ISPRS Journal of Photogrammetry and Remote Sensing 125 (2017) 33–49

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Line segment matching and reconstruction via exploiting coplanar cues

Kai Li, Jian Yao *

Computer Vision and Remote Sensing (CVRS) Lab, School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, PR China

ARTICLE INFO

Article history:
Received 13 July 2016
Received in revised form 26 December 2016
Accepted 9 January 2017
Available online 19 January 2017

Keywords:
Line segment matching
3D line segment reconstruction
Local region detector
Homography estimation
Markov random field

ABSTRACT

This paper introduces a new system for reconstructing 3D scenes from Line Segments (LS) on images. A new LS matching algorithm and a novel 3D LS reconstruction algorithm are incorporated into the system. Two coplanar cues that indicates image LSs are coplanar in physical (3D) space are extensively exploited in both algorithms: (1) adjacent image LSs are coplanar in space in a high possibility; (2) the projections of coplanar 3D LSs in two images are related by the same planar homography. Based on these two cues, we efficiently match LSs from two images firstly in pairs through matching the V-junctions formed by adjacent LSs, and secondly in individuals by exploiting local homographies. We extract for each V-junction a scale and affine invariant local region to match V-junctions from two images. The local homographies estimated from V-junction matches are used to match LSs in individuals. To get 3D LSs from the obtained LS matches, we propose to first estimate space planes from clustered LS matches and then back-project image LSs onto the space planes. Markov Random Field (MRF) is introduced to help more reliable LS match clustering. Experiments shows our LS matching algorithm significantly improves the efficiency of state-of-the-art methods while achieves comparable matching performance, and our 3D LS reconstruction algorithm generates more complete and detailed 3D scene models using much fewer images.

1. Introduction

Image-based 3D reconstruction is a widely studied research field and researchers have developed some remarkable works through exploiting feature points extracted from images (Agarwal et al., 2011; Furukawa and Ponce, 2010; Snavely et al., 2006, 2008; Wu, 2013). However, objects in man-made scenes are often structured and can be outlined by a bunch of Line Segments (LS). It is therefore advantageous to get the 3D wire-frame model of a scene by exploiting LSs on images. For example, for the house shown in Fig. 1(a), the 3D LS reconstruction method to be introduced in this paper generates the 3D model shown in Fig. 1(b) using only two images. It is easy to recognize the house from this 3D model, but hardly possible to do so from the extremely sparse point clouds obtained by some point based 3D reconstruction methods. Some works (Hofer et al., 2014; Sinha et al., 2009) also proved that 3D modeling by exploiting both feature points and LSs on images resulted in more accurate and complete results.

Despite of the above benefits of exploiting LSs for 3D scene reconstruction, it is yet hard to reliably reconstruct 3D LSs. The foremost reason is that LSs are difficult to be matched, such that even several 3D LS reconstruction algorithms (Hofer et al., 2013; Jain et al., 2010; Ramalingam and Brand, 2013) skipped LS matching and directly reconstructed extracted image LSs. The main cause for the difficulties of matching LSs is the absence of point-to-point correspondence between corresponding LSs. The endpoints of corresponding LSs do not reliably correspond with each other, and a short LS from one image is allowed to correspond to a long one from another image. This fact makes it unreliable to exploit some local region description based methods, which have been proved to be very effective in feature point matching, for LS matching because it is hard to extract invariant local regions around LSs.

Another factor complicating 3D LS reconstruction is the unstableness and low location accuracy of extracted LSs. LSs are the straight fittings of curve edges detected on images so that sometimes a 3D edge would result in two straight fittings that are not precisely corresponding on two images. The imprecise correspondence of corresponding LSs makes it difficult to reliably reconstruct their 3D correspondences. For example, to reconstruct 3D LSs in the scene shown in Fig. 1(c), all of which can roughly be regarded to lie on one space plane, when using traditional way to triangulate (forward intersect) LS correspondences identified from two

* Corresponding author.
E-mail address: jian.yao@whu.edu.cn (J. Yao).
URL: http://cvrs.whu.edu.cn/ (J. Yao).

http://dx.doi.org/10.1016/j.isprsjprs.2017.01.006
0924-2716/

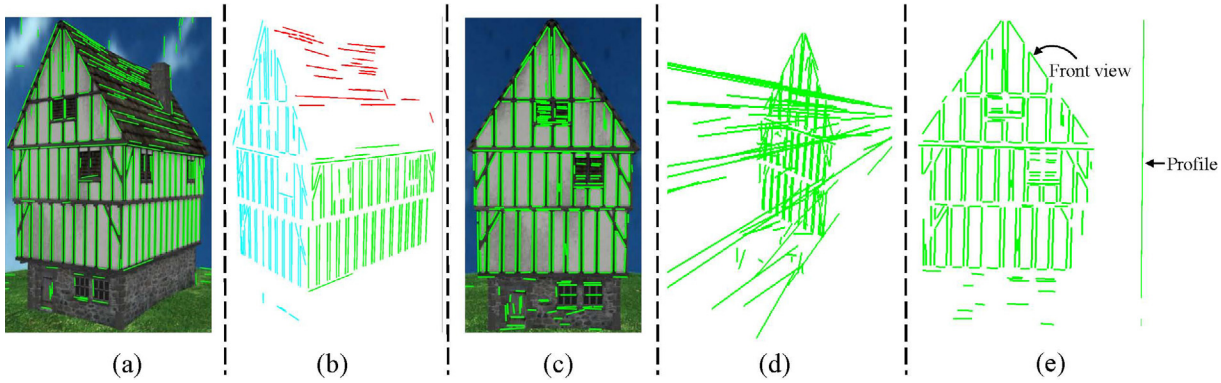

Fig. 1. Examples showing the benefits and difficulties of exploiting LSs on images for 3D reconstruction. (a) A multi-planar scene and the extracted LSs. (b) The reconstructed 3D LSs for the scene (a) obtained by the proposed 3D LS reconstruction method using two images. Different colors are used to differentiate 3D LSs lying on different space planes. (c) An image of a roughly planar scene and the extracted LSs. (d) The 3D LS reconstruction result for the scene shown in (c) by triangulating LS correspondences identified from two images. (e) The 3D LS reconstruction result obtained by our proposed algorithm after solving the problem existing in (d). The front view and profile of the obtained 3D model are shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

images, the 3D LSs shown in Fig. 1(d) are obtained. As we can see, a big fraction of the 3D LSs are incorrectly reconstructed.

We solve the aforementioned problems by exploiting the following two coplanar cues that indicate the coplanarity image LSs in space.

- $\mathcal{C}1$: Adjacent image LSs are coplanar in space in a high possibility.
- $\mathcal{C}2$: The projections of coplanar space LSs in two images shall be related by the same planar homography.

As for the no point-to-point correspondence problem in LS matching, based on $\mathcal{C}1$, we intersect adjacent LSs to form V-junctions in both images to be matched. Since adjacent image LSs are very likely to be coplanar in space and the intersecting junctions of coplanar space LSs are projectively invariant with camera motions, a portion of V-junctions constructed in one image would reappear in the other image. Through matching V-junctions from the two images, the LSs forming the obtained V-junction matches are matched accordingly. While matching V-junctions, we propose to extract for each V-junction a scale and affine invariant local region and describe it with SIFT. For LSs unable to be matched along with V-junctions (due to that they are not adjacent enough to others as to be used to form V-junctions) based on $\mathcal{C}2$, we use local homographies estimated from their adjacent V-junction matches to evaluate their correspondence.

When reconstructing 3D LSs, based on $\mathcal{C}2$, we group LS matches obtained from two images according to a set of homographies, such that LS matches in each group are related by the same homography, which is induced by the space plane where the 3D LSs corresponding to the LS matches in the group lie. The space plane for each LS matches group can then be recovered from the 3D LSs obtained by triangulating all the pairs of LS correspondences. As the space plane being recovered, the 3D LSs corresponding to LS matches in the group can be obtained easily by back-projecting LSs from one image onto the space plane. To reduce the incidence of incorrect LS match grouping, we introduce coplanar cue $\mathcal{C}1$ into LS match grouping, frame it to Markov Random Field (MRF) and solve it as a multi-label optimization problem. Fig. 1(e) shows our 3D LS reconstruction result for the scene shown in Fig. 1(d). It is easy to observe that our algorithm has successfully remedied the problem existing in Fig. 1(d).

In summary, the novelties of this paper are threefold: First, we propose to match V-junction from two images by extracting for each of them a scale and affine invariant local region. Second, we propose a new solution for solving the ambiguities in 3D LS reconstruction through LS match grouping, space plane estimation and image LS back-projection. Third, we formulate to solve the LS match grouping problem by solving a multi-label optimization problem.

The rest of this paper is organized as this: Section 2 presents relevant works to ours. The proposed LS matching algorithm and 3D LS reconstruction algorithm are introduced in Sections 3 and 4, respectively. Experimental results are reported in Section 5, and conclusions are drawn in Section 6.

2. Related works

We give in this section a brief introduction of existing LS matching and 3D LS reconstruction methods.

2.1. Line segment matching

LS matching methods in existing literatures can generally be classified into two groups: methods that match LSs in individuals and those in groups. Some methods matching LSs in individuals exploit the photometric information in the local regions around LSs, like intensity (Baillard et al., 1999; Schmid and Zisserman, 1997), gradient (Verhagen et al., 2014; Wang et al., 2009b; Zhang and Koch, 2013), and color (Bay et al., 2005). All these methods underlie the assumption that there are considerable overlaps between corresponding LSs, which might lead to the failure of these methods when corresponding LSs share insufficient overlapping parts.

Other methods matching LSs in individuals leverage point matches for LS matching (Chen and Shao, 2013; Fan et al., 2010, 2012; Lourakis et al., 2000). These methods first find point matches using the existing point matching methods, and then exploit geometric invariants between coplanar points and line(s) under certain image transformations to evaluate LSs from two images. The LSs which meet the invariants are regarded to be in correspondence. A common disadvantage of these methods is that they depend heavily on point matching results so that once insufficient point matches were found before, these methods would generate inferior results.

Methods matching LSs in groups are more complex, but more constraints are available for disambiguation. Wang et al. (2009a) exploited the stability of the relative positions of the endpoints of a group of LSs in a local region under various image transformations to describe and match LS groups. This method is robust in

some challenging situations, but its dependence on the approximately corresponding relationship between the endpoints of LS correspondences leads to its tendency to produce false matches when substantial disparity exists in the locations of the endpoints of corresponding LSs.

A more common way to match LSs in groups is to match them in pairs (Alshahri and Yilmaz, 2014; Kim et al., 2014; Li et al., 2016b; Ok et al., 2012b). For two LS pair correspondences from two images, $\mathcal{P}_l = (\mathbf{l}_m, \mathbf{l}_n)$ and $\mathcal{P}'_l = (\mathbf{l}'_m, \mathbf{l}'_n)$ for instance, suppose the intersecting junctions of $\mathbf{l}_m$ and $\mathbf{l}_n$ is $\mathbf{j}$, and $\mathbf{j}'$ for $\mathbf{l}'_m$ and $\mathbf{l}'_n$, a series of discriminative constraints were exploited by existing methods to match them. Commonly used constraints include: (1) Cross angle constraint. Since $\mathbf{l}_m$ and $\mathbf{l}_n$ ($\mathbf{l}'_m$ and $\mathbf{l}'_n$ as well) are in a local region, their cross angle should be of little difference with that of $\mathbf{l}'_m$ and $\mathbf{l}'_n$. (2) Epipolar line constraint. $\mathbf{j}$ and $\mathbf{j}'$ are point correspondences so that they should meet epipolar line constraint. (3) Local region similarity constraint. The local region determined by $\mathbf{l}_m, \mathbf{l}_n$ and $\mathbf{j}$, should be similar to the local region determined by $\mathbf{l}'_m, \mathbf{l}'_n$ and $\mathbf{j}'$, in light of their photometric characteristics (e.g., pixel intensity and gradient). Algorithms exploiting this constraint focus on how to give discriminative descriptions of the local regions. (4) Homography constraint. Since $\mathbf{l}_m$ and $\mathbf{l}_n$ ($\mathbf{l}'_m$ and $\mathbf{l}'_n$ as well) are in a local region, there is a high possibility that they are coplanar in physical space. In this case, there exists a planar homography that establishes point-to-point correspondence to help match LSs. How to estimate the planar homography is the crux in applying this constraint. The above constraints are not exclusive with each other, and are often assembled to produce a powerful LS matcher.

Our proposed LS matcher also matches LSs in pairs and employs all the four commonly used constraints mentioned above. The differences between our method with others in this category are as follows. When using the local region similarity constraint, Ok et al. (2012b) used spatiograms (a measure encoding both the color and coordinate information of pixels in a region) to measure the similarity of two quadrilateral regions; Kim et al. (2014) rectified local regions to squared patches and used normalized cross correlation (NCC) as a similarity measure for the patches; we instead find scale and affine invariant local regions around intersecting junctions and use SIFT to describe the extracted local regions. Since our local region extractor is very robust and SIFT is a powerful local region descriptor, our strategy is very effective in using this constraint. When applying the homography constraint, Alshahri and Yilmaz (2014) and Sun et al. (2015) estimated the planar homography between two local regions using adjacent point matches, which were provided by external point matching methods. But in our algorithm, the planar homography between two local regions is estimated from their adjacent LS pair match, which is obtained previously by our method and can therefore always be guaranteed. In this perspective, our algorithm is self-sustaining and independent. Our LS matcher is a direct promotion of a recent one, referred as LJL (Li et al., 2016b). We targets to improve the efficiency of LJL. We avoid to deal with scale changes among images by the time-consuming scale simulation procedure in LJL, but instead by extracting for each junction generated in the original images a scale and affine invariant local region. We conduct this local region extraction procedure only in the original images and hence tremendously improves the matching efficiency of LJL.

2.2. 3D line segment reconstruction

We divide existing 3D LS reconstruction methods into two categories: methods that require LS matching before reconstruction and those do not. Many methods in the former category focus on the exploitation of different mathematical representations for a 3D line to establish the projective relationship between a 2D line and its 3D correspondence, which is not as explicit as that for points (Hartley and Zisserman, 2003). These 3D line representations include plücker coordinates (Bartoli and Sturm, 2005; Martinec and Pajdla, 2003; Přibyl et al., 2015), pair of points (Baillard et al., 1999; Habib et al., 2002; Hartley and Zisserman, 2003; Ok et al., 2012a; Smith et al., 2006; Werner and Zisserman, 2002), pair of planes (Hartley and Zisserman, 2003), a unitary direction vector and a point on a line (Taylor and Kriegman, 1995), the intersections of a line with two orthogonal planes (Spetsakis and Aloimonos, 1990), and a recent one, Cayley representation (Zhang and Koch, 2014). With these representations, researchers proposed various methods for reconstructing 3D lines and/or estimating camera parameters. Some methods in the first category aim to reconstruct 3D LSs in certain types of scenes, like scenes meeting Manhattan World assumption (Kim and Manduchi, 2014; Schindler et al., 2006), piecewise planar scenes (Sinha et al., 2009) and poorly textured scenes (Bay et al., 2006). The prior knowledge of the scenes decreases reconstruction uncertainties and often leads to remarkable results.

Some recent algorithms in this area attempted to free the reconstruction procedure from the heavy dependence on the LS matching procedure because it is sometimes hard to get reliable LS correspondences in some kinds of scene types, such as poorly textured indoor environments and scenes containing wiry structures (e.g., power pylons (Hofer et al., 2013)). Most of these methods adopt the strategy of first generating a set of 3D hypotheses for each extracted LS, either by sampling the depths of the endpoints of 3D LSs to camera centers (Jain et al., 2010), or by triangulating gross LS correspondences obtained after enforcing some soft constraints on the extract LSs (Hofer et al., 2013, 2014, 2016). Next, they validate these hypotheses by projecting them back to images. The algorithm proposed by Ramalingam and Brand (2013) is able to obtain 3D LSs with an unknown global scale from a single image capturing a Manhattan World scene. It is possible to do so because LSs in this special type of scenes can only distribute in three orthogonal directions, hence dramatically decreasing the degrees of freedom when to reconstruct the scene LSs.

Our 3D LS reconstruction algorithm belongs to the first category and we focus only on 3D LS reconstruction. The camera parameters are obtained by some external camera calibration methods, or some existing SFM pipelines. The most similar method to ours is proposed by Kim and Manduchi (2014), who also focused on recovering planar structures of a scene from LSs. But their method is confined to be only applicable for structured scenes which meet Manhattan World assumption, while our method is a more general one and do not underlie this pretty strong assumption. Besides, their method exploits parallel LSs to determine their spatial coplanarity, while our method instead uses planar homographies.

3. Line segment matching algorithm

Our LS matching method matches LSs from two images in two forms, firstly in pairs by matching their intersecting V-junctions, and secondly in individuals through exploiting local homographies. This section presents the two LS matching forms in order.

3.1. V-junction matching

We present the ways of generating, describing and matching V-junctions orderly in this part.

3.1.1. V-junction generation

Only the intersecting junctions of 2D LSs whose 3D correspondences are coplanar in space are invariant with camera motions and can possibly find their 2D correspondences in other images.

Based on coplanar cue $\mathcal{C}1$ we stated in the introduction section, we intersect adjacent LSs to get repeatable junctions. Refer to Fig. 2(a), similar to the strategy of Kim et al. (2014), for a LS $\mathbf{l}_1$, we define the rectangle $\mathcal{R}$ as its impact zone (filled in yellow[1] in the figure), which centers at the midpoint of $\mathbf{l}_1$ and has the width of $|\mathbf{l}_1| + 2w$ and the height of $2w$, where $|\mathbf{l}_1|$ denotes the length of $\mathbf{l}_1$ and w is a user-defined parameter, set as 20 in this paper. A LS satisfying the following two conditions is assumed to be coplanar with $\mathbf{l}_1$ in 3D space: First, it has at least one endpoint dropping in $\mathcal{R}$. Second, its intersection with $\mathbf{l}_1$ is also within $\mathcal{R}$. Under these two conditions, in Fig. 2(a), only $\mathbf{l}_2$ is accepted to be used to intersect with $\mathbf{l}_1$ to construct V-junction(s).

There exist two distribution forms where two LSs are used to construct V-junction(s), as shown in Fig. 2(b). In the left form where the intersection of $\mathbf{l}_1$ and $\mathbf{l}_2$ lies on one of them (not on their extensions), two V-junctions, $\widehat{AOC}$ and $\widehat{BOC}$ are constructed. We call $\widehat{AOC}$ an acute V-junction because angle α is an acute angle (right angle cases are also included in this type); analogously, we call $\widehat{BOC}$ an obtuse V-junction. In this distribution form, we construct two V-junctions. We do this because it is unknown which type(s) of V-junction(s) is (are) constructed using the correspondence of $\mathbf{l}_1$ (let it be $\mathbf{l}'_1$) and the correspondence $\mathbf{l}_2$ (let it be $\mathbf{l}'_2$). If an acute V-junction is constructed using $\mathbf{l}'_1$ and $\mathbf{l}'_2$, it can be matched with $\widehat{AOC}$; if an obtuse V-junction is constructed using $\mathbf{l}'_1$ and $\mathbf{l}'_2$, it can be matched with $\widehat{BOC}$; lastly, if two V-junctions are constructed using $\mathbf{l}'_1$ and $\mathbf{l}'_2$, the acute V-junction and obtuse V-junction can be matched with $\widehat{AOC}$ and $\widehat{BOC}$, respectively. In the right distribution form where the intersection of the two LSs lies on their extensions, only one V-junction, $\widehat{AOC}$ is constructed.

3.1.2. V-junction description

To match V-junctions constructed in two images, we exact for each V-junction a scale and affine invariant region and describe it with SIFT descriptor. Our idea derives from the edge-based region (EBR) detector proposed by Tuytelaars and Van Gool (2004), which exploits features of edges on images to detect invariant regions. We exploit features of LSs to extract invariant regions for V-junctions. Besides, we consider the relationship between adjacent LSs to make extracted invariant regions less sensitive to noises and location imprecision of LSs.

For a V-junction, we detect for each of the two LSs forming it a stable point which has the greatest intensity change among all points on the LS. Suppose $\mathbf{l}$ is one of the two LSs forming a V-junction $\mathcal{V}$, if $\mathbf{l}$ was used to construct V-junctions besides $\mathcal{V}$, as LS $\overline{\mathbf{AB}}$ shown in Fig. 3(a), the junction points are selected as the candidates of the stable point. This is reasonable because LSs lie at region borders, so that if $\mathbf{l}$ meets a LS, it likely reaches a region border. The junction of $\mathbf{l}$ and the LS it meets is very possible to be a stable point. On the other hand, if $\mathbf{l}$ was not used to construct any V-junction other than $\mathcal{V}$, as LS $\overline{\mathbf{CD}}$ shown in Fig. 3(a), all points on $\mathbf{l}$ are regarded as candidate stable points. In this way, we collect a set of candidate stable points $\mathcal{C} = \{\mathbf{S}_i\}_{i=1}^{N_s}$ for $\mathbf{l}$. We select the best candidate as the one which has the most abrupt intensity change along $\mathbf{l}$. For a candidate $\mathbf{S}_i$, we collect 5 points in both its sides along $\mathbf{l}$ and gather two sets of the pixel intensities, $\mathcal{I}_1(\mathbf{S}_i)$ and $\mathcal{I}_2(\mathbf{S}_i)$. The intensity change of $\mathbf{S}_i$ is calculated as: $I_d(\mathbf{S}_i) = |med(\mathcal{I}_1(\mathbf{S}_i)) - med(\mathcal{I}_2(\mathbf{S}_i))|$, where $med(\cdot)$ means the median value of a set of numbers. The $\mathbf{S}_i$ with the maximal value of $I_d(\mathbf{S}_i)$ is selected as the stable point for $\mathbf{l}$. Here, median value of a set of intensities is used to compute the intensity change of a candidate point because this can reduce the influence of noises among the sets.

After finding a stable point for the two LSs forming a V-junction, as $\mathbf{S}_1$ for $\overline{\mathbf{AB}}$ and $\mathbf{S}_2$ for $\overline{\mathbf{CD}}$ shown in Fig. 3, the parallelogram determined by these two stable points and the junction is regarded as the invariant region for the V-junction, as shown in Fig. 3(b). Fig. 4 shows the extracted invariant regions by our method on two images with different sizes and a great viewpoint change. We can see that, in this extreme case, some (nearly) identical regions are extracted in the two images.

Since most signal variations exist in the vicinities of LSs, we expand the extracted parallelogram into a larger one to include more discriminative information, as shown in Fig. 3(c). Next, we normalize the expanded parallelogram into a square through affine transformation to make it affine invariant. Finally, we describe the square with SIFT (Fig. 3(d)). The size of the square is suggested to be 41×41 by Mikolajczyk et al. (2005), but we find in our case the matching results are better when it is set as 21×21.

3.1.3. V-junction matching

To match V-junctions from two images, the general way is to evaluate the Euclidean distances of their description vectors. But since the two LSs forming a V-junction are in a local region, their crossing angle should vary in a small range under most image transformations. We can use this simple constraint to discard many false candidates before evaluating the description vector distance. Let $(\mathcal{V}, \mathcal{V}')$ be a pair of V-junctions to be matched and (θ, θ') be the crossing angles of the two pairs of LSs. If $(\mathcal{V}, \mathcal{V}')$ is a correct match, the difference between θ and θ' should be less than a small threshold ϵ_1 (set as 30° in this paper), i.e., $|\theta - \theta'| < \epsilon_1$. Once $\mathcal{V}$ and $\mathcal{V}'$ meet this constraint, we evaluate them further by computing their description vector distance. When the distance is less than the given threshold d_t (set as 0.4 in this paper), we accept them as a candidate match. There may exist the case that a V-junction in one image is matched with several V-junctions in the other image, we keep only the pair of V-junctions with the smallest distance for later use.

There inevitably exist false matches after evaluating V-junctions from the two images using the above strategy. We eliminate the false matches using the following two constraints. First, we estimate the fundamental matrix for the two images from the obtained V-junction matches using RANSAC. The fundamental matrix enforces epipolar line constraint on the obtained V-junction matches and we keep only those meeting this constraint. Epipolar line constraint cannot filter out false matches that lie near corresponding epipolar lines. We refine the obtained V-junction matches further by exploiting the stability of the topological distribution of a group of V-junctions in a local region with image transformations.

Refer to Fig. 5, for a V-junctions $\mathcal{V}_c$, the two LSs forming it and their reverse extensions form a coordinate-like structure, in which the neighbors of $\mathcal{V}_c$ ($\mathcal{V}_{1\sim8}$ in the figure) distribute in the four quadrants. This topological distribution is quite stable with image transformations, i.e., after some kinds of image transformations, while $\mathcal{V}_c$ is transformed into $\mathcal{V}'_c$, its neighbors should change consistently. To apply this constraint to refine the obtained V-junction matches, for each candidate V-junction match $(\mathcal{V}_c, \mathcal{V}'_c)$, we collect the K ($K = 10$ used in this paper) nearest matched V-junctions as $\tilde{\mathcal{N}} = \{\mathcal{V}_i\}_{i=1}^{K}$ and $\tilde{\mathcal{N}}' = \{\mathcal{V}'_j\}_{j=1}^{K}$ for $\mathcal{V}_c$ and $\mathcal{V}'_c$, respectively. If $(\mathcal{V}_c, \mathcal{V}'_c)$ is a correct match, the following two conditions must be satisfied. First, there should exist a sufficiently large proportion (0.5 used in this paper) of correspondences in $\tilde{\mathcal{N}}$ and $\tilde{\mathcal{N}}'$. Second, the correspon-

[1] For interpretation of color in Figs. 2 and 7, the reader is referred to the web version of this article.

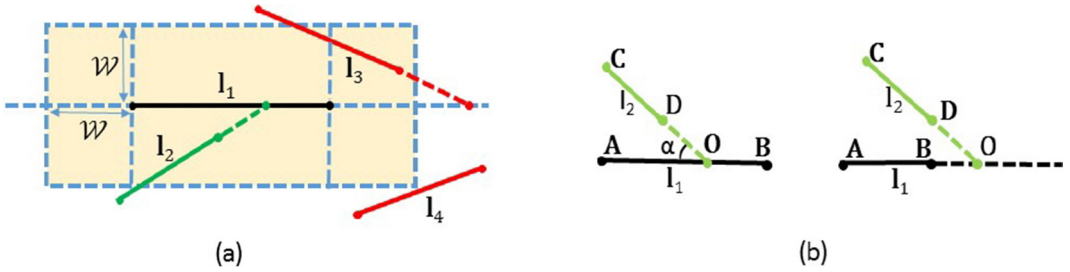

Fig. 2. V-junction generation. (a) Finding image LSs possibly coplanar in 3D space. (b) Two distribution forms of a pair of adjacent LSs.

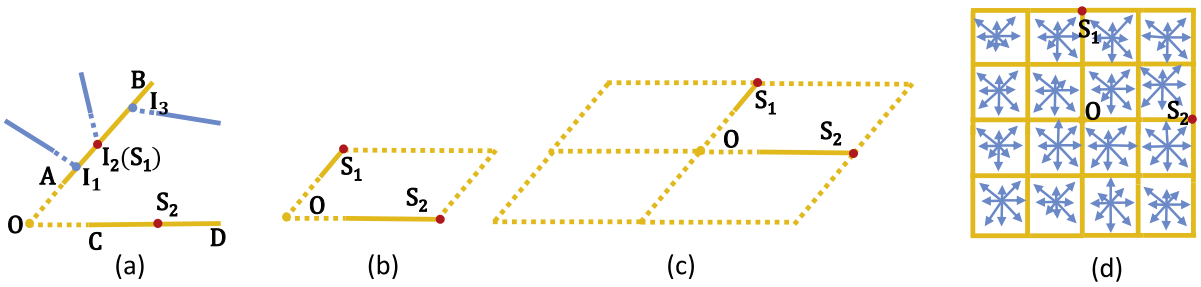

Fig. 3. Illustration of the scale and affine invariant local region extraction and description procedures of our proposed method. (a) Finding stable points on the two line segments forming V-junction $\widehat{BOD}$. (b) The extracted local region. (c) The expanded local region. (d) Describing the normalized local region with SIFT.

Fig. 4. Scale and affine invariant local region extraction on two images (Mishkin et al., 2013) with a great viewpoint change. The parallelograms drawn in different colors are the extracted local regions. Only a subset of all extracted parallelograms are shown in the two images for better interpretation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

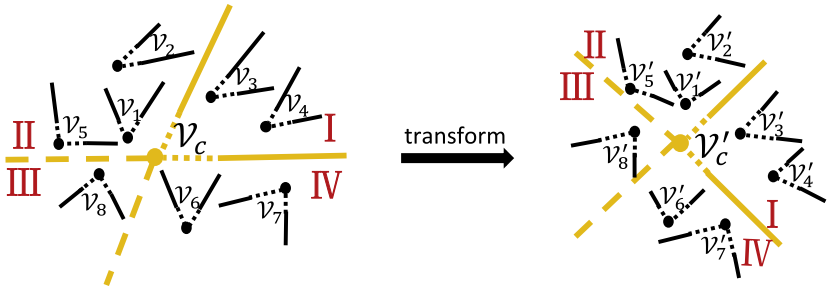

Fig. 5. Illustration of the stability of the topological distribution among adjacent V-junctions with image transformations.

dences in $\tilde{\mathcal{N}}$ and $\tilde{\mathcal{N}}'$ lying in the same quadrants of the coordinates formed by $\mathcal{V}_c$ and $\mathcal{V}_c'$ should account for a great ratio of the total correspondences; the ratio is set as 0.8 in this paper. For example, suppose $\mathcal{V}_i \in \tilde{\mathcal{N}}$ corresponds to $\mathcal{V}_j' \in \tilde{\mathcal{N}}'$, if $\mathcal{V}_i$ lies in quadrant III of the coordinates centered at $\mathcal{V}_c$, $\mathcal{V}_j'$ should also lie in quadrant III of the coordinates centered at $\mathcal{V}_c'$ in a high possibility.

With the guidance of epipolar line constraint and the topological distribution constraint among neighboring V-junctions, V-junctions from the two images can be matched exhaustively by alternatively adding new matches and deleting false ones until no more correct match can be added (Li et al., 2016b).

3.2. Individual line segment matching

LSs in the two images that lie far away from others and were not used to form V-junctions with others will be matched in individuals. They will first be grouped according to the matched V-junctions, and then matched in corresponding groups based on the local homographies estimated from V-junction correspondence pairs.

3.2.1. Local homography estimation

Coplanar cue $\mathcal{C}1$ we stated in the introduction section indicates that adjacent LSs on an image are very likely to be coplanar in space. Therefore, the two pairs of LS correspondences brought by a V-junction match can be regarded as the projections of two coplanar 3D LSs onto two images. We can estimate from the two pairs of LS correspondences the planar homography induced by the plane on which their corresponding 3D LSs lie, with the help of the estimated fundamental matrix (already obtained using V-junction matches, see Section 3.1.3).

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 $\mathbf{F}$ for the two images is known, then $\mathbf{H}^\top\mathbf{F}$ is skew-symmetric (Luong and Vieville, 1996), i.e.

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

The above equation gives five independent constraints on $\mathbf{H}$, and three others are required to fully describe a homography.

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

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

where the 3D plane is represented by $\boldsymbol{\pi} = (\mathbf{u}^\top, 1)$ in the projective reconstruction with camera matrices $\mathbf{C} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{C}' = [\mathbf{A}|\mathbf{e}']$. For a LS match $(\mathbf{l}, \mathbf{l}')$, suppose $\mathbf{x}$ is an endpoint of $\mathbf{l}$, the homography maps it to its corresponding point $\mathbf{x}'$ as: $\mathbf{x}' = \mathbf{H}\mathbf{x}$. Since $\mathbf{l}$ and $\mathbf{l}'$ correspond with each other, $\mathbf{x}'$ must be a point lying on $\mathbf{l}'$, that is $\mathbf{l}'^\top\mathbf{x}' = 0$. Therefore, we obtain

$$\mathbf{l}'^\top(\mathbf{A} - \mathbf{e}'\mathbf{u}^\top)\mathbf{x} = 0. \tag{3}$$

Arranging the above equations, we finally get

$$\mathbf{x}^\top\mathbf{u} = \frac{\mathbf{x}^\top\mathbf{A}^\top\mathbf{l}'}{\mathbf{e}'^\top\mathbf{l}'}, \tag{4}$$

which is linear in $\mathbf{u}$. Each endpoint of a LS in a LS match provides an equation, and the two LS matches brought by a V-junction match provide totally four constraint equations. A least-square solution of $\mathbf{u}$ can be obtained from the four equations, and the local homography $\mathbf{H}$ can then be computed from Eq. (2).

3.2.2. Individual line segment matching

Let $\mathcal{M} = \{(\mathcal{V}_m, \mathcal{V}'_m)\}_{m=0}^{S}$ be the set of S V-junction matches identified from the two images, where $(\mathcal{V}_m, \mathcal{V}'_m)$ denotes the m-th V-junction match. Let $\mathcal{K} = \{\mathbf{l}_i\}_{i=1}^{M_k}$ and $\mathcal{K}' = \{\mathbf{l}'_j\}_{j=1}^{N_k}$ be the two groups of individual LSs, which have not been matched before, from the two images, respectively. For each individual LS $\mathbf{l}_i \in \mathcal{K}$ or $\mathbf{l}'_j \in \mathcal{K}'$, we find k_l ($k_l = 4$ in this paper) of its nearest matched V-junctions and assign it into the corresponding k_l groups. After that, any matched V-junction in $\mathcal{K}$ and $\mathcal{K}'$ collects zero to multiple individual LS(s). We match individual LSs group by group, i.e., a individual LS from a group in one image is evaluated only with individual LSs from the corresponding group in the other image. Note here we redundantly assign each LS into k_l groups, which will cause some LS pairs being evaluated in multiple times. But it is still necessary to do so to ensure two true (correct) LS correspondences to be assigned into at least one pair of corresponding groups and evaluated at least one time.

Suppose $\mathbf{l}$ and $\mathbf{l}'$ are a pair of individual LSs to be evaluated and they are collected by the matched V-junctions $\mathcal{V}$ and $\mathcal{V}'$, respectively. Suppose $\mathcal{V}$ is formed by LS pair $(\mathbf{l}_p, \mathbf{l}_q), (\mathbf{l}'_p, \mathbf{l}'_q)$ for $\mathcal{V}'$. The directions of adjacent LSs should change similarly with image transformations. Let σ be the direction difference of $\mathbf{l}$ and $\mathbf{l}'$, σ_p for $\mathbf{l}_p$ and $\mathbf{l}'_p$, and σ_q for $\mathbf{l}_q$ and $\mathbf{l}'_q$. If $\left|\sigma - \frac{\sigma_p+\sigma_q}{2}\right| < \epsilon_2$, where ϵ_2 is a user-defined threshold set as 20° in this paper, we accept $(\mathbf{l}, \mathbf{l}')$ temporarily and take it for further evaluation. We next test $(\mathbf{l}, \mathbf{l}')$ again using the brightness constraint (Bay et al., 2005), which requires the brighter sides of two corresponding LSs to be the same. The brighter side of a LS refers to the side where the average intensity value of pixels in a small profile along the LS is greater than that of the other side.

If $(\mathbf{l}, \mathbf{l}')$ satisfies the above constraints, we evaluate it further by the local homography $\mathbf{H}_l$, which is estimated from $\mathcal{V}$ and $\mathcal{V}'$ using the strategy presented in Section 3.2.1. We map $\mathbf{l}$ and $\mathbf{l}'$ by $\mathbf{H}_l$, generating their correspondences $\mathbf{l}_h$ for $\mathbf{l}$, and $\mathbf{l}'_h$ for $\mathbf{l}'$. 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 $\mathbf{l}'$, is defined as the mapping error of $(\mathbf{l}, \mathbf{l}')$. After that, there may exist the cases that one LS in one image is matched with several LSs in the other image. We select the pair with the minimal mapping error as the correct match and reject the others.

4. 3D line segment reconstruction algorithm

This section first presents our method for 3D LS reconstruction from two views (images), and then introduces how we extend the two-view based method into multiple views. To be clear, in this paper, when we say *multiple views*, we mean three or more views.

4.1. Two-view based 3D line segment reconstruction

Given images $\mathbf{I}$ and $\mathbf{I}'$, suppose their corresponding camera poses are $\mathbf{C}$ and $\mathbf{C}'$, which can be obtained by some existing SFM pipelines, such as the famous *Bundler* (Snavely et al., 2006, 2008), or some camera calibration methods (Přibyl et al., 2015; Zhang, 1999). Suppose LS matches obtained from $\mathbf{I}$ and $\mathbf{I}'$ by a LS matcher is $\tilde{\mathcal{M}} = \{(\mathbf{l}_r, \mathbf{l}'_r)\}_{r=1}^{N_r}$. Note that the LS matcher is not necessarily the one we introduced above; our 3D LS reconstruction algorithm is independent to the LS matcher used. For every LS match $(\mathbf{l}_r, \mathbf{l}'_r) \in \tilde{\mathcal{M}}$, we search its spatial neighbors in $\tilde{\mathcal{M}}$ by finding matched LSs from $\mathbf{I}$ which are adjacent to $\mathbf{l}_r$. A LS match is regarded to be a neighbor of $(\mathbf{l}_r, \mathbf{l}'_r)$ when any one of the two endpoints of the matched LS from $\mathbf{I}$ is within a rectangle centered around $\mathbf{l}_r$. For example, if matched LS $\mathbf{l}_s$ is found to be adjacent enough to $\mathbf{l}_r$, LS match $(\mathbf{l}_s, \mathbf{l}'_s)$ is regarded as a neighbor of LS match $(\mathbf{l}_r, \mathbf{l}'_r)$. The width of the rectangle equals to the length of $\mathbf{l}_r$, while its height equals 20 pixels (10 pixels in both sides of $\mathbf{l}_r$) in this paper. If we find at least one neighbor for $(\mathbf{l}_r, \mathbf{l}'_r)$, we can estimate the corresponding local homography using the method presented in Section 3.2.1. Please note that Section 3.2.1 shows that it is enough to estimate a homography using two LS matches; if more LS matches are available so long as they are induced by coplanar 3D LSs, they can provide more constraints on homography estimation. Therefore, if we

find for $(\mathbf{l}_r, \mathbf{l}'_r)$ multiple neighbors, we can use all of them to estimate the homography. In contrast, if we fail to find any neighbor for $(\mathbf{l}_r, \mathbf{l}'_r)$, we are unable to calculate the homography. Having processed all LS matches in $\tilde{\mathcal{M}}$, we obtain a set of homographies, $\mathcal{H} = \{\mathbf{H}_i\}_{i=1}^{N_h}$, where N_h denotes the total number of homographies obtained and it is often much smaller than the number of elements in $\tilde{\mathcal{M}}$ because we often cannot find for some LS matches even one neighbor.

In most cases, it is enough to robustly estimate the planar homographies induced by main planes in the captured scene using only the obtained LS matches. But occasionally a LS matcher, especially when it is a weak one, cannot find enough LS matches that are induced by 3D LSs coming from some main scene planes. To avoid this situation, we employ point matches obtained from the two images to assist for homography estimation.

Suppose $(\mathbf{p}, \mathbf{p}')$ is a point match, a homography $\mathbf{H}$ relates the two points as $\mathbf{p}' = \mathbf{Hp}$. Replacing $\mathbf{H}$ using Eq. (2), we have

$$\mathbf{p}' = \mathbf{Ap} - \mathbf{e}'(\mathbf{u}^\top \mathbf{p}). \tag{5}$$

From this equation, we know vectors $\mathbf{p}'$ and $\mathbf{Ap} - \mathbf{e}'(\mathbf{v}^\top \mathbf{p})$ are parallel, so that their vector product is supposed to be zero:

$$\mathbf{p}' \times (\mathbf{Ap} - \mathbf{e}'(\mathbf{u}^\top \mathbf{p})) = (\mathbf{p}' \times \mathbf{Ap}) - (\mathbf{p}' \times \mathbf{e}')(\mathbf{u}^\top \mathbf{p}) = \mathbf{0}. \tag{6}$$

When using Eq. (6) to form the scalar product with the vector $\mathbf{p}' \times \mathbf{e}'$, we have

$$\mathbf{p}^\top \mathbf{u} = \frac{(\mathbf{p}' \times (\mathbf{Ap}))^\top (\mathbf{p}' \times \mathbf{e}')}{(\mathbf{p}' \times \mathbf{e}')^\top (\mathbf{p}' \times \mathbf{e}')}. \tag{7}$$

Same as to Eq. (4), this equation is also linear in $\mathbf{u}$ and provides one constraint.

Therefore, when point matches between the two images are available, for every LS match $(\mathbf{l}_r, \mathbf{l}'_r) \in \tilde{\mathcal{M}}$, while searching for its neighboring LS matches, we also find its point match neighbors. If a matched point from $\mathbf{I}$ is within the rectangle centered around $\mathbf{l}_r$, the corresponding point match is also used to estimate the local homography.

4.1.1. Line segment match grouping

Coplanar cue $\mathcal{C}2$ we stated in the introduction section indicates that the projections of coplanar space LSs into two images shall be related by the same homography. Based on this cue, we cluster LS matches in $\tilde{\mathcal{M}}$ using homographies in $\mathcal{H}$. For a LS match $(\mathbf{l}, \mathbf{l}') \in \tilde{\mathcal{M}}$, we find a homography $\mathbf{H} \in \mathcal{H}$ which minimizes the distance of a pair of LSs according to a homography:

$$d = \frac{\mathbf{l}'^\top \mathbf{H}\mathbf{x}_1 + \mathbf{l}'^\top \mathbf{H}\mathbf{x}_2 + \mathbf{l}^\top \mathbf{H}^{-1}\mathbf{x}'_1 + \mathbf{l}^\top \mathbf{H}^{-1}\mathbf{x}'_2}{4}, \tag{8}$$

where $\mathbf{x}_{i=1,2}$ and $\mathbf{x}'_{j=1,2}$ denote the two endpoints of $\mathbf{l}$ and $\mathbf{l}'$, respectively. Note that each of the four components of the right side of the above equation measures the distance from an endpoint of one LS to the other LS according to the given homography. For example, $\mathbf{l}'^\top \mathbf{H}\mathbf{x}_1$ measures the distance from $\mathbf{x}_1$ to $\mathbf{l}'$ according to $\mathbf{H}$. In other words, it is the distance between point $\mathbf{x}_1^h = \mathbf{H}\mathbf{x}_1$ and $\mathbf{l}'$: $\mathbf{l}'^\top \mathbf{x}_1^h = \mathbf{l}'^\top \mathbf{H}\mathbf{x}_1$, where $\mathbf{x}_1^h$ is the mapping of $\mathbf{x}_1$ under $\mathbf{H}$ from $\mathbf{I}$ to $\mathbf{I}'$.

Having found for every LS matches in $\tilde{\mathcal{M}}$ a most *consistent* homography (with the smallest distance measure defined in Eq. (8)) in $\mathcal{H}$, we get a set of LS match groups $\mathcal{S} = \{\mathcal{G}_t\}_{t=1}^{T}$, where $\mathcal{G}_t$ denotes the t-th LS match group which is formed based on a homography in $\mathcal{H}$. LS correspondences in each LS match group are related by the same homography induced by a space plane in the scene. Next, we merge some groups in $\mathcal{S}$ to ensure that LS matches induced by coplanar 3D LSs are clustered into only one group. For two LS match groups, $\mathcal{G}_u$ and $\mathcal{G}_v$, suppose they are formed based on homographies $\mathbf{H}_u$ and $\mathbf{H}_v$, respectively, if LS matches in $\mathcal{G}_u$ are *consistent* with $\mathbf{H}_v$, and the same goes for $\mathcal{G}_v$ and $\mathbf{H}_u$, we merge the two groups into one. Here, a group of LS matches are "consistent" with a homography means the average of their distances according to the homography (the distance measure is defined in Eq. (8)) is smaller than a given threshold (2 pixels in this paper). After this, we obtain an updated LS group set $\mathcal{S}$, in which the number of groups drops significantly.

4.1.2. Line segment match grouping result refinement

We found that it often brought in mistakes when we grouped LS matches only based on the distance of two LS correspondences according to the estimated homographies, such that some LS matches which should be assigned into one group but were clustered into another group mistakenly. This kind of mistakes frequently occur when there are several similar space planes in the scene and the estimated homographies are not so accurate. For instance, Fig. 6(a) shows an example of the LS match grouping result using the strategy presented above. We drawn in different colors the matched LSs in one of the two used images to differentiate the groups they belong. LSs drawn in the same color are supposed to appear on the same scene plane if they had been correctly grouped. But, as we can see, a considerable number of them are mistakenly clustered.

Coplanar cue $\mathcal{C}1$ we stated in the introduction section indicates that adjacent image LSs are projected from the same space plane in a high possibility. This cue can be framed into Markov Random Field (MRF). We propose to formulate the LS match grouping problem as a multi-label optimization problem and solve it by minimizing the following energy function

$$E = \sum_p D_p(l_p) + \sum_{p,q} V_{p,q}(l_p, l_q), \tag{9}$$

where the data term D_p is the cost of a LS match $p = (\mathbf{l}_p, \mathbf{l}'_p)$ being labeled to belong to a group l_p. Suppose the homography relating LS matches in l_p is $\mathbf{H}_{l_p}$, D_p can then be calculated from Eq. (8). The smoothness term $V_{p,q}$ measures the cost of two neighboring LS matches p and q being labeled to belong to groups l_p and l_q, respectively. To define $V_{p,q}$, an adjacency graph among the LS matches needs to be constructed. Inspired by Delong et al. (2012) and Pham et al. (2014) who constructed Delaunay triangles for feature points to define their adjacency, we construct Delaunay triangles using the midpoints of matched LSs in the first image to define the adjacent relationship among the LS matches. Fig. 6(b) shows the constructed Delaunay triangles corresponding to Fig. 6(a). With the adjacency graph derived from the Delaunay triangles, we set the smoothness term as

$$V_{p,q}(l_p, l_q) = \begin{cases} sw_{pq} & l_p \neq l_q \\ 0 & l_p = l_q, \end{cases}$$

where w_{pq} is the weight for the edge linking vertexes p and q in the adjacency graph. It is assigned by Gaussian function according to the distance between the two vertexes to encourage vertexes with smaller distances being assigned with the same label in a higher possibility. s is a constant serving to amplify the differences of weights and is empirically set as 4 pixels in this paper. Having defined all the terms, we resort to graph cuts (Boykov et al., 2001) to minimize the objective function. The regrouping result corresponding to the minimum of the objective function is shown in Fig. 6(c). Comparing Fig. 6(a) and (c), we can observe that almost all mistakes have been corrected.

4.1.3. Space plane estimation and trimming

For each LS match group $\mathcal{G}_i \in \mathcal{S}$, triangulating all the pairs of corresponding LSs obtains a group of 3D LSs, $\mathcal{L}_i$. All 3D LSs in $\mathcal{L}_i$

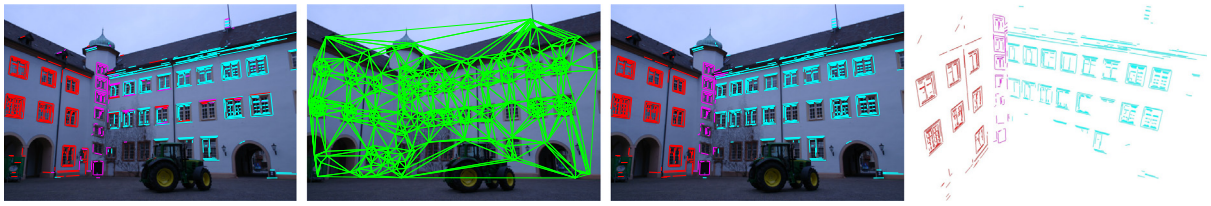

Fig. 6. An example used to illustrate some important steps of the proposed two-view based 3D LS reconstruction method. (a) The LS match grouping result before the refinement procedure. The grouping result of the matched LSs in the first image is shown. LSs drawn in the same color are regarded to belong to the same group. (b) The Delaunay triangles constructed using the middle points of matched LSs in the first image. (c) The LS match grouping result after applying the refinement procedure. (d) The final 3D LS reconstruction result for the scene. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are supposed to lie on a space plane $\mathbf{P}_i$. We estimate $\mathbf{P}_i$ from the endpoints of 3D LSs in $\mathcal{L}_i$ using RANSAC. Next, we recompute the homography induced by $\mathbf{P}_i$ and use it to check if LS matches in $\mathcal{G}_i$ are consistent with it or not. We accept $\mathbf{P}_i$ as a correct plane only when the majority (0.8 in this paper) of LS matches in $\mathcal{G}_i$ are consistent with it. This step can ensure only robust space planes are kept for further processing because an accidentally formed LS match group would not result in a robust space plane such that the majority of the LS matches are consistent with its induced homography. If $\mathbf{P}_i$ is accepted, the final reliable 3D LSs corresponding to LS matches in $\mathcal{G}_i$ can be obtained simply by back-projecting matched LSs from one image onto $\mathbf{P}_i$, producing an updated $\mathcal{L}_i$. After processing all LS match groups in $\mathcal{S}$, we obtain a space plane set $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^{N_p}$, the corresponding 3D LS set $\hat{\mathcal{L}} = \{\mathcal{L}_i\}_{i=1}^{N_p}$, and the updated LS match group set $\mathcal{S} = \{\mathcal{G}_i\}_{i=1}^{N_p}$.

To remove some falsely reconstructed 3D LSs brought by a few falsely grouped matches that still exist after the refinement procedure, we intersect adjacent 3D planes, trim each plane at the intersection and keep the half plane on which there are more 3D LSs than those on the other half plane. It is reasonable to do so because only a minor (if any) fraction of 3D LSs on a plane are falsely reconstructed and they are certain to lie on the opposite side (according to the intersection) of the correctly reconstructed majority. Illustration of this plane trimming strategy is shown in Fig. 7(a).

The way we determine the adjacency of space planes is as follows: We project all groups of 3D LSs in $\hat{\mathcal{L}}$ onto the first image and generate the corresponding 2D LS set $\hat{\mathcal{L}}^{2d} = \{\mathcal{L}_i^{2d}\}_{i=1}^{K}$. Refer to Fig. 7(b), for two space planes $\mathbf{P}_i, \mathbf{P}_j \in \mathcal{P}$, suppose their corresponding 2D LS sets are $\mathcal{L}_i^{2d}$ and $\mathcal{L}_j^{2d}$. Let the convex hulls determined by $\mathcal{L}_i^{2d}$ and $\mathcal{L}_j^{2d}$ be CH_i and CH_j, respectively. Let the convex hull determined by both $\mathcal{L}_i^{2d}$ and $\mathcal{L}_j^{2d}$ be CH_w (the region outlined by dashed red line in Fig. 7(b)). If there exists a third 2D LS set $\mathcal{L}_m^{2d} \in \hat{\mathcal{L}}^{2d}$, which determines a convex hull CH_m that has a big overlapping ratio (0.6 in this paper) with CH_w, we deem there is a third space plane lying between $\mathbf{P}_i$ and $\mathbf{P}_j$, and do not regard $\mathbf{P}_i$ and $\mathbf{P}_j$ to be adjacent. Otherwise, we treat $\mathbf{P}_i$ and $\mathbf{P}_j$ as adjacent planes. This strategy makes sense because it is very likely to be true in structured scenes that two space planes are adjacent if there is not a third space plane between them.

In Fig. 6, we show the final 3D LSs for the scene in sub-figure (d). We can see that the three main planes in the scene are correctly recovered and all 3D LSs are well reconstructed and correctly clustered w.r.t. the space planes they lie.

4.2. Multi-view based 3D line segment reconstruction

If more than two images are available, it is easy to extend the above two-view based 3D LS reconstruction method to deal with multiple views. We just need to combine the results obtained from every adjacent pair of images. In details, we begin with using the first two images to generate a set of space planes $\mathcal{P}_1$, and the corresponding set of 3D LSs $\hat{\mathcal{L}}_1$. The two sets are used to initialize the global space plane set $\mathcal{P}^g = \mathcal{P}$, and the global 3D LS set $\hat{\mathcal{L}}^g = \hat{\mathcal{L}}_1$, for the whole scene. The subsequent images are used to refine the two global sets. Each subsequent image is used to reconstruct 3D LSs with its previous image (we assume the input images are aligned), generating a new space plane set $\mathcal{P}_i$ and a new 3D LS set $\hat{\mathcal{L}}_i$. For each space plane $\mathbf{P}_{ij} \in \mathcal{P}_i$, suppose its corresponding 3D LS set is $\mathcal{L}_{ij} \in \hat{\mathcal{L}}_i$, if $\mathcal{L}_{ij}$ is consistent with a space plane $\mathbf{P}_m \in \mathcal{P}^g$, whose corresponding 3D LS set is $\mathcal{L}_m$, we merge $\mathbf{P}_{ij}$ and $\mathbf{P}_m$ into a new space plane using 3D LSs in $\mathcal{L}_{ij}$ and $\mathcal{L}_m$; we next project 3D LSs in $\mathcal{L}_{ij}$ and $\mathcal{L}_m$ onto the new space plane. Otherwise, we regard $\mathbf{P}_{ij}$ as a new plane and insert it into $\mathcal{P}^g$, and meanwhile insert $\mathcal{L}_{ij}$ into and $\hat{\mathcal{L}}^g$.

Algorithm 1. 3D Line Segment Reconstruction

Input: Images $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^{N} (N \geqslant 2)$, line segment matches $\hat{\mathcal{M}} = \{\mathcal{M}_i\}_{i=1}^{N-1}$
Output: 3D line segments $\hat{\mathcal{L}}^g$, space planes $\mathcal{P}^g$

```
Initialize L̂^g = ∅, P^g = ∅
for each M_i ∈ M̂ do
    Estimate local homographies H_i using M_i.
    Group line segment matches in M_i using H_i into clusters as S_i = {G_j}_{j=1}^{M}.
    Refine S_i through multi-label optimization.
    for each G_j ∈ S_i do
        Estimate the corresponding space plane P_j.
        Project line segments in G_j onto P_j and obtain 3D line segment set L_j.
        if P_j can be merged with a space plane P_m ∈ P then
            Merge P_j and P_m, update P^g and L̂^g.
        else
            Insert L_j into L̂^g, and P_j into P^g.
        end if
    end for
end for
Remove duplications in L̂^g.
```

After processing all images, there would exist a considerable number of duplications in $\hat{\mathcal{L}}^g$ because a space LS can be visible in multiple views and be reconstructed in multiple times. We need to remove these duplications. Since 3D LSs in our case are organized according to space planes, the duplications of a 3D LS must lie on the same space plane. We can therefore conduct duplication removal plane by plane in 2D space. For each space plane $\mathbf{P}_i \in \mathcal{P}^g$,

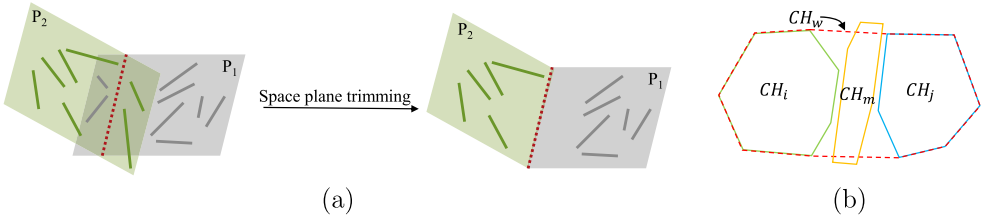

Fig. 7. Illustration of the strategy of removing falsely reconstructed 3D LSs. (a) Adjacent space plane intersection and trimming. (b) Finding adjacent space planes.

we project 3D LSs on it to a 2D plane $\mathbf{P}_i^{2d}$. For a LS $\mathbf{l}_m$ on $\mathbf{P}_i^{2d}$, we search its neighbors in a band around it. The band has the width equaling to the length of $\mathbf{l}_m$ and the height of 6 pixels (3 pixels in both sides of $\mathbf{l}_m$) in this paper. A LS $\mathbf{l}_n$ is regarded as a neighbor of $\mathbf{l}_m$ if it meets the two condition: First, both its two endpoints drop in the band around $\mathbf{l}_m$. Second, the direction difference between $\mathbf{l}_m$ and $\mathbf{l}_n$ is less than 5°. In this way, we obtain a set of neighbors for $\mathbf{l}_m$. All neighbors of $\mathbf{l}_m$ and $\mathbf{l}_m$ itself are merged into a single LS. After that, we project the merged new LSs from $\mathbf{P}_i^{2d}$ back to $\mathbf{P}_i$.

The above duplication removal strategy has advantages over those of some existing methods because it is easier and more reliable for us to define which LSs are adjacent enough to be merged into one. We only need to search in the band around a LS to find its possible duplications in a 2D plane, rather than in a cylinder in 3D space as that done by Jain et al. (2010) and Hofer et al. (2014). Therefore, the cases are rare in our method that the 3D reconstructions of multiple scene LSs are falsely regarded as the duplications of one scene LS, and one scene LS is reconstructed with multiple 3D representations. This benefits our method on delivering more accurate details of scenes.

Algorithm 1 outlines the main steps of the proposed method.

5. Experimental results

The experimental results of the proposed LS matching algorithm and 3D LS reconstruction algorithm are presented in this section first, followed by some discussions about the parameter settings of the two algorithms.

5.1. Line segment matching results

5.1.1. Natural scene images

We employed a recent line segment matching benchmark dataset (Li et al., 2016a) to evaluate our method on natural scene images. The benchmark dataset comprises of 15 image pairs characterized by various image transformations and scene types captured, 30 pairs of LS sets extracted from the 15 image pairs using two LS detectors (LSD (Grompone et al., 2010) and EDLines (Akinlar and Topal, 2011)), and the ground truth matches among all the 30 pairs of LS sets.[2] The 15 image pairs are shown in Fig. 8. We took as input all the 15 image pairs and the corresponding LS sets extracted by LSD for our method and the other two state-of-the-art ones. The comparative results are shown in Fig. 9. The LS sets extracted by EDLines were not experimented because it was reported that the matching results are similar when replacing LSs extracted by LSD with that by EDLines (Li et al., 2016a).

From Fig. 9, we can see that the three compared methods vary their relative ranks in the three performance evaluation measures: *recall*, i.e., the ratio of the number of correct matches and the number of ground truth matches; *accuracy*, i.e., the ratio of the number of correct matches and the number of obtained matches; *F-Measure* $= \frac{2\times accuracy\times recall}{accuracy+recall}$. Statistically, among the 15 image pairs, LJL (Li et al., 2016b) gains the highest recall scores in 9 of them, while our method achieves the best among the rest 6 image pairs. LPI (Fan et al., 2012) does not achieve the best recall score in any image pair, but it dominates the accuracy score and achieves the highest in 11 of the 15 image pairs. *F-Measure* reflects the matching performance of a method from both *recall* and *accuracy* aspects. Fig. 9 shows that our method, LJL and LPI win the best in 6, 7 and 2 image pairs, respectively. Based on these observations, we can conclude that our algorithm is slight inferior to LJL in term of matching performance, but significantly better than LPI. As for running time, our method owns overwhelming advantages over the other two methods: our method uses much less time than the other two in 14 of the 15 image pairs. We stated in the introduction section that our proposed algorithm targets to solve the low efficiency problem of LJL; these results convincingly substantiate our statement and our algorithm indeed tremendously improves the efficiency of LJL, with only a minor sacrifice of the matching performance.

The good performance of our algorithm owes to the local region extractor's robustness in generating repeatable local regions around V-junctions from images and SIFT's capability in powerfully describing the extracted local regions. Meanwhile, the inheritance of many benefits from LJL, like iteratively refining initial junction matches and matching individual LSs by local homographies, also contributes to our good performance. Unlike LJL which deals with scale changes among images to be matched through some sophisticated and time-consuming scale change simulation procedures, we extract scale invariant local regions in the original images and thus achieve significant improvements on the matching performance.

One may have noticed that the running time of our method on image pair (i) is anomalously high. This is due to the dense distribution of the extracted LSs. The two images have the size only of 800 × 600, but 1071 and 1016 LSs are extracted in them, respectively. Meanwhile, these extracted LSs crowd in the images, resulting in a great number of V-junctions being constructed in both images (11,851 in one image and 11,444 in the other). Constructing, describing and matching such two large sets of V-junctions definitively impose a great computation burden on the algorithm. However, it had been proved previously (Li et al., 2016b) that the LS matching efficiency for this type of scene can be tremendously improved by reducing the sizes of the impact zones of LSs, without impairing the good matching performance. This is because the abundance of LSs in this type of scenes makes it possible to generate sufficient numbers of V-junctions to match LSs even LSs has small impact zones.

The running time of our algorithm does not depend on the quantities of extracted LSs, but on the quantities of constructed V-junctions. Suppose the numbers of V-junctions constructed in two images are N_v and N'_v, and the number of matches found from them is N_c, the time complexity of each step of the proposed method is as follows: The time cost of the local region extraction

[2] Available at http://kailigo.github.io/projects/LineMatchingBenchmark.

Fig. 8. The 15 image pairs, referred later as (a)–(o), from a line segment matching benchmark dataset (Li et al., 2016a).

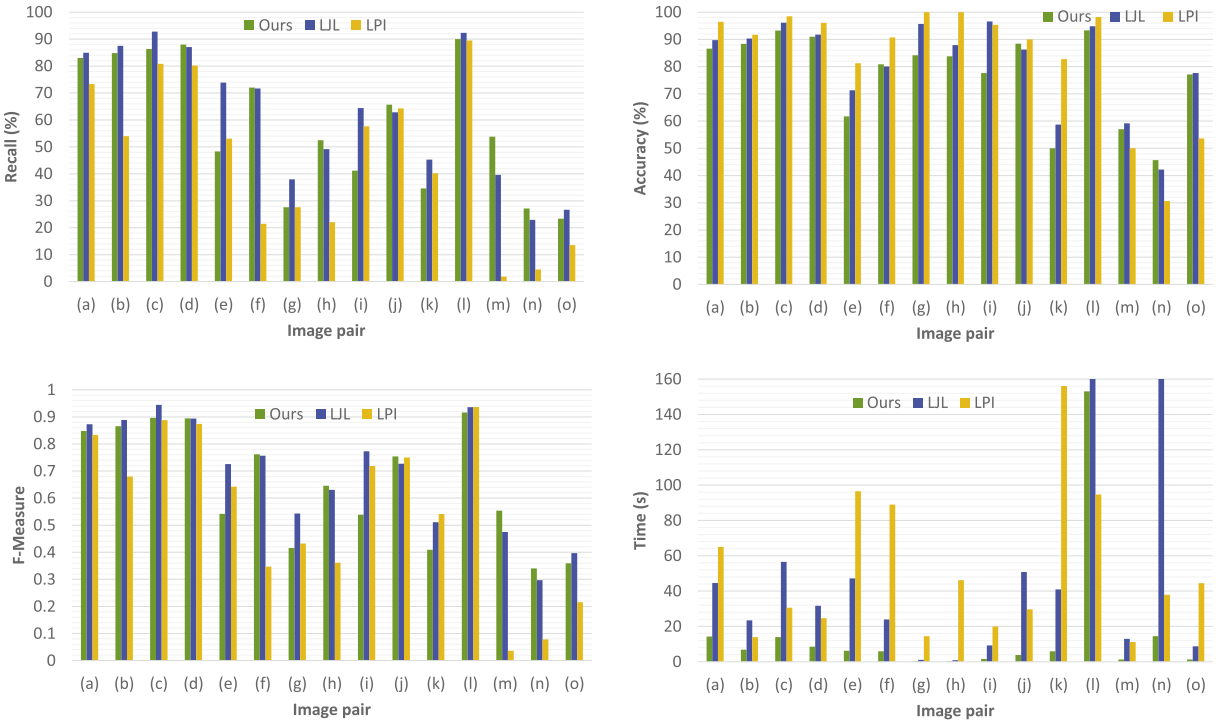

Fig. 9. The comparative results of our proposed LS matching algorithm and the other two. Note that in the right bottom sub-figure, bins with heights above 160 were truncated to limit the range of the heights.

and description is $O(N_v + N'_v)$. The time complexity of the V-junction matching step is $O(d_v N_v N'_v)$, where d_v is the dimension of the feature description vectors. The cost for estimating homographies from the N_c V-junction matches is $O(N_c)$, and the cost for matching individual LSs from two images is $O(k_l M_l)$, where k_l is the number of groups each individual LS is assigned to during the individual LS matching stage, and M_l is the number of individual LSs to be matched in the first image.

5.1.2. Aerial images

Sun et al. (2015) proposed an algorithm, referred later as PHLM, that targeted to match LSs in aerial images. They concluded that in this special type of images, their algorithm performed better than state-of-the-art (at that time) LS matching methods. It is well known that in most cases, feature matching on aerial images is easier to be performed than that on natural scene images because of the relatively little image distortions and relatively simple image transformations. We cannot employ the benchmark dataset shown in Fig. 8 to evaluate PHLM because it requires the 3D points for the captured scene and point matches among the images as input to serve for LS matching, whereas none of which are provided in the benchmark dataset. Besides, a component of PHLM, the case II assumes scene LSs are under terrain plane or are parallel to that plane; this strong assumption holds only for some aerial images. For these two reasons, we have to use aerial images that can be processed by PHLM to make a fair comparison between our matcher and PHLM. We took as input the same images[3] used by Sun et al. (2015) and employed also LSD for extracting LSs.

The matching results of our matcher and PHLM are shown in Table 1. In Fig. 10, we visualize our matching results. Please note that the matching data of PHLM in Table 1 is from the authors' paper. We can see that on all the four pairs of aerial images, our matcher produced much more matches with higher accuracies than PHLM. Our matcher used less than 2 s[4] on all image pairs, evidencing our good matching efficiency. Note that we do not provide the running time of PHLM in Table 1. Sun et al. (2015) did report in the paper the running time on these image pairs, but since the time was measured on their machine, while the listed running time of our matcher was measured on our machine, it is therefore of no meaning to make a comparison between the reported running time of PHLM with ours.

5.2. Point matching results

Another benefit of the proposed LS matching method is that while it generates LS matches, it also produces point matches. This benefit would make the proposed method favorable in some upper-level applications, such as SLAM (Engel et al., 2014) and 3D scene modeling (Sinha et al., 2009), where exploiting both point matches and LS matches was proved to produce better results. Our method owns this benefit because it matches some LSs from images by effectively matching the V-junctions they formed through extracting scale and affine invariant local regions for V-junctions. In fact, we found that our method is quite robust for obtaining point matches. Fig. 11 visualizes the point matches obtained by the proposed method on image pairs characterized by some extreme image transformations or the scarce textures of the captured scenes; the point matches obtained by SIFT[5] on the same images are also visualized as a comparison. It is easy to observe that our proposed method produced much more point matches than SIFT, with much higher accuracies as well in all the image pairs. (The optical flows of the matched points of our method in the first image are more consistent with each other in each of the image pairs than those of SIFT. It is sure that the more consistent the optical flows of the matched points are, the higher the matching accuracy is.)

5.3. 3D line segment reconstruction results

This part presents the experimental results of the proposed 3D LS reconstruction method. The two-view based 3D LS reconstruction results are presented first, followed by the results obtained from images sequences. All images employed for experiments are from public datasets (Jain et al., 2010; Jensen et al., 2014; Strecha et al., 2008).

5.3.1. Two views

Figs. 1 and 6 show two sample results of the two-view based 3D LS reconstruction method. Fig. 12 shows four additional sets of such results. From all these figures, we can observe that the proposed method has successfully reconstructed a large part of 3D LSs lying on main planes of the scenes, and correctly clustered them w.r.t. their respective space planes. The main structures of the scenes are well outlined by the reconstructed LSs. These experiments prove the feasibility of the proposed two-view based 3D LS reconstruction strategy.

5.3.2. Multiple views

Two-view based 3D reconstruction is limited by the scope of the images; when multiple images are available, more scene content can be covered and the reconstruction results generated from each two images can complement with each other, contributing to more complete and detailed scene models. In this part, we present the experiments of our method on two image datasets, a synthetic image dataset and a real image dataset.

Synthetic images. The synthetic image dataset has $80 \times 3 = 240$ images photographing around a CAD model from the upper, middle and bottom viewpoints. An example image from the dataset is shown in Fig. 13(a). We employed for experiments only the 80 images for the middle round because we found in our initial experiments that the reconstruction result generated by our method based on the 80 images is negligibly different from that based on all 240 images, but the running time dropped significantly. The result model O_{80} is shown in Fig. 13(b). We can observe from O_{80} that the main planes in this scene are correctly recovered, and LSs in the scene are precisely reconstructed and correctly clustered w.r.t. the planes they lie. We overlapped O_{80} with the ground truth CAD model to qualitatively evaluate the reconstruction accuracy, as shown in Fig. 13(c). As we can see, the vast majority of the reconstructed LSs (in black) cling to or closely approach the ground truth model, which indicates the high reconstruction accuracy. To test the robustness of the proposed method for 3D reconstruction from a small number of images, we sampled from the 80 used images by taking one from every three images, producing a new image sequence containing 27 images. Taking as input this new image sequence, our method generated the 3D model O_{27} shown in Fig. 13(d). Comparing O_{27} with O_{80}, we can see that there is no significant difference between them, except some missing LSs on the roof and bottom of the captured house in O_{27}; LSs on the walls of the house are identically and completely reconstructed in both models. Besides, LSs in O_{27} are also correctly clustered w.r.t. their respective planes.

For comparison, we show in Fig. 13(e)–(g) the reconstruction models of a recent algorithm, Line3D++ (Hofer et al., 2016),[6] using the whole 240 images of the dataset (C_{240}), our used 80 images (C_{80}) and 27 images (C_{27}), respectively. We can see that the reconstruction result of Line3D++ degenerates dramatically as the number of used images decreases. Line3D++ is able to generate good result when plentiful images are available, but cannot guarantee it with a small number of images. Our method, on other hand, is much less dependable on the availability of abundant images. Comparing Line3D++'s best model C_{240} with our model O_{80}, we can see that although C_{240} presents more details at the bottom of the house, our model is much neater and contains less short LSs that are arbitrarily distributed,

[3] Image patches with size of 500×500 selected from large aerial images.

[4] The running time was measured on a 3.4 GHz Inter (R) Core(TM) processor with 12 GB of RAM.

[5] The implementation is from http://www.cs.ubc.ca/~lowe/keypoints/.

[6] Implementation is available in https://github.com/manhofer/Line3Dpp.

Table 1
LS matching results of our method and PHLM (Sun et al., 2015) on four pairs of aerial images. #TM, #CM and #FM denote the numbers of total matches, correct matches and false matches, respectively. "–" represents the data are unavailable.

		#TM	#CM	#FM	Accuracy (%)	Time (s)
Aerial	Ours	100	89	11	89.0	0.9
	PHLM	56	45	11	80.3	–
Dunhuan	Ours	111	102	9	91.9	0.8
	PHLM	70	64	6	91.4	–
Tongzhou	Ours	126	122	4	96.8	1.2
	PHLM	82	79	3	96.3	–
Toronto	Ours	160	146	14	91.3	1.9
	PHLM	123	110	13	89.4	–

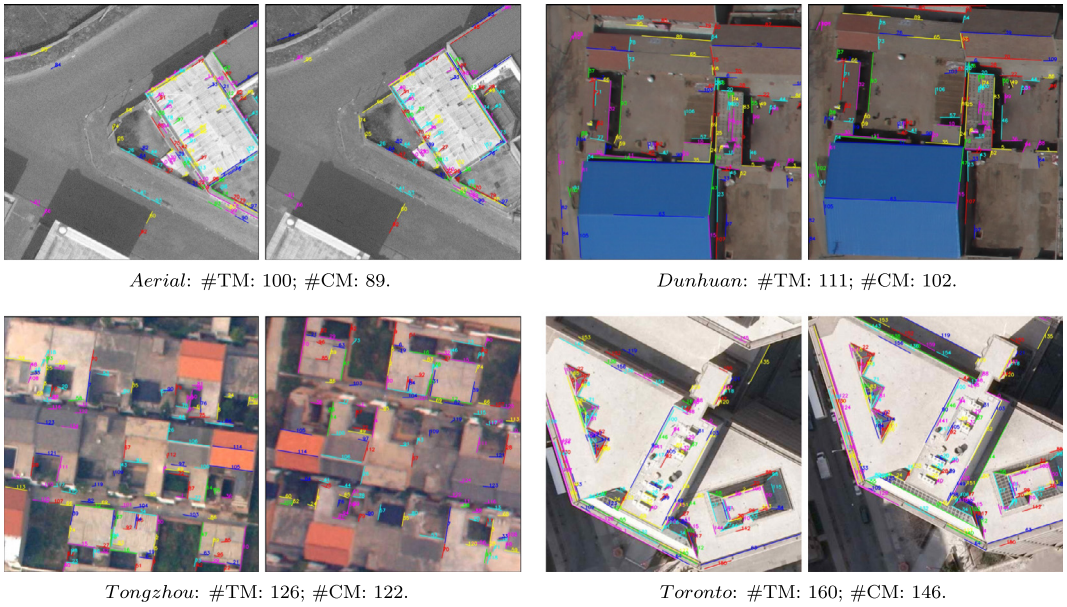

Fig. 10. Results of the proposed LS matching algorithm on four pairs of aerial images. #TM and #CM denote the numbers of total matches and correct matches, respectively. In each pair of images, two LSs in correspondence are drawn in the same color in both images and labeled with the same number at the middles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which, to some extent, indicates our model is a better wire-frame model for the scene. Besides, through carefully inspection, we can observe that for some scene LSs, C_{240} presents several duplications, while these cases are rare in our model. This proves the benefit of our duplication removal strategy.

To quantitatively evaluate the reconstruction accuracy, following previous works, we calculated the Hausdorff distances between densely sampled points along 3D LSs in our models and the ground truth CAD model, and computed the Mean Error (ME) and Root Mean Square Error (RMSE). We do not directly compare our measure data with those of Line3D++ because Line3D++ is based on the point clouds and camera parameters generated by some existing SFM systems, whose outputs are under arbitrary coordinates. 3D models generated by Line3D++ are hence inherently under the input arbitrary coordinates. It is thus not a trivial thing to evaluate models generated by Line3D++ because the underlying coordinates are inconsistent with that of the ground truth model. Alternatively, since Line3D++ is directly promoted from the same authors' two earlier algorithms (Hofer et al., 2013, 2014) and the two predecessors do not rely on SFM results, a comparison between our measure data with the report data of the two predecessors is also meaningful.[7] Meanwhile, we will immediately show that this indirect comparison does not affect us to reach a conclusion about the accuracies of our models and those of Line3D++.

Table 2 shows the measure data. We can see that when we set the cutoff distance threshold (distance values greater than this threshold are treated as gross errors and excluded for ME and RMSE calculations) $\rho = 1.0$, as that applied in ILGC (Hofer et al., 2013), the RMSEs of our two models O_{27} and O_{80}, are much better than the others, while the MEs are slightly inferior to that of ICGC. When we set $\rho = 0.6$ as that used in LBR (Hofer et al., 2014), our two models are better than that of EGCC (Jain et al., 2010), but worse than both those of ILGC and LBR. Since Line3D++ is promoted from ILGC and LBR, its generated model is supposed to be of even higher accuracy. It is thus reasonable to infer that the reconstruction accuracy of C_{240} is better than our models. But as can be obviously seen from Fig. 13, it is unlikely that the reconstruction accuracies of C_{80} and C_{27} are better than our two models, O_{80} and O_{27}, where the same numbers of images were fed to both algorithms. Therefore, we can reach the conclusion that Line3D++ can produce 3D models with higher reconstruction accuracy than our method, when plentiful images are available, but in the cases that there are only a small number of images, our method produces more accurate 3D models.

Real images. The real image dataset contains 30 images. Fig. 14 shows the result models of our method and Line3D++ generated from these images. As we can see, in our model, the 3D LSs lying on the main planes of the scene are well reconstructed; the details

[7] The authors of Line3D++ made the source code of Line3D++ publicly available, but did not do so for its predecessors. So, we can only compare our data with those reported in the papers.

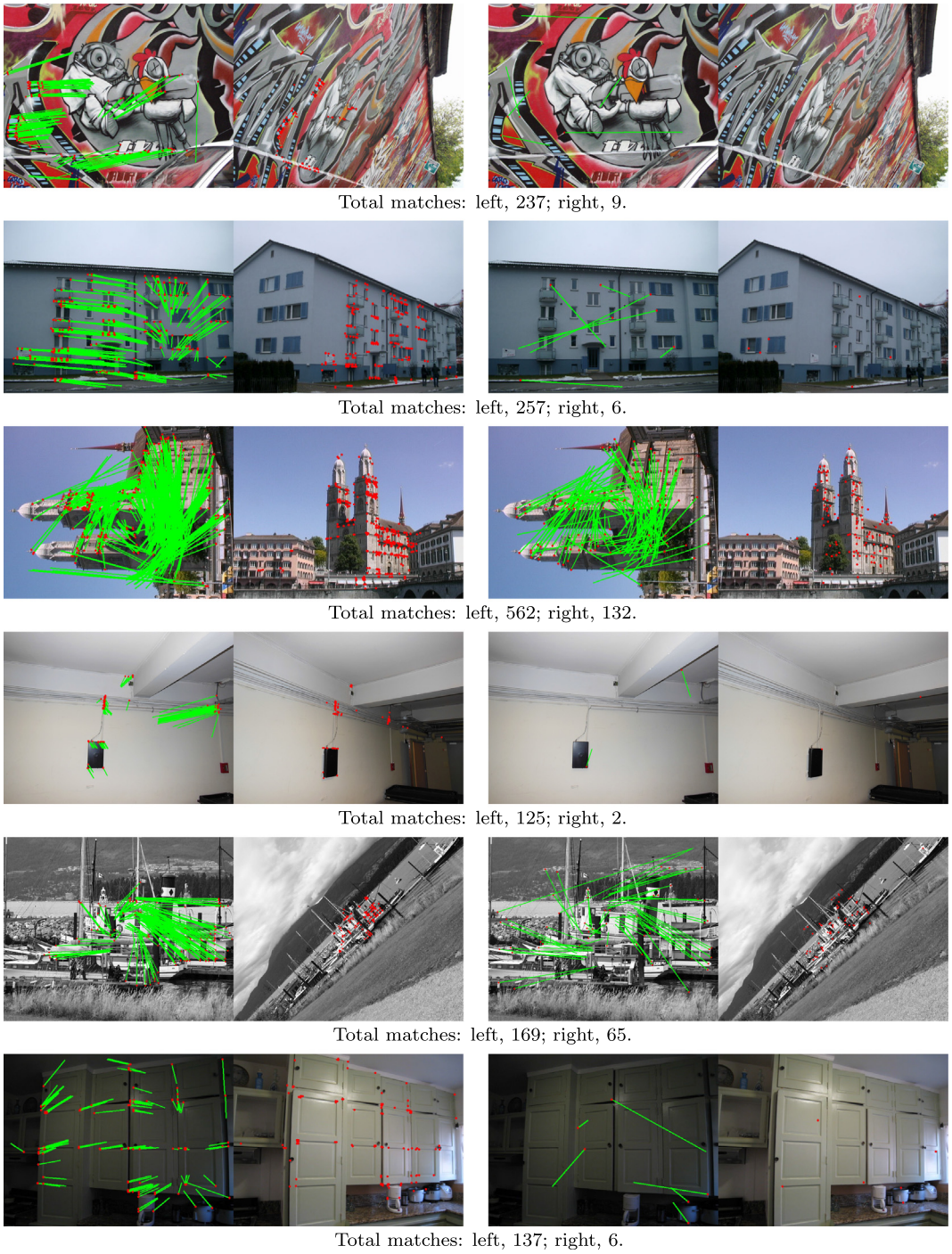

Fig. 11. Point matching results of our method (left column) and SIFT (right column) on some image pairs. Red dots are the matched points; green lines represent the optical flows of matched points in the first images. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the scene are precisely presented (see the bricks and windows of the selected dashed elliptical region shown in Fig. 14(a)). Our method failed to reconstruct 3D LSs on the main planes of this scene shown in the selected rectangle region in Fig. 14(b). This is because only several LSs were extracted on these two planes and even fewer LS matches were obtained. Our method is unable to reliably estimate a space plane when the quantity of LS matches induced by 3D LSs on the plane is too small, and hence incapable of obtaining the 3D LSs on it. Comparing with the model generated by Line3D++, our model is obviously much more complete and detailed.

Running time and limitations. The 3D LS reconstruction algorithm is currently implemented based on MATLAB. The unrefined codes took 631s on the 80 synthetic images and 1021s on the real image dataset on a 3.4 GHz Inter (R) Core(TM) processor with 12 GB of RAM. It is expected that the code can be substantially accelerated after refinements and being reimplemented in C++.

As stated above, this paper targets to reconstruct 3D LSs in structured scenes that comprise of a set of planes. Some strategies used in the two proposed algorithms are also specially designed for this targeted scene type. For example, we match V-junctions from images by describing local regions extracted around V-junctions

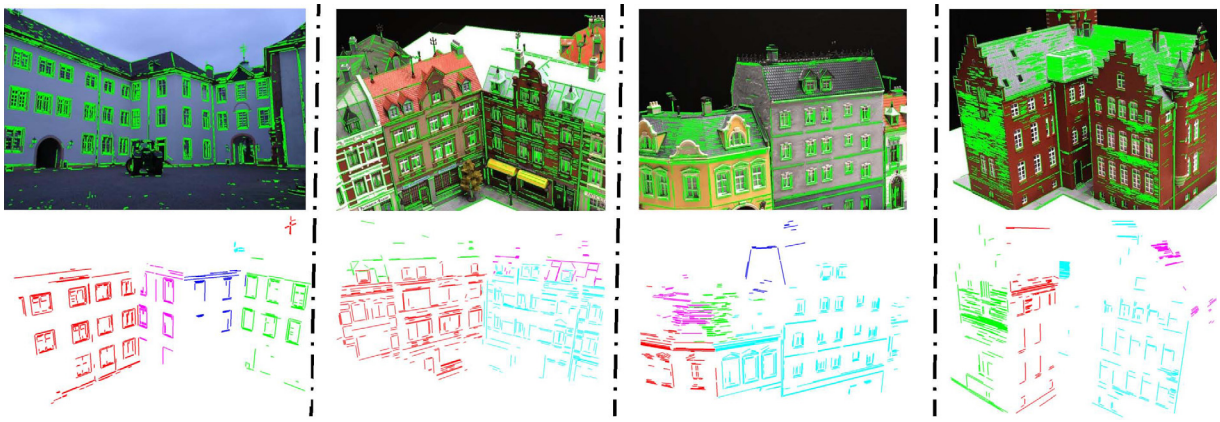

Fig. 12. Two-view based 3D LS reconstruction results. The top row shows the first images used for 3D LS reconstruction and the extracted LSs; the bottom row shows the obtained 3D LSs.

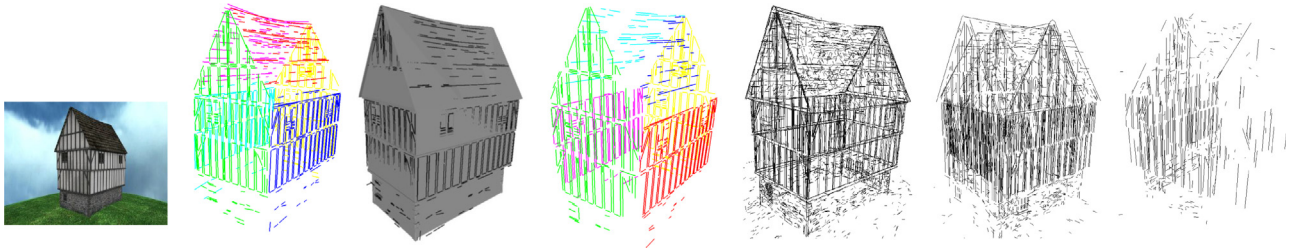

Fig. 13. 3D LS reconstruction results on a synthetic image dataset. (a) One of the used images. (b) The 3D model (referred later as O_{80}) obtained by the proposed method using 80 images. (c) The overlapping result of O_{80} with the ground truth scene model. (d) The 3D model (O_{27}) obtained by the proposed method using 27 images. (e)–(g) The 3D models generated by Line3D++ (Hofer et al., 2016) using 240, 80, and 27 images, respectively. The three models will orderly be referred later as C_{240}, C_{80} and C_{27}.

Table 2
The Mean Error (ME) and Root Mean Square Error (RMSE) of the reconstruction results obtained by our method, EGCC (Jain et al., 2010), ILGC (Hofer et al., 2013), and LBR (Hofer et al., 2014) on a synthetic dataset. "–" denotes the data are unavailable.

	$\rho = 1.0$		$\rho = 0.6$	
	ME	RMSE	ME	RMSE
EGCC	0.162	0.291	0.137	0.189
ILGC	0.065	0.196	0.044	0.080
LBR	–	–	0.029	0.046
O_{27}	0.077	0.114	0.075	0.104
O_{80}	0.89	0.135	0.082	0.109

with SIFT descriptor (Section 3.1.2). This strategy works only when the extracted local region around a V-junction is a solid planar patch in physical space because only in this situation, its corresponding local region can possibly be extracted and described in images captured from different viewpoints; otherwise, the surrounding of the V-junction shall change significantly even with slight viewpoint changes. For instance, for wiry objects (e.g., power pylons (Hofer et al., 2013)), what captured in images can change dramatically as viewpoint changes because the background shifts from one place to others. The appearance description based local region descriptor, SIFT would certain fail in this scenario. So, the first limitation of our 3D LS reconstruction method is that it cannot produce satisfactory results in scenes that are not characterized by a set of solid planes. Besides, as observed from Figs. 13 and 14, our method is unable to recover small planes in scenes where LS features are scarce and consequently incapable of reconstructing 3D LSs on the planes. So, in scenes dominated by small patterns, our method would also fail.

5.4. Discussions about parameter settings

The two proposed algorithms both have a fair number of parameters, so that it is nontrivial to tune all the parameters to the optimal state. However, we found that the majority of the parameters are easy to tune and can be fixed in some initial experiments because some reasonable fluctuations of their values do not cause much variation on the results. In this part, we first briefly explain how we chose the values of tractable parameters whose values are easy to tune, and next introduce in details how we fixed two intractable parameters whose values are hard to tune.

5.4.1. Tractable parameters

ϵ_1, the threshold for the cross angle difference of V-junction correspondences, is set as 30° in this paper. We know that the cross angle of the two LSs forming a V-junction remains unchanged with image translation, rotation and scale changes, and only changes moderately with great viewpoint changes. So, the cross angle dif-

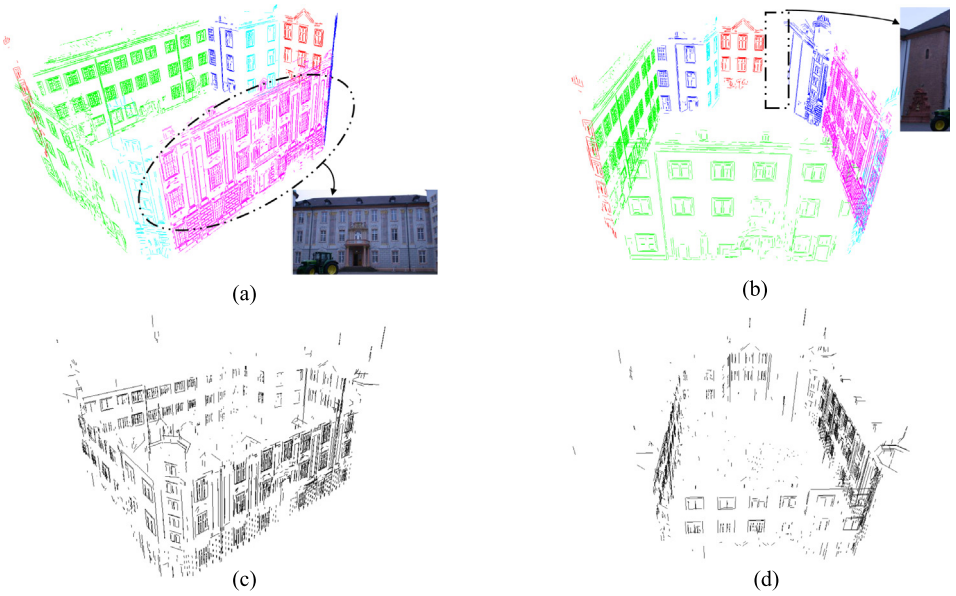

Fig. 14. The 3D LS reconstruction results of the proposed method and Line3D++ on a real image dataset. The top row shows our result model from two different viewpoints, while the bottom row shows that of Line3D++.

ference of two V-junction correspondences from two images shall be a fairly small angle in most cases. In this paper, we set ϵ_1 as a relatively big value $\epsilon_1 = 30^\circ$, to validate this constraint on images with great viewpoint changes. With the same idea, we set $\epsilon_2 = 20^\circ$, the threshold for the angle difference of two LS correspondences relative to those of their neighbors when matching LSs in individuals (Section 3.2.2). Other angle-related thresholds were also set in the similar way.

k_l, the number of groups each LS is assigned to when matching LSs in individuals (Section 3.2), is set as 4 in this paper. As mentioned before, we redundantly assign each individual LS into multiple groups to ensure potential LS correspondences from two images to be evaluated as least one time. This is meaningful because we match individual LSs between corresponding groups; once two LS correspondences fail to be assigned into any pair of corresponding groups, they will never be evaluated and thus cannot be matched. For complex image transformations, like wide-baseline viewpoint change, a big k_l is required, while a small k_l is good enough for simple image transformations, like rotation and scale changes. Simply choosing a relatively big k_l, as we did in this paper, is always safe, but the cost is that we need more time to evaluate some pairs LSs repeatedly. In similar way, we set parameter $K = 10$, which is the number of adjacent V-junctions we collect to apply the topological distribution constraint (Section 3.1.3), when matching V-junctions.

All the ratio (proportion) thresholds are set as values ranging from 0.5 to 0.8 in this paper. We use these thresholds to force the majority of a group samples meet certain constraints. It is in fact not that crucial how majority the inlier samples are. In other word, it is not a big deal when we vary the values of these ratio thresholds from 0.5 to 0.8.

Other tractable parameters we have not mentioned above were all fixed in the similar ways. All these tractable parameters are relatively easy to tune; we in fact did not change their values after testing them in several image pairs.

5.4.2. Intractable parameters

Different from tractable parameters discussed above, we found the two parameters are hard to tune: w and d_t. w determines how adjacently two LSs lie that they are intersected to form a junction, while d_t constraints on the description vector distances of V-junction correspondences.

A bigger w would result in more V-junctions, and consequently more V-junction matches. However, excessive V-junctions, especially when many of them cannot find correspondences in the other group of V-junctions, hamper the matching since more interferences are introduced. Besides, both more computation time and memory are required to match them. We adopted the strategy introduced by Mikolajczyk et al. (2005) to select a proper value for w by calculating the *repeatability* (a measure reflects the capability of a local region detector in extracting repeatable local regions in images to be matched) of extracted local regions for V-junctions from images. The famous local region detection and description datasets,[8] *graffiti*, *leuven*, *boat*, *bikes* and *ubc*, were employed for experiments. We sampled w from 5 to 30 at the step of 5. With every w, we calculated the repeatability scores of the extracted local regions in the first image and all the rest images in each of the above five datasets, and computed the average repeatability. The change of the average repeatability w.r.t. w is shown in Fig. 15(a). We can observe from this figure that the curve increases when w is less than 20, and is stable until w is bigger than 25, where the curve begins to drop. Thus, both 20 and 25 are proper values for w. To obtain less junctions and reduce computation time, $w = 20$ was selected.

d_t is the threshold for the description vector distance of V-junction correspondences. We had expected it to be very hard to tune. Surprisingly, we later found it is not that intractable and a reasonable fluctuations of its value do not cause big changes of the final LS matching results, except on images with great scale changes. The reason behind the tractability of d_t is that we only need to obtain some initial V-junction matches through evaluating the description vectors of V-junctions from two images to be matched. We do not need to maximize the match numbers and the accuracies because we will subsequently refine the initial V-junction matches under the guidance of epipolar line constraint and the topological distribution constraint. This refinement procedure would find back the missing V-junction matches due to a non-optimal threshold value d_t. So, a value of d_t is acceptable so long as

[8] Available in http://www.robots.ox.ac.uk/~vgg/research/affine/.

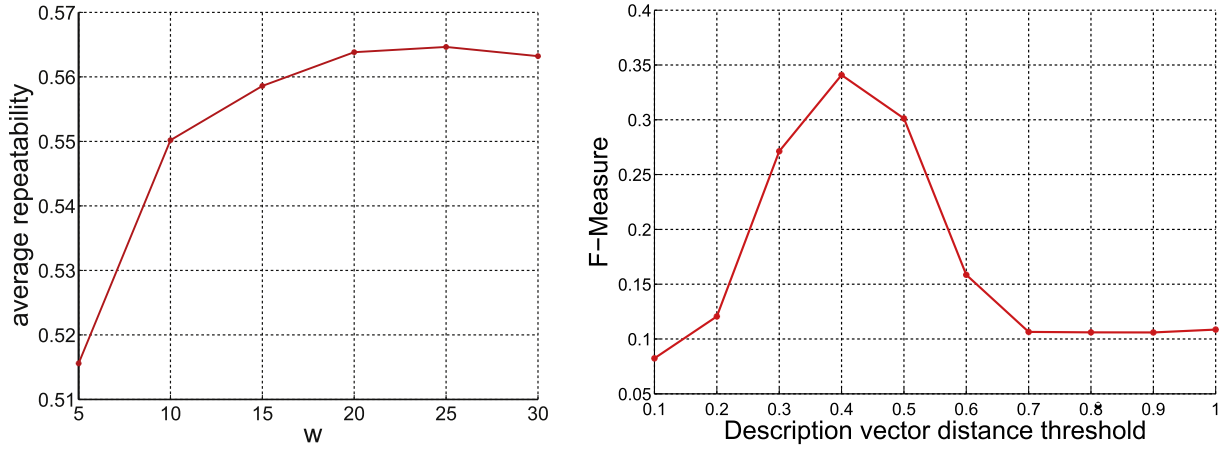

Fig. 15. Parameter tuning. (Left): The changes of the average repeatability with different values of parameter *w*. (Right): The changes of *F-Measure* w.r.t. the threshold for description vector distance.

it can result in a sufficient portion of correct V-junction matches that are able to be used to correctly estimate the epipolar geometry between two images. However, the tractability of d_t is not applicable to images with great scale changes. This is because our LS matcher is not so powerful to match LS from images with great scale changes. If d_t is not set as a fairly good value, our LS matcher might not be able to find a sufficient proportion of correct V-junction matches that enable a robust fundamental matrix estimation.

Based on the above observations, instead of setting d_t as the statistically optimal value after experimenting on a large set of images, we simply set d_t as the value that can bring in the best matching performance on images with scale change. The first and last images in the above-mentioned dataset *boat*, were utilized for this purpose. The extremely great scale change between the two images benefits us to select a proper value for d_t. Besides, the known global homography between the two images can help access the correctness of the obtained V-junction matches automatically and reliably. *F-Measure* of the obtained V-junction matches w.r.t. different values of d_t is shown Fig. 15(b). We can see that *F-Measure* reaches the maximum when d_t is set as 0.4. Therefore, $d_t = 0.4$ was adopted in the algorithm. An interesting observation from Fig. 15(b) is that when d_t is set to be greater than 0.7, *F-Measure* remains unchanged. This is because no more new matches, no matter correct or incorrect ones, can be added with a threshold greater than 0.7. Remember that when matching V-junctions, before evaluating description vector distances, we prefilter candidate matches by the cross angle difference constraint. It is this constraint that stops the increase of the matches.

6. Conclusions

We have presented in this paper a new system for 3D reconstruction based on LSs on images. The proposed LS matching algorithm and 3D LS reconstruction algorithm both utilize the two coplanar cues of image LSs that indicate their coplanarity in space: adjacent image LSs are coplanar in space in a high possibility, and corresponding image LSs shall be related by the same planar homography if they are coplanar in space. Based on these two cues, the proposed LS matching algorithm significantly improves the efficiency of existing methods through matching the V-junctions of adjacent LSs by extracting for each V-junction a scale and affine invariant local region. The 3D LS reconstruction method solves the ambiguities in 3D LS reconstruction through LS match grouping, space plane estimation and image LS back-projection. A Markov Random Field (MRF) based strategy is proposed to help more reliable LS match clustering. The benefit of the proposed 3D LS reconstruction algorithm is that it can use a small number of images to generate complete and detailed scene models.

Acknowledgments

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 Natural Science Foundation of China (Project No. 41271431), the National Natural Science Foundation of China under Grant 91438203, and the Jiangsu Province Science and Technology Support Program, China (Project No. BE2014866).

References

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R., 2011. Building Rome in a day. Commun. ACM 54 (10), 105–112.
Akinlar, C., Topal, C., 2011. Edlines: a real-time line segment detector with a false detection control. Pattern Recogn. Lett. 32 (13), 1633–1642.
Alshahri, M., Yilmaz, A., 2014. Line matching in wide-baseline stereo: a top-down approach. IEEE Trans. Image Process. 23 (9), 4199–4210.
Baillard, C., Schmid, C., Zisserman, A., Fitzgibbon, A., 1999. Automatic line matching and 3D reconstruction of buildings from multiple views. In: ISPRS Conference on Automatic Extraction of GIS Objects From Digital Imagery.
Bartoli, A., Sturm, P., 2005. Structure-from-motion using lines: representation, triangulation, and bundle adjustment. Comput. Vision Image Underst. 100 (3), 416–441.
Bay, H., Ess, A., Neubeck, A., Gool, L.V., 2006. 3D from line segments in two poorly-textured, uncalibrated images. In: International Symposium on 3d Data Processing, Visualization, and Transmission.
Bay, H., Ferraris, V., Van Gool, L., 2005. Wide-baseline stereo matching with line segments. In: Computer Vision and Pattern Recognition.
Boykov, Y., Veksler, O., Zabih, R., 2001. Efficient approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 36 (12), 1222–1239.
Chen, M., Shao, Z., 2013. Robust affine-invariant line matching for high resolution remote sensing images. Photogramm. Eng. Remote Sens. 79 (8), 753–760.
Delong, A., Osokin, A., Isack, H., Boykov, Y., 2012. Fast approximate energy minimization with label costs. Int. J. Comput. Vision 96 (1), 1–27.
Engel, J., Schops, T., Cremers, D., 2014. Lsd-slam: large-scale direct monocular slam. In: European Conference on Computer Vision.
Fan, B., Wu, F., Hu, Z., 2010. Line matching leveraged by point correspondences. In: Computer Vision and Pattern Recognition.
Fan, B., Wu, F., Hu, Z., 2012. Robust line matching through line-point invariants. Pattern Recognit. 45 (2), 794–805.
Furukawa, Y., Ponce, J., 2010. Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. 32 (8), 1362–1376.
Grompone, v.G.R., Jakubowicz, J., Morel, J.-M., Randall, G., 2010. Lsd: a fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Mach. Intell. 32 (4), 722–732.
Habib, A., Morgan, M., Lee, Y., 2002. Bundle adjustment with self–calibration using straight lines. Photogramm. Record 17 (100), 635–650.

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision. Cambridge University Press.
Hofer, M., Maurer, M., Bischof, H., 2014. Improving sparse 3d models for man-made environments using line-based 3d reconstruction. In: International Conference on 3D Vision.
Hofer, M., Maurer, M., Bischof, H., 2016. Efficient 3d scene abstraction using line segments. Comput. Vision Image Underst.
Hofer, M., Wendel, A., Bischof, H., 2013. Incremental line-based 3d reconstruction using geometric constraints. In: British Machine Vision Conference.
Jain, A., Kurz, C., Thormahlen, T., Seidel, H., 2010. Exploiting global connectivity constraints for reconstruction of 3d line segments from images. In: Computer Vision and Pattern Recognition.
Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanaes, H., 2014. Large scale multi-view stereopsis evaluation. In: Computer Vision and Pattern Recognition.
Kim, C., Manduchi, R., 2014. Planar structures from line correspondences in a manhattan world. In: Asian Conference on Computer Vision.
Kim, H., Lee, S., Lee, Y., 2014. Wide-baseline stereo matching based on the line intersection context for real-time workspace modeling. J. Opt. Soc. Am. A Opt. Image Sci. Vision-Opt. Image Sci. Vision 31 (2), 421–435.
Li, K., Yao, J., Lu, M., Yuan, H., Wu, T., Li, Y., 2016a. Line segment matching: a benchmark. In: IEEE Winter Conference on Applications of Computer Vision.
Li, K., Yao, J., Lu, X., Li, L., Zhang, Z., 2016b. Hierarchical line matching based on line-junction-line structure descriptor and local homography estimation. Neurocomputing 184, 207–220.
Lourakis, M.I.A., Halkidis, S.T., Orphanoudakis, S.C., 2000. Matching disparate views of planar surfaces using projective invariants. Image Vision Comput. 18 (9), 673–683.
Luong, Q.T., Vieville, T., 1996. Canonical representations for the geometries of multiple projective views. Comput. Vision Image Underst. 64 (2), 193–229.
Martinec, D., Pajdla, T., 2003. Line reconstruction from many perspective images by factorization. In: Computer Vision and Pattern Recognition.
Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L., 2005. A comparison of affine region detectors. Int. J. Comput. Vision 65 (1–2), 43–72.
Mishkin, D., Perdoch, M., Matas, J., 2013. Two-view matching with view synthesis revisited. In: Image and Vision Computing New Zealand.
Ok, A.Ö., Wegner, J.D., Heipke, C., Rottensteiner, F., Sörgel, U., Toprak, V., 2012a. Accurate reconstruction of near-epipolar line segments from stereo aerial image. Photogramm.-Fernerkundung-Geoinform. 2012 (4), 345–358.
Ok, A.Ö., Wegner, J.D., Heipke, C., Rottensteiner, F., Sörgel, U., Toprak, V., 2012b. Matching of straight line segments from aerial stereo images of urban areas. ISPRS J. Photogramm. Remote Sens. 74, 133–152.
Pham, T.T., Chin, T.J., Yu, J., Suter, D., 2014. The random cluster model for robust geometric fitting. IEEE Trans. Pattern Anal. Mach. Intell. 36 (8), 1658–1671.
Přibyl, B., Zemčík, P., Čadík, M., 2015. Camera pose estimation from lines using plücker coordinates. In: British Machine Vision Conference.
Ramalingam, S., Brand, M., 2013. Lifting 3d manhattan lines from a single image. In: International Conference on Computer Vision.
Schindler, G., Krishnamurthy, P., Dellaert, F., 2006. Line-based structure from motion for urban environments. In: International Symposium on 3d Data Processing, Visualization, and Transmission.
Schmid, C., Zisserman, A., 1997. Automatic line matching across views. In: Computer Vision and Pattern Recognition.
Sinha, S.N., Steedly, D., Szeliski, R., 2009. Piecewise planar stereo for image-based rendering. In: International Conference on Computer Vision.
Smith, P., Reid, I., Davison, A.J., 2006. Real-time monocular slam with straight lines. In: British Machine Vision Conference.
Snavely, N., Seitz, S.M., Szeliski, R., 2006. Photo tourism: exploring photo collections in 3d. ACM Trans. Graph. 25 (3), 835–846.
Snavely, N., Seitz, S.M., Szeliski, R., 2008. Modeling the world from internet photo collections. Int. J. Comput. Vision 80 (2), 189–210.
Spetsakis, M.E., Aloimonos, J., 1990. Structure from motion using line correspondences. Int. J. Comput. Vision 4 (3), 171–183.
Strecha, C., Hansen, W.V., Gool, L.V., Fua, P., Thoennessen, U., 2008. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: Computer Vision and Pattern Recognition.
Sun, Y., Zhao, L., Huang, S., Yan, L., Dissanayake, G., 2015. Line matching based on planar homography for stereo aerial images. ISPRS J. Photogramm. Remote Sens. 104, 1–17.
Taylor, C.J., Kriegman, D., 1995. Structure and motion from line segments in multiple images. IEEE Trans. Pattern Anal. Mach. Intell. 17 (11), 1021–1032.
Tuytelaars, T., Van Gool, L., 2004. Matching widely separated views based on affine invariant regions. Int. J. Comput. Vision 59 (1), 61–85.
Verhagen, B., Timofte, R., Van Gool, L., 2014. Scale-invariant line descriptors for wide baseline matching. In: IEEE Winter Conference on Applications of Computer Vision.
Wang, L., Neumann, U., You, S., 2009a. Wide-baseline image matching using line signatures. In: International Conference on Computer Vision.
Wang, Z., Wu, F., Hu, Z., 2009b. Msld: a robust descriptor for line matching. Pattern Recognit. 42 (5), 941–953.
Werner, T., Zisserman, A., 2002. New techniques for automated architectural reconstruction from photographs. In: European Conference on Computer Vision.
Wu, C., 2013. Towards linear-time incremental structure from motion. In: International Conference on 3D Vision.
Zhang, L., Koch, R., 2013. An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. J. Visual Commun. Image Rep. 24 (7), 794–805.
Zhang, L., Koch, R., 2014. Structure and motion from line correspondences: representation, projection, initialization and sparse bundle adjustment. J. Visual Commun. Image Rep. 25 (5), 904–915.
Zhang, Z., 1999. Flexible camera calibration by viewing a plane from unknown orientations. In: International Conference on Computer Vision.