

Hierarchical Line Matching Based on Line-Junction-Line Structure Descriptor and Local Homography Estimation

Kai Li^a, Jian Yao^{a,*}, Xiaohu Lu^a, Li Li^a, Zhichao Zhang^{a,b}

^a*School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, P.R. China*

^b*Collaborative Innovation Center of Geospatial Technology, Wuhan, Hubei, P.R. China*

Abstract

This paper presents a hierarchical method for matching line segments from two images. Line segments are matched first in groups and then in individuals. While matched in groups, adjacent line segments are first intersected to form junctions and then used to construct structures called Line-Junction-Line (LJL), consisting of two adjacent lines segments and their intersecting junction. To reliably deal with the possible scale change between two images, we proposed to describe LJLs from multi-scale pyramids of images constructed from two original images with a robust descriptor. By evaluating their descriptor vector distances, some candidate LJL matches are obtained, which are then refined and expanded by an effective match-propagation strategy. Line segments, which cannot be matched in the form of LJL, are further matched in individuals by utilizing the local homographies estimated from their neighboring corresponding LJLs. Experimental results on a set of representative image pairs captured from different scenes demonstrate the robustness of the proposed method both under various image transformations and in poorly-textured scenes. Compared with the state-of-the-art line matching methods, the proposed method has the superiority for quite more correct matches and higher accuracy in most cases.

Keywords: Line matching, Hierarchical matching, Local feature descriptor, Homography estimation

*Corresponding author.

Email address: jian.yao@whu.edu.cn (Jian Yao)

URL: <http://cvrs.whu.edu.cn/> (Jian Yao)

1. Introduction

As a low-level vision task, image matching is fundamental for many applications which require recovering the 3D scene structure from 2D images such as robotic navigation [1, 2], structure from motion [3, 4, 5], 3D reconstruction [6, 7], scene interpretation [8], etc. The majority of image matching methods are point-based [9, 10, 11, 12, 13, 14] which commerce the extraction of feature points from images, followed by the utilization of the photometric information adjacently associated with the extracted points to match them. Objects in real scenes, however, can be easily outlined by line segments, which indicates recovering 3D scene structure from line matches seems better than that from point matches. In some cases, for example, the scenes are poorly-textured, where feature points are hard to be detected and matched, recovering their 3D structures from line matches seems the only choice because their structures can be easily outlined by several edge line segments. Despite the advantages, both the lack of point-to-point correspondence and the lost of connectivity and completeness of the extracted line segments make line segment matching a tough task, which also explains why line segment matching has less been investigated to some extent.

1.1. Related Works

Line matching methods in existing literatures can generally be classified into two categories: methods that match line segments in individuals and those in groups. Some methods matching line segments in individuals exploit the photometric information associated with individual line segments, such as intensity [15, 16], gradient [17, 18, 19], and color [20] in the local regions around line segments. All these methods underlie the assumption that there are considerable overlaps between corresponding line segments, which leads to the failure of these methods when corresponding line segments share insufficient corresponding parts. Besides, in regions with repetitive textures, these methods tend to produce false matches since the lack of variations in the photometric information for some line segments. Other methods matching line segments in individuals leverage point matches for line matching [21, 22, 23, 24]. These methods first find a large group of point matches using the existing point matching methods, and then exploit invariants between coplanar points and line(s) under certain image transformations to evaluate the correspondence of line segments from two images. Line segments

which meet the invariants are regarded to be in correspondence. These methods utilize geometric relationship between line segments and points, rather than photometric information in the local regions around line segments, and thus are robust even when local shape distortions are severe and there are scale changes. These two factors are acknowledged to contribute to the majority of the most challenging image matching tasks. However, these methods share a common disadvantage that they are incapable of processing images where the scenes captured are poorly-textured due to that feature points are hard to be detected and matched in this kind of scenes where there are often not sufficient point matches that can be exploited for matching line segments.

Matching line segments in groups is more complex, but more constraints are available for disambiguation. Most of these methods [25, 26, 27, 28, 29] first use some strategies to intersect line segments to form junctions and then utilize features associated with those junctions for line segment matching. These methods transfer matching line segments to matching points which has been widely investigated and there exist many applicable methods. But junction points contain more information than feature points detected by some feature point detectors [30, 31, 32, 33]. They are the results of intersecting two lines segments and the relationship between junction points and line segments forming them is additional and important information that can be exploited for matching them. How to effectively exploit features associated with the junction points to promote the robustness of these methods is still an open issue. In [34, 35], rather than exploiting features of junctions for line segment matching, the stability of the relative positions of the endpoints of a group of line segments in a local region under various image transformations is exploited. This method first divides line segments into groups and then generates a descriptor of the configuration of the line segments in each group by calculating the relative positions of these line segments. Since the configuration of a group of line segments in a local region is somewhat stable under most image transformations, the description vectors of the configurations of two groups of line segments in correspondence should be similar, and thus groups of line segments can be matched. This method is robust in many challenging situations, but the dependence on the approximately corresponding relationship between the endpoints of corresponding line segments leads to the tendency of this method to produce false matches when substantial disparity exists in the locations of the endpoints.

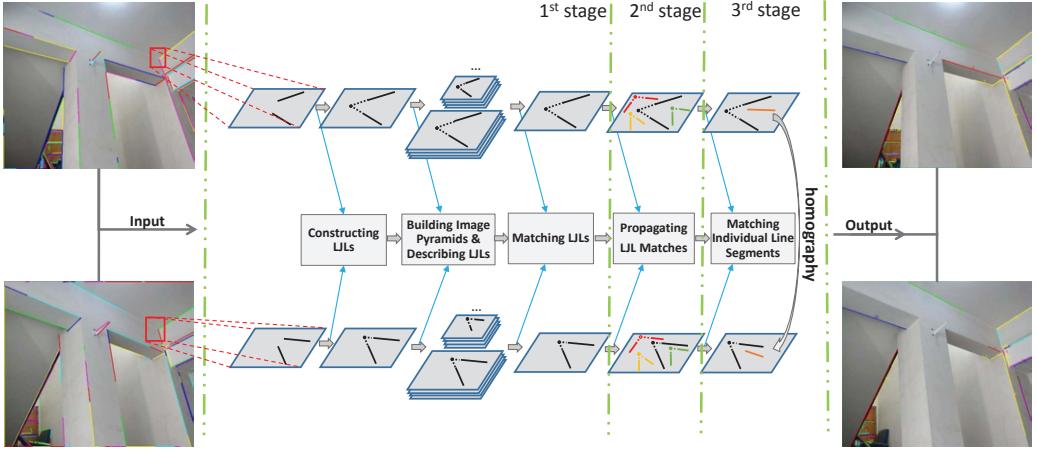


Figure 1: The flowchart of the proposed line matching algorithm.

1.2. Our Method

Our proposed line matching method in this paper is a combination of the two categories of above-described line matching methods, which matches line segments both in groups and in individuals under a hierarchical framework. The framework is comprised of three stages where line segments are matched in groups in the first two stages while in individuals in the third stage. The three-stage flowchart of the proposed line matching algorithm is illustrated in Figure 1. The first stage commences intersecting neighboring line segments to form junctions and then using them to construct the structures called Line-Junction-Line (LJL), consisting of two adjacent line segments and their intersecting junction. To greatly reduce the effect of the scale change between two images to be matched, we proposed to build Gaussian image pyramids for both images and the LJLs constructed in the original two images are adjusted to fit each image in the image pyramids and are described there by the proposed LJL descriptor. Some initial LJL matches can be found by matching these LJLs according to their corresponding description vector distances. These LJL matches are then refined and expanded in the second stage, where we propagate LJL matches by iteratively adding new matches while deleting possibly false ones. Line segments fail to be matched in the form of LJL are further matched in individuals in the third stage, where line segments are matched in individuals by utilizing the local homographies estimated from their neighboring corresponding LJLs.

This work is an extension of our work presented in [36]. Compared with

the previous one, this work makes promotions in the following aspects. First, a more general way is utilized to generate junctions using adjacent line segments. In [36], line segments extracted by some line segment detectors are refined by a series of procedures to obtain some line segments that are extended to be longer and are connected with others. In this work, line segments extracted by line detectors are not required to be refined in advance but only are used to generate junctions based on the local spatial proximity. What used as the input for the matching process are the original detected line segments and the junctions generated by them. This promotion helps to generate more junctions using the popular line segment extraction methods and benefits the better matching results. Second, a more robust descriptor is proposed to describe the structure (called Ray-Point-Ray (RPR) in [36], while Line-Junction-Line (LJL) in this paper) formed by two adjacent line segments and their intersecting junction. Third, a more reasonable strategy to deal with the possible scale change between images. To match line segments from two images with scale change, in [36], the global scale change factor between two original images is estimated and then one of the two images is adjusted to the same scale as that of another. This strategy is reasonable only when the scale change between two images is a global one. When there are scale changes in local regions, often introduced by viewpoint change, this strategy is unable to reliably work. This disadvantage is solved in this paper and the proposed strategy can deal with both global scale changes and local ones. Fourth, a more sophisticated way to match individual line segments. For line segments which cannot be matched in groups (RPR in [36] and LJL in this paper), in [36], they are used to intersect with those matched line segments to construct new RPRs and are matched in the form of RPR again, while in this paper, they are matched by the local homographies estimated from their neighboring corresponding LJLs.

All above promotions together contribute to the better performance of this work than the previous one. The advantages of this work over the previous one and other state-of-the-art line matching methods include its robustness for most kinds of image transformations, its fairly good performance in poorly-textured scenes and the larger amount of correct line matches with higher accuracy in the majority of cases.

The remainder of the paper is organized as follows. Section 2 introduces the construction, description and matching of LJLs from two images in the first stage. The adopted match propagation strategy in the second stage is described in Section 3. The step of matching individual line segments failed to

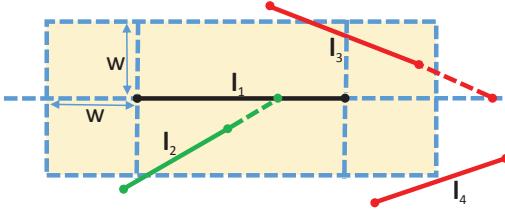


Figure 2: An illustration of finding neighbor line segments possibly coplanar with l_1 . The rectangle filled with yellow is the affect region of l_1 . l_2 , l_3 and l_4 are three neighbor line segments of l_1 . Only l_2 is accepted to be coplanar with l_1 because only it meets the two conditions (see the text) we use to determine coplanar line segments.

be matched in groups in the first and second stages is described in Section 4. Experimental results are presented in Section 5 and some discussion about the algorithm is given in Section 6. The conclusions are finally drawn in Section 7.

2. Initial LJL Matches Generation

The endpoints of line segments are unstable for line segment matching since there often does not exist accurate point-to-point correspondence for the endpoints of corresponding line segments. The intersecting junctions of line segments coplanar in 3D space are however invariant under projective transformation and thus are reliable to be exploited for matching line segments. If the two intersecting junctions of two pairs of line segments are successfully matched, it is then easy to determine the corresponding relationship between the two pairs of line segments. Therefore, what need to do first is to find some line segments coplanar in 3D space to generate junctions.

2.1. LJL Construction

It is hardly possible to determine which line segments are coplanar in 3D space only from a 2D image without the projective information of the camera. But adjacent line segments possess a higher probability to be coplanar in 3D space due to the spatial proximity. So, it is an alternative choice to intersect neighboring line segments to obtain reliable junctions. We use a similar method as that presented in [29] to generate junctions. Refer to Figure 2, for a line segments l_1 , we define its affect region as a rectangle, filled in yellow

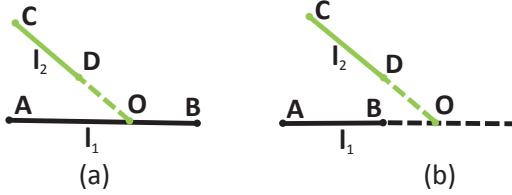


Figure 3: Two configurations of a pair of line segments intersecting with each other.

in the figure, which centers at the midpoint of \mathbf{l}_1 and has the length $|\mathbf{l}_1|+2w$ and the width of $2w$, where $|\mathbf{l}_1|$ denotes the length of \mathbf{l}_1 and w is a user-defined parameter. Any line segment satisfying the following two conditions is assumed to be coplanar with \mathbf{l}_1 in 3D space. First, at least one of the two endpoints drops in the affect region of \mathbf{l}_1 . Second, its intersection with \mathbf{l}_1 also drops in the affect region of \mathbf{l}_1 . Under these two conditions, in Figure 2, only \mathbf{l}_2 is accepted to be coplanar with \mathbf{l}_1 . \mathbf{l}_3 is rejected because its intersection with \mathbf{l}_1 is not within the affect region of \mathbf{l}_1 despite one of its endpoint drops inside it. \mathbf{l}_4 is rejected for neither of its two endpoints drops in the affect region of \mathbf{l}_1 .

The configurations of two line segments assumed to be coplanar in 3D space may exist the two forms shown in Figure 3. In Figure 3(a), the intersection lies on one of the two line segments (not on their extensions). In this case, two LJLs, $(\mathbf{OA}, \mathbf{O}, \mathbf{OC})$ and $(\mathbf{OB}, \mathbf{O}, \mathbf{OC})$, can be constructed. In Figure 3(b), the intersection lies on neither of the two line segments. Only one LJL, $(\mathbf{OA}, \mathbf{O}, \mathbf{OC})$, can be constructed in this case.

2.2. LJL Description

The relationship between the junction and the two line segments in a LJL is invariant under image transformations, which is exploited by our method to generate our proposed LJL descriptor. Inspired by SIFT [9], we construct gradient orientation histograms in the local region around the junction in a LJL to generate the LJL descriptor. Refer to Figure 4, the local region covered by two concentric circles centered at the junction is exploited. The radius r of the smaller circle is half as that of the bigger one. The two circles are divided by the two line segments in the LJL and their extensions into 4 parts. Each part contains a sector and a ring-shaped region, which is then evenly divided into 3 subregions, resulting in the 3 subregions have the same area with the sector. Therefore, there are totally 16 regions, two groups of 8

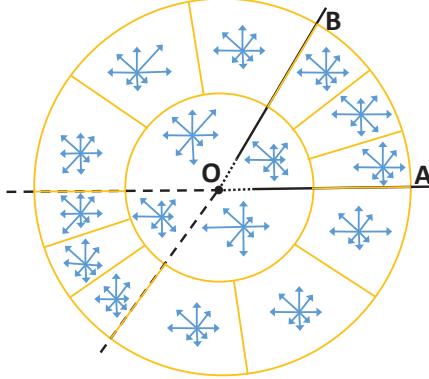


Figure 4: An illustration of describing a LJL (**OA**, **O**, **OB**) with the proposed LJL descriptor.

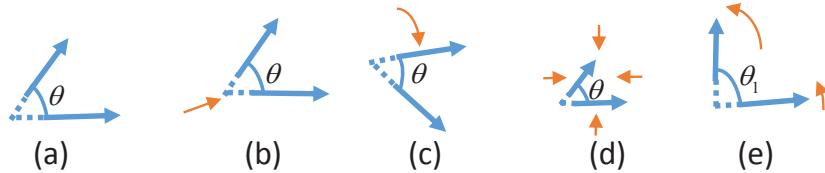


Figure 5: The changes of the crossing angle of the two line segments in a LJL with different transformations: (a) the original LJL; (b)-(e) the transformed LJLs with translation (b), rotation (c), scale (d) and affine changes (e), respectively.

regions with the same areas, for constructing gradient orientation histograms with 8 bins, producing a vector of 128 dimensions as the descriptor of a LJL. The strategy of assigning a weight to the gradient magnitude of a sample point and the way of eliminating boundary effects are the same as those of SIFT. A normalization on the descriptor vector is necessary to reduce the effect of illumination change. But since the LJL descriptor vector is from concatenating two groups of histograms constructed in regions with different areas and the numbers of sample points contributing to the histograms are different, the normalization should be conducted separately among each group of 8 histograms constructed in regions with the same area. After that, same as SIFT, a truncation of large gradient magnitudes at a certain value, v ($v = 0.3$ in this paper), is applied to give greater emphasis on the distribution of the orientations.

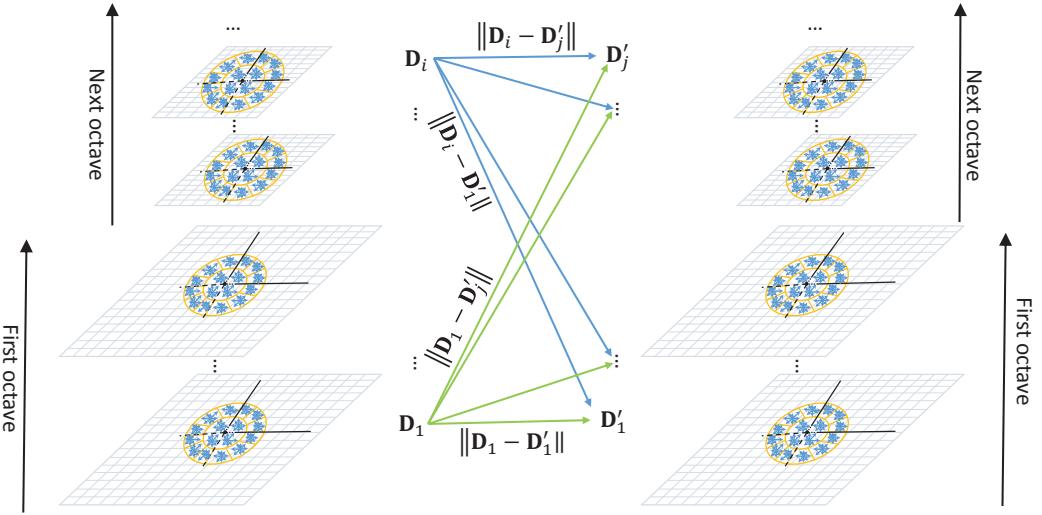


Figure 6: An illustration of matching a pair of LJLs from two images through building Gaussian image pyramids to deal with the possible scale change between two images.

2.3. LJL Matching

To match LJLs from two images, the general way is to evaluate their descriptor vector distances. But since each LJL consists of two line segments and their junction, there is additional information available for disambiguation. The two line segments in a LJL locate in a local region, indicating the crossing angle of the two line segment should vary at a small range under most image transformations. As illustrated in Figure 5, the crossing angle remains invariant with translation, rotation, scale transformations, and changes slightly with moderate affine change. For a test pair of LJLs, if θ_1 and θ_2 denote the crossing angle of the two line segments of the two LJLs, respectively, then, the absolute difference of θ_1 and θ_2 , $\theta = |\theta_1 - \theta_2|$ should be a small value ($\theta = 30^\circ$ in this paper) if the LJL pair is a correct match. This constraint can help to discard many false candidates before evaluating their descriptors and thus contributes to better matching results.

If a pair of LJLs satisfy the above constraint, we then evaluate their descriptors. The proposed LJL descriptor is based on the fixed-size window, which implies its inability of processing images with scale changes. The following strategy is proposed to solve this problem. Refer to Figure 6, we build Gaussian image pyramids for both the reference and query images

and adjust, if image sizes change, LJLs constructed in original images to fit each image in the pyramids and describe them in that image. Therefore, the descriptor of a LJL formed in the original images comprises of a set of sub-descriptors computed in different levels of the image pyramids, which leads to the descriptors of a pair of LJLs should be evaluated in a special manner. Suppose \mathcal{L} and \mathcal{L}' are two LJLs to be evaluated and $\mathcal{D} = \{\mathbf{D}_i\}_{i=0}^{L-1}$ and $\mathcal{D}' = \{\mathbf{D}'_j\}_{j=0}^{L-1}$ are their descriptors, where L denotes the number of levels of the pyramids which is the product of the number of the octaves in a pyramid, o , and the number of scales in each octave, s . In this paper, we empirically set o as 4 and s as 2, and thus L equals 8. \mathbf{D}_i and \mathbf{D}'_j denote the sub-descriptors of \mathcal{L} at the i -th level and the sub-descriptors of \mathcal{L}' at the j -th level, respectively. The distances between each sub-descriptor \mathbf{D}'_j in \mathcal{D}' and all sub-descriptors $\{\mathbf{D}'_j\}_{j=0}^{L-1}$ in \mathcal{D} are calculated, and then the average of the k (set as 2 in this paper) smallest distance values among $\{\|\mathcal{D}_i - \mathcal{D}'_j\|\}_{i=j=0}^{L-1}$ is regarded as the distance of \mathcal{D} and \mathcal{D}' . Those LJL pairs whose descriptor vector distances are smaller than a given threshold d_v ($d_v = 0.5$ in this paper) are regarded as candidate LJL matches.

This above strategy is reasonable because if \mathcal{L} and \mathcal{L}' are two LJLs in correspondence, since their descriptors are comprised of a set of sub-descriptors calculated in different scales, there always exist at least one pair of sub-descriptors which are computed in the (almost) same scale if the levels of image pyramids are sufficient, and, theoretically, their descriptor vector distance is smaller than other pairs where the two sub-descriptors are calculated in different image scales. In case that a pair of sub-descriptors which are calculated in different scales but accidentally with the smallest descriptor distance, which is often the case when the number of candidates is great and the dimension of the descriptors is high, it is better to use the average of the k smallest values among the descriptor distances $\{\|\mathcal{D}_i - \mathcal{D}'_j\|\}_{i=j=0}^{L-1}$ of all sub-descriptors as the descriptor vector distance of the candidate LJL pair.

3. LJL Match Propagation

Point matches, the junction pairs of LJL matches, can be used to recover the fundamental matrix by using RANSAC [37]. After that, we obtain the fundamental matrix as well as a group of LJL matches consistent with it, based on which we commence propagating LJL matches among the unmatched LJLs. The LJL match propagation is achieved by progressively

increasing the threshold for the point-to-epipolar-line distance of an accepted point match according to the fundamental matrix. LJL pairs with smaller distances of their junctions according to the fundamental matrix are matched first and then serve as the basis for next iteration to introduce new LJL matches. The topological constraint on corresponding LJLs and their neighboring corresponding points is exploited to filter false matches while guide the process of adding new matches.

A correct LJL match would be consistent with their neighboring point matches in the topological distribution. Refer to Figure 7, $(\mathbf{OA}, \mathbf{O}, \mathbf{OB})$ and $(\mathbf{O}'\mathbf{A}', \mathbf{O}', \mathbf{O}'\mathbf{B}')$, referred as \mathcal{L} and \mathcal{L}' , are a pair of LJLs in correspondence from two images. The junction and the two line segments as well as their extensions in each matched LJL form a coordinate-like structure. Neighboring matched points, the junctions of matched LJLs, distribute in different quadrants of the coordinates. We collect the n (n equals the smaller one between 10 and the total number of matched points in this paper) nearest matched points as $\tilde{\mathcal{M}} = \{\mathbf{m}_i\}_{i=1}^n$ and $\tilde{\mathcal{M}}' = \{\mathbf{m}'_j\}_{j=1}^n$ to the junctions \mathbf{O} and \mathbf{O}' of \mathcal{L} and \mathcal{L}' , respectively. If $(\mathcal{L}, \mathcal{L}')$ is a correct match, the following two conditions must be satisfied. The first one is that there should exist a sufficient large proportion, p_1 ($p_1 = 0.5$ in this paper), of correspondences in $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{M}}'$. In addition, if $\mathbf{m}_i \in \tilde{\mathcal{M}}$ and $\mathbf{m}'_j \in \tilde{\mathcal{M}}'$ are the correct correspondences, they should be in the same quadrants of the two coordinates formed by \mathcal{L} and \mathcal{L}' in a high probability. So, the second condition is that among the correspondences in $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{M}}'$, those with the same quadrants should account for a big proportion, p_2 ($p_2 = 0.8$ in this paper).

After the first stage of our method, we obtain the set of LJL matches, \mathcal{M}_L , and the two sets of unmatched LJLs, \mathcal{U}_L and \mathcal{U}'_L , from the two images, respectively. In this match propagation stage, we refine the set \mathcal{M}_L by adding new LJL matches from \mathcal{U}_L and \mathcal{U}'_L , and eliminating the possible false ones in an iterative scheme. The pseudo-code of the proposed LJL match propagation strategy is presented in Algorithm 1. While adding new LJL matches, each LJL in \mathcal{U}_L is evaluated to all LJLs in \mathcal{U}'_L . For a test pair of LJLs, it will be checked by the following three constraints in order. First, the point-to-epipolar-line distances of the pair of junctions in the two LJLs should be less than some threshold d_e . The value of d_e was set to 1 in the initial iteration and was increased by adding 1 in each subsequent iteration. Most false test pairs can be filtered out based on this constraint. Second, their description vector distance is less than a given threshold d_v ($d_v = 0.5$ in this paper). Third, the two LJLs should meet the topological constraint presented

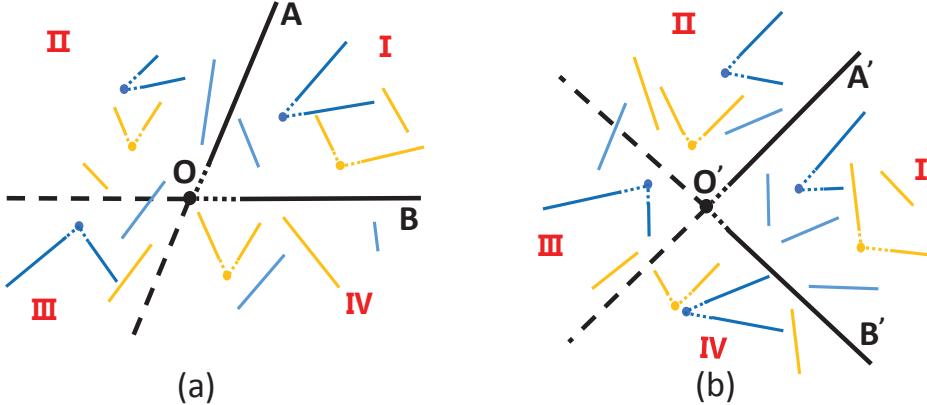


Figure 7: An illustration of the pair of LJLs, (OA, O, OB) and $(O'A', O', O'B')$, in correspondence and the distribution of their neighboring line segments and LJLs. Neighboring matched LJLs and individual line segments are marked in yellow while unmatched ones are marked in blue.

in the above paragraph. Some new LJL matches would be generated after the above steps and bring in new point matches. Under the new group of point matches, some LJL matches used to be consistent with their neighboring point matches may turn out to be inconsistent with them and therefore need be filtered out. The process of adding new LJL matches and filtering out possibly false ones are conducted iteratively until no new LJL matches are added or the iterative times is greater than 5.

4. Individual Line Segment Matching

Line segments that have not been matched along with LJLs in the above two stages are further matched in individuals in the last stage of our method, in which they are first grouped according to those matched LJLs, and then matched in each two corresponding groups based on the local homography recovered from the pair of matched LJLs in these two groups.

4.1. Individual Line Segment Grouping

Let $\mathcal{M}_L = \{(\mathcal{L}_v, \mathcal{L}'_v)\}_{v=1}^V$ be the set of V LJL matches identified before, where $(\mathcal{L}_v, \mathcal{L}'_v)$ denotes the v -th LJL match. Let $\mathcal{K} = \{l_i\}_{i=1}^M$ and $\mathcal{K}' = \{l'_j\}_{j=1}^N$ be the two groups of individual line segments, which have not been matched before, from two images, respectively. For each individual line segment $l_i \in$

Algorithm 1 LJL Match Propagation

Input: The two sets of unmatched LJLs from two images: \mathcal{U}_L and \mathcal{U}'_L , and the set of LJL matches, \mathcal{M}_L

Output: The updated \mathcal{U}_L , \mathcal{U}'_L and \mathcal{M}_L

```
1: Set #Iterations  $\leftarrow 0$ , #NewLJLMatches  $\leftarrow 1$ ,  $d_e \leftarrow 1$ , and  $d_v \leftarrow 0.5$ .
2: while #NewLJLMatches  $> 0$  and #Iterations  $< 5$  do
3:   Set #NewLJLMatches  $\leftarrow 0$ .
4:   for each  $\mathcal{L}_i \in \mathcal{U}_L$  do
5:     for each  $\mathcal{L}'_j \in \mathcal{U}'_L$  do
6:       Calculate the point-to-epipolar-line distances of the pair of
junctions in  $\mathcal{L}_i$  and  $\mathcal{L}'_j$ ,  $D_e(\mathcal{L}_i, \mathcal{L}'_j)$  and  $D_e(\mathcal{L}'_j, \mathcal{L}_i)$ .
7:       if  $D_e(\mathcal{L}_i, \mathcal{L}'_j) < d_e$  and  $D_e(\mathcal{L}'_j, \mathcal{L}_i) < d_e$  then
8:         Calculate the description vector distance of  $\mathcal{L}_i$  and  $\mathcal{L}'_j$ ,
 $D_v(\mathcal{L}_i, \mathcal{L}'_j)$ .
9:         if  $D_v(\mathcal{L}_i, \mathcal{L}'_j) < d_v$  then
10:           if  $(\mathcal{L}_i, \mathcal{L}'_j)$  satisfies the topological constraint then
11:             Remove  $\mathcal{L}_i$  and  $\mathcal{L}'_j$  from  $\mathcal{U}_L$  and  $\mathcal{U}'_L$ , respectively.
12:             Update  $\mathcal{M}_L \leftarrow \mathcal{M}_L \cup \{(\mathcal{L}_i, \mathcal{L}'_j)\}$ .
13:             Update #NewLJLMatches++.
14:           end if
15:         end if
16:       end if
17:     end for
18:   end for
19:   for each  $(\mathcal{L}_i, \mathcal{L}'_i) \in \mathcal{M}_L$  do
20:     if  $(\mathcal{L}_i, \mathcal{L}'_i)$  satisfies the topological constraint then
21:       Remove  $(\mathcal{L}_i, \mathcal{L}'_i)$  from  $\mathcal{M}_L$ .
22:       Update  $\mathcal{U}_L \leftarrow \mathcal{U}_L \cup \{\mathcal{L}_i\}$  and  $\mathcal{U}'_L \leftarrow \mathcal{U}'_L \cup \{\mathcal{L}'_i\}$ .
23:     end if
24:   end for
25:   Update #Iterations++ and  $d_e \leftarrow d_e + 1$ .
26: end while
```

\mathcal{K} or $\mathcal{L}'_j \in \mathcal{K}'$, we search u ($u = 3$ in this paper) of its nearest matched LJLs whose junctions close to the line segment. The line segment is then assigned into these corresponding u groups. In this way, each matched LJL collects zero to multiple individual line segment(s). Then, for each matched

LJL, we divide the individual line segments that it collects into four groups according to the positions of these line segments relative to the matched LJL. As illustrated in Figure 7, a LJL forms a coordinate-like structure that the line segments and the constructed LJLs are distributed into its four quadrants. For each individual line segment that a matched LJL collects, if any of its two endpoints lies in a certain quadrant of the coordinates formed by the matched LJL, the line segment is put into the corresponding group. In this way, the individual line segments in \mathcal{K} and \mathcal{K}' form two sets, $\mathcal{U} = \{\mathcal{I}_m^p | m = 1, 2, \dots, V; p = 1, 2, 3, 4\}$ and $\mathcal{U}' = \{\mathcal{I}'_n^q | n = 1, 2, \dots, V; q = 1, 2, 3, 4\}$, respectively, where \mathcal{I}_m^p and \mathcal{I}'_n^q are the individual line segment sets whose elements are collected from \mathcal{K} and \mathcal{K}' , respectively. Then the line segments from each two corresponding sets, \mathcal{I}_m^p and \mathcal{I}'_n^q when $m = n$ and $p = q$, are evaluated and matched separately. Under this grouping strategy, each line segment may be put into several groups. Despite, in most cases, this may lead to multiple evaluations of some pairs of line segments, it is still necessary to do so to ensure potential corresponding line segments would be assigned into at least one pair of groups in correspondence and be evaluated at least one time.

4.2. Local Homography Estimation

We have assumed the two line segments forming a LJL are coplanar in 3D space, and therefore a LJL match provides two coplanar line segment matches, which can be used to estimate a local homography with the combination of the estimated fundamental matrix.

A planar homography \mathbf{H} is determined by eight degrees of freedom, necessitating 8 independent constraints to find a unique solution. However, when the fundamental matrix \mathbf{F} between two images is known, then $\mathbf{H}^\top \mathbf{F}$ is skew-symmetric [38] as

$$\mathbf{H}^\top \mathbf{F} + \mathbf{F}^\top \mathbf{H} = 0. \quad (1)$$

The above equation gives five independent constraints on \mathbf{H} , and the other three are required to fully describe a homography. One line match provides two independent constraints [39], resulting in the system is over-constrained since two coplanar line matches exist in our case.

The homography induced by a 3D plane $\boldsymbol{\pi}$ can be represented as

$$\mathbf{H} = \mathbf{A} - \mathbf{e}' \mathbf{v}^\top, \quad (2)$$

where the 3D plane is represented by $\boldsymbol{\pi} = (\mathbf{v}^\top, 1)$ in the projective reconstruction with the camera matrices $\mathbf{C} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{C}' = [\mathbf{A}|\mathbf{e}']$. The homography

maps a point from one 2D plane to another 2D plane. For a line segment match $(\mathbf{l}, \mathbf{l}')$, suppose \mathbf{x} is an endpoint of \mathbf{l} , the homography maps it to its correspondence point \mathbf{x}' as

$$\mathbf{x}' = \mathbf{H}\mathbf{x}. \quad (3)$$

Since \mathbf{l} and \mathbf{l}' correspond with each other, \mathbf{x}' must be a point lying on \mathbf{l}' , which results in

$$\mathbf{l}'^\top \mathbf{x}' = 0. \quad (4)$$

Combining Eqs. (2)–(4), we obtain

$$\mathbf{l}'^\top (\mathbf{A} - \mathbf{e}' \mathbf{v}^\top) \mathbf{x} = 0. \quad (5)$$

Arranging the above equation, we finally get

$$\mathbf{x}^\top \mathbf{v} = \frac{\mathbf{x}^\top \mathbf{A}^\top \mathbf{l}'}{\mathbf{e}'^\top \mathbf{l}'}, \quad (6)$$

which is linear in \mathbf{v} . Each endpoint of a line segment in a line match provides a constraint equation, and two line segment matches totally provide four constraint equations. A least-square solution of \mathbf{v} can be obtained from the four equations. The local homography \mathbf{H} is then computed from Eq. (2).

4.3. Individual Line Segment Matching

After grouping, the individual line segments in one group from an image are only evaluated and matched with the individual line segments in the corresponding group from the other image, which decreases the candidate pairs that need to be evaluated and thus improve the efficiency of our method and also the accuracy of line matching results since less interferences are involved when finding the correspondence for each line segment. Suppose \mathbf{l} and \mathbf{l}' are a pair of individual line segments to be evaluated and they are collected by the pair of matched LJLs, \mathcal{L} and \mathcal{L}' , respectively. The LJL match, $(\mathcal{L}, \mathcal{L}')$, brings one point match, $(\mathbf{j}, \mathbf{j}')$, and two line segment matches, $(\mathbf{l}_m, \mathbf{l}'_m)$ and $(\mathbf{l}_n, \mathbf{l}'_n)$.

We first check whether the rotation angle of \mathbf{l} and \mathbf{l}' is consistent with the rotation angles of the two pairs of matched line segments brought by \mathcal{L} and \mathcal{L}' . Correctly matched line segments in local regions possess similar rotation angles under image transformations. Suppose the rotation angle of \mathbf{l}_m and \mathbf{l}'_m is σ_m , and that of \mathbf{l}_n and \mathbf{l}'_n is σ_n , If there exists

$$\left| \sigma - \frac{\sigma_m + \sigma_n}{2} \right| < \beta, \quad (7)$$

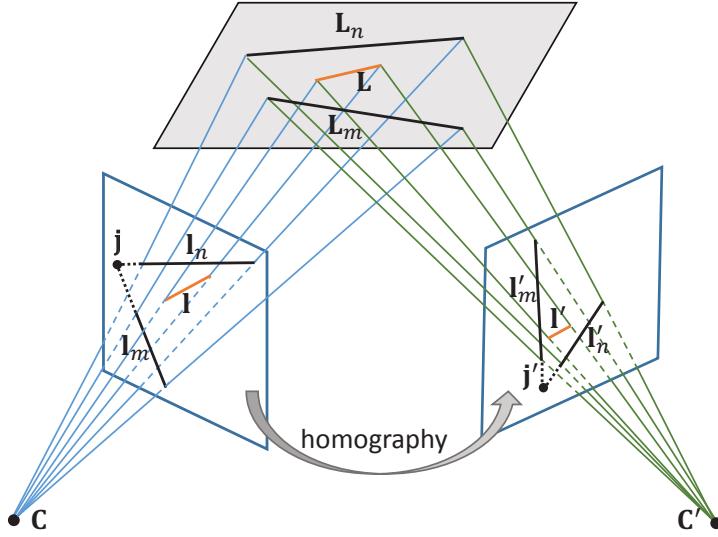


Figure 8: An illustration of the principle of matching the individual line segments by the estimated local homography. (l_m, j, l_n) and (l'_m, j, l'_n) are two matched LJJs from two images. l and l' are two line segment to be matched. C and C' are the camera centers for the two images, respectively. L_m , L_n and L are the 3D projections of the two groups of line segments from the two images. From (l_m, j, l'_n) and (l'_m, j, l'_n) , we can estimate a local homography, based on which, the test pair of line segments, l and l' are evaluated and matched.

where σ denotes the rotation angle of l and l' , and β is a user-defined angle threshold set as 20° in this paper, we accept (l, l') temporarily as a candidate match and take it for further evaluation.

We then evaluate the candidate match (l, l') by the local homography estimated from $(\mathcal{L}, \mathcal{L}')$. This method is reasonable only when it is the situation shown in Figure 8 where the 3D projections of the 6 line segments, l , l' , l_m , l'_m , l_n and l'_n are on the same 3D plane. It is hardly possible to determine whether the 3D projections of several 2D line segments are on same 3D planar without projective information of the cameras. But the strategies used in our algorithm ensure the rationality of this method. The first is the exploitation of the local spatial proximity. Line segments adjacent with each other in 2D image are highly possible to be coplanar in 3D space. Line segments in the two triples, (l_m, l_n, l) and (l'_m, l'_n, l') , are clustered based on the local spatial

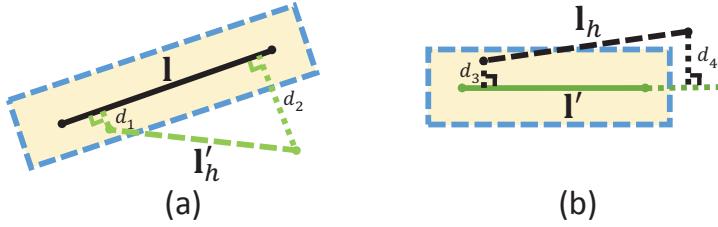


Figure 9: An illustration of calculating the mapping error of a pair of line segments, \mathbf{l} and \mathbf{l}' , by the estimated homography. \mathbf{l}_h and \mathbf{l}'_h are the correspondences of \mathbf{l} and \mathbf{l}' calculated by the estimated homography, respectively.

proximity, which guarantees a fairly high possibility that the 3D projections of line segments in the two triples are on the same plane. On the other hand, the success of matching the two LJLs, despite we cannot absolutely ensure the correctness of the matching, substantiates the 3D projections of \mathbf{l}_m , \mathbf{l}'_m , \mathbf{l}_n and \mathbf{l}'_n are on the same plane. The second is the redundant grouping strategy. Each individual line segment is redundantly collected by several neighboring matched LJLs, which greatly increases the possibility that two potential corresponding line segments are distributed into at least one pair of groups where they are coplanar with the two pairs of matched lines in 3D space.

If $(\mathbf{l}, \mathbf{l}')$ is a correct match, the correspondences of the two endpoints of \mathbf{l} , mapped by the estimated local homography, must be adjacent with (ideally on) \mathbf{l}' , and the same goes with the endpoints of \mathbf{l}' . We use the affect region of a line segment to apply this constraint. The affect region of a line segment is illustrated in Figure 2. It is a rectangle around the line segment with a parameter controlling the size of the rectangle. This parameter w (see Figure 2) was set as 3 in pixels when applied this constraint in this paper. Refer to Figure 9, We map \mathbf{l} and \mathbf{l}' to their correspondences by the estimated local homography, generating \mathbf{l}_h and \mathbf{l}'_h for \mathbf{l} and \mathbf{l}' , respectively. If both \mathbf{l}_h and \mathbf{l}'_h intersect with the affect regions of \mathbf{l}' and \mathbf{l} (the two rectangles filled in yellow around \mathbf{l}' and \mathbf{l}), the match $(\mathbf{l}, \mathbf{l}')$ is temporarily accepted and is taken for further evaluation. Here, a line segment intersects with a region means there exists at least one point (not just the two endpoints) on the line segment is within the region. we define the average of the four distances, including the perpendicular distances of two endpoints of \mathbf{l}'_h to \mathbf{l} and the perpendicular distances of the two endpoints of \mathbf{l}_h to \mathbf{l}' , as the *mapping error* of $(\mathbf{l}, \mathbf{l}')$, which is used to measure the similarity of \mathbf{l} and \mathbf{l}' . The four distances are



Figure 10: An illustration of using the brighter side constraint on matching line segments. \mathbf{l} is a line from a image. \mathbf{l}'_1 and \mathbf{l}'_2 are the two candidate correspondences for \mathbf{l} from another image. \mathbf{l}'_1 is accepted as the correspondence for \mathbf{l} because its brighter side is the same as that of \mathbf{l} .

denoted as d_1 , d_2 , d_3 and d_4 in Figure 9, then, the mapping error of $(\mathbf{l}, \mathbf{l}')$ is

$$E(\mathbf{l}, \mathbf{l}') = \frac{d_1 + d_2 + d_3 + d_4}{4}. \quad (8)$$

While finding the correspondence for a line segment, the above constraints may not be able to filter out those false ones if they have similar directions with the correct one and are adjacent with it. Refer to Figure 10, while we find the correspondence for \mathbf{l} , both \mathbf{l}'_1 and \mathbf{l}'_2 may be accepted by the above constraints because they are close with each other and have similar directions. We use a simple but effective way to enforce the constraints on correct matches by finding the brighter sides of line segments. The brighter side of a line segment is the side in which the average intensity value of pixels is higher than the other side. The brighter side of a line segment is located in this way. Refer to Figure 10, we calculate the average intensity values of pixels in the two rectangles lying in the two sides of a line segment and regard the side where the average intensity value is higher as the brighter side. The two rectangles along with the line segment have the same size with a length equaling the length of the line segment and a small width which equals 5 in pixels in this paper. Since the brighter side of a line segment indicates the relative brightness of the two small regions along with a line segment, it is invariant under almost all image transformations and thus can be exploited to find the correct correspondences for line segments. In Figure 10, we finally pick out \mathbf{l}'_1 as the correct correspondence of \mathbf{l} since it has the same brighter side as \mathbf{l} .

After that, there may exist the cases that one line segment in one image is matched with several line segments in another image. We select the pair with the minimal mapping error as the correct match and reject the others.

5. Experimental Results

Extensive experiments on representative image pairs were conducted to select proper values for some parameters of the proposed method ¹ and to evaluate its performance under various image transformations and in some special scenes, as well as to compare it with the state-of-the-art line matching methods.

5.1. Parameters Selection

Our algorithm has some parameters, but only two of them are key to influence the performance of the algorithm. Other parameters are used to strengthen some constraints and the fluctuations of their values make slight differences on the results. The first parameter is w which controls the size of the affect region of a line segment when constructing LJLs. The second parameter is r , the radius of the smaller of circle when describing LJLs. The five representative image pairs [43] shown in Figure 11 were employed to evaluate these two parameters. There are illumination change, image blur, JPEG compression, viewpoint change, scale and rotation changes between the two images in the five image pairs in order. Since the two images in each image pair are related by a known homography, we can evaluate the performance generated by different parameters conveniently and reliably, and thus help select proper values for the parameters.

We first conducted experiments for selecting a proper value of w . It is obvious that a big value of w results in a big affect region of a line segment and generates more intersections of line segments and thus more LJLs. Sufficient LJLs are the guarantee to produce enough initial LJL matches and therefore are crucial to the final line matching results because the subsequent steps of adding more line segment matches are based on the initial LJL matches. However, excessive LJLs, especially when many of them cannot find their correspondences in another group of LJLs, will harm the matching of them since more interferences are introduced, and both more computation time and memory are required to match them. We employed the way introduced in [43] by calculating the repeatability of two groups of points to select a proper value of w . In [43], repeatability was used to evaluate different local region detectors under various image transformations. It measures how the

¹The source codes, the executable program and test images are available in <http://cvrs.whu.edu.cn/projects/ljlLineMatcher/>.



Figure 11: Five image pairs used for selecting proper values of the key two parameters in our method and the L_{JL} descriptor evaluation. The five image pairs will be referred as “leuven”, “bikes”, “ubc”, “graffiti”, and “boat” from left to right in order.

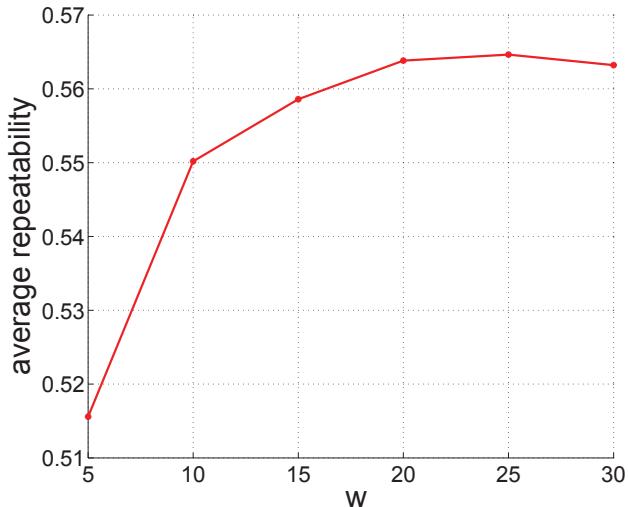


Figure 12: The changes of the average repeatability with different values of the parameter w .

number of correspondences depends on the transformation between two images. Higher repeatability indicates better performance of the image feature detection and is generally more conducive for matching these features. We calculated the repeatability of the formed junction points from images by varying value of w and fixed w at the value with the highest repeatability in all the following experiments. All the five image pairs shown in Figure 11 were employed and used to detect line segments and form junctions. We

varied w from 5 to 30 at the step of 5 and then calculated the average repeatability of the five pairs of point groups for each w . The change of the average repeatability along with w is shown in Figure 12, from which we observe that the repeatability curve increases when w is less than 20, and is stable until w is bigger than 25, where the curve begins to drop. Thus, both 20 and 25 are proper values for w . To obtain less junctions and reduce the computation time, $w = 20$ was selected in this paper.

Then, we conducted experiments to find a proper value of the parameter r , which determines the size of the local patch exploited for describing a LJL. The LJL descriptor is often more discriminative when a bigger patch is used but is often more sensitive to shape distortion. Table 1 shows accuracy of the matching results of the LJLs with various values of r on the five image pairs shown in Figure 11. Since the average accuracy reaches its maximum at $r = 10$, therefore $r = 10$ was used in all the following experiments. Note that the correctness of a LJL match is assessed in this way: let $(\mathbf{j}, \mathbf{j}')$ be the pair of junctions in a LJL match, we map \mathbf{j} and \mathbf{j}' by the known homography (\mathbf{j}' is mapped by the inverse of the known homography matrix), generating their estimated correspondences, \mathbf{j}_h for \mathbf{j} and \mathbf{j}'_h for \mathbf{j}' , respectively. If both the distance between \mathbf{j}'_h and \mathbf{j} and the distance between \mathbf{j}_h and \mathbf{j}' are less than 5 in pixels, we regard the LJL as a correct match. This correctness-assess strategy for point matches seems problematical since it may access a false point match as a correct one when one point in the match lies near the actual and exact correspondence of another point. However, since in our cases, the points are the intersecting junctions of line segments, whose distributions are often much sparser and whose numbers are often smaller than the detected feature points by some detectors. This fact ensures the reasonableness and reliability of the correctness-access strategy.

5.2. Evaluation of the LJL Descriptor

After the key parameter for constructing LJL descriptor being fixed, we conducted experiments to compare our LJL descriptor with other local region descriptor(s) for the effectiveness to describe the local regions around LJLs. The famous SIFT descriptor [9] was employed for the comparison. The two descriptors, LJL and SIFT, were used to describe the junctions of the constructed LJLs on the five image pairs shown in Figure 11, and then the junctions were matched under the same threshold according to their descriptor vector distances. Note that since SIFT is also based on fixed-size window and is unable to deal with scale change, describing the junctions in

	$r = 4$	$r = 6$	$r = 8$	$r = 10$	$r = 12$	$r = 14$
“leuven”	0.81	0.85	0.87	0.84	0.85	0.84
“bikes”	0.63	0.64	0.61	0.59	0.58	0.60
“ubc”	0.74	0.74	0.77	0.72	0.72	0.69
“graffiti”	0.30	0.45	0.64	0.76	0.69	0.65
“boat”	0.09	0.21	0.31	0.44	0.40	0.27
average	0.51	0.58	0.64	0.67	0.65	0.61

Table 1: The accuracies of the line matching results of the proposed method on the five image pairs under various values of the parameter r .

LJLs with SIFT is also applied in the Gaussian image pyramids. Table 2 shows the accuracy of the matching results of the two descriptors. It can be observed from this table that on all of the five image pairs, where various extreme image transformations exist, the proposed LJL descriptor produced matching results with higher accuracy and on some image pairs, “graffiti” and “boat”, this advantage is fairly remarkable: the results of LJL descriptor present the accuracy more than twice as that of SIFT on these two image pairs. This advantage of our LJL descriptor on the accuracy is crucial for our method because it requires estimating a precise fundamental matrix from the initial LJL matches, based on which, more LJL matches would be found subsequently. A large proportion of correct matches certainly contribute to better estimation result of the fundamental matrix. We did not use the well-known local descriptor evaluation method introduced in [42] because the proposed LJL descriptor is specially designed for LJLs. It describes the circular regions centered at the junctions of LJLs, rather than affine invariant regions detected by some detectors, which is the prerequisite of that famous local descriptor evaluation method.

The proposed LJL descriptor is specially designed for LJL structures. It exploits the gradients of pixels in the local patch covered by two concentric circles around the junction of a LJL while taking into consideration of the relationship between the junction point and the two line segments forming it. Its better performance over SIFT in describing the local regions around the junctions of LJLs owes to the following two factors. The first one is that the regions for constructing the orientation histograms for the LJL descriptor are more likely to correspond with each other for corresponding junctions

	“leuven”	“bikes”	“ubc”	“graffiti”	“boat”
LJL	0.84	0.59	0.72	0.76	0.44
SIFT	0.71	0.48	0.71	0.37	0.19

Table 2: The precisions of the matching results on the five image pairs generated by the proposed LJL descriptor and the SIFT descriptor by describing the junction points in LJLs.

than that of SIFT. We have clear and precise dominant directions to deal with possible rotation changes. Either of the directions of the two line segments forming the junction can be regarded as the dominant direction of the junction, according to which the regions for constructing the orientation histograms are rotated. While in SIFT, the dominant direction of a point is calculated from its neighboring region, which is absolutely less precise than ours. Besides, the configuration of the two line segments forming the junction is exploited for dividing the region around the junction into subregions, where the orientation histograms are constructed. Since the two line segments are clear and precise, subregions divided by them are more likely to correspond with each other for corresponding junctions than that of SIFT, in which subregions are obtained by dividing the region regularly with the same angle span (90°). The second one is that the exploitation of the constraint that the crossing angle of the two line segments in a LJL should vary in a small range under image transformations, which helps discard many false candidate LJL pairs before evaluating their descriptor vector distances and thus contributes to better matching results.

5.3. Line Matching Results

Figures 13–20 show the line matching results of the proposed method on 8 representative image pairs which contain various image transformations and were captured from planar scenes and non-planar scenes. All these image pairs are those used in the published papers [23, 43] except the image pair (d), in which a poorly-textured scene was captured in the two images. The aim for employing this image pair is to evaluate the performance of our method in poorly-textured scenes. These line matching results are based on line segments extracted by the famous line segment detector, LSD [40]. In all of these 8 figures, two line segments in correspondence are drawn in the same color and are labeled with the same number at the middles in two

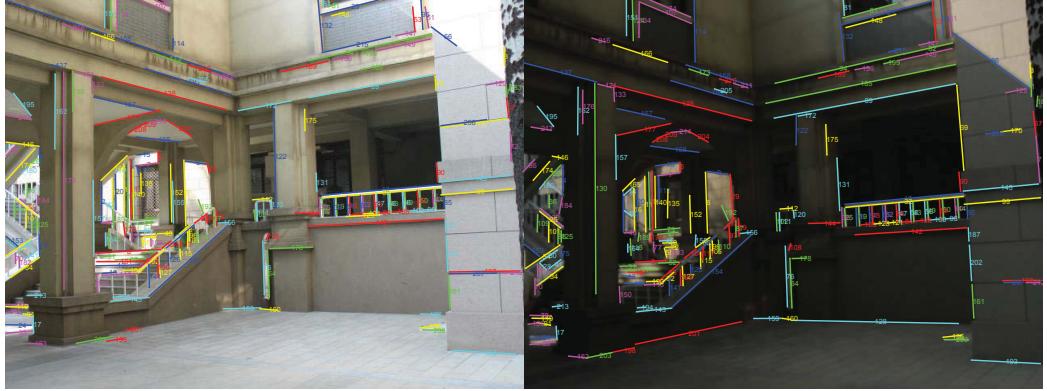


Figure 13: (a) illumination change. #DetectedLines: (572, 275), #TotalMatches: 217, #FalseMatches: 3, accuracy: 97.5%.



Figure 13: (b) rotation change. #DetectedLines: (537, 556), #TotalMatches: 363, #FalseMatches: 1, accuracy: 99.7%.

images. The correctness of matches was carefully visually inspected. It is observed that our algorithm is robust under common image transformations, namely illumination, scale, rotation, viewpoint changes, image blur, and JPEG compression and in poorly-textured scene. The accuracies are above 95% on all image pairs. The robustness of our method owes to the following factors. The first one that the proposed LJJ descriptor is robust, which is specially designed for LJJ while incorporates many benefits of SIFT, leading to its robustness and high effectiveness for matching LJJs. The second one is the adopted match-propagation strategy. Through the exploitation of the topological consistency between LJJ matches and their neighboring point



Figure 15: (c) viewpoint change. #DetectedLines: (1071, 1016), #Total-Matches: 792, #FalseMatches: 3, accuracy: 99.6%.



Figure 16: (d) poorly-textured scene. #DetectedLines: (102, 82), #Total-Matches: 33, #FalseMatches: 1, accuracy: 97.0%.

matches and the recursive scheme of adding new matches while deleting possibly false ones, the group of LJL matches is expanded while false matches are well limited, which is crucial because the subsequent step of matching line segments in individuals is based on these LJL matches. False LJL matches may bring in more false line segment matches. The third one is the novel approach of matching individual line segments by utilizing local homographies recovered from neighboring corresponding LJLs. This strategy is unaffected by most image transformations.

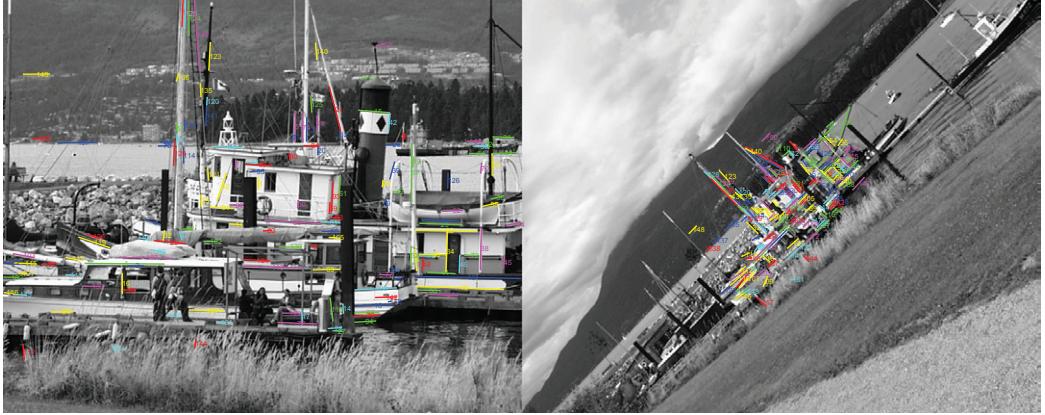


Figure 17: (e) scale & rotation change. #DetectedLines: (1334,569), #TotalMatches: 151, #FalseMatches: 7, accuracy: 95.4%.



Figure 18: (f) rotation change. #DetectedLines: (526, 398), #TotalMatches: 311, #FalseMatches: 2, accuracy: 99.4%.

Figure 21 shows the incremental process of finding correct line matches in different stages of our method on the 8 image pairs shown in Figures 13–20. It is observed from this figure that the proportions of the numbers of correct matches found in different stages accounting for the numbers of the total correct matches vary greatly on different image pairs. For example, the numbers of correct matches found in the first stage account for nearly 90% of the numbers of the total correct matches on the image pairs (b) and (c) while less than 50% on the image pairs (e) and (g). On all image pairs, the numbers of correct matches found in the second stage account for a small



Figure 19: (g) image blur. #DetectedLines: (1700, 454), #TotalMatches: 334, #FalseMatches: 11, accuracy: 96.7%.



Figure 20: (h) JPEG compression. #DetectedLines: (590, 1083), #TotalMatches: 356, #FalseMatches: 14, accuracy: 96.1%.

part of the numbers of the total correct matches. This is because most L_{JL} matches were found correctly in the first stage, and there were only a few left and could be found. However, the second stage still plays an important role in the algorithm not only because some new L_{JL} matches would be added, but also because false matches introduced in the first stage would be eliminated in this stage, which is significant for limiting false line matches.

While counting the numbers of correct matches obtained in different stages of our method, we found that, on all image pairs, almost all false matches among the final line matches are introduced in the third stage, in which line segments are matched in individuals by utilizing the local homo-

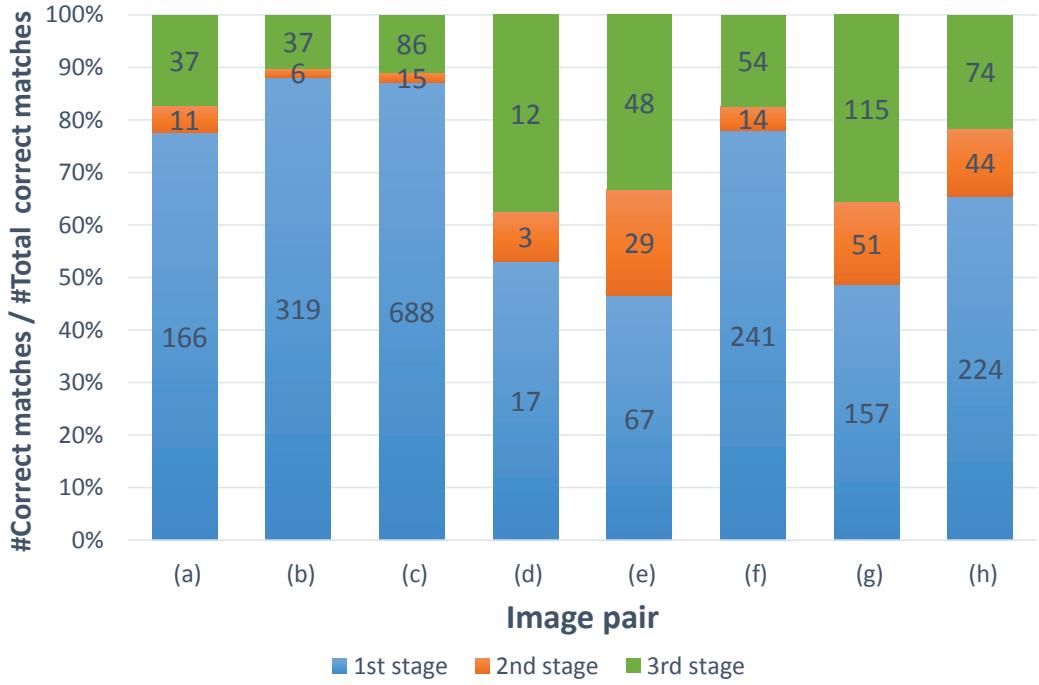


Figure 21: The incremental process of finding correct line matches in different stages of the method. The number in each bin denote the number of correct line matches found in certain stage.

graphies estimated from their neighboring LJJ matches. This is because the homography estimated from a pair of matched LJJ is not so precise. If the line segments in a LJJ match are not so precisely detected and located, the homography results from them would fluctuate around the precise one. When there are several line segments clustered in a small region in an image, this not so precise homography may lead to the method’s incapability of finding the correct one as the correspondence for a line segment from another image. However, since some additional constraints are applied when matching line segments in individuals, false matches are well controlled in our method.

We then compared our method with the state-of-the-art line matching methods. Three methods were employed for the comparison. The first one matches line segments in individuals, Lines-Points Invariant (LPI) [23], the second one matches line segments in groups, Line Signature (LS) [34], and the last one is our previous work Ray-Point-Ray (RPR) [36]. The implementations of LS and LPI were provided by their authors. To eliminate the

	LPI lines		LS lines		RPR lines	
	Ours	LPI	Ours	LS	Ours	RPR
(a)	(214, 97.5%)	(136, 99.3%)	(257, 99.2%)	(189, 97.9%)	(171, 97.2%)	(124, 99.2%)
(b)	(362, 99.7%)	(328, 99.4%)	(579, 99.3%)	(241, 99.6%)	(298, 100%)	(240, 100%)
(c)	(789, 99.6%)	(735, 99.5%)	(1229, 99.1%)	(269, 99.3%)	(699, 99.6%)	(546, 99.6%)
(d)	(32, 97.0%)	(16, 100%)	(31, 96.9%)	(42, 95.5%)	(24, 100%)	(23, 95.8%)
(e)	(144, 95.4%)	(82, 94.3%)	(33, 49.3%)	(0, 0)	(118, 94.4%)	(16, 94.1%)
(f)	(309, 99.4%)	(276, 99.3%)	(526, 99.1%)	(214, 99.7%)	(260, 97.7%)	(124, 99.2%)
(g)	(323, 96.7%)	(82, 93.2%)	(205, 94.0%)	(17, 73.9%)	(151, 92.1%)	(0, 0)
(h)	(342, 96.1%)	(225, 93.4%)	(311, 95.4%)	(126, 92.6%)	(259, 95.6%)	(137, 94.5%)
	97.7%	97.3%	91.5%	82.3%	97.1%	85.3%

Table 3: Comparative line matching results of our method with three state-of-the-art line matching methods: LPI [23], LS [34] and RPR [36]. For each of these three methods, we took its used line segments as input for our method and generated our line matching results. The dual elements shown in the table represent the number of correct matches and the accuracy, respectively. The last row represents the average accuracy of each method.

influence of different line detection methods on the line matching results, we took line segments used by these three methods as input for our method. The comparative results are shown in Table 3. The results of LPI on some image pairs shown in this table is somewhat different (generally better) with that shown our previous paper [36]. This is because the provided implementation of LPI only uses the line segments whose lengths are above 20 pixels as input for line matching. But while doing experiments, we found that the line matching results of LPI became better, quite more correct matches and comparable accuracy, if the requirement for the lengths of line segments is removed and all detected line segments are used for the matching. So, this paper shows only the better results of LPI while our previous paper shows the results generated by the original implementation of LPI. Since the provided implementation of LPI also uses LSD [40] to extract line segments, the results of our methods using line segments provided by LPI shown in the table are the same as those shown in Figures 13–20.

Several interesting observations can be made from Table 3. The first is that on the same image pairs, when using different line segments as input, the line matching results of the same method vary a lot. Our method gen-

erated quite different results, both in the numbers of correct matches and the corresponding accuracy, when using different input line segments. For example, on the image pair (e), where great scale and rotation changes exist, our method generated fairly good result, 144 correct matches with the accuracy of 95.4% when using line segments provided by LPI, however, the result drops drastically with only 33 correct matches and the accuracy is only 49.3% when using line segments provided by LS. The second is that when using line segments provided by LPI as input, our method produced more correct matches on all image pairs than that of LPI, and the average accuracy is higher despite that on the image pairs (a) and (d), the accuracies are slightly lower. On the image pair (g), where great image blur exists, our method produced nearly 4 times of correct matches than that of LPI. The third is that when using line segments detected by LS as input, our method has quite better performance than LS itself. Due to the multi-scale scheme, LS produced large groups of line segments, which caused matching them being very time-consuming and memory-consuming. With such large groups of line segments as input, compared with LS, our method produced line matches with quite higher average accuracy and owned an overwhelming advantage for the amount of correct matches on some images pairs. Our method found more than 12 times of correct matches than that of LS on the image pair (g) and nearly 5 times of correct matches on the image pair (c). Besides, on the image pair (e), LS failed to produce any correct line match, while our method can still produce some despite with a low accuracy. The fourth is that, by taking line segments used in RPR (detected by EDLines [41]) as input, the proposed method excels both at the amount of correct matches and at the accuracy. Remarkably, RPR failed on the image pair (g), where there is great blur between two images, while the proposed method can still produce good results with 151 correct matches with the accuracy of 92.1%. These advantages of the proposed method over RPR prove the effectiveness of the promotions we have made based on RPR.

5.4. Further Comparison with LPI

From Table 3, we can conclude that our method and LPI are the two most robust methods. So, we conducted experiments to further evaluate these two methods.

We first conducted experiments on some datasets related by global homographies to evaluate these two methods . The six image datasets [43, 18]



Figure 22: Six image datasets characterized by various image transformations. There are 6 images with gradual image transformation in each dataset, and only the first and the last are shown here.

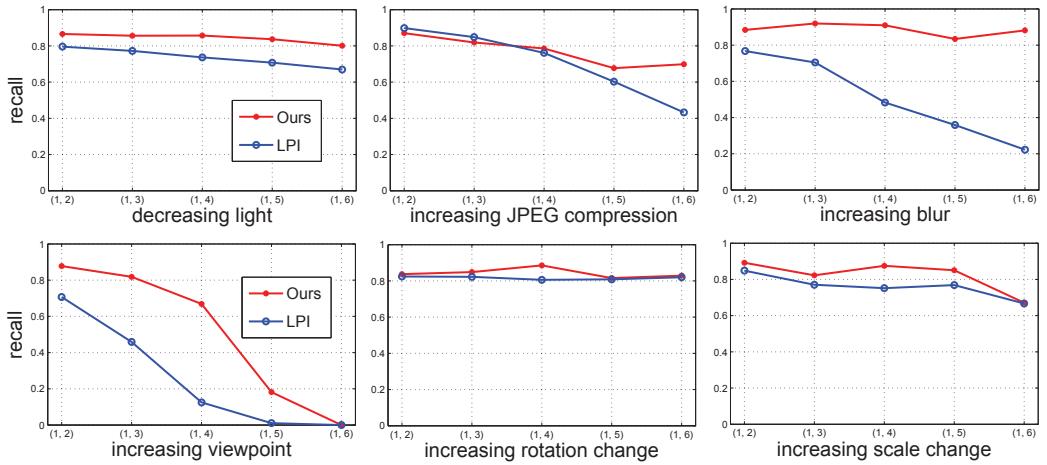


Figure 23: The recalls of the line matching results of the proposed method and LPI [23] on six image datasets characterized by various image transformations.

with various image transformations, including illumination, rotation, viewpoint and scale changes, image blur and JPEG compression, were employed for the following experiments. There are six images in each dataset, and the first and the last are shown in Figure 22. Since the image homographies between the first and the other five images in each dataset are available, the ground truth of the line segment matches between images can be established by mapping line segments detected in one image to the other images and finding if there are line segments in the very close regions around the mapped line segments. With the ground truth of line segment matches between im-

ages, the recalls (the ratio between the number of correct matches and the number of ground truth correspondences) of line matching results of these two methods can be calculated. In each datasets, the first image was used to match the other five images for line segments by these two methods, and the corresponding recalls are shown in Figure 23. It is observed from this figure that the recalls of the line matching results generated by our method are higher than those of LPI on almost all image pairs under all these six kinds of image transformations except the two image pairs where JPEG compression between images exists.

Beside experimenting on the common datasets, we conducted additional experiments on some very challenging image pairs. The experimental results are shown in Figure 24, from which we can observe that under these challenging cases, our method is more robust and produces quite more correct matches.

6. Discussion

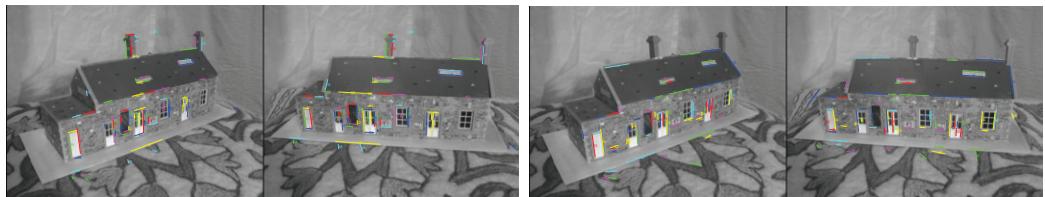
All the line matching results of our method presented above are based on the fixed parameters, which we have discussed in Section 5.1. In this section, we will further discuss about how to adjust the values of some parameters to improve the performance of our method for some specific applications.

6.1. Time Performance

Figure 25 shows the elapsed time (in seconds) of each stage of our method and the percentage it accounts for the corresponding total elapsed time on each of 8 image pairs shown in Figures 13–20. The proposed method was implemented with C++ and the computation time was measured on a 3.4GHz Inter (R) Core(TM) processor with 12 GB of RAM. It can be observed from this figure that the total elapsed time of our method varies a lot on different image pairs. Our method took nearly 660 seconds on the image pair (c) while less than 2 seconds on the image pair (d). Generally, the more complex the scenes captured are, the more time it takes for our method to match the line segments extracted from the images. This is because more line segments can be detected in images of complex scenes and matching larger groups of line segments costs more time. Another observation from this figure is that the time spent in the first stage of our method makes a dominant account for the total elapsed time on all image pairs. The reason behind this is that by building image pyramids, each LJL constructed in the original images is adjusted



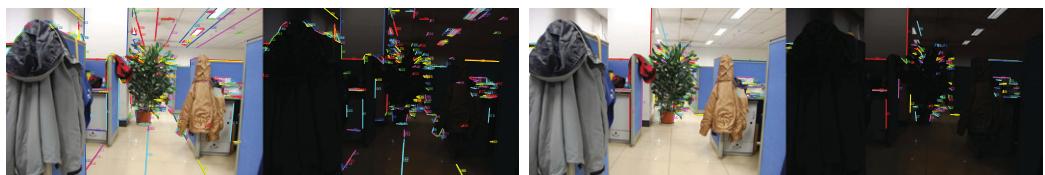
(a) Wide-baseline: (Left) Ours, #TotalMatches: 178, #CorrectMatches: 133; (Right) LPI, #TotalMatches: 62, #CorrectMatches: 24.



(b) Wide-baseline: (Left) Ours, #TotalMatches: 81, #CorrectMatches: 66; (Right) LPI, #TotalMatches: 69, #CorrectMatches: 34.



(c) Poorly-textured & light change & wide-baseline: (Left) Ours, #TotalMatches: 75, #CorrectMatches: 54; (Right) LPI, #TotalMatches: 4, #CorrectMatches: 2.



(d) Light change: (Left) Ours, #TotalMatches: 236, #CorrectMatches: 233; (Right) LPI, #TotalMatches: 83, #CorrectMatches: 81.

Figure 24: Comparative results between our method and LPI on some challenging image pairs. Please zoom in for better interpretation.

to all images in the pyramids and is described there. The time of describing and matching LJLs from two images increase with the number of the levels of the image pyramids . For example, on the image pair (c), which took the

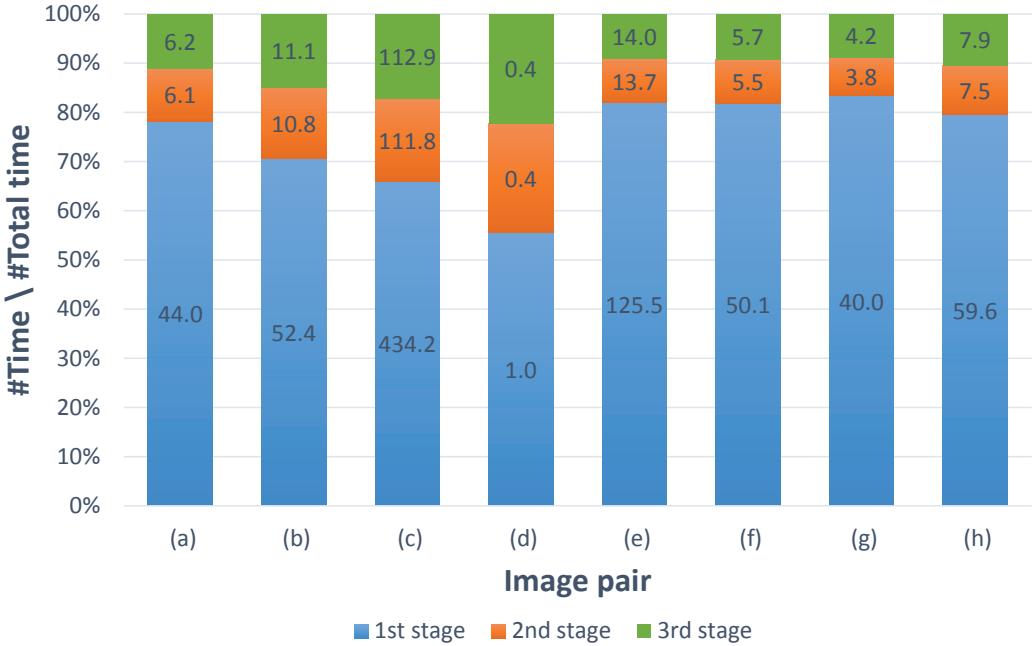


Figure 25: The elapsed time (in seconds) of each stage of our method and the percentage it account for the corresponding total elapsed time of the method on each of the eight image pairs shown in Figures 13–20. The number in each bin denotes the elapsed time in certain stages of the method on certain image pair.

most time by our method, 4856 LJLs were constructed in the first image and 4693 LJLs in the second image. When the Gaussian image pyramids built for two original images have 4 octaves with 2 scales in each octave, there are $4856 \times 8 = 38848$ LJLs and $4693 \times 8 = 37544$ LJLs need to be described for the two images. A LJL descriptor is a vector of 128 dimensions and matching such large two groups of LJLs by evaluating the distances between their descriptor vectors of such high dimension is definitely time-consuming.

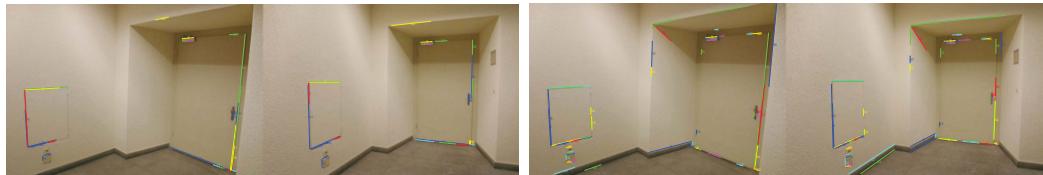
It seems that our method is impractical for some applications which have strict requirement on the running time. However, the time performance of our method can tremendously be improved through the following aspects. The first is to adjust some parameters for specific scenes. The majority of time is spent in describing and matching LJLs from two images and there are three parameters that control the number of the LJLs to be described

	Matching Results		Running Times (s)		
	Method I	Method II	Method I	Method II	Reduction
(a)	(214, 97.5%)	(212, 99.1%)	56.3	11.2	80.1%
(b)	(362, 99.7%)	(379, 99.2%)	74.3	27.2	63.4%
(c)	(789, 99.6%)	(786, 99.6%)	658.9	241.5	63.3%
(d)	(32, 97.0%)	(24, 100%)	1.9	0.7	62.0%
(f)	(309, 99.4%)	(317, 99.1%)	61.4	13.9	77.4%
(h)	(342, 96.1%)	(334, 96.5%)	75.0	12.4	83.5%

Table 4: The line matching results of the proposed algorithm generated by building Gaussian image pyramids (Method I) and that of without building Gaussian image pyramids (Method II) on some image pairs shown in Figures 13–20. The last column shows reduction ratio of the running time of Method II relative to Method I.

and matched. These three parameters are w that controls the size of the affect region of a line segment, the number of octaves of the image pyramids, o , and the number of scales per octave of a pyramid, s . Both o and s are introduced when building Gaussian image pyramids to deal with the possible scale changes between images. If we have the priori that there is no or merely slight scale change or some fixed scale change between images, then all the steps intended to deal with scale changes between images are needless. We can match LJLs constructed in the original images or some specifically scaled ones directly, which would save plenty of time.

Table 4 shows the line matching results on the six image pairs, (a)–(d), (f) and (h) by building the Gaussian image pyramids (Method I) and that of without building the Gaussian image pyramids (Method II), and the corresponding running time. All these six image pairs share the similarity that there are very little scale changes between two images and thus building Gaussian image pyramids is unnecessary for them. From Table 4, we can see that the matching results generated by Method II are similar with those of Method I both in the amounts of correct matches and the accuracy, but the running time is reduced drastically. On all image pairs, Method II took less than half of the running time of Method I, and on some image pair, (a) and (h), the reduction ratios are more than 80%, which means Method II used less than 20% of the running time of Method I. Apart from choosing to not build image pyramids for images with very little scale change to save



(a) “Corridor”. (Left) $w = 10$, #CorrectMatches: 30; (Right) $w = 60$, #CorrectMatches: 55



(b) “Corner”. (Left) $w = 10$, #CorrectMatches: 28; (Right) $w = 60$, #CorrectMatches: 36

Figure 26: Line matching results of the proposed method in two poorly-textured scenes, “Corridor” and “Corner”, when the parameter w was set as 10 and 60. Please zoom in for better interpretation.

time, decreasing the value of the parameter w can also help to reduce the running time since less LJLs are constructed with a smaller value of w . This strategy is especially efficient when scenes are rich-textured. Our method requires a fairly large value of w to produce considerable amounts of intersecting junctions of line segments, but for rich-textured scenes, plenty of line segments can be detected and a small value of w can produce sufficient junctions for matching line segments. For example, on the image pair (c), where the scene has rich texture and more than 1000 line segments were extracted in both images, when the value of w was set as 20 in pixels, our method spent 658.9 seconds matching line segments from the two images when building the Gaussian image pyramids and 241.5 seconds without building Gaussian image pyramids. But when we set $w = 5$ without building the Gaussian image pyramids, our method spent only 17.2 seconds and produced 788 correct line matches with the accuracy of 99.6%. The matching result is similar with those generated under a greater value of w , but the cost time drops drastically. So, for scenes with the rich texture, selecting a smaller value of w can greatly promote the efficiency of the method. Another way to accelerate our method is to introduce parallel computing into the algorithm. Most of the time of our method is spent on describing and matching LJLs, which is very suitable for parallel computing.

	$w = 10$	$w = 20$	$w = 30$	$w = 40$	$w = 50$	$w = 60$
“Corridor”	27	33	35	36	53	55
“Corner”	28	33	36	37	37	36

Table 5: The numbers of correct line matches obtained by the proposed method in two poorly-textured scenes when the value of the parameter w was progressively increased.

6.2. Poorly-Textured Scenes

While conducting experiments, we found that for image pairs that were captured from poorly-textured scenes, if we increase the value of the parameter w that controls the size of the affect region of a line segment when constructing LJLs, the line matching results are generally better. Figure 26 shows the line matching results of the proposed method in two poorly-textured scenes when w was set as 10 and 60. Table 5 shows how the numbers of correct matches produced by our method on the two scenes change by varying the value of w from 10 to 60 at the step of 10. We can observe from the table the trend that the number of correct matches increases as the value of w becomes bigger and turn relatively stable when the value of w has been great enough.

This trend can be explained as follows. In poorly-textured scenes, only a small amounts of line segments can be detected, and with a greater value of w , more line segments can be regraded as adjacent line segments and used to generate junctions and construct LJLs. More LJLs in poorly-textured scenes often result in a larger group of initial LJL matches, which improves the line matching results in the following three aspects. First, more line segments are matched in the form of LJL. Second, a generally preciser fundamental matrix can be obtained, which helps to both propagate LJL match (in the second stage) and match line segments in individuals (in the third stage). Third, LJL matches may distribute in more 3D planes. The third stage of our method, matching line segments in individuals, underlies the assumption that the 3D projections of two line segments to be matched lie in the same 3D plane that lay by the 3D projections of the two pairs of matched line segments brought by a pair of matched LJLs. If two individual line segments whose 3D projections lie in planes where none of the 3D projections of matched LJLs exists, the two line segments cannot be matched by our method. Thus,

a larger group of initial L JL matches can help to bring in more individual line segment matches. The number of correct matches stops increasing if the value of w has been big enough because only these line matches can be found by our method in these scenes.

7. Conclusions

This paper has presented a hierarchical line matching method in which line segments are first matched in the form of a structure called Line-Junction-Line (L JL) formed by two adjacent line segments and their intersecting junction and then matched in individuals. While matching L JLs, a robust descriptor as well as an effective strategy to deal with the possible scale changes between images are proposed to obtain initial L JL matches, which are then refined and expanded by an effective match-propagation scheme. Those left unmatched line segments are further matched by exploiting the local homographies estimated from their neighboring L JL matches. The experimental results show the robustness of the proposed L JL descriptor for matching L JLs and the good performance of the proposed method under most kinds of image transformations and in poorly-textured scenes. The superiorities of the proposed method to the state-of-the-art line matching methods include its robustness for more kinds of situations, the larger amounts of correct matches, and the higher accuracy in most cases.

Acknowledgement

This work was partially supported by the Hubei Province Science and Technology Support Program, China (Project No. 2015BAA027), the National Basic Research Program of China (Project No. 2012CB719904), the National Natural Science Foundation of China (Project No. 41271431), and the South Wisdom Valley Innovative Research Team Program. Thanks for Bin Fan, Lilian Zhang and Lu Wang for providing the implementations of their methods and some test images.

- [1] G. N. DeSouza, A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(2): 237-267, 2002.
- [2] 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(2): 229-246, 2005.

- [3] A. Bartoli, P. Sturm. Multiple-view structure and motion from line correspondences In ICCV, 2003.
- [4] A. Bartoli, P. Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding (CVIU)*, 100(3): 416-441, 2005.
- [5] C. J. Taylor, D. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(11): 1021-1032, 1995.
- [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): 889-896, 2006.
- [7] N. Snavely, S. M. Seitz, R. Szeliski. Photo tourism: exploring photo collections in 3D. *ACM transactions on graphics (TOG)*, 25(3): 835-846, 2006.
- [8] B. Neumann, R. Mller. On scene interpretation with description logics. *Image and Vision Computing*, 26(1): 82-101, 2008.
- [9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [10] Z. Wang, B. Fan, F. Wu. Local intensity order pattern for feature description. In ICCV, 2011.
- [11] J. M. Morel, G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2): 438-469, 2009.
- [12] S. Winder, G. Hua, M. Brown. Picking the best daisy. In CVPR, 2009.
- [13] Y. Ke, R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors In CVPR, 2004.
- [14] H. Bay, T. Tuytelaars, G. L. Van. Surf: Speeded up robust features In ECCV, 2006.

- [15] C. Schmid, A. Zisserman. Automatic line matching across views. In CVPR, 1997.
- [16] C. Baillard, C. Schmid, A. Zisserman, A. Fitzgibbon. Automatic line matching and 3D reconstruction of buildings from multiple views. In ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, 1999.
- [17] Z. Wang, F. Wu, Z. Hu. MSLD: A robust descriptor for line matching. Pattern Recognition, 42(5):941–953, 2009.
- [18] L. Zhang, R. Koch. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. J. Visual Communication and Image Representation, 24(7): 794–805, 2013.
- [19] B. Verhagen, R. Timofte, L. G. Van. Scale-invariant line descriptors for wide baseline matching. In WACV, 2014.
- [20] H. Bay, V. Ferrari, L. Van Gool. Wide-baseline stereo matching with line segments. In CVPR, 2005.
- [21] M. I. Lourakis, S. T. Halkidis, S. C. Orphanoudakis. Matching disparate views of planar surfaces using projective invariants. Image and Vision Computing, 18(9):673–683, 2000.
- [22] B. Fan, F. Wu, Z. Hu. Line matching leveraged by point correspondences. In CVPR, 2010.
- [23] B. Fan, F. Wu, Z. Hu. Robust line matching through line-point invariants. Pattern Recognition, 45(2):794–805, 2012.
- [24] M. Chen, Z. Shao. Robust affine-invariant line matching for high resolution remote sensing images. Photogrammetric Engineering & Remote Sensing, 79(8):753–760, 2013.
- [25] H. Bay, A. Ess, A. Neubeck, L. Van Gool. 3D from line segments in two poorly-textured, uncalibrated images. In 3DPVT, 2006.
- [26] B. Micusik, H. Wildenauer, J. Kosecka. Detection and matching of rectilinear structures. In CVPR, 2008.

- [27] H. Kim, S. Lee. A novel line matching method based on intersection context. In ICRA, 2010.
- [28] H. Kim, S. Lee. Simultaneous line matching and epipolar geometry estimation based on the intersection context of coplanar line pairs. Pattern Recognition Letters, 33(10):1349–1363, 2012.
- [29] H. Kim, S. Lee, Y. Lee. Wide-baseline stereo matching based on the line intersection context for real-time workspace modeling. Journal of the Optical Society of America A, Optics, Image Science, and Vision, 31(2):421–435, 2014.
- [30] K. Mikolajczyk, C. Schmid. Scale and Affine invariant interest point detectors. International Journal of Computer Vision, 60(1):63-86, 2004.
- [31] T.Tuytelaars, L. Van Gool. Matching widely separated views based on affine invariant regions. newblock International Journal of Computer Vision, 59(1):61-85, 2004.
- [32] J. Matas, O. Chum, M. Urban, T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In BMVC, 2004.
- [33] T. Kadir, A. Zisserman, M. Brady. An affine invariant salient region detector. In ECCV, 2004.
- [34] L. Wang, U. Neumann, S. You. Wide-baseline image matching using line signatures. In ICCV, 2009.
- [35] L. Wang, Adviser U. Neumann. Line segment matching and its applications in 3d urban modeling[M]. University of Southern California, 2010.
- [36] K. Li, J. Yao, X. Lu. Robust line matching based on Ray-Point-Ray descriptor. In ACCV 2014 Workshop on robust local descriptors for computer vision, 2014.
- [37] M. Zuliani. RANSAC for dummies. With examples using the RANSAC toolbox for Matlab and more, 2009.
- [38] Q. -T. Luong, T. Viville. Canonical representations for the geometries of multiple projective views. Computer Vision and Image Understanding, 64(2):193–229, 1996.

- [39] A. Anubhav, C. V. Jawahar, P. J. Narayanan. A survey of planar homography estimation techniques. Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12, 2005.
- [40] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall. LSD: A fast line segment detector with a false detection control. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 32(4):722–732, 2012.
- [41] C. Akinlar, C. Topal. EDLines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633–1642, 2011.
- [42] K. Mikolajczyk, C. Schmid. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 27(10): 1615–1630, 2005.
- [43] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir L. Van Gool. A comparison of affine region detectors. International Journal of Computer Vision, 65(1/2):43-72, 2005.