Pattern Recognition 45 (2012) 794–805

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Robust line matching through line–point invariants

Bin Fan *, Fuchao Wu, Zhanyi Hu

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

ARTICLE INFO

Article history:
Received 13 November 2010
Received in revised form
23 July 2011
Accepted 3 August 2011
Available online 10 August 2011

Keywords:
Line matching
Point matching
Line–point invariants

ABSTRACT

This paper is about line matching by line–point invariants which encode local geometric information between a line and its neighboring points. Specifically, two kinds of line–point invariants are introduced in this paper, one is an affine invariant constructed from one line and two points while the other is a projective invariant constructed from one line and four points. The basic idea of our proposed line matching methods is to use cheaply obtainable matched points to boost line matching via line–point invariants, even if the matched points are susceptible to severe outlier contamination. To deal with the inevitable mismatches in the matched points, two line similarity measures are proposed, one is based on the maximum and the other is based on the maximal median. Therefore, four different line matching methods are obtained by combining different line–point invariants with different similarity measures. Their performances are evaluated by extensive experiments. The results show that our proposed methods outperform the state-of-the-art methods, and are robust to mismatches in the matched points used for line matching.

1. Introduction

Feature matching is a fundamental task in computer vision and has been widely used in many applications [1–6]. It aims to find corresponding features, such as points and lines, across images of the same scene. Imaging condition variations, such as illumination and viewpoint changes, make the feature matching a challenging task. Although point matching has been well studied in the past two decades, methods for line matching are less investigated. Line matching is less successful is mainly due to its inherent difficulties: inaccurate locations of endpoints, fragmentation of a single line, less distinctive appearance of line segments, no available global geometric constraint (such as the epipolar constraint in point matching). However, lines contain more structural information about scenes and objects than that contained by points. Therefore, line matching is both desirable and even indispensable in many applications [6,4,7,8].

This paper is focused on line matching through line–point invariants, i.e., rudimentary matched points are used to leverage line matching. Specifically, two kinds of line–point invariants are introduced in this paper (Section 4), one is an affine invariant derived from one image line and two points and the other is a projective invariant derived from one image line and four points. Although these invariants are planar ones, they can be used in general non-planar scenes by computing them from the points in the vicinity of lines. Meanwhile, in order to deal with the inevitable mismatches in the matched points, two line similarity measures, the maximum-based one and the maximal median-based one, are proposed respectively (Section 5). Consequently, four different line matching methods are obtained by combining different line–point invariants with different similarity measures. These line matching methods have been extensively evaluated by experiments. The experimental results show that they can successfully match image lines with high accuracy under various image transformations, including scale changes, rotation, partial occlusion, illumination changes, and viewpoint changes to some extent. They outperform the state-of-the-art methods, i.e., Line Signature [9] and MSLD [10]. In addition, although the proposed methods use matched points to boost line matching, they are largely insensitive to point matching outliers.

A preliminary version of this work was reported in [11] which is to use the affine invariant for line matching. The current work is a largely extended one, including:

(1) A more general line–point invariant (projective invariant, Section 4.2) is introduced and assessed.
(2) A maximum-based line similarity measure (Section 5.1) is proposed and the maximal median-based similarity measure is extended with the projective invariant.
(3) Thorough analysis and experimental evaluation are carried out for the two kinds of line–point invariants (the affine invariant and the projective invariant) as well as the proposed two kinds of similarity measures (the one based on maximum and the one based on maximal median).

The rest of this paper is organized as follows: Section 2 gives a brief overview of the related work. The formal problem definition

* Corresponding author. Tel.: +86 10 62542946; fax: +86 10 62551993.
E-mail addresses: bfan@nlpr.ia.ac.cn (B. Fan), fcwu@nlpr.ia.ac.cn (F. Wu), huzy@nlpr.ia.ac.cn (Z. Hu).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.08.004

is stated in Section 3. The line–point invariants are introduced in Section 4, followed by the proposed line similarity measures in Section 5. A fast matching strategy is proposed in Section 6, then Section 7 gives the proposed line matching algorithm. Extensive experiments are reported in Section 8. We finally conclude this paper in Section 9.

2. Related work

2.1. Point matching

Point matching has received much attention and various methods have been proposed. Most of the point matching methods first construct a local descriptor to describe the neighborhood distribution of a point, then match points by comparing their local descriptors. Through local descriptors, point matching becomes robust to many photometric and geometric transformations. One of the most famous local descriptors is SIFT (Scale Invariant Feature Transformation) [3]. According to the comparative study of Mikolajczyk and Schmid [12], SIFT outperforms other local descriptors including shape context, spin images, differential invariants, moment invariants and so on. Since SIFT was proposed, researchers have developed many other local descriptors which are said to outperform SIFT in some aspect. Ke and Sukthankar [13] applied PCA (Principal Component Analysis) [14] to gradient patch of keypoint and introduced the PCA-SIFT descriptor. Mikolajczyk and Schmid [12] replaced the spatial grid cells of SIFT with cells in a polar coordinate system. Tola et al. [15] developed a fast descriptor named DAISY for dense matching. Winder et al. [16] learned local descriptors with different local features and different spatial pooling strategies. A DAISY-like descriptor is said to be the best among all configurations. Then the best DAISY was picked in [17]. Heikkila et al. [18] used a variant of LBP (Local Binary Pattern) [19] instead of gradient features to improve SIFT for illumination changes. To deal with complex brightness changes, some researchers have proposed local descriptors based on intensity orders [20–22] since the intensity orders are invariant to monotonic illumination changes. Besides using local descriptors for point matching, there are other methods based on geometric constraints among correctly corresponding points along with the appearance of the keypoint's patch [23–27]. These methods can generally deal with more challenging problems in point matching such as non-rigid deformation and large viewpoint changes. The key of this kind of methods is to define a proper measure about the geometric consistency among point correspondences. Such geometric constraints are usually formulated as an objective function so that the correspondence problem can be solved as an optimization problem. These methods can usually achieve higher accuracy, but also have higher computational cost, especially when the set of keypoints is large.

2.2. Line matching

Due to the various inherent difficulties, only a few line matching methods are reported in the literature up to now. Hartley [28] used the trifocal tensor to match lines across three views. Schmid and Zisserman [29] proposed to first find point correspondences on the matched lines by the known epipolar geometry and then to average the cross-correlation scores over all the corresponding points as the line similarity for matching. Both of these two methods need to know the epipolar geometry in advance. Lourakis et al. [30] used two lines and two points to construct a projective invariant for matching planar surfaces with lines and points. But their method can hardly be extended to non-planar scenes and has high computational complexity. Bay et al. [31] proposed to match lines based on their appearance and topological layout. Firstly line segments are matched based on the color histograms of their neighboring profiles, then the topological relations among 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 intensive. Meanwhile, the initial matches are obtained based on color histograms which make it less robust to illumination and other image changes. Recently, Wang et al. [10] proposed a descriptor named MSLD for line matching. It is analogous to SIFT by histogramming gradient orientations in pixel support region of each pixel on a line. Then the MSLD descriptor is constructed by the mean and standard deviation of these histograms. It only relies on the neighboring appearance of the line segment. However, it is less distinctive than local descriptors used for point matching since line segments usually lack of rich textures in their local neighborhoods. Moreover, their method cannot handle scale changes. Wang et al. [9] used Line Signatures to match lines between wide-baseline images. They used angles and length ratios between line segments, which are computed by the endpoints of the line segments, 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 the line segments, their method may fail when a large error is presented in the location of the endpoints.

3. Preliminaries and problem formulation

In this paper, image points are denoted as $\boldsymbol{X}_i$ and lines are denoted as $\boldsymbol{p}_i$ in the reference image. In the query image, points are denoted as $\boldsymbol{Y}_i$ and lines are denoted as $\boldsymbol{q}_i$. Lines refer to straight line segments and are represented as 3d vectors.[1] $\boldsymbol{X}_i$ and $\boldsymbol{Y}_i$ are represented as 2d vectors.

Given two sets of line segments extracted from a reference image and a query image, $\mathcal{L}_1=\{\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_M\}$ and $\mathcal{L}_2=\{\boldsymbol{q}_1,\boldsymbol{q}_2,\ldots,\boldsymbol{q}_N\}$, and additionally a set of tentative matched points which is susceptible to mismatches (for example matched points obtained by SIFT matching), denoted as $\mathcal{C}=\{(\boldsymbol{X}_i,\boldsymbol{Y}_i),i=1,2,\ldots,K\}$. The goal of this work is to match lines in $\mathcal{L}_1$ and $\mathcal{L}_2$ through matched points in $\mathcal{C}$. The resulting set of matched lines is represented as

$$\mathcal{LM}=\{(\boldsymbol{p}_{g(i)},\boldsymbol{q}_{f(i)}),i=1,2,\ldots,N_L\}$$

where $(\boldsymbol{p}_{g(i)},\boldsymbol{q}_{f(i)})$ represents a pair of matched lines and $g(i)\in[1,M],f(i)\in[1,N]$. Since local descriptors of keypoints usually contain orientation information to achieve rotation invariance, a coarse relationship of rotation between the reference and the query images can be estimated from $\mathcal{C}$. Such a not-so-accurate rotation relationship can be used to improve the line matching speed as we will describe in Section 6.

4. Line–point invariants

Here we introduce two kinds of planar line–point invariants (affine invariant and projective invariant) which are used in our line matching methods. The affine invariant is calculated by one line and two points while the projective invariant needs one line and four points to calculate.

4.1. The affine invariant

As shown in Fig. 1(a), assume that space points Q_1, Q_2 and line L all lie on plane Π, and Q_1, Q_2 are not on L. $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{p}$ and $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, $\boldsymbol{q}$

[1] For a line $ax+by+c=0$, it is denoted as $(a,b,c)^T$.

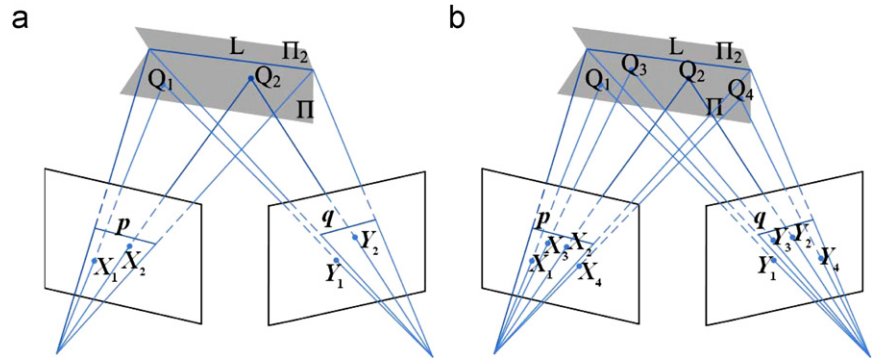

Fig. 1. Illustration of line–point invariants. (a) The affine invariant; (b) the projective invariant.

are their images taken by two cameras respectively. Since Q_1 and Q_2 are coplanar in 3D space, $\boldsymbol{Y}_i$ and $\boldsymbol{X}_i$ are related by a homography induced by the plane Π. Now considering a specific situation, the homography is an affine transformation which is approximately held when Q_1, Q_2 lie in the neighborhood of L. Such an affine transformation is denoted as $\boldsymbol{H}_a$. In the image coordinates, we represent $\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{Y}_1,\boldsymbol{Y}_2$ in homogeneous coordinates as $\tilde{\boldsymbol{X}}_1,\tilde{\boldsymbol{X}}_2,\tilde{\boldsymbol{Y}}_1,\tilde{\boldsymbol{Y}}_2$. By affine transformation, they satisfy

$$\boldsymbol{q} = s\boldsymbol{H}_a^{-T}\boldsymbol{p} \tag{1}$$

$$\tilde{\boldsymbol{Y}}_i = \boldsymbol{H}_a\tilde{\boldsymbol{X}}_i, \quad i=1,2 \tag{2}$$

where s is a scale factor determined by $\boldsymbol{H}_a$ and $\boldsymbol{p}$. We denote

$$D(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{p}) = \frac{\boldsymbol{p}^T\tilde{\boldsymbol{X}}_1}{\boldsymbol{p}^T\tilde{\boldsymbol{X}}_2} \tag{3}$$

$$D(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{q}) = \frac{\boldsymbol{q}^T\tilde{\boldsymbol{Y}}_1}{\boldsymbol{q}^T\tilde{\boldsymbol{Y}}_2} \tag{4}$$

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

$$D(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{p}) = D(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{q}) \tag{5}$$

Eq. (5) means that the ratio of distances from two points to a line is affine invariant. That is to say, $D(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{p})$ is an affine invariant. For the presentation convenience, $(\boldsymbol{X}_2,\boldsymbol{Y}_2)$ is named as base pair of points and $(\boldsymbol{X}_1,\boldsymbol{Y}_1)$ is named as reference pair of points.

4.2. The projective invariant

In this subsection, we discuss a more general situation, i.e., the transformation is a projective homography denoted as $\boldsymbol{H}$. In this case, Eq. (2) becomes

$$\tilde{\boldsymbol{Y}}_i = k_i\boldsymbol{H}\tilde{\boldsymbol{X}}_i, \quad i=1,2 \tag{6}$$

where k_i is an unknown scale factor. Thus Eq. (5) becomes

$$\frac{k_1}{k_2}D(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{p}) = D(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{q}) \tag{7}$$

Eq. (7) indicates that if the transformation is a projective one, $D(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{p})$ is not invariant any more. $D(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{p})$ and $D(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{q})$ differ in a factor which is determined by $\boldsymbol{X}_1,\boldsymbol{X}_2$ and $\boldsymbol{H}$. In order to eliminate this factor, we bring in two points Q_3 and Q_4 in Π which are neither on L nor on the line determined by Q_1 and Q_2, as shown in Fig. 1(b). $\boldsymbol{X}_3,\boldsymbol{X}_4$ and $\boldsymbol{Y}_3,\boldsymbol{Y}_4$ are their projected image points in the two cameras respectively. Their homogeneous coordinates are $\tilde{\boldsymbol{X}}_3,\tilde{\boldsymbol{X}}_4,\tilde{\boldsymbol{Y}}_3,\tilde{\boldsymbol{Y}}_4$. Then Q_3, Q_4 are used to constitute an auxiliary space line on the plane Π and its projection on the two cameras are $\boldsymbol{r} = (\tilde{\boldsymbol{X}}_3 \otimes \tilde{\boldsymbol{X}}_4)$ and $\boldsymbol{s} = (\tilde{\boldsymbol{Y}}_3 \otimes \tilde{\boldsymbol{Y}}_4)$ respectively. We denote

$$\mathrm{Proj}(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{X}_3,\boldsymbol{X}_4,\boldsymbol{p}) = \frac{\boldsymbol{p}^T\tilde{\boldsymbol{X}}_1}{\boldsymbol{p}^T\tilde{\boldsymbol{X}}_2} : \frac{\boldsymbol{r}^T\tilde{\boldsymbol{X}}_1}{\boldsymbol{r}^T\tilde{\boldsymbol{X}}_2} \tag{8}$$

$$\mathrm{Proj}(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{Y}_3,\boldsymbol{Y}_4,\boldsymbol{q}) = \frac{\boldsymbol{q}^T\tilde{\boldsymbol{Y}}_1}{\boldsymbol{q}^T\tilde{\boldsymbol{Y}}_2} : \frac{\boldsymbol{s}^T\tilde{\boldsymbol{Y}}_1}{\boldsymbol{s}^T\tilde{\boldsymbol{Y}}_2} \tag{9}$$

Since $\boldsymbol{r}$ and $\boldsymbol{s}$ are corresponding lines in the two views, they satisfy

$$\boldsymbol{s} = k\boldsymbol{H}^{-T}\boldsymbol{r} \tag{10}$$

where k is an unknown scale factor. Therefore, we have

$$\frac{\boldsymbol{s}^T\tilde{\boldsymbol{Y}}_1}{\boldsymbol{s}^T\tilde{\boldsymbol{Y}}_2} = \frac{k_1}{k_2}\frac{\boldsymbol{r}^T\tilde{\boldsymbol{X}}_1}{\boldsymbol{r}^T\tilde{\boldsymbol{X}}_2} \tag{11}$$

Combining Eqs. (7)–(9) and (11), we have

$$\mathrm{Proj}(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{X}_3,\boldsymbol{X}_4,\boldsymbol{p}) = \mathrm{Proj}(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{Y}_3,\boldsymbol{Y}_4,\boldsymbol{q}) \tag{12}$$

It is meant that $\mathrm{Proj}(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{X}_3,\boldsymbol{X}_4,\boldsymbol{p})$ is a projective invariant.

Note that such a planar projective invariant has been used in [30] in the form of two lines and two points. In this paper we derive it in the form of one line and four points. Such a modification has several advantages:

(1) It makes the invariant applicable for line matching in general cases, not merely to planar scenes, because the corresponding space line and space points of one line and four points (these points are obtained in the neighborhood of the line) are more likely to be coplanar compared with those of two lines and two points.
(2) It requires combinatorial search for the other corresponding lines when the invariant associated with 'two lines+two points' is directly used for line matching since the tentatively matched lines are not available. In contrast, our proposed 'one line+four points' invariant can be directly used for line matching without complicated combinatorial optimization, since the results of point matching are used as the tentatively matched points. Therefore, with the 'one line+four points' invariant, all we need is to design a line similarity measure that is robust to mismatches in the matched points. We will describe our similarity measures in Section 5.

4.3. Discussion

For either the affine invariant or the projective invariant, all the points and the line are required to be coplanar. To this end, a support region for the line is defined to find possible coplanar points in Section 5. In order to use the affine invariant for line matching, at least two correctly matched points are required to be

coplanar with the matched lines while at least four correctly matched points are required for the projective invariant. Since there are mismatches in the matched points, say the correct rate is p_r, then p_r can be regarded as the probability that a pair of matched points is correct. If we further assume that for a pair of matched lines, there exist n pairs of matched points in their support regions, then the probability that at least two pairs of matched points are correct is

$$\mathcal{P}_2 = 1-(1-p_r)^n - C_n^1(1-p_r)^{n-1}p_r \tag{13}$$

while the probability that at least four pairs of matched points are correct is

$$\mathcal{P}_4 = \mathcal{P}_2 - C_n^2(1-p_r)^{n-2}p_r^2 - C_n^3(1-p_r)^{n-3}p_r^3 \tag{14}$$

Obviously, $\mathcal{P}_2 > \mathcal{P}_4$, which means that it is more likely to construct reliable affine invariants than projective invariants for line matching. As a result, a line matching method based on the affine invariant could find more line matches.

5. Similarity measure of two image lines based on line–point invariants

In order to match two lines $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ by the line–point invariants proposed in Section 4, the first thing we need to do is to find out matched points from set $\mathcal{C}$ so that their corresponding space points may be coplanar with the corresponding space line of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$. Note that line segments detected in images seldom correspond to isolated 3D lines in real world, they usually correspond to edges of surfaces. Therefore, in the neighborhood of at least one side of an image line, some coplanar points with the line could be obtained. The side of an image line is defined by its gradient, which is defined as the average gradient of the constituent points of the line. More specifically, if a point is located on the region directed by the gradient of a line, it is said to be on the right side of this line, otherwise it is on the left side. For each side of a line, a support region is defined in order to find matched points that may be coplanar with the line. The support region of a line is defined according to its length in order to be scale invariant. If the distance from a point $\boldsymbol{X}$ to a line $\boldsymbol{l}$ is smaller than $\alpha \times length(\boldsymbol{l})$ and the distance from $\boldsymbol{X}$ to the perpendicular bisector of $\boldsymbol{l}$ is smaller than $\beta \times length(\boldsymbol{l})$, $\boldsymbol{X}$ is said to be in the support region of $\boldsymbol{l}$. $length(\boldsymbol{l})$ is the length of $\boldsymbol{l}$, α and β are two ratios defining the shape of the support region and their values are set by experiments (see Table 1). Due to the inevitable localization errors of line endpoints, the support regions of two corresponding lines may not correspond to each other exactly. However, our method needs only part of them to be correspondences since we just need to find matched points in these support regions. The correctly matched points must be located on the corresponding subregions. We denote

$$\mathcal{S}_l^{right} = \{\boldsymbol{X} : \boldsymbol{X} \in region(\boldsymbol{l}) \ \&\ \boldsymbol{X} \in right(\boldsymbol{l})\}$$

$$\mathcal{S}_l^{left} = \{\boldsymbol{X} : \boldsymbol{X} \in region(\boldsymbol{l}) \ \&\ \boldsymbol{X} \in left(\boldsymbol{l})\}$$

as the sets of points in the right and left support regions of $\boldsymbol{l}$ respectively, where $region(\boldsymbol{l})$ denotes the support region of $\boldsymbol{l}$ while $right(\boldsymbol{l})$ and $left(\boldsymbol{l})$ denote right and left side of $\boldsymbol{l}$ respectively.

Table 1
The parameter settings of α and β in different methods.

	Method I	Method II	Method III	Method IV
α	2.0	2.5	2.0	2.0
β	0.5	0.5	0.7	0.7

Then, we denote the set of matched points located in the right support region of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ as

$$\mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \{(\boldsymbol{X}_k \in \mathcal{S}_{p_m}^{right}, \boldsymbol{Y}_k \in \mathcal{S}_{q_n}^{right}), k = 1,2,\ldots,N^r\}$$

where $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ are a pair of matched points while N^r is the total number of matched points in the support region. Similarly

$$\mathcal{G}^{left}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \{(\boldsymbol{X}_k \in \mathcal{S}_{p_m}^{left}, \boldsymbol{Y}_k \in \mathcal{S}_{q_n}^{left}), k = 1,2,\ldots,N^l\}$$

represents the set of matched points that are located in the left support region. For the notation convenience, we denote $g_k = (\boldsymbol{X}_k, \boldsymbol{Y}_k)$ as a pair of matched points. Given $\mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ or $\mathcal{G}^{left}(\boldsymbol{p}_m,\boldsymbol{q}_n)$, two kinds of line similarity measures of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ are defined in the following subsections. For a pair of lines $(\boldsymbol{p}_m,\boldsymbol{q}_n)$, we first calculate two similarities based on $\mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ and $\mathcal{G}^{left}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ respectively, then the larger one is served as the final similarity of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$.

5.1. The similarity measure based on maximum

Take the right side as an example. Since not all the matched points in $\mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ are correct ones, we first calculate all possible similarities with the matched points in $\mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n)$, then the maximum is taken as the similarity of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$. The underlying principle is that if there exist several correctly matched points in $\mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ when $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ are correctly matched lines, the similarity calculated by these correctly matched points tends to 1, which is the largest possible similarity value of a pair of lines. Specifically, the affine invariant needs at least two pairs of correctly matched points while the projective invariant needs at least four pairs. If the affine invariant is used

$$\mathrm{SimA}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \max_{g_i,g_j \in \mathcal{G}^{right} \ \&\ i \neq j} \{\mathrm{AffSim}(g_i,g_j)\} \tag{15}$$

$$\mathrm{AffSim}(g_i,g_j) = e^{-\|D(\boldsymbol{X}_i,\boldsymbol{X}_j,\boldsymbol{p}_m) - D(\boldsymbol{Y}_i,\boldsymbol{Y}_j,\boldsymbol{q}_n)\|} \tag{16}$$

While if the projective invariant is used

$$\mathrm{SimA}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \max_{g_i,g_j,g_k,g_l \in \mathcal{G}^{right} \ \&\ i \neq j \neq k \neq l} \{\mathrm{ProjSim}(g_i,g_j,g_k,g_l)\} \tag{17}$$

$$\mathrm{ProjSim}(g_i,g_j,g_k,g_l) = e^{-\|\mathrm{Proj}(\boldsymbol{X}_i,\boldsymbol{X}_j,\boldsymbol{X}_k,\boldsymbol{X}_l,\boldsymbol{p}_m) - \mathrm{Proj}(\boldsymbol{Y}_i,\boldsymbol{Y}_j,\boldsymbol{Y}_k,\boldsymbol{Y}_l,\boldsymbol{q}_n)\|} \tag{18}$$

Similarly, we can get $\mathrm{SimA}^{left}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ and the final similarity of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ is defined as

$$\mathrm{SimA}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \max\{\mathrm{SimA}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n), \mathrm{SimA}^{left}(\boldsymbol{p}_m,\boldsymbol{q}_n)\} \tag{19}$$

Note that in Eq. (18), four pairs of matched points are used to calculate the similarity of two lines. Given four pairs of matched points, a straightforward way to calculate the line similarity is to compute the local homography between the two views from these four pairs of matched points, and then calculate the similarity of the query line and the transformed reference line (according to the computed homography) as the line similarity. However, this homography-based approach is rather time consuming as shown by our experiments (see Table 3).

5.2. The similarity measure based on maximal median

The similarity based on maximum proposed in the proceeding subsection is a straightforward undertaking, it may be susceptible to non-robustness. For instance, supposing that:

(1) $\boldsymbol{p}$ is a line in the reference image;
(2) $\boldsymbol{q}_n$ is $\boldsymbol{p}$'s corresponding line in the query image;
(3) $\boldsymbol{q}_m$ is a line parallel to $\boldsymbol{q}_n$ in the query image;

(4) g_1,g_2 are two correctly matched point pairs in the support regions of these lines, i.e.,

$\{g_1,g_2\} \in \mathcal{G}(\boldsymbol{p},\boldsymbol{q}_m)$ & $\{g_1,g_2\} \in \mathcal{G}(\boldsymbol{p},\boldsymbol{q}_n)$

in which $\mathcal{G}$ represents either $\mathcal{G}^{right}$ or $\mathcal{G}^{left}$;

(5) $D(\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{p}) \approx D(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{q}_n) \approx 1$, which means that $\boldsymbol{q}_n^T\tilde{\boldsymbol{Y}}_1 \approx \boldsymbol{q}_n^T\tilde{\boldsymbol{Y}}_2$.

Since $\boldsymbol{q}_m$ is parallel to $\boldsymbol{q}_n$ and $\boldsymbol{q}_n^T\tilde{\boldsymbol{Y}}_1 \approx \boldsymbol{q}_n^T\tilde{\boldsymbol{Y}}_2$, thus $\boldsymbol{q}_m^T\tilde{\boldsymbol{Y}}_1 \approx \boldsymbol{q}_m^T\tilde{\boldsymbol{Y}}_2$. Therefore, $D(\boldsymbol{Y}_1,\boldsymbol{Y}_2,\boldsymbol{q}_m) \approx 1$. In this case, it may result in a false match if the maximum is taken as the similarity since both $\mathrm{SimA}(\boldsymbol{p},\boldsymbol{q}_n)$ and $\mathrm{SimA}(\boldsymbol{p},\boldsymbol{q}_m)$ have values near 1. The similar problem exists for the projective invariant.

In this subsection, the line similarity based on median statistic is developed which is much more robust, as shown later by our experiments (Section 8).

5.2.1. The similarity measure based on maximal median with the affine invariant

Firstly, we select one pair of matched points $g_k \in \mathcal{G}_l^{right}$ as the base pair of points for constructing affine invariants, then each of the remaining N^r-1 pairs of matched points is taken as the reference pair to obtain N^r-1 corresponding affine invariants. So, N^r-1 similarities can be calculated by

$$\mathrm{SimB}_k^{right}(i) = \mathrm{AffSim}(g_i,g_k), \quad i \in [1,N^r] \ \&\ i \neq k \tag{20}$$

Then we can get the median of them as

$$\mathrm{SimB}_k^{right} = \underset{i \in [1,N^r] \ \&\ i \neq k}{\mathrm{median}} \{\mathrm{SimB}_k^{right}(i)\} \tag{21}$$

Choosing the median can improve the robustness to incorrectly matched points in $\mathcal{G}_l^{right}$ while preserving good discriminativeness. Finally, each $g_k \in \mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n)$, $k=1,2,\ldots,N^r$ is taken as the base pair of points to obtain N_r medians: SimB_k^{right}, $k=1,2,\ldots,N^r$. The maximum of them is then taken as the similarity of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ with matched points in the right side. That is

$$\mathrm{SimB}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \max_{k \in [1,N^r]} \{\mathrm{SimB}_k^{right}\} \tag{22}$$

In a similar way, we can get $\mathrm{SimB}^{left}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ for the left side. As we have said before, the final similarity of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ is the larger one of these two similarities:

$$\mathrm{SimB}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \max\{\mathrm{SimB}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n),\mathrm{SimB}^{left}(\boldsymbol{p}_m,\boldsymbol{q}_n)\} \tag{23}$$

5.2.2. The similarity measure based on maximal median with the projective invariant

In the case of projective invariant, two pairs of matched points $(g_i,g_j) \in \mathcal{G}_l^{right}$ are firstly selected to constitute the two auxiliary lines: $\boldsymbol{r} = (\tilde{\boldsymbol{X}}_i \otimes \tilde{\boldsymbol{X}}_j)$ and $\boldsymbol{s} = (\tilde{\boldsymbol{Y}}_i \otimes \tilde{\boldsymbol{Y}}_j)$. Then, one pair of matched points g_k from the remaining N^r-2 pairs is fixed and so N^r-3 similarities can be obtained as

$$\mathrm{SimB}_{i,j}^{right}(k,l) = \mathrm{ProjSim}(g_k,g_l,g_i,g_j), \quad l \in [1,N^r] \ \&\ l \neq k \neq i \neq j \tag{24}$$

Then, we can get the median of them as

$$\mathrm{SimB}_{i,j}^{right}(k) = \underset{l \in [1,N^r] \ \&\ l \neq k \neq i \neq j}{\mathrm{median}} \{\mathrm{SimB}_{i,j}^{right}(k,l)\} \tag{25}$$

Taking each $g_k \in \mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n)$, $k \in [1,N^r]$ & $k \neq i \neq j$ as the fixed pair of matched points, we can get N^r-2 medians: $\mathrm{SimB}_{i,j}^{right}(k)$, $k \in [1,N^r]$ & $k \neq i \neq j$. The maximum of these medians is taken as the similarity of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ when the auxiliary lines are constituted by $(g_i,g_j) \in \mathcal{G}_l^{right}$, denoted as

$$\mathrm{SimB}_{i,j}^{right} = \max_{k \in [1,N^r] \ \&\ k \neq i \neq j} \{\mathrm{SimB}_{i,j}^{right}(k)\} \tag{26}$$

Therefore, by taking each different combination of two pairs (g_i,g_j) belonging to $\mathcal{G}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ to constitute the two auxiliary lines, we can obtain $C_{N^r}^2$ maximums: $\mathrm{SimB}_{i,j}^{right}, i,j \in [1,N^r]$ & $i \neq j$. Since a line matching method based on the projective invariant requires at least four correctly matched points in the support regions of the matched lines, thus there are at least $C_4^2 = 6$ such maximums having values near 1. Consequently, the median of the top six maximums is taken as the similarity of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$, i.e., the third largest one.[2] That is

$$\mathrm{SimB}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \underset{i,j \in [1,N^r] \ \&\ i \neq j}{\mathrm{Third}} \{\mathrm{SimB}_{i,j}^{right}\} \tag{27}$$

Finally, the similarity of $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ is defined as

$$\mathrm{SimB}(\boldsymbol{p}_m,\boldsymbol{q}_n) = \max\{\mathrm{SimB}^{right}(\boldsymbol{p}_m,\boldsymbol{q}_n),\mathrm{SimB}^{left}(\boldsymbol{p}_m,\boldsymbol{q}_n)\} \tag{28}$$

6. Fast matching

Our work uses tentatively matched points to leverage line matching. Such a set of matched points can be easily obtained with keypoint matching, such as SIFT. In order to be rotation invariant, these keypoints are detected along with their orientations. Therefore, once matched points are obtained, one can use orientations of the matched points to estimate an approximate global rotation between the reference and the query images. Here, we propose a fast matching strategy using such a global rotation to speed up the matching process.

For each pair of matched points in $\mathcal{C}$, an angle between the point in the query image and its matched point in the reference image can be calculated by their orientations. Then with K (the number of matched points in $\mathcal{C}$) angles, an angle histogram can be obtained. The angle corresponding to the largest peak in the histogram is taken as the global rotation between the reference and the query images. Note that for real images taken at different viewpoints, different matched points may have different rotations, and there does not exist a global rotation between two images in theory. Therefore, in our work the estimated global rotation is not used as a hard constraint, but is only used to discard those line pairs that are obviously impossible matches. Suppose the estimated global rotation is θ. Given a pair of lines to be matched, if $|\theta'-\theta| > t_\theta$ in which θ' is the orientation difference between the two lines, this line pair is simply considered to be a non-match without further calculating their similarity by Eq. (19) or (23) or (28). This can in practice filter out most of the non-matches, thus speeding up the matching process substantially. Here t_θ is a pre-set threshold to decide whether the angle between the two lines violates θ too much. The lower the threshold is, the faster the matching process will be, but the number of line matches will be reduced if it is set too low. If it is set too high, then it will have no effect on speeding up. We found in our experiments that t_θ setting is not a delicate matter and our line matching methods perform stably with t_θ in a wide range. In all experiments in Section 8, t_θ is set to 20°.

[2] Note that the direct use of the median of all the maximums is not a good choice here. For instance, if there are six matched points in the support region and four of them are correct, then there are $C_6^2 = 15$ maximums, $C_4^2 = 6$ of them have values near 1, so the median of them, i.e., the eighth largest one, is incorrect. This will results in incorrectly matched lines.

Fig. 2. Some image pairs from the dataset used for comparison of different line matching methods.

7. The algorithm

Given a reference image and a query image, firstly, two sets of line segments $\mathcal{L}_1=\{\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_M\}$ and $\mathcal{L}_2=\{\boldsymbol{q}_1,\boldsymbol{q}_2,\ldots,\boldsymbol{q}_N\}$ are extracted. Secondly, a set of matched points $\mathcal{C}=\{(\boldsymbol{X}_i,\boldsymbol{Y}_i),i=1,2,\ldots,K\}$ is obtained by a keypoint matching method. In the experiments of this paper, we used DAISY [17] to describe keypoints and then matched them by the nearest neighbor distance ratio (NNDR) [12]. The threshold of NNDR is set to 0.8. Then the global rotation θ is estimated as described in Section 6. Along with the threshold t_θ which is used for fast matching, all of these are served as input to the line matching algorithm outlined in Algorithm 1. In Algorithm 1, line matching is conducted under the uniqueness constraint. Here, the uniqueness of matching is meant that if $\boldsymbol{p}_m$ and $\boldsymbol{q}_n$ are a pair of matched lines and $\mathrm{SimL}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ is their similarity, then there is no other line in $\mathcal{L}_2$ such that its similarity to $\boldsymbol{p}_m$ is larger than $\mathrm{SimL}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ and there is no other line in $\mathcal{L}_1$ such that its similarity to $\boldsymbol{q}_n$ is larger than $\mathrm{SimL}(\boldsymbol{p}_m,\boldsymbol{q}_n)$ either. It allows many-to-one matches which happen when a line in one image is split into multi-segments.

Algorithm 1. LineMatching

Input:
line set from reference image $\mathcal{L}_1$, line set from query image $\mathcal{L}_2$, corresponding point set $\mathcal{C}$, the estimated global rotation between the two images θ, threshold for rotation t_θ.
Output:
A set of corresponding lines $\mathcal{LM}$.

```
Initialization: LM = ∅.
for each p_i ∈ L_1 do
  for each q_j ∈ L_2 do
    if |angle(p_i,q_j) − θ| > t_θ then
      SimL(p_i,q_j) = 0
    else
      Calculate SimL(p_i,q_j) according to Eq. (19) or (23) or (28).
    end if
  end for
  Find the line in L_2 with the largest similarity to p_i according to SimL(p_i,q_j), j ∈ [1,N]. Such a line is represented as q_k.
  if SimL(p_i,q_k) ≥ 0.95 then
    Check whether p_i is the line with the largest similarity to q_k in L_1
    if TRUE then
      LM = LM ⋃ {(p_i,q_k)}
    end if
  end if
end for
Output: LM
```

8. Experimental results

Line segments extraction: In our experiments, line segments are extracted based on Canny [32] edge detector similar to [31]. Firstly, edges are detected from images by Canny detector. Then they are split at points with high curvature. Finally for each set of connected edge points, a line is fitted by the least-squares method. In our experiments, the high and low thresholds of Canny detector are set to 0.2 and 0.1 respectively. The standard deviation of the Gaussian filter in Canny detector is set to 1.0. In order to reduce the influence of noise and errors in line segments extraction, only the line segments whose length are larger than 20 pixels are used for line matching.

8.1. Line matching results

In this paper, two line–point invariants and two kinds of similarity measures are proposed. Therefore, four different combinations could be formed, and each could act as a method for line matching. They are as follows:

Method I: The affine invariant+the similarity measure based on maximum (Section 5.1).
Method II: The affine invariant+the similarity measure based on maximal median (Section 5.2.1).
Method III: The projective invariant+the similarity measure based on maximum (Section 5.1).
Method IV: The projective invariant+the similarity measure based on maximal median (Section 5.2.2).

In these methods, parameters α and β which define the support region of a line are set by experiments[3] as shown in Table 1.

Firstly, we conducted experiments to compare the performances of these four methods. To this end, 30 pairs of images are selected for experiments, which contain both planar and non-planar scene images, including various image transformations: viewpoint changes, scale changes, illumination changes, image rotation and occlusion. In order to make the evaluation representative, some of these images are from images used in [10], and some are from the publicly available dataset on the Internet[4,5] while others are captured by ourselves. Fig. 2 shows some image pairs in the database. For each pair of images, keypoints are described by DAISY and matched with the NNDR to obtain the set of tentatively matched points used for line matching. The threshold of NNDR is set to 0.8 as described in Section 7. Then we

[3] For image pairs shown in Fig. 3, we tested $\alpha=\{1.0,1.5,2.0,2.5\}$ and $\beta=\{0.4,0.5,0.6,0.7\}$ for the four line matching methods. α and β are then set to those values that lead to the best matching results.
[4] http://www.robots.ox.ac.uk/~vgg/data.
[5] http://lear.inrialpes.fr/people/mikolajczyk/Database/index.html.

conducted line matching according to Algorithm 1. We have recorded the matching results for each image pair and averaged them as shown in Table 2. In all experiments of this paper, whether a line match is correct or not is assessed one by one manually.

From Table 2 we can see that:

(1) By fixing the invariant (the affine invariant or the projective invariant), the line matching methods using the maximum-based similarity measure (i.e., Method I and Method III) are not as good as the ones using the maximal median-based similarity measure (i.e., Method II and Method IV). This indicates that the maximum-based similarity measure is less robust, which is consistent with the analysis in Section 5.1.

(2) By fixing the similarity measure (the maximum based or the maximal median based), the line matching methods based on the affine invariant are superior to the ones based on the projective invariant, i.e., Method I is better than Method III and Method II is better than Method IV. This is consistent with the analysis in Section 4.3.

Table 2
Comparison of different line matching methods.

Method	Total matches	Correct matches	Matching accuracy (%)
I	62.7	55.8	89.0
II	65.1	61.9	95.1
III	52.5	46.2	88.0
IV	61.1	57.0	93.3

Since the similarity measure based on maximal median is much more robust than the similarity measure based on maximum, in the following experiments, only the line matching methods using the maximal median-based similarity measure are evaluated, i.e., Method II and Method IV.

Then experiments are conducted on image pairs with specific transformations. Seven pairs of images are used as shown in Fig. 3, including five types of transformations: scale, rotation, illumination, occlusion and viewpoint changes. The matching results are shown in Table 3, and Figs. 4–10 show the matched lines with Method II, where matched lines are shown with the same color, with labels at their middle points. The incorrect matches are shown in blue. For better viewing, please see original color pdf file and zoom in to check the line matches.

It can be seen from Table 3 that both Method II and Method IV perform fairly well to all kinds of the tested image transformations and Method II performs slightly better than Method IV.

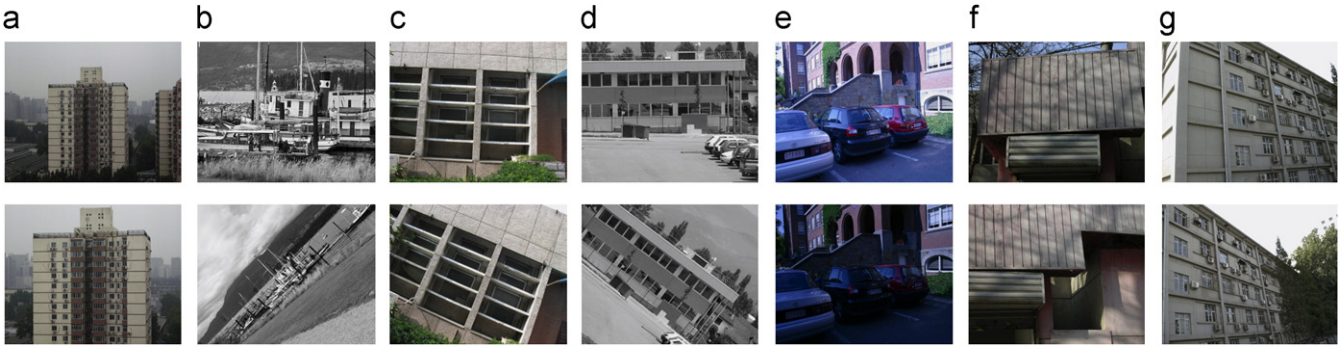

Fig. 3. Images with various transformations.

Table 3
The matching results (total matches, correct matches) of image pairs in Fig. 3.

	(a)	(b)	(c)	(d)	(e)	(f)	(g)
Method II	(92,92)	(15,14)	(148,148)	(182,177)	(97,97)	(44,44)	(241,239)
Time (s)	1.18	0.34	0.82	7.18	0.55	0.27	3.19
Method IV	(91,82)	(16,14)	(152,149)	(183,177)	(99,98)	(43,43)	(239,236)
Time (s)	73.60	0.40	110.08	224.14	34.54	77.51	188.69
Homo	(92,84)	(12,10)	(147,147)	(183,171)	(103,96)	(40,40)	(243,241)
Time (s)	2699	2.4	3531	7911	1468	3650	6432

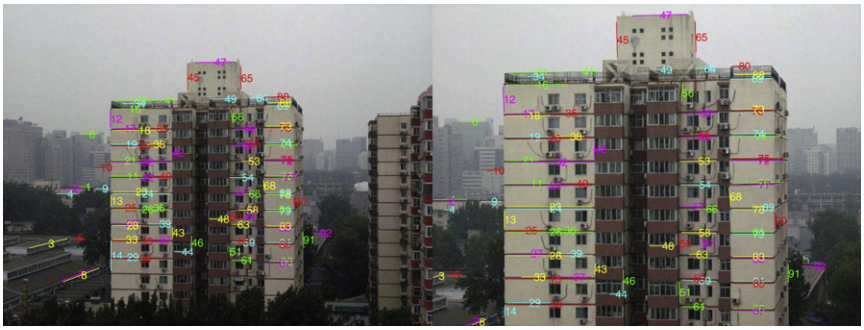

Fig. 4. Scale change. Extracted lines: 182,194. Total matches: 92. Correct matches: 92.

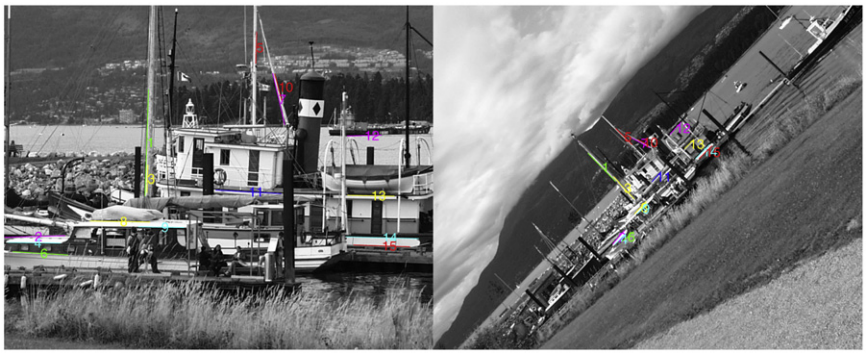

Fig. 5. Large-scale change + rotation. Extracted lines: 380,100. Total matches: 15. Correct matches: 14. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

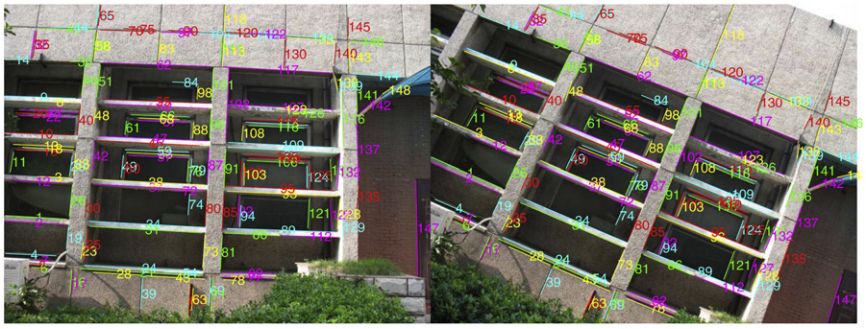

Fig. 6. Rotation. Extracted lines: 209,179. Total matches: 148. Correct matches: 148.

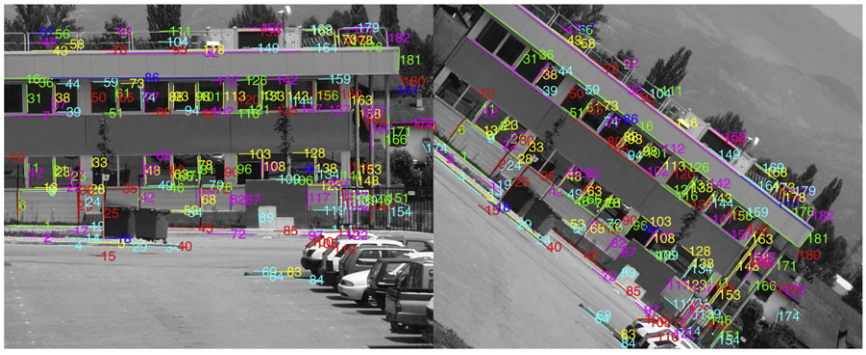

Fig. 7. Rotation. Extracted lines: 385,241. Total matches: 182. Correct matches: 177. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 8. Illumination change. Extracted lines: 211,131. Total matches: 97. Correct matches: 97.

Such a good performance may largely attributes to our introduced line–point invariants which are derived from one line and several neighboring points. Since such line–point invariants are good descriptors of the local geometric configuration between line and points, our proposed methods are robust as well as distinctive. For image pair in Fig. 3(b), the large-scale change makes a low repeatability of the extracted line segments in these two images. Meanwhile, the obtained point matching results that are used for line matching are not good due to large-scale change in Fig. 3(b), so there are many matched lines that do not have enough

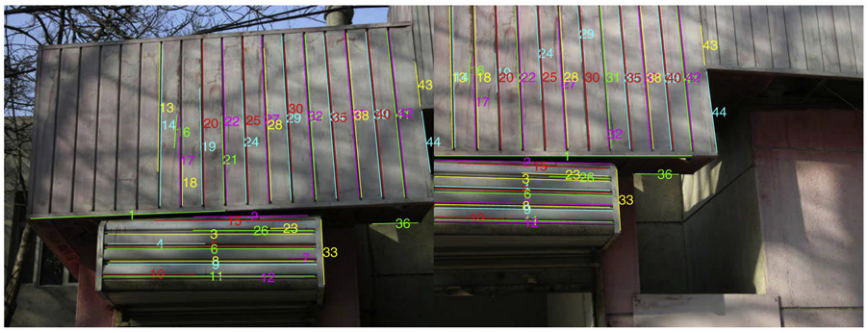

Fig. 9. Partially occlusion. Extracted lines: 159,95. Total matches: 44. Correct matches: 44.

Fig. 10. Viewpoint change. Extracted lines: 347,292. Total matches: 241. Correct matches: 239. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

correctly matched points in their support regions. Thus cannot obtaining correctly line–point invariants for these matched lines. These might be the reasons why the obtained line matches of Fig. 3(b) are not as many as that of other image pairs in Fig. 3. Note that the line segments can be matched correctly even their endpoints are very different such as matches 3, 35, etc., in Fig. 4 and most of the matches in Fig. 9 (the occlusion makes the endpoints of corresponding line segments do not correspond). This is because Method II does not critically rely on endpoints, thus it can successfully cope with the incorrect localization problem of endpoints. Method IV has a similar performance. For comparison, Table 3 also gives the matching results by the homography-based method (see the last paragraph in Section 5.1). Although it can also obtain satisfactory results, its computational load is too high.

8.2. Comparison with the state-of-the-art methods

Here, we conducted experiments to compare our line matching methods with two state-of-the-art methods : Line Signature (LS) [9] and MSLD [10]. The implementations of them are provided by their authors. Both of them have their own methods for line segments extraction as described in [9,10]. Since matching results may vary with different line segment extracting methods even for the same test images, we compare our methods to them with the line segments extracted by their methods respectively, i.e., the line segments extracted by LS are used for comparing our methods with LS while the line segments extracted by MSLD are used for comparing our methods with MSLD.

Comparison with LS: Table 4 shows the matching results of our methods and LS for image pairs in Fig. 3. For each image pair in the table, the top row shows the number of extracted line segments by LS while the bottom row gives the line matching results (total matches, correct matches, matching accuracy). It can be seen from Table 4 that LS tends to extract a large number of line segments owing to its multi-scale scheme. Except for image pair (f), Method II and Method IV not only have more matched lines than LS, but also have higher accuracy. Although Method II and Method IV are less accurate than LS for image pair (f), they have more correct matches, i.e., 61 correct matches of Method II and 62 correct matches of Method IV compared with 36 correct matches of LS.

Table 4
Line matching results (total matches, correct matches, matching accuracy) of our methods and LS. The line segment extracting method of LS is used for extracting line segments.

Image pair	Method II	Method IV	LS
(a)	Extracted lines: 3000, 3000		
	(307,280,**91.2**%)	(257,232,**90.3**%)	(154,137,**89.0**%)
(b)	Extracted lines: 3000, 3000		
	(150,120,**80.0**%)	(114,72,**63.2**%)	(31,24,**77.4**%)
(c)	Extracted lines: 1929, 1861		
	(285,275,**96.5**%)	(245,239,**97.6**%)	(245,234,**95.5**%)
(d)	Extracted lines: 3000, 2970		
	(478,449,**93.9**%)	(424,402,**94.8**%)	(227,207,**91.2**%)
(e)	Extracted lines: 3000, 2358		
	(372,352,**94.6**%)	(329,307,**93.3**%)	(256,243,**94.9**%)
(f)	Extracted lines: 1406, 676		
	(69,61,**88.4**%)	(68,62,**91.2**%)	(36,36,**100**%)
(g)	Extracted lines: 3000, 3000		
	(553,528,**95.5**%)	(506,485,**95.8**%)	(273,260,**95.2**%)

Comparison with MSLD: The matching results of image pairs in Fig. 3 by Method II, Method IV and MSLD are presented in Table 5. Note that the results of MSLD shown here are slightly different from [10]. This is due to the fact that different extracted line segments are used for line matching owing to different parameter settings. Although we cannot extract the same line segments as in [10], all the evaluated methods are tested with the same line segments. In addition, we also tried several other parameters to extract line segments for matching. The ranking of the evaluated methods does not change. Therefore, such a comparison is meaningful and representative. It can be seen from Table 5 that MSLD

has bad matching results for image pairs (a) and (b). This is due to its inability to deal with scale changes. While for other image pairs, MSLD can achieve high accuracy. Compared with MSLD, Method II and Method IV not only obtain good results for image pairs (a) and (b), but also have better performance than MSLD for the other image pairs. It demonstrates that, besides the ability to deal with scale changes, our line matching methods also outperform MSLD under other image transformations.

Table 5
Line matching results (total matches, correct matches, matching accuracy) of our methods and MSLD. The line segment extracting method of MSLD is used for extracting line segments.

Image pair	Method II	Method IV	MSLD
(a)	Extracted lines: 170, 171		
	(68,68,**100**%)	(68,63,**92.6**%)	(30,14,**46.7**%)
(b)	Extracted lines: 305, 70		
	(11,9,**81.8**%)	(11,9,**81.8**%)	(22,0,**0**%)
(c)	Extracted lines: 165, 140		
	(120,120,**100**%)	(115,115,**100**%)	(98,94,**95.9**%)
(d)	Extracted lines: 279, 162		
	(109,107,**98.2**%)	(100,97,**97**%)	(91,81,**89.0**%)
(e)	Extracted lines: 161, 86		
	(69,67,**97.1**%)	(67,65,**97**%)	(57,55,**96.5**%)
(f)	Extracted lines: 107, 71		
	(40,40,**100**%)	(38,38,**100**%)	(31,31,**100.0**%)
(g)	Extracted lines: 259, 221		
	(178,176,**98.8**%)	(178,174,**97.8**%)	(155,151,**97.4**%)

8.3. Robustness to mismatches in matched points

Since our line matching methods use matched points to boost line matching, in this subsection we performed experiments to show that our methods are not sensitive to mismatches in the set of matched points. As we have said in Section 7, in the experiments of this paper, the matched points used for line matching are obtained by firstly describing points with DAISY and then matching them by NNDR. By changing the threshold of NNDR, different sets of matched points that have different matching accuracy and total matches can be obtained. For each image pair in Fig. 3, we varied the threshold of NNDR from 0.9 to 0.7 and then conducted line matching with the obtained matched points. Line matching results are reported in Tables 6 and 7, in which the top row of each image pair presents the obtained sets of matched points (total matches, matching accuracy) used for line matching while the bottom row shows our line matching results (total matches, correct matches, matching accuracy). Table 6 gives the matching results of Method II and Table 7 gives the matching results of Method IV. RANSAC [33] is used to find inliers among matched points, and the matching accuracy is computed as

$$\text{accuracy} = \frac{\text{number of inliers by RANSAC}}{\text{total number of matched points}} \tag{29}$$

Note that here RANSAC is only used to compute the accuracy of the matched points, it is the whole set of matched points including mismatches used for line matching.

Table 6
Line matching (Method II) results with different sets of matched points. For each image pair, the top row presents different sets of matched points while the bottom row shows line matching results with these matched points. See text for detail.

Threshold of NNDR	0.9	0.85	0.8	0.7
Image pair (a)	(1388,67.0%)	(1114,75.7%)	(962,82.2%)	(794,89.4%)
	(93,92,**98.9**%)	(92,91,**98.9**%)	(92,92,**100**%)	(91,91,**100**%)
Image pair (b)	(717,13.1%)	(326,31.9%)	(183,48.6%)	(94,70.2%)
	(18,16,**88.9**%)	(16,15,**93.8**%)	(15,14,**93.3**%)	(13,12,**92.3**%)
Image pair (c)	(849,70.9%)	(712,81.9%)	(633,88.0%)	(546,93.0%)
	(150,149,**99.3**%)	(149,148,**99.3**%)	(148,148,**100**%)	(147,147,**100**%)
Image pair (d)	(2316,73.3%)	(1981,76.8%)	(1787,85.3%)	(1469,89.1%)
	(184,173,**93.5**%)	(178,171,**96.1**%)	(182,177,**97.3**%)	(181,177,**97.8**%)
Image pair (e)	(997,68.3%)	(852,77.9%)	(739,85.5%)	(622,93.1%)
	(98,98,**100**%)	(97,97,**100**%)	(97,97,**100**%)	(97,97,**100**%)
Image pair (f)	(266,25.6%)	(163,46.6%)	(122,65.6%)	(84,75%)
	(46,45,**97.8**%)	(46,46,**100**%)	(44,44,**100**%)	(44,44,**100**%)
Image pair (g)	(1448,77.6%)	(1315,83.2%)	(1225,87.8%)	(1110,93.3%)
	(240,236,**98.3**%)	(240,237,**98.8**%)	(241,239,**99.2**%)	(243,240,**98.8**%)

Table 7
Line matching (Method IV) results with different sets of matched points. For each image pair, the top row presents different sets of matched points while the bottom row shows line matching results with these matched points. See text for detail.

Threshold of NNDR	0.9	0.85	0.8	0.7
Image pair (a)	(1388,67.0%)	(1114,75.7%)	(962,82.2%)	(794,89.4%)
	(94,85,**90.4**%)	(92,84,**91.3**%)	(91,82,**90.1**%)	(89,81,**91.0**%)
Image pair (b)	(717,13.1%)	(326,31.9%)	(183,48.6%)	(94,70.2%)
	(22,15,**68.2**%)	(18,16,**88.9**%)	(16,14,**87.5**%)	(10,5,**50.0**%)
Image pair (c)	(849,70.9%)	(712,81.9%)	(633,88.0%)	(546,93.0%)
	(152,149,**98.0**%)	(153,150,**98.0**%)	(152,149,**98.0**%)	(148,145,**98.0**%)
Image pair (d)	(2316,73.3%)	(1981,76.8%)	(1787,85.3%)	(1469,89.1%)
	(178,168,**94.4**%)	(182,176,**96.7**%)	(183,177,**96.7**%)	(180,176,**97.8**%)
Image pair (e)	(997,68.3%)	(852,77.9%)	(739,85.5%)	(622,93.1%)
	(102,100,**98.0**%)	(100,98,**98.0**%)	(99,98,**99.0**%)	(100,99,**99.0**%)
Image pair (f)	(266,25.6%)	(163,46.6%)	(122,65.6%)	(84,75%)
	(44,44,**100**%)	(43,43,**100**%)	(43,43,**100**%)	(41,41,**100**%)
Image pair (g)	(1448,77.6%)	(1315,83.2%)	(1225,87.8%)	(1110,93.3%)
	(237,234,**98.7**%)	(239,236,**98.7**%)	(239,236,**98.7**%)	(240,238,**99.2**%)

Table 8
Comparison of the fast matching and the normal matching.

Image pair		(a)	(b)	(c)	(d)	(e)	(f)	(g)	Ave.
Method II (affine invariant+maximal median-based similarity measure)									
Time (s)	Fast	**1.18**	**0.34**	**0.82**	**7.18**	**0.55**	**0.27**	**3.19**	**1.93**
	Normal	3.30	0.62	2.50	15.82	2.02	0.60	12.30	5.31
Total matches	Fast	**92**	**15**	**148**	**182**	**97**	**44**	**241**	–
	Normal	92	15	148	181	98	44	241	–
Correct matches	Fast	**92**	**14**	**148**	**177**	**97**	**44**	**239**	–
	Normal	92	14	148	175	97	44	238	–
Method IV (projective invariant+maximal median-based similarity measure)									
Time (s)	Fast	**73.60**	**0.40**	**110.08**	**224.14**	**34.54**	**77.51**	**188.69**	**101.28**
	Normal	133.41	0.67	194.96	384.22	66.62	203.75	589.93	224.79
Total matches	Fast	**91**	**16**	**152**	**183**	**99**	**43**	**239**	–
	Normal	89	17	149	183	98	44	237	–
Correct matches	Fast	**82**	**14**	**149**	**177**	**98**	**43**	**236**	–
	Normal	81	14	149	176	96	43	235	–

It can be found that although the total matched points and matching accuracy are varied under different NNDR thresholds for a given image pair, both the total matched lines and matching accuracy of our line matching results do not change too much as shown in Tables 6 and 7 except for image pair (b). Since the projective invariant requires at least four correctly matched points in the support regions of matched lines to construct, it is harder to satisfy than the affine invariant which only requires at least two correctly matched points to construct, thus Method II is slightly better than Method IV. For image pair (b), the large-scale change makes it a challenging one for point matching, where the number of the matched points is small and the matching accuracy is low. Consequently, many matched lines do not have enough correctly matched points in their support regions, and it is more pronounced for the projective invariant-based method, i.e., Method IV. Therefore, as shown in Table 7, Method IV does not work well when the point matching accuracy is too low (NNDR threshold is 0.9) or the number of matched points is too small (NNDR threshold is 0.7). It works well in other cases. To sum up, although our proposed methods use matched points for line matching, they are rather robust to mismatches and can achieve very good performance even when the percentage of point matching outlier is as high as 50%.

8.4. The effectiveness of fast matching

In order to validate the effectiveness of the fast matching strategy proposed in Section 6, we performed experiments to compare the running time on matching lines with or without the fast matching strategy. Normal matching (without the fast matching strategy) needs to calculate all the similarities between two lines, while fast matching only calculates some of them, thus speeding up the matching process. For image pairs in Fig. 3, we recorded running time for line matching, with the fast matching strategy and without it respectively, denoted as *Fast* and *Normal* in Table 8. Besides running time, line matching results are listed in Table 8 too.

It can be seen from Table 8 that with the fast matching strategy, the running time of Method II for line matching is about three times faster than that with normal matching strategy, and the fast matching strategy contributes to two times of speed-ups when applied to Method IV. The running time differs with image pairs is due to different numbers of extracted line segments and different numbers of matched points. When using the projective invariant for line matching, it needs to search for all the combinations of four matched points in the support regions of matched lines while the affine invariant-based method only needs to search for the combinations of two matched points, so the computational complexity of Method IV is much higher than Method II, as indicated in Table 8. What is more, it is worth noting that the proposed fast matching strategy also reduces the searching space of line matching, thus the line matching method with the fast matching strategy performs slightly better than that without the fast matching strategy as shown in Table 8.

9. Conclusion

In this paper, we have proposed line matching methods based on a set of matched points susceptible to a significant ratio of mismatches using two kinds of line–point invariants, i.e., an affine invariant derived from one line and two points and a projective invariant derived from one line and four points, which encode local geometric information between a line and its neighboring points. In order to deal with the inevitable mismatches in the set of matched points, the maximum-based and the maximal median-based similarity measures are proposed to calculate line similarity. Since the affine invariant requires less points to construct than the projective invariant, it has a higher probability to encode correct local geometric information of matched lines, thus the affine invariant-based method is generally better than the projective invariant-based method. On the other hand, compared with the maximum-based similarity, the maximal median-based similarity is more robust to outliers in the matched points used for line matching, so the line matching method using the maximal median-based similarity measure is better than that using the maximum-based similarity measure. Extensive experiments under various image transformations have validated the effectiveness of our proposed line matching methods as well as their superiorities to the state-of-the-art methods.

Acknowledgments

This work is supported by the National Science Foundation of China (60835003, 61075038). Thanks for Lu Wang and Zhiheng Wang providing implementations of their line matching methods to us. Thanks for the anonymous suggestions by the reviewers.

References

[1] C.J. Taylor, D.J. Kriegman, Structure and motion from line segments in multiple images, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (11) (1995) 1021–1032.
[2] M. Brown, D. Lowe, Recognising panoramas, Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 1218–1225.

[3] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[4] O.A. Aider, P. Hoppenot, E. Colle, A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences, Robotics and Autonomous Systems 52 (2005) 229–246.
[5] N. Snavely, S.M. Seitz, R. Szeliski, Photo tourism: exploring photo collections in 3D, ACM Transactions on Graphics (TOG) 25 (2006) 835–846.
[6] A.W.K. Tang, T.P. Ng, Y.S. Hung, C.H. Leung, Projective reconstruction from line-correspondences in multiple uncalibrated images, Pattern Recognition 39 (5) (2006) 889–896.
[7] A. Bartoli, P. Sturm, Multiple-view structure and motion from line correspondences, Proceedings of the IEEE International Conference on Computer Vision, vol. 1, 2003, pp. 207–212.
[8] A. Bartoli, M. Coquerelle, P. Sturm, A framework for pencil-of-points structure-from-motion, in: European Conference on Computer Vision, 2004, pp. 28–40.
[9] L. Wang, U. Neumann, S. You, Wide-baseline image matching using line signatures, in: Proceedings of the IEEE International Conference on Computer Vision, 2009, pp. 1311–1318.
[10] Z.H. Wang, F.C. Wu, Z.Y. Hu, MSLD: a robust descriptor for line matching, Pattern Recognition 39 (2009) 889–896.
[11] B. Fan, F. Wu, Z. Hu, Line matching leveraged by point correspondences, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 390–397.
[12] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Transaction on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1615–1630.
[13] Y. Ke, R. Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2004, pp. 511–517.
[14] I.T. Joliffe, Principal Component Analysis, Springer-Verlag, 1986.
[15] E. Tola, V. Lepetit, P. Fua, A fast local descriptor for dense matching, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[16] S. Winder, M. Brown, Learning local image descriptors, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[17] S. Winder, G. Hua, M. Brown, Picking the best DAISY, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2009, pp. 178–185.
[18] M. Heikkila, M. Pietikainen, C. Schmid, Description of interest regions with local binary patterns, Pattern Recognition 42 (2009) 425–436.
[19] T. Ojala, M. Pietikainen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognition 29 (1996) 51–59.
[20] R. Gupta, A. Mittal, SMD: a locally stable monotonic change invariant feature descriptor, in: European Conference on Computer Vision, 2008, pp. 265–277.
[21] F. Tang, S.H. Lim, N.L. Chang, H. Tao, A novel feature descriptor invariant to complex brightness change, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2009, pp. 2631–2638.
[22] R. Gupta, H. Patil, A. Mittal, Robust order-based methods for feature description, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 334–341.
[23] Z. Zhang, R. Deriche, O. Faugeras, Q.-T. Luong, A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, Artificial Intelligence 78 (1995) 87–119.
[24] A.C. Berg, T.L. Berg, J. Malik, Shape matching and object recognition using low distortion correspondences, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 26–33.
[25] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: Proceedings of the IEEE International Conference on Computer Vision, 2005, pp. 1482–1489.
[26] O. Choi, I.S. Kweon, Robust feature point matching by preserving local geometric consistency, Computer Vision and Image Understanding 113 (2009) 726–742.
[27] O. Duchenne, F. Bach, I. Kweon, J. Ponce, A tensor-based algorithm for high-order graph matching, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2009, pp. 1980–1987.
[28] R. Hartley, A linear method for reconstruction from lines and points, in: Proceedings of the IEEE International Conference on Computer Vision, 1995, pp. 882–887.
[29] C. Schmid, A. Zisserman, The geometry and matching of lines and curves over multiple views, International Journal of Computer Vision 40 (3) (2000) 199–233.
[30] M.I.A. Lourakis, S.T. Halkidis, S.C. Orphanoudakis, Matching disparate views of planar surfaces using projective invariants, Image and Vision Computing 18 (2000) 673–683.
[31] H. Bay, V. Ferrari, L.V. Gool, Wide-baseline stereo matching with line segments, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 329–336.
[32] J. Canny, A computational approach to edge detection, IEEE Transaction on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.
[33] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (1981) 381–385.

Bin Fan is an assistant professor at the Institute of Automation, Chinese Academy of Sciences. He received his BS degree from Beijing University of Chemical Technology in 2006, and PhD degree from the Institute of Automation, Chinese Academy of Sciences in 2011. His research interest covers image matching and feature extraction.

Fuchao Wu is a Professor at the Institute of Automation, Chinese Academy of Sciences. His research interest covers computer vision, which includes camera calibration, 3D reconstruction, active vision and image-based modeling and rendering.

Zhanyi Hu is a Professor at the Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, geometric primitive extraction, vision guided robot navigation and pattern recognition.