Line Matching Across Views Based on Multiple View Stereo

FU Kang-Ping[1] SHEN Shu-Han[1] HU Zhan-Yi[1]

Abstract A graph-based multiple view line matching method is proposed based on results of multiple view stereo (MVS) algorithms. With the 3D points and their visibility information provided by MVS, point-line correspondences are firstly established through 3D-to-2D re-projection. Each image line detected in different views is described using a 3D point set as well as a unit vector representing its coarse 3D direction. From such a description, pairwise similarity and consistency are evaluated. Then, a graph is constructed to contain all image lines as nodes. To get a unified node distance measure, a spectral graph analysis method is employed. Finally, a modified DBSCAN algorithm is introduced to obtain reliable line matches from the graph. Experiments show that our method is more robust and exhibits better accuracy than the existing methods.

Key words Multi-view line matching, multiple view stereo (MVS), feature matching, 3D point clouds

Citation Fu Kang-Ping, Shen Shu-Han, Hu Zhan-Yi. Line matching across views based on multiple view stereo. *Acta Automatica Sinica*, 2014, **40**(8): 1680−1689

DOI 10.3724/SP.J.1004.2014.01680

Feature matching is a fundamental task in computer vision, especially in image-based modeling. In the past decade, point matching has been well studied and developed. Progress in point matching has made it possible for structure-from-motion (SfM) and multiple view stereo (MVS) methods[1−3] to reconstruct quite acceptable 3D point clouds from large-scale photo collections.

Meanwhile, line matching is also needed in many computer vision applications, such as 3D reconstruction and robot navigation. For example, [4] described an architecture reconstruction method which uses line matches to search for planes and polygons. And [5] showed an application of line matches in robot localization.

Compared to point matching, line matching faces more challenges and is relatively less successful[6]. In recent years, only a few influential line matching methods have been proposed and proved effective[6−15]. Furthermore, among all the line matching methods, few are explicitly targeted at line matching across multiple views while most of them focus on two-view scenarios. But for applications like 3D line and plane reconstruction, it is an essential step to match image lines among three or more views. Conceptually speaking, line matching methods for two views can be extended to multiple views, but the weak distinctiveness of line features and accumulation of errors may greatly lower the matching accuracy as the number of views grows.

In this paper, we propose a graph-based method to efficiently solve the multiple view line matching problem by fully exploiting both photometric and global geometric information provided by MVS. We choose to match image lines based on MVS for two reasons: 1) MVS has been well developed and the currently existing MVS algorithms are able to provide satisfactory results; 2) for some applications that require line matching across multiple views, like image-based modeling, recovering 3D points and camera parameters using SfM/MVS methods is usually a routine step.

Manuscript received September 9, 2013; accepted January 3, 2014
Supported by National High Technology Research and Development Program of China (863 Program) (2013AA12A202) and National Natural Science Foundation of China (61227804, 61105032)
Recommended by Associate Editor JIA Yun-De
1. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

1 Related work and our approach

Most existing line matching methods focus on two-view scenarios. In this section, we firstly give an overview on these two-view line matching methods. Then, challenges faced by such two-view methods are illustrated and explanations are given on why it is difficult to extend them to multiple view scenarios. Finally, existing multiple view line matching methods are summarized and analyzed.

The two-view line matching methods differ mainly in the way image lines are described. In the methods of [7−8], only appearance attributes of pixels in the neighborhood of image lines are considered. In [7], lines are described using color histograms of pixels in their neighborhoods, which means this method is not adequate to handle imaging changes, like illumination changes. Wang et al.[8] proposed a descriptor called MSLD to describe pixels in an image line′s neighborhood. This descriptor is built by computing the mean and standard deviation of histograms on neighboring pixels. It exhibits good robustness to many kinds of imaging condition changes except for scale change, as shown in [6]. Most recently, López et al. proposed a line detection and matching method for photogrammetry in [9]. In this method, extracted line segments are matched based on their local appearance and photometric relationship. Local appearance similarity is estimated by computing the gray level intensity distribution. And photometric relationship is built based on line orientation, length and location of endpoints. Experiments show that this method could acquire high matching accuracy in close-range industrial photogrammetry applications. However, it cannot be applied to image sets with view-point changes.

Besides appearance attributes, relative position of neighboring image lines could provide additional local geometric constraints for line matching. Wang et al.[10] proposed a line descriptor called line signature. A line signature describes an image line by encoding geometric properties of its neighboring lines, like angles and length ratios. However, as shown in [6], erroneous end points of detected lines may reduce the matching accuracy.

To improve the matching robustness, some methods[6, 11] try to combine both appearance and local geometric information. Fan et al.[6] proposed a method that matches lines using point correspondences. Keypoints are firstly extracted and matched between two views. Then, lines are matched based on a geometric line-point invariant. Since point matching is a well-studied problem, it exhibits good robustness and accuracy. Most recently, Zhang et al.[11] proposed a new line descriptor called LBD, which encodes both local appearance information and local geometric constraints. The geometric constraints are established for line pairs. Reported experiments show that their method outperforms other methods in accuracy.

All the aforementioned line matching methods use only local attributes — photometric, geometric, or both. No global geometric information is considered, since it is usually unknown for the scenario of two-view line matching. However, for multi-view line matching, global geometric constraints are sometimes crucial. Methods without taking global geometric information into account will be less robust, or may even fail. Fig. 1 gives an example. When matching repetitive line patterns across different views, which is very common in man-made scenes, additional geometric information will largely improve the robustness. Without such information, line matching may be inaccurate due to the weak distinctiveness of line features. Meanwhile, for most multi-view applications, it is usually not difficult to obtain reliable global geometric information with the help of existing SfM or other methods. Thus, for multi-view line matching, it is necessary to combine both local attributes and global geometric information.

Fig. 1 The two wide-baseline views contain repetitive patterns, but they have no overlapping areas (Since no global geometric information is considered, two-view line matching methods tend to find erroneous matches between the two views.)

References [12] and [13] described two similar multiple view line matching methods. In [4], Werner et al. extended the two methods and released an open-source toolkit called LMatch[16]. In their method, starting from two views, image lines are traversed and matched based on normalized cross-correlation scores. To reduce time complexity, a pairwise epipolar geometric constraint is enforced as a pruning strategy. Two-view match is later extended to other views using similar criteria. This method exhibits good accuracy on multi-view data sets of small scales. However, one problem with this method is that the computational cost is high, since all views have to be traversed to build a single match. In the implementation of this method, line matches are first generated from two views with nearest camera centers in order to get a better initial two-view match. However, nearer camera centers will not guarantee more overlapping areas in the two views. In addition, the computed cross-correlation score which is used to evaluate image line similarity is not reliable when the scale changes dramatically.

The method proposed in [14] uses uncertain projective geometry for 3D line matching and reconstruction. A 3D line is firstly estimated using two arbitrary image lines from different views. Then, the 3D line is re-projected to other views in order to include consistent image lines and extend the match. Consistency is evaluated by geometric attributes, like pairwise collinearity. Since uncertainty is represented in geometric computations like plane intersection, the method exhibits good robustness. However, since photometric similarity is not taken into consideration, the result could be very vulnerable to occlusion. Besides, since initial 3D lines are estimated from arbitrary image line pairs, the computational cost would be very high for large-scale applications.

The work described in [15] computed both geometric and photometric matching costs for each image line match hypothesis. Specifically, the geometric matching cost is evaluated by the residual vectors derived from the iterative reconstruction of 3D lines from 2D line matches. For the photometric cost, a fixed-size window is placed along each image line and the intensity values of pixels in the window are statistically analyzed. All possible matching hypotheses are generated and evaluated using the two cost functions. The match set with the minimum matching cost is then chosen and 3D lines are computed. This method was proposed for aerial image sets where images are ordered and overlap between adjacent images is guaranteed. However, for unordered wide-baseline image sets, the described photometric cost is not suitable to scale changes due to the fixed window size, nor the computational cost is acceptable due to its big search space.

1.1 Our approach

In this paper, we propose a method that obtains reliable image line matches across multiple views based on the results of MVS.

3D points generated by MVS are firstly re-projected to their visible images to obtain correspondences between 3D points and 2D lines. 3D points are reconstructed from feature points which contain rich photometric information. Meanwhile, re-projection from 3D to 2D could geometrically relate feature points in different views. Thus, both photometric information and global geometric constraints could be expressed within such correspondences. In addition, line match hypotheses can also be derived from the obtained correspondences, which largely reduces the computational cost brought by the traverse of image lines across views needed in traditional line matching methods.

3D-to-2D re-projection is a common method for error evaluation[17] and consistency check. In [18], the authors re-projected 3D points obtained using SfM algorithms to images to detect occlusion for facade modeling. Chen et al.[19] proposed a 3D line extraction method from point clouds generated by MVS based on 3D-to-2D re-projection. However, the point-line correspondences obtained from 3D-to-2D re-projection are extremely unreliable due to 3D-to-2D ambiguities and inevitable errors. Any image lines that are closely located could be related by such correspon-

dences. To further analyze these correspondences and get reliable line matches, we propose a graph-based method.

Pairwise similarity of two image lines could be defined based on the obtained correspondences. A graph is constructed with all image lines being the nodes, and image line pairs with non-zero similarity scores are connected in the graph. Then, a spectral graph analysis method is employed. In this method, a Markov-chain random walk model is used to provide a unified distance measure for nodes. Under such a measure, principal components of the graph are analyzed and a dimensionally reduced Euclidean distance measure for the nodes is presented. Based on the obtained Euclidean distance measure, we cluster nodes into groups. To enforce known pairwise constraints, a modified clustering algorithm DBSCAN[20] is introduced. After clustering, image line matches are finally obtained.

The rest of this paper is organized as follows. Section 2 are some preliminaries. Details of the graph construction, graph analysis, and node clustering are given in Section 3. Experiments are reported and analyzed in Section 4 before we conclude this paper in Section 5.

2 Preliminaries

In this paper, the 3D point set reconstructed by MVS is denoted by $X = \{X_k\}$, $1 \le k \le K$. We denote by $I = \{I_i\}$, $1 \le i \le N$, the set of N images and by $P = \{P_i\}$, $1 \le i \le N$, the set of N corresponding camera matrices for the N views, respectively.

Image lines detected for view i form set L^i_{2D} $(1 \le i \le N)$, and its elements are denoted by l^i_m $(1 \le i \le N, 1 \le m \le |L^i_{2D}|)$, which means the mth line segment in view i. All image lines form a global image line set L_{2D} as follows.

$$L_{2D} = \bigcup_{1 \le i \le N} L^i_{2D} \tag{1}$$

The goal of this work is to obtain an image line match set of $M = \{(l^i_m, l^j_n, \cdots, l^k_o), 1 \le i, j, k \le N\}$. The rth element in set M is denoted by M_r. We require that $|M_r| \ge 2$, since it takes at least two image lines to constitute a line match.

Note that, unless otherwise stated, the lines we talk about in this paper refer to lines with finite lengths, also known as line segments.

3 Image line description and graph-based matching

3.1 Image line description

In this method, an image line is described using a 3D point set and a 3D unit vector. All 3D points in the set fall on the image line after re-projection, and are considered to be potential points lying on the 3D line that corresponds to the image line. Considering inevitable errors in MVS, we define a neighborhood for each l^i_m. All re-projected points falling in the neighborhood are considered as having fallen on l^i_m. The neighborhood N^i_m of l^i_m is a small region centered along l^i_m with a width of T. Each pixel in the image can be labeled either with a line′s ID (a positive integer) to whose neighborhood it belongs, or with 0 if it belongs to none of the line neighborhoods. For a region where two or more neighborhoods overlap, boundaries are set up to ensure each pixel in this region only belongs to the nearest line′s neighborhood. Note that pixels that are equally near to two or more lines are labeled 0. An illustration is given in Fig. 2.

							2	2	②	2	2	
							2	2	②	2	2	
							2	2	②	2	2	
1	1	1	1	1	1	1		2	②	2	2	
1	1	1	1	1	1	1	1		②	2	2	
①	①	①	①	①	①	①	①		②	2	2	
1	1	1	1	1	1	1	1		②	2	2	
1	1	1	1	1	1	1		2	②	2	2	
							2	2	②	2	2	
							2	2	②	2	2	

Fig. 2 Pixels labeled with image line IDs (The circled numbers stand for the original image line pixels and the plain numbers stand for the neighborhood pixels. All pixels labeled 0 are left empty for simplicity, and the neighborhood width is 5.)

For each image line, all 3D points that fall into its neighborhood in the re-projection form a point set to describe this line as follows.

$$D(l^i_m) = \{X_k | P_i X_k \in N^i_m, 1 \le k \le K\} \tag{2}$$

In many line matching methods, such as [4] and [6], image lines are oriented based on the intensity gradient. For example, the start point and the end point of an image line could be defined so that the brighter side of this image line is always the right side. In our method, we extend the orientation to 3D and compute a 3D unit vector for each image line. The two re-projected 3D points that are closest to end points are selected, and a normalized 3D orientation vector $E(l^i_m)$ is computed. The computation of $E(l^i_m)$ is illustrated in Fig. 3.

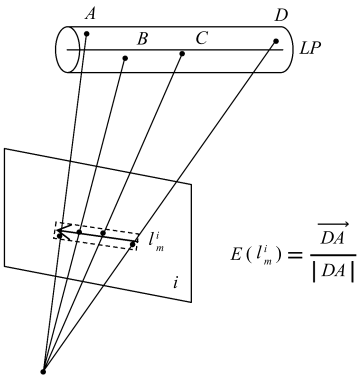

Fig. 3 Computation of $E(l^i_m)$ for oriented image line l^i_m

3.2 Graph construction and its structure

The similarity score between two image line segments is defined as follows.

$$Sim(l_a^i, l_b^j) = \begin{cases} \dfrac{|D(l_a^i) \cap D(l_b^j)|}{\max(|D(l_a^i)|, |D(l_b^j)|)}, & E(l_a^i) \cdot E(l_b^j) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

In the definition, orientation consistency is evaluated based on the 3D orientation vectors, and similarity between two consistent image lines is evaluated based on the overlap ratio of their corresponding 3D point sets. Two image lines are considered inconsistent in their orientations only if the angle between their orientation vectors is bigger than 90°. This is a loose requirement since the orientation vectors are estimated using the 3D point set which may contain erroneous elements.

Note that $Sim(l_a^i, l_b^i) = 0$ holds, because neighborhoods of lines from the same image never overlap based on the definition.

With pairwise similarity scores, we build a graph where all image lines with non-empty description point sets act as nodes. Edges are added for the node pairs whose corresponding image lines have non-zero similarity scores, and the similarity scores serve as edge weights. In the constructed graph, potential similar nodes appear in the same connected components. Before further analyzing the graph, we divide the graph into connected components and process one connected component at a time.

Image lines in the same connected component can be considered as a tentative match. However, such a match is not accurate enough. For one thing, two nodes may be directly connected by an edge with a very small, though non-zero, edge weight. It is possible that such a small edge weight (similarity score) is introduced by noise. And, more importantly, the relation of similarity used here is intransitive. That is, line i and line j both are similar to line k does not necessarily mean line i and line j are similar to each other. This is also the reason why matching accuracy is not guaranteed when we directly extend two-view matching to more views. In our case, it is not safe to assume those nodes that are not directly connected in a connected component to be within the same match.

Fig. 4 shows the structure of a connected component and its corresponding image lines. The first row shows the graph structure and a sample image of the scene. In the graph structure image, a node is labeled by the view ID and line ID, while an edge is labeled using its weight. The bottom two rows show image lines represented by nodes of this connected component. We can see that these image lines can be divided into two image line matches, as marked

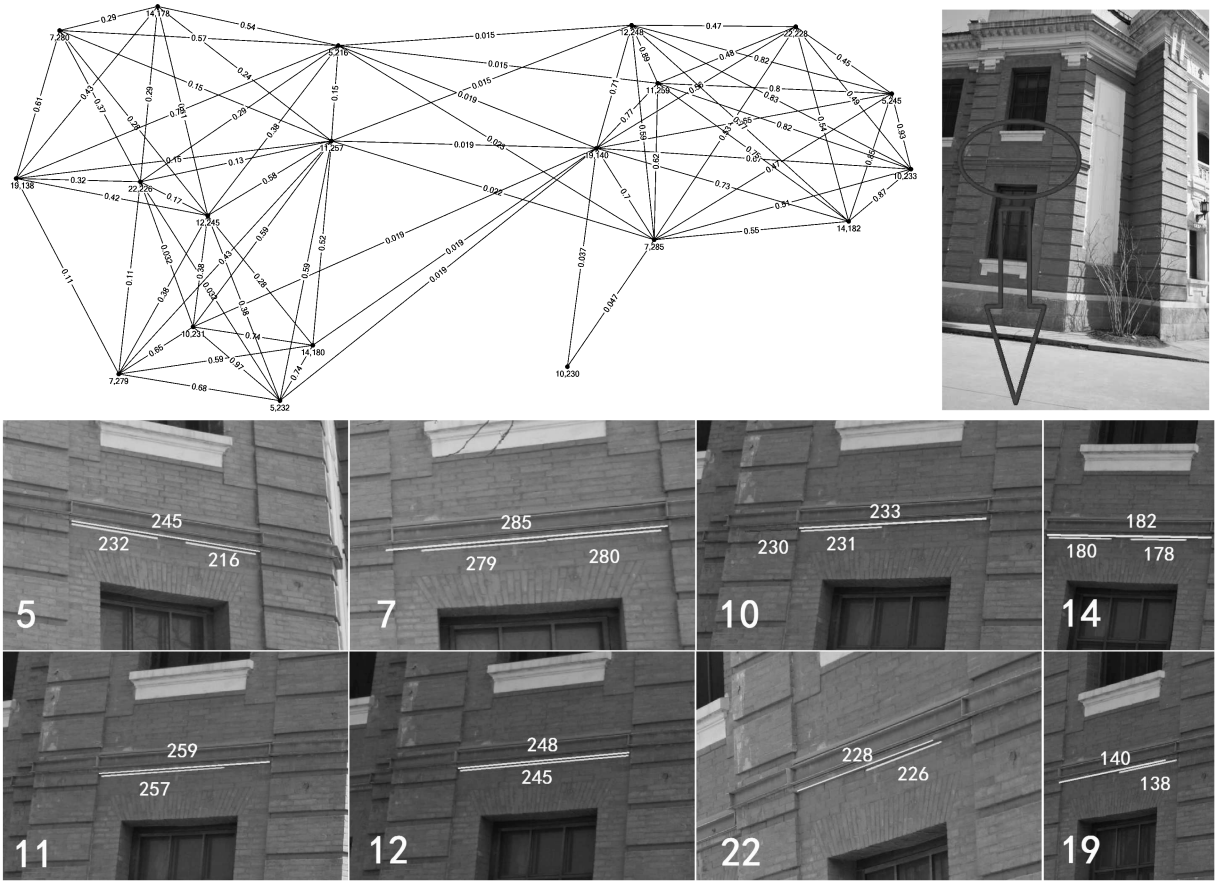

Fig. 4 Example of a connected component of the match graph (The top row shows graph structure and a sample image of the scene. Vertices are labeled with (View ID, Line ID), and edges are labeled with their weights. The bottom two rows show the image lines contained in this connected component. The lines are labeled with their line IDs, and the labeling numbers at the bottom-left corners in the image are their view IDs.)

in the figure. And there is an error image line (Line 230 in View 10) that belongs to neither of the two matches. Note that, based on the matches observed from images, the edges connecting nodes inside the two matches have relatively big edge weights while the weights of edges connecting nodes from different matches are very small. Meanwhile, the node corresponding to the noise has only two edges whose weights are less than 0.1. This makes sense because neighboring image lines can be easily connected in the re-projection procedure due to various factors, such as inaccurate 3D points and camera parameters, ambiguities of 3D-to-2D projection, and inaccurate image line localization.

In summary, further analysis of the graph structure is needed to remove weakly connected nodes and divide the node groups connected by only weak edges into different clusters.

3.3 Graph principal components analysis

In order to cluster nodes into groups, we need to evaluate distances between nodes under a unified distance measure. To achieve this, we employ the method proposed in [21] to analyze the principal components of the graph and compute a dimensionally reduced Euclidean distance between each node pair.

Without loss of generality, we denote by G the current connected component with n nodes. The elements in the adjacency matrix A of G are defined as $a_{pq} = Sim(l_a^i, l_b^j)$, where nodes p and q correspond to lines l_a^i and l_b^j, respectively. The Laplacian matrix L of G is defined as $L = D - A$, where $D = \mathrm{diag}\{a_p\}$ is the degree matrix and $d_{pp} = [D]_{pp} = a_p = \sum_{q=1}^{n} a_{pq}$. The Moore-Penrose pseudo-inverse of L is denoted by L^+. Since L^+ is a positive semi-definite matrix, L^+ can be transformed to a diagonal matrix Λ through $\Lambda = U^{\mathrm{T}} L^+ U$, where U is an orthonormal matrix made of eigenvectors of L^+.

$$P(s(t+1) = q | s(t) = p) = \frac{a_{pq}}{d_{pp}} = p_{pq} \tag{4}$$

Saerens et al.[21] proposed a node distance measure called the Euclidean commute time distance (ECTD), which is based on a Markov-chain random walk model. In the model, the transition probability of a random walker jumping from node p to its adjacent node q is defined in (4), where $s(t) = p$ means the random walker is in state p (node p) at time t.

$$n(p,q) = V_G (e_p - e_q)^{\mathrm{T}} L^+ (e_p - e_q) \tag{5}$$

The average commute time $n(p,q)$ is defined as the average number of steps for a random walker to start from node p, enter node q, and go back to node p for the first time. It is proved that $n(p,q)$ can be computed from (5), where V_G, defined as $V_G = \sum_{p=1}^{n} d_{pp}$, is the volume of graph G, and e_p is the pth column of the $n \times n$ identity matrix I. Since L^+ is positive semi-definite, the ECTD, defined as the square root of $n(p,q)$, is thus a node distance measure in the Euclidean space.

It is also proved that the transformation by (6) transforms the ECTD to a new n-dimensional Euclidean space where the dimensionality can be easily reduced.

$$X' = U\Lambda^{\frac{1}{2}} \tag{6}$$

Here, X' denotes the data matrix containing the coordinates of the nodes in the transformed space. According to [21], the kth principal component of X' yields the kth eigenvector of Λ.

In our method, we choose the projection of X' on the first two principal components for easier node clustering and data visualization. The left image in Fig. 5 shows the first two principal components′ projection of the graph in Fig. 4. It is consistent with our conclusion in Subsection 3.2 that the entire graph can be clustered into three groups: two of them have many nodes while the third one has only one node.

3.4 Constrained node clustering

In this step, we cluster all nodes into groups under the ECTD measure, ensuring that distances of nodes within each group are as small as possible while distances between different groups are as big as possible. We choose the DBSCAN[20] algorithm which is a density-based spatial clustering method. In this method, noisy data are clustered

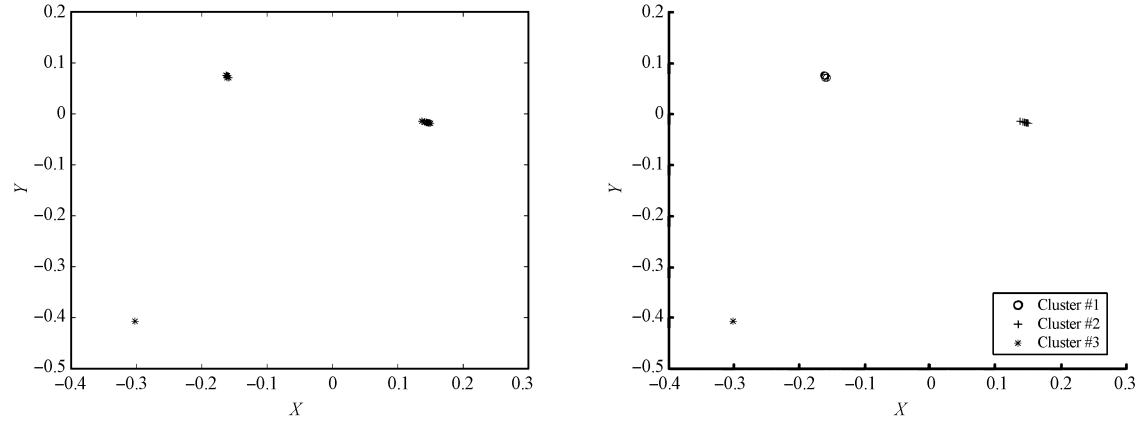

Fig. 5 The graph-PCA results projected to the 2D space (left) and the clustering results (right) (In the clustering results, the points in Cluster #1 and Cluster #2 are inliers of different labels. Whereas the one in Cluster #3 is treated as a noise point, since one line segments do not form a valid match.)

based on the density distribution. It is able to successfully cluster data into groups without knowing the cluster number beforehand.

Note that two closely located image lines from the same image could easily be included in the same G, as shown in the example. In our matching method, we allow two or more lines that are from the same image to be matched together, like the two collinear lines in View 5 of Fig. 4. This makes sense because in natural scenes, line segments detected in different views can easily be broken into pieces due to illumination changes or occlusion. However, for a situation like this, an additional constraint, called collinearity constraint, is enforced to improve the accuracy. We call two image lines from the same image conflicting with each other if they are not collinear. For example, Lines #232 and #216 in View 5 of Fig. 4 are not conflicting, while they both conflict with Line #245.

In the implementation of DBSCAN in [22], an expansion diameter r_e is estimated based on the point density. Starting from each unvisited point, the algorithm finds all its neighboring points within the distance of r_e and includes them in the same cluster. Then, the cluster is expanded recursively for all its elements under the same r_e. And a point is labeled as a noise point if it has too few neighboring points.

To take the collinearity constraint into account, we modified the DBSCAN algorithm. Each time we expand the current cluster from one node, the diameter dynamically shrinks from r_e until it includes neither nodes that conflict with any existing elements in the current cluster, nor nodes that conflict with each other.

The right image in Fig. 5 shows the clustering result of the constrained-DBSCAN. The nodes are clustered into three group as marked. Clusters #1 and #2 both contain several nodes while Cluster #3 contains only one node. Since at least two image lines are needed to form a valid line match, Cluster #3 is treated as a noise group.

The corresponding image lines of the clustered nodes are drawn in the bottom row of Fig. 4.

3.5 Graph-based image line matching

Matches are established with the above clustering results. As to the example in Fig. 4, these image lines form two line matches and the noise line is marked discarded. Note that the match model we are using allows multiple image lines from the same view in one match (e.g. Lines #232 and #216 in View 5 of Fig. 4). Such a matching model could deal with broken lines across views and give cues for further image line merging.

The entire line matching algorithm is shown in Algorithm 1.

Algorithm 1. Matching algorithm
Input:
I, P and X from MVS;
L_{2D};
Output:
Matching set M;
1. Generate line-ID-labeled images for elements in I;
2. Re-project points in X to their visible views and build line descriptor set D;
3. Construct global graph G_{global} using L_{2D} and D;
4. Divide G_{global} into connected components $\{G_i\}$;
5. $M = \emptyset$;
6. **For** each G_i in $\{G_i\}$ **do**
7. Compute pairwise conflicts ($Conf$) for each node;
8. $Comps = \text{PCA}(G_i)$;
9. $NdLbls = \text{Constrained-DBSCAN}(Comps, Conf)$;
10. **For** each valid node cluster in $NdLbls$ **do**
11. $M_r = \{$image lines corresponding to all nodes in current node cluster$\}$;
12. $M = M \cup \{M_r\}$;
13. **Endfor**
14. **Endfor**
15. **Return** M.

4 Experimental results and analysis

In image-based modeling, the open-source tools Bundler[1] and PMVS[2] could be considered as the most influential SfM and MVS algorithms. Bundler can efficiently compute camera parameters from images, which are used as inputs to PMVS which in turn generates quasi-dense point clouds. Here, we choose the Bundler + PMVS pipeline for all our experiments.

Experiments were carried out upon two different scenes: the Tsinghua School and the Longquan Temple, both of which are available on-line[1]. The Tsinghua School image set shows a single building viewed from different viewpoints. The Longquan Temple image set shows a more complex scene, which consists of several building with occlusion. 3D point clouds of both scenes are shown in Fig. 6, which are obtained by the Bundler + PMVS pipeline.

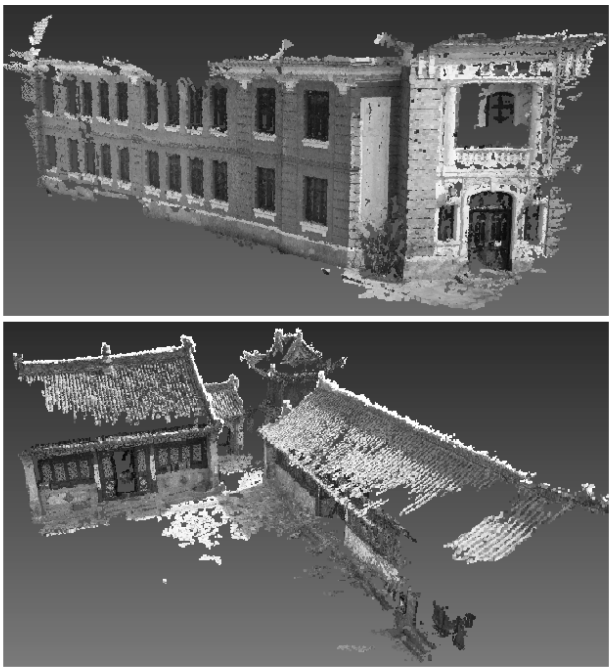

Fig. 6 3D point clouds of the tested scenes: the Tsinghua School (upper) and the Longquan Temple (lower)

[1] http://vision.ia.ac.cn/data/index.html

For comparison, all image line segments in our experiments were extracted using the method provided by LMatch[16], which is based on the Canny edge detector[23].

4.1 Parameter setting

In our proposed algorithm, the image line neighborhood width threshold T needs to be determined. This parameter is used to tolerate possible errors introduced in the 3D point re-projection. According to [2], re-projection errors of reconstructed 3D points are rather small (no more than 2 pixels for experiments on their reported image sets). Thus, in all our experiments, we used a fixed T of 5 pixels for all scenes, which is big enough to counteract the effect of the re-projection errors.

Additionally, matches that are composed by lines from less than 3 views are dumped. This is applied as a minimum requirement for line triangulation.

4.2 Comparison of accuracy

We compared our method with the multi-view line matching toolbox LMatch[16]. The statistics on the accuracy of both methods are shown in Table 1. Due to lack of ground truth data in multi-view line matching, all results are verified by hand. The results show that our method could generate more reliable matches.

4.3 Comparison of matching quality

In order to visualize the matching results, we reconstructed 3D lines from the obtained matches using the nonlinear triangulation method implemented in [17][2]. Figs. 7 and 8 show the 3D lines reconstructed from matches obtained using LMatch and our method. The quality of reconstructed 3D lines could reflect the reliability of the line match results qualitatively .

Note that a variety of factors would lead to poor-quality 3D lines: mismatches, degeneracy, errors in image line detection, and inaccurate camera parameters. Meanwhile, different triangulation algorithms may also exhibit different reliability and stability. However, under exactly the same inputs and the same triangulation method, we can see from Figs. 7 and 8 that our method could obtain more reliable results than LMatch. This is consistent with the conclusion of the accuracy comparison.

4.4 Comparison of robustness

For the line matching problem, it is a big challenge to distinguish closely located parallel image lines. For one thing, such image lines usually have quite similar appearance. Meanwhile, geometric attributes of such lines may also not be distinctive enough. Fig. 9 shows a mismatch generated by the two-view matching method[6].

Table 1 Accuracy comparison between LMatch[16] and our method

	Tsinghua School			Longquan Temple		
	Correct	Total	Accuracy (%)	Correct	Total	Accuracy (%)
LMatch	514	603	85.2	700	903	77.5
Our method	608	613	99.2	1 164	1 190	97.8

Fig. 7 3D lines reconstructed for the Tsinghua School scene from matches obtained by LMatch[16] (the upper row) and our method (the lower row) (Note that degeneracy in triangulation may result in erroneous 3D lines reconstructed from correct matches, like the irregular lines shown in the top-down view (the right image) of the lower row.)

[2]http://www.robots.ox.ac.uk/vgg/hzbook/code/

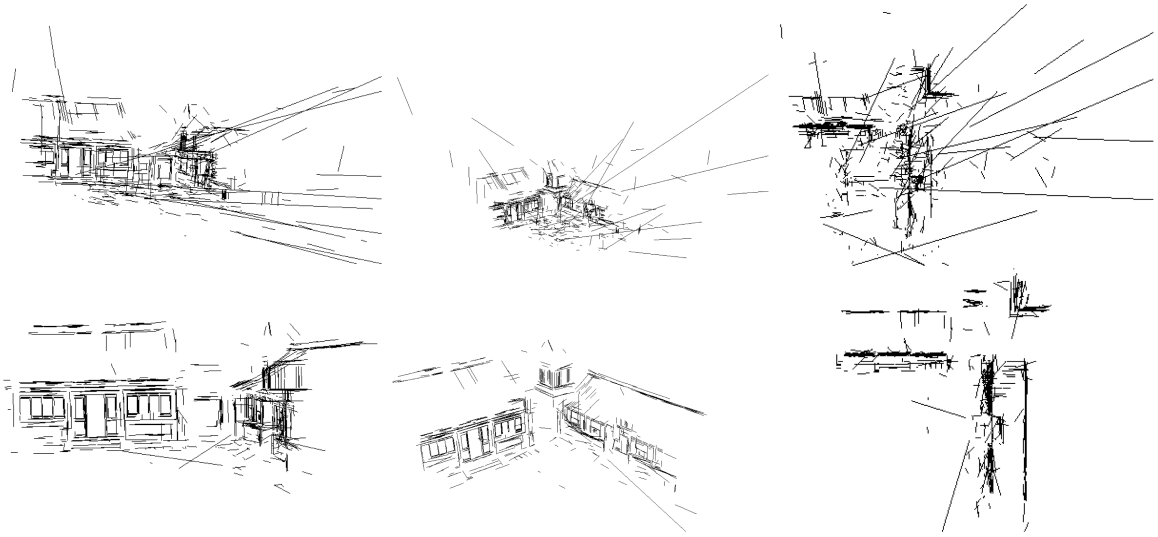

Fig. 8 3D lines reconstructed for the Longquan Temple scene from matches obtained by LMatch[16] (the upper row) and our method (the lower row)

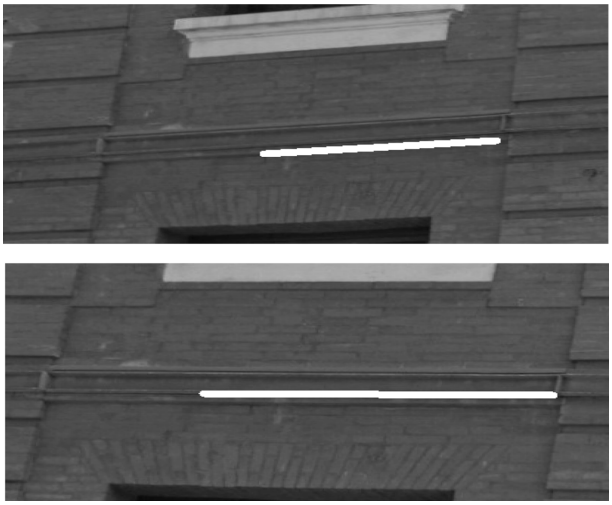

Fig. 9 A mismatch on closely located parallel lines generated by the two-view line matching method[6]

Even in the results of LMatch, which takes global geometric information into account, such mismatches still exist. Fig. 10 gives an example.

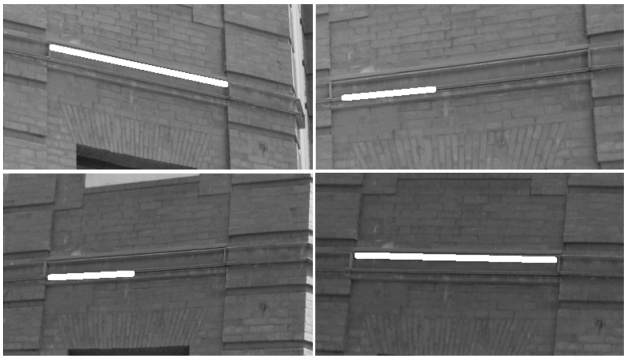

Fig. 10 A mismatch generated by LMatch[16]

In our method, with the statistical information provided by point-line correspondences, the graph-based analysis could give a reliable image line similarity measure. Based on this, the constrained DBSCAN clustering would separate these lines successfully. We selected image lines in the same region as shown in Fig. 10 and showed the resulting matches containing these lines in Fig. 11. The results show that our method is more effective and robust.

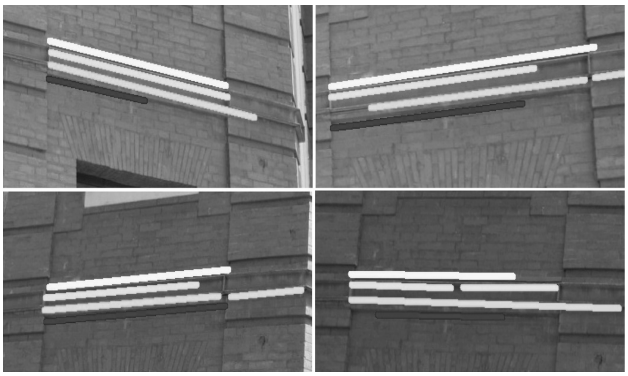

Fig. 11 Matches obtained using our method (Only the ones in the region of interest are shown. All lines drawn in the four views are correctly matched.)

4.5 Computational complexity

Our method can be divided into two main steps: the graph construction step and the graph analysis step.

In the graph construction step, the most time-consuming part is the re-projection step, where all 3D points are re-projected to their visible images. The complexity of this step is $O(K \cdot N)$, where K is the number of 3D points and N is the number of views. Since the visibility information, which is a by-product of PMVS, is used in this process, the re-projection is more efficient and the actual complexity is much smaller than this.

For each connected component of the graph, the com-

plexity is $O(n^2)$, where n is the number of nodes in the connected component. So the overall complexity of graph analysis step could be approximated as $O(|L_{2D}|^2)$.

In summary, our method has a polynomial computational complexity. Meanwhile, both re-projection and graph analysis in our method could be conveniently parallelized, which makes our method scalable for large image sets.

Our experiments were carried out on a workstation with a 2.1 GHz Intel® Xeon CPU and 8 G internal memory. And for each scene, the point-based reconstruction by the Bundler + PMVS pipeline took less than 30 minutes, and our method could finish line matching within 20 minutes.

In contrast, the computation of photometric similarity in LMatch makes it less efficient. Line matching on the two reported scenes using LMatch took 2.8 hours and 4.2 hours, respectively. It can be seen that, even by taking into account the runtime of MVS, our method still took far less time. Meanwhile, the sequential scheme of LMatch greatly limits its application on large image sets.

5 Conclusion and future work

We propose a line matching method across multiple views based on MVS and demonstrate its robustness, accuracy and efficiency on real-world image sets. Within this method, information provided by point-based MVS is fully exploited, and common limitations of existing line matching methods are largely alleviated.

From the results of MVS, point-line correspondences across multiple views are firstly established, encoding both photometric information and geometric constraints. A graph is constructed to help model image lines and their pairwise similarities derived from the correspondences. The spectral graph analysis method provides a dimensionally reduced distance measure between any pair of nodes. With this measure, a constrained clustering algorithm is introduced to eliminate noise introduced during re-projection and establish reliable matches. Experiments show that our method exhibits better accuracy and robustness than the existing methods.

Our method is able to provide reliable multi-view line matches for given 3D point clouds generated by MVS, which could be used for further 3D line or plane modeling. As to the limitation, point information plays such a crucial role in the proposed method that our method could not get satisfactory line matches for areas where 3D points cannot be sufficiently reconstructed, like interior walls. In our future work, we will seek to solve the problem by reducing the dependence on points during line matching.

References

1 Snavely N, Seitz S M, Szeliski R. Photo tourism: exploring photo collections in 3D. *ACM Transactions on Graphics*, 2006, **25**(3): 835−846

2 Furukawa Y, Ponce J. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, **32**(8): 1362−1376

3 Snavely N, Seitz S M, Szeliski R. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 2008, **80**(2): 189−210

4 Werner T, Zisserman A. New techniques for automated architectural reconstruction from photographs. In: Proceedings of the 7th European Conference on Computer Vision. London, UK: Springer-Verlag, 2002. 541−555

5 Aider O A, Hoppenot P, Colle E. A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences. *Robotics and Autonomous Systems*, 2005, **52**(2−3): 229−246

6 Fan B, Wu F C, Hu Z Y. Robust line matching through line-point invariants. *Pattern Recognition*, 2012, **45**(2): 794−805

7 Bay H, Ferraris V, Van Gool L. Wide-baseline stereo matching with line segments. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington DC, USA: IEEE, 2005. 329−336

8 Wang Z H, Wu F C, Hu Z Y. Msld: a robust descriptor for line matching. *Pattern Recognition*, 2009, **42**(5): 941−953

9 López J, Fuciños M, Fdez-Vidal X R, Pardo X M. Detection and matching of lines for close-range photogrammetry. In: Proceedings of the 6th Iberian Conference on Pattern Recognition and Image Analysis. Madeira, Portugal: Springer, 2013. 732−739

10 Wang L, Neumann U, You S. Wide-baseline image matching using line signatures. In: Proceedings of the 12th IEEE International Conference on Computer Vision. Kyoto, Japan: IEEE, 2009. 1311−1318

11 Zhang L L, Koch R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 2013, **24**(7): 794−805

12 Schmid C, Zisserman A. Automatic line matching across views. In: Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Juan, Puerto Rico: IEEE, 1997. 666−671

13 Schmid C, Zisserman A. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 2000, **40**(3): 199−233

14 Heuel S, Förstner W. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Kauai, USA: IEEE, 2001. II517−II524

15 Elaksher A F. Automatic line matching across multiple views based on geometric and radiometric properties. *Applied Geomatics*, 2011, **3**(1): 23−33

16 Werner T. Lmatch: Matlab toolbox for matching line segments accross multiple calibrated images [Online], available: http://cmp.felk.cvut.cz/cmp/software/lmatch/, July 30, 2013

17 Hartley R I, Zisserman A. *Multiple View Geometry in Computer Vision (2nd edition)*. Cambridge: Cambridge University Press, 2004. 321−323

18 Xiao J X, Fang T, Tan P, Zhao P, Ofek E, Quan L. Image-based facade modeling. *ACM Transactions on Graphics*, 2008, **27**(5): Article No. 161

19 Chen T W, Wang Q. 3d line segment detection for unorganized point clouds from multi-view stereo. In: Proceedings of the 10th Asian conference on Computer vision. Queenstown, New Zealand: Springer-Verlag, 2010. 400−411

20 Sander J, Ester M, Kriegel H P, Xu X W. Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, 1998, **2**(2): 169−194

21 Saerens M, Fouss F, Yen L, Dupont P. The principal components analysis of a graph, and its relationships to spectral clustering. In: Proceedings of the 15th European Conference on Machine Learning. Pisa, Italy: Springer-Verlag, 2004. 371−383

22 Daszykowski M, Walczak B, Massart D L. Looking for natural patterns in data: Part 1. Density-based approach. *Chemometrics and Intelligent Laboratory Systems*, 2001, **56**(2): 83−92

23 Canny J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, **8**(6): 679−698

FU Kang-Ping Ph. D. candidate at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He received his bachelor degree in computer science from University of Science and Technology of China in 2009. His research interest covers computer vision and pattern recognition. Corresponding author of this paper.
E-mail: fukangping@163.com

SHEN Shu-Han Associate professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He received his Ph. D. degree from Shanghai Jiao Tong University in 2010. Before that he received his bachelor and master degrees both from Southwest Jiao Tong University in 2003 and 2006, respectively. His research interest covers 3D reconstruction and image based modeling. E-mail: shshen@nlpr.ia.ac.cn

HU Zhan-Yi Professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interest covers computer vision, geometric primitive extraction, vision guided robot navigation, and pattern recognition.
E-mail: huzy@nlpr.ia.ac.cn