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, the line segments lying adjacently are intersected to form junctions. At the places of the junctions, the structures are constructed called Line-Junction-Line (LJL), which consists of two adjacent lines segments and their intersecting junction. To reliably deal with the possible scale change between the two images to be matched, we propose to describe LJLs by a robust descriptor in the multi-scale pyramids of images constructed from two original images. By evaluating the description vector distances of LJLs from the two images, some candidate LJL matches are obtained, which are then refined and expanded by an effective match-propagation strategy. The line segments used to construct LJLs are matched when the LJLs they formed are matched. For those left unmated line segments, we match them in individuals by utilizing the local homographies estimated from their neighboring matched LJLs. Experiments demonstrate the superiorities of the proposed method to the state-of-the-art methods for its robustness in more difficult situations, the larger amounts of correct matches, and the higher accuracy in most cases.

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

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, like robotic navigation, structure from motion, 3D reconstruction, scene interpretation, etc. The majority of image matching methods are feature point-based [1, 2, 3, 4, 5, 6] 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, especially for man-made scenes. This indicates that it is better to recover 3D scene structures based on line segments than that on feature points, at least for some scenes [7, 8, 9, 10, 11, 12]. For example, for poorly-textured scenes, 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 [13]. Despite of these 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 partly explains why line segment matching has been less investigated.

Line matching methods in existing literatures can generally be classified into two categories: the 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 [14, 15], gradient [16, 17, 18], and color [19] in the local regions around line segments. All these methods underlie the assumption that there are considerable overlaps between corresponding line segments. This assumption leads to the failure of these methods in situations where 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 [20, 21, 22, 23]. These methods first find a large group of point matches using the existing point matching methods, and then exploit the invariants between coplanar points and line(s) under certain image transformations to evaluate the correspondence of the line segments from two images. The 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 are thus robust even when local shape distortions are severe. However, these methods share a common disadvantage that they are incapable of processing images in which the scenes captured are poorly-textured since feature points are hard to be detected and matched in this kind of scenes, which consequently disables the use of point matches for line segment matching.

Matching line segments in groups is more complex, but more

*Corresponding author.
Email address: jian.yao@whu.edu.cn (Jian Yao)
URL: http://cvrs.whu.edu.cn/ (Jian Yao)

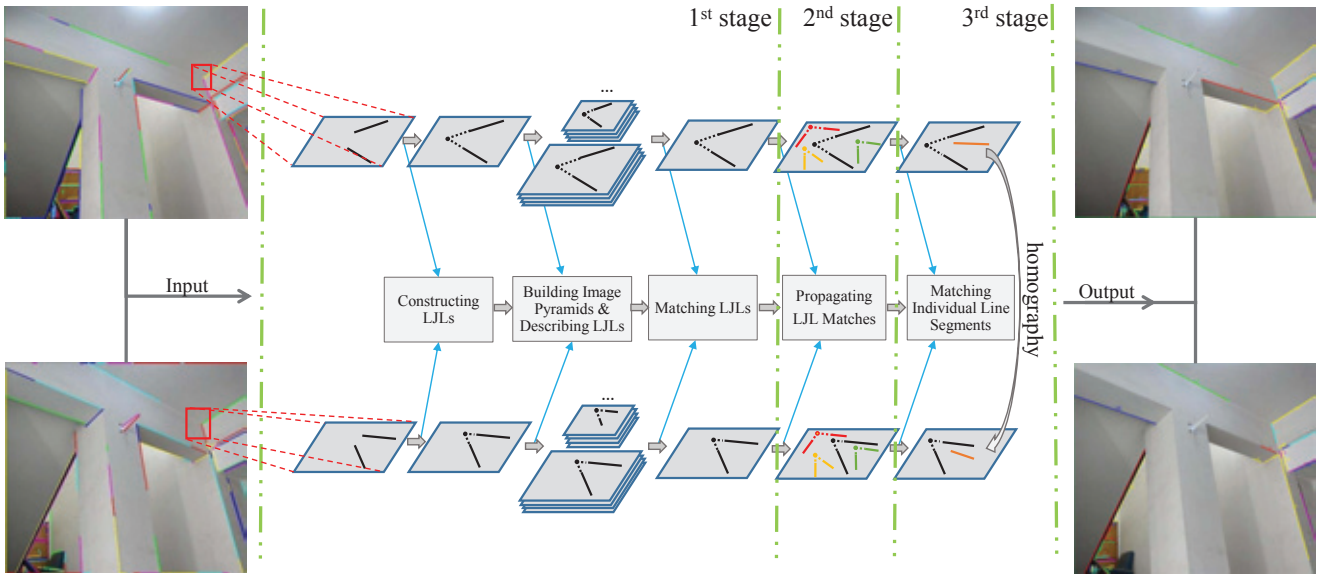


Figure 1: The flowchart of the proposed line matching algorithm.

constraints are available for disambiguation. Most of these methods [13, 24, 25, 26, 32] first use some strategies to intersect line segments to form junction points and then utilize features associated with the generated junction points for line segment matching. These methods transfer line matching to point matching, a widely investigated problem which many effective algorithms target to solve. But junction points contain more information than feature points detected by some detectors [27, 28, 29, 30]. They are the results of intersecting pairs of adjacent line 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 junction points to help match them is still an open issue. In [31], 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 description 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 fairly stable under most image transformations, the description of the configurations of two groups of line segments in correspondence should be similar. In this way, 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.

Our proposed line matching method in this paper is a combination of the two categories methods described above. It 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. For two sets of line segments extracted from two images, the first stage commences intersecting neighboring line segments to form junction points. At the places of the formed junction points, we form the structures called Line-Junction-Line (LJL), utilizing two adjacent line segments and their intersecting junction. To greatly reduce the effect of the scale change between the two images, we propose to build Gaussian image pyramids for the original images and adjust the LJLs constructed in the original images to fit each image in the image pyramids and described them there by the proposed LJL descriptor. Some initial LJL matches can be found by evaluating the description vector distances of LJLs from the two images. 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. In the above two stages, the line segments lying closely with each other from the two images are matched along with their constructed and matched LJLs. For those line segments lying far away from others and are not used to constructed LJLs, we match them in individuals in the third stage by utilizing the local homographies estimated from their neighboring matched LJLs.

This work is an extension of our work presented in [32]. 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 [32], a sets of line segments extracted by some line segment detectors in a image are processed in advance before they are used for matching by a series of procedures to obtain a new line segment set where some line segments are extended to be longer and are connected with others. In this work, the line segments extracted by line detectors are not required to be refined in advance, and can be used to generate junctions directly based on the local spatial proximity. This promotion helps generate more

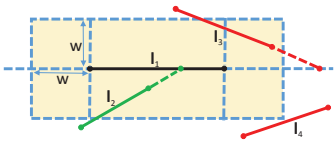


Figure 2: An illustration of finding line segments possibly coplanar in 3D space. The rectangle filled in yellow is the affect region of $\mathbf{l}_1$. $\mathbf{l}_2$, $\mathbf{l}_3$ and $\mathbf{l}_4$ are the three neighbors of $\mathbf{l}_1$. w is a parameter controlling the size of the affect region of $\mathbf{l}_1$.

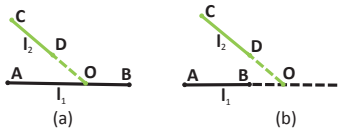


Figure 3: Two configurations of a pair of line segments intersecting with each other.

junctions and contributes to better matching results. Second, a more robust descriptor is proposed to describe the structure (called RPR in [32], while LJL in this paper) formed by two adjacent line segments and their intersecting junction. Third, we propose a more reasonable strategy to deal with the possible scale change between images. To match line segments from two images with scale change, in [32], the global scale change factor between the two original images is estimated and one of the two images is adjusted to the same scale as the other one. This strategy is reasonable only when the scale change between the two images is a global one. When scale changes between the two images vary with local regions (often introduced by viewpoint changes), 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 locally variant ones. Fourth, a more sophisticated way to match individual line segments is proposed. For line segments which cannot be matched in groups (in RPRs in [32] while in LJLs in this paper), in [32], they are used to intersect with those matched line segments to construct new RPRs and matched along with the newly constructed RPR again. In this paper, they are matched by the local homographies estimated from their neighboring corresponding LJLs. All the above promotions together contribute to the better performance of this work than our previous one.

Experimental results substantiate the advantages of this work over the previous one and other state-of-the-art line matching methods for its robustness under more severe image transformations, its better performance for poorly-textured scenes, the larger amount of correct line matches obtained, and higher accuracy in the majority of cases.

The remainder of the paper is organized as follows. Section 2 introduces the details of constructing, describing and matching LJLs. The adopted match propagation strategy is described in Section 3. The step of matching individual line segments failed to be matched along with LJLs 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 Match Generation

The endpoints of line segments are unreliable 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 are thus 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 forming them. Therefore, what needs 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 a good choice to intersect neighboring line segments to obtain reliable junctions. We use a similar method as that presented in [26] to generate junctions. Refer to Figure 2, for a line segments $\mathbf{l}_1$, we define its affect region as a rectangle (filled in yellow 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 its 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 $\mathbf{l}_1$ in 3D space. $\mathbf{l}_3$ is rejected because its intersection with $\mathbf{l}_1$ is not within the affect region of $\mathbf{l}_1$ despite that 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 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, (**OA**, **O**, **OC**) and (**OB**, **O**, **OC**) , are constructed. In Figure 3(b), the intersection lies on neither of the two line segments. Only one LJL, (**OA**, **O**, **OC**), is constructed.

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 [1], 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 four parts. Each part

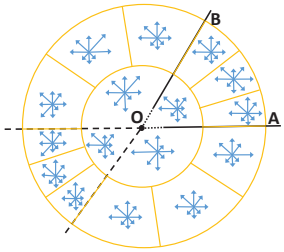


Figure 4: An illustration of describing a LJL, (**OA**, **O**, **OB**), with the proposed LJL descriptor.

contains a sector and a ring-shaped region, which is then evenly divided into three subregions, resulting in that the three subregions have the same area with the sector. Therefore, there are totally 16 regions, two groups of eight 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 in SIFT. A normalization on the description vector is necessary to reduce the effect of illumination change. Since a LJL descriptor is generated by 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 eight 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.

2.3. LJL Matching

To match LJLs from two images, the general way is to evaluate their description vector distances. But since each LJL consists of two line segments and their intersecting 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, suppose θ_1 and θ_2 denote the crossing angle of the two line segments of the two LJLs, respectively, the absolute difference of θ_1 and θ_2, $\theta = |\theta_1 - \theta_2|$ should be a small value ($\theta = 30°$ in this paper) if the LJL pair is a correct match. This constraint can help discard many false candidates before evaluating their description vectors and thus contributes to better matching results.

If a pair of LJLs satisfy the above constraint, we then evaluate their description vectors. The proposed LJL descriptor is based on a 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 two images to be matched and adjust, if image sizes change, LJLs constructed in original images to

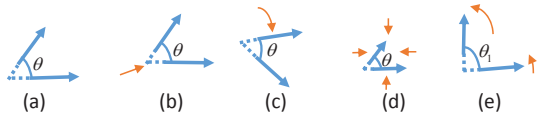


Figure 5: The changes of the crossing angle of the two line segments in a LJL with different transformations. (a): the original LJL; (b)-(e): the transformed LJLs, with translation (b), rotation (c), scale (d) and affine (e) changes.

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 makes 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. It 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 $\mathcal{L}$ in the i-th level and the sub-descriptors of $\mathcal{L}'$ in the j-th level, respectively. The distances between each sub-descriptor in $\mathcal{D}$ and all sub-descriptors 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 between $\mathcal{D}$ and $\mathcal{D}'$. Those LJL pairs whose description vector distances are smaller than a given threshold d_v ($d_v = 0.5$ in this paper) are regarded as candidate LJL matches.

The 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 description vector distance is smallest among all the sub-descriptor pairs. In case that a pair of sub-descriptors which are calculated in different scales but accidentally have the smallest description vector 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 vector distances of all the sub-descriptors as the description 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. After that, we obtain the fundamental matrix as well as a group of LJL matches consistent with it. Based on these, 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 served as the basis for the next iteration of introducing new LJL matches.

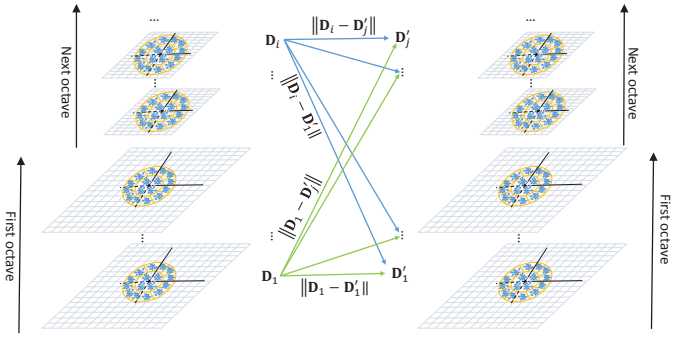


Figure 6: An illustration of describing and matching a pair of LJLs in a multi-scale Guassian image pyramid scheme.

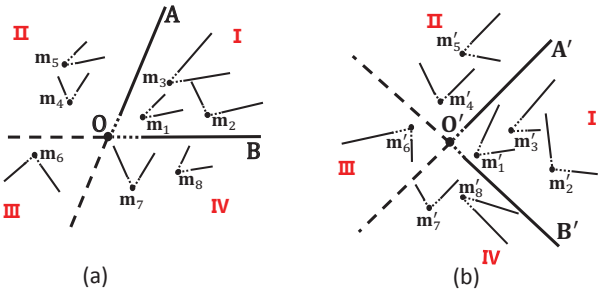


Figure 7: An illustration of the distribution of a pair of corresponding LJLs, (**OA**, **O**, **OB**) and (**O′A′**, **O′**, **O′B′**), and their neighboring matched points brought by matched LJLs.

The topological distribution 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 should be consistent with their neighboring point matches in the topological distribution. Refer to Figure 7, (**OA**, **O**, **OB**) and (**O′A′**, **O′**, **O′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 in 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 **O** and **O′** in $\mathcal{L}$ 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. 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 is set to 1 in the initial iteration and 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 distribution constraint presented 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 procedures 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. 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 the 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 $\mathcal{K} = \{\mathbf{l}_i\}_{i=1}^M$ and $\mathcal{K}' = \{\mathbf{l}'_j\}_{j=1}^N$ be the two groups of individual line segments, which have not been matched before, from two images, respectively. For each individual line segment $\mathbf{l}_i \in \mathcal{K}$ or $\mathbf{l}'_j \in \mathcal{K}'$, 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 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. The individual line segments this LJL collects distribute in its four quadrants. For each such line segment, if any of its two endpoints lies in a certain quadrant of the coordinates formed by the the 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, $\mathcal{U} = \{\mathcal{I}_m^p | m = 1, 2, \cdots, V; p = 1, 2, 3, 4\}$ and $\mathcal{U}' = \{\mathcal{I}'^q_n | n = 1, 2, \cdots, V; q = 1, 2, 3, 4\}$, respectively, where $\mathcal{I}_m^p$ and $\mathcal{I}'^q_n$ 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 $\mathcal{I}'^q_n$ 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 that, 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 [33] as

$$\mathbf{H}^\top\mathbf{F} + \mathbf{F}^\top\mathbf{H} = 0. \tag{1}$$

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 [34], 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

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

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

$$\mathbf{x}' = \mathbf{H}\mathbf{x}. \tag{3}$$

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

$$\mathbf{l}'^\top\mathbf{x}' = 0. \tag{4}$$

Combining Eqs. (2)–(4), we obtain

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

Arranging the above equation, we finally get

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

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 homograpy $\mathbf{H}$ 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 a 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 $(\mathbf{l}_n, \mathbf{l}'_n)$. From $\mathcal{L}$ and $\mathcal{L}'$, the local homography, $\mathbf{H}$ is estimated, using the strategy presented in Section 4.2.

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

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

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

We then evaluate the candidate match $(\mathbf{l}, \mathbf{l}')$ by the local homography estimated from $(\mathcal{L}, \mathcal{L}')$. This method is reasonable only when the 3D correspondence(s) of $\mathbf{l}$ and $\mathbf{l}'$ lie in the same 3D plane as that of the 3D correspondences of $\mathbf{l}_m$ ($\mathbf{l}'_m$) and $\mathbf{l}_n$ ($\mathbf{l}'_n$). It is hardly possible to determine whether the 3D correspondences of several 2D line segments are in same 3D plane without the 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 images are highly possible to be coplanar in 3D space. The two line segment triples, ($\mathbf{l}_m$, $\mathbf{l}_n$, $\mathbf{l}$) and ($\mathbf{l}'_m$, $\mathbf{l}'_n$, $\mathbf{l}'$), are clustered based on the local spatial proximity, which guarantees a fairly high possibility that the 3D correspondences of the line segments in the two triples are on the same plane. On the other hand, the success of matching the two LJLs, though we cannot absolutely ensure the correctness of the matching, substantiates the 3D correspondences of $\mathbf{l}_m$ ($\mathbf{l}'_m$) and $\mathbf{l}_n$ ($\mathbf{l}'_n$) are on the same 3D 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

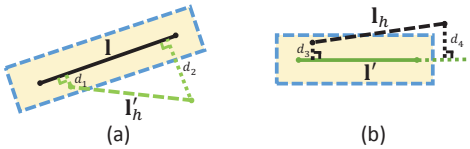


Figure 8: An illustration of evaluating a pair of candidate line segment correspondences using the estimated local homography. $\mathbf{l}$ and $\mathbf{l}'$ are the two line segments to be evaluated, $\mathbf{l}_h$ and $\mathbf{l}'_h$ are their correspondences mapped by the estimated homography.

Figure 9: An illustration of using the brighter side constraint to help match line segments. $\mathbf{l}$ is a line segment from a image. $\mathbf{l}'_1$ and $\mathbf{l}'_2$ are the two candidate correspondences for $\mathbf{l}$ from the other image. $\mathbf{l}'_1$ is accepted as the correspondence for $\mathbf{l}$ because its brighter side is the same as $\mathbf{l}$.

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) is set as 3 in pixels when applying this constraint. Refer to Figure 8, 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 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 denoted as d_1, d_2, d_3 and d_4 in Figure 8, then, the mapping error of $(\mathbf{l}, \mathbf{l}')$ is:

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

While finding the correspondence for a line segment, the above constraints are unable to discern the false candidates when they have similar directions with the correct one and are adjacent with it. Refer to Figure 9, 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 line segment matches by finding the brighter sides of line segments. Refer to Figure 9, we calculate the average intensity values of pixels in the two small 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. 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

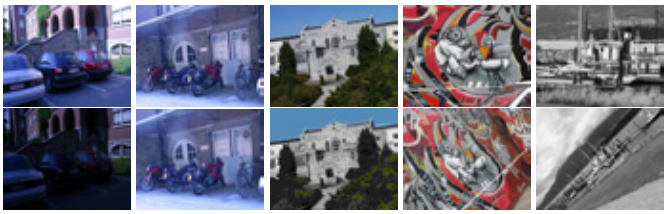


Figure 10: The five image pairs, *leuven*, *bikes*, *ubc*, *graffiti*, and *boat*, used for selecting proper values for the two key parameters in our method and for evaluating the proposed LJL descriptor.

can be exploited to find the correct correspondences for line segments. In Figure 9, 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 the other 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 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. Parameter 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 a LJL. The five representative image pairs [37] shown in Figure 10 were employed to help fix the 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, which benefits us to select optimal values for the parameters.

We first conducted experiments to select a proper value for w. It is obvious that a big value for w results in a big affect region of a line segment and more intersections of line segments and more LJLs. Sufficient LJLs are the guarantee to produce enough initial LJL matches and are therefore 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 the other group of LJLs, will harm the matching of them since more interferences are introduced, and will increase both the computation

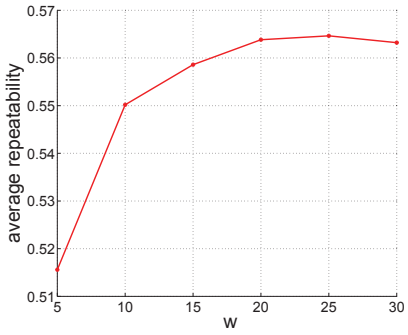


Figure 11: The changes of the average repeatability of the junction points in the constructed LJL from images shown in Figure 10 with different values of the parameter w.

time and memory to match them. We employed the way introduced in [37] by calculating the *repeatability* of two groups of points to select a proper value for w. In [37], the repeatability score is used to evaluate different local region detectors under various image transformations. It measures how the 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 junction points in constructed LJLs under various values of w and fixed w at the value with the highest repeatability. All the five image pairs shown in Figure 10 were employed for experiments and the average repeatability of the junctions in constructed LJLs in all image pairs is calculated. The change of the average repeatability with respective to w is shown in Figure 11.We observe from this figure 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 reduce the computation time, $w = 20$ was selected in this paper.

We then conducted experiments to find a proper value for the parameter r, which determines the size of the local patch exploited for describing a LJL. The LJL descriptor is often more discriminative but more sensitive to shape distortion when a bigger patch is used. Table 1 shows the accuracies of the matching results of the LJLs with various values of r on the five image pairs shown in Figure 10. Since the average accuracy reaches its maximum at $r = 10$, this setting was therefore applied in our algorithm. 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, generating their estimated correspondences, $\mathbf{j}_h$ and $\mathbf{j}'_h$, for $\mathbf{j}$ and $\mathbf{j}'$, 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 pixels, we regard the LJL match as a correct one. This correctness-assess strategy for point matches seems problematical since it may label a false point match as a correct one when one point in the match lies near the actual correspondence of the other point. However, since in our situation, the matched points are the intersecting junctions of line segments, the distribution of which is often much sparser and their numbers are often smaller than the detected feature points by some detectors. The cases that several matched points cluster in a very small region are scarce. This fact ensures the reasonableness of our correctness-access strategy and the reliableness of the value we set for r.

	$r = 4$	$r = 6$	$r = 8$	$r = 10$	$r = 12$	$r = 14$
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 LJL structure matching results on the five image pairs shown in Figure 10 under various values of the parameter r.

5.2. Robustness of the LJL Descriptor

After the key parameter for constructing a 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 [1] was employed for the comparison. The two descriptors, LJL and SIFT, were used to describe the junctions in the constructed LJLs on the five image pairs shown in Figure 10. The junctions were matched under the same threshold for their description vector distances for both the two descriptors. Note that since both SIFT and LJL are based on fixed-size windows and are unable to deal with scale changes, we described the junctions in LJLs using the two descriptors both in the multi-scale image pyramid framework we proposed in Section 2.3.

Table 2 shows the accuracies 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 accuracies than SIFT. On some image pairs, *graffiti* and *boat*, the advantage is fairly remarkable: the results of LJL descriptor present the accuracy more than twice as that of SIFT. This good performance of our LJL descriptor on the matching accuracy is crucial for our method because it requires estimating a precise fundamental matrix from the initial LJL matches. 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 [36] because the proposed LJL descriptor is specially designed for LJLs. It describes the circular regions centered at the junctions in LJLs, rather than affine invariant regions detected by some detectors, which is the prerequisite of that famous local descriptor evaluation method.

The better performance of our proposed LJL descriptor 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 LJL descriptor are more likely to correspond with each other for cor-

	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 comparative junction point matching accuracies based on the description by the proposed LJL descriptor and SIFT descriptor on the five image pairs shown in Figure 10.

responding junctions 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 a junction can be regarded as the dominant direction of the junction, according to which the region exploited 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 in a LJL 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 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 description vector distances and hence contributes to better matching results.

5.3. Line Matching Results

Figure 12 shows the line matching results of the proposed method on eight representative image pairs which contain various image transformations and were captured from both planar and non-planar scenes. All these image pairs were used in the published papers [22, 37], except the image pair (d), in which a poorly-textured scene was captured in the two images. The aim we employed this image pair is to evaluate the performance of our method in poorly-textured scenes. The line segments used for matching were extracted by the famous line segment detector, LSD [35]. The correctness of the obtained matches was accessed by visual inspection.

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 the image pairs. The robustness of our method owes to the following factors. The first one is the robust LJL descriptor. It is specially designed for LJL while incorporates many benefits of SIFT, leading to its robustness and high effectiveness for matching LJLs. The second one is the adopted match-propagation strategy. Through the exploitation of the topological distribution consistency between LJL matches and their neighboring point matches and the recursive scheme of adding new matches while deleting possibly false ones, the group of LJL matches is

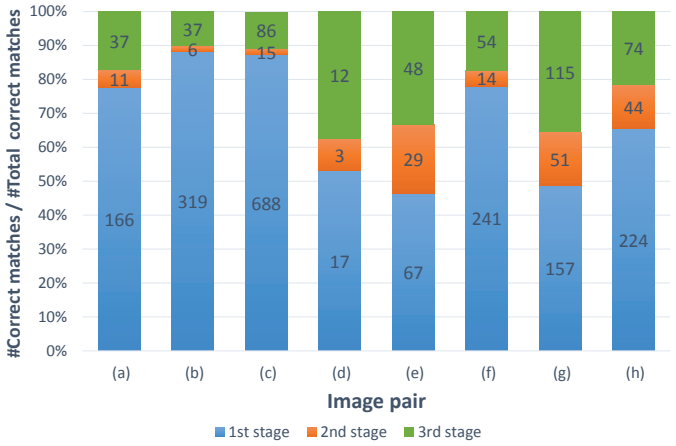
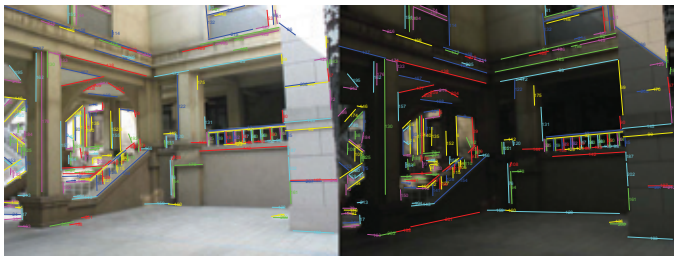


Figure 13: The incremental process of finding correct line matches in different stages of the proposed method. The number in each bin denotes the number of correct line matches found in the certain stage on the certain image pair.

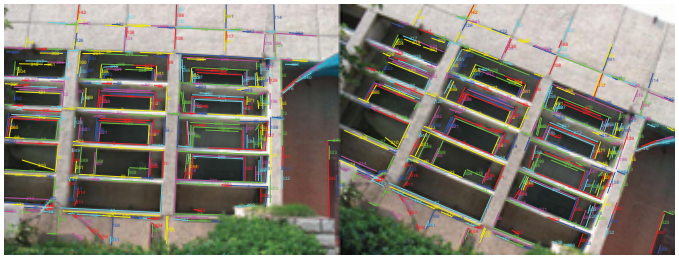
expanded while false matches are well limited. The third one is the novel approach of matching individual line segments by utilizing local homographies recovered from neighboring corresponding LJLs. This strategy is unaffected by most image transformations.

Figure 13 shows the incremental process of finding correct line matches in different stages of our method on the eight image pairs shown in Figure 12. It is observed from this figure that the proportions of the correct matches found in different stages accounting for the total correct matches vary greatly on different image pairs. For example, the correct matches found in the first stage account for nearly 90% 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 the image pairs, the correct matches found in the second stage account for a small part of the total correct matches. This is because most LJL matches were found correctly in the first stage, and there were only a few left to be found. However, the second stage still plays an important role in the algorithm, not only because some new LJL matches would be added, but also because the false matches introduced in the first stage are eliminated in this stage, which is significant for limiting false line matches.

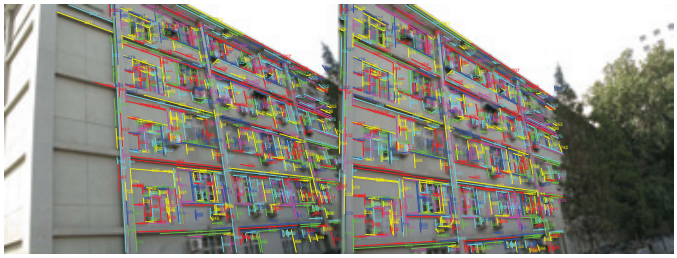
While counting the numbers of correct matches obtained in different stages of our method, we found that, on all the image pairs, almost all false matches among the final line matches are introduced in the third stage, in which line segments are matched in individuals by utilizing the local homographies estimated from their neighboring LJL matches. This is because the homography estimated from a pair of matched LJL is not so precise. If the line segments in a LJL match are not so precisely detected and located, the homography estimated from them would fluctuate around the precise one. When several line segments cluster in a small region of an image, this not so precise homography may lead to the method's incapability of picking out the best correspondence. However, since some additional constraints are applied when matching line segments in individuals, false matches are well controlled in our method.



(a) Illumination change. $(D_1, D_2) = (572, 275)$, $T = 217$, $F = 3$, $A = 97.5\%$.

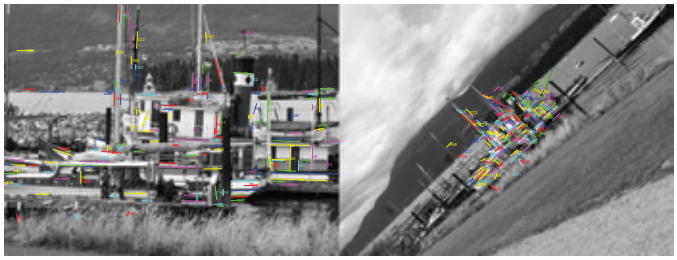


(b) Rotation change. $(D_1, D_2) = (537, 556)$, $T = 363$, $F = 1$, $A = 99.7\%$.

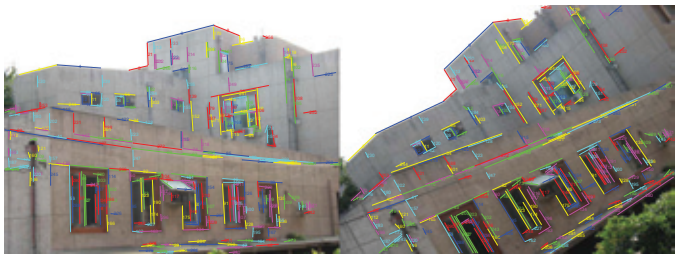


(c) Viewpoint change. $(D_1, D_2) = (1071, 1016)$, $T = 792$, $F = 3$, $A = 99.6\%$.

(d) Poorly-textured scene. $(D_1, D_2) = (102, 82)$, $T = 33$, $F = 1$, $A = 97.0\%$.



(e) Scale & rotation change. $(D_1, D_2) = (1334, 569)$, $T = 151$, $F = 7$, $A = 95.4\%$.



(f) Rotation change. $(D_1, D_2) = (526, 398)$, $T = 311$, $F = 2$, $A = 99.4\%$.

(g) Image blur. $(D_1, D_2) = (1700, 454)$, $T = 334$, $F = 11$, $A = 96.7\%$.

(h) JPEG compression. $(D_1, D_2) = (590, 1083)$, $T = 356$, $F = 14$, $A = 96.1\%$.

Figure 12: The line matching results of the proposed method on eight representative image pairs. We will refer the eight image pairs as (a)∼(h) for later use. In the caption of each subfigure, (D_1, D_2) denotes the pair of numbers of the extracted line segments in the two images. T represents the number of total matches found by the proposed method between the two sets of line segments extracted from the two images. F and A denote the number of false matches among the total matches and the corresponding accuracy, respectively. Two line segments in correspondence from a pair of images are drawn in the same color and are labeled with a same number at the middles.

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) [22]; the second one matches line segments in groups, Line Signature (LS) [31]; and the last one is our previous work Ray-Point-Ray (RPR) [32]. The implementations of LS and LPI were provided by their authors. To eliminate the 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. Note that the results of LPI on some image pairs shown in this table is somewhat different (generally better) from that shown our previous paper [32]. 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

	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%)
average	97.7%	97.3%	91.5%	82.3%	97.1%	85.3%

Table 3: The comparative line matching results of our method and three state-of-the-art line matching methods: LPI [22], LS [31] and RPR [32]. 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 the generated results.

Figure 14: Six image datasets characterized by various image transformations. There are six images with gradual image transformation in each dataset, and only the first and the last are shown here.

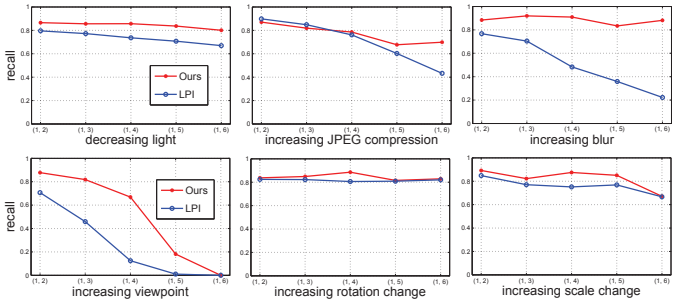


Figure 15: The recalls of the line matching results of the proposed method and LPI [22] on the six image datasets shown in Figure 14.

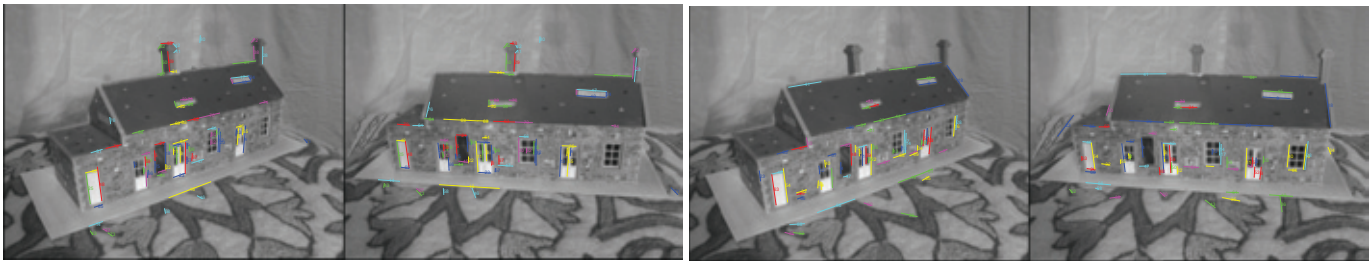
results of LPI became better, quite more correct matches and comparable accuracy, if we removed the requirement for the lengths of the line segments to be used for matching. So, this paper shows only the better results of LPI while our previous paper shows the results generated by the original implementation of LPI. Since the provided implementation of LPI also uses LSD [35] for extracting line segments, the results of our methods using line segments provided by LPI shown in the table are the same as those shown in Figure 12.

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, both in the numbers of correct matches and the corresponding accuracies. 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 the line segments provided by LPI. However, the corresponding result drops drastically with only 33 correct matches at the accuracy of only 49.3% when using the line segments provided by LS. The second is that when using the 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 the 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 much higher average accuracies and owned an overwhelming advantage in 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 through with a low accuracy. The fourth is that, by taking the line segments used in RPR as input, the proposed method excels both at the amount of correct matches and the accuracy. Remarkably, RPR failed on the image pair (g), where there is great blur between the two images, while the proposed method can still produce good results. It generated 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

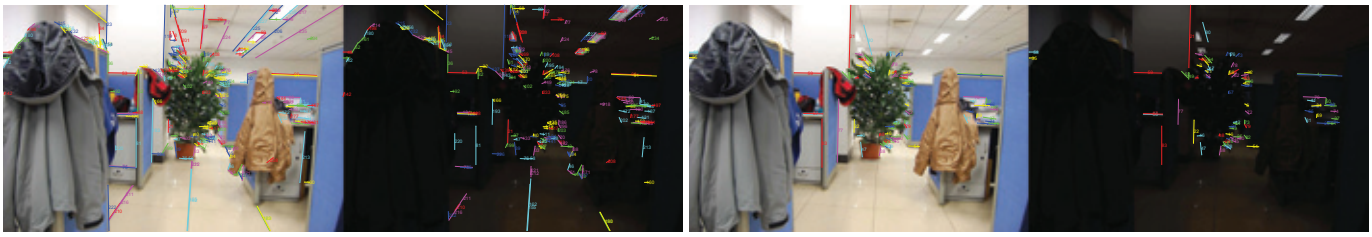
From Table 3, we can conclude that our method and LPI are the two most robust line matching methods. We conducted additional experiments to further evaluate the two methods.

(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 16: The comparative results between our method and LPI on some challenging image pairs. Please zoom in for better interpretation.

We first experimented on some datasets related by global homographies. The six image datasets [37, 17] shown in Figure 14 were employed. These image datasets are characterized by illumination, rotation, viewpoint and scale changes, image blur and JPEG compression among the images, respectively. The reason we employed them for experiments is because the global homographies between images in the datasets are known. Thus, the ground truth of the line segment matches between images can be established by mapping line segments detected in one image to another one 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 images, 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 line segments detected in the first image were matched with those detected in the other five images. The recalls of the matching results for the two methods are shown in Figure 15. 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 had con-

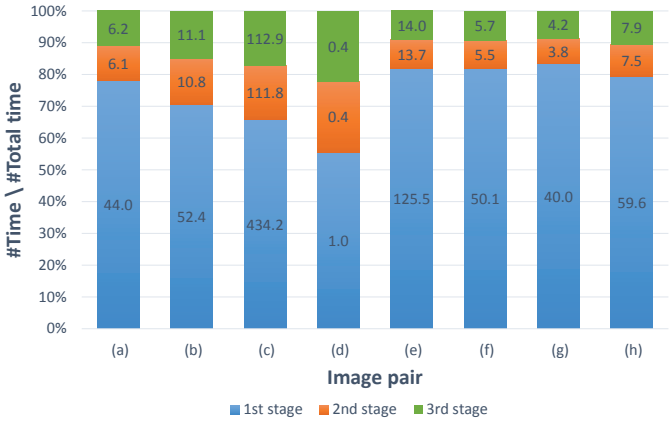


Figure 17: The elapsed time (in seconds) of each stage of our method and the percentage it accounts for the corresponding total elapsed time on each of the eight image pairs shown in Figure 12. The number in each bin denotes the elapsed time of the proposed method in a certain stage on a certain image pair.

ducted additional experiments on some very challenging image pairs. The experimental results are shown in Figure 16, 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 discuss further 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 17 shows the elapsed time of each stage of our method and the percentage it accounts for the corresponding total elapsed time on each of the eight image pairs shown in Figure 12. 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 is that by building image pyramids, each LJL constructed in the original images is adjusted 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,

	Matching Results		Running Times (s)		
	M-I	M-II	M-I	M-II	Drop
(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 and running time of the proposed algorithm generated by building Gaussian image pyramids (M-I) and without building Gaussian image pyramids (M-II) on some image pairs shown in Figure 12. The last column shows the drop ratios of the running time of M-II relative to M-I.

on the image pair (c), which took the 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 the two original images have 4 octaves with 2 scales in each octave, there are 4856×8=38848 LJLs and 4693×8=37544 LJLs required to be described for the two images, respectively. A LJL descriptor is a vector of 128 dimensions. It is sure that matching such large two groups of LJLs by evaluating the distances between their description vectors in such a high dimension is 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 be tremendously improved by adjusting some parameters for specific scenes.

The majority of the running time of our method was spent in describing and matching LJLs from two images. There are three parameters that control the number of the LJLs to be described and matched. These three parameters are w that controls the size of the affect region of a line segment, o, the number of octaves of the image pyramids, 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 comparative line matching results and the corresponding running time on the six image pairs, (a)–(d), (f) and (h) shown in Figure 12, with building the Gaussian image pyramids (M-I), and without building the Gaussian image pyramids (M-II). All these six image pairs share the similarity that there are very little scale changes between the two images and thus building Gaussian image pyramids is unnecessary for them. From Table 4, we can see that the matching results generated by M-II are similar with M-I both in the amounts of correct matches and the accuracy, but the running time dropped drastically. On the all image pairs, M-II took less than half of the running time of M-I. Remarkably, on the image pairs, (a)

and (h), the drop ratios are more than 80%, which means M-II used less than 20% of the running time of M-I.

Apart from choosing to not build image pyramids for images with very little scale change to save time, decreasing the value of the parameter w can also help 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. 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 the extracted line segments 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 rich texture, selecting a smaller value for w can greatly promote the efficiency of the method.

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. For example, on the image pair (d) shown in Figure 12, when we varied w from 10 to 60 at the step of 10, we obtained 28, 33, 36, 37, 37 and 36 correct matches in order. Though the increasing is not quantitatively significant, it is particularly meaningful for this special scene because a more complete sketch of the scene can be obtained with even slight increasing of the obtained line segment matches.

The reason for the better matching results of our proposed method on poorly-textured scenes with bigger w is as follows. In poorly-textured scenes, only a small amounts of line segments can be detected. 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 can be matched along with LJL. Second, a generally preciser fundamental matrix can be obtained, which helps both propagate LJL match (in the second stage) and match line segments in individuals (in the third stage). Third, the obtained 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 correspondences of the two line segments to be matched lie in the same 3D plane lay by the 3D correspondences of the two pairs of matched line segments brought by a pair of matched LJLs. If two individual line segments whose 3D correspondences lie in a 3D plane where none of the 3D correspondences of matched LJLs exists, the two line segments cannot be matched by our method. Thus, a larger group of initial LJL matches can help to bring in more individual line segment matches.

7. Conclusions

This paper has presented a hierarchical line matching method in which line segments are first matched along with the structures called Line-Junction-Line (LJL) formed by two adjacent line segments and their intersecting junction, and then matched in individuals. While matching LJLs, a robust descriptor as well as an effective strategy to deal with the possible scale changes between images are proposed to obtain the initial LJL 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 LJL matches. The experimental results show the robustness of the proposed LJL descriptor for matching LJLs 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 National Natural Science Foundation of China (Project No. 41571436), the Hubei Province Science and Technology Support Program, China (Project No. 2015BAA027), the National Basic Research Program of China (Project No. 2012CB719904), and the National Natural Science Foundation of China (Project No. 41271431). Thanks for Bin Fan, Lilian Zhang and Lu Wang for providing the implementations of their methods and some test images.

[1] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91–110, 2004.

[2] Z. Wang, B. Fan, F. Wu. Local intensity order pattern for feature description. In ICCV, 2011.

[3] 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.

[4] S. Winder, G. Hua, M. Brown. Picking the best daisy. In CVPR, 2009.

[5] Y. Ke, R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors In CVPR, 2004.

[6] H. Bay, T. Tuytelaars, G. L. Van. Surf: Speeded up robust features In ECCV, 2006.

[7] 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.

[8] 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.

[9] 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.

[10] M. Chandraker, J. Lim, D. Kriegman. Moving in stereo: Efficient structure and motion using lines. In *ICCV*, 2009.

[11] S. Ramalingam, M. Brand. Lifting 3D Manhattan lines from a single image. In *ICCV*, 2013.

[12] M. Hofer, M. Maurer, H. Bischof. Improving sparse 3D models for man-made environments using line-based 3D reconstruction. In *3DV*, 2014.

[13] H. Bay, A. Ess, A. Neubeck, L. Van Gool. 3D from line segments in two poorly-textured, uncalibrated images. In *3DPVT*, 2006.
[14] C. Schmid, A. Zisserman. Automatic line matching across views. In CVPR, 1997.
[15] 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.
[16] Z. Wang, F. Wu, Z. Hu. MSLD: A robust descriptor for line matching. Pattern Recognition, 42(5):941–953, 2009.
[17] 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.
[18] B. Verhagen, R. Timofte, L. G. Van. Scale-invariant line descriptors for wide baseline matching. In WACV, 2014.
[19] H. Bay, V. Ferrari, L. Van Gool. Wide-baseline stereo matching with line segments. In CVPR, 2005.
[20] 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.
[21] B. Fan, F. Wu, Z. Hu. Line matching leveraged by point correspondences. In CVPR, 2010.
[22] B. Fan, F. Wu, Z. Hu. Robust line matching through line-point invariants. Pattern Recognition, 45(2):794–805, 2012.
[23] M. Chen, Z. Shao. Robust affine-invariant line matching for high resolution remote sensing images. Photogrammetric Engineering & Remote Sensing, 79(8):753–760, 2013.
[24] B. Micusik, H. Wildenauer, J. Kosecka. Detection and matching of rectilinear structures. In CVPR, 2008.
[25] 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.
[26] 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.
[27] K. Mikolajczyk, C. Schmid. Scale and Affine invariant interest point detectors. International Journal of Computer Vision, 60(1):63-86, 2004.
[28] T.Tuytelaars, L. Van Gool. Matching widely separated views based on affine invariant regions. International Journal of Computer Vision, 59(1):61-85, 2004.
[29] J. Matas, O. Chum, M. Urban, T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In BMVC, 2004.
[30] T. Kadir, A. Zisserman, M. Brady. An affine invariant salient region detector. In ECCV, 2004.
[31] L. Wang, U. Neumann, S. You. Wide-baseline image matching using line signatures. In ICCV, 2009.
[32] 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.
[33] Q. -T. Luong, T. Viville. Canonical representations for the geometries of multiple projective views. Computer Vision and Image Understanding, 64(2):193–229, 1996.
[34] 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.
[35] 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.
[36] K. Mikolajczyk, C. Schmid. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 27(10): 1615–1630, 2005.
[37] 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.