J. Vis. Commun. Image R. 24 (2013) 794–805

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

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

An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency

Lilian Zhang *, Reinhard Koch

Institute of Computer Science, University of Kiel, 24098 Kiel, Germany

ARTICLE INFO

Article history:
Received 18 July 2012
Accepted 12 May 2013
Available online 24 May 2013

Keywords:
Line segment matching
Multi-scale line detection
Line band descriptor
Unary geometric attribute
Pairwise geometric consistency
Relational graph
Graph matching
Spectral method

ABSTRACT

We present a line matching algorithm which utilizes both the local appearance of lines and their geometric attributes. To overcome the problem of segment fragmentation and geometric variation, we extract lines in the scale space. To depict the local appearance of lines, we design a novel line descriptor called Line Band Descriptor (LBD). To evaluate the pairwise geometric consistency, we define the pairwise geometric attributes between line pairs. Then we built a relational graph for candidate line matches and employ a spectral technique to solve this matching problem efficiently. The advantages of the proposed algorithm are as follows: (1) it is robust to image transformations because of the multi-scale line detection strategy; (2) it is efficient because the designed LBD descriptor is fast to compute and the appearance similarities reduce the dimension of the graph matching problem; (3) it is accurate even for low-texture images because of the pairwise geometric consistency evaluation.

1. Introduction

One of the challenging areas in computer vision is feature matching, which is a basic tool for applications in scene reconstruction [1], pattern recognition and retrieval [2], stereo SLAM [3] and so on. Most of the existing matching methods in the literature are based on local points or region features [4] which are deficient for low-texture scenes [5]. On the contrary, line features are often abundant in these situations. Moreover, line features and other local features provide complementary information about scenes. Therefore, line segment matching is both desirable and indispensable in many applications. Although some progress was achieved recently for the line matching problem [5,6], these approaches are quite computationally expensive, prohibiting their usage in many applications. This paper addresses the problem of robust and efficient line feature matching based on the results of our previous work [7] with the following extensions: (i), the Line Band Descriptor (LBD) is presented in detail; (ii), the descriptor performance is experimentally evaluated; (iii), the robustness of the histogram based rotation estimation method is discussed; (iv), in this paper, we also conduct an experiment to reveal the influence of the multi-scale line detection strategy and the geometric constraints.

1.1. Related work

Existing approaches to match lines are of three types: those that match individual line segments, those that match groups of line segments and those that perform line matching by employing point correspondences.

For matching lines in image sequences or small baseline stereos where extracted corresponding segments are similar, approaches based on matching individual lines are suitable [8–10] because of their better computational performance. Among the first group, Wang et al. [11] proposed a descriptor named Mean-Standard deviation Line Descriptor (MSLD) for line matching based on the appearance of the pixel support region. This approach achieves good matching results for moderate image variations in textured scenes.

Generally, approaches which match groups of line segments have the advantage that more geometric information is available for disambiguation. A large number of methods have been developed around the idea of graph-matching [12–15], however, most of them are for small baseline stereo image pairs. Bay et al. [16] presented a wide baseline stereo line matching method which compares the histograms of neighboring color profiles and iteratively eliminates mismatches by a topological filter. The results shown in their work are for structured scenes with small number of lines, thus the performance on images featuring a larger range of conditions is not clear. Wang et al. [5] used line signatures to match lines between wide baseline images. To overcome the unreliable line detection problem, a multi-scale line extraction strategy

* Corresponding author.
E-mail addresses: lz@mip.informatik.uni-kiel.de (L. Zhang), rk@mip.informatik.uni-kiel.de (R. Koch).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.05.006

which extracts lines by verifying multiple merging thresholds from the edge image is employed. A line signature is constructed for each extracted line. This approach significantly improves the repeatability of line signatures and therefore has a good matching performance. However, this method is quite computationally expensive because of the huge number of line signatures.

Given a set of point correspondences, Schmid and Zisserman [17] took the epipolar constraint of line endpoints for short baseline matching and presented a plane sweep algorithm for wide baseline matching. Lourakis et al. [18] used two lines and two points to construct a projective invariant for matching planar surfaces. Kim and Lee [19] presented a line matching method by using coplanar Line Intersection Context Features (LICF). More recently, Fan et al. [6] explored an affine invariant from two points and one line. They utilized this affine invariant to match lines with known point correspondences. The main drawback of these approaches is the requirement of known epipolar geometry or point correspondences. Besides, their performance in low texture scenes is limited because of the lack of good point correspondences.

1.2. Our approach

Several reasons make line matching a difficult problem [6,17], including: inaccurate locations of line endpoints, fragmentation of lines, lack of strongly disambiguating geometric constraints (e.g. Epipolar Constraints), lack of distinctive appearance in low-texture scenes, instabilities for large image transformations. To deal with these challenges, the approach in this paper is built on three strategies.

The first is to extract lines in the scale space making the matching algorithm robust to the scale changes. Though there is some work on detecting and tracking scale invariant lines [20,21], the proposed multi-scale line extraction approach simply apply the EDLine [22] detector to a scale-space pyramid consisting a set of octave images, because it is more efficient to detect features in the scale space [23–26] than to directly extract scale invariant regions [27].

The second strategy is to characterize the local appearance of line segments by the Line Band Descriptor (LBD) which is more efficient to compute than MSLD [11]. Different from the edge descriptors proposed in [28,29], the proposed line descriptors are not designed to overcome the large scale changes because it is inefficient to adjust the scale of support region for each line segment. Instead, the multi-scale line extraction approach is adopted to solve this problem more efficiently.

The third novel part is to combine the local appearance of lines and the geometric constraints between line pairs to build a relational graph. The dimension of the graph matching problem is reduced by checking the appearance similarities and geometric consistencies. A spectral method [30] is employed to solve the matching problem which avoids the combinatorial explosion inherent to the graph matching problem. The geometric relationship of corresponding line pairs in two images may be not exactly affine invariant because they are often not coplanar. However, for images without strong view point changes, most of the correctly corresponding line pairs tend to establish strong agreement links among each other while the incorrect assignments have weak links in the graph and few of them have strong links by accident. This property makes the spectral technique a promising strategy to efficiently solve the matching problem.

Compared to state-of-the-art methods, experiments validate that the proposed line matching approach is faster to generate the matching results. It is also robust against various image transformations including occlusion, rotation, blurring, illumination changes, scale changes, and moderate view point changes even for non-planar scenes or low-texture scenes.

The rest of this paper is organized as follows. Section 2 presents the way to extract lines in the scale space and to construct the line descriptors. Section 3 introduces the processes to generate candidate matching pairs, to build the relational graph and to solve the graph matching problem via spectral technique. The descriptor performance evaluation is presented in Section 4 and the experimental matching results are reported in Section 5. Finally, we draw the conclusion in the last section.

2. Line detection and description

In this section, we first present the approach to detect lines in the scale space. Then the way to construct the line descriptor is introduced. The main reason for proposing this new line descriptor is to depict the local appearances of lines more efficiently than MSLD [11] without losing the matching performance.

2.1. Detecting lines in the scale space

To overcome the fragmentation problem of line detection and to improve the performance for large scale changes, in our line detection framework we employ a scale-space pyramid consisting of N octave images which are generated by down-sampling the original image with a set of scale factors and Gaussian blurring. There is no intra-layer between two consecutive octaves. We first apply the EDLine [22] algorithm to each octave producing a set of lines in the scale space. Each line has a direction which is given by making the gradients of most edge pixels pointing from the left side of the line to its right side. Then we re-organize them by finding corresponding lines in the scale space. For all lines extracted in the scale space, they are assigned a unique ID and stored into a vector called LineVec if they are related to the same event in the image (i.e. the same region of the image with the same direction). The final extracted results are a set of LineVecs as illustrated in Fig. 1. The line detection approach is different from Wang et al. [5] by re-organizing all the line segments extracted in the scale space to form LineVecs, which reduces the dimension of the graph matching problem.

As shown in Fig. 1, each LineVec may include more than one line in the scale space. To depict the local appearance of a LineVec, for each line in it, we will generate a line descriptor from the octave image where the line is extracted. The representation of the line support region and the construction of the line descriptor are introduced in the following.

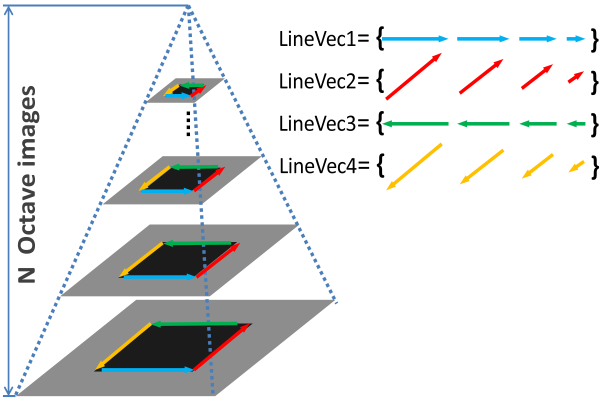


Fig. 1. Illustration of the line detection in the scale space. The original image is down-sampled to generate a set of octave images. For each octave image, line segments are extracted by the EDLine [22] detector. For all the extracted lines, they are re-organized to form a set of LineVecs. Lines in the same LineVec have the same direction and are corresponding to the same region in the original image.

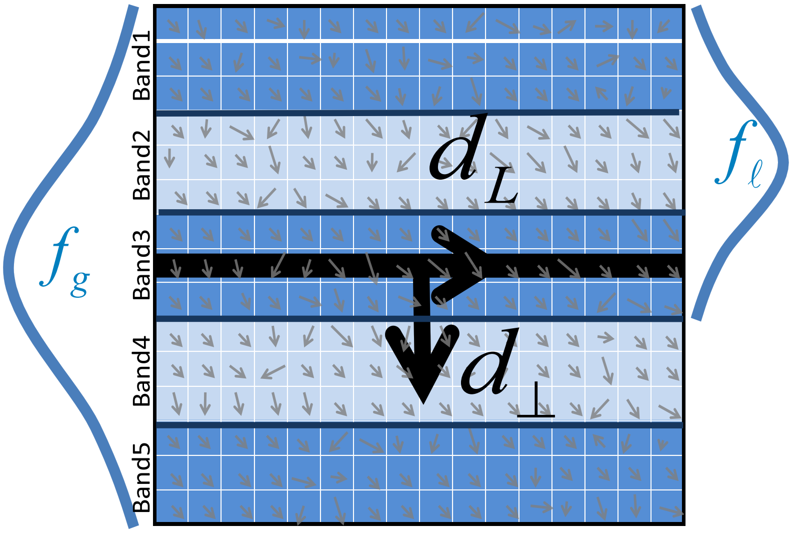


Fig. 2. Illustration of the band representation. A local rectangular region around the line is chosen as the line support region (LSR). Two directions $\mathbf{d}_L$ and $\mathbf{d}_\perp$ are introduced. The LSR is divided into m bands with the width of w, (e.g. $m = 5,\ w = 3$). A global Gaussian function f_g is applied to all rows in the LSR. For each band (e.g. Band2), a local Gaussian function f_l is applied to rows in the band and its nearest two neighbor bands. Small arrows represent the gradients of pixels in the LSR.

2.2. The band representation of the line support region

Given a line segment in the octave image, the descriptor will be computed from the line support region (LSR) which is a local rectangular region centered at the line as shown in Fig. 2. This support region is divided into a set of bands $\{B_1, B_2, \ldots, B_m\}$ where each band is a sub-region of the LSR and parallel with the line. The numbers of bands m and the width of each band w will be discussed in Section 4.1. Fig. 2 illustrates an example of the LSR when $m = 5,\ w = 3$. The length of the band naturally equals to the length of the segment.

Similar to MSLD [11], two directions which form a local 2D coordinate frame are introduced to distinguish parallel lines with opposite gradient directions and to make the descriptor rotation invariant. According to the line direction $\mathbf{d}_L$, the orthogonal direction $\mathbf{d}_\perp$ is defined as the clockwise orthogonal direction of $\mathbf{d}_L$. The middle point of the line is chosen as the origin of this local coordinate frame. The gradient of each pixel in the LSR is projected into this local frame $\mathbf{g}' = (\mathbf{g}^T \cdot \mathbf{d}_\perp, \mathbf{g}^T \cdot \mathbf{d}_L)^T \triangleq (\mathbf{g}'_{d_\perp}, \mathbf{g}'_{d_L})^T$ in which $\mathbf{g}$ and $\mathbf{g}'$ are the pixel gradients in the image frame and the local frame respectively.

Motivated by SIFT [23] and MSLD [11], two Gaussian functions are applied to each row of the LSR along the $\mathbf{d}_\perp$ direction. First, a global weighting coefficient $f_g(i) = (1/\sqrt{2\pi}\sigma_g)e^{-d_i^2/2\sigma_g^2}$ is assigned to the ith row in the LSR, in which d_i is the distance of the ith row to the center row of LSR and $\sigma_g = 0.5(m \cdot w - 1)$. Second, considering a band B_j, for rows in the band B_j and in its nearest neighbor bands $B_{j-1},\ B_{j+1}$, a local weighting coefficient $f_l(k) = (1/\sqrt{2\pi}\sigma_l)e^{-d_k'^2/2\sigma_l^2}$ is assigned to the kth row, in which d'_k is the distance of the kth row to the center row of B_j and $\sigma_l = w$. The purpose of the global Gaussian window is to give less emphasis to gradients that are far from the line mitigating the sensitivity to small changes in the position of the LSR along the direction $\mathbf{d}_\perp$. The purpose of the local Gaussian window is to reduce boundary effects. It avoids that the descriptor changes abruptly as pixels move from one band to the next.

By this representation, we gain two advantages compared to the sub-region representation introduced in [11]: First, it is more robust to small position changes in the direction $\mathbf{d}_L$ because in this case, most part of the image content in the band keeps unchanged with a little variation in the band boundary. Note that, this feature is important since generally the position accuracy of a line is lower in the direction $\mathbf{d}_L$ than in the direction $\mathbf{d}_\perp$ due to the unstable line endpoints. Second, it is more computationally efficient because there is no overlap between bands in the direction $\mathbf{d}_L$ and the Gaussian weights are applied to each row rather than each pixel directly.

2.3. The construction of the Line Band Descriptor

For a band B_j in the LSR, the band descriptor BD_j is computed from rows of B_j and its nearest two neighbor bands $B_{j-1},\ B_{j+1}$. Specially, for the top and bottom bands B_1 and B_m, rows which are outside of the LSR will not be considered when computing the band descriptor of B_1 and B_m. After computing $\{BD_j\}$, the Line Band Descriptor LBD is simply generated by concatenating them:

$$LBD = (BD_1^T, BD_2^T, \ldots, BD_m^T)^T. \tag{1}$$

Now, we construct the band descriptor BD_j. For the kth row in the band B_j or its neighbors, we accumulate the gradients of pixels within this row as:

$$\begin{aligned} v1_j^k &= \lambda \sum_{\mathbf{g}'_{d_\perp} > 0} \mathbf{g}'_{d_\perp}, \quad & v2_j^k &= \lambda \sum_{\mathbf{g}'_{d_\perp} < 0} -\mathbf{g}'_{d_\perp}, \\ v3_j^k &= \lambda \sum_{\mathbf{g}'_{d_L} > 0} \mathbf{g}'_{d_L}, \quad & v4_j^k &= \lambda \sum_{\mathbf{g}'_{d_L} < 0} -\mathbf{g}'_{d_L}. \end{aligned} \tag{2}$$

where the Gaussian coefficient $\lambda = f_g(k)f_l(k)$.

By stacking these four accumulated gradients of all rows associated with the band B_j, the band description matrix (BDM) is constructed as:

$$BDM_j = \begin{pmatrix} v1_j^1 & v1_j^2 & \cdots & v1_j^n \\ v2_j^1 & v2_j^2 & \cdots & v2_j^n \\ v3_j^1 & v3_j^2 & \cdots & v3_j^n \\ v4_j^1 & v4_j^2 & \cdots & v4_j^n \end{pmatrix} \in \Re^{4\times n}, \tag{3}$$

where n is the number of rows associated with B_j:

$$n = \begin{cases} 2w, & j = 1 \| m; \\ 3w, & \text{else}. \end{cases}$$

Now BD_j is simply constructed using the mean vector M_j and the standard deviation vector S_j of the matrix $BDM_j : BD_j = (M_j^T, S_j^T)^T \in \Re^8$. Substituting in Eq. (1), yields:

$$LBD = (M_1^T, S_1^T, M_2^T, S_2^T, \ldots, M_m^T, S_m^T)^T \in \Re^{8m}. \quad (4)$$

Similar to [11], the mean part $(M_1^T, M_2^T, \ldots, M_m^T)$ and the standard deviation part $(S_1^T, S_2^T, \ldots, S_m^T)$ of *LBD* are normalized separately because of their different magnitudes. Furthermore, to reduce the influence of non-linear illumination changes, the value of each dimension of *LBD* is restrained such that it is smaller than a threshold (0.4 is empirically found to be a good value). Finally, we re-normalize the restrained vector to get a unit *LBD*.

3. Graph matching using spectral technique

After introducing the line detection and description, in this section we present the method to construct the relational graph between two groups of LineVecs and to establish the matching results from this graph. Before that, some pre-processes are introduced first to reduce the dimension of the graph matching problem by excluding the clear non-matches.

3.1. Generating the candidate matching pairs

LineVecs detected in the reference and query images are deemed to be non-matches if they fail to pass the following tests according to their unary geometric attributes and their local appearance similarities.

• *Unary Geometric Attribute:* The unary geometric attribute considered in our work is the direction of LineVecs. Note that lines in the same LineVec have the same direction, so each LineVec has a unique direction. At first glance, the directions of corresponding LineVecs in the image pair are ambiguous and unreliable as image pairs can have arbitrary rotation changes. Though this is exactly true, there is often an approximate global rotation angle between image pairs. We could employ this attribute whenever it is available to reduce the number of candidate matches.

In [6], the approximate rotation relationship between the reference and query images are calculated from the point feature correspondences. Inspired by this, although we do not have such point correspondence information, we can directly compute the LineVec direction histograms of the reference and query images. We first calculate the two direction histograms of LineVecs from two images, then get the normalized histograms $(\mathbf{h}_r, \mathbf{h}_q)$ in which the subscript r denotes the reference image and q denotes the query image. Then we shift $\mathbf{h}_q$ by an angle θ varying from 0 to 2π and search for the approximate global rotation angle θ_g. By taking the angle as index in the histogram for simplicity, θ_g is estimated as:

$$\theta_g = \underset{0 \leqslant \theta \leqslant 2\pi}{\operatorname{argmin}} \|\mathbf{h}_r(x) - \mathbf{h}_q(x - \theta)\|. \quad (5)$$

Since it is not always suitable to approximate the perspective transformation of images by a global rotation change, we have to check whether the estimated rotation angle is genuine. In practice, if the perspective transformation can be approximated by a rotation, then the shifted histogram distance $\|\mathbf{h}_r(x) - \mathbf{h}_q(x - \theta_g)\|$ is small. Fig. 3 gives an example of line direction histograms of an image pair. The estimated θ_g is 0.349 rad and the shifted histogram distance is 0.243. Besides, if the repeatability of the extracted lines in the images is low, then the histogram based method may fail, i.e. a wrong rotation angle may be accepted by the algorithm. To improve the robustness of this method, for lines falling in the same bin of the direction histogram, their length are accumulated as well. So, corresponding to a direction histogram, there is a length vector whose ith element is the accumulated length of all lines falling in the ith bin of the direction histogram. In our implementation, we accept the estimated global rotation angle when the minimal shifted histogram distance is smaller than a threshold t_h and the minimal shifted length vector distance is smaller than a threshold t_l. In Section 5.1, we will experimentally discuss these two thresholds. Once θ_g is accepted, for a pair of LineVecs to be matched, if $|\alpha - \theta_g| > t_\theta$ in which α is the angle between their directions, they are considered to be a non-match without further checking their appearance similarities. If there is no accepted rotation angle between two images, then only the appearance similarities will be tested.

• *Local Appearance Similarity:* The local appearance similarity is measured by the distance of line descriptors. For each line in the LineVec, we generate a LBD descriptor vector V from the octave image where the line is extracted. When matching two sets of LineVecs extracted from an image pairs, the distances between all descriptors of a reference LineVec and a test LineVec are evaluated, and the minimal descriptor distance is used to measure the LineVec appearance similarity s. If $s > t_s$ in which t_s is the local appearance dissimilarity tolerance, then the corresponding two LineVecs will not be considered further.

After checking the unary geometric attribute of LineVecs and their local appearance similarities, the pairs passing these tests are taken as candidate matches. A set of loose thresholds should be chosen, otherwise there will be a larger chance of missing correct matches. In our implementation, the thresholds are empirically set as $t_\theta = \pi/4$, and $t_s = 0.35$. The number of candidate matches is quite larger than the number of real matches because one can not only rely on the aforementioned verifications to decide the final matching results. However, the checking still significantly reduces the dimension of the following graph matching problem compared with direct combinations.

3.2. Building the relational graph

For a set of candidate matches, we build a relational graph whose nodes represent the potential correspondences and the weights on the links represent pairwise consistencies between them.

Given a set of κ candidate matches, the relational graph is represented by an adjacency matrix A with a size of $\kappa \times \kappa$ following the terminology in [30]. The value of the element in row i and column j of A is the consistent score of candidate LineVec matches (L_r^i, L_q^i) and (L_r^j, L_q^j) where L_r^i, L_r^j are LineVecs in the reference image

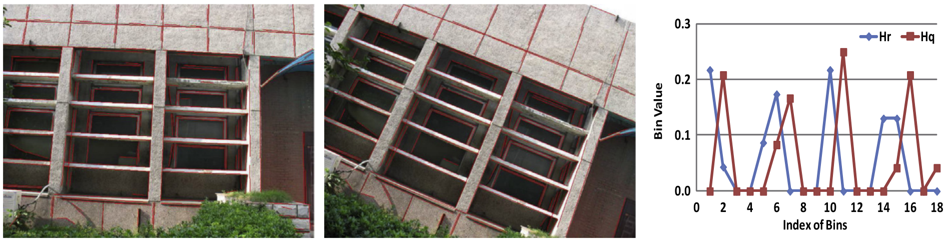


Fig. 3. Illustration of line direction histograms. The first two images show the reference and query images with detected lines and the plot shows their direction histograms. The resolution of each bin is 20°, so there are 18 bins for each histogram.

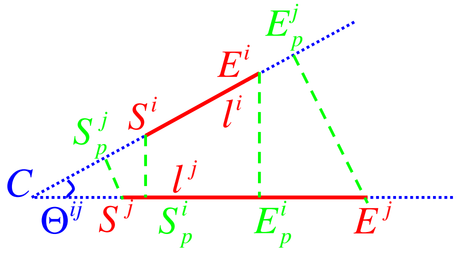


Fig. 4. Illustration of the pairwise geometric attributes. C is the intersection of two lines. (S^i, E^i) are endpoints of the line l^i and (S_p^i, E_p^i) are their projections onto the line l^j. Similarly, (S^j, E^j) are endpoints of the line l^j and (S_p^j, E_p^j) are their projections onto the line l^i.

and $L_q^i,\ L_q^j$ are LineVecs in the query image. The consistent score is computed from the pairwise geometric attributes and appearance similarities of the candidate matched pairs.

For describing the pairwise geometric attributes of two LineVecs (L^i, L^j), we choose two lines (l^i, l^j) which lead to the minimal descriptor distance between these two LineVecs and locate their endpoint positions in the original images. Then we describe the geometric attributes of (l^i, l^j) by their intersection ratios (I^i, I^j), projection ratios (P^i, P^j) and relative angle Θ^{ij} as shown in Fig. 4. I^i and P^i are computed as:

$$I^i = \frac{\overrightarrow{S^iC} \cdot \overrightarrow{S^iE^i}}{|\overrightarrow{S^iE^i}|^2}, \quad P^i = \frac{|\overrightarrow{S^iS^i_p}| + |\overrightarrow{E^iE^i_p}|}{|\overrightarrow{S^iE^i}|}. \tag{6}$$

I^j and P^j can be calculated in the same way. The relative angle Θ^{ij} is easily calculated from the line directions. These three attributes are invariant to changes of translation, rotation, and scale.

As introduced in Section 3.1, we use the LBD descriptor vector V to represent the local appearance of a line. Supposing the descriptors with minimal distances for LineVecs (L_r^i, L_q^i) in the reference and query images are (V_r^i, V_q^i) and for LineVecs (L_r^j, L_q^j) are (V_r^j, V_q^j) respectively, we get two sets of pairwise geometric attributes and local appearances for two candidate matches (L_r^i, L_q^i) and (L_r^j, L_q^j) as: $\{I_r^i, I_r^j, P_r^i, P_r^j, \Theta_r^{ij}, V_r^i, V_r^j\}$ and $\{I_q^i, I_q^j, P_q^i, P_q^j, \Theta_q^{ij}, V_q^i, V_q^j\}$. Then the consistent score A^{ij} is computed as:

$$A^{ij} = \begin{cases} 5 - d_I - d_P - d_\Theta - s_V^i - s_V^j, & \text{if } \Gamma \text{ is true;} \\ 0, & \text{else,} \end{cases} \tag{7}$$

where $d_I,\ d_P$ and d_Θ are the geometric similarities; $s_V^i,\ s_V^j$ are the local appearance similarities; and Γ is the condition. They are defined as:

$$\begin{cases} d_I = min\left(\frac{|I_r^i - I_q^i|}{t_I}, \frac{|I_r^j - I_q^j|}{t_I}\right); \\ d_P = min\left(\frac{|P_r^i - P_q^i|}{t_P}, \frac{|P_r^j - P_q^j|}{t_P}\right); \\ d_\Theta = \frac{|\Theta_r^{ij} - \Theta_q^{ij}|}{t_\Theta}; \\ s_V^i = \frac{\|V_r^i - V_q^i\|}{t_s}; \\ s_V^j = \frac{\|V_r^j - V_q^j\|}{t_s}; \\ \Gamma \equiv \{d_I, d_P, d_\Theta, s_V^i, s_V^j\} \leqslant 1, \end{cases} \tag{8}$$

where $\Gamma \leqslant 1$ means that each element in Γ is not larger than 1. Compared with [5], the definition of d_I in our work is more robust against the fragmentation problem of line detection because only if one pair of matched lines in the reference and query images is well extracted, then d_I could be very small no matter how badly the other pair is extracted. The definition of d_P shares the same advantage. $t_I,\ t_P,\ t_\Theta$ and t_s are thresholds. In our implementation, they are set as $t_I = 1,\ t_P = 1,\ t_\Theta = \pi/4$ and $t_s = 0.35$. For all the candidate matches, we compute the consistent score among them and obtain the adjacency matrix A. The diagonal elements of A equal zero as suggested by Leordeanu [31] for better results and let $A^{ji} = A^{ij}$ to keep the symmetry.

3.3. Generating the final matching results

The matching problem is now reduced to finding the cluster of matches $\mathcal{LM}$ that maximizes the total consistent scores $\sum_{(L_r^i, L_q^i),(L_r^j, L_q^j) \in \mathcal{LM}} A^{ij}$ such that the mapping constraints are met. We use an indicator vector x to represent the cluster such that $x(i) = 1$ if $(L_r^i, L_q^i) \in \mathcal{LM}$ and zero otherwise. Thus, the problem is formulated as:

$$x^* = argmax(x^T A x), \tag{9}$$

where x is subject to the mapping constraints. The general quadratic programming techniques are too computationally expensive to solve this problem. We employ the spectral technique which relaxes both the mapping constraints and the integral constraints on x such that its elements can take real values in $[0, 1]$.

By the Raleigh's ratio theorem [30], the x^* that will maximize $x^T A x$ is the principal eigenvector of A. What still remains is to binarize the eigenvector using mapping constraints and obtain a robust approximation of the optimal solution. The mapping constraints applied here are the sidedness constraint [16,14] and the one-to-one constraint. Details of the algorithm are as follows:

1. Extract LineVecs from the reference and query images by EDLine [22] in the scale space to obtain two sets of LineVecs $\mathcal{L}_r$ and $\mathcal{L}_q$;
2. Estimate the global rotation angle θ_g of the image pair from the direction histograms of $\mathcal{L}_r$ and $\mathcal{L}_q$;
3. Compute the LBD descriptors of LineVecs in $\mathcal{L}_r$ and $\mathcal{L}_q$;
4. Generate a set of candidate matches $\mathcal{CM} = \{(L_r^1, L_q^1), (L_r^2, L_q^2), \ldots, (L_r^\kappa, L_q^\kappa)\}$ by checking the unary geometric attribute and local appearance similarities of LineVecs in $\mathcal{L}_r$ and $\mathcal{L}_q$;
5. Build the adjacency matrix A with a size of $\kappa \times \kappa$ according to the consistence scores of pairs in $\mathcal{CM}$;
6. Get the principal eigenvector x^* of A by using ARPACK[32];
7. Initialize the matching result: $\mathcal{LM} \leftarrow \text{ø}$;
8. Find $a = \underset{1 \leqslant i \leqslant \kappa}{\text{argmax}}\,(x^*(i))$. If $x^*(a) = 0$, then stop and return the matching result $\mathcal{LM}$. Otherwise, set $\mathcal{LM} = \mathcal{LM} \cup \{(L_r^a, L_q^a)\}$, $\mathcal{CM} = \mathcal{CM} - \{(L_r^a, L_q^a)\}$ and $x^*(a) = 0$.
9. Check all the candidates in $\mathcal{CM}$. If (L_r^j, L_q^j) conflicts with (L_r^a, L_q^a), then set $\mathcal{CM} = \mathcal{CM} - \{(L_r^j, L_q^j)\}$ and $x^*(j) = 0$.
10. If $\mathcal{CM}$ is empty, then return $\mathcal{LM}$. Otherwise go back to Step 8.

The final line matches can be directly retrieved from the matching results of LineVecs $\mathcal{LM}$. Notice that, lines in the LineVec are located in the same region of image with the same direction, hence, for each pair of LineVec matches, it is enough to retrieve only one pair of line matches.

4. The descriptor performance evaluation

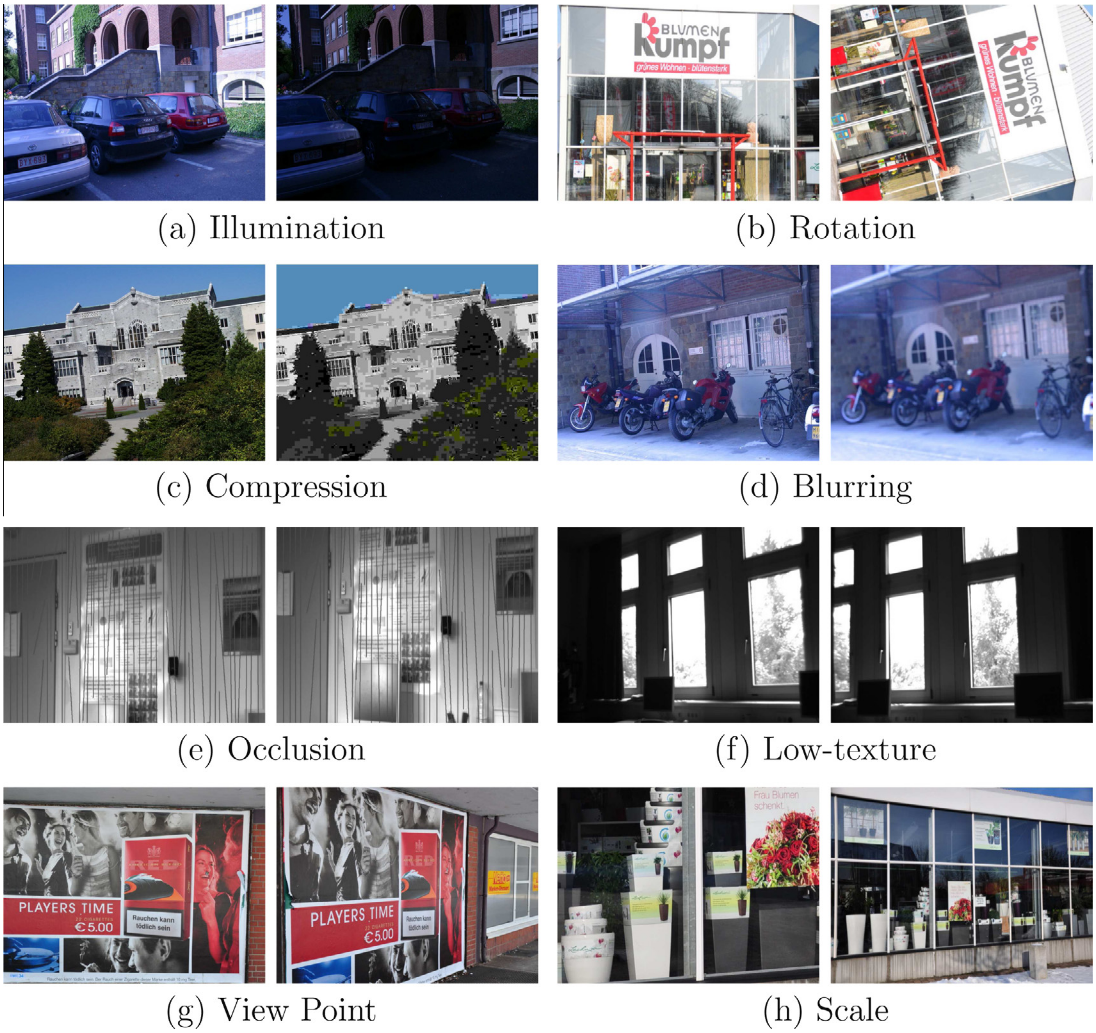
Before testing the proposed graph matching algorithm, we first analyze the influence of the LSR parameters, i.e. the number of bands m and the width of each band w, then evaluate the performance of LBD by comparing it with the well-known MSLD [11] descriptor.

Mikolajczyk and Schmid [33] established a benchmark to evaluate the performance of the local descriptors. We employ this framework to compare the performance of the line descriptors. The dataset in this experiment includes eight groups of images with following transformations: illumination changes, in-plane rotation, JPEG compression, image blurring, image occlusion, view point changes in the low-texture scene and the texture scene, and scale variations. There are six images in each group of sequence raising from small to large image transformations. Fig. 5 shows example images of our dataset, image sets of (a), (c) and (d) are from [33], and the rest are captured by ourselves to make sure images contain some line features. The images are either of planar scenes or the camera position was fixed during acquisition. Therefore, they are always related by a homography (plane projective transformation). The ground truth homographies are known. In order to better evaluate the descriptor performance for different image transformations, in this section we only consider lines extracted in the original image rather than in the octave images.

Since the ground truth of the image homographies is available, we first transfer the extracted lines in the query image into the reference image, then establish the ground truth of line correspondences by searching the parallel and close reference lines of the transferred lines. For the matching performance of descriptors reported in this section, we choose the nearest neighbor matching criterion to match lines according to their descriptor distance avoiding the prejudice of a distance threshold because different kinds of descriptors prefer different thresholds. Another advantage of this matching criterion is that the recall ratio (the number of correct matches divided by the number of ground correspondences) and the matching precision (the number of correct matches divided by the number of total matches) are only decided by the number of correct matches because the denominators for different descriptors are equal.

4.1. The descriptor dimension

The influence of the LSR parameters are analyzed experimentally. We vary m and w from 3 to 13, respectively. Fig. 6 shows how the number of correct matches of all images is influenced by these two parameters. It is clear that LBD and MSLD share similar rules: the performance increases fast at the beginning with the increment of m or w, then reaches the best performance when $m = 9$ and w is about 7 or 9, after that there is a steady performance decrease. The results are well explained by the fact that larger values of m and w (i.e. larger LSR) make the descriptor more distinctive while they also reduce the repeatability of the LSR.


(a) Illumination

(b) Rotation

(c) Compression

(d) Blurring

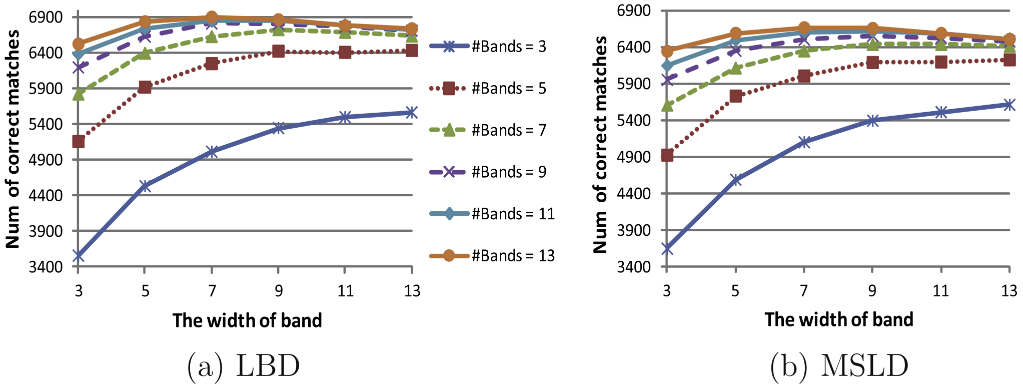
(e) Occlusion

(f) Low-texture

(g) View Point

(h) Scale

Fig. 5. Examples in the image dataset including eight groups of image transformations. For each group, there are six images in the dataset raising from small to large transformations (Images in (f) are generated by increasing baseline between views in the low texture environment). The first and the last images of each group are shown here. The left image in each pair is chosen as the reference image.



(a) LBD (b) MSLD

Fig. 6. The analysis of the descriptor dimension. The numbers of bands and the width of each band both vary from 3 to 13.

Table 1
The time performance of descriptors (LBD, MSLD). The results are given in ms by varying the number of bands m and the width of each band w. The time is measured on a 3.4 GHz Intel (R) Core2 processor with 8 GB of RAM.

m	w					
	3	5	7	9	11	13
3	4 , 13	7, 27	9, 41	12, 62	14, 85	17, 115
5	6, 23	12, 44	15, 72	20, 107	24, 151	28, 208
7	9, 33	15, 63	21, 103	28, 154	34, 217	42, 303
9	12, 42	20, 82	28, 137	37, 200	45, 281	52, 383
11	15, 52	24, 100	34, 165	44, 245	53, 346	63, 470
13	17 , 61	29, 119	40, 195	51, 296	65, 418	74, 566

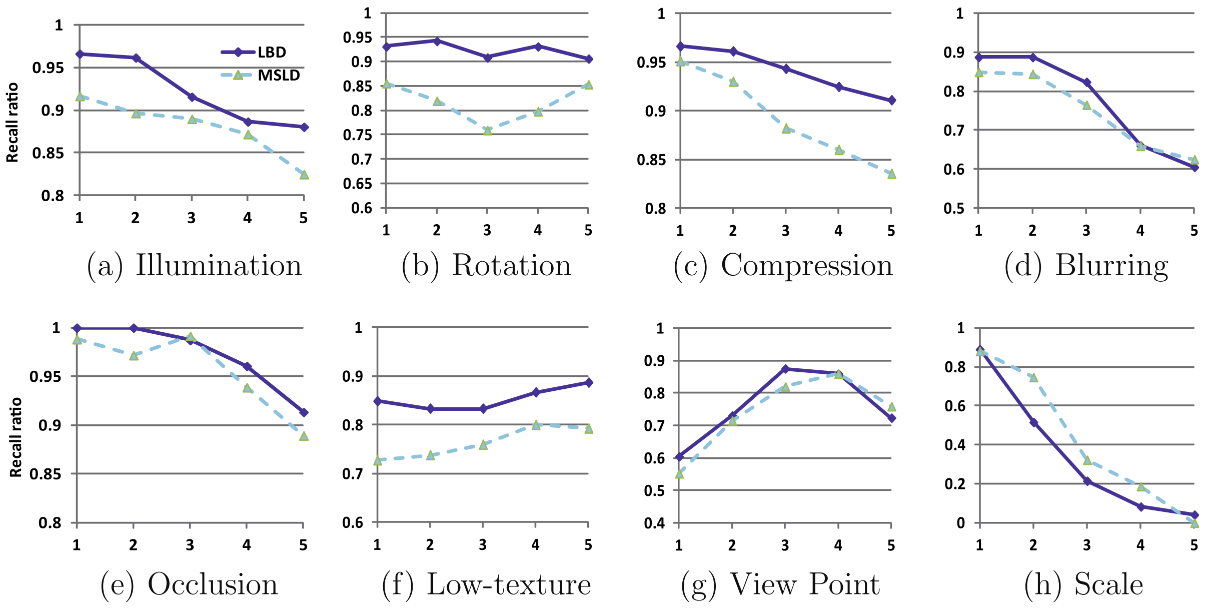
We also evaluate the time performances of these two descriptors which are given in Table 1. Although the time performances may change from image to image, their relative relationship will keep the same. We only show the results which are generated from an example image with the size of 900 × 600 and 573 extracted lines. Basically, the larger m and w are, the more computational time is consumed. LBD is less sensitivity to the increase of m and w than that of MSLD, especially for the increase of w.

Based on the aforementioned evaluation, through the rest of the paper, the descriptor will be computed from a LSR with $m = 9$ and $w = 7$, resulting in a 72-dimensional descriptor. Then the computing times of LBD and MSLD for the example image are 28 ms, and 137 ms, respectively.

4.2. Further comparison of MSLD and LBD

In this section, we report the comparison details of the descriptor performance for each group of images in our dataset (Fig. 5). For each group of images, the recall ratios of MSLD and LBD are given in Fig. 7.

Fig. 7(a) shows the performances of MSLD and LBD for the image illumination changes. From image 1 to image 5, the lighting condition gets worse. The recall ratios decrease with the increment of the lighting distortion. Fig. 7(b) shows the results for images which are generated by a set of in-plane rotation varying from 15° to 75°. It is interesting that when the rotation angle is 45° (between image 3 and the reference image), LBD and MSLD perform worst because of the aliasing of discrete lines. Fig. 7(c) and (d) show the descriptor performance against the image compression and the image blurring, respectively. Not surprisingly, the performances decrease with the increment of the image compression ratio or the image blurring. Fig. 7(e) shows the descriptor performance against image occlusion. To evaluate the occlusion effect, we first artificially add some vertical line features in a background image, then shift the region of interest along the vertical



(a) Illumination (b) Rotation (c) Compression (d) Blurring

(e) Occlusion (f) Low-texture (g) View Point (h) Scale

Fig. 7. The comparison of the descriptor performance in terms of the recall ratio over five test images for each image transformation in the dataset.

Table 2
The experimental analysis of the direction histogram based rotation estimation method. For each image pair, the following results are reported: the ground truth and the estimated rotation angle in degrees, the minimal shifted histogram distance and length vector distance.

Img	Ground truth and estimated angle					Histogram distance and length vector distance				
	1	2	3	4	5	1	2	3	4	5
a	0, 0	0, 0	0, 0	0, 0	0, 0	0.064, 0.059	0.065, 0.054	0.048, 0.052	0.079, 0.129	0.108, 0.101
b	15, 10	30, 30	45, 40	60, 60	**75, 70**	0.493, 0.482	0.415, 0.472	0.379, 0.420	0.152, 0.133	**0.528, 0.561**
c	0, 0	0, 0	0, 0	0, 0	0, 0	0.088, 0.087	0.049, 0.406	0.075, 0.052	0.138, 0.081	0.221, 0.161
d	0, 0	0, 0	0, 0	0, 0	0, 0	0.077, 0.067	0.189, 0.134	0.339, 0.251	0.367, 0.259	0.375, 0.301
e	0, 0	0, 0	0, 0	0, 0	0, 0	0.043, 0.043	0.036, 0.055	0.076, 0.145	0.090, 0.150	0.065, 0.122
f	0, 0	0, 0	0, 0	0, 0	0, 0	0.167, 0.103	0.235, 0.286	0.182, 0.196	0.169, 0.208	0.365, 0.299
g	**0, 80**	0, 0	0, 0	0, 0	0, 0	**0.578, 0.626**	0.292, 0.366	0.085, 0.099	0.207, 0.437	0.222, 0.448
h	0, 0	0, 0	0, 0	0, 0	0, 0	0.132, 0.105	0.157, 0.240	0.290, 0.299	0.357, 0.331	0.296, 0.344

Bold font marks the false negative error and the false positive error.

direction of the artificial image to generate a set of smaller images as shown in Fig. 5(e). This process makes sure that for the most of the lines, their LSR in the image sequence will change gradually (some part of the LSR moves out or in). The results show that the descriptor performance decreases with the increment of the image occlusion. Fig. 7(f) shows the descriptor performance in the low-texture scene. Images in this sequence are captured in front of the window with small view point changes. The results do not show drastic change in performance because of the small baseline between images. Fig. 7(g) shows the descriptor performance against large view point change. The view angles between the query images and the reference image range approximately from -70° to 60°. No doubt, the descriptors perform better when the absolute value of the view angle is smaller (image3 and image4). Fig. 7(h) shows the most challenging case for the descriptors, i.e, the large scale change. The scale ratio between the query images and the reference image range from 0.9 to 0.3. The performance decreases fast with the scale change.

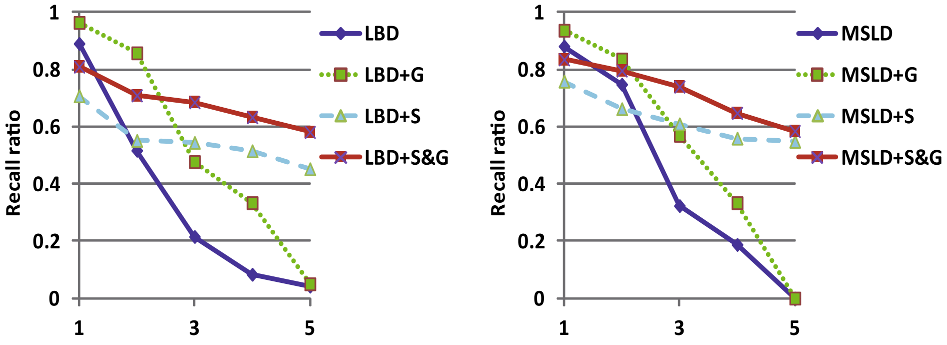
Conclusively, for the most kinds of image transformations (Fig. 7(a)–(f)), it is clear that LBD performs better than MSLD. For the large scale change (Fig. 7(h)), as explained in Section 1.2, all these two descriptors perform badly though MSLD is slightly better, because in this experiment, lines are only extracted in the original image. However, this can be made up by extracting lines in the scale space as addressed in Section 2.1 and will be illustrated in the following experiments.

5. Line matching experiments

In this section, we first experimentally analyze the direction histogram based rotation estimation method proposed in Section 3.1. Then we illustrate the performance improved by the multi-scale line extraction strategy and the geometric consistency verification. At last, we compare the proposed line matching algorithm with the state-of-the-art methods. The following experiments are performed on a 3.4 GHz Intel (R) Core 4 processor with 8 GB of RAM.

5.1. The analysis of the rotation estimation method

In Section 3.1, we propose the direction histogram based rotation estimation method which is employed to reduce the number of candidate matches when the estimated rotation angle is accepted. The robustness of this method is very important for the matching algorithm. Generally, there are two kinds of errors: false negative (i.e. a correct rotation angle is rejected) and false positive (i.e. a wrong rotation angle is accepted). For false negative errors, the matching algorithm will be less efficient because more candidate line matches will be generated without checking their directions, but the matching algorithm can still match lines accurately. For false positive errors, the matching algorithm may fail because the wrong rotation angle is employed to generate wrong candidate line matches. Therefore, our priority goal is to control the false positive error as low as possible while keep small false negative error. In Section 3.1, there are two thresholds: the minimal shifted histogram distance threshold t_h and the minimal shifted length vector distance threshold t_l. It is hard to give a theoretic analysis about the optimal setting of these thresholds because they depends on the scene environment, the image transformation and the line detection method. Here, we use the image set in Fig. 5 to experimentally choose the proper values of these thresholds. For each image transformation, five images are compared to the reference image. Table 2 reports the ground truth and the estimated rotation angle, the minimal shifted histogram distance and the minimal shifted length vector distance between two images. Note that the precision of the estimated rotation angle equals to the resolution of the direction histogram. In our implementation, it is 20°. In this experiment, for the image pair g.1,



(a) LBD (b) MSLD

Fig. 8. Illustration of the performance improvement. In (a), LBD is the original descriptor performance, LBD+G is generated by applying the geometric constraints, LBD+S is generated by detecting lines in the scale space and LBD+S&G is generated by employing both strategies. (b) is related to the MSLD descriptor.

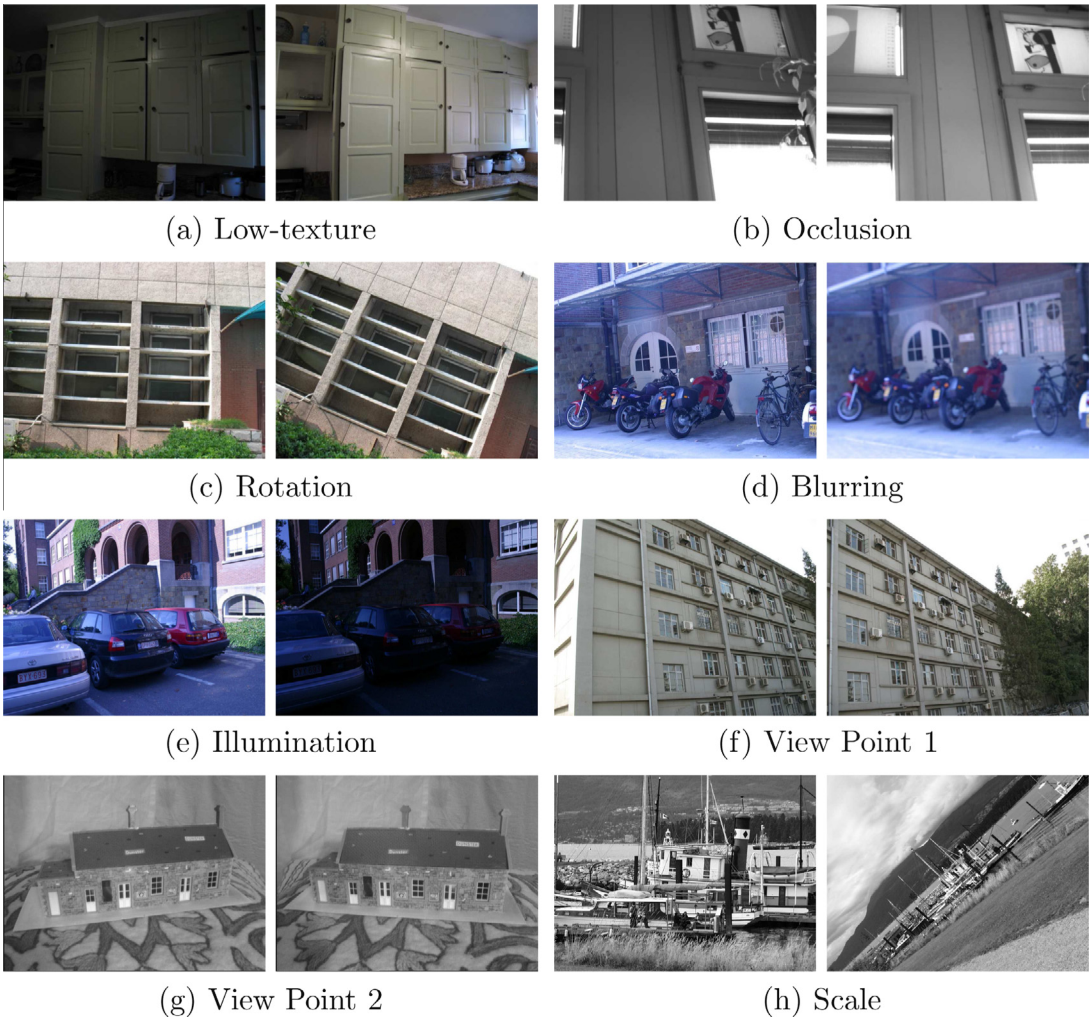

(a) Low-texture
(b) Occlusion
(c) Rotation
(d) Blurring
(e) Illumination
(f) View Point 1
(g) View Point 2
(h) Scale

Fig. 9. Image dataset for comparison experiment. In each group, there are two images with large transformations.

the estimated rotation angle is obviously wrong. Hence, it should be rejected. In order to control the false positive error, we choose a pair of conservative thresholds ($t_h = t_l = 0.5$) although there is a false negative error (the correct estimated rotation angle is rejected for image pair b.5). This threshold setting generally works well even for a challenging image set as shown in the following experiment. If the efficiency of the matching algorithm is not important while matching failure is definitely not acceptable, then one can set t_h and t_l to zero, i.e. the estimated rotation angle will always be rejected and the unary attribute of lines will never be employed to generate the candidate matches.

Table 3
The summary of parameters of the LBD+S&G algorithm and their settings in the experiments.

Descriptor	m	Number of bands	9
	w	Width of band	7
Histogram	t_h	Histogram distance threshold	0.5
	t_l	Length vector distance threshold	0.5
Consistency	t_I	Intersection ratio difference threshold	1
	t_P	Projection ratio difference threshold	1
	t_θ	Relative angle difference threshold	$\pi/4$
	t_s	Appearance dissimilarity threshold	0.35

5.2. The improvement of the matching performance

As introduced in Section 1.2, in order to mitigate the problem of segment fragmentation and image variation, we propose to extract lines in the scale space (marked as +S). Besides, the geometric constraints are applied to improve the matching robustness (marked as +G). The sign +S&G denotes both strategies are applied. We now show the influence of these two strategies on the matching performance. Limited by the space, we only illustrate the performance improvement for the large scale changes although the approaches are generally effective for other image transformations. Fig. 8 shows the improved matching performance of the local appearance based algorithms (LBD and MSLD). Taken Fig. 8(a) for example, when the scale variation is large, the performance gain of LBD+S&G is obvious. Although when the scale variation is small, the recall ratio of LBD+S&G is slightly smaller than that of LBD or LBD+G because a larger number of lines are extracted in the scale space than in the original image. The results also show that LBD+G is better than LBD, but it is still not robust to scale changes because we employ the geometric consistency verification as post process. When both strategies are applied, the matching performance of LBD+S&G is less sensitive to the scale changes and always better than LBD+S. The proposed two strategies are also effective for the MSLD based matching algorithms as shown in Fig. 8(b). The performance comparison between LBD+S&G and MSLD+S&G together with other state-of-the-art methods are presented in the next section.

Table 4
Comparison of our approach (LBD+S&G) with three line matching algorithms (MSLD[11]+S&G, LP [6], LS [5]). For each image pair, the following results are reported: the number of total matches, the matching precision and the computing time.

	Img	LBD+S&G	MSLD+S&G	LP	LS		Img	LBD+S&G	MSLD+S&G	LP	LS		Img	LBD+S&G	MSLD+S&G	LP	LS
Total matches	a	54	44	12	54	Match precision (%)	a	94	92	67	96	Time (s)	a	0.11	0.35	5.5	8
	b	54	57	50	76		b	100	100	94	100		b	0.04	0.07	4.5	1
	c	263	240	253	188		c	100	100	100	100		c	0.38	0.48	13	26
	d	106	121	101	43		d	100	98	100	100		d	0.55	0.59	38	5
	e	245	223	262	241		e	100	100	100	100		e	0.59	0.65	28	8
	f	446	445	422	281		f	100	100	100	100		f	1.75	2.49	31	10
	g	87	78	117	151		g	100	100	91	98		g	0.20	0.42	22	8
	h	44	33	54	14		h	95	88	76	29		h	0.51	0.54	54	8

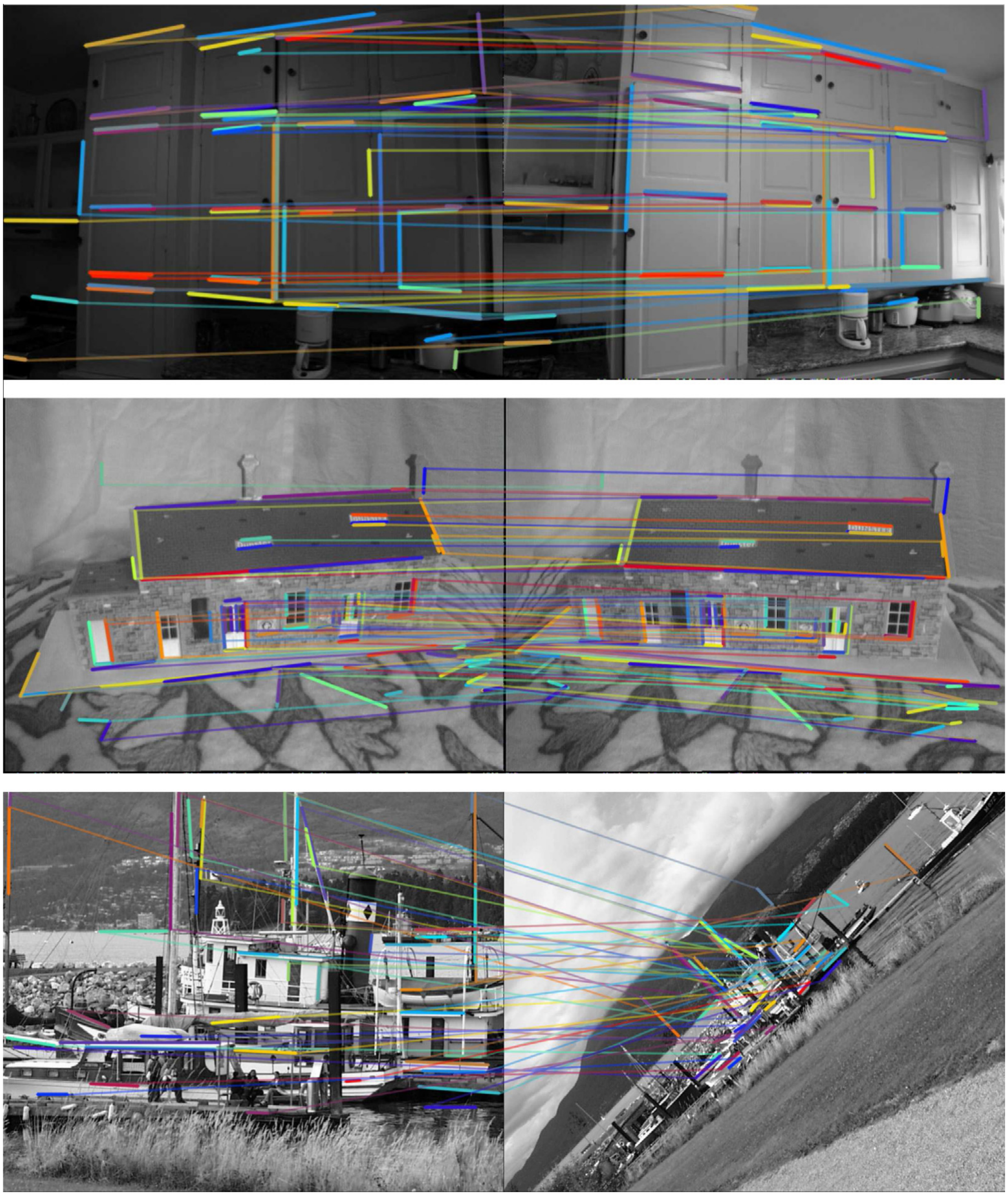


Fig. 10. Illustration of the LBD+S&G matching results.

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

For a fair comparison with previous work, in this section we conduct the comparison experiment on the images demonstrated in the literature [5,6] (except the occlusion image pair which is captured in our office). The differences between the image set in Fig. 9 and the previous image set (Fig. 5) are that: (1), this image set only includes an image pair rather than an image sequence for each group of transformation; (2), most of the image pairs in this image set are more challenging. As introduced in Section 1, the existing approaches to match lines are mainly of three types. Hence, the proposed algorithm LBD+S&G, is compared with three representatives from three groups which are recently reported to feature remarkable performance: the Line Signature (LS) [5], the Line matching leveraged by Point correspondences (LP) [6] and the Mean-Standard deviation Line Descriptor (MSLD) [11] (here, we use MSLD+S&G instead because it performs better than MSLD as illustrated in Fig. 8(b)). The implementations of LS and LP are by the courtesy of their authors while MSLD+S&G is implemented by ourselves with parameter settings as recommended by its authors.

In this experiment, the line detectors used by LBD+S&G and MSLD+S&G are the same as described in Section 2.1. The extracted LineVecs are also used as input to LP. For LS, it uses its own multiscale line detector because the structure of line signatures in [5] is quite different from the structure of LineVecs. The parameter settings of the proposed algorithm are summarized in Table 3. The comparison results are given in Table 4. All the matched lines are checked one by one manually to test whether a matched line pair is correct or not. It is clear that LP is less performing for the low-texture scene, because the local appearances of lines are indistinguishable and the images lack stable point correspondences. The results also show that LBD+S&G performs slightly better than MSLD+S&G even for images with large scale variations because the drawback of LBD is made up by the multi-scale line extraction strategy. Surprisingly, LS has a bad matching result for image pair (h) which is inconsistent with the result presented in [5]. If we change the role of reference and query images, then we can get the same good result. This illustrates that the matching results of LS are dependent on the order of images in a pair. Compared to MSLD+S&G, LP and LS, the most superior feature of LBD+S&G is its time performance. Here, the time of LP given in Table 4 is its complete processing time which includes generating point correspondences and matching lines.

Fig. 10 illustrates the matching results of LBD+S&G in three challenging scenes. The matched lines in each pair are assigned the same color and one of their endpoints are connected to illustrate their correspondences. These figures are better viewed in color. The first image pair in Fig. 10 is a low-texture planar scene with illumination and view point changes. The second image pair in Fig. 10 is a non-planar scene with moderate view point changes. The last image pair in Fig. 10 is a textured scene with strong scale and rotation variations. The matching algorithm is less performing for these three image pairs than for the rest of image pairs. Nevertheless, the results shown in Fig. 10 are still quite acceptable and establish many line correspondences with few mismatches.

Note that similar to the parameter detection methods adopted in [5,6], it is empirical to find the good parameter settings. However, these parameter settings are fixed as presented in Table 3 for all the experiments. The results show that the algorithm works well for a large range of image variations.

6. Conclusion and discussion

We address the problem of line matching for image pairs under various situations: low-texture scenes, partial occlusion, rotation changes, blurred images, illumination changes, moderate view-point changes, and scale changes. We show the robustness and the efficiency of our graph matching process. The good performance achieved by the proposed algorithm is mainly because we detect lines in the scale space and combine the local appearance and geometric constraints together which eliminates lots of mismatches. The source code of the proposed algorithm, the image dataset and the matching results are available on our website.[1]

The dimension of the LBD descriptor to characterize the local appearance of line segment is a 72-vector with float elements. For retrieving line features from a large data base, it may be not efficient enough. The idea of Brief [34] or its enhancements Brisk [25] and ORB [26] could be employed to improve the matching process. In order to conduct a set of meaningful binary tests, the LSR should be normalized to a regular region with a fixed size. It is worth to address this in the future.

Besides, the geometric constraints are enforced globally in this paper by using the spectral technique. For images undergoing a moderate view point transformation, the global geometric constraints are maintained well. For strong wide baseline images of the non-planar scenes, the global constraints may be violated, then it is better to enforce the local geometric constraints like the approach in [5] although it is more time consuming.

Acknowledgments

This work was supported by China Scholarship Council (No. 2009611008). The authors thank the anonymous reviewers for their valuable comments that helped to improve the paper.

References

[1] C.J. Taylor, D.J. Kriegman, Structure and motion from line segments in multiple images, PAMI 17 (1992) 1021–1032.
[2] P. David, D. DeMenthon, Object recognition in high clutter images using line features, in: ICCV, 2005, pp. 1581–1588.
[3] M. Chandraker, J. Lim, D. Kriegman, Moving in stereo: efficient structure and motion using lines, in: ICCV, 2009, pp. 1741–1748.
[4] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, PAMI 27 (2005) 1615–1630.
[5] L. Wang, U. Neumann, S. You, Wide-baseline image matching using line signatures, in: ICCV, 2009, pp. 1311–1318.
[6] B. Fan, F. Wu, Z. Hu, Line matching leveraged by point correspondences, in: CVPR, 2010, pp. 390–397.
[7] L. Zhang, R. Koch, Line matching using appearance similarities and geometric constraints, DAGM-OAGM, Graz, LNCS 7476 (2012) 236–245.
[8] R. Deriche, O. Faugeras, Tracking line segments, Image and Vision Computing 8 (1990) 261–270.
[9] P. Neubert, P. Protzel, T. Vidal-Calleja, S. Lacroix, A fast visual line segment tracker, in: ETFA, 2008, pp. 353–360.
[10] D.-M. Woo, D.-C. Park, S.-S. Han, S. Beack, 2d line matching using geometric and intensity data, in: AICI, 2009, pp. 99–103.
[11] Z. Wang, F. Wu, Z. Hu, Msld: a robust descriptor for line matching, PR 42 (2009) 941–953.
[12] N. Ayache, B. Faverjon, Efficient registration of stereo images by matching graph descriptions of edge segments, IJCV 1 (1987) 107–131.
[13] W. Christmas, J. Kittler, M. Petrou, Structural matching in computer vision using probabilistic relaxation, PAMI 17 (1995) 749–764.
[14] R. Horaud, T. Skordas, Stereo correspondence through feature grouping and maximal cliques, PAMI 11 (1989) 1168–1180.
[15] R.C. Wilson, E.R. Hancock, Structural matching by discrete relaxation, PAMI 19 (1997) 634–648.
[16] H. Bay, V. Ferrari, L.V. Gool, Wide-baseline stereo matching with line segments, in: CVPR, 2005, pp. 329–336.
[17] C. Schmid, A. Zisserman, Automatic line matching across views, in: CVPR, 1997, pp. 666–671.
[18] 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.
[19] H. Kim, S. Lee, Wide-baseline image matching based on coplanar line intersections, in: IROS, 2010, pp. 1157–1164.
[20] L.J. Chmielewski, Scale and rotation invariance of the evidence accumulation-based line detection algorithm., in: CORES, 2005, pp. 363–370.
[21] J.L.C. Amaury Negre, C. Laugier, Scale invariant segment detection and tracking, in: ISER, 2008.
[22] C. Akinlar, C. Topal, Edlines: a real-time line segment detector with a false detection control, Pattern Recognition Letters 32 (2011) 1633–1642.
[23] D.G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2004) 91–110.
[24] H. Bay, A. Ess, T. Tuytelaars, L.V. Gool, Surf: speeded up robust features, CVIU 110 (2008) 346–359.
[25] M.C. Stefan Leutenegger, R. Siegwart, Brisk: Binary robust invariant scalable keypoints, in: ICCV, 2011.
[26] K.K. Ethan Rublee, Vincent Rabaud, G. Bradski, Orb: an efficient alternative to sift or surf, in: ICCV, 2011.
[27] T. Tuytelaars, L. Van Gool, Matching widely separated views based on affine invariant regions, IJCV 59 (2004) 61–85.

[1] http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=Lilian+Zhang.

[28] K. Mikolajczyk, A. Zisserman, C. Schmid, Shape recognition with edge-based features, in: BMVC, 2003.

[29] J. Meltzer, S. Soatto, Edge descriptors for robust wide-baseline correspondence, in: CVPR, 2008, pp. 1–8.

[30] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: ICCV, 2005, pp. 1482–1489.

[31] M. Leordeanu, R. Sukthankar, M. Hebert, Unsupervised learning for graph matching, IJCV (2011) 1–18.

[32] R. Lehoucq, K. Maschhoff, D. Sorensen, C. Yang, Arpack software, 2011, <http://www.caam.rice.edu/software/ARPACK/>.

[33] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, PAMI 27 (2005) 1615–1630.

[34] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, in: ECCV, 2010.