IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 21, 2019, accepted March 11, 2019, date of publication March 20, 2019, date of current version April 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906368

Robust Line Matching for Image Sequences Based on Point Correspondences and Line Mapping

XIANGYANG JIA, XIANFENG HUANG, FAN ZHANG, YUNLONG GAO, AND CHONG YANG

State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

Corresponding author: Xianfeng Huang (huangxf@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 41571437, and in part by the National Key Technologies Research and Development Program of the Ministry of Science and Technology of China under Grant 2014BAK07B04.

ABSTRACT This paper proposes a line segment matching method by performing line mapping and unmapping based on point correspondences. The goal of this paper is to improve the accuracy and the robustness of line segment matching for two views, which will be conducive for generating a full single-line structure for image sequences. In this paper, to improve the quantity and quality of line matches, the topological adjacency of a point-line is first introduced for two goals: to find and filter the candidate line segment by resorting to KD-tree data-index structure efficiently in the corresponding image and to leverage the candidate line segment to improve the performance of planar homography. In addition, the shift threshold parameter is theoretically analyzed, and trials are validated to determine the matching degree. Line mapping and unmapping are then used for image sequences to reduce missing matches. The extensive experimental results validate and demonstrate that our method is both more accurate and robust than existing line matching methods for two views under the circumstance of a higher recall rate. In addition, our method contributes in finding full line segments for image sequences using line mapping, with higher completeness of the 3D line model than that obtained by the state-of-the-art methods.

INDEX TERMS Line mapping, line segment matching, planar homography, topological adjacency.

I. INTRODUCTION

Line segment matching [1]–[3] is a challenging and fundamental area with extensive applications, including 3D scene reconstruction, scene recognition and image merging. The above applications can be found in diverse fields, such as reverse engineering, computer vision and photogrammetry, where feature points and line matching are the focuses of current research. Feature points play an important role in 3D scene reconstruction [4], but they cannot usually capture geometric information such as the outlines of buildings, windows, and doors. Compared with feature points, line segment information can more fully describe structural details and complement image line features [5], [6]. For example, a box is clearly recognizable when using line segments [7]. Therefore, full line segment matching has great research significance and application value.

However, line segments are typically not comprehensively extracted by existing line segment detection methods, which sometimes divide a 3D line segment into multiple fragments in different image views [8]. A fragmented single-line segment is more difficult to match than feature points [9], [10]. This problem not only increases the probability of mismatches and missing matches but also fails to deliver good performance on a full line structure.

This paper proposes a novel line-matching method based on image sequences to decrease the ratios of mismatches and missing matches and generate a full line segment using line mapping and unmapping. The workflow of the proposed method is shown in Fig.1. First, the planar homography is computed through pairs of keypoints in the vicinity of the line segment, which can be determined by the topological adjacency of the point-line. Candidate line segments are then efficiently found in the corresponding image using descriptors consisting of keypoints by the KD-tree data-index approach. The line segment is then mapped onto the corresponding image using the planar homography matrix, and the candidate line segments are evaluated by the shift threshold to determine the matched degree. For image sequences, a line unmapping approach with inverse transformation is proposed

The associate editor coordinating the review of this manuscript and approving it for publication was Naveed Akhtar.

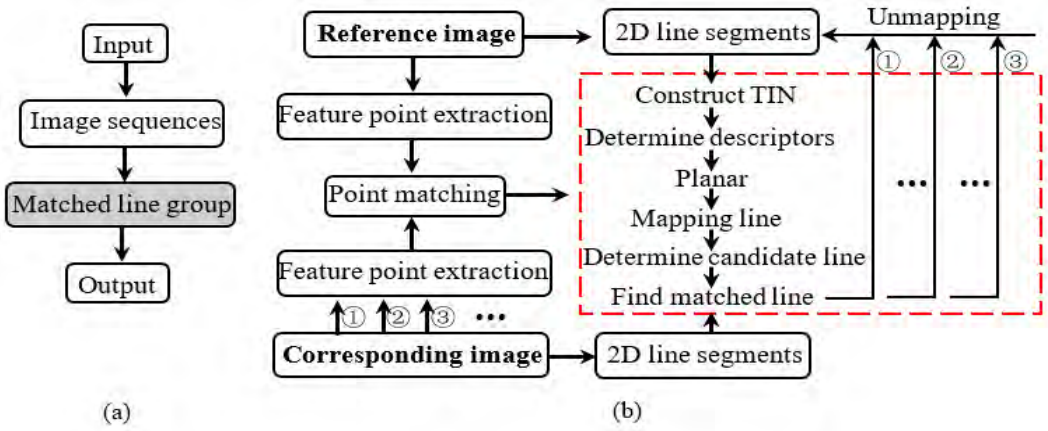

FIGURE 1. The workflow of line matching (a) denotes the whole pipeline and (b) represents the stage of the matched line group, in which the red dash rectangle denotes the line segment mapping process from the reference image to the corresponding image sequences and the determination of the matched line. TIN denotes triangulated irregular network. ①, ② and ③ in (b) represent the serial numbers of the corresponding images, respectively.

to overcome incomplete line structure and reduce missed matches.

This study is based on two research foundations. The first is the keypoint matching algorithm [11], [12], whose robustness and accuracy have been greatly improved in the past three decades. Researchers have also attempted to find an optimal algorithm [1], [13]–[17] based on the scale invariant feature transform (SIFT) proposed by Lowe [18] to achieve the goals of scale invariance and algorithm robustness. The Affine-SIFT algorithm, which is a prominent example of a SIFT variant and is also used in our experiments, solves the nonrobustness issue that occurs in affine transformation caused by changing viewpoints. The second foundation is the line segment detection algorithm. This algorithm, named line segment detection (LSD) [19], adopts the pixel-merging approach and is both widely used and effective. Here, this paper focuses on line segment matching based on the LSD algorithm because this algorithm performs well for detecting line segments without requiring tuning parameters for multiscale images [20].

Our contributions are as follows. To further improve the mismatches and missing matches of line matching, this paper makes some contributions to line matching from the following three perspectives: 1) building a 2D point-line topological adjacency, which can improve the performance of computing the planar homography matrix based on matched keypoints, to find the candidate line segments resorting to the KD-tree efficiently (Part A of Section IV); 2) theoretically analyzing and validating the shift threshold for filtering candidate line segments to reduce mismatches and applying wide-/short-baseline or scale changes with accurate and robust performance (Part B of Section IV); and 3) introducing line mapping for image sequences to integrate all fragments into full segments, which not only decreases the possibility of missed matches but also completes the details of the 3D line model (Part C and Part D of Section IV).

The remainder of this study is organized as follows. Section II introduces related work on line segment matching. Section III defines the parameters used to describe the algorithm and the method. Section IV introduces the main processes and the algorithm of the proposed method. Section V presents the experimental results of comparisons with state-of-the-art methods. Finally, Section VI presents the discussions, conclusions and future work.

II. RELATED WORK

Researchers and experts have always aimed to improve the accuracy and robustness of line segment matching through experiments. The existing matching methods can be divided into two classes [6], [21]: individual line segments, where similarity is calculated based on the gray levels around the line segment, length and direction, and line-line or point-line group matching based on topological constraints.

In 1995, Tavares *et al.* examined the similarities between line segments by considering Mahalanobis normalized distances or geometric constraints and estimating the values of direction, length and midpoint position. In 1997, Schmid and Zisserman [21] combined the gray level of an individual line segment with multiview geometric constraints and determined the homography between two views based on coplanar curves and geometric characteristics. These methods are highly dependent on the overlap between the corresponding line segments. In 2000, Schmid and Zisserman [23] investigated the correctness of line matching using a third corresponding view that differed from the two views. However, this method fails on wide-baseline images with low overlap. In 2002, Werner and Zisserman [24] provided an implementation of the Schmid and Zisserman technique and made the source code available. However, whether this method works well depends on the quality of the auxiliary information, and it is also sensitive to occluded areas. In 2005, Bay *et al.* [25] proposed a line matching method for highly textured scenes that obtained an initial set of line segment correspondences

iteratively. However, this method fails on images with little texture. Therefore, in 2013, Witt and Weltin [26] proposed the iterative closest multiple lines (ICML) algorithm with weight matching criteria to find the optimal matches considering neighborhood lines, which can be applied to weak texture scenes and sparse point regions with robustness. However, the ICML algorithm requires a rectified image and a disparity for stereo matching of 140 pixels or less. In 2011, Elaksher [6] proposed a matching hypothesis using multiview geometric constraints and evaluated the similarity of line segments between corresponding arrays using the correlation coefficient. The problem of matching failures due to inaccurate endpoints can be solved, but failure is inevitable because the level of correlation in the matching window decreases when the illumination or viewpoint changes. In 2012, Ok *et al.* [27] used probability density functions and seven relational pairwise constraints to obtain more robust matches. In 2016, Gao *et al.* [28] combined the disparity map in the rectified space with planar homography to determine candidate line segments; they then removed incorrect segments from the candidate pool based on the geometric information of the individual line segments. This method requires a longer runtime to calculate the density functions and seven relational constraints. In 2017, to reconstruct a 3D scene line-based model with good quality, Hofer *et al.* [5] exploited weak epipolar constraints to establish line segment correspondences and then evaluated them by directly analyzing their 3D similarities. Unfortunately, this method requires prior knowledge of spatial regularizers.

In 2009, Wang *et al.* [8] used the SIFT algorithm as the basis for building the line-line description group and selected the algorithm with the highest similarities as the correct match. This method does not require prior information or constraint conditions; however, the method's shortcomings are its nonrobustness and low accuracy under conditions involving scale changes. In 2009, Wang *et al.* [29] proposed the LS algorithm to match line segments from corresponding wide baseline views. Line segment matching is extremely sensitive to endpoint inaccuracies due to the limitations of calculating similarity based on the length and direction of line segments, thereby increasing the mismatch probability. In 2014, Al-Shahri and Yilmaz [30] exploited epipolar geometry to leverage intersections and endpoints of the line-pair to remove ambiguous line-pair matches. They then mapped a line-pair to the corresponding image to compute the projected error and constructed an affinity matrix of the geometric similarity metric to provide the final line matches. In 2015, Sun *et al.* [2] triangulated coarse matching points to 3D points and examined the correctness of homography based on planar homography, which filtered mismatches by line length, consistency of point-line topology and projective invariance. However, the given threshold in this method has no theoretical basis and applies only to aerial orthogonal images. Fan *et al.* [31] used an existing point matching algorithm and point-line topology to determine the correct matching line in the query image. The method was proposed in 2010 under the

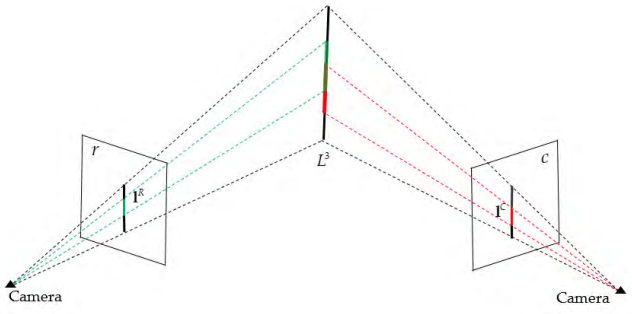

FIGURE 2. Illustration of line fragments generated by inaccurate detection of image line segment: the red and green lines are different parts of L^3, which is a real 3D structure line, and correspond to l^R and l^C in the reference and corresponding image, respectively. The solid black triangles represent the position of the camera. (Illustration by [5]).

circumstance of affine-only invariance. In 2012, Fan *et al.* [9] complemented the projective invariant in the original method. They analyzed four different combinations based on two point-line invariants and two types of similarity measures through extensive experiments. Their conclusion showed that combining the affine-invariant and similarity measure based on the maximal median delivered the best performance. However, the search range of keypoints depends on the length and position of the perpendicular line. Therefore, this matching approach often fails due to the deficiency of keypoints in the support region generated by short line segments. Moreover, this approach is time consuming. Lopez *et al.* [32] proposed a novel line detection algorithm in the scale-space pyramid in which the real perceived line corresponding to fragments is generated and then applied to the subsequent line matching process in low-texture scenes to improve the matched line quality. However, the process costs increase for rich texture scenes due to increasing line neighbors and the involvement of some important thresholds (e.g., minimum distance between endpoints and parallel or perpendicular distance) in line matching without theoretical analysis. In 2016, Li *et al.* [33] and Li and Yao [34] evaluated the Euclidean distance between V-junction descriptors to match line segments; then, they matched individual line segments by estimating the local homography. Nevertheless, this method assumes that adjacent line segments possess a higher probability of being coplanar in 3D space, which causes this method to fail when the outline of a building or window is matched. Subsequently, Li *et al.* [35] proposed a line-junction-line (LJL) structure descriptor to match line segments using a hierarchical method in multi-scale pyramids of an image, which demonstrated good performance on most scenes, including poorly textured ones. Obviously, this method is very time consuming, especially for large-scale scenes. Jia *et al.* [36], [37] developed the characteristic number, a new projective invariant, to match line segments with the line-point invariant (LPCN), which is not disturbed by mismatched points of interest and even generates more matches from a shorter fragment across views than [31] and [32]. However, the accuracy of line matches declines for shorter fragments, which are usually noisy [38]

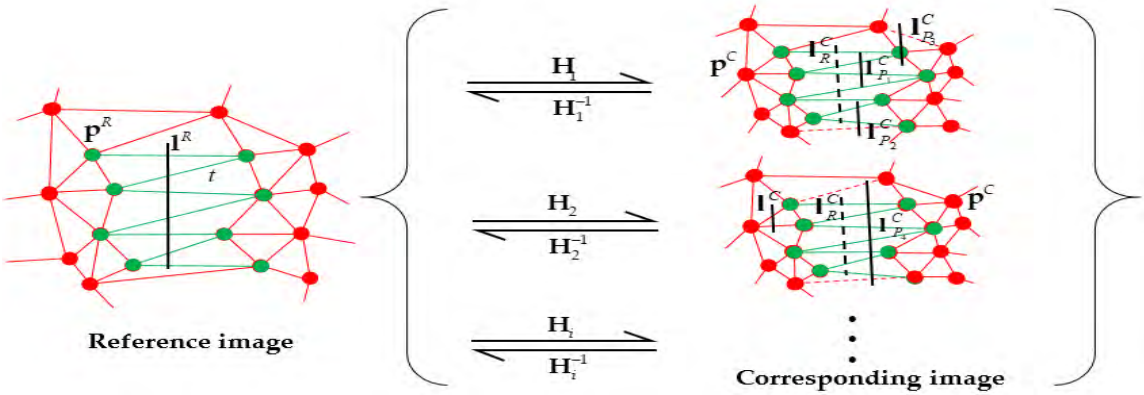

FIGURE 3. Illustration of the process of distinguishing candidate line segments using TIN (namely, t in the reference image): the red circles represent feature points that do not belong to the line descriptors. The green circles represent the endpoints of the line descriptors. The solid red lines show the edges of the triangulated network intersecting without $\mathbf{l}^R$ or $\mathbf{l}^C_P$, and the dotted red lines exist only in C due to the inaccurate endpoints of the line segments or the inevitable L^C_P. The green lines denote D^R or D^C. $\mathbf{H}_i(\mathbf{H}_i^{-1})$ represents the homography matrix (and the inverse of the homography matrix). $\mathbf{p}^R$ and $\mathbf{p}^C$ denote the keypoints in the reference and corresponding images, respectively, and $\mathbf{l}^C$ denotes a noise line segment. All symbols are also defined in Section III.

under high matches, and it can be seen from the computational complexity $O(kNM^2)$(the parameters are illustrated in [37]) that this method is also time consuming for large-scale scenes.

III. DEFINITIONS

In this section, for convenience, some notations are first defined. The L^3 in Fig.2 represents a 3D space line segment in the real world. Two types of image views, namely, reference and corresponding images, are available in the matching and mapping processes of every line segment. The former is represented as $R = \{r_i,\ i = 1,2,3, ...\}$ and the latter as $C = \{c_i,\ i = 1,2,3, ...\}$. The line segment[1] is denoted as $L^R = \{\mathbf{l}^R_i,\ i = 1,2,3, ...\}$ in R and as $L^C = \{\mathbf{l}^C_i,\ i = 1,2,3, ...\}$ in C. Inevitably, line segment matching generates tentative segments (also termed candidate line segments), which are denoted as $L^C_P = \{\mathbf{l}^C_{P_i},\ i = 1,2,3, ...\}$. Here, $L = \{\mathbf{l}_i,\ i = 1,2,3, ...\}$ refers to the 2D line segment, and L^3 denotes the 3D line segment. The two endpoints of the line segment are denoted as $\mathbf{e}^i (i = 1,2)$. To reduce the number of candidate line segments and improve the efficiency of determining matched line segments, this study introduces TIN, which is denoted as $T = \{t_i, i = 1,2,3, ...\}$ [39]. An expression for the topology of a point-point or point-line is provided and constructed for the keypoints in the image view. Thus, $P^R = \{\mathbf{p}^R_i,\ i = 1,2,3, ...\}$ and $P^C = \{\mathbf{p}^C_i,\ i = 1,2,3, ...\}$ represent the keypoints[2] in R and C, respectively. To search the candidate line segments, this study defines the line descriptor that intersects with the line segment and denotes it as D^R and D^C in R and C, respectively, as shown in Fig.3. The line segments that intersect with the line descriptors (the solid green lines in Fig.3) are selected as the candidate line segments. For example, as shown in Fig.3, $\mathbf{l}^C_{P_3}$, which passes through the green circle, is added to L^C_P even though it does not rigorously intersect the line descriptors; however, $\mathbf{l}^C$ is not added because it does not intersect with the line descriptors. In addition, please note that the subscript denoting the index number is often omitted unless it would be ambiguous for reading and understanding.

[1]Line segment is denoted as (A, B, C) from line: $Ax + By + C = 0$.
[2]All point coordinates in this paper are homogeneous.

IV. LINE SEGMENT MAPPING MATCHING

Given that a line segment consists of sequential points and possesses the properties of those points, one can rely on research results that address robust point correspondence instead of viewing line segment matching as an independent study. However, a line segment not only has the properties of its points but also its own unique characteristics. For example, points are dimensionless (zero dimension), but a line segment can be described by its length and direction (one dimension). Therefore, a full line segment in the physical world might be divided into several parts due to scale, viewpoint or illumination changes, as shown in Fig.4(left), which represents the full line segment in the physical world, and Fig.4(a, b, c and d), which shows the different parts belonging to a real line. The inaccurate locations of endpoints and the existence of multiple different parts of a single line might cause failure to match the line segment with high quality and quantity [9].

In this section, this approach is introduced mainly to improve the robustness and accuracy of line matching based on point correspondences using the planar homography and topological adjacency of the point-line. Subsequently, based on the shift threshold, as analyzed from the LSD method theoretically, the details of line mapping and unmapping approaches are introduced. The following subsections introduce the key technologies of the proposed method.

A. ESTABLISHING CANDIDATE LINE SEGMENTS

Because only the planar metric is considered, this paper resorts to planar homography of the line segment to connect the image pairs conveniently, which can be computed from

FIGURE 4. Structure line segments: The left-hand image contains a line segment that is not broken. The a, b, c and d fragments show several fragments due to changes in scale, viewpoint and illumination. Matching these fragments in different views is difficult due to the lack of exact endpoints. For example, $\mathbf{l}_i$ ($i = 1, 2$ or 3) in *a*, *b*, *c* and *d* represent parts of line segment **l**.

the point correspondences obtained from keypoint matching. However, the correlation and probability of coplanar points and lines increases as the distance decreases. Another important factor is that the accuracy of homography becomes worse at points farther from the line [40]. That is, the precision of homography transformation would disturb the line matching directly if these points are only near the line segment. In other words, to improve the line segment match rate to the maximum probability, it is necessary to find the closest points to the line segment. For example, in Fig.3 (reference image), one can see that the red circles are farther from the reference line segment than the green circles. Hence, compared with the green circles, these red circles are unsuitable for estimating the homography.

In this proposed method, the TIN is considered to represent the topological adjacency in the image and planar surface, which can search the line that is topologically near the line segment. The TIN of the keypoints is built to describe the point-line topology and determine that the reference descriptor (D^R) lies in the triangle intersecting the reference line segment. Subsequently, one can determine the corresponding descriptor (D^C) in the corresponding image directly based on keypoint correspondences constructed by the mature Affine-SIFT algorithm. The TIN has two functions in the proposed method: the first is to determine the keypoints near the line segment and remove noisy line segments using point-line topology, such as $\mathbf{l}^C$ in Fig.3, which is removed using TIN. The second is to improve the accuracy of the planar homography matrix based on the topological adjacency of the point-line.

A line segment usually occurs between two intersecting planes. Due to the inaccurate keypoint correspondence and the long distances between line segments and keypoints, the 3D points generated by corresponding keypoints located on one side of a line segment may not be conducive to fitting into a real and precise plane. That is, whether the keypoints lying on the left and right of the line segment are coplanar is unknown. Thus, the RANSAC [41] algorithm is exploited to check the reliability of the fitted plane and remove those noise points iteratively until the real plane is produced. If successful, the planar homography can be obtained using Equation 1 [40]:

$$\mathbf{x}' = \mathbf{H}\mathbf{x} \tag{1}$$

where $\mathbf{x}'$ and $\mathbf{x}$ are the coordinates of the keypoints in the reference and corresponding image, respectively. In addition, the mapped line segment $\mathbf{l}_R^C = ((\mathbf{e}_R^C)^1, (\mathbf{e}_R^C)^2)$ is provided by $\mathbf{l}^R = ((\mathbf{e}^R)^1, (\mathbf{e}^R)^2)$ in the corresponding image by Equation 2:

$$(\mathbf{e}_R^C)^i = \mathbf{H}(\mathbf{e}^R)^i, \quad i = 1, 2 \tag{2}$$

in which the symbols are introduced in Section III. In this process, to calculate the planar homography successfully, the number of keypoints lying to the right or left of the line segment should be no fewer than 4, and the planar homography transformation may fail when three of the keypoints are collinear.

For the corresponding image, the opposite process is conducted compared with that in the reference image: the candidate line segment will be determined by the descriptors in the corresponding image. The above process of establishing line segment correspondence removes most of the noisy line segments using point-line topology. Additionally, the KD-tree data-index structure is used to improve the computation time for finding descriptors and candidate line segments, which was inspired by Sun et al. [2]. The vertices and center of each triangle in the TIN are built in the KD-tree, as are the endpoints and midpoints of the line segments. Note that they are two different KD-trees for the reference and corresponding image. Subsequently, we select the midpoint of the line segment and its length as the target and search radius respectively to find the triangles by traversing the KD-tree in the reference image. Similarly, in the corresponding image, the center of the minimum bounding rectangle (MBR) of the line descriptors and the diagonal line of the MBR are used as the anchor and search range to determine the candidate line segment using the other KD-tree. The KD-tree is exploited only to approximately determine triangles (candidate line segments) to reduce the computation time, and then we filter the remaining noises by intersecting with the line segment (line descriptors) in the reference image (corresponding image).

B. THEORETICAL ANALYSIS OF LINE SEGMENT DETECTION

When the endpoints of line segments are accurate in the orthogonal direction, the mapped line segment will partially or completely overlap with the matched line segment if it exists. However, due to the errors resulting from keypoint extraction and matching and from line segment extraction, the ideal mapped line segment may not appear. To determine a line segment from all candidate line segments, the common solution to find a matching line segment is to set a constant threshold with an empirical value to remove noisy line segments for a given image. However, the subjectivity

and limits of this solution are obvious. Selecting a too-large threshold will produce a mismatch towards a candidate line segment that is not a corresponding line segment, and a too-small value will prevent the seed line segment from being matched. To prevent this problem, it is necessary to improve the correct matching rate and the number of matches and select a reasonable threshold.

For a gray-level image, each pixel or point corresponds to a specific gray value that is quantized and therefore inaccurate. In a natural image, straight edges usually have a gray-level transition corresponding to many pixels, which is particularly thick due to the quantization. Desolneux [42] applies the Shannon principles [43] for an image to a level line with a constant gray level. In other words, the level line can collect the neighboring points with the same gray level and contribute to the geometric structure. Additionally, an a-contrario noise model for the Helmholtz principle is used to determine the meaningful boundaries and edges of infinite details in the finite resolution image. In the LSD algorithm, rectangles, which denote line segments with a certain width, are involved in describing the straight geometric structure on the basis of the above mentioned a-contrario noise model where pixels at a distance larger than two pixels are independent. That is, whether less than two pixels at a distance lie in the line segment or not depends on the rule of the LSD algorithm to the maximum extent.

The gradient orientation orthogonal to the level line is computed by the neighbor (right-, down- and downright-) pixels, which is the simplest local contrast invariant information. In this transition field, the contrast of the gradient level would be small compared with the highly contrasted edges at the meaningful boundary. Note that the final segment is from the meaningful segments in the middle by the exclusion principle, which denotes that any two alignments cannot overlap. Likewise, the line-support region proposed by [44] is introduced into the LSD algorithm to describe the criteria for detecting line segments using several parameters (e.g., image scale, balance value, gradient threshold, angle error, aligned point density and number of false alarm thresholds).

However, some of those parameters are unable to improve line segment extraction from most images. This is evident by reports such as "works better sometimes before and sometimes after the width refinement, and there is no serious caveat in performing both ", as stated in [20]. Hence, in this study, only the image scale and balance value, which play prominent roles, are considered. The LSD algorithm usually select half-pixels as a balance value. Following [43] and [45], the gray-level distribution is Gaussian or uniform in the noise image, while the direction is always uniform distribution. Therefore, the triple balance value [46] is determined as a believable threshold conservatively; when subsequently changed back to full scale (from 0.8 to 1.0), approximately two pixels are obtained, the result of which coincides with that of the a-contrario noise model.

Assumption that there is no error for computing planar homography using the point correspondences and occlusion, the matched line segment should be found in two pixels if it exists. Nevertheless, if the threshold is not reasonable to distinguish two segments, two cases might occur: one is that two segments at a distance of less than two pixels are not matched, and the other is that two segments at a distance of greater than two pixels are actually a matched pair. Given inaccurate quantization of image, the first case can be excluded. The other occurs only if the assumption is rejected. Therefore, the final threshold is relaxed by half-pixels to solve the second case through the different types of scene experiments, the details of which are demonstrated in Part A of Section V. As the shift threshold increases, the mismatching probability of the line segment increases, especially for the line segment for which matches do not actually exist. The final threshold could give attention to both precision and recall rate and then match line segment(see Part A of Section V).

C. DETERMINE THE MATCHED LINE SEGMENT

As mentioned above, the LSD algorithm offers a meaningful alignment threshold and is applicable to any scale and scene without tuning the value. Hence, one can find matched line segments using the orientation and fixed position shift between the candidate and mapped line segments. Then, the orientation between the candidate line segment and mapped line segment can be computed by Equation 3 where θ represents the angle between $\mathbf{l}_{P_i}^C$ and $\mathbf{l}_R^C$. To calculate the position shift between $\mathbf{l}_{P_i}^C$ and $\mathbf{l}_R^C$, the means of the distances, including those between the two endpoints of $\mathbf{l}_{P_i}^C$ and $\mathbf{l}_R^C$ and between the two endpoints of $\mathbf{l}_R^C$ and $\mathbf{l}_{P_i}^C$, are evaluated by Equation 4, where d denotes the position shift, d_1, d_2, d_3, and d_4 denote the four distances, and $d_1 = \frac{|(\mathbf{e}_{P_i}^C)^1 \cdot \mathbf{l}_R^C|}{\sqrt{(A_R^C)^2+(B_R^C)^2}}$, $d_2 = \frac{|(\mathbf{e}_{P_i}^C)^2 \cdot \mathbf{l}_R^C|}{\sqrt{(A_R^C)^2+(B_R^C)^2}}$, $d_3 = \frac{|(\mathbf{e}_R^C)^1 \cdot \mathbf{l}_{P_i}^C|}{\sqrt{(A_{P_i}^C)^2+(B_{P_i}^C)^2}}$ and $d_4 = \frac{|(\mathbf{e}_R^C)^2 \cdot \mathbf{l}_{P_i}^C|}{\sqrt{(A_{P_i}^C)^2+(B_{P_i}^C)^2}}$, in which $(\mathbf{e}_{P_i}^C)^1$, $(\mathbf{e}_{P_i}^C)^2$, $(\mathbf{e}_R^C)^1$, $(\mathbf{e}_R^C)^2$ represent the endpoints of $\mathbf{l}_{P_i}^C$ and $\mathbf{l}_R^C$, respectively.

$$\theta = arccos \frac{|\mathbf{l}_{P_i}^C \cdot \mathbf{l}_R^C|}{|\mathbf{l}_{P_i}^C||\mathbf{l}_R^C|} \tag{3}$$

$$d = \frac{1}{4} \cdot (d_1 + d_2 + d_3 + d_4) \tag{4}$$

D. LINE SEGMENT MAPPING AND UNMAPPING

It is inevitable that the LSD algorithm will generate line fragments because the gradient orientation and level-line angle change when the image scale changes. A long spatial line in an image contains more pixels than a short spatial line at the same scale; thus, a long spatial line is more likely to be detected as many fragments. The existence of such fragments increases the probabilities of mismatching and missed matches.

To solve this problem, this paper proposes line segment mapping and unmapping to integrate such fragments into full line segments based on image sequences. Significantly, the proposed method not only improves the number of line

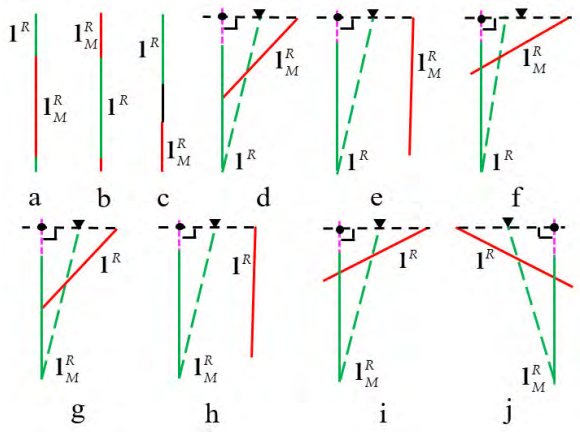

FIGURE 5. Schematic of updating line segment endpoints: a-c denote three cases ideally and d-i represent the approaches to update the new line segment (the green dash line segment) when the endpoint's position or orientation of fragment in case c is inaccurate. The black solid and dash line represents the overlap parts and biases between $\mathbf{l}_M^R$ and $\mathbf{l}^R$, respectively. The black dot represents the intersection point of $\mathbf{l}^R$ ($\mathbf{l}_M^R$) and the black dash line located in the extended line of $\mathbf{l}^R$ ($\mathbf{l}_M^R$)(the purple dash line). The black solid triangle denotes the midpoint of the black dot and the endpoint of $\mathbf{l}_M^R$ ($\mathbf{l}^R$), which is also the endpoint of the new line segment. Note that the green solid line is longer than the red solid one in cases d-j.

matches but also produces more full-line segments. First, the line segment is mapped from R to C to determine the matched line ($\mathbf{l}_M^C$), which is then unmapped into R to update $\mathbf{l}^R$. The method aims to collect fragments in the different views iteratively and solve the problem of one-to-many matching in one image to complete the line segment structure gradually. This strategy is helpful for improving the missing matches rate and 3D line structure using topological adjacency. In this unmapping process, our strategy is to map $\mathbf{l}_M^C$ to R as $\mathbf{l}_M^R$ and merge $\mathbf{l}_M^R$ and $\mathbf{l}^R$ as a new line in R using planar homography. Ideally, $\mathbf{l}_M^R$ and $\mathbf{l}^R$ should be completely collinear along the fragment orientation with the different lengths and endpoint positions (see Fig.5(a-c)). Due to the inevitable position and orientation errors between $\mathbf{l}^R$ and $\mathbf{l}_M^R$, the endpoints of the new line segment are updated using the mean values of their endpoints, a simple approach to adjusting bias, to produce new endpoints. In Fig.5(a-b), the new line segment is updated using the longer fragment (e.g., $\mathbf{l}^R$ in case a and $\mathbf{l}_M^R$ in case b) since the shorter fragment is more likely to be noise [38]. In Fig.5(c), there are six cases in which the endpoints of the line segment must be updated according to the relative position between $\mathbf{l}^R$ and $\mathbf{l}_M^R$: a schematic is shown in Fig.5(d-i), in which cases d-f denote that $\mathbf{l}^R$ is longer than $\mathbf{l}_M^R$ and cases g-i demonstrate the opposite cases. Please note that these six cases apply to their symmetric circumstance (e.g., in Fig.5(i) and (j)). When the line segment is regenerated, more fragments are produced by the line detection algorithm to be matched, and a real and full single line will be obtained.

E. ALGORITHM IMPLEMENTATION

This study employs two main procedures to achieve line segment matching. The first procedure involves keypoint extraction, matching and line segment detection from the image. The Affine-SIFT method is used in the experiments to find additional keypoints to support this algorithm. The 2D point correspondences are then filtered through the double cross-direction consistency method, in which the coarsely matched point-pairs in R and C are checked as references for each other. To refine matching, the process of filtering 2D point correspondences is considered in this paper. The LSD algorithm is used to extract the feature line from the image because it performs better than the Hough transform [47] and the line detection method introduced by Wang *et al.* [29]. The second step is line mapping and unmapping. This approach requires no parameter tuning (the orientation is set to $cos\theta = 0.99$, and the position shift is set to 2.5 pixels), excluding special instructions. The pipeline of the algorithm is shown in Algorithm1.

Assume N_t denotes the number of center and vertex of TIN and N_e as the midpoints and endpoints of line segments. For each view, the KD-tree is constructed with the worst-case complexity $O(N_t log^2 N_t)$ for TIN and $O(N_e log^2 N_e)$ for line segments, and the worst performance on the range search is $O(N^{1/2})$, in which N represents the maximum number of N_t and N_e. Thus, the computational complexity of this proposed method is approximately the maximum between $O(MN^{1/2})$ and $O(Nlog^2 N)$, in which M represents the number of line segments.

V. EXPERIMENTS AND ANALYSIS

All experiments were executed on a computer running a Windows 7 operating system with 1.6 GHz. In the proposed method, the shift threshold has a strong effect on the line segment matching accuracy. Hence, in the experiments, original images of five different types of scenes were selected to analyze the relationship between the line-matching precision and recall rate (PR curve) for the different shift values. Additionally, to analyze and compare the efficiency and accuracy of the proposed method with the existing line-matching algorithms, three comparison experiments were designed. The first experiment, belonging to the first class (Section II), compares the proposed method with the method of Schmid [21], [23] using the three datasets from Werner and Zisserman [24]. The second experiment, for the second class (Section II), analyzes the accuracy and time cost of line segment matching using the abovementioned five types of scenes. The last experiment focuses primarily on comparing the total matches, correct matching rate, match completeness and line segment completeness for image sequences of the proposed method with that of the state-of-the-art 3D scene reconstruction method (Line3D++ software) using a 3D line-based model.

All the line segments used in the experiment were detected using the same line segment extraction algorithm (LSD), and only those greater than 1% of the image diagonal line remain to be matched since shorter fragments are usually noise and do not represent line features. Furthermore, shorter fragments are usually not supported by at least two other views [38]

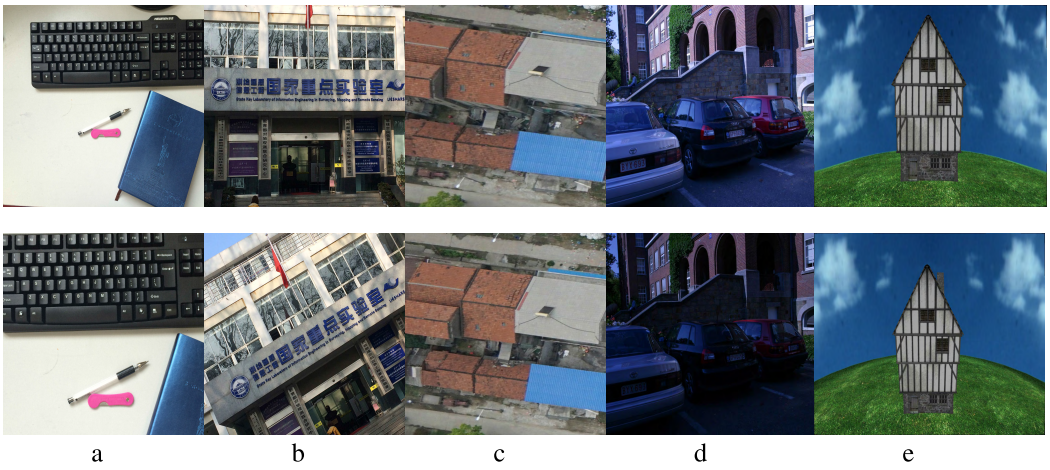

FIGURE 6. Images with four transformation types and UAV images.

and thus would not play an active role in reconstructing the 3D lined model or other models. The experiments analyze the running efficiency (T), total matches (TM), mismatches (MM), extracted 2D segments (S2D), total of 3D lines (L3D) and correct matching rate (CMR) for comparison. Note that all the matched line segments are drawn into the original image with different colors to aid distinguishing the different matched pairs, and a label is created with the image serial number near one endpoint of the line segment to conveniently check for mismatches.

A. THRESHOLD TRIALS AND VALIDATION

In this proposed method, the accuracy of line segment matching depends heavily on the shift threshold. A too-small threshold parameter results in missed matches, while a too-large threshold parameter produces mismatches. Although the maximum similarity is introduced into the process of filtering the candidate line segments to determine the matched line after the larger threshold is applied, mismatches still exist. In addition, the accuracy of keypoint extraction and matching and the planar homography are also critical factors in line matching, but this problem can be controlled within an acceptable range based on the mature Affine-SIFT and LSD algorithms.

In this experiment, five different scenes, including four types of transformation and unmanned aerial vehicle (UAV) images, are selected as shown in Fig.6 to validate the robustness of the proposed method. These images were captured from a desk (Fig.6(a)) and entrance (Fig.6(b)) in our laboratory; the image in Fig.6(c) was acquired from a district in Hubei Province; the image in Fig.6 was introduced by [9] and [31], and the image in Fig.6(e) comes from the Hofer [5] algorithm and is a synthetic case. In addition, a portion of the UAV image (733×833) is selected as the research area to simplify checking and aid in demonstrating clearly in Part C of Section V. Table 1 lists some of the details of the experimental dataset, including the types and sizes of the images, the total

TABLE 1. The datasets of the five scene descriptions.

Scene	Type	Size	Keypoints	Total of segments	
				Ref	Corr
a	scale	3264×2448	7956	1231	1080
b	rotation	1632×1224	31577	1550	1512
c	UAV image	733×833	3946	855	790
d	illumination	900×600	6057	826	486
e	viewpoint	1280×960	6591	370	311

[1] "Ref": denotes the reference image.
[2] "Corr": denotes the corresponding image.

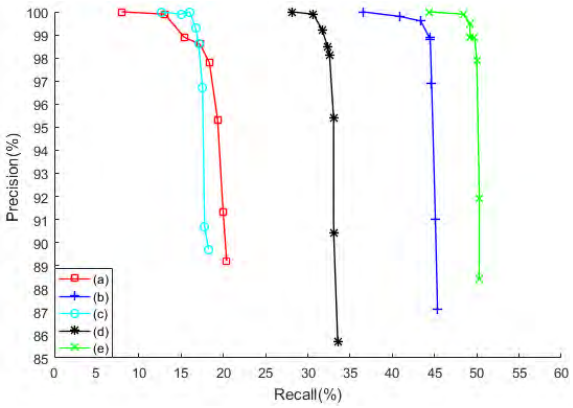

FIGURE 7. The different color curves represent the PR curves of the five scenes. Each curve varies with the range of different threshold sets of 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 5.0 and 6.0 pixels, in which the smaller the threshold, the higher the precision and the lower the recall rate.

number of keypoint pairs based on the Affine-SIFT algorithm and the line segments detected by the LSD algorithm.

The line-matching rate is counted and analyzed, given accurate keypoint extraction and matching and line segment detection, for the above scene datasets at full scale (in Fig.6). The proposed method under the different shift threshold settings is validated using the PR curve. The ideal line-matching PR curve would fall in the top-right corner with high precision and recall rate. As shown in Fig.7, the best line matching

Algorithm 1 Line Segment Matching for Image Sequences

Require: A sequence of images
Ensure: The matched line group Ω
Preliminary:
1. Keypoint extraction, coarse/refined matching and structure line extraction;
2. Generate TIN;
3. Construct the KD-tree for line segments and TIN;

for $i = 0$ to sum of images **do**
  for $j = 0$ to sum of line segments **do**
    Initialize $t = 1$;
    while t **do**
      Initialize $t = 0$ && $d_0 = 2.5$
      for $k = 0$ to 10 nearest neighbors of images **do**
        Determine the descriptors D^R of image i;
        Find the 2D keypoints correspondence;
        Determine the descriptors D^C of image j;
        Divide all the keypoints into two parts by the line segment;
        Refine the corresponding keypoints of each part using RANSAC;
        if the sum of keypoints is less than 4 **then**
          break;
        else if three existing keypoints are collinear **then**
          break;
        end if
        Find L_P^C;
        Map $\mathbf{l}^R$ from image i to image j as $\mathbf{l}_R^C$;
        for $m = 0$ to sum of L_P^C **do**
          Compute the orientation Θ and position shift between $\mathbf{l}_R^C$ and $\mathbf{l}_{P_m}^C$;
          if $fabs(cos\Theta) \geq 0.99$ and $fabs(d) \leq 2.5$ and $d_0 > d$ **then**
            $d_0 = d$;
          end if
        end for
        if $d_0 < 2.5$ **then**
          Add $\mathbf{l}_{P_m}^C$ into Ω and mark it as "Use "if $\mathbf{l}_{P_m}^C$ meets the condition;
          Map $\mathbf{l}_{P_m}^C$ to R if $\mathbf{l}_{P_m}^C$ meets the condition;
          $t = 1$ && break;
        end if
      end for
    end while
    Map a new $\mathbf{l}^R$ to the matched line segment in the Ω;
  end for
end for

results could be produced using 2.5 pixels for five different scenes.

B. COMPARISON WITH SCHMID

In this section, the proposed method is compared with the Schmid method, which is the classical method in the first class. Werner [48] implemented the method proposed by Schmid, and an implementation version of this approach is available in the MATLAB toolbox.[3] Hence, their scenes and the matching projection matrix are used to conduct the comparison experiment. The line segment matching results are listed in Table 2 (bold font denotes the top-rank performance on the corresponding metric among all line-matching methods in this table and in Table 3 and Table 4).

TABLE 2. Comparison of the proposed method with Schmid.

Scene	Item	Merton	Merton-a	Valbonne
Schmid	TM	248	396	**153**
	CMR(%)	87.9	85.1	84.3
Schmid+	TM	227	378	144
	CMR(%)	88.5	87.3	79.2
Our	TM	**331**	**553**	107
	CMR(%)	**99.7**	**99.8**	**98.1**

[1] "Schmid+": denotes the Schmid method with epipolar ordering constraints.

One can see that the proposed method has better performance regarding the correct matching rate and achieves higher completeness than the Schmid method either with or without epipolar ordering constraints, especially for the correct matching rate. There are two reasons for our method's lower completeness on the Valbonne scene. The first is the lack of keypoints caused by the low resolution, and the second one is the occurrence of many sharp edges in the keypoint distribution area. Hence, when there are sufficient numbers of keypoints, the proposed method achieves higher accuracy on line segment matching. The matched line segments from the proposed method are drawn in the corresponding image pairs (Fig.8-Fig.10 demonstrate the extracted line segments, total matches and correct matches) using different colors. Please note that in this experiment, the processing time was not evaluated because the implementation of Schmid with and without epipolar beam ordering constraints runs on the MATLAB platform, which is different from the proposed method.

C. COMPARISON WITH MSLD, LPI, PHLM, LJL AND LPCN

In this section, the proposed method is compared with the MSLD [8], LPI [9], PHLM [2] and LJL [35] algorithms using the same keypoint pairs and line segments. LPCN [37] only runs with the same line segments, and the keypoints are produced by the shared code available in [37] since a large number of keypoints would be very time consuming. In addition, please note that Method IV, which is the general case and the most robust of all methods in LPI, is applied to our experiment. In addition, only Case I of the PHLM method is used for this comparison because Case II is suitable only for approximate orthogonal aerial images. The matching results are shown in Table 3, where one can see that the proposed method performs well in terms of the completeness of line segment matching, correct matches and time consumption.

[3]The source code and dataset of the Schmid algorithm are available at http://cmp.felk.cvut.cz/cmp/software/lmatch.

FIGURE 8. Line matches of Merton: Extracted line segments: 786, 757. Total matches: 331. Correct matches: 330.

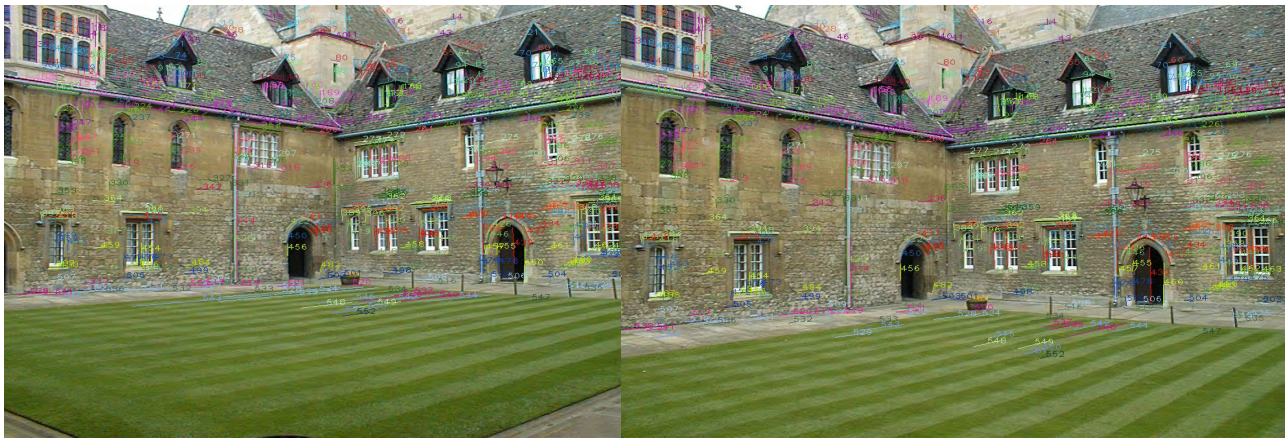

FIGURE 9. Line matches of Merton-a: Extracted line segments: 1095, 1012. Total matches: 553. Correct matches: 552.

Note that the underline denotes the second-best performance. For all scenes, this proposed method achieves robust performance on the correct matching rate (1%-2%) for the different scenes with different types of transformations because of higher completeness, especially for the accuracy, as scenes b-e are better than others and scene a is the second-best one. The MSLD algorithm performs better regarding line matching efficiency, but it usually fails in the rotation transformation (e.g., (b) in Table 3) and has a lower line matching completeness than the other methods. For the LPI method, the total matches and correct matches of (d) are slightly different from those reported by [31] because the line segments are generated by a different method in this experiment. However, the results are still convincing and can be compared with our method.

Notably, this proposed method could produce more line matches than PHLM with higher accuracy and lower time consumption per matched line. By contrast, compared with the proposed method, the LJL and LPCN methods can produce more line matches, but the accuracy is lower, and the time consumption is greater per matched line, especially for the scene with rich lines (scene b) or keypoints, as stated by [35] and [37]. Additionally, the proposed method also performs more robustly than the others in terms of accuracy, which varies in the range of 98.1 to 99.3 for all scenes. Note

TABLE 3. Comparison of the proposed method with MSLD, LPI, PHLM, LJL and LPCN.

Scene/method	Item	a	b	c	d	e
MSLD	TM	11	-	10	54	59
	CMR(%)	36.4	-	90.0	96.3	89.8
	T(s)	16	-	2	**2**	3
LPI	TM	67	259	50	84	80
	CMR(%)	74.6	93.8	80.0	90.5	80
	T(s)	128	781	44	149	35
PHLM	TM	32	235	71	151	97
	CMR(%)	**100**	95.8	94.4	96.0	94.8
	T(s)	**3**	**42**	**1**	**2**	**2**
LJL	TM	248	802	168	360	174
	CMR(%)	65.7	97.2	91.6	96.9	93.7
	T(s)	45	707	9	128	34
LPCN	TM	**386**	**898**	**389**	**398**	**197**
	CMR(%)	89.6	97.1	77.6	97.5	94.9
	T(s)	389	8744	229	279	63
Our method	TM	213	680	136	264	184
	CMR(%)	98.1	**98.8**	**99.3**	**98.5**	**98.9**
	T(s)	5	60	5	11	16

[1] "-"denotes that the method failed.
[2] "_"denotes the second-best performance.

FIGURE 10. Line matches of Valbonne: Extracted line segments: 575, 556. Total matches: 107. Correct matches: 105.

FIGURE 11. Line matches of (a) with scale: Extracted line segments: 1231, 1080. Total matches: 213. Correct matches: 209.

that LPCN exploits the matched line neighbors to find more potential matching lines (e.g., short lines) for two views, whereas the proposed method does not because the shorter fragments might be noise that might produce mismatches. However, in this paper, an iterative method of line mapping and unmapping is introduced for image sequences (more than two views) to collect and merge fragments into the same perceived line to generate a full line structure resorting to topological adjacency and thereby avoid missing short line matches or mismatches. The details are shown in Part D of Section IV.

Fig.11-Fig.15 show the results of line segment matching by the proposed method. Likewise, a different color is also used to distinguish the different line segments to verify the validation of the proposed method. However, due to occlusion and viewpoint changes, there are still fractions of the reference line (e.g., line segment 97(case(c)) in Fig.13) for the two views that produce missed matches. Line mapping for image sequences paved the way to solve the above problem, and its performance will be verified in Part D of Section V.

In order to validate the reliability of the line matching methods, five groups of experiments, each consisting of two unrelated views (e.g., a-b, b-c, c-d, d-e, e-a) in the top row of Fig. 6, are selected to match line segments inspired by [32]. There are no line matches, which is the desired line matching result since the two views are actually unrelated. As shown in Table 4, PHLM and the proposed method do not find

FIGURE 12. Line matches of (b) with rotation: Extracted line segments: 1550, 1512. Total matches: 680. Correct matches: 672.

FIGURE 13. Line matches of (c) from the UAV image: Extracted line segments: 855, 790. Total matches: 136. Correct matches: 135.

TABLE 4. Comparison of line matching on two unrelated views among the proposed method, MSLD, LPI, PHLM, LJL and LPCN.

Scene/method	e-a	a-b	b-c	c-d	d-e
MSLD	9	4	1	3	2
LPI	1	1	3	**0**	3
PHLM	**0**	**0**	**0**	**0**	**0**
LJL	4	2	2	3	**0**
LPCN	5	5	1	3	**0**
Our method	**0**	**0**	**0**	**0**	**0**

line matches for all scenes, and MSLD, with a maximum of 9 matches, performs behind LPI(3), LJL(4) and LPCN(5).

D. LINE MATCHING BASED LINE MAPPING FOR IMAGE SEQUENCES

For image sequences, the line-matching rate cannot be counted and evaluated easily using the above method in Part C of Section V. Therefore, one could adopt the analysis method following Li *et al.* [33] and Elaksher [6] to evaluate the total matches, mismatching rate, match completeness and line segments. Then, one reconstructs a 3D line-based model using matched line segments with at least three support lines each to count the total matches, and all matched 2D line segments whose reprojection error is greater than the shift threshold value are counted as mismatches. Although using the reprojection error may be inaccurate for small baselines, it does not change the score substantially when evaluating the line matching results.

One defines the ratio of the mismatches and the total sum of matched lines from image sequences as *RMM* and the ratio of the total matches and all the extracted line segments as *ROM* to evaluate the completeness of matches. The relevant rules are shown in Equation 5. In this section, five scenes, including synthetic, street and UAV image sequences (three scenes), are designed to analyze and verify the robustness and advantage of the proposed method. The synthetic scene consists of 21 views (the details are in Table 1(e)), and both the street (3072×2048) and UAV scenes (4000×3000) include

FIGURE 14. Line matches of (d) with illumination: Extracted line segments: 826, 486. Total matches: 264. Correct matches: 260.

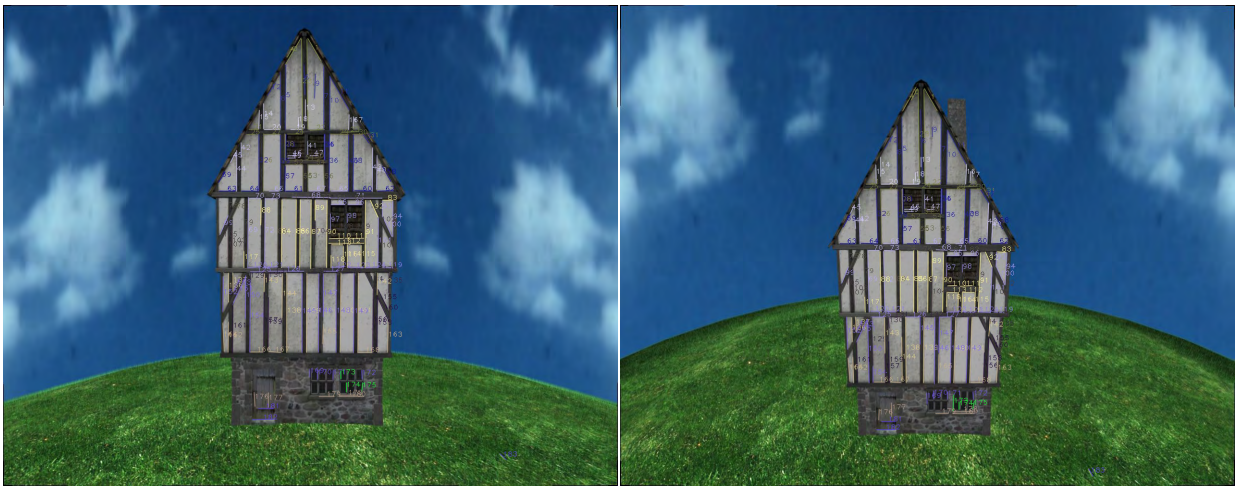

FIGURE 15. Line matches of (e) with viewpoint: Extracted line segments: 370, 311. Total matches: 184. Correct matches: 182.

FIGURE 16. 3D scene line-based model of the synthetic image: 3D lines: 321, 440. Time: 1 min, 10 mins.

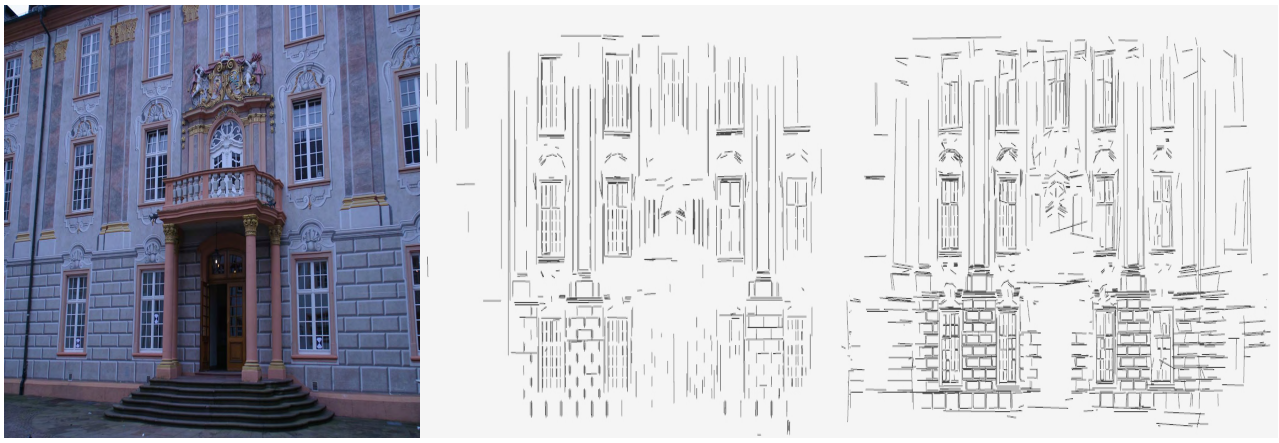

FIGURE 17. 3D scene line-based model of the synthetic image: 3D lines: 1267, 1645. Time: 6 mins, 58 mins.

FIGURE 18. 3D scene line-based model of UAV-1: 3D lines: 504, 1245. Time: 2 mins, 22 mins.

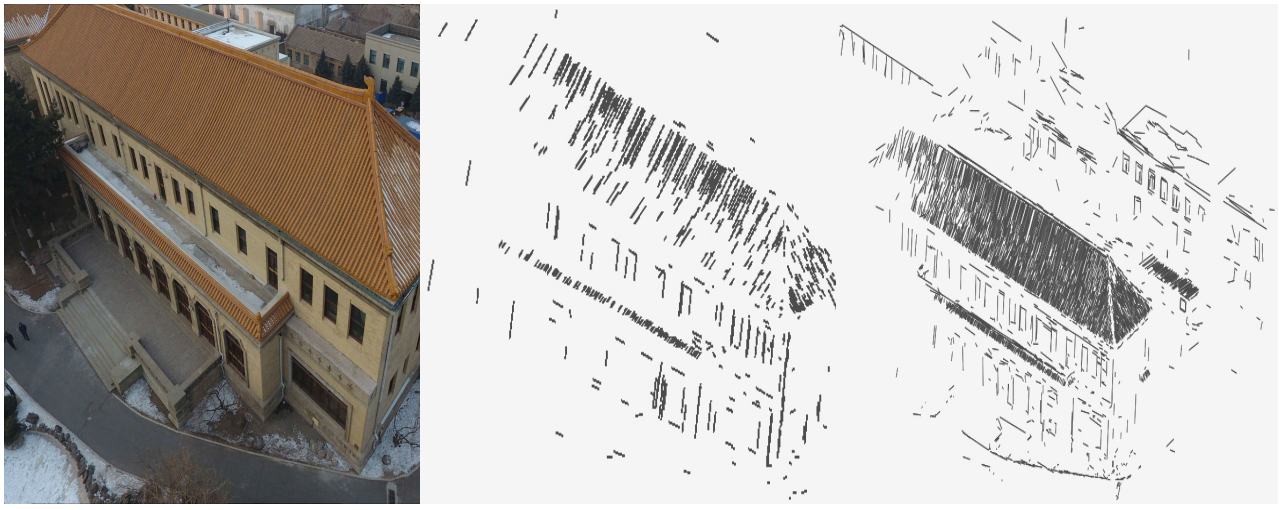

FIGURE 19. 3D scene line-based model of UAV-2: 3D lines: 598, 2865. Time: 7 mins, 64 mins.

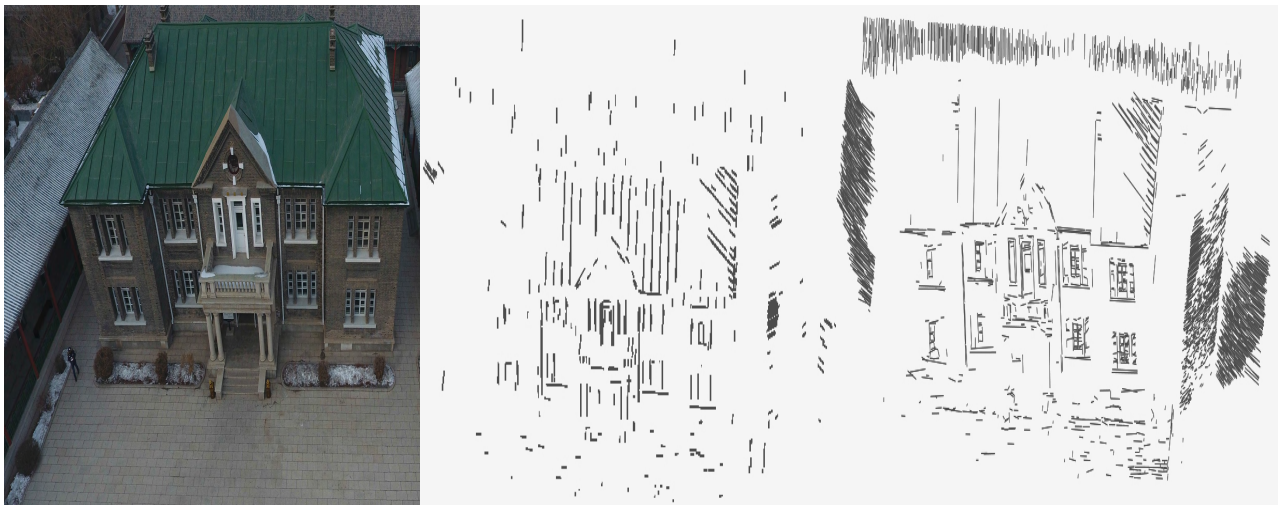

FIGURE 20. 3D scene line-based model of UAV-3: 3D lines: 463, 2537. Time: 6 mins, 60 mins.

TABLE 5. Results of line matching and 3D line reconstruction.

Scene	S2D	TM	ROM(%)	MM	RMM(%)	L3D
synthetic	9463	4847	51	226	4.7	440
Street	41780	15556	37	314	2.0	1645
UAV-1	23630	10950	46	686	6.3	1245
UAV-2	57823	27647	48	2045	7.4	2865
UAV-3	55813	25307	45	1987	7.8	2537

10 views. For the Line3D++ algorithm, *ROM* and *RMM* are not evaluated because these metrics are not mentioned in [5] and [38]. Hence, the experimental details for only the proposed method are listed in Table 5. One can see that the proposed method not only produces more matches but also results in fewer mismatches. In addition, for the different scene types, the proposed method performs robustly regarding *ROM* and *RMM* (approximately $5\% \pm 2\%$).

$$RMM = \frac{the\ total\ sum\ of\ mismatches}{the\ total\ sum\ of\ matched\ lines} \times 100\%$$
$$ROM = \frac{the\ total\ sum\ of\ matched\ lines}{the\ total\ sum\ of\ extracted\ lines} \times 100\% \quad (5)$$

To intuitively analyze the robustness and completeness of the proposed method, the experiment reconstructs the matched line segments from the above five scenes to a 3D line-based model. In addition, it also compares the 3D line-based model generated by mapped line segments from the state-of-the-art 3D reconstruction method (Line3D++ software), which is based on line segment matching. Fig.16-Fig.20 show real images and two 3D models for each scene reconstructed by Line3D++ and our approach based on line mapping, ordered from left to right. This experiment uses Ceres-Solver [49] for bundle adjustment to solve the optimization problem in our method as well as Line3D++.

In Fig.16-Fig.20, one can see clearly that the proposed line segment mapping matching method not only produces many more line matches than the existing method but also completes the line segment structures to obtain a full 3D line-based model (e.g., the roof in Figure 16). Line3D++ generates more details for the synthetic and street scenes than for the UAV scenes. By contrast, our method describes more structural information using a line-based model. However, due to its dependence on the keypoint matching performance, our method will fail when there are mismatched points or when feature points are missing near the line segment (e.g., some of the windows and doors in Figure 17 and the roof in Figure 20). In addition, although one can see that the proposed method consumes slightly more time than Line3D++, which uses massive parallelism, the proposed method generates more 3D lines (up to 5 times more than the Line3D++ method). In general, the proposed algorithm is more efficient than the existing method and performs well overall regarding the robustness and completeness of line structures and line matches on most types of scenes.

VI. DISCUSSION AND CONCLUSIONS

Note that this proposed method builds on [2], [9], and [31]; hence, it is necessary to discuss the differences between our method and their methods. In Sun's method, the candidate line segments are determined by surrounding points corresponding to those of the reference line segment, which are searched in the specific range to obtain the center and length of the line segment. The LPI method [31] finds point correspondences using a similar approach. Additionally, the LPI method [9] introduces the affine-invariant and projective-invariant into different cases with similar measures based on the maximum and maximum median, respectively. Jia *et al.* [36], [37] extends the cross ratio using CN to find more potential matched lines, especially for shorter fragments.

However, none of these methods find the most ideal point-line pairs to compute the planar homography matrix. In [40],

the precision of the homography transformation will disturb the line matching directly if these points are not closest to the line segment. That is, the accuracy of planar homography will decrease with increasing distance of the point-line. The error is also influenced by the choice of the shift threshold value.

In this paper, topological adjacency is introduced to eliminate the negative factor of the maximum probability. Another advantage is the ability to find the coplanar point-line resorting to the KD-tree data-index structure more efficiently than the above methods. In addition, to determine the matched line segments accurately, the shift threshold is introduced for line matching, and its feasibility is then validated by theory and trials. In Part A of Section V, five different scenes are designed with four different transformation types, and the shift threshold settings are verified using the PR curve, which demonstrates higher precision under the stable recall rate. Subsequently, in Part B and Part C of Section V, two groups of experiments are designed: the first compares Schmid with and without epipolar ordering constraints, in which the correct matching rates of the proposed method for three scenes are much better than Schmid(+) excluding the total matches of Valbonne caused by weak texture. The next compares the second class of line segment matching methods. Again, our method also improves the performance on the correct matching rate and time cost for each matched line segment (more matched line segments per second) for the two views, whereas LJL and LPCN focus on matching more line segments. However, the proposed method resorting to line mapping and unmapping is devoted to matched lines that are supported by at least three views for image sequences, which is more robust and results in lower mismatching. This is the main reason that the total matches are less than for the LJL and LPCN methods.

Line mapping is performed not only for the sake of improving line segment matching but also to complete the 3D line structure. In Part D of Section V, synthetic, street and UAV scenes are designed to verify and compare the robustness of our method and the state-of-the-art Line3D++ method. Our method achieves better performance in terms of completeness and robustness by combining geometric and texture information compared to Line3D++, which depends only on epipolar constraints.

In conclusion, the experiments indicated that for two or more views, the proposed method performs better in terms of the quality and quantity of matched line segments than existing methods and is practical for matching line segments from different types of scenes. However, the proposed method fails for sharp or wiry edges (e.g., eaves and power towers) due mainly to changes in neighborhood keypoints caused by the noncoplanar line-point topology, which is introduced in the Valbonne scene (Fig.10) in Part B of Section V and the case in which line segments have few or incorrect nearby keypoint pairs due to weak textures. As shown in Fig.21, there are fewer feature points near line segments and keypoint pairs even with many mismatches (e.g., 27, 56, 76, 98, 99 in Fig.21), which might directly produce an incorrect TIN. As a result, the line segment matching fails completely. Future studies should strive to improve the approach to better determine candidate line segments combining pure geometric and machine learning and correct the 3D surface mesh model using a 3D line-based model.

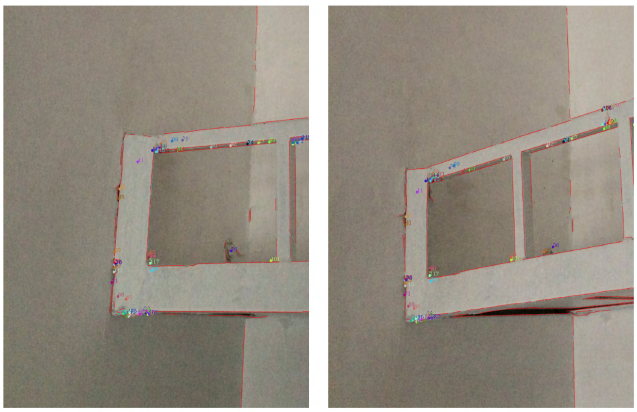

FIGURE 21. The line segment extraction and key point pairs for two views in the sparse texture: all feature points are marked using different colors and serial numbers. The red solid segment is detected by the LSD algorithm.

ACKNOWLEDGMENTS

The authors would like to thank M. Hofer, Ph.D., and Yanbiao Sun, Ph.D., for providing the experimental datasets and the implementation code. They also would like to thank Kai Li, Ph.D., and Qi Jia, Ph.D., for making the source code available.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[2] Y. Sun, L. Zhao, S. Huang, L. Yan, and G. Dissanayake, "Line matching based on planar homography for stereo aerial images," *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 1–17, Jun. 2015.
[3] C. J. Taylor and D. J. Kriegman, "Structure and motion from line segments in multiple images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 11, pp. 1021–1032, Nov. 1995.
[4] M. Rothermel, K. Wenzel, D. Fritsch, and N. Haala, "Sure: Photogrammetric surface reconstruction from imagery, " in *Proc. LC3D Workshop*, Dec. 2012, p. 12.
[5] M. Hofer, M. Maurer, and H. Bischof, "Efficient 3D scene abstraction using line segments," *Comput. Vis. Image Understand.*, vol. 157, pp. 167–178, Apr. 2017.
[6] A. F. Elaksher, "Automatic line matching across multiple views based on geometric and radiometric properties," *Appl. Geomatics*, vol. 3, no. 1, pp. 23–33, Mar. 2011.
[7] M. Hofer, M. Maurer, and H. Bischof, "Improving sparse 3D models for man-made environments using line-based 3D reconstruction," in *Proc. Int. Conf. 3d Vis.*, Dec. 2014, pp. 535–542.
[8] Z. Wang, F. Wu, and Z. Hu, "MSLD: A robust descriptor for line matching," *Pattern Recognit.*, vol. 42, no. 5, pp. 941–953, 2009.
[9] B. Fan, F. Wu, and Z. Hu, "Robust line matching through line-point invariants," *Pattern Recognit.*, vol. 45, no. 2, pp. 794–805, Feb. 2012.
[10] Q. Wang *et al.*, "Line matching based on viewpoint-invariance for stereo wide-baseline aerial images," *Appl. Sci.*, vol. 8, no. 6, p. 938, Jun. 2018.

[11] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[12] L. V. G. H. Bay and T. Tuytelaars, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.

[13] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 438–469, 2009.

[14] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

[15] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[16] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2004, pp. 506–513.

[17] F. Tang, S. H. Lim, L. Nelson Chang, and H. Tao, "A novel feature descriptor invariant to complex brightness changes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2631–2638.

[18] G. D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1159.

[19] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[20] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector," *IPOL J.*, vol. 2, pp. 35–55, Mar. 2012.

[21] C. Schmid and A. Zisserman, "Automatic line matching across views," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 666–671.

[22] R. Manuel, "Matching lines in image sequences using geometric constraits," in *Proc. 7th Portuguese Conf. Pattern Recognit.*, 1995, pp. 10–45.

[23] C. Schmid and A. Zisserman, "The geometry and matching of lines and curves over multiple views," *Int. J. Comput. Vis.*, vol. 40, no. 3, pp. 199–233, Dec. 2000.

[24] T. Werner and A. Zisserman, "New techniques for automated architectural reconstruction from photographs," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 541–555.

[25] H. Bay, V. Ferraris, and L. V. Gool, "Wide-baseline stereo matching with line segments," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 329–336.

[26] J. Witt and U. Weltin, "Robust stereo visual odometry using iterative closest multiple lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 4164–4171.

[27] A. O. Ok, J. D. Wegner, C. Heipke, F. Rottensteiner, U. Soergel, and V. Toprak, "Matching of straight line segments from aerial stereo images of urban areas," *ISPRS J. Photogram. Remote Sens.*, vol. 74, pp. 133–152, Nov. 2012.

[28] Y. Gao, S. Liu, Y. Sun, S. Fan, and X. Tan, "Line matching using a disparity map in rectified image space for stereo aerial images," *Remote Sens. Lett.*, vol. 7, no. 8, pp. 751–760, Aug. 2016.

[29] L. Wang, U. Neumann, and S. You, "Wide-baseline image matching using line signatures," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2009, pp. 1311–1318.

[30] M. Al-Shahri and A. Yilmaz, "Line matching in wide-baseline stereo: A top-down approach," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4199–4210, Sep. 2014.

[31] B. Fan, F. Wu, and Z. Hu, "Line matching leveraged by point correspondences," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 15–25.

[32] J. López, R. Santos, R. X. Fdez-Vidal, and X. M. Pardo, "Two-view line matching algorithm based on context and appearance in low-textured images," *Pattern Recognit.*, vol. 48, no. 7, pp. 2164–2184, Jul. 2015.

[33] K. Li, J. Yao, M. Xia, and L. Li, "Joint point and line segment matching on wide-baseline stereo images," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[34] K. Li and J. Yao, "Line segment matching and reconstruction via exploiting coplanar cues," *ISPRS J. Photogram. Remote Sens.*, vol. 125, pp. 33–49, Mar. 2017.

[35] K. Li, J. Yao, X. Lu, L. Li, and Z. Zhang, "Hierarchical line matching based on Line-Junction-Line structure descriptor and local homography estimation," *Neurocomputing*, vol. 184, pp. 207–220, Apr. 2016.

[36] Q. Jia, X. Gao, X. Fan, Z. Luo, H. Li, and Z. Chen, "Novel coplanar line-points invariants for robust line matching across views," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 599–611.

[37] Q. Jia, X. Fan, X. Gao, M. Yu, H. Li, and Z. Luo, "Line matching based on line-points invariant and local homography," *Pattern Recognit.*, vol. 81, pp. 471–483, Sep. 2018.

[38] M. Hofer, "Building with lines efficient 3D scene abstraction for built environment," Tech. Rep., 2016.

[39] C. T. Silva, J. S. B. Mitchell, and A. E. Kaufman, "Automatic generation of triangular irregular networks using greedy cuts," in *Proc. Visualizat.*, Nov. 1995, pp. 201–208.

[40] R. I. Hartley and A. Zisserman, *Multiple view geometry Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[41] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[42] A. Desolneux, "From gestalt theory to image analysis: A probabilistic approach," *Interdiscipl. Appl. Math.*, vol. 34, no. 3, pp. 1255–1259, 2008.

[43] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.

[44] J. B. Burns, A. R. Hanson, and E. M. Riseman, "Extracting straight lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 4, pp. 425–455, 1986.

[45] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, no. 3, pp. 183–193, 1954.

[46] A. G. Burr, "Limiting error bounds for the continuous channel," *IEE Proc. F Commun., Radar Signal Process.*, vol. 134, no. 6, pp. 571–582, Oct. 1987.

[47] R. O. Duda and R. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[48] T. Werner, *Matching of Line Segments Across Multiple Views: Implementation Description*. London, U.K.: Vis. Geometry Group, 2002.

[49] K. Mierle and S. Agarwal. (2010). *Ceres Solver*. [Online]. Available: http://ceres-solver.org

XIANGYANG JIA received the B.E. and M.E. degrees from the School of Civil and Architectural Engineering, Shandong University of Technology, in 2013 and 2016, respectively. He is currently pursuing the D.E. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University. His research interests include computer vision, image processing, and 3D laser scanning.

XIANFENG HUANG received the B.E. degree from the School of Power and Mechanical Engineering, Wuhan University, in 2000, the M.E. degree from the School of Computer Science, Wuhan University, in 2003, and the D.E. degree from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, in 2006, where he is currently a Professor. From 2011 to 2013, he was with Microsoft Research Asia and the Singapore ETH Center, as a Senior Researcher. His work has been applied in world heritage protection and public security applications. His research interests include laser scanning, 3D point cloud processing, and high-quality 3D modeling.

FAN ZHANG received the D.E. degree from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, in 2009, where he is currently an Associate Professor. His research interests include laser scanning, high-quality 3D modeling and texture reconstruction, and artificial intelligence.

YUNLONG GAO received the D.E. degree from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, in 2017, where he is currently a Postdoctoral Researcher. His research interests include terrestrial laser scanning, high-quality 3D model reconstruction, and image processing and analysis.

CHONG YANG received the B.E. degree from the School of Remote Sensing and Information Engineering, Wuhan University, in 2013, and the D.E. degree from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, in 2018, followed by postgraduate and doctoral programs of study. His research interests include computer vision, image processing and analysis, and 3D modelling.