Robust Line Matching Based on Ray-Point-Ray Structure Descriptor

Kai Li   Jian Yao[†]   Xiaohu Lu

School of Remote Sensing and Information Engineering
Wuhan University, Wuchang District, Wuhan, Hubei, P.R. China
[†]Email: jian.yao@whu.edu.cn

Abstract. In this paper, we propose a novel two-view line matching method through converting matching line segments extracted from two uncalibrated images to matching the introduced Ray-Point-Ray (RPR) structures. The method first recovers the partial connectivity of line segments through sufficiently exploiting the gradient map. To efficiently matching line segments, we introduce the Ray-Point-Ray (RPR) structure consisting of a joint point and two rays (line segments) connected to the point. Two sets of RPRs are constructed from the connected line segments extracted from two images. These RPRs are then described with the proposed SIFT-like descriptor for efficient initial matching to recover the fundamental matrix. Based on initial RPR matches and the recovered fundamental matrix, we propose a match propagation scheme consisting of two stages to refine and find more RPR matches. The first stage is to propagate matches among those initially formed RPRs, while the second stage is to propagate matches among newly formed RPRs constructed by intersecting unmatched line segments with those matched ones. In both stages, candidate matches are evaluated by comprehensively considering their descriptors, the epipolar line constraint, and the topological consistency with neighbor point matches. Experimental results demonstrate the good performance of the proposed method as well as its superiority to the state-of-the-art methods.

1   Introduction

Image matching is an indispensable procedure in almost all applications which require recovering 3D scene structure from 2D images such as 3D reconstruction, scene interpretation, robotic navigation, structure from motion, etc. A lot of objects in real scenes can be outlined easily by line segments. So, recovering 3D scene structure from line matches has advantages over that from point matches. In some cases, for example, the scenes are poorly-textured, recovering their 3D structures from line matches seems the only choice because point matches are often insufficient in this kind of scenes. Despite that recovering 3D scene structure from line segment matches seems a better choice than that from point matches, both the instability of endpoints of line segments and the lost of connectivity of line segments complicate the matching of line segments.

Line matching methods can be roughly divided into two categories: methods that match individual line segments and methods that match a group of line segments. Some methods matching individual line segments take advantages of the photometric information associated with the line segments, such as intensity [1, 2], color [3], or/and gradient [4, 5] in the regions around the line segments. All of these methods are based on the assumption that there are considerable overlaps between corresponding line segments. But if two line segments in correspondence don't share sufficient corresponding part, it is hardly possible to match them correctly. Moreover, these methods tend to produce false matches in regions with repeated textures because of the lack of variation in the photometric information. Another group of methods matching individual line segments incorporate point matches into line matching [6–9]. These methods first find a large group of point matches using existing point matching methods [10, 11], and then exploit invariants between coplanar points and line(s) under certain image transformations. Line segments which meet the invariants are regarded to be in correspondence. All these methods share the same disadvantage that they tend to fail when scenes captured are poorly-textured since there are often not sufficient point matches to be found in these scenes.

Matching group of line segments is more complex, but more constraints are available for disambiguation. Most of these methods [13, 14, 16] first use some strategies to intersect line segments to form junctions and then utilize features associated with the junctions for line matching. In [18, 19], line segments are not intersected to form junctions, but 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. Their method is robust in some very challenging cases. But the dependence on approximately corresponding relationship between the endpoints of line matches leads to the tendency of the method to produce false matches when substantial disparity exists in the locations of the endpoints.

Our proposed line matching method exploits features of junctions too, but in a quite different way. At the place of the junctions, we form a Ray-Point-Ray (RPR) structure, consisting of a junction point and two rays (line segments) connected to the point. RPRs are described with a robust SIFT-like descriptor. Through exploiting photometric and geometric constraints associated with RPR matches as well as their topological relationship with neighbor point matches, we propagate RPR matches in an iterative scheme. Experimental results show our method generates more correct line matches with higher accuracy than the state-of-the-art methods in most cases.

2   Partial Line-Connectivity Recovery

Line segments extracted by the existing line segment detectors [20, 21] are often separated with each other since the segmentation procedure. We refine them by partially recovering their connectivity through exploiting the gradient map of the original image. Given a line segment $\mathbf{l}_1$, we search its neighbors using two circles centered at its two endpoints with the same radius, d. Other line segments fall
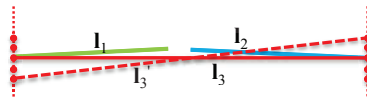
Fig. 1: An illustration of merging two line segments and adjusting the endpoints of the merged line segment to maximize the gradient magnitude of the line segment.

inside these two circles are neighbors of l_1. For a searched neighbor line segment l_2, if the orientation difference between l_1 and l_2 is less than a given threshold α (α was set to 5° in this paper), we tentatively merge the two line segments to a long one and adjust the endpoints of the new line segment to maximize its Gradient Magnitudes (GM), which is the mean of gradient magnitudes of all pixels in the line segment. The direction of a line segment is the direction of the vector from one endpoint of the line segment to another one. The way of adjusting the endpoints of a merged line segment to maximize its GM is illustrated in Fig. 1 where l_1 and l_2 are two line segments to be merged, l_3 is the merged line segment, l_3' is an example of adjusting the endpoints of l_3 to find the line segment with maximal GM, and the red dots are pixels in the lines orthogonal with l_3 and passing through its endpoints. 5 pixels for each endpoints is used, generating 25 candidate line segments by linking any pixel in one side to all pixels in the other side. The one with maximal GM is selected as the merged line segment of l_1 and l_2. If the GM of the merged line segment is greater than 80% of the sum of that of l_1 and l_2, we accept it and use it to replace l_1 and l_2 for further steps. If the direction difference between l_1 and l_2 is above α, we intersect them and generate a junction. If the distance between the junction and the endpoint of l_1 is less than d, we accept the junction and extend l_1 to the junction. After all line segments being processed by above steps, we obtain a new set of line segments in which some of which are connected with each other. The steps above are conducted iteratively until no more line segment could be merged or extended. Fig. 2 is a demonstration of our line refinement method on a real scene. Comparison between the second image and the third one shows that some line segments are extended to be longer while some line segments are merged with others.
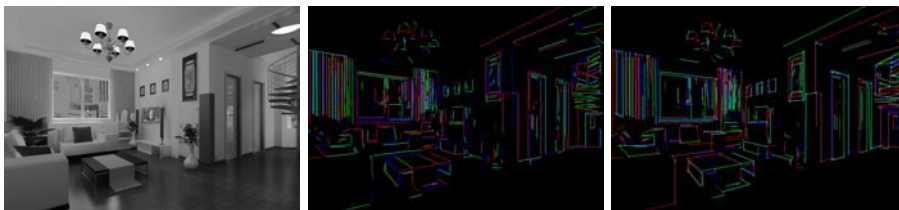

Fig. 2: An example of the method of recovering the partial connectivity of line segments on a real scene: (Left) the original image; (Middle) the line segments detected by EDLines [20]; (Right) the refined line segments by our method. The small circles in the right image denote the added junctions.

3 Line Matching

After recovering the partial connectivity of line segments, some line segments are connected with others from which we construct two sets of initial RPRs from two images respectively. Through describing and matching these RPRs, we find a set of initial RPR matches, which are the basis for the propagation of RPR match in an iterative scheme.

3.1 RPR Construction

Line segments may connect with each other in four forms as shown in Fig. 3. The numbers of RPRs to be constructed are different in different forms. The principle of forming RPRs from connected line segments is that any two line segments which connect with a common joint point but are not in the same line can be used to form a RPR. Under this principle, 1, 2 and 4 RPRs can be constructed in Figs. 3(a)-(c), respectively. In Fig. 3(d), the number of RPRs to be constructed is dependent on the number of line segments connected to the intersection and their configuration.

3.2 RPR Descriptor

A RPR consists of a point and two rays (line segments) connected with the point. Such relationship between the point and the two rays is stable under image transformations. Inspired by SIFT [10], we use the directions of both rays as dominant directions of the point and generate orientation histograms. Entries in both groups of orientation histograms are concatenated to get the descriptor of the RPR.

The way of generating orientation histograms using one of the rays of a RPR is illustrated in Fig. 4, in which the direction of ray **OA** is currently used as dominant direction. The region is a rectangle centered at the point with the width along the ray of 8 pixels and the height of 16 pixels. The region is evenly split into 4 subregions along the height, generating 4 rectangles with the same size of 8×4. In each subregion, an orientation histogram containing 8 bins is constructed. Thus, using one ray as dominant direction results in a vector of 8×4=32 dimensions, and the final descriptor of a RPR is a vector of 64 dimensions. For each pixel in the rectangle, its gradient magnitude is weighted and added to corresponding entry of certain histogram according to its gradient
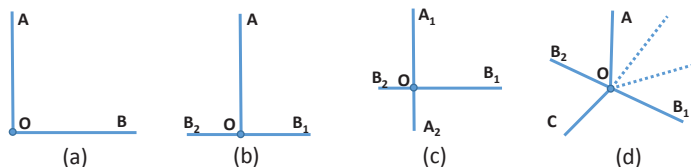

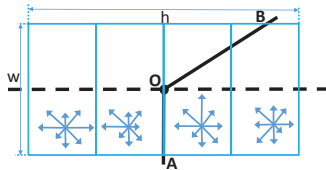Fig. 3: Four forms of line segments connecting with each other.

Fig. 4: An illustration of generating orientation histogram using one ray of a RPR.

orientation. The weighted gradient magnitude of a pixel is calculated as:

$$G(x, y) = |g(x, y)| \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\Delta y)^2}{2\sigma^2}\right),$$ (1)

where $g(x, y)$ denotes the gradient magnitude of the pixel located at (x, y), Δy is the distance of the pixel to the ray, σ is a parameter determining the weighting function, and $G(x, y)$ is the weighted gradient magnitude.

We utilize these ways to construct histograms base on the following two reasons. The first reason is to promote efficiency of the descriptor while keeping its distinctiveness. The choices of the height and width of the rectangle derive from SIFT descriptor in which the region forming an orientation histogram is a 4×4 square, while ours is a 8×4 rectangle. We do not split the region into subregions along the ray like SIFT because the RPR descriptor is only 64 dimensions in our way, rather than 256 dimensions when using the strategy of SIFT. It is certainly more efficient. Besides, in most cases, the numbers of RPRs to be matched are far less than the numbers of detected keypoints of SIFT. A descriptor with lower dimension is distinctive enough for matching. The second one is to avoid the possible shift of the point along the ray. The point in each RPR results from intersecting two lines in neighbor. Its position may shift slightly along the ray since the positions of the two lines forming the point may vary slightly under image transformations. If we split the region along the ray and use the point as the basis for assigning weights, the histograms of regions in correspondence would differ with each other mistakenly.

Two sets of RPR descriptors would be obtained after describing all RPRs constructed from two images. The general way of matching these descriptors is to like the SIFT matching procedure by computing their Euclidean distances and selecting pairs of descriptors with the smallest distances as matches. But since each RPR consists of a point and two rays connected with the point, there is additional information available for disambiguation. The two rays in each RPR locate in a local region. The difference between their directions should vary at a small range under most image transformations. For a test pair of RPRs, if θ_1 and θ_2 denote the direction differences of the two rays of the pair of RPRs, then, $|\theta_1 - \theta_2|$ should be a small value if the RPR pair is a correct match. This constraint can be used to discard many false candidates before evaluating their descriptor distances and thus contributes to better matching results.

3.3   Global Image Scale Change Estimation

Our strategy for global image scale change estimation is based on the fact that two images in the (approximately) same scale produce the most putative R-PR matches. We first build Gaussian pyramids for original reference and query images. At the same time, line segments extracted from original images are accordingly adjusted to fit the new images. In this paper, the pyramids have 4 octaves with 4 layers in each octave. After that, the original reference image is to match all images in the pyramid built for the query image with their RPRs. The same procedure is applied to the original query image and all images in the pyramid built for the reference image. The pair of images producing the most putative RPR matches is regarded to be in the same scale and will be used in further steps.

3.4   RPR Match Propagation

Point matches obtained along with RPR matches can be used for the estimation of the fundamental matrix. To achieve a stable and precise fundamental matrix, we first use RANSAC [22] to estimate an initial fundamental matrix and refine it by the Normalized 8-point Method [23] and Levenberg-Marquardts optimization [24] in order. After that, we obtain the fundamental matrix and the corresponding group of RPR matches, from which we commence propagating RPR matches. The RPR match propagation is achieved by progressively increasing the threshold for the distance of an accepted point match according to the fundamental matrix, which is defined as follows:

$$d\left(\mathbf{x}_i, \mathbf{x}_i'\right) = \left(\mathbf{x}_i'^{\top}\mathbf{F}\mathbf{x}_i\right)^2 \times \left(\frac{1}{(\mathbf{F}\mathbf{x}_i)_1^2 + (\mathbf{F}\mathbf{x}_i)_2^2} + \frac{1}{(\mathbf{F}^{\top}\mathbf{x}_i')_1^2 + (\mathbf{F}^{\top}\mathbf{x}_i')_2^2}\right), \quad (2)$$

where $\mathbf{x}_i$ and $\mathbf{x}_i'$ represent the i-th candidate corresponding points in the reference image and query image respectively, and $(\mathbf{F}\mathbf{x})_k^2$ denotes the square of the k-th entry of the vector $\mathbf{F}\mathbf{x}$. The distance will later be referred in a simplified manner, as Distance according to Fundamental Matrix (DFM). RPR pairs whose DFMs are smaller will be matched first and then serve as the basis for the next iteration to introduce new RPR matches. Unmatched RPRs are first grouped according to matched RPRs and then matched in corresponding groups. Point matches are introduced to filter false matches while guide the process of adding new matches.

Point Match Expanding There are 3 pairs of points in each RPR match: one pair of junctions and two pairs of endpoints of the two pairs of rays. The pair of junctions are matched along with the RPR match. Point match expanding process aims only at the two pairs of endpoints. For a test pair of endpoints, we check if they can be regarded as a match based on the following criteria. First, the distance of their descriptors should be less than a given threshold η ($\eta = 0.5$ in this paper). We describe the pair of endpoints using the similar way as that

describing RPRs. But since only one ray is connected with each endpoint, their descriptors are therefore only 32 dimensions by using the direction of one ray as the dominant direction. Second, the DFM of the test endpoint pair should be less than a pre-defined threshold (18 was used in this paper).

RPR Match Filtering False RPR matches with the two points locating near the corresponding epipolar lines may have been accepted by the fundamental matrix mistakenly. These false matches can be eliminated by exploiting the topological consistency between RPR matches and their neighbor point matches.

For a RPR match $(\mathbf{M}_1, \mathbf{M}_2)$, we first find out a certain number of the nearest matched points to the points of $\mathbf{M}_1$ and $\mathbf{M}_2$. The number is set as 8 to make a balance between efficiency and in-sensitiveness with noises among the point match group. After that, we obtain two sets of neighbor matched points, $\tilde{\mathcal{M}}_1$ and $\tilde{\mathcal{M}}_2$ for $\mathbf{M}_1$ and $\mathbf{M}_2$ respectively. If $(\mathbf{M}_1, \mathbf{M}_2)$ is a correct match, elements in $\tilde{\mathcal{M}}_1$ and $\tilde{\mathcal{M}}_2$ should meet the following two conditions. First, a proportion of elements in $\tilde{\mathcal{M}}_1$ and $\tilde{\mathcal{M}}_2$ should be correspondences. The proportion was set as 0.5 in this paper. Second, the distribution of elements in $\tilde{\mathcal{M}}_1$ and $\tilde{\mathcal{M}}_2$ and their correspondences according to $\mathbf{M}_1$ and $\mathbf{M}_2$ should also be consistent with each other. Refer to Fig. 5, $\mathbf{AOB}$ and $\mathbf{A'O'B'}$ are a pair of RPRs in correspondence from two images. The other RPRs in yellow are matched RPRs nearby. Matched points generated by RPR matches are represented by yellow dots, which can be the junctions or the endpoints of the matched rays. The junction and the two rays in addition to their reverse extensions in each matched RPR form a coordinate-like structure. Matched points distribute in different quadrants of the coordinate. This kind of distribution is invariant under projective transformation if the matched points are coplanar with the matched RPR in 3D space. So, a proportion of elements in $\tilde{\mathcal{M}}_1$ and $\tilde{\mathcal{M}}_2$ and their correspondences should lie in the same quadrants of the two coordinates formed by $\mathbf{M}_1$ and $\mathbf{M}_2$. The proportion was set as 0.8 empirically in this paper. The kind of topological consistency between RPR matches and their neighboring point matches will be referred in a simplified manner, as Point Distribution Consistency (PDC) for subsequent use.

RPR Match Propagation RPR match propagation consists of two main steps: unmatched RPRs grouping and unmatched RPRs matching. For each
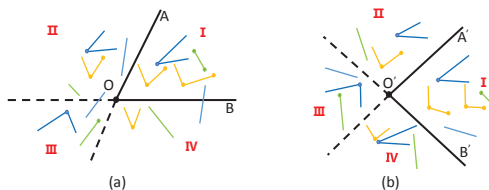

Fig. 5: An illustration of a pair of RPRs in correspondence and the distribution of their neighboring line segments and RPRs. $\mathbf{AOB}$ and $\mathbf{A'O'B'}$ are a pair of RPRs in correspondence from two images. The other matched RPRs are marked in yellow. Matched single line segments are marked in green. Unmatched single line segments and RPRs are marked in blue. The yellow and green dots are matched points generated by RPR matches and single line matches respectively.

unmatched RPR, we find 3 of its nearest matched RPRs whose points are the nearest 3 points to the point of the unmatched RPR among all points in matched RPRs. The unmatched RPR is then put into the 3 groups. The reason that each unmatched RPR is redundantly put into multiple groups is to ensure potential matches will be distributed into at least one pair of groups in correspondence. After that, there are one matched RPR and some unmatched RPRs in each group. As shown in Fig. 5, unmatched RPRs, marked in blue, are distributed in the four quadrants of the coordinates formed by the matched RPR. These unmatched RPRs are divided into 4 groups according to the quadrants their points belong.

After grouping unmatched RPRs in two images, the matching process is performed in each pair of groups in correspondence. For each test RPR pair, it will be accepted as a candidate match if it meets all these requirements. First, the distance of their descriptors is less than a given threshold η_1 (η_1 equals 0.5 in the paper). Second, the DFM of the pair of points is less than η_2. The value of η_2 changes in every iteration. Third, the two RPRs meet the PDC.

Iterative Scheme Some new RPR matches has been generated in previous step. The point match group can be expanded again from the newly obtained RPR matches. Under the new point match group, some false RPR matches may be filtered out by rechecking their PDC since their neighbor matched points may have varied. After that, it is necessary to update the point match group and line match group since some RPR matches may have been deleted.

Up to now, an iteration of RPR match propagation has been fulfilled. Before increasing the value of η_2 and starting a new iteration, we need to conduct the same iteration again without changing η_2 to add those RPR matches neglected in the previous iteration. In the previous iteration, several RPRs in one image may be matched to the same RPR in another image as their best correspondence. We select the pair whose descriptor distance is the smallest and reject others. The left RPRs may find their correct correspondences under the new pair of groups of unmatched RPRs. After that, we increase the value of η_2 and begin a new iteration.

The way of tuning η_2 at different iterations determines the number of RPR matches to be obtained and the times of iterations. So, it is necessary to explain it separately. The same strategy is employed in the propagation of RPR match among single line segments discussed in next section. DFM for a pair of point, as defined in Eq. (2), is actually the quadratic sum of the distances of a pair of points to corresponding epipolar lines. For a pair of points, $\mathbf{P}_1$ and $\mathbf{P}_2$, their DFM can be represented as $\sigma = d_1^2 + d_2^2$ where d_1 denotes the distance from $\mathbf{P}_2$ to the epipolar line determined by $\mathbf{P}_1$ and d_2 denotes the distance from $\mathbf{P}_1$ to the epipolar line determined by $\mathbf{P}_2$. Point matches in our case are not in precise correspondence, but in an approximate manner. Their DFM is a relative great value. At the initial iteration, we accept two points as a match if η_2 is less than α. Then, we progressively increase η_2 at a certain step until it reaches the upper limit β. In this paper, α was set as 8 while β equals 18 and the step is 5 for two adjacent iterations.

3.5   Single Line Segment Matching

RPR matches among initially formed RPRs are mostly found after propagation. Each RPR match brings two line matches. Besides these line matches, there exist a considerable amount of single line segments which do not connect with other line segments but can find their correspondences in the other image. To find these line matches, we first group single line segments based on matched RPRs and then match them in corresponding groups.

Single Line Segment Grouping For each single line segment, we find 3 nearest RPRs whose points are the nearest 3 points to the midpoint of line segment among all points in matched RPRs. The line segment is then distributed into the corresponding 3 groups. After that, in each group, there are a matched RPR and some unmatched single line segments. Refer to Fig. 5, unmatched single line segments, marked in blue, distribute in different quadrants of the coordinates formed by the matched RPR. These unmatched line segments are then divided into four groups according to the quadrants their endpoints belong to. For each line segment, if any of its two endpoints lies in a certain quadrant, the line segment is put into the corresponding group. Under this grouping strategy, each line segment may be put into several groups. Despite, in most cases, this may lead to multiple evaluations of the same pair of line segments, it is still necessary to do so to ensure potential line matches will are distributed into at least one pair of groups in correspondence.

Single Line Segment Matching The matching process is conducted in each pair of groups in correspondence. For such a group pair $(\mathcal{G}_1, \mathcal{G}_2)$, there are a RPR match denoted as $(\mathbf{M}_1, \mathbf{M}_2)$ and two sets of single line segments. Each single line segment in $\mathcal{G}_1$ will be evaluated with all single line segments in $\mathcal{G}_2$. For a test pair $(\mathbf{l}_1, \mathbf{l}_2)$, we first check whether the direction difference of $\mathbf{l}_1$ and $\mathbf{l}_2$ is consistent with the direction differences of the two line segment matches generated by $(\mathbf{M}_1, \mathbf{M}_2)$. Correct line segment matches in local region share similar direction differences under image transformations. So we calculate the mean value of the direction differences of the two pairs of matched rays, denoted as σ_1. Suggest the direction difference between $\mathbf{l}_1$ and $\mathbf{l}_2$ is σ_2. If $|\sigma_2 - \sigma_1| < \epsilon$, where ϵ is a user-defined threshold set as $20°$ in this paper, we accept the test pair temporarily and take it for further evaluation.

Then, we use the epipolar constraint to check the approximate correspondence of the endpoints of $\mathbf{l}_1$ and $\mathbf{l}_2$. It is unreasonable to count on the correspondence of the endpoints of corresponding line segments to match them. However, we notice almost all line segment matches own at least one pair of endpoints which approximately correspond with each other. Thus, if neither the DFMs of the two pairs of endpoints of $\mathbf{l}_1$ and $\mathbf{l}_2$ is below a relatively small value (100 is used in the paper), we regard $(\mathbf{l}_1, \mathbf{l}_2)$ as a false match.

As mentioned before, the fundamental matrix is incapable to be used to discard false corresponding points that approach the corresponding epipolar lines. So some pairs of line segments may have passed previous constraint mistakenly. We employ topological consistency between line segment matches and their

neighbor point matches to avoid this problem. For $\mathbf{l}_1$ and $\mathbf{l}_2$, we find n (set as 8 in this paper) of the nearest matched points to the midpoints of the two line segments, generating two point sets, $\mathcal{S}_1$ and $\mathcal{S}_2$. If $(\mathbf{l}_1, \mathbf{l}_2)$ is a correct match, $\mathcal{S}_1$ and $\mathcal{S}_2$ should meet the following two conditions. First, a proportion of elements in $\mathcal{S}_1$ and $\mathcal{S}_2$ should be correspondences. The proportion was set as 0.5 in this paper. Second, a certain ratio, set as 0.8 in this paper, of elements in $\mathcal{S}_1$ and $\mathcal{S}_2$ and their correspondences should lie on the same sides of $\mathbf{l}_1$ and $\mathbf{l}_2$. The side of a line segment is defined as clockwise or anticlockwise direction relative to the line segment based on the endpoint whose x-coordinate is smaller.

If $(\mathbf{l}_1, \mathbf{l}_2)$ has passed all tests above, we use the two line segments to intersect with the two pairs of rays of $(\mathbf{M}_1, \mathbf{M}_2)$ to form new RPRs. Through evaluating the newly formed RPR pairs, we determine whether $(\mathbf{l}_1, \mathbf{l}_2)$ can be accepted or not. The way of matching the newly formed RPR pairs is same as that presented in the step of RPR match propagation. If any of the two pairs of newly formed RPRs is finally accepted as a RPR match. We regard $(\mathbf{l}_1, \mathbf{l}_2)$ as a candidate match.

After matching single line segments in all corresponding groups, we obtain a set of new line segment matches and the corresponding set of RPR matches. Among these RPR matches, there may exist several RPRs in one image match with the same RPR in another image. We select the best one with the smallest descriptor distance.

From the new line segment matches and RPR matches, we can expand the point match group again using the same strategy as that presented in Section 3.4. False RPR matches used to be accepted can possibly be discerned by rechecking the PDC under the expanded point match group. Some false RPR matches may have been removed after previous step. The line match group and the point match group should accordingly be updated again.

Up to now, an iteration of propagating RPR matches among single line segments has been finished. Before increasing the threshold for DFM and conducting a new iteration, the iteration without changing the threshold should be conducted again to pick up matches neglected in previous iteration. After that, we increase the value of the threshold and begin a new iteration to get more line segment matches. The iteration will halt if the threshold reaches the upper limit. The way of tuning the threshold at different iterations is the same as that presented in Section 3.4.

4   Experimental Results

Experiments on a set of representative image pairs were conducted to substantiate the robustness of the method and to prove its superiority by comparing with the state-of-the-art line matching methods.

Line Matching Results Fig. 6 shows the line matching results on some representative image pairs, denoted as $\mathcal{P}_1$–$\mathcal{P}_8$ in order. Two line segments in correspondence are drawn in the same color with the same label. The statistical results shown below each sub-figure in Fig. 6 is a triple consisting of the number
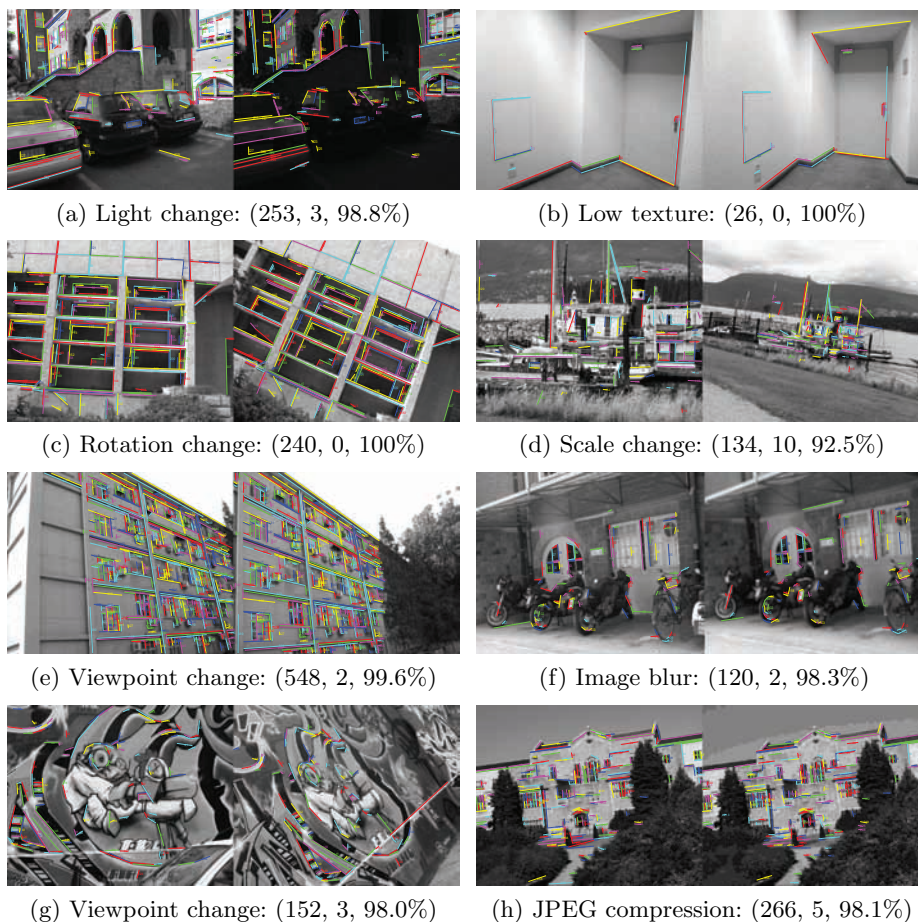
(a) Light change: (253, 3, 98.8%)

(b) Low texture: (26, 0, 100%)

(c) Rotation change: (240, 0, 100%)

(d) Scale change: (134, 10, 92.5%)

(e) Viewpoint change: (548, 2, 99.6%)

(f) Image blur: (120, 2, 98.3%)

(g) Viewpoint change: (152, 3, 98.0%)

(h) JPEG compression: (266, 5, 98.1%)

Fig. 6: Results of the proposed line matching method on some representative image pairs. The eight image pairs will be represented as $\mathcal{P}_1$–$\mathcal{P}_8$ in order for subsequent use. See the text for details.

of total matches, the number of false matches and the accuracy in order. It is noticed that our algorithm is robust under common image transformations, namely light, scale, rotation, viewpoint changes, image blur, JPEG compression, and in poorly-textured scene. The accuracy is above 98% on all image pairs except $\mathcal{P}_4$, where there exists great scale change between the two images. The reason that the accuracy of our method on image pairs with great scale change is relative lower is that after being adjusted to the same scale, line segments lying closely in the original image become so adjacent with each other that it is very hard to pick out the correct one.

Evaluation of the RPR Descriptor The RPR descriptor describes the local regions formed by a point and two rays connected with the point. The relationship between the point and the two rays is fully exploited, which results in

the descriptor is more robust and efficient than other famous local region descriptors which directly describe the local regions centered at the points. We use the same way presented in [25] to evaluate our descriptor and SIFT descriptor [10]. We describe initially formed RPRs in image pairs $\mathcal{P}_1$–$\mathcal{P}_8$ by using our proposed RPRdescriptor and the SIFT descriptor. The RPRs are matched by evaluating their descriptor distances under the same threshold. For each image pair, we count the number of correct matches, the number of total matches, and the 1-precision, which is the ratio between the number of false matches and the number of total matches. The comparative results are shown in Table 1. Note that any pair of descriptors whose distance below the given threshold is regarded as a match. So, one RPR in one image may match with several RPRs in another image and all these matches are included into the total matches. This is why the 1-precisions of the matching results on some image pairs are so high. From the table, it can be noticed that on some image pairs, our RPR descriptor generates more correct matches, while the SIFT descriptor generates more correct matches on the others. But on all image pair, the 1-precisions of our descriptor are lower than that of the SIFT descriptor, which indicates our RPR descriptor is better than the SIFT descriptor for the specific local regions.

Comparison with Single Line Based Methods Two methods are selected for comparison. They are Lines-Points Invariants (LPI) [8] and Line Signature (LS) [19]. Their implementations are provided by their own authors. Except experimenting on image pairs $\mathcal{P}_1$–$\mathcal{P}_8$ using the three methods (LPI, LS and RPR), we do another two groups of experiments on the same image pairs to eliminate the influence of different line detection methods on the line matching results. The first is that we take line segments used in LPI, detected by LSD [21], as input for our method. The second is that we use our line segments, detected by EDLines [20], as input for LPI. All comparative results are listed in Table 2. From the table it can be concluded that no matter using line segments detected by EDLines or LSD as input, our proposed RPR method generates much better results than that of LPI. Our method can find quite more line matches with higher accuracy in almost all cases. When comparing with LS, our method produces comparable number of matches but with higher accuracy on the image

Table 1: Comparative results of our RPR descriptor and SIFT descriptor on image pairs $\mathcal{P}_1$–$\mathcal{P}_8$ by describing the local regions formed by RPRs. The triple elements shown in the table represent the number of the correct matches, the number of total matches, and the corresponding 1-precision.

	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$
RPR	(116, 163, 28.8%)	(12, 16, 25.0%)	(241, 330, 27.0%)	(53, 77, 31.2%)
SIFT	(113, 217, 47.9%)	(16, 26, 38.5%)	(170, 271, 37.3%)	(72, 134, 46.3%)
	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_7$	$\mathcal{P}_8$
RPR	(460, 601, 23.5%)	(17, 132, 87.1%)	(25, 110, 77.3%)	(23, 34, 32.4%)
SIFT	(325, 493, 34.1%)	(10, 259, 96.1%)	(41, 424, 90.3%)	(47, 95, 50.5%)

Table 2: Comparative results of the proposed RPR algorithm, LPI [8] and LS [19]. The columns from the left to right are the results of our RPR method using line segments detected by EDLines [20], our RPR method based on line segments detected by LSD [21], LPI using line segments detected by LSD, LPI based on line segments detected by EDLines, and LS. The triple elements shown in the table represent the number of line matches, the number of false matches, and the accuracy respectively. The last row represents the average accuracy of each method.

	RPR (EDLines)	RPR (LSD)	LPI (LSD)	LPI (EDLines)	LS
$\mathcal{P}_1$	(253, 3, 98.8%)	(295, 5, 98.3%)	(219, 2, 99.1%)	(185, 9, 95.1%)	(248, 5, 98.0%)
$\mathcal{P}_2$	(26, 0, 100%)	(30, 0, 100%)	(12, 0, 100%)	(15, 0, 100%)	(53, 12, 77.4%)
$\mathcal{P}_3$	(240, 0, 100%)	(219, 0, 100%)	(227, 2, 99.1%)	(235, 0, 100%)	(242, 1, 99.6%)
$\mathcal{P}_4$	(134, 10, 92.5%)	(153, 5, 96.7%)	(78, 3, 96.2%)	(92, 8, 91.3%)	(40, 3, 92.5%)
$\mathcal{P}_5$	(548, 2, 99.6%)	(582, 3, 99.5%)	(390, 3, 99.2%)	(356, 2, 99.4%)	(271, 2, 99.3%
$\mathcal{P}_6$	(120, 2, 98.3%)	(287, 3, 99.0%)	(116, 5, 95.7%)	(142, 4, 97.2%)	(81, 2, 97.5%)
$\mathcal{P}_7$	(152, 3, 98.0%)	(123, 1, 99.2%)	(90, 11, 87.8%)	(107, 8, 92.5%)	(139, 3, 97.8%)
$\mathcal{P}_8$	(266, 5, 98.1%)	(292, 3, 99.0%)	(148, 5, 96.6%)	(193, 7, 96.4%)	(205, 4, 98.0%)
	98.2%	99.0%	96.7%	96.5%	95.0%

pairs $\mathcal{P}_1$–$\mathcal{P}_3$. On the image pairs $\mathcal{P}_4$–$\mathcal{P}_8$, the accuracy of the two methods is similar, but our method produces more correct matches.

Comparison with Junction Based Methods Our method exploits features of junctions for matching line segments. So comparisons with junction-based line matching methods are necessary to comprehensively evaluate our method. The method introduced in [16], represented as LICF, is used for the comparison. We cannot obtain the implementation of the method, so the comparison is conducted by evaluating the different performances of the two methods on same image pairs. Fig. 7 shows three image pairs used in the published paper. The corresponding comparative results are shown in Table 3. Comparing the results of the two methods shows that our method produces fairly more matches with similar accuracy, which means quite more correct matches are obtained by our method.

Comparative Performances under Image Transformations. Compared with other line matching methods, one remarkable merit of our method is that it produces more correct matches with high accuracy under various image trans-

Fig. 7: Three pairs of images used to compare our method with the method presented in [16]. From left to right, the three image pairs will be represented as "apt", "tcorner" and "valbonne" in order for later use.

Table 3: Comparative results of our RPR method and LICF [16] on some image pairs. Refer to Table 2 for details.

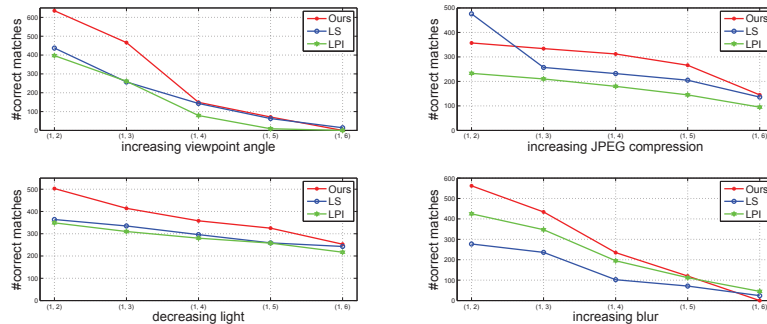|	apt	tcorner	valbonne
RPR	(190, 8, 95.8%)	(122, 3, 97.5%)	(123, 5, 95.9%)
LICF	(53, 3, 94.3%)	(70, 8, 88.6%)	(34, 1, 97.1%)

Fig. 8: The results of our proposed RPR method, LS [19], and LPI [8] on image datasets with various image transformations.

formations. Fig. 8 shows the results of our method, LPI [8], and LS [19] on the famous image datasets, "graffiti", "ubc", "leuven", and "bikes" [1], in which viewpoint change, JPEG compression, light change and image blur between images exist respectively. Each dataset contains 6 images and the image transformations from the second to the last image relative to the first image is stronger and stronger. Image pairs $\mathcal{P}_7$, $\mathcal{P}_8$, $\mathcal{P}_1$, and $\mathcal{P}_6$ are token from the four datasets respectively. Fig. 8 shows the matching results between the first and the second to sixth image in each dataset. It is obvious that our method produces quite more correct matches under these image transformations in most cases.

5   Conclusions

We propose a novel line matching method through converting matching line segments to the newly introduced Ray-Point-Ray (RPR) structures. A SIFT-like robust descriptor is proposed to match RPRs under an efficient iterative scheme by progressively adding new matches while deleting false matches. Experimental results demonstrate the robustness of the method and its superiority to the state-of-the-art methods for the larger group of correct matches and the higher accuracy in most cases.

---

[1] http://lear.inrialpes.fr/people/mikolajczyk/Database/index.html

References

1. Schmid, C., Zisserman, A.: Automatic line matching across views. CVPR (1997)
2. Baillard, C., Schmid, C., Zisserman, A., Fitzgibbon, A.: Automatic line matching and 3D reconstruction of buildings from multiple views. ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery (1999)
3. Bay, H., Ferrari, V., Van Gool, L.: Wide-baseline stereo matching with line segments. CVPR (2005)
4. Wang, Z., Wu, F., Hu, Z.: MSLD: A robust descriptor for line matching. Pattern Recognition **42** (2009) 941–953
5. Zhang, L., Koch, R.: Line matching using appearance similarities and geometric constraints. Springer Berlin Heidelberg (2012)
6. Lourakis, M.I., Halkidis, S.T., Orphanoudakis, S.C.: Matching disparate views of planar surfaces using projective invariants. Image and Vision Computing **18** (2000) 673–683
7. Fan, B., Wu, F., Hu, Z.: Line matching leveraged by point correspondences. CVPR (2010)
8. Fan, B., Wu, F., Hu, Z.: Robust line matching through line-point invariants. Pattern Recognition **45** (2012) 794–805
9. Chen, M., Shao, Z.: Robust affine-invariant line matching for high resolution remote sensing images. Photogrammetric Engineering & Remote Sensing **79** (2013) 753–760
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
11. Winder, S., Hua, G., Brown, M.: Picking the best daisy. CVPR (2009)
12. Vincent, E., Laganiere, R.: Junction matching and fundamental matrix recovery in widely separated views. BMVC (2004)
13. Bay, H., Ess, A., Neubeck, A., Van Gool, L.: 3D from line segments in two poorly-textured, uncalibrated images. 3DPVT (2006)
14. Micusik, B., Wildenauer, H., Kosecka, J.: Detection and matching of rectilinear structures. CVPR (2008)
15. Kim, H., Lee, S.: A novel line matching method based on intersection context. ICRA (2010)
16. Kim, H., Lee, S.: Simultaneous line matching and epipolar geometry estimation based on the intersection context of coplanar line pairs. Pattern Recognition Letters **33** (2012) 1349–1363
17. Kim, H., Lee, S., Lee, Y.: 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** (2014) 421–435
18. Wang, L., Neumann, U., You, S.: Wide-baseline image matching using line signatures. ICCV (2009)
19. Wang, L., Adviser-Neumann, U.: Line segment matching and its applications in 3D urban modeling. PhD thesis, University of Southern California (2010)
20. Akinlar, C., Topal, C.: EDLines: A real-time line segment detector with a false detection control. Pattern Recognition Letters **32** (2011) 1633–1642
21. Von Gioi R G, Jakubowicz J, M.J.M.e.a.: von Gioi, R. G., Jakubowicz, J., Morel, J.-M., Randall, G.: LSD: A fast line segment detector with a false detection control. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 722–732
22. Zuliani, M.: RANSAC for dummies (2009). With examples using the RANSAC toolbox for Matlab and more (2009)

23. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press (2003)
24. Gavin, H.P.: The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems. Technical report, Department of Civil and Environmental Engineering, Duke University (2013)
25. Mikolajczyk K., Schmid C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1615–1630