3D Line Segment Reconstruction in Structured Scenes via Coplanar Line Segment Clustering

Kai Li, Jian Yao(✉), Li Li, and Yahui Liu

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, People's Republic of China
jian.yao@whu.edu.cn
http://cvrs.whu.edu.cn/

Abstract. This paper presents a new algorithm aiming for 3D Line Segment (LS) reconstruction in structured scenes that are comprised of a set of planes. Due to location imprecision of image LSs, it often produces many erroneous reconstructions when reconstructing 3D LSs by triangulating corresponding LSs from two images. We propose to solve this problem by first recovering space planes and then back-projecting image LSs onto the recovered space planes to get reliable 3D LSs. Given LS matches identified from two images, we estimate a set of planar homographies and use them to cluster the LS matches into groups such that LS matches in each group are related by the same homography induced by a space plane. In each LS match group, the corresponding space plane can be recovered from the 3D LSs obtained by triangulating all the LS correspondences. To reduce the incidence of incorrect LS match grouping, we formulate to solve the LS match grouping problem into solving a multi-label optimization problem. The advantages of the proposed algorithm over others in this area are that it can generate more complete and detailed 3D models of scenes using much fewer images and can recover the space planes where the reconstructed 3D LSs lie, which is beneficial for upper level applications, like scene understanding and building facade extraction.

1 Introduction

3D reconstruction from images has been a widely studied field of research and some remarkable works have been done through exploiting feature points extracted from images [1,9,30,37]. However, objects in man-made scenes are often structured and can be outlined by a bunch of LSs. It is therefore advantageous to get the 3D wireframe model of a scene by exploiting LSs on the images. For example, for the house shown in Fig. 1(d), our proposed 3D LS reconstruction method to be introduced generates the 3D model shown in Fig. 1(e) using only two images. It is easy to recognize the house from this 3D model, but hardly possible to achieve this from the extremely sparse point clouds obtained by some point based 3D reconstruction methods. Some works [12,28] also proved that 3D modeling by exploiting both feature points and line segments on images resulted in more accurate and complete results.

C.-S. Chen et al. (Eds.): ACCV 2016 Workshops, Part II, LNCS 10117, pp. 46–61, 2017.
DOI: 10.1007/978-3-319-54427-4_4

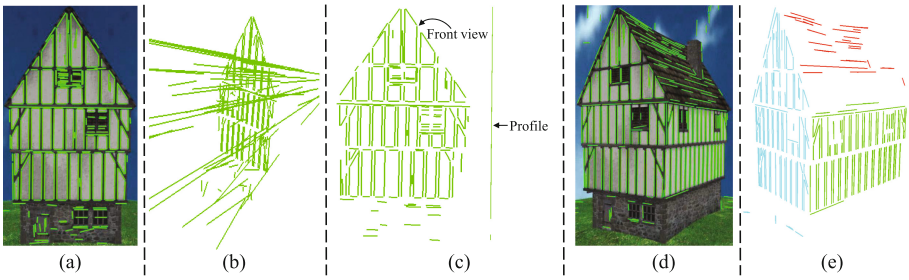


Fig. 1. An example showing problems in 3D LS reconstruction and the results obtained by our proposed solution. (a) An image of a roughly planar scene and the extracted LSs. (b) The 3D LS reconstruction result for the scene shown in (a) by triangulating LS correspondences from two images. (c) The 3D LS reconstruction result obtained by the proposed method for the scene shown in (a) by using two images. The front view and profile of the obtained 3D model are shown. (d) An image of a scene comprised of multiple planes, and the extracted LSs. (e) The 3D LS reconstruction result for the scene shown in (d) obtained by the proposed method using two images. Different colors are used to differentiate 3D LSs lying on different space planes. (Color figure online)

Despite of the above benefits of exploiting LSs on images for 3D reconstruction, it is often hard to reliably reconstruct 3D LSs because of the unstableness and low location accuracy of image LSs. Image LSs are the straight fittings of curve edges detected on images so that sometimes a 3D edge results in two straight fittings that are not precisely corresponding on two images. This fact makes it difficult to reliably reconstruct 3D LSs through triangulating corresponding LSs from two images. For example, to reconstruct 3D LSs in the scene shown in Fig. 1(a), all of which can approximately be regarded to lie on a single space plane, triangulating LS correspondences from two images capturing the scene produced 3D LSs shown in Fig. 1(b). Obviously, there are many mistakes. To solve this problem, some methods [12–14,17] resorted to exploit multiple (three or more) images photographing a scene to eliminate the mistakes. These methods involved establishing LS correspondences among multiple images, or some sophisticated hypothesizing-and-testing procedures. In this paper, we propose a simple, yet effective, solution to this problem. Our solution requires only two images and can be easily extended to more images, if available.

We observed from Fig. 1(b) that despite of many false reconstructions, the 3D model contains a big fraction of 3D LSs approaching to one space plane. This fact makes it possible to recover the space plane from the 3D LSs using RANSAC. Once the space plane being recovered, reliable 3D LSs can be obtained easily by back-projecting image LSs onto the space plane. With this idea, errors shown in Fig. 1(b) can be completely eliminated, as shown Fig. 1(c). The scene shown in Fig. 1(a) comprises of only one main space plane, which enables us to use all 3D LSs obtained by image LS triangulation to recover the space plane. But for scenes comprised of multiple planes, such as the one shown in Fig. 1(d), it is not clear which part of the obtained 3D LSs come from one plane, while some others come from another one. We need to cluster the 3D LSs according

to their coplanarity. Instead of directly analyzing the 3D LSs, we propose to achieve this by exploiting LS matches obtained from the images.

Our solution is based on the fact that the projections of 3D LSs from a space plane onto two images shall be related by the planar homography induced by the space plane. Given LS matches identified from two images, we first estimate a set of planar homographies and use them to cluster the LS matches into groups such that LS matches in each group are related by a homography induced by a space plane. Then, in each LS match group, the corresponding 3D LSs are supposed to come from the same space plane, so that the final reliable 3D LSs can be obtained as the single plane case shown in Fig. 1(a). To reliably cluster LS matches, we formulate to solve the LS match clustering problem by solving a multi-label optimization problem. With our solution, for the multi-plane scene shown in Fig. 1(d), the 3D model shown in Fig. 1(e) are obtained. We can observe that a big fraction of the scene LSs are correctly reconstructed and categorized according to the space planes they lie.

In summary, the major contributions of this paper are twofold: First, 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. Second, we formulate to solve the LS match grouping problem by solving a multi-label optimization problem.

2 Related Works

We divide existing 3D LS reconstruction methods into two categories: methods that require LS matching before 3D 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 the foundation of 3D LS reconstruction and camera calibration based on lines. A series of representations for a line in 3D space have been proposed. They are plücker coordinates [3,20,25], pair of points [2,10,11,22,29,36], pair of planes [11], a unitary direction vector and a point on a line [34], the intersections a line with two orthogonal planes [32], and a more recent one, Cayley representation [38]. 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 [16,27], piecewise planar scenes [28] and poorly textured scenes [4]. The prior knowledge of these scenes decreases reconstruction uncertainties and often benefits for remarkable results.

Some recent algorithms in this area attempt to free the reconstruction procedure from the heavy dependence on the LS matching procedure because it is hard to get reliable LS correspondences in some kinds of scenes, such as poorly textured indoor environments [21] and scenes containing wiry structures (e.g., power pylons [13]). Most of these methods adopted the strategy of firstly generating a set of 3D hypotheses for each extracted LSs, either by sampling the depths

of the endpoints of 3D LSs to camera centers [17], or triangulating putative LS correspondences after enforcing some soft constraints on the extract LSs [12,13], and then validating the hypotheses by projecting them back to images. In [26], a novel algorithm is proposed 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 3D LSs in this special type of scenes can only distribute in three orthogonal directions. This fact tremendously deceases the degrees of freedom when to reconstruct the scene LSs.

Our method 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 the one proposed by Kim and Manduchi [16], which also focuses 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. 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 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, as a differentiation with two views.

3.1 Two-View Based 3D Line Segment Reconstruction

Given two 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* [30,31], or some camera calibration methods [25,39]. Suppose LS matches obtained from $\mathbf{I}$ and $\mathbf{I}'$ by a LS matcher is $\mathcal{M} = \{(\mathbf{l}_m, \mathbf{l}'_m)\}_{m=1}^{M}$. The 3D LS reconstruction procedures begin with estimating a set of planar homographies.

Homography from Line Segment Matches. A planar homography $\mathbf{H}$ is determined by eight degrees of freedom, necessitating 8 independent constraints to find a unique solution. However, when the fundamental matrix $\mathbf{F}$ between the two images is known, $\mathbf{H}^\top\mathbf{F}$ is skew-symmetric [19],

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

The above equation gives five independent constraints on $\mathbf{H}$, and the other three are required to fully describe a homography. The fundamental matrix $\mathbf{F}$ can be obtained easily by using some point matching methods, or computing from the projection matrices of the two images [11], as they are known in our case.

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

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

where the 3D plane is represented by $\boldsymbol{\pi} = (\mathbf{v}^\top, 1)$ in the projective reconstruction with camera matrices $\mathbf{C} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{C}' = [\mathbf{A}|\mathbf{e}']$. A homography maps a point from one 2D plane to another 2D plane. For a line segment match $(\mathbf{l}, \mathbf{l}')$, suppose $\mathbf{x}$ is an endpoint of $\mathbf{l}$, $\mathbf{H}$ maps it to its correspondence $\mathbf{x}'$ 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

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

Arranging the above equations, we get

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

which is linear in $\mathbf{v}$. Each endpoint of a LS in a LS match provides one such equation. Two line segment matches, which totally provide four such equations, are sufficient to compute $\mathbf{v}$, and accordingly $\mathbf{H}$ from Eq. (2). If more such LS matches are available, as long as they are induced by 3D LSs from space plane $\boldsymbol{\pi}$, additional constraints can be used to help more robust homography estimation.

Homography from Point Matches (Optional). If point matches from the two images which are induced by 3D points also coming from space plane $\boldsymbol{\pi}$ are available, they can be incorporated into the above LS match based local homography estimation method. Note that point matches are optional for the proposed method, and they are used only to provide additional constraints for homography estimation. Suppose $(\mathbf{p}, \mathbf{p}')$ is a such point match, there exists $\mathbf{p}' = \mathbf{H}\mathbf{p}$. Replacing $\mathbf{H}$ using Eq. (2), we get

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

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

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

It holds when using Eq. (6) to form the scalar product with the vector $\mathbf{p}' \times \mathbf{e}'$

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

This equation is also linear in $\mathbf{v}$ and provides one constraint.

Line Segment Match Grouping. The last section presents how to estimate a local homography from (at least two) LS matches (and optional point matches when available) under the condition that they are induced by coplanar 3D LSs (or points). It is yet hard to determine which LS matches meet this condition

only from images. However, due to spatial adjacency, the projections of coplanar 3D LSs onto image planes are likely to be adjacent. Therefore, it is alternative to use spatially adjacent LS matches to estimate local homographies.

For every LS match $(\mathbf{l}_m, \mathbf{l}'_m) \in \mathcal{M}$, we search its spatial neighbors in $\mathcal{M}$ by finding matched LSs from $\mathbf{I}$ which are adjacent to $\mathbf{l}_m$. A matched LS from $\mathbf{I}$ which has at least one of its two endpoints dropping in the rectangle centered around $\mathbf{l}_m$ is regarded as a neighbor of $\mathbf{l}_m$, and the corresponding LS match is regarded as a neighbor of $(\mathbf{l}_m, \mathbf{l}'_m)$. For example, if matched LS $\mathbf{l}_n$ is found to be adjacent with $\mathbf{l}_m$, then LS match $(\mathbf{l}_n, \mathbf{l}'_n)$ is treated as a neighbor of $(\mathbf{l}_m, \mathbf{l}'_m)$. The rectangle around $\mathbf{l}_m$ has the width equaling to the length of $\mathbf{l}_m$ and the height of 20 pixels (10 pixels in both sides of $\mathbf{l}_m$) in this paper. When point matches are available, we can also find point match neighbors for $(\mathbf{l}_m, \mathbf{l}'_m)$ using the same strategy. Having found the neighbors for $(\mathbf{l}_m, \mathbf{l}'_m)$, we estimate the corresponding local homography using the method presented in the last section.

A set of homographies, $\mathcal{H} = \{\mathbf{H}_i\}_{i=1}^{H}$, can be obtained after processing all LS matches in $\mathcal{M}$. H denotes the number of homographies obtained and it is often smaller than the number of elements of $\mathcal{M}$ because we sometimes cannot find for a LS match even one neighbor, and a LS match alone is insufficient to define a unique homography.

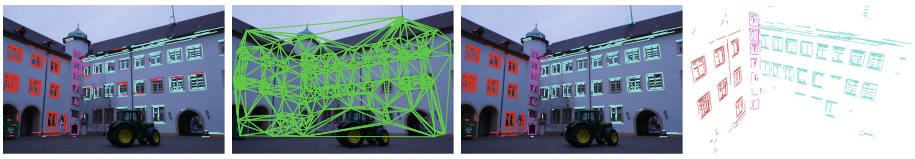
The projections of 3D LSs from a space plane onto two images would be related by the homogrpahy induced by the space plane. Based on this fact, we cluster LS matches in $\mathcal{M}$ using homographies in $\mathcal{H}$. For a LS match $(\mathbf{l}, \mathbf{l}') \in \mathcal{M}$, we find its most consistent homography matrix $\mathbf{H} \in \mathcal{H}$ 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 $\mathbf{I}'$.

After finding for each LS match in $\mathcal{M}$ a most consistent homography, some homographies in $\mathcal{H}$ are assigned with some LS matches from $\mathcal{M}$, forming a LS match group set $\mathcal{S} = \{\mathcal{G}_i\}_{i=1}^{N_s}$, where $\mathcal{G}_i$ denotes the i-th LS match group whose elements are from $\mathcal{M}$. Each LS match group in $\mathcal{S}$ is formed based on a homography, induced by a space plane. Next, we merge some groups in $\mathcal{S}$ to ensure LS matches induced by 3D LSs coming from the same space plane are clustered into only one group.

For two LS match groups, $\mathcal{G}_i$ and $\mathcal{G}_j$, suppose they are formed based on homographies, $\mathbf{H}_i$ and $\mathbf{H}_j$, respectively. If LS matches in $\mathcal{G}_i$ are *consistent* with $\mathbf{H}_j$, and the same goes for $\mathcal{G}_j$ and $\mathbf{H}_i$, 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



(a) (b) (c) (d)

Fig. 2. 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 used images 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 to define the adjacent relationship among the LS matches. (c) The LS match grouping result after applying the refinement procedure. (d) The final 3D LS reconstruction result for the scene. (Color figure online)

Eq. (8)) is a small value (2 pixels in this paper). After this, we obtain an updated LS match group set $\mathcal{S}$, in which the elements decrease significantly.

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 estimated homographies, such that some LS matches which should be assigned into one group but were clustered into another group mistakenly. This kinds 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. 2(a) shows an example of the LS match grouping result using the strategy presented above. We draw 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 have been correctly grouped. But, as we can see, a considerable number of them are mistakenly clustered.

We propose to refine the LS match grouping result by enforcing spatial smoothness constraint that requires LS matches induced by coplanar 3D LSs are more likely to be adjacent with each other. We formulate to solve the LS match grouping problem by solving a multi-label optimization problem and minimizing

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

where the data term, D_p measures the cost of an object p being assigned with the label l_p, and the smoothness term, $V_{p,q}$ encourages a piecewise smoothness labeling by assigning a cost whenever neighboring objects p and q are assigned with labels l_p and l_q, respectively. Specifically to our problem, 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 it, an adjacency graph among the LS matches needs to be constructed. Inspired by [8,24], which 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, as shown in Fig. 2(b). Under this adjacency graph, 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 a same label in a higher possibility. s is a constant to amplify the differences of weights. It is empirically set as 4 pixels in this paper. Having defined all the terms, we resort to graph cuts [5] to minimize the objective function. The refined LS match grouping result corresponding to the minimum of the objective function is shown in Fig. 2(c). Comparing Figs. 2(a) and (c), we can observe that almost all mistakes have been corrected.

Space Plane Estimation and Trimming. For a 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 $\mathcal{L}_i$ 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 $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^K$, and the corresponding 3D LS set $\hat{\mathcal{L}} = \{\mathcal{L}_i\}_{i=1}^K$.

To remove some falsely reconstructed 3D LSs brought by a few falsely grouped matches that exist even after enforcing the smoothness constraint, 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 sure 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. 3(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, generating 2D LS set $\hat{\mathcal{L}}^{2d} = \{\mathcal{L}_i^{2d}\}_{i=1}^K$. Refer to Fig. 3(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, and the convex hull determined by

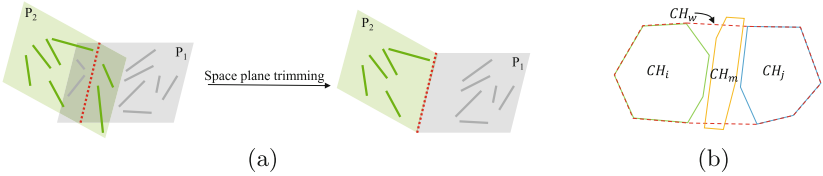


Fig. 3. Illustration of the strategy of removing falsely reconstructed 3D LSs. (a) Adjacent space plane intersection and trimming. (b) Finding adjacent space planes. (Color figure online)

both $\mathcal{L}_i^{2d}$ and $\mathcal{L}_j^{2d}$ be CH_w (the region outlined by dashed red line in Fig. 3(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. 2, we show the final 3D LSs for the scene in sub-figure (d). We can see that the three main planes in this scene are recovered and all 3D LSs are correctly reconstructed and clustered w.r.t. the space planes they lie.

3.2 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. Specifically, we begin to use 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), and generate 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 $\mathcal{L}_m$. Next, we 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 $\hat{\mathcal{L}}^g$.

After processing all images, there would exist a considerable number of duplications in $\hat{\mathcal{L}}^g$ because a same 3D LS can be visible in more than two views and be reconstructed in multiple times. We need to remove these duplications. Since 3D LSs in our case are organized w.r.t. 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 $\mathbf{P}_i \in \mathcal{P}^g$, 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

Algorithm 1. 3D Line Segment Reconstruction

Input: Images $\mathcal{I} = \{\boldsymbol{I}_i\}_{i=1}^{N} (N \geqslant 2)$, 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.
```

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 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 in [12,17]. Therefore, the cases that the 3D reconstructions of multiple scene LSs being falsely regarded as the duplications of one scene LS, and one scene LS being reconstructed with multiple 3D representations are rare in our method. This contributes to the benefit of our method on delivering more details of scenes.

Algorithm 1 outlines the main steps of the proposed method.

4 Experiments

This section presents the experimental results of the proposed method. All images employed for experiments come from public datasets [17,18,33]. We used the method presented in [36] for LS extraction and the method presented [15] for LS matching.

4.1 Two Views

We presented in Figs. 1 and 2 two sample 3D LS reconstruction results based on two views, and Fig. 4 shows four additional such results. We can observe from

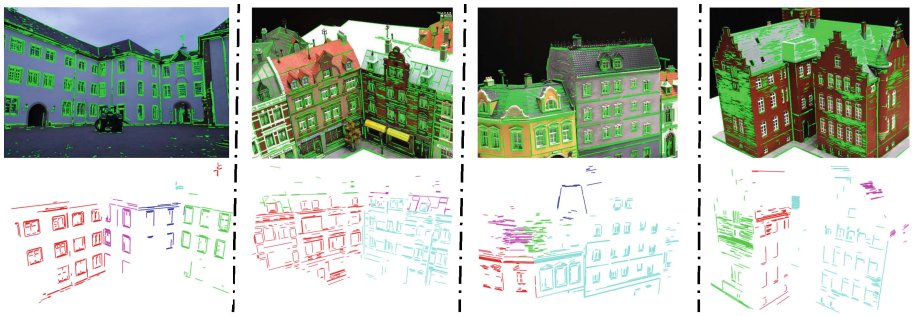


Fig. 4. Two-view based 3D LS reconstruction results. The top row shows the first ones of two images used for 3D LS reconstruction and the extracted LSs; the bottom row shows the obtained 3D LSs.

these results that the proposed method successfully reconstructed a large part of space LSs lying on main planes of the scenes, and correctly clustered them according to the space planes they lie. The main structures of the scenes are outlined by the reconstructed 3D LSs. This proves the feasibility of the proposed two-view based 3D LS reconstruction method.

4.2 Multiple Views

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. Each round consists of 80 images separated by a constant angle interval. An example image from the dataset is shown in Fig. 5(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. 5(b). We can observe from O_{80} that the main planes in this scene are correctly recovered, and LSs in the scene are precisely presented 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. 5(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 LS 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. 5(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; LSs in O_{27} are also correctly clustered w.r.t. their respective planes.

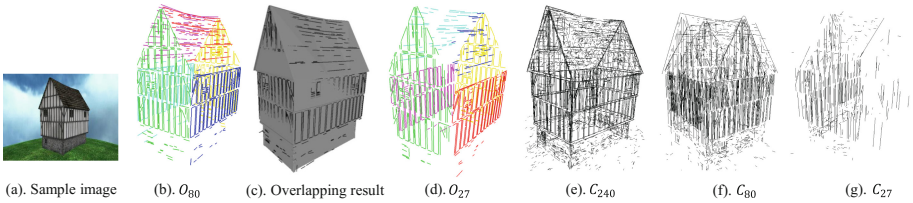


Fig. 5. 3D LS reconstruction results on a synthetic image dataset. (a) One of the used images. (b) The 3D model (referred later as O_{80}) obtained by the proposed method using 80 images. (c) The overlapping result of O_{80} with the ground truth model of the scene. (d) The 3D model (O_{27}) obtained by the proposed method using 27 images. (e)–(g) The 3D models generated by Line3D++ [14] using 240, 80, and 27 images, respectively. The three models will orderly be referred later as C_{240}, C_{80} and C_{27}.

For comparison, we show in Fig. 5(e)–(g) the reconstruction models of a recent algorithm, Line3D++ [14], using the whole 240 images of the dataset, our used 80 images and 27 images, 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 good performance 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, which, to some extent, indicates our model is a better wireframe 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 [12,13,17], we calculated the Hausdorff distances between densely sampled points along the 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 that 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. This fact hinders the quantitative evaluations of models generated by Line3D++ because the underlying coordinates are inconsistent with that of the ground truth model. ICP [6] is a powerful way to align point clouds from different coordinates, which makes it possible to quantitatively evaluate the 3D models generated by Line3D++. However, ICP itself shall introduce alignment errors, and these errors would be counted into the errors between models generated by Line3D++ and the ground truth model. Therefore, the error data calculated in this situation cannot reflect

the true accuracy of the models. On the other hand, the camera matrices corresponding to images in the synthetic dataset are provided and they are consistent with the coordinates of the ground truth model. Our proposed method took the provided camera matrices as input and produced 3D models that are naturally aligned with the ground truth model. So, the error data of our models do not contain alignment errors. For this reason, it is an unfair comparison if we compare our error data (without alignment errors) with that of Line3D++ (with alignment errors). Alternatively, since Line3D++ is a promoted version of the methods presented in [13] and [12] by the same authors, and these two methods do not rely on SFM results, a comparison between our measure data with the report data in [13] and [12] is also some kind of meaningful[1]. Meanwhile, we will show that this indirect comparison does not affect us to reach a conclusion about the accuracy performances of our method and Line3D++.

Table 1. The Mean Error (ME) and Root Mean Square Error (RMSE) data of the reconstruction results obtained by our method and several other ones on a synthetic dataset. "–" denotes the corresponding measure datum was not reported in the paper.

	$\rho = 1.0$					$\rho = 0.6$				
	O_{27}	O_{80}	[17]	[13]	[12]	O_{27}	O_{80}	[17]	[13]	[12]
ME	0.077	0.89	0.162	0.065	–	0.075	0.082	0.137	0.044	0.029
RMSE	0.114	0.135	0.291	0.196	–	0.104	0.109	0.189	0.080	0.046

Table 1 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 [13], 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 [13]. When we set $\rho = 0.6$ as that used in [12], our two models are better than [17], but worse than both [12,13]. Since Line3D++ is promoted from [12,13], 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. 5, it is unlikely that the reconstruction accuracy of C_{80} and C_{27} is better than our two models, O_{80} and O_{27}, when the same numbers of images are used. Therefore, we can reach the conclusion that Line3D++ can produce 3D models with higher accuracy than our method, when plentiful images are available, but in the case 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. Figure 6 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 of the scene are precisely presented (see the bricks and

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

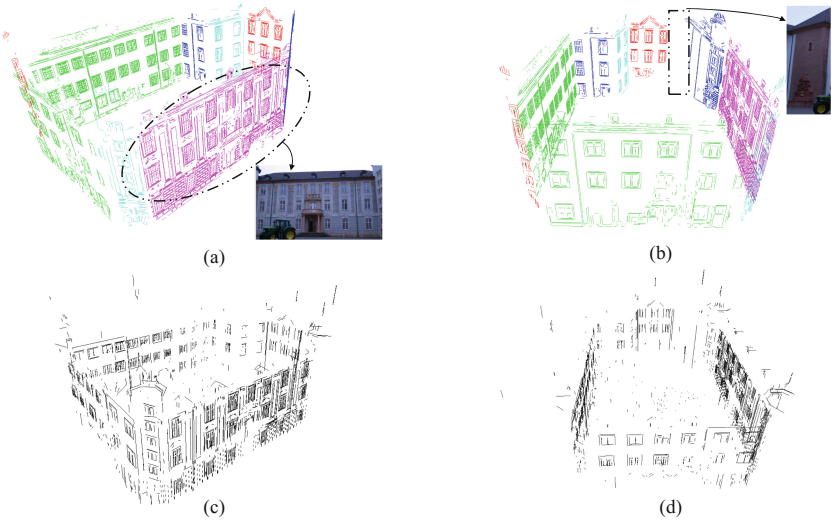


Fig. 6. The 3D LS reconstruction results of the proposed method and Line3D++ on a real image dataset. The top row shows our result model from two different viewpoints, while the bottom row shows that of Line3D++.

windows of the selected dashed elliptical region shown in Fig. 6(a)). Our method failed to reconstruct 3D LSs on the main planes of this scene shown in the selected rectangle region in Fig. 6(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 LS matches induced by 3D LSs on it are too few, and hence incapable to reconstruct the 3D LSs on it. Comparing our model with that of Line3D++, it is obvious that our model is much more complete and detailed.

Running Time. The algorithm is currently implemented based on MATLAB. The unrefined codes took 631 s on the 80 synthetic images and 1021 s 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++.

5 Conclusions

We have presented in this paper a new method about 3D LS reconstruction in structured scenes. A new solution is proposed to solve the uncertainties in 3D LS reconstruction by estimating space planes from clustered LS matches and back-projecting image LSs onto the space planes. We introduce a multi-label optimization framework to improve LS match grouping results. Experiments show the superiority of the proposed method to others in this area for its better performance in using small numbers of images and its ability of clustering 3D LSs w.r.t.

their respective space planes, which is beneficial for upper level applications, like scene understanding [23] and building facade extraction [7,35].

Acknowledgment. This work was partially supported by the National Natural Science Foundation of China (Project No. 41571436), the National Natural Science Foundation of China under Grant 91438203, the Hubei Province Science and Technology Support Program, China (Project No. 2015BAA027), the Jiangsu Province Science and Technology Support Program, China (Project No. BE2014866), and the South Wisdom Valley Innovative Research Team Program.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building Rome in a day. Commun. ACM **54**, 105–112 (2011)
2. Baillard, C., Schmid, C., Zisserman, A., Fitzgibbon, A.: Automatic line matching and 3D reconstruction of buildings from multiple views. In: ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery (1999)
3. Bartoli, A., Sturm, P.: Structure-from-motion using lines: representation, triangulation, and bundle adjustment. Comput. Vis. Image Underst. **100**, 416–441 (2005)
4. Bay, H., Ess, A., Neubeck, A., Van Gool, L.: 3D from line segments in two poorly-textured, uncalibrated images. In: 3DPVT (2006)
5. Boykov, Y., Veksler, O., Zabih, R.: Efficient approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **36**, 1222–1239 (2001)
6. Chetverikov, D., Svirko, D., Stepanov, D., Krsek, P.: The trimmed iterative closest point algorithm. In: ICPR (2002)
7. Delmerico, J.A., David, P., Corso, J.J.: Building facade detection, segmentation, and parameter estimation for mobile robot stereo vision. Image Vis. Comput. **31**, 841–852 (2013)
8. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. Int. J. Comput. Vis. **96**, 1–27 (2012)
9. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 1362–1376 (2010)
10. Habib, A.F., Morgan, M., Lee, Y.R.: Bundle adjustment with selfcalibration using straight lines. Photogram. Rec. **17**, 635–650 (2002)
11. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
12. Hofer, M., Maurer, M., Bischof, H.: Improving sparse 3D models for man-made environments using line-based 3D reconstruction. In: 3DV (2014)
13. Hofer, M., Wendel, A., Bischof, H.: Incremental line-based 3D reconstruction using geometric constraints. In: BMVC (2013)
14. Hofer, M., Maurer, M., Bischof, H.: Efficient 3D scene abstraction using line segments. Comput. Vis. Image Underst. (2016). doi:10.1016/j.cviu.2016.03.017
15. Li, K., Yao, J., Lu, X., Xia, M., Li, L.: Joint point and line segment matching on wide-baseline stereo images. In: WACV (2016)
16. Kim, C., Manduchi, R.: Planar structures from line correspondences in a Manhattan World. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9003, pp. 509–524. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16865-4_33

17. Jain, A., Kurz, C., Thormahlen, T., Seidel, H.P.: Exploiting global connectivity constraints for reconstruction of 3D line segments from images. In: CVPR (2010)
18. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E.: Large scale multi-view stereopsis evaluation. In: CVPR (2014)
19. Luong, Q.-T., Viéville, T.: Canonical representations for the geometries of multiple projective views. Comput. Vis. Image Underst. **64**, 193–229 (1996)
20. Matinec, D., Pajdla, T.: Line reconstruction from many perspective images by factorization. In: CVPR (2003)
21. Micusik, B., Wildenauer, H.: Structure from motion with line segments under relaxed endpoint constraints. In: 3DV (2014)
22. Micusik, B., Wildenauer, H.: Descriptor free visual indoor localization with line segments. In: 3DV (2015)
23. Pan, J.: Coherent scene understanding with 3D geometric reasoning. Ph.D. thesis, Carnegie Mellon University (2014)
24. Pham, T.T., Chin, T.J., Yu, J., Suter, D.: The random cluster model for robust geometric fitting. IEEE Trans. Pattern Anal. Mach. Intell. **36**, 1658–1671 (2014)
25. Přibyl, B., Zemčík, P., Čadík, M.: Camera pose estimation from lines using Plücker coordinates. In: BMVC (2015)
26. Ramalingam, S., Brand, M.: Lifting 3D Manhattan lines from a single image. In: ICCV (2013)
27. Schindler, G., Krishnamurthy, P., Dellaert, F.: Line-based structure from motion for urban environments. In: 3DPVT (2006)
28. Sinha, S.N., Steedly, D., Szeliski, R.: Piecewise planar stereo for image-based rendering. In: ICCV (2009)
29. Smith, P., Reid, I.D., Davison, A.J.: Real-time monocular SLAM with straight lines. In: BMVC (2006)
30. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. ACM Trans. Graph. **25**, 835–846 (2006)
31. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. Int. J. Comput. Vis. **80**, 189–210 (2008)
32. Spetsakis, M.E., Aloimonos, J.Y.: Structure from motion using line correspondences. Int. J. Comput. Vis. **4**, 171–183 (1990)
33. Strecha, C., Hansen, W.V., Gool, L.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR (2008)
34. Taylor, C.J., Kriegman, D.J.: Structure and motion from line segments in multiple images. IEEE Trans. Pattern Anal. Mach. Intell. **17**, 1021–1032 (1995)
35. Teboul, O., Simon, L., Koutsourakis, P., Paragios, N.: Segmentation of building facades using procedural shape priors. In: CVPR (2010)
36. Werner, T., Zisserman, A.: New techniques for automated architectural reconstruction from photographs. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 541–555. Springer, Heidelberg (2002). doi:10.1007/3-540-47967-8_36
37. Wu, C.: Towards linear-time incremental structure from motion. In: 3DV (2013)
38. Zhang, L., Koch, R.: Structure and motion from line correspondences: representation, projection, initialization and sparse bundle adjustment. J. Vis. Commun. Image Represent. **25**, 904–915 (2014)
39. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: ICCV (1999)