

Line Matching Leveraged By Point Correspondences

Bin Fan Fuchao Wu Zhanyi Hu

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, 100190, Beijing, China

{bfan, fcwu, huzy}@nlpr.ia.ac.cn

Abstract

A novel method for line matching is proposed. The basic idea is to use tentative point correspondences, which can be easily obtained by keypoint matching methods, to significantly improve line matching performance, even when the point correspondences are severely contaminated by outliers. When matching a pair of image lines, a group of corresponding points that may be coplanar with these lines in 3D space is firstly obtained from all corresponding image points in the local neighborhoods of these lines. Then given such a group of corresponding points, the similarity between this pair of lines is calculated based on an affine invariant from one line and two points. The similarity is defined on the basis of median statistic in order to handle the problem of inevitable incorrect correspondences in the group of point correspondences. Furthermore, the relationship of rotation between the reference and query images is estimated from all corresponding points to filter out those pairs of lines which are obviously impossible to be matches, hence speeding up the matching process as well as further improving its robustness. Extensive experiments on real images demonstrate the good performance of the proposed method as well as its superiority to the state-of-the-art methods.

1. Introduction

Feature matching is a fundamental task in computer vision and has been widely studied in the past decades. It can be widely used in many applications, such as object recognition [14], 3D reconstruction [17, 18, 22], robot navigation [1], structure from motion [2] and so on. Concerning feature matching, point matching has received much attention and various approaches have been proposed [5, 7, 9, 11, 12, 14, 15, 19, 23]. These approaches first construct a local descriptor to describe the neighborhood distribution of a point, then the point matching is conducted by comparing local descriptors. Through local descriptors, point matching becomes robust to changes of illumination, affine

transformation, scale as well as some extent of viewpoint changes. Since most keypoints, which are popularly used for matching, are not localized at edges, they fail to capture geometrical and structural information about the scene. In contrast, lines supply sufficient geometrical information and thus contain structural information about the scene and objects. Therefore, line matching is both desirable and even indispensable in many applications [1, 3, 18]. However, approaches for line matching are not many due to various inherent difficulties.

Besides the traditional challenges in point matching, there are other factors making line matching a more challenging task. These factors include: inaccurate locations of line endpoints, fragments of same line, less distinctive appearance of line segments, no global geometric constraint such as the epipolar constraint in point matching is available. Up to now, only a few methods are reported in the literatures for automatic line segment matching. Hartley [8] used the trifocal tensor to match lines across three views. Schmid and Zisserman [16] proposed to first find point correspondences in the matching lines by known epipolar geometry and then to average the cross-correlation scores of all corresponding points as the similarity measure of lines. Both of these two methods need to know the epipolar geometry in advance. Lourakis et al. [13] used 2 lines and 2 points to construct a projective invariant for matching planar surfaces with lines and points, their method is rather time consuming and can hardly be extended to non-planar scenes. Herbert et al. [4] proposed to match lines based on their appearance and their topological layout. Firstly line segments are matched based on the histograms of the neighboring color profiles, then the topological relations between all line segments are used to remove false matches as well as to find more matches. Since the matching propagation is an iterative process, this method is computationally expensive. Meanwhile, initial matches are found based on color histogram that makes it less robust to illumination and other image changes. Recently, Wang et al. [21] proposed a descriptor named MSLD for line matching. It is analogous to SIFT [14] since it defines a pixel support region for each

pixel in a line, then a histogram of image gradient is accumulated for each pixel support region. The final MSLD descriptor is composed of the mean and standard deviation of these histograms. It only relies on the neighboring appearance of line segment. However, it is less distinctive than local point descriptors since line segments usually lack of rich textures in their local neighborhood and the method may fail when encountering repeated textures such as building facades. What is more, the method can not handle the problem of scale changes. Wang et al. [20] used Line Signatures to match lines between wide-baseline images. They used angles and length ratios between lines, which are computed by endpoints, to describe a pair of line segments, and then do the line matching on the basis of pairs of line segments. Since the descriptor of a pair of line segments relies on the endpoints of line segments, their method may fail when the localization of endpoints is not accurate enough.

This paper proposes a novel method for automatic line matching between a pair of images. Since methods of point matching are much advanced and can be applied to many image pairs with high reliability, the proposed method uses result of point matching to leverage line matching. It explores an affine invariant from two points and one line and utilizes this invariant to match lines. The result of point matching obtained from keypoint matching are served as point correspondences for the computation of invariants. Although the invariant is only available for planar surface, it is easy to find points that are coplanar with a line in the neighborhood of the line. Therefore, it can be used in general non-planar scenes. Here are some underlying principles of our method: 1) 3D Lines always lie in plane edges which implies that at least one side of a line has points coplanar with it; 2) It is relatively easy to find at least two points coplanar with a line in its local neighbors given a set of corresponding points, thus can obtain at least one invariant; 3) In the local neighborhood region of line, the homography can be suitably approximated by affine transformation. Therefore, by directly comparing the affine invariants which are computed from lines and the points that lie in their neighborhood, one can conduct line matching. Extensive experiments have demonstrated that our proposed method can match lines with high accuracy under various conditions, such as scaling, rotation, illumination changes as well as some extent of view point changes. Meanwhile, our method is robust when the set of corresponding points obtained by keypoint matching contains outliers.

The rest of this paper is organized as follows. Section 2 introduces the proposed method of line matching by point correspondences on the basis of an affine invariant. The algorithmic process of our method is presented in Section 3 while experimental results are reported in Section 4. Finally, we conclude this paper in Section 5.

2. Line Matching by Point Correspondences

Given two sets of line segments extracted from a reference image and a query image, $\mathcal{L}_1 = \{l_1^1, l_1^2, \dots, l_1^M\}$ and $\mathcal{L}_2 = \{l_2^1, l_2^2, \dots, l_2^N\}$, and additionally a set of tentative point correspondences which is susceptible to mismatches (for example point correspondences obtained by SIFT matching), denoted as $\mathcal{C} = \{(x_1^i, y_1^i), (x_2^i, y_2^i), i = 1, 2, \dots, K\}$, where (x_1^i, y_1^i) is a point in the reference image and (x_2^i, y_2^i) is its corresponding point in the query image. The goal of this paper is to find matching lines in \mathcal{L}_1 and \mathcal{L}_2 through correspondences in \mathcal{C} . The resulting set of matching lines is represented as

$$\mathcal{LM} = \{(l_1^{g(i)}, l_2^{f(i)}), i = 1, 2, \dots, N_L\}$$

in which $(l_1^{g(i)}, l_2^{f(i)})$ represents a pair of matching lines and $g(i) \in [1, M], f(i) \in [1, N]$. Since local descriptors of keypoint usually contain orientation information to achieve rotation invariant, a coarse relationship of rotation between the reference and query images can be estimated from \mathcal{C} . Such a not-so-accurate rotation can be used to improve the speed and robustness of our line matching as we will describe later on.

2.1. The Affine Invariant

Considering a pair of image lines (l and l') which are related by an affine transformation H , and two pairs of point correspondences (X_1, X'_1) and (X_2, X'_2) which are also related by H . These points can be represented in homogeneous coordinates as $(\tilde{X}_1, \tilde{X}'_1)$ and $(\tilde{X}_2, \tilde{X}'_2)$. They satisfy

$$l' = sH^{-T}l \quad (1)$$

$$\tilde{X}'_i = H\tilde{X}_i, i = 1, 2 \quad (2)$$

where s is a scalar factor. We denote

$$D(X_1, X_2, l) = \frac{l^T \tilde{X}_1}{l^T \tilde{X}_2} \quad (3)$$

$$D(X'_1, X'_2, l') = \frac{l'^T \tilde{X}'_1}{l'^T \tilde{X}'_2} \quad (4)$$

Substituting Eq. (1) and Eq. (2) to Eq. (4), then the following equality holds

$$D(X_1, X_2, l) = D(X'_1, X'_2, l') \quad (5)$$

Eq. (5) means that the ratio of distances from point to the line is affine invariant. That is, $D(X_1, X_2, l)$ is an affine invariant. Therefore, it can be used for line matching. We call points (X_2, X'_2) as basis points and (X_1, X'_1) as reference points. Our method for line matching is based on the above affine invariant, as will described in the next section.

2.2. The Similarity of Two Lines

In order to match two lines l_1^m and l_2^n with the invariant described above, the first thing we need to do is to find out corresponding points from set \mathcal{C} that may be coplanar with these lines since they can be regarded as related by an affine transformation approximately in the neighborhoods of lines. Note that line segments detected in images seldom correspond to separated lines in real world. They are usually from edges of surfaces. Thus in the neighboring region on at least one side of a line, some coplanar points with the line can be found. The side of a line is defined by the gradient of line, which is defined as the average gradient of all the points on the line. If one point located on the region directed by the gradient of line, it is said to be on the right side of the line, otherwise the point is on its left side. For each side of a line, we seek corresponding points that are located in the neighborhood of this line from set \mathcal{C} . The neighborhood of a line is defined according to its length in order to be scale invariant. If the distance from a point P to a line l is smaller than $\alpha \times length(l)$ and the distance from P to the perpendicular bisector of l is smaller than $\beta \times length(l)$, P is said to be in the neighborhood of l . $length(l)$ is the length of line l , α and β are two ratios defining the local neighborhood of a line. Due to the inevitable localization errors of endpoints, such defined neighborhoods of two corresponding lines may not correspond exactly each other. However, our method needs only part of them corresponding since we just need to find some corresponding points in their neighborhoods. The correct corresponding points must locate on the corresponding subregion. In order to reduce the influence of noise and error detection of lines, only lines whose length are larger than 20 pixels are used for matching. In the experiments of this paper, α is set to 2.0 while β is set to 0.5. We denote

$$\mathcal{S}_l^{right} = \{P : P \in neigh(l) \& P \in \mathcal{C} \& P \in right(l)\}$$

and

$$\mathcal{S}_l^{left} = \{P : P \in neigh(l) \& P \in \mathcal{C} \& P \in left(l)\}$$

as the sets of points in the right and left neighborhood of l respectively, where $neigh(l)$ denotes the neighborhood of l while $right(l)$ and $left(l)$ denote right and left side of l respectively.

Given a pair of lines, we calculate their similarity by corresponding points in their neighborhood of right and left side separately. Then the maximum of these two values is taken as the final similarity of these two lines. Take right side for example. We assume that $\mathcal{G}^{right}(l_1^m, l_2^n) = \{(x_1^k, y_1^k) \in \mathcal{S}_{l_1^m}^{right}, (x_2^k, y_2^k) \in \mathcal{S}_{l_2^n}^{right}, k = 1, 2, \dots, N^r\}$ is the set of point correspondences lying in the right neighboring area of l_1^m in reference image and l_2^n in query image, where N^r is the number of corresponding points in the

supported local areas while (x_1^k, y_1^k) and (x_2^k, y_2^k) denote a pair of corresponding points. It should be noted that not all of these point correspondences are correct, some of them may be incorrect. In order to handle this problem, we select one pair of corresponding points $\{P_1^k = (x_1^k, y_1^k), P_2^k = (x_2^k, y_2^k)\}$ as the basis points of two compared invariants, then each of remaining $N^r - 1$ pairs of corresponding points are taken as reference points to calculate $N^r - 1$ similarities between their corresponding invariants. We get $N^r - 1$ similarities as below:

$$\text{sim}_k^{right}(i) = e^{-\|D(P_1^k, P_1^i, l_1^m) - D(P_2^k, P_2^i, l_2^n)\|}, \quad (6)$$

$$i \in [1, N^r], i \neq k$$

where $D(P_1^k, P_1^i, l_1^m)$ and $D(P_2^k, P_2^i, l_2^n)$ are affine invariants as defined by Eq. (3) and Eq. (4). Then we can get the median of them as:

$$\text{sim}_k^{right} = \text{median}\{\text{sim}_k^{right}(i), i \in [1, N^r], i \neq k\} \quad (7)$$

Choosing the median can improve the robustness to possible incorrect correspondences existing in $\mathcal{G}^{right}(l_1^m, l_2^n)$ while preserving discriminability. Finally, we take each $\{P_1^k, P_2^k\} \in \mathcal{G}^{right}(l_1^m, l_2^n)$ as the basis points so that we can get a set of medians: $\text{sim}_k^{right}, k = 1, 2, \dots, N^r$. The maximum is served as the similarity between l_1^m in reference image and l_2^n in query image with corresponding points in the right neighborhood. That is

$$\text{sim}^r(l_1^m, l_2^n) = \max_{k \in [1, N^r]} \{\text{sim}_k^{right}\} \quad (8)$$

In the similar way, we can get $\text{sim}^l(l_1^m, l_2^n)$. As we have said before, the final similarity of l_1^m and l_2^n is the maximum of these two similarities.

$$\text{SimL}(l_1^m, l_2^n) = \max\{\text{sim}^r(l_1^m, l_2^n), \text{sim}^l(l_1^m, l_2^n)\} \quad (9)$$

2.3. Fast Matching

The above proposed matching method uses point correspondences as additional information for line matching, and it can tolerate mismatches in the set of point correspondences by our defined similarity measure. The set of point correspondences can be easily obtained with keypoint matching, such as SIFT matching. In order to be rotation invariant, these keypoints are detected along with their orientations. Therefore, once correspondences have been obtained, one can use their orientations of corresponding points to estimate the coarse relationship of rotation between the reference and query images. As we have mentioned before, such a rotation relationship can be used to speed up the matching process as well as to improve its robustness.

For each pair of corresponding points in \mathcal{C} , we can calculate an angle between the point in query image and the

corresponding point in reference image. Then with these K (the number of corresponding points in \mathcal{C}) angles, a histogram of angles can be obtained. The angle corresponds to peak in the histogram is taken as the rotation relationship between the reference and query images. Note that in real images taken from different views, different points may have different rotation relationship. There does not exist a global rotation between two images. Thus there are errors when using the estimated relationship to approximate relationships for all pairs of corresponding points. Here, it is not used as a hard constraint, but only as a reference to find out those candidate pairs of lines that are obviously impossible to be matches. Suppose the estimated rotation is θ . For a pair of lines to be matched, if $|\theta' - \theta| > t_\theta$ in which θ' is the angle between them, they are considered to be a non-match without further calculating their similarity by Eq. (9). This can in practice filter out most of the non-matches, thus speed up the matching process. Here t_θ is the threshold to decide whether the angle between two lines violates θ too much. The lower the threshold is, the faster the matching process will be, but the number of matches will diminish. If it is set too high, then it will have no effect on speeding up. It turns out that t_θ setting is not a too delicate matter based on our experiments and it is set to be 20 degrees in all our experiments. Since the searching space for matching is reduced with such a fast matching strategy, the robustness of our method is also improved. Note that the pruning strategy for fast matching may fail when large viewpoint changes are involved in images since it relies on 2D inter-image rotation estimation. However, the proposed line matching method works well as long as the local homography can be reasonably approximated by an affine transformation, which is a basic assumption of our method. With the pruning strategy, the running time is around 3 times faster than that without it in our experiments.

3. The Algorithm

Given a reference image and a query image, we first extract line segments from them to obtain two sets of line segments $\mathcal{L}_1 = \{l_1^1, l_1^2, \dots, l_1^M\}$ and $\mathcal{L}_2 = \{l_2^1, l_2^2, \dots, l_2^N\}$. Secondly, a set of corresponding points $\mathcal{C} = \{(x_1^i, y_1^i), (x_2^i, y_2^i), i = 1, 2, \dots, K\}$ is obtained by a keypoint matching method. In the experiments of this paper, we used DAISY [23] to describe keypoints and then matched them by the nearest neighbor distance ratio (NNDR) [15]. The threshold for matching is set to 0.8. We then utilize the histogram technique described above to obtain θ from the set \mathcal{C} . All of these are served as input to the line matching algorithm along with threshold t_θ , which is the threshold used for fast matching as mentioned above. Then the line matching is done in keeping the uniqueness constraint. Here, the uniqueness of matching is meant that if l_1^i and l_2^j are a pair of matching lines, then there is no

other line in \mathcal{L}_2 such that its similarity to l_1^i is larger than $\text{SimL}(l_1^i, l_2^j)$ and there is no other line in \mathcal{L}_1 such that its similarity to l_2^j is larger than $\text{SimL}(l_1^i, l_2^j)$ too. It allows many-to-one correspondences which happens when a line in one image is split into multi-segments. The proposed method is outlined in Algorithm 1.

```

1: Function: LineMatching
2: Input: line set from reference image  $\mathcal{L}_1$ , line set from query image  $\mathcal{L}_2$ , corresponding point set  $\mathcal{C}$ , the estimated relationship of rotation between two images  $\theta$ , threshold for rotation  $t_\theta$ .
3: Initialization:  $\mathcal{LM} = \emptyset$ .
4: for each  $l_1^i \in \mathcal{L}_1$  do
5:   for each  $l_2^j \in \mathcal{L}_2$  do
6:     if  $|\text{angle}(l_1^i, l_2^j) - \theta| > t_\theta$  then
7:        $\text{SimL}(l_1^i, l_2^j) = 0$ 
8:     else
9:       Calculate  $\text{SimL}(l_1^i, l_2^j)$  according to Eq. (9)
10:    end if
11:  end for
12:  Find the line in  $\mathcal{L}_2$  with the largest similarity to  $l_1^i$  according to  $\text{SimL}(l_1^i, l_2^j), j \in [0, N]$ . Such a line is represented as  $l_2^k$ .
13:  if  $\text{SimL}(l_1^i, l_2^k) \geq 0.95$  then
14:    Check whether  $l_1^i$  is the line with the largest similarity to  $l_2^k$  in  $\mathcal{L}_1$ 
15:    if TRUE then
16:       $\mathcal{LM} = \mathcal{LM} \cup \{(l_1^i, l_2^k)\}$ 
17:    end if
18:  end if
19: end for
20: Output:  $\mathcal{LM}$ 

```

Algorithm 1: The proposed method for line matching.

4. Experimental Results

In this section we conducted three sets of experiments to demonstrate the effectiveness of our proposed method. First, we evaluated the proposed method on image pairs under various transformations. Then we tested the robustness of the proposed method to mismatches in the set of point correspondences. Finally, the proposed method was compared with some of the state-of-the-art methods. Figure. 1 shows the image pairs used for our experiments. Each column shows a pair of tested images. These image pairs are under a variety of transformations, including scale change, rotation change, illumination change, occlusion and view point change. The method for line segment extraction we used is similar to the one in [4]. First, we extracted line segments by Canny edge detector [10]. Then these edges are split at points with high curvature. Finally for each set of connected edge points, we fitted a line to it by the least squares method. The high and low threshold of Canny de-

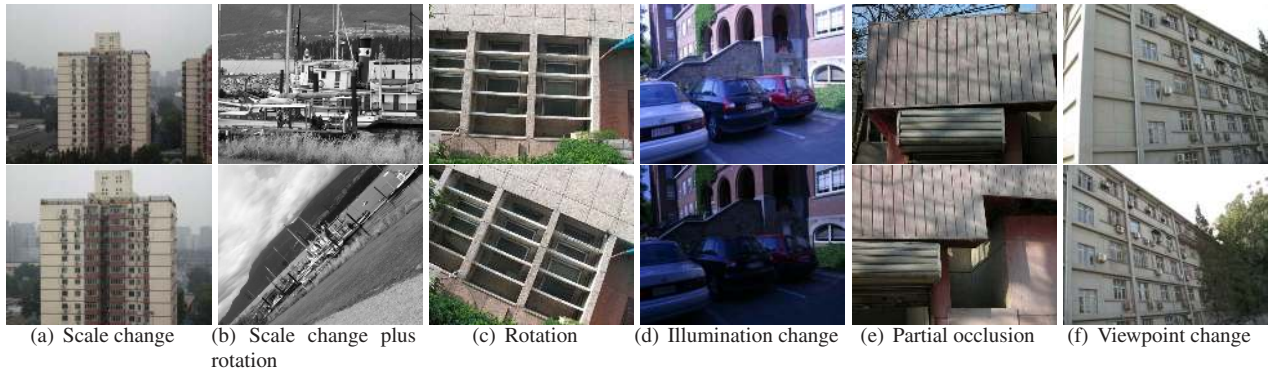


Figure 1. Data set for experiments.

tector are set to 0.2 and 0.1 respectively. The standard deviation of the Gaussian filter in Canny detector is set to 1.0.

4.1. Line Matching Results

In this subsection, extensive experiments are conducted to validate the proposed method on image pairs under various transformations. Six pairs of images are used in our experiments as shown in Fig. 1 and the matching results are shown from Fig. 2 to Fig. 7. The matched lines are shown with the same colors, with labels at their middle points. Whether a match is correct or not is assessed by hand. All matches are assessed one by one manually. The incorrect matches are shown in blue. For better viewing, please see the original color pdf file and zoom in to check the line matches.

Scale changes: We first tested our method of line matching on image pairs with scale changes as shown in Fig. 1(a) and Fig. 1(b). The scale between two images in Fig. 1(a) is about 1.5. As shown in Fig. 2, there are 91 matches, all of them are correct. Note that lines can be matched correctly even their endpoints are very different such as matches 3, 36 etc. It shows that our method can handle the problem of incorrect locations of endpoint successfully. A large scale change image pair is also tested with our method. As shown in Fig. 1(b), the scale between two images is larger than 3. In this case, our method can detect 12 matched lines, only 1 of them is incorrect. It demonstrates that our method is robust to large scale changes.

Rotation changes: Here we tested our method on images with rotation change. Besides image pair in Fig. 1(b) that contains both rotation and scale change, a pair of images shown in Fig. 1(c) with only rotation change is used to test the performance of our method on rotation change. The rotation between two images is about 20° . It can find totally 148 matches with our method, all of them are correct, as shown in Fig. 4.

Brightness changes: The proposed method is also tested under illumination change using a pair of images shown in Fig. 1(d). As can be observed in Fig. 5, there are totally 97

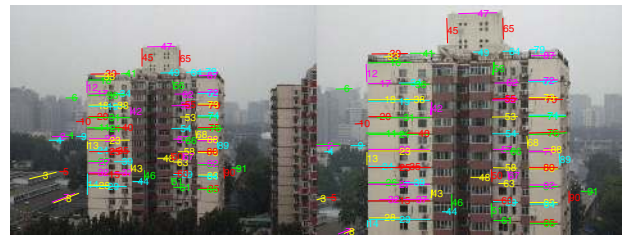


Figure 2. Results of line matching on a pair of images with scale change. This figure is better viewed in color. Extracted Lines : 182,194. Total matches: 91, Correct matches: 91.

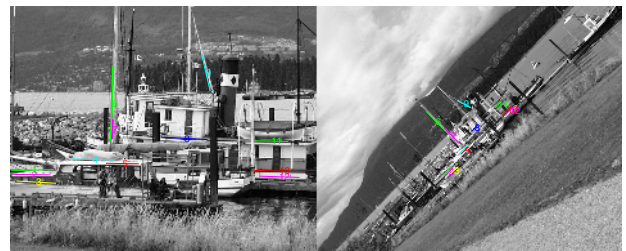


Figure 3. Results of line matching on a pair of images with scale and rotation change. This figure is better viewed in color. Extracted Lines : 380,100. Total matches: 12, Correct matches: 11.

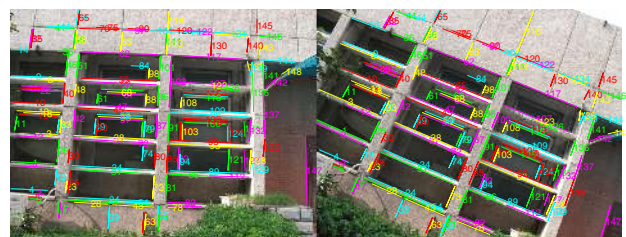


Figure 4. Results of line matching on a pair of images with rotation change. This figure is better viewed in color. Extracted Lines : 209,179. Total matches: 148, Correct matches: 148.

matches with our method and all of them are correct.

Occlusion: As shown in Fig. 1(e), many lines in the bottom image are partially occluded compared to the lines in



Figure 5. Results of line matching on a pair of images with brightness changes. This figure is better viewed in color. Extracted Lines : 211,131. Total matches: 97, Correct matches: 97.

the top image. Due to occlusion, endpoints of corresponding line segments do not correspond. Hence how to deal with the problem of occlusion is a challenge for line matching. As shown by Fig. 6, our method can handle this problem successfully since it does not critically rely on endpoints. There are 45 matches, none of them is incorrect. The results demonstrate that our method is robust to incorrect localization of endpoints of line segments, thus can handle the problem of partial occlusion successfully.



Figure 6. Results of line matching on a pair of images with occlusion. This figure is better viewed in color. Extracted Lines : 159,95. Total matches: 45, Correct matches: 45.

Viewpoint changes: Finally, we tested our method under viewpoint change. Fig. 1(f) shows two images taken from different viewpoints. As shown in Fig. 7, among 242 matches, there are 240 correct matches. The result shows that our method is robust to viewpoint change.

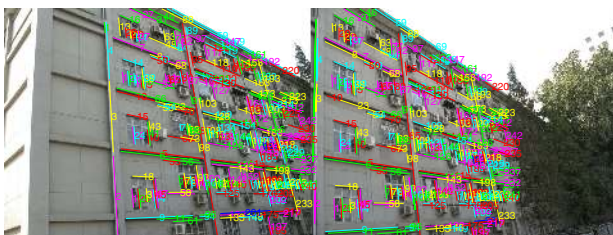


Figure 7. Results of line matching on a pair of images with viewpoint changes. This figure is better viewed in color. Extracted Lines : 347,292. Total matches: 242, Correct matches: 240.

From all of the above experiments, it is clear that the proposed method for line matching is robust to many kinds of image changes, including scale, rotation, brightness, viewpoint changes and occlusion. These image transformations

are very common in practice. For all tested image pairs, the proposed method can achieve very high matching accuracy. Such a good performance of our method should be attributed to the affine invariant which is derived from one line and two points. Since such an invariant is based on the local geometric configuration between lines and points, our method is robust as well as distinctive.

4.2. Robustness to Mismatches of Point Correspondences

Since our method uses point correspondences to boost line matching, in this subsection some experiments are performed to assess our method's robustness to mismatches in the set of point correspondences. As well known, the accuracy of point correspondences and the total number of correspondences are both varied with the threshold used for matching. Therefore, we vary this threshold to obtain different groups of point correspondences. Given a set of point correspondences, RANSAC [6] is used to find out inliers of the set. Thus we can calculate the accuracy of a set of point correspondences by RANSAC.

For each pair of images, we varied the threshold for point matching from 0.9 to 0.7 and then conducted line matching with obtained point correspondences. The results are shown in Table. 1. In the table, for each pair of images, the top row are obtained point correspondences (Total point matches, Matching accuracy) with DAISY matching while the bottom row shows line matching results (Total line matches, Matching accuracy). From Tab. 1, we can see that our method is robust to outliers in the set of point correspondences. The ratio of mismatches to total matches can be up to about 50% in some cases, however, our method can still achieve good line matching results in such cases. It indicates that our method is not sensitive to large number of outliers in point correspondences. Although the total point correspondences and correct point correspondences are varied with different threshold for a same pair of images, both the total matches and matching accuracy of our line matching method are robust as shown in Tab. 1 except for image pair (b). The large scale change of image pair (b) makes it a challenging one for both point and line matching. As can be seen from Tab. 1, when the threshold of point matching is set to 0.9, only 14.2% accuracy can DAISY matching achieve. Such a terrible accuracy plagued the subsequent line matching method. However, when the accuracy of point matching increased to about 50%, our method can achieve very good performance, only 1 of total 12 matches is incorrect. It demonstrates that although our method uses point correspondences for line matching, it is robust when there are outliers in the set of point correspondences and it can achieve very good performance even when the percentage of outlier is about 50%.

	0.9	0.8	0.7
(a)	(1388,67.0%)	(962,82.2%)	(794,89.4%)
	(91,98.9%)	(91,100%)	(90,100%)
(b)	(529,14.2%)	(170,47.1%)	(97,72.2%)
	(21,66.7%)	(12,91.7%)	(9,88.9%)
(c)	(849,70.9%)	(633,88.0%)	(546,93.0%)
	(151,99.3%)	(148,100%)	(146,100%)
(d)	(997,68.3%)	(739,85.5%)	(622,93.1%)
	(98,100%)	(97,100%)	(98,100%)
(e)	(266,25.6%)	(122,65.6%)	(84,75%)
	(46,100%)	(45,100%)	(44,100%)
(f)	(1448,77.6%)	(1225,87.8%)	(1110,93.3%)
	(238,98.8%)	(242,99.2%)	(244,98.8%)

Table 1. Results of line matching with different groups of point correspondences. For each pair of images, the top row is different groups of point correspondences while the bottom row shows line matching results with these point correspondences. Please see text for details.

4.3. Comparison with State-of-the-art Methods

Here, we conducted experiments to compare our method with two state-of-the-art line matching methods : Line Signature (LS) [20] and MSLD [21]. The implementations of them are supplied by their authors. Both of them have their own methods for line segment extraction as described in [20] and [21]. Since matching results may vary with different line segment extracting method even for the same tested images, we compare our method to them with the line segments extracted by their methods respectively, e.g. the line segments extracted by LS are used for comparing our method with LS while the line segments extracted by MSLD are used for comparing our method with MSLD.

Comparison with LS: In order to compare the matching performance of our method to LS, we took the line segments extracted by LS as inputs for our method. Tab. 2 shows the matching performance of our method and LS with tested image pairs in Fig. 1. In the table, for each pair of images the top row shows the number of extracted line segments by LS while the bottom row are matching results by our method and LS respectively. It can be seen from Tab. 2 that LS tends to extract a large number of line segments, owing to its multi scale scheme. Except for image pair (e), our method not only has a larger number of total matches than LS but also has a higher matching accuracy. In some cases such as (a) and (f), our method has about twice of total matches of LS. For image pair (b), our method can even detect more than 4 times of total matches of LS. Although our method is less accurate than LS for image pair (e), our method has more correct matches, e.g. 57 correct matches

Image Pair	Our method	LS
(a)	Extracted Lines: 3000, 3000	
	(304,90.1%)	(154,89.0%)
(b)	Extracted Lines: 3000, 3000	
	(134,82.8%)	(31,77.4%)
(c)	Extracted Lines: 1929, 1861	
	(294,95.6%)	(245,95.5%)
(d)	Extracted Lines: 3000, 2358	
	(360,96.4%)	(256,94.9%)
(e)	Extracted Lines: 1406, 676	
	(65,87.7%)	(36,100%)
(f)	Extracted Lines: 3000, 3000	
	(558,96.4%)	(273,95.2%)

Table 2. Line matching results of our method and LS.

compared with 36 correct matches of LS.

Comparison with MSLD: We have also compared our method with MSLD. The tested image pairs are also those shown in Fig. 1. As we have mentioned before, the line segments extracted by MSLD are used for this experiment in order to get a fair comparison between our method and MSLD. The results are shown in Tab 3. Note that the results of MSLD shown here are slightly different from [21]. This is due to the fact that different extracted line segments are used for line matching because of different parameters setting. Although we can not extract exactly the same lines as published in [21], both our method and MSLD are tested with the same lines. What is more, we also tried several other parameters to extract line segments, and input them for matching, our method consistently outperforms the MSLD. Therefore, such a comparison is convincing. It can be seen from Tab. 3 that MSLD has bad matching result for image pairs (a) and (b). This is due to its inability to deal with scale change. While for other image pairs, MSLD can achieve high accuracy. Compared with MSLD, our method not only obtains good results for image pairs (a) and (b), but also has better performance than MSLD for other image pairs. It indicates that, besides the capability to deal with scale change, our method outperforms MSLD in many cases too.

5. Conclusion

In this paper, we have proposed a method for line matching based on a set of supporting corresponding points susceptible to a significant ratio of mismatches. The result of keypoint matching using DAISY descriptor is served as the set of point correspondences for line matching in our experiments. An affine invariant derived from one line and two points is used to calculate the similarity of two line

Image Pair	Our method	MSLD
(a)	Extracted Lines: 170, 171	
	(67,100%)	(30,46.7%)
(b)	Extracted Lines: 305,70	
	(9,88.9%)	(22,0%)
(c)	Extracted Lines: 165, 140	
	(120,100%)	(98,95.9%)
(d)	Extracted Lines: 161, 86	
	(68,97.1%)	(57,96.5%)
(e)	Extracted Lines: 107, 71	
	(40,100%)	(31,100.0%)
(f)	Extracted Lines: 259, 221	
	(178,98.8%)	(155,97.4%)

Table 3. Line matching results of our method and MSLD.

segments. This similarity is based on the geometry configuration between lines and points, thus it is robust to many changes of image, including scale, rotation, brightness, viewpoint changes and occlusion. Experimental results on image pairs with a variety of transformations validate our proposed method.

6. Acknowledgements

This work is supported by the National Science Foundation of China(60835003, 60633070). Thanks for Lu Wang and Zhiheng Wang providing implementations of their line matching methods to us. Thanks Xiaotong Yuan for his helpful suggestions on this paper.

References

- [1] O. A. Aider, P. Hoppenot, and E. Colle. A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences. *Robotics Autonomous Syst.*, 52:229–246, 2005.
- [2] A. Bartoli, M. Coquerelle, and P. Sturm. A framework for pencil-of-points structure-from-motion. In *European Conference on Computer Vision*, pages 28–40, 2004.
- [3] A. Bartoli and P. Sturm. Multiple-view structure and motion from line correspondences. In *International Conference on Computer Vision*. IEEE, 2003.
- [4] H. Bay, V. Ferrari, and L. V. Gool. Wide-baseline stereo matching with line segments. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2005.
- [5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(24):505–522, 2002.
- [6] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–385, 1981.
- [7] R. Gupta and A. Mittal. SMD: A locally stable monotonic change invariant feature descriptor. In *European Conference on Computer Vision*, 2008.
- [8] R. Hartley. A linear method for reconstruction from lines and points. In *International Conference on Computer Vision*. IEEE, 1995.
- [9] M. Heikkila, M. Pietikainen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42:425–436, 2009.
- [10] J. Canny. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [11] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume I, pages 511–517. IEEE, 2004.
- [12] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2003.
- [13] M. I. A. Lourakis, S. T. Halkidis, and S. C. Orphanoudakis. Matching disparate views of planar surfaces using projective invariants. *Image and Vision Computing*, 18:673–683, 2000.
- [14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
- [16] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–233, 2000.
- [17] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (TOG)*, 25:835–846, 2006.
- [18] A. W. K. Tang, T. P. Ng, Y. S. Hung, and C. H. Leung. Projective reconstruction from line-correspondences in multiple uncalibrated images. *Pattern Recognition*, 39:889–896, 2006.
- [19] F. Tang, S. H. Lim, and N. L. Chang. A novel feature descriptor invariant to complex brightness change. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.
- [20] L. Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *International Conference on Computer Vision*. IEEE, 2009.
- [21] Z. Wang, F. Wu, and Z. Hu. MSLD: A robust descriptor for line matching. *Pattern Recognition*, 39:889–896, 2009.
- [22] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. In *European Conference on Computer Vision*, 2002.
- [23] S. Winder, G. Hua, and M. Brown. Picking the best DAISY. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.