4FP-Structure: A Robust Local Region Feature Descriptor

Jiayuan Li, Qingwu Hu, and Mingyao Ai

Abstract

Establishing reliable correspondence for images of the same scene is still challenging work due to repetitive texture and unknown distortion. In this paper, we propose a region-matching method to simultaneously filter false matches and maximize good correspondence between images, even those with irregular distortion. First, a novel region descriptor, represented by a structure formed by four feature points (4FP-Structure), is presented to simplify matching with severe deformation. Furthermore, an expansion stage based on the special 4FP-Structure is adapted to detect and select as many high location accuracy correspondences as possible under a local affine-transformation constraint. Extensive experiments on both rigid and non-rigid image datasets demonstrate that the proposed algorithm has a very high degree of correctness and significantly outperforms other state-of-the-art methods.

Introduction

As a basic step for many remote sensing and computer vision applications, such as image registration (Brown and Lowe, 2003), structure from motion (Snavely *et al.*, 2006), and simultaneous localization and mapping (SLAM) (Montemerlo *et al.*, 2002), automatic image matching has been well studied in recent years. Current feature matching algorithms (Bay *et al.*, 2008; Ke and Sukthankar, 2004; Lourenço *et al.*, 2012; Lowe, 2004; Rublee *et al.*, 2011; Tola *et al.*, 2010) typically consist of three major stages: keypoint detection, keypoint description and keypoint matching. In the first stage, salient and stable interest points are extracted. These keypoints are then described based on their photometric neighborhoods using properties such as local gradients. In the third step, the distances between the descriptor vectors are calculated to recognize reliable correspondences. Among these methods, the most famous is the scale-invariant feature transform (SIFT) (Lowe, 2004) due to its robustness to image scale, rotation, illumination and viewpoint change.

For rigid scenes, such a framework can achieve remarkable results. Point correspondences can be produced with high correctness rate. Although there are some false matches because of ambiguities that arise from poor or repetitive texture, a postprocessing step such as RANSAC (Fischler and Bolles, 1981) or graph matching (Conte *et al.*, 2004) can be adopted.

The RANSAC algorithm is a robust technique for model fitting with noise and outliers, which has been widely used in computer vision and machine learning. The basic idea of RANSAC is simple but effective: first, randomly select a subset of correspondences to compute the candidate fundamental or homography matrix because perspective images satisfy the epipolar or homography constraint. Then, count the number of correspondences that support this transformation model. If the number is sufficiently large, the transformation matrix can be considered a good solution. The matches that support it will be accepted as inliers; in contrast, others will be discarded as outliers. RANSAC, however, works well only if two prerequisites are satisfied. The first is a sufficiently high inlier rate. Literature (Liu and Marlet, 2012) reports that RANSAC-like (Chum and Matas, 2005b; Chum *et al.*, 2003; Torr and Zisserman, 2000) methods may fail and become very time-consuming when the inlier rate is less than 50 percent. If the inlier rate is very small, the number of required iterations becomes huge. The other is the transformation model. A putative model must be given in advance, and the inlier set should satisfy this model well.

Graph matching (Cho and Lee, 2012; Conte *et al.*, 2004; Duchenne *et al.*, 2011) is another powerful and general tool for feature matching. It represents scene images as graphs using feature points, and correct correspondences can be extracted by solving a global optimization function to minimize the structural distortions between graphs (Cho and Lee, 2012). Unlike the RANSAC algorithm which only uses rigid geometric constraints, graph matching can also be applied to non-rigid scenes. However, current methods still assume that the inlier rate is relatively high. The large number of outliers arising from strong distortion may make them impractical. For instance, Duchenne *et al.* (2011) show that if the outlier rate is more than 70 percent, the performance of graph matching will severely drop. Another problem of graph matching is that it is NP-hard, so the computational costs in time and memory limit the permissible sizes of input graphs.

In this paper, we also focus on feature matching for non-rigid scenes, e.g., fisheye images. A fisheye lens has a large field of view (FOV), which is needed for many vision tasks in photogrammetry and computer vision. For instance, five fisheye images are sufficient for 360° panoramic stitching, but nine perspective images are needed; self-driving vehicles (Geiger *et al.*, 2012) need a large FOV to accurately sense the environment to plan their route. However, fisheye images have an inherent drawback: distortion is severe. Because of that, SIFT (Lowe, 2004) usually cannot work well, and the outlier rate may be very high (higher than 50 percent). In addition, a fisheye image no longer satisfies the homography constraint and has its own epipolar geometry, which can be applied only if the calibration information is provided. These issues make feature matching challenging, as the prerequisites of RANSAC and graph matching are not well satisfied.

To exactly distinguish inliers from outliers for both rigid and non-rigid images, a region-matching method is proposed. We first define a 4FP-Structure, formed by four neighborhood feature points, to represent the local region. Using local regions instead of feature points for matching has two advantages: (a) The 4FP-Structure is a 4-node graph that has the ability to resist the distortion in a small region, and it contains four feature points that can restrain each other to

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China (huqw@whu.edu.cn).

Photogrammetric Engineering & Remote Sensing
Vol. 83, No. 12, December 2017, pp. 813–826.
0099-1112/17/813–826

doi: 10.14358/PERS.83.12.813

maintain geometric consistency. These properties ensure that good matches can be separated from false matches. (b) Once the local region has been correctly matched, the local affine transformation of this region can be computed based on the four good correspondences. We use four feature points to construct the structure because of the consideration of redundant observations. Taking advantage of the special 4FP-Structure, we propose an expansion stage. Many good matches are undetected by SIFT because of the strong distortion, and the expansion stage has the ability to find as many of them as possible. In addition, our 4FP-Structure consists of only six connecting lines, so the computational complexity is small. The results on a variety of datasets indicate that our work is robust to outliers and can outperform recent state-of-the-art algorithms.

Figure 1 shows the schematic diagram of the main ideas in this paper. Given a pair of images, we first compute a set of initial matches based on feature matching methods, such as SIFT (Lowe, 2004), SURF (Bay *et al.*, 2008), and ORB (Rublee *et al.*, 2011). For each correspondence, we carefully select its three neighbors to construct a local structure called the 4FP-Structure. The 4FP-Structure is then matched based on a region descriptor. The local affine transformation is also established simultaneously by the specially designed structure. Finally, matches inconsistent with the transformation are removed as outliers, and keypoint pairs consistent with the transformation are accepted as inliers. The proposed method is able to eliminate outliers and maximize good correspondences simultaneously. In addition, it is suitable for both rigid and non-rigid image scenes.

The rest of the paper is organized as follows: The next section reviews several related works, followed by details of the proposed region matching method. In the next section, we show the experimental results on both rigid and non-rigid image datasets.Finally, we present a conclusion.

Related Work

Establishing reliable correspondence between a pair of images has been studied extensively. This section briefly reviews several related works, including RANSAC-like methods, graph-based group and other recently presented techniques.

As mentioned earlier, RANSAC uses a hypothesize-and-verify technique, which alternately computes a putative correspondence set and estimates the geometric transformation. There are several variants of RANSAC, such as MLESAC (Torr and Zisserman, 2000), LO-RANSAC (Chum et al., 2003), and PROSAC (Chum and Matas, 2005a). MLESAC introduces a new robust estimator to maximize the likelihood rather than just the inliers. LO-RANSAC enhances RANSAC with a local optimization step, which improves both the speed and the quality of traditional RANSAC. In PROSAC, the random sampling step is improved. PROSAC draws random samples based on local similarity ordering instead of uniform sampling. Similarly, Li *et al.* (Li *et al.*, 2017a) propose a normalised barycentric coordinate system (NBCS) to improve the random sampling step. A comparative analysis of the RANSAC family may be found in (Raguram *et al.*, 2008).

Unlike the RANSAC family, which assumes rigid geometric constraints, the graph-matching group is also suitable for non-rigid scenes. Progressive graph matching (PGM+MPM) (Cho and Lee, 2012) focuses on the initial construction problem of graphs and presents a move-making approach for graph matching. The progressive framework alternately performs graph probabilistic progression and graph matching steps. Torresani *et al.*,(2008) cast this task as an energy minimization problem with an objective function depending on feature similarity, geometric consistency, and spatial coherence. The dual decomposition approach is then adopted to optimize this function. In this work, the geometric consistency term is measured by a "neighborhood system." Cho *et al.* (2014) introduce a max-pooling strategy to graph matching (MPM), which is robust to deformations and outliers. Each candidate correspondence is evaluated by its neighbors with contextual information, and the matching scores are gradually propagated to update the most promising neighbors. Similar to Torresani's method and MPM, our method also uses local feature neighbors to improve the matching performance. Unlike these methods, the local neighbors of our method form a special structure that can identify the local neighbors as good matches or not and estimate the local affine transformation. In addition, in Torresani's method and MPM, local feature neighbors are only used to compute geometric consistency and matching scores in the graphs, respectively. Compared with these global optimization problems, the proposed local-region matching approach is much more efficient and requires less memory.

Cho *et al.* (2009) formulate image matching as a clustering problem and present a novel linkage model and a new dissimilarity metric in the framework of hierarchical agglomerative clustering (ACC). In this approach, initial compact correspondence clusters are first formed, and then, locally connected matches are progressively merged with high confidence. Literature (Ma *et al.* (2015a) introduces the L2E estimator to estimate the transformation between correspondence sets. Similar to the famous iterative closest point (ICP) (Best and McKay, 1992) algorithm, it also iteratively establishes the point correspondences and estimates the transformation. Wang *et al.* (2015) use a probability model, called the mixture of asymmetric Gaussian model (MoAG) (Kato *et al.*, 2002), to represent each point set. The matching task is then formulated as an optimization problem and solved under

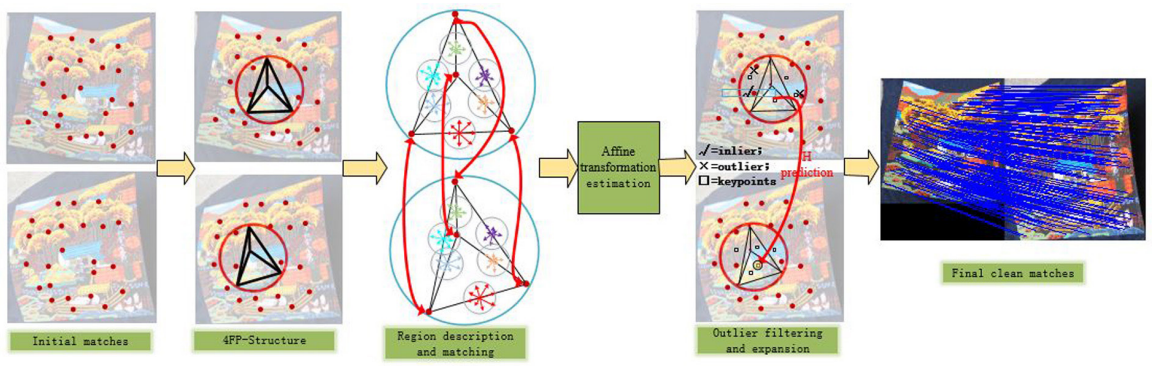

Figure 1. The schematic diagram of the proposed method. Given a set of putative correspondences, we first construct a local region structure, i.e., 4FP-Structure, for each correspondence. The special 4FP-Structure is then described by a compact descriptor and an expansion stage is performed to extract as many good matches as possible.

regularization theory in Reproducing Kernel Hilbert Space (RKHS). These methods are time consuming and memory consuming because of the techniques they use. To focus on both correctness and efficiency, Li *et al.* propose a l_q-estimator (Li *et al.*, 2016; Li *et al.*, 2017c) and affine-invariants (Li *et al.*, 2017b) for outlier removal; Ma *et al.*, (2014) propose vector field consensus (VFC) and locally linear transforming (LLT) (Ma *et al.*, 2015b) for robust feature matching. VFC uses a vector field to estimate a consensus of inlier correspondences that follow a non-parametric geometrical transformation. This is formulated as a *posteriori* maximization problem of the Bayesian model and solved using the Expectation Maximization (EM) algorithm. FastVFC (Ma *et al.*, 2014) reduces the complexity of original VFC; however, the iterative-based method is still slow when the number of tentative matches is large. Similar to VFC, LLT formulates this task as a maximum-likelihood estimation of a Bayesian model. In this model, the local geometrical constraint, represented by locally linear transforming, is used to preserve local structures. LLT is similar to the proposed method; however, it only uses the local geometrical information among neighboring features. In contrast, our method uses both geometrical information and texture information among neighboring feature points, making it more robust.

Todorovic and Ahuja (2008) proposed a region matching method by posing image matching as a tree matching problem. First, directed acyclic graphs (DAGs) were generated by merging adjacent nodes of the trees. Then, transitive closures of the DAGs were produced, and a bijection between the two transitive closures on DAGs was established. Thus, the matching is to search a maximum subtree isomorphism between the transitive closures. Similar to our method, both geometric and photometric properties are considered. Different from ours, their method is a global region matching method which identifies the largest region with similar information between an image pair.

Closer to our 4FP-Structure descriptor is the KVLD (Liu and Marlet, 2012) method, which also uses virtual lines that join neighboring feature points. KVLD uses k local line matches as photometric constraints to identify inliers or outliers. The main limitation is that KVLD is sensitive to repetitive texture. Differently, we make extensions to KVLD that we consider both photometric and geometric constraints in our method. We define a more compact structure based on local lines called 4FP-Structure descriptor, which not only inherits the advantages of KVLD, but also represents a local region. Local geometric constraint can be efficiently established based on the proposed 4FP-Structure; thus, verification step can be easily adapted to improve the robustness to repetitive texture. In addition, local affine expansion can help us to find as many high location accuracy correspondences as possible, which is important to sparse 3D reconstruction and TIN surface model.

Local Region Matching

In this section, we will detail our method, local region matching, in the order of the three stages 4FP-Structure construction, 4FP-Structure description and correspondence expansion (Figure 1). The main contribution is the proposed 4FP-Structure region descriptor. 4FP-Structure descriptor is based on line segments which are described by a KVLD-like (Liu and Marlet, 2012) method. It is robust to local geometric distortions and repetitive texture. Benefit from the proposed descriptor, local affine geometric constraints can be efficiently adapted to remove outliers and find more good matches. We assume that initial matches and keypoints are given as the input of our algorithm, e.g., correspondences generated by SIFT. Certainly,

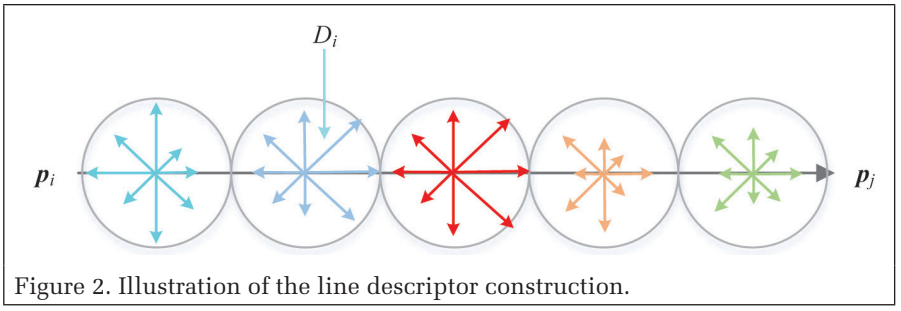

Figure 2. Illustration of the line descriptor construction.

the performance of the proposed approach does not rely on the particular method used to extract initial feature correspondences. This will be verified in the experiment section.

4FP-Structure Construction

We define a structure formed by four neighborhood feature points, i.e., 4FP-Structure, to represent a local region. The 4FP-Structure is composed of six connecting lines. To describe this special structure, we first introduce a simple line descriptor similar to KVLD (Liu and Marlet, 2012).

Line Descriptor

Given a straight line $l_{\boldsymbol{p}_i\boldsymbol{p}_j}$ with length d in image I, we first divide it into m small segments S of equal length $len = d/m$, the descriptor will be computed from the circular local support region D_i with radius len / 2 centered at each segment (Figure 2). This is necessary because the distortion in a small support region is not so severe. For each local support region D_i, the line direction $\boldsymbol{d}_l$ is assigned as its dominant orientation to make the descriptor rotation invariant. According to $\boldsymbol{d}_l$ and its clockwise orthogonal direction $\boldsymbol{d}_\perp$, pixels in region D_i are resampled to a local coordinate frame $F(x,y)$ whose row direction is $\boldsymbol{d}_l$ and origin is the center of D_i. Then, we can convolve the local frame $F(x,y)$ with a Gaussian weight function, $G(x,y)$, along direction $\boldsymbol{d}_\perp$ to produce a weighted frame $F'(x,y)$, as in the LBD (Zhang and Koch, 2013) and SIFT.

$$F'(x,y) = G(x,y) * F(x,y) \tag{1}$$

where * is the convolution operation in x and y, and $G(x,y) = (1 / \sqrt{2\pi}\,\sigma)e^{-y^2/2\sigma^2}$, in which $\sigma = len$ / 2 is a scale factor. The purpose of generating the weighted frame is to give more emphasis to gradients that are close to the line segment and less to those are far from it.

As mentioned previously, the presented line descriptor is invariant to rotation and scale changes. To make it robust to illumination and viewpoint changes, a SIFT-like gradient histogram is introduced. Each segment $s \in S$ can be described by sampling the orientations and magnitudes of the pixel gradients in the weighted frame of s_i . Thus, a histogram representing the rough spatial structure of the local region with eight orientation bins can be built. After computing the descriptors of all segments, denoted by D_{seg}, concatenate them to form an 8 × m-element vector:

$$\begin{aligned} &\boldsymbol{d}_{line}(l_{\boldsymbol{p}_i\boldsymbol{p}_j}) = (\boldsymbol{d}_{seg}(s_1)^T, \boldsymbol{d}_{seg}(s_2)^T, \ldots, \boldsymbol{d}_{seg}(s_m)^T), \\ &s.t.\ \boldsymbol{d}_{seg}(s_i)^T \in D_{seg}(i = 1, 2, \ldots, m) \end{aligned} \tag{2}$$

where $\boldsymbol{d}_{seg}(s_i)$ stands for the description of segment s_i and $\boldsymbol{d}_{line}(l_{\boldsymbol{p}_i\boldsymbol{p}_j})$ is the line descriptor.

4FP-Structure Configuration

Suppose correspondences (P,Q) with both inliers and outliers are provided by SIFT, where P represents a set of feature points in image I_1 and Q represents the corresponding matched points of P in image I_2 (Figure 3). For any matched pair $(\boldsymbol{p}_i,\boldsymbol{q}_i)$

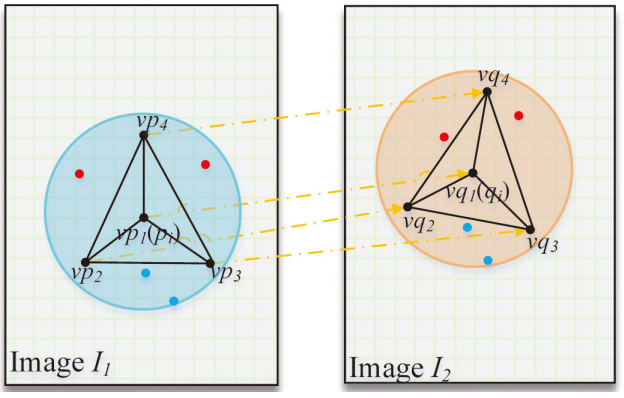

Figure 3. 4FP-Structure configuration. For a pair of correspondence $\boldsymbol{p}_i$, $\boldsymbol{q}_i$, the neighborhoods of $\boldsymbol{p}_i$ are firstly found (features inside the green circular area), then, other three suitable matches are selected by performing line matching algorithm and following some predefined principles, and the 4FP-Structures can be obtain ed by connecting the points each other. See text for more details.

from (P,Q), the goal of 4FP-Structure is to transform the description of $(\boldsymbol{p}_i,\boldsymbol{q}_i)$ into a more compact one in which regional texture information and local geometric consistency are considered and to determine $(\boldsymbol{p}_i,\boldsymbol{q}_i)$ is a good correspondence.

For each feature point $\boldsymbol{p}_i$ in set P, the following procedure is performed to generate a local region represented by a 4FP-Structure $4FPS$ $(\boldsymbol{p}_i)$. We first search the neighborhoods of $\boldsymbol{p}_i$, denoted by $N_{\boldsymbol{p}_i}$. If the Euclidean distance between $\boldsymbol{p}_j$ and $\boldsymbol{p}_i$ is less than the search radius r, $\boldsymbol{p}_j$ is considered to be a neighbor of $\boldsymbol{p}_i$:

$$N_{\boldsymbol{p}_i} \equiv \{\boldsymbol{p}_j \mid \boldsymbol{p}_j \in P\} \; s.t. \; \|\boldsymbol{p}_j - \boldsymbol{p}_i\|_{i\neq j} < r. \tag{3}$$

Meanwhile, the neighborhoods $N_{\boldsymbol{p}_i}$ of $\boldsymbol{q}_i$ in set Q are also found because of the correspondence relationship between point sets P and Q. For efficiency, point set P should be organized by a more powerful data structure such as KD-Tree (Zhou et al., 2008). After that, line segments $L_{\boldsymbol{p}_i}(L_{Q_i})$ are formed by linking neighborhoods $N_{\boldsymbol{p}_i}(N_{Q_i})$ with $\boldsymbol{p}_i(\boldsymbol{q}_i)$ and described by the aforementioned line descriptor. Line segments $l_{\boldsymbol{p}_i,\boldsymbol{p}_j} \in L_{\boldsymbol{p}_i}$ and $l_{\boldsymbol{q}_i,\boldsymbol{q}_j} \in L_{\boldsymbol{q}_i}$ are unlikely to be a pair of correct line correspondence unless both $(\boldsymbol{p}_i,\boldsymbol{q}_i)$ and $(\boldsymbol{p}_j,\boldsymbol{q}_j)$ are good matches. Thus, we assume that $(\boldsymbol{p}_j,\boldsymbol{q}_j)$ is a possible correct match when the distance between $l_{\boldsymbol{p}_i,\boldsymbol{p}_j}$ and $l_{\boldsymbol{q}_i,\boldsymbol{q}_j}$ is below a certain threshold τ:

$$(C_{\boldsymbol{p}_i}, C_{\boldsymbol{q}_i}) \equiv \{(\boldsymbol{p}_j,\boldsymbol{q}_j) \mid \boldsymbol{p}_j \in P, \boldsymbol{q}_j \in Q\} \tag{4a}$$

$$s.t. \begin{cases} \|\boldsymbol{p}_j - \boldsymbol{p}_i\| < r \\ \left\| \boldsymbol{d}_{line}\left(l_{p_i,p_j}\right) - \boldsymbol{d}_{line}\left(l_{q_i,q_j}\right) \right\| < \tau \end{cases} \tag{4b}$$

where $(C_{\boldsymbol{p}_i}, C_{\boldsymbol{q}_i})$ is a potential good correspondence set whose elements are the neighbors of $(\boldsymbol{p}_i,\boldsymbol{q}_i)$. If $(C_{\boldsymbol{p}_i}, C_{\boldsymbol{q}_i})$ is empty, $(\boldsymbol{p}_i,\boldsymbol{q}_i)$ is discarded as an outlier. Otherwise, we select another three points from $C_{\boldsymbol{p}_i}$ with $\boldsymbol{p}_i$ as the vertexes of the structure, denoted by $V(\boldsymbol{p}_i)$, and connect them to configure $4FPS$ $(\boldsymbol{p}_i)$ (Figure 3); the configuration details are summarized in Algorithm 1. There are two principles for selecting these vertexes: (1) motivated by the Delaunay (Cignoni *et al.*, 1998) triangulation algorithm, the minimum angle of the triangle formed by $(\boldsymbol{vp}_2, \boldsymbol{vp}_3, \boldsymbol{vp}_4)(\boldsymbol{vp}_i \in V(\boldsymbol{p}_i))$ should not be small, and $\boldsymbol{p}_i$ should be near the center of the triangle. This is to avoid skinny triangles in the 4FP-Structure. The three vertexes of a skinny triangle tend to be nearly collinear, which may cause degenerated expression of the local region, e.g., narrow and long. Such degenerated expression will make the expansion stage difficult since the affine transformation will not be as reliable as usual. (2) Selected points should be far from one another so that the constructed structure better represents the circular local region. Suppose that an affine transformation is computed based on four nearby correspondences. It well models the relationship between images I_1 and I_2 in the envelope area of these correspondences. However, if the distortion of the image pair is strong and cannot be ignored even in the small circular local region, the transformation will be not suitable for the area outside the envelope but inside the circular local region.

Algorithm 1: 4FP-Structure configuration

Input: correspondence set (P,Q)
Output: 4FP-Structures

```
Build KD-Tree of P;
repeat:
  for each point p_i ∈ P do
      search N_{p_i}(N_{q_i}), link with p_i → L_{p_i}(L_{q_i})
      line matching;
  for each neighbor do
      if ‖ d_line(l_{p_i,p_j}) − d_line(l_{q_i,q_j}) ‖ <τ do
          (p_j, q_j) → (C_{p_i}, C_{q_i});
  if (C_{p_i}, C_{q_i}) is empty do
      (p_i, q_i) → outlier;
  else if the number of (C_{p_i}, C_{q_i}) ≤3 do
      increase the search radius
      return line 3;
  else do
      pick other 3 suitable points → V(p_i), V(q_i)
      link each other V(p_i), V(q_i) → 4FPS(p_i), 4FPS(q_i)
```

4FP-Structure Description

A 4FP-Structure contains six line segments. We first describe these segments and then concatenate their descriptors by order to form a $6 \times 8 \times$ m-dimensional vector $\boldsymbol{d}_{4FPS}$. We use Euclidean distance as the similarity metric for matching the descriptors. $4FPS(\boldsymbol{p}_i)$ and $4FPS(\boldsymbol{q}_i)$ are identified as a reliable correspondence when the matching score is high (the Euclidean distance is small), and all four point correspondences in the pair of structures are accepted as inliers to form a base $B(\boldsymbol{p}_i, \boldsymbol{q}_i)$, which is used to compute the geometrical relationship between the circular local regions of $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$:

$$B(\boldsymbol{p}_i,\boldsymbol{q}_i) \equiv \left\{ \begin{matrix} \left\{ (\boldsymbol{vp}_1,\boldsymbol{vq}_1),\ldots,(\boldsymbol{vp}_4,\boldsymbol{vq}_4) \right\} \\ \boldsymbol{vp}_i \in V(\boldsymbol{p}_i), \boldsymbol{vq}_i \in V(\boldsymbol{q}_i) \end{matrix} \right\} s.t. \left\| \boldsymbol{d}_{4FPS}(\boldsymbol{p}_i) - \boldsymbol{d}_{4FPS}(\boldsymbol{q}_i) \right\| < \tau \tag{5a}$$

In the above, we have introduced the steps of 4FP-Structure construction. In fact, during the construction of this

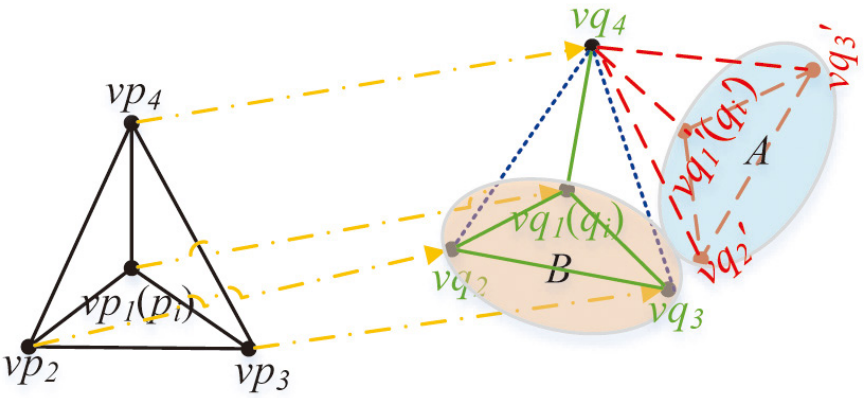

Figure 4. An example to explain the robustness of our local region descriptor. The proposed method can easily filter the outliers, even if some matches have similar textures (three false matches have similar textures to the correct matches in this case). See text for more details.

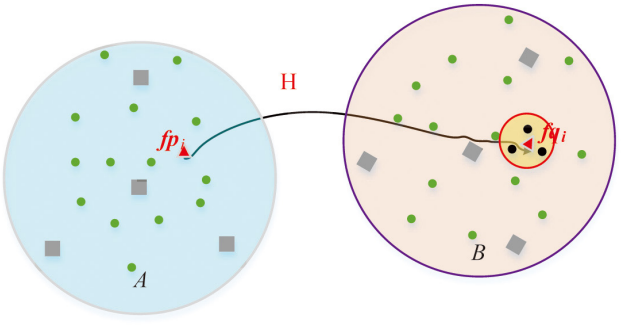

Figure 5. Match exploration.

structure, most outliers can be filtered by matching the line segments because the matching score is high only if both vertexes of the line segment are good correspondences. That is, if the vertexes $\boldsymbol{p}_i$ of $L_{\boldsymbol{p}_i}$ and $\boldsymbol{q}_i$ of $L_{\boldsymbol{q}_i}$ are mismatched, it is hard to find even one reliable line match for configuring the structure. However, repetitive texture may reduce the power of the line matching method. Consider the following special case: points ($\boldsymbol{q}'_i$, $\boldsymbol{vq}'_2$, $\boldsymbol{vq}'_3$, $\boldsymbol{vq}'_4$) are the true matches of ($\boldsymbol{p}_i$, $\boldsymbol{vp}_2$, $\boldsymbol{vp}_3$, $\boldsymbol{vp}_4$); area A (rendered in cyan) and area B (rendered in orange) have the same texture (Figure 4). Because of the repetitive texture, ($\boldsymbol{p}_i$, $\boldsymbol{vp}_2$, $\boldsymbol{vp}_3$, $\boldsymbol{vp}_4$) are matched with ($\boldsymbol{q}_i$, $\boldsymbol{vq}_2$, $\boldsymbol{vq}_3$, $\boldsymbol{vq}_4$) by SIFT. Moreover, if the pixel information around lines $L_{\boldsymbol{q}_i,\boldsymbol{vq}_4}$ and $L_{\boldsymbol{q}'_i,\boldsymbol{vq}_4}$ is similar, the line matching algorithm will construct a false corresponding structure $4FPS(\boldsymbol{q}_i)$ of $4FPS(\boldsymbol{p}_i)$. In this case, especially for deformable images, the proposed approach is much more powerful than most current algorithms (including line- and point-matching methods and RANSAC-like post-processing algorithms) since additional information ($L_{\boldsymbol{vq}_2,\boldsymbol{vq}_4}$, $L_{\boldsymbol{vq}_3,\boldsymbol{vq}_4}$) is considered in our novel descriptor for distinguishing $4FPS(\boldsymbol{q}_i)$ from $4FPS(\boldsymbol{q}'_i)$. In addition, our region matching algorithm can handle the extreme case that all four matches of the constructed structures are false correspondences with similar information. In other words, the four correspondences are all in the repetitive texture region B and all are false matched. We will illustrate how our method works in the following section.

Correspondence Expansion

So far, we have obtained a set of local region correspondences that each contains a four-match base. To model the relationship of each correspondence, affine transformation is chosen. The reason why we prefer affine over other rigid transformations (such as similarity) is that it has the ability to resist distortion in a small local region (Guo and Cao, 2012). Three non-collinear matches are sufficient for computing the affine matrix since it only has six degrees of freedom. However, for estimation accuracy, redundant observations are considered. Moreover, using four matches to compute the affine matrix also allows the correctness of the constructed 4FP-Structure to be verified: if the 4FP-Structure is constructed incorrectly, the residuals of some vertexes will be very large.

Figure 5 gives an example to explain the procedure of expansion under the affine constraint. Areas A and B are a pair of local region correspondence, which is represented by the 4FP-Structure whose vertexes $V(\bullet)$ are denoted by gray squares in the figure. The relationship of the base $B(\boldsymbol{p}_i, \boldsymbol{q}_i)$ is modeled as follows:

$$\boldsymbol{X}_B = \boldsymbol{H}_{affine}\boldsymbol{X}_A \tag{6}$$

where $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$ are matrixes whose columns are the homogeneous coordinate vectors of $V(\boldsymbol{p}_i)$ and $V(\boldsymbol{q}_i)$, respectively. $\boldsymbol{H}_{affine}$ is an affine transformation matrix. Ideally, $\boldsymbol{vp}_j \in V(\boldsymbol{p}_i)(j = 1,2,3,4)$ from region A is projected into $\boldsymbol{vq}_j \in V(\boldsymbol{q}_i)(j = 1,2,3,4)$ in region B when the affine transformation $\boldsymbol{H}_{affine}$ is performed. Once we obtain the affine matrix of a local region correspondence, the point matches inside the region can be classified into inliers and outliers by checking if they support the transformation model. In fact, each match $(\boldsymbol{p}_i, \boldsymbol{q}_i) \in (P,Q)$ can only be assigned one property, inlier or outlier. If some matches are classified as inliers in one region and outliers in another region, these matches will be treated as unreliable. We will increase the area of their local regions and reconstruct the 4FP-Structure to identify these unreliable matches. Because of this strategy, our method has the ability to address some difficult extreme cases, such as the case shown in Figure 6. In this case, the vertexes of structure $4FPS(\boldsymbol{p}_i)$ are all in the repetitive texture region (orange circular area in Figure 6a) and are all falsely matched, i.e., $(\boldsymbol{p}_i \rightarrow \boldsymbol{q}_i, \boldsymbol{vp}_2 \rightarrow \boldsymbol{vq}_2, \boldsymbol{vp}_3 \rightarrow \boldsymbol{vq}_3, \boldsymbol{vp}_4 \rightarrow \boldsymbol{vq}_4)$; furthermore, these matches have almost the same information as the true matches, so the match $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ is classified as an inlier. Suppose that there is another region (cyan circular area in Figure 6b) that also contains match $(\boldsymbol{p}_i, \boldsymbol{q}_i)$. The match will be classified as an outlier if the affine matrix of this region is computed correctly. As a result, match $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ has two properties, both inlier and outlier, so match $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ will be treated as an unreliable correspondence that needs to be identified further.

In addition, we implement a match-exploration stage for the remaining SIFT keypoints (feature points unmatched by SIFT) to extract as many good matches as possible. In Figure 5, we use *FP* and *FQ* to denote the feature points of region *A* and

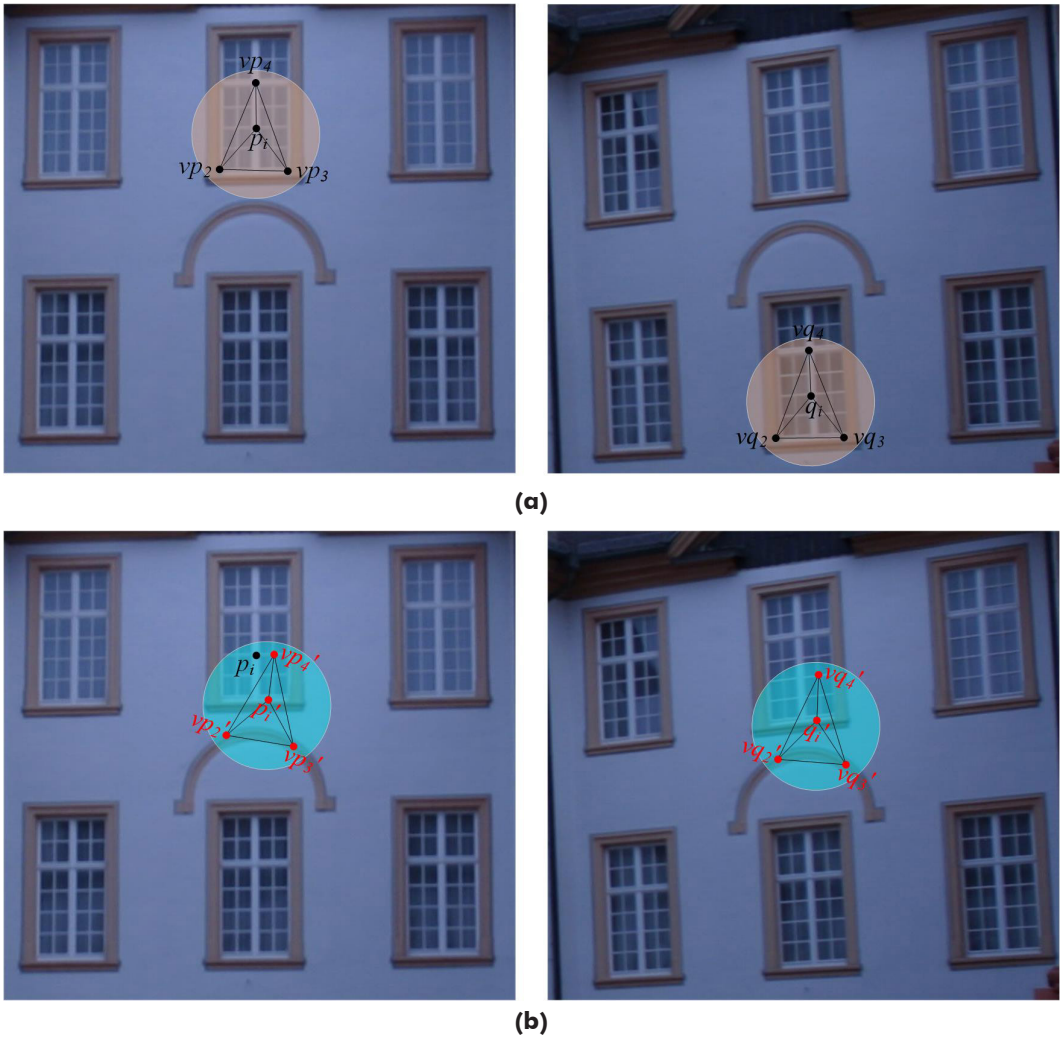

Figure 6. An extreme case involving repetitive texture. See text for more details.

B, respectively. For each feature point $\boldsymbol{fp}_i \in FP$ inside region A, the ideal location of its matching point $\boldsymbol{fq}_i'$ in region B can be predicted through $\boldsymbol{H}_{affine}$ (Figure 5). In practice, the model noise cannot be ignored, and projection residuals exist. A candidate circular area with radius R around $\boldsymbol{fq}'$(small orange area in Figure 5) is introduced. The features $\boldsymbol{fq}_j \in FQ$ inside the circle (black dots) are defined as candidate matches of $\boldsymbol{fp}_i$. We use the formulation in Equation 7) to measure the similarity between $\boldsymbol{fp}_i$ and its candidate match $\boldsymbol{fq}_j$:

$$dist_{ij} = e^{-R/d_{ij}} \cdot \| \boldsymbol{d}_{SIFT}(\boldsymbol{fp}_i) - \boldsymbol{d}_{SIFT}(\boldsymbol{fq}_j) \| \quad (7)$$

where d_{ij} is the Euclidean distance between $\boldsymbol{fq}_j$ and the ideal match $\boldsymbol{fq}_i'$, $\boldsymbol{d}_{SIFT}(\boldsymbol{x})$ stands for the SIFT descriptor of feature $\boldsymbol{x}$, and $\|\cdot\|$ is the two-dimensional norm operator. The first term $e^{-R/d_{ij}} \in (0,1)$ is a weight function whose role is to give more emphasis to the candidate matches that are close to the ideal match $\boldsymbol{fq}_i'$. If the minimum distance $dist_{\min}$ is below τ, the corresponding candidate feature is accepted as the true match of feature $\boldsymbol{fp}_i$.

Experimental Evaluation

We evaluate the proposed algorithm on both rigid and non-rigid image datasets. All the experiments are performed on a laptop PC with an Intel Core i5-3210M 2.5 GHz CPU and 8 GB of RAM. In the Experimental Settings Section, we first describe the dataset information and experimental settings; in Parameter Settings Section and Robustness to Feature Extractor Section, we study the parameters and validate the robustness of the proposed method to different feature extractors, respectively; we then compare our method with six state-of-the-art algorithms in the next Section.. The robustness to noise and running time are reported in following Sections.

Experimental Settings

Table 1 gives the details of the experimental settings, including parameters, dataset information, algorithms for comparison and evaluation metrics. We use both rigid and non-rigid datasets for evaluation. The rigid Oxford dataset (Mikolajczyk and Schmid, 2005) contains eight categories that have different geometric and photometric transformations, such as blur, viewpoint change, zoom, rotation, illumination change,and JPEG compression (Figure 7**)**. Each sequence of the Oxford dataset contains six images with increasing variation, and the first image is matched with the others. As suggesting by Liu *et al.* (Liu and Marlet, 2012), matches whose projection error is less than five pixels under the ground-truth transformation are considered inliers. For the non-rigid experiment, 24 challenging image pairs without ground truth are collected, as shown in Figure 8.

To establish the ground truth, i.e., determine the correctness of each correspondence, we confirm the results artificially. The putative correspondences are determined by SIFT, ORB, or SURF. The ratio of the Euclidean distances of the

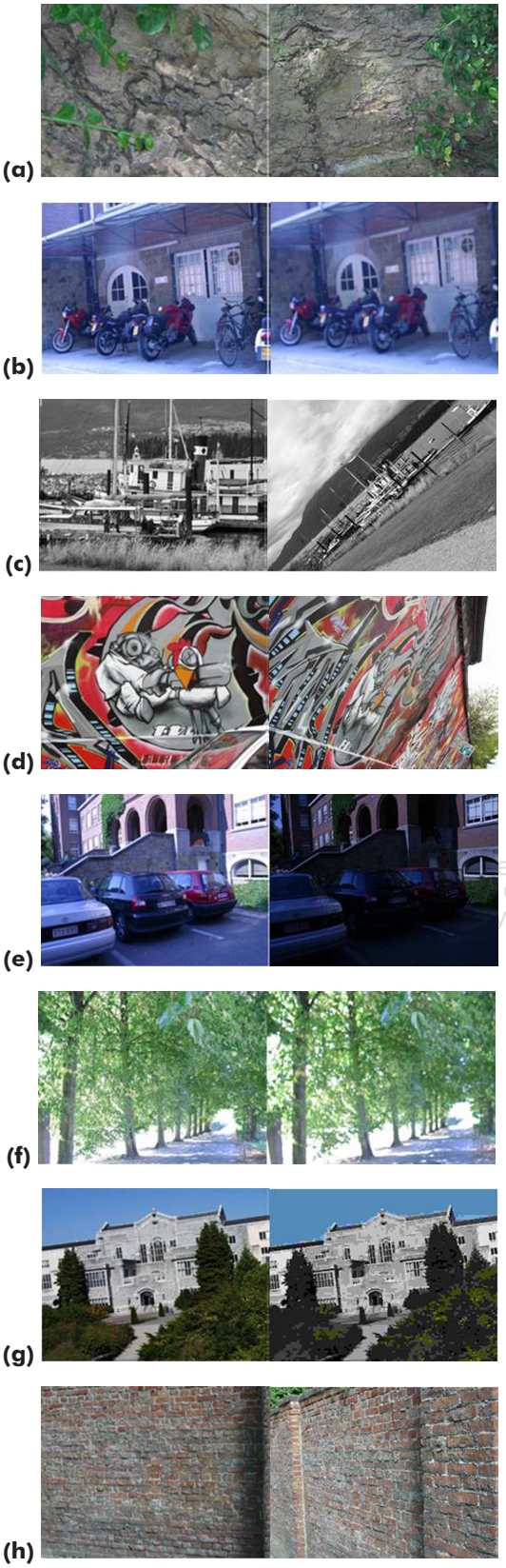

Figure 7. Example images of Oxford dataset. Each image dataset has different geometric and photometric transformation. For example, Bikes and Trees, blur; Graf and Wall, viewpoint change; Bark and Boat; zoom and rotation; Leuven, illumination change; UBC, JPEG compression.

Table 1. The details of experimental settings.

Settings	Information
Parameters	segments number: $m = 5$; neighborhood search radius: $r = \max(w,h)/20$, w and h stand for the image width and height, respectively; match score threshold: $\tau = 0.3$; expansion radius: $R = 3$.
Datasets	Rigid: the Oxford dataset (Mikolajczyk and Schmid, 2005), total 40 image pairs with ground truth (Figure 7); Non-rigid: 24 challenging image pairs, ground truth is established artificially (Figure 8).
Putative correspondence producer	ORB, opencv; SURF, opencv; SIFT, implemented by Lowe.
Methods for comparison Section *Experimental Evaluation Section*	ACC, Matlab & C++, open source; PGM+MPM, Matlab & C++, open source; FastVFC, Matlab, open source; LLT, Matlab, open source; KVLD, C++, open source; RANSAC, Matlab, open source;
Evaluation metrics	Number of correct matches: N_{total}; Matching correctness: $\text{Precision} = \frac{N_{correct}}{N_{total}}$ where N_{total} is the number of matches.

closest neighbor and the second-closest neighbor is set to 0.8. We compare the proposed method with six state-of-the-art algorithms whose source codes are publicly available. Similar to TCM (Guo and Cao, 2012), we use two evaluation metrics, i.e., the number of correct matches $N_{correct}$ and the matching correctness *Precision*. These two aspects are sufficient to evaluate the performances of the abovementioned algorithms because the feature detection scheme, initial matches and ground-truth rule are the same.

Parameter Study

We study the parameters τ, m, and r on the Oxford dataset. We use SIFT to generate the initial matches and refine these matches using the proposed method with different parameters. We perform three independent experiments in this section, in which each experiment has only one parameter as the variable and the others are constant. The details can be found in Table 2. We report the average $N_{correct}$ and *Precision* of all 40 image pairs in Figure 9.

From the plots, we can make the following observations: first, the highest $N_{correct}$ and the highest *Precision* cannot be achieved simultaneously. Higher $N_{correct}$ means lower *Precision* and vice versa. Second, $N_{correct}$ is proportional to τ and r and is inversely proportional to m. In contrast, *Precision* is proportional to m and is inversely proportional to τ and r. Third, the proposed method is robust to different parameter settings. The ranges of $N_{correct}$ and *Precision* are small. The lowest *Precision* and $N_{correct}$ values are 92.38 percent and 1698.35, respectively, while the initial *Precision* and $N_{correct}$ values are 74.15 percent and 1167.73 (Figure 9). With the analysis described above, we make a tradeoff between $N_{correct}$ and *Precision*. We fix $\tau = 0.3$, $m = 5$ and $r = \max(w,h)/20$ in the following experiments. For m and r, we do not use large values because large m and r will increase the computational complexity of the proposed method.

Robustness to Feature Extractor

In this section, we validate the robustness of the proposed method to different feature extractors on the Oxford dataset.

Figure 8. Non-rigid image pair dataset. This dataset contains 24 image pairs with different transformations.

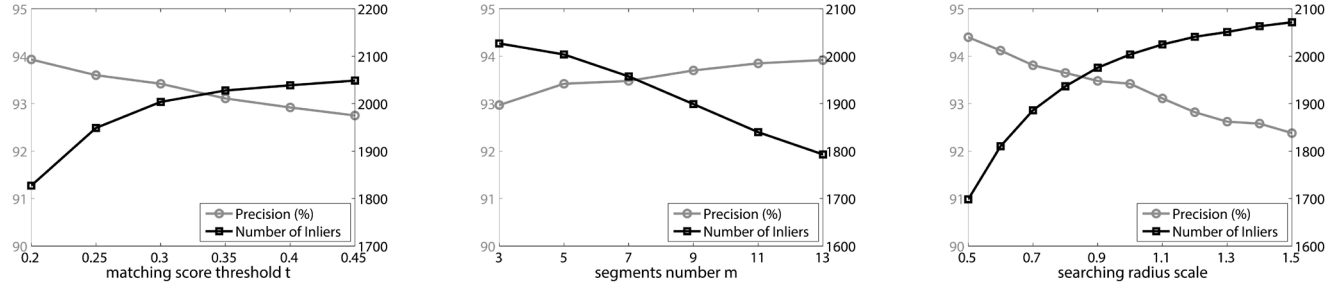

Figure 9. The results of parameters τ, m, and r. We perform three independent experiments, in which each experiment has only one parameter as the variable and the others are constant. The experiment setting details can be found in Table 2.

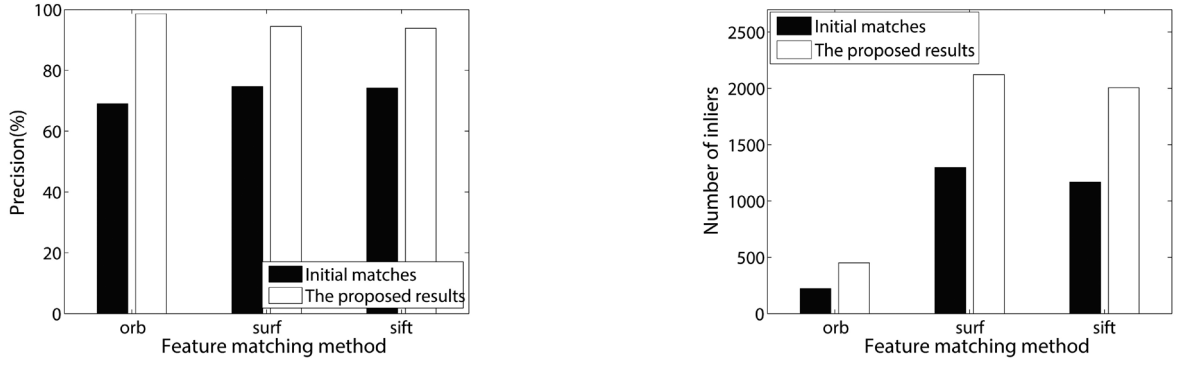

Figure 10. The results of different feature extractors. ORB, SURF, and SIFT are applied for initial feature matching on the Oxford dataset.

Table 2. The details of parameter settings.

Experiments	Variable	Fixed parameters
Parameter τ study	τ = [0.2, 0.25, 0.3, 0.35, 0.4, 0.45]	m = 5, r = max(w,h)/20, R = 3
Parameter m study	m = = [3, 5, 7, 9, 11, 13]	τ = 0.3, r = max(w,h)/20, R = 3
Parameter r study	r = a*max(w,h)/20, a = [0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5]	m = 5, τ = 0.3, R = 3

We use ORB, SURF, and SIFT to determine putative correspondences. The average values of $N_{correct}$ and *Precision* of all 40 image pairs (excluding failure cases) are shown in Figure 10.

As can be seen, the proposed method achieved 29.6 percent, 19.79 percent, and 19.68 percent growth rates of *Precision* compared with the initial matches of ORB, SURF and SIFT, respectively. It also achieved 103.18 percent, 63.49 percent and 71.74 percent growth rates of $N_{correct}$ compared with the initial matches of ORB, SURF and SIFT, respectively. According to this figure, the proposed method can achieve very impressive performance (*Precision* is higher than 93 percent) for all the three feature extractors. It seems that ORB is the best. However, the failure cases, i.e., the cases in which there are no correct matches in the putative correspondence set, are not included in the reported performance. The ORB feature extractor is sensitive to scale and viewpoint changes and has 11 failure cases among 40 image pairs, while SURF and SIFT only have 2 and 0 failure cases, respectively. We choose SIFT as the putative correspondence provider in the following sections because of its robustness and stability.

Individual Contribution of Single Step

Our matching method consists of three steps, i.e., 4FP-Structure matching, local affine verification, and local expansion. The first two steps improve the matching correctness *Precision* and the last one improves the number of correct matches $N_{correct}$. We perform an experiment to study the individual contribution of each single step on the non-rigid dataset. We first perform 4FP-Structure matching on initial SIFT matches (4FP-Structure); then, we add local affine verification step to clean the 4FP-Structure matching result (4FP-Structure & verification); finally, we adapt local expansion to find as many high location accuracy correspondences as possible (4FP-Structure & verification & expansion). The result of each image pair is shown in Figure 11 and Figure 12.

Compared to UNN (original SIFT matching), 4FP-Structure can largely improve the matching correctness *Precision*. However, it is not very robust to repetitive texture. For example, 4FP-Structure only gets 79.34 percent average *Precision* on image pair 18~22. A Verification step can effectively draw this issue. The average *Precision* on image pair 18 through 22 is increased to 94.02 percent. The Expansion step can help us to find as many high location accuracy correspondences as possible, which is important to sparse 3D reconstruction and TIN surface model. It extracts 407.33 more correct correspondences compared with UNN.

Figure 11. Number of correct matches measured on the non-rigid dataset for single step contribution study.

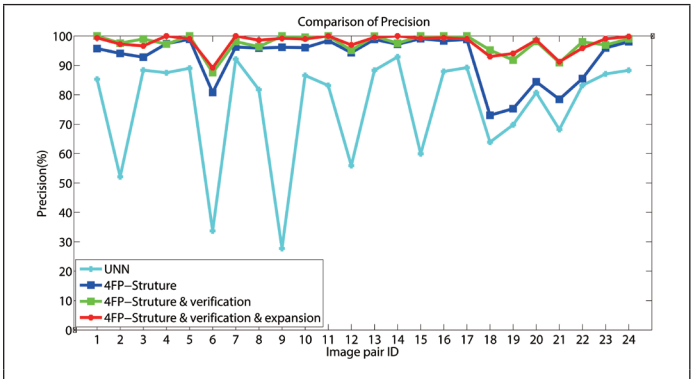

Figure 12. The matching correctness measured on the non-rigid dataset for single step contribution study.

Comparison with State-of-the-Art Methods

We compare the proposed approach with six state-of-the-art methods, i.e., PGM+MPM, ACC, FastVFC, LLT, KVLD, and RANSAC (for the rigid Oxford dataset only). We obtain the implementations of these methods from the authors' websites and use the same SIFT implementation to provide the putative correspondences. For PGM, we use MPM as the graph-matching module. For RANSAC, we use the homography model. The parameters are set according to the authors' recommendations. We also report the performance of the distance ratio method (UNN) (as a baseline) which provides the initial correspondences. Throughout all the experiments, the seven algorithms' parameters are all fixed. Note that we regard $N_{correct}$ and *Precision* as zero if the algorithm fails to match an image pair.

Rigid image matching: The results on the rigid Oxford dataset are summarized in Figure 13 and Figure 14. The average ($N_{correct}$, *Precision*) pair of each method is reported in Table 3. Our method significantly increases the number of correct matches, benefitting from the match expansion stage. It extracts 836 more correct correspondences than UNN, which ranks second in terms of $N_{correct}$. For matching correctness *Precision*, RANSAC achieves the best performance. This may be expected because the image pairs of the Oxford dataset are either of planar scenes or taken by pure rotation. The images, therefore, always obey the homography constraint. In addition, the inlier rates are high in most situations. PGM+MPM does not perform well on this dataset, which is just slightly better than the initial matches (UNN). It is sensitive to zoom and rotation changes. For some cases of the Bark and Boat sequences, PGM+MPM fails to extract even one good match. LLT and ACC have similar performance; KVLD performs well on this dataset. However, their accuracies are not as high as those of VFC, Our method and RANSAC. They are not very stable when the geometric or photometric transformation is severe. For example, the values of *Precision* of LLT, ACC, and KVLD on the fourth image pair of Graf are 30 percent, 0 percent, and 0 percent, respectively. FastVFC achieves very good performance. It performs as well as our method and RANSAC in most image

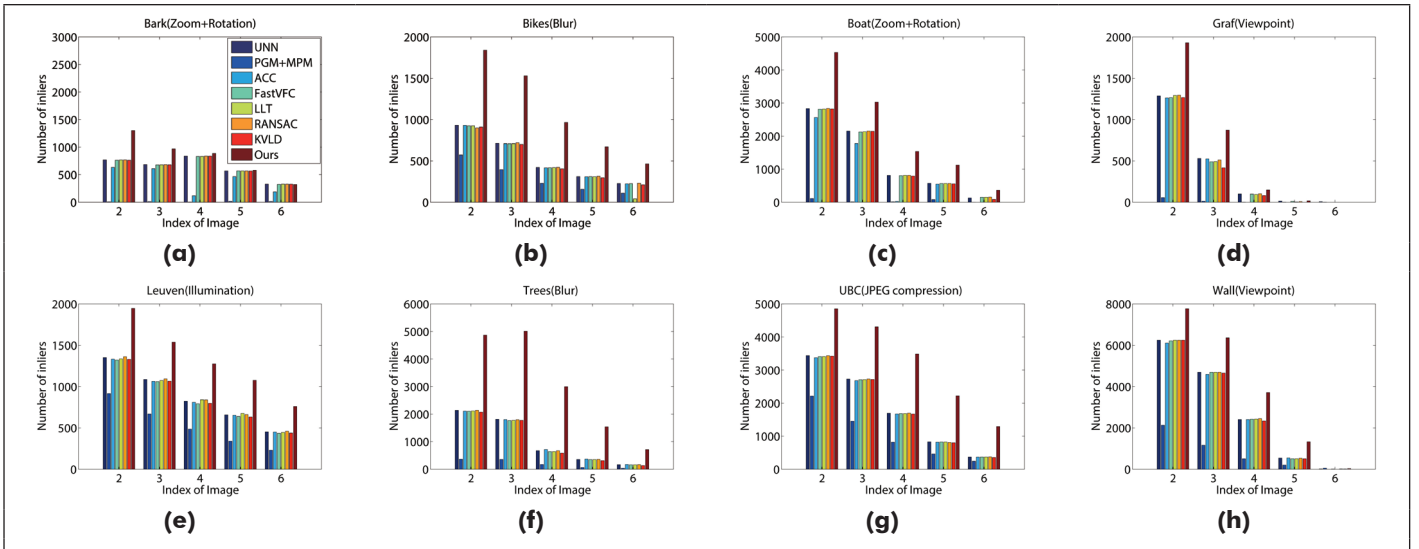

Figure 13. Number of correct matches measured on the Oxford dataset.

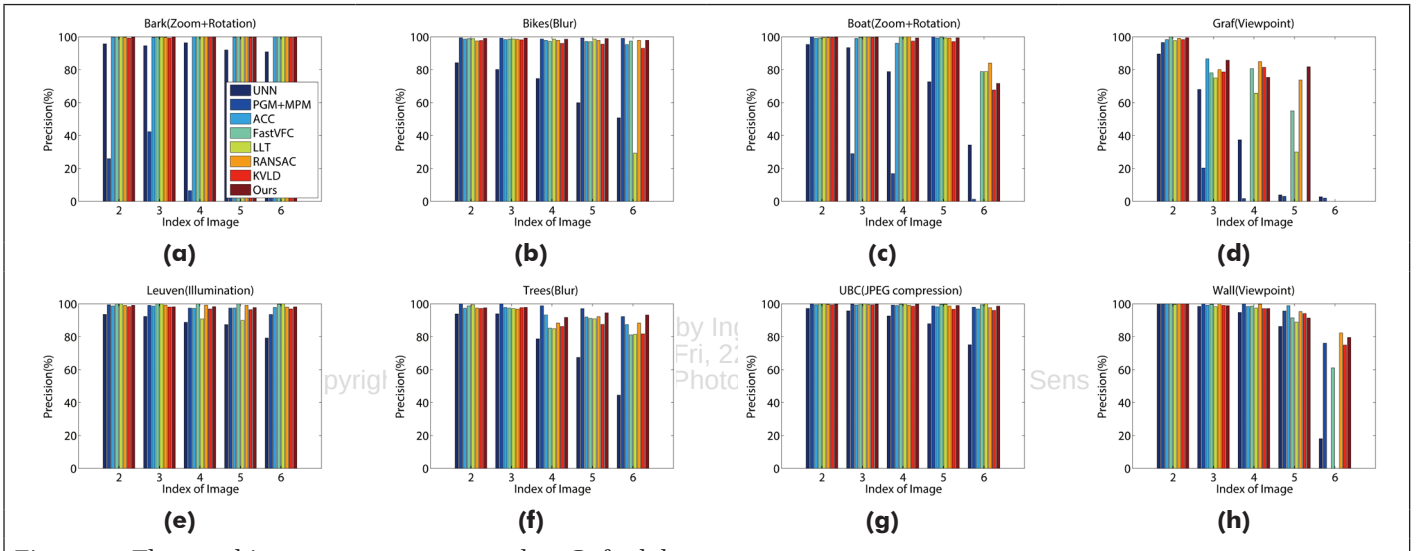

Figure 14. The matching correctness measured on Oxford dataset.

Table 3. Performance comparison on the Oxford dataset.

Metric	UNN	PGM+MPM	ACC	FastVFC	LLT	RANSAC	KVLD	Ours
$N_{correct}$	1167.73	367.03	1083.7	1154.9	1155.8	1162.18	1143.1	2003.7
Precision/%	74.15	75.04	85.3	92.09	87.23	93.5	89.78	93.42

pairs. However, it is worse than our method for extremely difficult cases. The *Precision* of our method ranks second and is only 0.08 percent lower than that of RANSAC. It achieves 8.12 percent, 1.33 percent, 6.19 percent, and 3.64 percent growth rates compared with ACC, FastVFC, KVLD, and LLT, respectively. Considering both $N_{correct}$ and *Precision*, the proposed method performs the best among these eight methods.

Non-rigid image matching: Figure 15 shows the results on five selected non-rigid image pairs in Figure 8. These five image pairs have different transformations, i.e., image pair 1 (multiple geometric models), image pair 9 (strong deformation + 180° rotation), image pair 11 (small deformation), image pair 12 (strong deformation + viewpoint change) and image pair 19 (deformation + repetitive texture). As shown, PGM+MPM performs well only if the transformation is simple. It is sensitive to multiple geometric models, strong deformations, large viewpoint changes, large rotations and repetitive texture. ACC achieves good precision on image pairs 1, 11, 12 and 19. However, it may fail when the rotation change is sufficiently large, such as in image pair 9. LLT is also very sensitive to large rotations. In addition, LLT becomes unreliable in cases with strong deformation and repetitive texture regions, such as image pair 12 and the white building of image pair 19. KVLD is sensitive to repetitive texture, for example, the *Precision* of KVLD on image pair 19 is only 69.6 percent. Both FastVFC and our method can achieve very impressive performance on all five image pairs. Our method is more robust than FastVFC for multiple geometric model transformations. For example, FastVFC and LLT extract two of the three cluster matches in image pair 1, while the proposed method extracts all three cluster matches.

Figure 16 and Figure 17 plot the results of $N_{correct}$ and *Precision* for each image pair in the non-rigid dataset, respectively. As can be seen, the proposed method achieves

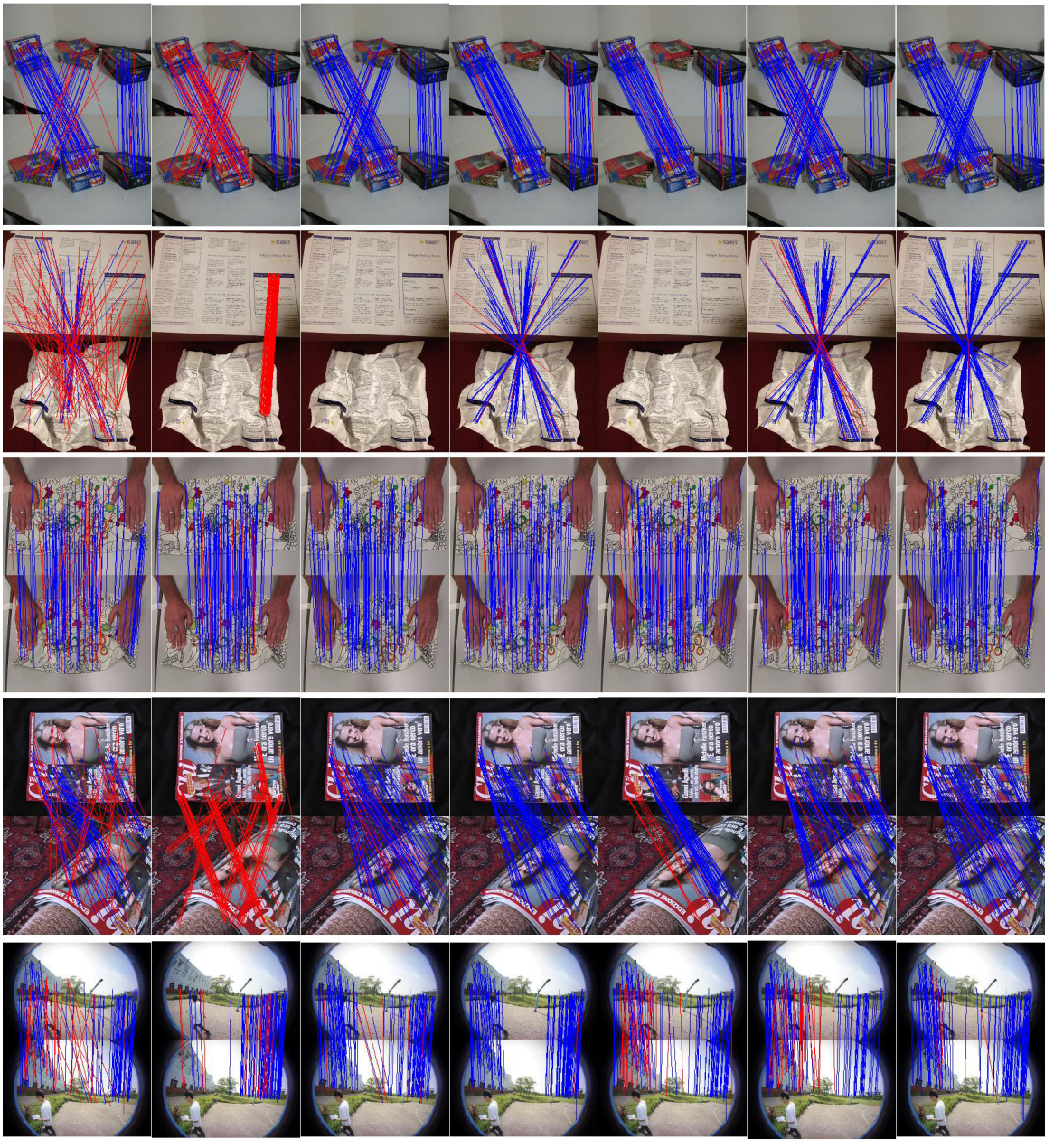

Figure 15. The results of five selected non-rigid image pairs in Figure 8. From left to right, columns are the results of image pair 1 (multiple geometric models), image pair 9 (strong deformation + 180° rotation), image pair 11 (small deformation), image pair 12 (strong deformation + viewpoint change) and image pair 19 (deformation + repetitive texture). For better visualization, no more than 100 randomly selected matches are plotted.

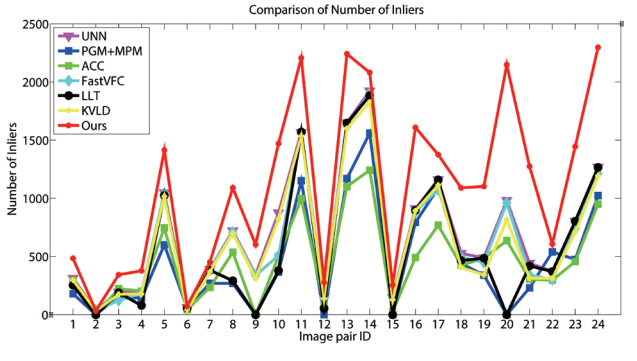

Figure 16. Number of correct matches measured on the non-rigid dataset.

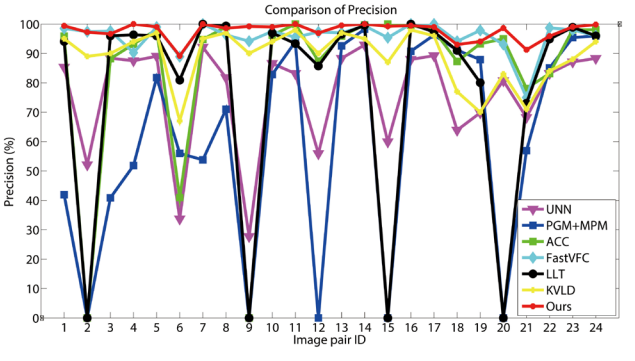

Figure 17. The matching correctness measured on the non-rigid dataset.

Table 4. Performance comparison on the non-rigid dataset.

Metric	UNN	PGM+MPM	ACC	FastVFC	LLT	KVLD	Ours
$N_{correct}$	690.5	452.33	460.5	644.71	569.75	628.68	1097.83
Precision/%	75.94	61	84.14	95.69	77.73	88.73	97.7

Table 5. Robustness to noise.

Metric	UNN	PGM+MPM	ACC	FastVFC	LLT	RANSAC	KVLD	Ours
number of matches	97.58	506.77	7.17	17.63	27.18	8.22	0.36	0.05
standard deviation	72.5	168.93	18.73	27.13	37.11	3.07	3.7	1.05

the best performance on both $N_{correct}$ and *Precision*, i.e., the red curves of our method are the highest for most image pairs. Our method has the ability to detect a larger number of correct matches than other state-of-the-art methods, while preserving extremely high detection correctness. The average ($N_{correct}$, *Precision*) pair of each method is reported in Table 4. Our method identifies 407.33 more correct correspondences compared with UNN. For matching correctness *Precision*, the matches obtained by PGM+MPM are even worse than the initial matches (UNN). This method is sensitive to all kinds of transformations. ACC performs well in most cases, except for large rotations. LLT becomes unreliable for image pairs with strong deformation, repetitive texture, and large rotation. FastVFC achieves very good results. It performs even better than our method in several cases. KVLD is sensitive to repetitive texture. Its performance largely decreases on image pair 18, 19, 20, 21, and 22 which contains many repetitive texture regions. Our method ranks first in terms of *Precision*. It achieves 13.56 percent, 2.01 percent, 19.97 percent, and 8.97 percent *Precision* growth rates compared with ACC, FastVFC, LLT, and KVLD, respectively.

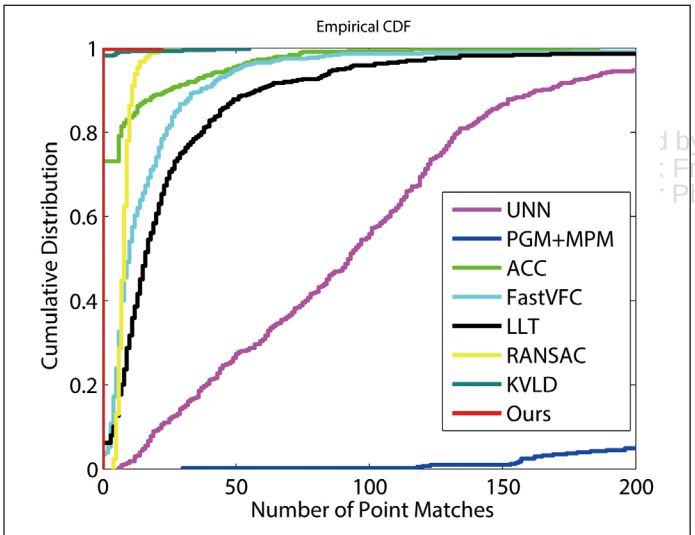

Figure 18. The cumulative distribution curves of noise image test.

Robustness to Noise

A good correspondence detection algorithm should be robust to noisy images, which is very important for some applications, such as image searching and scene recognition. Therefore, an additional experiment is performed to investigate the validity of the proposed method. We use 30 images, including the first image of each category in the Oxford dataset and the first image of 22 non-rigid image pairs in Figure 8 (non-rigid image pairs 4 and 13 are not used), to form $C_{30}^2 = 435$ image pairs for evaluation. There is no doubt that the ground-truth number of correct matches of each image pair is zero. The cumulative distribution curve of each method is reported in Figure 18, and the average (number of matches, standard deviation) pairs are displayed in Table 5. As shown, KVLD and the proposed method are the most robust among all the compared methods. ACC and RANSAC are better than FastVFC and LLT. PGM+MPM is much worse than even UNN. There is only one image pair that has a non-zero number of matches among the 435 image pairs for our method. This image pair is shown in Figure 19. As can be seen, the matched areas almost have the same information, including texture and geometric structure.

Computational Complexity and Running Time

When searching neighbors for each matching point in P using the K-D Tree, the time complexity is close to $O((\bar{K} + N) \log N)$, where $\bar{K}$ is the average number of neighbors and N is the number of initial matches. The time complexity of the line descriptor for each line is $O(N)$. Hence, the time complexity of 4FP-Structure construction is $O(\bar{K}N + N \log N)$. The time complexity of 4FP Structure description is $O(N)$. The time complexity of expansion is $O(\bar{M}N)$, where $\bar{M}$ is the average number of keypoints inside the local circular region. The total time complexity of the proposed method is $O((\bar{K}+\bar{M})N + N \log N)$.

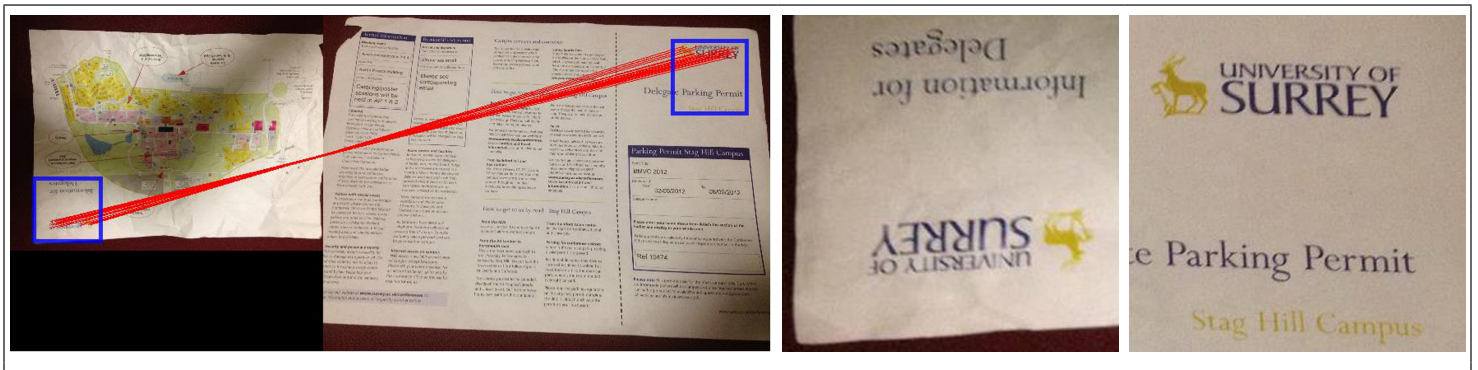

Figure 19. The only one image pair that our method has non-zero number of matches. Left is our matching result. Middle and right are the enlarged areas of the blue box in the left figure.

Table 6. Running time comparison on the Oxford dataset.

Metric	PGM+MPM	ACC	FastVFC	LLT	RANSAC	KVLD	Ours
mean	73.13	106.71	1.67	0.86	8.92	4.78	3.62
std	41.41	331.74	3.99	1.24	37.34	2.91	2.78
max	204.27	1942.8	25.6	7.02	171.69	15.48	11.36

Table 7. The comprehensive comparison of different methods.

metric	PGM+MPM	ACC	FastVFC	LLT	RANSAC	KVLD	Ours
rigid (Y/N)	Y	Y	Y	Y	Y	Y	Y
non-rigid (Y/N)	Y	Y	Y	Y	N	Y	Y
Precision (√)	√	√√	√√√	√√	√√√	√√	√√√
$N_{correct}$(√)	√	√√	√√	√√	√√	√√	√√√
robustness to transformations (√)	√	√√	√√√	√√	√√√	√√	√√√
robustness to noise (√)	×	√√	√	√	√√	√√√	√√√
efficiency (√)	√	√	√√√	√√√	√√	√√	√√

We measure the running times of each method on 40 image pairs of the Oxford dataset. FastVFC, LLT, and RANSAC are implemented in MATLAB, PGM+MPM, ACC, and Ours are implemented based on MATLAB&C++; KVLD is implemented in C++. There are many matrix operations in FastVFC and LLT; thus, rewriting them in MATLAB&C++ would likely not result in significant improvements in their running times. The average, maximum and standard deviation of the running time are summarized in Table 6. Although this is only a rough comparison because the codes are not implemented on a uniform platform, it can reflect many problems. As shown, our method is much more efficient than graph-based methods, i.e., PGM+MPM and ACC. The time complexity of FastVFC is $O(D^3N^3)$, which means that the running time of FastVFC increases quickly with the number of initial matches. For example, the running time of FastVFC on the first image pair of the Wall category (6268 initial matches) is 25.6 seconds. RANSAC becomes very slow on low inlier rate image pairs. For instance, the running time is up to 171 seconds for the second last image pair of the Graf category (3.95 percent inlier rate). KVLD contains scale space construction stage, thus, it is slightly slower than the proposed method. The standard deviation of the proposed method is only larger than that of LLT, whose time complexity is $O(K^3N + N\log N)$, where K is the parameter for the number of nearest neighbors. In addition, the implementation of the proposed method is very rough and could be greatly optimized.

Table 7 summarizes the comprehensive performances of these seven methods in several aspects, including suitability for rigid and non-rigid image pairs, *Precision*, $N_{correct}$, robustness to transformations, robustness to noise and efficiency.

Conclusions

In this paper, we proposed a region matching method for simultaneous outlier removal and good match maximization. We introduced a special and novel 4FP-Structure to describe this local region. In addition, an expansion stage to detect and select high location accuracy correspondences under a local affine transformation constraint was presented. The method is general and robust, able to handle both rigid and non-rigid image pairs in cases of severe outliers, and hence applicable to various vision tasks. We tested our method in various situations, including rigid scenes, non-rigid scenes and irrelative scenes. As shown in the results, the proposed method achieved impressive performance in terms of correct matches and correctness compared with other state-of-the-art methods. We believe the local region matching method introduced here is worth further research for applications such as structure from motion and 3D reconstruction from non-rigid images.

Acknowledgments

The authors would like to express their gratitude to the editors and the reviewers for their constructive and helpful comments for substantial improvement of this paper. This work was supported by National Natural Science Foundation of China (No. 41271452 and No. 41701528), the Fundamental Research Funds for the Central Universities (No. 2042017KF0235), the Key Technologies R&D Program of China (No. 2015BAK03B04), and the National High-tech R&D Program of China (863 Program) Grant No.2013AA102401.

References

Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool, 2008. Speeded-up robust features (SURF), *Computer Vision and Image Understanding*, 110:346–359.

Best, P.J., and N.D. McKay, 1992. A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239-256.

Brown, M., and D.G. Lowe, 2003. Recognising panoramas, *Proceedings of the International Conference on Computer Vision 2003*, pp. 1218.

Cho, M., and J. Lee., 2009. Feature correspondence and deformable object matching via agglomerative correspondence clustering, *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*,. IEEE, pp. 1280–1287.

Cho, M., and K.M. Lee, 2012. Progressive graph matching: Making a move of graphs via probabilistic voting, *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 398–405.

Cho, M., J. Sun, O. Duchenne, and J. Ponce, 2014. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers, 2014. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 2091–2098.

Chum, O., and J. Matas, 2005a. Matching with PROSAC-progressive sample consensus, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR, IEEE, pp. 220–226.

Chum, O., and J. Matas, 2005b. Matching with PROSAC-progressive sample consensus, 2005 *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, pp. 220–226.

Chum, O.,J. Matas, and J. Kittler, 2003. Locally optimized RANSAC, *Proceedings of the Joint Pattern Recognition Symposium*, Springer, pp. 236–243.

Cignoni, P., C. Montani, and R. Scopigno, 1998. DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed, *Computer-Aided Design*, 30:333–341.

Conte, D., P. Foggia, C. Sansone, and M. Vento, M., 2004. Thirty years of graph matching in pattern recognition, *International Journal of Pattern Recognition and Artificial Intelligence*, 18:265–298.

Duchenne, O., F. Bach, I.-S. Kweon, and J. Ponce, 2011. A tensor-based algorithm for high-order graph matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,33: 2383–2395.

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

Geiger, A., M. Lauer, F. Moosmann, B. Ranft, H. Rapp, C. Stiller, and J. Ziegler, 2012. Team AnnieWAY's entry to the 2011 Grand Cooperative Driving challenge, *IEEE Transactions on Intelligent Transportation Systems*, 13:1008–1017.

Guo, X., and X. Cao, 2012. Good match exploration using triangle constraint, *Pattern Recognition Letters*, 33:872–881.

Kato, T., S. Omachi, and H. Aso, 2002. Asymmetric gaussian and its application to pattern recognition, *Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, Springer, pp. 405–413.

Ke, Y., and R. Sukthankar, 2004. PCA-SIFT: A more distinctive representation for local image descriptors, , 2004. CVPR 2004. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, pp. II-506–II-513.

Li, J., Q. Hu, and M. Ai, 2016. Robust Feature Matching for Remote Sensing Image Registration Based on $ L_ {q} $-Estimator, *IEEE Geoscience and Remote Sensing Letters*, 13:1989–1993.

Li, J., Q. Hu, and M. Ai, 2017a. Robust feature matching for geospatial images via an affine-invariant coordinate system, *The Photogrammetric Record*, September 2017, pp. 317–331.

Li, J., Q. Hu, M. Ai, and R. Zhong, 2017b. Robust feature matching via support-line voting and affine-invariant ratios, *ISPRS Journal of Photogrammetry and Remote Sensing*, 132:61–76.

Li, J., Q. Hu, R. Zhong, and M. Ai, 2017c. Exterior orientation revisited: A robust method based on l q-norm, *Photogrammetric Engineering & Remote Sensing*, 83:47–56.

Liu, Z., and R. Marlet, 2012. Virtual line descriptor and semi-local matching method for reliable feature correspondence, *Proceedings of the British Machine Vision Conference 2012*, pp. 16.11–16.11.

Lourenço, M., J.P. Barreto, and F. Vasconcelos, 2012., *IEEE Transactions on srd-sift: Keypoint Detection and Matching in Images with Radial Distortion, Robotics*, 28:752–760.

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

Ma, J., W. Qiu, J. Zhao, Y. Ma, A.L. Yuille, and Z. Tu, 2015a. Robust estimation of transformation for non-rigid registration, *IEEE Transactions on Signal Processing*, 63:1115–1129.

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

Ma, J., H. Zhou, J. Zhao, Y. Gao, J. Jiang, and J. Tian, 2015b. Robust feature matching for remote sensing image registration via locally linear transforming, IEEE Transactions on Geoscience and Remote Sensing, 53:6469–6481.

Mikolajczyk, K.,and C. Schmid, 2005. A performance evaluation of local descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1615–1630.

Montemerlo, M., S. Thrun, D. Koller, B. Wegbreit, 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem, *Proceedings of the Innovative Applications of Artificial Intelligence Conference*, AAAI/IAAI, pp. 593–598.

Raguram, R., J.-M. Frahm, and M. Pollefeys, 2008. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus, *Proceedings of the European Conference on Computer Vision*, Springer, pp. 500–513.

Rublee, E., V. Rabaud, K. Konolige, and G. Bradski, 2011. ORB: An efficient alternative to SIFT or SURF, 2011 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, IEEE, pp. 2564–2571.

Snavely, N., S.M. Seitz, and R. Szeliski, 2006. Photo tourism: Exploring photo collections in 3D, *ACM Transactions on Graphics (TOG)*, Association for Computing Machinery, pp. 835–846.

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

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

Torresani, L., V. Kolmogorov, and C. Rother, 2008. Feature correspondence via graph matching: Models and global optimization, *Proceedings of the European Conference on Computer Vision*, Springer, pp. 596–609.

Wang, G., Z. Wang, Y. Chen, and W. Zhao, 2015. A robust non-rigid point set registration method based on asymmetric gaussian representation, *Computer Vision and Image Understanding*, 141:67–80.

Zhang, L., and R. Koch, 2013. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency, *Journal of Visual Communication and Image Representation*, 24:794–805.

Zhou, K., Q. Hou, R. Wang, B. and Guo, B., 2008. Real-time KD-tree construction on graphics hardware, *ACM Transactions on Graphics* (TOG), 27:126.