# Line Matching Using Appearance Similarities and Geometric Constraints

Lilian Zhang* and Reinhard Koch

Institute of Computer Science, University of Kiel, Germany
{lz,rk}@mip.informatik.uni-kiel.de

**Abstract.** Line matching for image pairs under various transformations is a challenging task. In this paper, we present a line matching algorithm which considers both the local appearance of lines and their geometric attributes. A relational graph is built for candidate matches and a spectral technique is employed to solve this matching problem efficiently. Extensive experiments on a dataset which includes various image transformations validate the matching performance and the efficiency of the proposed line matching algorithm.

## 1 Introduction

One of the challenging areas in computer vision is feature matching which is a basic tool for applications in scene reconstruction, pattern recognition and retrieval, motion estimation, and so on. Most of the existing matching methods in the literature are based on local point or region features [1] which are deficient for low-texture scenes [2]. On the contrary, line features are often abundant in these situations. Moreover, line features and other local features provide complementary information about the scenes. Therefore line matching is both desirable and indispensable in many applications. Although some progress was achieved recently for the line matching problem [2,3], they are quite computationally expensive, prohibiting their usage in many applications. This paper addresses the problem of robust and efficient line feature matching.

Several reasons make line matching a difficult problem, including: inaccurate locations of line endpoints, fragmentation of lines, lack of strongly disambiguating geometric constraints, and lack of distinctive appearance in low-texture scenes [3,4]. 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 image scale changes. The second is to check the consistency of line pairs in a loose way which combines the appearance similarities and geometric constraints. The third novel part is to solve the matching problem by a spectral method [5] 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

* This work was supported by China Scholarship Council (No.2009611008). Special thanks to Markus Franke whose work was valuable in improving this paper.

A. Pinz et al. (Eds.): DAGM/OAGM 2012, LNCS 7476, pp. 236–245, 2012.

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's 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.

## 2 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 stereo where extracted corresponding segments are similar, approaches based on matching individual lines are suitable [6,7] because of their better computational performance. Among the first group, Wang et al. [8] propose 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 [9,10,11]. Bay et al. [12] present 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. [2] use line signatures to match lines between wide baseline images. To overcome the unreliable line detection problem, a multi-scale line extraction strategy is employed which significantly improves the repeatability of line signatures and therefore has a good matching performance. However, this method is quite computationally expensive.

Given a set of point correspondences, Schmid and Zisserman [4] take the epipolar constraint of line endpoints for short baseline matching and present a plane sweep algorithm for wide baseline matching. More recently, Fan et al. [3] explore an affine invariant from two points and one line. They utilize 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.

The rest of this paper is organized as follows. Sec.3 presents the way to extract lines in the scale space and generate the candidate matched pairs. Sec.4

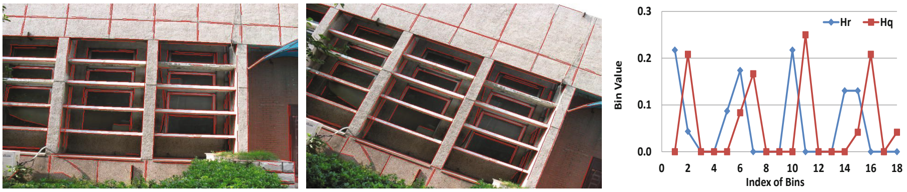

**Fig. 1.** Illustration of line direction histograms. The first two images show the reference and query images with detected lines and the plot shows their direction histograms. The resolution of each bin is 20 degrees, so there are 18 bins for each histogram.

introduces the process to build the relational graph and the spectral technique to solve the graph matching problem. The experimental results are reported in Sec.5. Finally, we draw the conclusion in the last section.

## 3 Generating the Candidate Matching Pairs

### 3.1 Detecting Lines in the Scale Space

To overcome the fragmentation problem of line detection and 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 downsampling 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 [13] algorithm to each octave producing a set of lines in the scale space. Then we re-organize them by finding corresponding lines in the scale space. For all lines extracted in the scale space, they will be 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. The line detecting approach is different from Wang's [2] by reorganizing all line segments detected in the scale-space to form LineVecs which reduce the dimension of the graph matching problem.

### 3.2 Unary Geometric Attribute

The unary geometric attribute considered in our work is the direction of lines. At first glance, this attribute is 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 [3], the approximate rotation relationship between the reference and query images are calculated from the point feature correspondences. Inspired by this, although we don't have such point correspondence information, we can directly compute the line direction histograms of the reference and query images. Here, the line direction is given by making the gradients of most edge pixels pointing from its left side to its right side. Note that, lines in the same LineVec have the same direction, so each LineVec has a unique direction. We first calculate the two direction histograms of LineVecs from two images, then normalize them to get $(\mathbf{h}_r, \mathbf{h}_q)$ in which the subscript $r$ denotes the reference image and $q$ denotes the query image. Then we shift $\mathbf{h}_q$ by a angle $\theta$ varying from 0 to $2\pi$ and search for the approximate global rotation angle $\theta_g$. By taking the angle as index in the histogram for simplicity, $\theta_g$ is estimated as:

$$\theta_g = \underset{0 \leq \theta \leq 2\pi}{\operatorname{argmin}} \|\mathbf{h}_r(x) - \mathbf{h}_q(x - \theta)\|. \tag{1}$$

Since it's 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.1 gives an example of line direction histograms of an image pair. The estimated $\theta_g$ is 0.349 rad and the shifted histogram distance is 0.243. In our implementation, we accept the estimated global rotation angle when the shifted histogram distance is smaller than 0.5. Once $\theta_g$ is accepted, for a pair of LineVecs to be matched, if $|\alpha - \theta_g| > t_\theta$ in which $\alpha$ 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 are applied.

### 3.3 Local Appearance Similarity

There are few robust line descriptors based on the local appearance of a line in the literature. Two of them have remarkable performance: one takes the mean and standard deviation of the pixel gradients in a region centered at a line as its descriptor [8] and the other computes the line band descriptor (LBD) based on the band representation of lines [14]. As LBD is computationally more efficient and has better performance for most of image transformations, in our work we choose it to measure the appearance similarity of a pair of LineVecs.

Each LineVec may include more than one line in the scale space. For each line in the LineVec, we will generate a LBD1 descriptor [14] from the octave image where the line is extracted. In our work, for a line we use 9 bands each with width of 7 pixels, resulting in a 72-dimensional LBD1 descriptor vector $V$. 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 won't be considered further.

After checking the unary geometric attribute of LineVecs and their local appearance similarities, the pairs passing the test 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 set as $t_\theta = \pi/4$, and $t_s = 0.35$. The number of candidate matches is quite larger

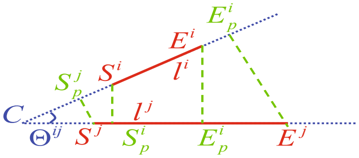

**Fig. 2.** Illustration of the pairwise geometric attributes. $C$ is the intersection of two lines. $(S^i, E^i)$ are endpoints of the line $l^i$ and $(S^i_p, E^i_p)$ are their projections onto the line $l^j$. Similarly, $(S^j, E^j)$ are endpoints of the line $l^j$ and $(S^j_p, E^j_p)$ are their projections onto the line $l^i$.

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.

# 4 Graph Matching Using Spectral Technique

For a given set of candidate matches, we build the relational graph whose nodes represent the potential correspondences and the weights on the links represent pairwise consistencies between them. Then recover the correct matches by using the spectral technique and imposing the mapping constraints.

## 4.1 Building the Relational Graph

Given a set of $n$ candidate matches, the relational graph is represented by an adjacency matrix $M$ with a size of $n \times n$ following the terminology in [5]. The value of the element in row $i$ and column $j$ of $M$ is the consistent score of candidate LineVec matches $(L^i_r, L^i_q)$ and $(L^j_r, L^j_q)$ where $L^i_r, L^j_r$ are LineVecs in the reference image and $L^i_q, L^j_q$ 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. Then referring to the work in [2], 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 $\Theta^{ij}$ as shown in Fig.2. $I^i$ and $P^i$ are computed as:

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

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

As introduced in Sec.3.3, we use the LBD1 descriptor vector $V$ to represent the local appearance of a line. We now get two sets of pairwise geometric attributes and local appearances for two candidate matches $(L^i_r, L^i_q)$ and $(L^j_r, L^j_q)$

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 $M^{ij}$ is computed as:

$$M^{ij} = \begin{cases} 5 - d_I - d_P - d_\Theta - s_V^i - s_V^j, & if \quad \Gamma = true; \\ 0, & else. \end{cases} \tag{3}$$

where $d_I$, $d_P$ and $d_\Theta$ are the geometric similarities and $s_V^i$, $s_V^j$ are the local appearance similarities. They are defined as:

$$\begin{cases} d_I = min(\frac{|I_r^i - I_q^i|}{t_I}, \frac{|I_r^j - I_q^j|}{t_I}); d_P = min(\frac{|P_r^i - P_q^i|}{t_P}, \frac{|P_r^j - P_q^j|}{t_P}); d_\Theta = \frac{|\Theta_r^{ij} - \Theta_q^{ij}|}{t_\Theta}; \\ s_V^i = \frac{\|V_r^i - V_q^i\|}{t_s}; \qquad s_V^j = \frac{\|V_r^j - V_q^j\|}{t_s}; \qquad \Gamma \equiv \{d_I, d_P, d_\Theta, s_V^i, s_V^j\} \leq 1. \end{cases} \tag{4}$$

Where $\Gamma \leq 1$ means each element in $\Gamma$ is less than 1. Compared with [2], 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 bad 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 $t_s = 0.35$. The five terms in the consistent score equation (3) are given the same weights while [15] introduces a method to learn weights of these terms which will be part of our future considerations. For all the candidate matches, we compute the consistent score among them and obtain the adjacency matrix $M$. The diagonal elements of $M$ equal zero as suggested by [15] for better results and let $M^{ji} = M^{ij}$ to keep the symmetry.

### 4.2 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 $\sum_{(L_r^i, L_q^i),(L_r^j, L_q^j) \in \mathcal{LM}} M^{ij}$ 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 [5]:

$$x^* = argmax(x^T M x) \tag{5}$$

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 [5], the $x^*$ that will maximize $x^T M x$ is the principal eigenvector of $M$. 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 [12,9] and the one-to-one constraint. Details of the algorithm are as follows:

1. Extract LineVecs from the reference and query images by EDLine [13] in the scale space to obtain two sets of LineVecs $\mathcal{L}_r$ and $\mathcal{L}_q$;
2. Estimate the global rotation angle $\theta_g$ of the image pair from the direction histograms of $\mathcal{L}_r$ and $\mathcal{L}_q$;
3. Compute the LBD1 descriptors [14] of LineVecs in $\mathcal{L}_r$ and $\mathcal{L}_q$;
4. Generate a set of candidate matches $\mathcal{CM} = \{(L_r^1, L_q^1), (L_r^2, L_q^2), \ldots, (L_r^n, L_q^n)\}$ 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 $M$ with a size of $n \times n$ according to the consistence scores of pairs in $\mathcal{CM}$;
6. Get the principal eigenvector $x^*$ of $M$ by using ARPACK[16];
7. Initialize the matching result: $\mathcal{LM} \leftarrow \oslash$;
8. Find $a = \text{argmax}(x^*(i)),\ i = 1, \cdots, n$. 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.

## 5 Experiments

To evaluate the performance of the proposed line matching algorithm, we conduct a set of experiments for image pairs under various transformations. We also compare our results to state-of-the-art methods. The following experiments are performed on a 3.4GHz Intel(R) Core 2 processor with 8 GB of RAM.

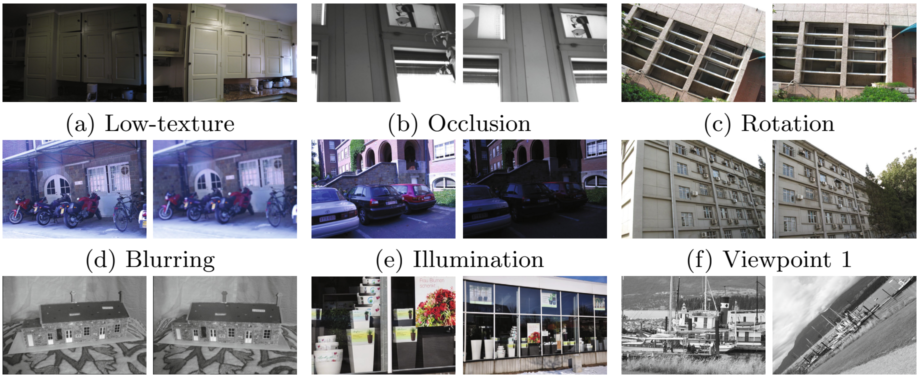
(a) Low-texture (b) Occlusion (c) Rotation
(d) Blurring (e) Illumination (f) Viewpoint 1
(g) Viewpoint 2 (h) Scale 1 (i) Scale 2

**Fig. 3.** Dataset for experiments

• **Line Matching Results:** We present the results of our line matching algorithm on nine image pairs as shown in Fig.3. The occlusion images are captured

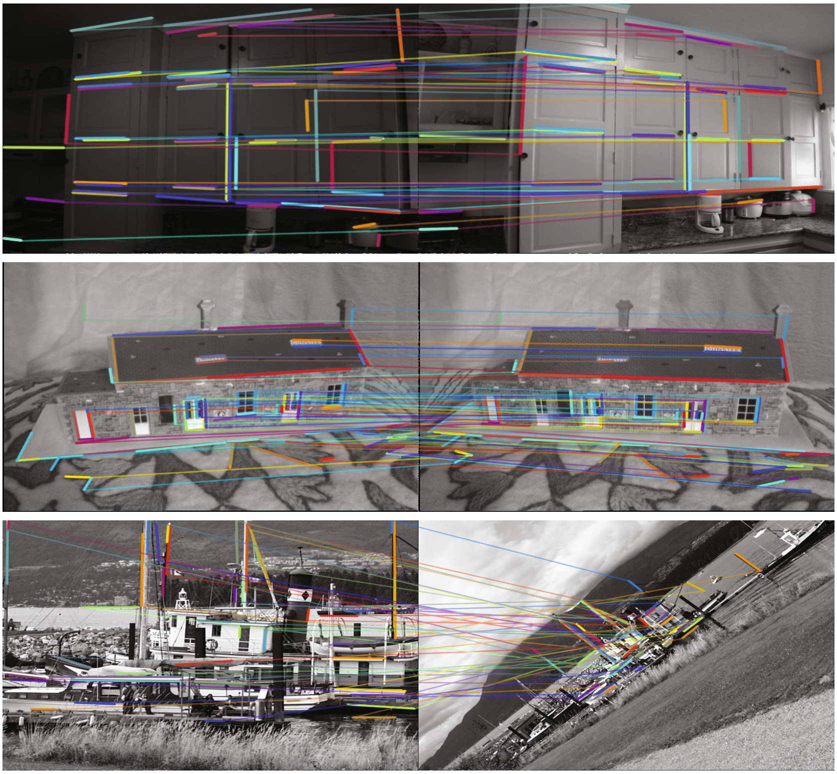

**Fig. 4.** Illustration of the line matching results

in our office and the others are from some publicly available dataset[2,3]. The matching results in three challenging scenes are shown in Fig.4 (The rest of matching results are presented in the supplementary materials because of the space limitation). 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.4 is low-texture planar scene with illumination and view point changes. The second image pair in Fig.4 is non-planar scene with moderate view point changes. The last image pair in Fig.4 is 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.4 are still quite acceptable and establish many matched lines with few mismatches.

It's worth to note that similar to the parameter detection methods adopted in [2] and [3], it's quite empirical to find the good parameter settings. However, these parameter settings are fixed as presented in Sec.3 and Sec.4 for all the experiments. The results show that the algorithm works well for a large range of image variations as illustrated in Fig.3.

**Table 1.** Comparison of our approach (AG) with three other line matching algorithms (MSLD [8], LP [3], LS [2]). For each image pair, the following results are reported: the number of total matches, the matching precision and the computational time.

| | Img | AG | MSLD | LP | LS | | Img | AG | MSLD | LP | LS | | Img | AG | MSLD | LP | LS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Total Matches | a | 54 | 20 | 5 | 54 | Matching Precision (%) | a | 94 | 60 | 60 | 96 | Time(s) | a | 0.11 | 0.30 | 5 | 8 |
| | b | 54 | 42 | 37 | 76 | | b | 100 | 95 | 100 | 100 | | b | 0.04 | 0.24 | 6.5 | 1 |
| | c | 263 | 238 | 230 | 188 | | c | 100 | 92 | 100 | 100 | | c | 0.38 | 0.42 | 14 | 26 |
| | d | 106 | 96 | 58 | 43 | | d | 100 | 92 | 100 | 100 | | d | 0.55 | 0.54 | 35 | 5 |
| | e | 245 | 245 | 211 | 241 | | e | 100 | 98 | 100 | 100 | | e | 0.59 | 0.50 | 25 | 8 |
| | f | 446 | 568 | 364 | 281 | | f | 100 | 96 | 100 | 100 | | f | 1.75 | 0.61 | 29 | 10 |
| | g | 87 | 82 | 80 | 151 | | g | 100 | 73 | 86 | 98 | | g | 0.20 | 0.24 | 20 | 8 |
| | h | 19 | 18 | 30 | 37 | | h | 95 | 67 | 95 | 97 | | h | 0.17 | 0.32 | 13 | 9 |
| | i | 44 | 30 | 48 | 14 | | i | 95 | 33 | 88 | 29 | | i | 0.51 | 0.42 | 72 | 8 |

• **Comparison with State-of-the-art Methods:** We compare our matching results with some state-of-the-art methods to further show its performance. As introduced in Sec.2, the existing approaches to match lines are mainly of three types. We choose three representatives from the three groups which are recently reported to feature highly remarkable performance: the Mean-Standard deviation Line Descriptor (MSLD) [8], the Line Signature (LS) [2] and the Line matching leveraged by Point correspondences (LP) [3]. As our method combines the Appearance and Geometric constraints together, it's called AG here. The implementations of LS and LP are supplied by their authors while MSLD is implemented by ourselves. In our implementation of MSLD, the parameters are chosen as recommended by its authors.

The test dataset used in the comparison is the same as in Fig.3. The comparison results are given in Tab.1. All the matched lines are checked one by one manually to test whether a matched line pair is correct or not. The input lines for different matching methods are not exactly the same except for AG and MSLD, because the binary codes of these algorithms have their default line detection methods embedded. However, the results shown in Tab.1 represent their performance for the nine image pairs. It's clear that MSLD and LP are less performing for the low-texture scene, because the local appearances of lines are indistinguishable and the images lack of stable point correspondences. The results also show that AG achieves higher accuracy than MSLD which proves that the graph matching process using spectral technique is necessary especially for these challenging situations. Surprisingly, LS has a bad matching result for image pair (i) which is inconsistent with the result presented in [2]. 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 depending on the order of images in a pair. Compared to LP and LS, the most superior feature of AG is its time performance. Here, the time of LP given in Tab.1 is its complete processing time which includes generating point correspondences and matching lines.

## 6 Conclusion

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 viewpoint 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 geometric constraints are enforced globally in this paper by using the spectral technique. For images undergoing a moderate 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's better to enforce the local geometric constraints like the approach in [2] although it is more time consuming.

## References

1. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI 27, 1615–1630 (2005)
2. Wang, L., Neumann, U., You, S.: Wide-baseline image matching using line signatures. In: ICCV, pp. 1311–1318 (2009)
3. Fan, B., Wu, F., Hu, Z.: Line matching leveraged by point correspondences. In: CVPR, pp. 390–397 (2010)
4. Schmid, C., Zisserman, A.: Automatic line matching across views. In: CVPR, pp. 666–671 (1997)
5. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV, pp. 1482–1489 (2005)
6. Neubert, P., Protzel, P., Vidal-Calleja, T., Lacroix, S.: A fast visual line segment tracker. In: ETFA, pp. 353–360 (2008)
7. Woo, D.M., Park, D.C., Han, S.S., Beack, S.: 2d line matching using geometric and intensity data. In: AICI, pp. 99–103 (2009)
8. Wang, Z., Wu, F., Hu, Z.: Msld: A robust descriptor for line matching. PR 42, 941–953 (2009)
9. Horaud, R., Skordas, T.: Stereo correspondence through feature grouping and maximal cliques. PAMI 11, 1168–1180 (1989)
10. Christmas, W., Kittler, J., Petrou, M.: Structural matching in computer vision using probabilistic relaxation. PAMI 17, 749–764 (1995)
11. Wilson, R.C., Hancock, E.R.: Structural matching by discrete relaxation. PAMI 19, 634–648 (1997)
12. Bay, H., Ferrari, V., Van Gool, L.: Wide-baseline stereo matching with line segments. In: CVPR, pp. 329–336 (2005)
13. Akinlar, C., Topal, C.: Edlines: A real-time line segment detector with a false detection control. Pattern Recognition Letters 32, 1633–1642 (2011)
14. Zhang, L., Koch, R.: Lbd: A fast and robust line descriptor based on line band representation. Technical Report (2012), http://www.mip.informatik.uni-kiel.de
15. Leordeanu, M., Sukthankar, R., Hebert, M.: Unsupervised learning for graph matching. IJCV, 1–18 (2011)
16. http://www.caam.rice.edu/software/ARPACK/