** $\Omega$ is the index set of outliers.

*1) AHC Matrix Based Approach Compared with Least Squares Based One in Robustness of Estimating Matched Points:* At the first glance, it seems that the framework of our method is similar to the classic RANSAC in that both are iterative and refine the anchor matches for better estimation. However, they are of different mechanisms. RANSAC randomly initializes the set of anchor matches multiple times, while our method does not. Moreover, our method does not explicitly recover the projective transformation, while RANSAC estimates the transformation via least squares. Instead, we use the AHC matrix based approach to directly estimate matched points in the target image, thus choosing the anchor matches more straightforwardly. Here we will show that our AHC matrix based approach is more robust than the least squares based approach in estimating the matched points, in particular when the noises and outliers are severe.

We randomly generate a transformation matrix $\mathbf{P} = [0.189, 0.362, 7.342E-10; -0.423, 0.220, -9.778E-9; -5.334E-4, 2.777E-4, 1]$ and a point set including $10,000$ points within an image of $1,000 \times 1,000$ pixels. To test the robustness to noises and outliers, we first add weak Gaussian noises with 1 pixel standard deviation ($\mu = 0, \sigma = 1$) to all the true matched points in the image to simulate the noises caused by the imperfection of the feature point detector and quantization. Then we replace 70% of the true matched points in image $\widetilde{Q}$ with uniformly randomly chosen points in $\widetilde{Q}$. The resulted matches are regarded as outliers.

After BOF matching, we take all the resulted putative matches to calculate the estimated matched points in image $\widetilde{Q}$. We call the distance between the true matched point (mapped by the ground truth transformation) and the actually matched point as *ground truth error*. If the ground truth error of a match is greater than 5 pixels, the match is considered a mismatch, or outlier. If the error is below 5 pixels, the match is considered a correct match, or inlier.

Figure 2 displays the reprojection errors (2D vectors) after

certain iterations. We can see the reprojection errors of inliers distribute in a much smaller region than those of outliers. Thus it is easy to separate the inliers from the outliers. Therefore, the anchor matches can be more and more trustworthy. In comparison, as shown in Figure 3(b), if the estimated matched points are mapped by the projective transformation estimated by the least squares method, it is very hard to separate the inliers from the outliers because they are mixed up and distribute across a large area.

*2) Stability of Anchor Matches:* As the reprojection errors of inliers and outliers are relatively easy to identify, the quality of anchor matches can be improved constantly through the iterations, as shown in Figure 2. In this part, we show the stability of anchor matches through the iterations.

As described in subsection III-D, we determine an index set of anchor matches in each iteration. If Algorithm 1 works ideally, the set of anchor matches determined in the last iteration should naturally include the one in the current iteration without extra processing. In the experiment, there are $10,000$ given matches with 100% points added with 1-pixel-standard-deviation noises and 80% true matched points are made outliers by replacing them with uniformly random points in the target image. Algorithm 1 terminates after 9 iterations with $F$-*score* (defined in (10)) 1, which means that all the outliers are detected correctly.

TABLE II
TRACKING WHETHER A POINT IS IN THE SET OF ANCHOR MATCHES (A) OR NOT (N) THROUGH THE ITERATIVE PROCESS.

| index / iter. # | ··· | 5010 | 5011 | 5012 | 5013 | 5014 | 5015 | 5016 | 5017 | ··· |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ··· | A | A | A | A | A | A | A | A | ··· |
| 2 | ··· | A | N | A | A | A | A | A | A | ··· |
| 3 | ··· | A | N | A | A | A | A | A | A | ··· |
| 4 | ··· | A | N | A | A | A | A | A | A | ··· |
| 5 | ··· | A | N | A | A | A | A | N | A | ··· |
| 6 | ··· | A | N | A | A | N | A | N | A | ··· |
| 7 | ··· | N | N | A | A | N | A | N | A | ··· |
| 8 | ··· | N | N | N | A | N | A | N | A | ··· |
| 9 | ··· | A | N | N | A | N | A | N | A | ··· |

Part of the tracking results of anchor matches is shown in Table II. We can see that the set of anchor matches is almost always shrinking along with iteration. Actually, 97.31% of all the points have zero or one label change. This experiment verifies the great stability of our method.

## IV. EXPERIMENTS

In this section, we first compare our method with nine state-of-the-arts for removing mismatches from putative matches in synthetic datasets, including RANSAC [19], MLESAC [20], SparseVFC [21], VFC [21], ICF [22], LPM [25], SIM [18], L1GGC [17], and LRGGC [16]. Then we compare with five more state-of-the-arts for removing mismatches from putative matches in real datasets, including GC [14] [15], PGM [13], SGC [12], EWGC [3], and WGC [11]. These five methods are not compared with on the synthetic datasets because they need scale and orientation of SIFT features as input, which is unavailable from synthetic datasets. Finally, all these fourteen methods are applied to partial-duplicate image search to see how they improve the retrieval performance. All the experiments are done using MATLAB R2016b on a desktop computer with an Intel Core i5-4570 3.2GHz processor and 32GB of RAM.

We use $F$-*score* to evaluate the performance of all the methods. $F$-*score* is defined as:

$$F\text{-}score = 2 \times (precision \times recall)/(precision + recall), \quad (10)$$

where *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. *precision* and *recall* are commonly used measure to evaluate the performance of all the methods on detecting mismatches. In our experiment, *precision* and *recall* have an equal weight in $F$-*score*.

### A. Mismatch Removal on Synthetic Datasets

We generate $1,000$ trials in total. For each trial, we randomly generate a projective transformation matrix and a point set including 200 points within an image of $1,000 \times 1,000$ pixels. To make sure that the transformation matrix is reasonable and meaningful in image matching, we construct a rectangular pyramid with a square base. Then we use a random plane to intersect with the pyramid, obtaining a quadrilateral. Then $\mathbf{P}$ is computed as the transformation matrix from the quadrilateral to the square.

In our evaluation we consider the following two scenarios: affine transformation and projective transformation. So we make two copies of the given point set to generate true matches. One is for affine transformation and the other is for projective transformation. For affine transformation, we set $\mathbf{P}(3,:) = [0,0,1]$ after generating the random $\mathbf{P}$. When two transformated point sets are generated, we add Gaussian noises with different standard deviations and outliers with different percentages to them.

Now we analyze the performance of the methods in comparison on synthetic data.

*1) Comparison at Different Levels of Noises:* For different levels of noises, we set the standard deviation of Gaussion noises to be 1, 2, 3, 4, 5, 6, 7, and 8 pixels, respectively. They are added to 100% data. With added noises, the matches whose ground truth errors are greater than 2 (3, 4, 5, 6, 7, 8, and 9 pixels, respectively) are considered as mismatches (outliers). As is shown in Figures 5(a) and (c), our method achieves the highest $F$-*score* among all the state-of-the-arts at all levels of Gaussian noises. For affine transformations, only our method can achieve an average $F$-*score* higher than 0.80 at 8 pixels noises. For projective transformations, only our method can achieve an average $F$-*score* higher than 0.78 at all levels of noises. Moreover, we note that, compared with results on affine transformations, our method can handle projective transformations well with less drop in $F$-*score*. Though SIM is the only algorithm other than our method that can achieve an average $F$-*score* higher than 0.70 under affine transformations at 8 pixels, its $F$-*score* under projective transformation is about 0.1 lower than that under affine transformation, because it is designed for affine transformation only. This testifies to the effectiveness of our method on projective transformations.

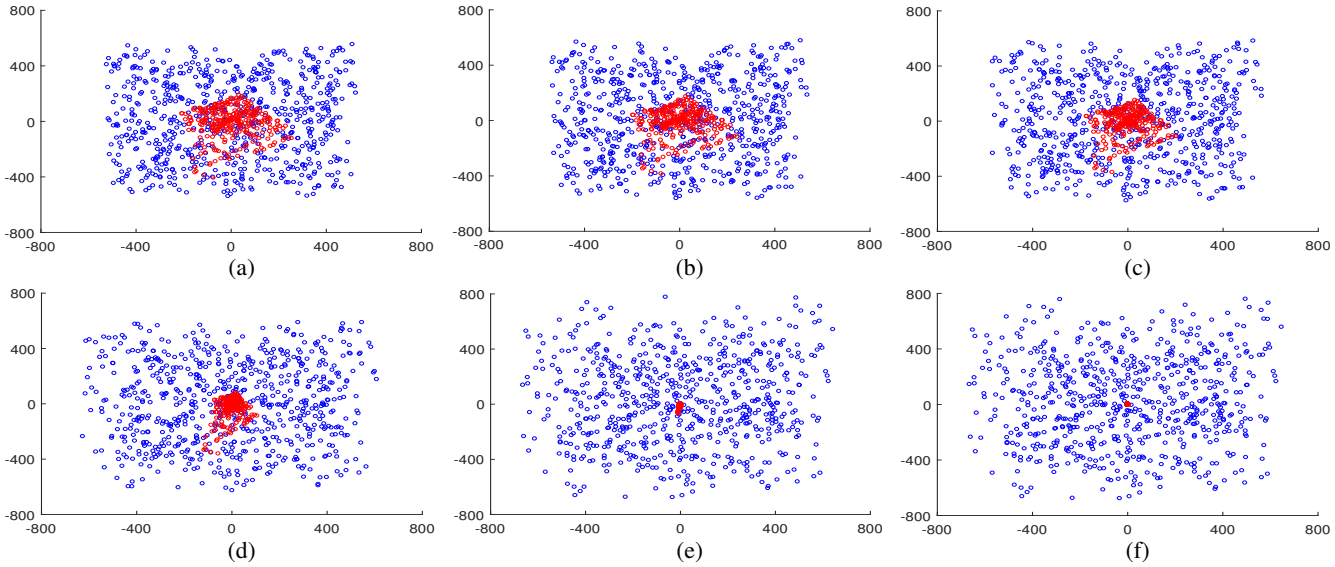


Fig. 2. (a)-(f) display the reprojection errors (2D vectors) after iteration 2, 4, 6, 8, 10, and 12, respectively. All the error vectors of outliers are marked as blue scattered points and those for inliers are red. Through the iterations, the reprojection errors gradually approach the ground truth errors. This experiment ends up with *F-score* 1 after 12 iterations.

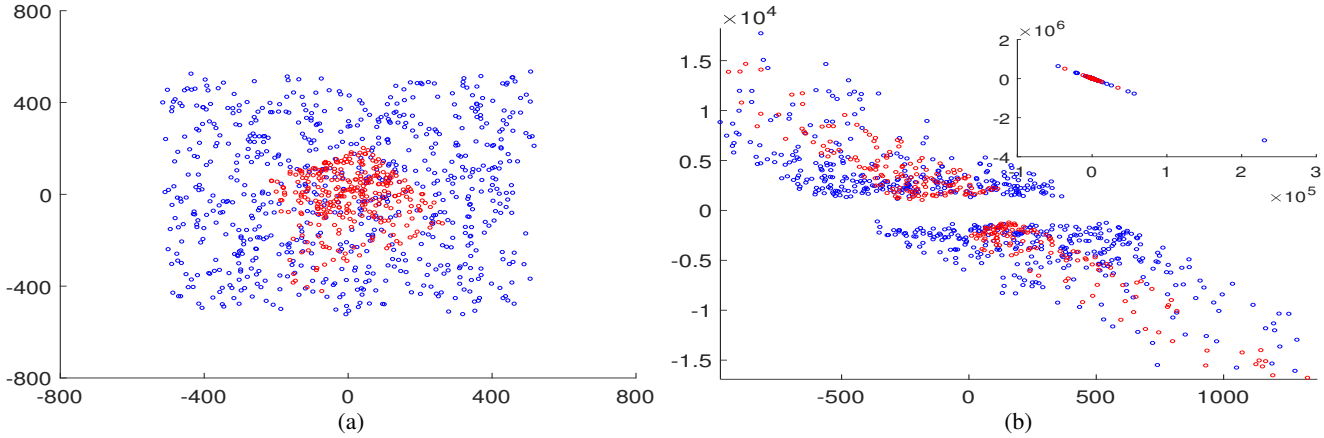


Fig. 3. The reprojection errors after the first iteration by (a) our AHC based method and (b) the least square method. All the reprojection errors (2D vectors) of outliers are marked as blue scattered points and those for inliers are red. In (a), the reprojection errors of inliers distribute in a much smaller area than those of outliers, which makes more accurate selection of inliers possible. In contrast, in (b) the reprojection errors of inliers and outliers are mixed up and distribute across a large area, making the separation of inliers and outliers difficult. The inset figure give the complete range of all the error vectors.

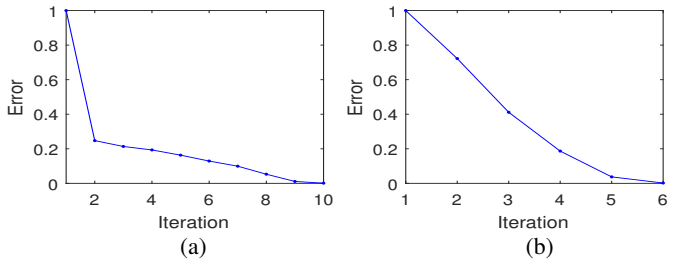


Fig. 4. Convergence results on synthetic datasets. (a) synthetic dataset with noise, (b) synthetic dataset with outliers.

*2) Comparison at Different Percentages of Outliers:* To make the data even more challenging, we combine weak noises with outliers in this experiment. We first add weak Gaussian noises with the standard deviation of 1 pixel ($\mu = 0, \sigma = 1$) to all the points to simulate measurement errors. Next, by replacing 10%, 20%, 30%, 40%, 50%, 60%, 70%, and 80% true matched points with uniformly random ones in the target image, we have different percentages of outliers. As is shown in Figures 5(b) and (d). Though our method, RANSAC, MLESAC, VFC, and SparseVFC all have satisfactory robust performance in this test, the time cost of RANSAC, MLESAC, and VFC is much higher.

*3) Comparison on Computing Efficiency:* The computing time of the methods in comparison are shown in Table III. Our method achieves the best performance on detecting true matches with almost the shortest time. Its time cost is only higher than SIM and LPM, two non-iterative methods.

To conclude, our AHC matrix based method demonstrates its capability of handling strong affine and projective transformations and achieves higher *F-score*'s than those of the state-of-the-arts. It is also shown to be quite robust under noises and outliers.

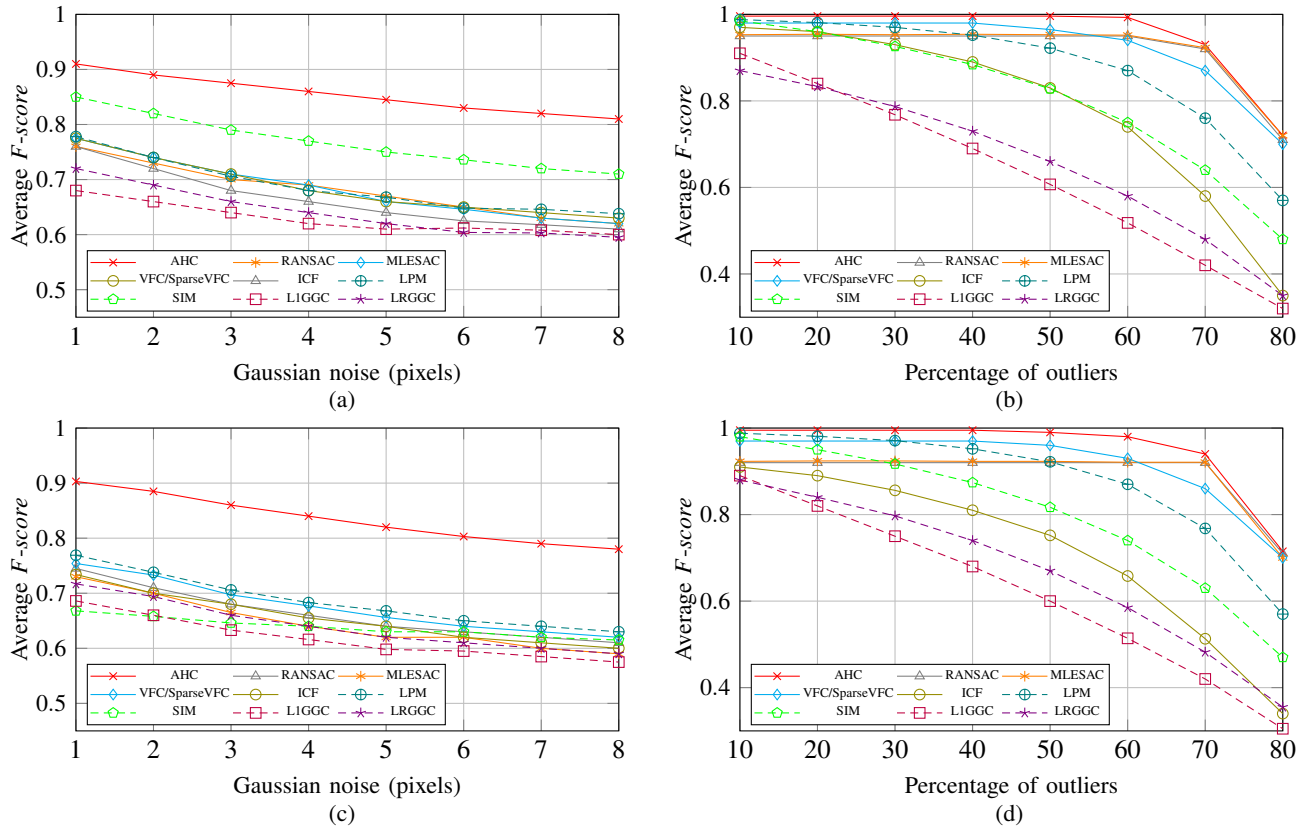


Fig. 5. The average *F-score*'s of ten methods on synthetic datasets under different: (a) standard deviations of Gaussian noises under affine transformation, (b) percentages of outliers under affine transformation, (c) standard deviations of Gaussian noises under projective transformation, (d) percentages of outliers under projective transformation.

TABLE III
AVERAGE *F-score* AND TIME COST COMPARISON OF TEN METHODS ON THE SYNTHETIC DATASET WITH AFFINE OR PROJECTIVE TRANSFORMATION.

| Category | Method | Average F-score | Average time cost(s) |
|---|---|---|---|
| Iterative | **AHC** | **0.885** | **0.0021** |
| | RANSAC [19] | 0.793 | 0.0932 |
| | MLESAC [20] | 0.794 | 0.0966 |
| | SparseVFC [21] | 0.798 | 0.0050 |
| | VFC [21] | 0.789 | 0.0298 |
| | ICF [22] | 0.710 | 0.0457 |
| Non-iterative | LPM [25] | 0.784 | 0.0013 |
| | SIM [18] | 0.750 | 0.0012 |
| | L1GGC [17] | 0.629 | 0.0044 |
| | LRGGC [16] | 0.654 | 0.0080 |

### *B. Mismatch Removal on Real Image Dataset*

In this subsection, to test the performace of our method under several types of transformations and distortions, we select Mikolajczyk and Schmid [26] as the benchmark dataset, which has fourty images in eight groups. Each group includes five matched images with similarity, affine and projective transformations and different lighting conditions. The ground truth transformation matrices between the query image and the five matched images are provided respectively, which help calculate the true matched points. We take the points whose ground truth errors are within 5 pixels as the correct matches. We use ASIFT [7] with default settings as the detector and descriptor to obtain putative matches. The average percentage of the correct matches in the dataset is 94.34%, and the average number of putative matches is about 3,468.

Table IV shows comparisons between our method and other fourteen state-of-the-arts on the average *F-score* and the average time cost. In the experiment, our method achieves the highest *F-score* with almost the fastest speed, our average runtime is merely 4ms. Note that we also count the average number of iterations of each method. Results in Table IV display that in average our method only requires 1.5 iterations, while all the others require more than 5. This also explains why our method excels in speed on the dataset though it is iterative.

In Table IV, the competitive methods are those whose average runtime is shorter than 1*s* and whose average *F-score*'s are higher than 0.95, including our method, SparseVFC, SIM, and LPM. Figure 8 shows the curves of these four algorithms, reflecting the increase of their average runtime with respect to the increase of the number of putative matches. We can see that the average runtime of SIM is sensitive to the number of putative matches on the Mikolajczyk and Schmid dataset, but the speed of our method and SparseVFC is not significantly influenced by the number of putative matches. Note that SIM is *non-iterative*. In the inset figure, we can see that the growth of time cost of our method is much more stable than that of

TABLE IV
COMPARISONS AMONG ALL THE METHODS ON THE AVERAGE *F-score* AND TIME COST ON THE MIKOLAJCZYK AND SCHMID DATASET.

| Category | Method | Average iterations | Average F-score | Average time cost(s) |
|---|---|---|---|---|
| Iterative | **AHC** | **1.5** | **0.983** | **0.004** |
| | RANSAC [19] | 10.5 | 0.973 | 1.533 |
| | MLESAC [20] | 10.5 | 0.973 | 1.539 |
| | VFC [21] | 8.5 | 0.950 | 9.124 |
| | SparseVFC [21] | 10.0 | 0.950 | 0.016 |
| | ICF [22] | 6.0 | 0.967 | 8.557 |
| Non-iterative | LPM [25] | | 0.968 | 0.030 |
| | SIM [18] | | 0.953 | 0.285 |
| | L1GGC [17] | | 0.915 | 3.385 |
| | LRGGC [16] | | 0.941 | 4.687 |
| | GC [14] [15] | | 0.955 | 1.616 |
| | PGM [13] | | 0.877 | 2.684 |
| | SGC [12] | | 0.832 | 0.087 |
| | EWGC [3] | | 0.802 | 0.018 |
| | WGC [11] | | 0.708 | 0.002 |

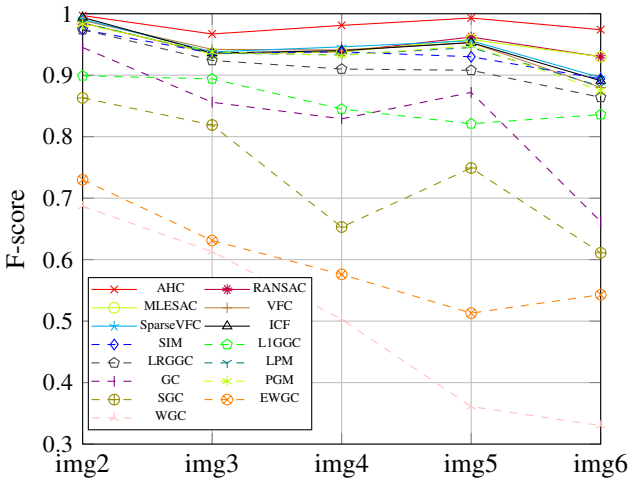


Fig. 6. The *F-score* of 2nd-6th 'graf' image matching 1st image with the strongly projective transformation on the Mikolajczyk and Schmid dataset.

SparseVFC, besides being lower.

In addition, the *F-score* of 'graf' image group is shown alone in Figure 6, The group includes five matched images with strongly projective transformations on the Mikolajczyk and Schmid dataset. Our method achieves the highest *F-score* among all the state-of-the-art geometric verification methods under strongly projective transformation.

As the Mikolajczyk and Schmid dataset includes only two groups with strongly projective transformations, we have manually collected 50 image pairs with strong projective transformations in wild. In figure 7, due to the limitation of space, we only display 5 image pairs capturing different scenes and 6 subfigures comparing different visual effect by 6 state-of-the-arts on the same scene as an example. In Figure 7(a), the distortions from images in the second line to corresponding images in the first line are mainly projective transformations caused by different viewpoints of acquisition. After obtaining the putative matches of these 50 image pairs by ASIFT, we follow the strategy from [25] to determine the ground-truth of each image pair. The sizes of the captured images are all $800 \times 600$. The average percentage of the correct matches is 85.25%. And the average number of putative matches is about $1,196$. The average *F-score* and time cost of all methods on the self-captured projective image pairs are reported in Table V. According to the statistics, our method achieves the highest *F-score* with the second fastest speed. However WGC, which is the only method faster than ours, achieves the unsatisfactory *F-score* 0.59 compared with our 0.95 and our average time cost is merely 2ms compared with the 0.5ms of WGC. The competitive methods are those whose average *F-score*'s are higher than 0.90. Figures 7(b), (c), (d), and (e) display the visual effect of mismatch removal result by these algorithms, that is AHC, RANSAC, VFC/SparseVFC, and LPM, respectively. The visual effect of MLESAC is the same with RANSAC. In comparison, Figures 7(f) and (g) display the visual effect of ICF and PGM respectively, which have the less competitive *F-score* of 0.89 and 0.85, presenting obviously more false matches than the former ones.

TABLE V
COMPARISONS AMONG ALL THE METHODS ON THE AVERAGE *F-score* AND TIME COST ON THE SELF CAPTURED DATASET.

| Category | Method | Average iterations | Average F-score | Average time cost(s) |
|---|---|---|---|---|
| Iterative | **AHC** | **3.0** | **0.954** | **0.002** |
| | RANSAC [19] | 43.5 | 0.948 | 0.350 |
| | MLESAC [20] | 46.6 | 0.949 | 0.360 |
| | VFC [21] | 11.2 | 0.930 | 2.245 |
| | SparseVFC [21] | 23.0 | 0.930 | 0.011 |
| | ICF [22] | 12.0 | 0.894 | 2.025 |
| Non-iterative | LPM [25] | | 0.912 | 0.007 |
| | SIM [18] | | 0.850 | 0.076 |
| | L1GGC [17] | | 0.833 | 0.843 |
| | LRGGC [16] | | 0.610 | 2.870 |
| | GC [14] [15] | | 0.735 | 0.382 |
| | PGM [13] | | 0.858 | 0.610 |
| | SGC [12] | | 0.720 | 0.030 |
| | EWGC [3] | | 0.654 | 0.006 |
| | WGC [11] | | 0.596 | 0.0005 |

### C. Convergene Analysis on Our Algorithm

In this section, we present the empirical convergence analysis of our algorithm by the data from the synthetic dataset. We generate a projective transformation matrix and a point set including 200 points in each trial and run 2,000 times for synthetic experiments with noises and outliers respectively.

For the data with noise, we record the squared error from the estimated coordinates to the real coordinates of the batch of images in each iteration. The average error vs. iteration is plotted in Figure 4(a). Similarly, Figure 4(b) show the convergence performance on data with outliers.

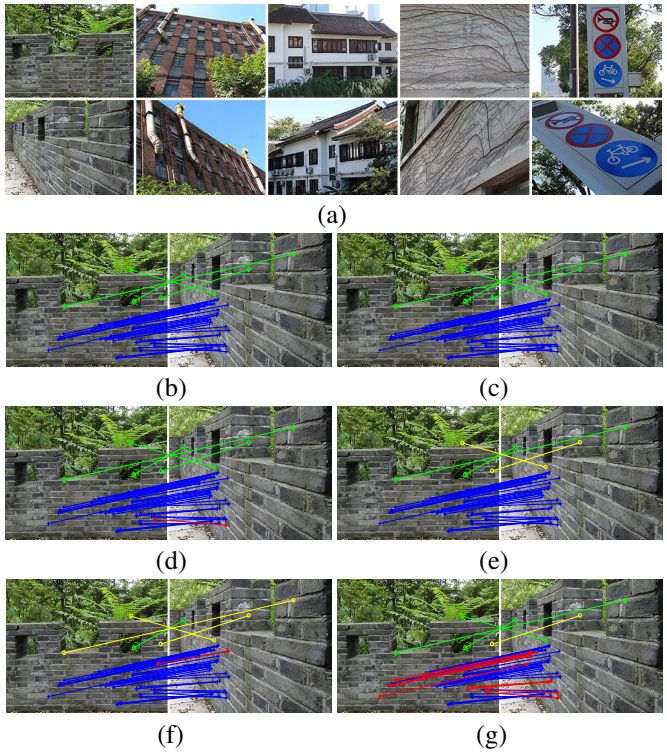


Fig. 7. (a) Five examples of image pairs with projective transformations captured naturally in wild. (b), (c), (d), (e), (f), and (g) The visual effect of mismatch removal result by our AHC, RANSAC, VFC/SparseVFC, LPM, ICF, and PGM, respectively. The visual effect of MLESAC is the same with RANSAC. The true positive pairs are in blue. The true negative pairs are in green. The false positive pairs are in yellow. The false negative pairs are in red.

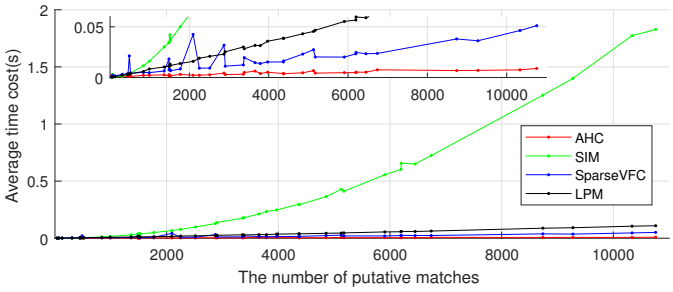


Fig. 8. The curves of computing time of our AHC matrix based method, SIM, SparseVFC and LPM vs. number of putative matches. The inset figure gives more details near the horizontal axis.

### D. Mismatch Removal for Partial-Duplicate Image Search

In this subsection, we apply our method to partial-duplicate image search to test its robustness and efficiency. We compare it with other fourteen state-of-the-art geometric verification methods on the performance of filtering outliers. The number of remaining matches is considered as a measure for re-ranking the coarse retrieval results by BOF matching.

*1) Datasets:* We test and evaluate on three popular benchmark datasets, the GCDup dataset [27] with 1,104 partial-duplicate images in 33 groups collected from the Internet, the Holiday dataset [11] with 1,491 near-duplicate images in 500 groups taken on many different scenes and the Oxford5k dataset [28] with 5,062 high resolution photos in 55 groups collected from Flickr by searching for some famous landmarks in Oxford. We set the first five images of each group in the three benchmark datasets as the query images. Then the rest of the images in the same group are expected to be ranked at the top of the retrieval results. We also use the MIRflickr1M dataset [29][2] as distractors. It contains one million unrelated images downloaded from Flickr.

*2) Experiment Settings:* After getting features from ASIFT [7] on the three benchmark datasets, we use the method of hierarchical k-means clustering [5] to train a codebook, which have one million visual words. Each 128-dimension feature descriptor is quantized into a visual word with the trained codebook. Once the visual words of one pair of feature points are the same, the pair is determined as a putative match.

Then we filter the mismatches with all the mismatch removal methods. Every method has its own number of remaining matches for re-ranking the coarse retrieval results. With different numbers of distracting images $1K$, $10K$, $100K$, and $1M$, the re-ranking range is set to be the top 500, 1,000, 5,000, and 10,000 images of the coarse retrieval results, respectively.

*3) Evaluation Metrics:* We evaluate the accuracy of compared methods with mean average precision (mAP) [27] and evaluate their speed with average computing time.

The mAP is defined as: $mAP = \sum_{q=1}^{Q_N} AP(q)/Q_N$, where $Q_N$ is the number of queries and $AP(q)$ is the average precision of the $q^{th}$ query. $AP$ [28] is the area under the precision-recall curve. $AP = \sum_{r \in R}(N_{rel}/r)/N_{all}$, where $r$ means the $r$-th ranked images, $R$ is the set of all the truly relevant images, $N_{rel}$ is the total number of $R$, and $N_{all}$ is the number of all the images.

TABLE VI
THE AVERAGE MAP AND THE AVERAGE TIME COST COMPARISON AMONG ALL THE METHODS ON THE THREE DATASETS WITH 10K DISTRACTORS.

| Method | Average mAP | | | Average time cost (s) | | |
|---|---|---|---|---|---|---|
| | Holiday | GCDup | Oxford5k | Holiday | GCDup | Oxford5k |
| **AHC** | **0.780** | **0.880** | **0.925** | **0.042** | **0.064** | **0.068** |
| RANSAC [19] | 0.761 | 0.860 | 0.858 | 0.901 | 0.967 | 1.219 |
| MLESAC [20] | 0.762 | 0.860 | 0.867 | 1.029 | 1.042 | 1.351 |
| SparseVFC[21] | 0.740 | 0.875 | 0.910 | 0.205 | 0.252 | 0.344 |
| VFC [21] | 0.588 | 0.857 | 0.828 | 0.430 | 0.954 | 0.889 |
| ICF[22] | 0.769 | 0.876 | 0.918 | 1.518 | 2.555 | 2.667 |
| LPM [25] | 0.766 | 0.870 | 0.916 | 0.050 | 0.081 | 0.090 |
| SIM [18] | 0.760 | 0.865 | 0.919 | 0.051 | 0.131 | 0.120 |
| L1GGC[17] | 0.771 | 0.877 | 0.917 | 0.174 | 0.435 | 0.416 |
| LRGGC[16] | 0.772 | 0.877 | 0.917 | 0.479 | 1.124 | 1.042 |
| GC[14] [15] | 0.516 | 0.853 | 0.716 | 0.301 | 0.835 | 0.779 |
| PGM[13] | 0.675 | 0.874 | 0.891 | 1.096 | 2.361 | 2.666 |
| SGC [12] | 0.765 | 0.876 | 0.916 | 0.108 | 0.266 | 0.260 |
| EWGC[3] | 0.714 | 0.872 | 0.894 | 0.032 | 0.077 | 0.076 |
| WGC[11] | 0.501 | 0.855 | 0.885 | 0.023 | 0.021 | 0.028 |

*4) Performance and Discussions:* Figure 9 displays the mAPs of all methods on the three datasets. Our method achieves the highest performance in mAP among all compared

[2]The image retrieval community often utilizes MIRFlickr to examine the scalability and robustness performance of a method by adding different numbers of its images to the benchmark datasets as distractors.

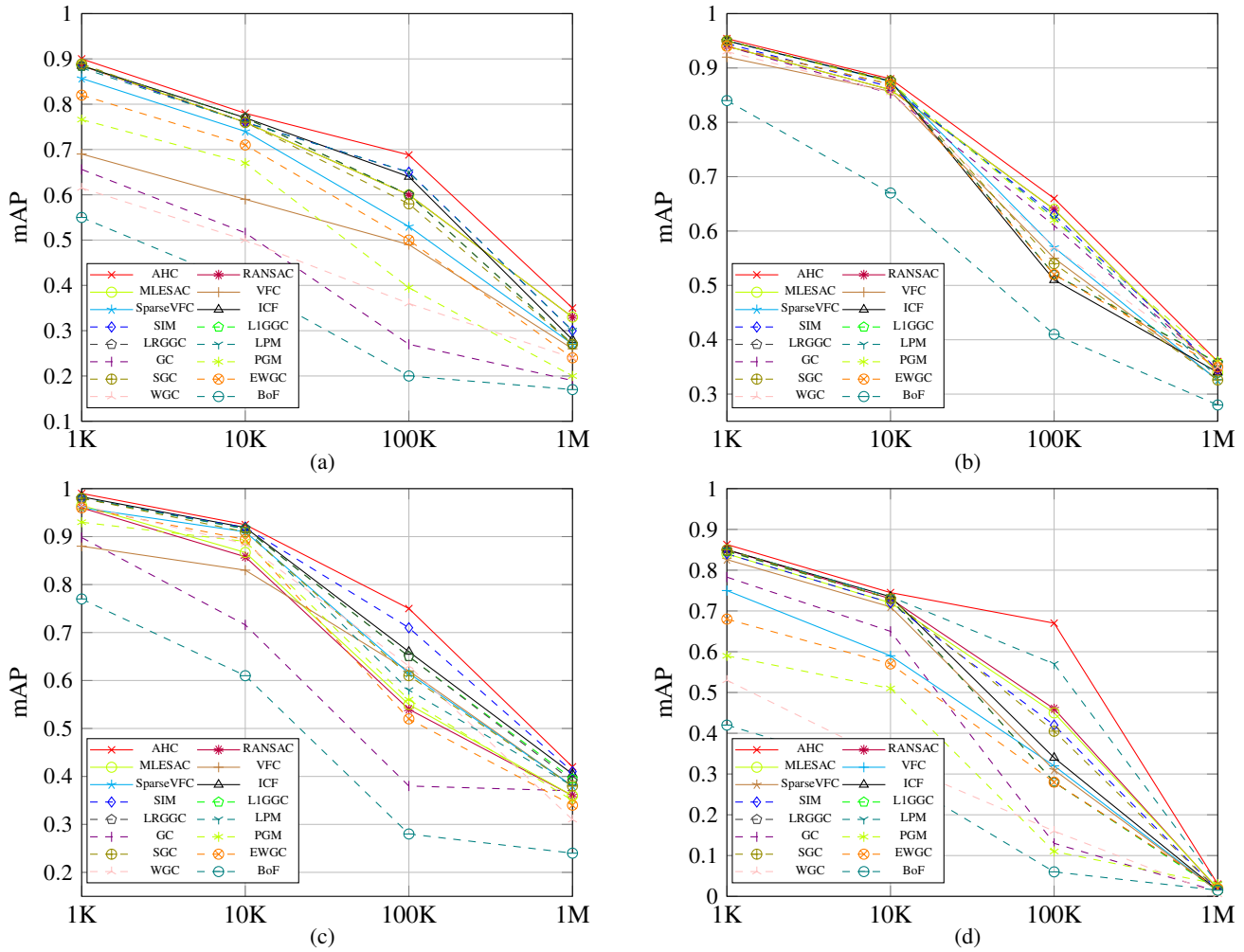


Fig. 9. The mAP on the three benchmark datasets after applying all the geometric verification methods with different number of distractor images. (a) Holiday dataset. (b) GCDup dataset. (c) Oxford5k dataset. (d) The mAPs on the strongly projective dataset with different numbers of distractors.

Fig. 10. Examples of the pairs of images with strongly projective transformations selected from the three benchmark datasets.

methods. Without any geometric verifications, BOF gets the worst results.

To put more emphasis on images with strongly projective transformation, we manually select 20 groups of images from the three benchmark datasets which have strongly projective transformation between them. Part of the examples are shown in Figure 10. Figure 9(d) gives the mAPs of all methods on this dataset. As one can see, our method performs better than all the other ones.

Table VI shows the average mAP and the average time cost per image query of all methods on the three datasets with $10K$ distractors. The time costs of feature extraction, codebook generation, and feature matching are excluded, as they are common steps in all these methods. We can see that our method is only slower than EWGC and WGC, two *non-iterative* methods. But they both achieve lower mAPs.

According to the above results, we conclude that our method is more suitable for handling projective transformation efficiently.

## V. Conclusions

We propose a novel mismatch removal method based on the newly constructed augmented homogeneous coordinates matrix. Given anchor matches, we only use the coordinates of feature points to construct augmented homogeneous coordinates matrices to establish the geometric and algebraic correspondence between two images. The AHC matrix based approach gives robust estimation on the matched points in the target image and can select high-quality anchor matches iteratively. Compared with state-of-the-arts, our method is simpler, faster, and is robust to projective transformations and noises and outliers, as shown in experimental results on both synthetic data and real data. In the future, we will target on giving in-depth theoretical analysis on the robustness of

our AHC matrix based approach and generalizing our method to handle more complex transformations, such as articulated motion and non-rigid deformation.

## REFERENCES


[1] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 36–51, 2008.

[2] D.-Q. Zhang and S.-F. Chang, "Detecting image near-duplicate by stochastic attributed relational graph matching with learning," in *Proc. ACM Conf. Multimedia*, 2004, pp. 877–884.

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

[4] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[5] 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.

[6] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Image Vis. Comput.*, vol. 22, no. 10, 2004, pp. 761–767.

[7] J. 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.

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

[9] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[10] 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.

[11] 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.

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

[13] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5153–5161.

[14] 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.

[15] 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.

[16] 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.

[17] Y. Lin, C. Xu, L. Yang, Z. Lin, and H. Zha, "L1-norm global geometric consistency for partial-duplicate image retrieval," in *Proc. IEEE Int. Conf. Image Process.*, Oct 2014, pp. 3033–3037.

[18] 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.

[19] M. A. Fischler and R. C. 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.

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

[21] 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.

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

[23] G. Wang, Z. Wang, Y. Chen, and W. Zhao, "Robust point matching method for multimodal retinal image registration," *Biomed. Signal Proc. and Control*, vol. 19, pp. 68–76, 2015.

[24] J. Ma, J. Zhao, J. Jiang, and H. Zhou, "Non-rigid point set registration with robust transformation estimation under manifold regularization," in *Proc. AAAI*, 2017, pp. 4218–4224.

[25] J. Ma, J. Zhao, H. Guo, J. Jiang, H. Zhou, and Y. Gao, "Locality preserving matching," in *Proc. IJCAI*, 2017, pp. 4492–4498.

[26] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, Oct. 2005, pp. 1615–1630.

[27] 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.

[28] 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.

[29] M. J. Huiskes, B. Thomee, and M. S. Lew, "New trends and ideas in visual concept detection," in *Proc. ACM Int. Conf. Multimedia Inf. Retr.*, 2010, pp. 527–536.