ISPRS Journal of Photogrammetry and Remote Sensing 171 (2021) 49–62

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Robust line segment matching across views via ranking the line-point graph

Dong Wei[a], Yongjun Zhang[a,*], Xinyi Liu[a,*], Chang Li[b], Zhuofan Li[b]

[a] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[b] Key Laboratory for Geographical Process Analysis & Simulation, Central China Normal University, Wuhan 430079, China

ARTICLE INFO

Keywords:
Line segment matching
Graph rank
Scene plane theory
3D reconstruction

ABSTRACT

Line segment matching in two or multiple views is helpful to 3D reconstruction and pattern recognition. To fully utilize the geometry constraint of different features for line segment matching, a novel graph-based algorithm denoted as GLSM (Graph-based Line Segment Matching) is proposed in this paper, which includes: (1) the employment of three geometry types, i.e., homography, epipolar, and trifocal tensor, to constrain line and point candidates across views; (2) the method of unifying different geometry constraints into a line-point association graph for two or multiple views; and (3) a set of procedures for ranking, assigning, and clustering with the line-point association graph. The experimental results indicate that GLSM can obtain sufficient matches with a satisfactory accuracy in both two and multiple views. Moreover, GLSM can be employed with large image datasets. The implementation of GLSM will be available soon at https://skyearth.org/research/.

1. Introduction

Line segment matching is of paramount importance for 3D reconstruction in manmade scenes, which can obtain the abstractive 3D structure (Hofer et al., 2016) and provide extensive clues to reconstruct more accurate and complete 3D scenes (Li et al., 2016; Liu et al., 2019). Although there is much available software for point matching, e.g., VisualSFM (Visual Structure from Motion) (Wu, 2013), Bundler (Snavely et al., 2008), and OpenMVG (Moulon et al., 2013), line segment matching is still considered as an open field for its inherent problems: lack of strong epipolar constraint, poor texture along line segments, and indefinite endpoint of the line segment. Our research motivation is to develop a robust line segment matching algorithm that can be employed in both two and multiple views for manmade scenes.

Up to now, a significant number of studies about line segment matching have been published. These algorithms can be categorized into three types: matching with texture similarity, matching with planar geometry, and matching with multiple-view geometry.

Matching with texture similarity describes the textures around line segments by histogram (Bay et al., 2005) or vector (Wang et al., 2009) like SIFT (Lowe, 2004). A pyramid strategy was proposed (Zhang et al., 2013) to make the descriptor scale invariant. To make the descriptor affine invariant, the affine invariant region was first confirmed with the junctions of at least three lines (Li et al., 2016; Li and Jian, 2017). Because the textures around line segments are not unique enough to obtain reliable matches, other geometry constraints, such as epipolar constraint (Wang et al., 2009), pairwise constraint (Zhang et al., 2013), and a combination of these constraints (Ok et al., 2012; Al-Shahri and Yilmaz, 2014; López et al., 2014) were exploited. Matching with planar geometry exploits the cues that are invariant under affine or projective transformation. These algorithms assume that the local scene around the line segment is coplanar. Schmid et al. (1997, 2000) proposed the classical algorithm, which first calculates the homography between line candidates, then computes the cross correlation with the local homography. Generally, matching points is easier than matching lines, thus point correspondences can be used to guide line matching based on the local affine or projective transformation (Fan et al., 2010, 2012; Sun et al., 2015; Jia et al., 2019), or obtain more reliable descriptors (Shi and Wang, 2017), but they require sufficient and high quality point correspondences. Recently, Wang et al. (2020) exploit point-to-line distance ratio in an affine projection space to solve the problem of matching in aerial oblique images.

The above algorithms match line segments with the cues between stereo images. When dealing with multiple images, the multiple-view geometry can be exploited, which does not employ explicit line matching with stereos, but focuses on the cluster in 3D space. Jain et al. (2010) used the global topology of line connections as the constraint, and the 3D line segment was estimated with a probability method. In

* Corresponding authors.
E-mail addresses: weidong@whu.edu.cn (D. Wei), zhangyj@whu.edu.cn (Y. Zhang), liuxy0319@whu.edu.cn (X. Liu), lichang@mail.ccnu.edu.cn (C. Li), lizhoufan@qq.com (Z. Li).

https://doi.org/10.1016/j.isprsjprs.2020.11.002
Received 14 June 2020; Received in revised form 29 September 2020; Accepted 3 November 2020
Available online 16 November 2020

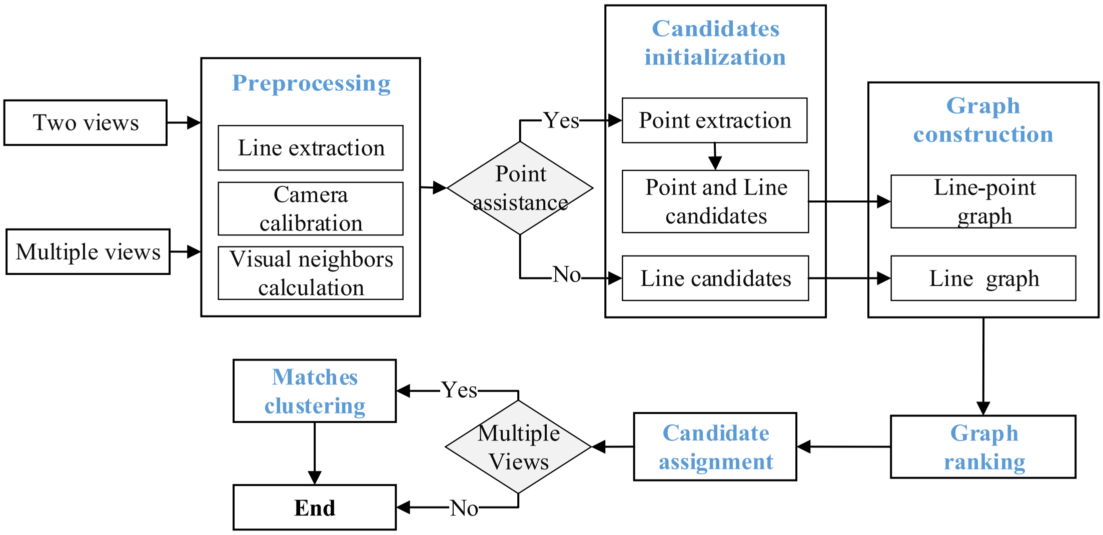


Fig. 1. The flowchart of GLSM.

fact, Jain's method skipped the explicit line matching step. The number of different views from which these underlying correspondences emerge is a good indicator to identify whether a 3D segment is reasonable to exist in reality or no. Based on this, Beder (2004) proposed a graph method for matching line and point with multiple images, which first groups the candidate that intersects with other candidate into a graph, then the minimum clique partition is employed with the graph to find reliable matches. Beder's work is instructive while the geometry constraint to valid the intersect in the object space is not well studied. Also, there is no edge between line and point candidate in the graph, thus the matching of line and point candidate is isolated. Hofer et al. (2016) proposed a rather intuitive and fast method named Line 3D++ based on the intersect assumption in object space, and a detailed study is carried out to validate the cluster in 3D space. In practice, Line 3D++ runs efficiently with GPU but cannot work for stereo images and requires sufficient images to obtain a reliable cluster. In addition, these clustering methods may fail when the image is with a long range (Jain et al., 2010).

There are different constraints for matching line segments, such as point correspondence and planar geometry for stereo images, trifocal tensor for multiple views. However, these constraints are local and independent, in which a wrong candidate may even satisfy all constraints, and there is no generalized approach to unify these geometry constraints. GLSM is designed to unify the geometry constraint across views for robust line segment matching, which is inspired by the two order graph matching algorithm named RRWM (Reweighted Random Walks for Graph Matching) (Cho et al., 2010). RRWM is similar to the personalization strategy of web ranking algorithms (Page et al., 1999), in which the node (candidate) is considered as a page in the web, and the edge (geometry constraint) is viewed as the link between nodes. Then, the candidate is ranked with random walks under the democratic assumption: a candidate has high rank if the sum of the candidate of its links is high. Note that although there are many graph matching algorithm (McAuley et al., 2010; Zhou et al., 2012; Yang et al., 2017) like RRWM that can match patterns considering the global constraints, their graph is just designed for points in two small image patches.

The core idea in RRWM is very versatile, which is also appropriate for line segment matching, because a correct line match is likely to be consistent in geometry with other correct matches. Motivated by RRWM and the aforementioned studies and concerns, three main objectives have been conducted in this paper:

(1) To offer a set of geometry constraints to build connections for pairwise candidates across views, i.e., line and line candidates, line and point candidates, and triplet candidates across three views.
(2) To unify the line and point candidate, and their geometry constraints across views, in one association graph, namely, the line-point graph. In the graph the candidate can be ranked and assigned by considering not only the nodes in the same stereo, but also the nodes in other stereos.
(3) To improve the precision and reliability of assigning candidates. The constrained greedy algorithm for two-view matching, and the clustering algorithm for multiple-view matching are proposed. The false match is efficiently reduced.

The main contribution of this paper lies in that a generalized graph-based framework is proposed based on the coplanar assumption, which unifies different geometry constraints of line and point candidates across stereos or multiple-views into one association graph for matching line segments. Compared with previous studies, it is efficient for matching line segments in both two and multiple views by considering not only the neighboring line and point candidate in two views, but also the candidate in other stereos. The remainder of the paper is organized as follows: The overview of GLSM is presented in Section 2. Section 3 introduces the geometry constraints in the association graph. Section 4 describes how to construct the line-point association graph. The ranking approach to confirm matches is introduced in Section 5. Our performance evaluation is introduced in Section 6. And Section 7 concludes the paper.

2. Overview

Basically, the candidate that is aligned with many other candidates in geometry demonstrates a higher chance to be correct, thus after unifying the geometry constraint of the candidate into an association graph. The line matching problem can be converted into ranking the graph node. Fig. 1 presents the flowchart of GLSM, which consists of several steps:

(1) Establish geometry constraint between pairwise candidates. First, the additional one constraint in the scene plane theory is exploited for pairwise line candidates. Then, the local homography induced with pairwise line candidates is used to verify line and point candidates. Finally, for multiple-view matching, the trifocal tensor is employed to join the candidate in different stereos.

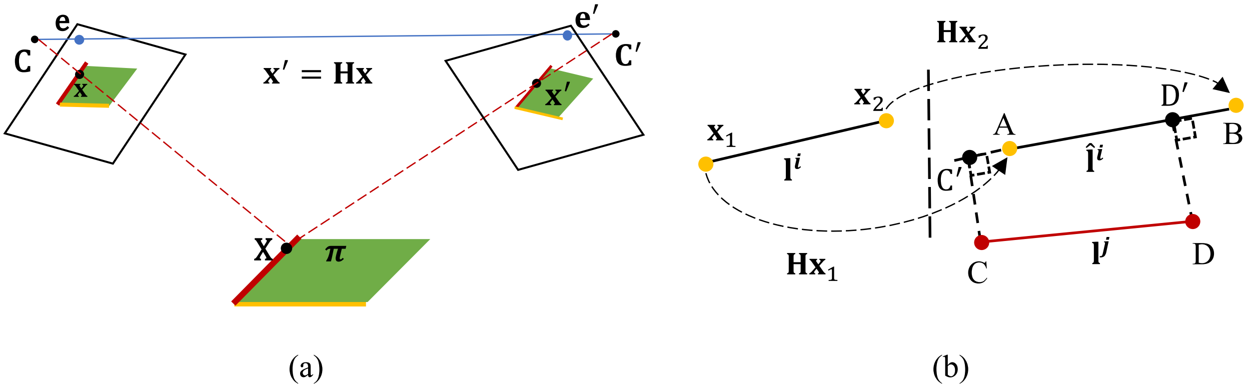


Fig. 2. The illustration of the homography constraint. (a) The scene plane $\boldsymbol{\pi}$ that can be represented with two scene lines determines the homography$\mathbf{H}$, thus any image point $\mathbf{x}$ in the green can be mapped to the second view by $\mathbf{x}^{'} = \mathbf{H}\mathbf{x}$. (b) The mapping distance and the overlapping rate to validate if one candidate is aligned with the local homography. The overlapping rate is calculated as $|\mathrm{AD}^{'}|/|\mathrm{CD}|$, in which CD is the shorter line segment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(2) Construct the general association graph to unify the geometry constraints of different features (line and point) across multiple views. In the graph, the line and point candidate is denoted as the node and the edge between two nodes represents the geometry constraint. The proposed association graph is capable of embedding the geometry constraint for both two-view and multiple-view matching.

(3) Rank the node in the association graph. The association graph can be ranked by many soft assigned graph matching algorithm, and the reweighted random walk (RRW) is employed for its robustness and the potential for further optimization. Accordingly, we have optimized RRW for matching with large quantities of nodes in multiple views, which is originally designed to match with two patterns

(4) Assign the matches after ranking the nodes in the association graph, and a clustering method is proposed for multiple-view matching.

Note that the three types of geometry constraints in step (1) are hierarchical but not a package in the association graph: First, the constraint of pairwise line candidates is essential and necessary for the further construction of the line-point candidate, and the constraint of the triplet candidate is established based on the former two constraints. Second, the constraint becomes stricter from pairwise line candidates to triplet candidates, thus providing more useful cues but not noise for ranking the node.

3. Geometry constraints

GLSM exploits pure geometry constraint to build edges between nodes. There are three types of edges: the edge between pairwise line candidates, the edge between line and point candidates, and the edge between two point or line candidates in triple views. The geometry constraints are as follows.

3.1. Pairwise line candidates

The scene planes theory (Luong and Viéville, 1994), which describes the projective geometry of two cameras and a world plane, is introduced to constrain the pairwise line candidates. As illustrated in Fig. 2 (a), denote the scene plane as $\boldsymbol{\pi} = (\mathbf{v}^{\mathrm{T}}, 1)^{\mathrm{T}}$, of which $\mathbf{v} = (a, b, c)^{\mathrm{T}}$, thus any 3D point $\mathbf{X} = (x, y, z, 1)^{\mathrm{T}}$ on the scene plane satisfies $\boldsymbol{\pi}^{T}\mathbf{X} = 0$, Luong and Viéville (1994) showed that the local homography corresponding to $\boldsymbol{\pi}$ can be calculated as

$$\mathbf{H} = \mathbf{A} - \mathbf{e}^{'}\mathbf{v}^{\mathrm{T}} \tag{1}$$

Such that the images of points on $\boldsymbol{\pi}$ are related by $\mathbf{x}^{'} = \mathbf{H}\mathbf{x}$ (a point $\mathbf{x} = (x, y, 1)^{T}$); $\mathbf{e}^{'}$ is the epipole of the second view. $\mathbf{A}$ is the first three columns of the second camera matrix when we choose $\mathrm{P} = [\mathbf{I}|0]$, $\mathrm{P}^{'} = [\mathbf{A}|\mathbf{e}^{'}]$, and it can be obtained by $\mathbf{A} = [\mathbf{e}^{'}]_{\times}\mathbf{F}$ (Hartley and Zisserman, 2008). With the epipolar geometry any point $\mathbf{x}$ on the line $\mathbf{l}$ is mapped to a fixed point on $\mathbf{l}^{'}$ in the second view, thus Eq. (1) can be written as

$$\mathbf{l}^{'\mathrm{T}}\left(\mathbf{A} - \mathbf{e}^{'}\mathbf{v}^{\mathrm{T}}\right)\mathbf{x} = 0 \tag{2}$$

Given two line candidates, four equations can be listed with Eq. (2), thus $\mathbf{v}$ can be estimated by least squares with additional one constraint since $\mathbf{v}$ is a 3D vector. Li et al. (2016), Li and Jian (2017) have employed Eq. (2) to obtain the local homography with pairwise line candidates, but the additional one constraint is not furtherly explored.

The geometry constraint of pairwise line candidates with the scene planes theory is under the assumption that a 3D line segment has a high chance to be coplanar with some of its neighbors. Given two line matches denoted as $L_a^{ij} = (\mathbf{l}_{a1}^{i}, \mathbf{l}_{a2}^{j})$ and $L_b^{ij} = (\mathbf{l}_{b1}^{i}, \mathbf{l}_{b2}^{j})$ (the superscript denotes the image index and the subscript represents the line index), the local homography $\mathbf{H}$ can be estimated by Eq. (2) with one constraint. As illustrated in Fig. 2 (b), for each candidate $L^{ij} = (\mathbf{l}^{i}, \mathbf{l}^{j})$, mapping $\mathbf{l}^{i}$ to the second view as $\widehat{\mathbf{l}}^{i}$ with $\mathbf{H}$; denote the shorter of $\widehat{\mathbf{l}}^{i}$ and $\mathbf{l}^{j}$ as l_{CD} and the longer as l_{AB}; both L_a^{ij} and L_b^{ij} should satisfy

$$\begin{cases} |\mathrm{CC}^{'}| < T_{map} \\ |\mathrm{DD}^{'}| < T_{map} \\ |\mathrm{AD}^{'}|/|\mathrm{CD}| > T_{ove} \end{cases} \tag{3}$$

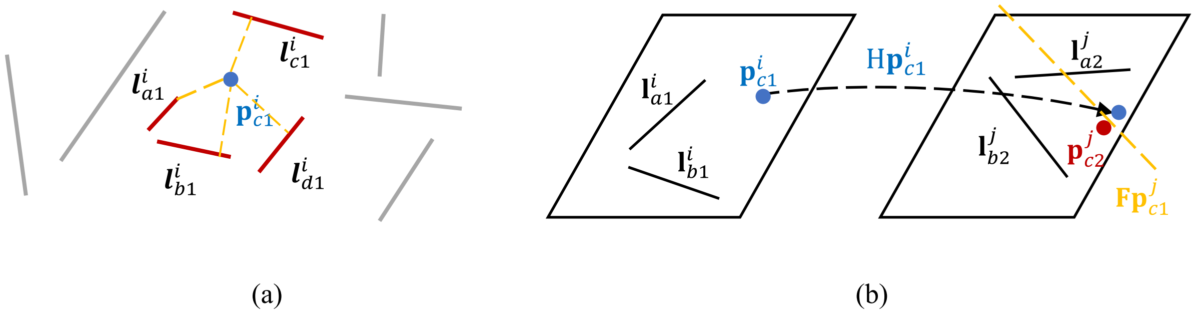
to be aligned with $\mathbf{H}$. Otherwise, the scene lines of L_a^{ij} and L_b^{ij} are incorrect or not coplanar. T_{map} and T_{ove} in Eq. (3) represent the thresholds of mapping distance and overlapping rate, respectively. Since the endpoint of a line segment is indefinite, the consistency of L^{ij} with $\mathbf{H}$ is scored by the average endpoint-to-line distance between $\widehat{\mathbf{l}}^{i}$ and $\mathbf{l}^{j}$ as

$$\Omega_L\left(L^{ij}\right) = T_{map} - \frac{|\mathrm{CC}^{'}| + |\mathrm{DD}^{'}|}{2} \tag{4}$$

If both L_a^{ij} and L_b^{ij} satisfy Eq. (3), we assume the pairwise candidates satisfy the homography constraint, and take the average of $\Omega_L\left(L_a^{ij}\right)$ and $\Omega_L\left(L_b^{ij}\right)$ to score the pairwise line candidates

$$\Omega_{L,L}\left(L_a^{ij}, L_b^{ij}\right) = \frac{\Omega_L\left(L_a^{ij}\right) + \Omega_L\left(L_b^{ij}\right)}{2} \tag{5}$$

Eq. (5) is the core formula to assess the two line-matching score.



(a) (b)

Fig. 3. The geometry constraint of the point and line candidate. (a) P_c^{ij} is a neighbor of L_a^{ij} if $\mathbf{l}_{a1}^{i}$ is one of the T_{nei} ($T_{nei} = 4$ in this figure) closest line segments to $\mathbf{p}_{c1}^{i}$. The line-point distance is calculated as the shortest distance between $\mathbf{p}_{c1}^{i}$ and the point on the line segment. (b) $\mathbf{p}_{c1}^{i}$ is mapped via $\mathbf{H}\mathbf{p}_{c1}^{i}$ to the second view to validate the mapping error.

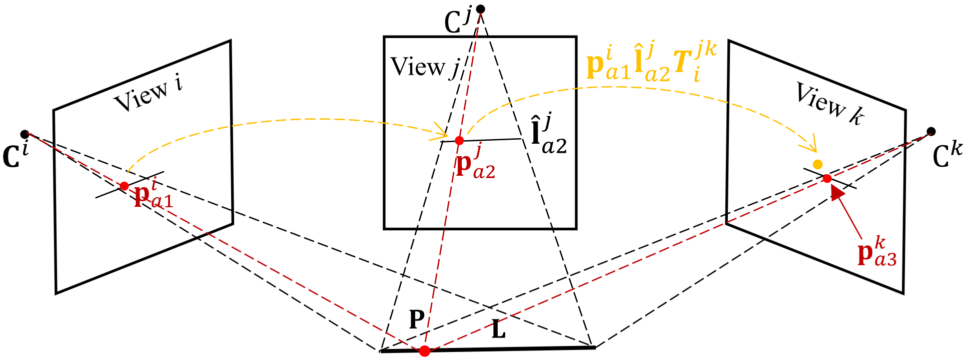


Fig. 4. Trifocal tensor constraint for triplet candidate. For pairwise point candidates $P_a^{ij} = (\mathbf{p}_{a1}^{i}, \mathbf{p}_{a2}^{j})$ and $P_a^{jk} = (\mathbf{p}_{a2}^{j}, \mathbf{p}_{a3}^{k})$, $\mathbf{p}_{a3}^{k}$ in the third view can be predicted by $\mathbf{p}_{a1}^{i}$ and $\mathbf{p}_{a2}^{j}$ in the first and second views, thus the inconsistency of $\mathbf{p}_{a1}^{i} \leftrightarrow \mathbf{p}_{a2}^{j} \leftrightarrow \mathbf{p}_{a3}^{k}$ is calculated as $\|\mathbf{p}_{a1}^{i}\widehat{\mathbf{l}}_{a2}^{j}\boldsymbol{T}_{i}^{jk} - \mathbf{p}_{a3}^{k}\|$.

3.2. Line and point candidates

The line candidate is likely to be correct if it is aligned with many point correspondences. The alignment is also based on the coplanar assumption, i.e., a 3D line segment has a high chance to be coplanar with some neighbor 3D points. Fan et al. (2012) and Sun et al. (2015) have used at least three point correspondences to validate line candidates, while GLSM is on the contrary: the point candidate is validated based on the local homography induced by pairwise line candidates (Section 3.1). GLSM is able to establish the edge of line and point candidate with less correspondence but more robust constraint, because there is one additional constraint in estimating the local homography with pairwise line candidates (Section 3.1). Given the individual point candidate $P_c^{ij} = (\mathbf{p}_{c1}^{i}, \mathbf{p}_{c2}^{j})$, if P_c^{ij} is a neighbor (Fig. 3 (a)) of L_a^{ij} or L_b^{ij} and is consistent with $\mathbf{H}$ that is induced by L_a^{ij} and L_b^{ij}(Fig. 3 (b)), the consistency is scored by the point-to-point distance as

$$\Omega_{P,L}\left(P_c^{ij}, L_a^{ij}\right) = \Omega_{P,L}\left(P_c^{ij}, L_b^{ij}\right) = T_{map} - \left\|\mathbf{H}\mathbf{p}_{c1}^{i} - \mathbf{p}_{c2}^{j}\right\| \tag{6}$$

P_c^{ij} should also be constrained because Eq. (6) does not hold if P_c^{ij} is incorrect, thus the epipolar geometry is introduced to validate P_c^{ij}. Given the fundamental matrix $\mathbf{F}$ of the i-th and j-th view, the consistency score of P_c^{ij} is

$$\Omega_P\left(P_c^{ij}\right) = T_{map} - dis\left(\mathbf{F}\mathbf{p}_{c1}^{i}, \mathbf{p}_{c2}^{j}\right) \tag{7}$$

where $dis(*, *)$ computes the orthogonal distance from the point to the epipolar line that induced from the corresponding point in the first view.

3.3. Triplet candidates

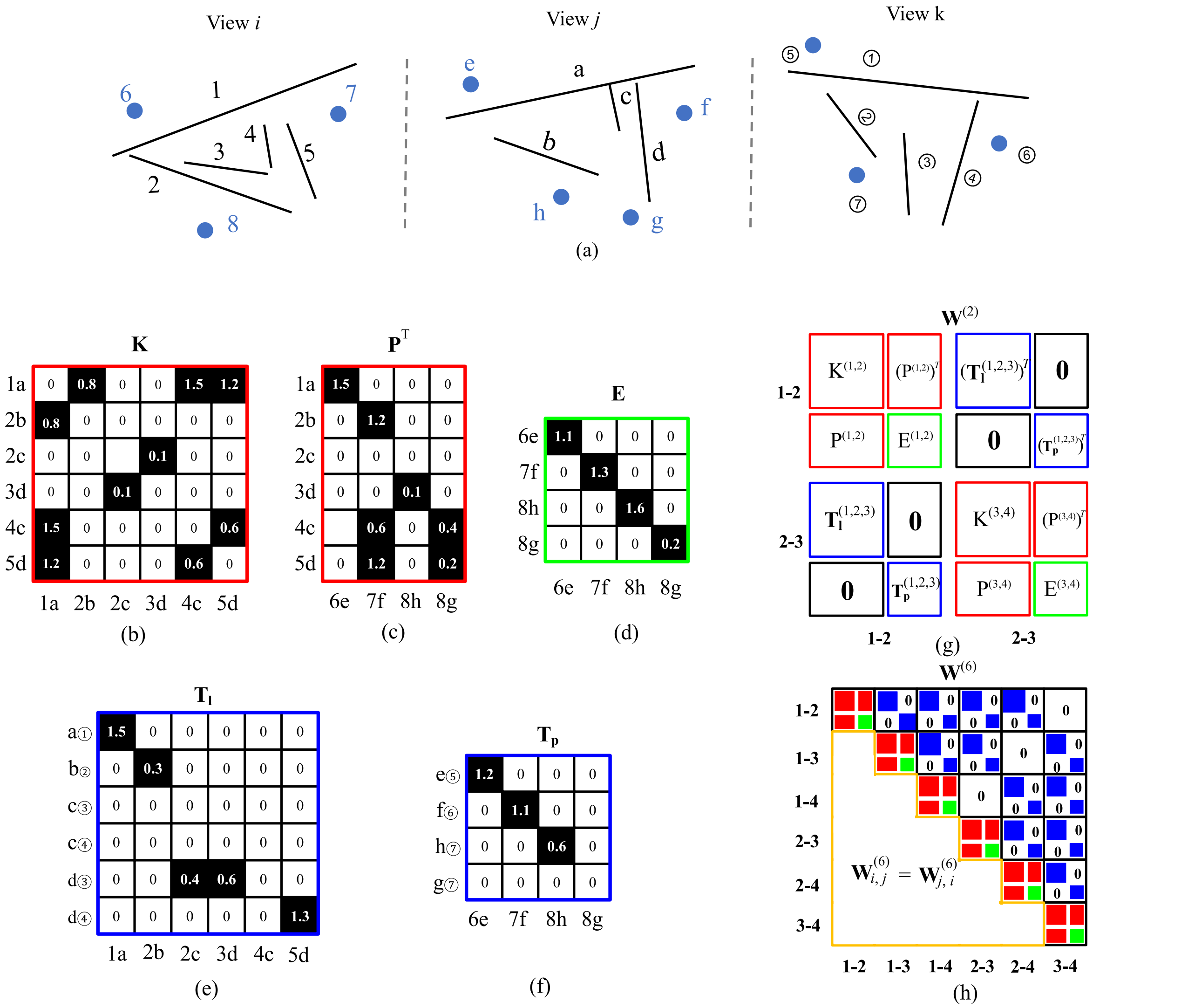
All line triplet combinations can be considered to improve the line segment matching (Schmid et al., 1997; 2000), thus trifocal tensor is introduced in GLSM to establish connections for candidates in different stereos. Given the pairwise point candidates $P_a^{ij} = (\mathbf{p}_{a1}^{i}, \mathbf{p}_{a2}^{j})$ and $P_a^{jk} = (\mathbf{p}_{a2}^{j}, \mathbf{p}_{a3}^{k})$, there is a triplet-combination denoted as $\mathbf{p}_{a1}^{i} \leftrightarrow \mathbf{p}_{a2}^{j} \leftrightarrow \mathbf{p}_{a3}^{k}$ (Fig. 4). The position of $\mathbf{p}_{a3}^{k}$ can be predicted via $\mathbf{p}_{a3}^{k} = \mathbf{p}_{a1}^{i}\widehat{\mathbf{l}}_{a2}^{j}\boldsymbol{T}_{i}^{jk}$, in which $\widehat{\mathbf{l}}_{a2}^{j}$ is any line except for the epipolar line passing through $\mathbf{p}_{a2}^{j}$, and $\boldsymbol{T}_{i}^{jk}$ is the trifocal tensor (a homogeneous $3 \times 3 \times 3$ array) that can be calculated with $\mathbf{F}$ matrix (Hartley and Zisserman, 2008). If the triplet point candidate is correct, the error that mapping $\mathbf{p}_{a1}^{i}$ to $\mathbf{p}_{a3}^{k}$ with trifocal tensor should be small, thus, the triplet point correspondence is scored by the error between the mapping point and the candidate point

$$\Omega_{P-P}\left(P_a^{ij}, P_a^{jk}\right) = T_{map} - \left\|\mathbf{p}_{a1}^{i}\widehat{\mathbf{l}}_{a2}^{j}\boldsymbol{T}_{i}^{jk} - \mathbf{p}_{a3}^{k}\right\| \tag{8}$$

Evaluating the triplet line candidate is similar to evaluate the point candidate, since a line segment is represented by two endpoints. The endpoints of the line segment in the first view are mapped to the third with trifocal tensor, and then, the mapping can also be verified by the overlapping rate and the mapping distance as Eq. (3). Thus, the matching can be scored by the average endpoint-to-line distance, which is the same as Eq. (5)

$$\Omega_{L-L}\left(L_a^{ij}, L_a^{jk}\right) = \Omega_L\left(L_a^{ik}\right) \tag{9}$$

Scoring all triplet candidates would have a significantly higher complexity, but this cost can be largely avoided by initializing candidates based on the homography constraint on stereo images to remove ridiculous matches, which will be discussed in Section 4.



K

	1a	2b	2c	3d	4c	5d
1a	0	0.8	0	0	1.5	1.2
2b	0.8	0	0	0	0	0
2c	0	0		0.1	0	0
3d	0	0	0.1	0	0	0
4c	1.5	0	0	0	0	0.6
5d	1.2	0	0	0	0.6	0

P^T

	6e	7f	8h	8g
1a	1.5	0	0	0
2b	0	1.2	0	0
2c	0	0	0	0
3d	0	0	0.1	0
4c		0.6	0	0.4
5d	0	1.2	0	0.2

E

	6e	7f	8h	8g
6e	1.1	0	0	0
7f	0	1.3	0	0
8h	0	0	1.6	0
8g	0	0	0	0.2

T$_l$

	1a	2b	2c	3d	4c	5d
a①	1.5	0	0	0	0	0
b②	0	0.3	0	0	0	0
c③	0	0	0	0	0	0
c④	0	0	0	0	0	0
d③	0	0	0.4	0.6	0	0
d④	0	0	0	0	0	1.3

T$_p$

	6e	7f	8h	8g
e⑤	1.2	0	0	0
f⑥	0	1.1	0	0
h⑦	0	0	0.6	0
g⑦	0	0	0	0

Fig. 5. Illustration of the association graph. (a) The points and lines in three views. (b)-(f) The matrices of $\Omega_{L,L}$, $\Omega_{P,L}$, Ω_P, Ω_{L-L}, and Ω_{P-P}. (g) The partitioned matrices of three views with two image pairs. (h) The partitioned matrices of four views with six image pairs. The filled color is aligned with the outline in (g).

4. Line-point association graph

The association graph unifies the geometry constraints for pairwise candidates, in which the node represents the point or line candidate and the edge represents the geometry constraint. Compared with other graph based algorithm (Beder, 2004; McAuley et al., 2010; Yang et al., 2017), the association graph in GLSM is unique for two aspects: (1) the geometry constraint is manly based on the local coplanar assumption, and the edge between line and point candidate is able to be established. (2) The association graph can embed the geometry constraints of line and point features in both two and multiple views. Before graph construction, both line and point candidates need to be initialized.

4.1. Candidates initialization

Initial point candidates: Because **F** matrix is a priori knowledge, any point pair satisfying the epipolar geometry, i.e., $\Omega_P > 0$, is initialized as the candidate.

Initial line candidates: If L_a^{ij} and L_b^{ij} satisfy the pairwise constraint (Section 3.1), they are initialized as candidates. Because the homography constraint is only effective when the pairwise candidates are coplanar and validating all pairwise candidates will bring about large computations, line candidates are initialized with two steps:

(1) Setting geometry constraint on line pairs: only if the intersection of the line pair is within T_{int} pixels to the nearest endpoint, the pair is selected for further validation.
(2) After the validation of pairwise candidates, the individual candidate that is consistent with one of its T_{nei} neighboring homographies is added as the initial candidate. The consistency is validated based on Eq. (3).

4.2. Graph construction

Denote n^1 as the number of the candidates, we use a matrix $\mathbf{W}^{(\delta)} \in [0, T_{map}]^{n^1 \times n^1}$ to encode the association graph, in whichδ is the number of image pairs.

For two-view matching, there are three types of geometry constraints: $\Omega_{L,L}$, $\Omega_{P,L}$, and Ω_P. They are encoded in $\mathbf{W}^{(1)}$ as the partitioned matrices.

Table 1
The image dataset in each section.

Section	Name	Number	Image size (px)	Feature point	Line segment	Camera matrix
6.1	Vaihingen	30	$< 750 \times 750$[a]	SURF (Bay et al., 2006)	Provided in (Ok et al., 2012)	
6.2	Herz-Jesu	8	3072×2048		LSD (Von Gioi et al., 2010)	Provided in (Furukawa et al., 2007)
6.2	Castle	30	3072×2048		LSD	
6.3	Dublin	368	$7360 \times 49129000 \times 6732$		LSD	VisualSFM (Wu, 2013)

[a] The image size is varied, thus the biggest value of the width and height is presented.

$$\mathbf{W}^{(1)} = \begin{bmatrix} K\mathbf{P}^T \\ PE \end{bmatrix} \tag{10}$$

Fig. 5(a)-(d) illustrates the example of the partitioned matrix. $\mathbf{K}$ encodes the homography constraint of pairwise line candidates, in which $\mathbf{K}_{m,n} = \Omega_{L,L}\left(L_m^{ij}, L_n^{ij}\right)$. $\mathbf{P}$ encodes the homography constraint between point and line candidates, in which $\mathbf{P}_{m,n} = \Omega_{P,L}\left(P_m^{ij}, L_n^{ij}\right)$. $\mathbf{E}$ is a diagonal matrix encoding the epipolar constraint of the individual point candidate, in which $\mathbf{E}_{\boldsymbol{m},\boldsymbol{m}} = \Omega_P\left(P_m^{ij}\right)$. Note that $\mathbf{W}^{(1)}$ is a symmetric matrix.

For multiple-view matching, the trifocal tensor constraint is introduced to the association graph

$$\boldsymbol{W}^{(\delta)} = \begin{bmatrix} \boldsymbol{W}_1^{(1)} & \boldsymbol{T}_{1-2} & \cdots & \boldsymbol{T}_{1-\delta} \\ \boldsymbol{T}_{2-1} & \boldsymbol{W}_2^{(1)} & & \\ \vdots & & \ddots & \vdots \\ T_{\delta-1} & & \cdots & \boldsymbol{W}_\delta^{(1)} \end{bmatrix} \qquad \boldsymbol{T}_{i-j} = \begin{bmatrix} \boldsymbol{T}_1^{(i,j)} & 0 \\ 0 & \mathrm{T}_\mathrm{p}^{(i,j)} \end{bmatrix} \tag{11}$$

where i and j denote the indices of the image pair. Note that $\mathbf{T}_{i-j} = 0$ when the i-th image pair has no common image with the j-th image pair. The partitioned matrices T_l and T_p (see Fig. 5 (e)-(f)) are the basic blocks that encode the trifocal tensor constraint, in which $\mathbf{T}_{\mathrm{l}(m,n)} = \Omega_{L-L}\left(L_m^{ij}, L_n^{jk}\right)$ and $\mathbf{T}_{\mathrm{p}(m,n)} = \Omega_{P-P}\left(P_m^{ij}, P_n^{jk}\right)$. Fig. 5 (g)-(h) show the examples of $\mathbf{W}^{(2)}$ and $\mathbf{W}^{(6)}$.

Because the homography constraint is only effective when two nodes are coplanar, the edge between two nodes is constrained: each node only connects to its T_{nei} neighboring nodes, and the edge with a negative score will be removed.

5. Matches confirmation

Having obtained the line-point graph, the correct matches are confirmed with two steps: ranking and assigning. Ranking is employed based on reweighted random walk (RRW) (Cho et al., 2010; Lee et al., 2011), and we make it workable for large image datasets in common computers. Assigning is employed with the greedy algorithm (Leordeanu and Hebert, 2005; Lee et al., 2011). For multiple images, a cluster method is proposed to obtain the image lines that correspond to the same 3D line.

5.1. Efficient ranking with random walks

Defining an assignment vector $\mathbf{x} \in [0,1]^{n_1}$ and $\sum_{i=1}^{n_1} \mathbf{x}_i = 1$, the matching score is calculated as

$$\mathrm{score}(\mathbf{x}) = \mathbf{x}^T \mathbf{W}^{(\delta)} \mathbf{x} \tag{12}$$

Then, $\mathbf{x}$ is solved by maximizing the function

$$\mathbf{x}^* = \mathrm{argmaxscore}(\mathbf{x}) \tag{13}$$

RRW is used to obtain the soft assignment vector, in which the k-th random walk is formulated by

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \alpha \mathbf{P} \mathbf{x}^{(k)} + (1-\alpha) f_C\left(\mathbf{W}^{(\delta)} \mathbf{x}^{(k)T}\right) \\ \mathbf{P} &= \begin{pmatrix} \mathbf{W}^{(\delta)}/d_{\max} & 1 - d/d_{\max} \\ 0^T & 1 \end{pmatrix} \end{aligned} \tag{14}$$

in which $d_{\max}$ is the maximum of the column summary of each row and α is the reweight factor in the random walk. f_C contains two functions: (1) The inflation function, $\widehat{\mathbf{x}} = \exp\left(\frac{\beta \widehat{\mathbf{x}}}{\max(\widehat{\mathbf{x}})}\right)$, to improve the influence of strong nodes. (2) Bistochastic normalization via Sinkhorn algorithm (Sinkhorn, 1964) to constrain the one-to-one match. When the random walk reaches a stationary state, i.e., $\mathbf{x}^{(k+1)} = \mathbf{x}^k$, the value in $\mathbf{x}$ represents the rank score of the candidate.

RRW is optimized for GLSM because $\mathbf{W}^{(\delta)}$ is not just designed for small image patches but also for hundreds or thousands of images with large image size, which may have millions of rows and columns. A set of 3D vectors $\mathbf{w}_i = \{m, n, \Omega(m,n)\}$ is used to represent the association graph, in which m and n are the indices of the pairwise candidates and $\Omega(m,n)$ is the matching score. After dividing $\Omega(m,n)$ by $d_{\max}$, RRW can be employed efficiently with three steps.

(1) $\alpha \mathbf{P} \mathbf{x}^{(k)}$ is calculated by

$$\widehat{\mathbf{x}}_{\mathbf{w}_{i,1}}^{(k)} = \alpha \sum_{i=1}^{n} \mathbf{w}_{i,3} \mathbf{x}_{\mathbf{w}_{i,2}}^{(k)} \tag{15}$$

through which the memory space can be saved, and also the useless computation is reduced.

(2) Then, $f_C(\widehat{\mathbf{x}}^{(k)})$ is calculated with the inflation and Bistochastic normalization. Bistochastic normalization with the whole $\widehat{\mathbf{x}}^{(k)}$ takes large computations for convergence. Because the one-to-one match is just a constraint for two views, GLSM normalizes $\widehat{\mathbf{x}}^{(k)}$ separately with each image pair, e.g. if the rank scores of the matches of view i and j covers from $\widehat{\mathbf{x}}_m^{(k)}$ to $\widehat{\mathbf{x}}_n^{(k)}$, then the Bistochastic normalization is employed for $\widehat{\mathbf{x}}^{(k)}$ in the range between m and n.

(3) Finally, $\mathbf{x}^{(k+1)}$ is calculated by $\mathbf{x}^{(k+1)} = \widehat{\mathbf{x}}^{(k)} + (1-\alpha) f_C(\widehat{\mathbf{x}}^{(k)})$. There is no iteration in this process, thus can be calculated efficiently.

5.2. Candidates assignment

The assignment of $\mathbf{x}$ in two-view matching is different with that in multiple-view matching. For the two-view matching, matches can be assigned via greedy algorithm with $\mathbf{x}$, but the wrong candidate that is not contradicted with the correct will not be rejected. Because the match ranking in the head has a higher confidence than that in the back, a truncation strategy is proposed: the top T_{tru} (percent) matches are considered as reliable matches, and the rest should satisfy at least one homography, which is induced by its neighbors.

For multiple-view matching, the candidate can also be assigned like that in two-view matching. However, it is uncertain to identify which image lines represent the same 3D line segment. As presented in the pseudocode (Algorithm 1), after the assignment of each stereo via the greedy algorithm with $\mathbf{x}$, the cluster is then obtained with a mapping strategy. To obtain a reliable result, the cluster with less than T_{clu} image lines will be rejected. Note that the truncation strategy for two-view matching is not employed here.

Table 2
The parameters in GLSM.

Section	Parameters	Symbol	Value
3	Mapping error	T_{map}	2 pixels
3	Overlapping rate	T_{ove}	40%
3	Nearest neighbors	T_{nei}	40
4	Intersection distance	T_{int}	20 pixels
5	Truncation proportion	T_{tru}	70%
5	Lines in a cluster	T_{clu}	4

Algorithm 1: Clustering the image line for multiple-view matching

```
Input: Ms contains the assignment for all stereos; Ls contains all image lines; Ps
  contains all cameras; $\hat{m}$ represents the first match of a cluster; used records the usage
  of Ms.
  Output: a set C contains the image lines in the cluster.
  add $\hat{m} \rightarrow C$
  used $(\hat{m}) \leftarrow 1$
  $l3d$←resconstruct_3DLine $(C, Ls, Ps)$
  Foreach assignment $m$ in $Ms$
    if $used\ (m) = 1$
      continue
    $l_1 \leftarrow$ Map_ l3d_to_images $(l3d, Ps, Ls, m\ (1))$
    $l_2 \leftarrow$ Map_ l3d_to_images $(l3d, Ps, Ls, m\ (2))$
    if $l_1$ and $l_2$ are aligned with $Ls\ (m\ (1))$ and $Ls\ (m\ (2))$ with    Equation 3
      add $m \rightarrow C$
      $used\ (m) \leftarrow 1$
      $l3d \leftarrow$ resconstruct_3DLine $(C, Ls, Ps)$
```

6. Experiments and discussions

Our proposed GLSM algorithm is evaluated on several open source datasets shown in Table 1. Note that SURF features are extracted with the internal function of MATLAB with its default parameters, and the point candidate is initialized based on the given fundamental matrix (Please refer to Section 4.1 for the initialization) but not the texture correlation, thus there is no specific threshold in feature point extraction and initialization. Table 2 describes the parameters of GLSM, which are easy to set and fixed for all datasets. GLSM is implemented in mixed programming of MATLAB and C++. The test system is an ordinary laptop machine that is equipped with an Intel Core i7-7700HQ CPU, and 16 GB RAM.

6.1. *Aerial image patch pairs*

In this section, GLSM is evaluated on the aerial image patch pairs presented in Fig. 6. (Ok et al., 2012), and is compared with the two state-of-the-art methods, i.e., LJL (Li et al., 2016; Li, 2020) and LPI (Fan et al., 2010, 2012; Li, 2020). Because the benchmark provided by OK et al. contains not only image patches, but also the line segment and the manually-determined correct match, the result is evaluated in line-matching-level with three indicators: the ratio of the number of correct matches and the number of ground truth matches (Recall); the ratio of the number of correct matches and total matches (Accuracy); and the F-score $= \frac{2 \times Recall \times Accuracy}{Recall + Accuracy}$. Note that we say one match is correct means it exists in the benchmark.

The evaluation results are presented in Fig. 7. For the accuracy,

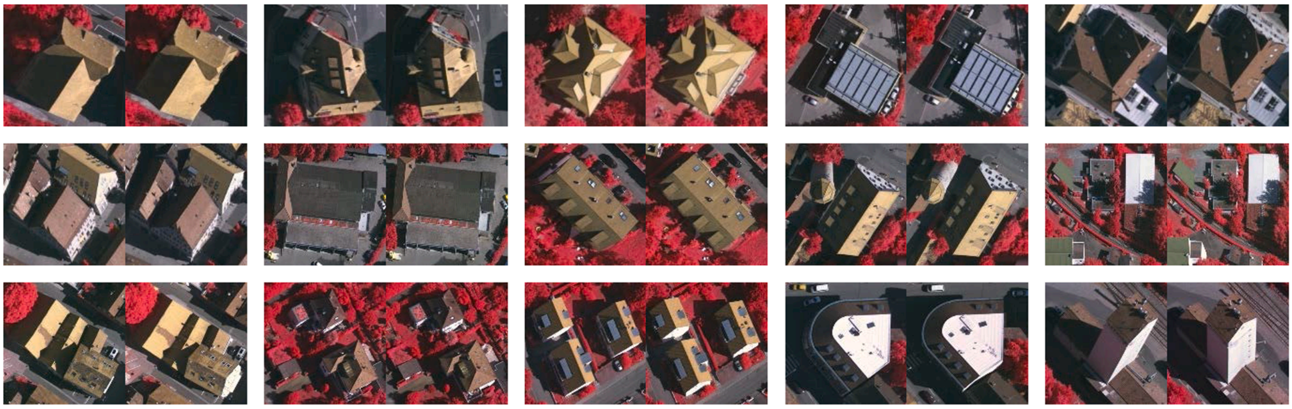


Fig. 6. The aerial image patch pairs of Vaihingen.

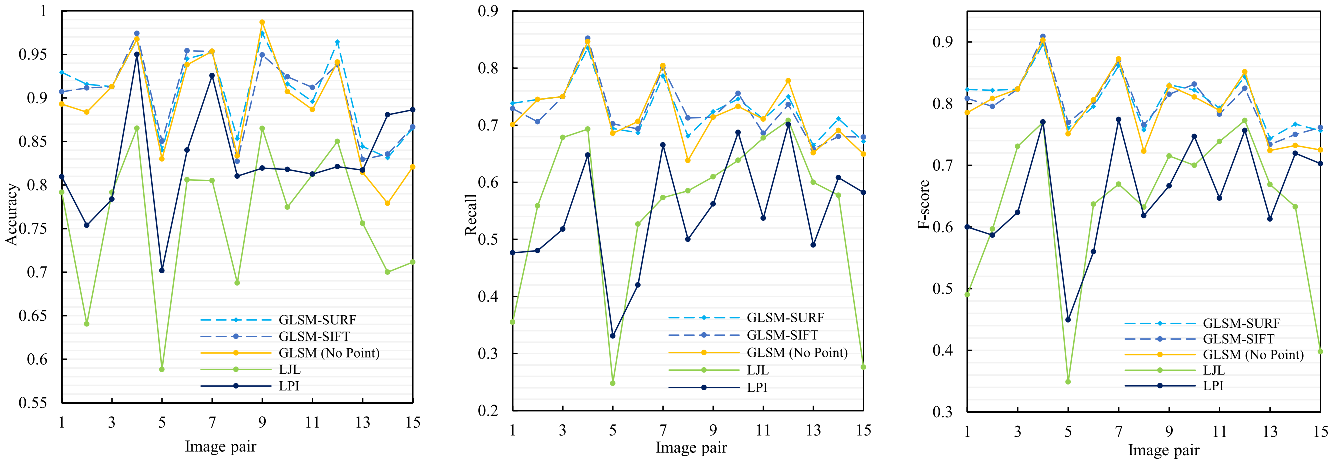
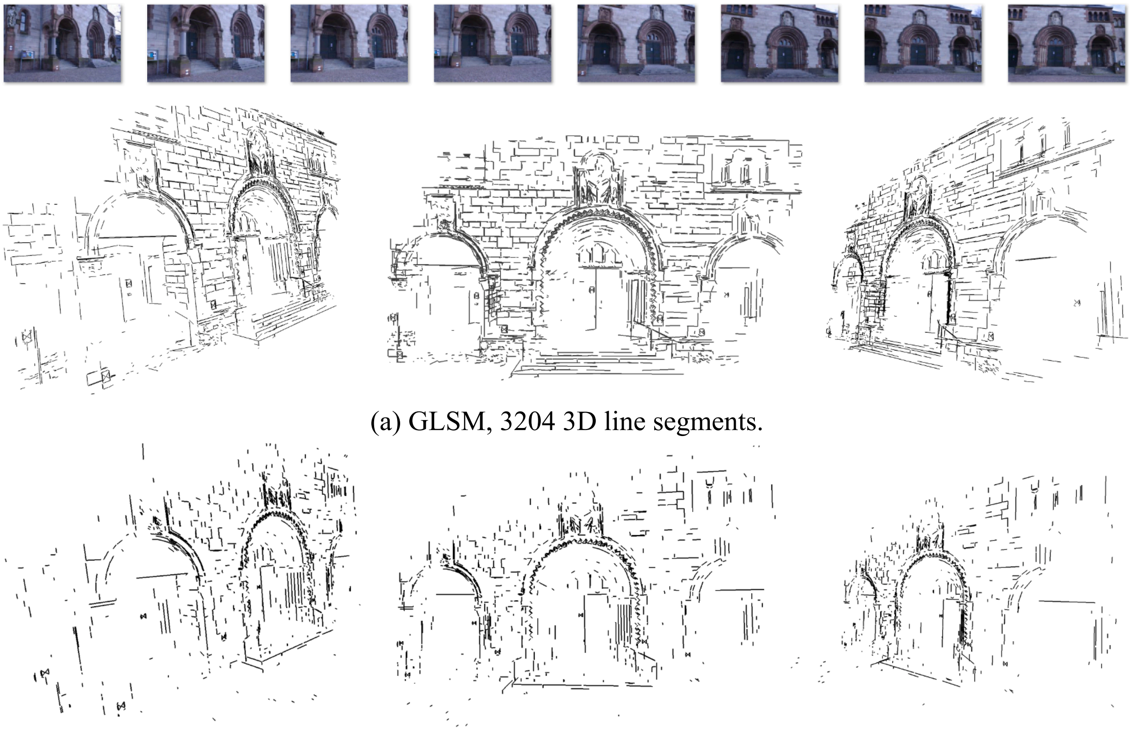


Fig. 7. The quantitative evaluation results of Vaihingen.



(a) GLSM, 3204 3D line segments.

(b) Line3D++, 1079 3D line segments.

Fig. 8. Herz-Jesu images (3072 × 2048) and the result of GLSM and Line3D++. No ristriction is set for line extraction.

GLSM ranks first in thirteen image pairs. In the last two pairs, GLSM ranks second and is lower than LPI for about 3% in average. LJL ranks third in most image pairs. For the recall, GLSM ranks first in all image pairs. LJL performs better than LPI in recall, because LPI relies on local point matches while LJL employs global epipolar geometry and the scene plane theory to add individual matches. GLSM relies mainly on the planar geometry, and obtains more correct matches than LPI and LJL in most image pairs. For the F-score, because GLSM obtains a better performance in both recall and accuracy, it ranks first in all stereo images. The evaluation of GLSM without the point assistance is shown in Fig. 7, which indicates that GLSM obtains reliable matches with only line association graph. Fig. 7 also presents the evaluation of GLSM with SIFT points (Lowe, 2004; vlfeat.org, 2020). Although the performances of matching with SURF and SIFT are different for each stereo, both of them indicate that GLSM can obtain more robust matches with the point assistance.

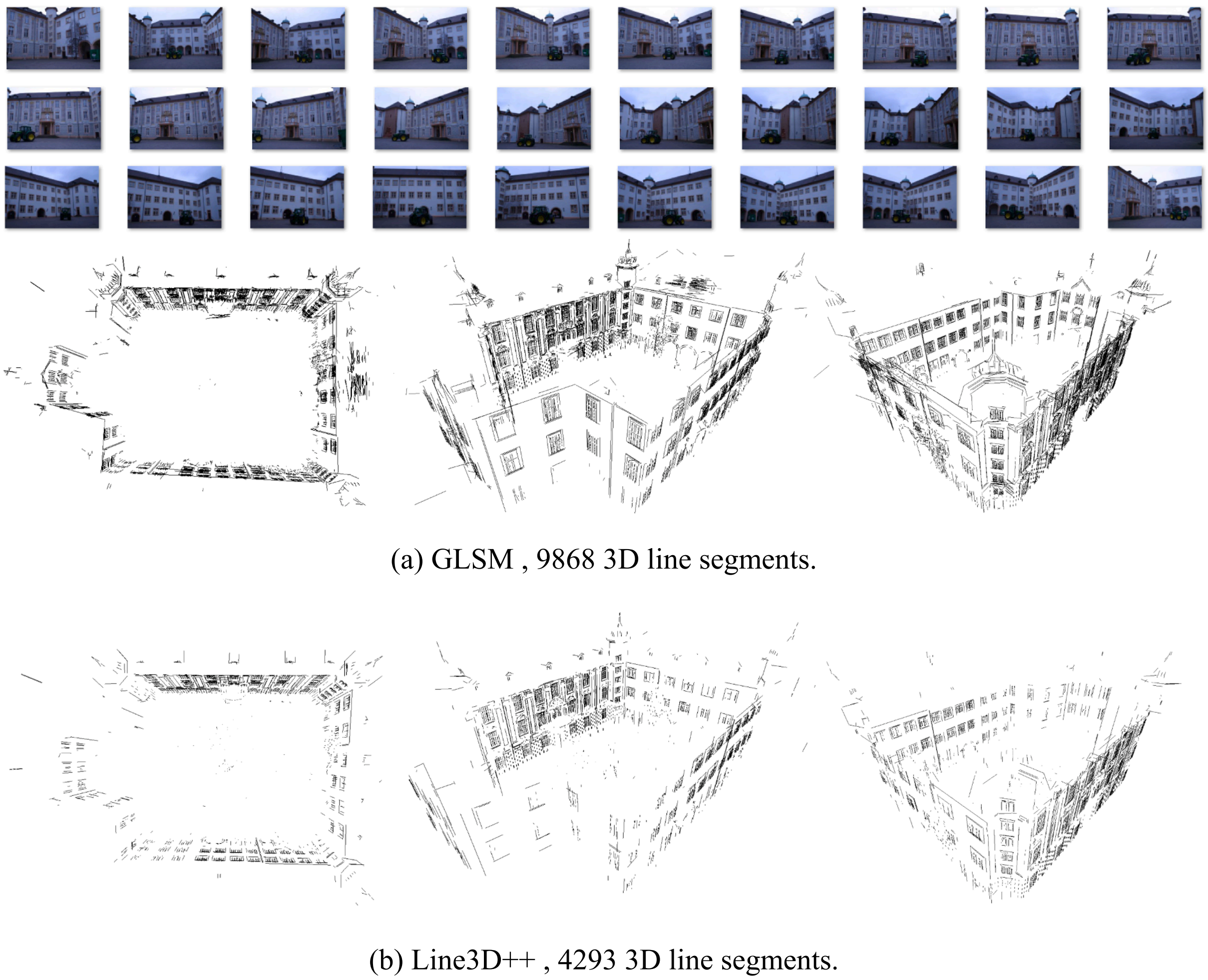
6.2. Multiple-view ground images

Two ground image datasets with high resolution (Strecha et al., 2008, 2020) are used to evaluate GLSM. Considering verifying thousands of matches for multiple views is a tough work, the 3D line segment is reconstructed based on the cluster of the matches (Algorithm 1) for intuitive presentation and evaluation.

Additionally, Line3D++ (Hofer et al., 2014, Hofer, 2020), the cutting-edge in 3D line abstraction, is employed as the baseline. Line 3D++ first initials line candidate based on the epipolar line constraint. Then, the candidate is evaluated in the 3D space based on their support in neighboring images, and the candidate in the significant 3D lines cluster is selected as the most plausible correspondence. Finally, 3D line segments are clustered based on their spatial proximity. GLSM relies on the scene plane theory, which exploits line and point candidates with coplanar assumption in the same stereo or across views to find the correct match, thus it is quite different with Line3D++, which relies on the cluster of hypothetical 3D lines and is not capable for two-view matching. Although Line3D++ employs LSD as the line segment detector, many short line segments are removed by down sampling and truncation for acceleration. For a fair compare, we have altered some internal parameters of Line3D++ to make GLSM and Line3D++ share the same parameters and the result in line segment extraction: no downsampling is employed for images, and no restriction is set on the number and the length of line segments; the others of Line3D++ remain unchanged.

3D lines lying on epipolar planes cannot be determined from their images in two views, in which case the epipoles lie on the image lines (Ok et al., 2012). Thus, the image line that is within 500 pixels to the epipole will not be used to reconstruct 3D lines. This strategy and the same parameter will also be used in Section 6.3. Note that "500 pixels" is an empirical parameter since dealing with the degeneracy in 3D reconstruction is out of the scope of this paper.

Fig. 8 displays the results of Herz-Jesu images. Because GLSM exploits the geometry constraints in both two and multiple views, it obtains three times more 3D line segments than Line3D++ that only exploits the multiple-view geometry. Fig. 9 shows the 3D Line segments of Castle. It can be seen that although GLSM also resconstructed more 3D line segments than Line3D++. GLSM reconstructs many details about the whole structure, such as doors, windows, and ridges, while Line3-D++ obtains many broken lines in the ground, which is useless for 3D reconstruction.



(a) GLSM , 9868 3D line segments.

(b) Line3D++ , 4293 3D line segments.

Fig. 9. Castle images (3072 × 2048) and the result of GLSM and Line3D++. No ristriction is set for line extraction.

6.3. Multiple-view aerial images for large areas

In this section, GLSM is evaluated on the aerial image dataset (368 images are selected randomly) for a 2 km^2 area of Dublin, Ireland that is obtained at an average flying altitude of 300 m (Laefer et al., 2020). To the best of our knowledge, no published paper in line segment matching is larger than ours in test images and areas. Fig. 10 (a) displays the ortho-image of the central area. Fig. 10 (b) illustrates the distribution of images and sparse 3D points that are generated by VisualSFM; Fig. 10 (c) shows the two shooting angles. Considering the image is large and with high resolution, the 2D line segments shorter than 50 pixels are removed in both Line3D++ and GLSM to reduce noise and computations.

Fig. 11 displays the 3D line segments reconstructed with the matches of GLSM, which can be concluded that GLSM is able to deal with high resolution aerial images of large urban areas because of the proposed procedures for RRW. GLSM obtains the structure of the mapping area, such as buildings and roads. Moreover, the details are also preserved, such as windows on the building, lamb-posts aside the road, and the boat in the river. Line3D++ also obtains the structure of the mapping area, but too many details are lost when compared with GLSM. Consequently, GLSM obtains 116,151 3D lines, which is almost two times more than that of Line3D++. Fig. 12 presents the results of four local areas. GLSM obtains many details in the buildings and roads, moreover, it is able to obtain reliable matches when images are dotted with dense line segments. As shown in Fig. 13, the four textured building models are reconstructed via TopoLAP (Liu et al., 2019), which illustrates the fact that GLSM obtains almost all the building lines, including the dense lines on the roof and the details in the window. Thus, GLSM has a great potential for improving 3D resconstruction.

6.4. Quantitative evaluation with the 3D mesh

To quantitatively evaluate the matches, we check the 3D line segments in object space about their distance to the 3D Mesh. Fig. 14 presents the 3D mesh of the three datasets in Section 6.2 and 6.3. The mesh of Herz-Jesu is available on the website (Strecha, 2020), the mesh of Castle is obtained by Photoscan with all of the images, and the mesh of Dublin is constructed from the LIDAR points provided in the dataset. Because the camera matrix of Castle and Dublin images are not aligned with the point clouds, the registration between 3D line segments and point clouds is employed in CloudCompare with manual correction and ICP optimization.

Having obtained the 3D mesh, the distance between the 3D line segment and the mesh is computed by three steps: (1) Sampling dense points on the 3D line segments; (2) computing point-to-mesh distance by CloudCompare with the “Cloud/Mesh Dist” function; and (3) taking the

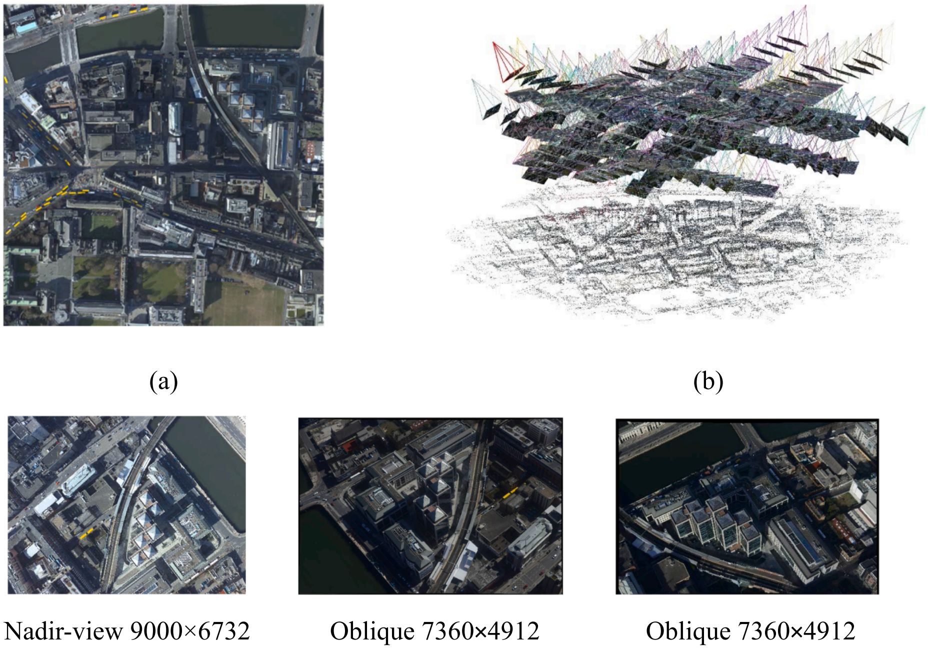


Fig. 10. The Dublin aerial image dataset. (a) The ortho-image of the central area. (b) The flight paths and sparse 3D points. (c) There are 200 oblique images and 168 nadir images, and the angle of the oblique image is about 25° to the horizontal plane.

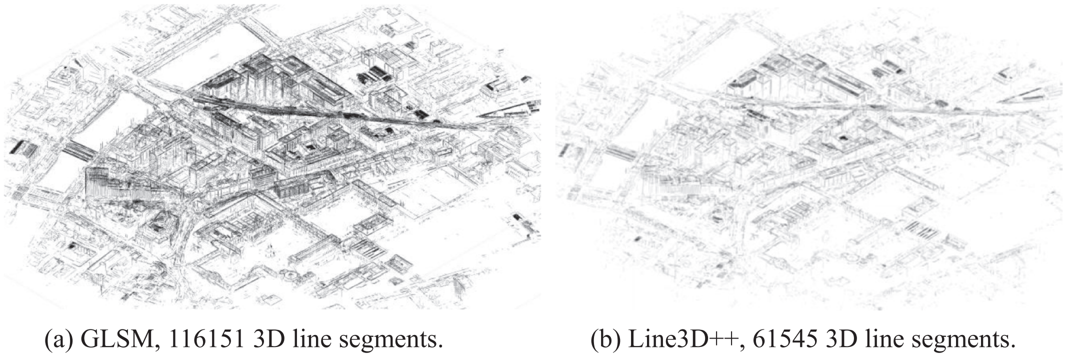

(a) GLSM, 116151 3D line segments.
(b) Line3D++, 61545 3D line segments.

Fig. 11. The 3D line segments reconstructed with Dublin image dataset. The 2D line segment shorter than 50 pixels is removed before matching.

average of point-to-mesh distances of the dense points for each 3D line segment. Note that the LIDAR points of Dublin have not covered the object space of the images, thus the 3D line segments outside the 3D mesh will not be validated (Fig. 14 (c)). Because there is no accurate distance threshold to identify whether a 3D line segment is correct, two indicators are counted: (1) the quantity of the 3D line segments in different error intervals; and (2) the proportion between the number of 3D lines within the maximum error and the number of the total segments.

Fig. 15 shows the evaluation results, in which the 3D line segment with a smaller error has a higher chance to be correct. The error of both Line3D++ and GLSM raise from Herz-Jesu to Dublin because the scene becomes larger and the range becomes longer. All the three subfigures indicate the same fact: GLSM obtains much more 3D line segments than Line3D++ while it also brings about more wrong 3D line segments, which falls into the contradictory of controlling the false positive and the false negative. Comparing with Line3D++, we think the sacrifice in controlling the false positive is acceptable, because GLSM obtains a lot more 3D line segments than the false positives when roughly using the midpoint of the horizontal axis to identify the correct. Note that there are some factors that may influence the accuracy when evaluating 3D line segments by 3D meshes: (1) There are holes in 3D mesh because the degeneracy of LIDAR (Fig. 14 (a)) or dense point matching (Fig. 14 (b)). (2) The 3D line segment is usually at the edge of an object, which is usually the coarsest part of the point cloud (Fig. 14 (d)).

6.5. *Discussions*

The experiments have manifested three advantages of GLSM: (1) It has matched line segments with a satisfactory accuracy. (2) It has been

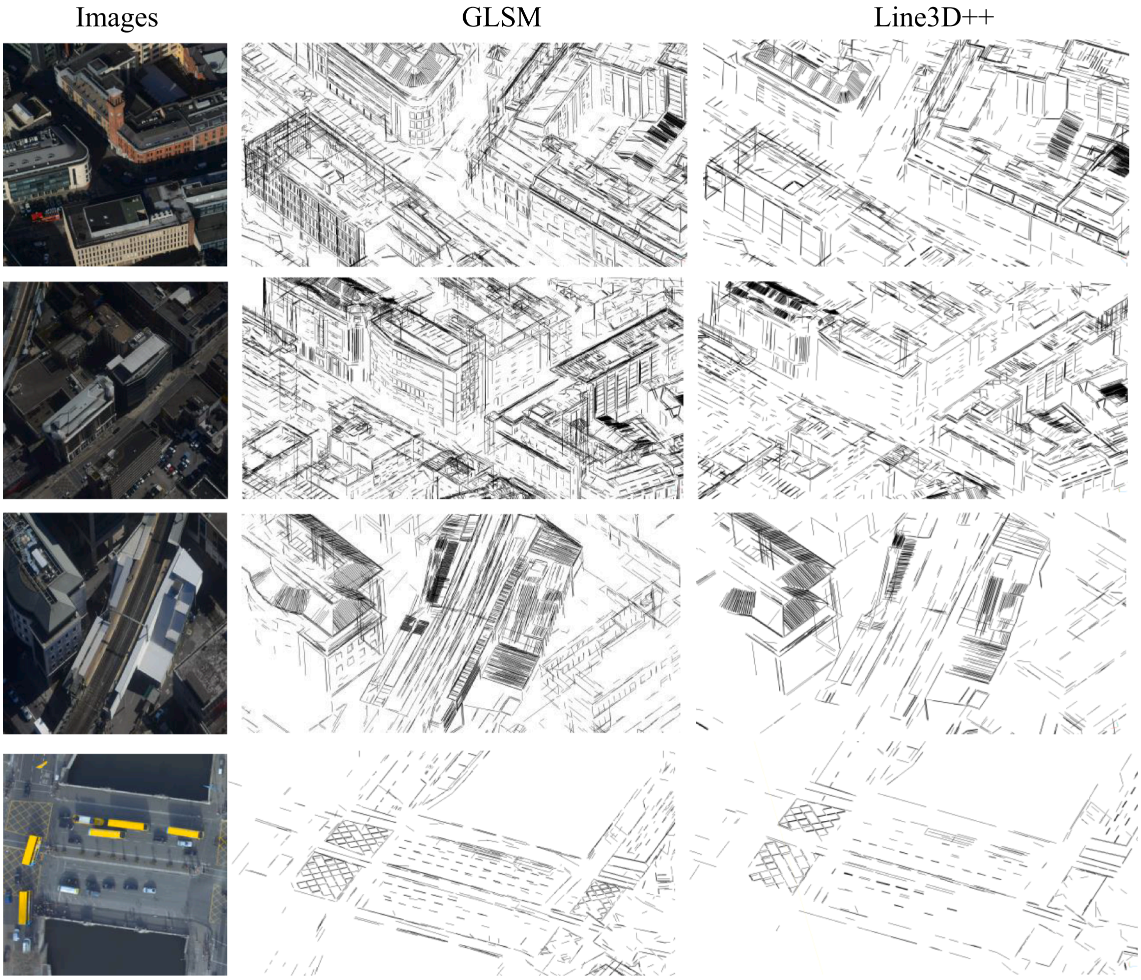


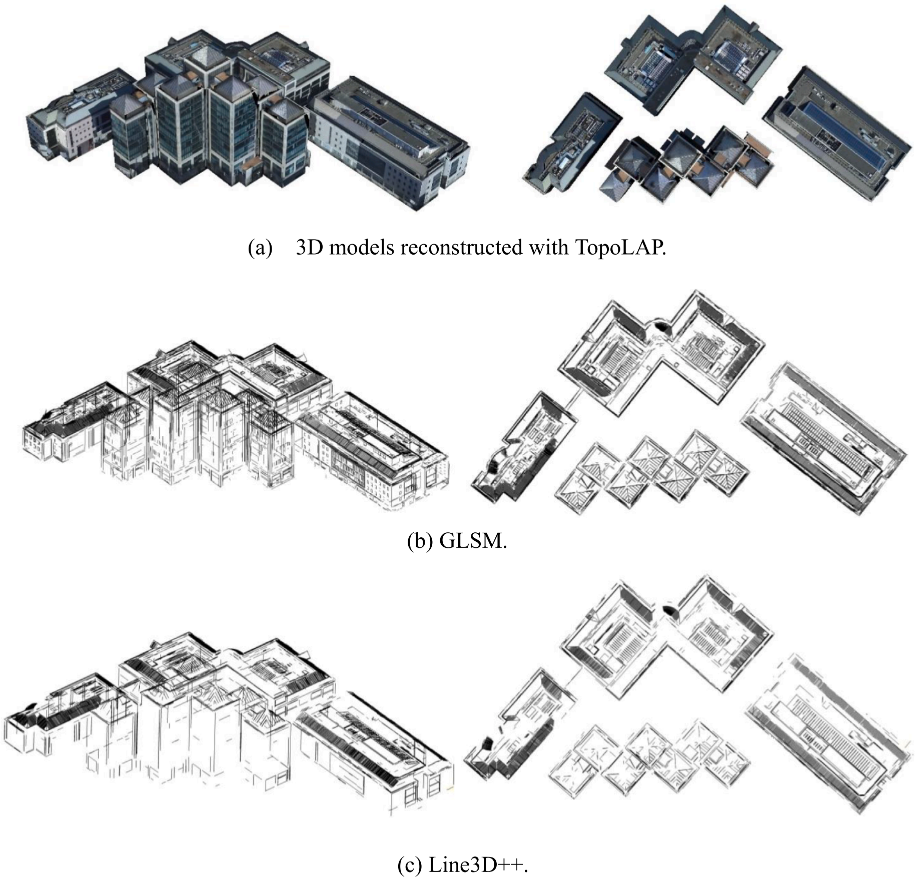
Fig. 12. Local areas and 3D line segments.

workable for large images and areas, as well as small stereo patches. (3) It has obtained much more details about the scene structure than that of Line3D++. These advantages come with four reasons: (1) GLSM makes full use of the geometry constraint among features, and is able to confirm matches in two views. (2) The line-point graph unifies the geometry constraint across views, which is able to obtain global optimal matches with graph ranking. (3) The measures to save computations and memories are proposed for the employment in large areas and datasets. (4) Line3D++ only exploits the cluster of hypothetical 3D lines, which requires sufficient images to confirm a real 3D line.

However, there are also some current limitations in GLSM. Running speed is the most obvious one, because half of the code is written by MATLAB (The rest is written by C++ and called by MATLAB) and no GPU acceleration is implemented. As to the small image dataset like image patches in Section 6.1, GLSM has run in four seconds for each image pair. For the castle images in Section 6.2, GLSM has run about half an hour to obtain the match, and has took another half an hour to cluster 3D lines with Algorithm 1. For the Dublin images in Section 6.3, GLSM has took more than one day to obtain matches and 3D line segments. Thus, compare with Line3D++ that is with GPU acceleration and only takes a minute for Castle images and within half an hour for Dublin images, GLSM manifests slowly in term of the speed. Additionally, the core geometry constraint of GLSM is under the assumption that the line segment is coplanar with some neighbors. Thus, it can be concluded that GLSM is more suitable to deal with manmade scenes.

7. Conclusions

This paper presents a novel line segment matching algorithm based on the geometry constraint of different features across views. The proposed algorithm exploits the pairwise line constraint, line-point constraint, and triplet constraint to construct the line-point graph, and the reweighted random walk method is employed efficiently on this graph to rank the candidate. Finally, the robust matches are obtained with the greedy algorithm, and a cluster method is employed to obtain 3D line segments from multiple images. The experiments demonstrate the robustness of our algorithm when compared with the state-of-the-art methods, and our algorithm can be employed for both stereo patches and large area image datasets. However, the running speed of the proposed algorithm need to be improved for large image datasets, thus, we will speed up the proposed algorithm with GPU in future works. In addition, considering this paper focuses on the matching across multiple views and only takes one available stereo-dataset to evaluate the performance of two-view matching, another study is in process to sufficiently evaluate the performance in different kinds of complex stereos.

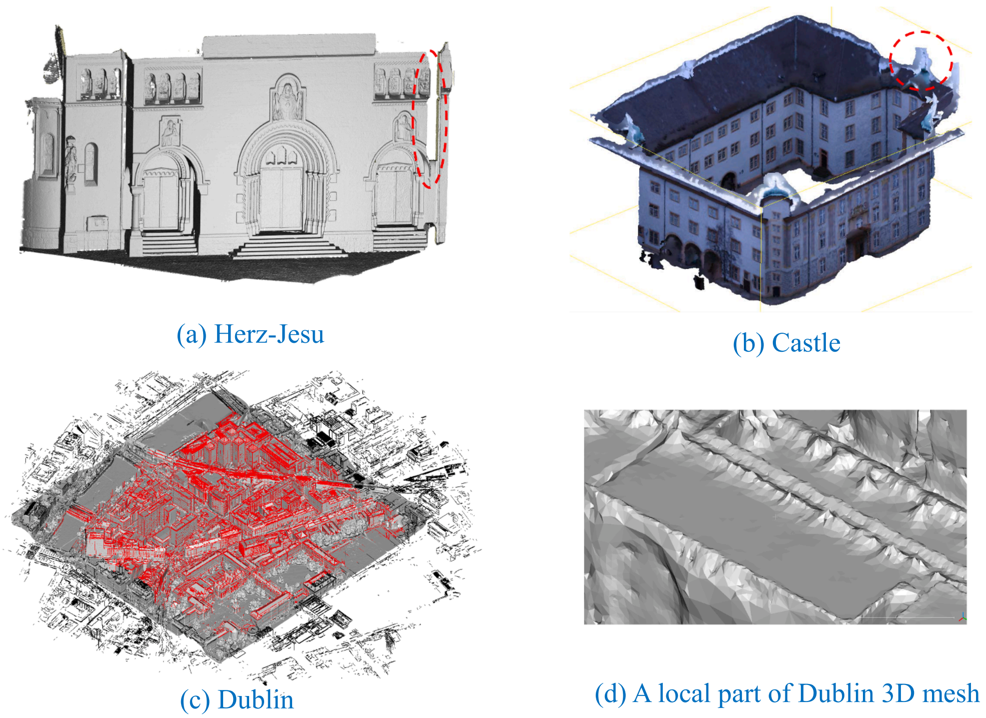


(a) 3D models reconstructed with TopoLAP.

(b) GLSM.

(c) Line3D++.

Fig. 13. The 3D models and 3D line segments of the four buildings.



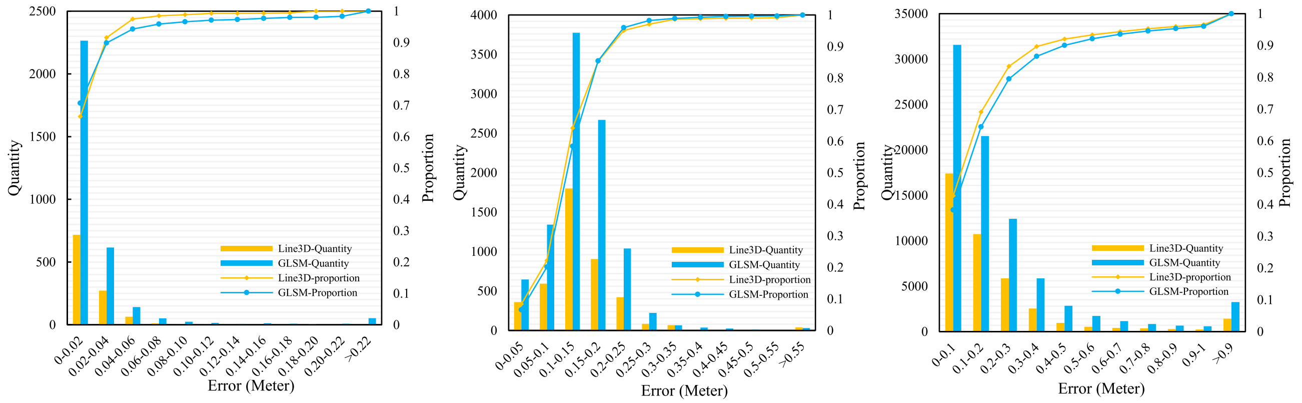
(a) Herz-Jesu

(b) Castle

(c) Dublin

(d) A local part of Dublin 3D mesh

Fig. 14. The 3D mesh of the scenes in Section 6.2 and 6.3. (1) The red ellipses in (a) and (b) mark the example of the holes in the 3D mesh. (2) Only the red 3D line segments in (c) are counted because the LIDAR points do not cover the scene in the image sequence. (3) As shown in (d), the 3D mesh is coarse at the edge of the building, which may influence the accuracy of the evaluation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Herz-Jesu (b) Castle (c) Dublin

Fig. 15. Evaluation of the error between 3D points and meshes. GLSM is superior than Line 3D++ in the quantity of 3D line segments while it brings about more false positives.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 41871368 and 41771493, and Huawei Technologies Co., Ltd. under Grant YBN2018095106.

We would like to thank the authors Li et al. (2016), Hofer et al. (2016), and Fan et al. (2010) to share their codes, and thank the author Ok et al. (2012) for their kind reply and providing the aerial image dataset with the ground truth. They are quite helpful for our work in this paper.

References

Al-Shahri, M., Yilmaz, A., 2014. Line matching in wide-baseline stereo: A top-down approach. IEEE Trans. Image Process. 23 (9), 4199–4210.

Bay, H., 2005. Vittorio Ferrari and Luc J. Van Gool. Wide-Baseline Stereo Matching with Line Segments, CVPR.

Bay, H., 2006. Tinne Tuytelaars and Luc Van Gool. Speeded Up Robust Features. ECCV, SURF.

Beder, Christian, 2004. A Unified Framework for the Automatic Matching of Points and Lines in Multiple Oriented Images. ISPRS Congress.

Cho, M., 2010. Jungmin Lee and Kyoung Mu Lee. Reweighted Random Walks for Graph Matching, ECCV.

Laefer, Debra F., Abuwarda, Saleh, Vo, Anh-Vu, Truong-Hong, Linh, Gharibi, Hamid, 2020. 2015 Aerial Laser and Photogrammetry Survey of Dublin City Collection Record. From https://geo.nyu.edu/catalog/nyu_2451_38684.

Fan, B., Fuchao, Wu, Zhanyi, Hu, 2010. Line Matching Leveraged by Point Correspondences. CVPR.

Fan, B., Fuchao, Wu, Zhanyi, Hu, 2012. Robust line matching through line-point invariants. Pattern Recogn. 45 (2), 794–805.

Furukawa, Yasutaka and Jean Ponce. Accurate, Dense, and Robust Multi-View Stereopsis. CVPR, 2007.

Hartley, Richard, Zisserman, Andrew, 2008. Multiple view geometry in computer vision. Kybernetes 30 (9/10), 1865–1872.

Hofer, M., 2020. "Line3D++ Source Code." from https://github.com/manhofer/Line3Dpp.

Hofer, M., Maurer, M., Bischof, H., 2014. Improving sparse 3d models for man-made environments using line-based 3D reconstruction. International Conference on 3D Vision.

Hofer, Manuel, Maurer, Michael, Bischof, Horst, 2016. Efficient 3D scene abstraction using line segments. Comput. Vis. Image Underst. 157, 167–178.

Jain, Arjun, Kurz, Christian, Thormählen, Thorsten, Seidel, Hans Peter, 2010. Exploiting global connectivity constraints for reconstruction of 3D line segments from images. CVPR.

Jia, Xiangyang, Huang, Xianfeng, Zhang, Fan, Gao, Yunlong, Yang, Chong, 2019. Robust line matching for image sequences based on point correspondences and line mapping. IEEE Access 7, 39879–39896.

Lee, Jungmin, Cho, Minsu, Kyoung Mu Lee, 2011. Hyper-graph matching via reweighted random walks. CVPR.

Leordeanu, Marius, Hebert, Martial, 2005. A spectral technique for correspondence problems using pairwise constraints. ICCV.

Li, Chang, Zhang, Yongjun, Zhang, Zuxun, 2016a. Automatic keyline recognition and 3D reconstruction for quasi-planar façades in close-range images. Photogrammetric Record 31 (153), 29–50.

Li, Kai, 2020. Line Segment Matching: A Benchmark. Retrieved 8, 2020, from https://kailigo.github.io/projects/LineMatchingBenchmark.

Li, Kai, Jian, Yao, 2017. Line segment matching and reconstruction via exploiting coplanar cues. ISPRS J. Photogramm. Remote Sens. 125, 33–49.

Li, Kai, Yao, Jian, Xiaohu, Lu., Li, Li, Zhang, Zhichao, 2016b. Hierarchical line matching based on line-junction-line structure descriptor and local homography estimation. Neurocomputing 184, 207–220.

Liu, Xinyi, Zhang, Yongjun, Ling, Xiao, Wan, Yi, Liu, Linyu, Li, Qian, 2019. TopoLAP: Topology recovery for building reconstruction by deducing the relationships between linear and planar primitives. Remote Sens. 11 (11), 1372.

López, Juan, Santos, Roi, Fdez-Vidal, Xosé R., Pardo, Xosé M., 2014. Two-view line matching algorithm based on context and appearance in low-textured images. Pattern Recogn. 48 (7), 2164–2184.

Lowe, David G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60 (2), 91–110.

Luong, Q.T., Viéville, T., 1994. Canonic Representations for the Geometries of Multiple Projective Views. Springer, Berlin Heidelberg.

McAuley, Julian J., De Campos, Teófilo, Caetano, Tibério S., 2010. Unified graph matching in Euclidean spaces. CVPR.

Moulon, Pierre, Monasse, Pascal, Marlet, Renaud, et al., 2013. Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion. ICCV.

Ok, Ali Ozgun, Wegner, Jan Dirk, Heipke, Christian, Rottensteiner, Franz, Soergel, Uwe, Toprak, Vedat, 2012. Matching of straight line segments from aerial stereo images of urban areas. ISPRS J. Photogrammetry Remote Sens. 74(6), 133–152.

Ok, Ali, Wegner, Jan Dirk, Heipke, Christian, Rottensteiner, Franz, Soergel, Uwe, Toprak, Vedat, 2012b. Accurate reconstruction of near-epipolar line segments from stereo aerial images. Photogrammetrie Fernerkundung Geoinformation 2012 (4), 345–358.

Page, Lawrence, Brin, Sergey, Motwani, Rajeev, Winograd, Terry, 1999. The Pagerank Citation Ranking: Bringing Order to The Web. the web conference.

Schmid, Cordelia, Zisserman, Andrew, 1997. Automatic line matching across views. CVPR.

Schmid, Cordelia, Zisserman, Andrew, 2000. The geometry and matching of lines and curves over multiple views. Int. J. Comput. Vision 40 (3), 199–233.

Shi, Jiacha, Wang, Xuanyin, 2017. A local feature with multiple line descriptors and its speeded-up matching algorithm. Comput. Vis. Image Underst. 162, 57–70.

Sinkhorn, Richard, 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. Ann. Math. Statistics 35 (2), 876–879.

Snavely, Noah, Seitz, Steven M., Szeliski, Richard, 2008. Modeling the world from internet photo collections. Int. J. Comput. Vision 80 (2), 189–210.

Strecha, Christoph, 2020. Strecha Dense MVS. From https://documents.epfl.ch/groups/c/cv/cvlab-unit/www/data/multiview/denseMVS.html.

Strecha, Christoph, Von Hansen, W., Van Gool, L., Fua, Pascal, Thoennessen, U., 2008. On benchmarking camera calibration and multi-view stereo for high resolution imagery. CVPR.

Sun, Yanbiao, Zhao, Liang, Huang, Shoudong, Yan, Lei, Dissanayake, Gamini, 2015. Line matching based on planar homography for stereo aerial images. ISPRS J. Photogramm. Remote Sens. 104, 1–17.

vlfeat.org., 2020. SIFT Implementation, 2020, from https://www.vlfeat.org/.

Von Gioi, R.G., Jakubowicz, Jeremie, Morel, Jeanmichel, Randall, Gregory, 2010. LSD: A fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Mach. Intell. 32 (4), 722–732.

Wang, Qiang, Zhang, Wei, Liu, Xiaolong, Zhang, Zhenxin, Baig, Muhammad Hasan Ali, Wang, Guo, He, Long, Cui, Tiejun, 2020. Line matching of wide baseline images in an affine projection space. Int. J. Remote Sens. 41 (2), 632–654.

Wang, Zhiheng, Fuchao, Wu, Zhanyi, Hu, 2009. MSLD: A robust descriptor for line matching. Pattern Recogn. 42 (5), 941–953.

Wu, Changchang, 2013. Towards Linear-Time Incremental Structure from Motion. International conference on 3d vision.

Yang, Xu, Liu, Zhi Yong, 2017. Adaptive graph matching. IEEE Trans. Cybernetics, pp. 1–14.

Zhang, Lilian, Koch, Reinhard, Jvcir, J., 2013. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. J. Vis. Commun. Image Represent. 24 (7), 794–805.

Zhou, Feng, Fernando De la Torre, 2012. Factorized Graph Matching. CVPR.