IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 24, 2019, accepted March 5, 2019, date of publication March 15, 2019, date of current version April 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2905273

Optimization of Line Extraction Based on an Affine Camera Model

QIANG WANG[1,2], YONGSHENG ZHANG[3], YANYAN LI[4], ZHENXIN ZHANG[5], TIEJUN CUI[1], AND LEI YAN[2], (Senior Member, IEEE)

[1]School of Geographic and Environment Science, Tianjin Normal University, Tianjin 300387, China
[2]Beijing Key Laboratory of Spatial Information Integration and Its Application, Peking University, Beijing 100871, China
[3]China Nuclear Industry 23 Construction Co., Ltd., Shenzhen 518120, China
[4]Department of Informatics, Technical University of Munich, 85748 Munich, Germany
[5]Beijing Advanced Innovation Center for Imaging Theory and Technology, Capital Normal University, Beijing 100048, China

Corresponding authors: Zhenxin Zhang (zhenxin066@163.com) and Lei Yan (lyan@pku.edu.cn)

This work was supported in part by funds by the National Natural Science Foundation of China under Grant 41701533 and Grant 41471314, in part by the Open Fund of the State Key Laboratory of Remote Sensing Science under Grant OFSLRSS201818, and in part by the Open Fund Key Laboratory for National Geography State Monitoring (National Administration of Surveying, Mapping and Geoinformation) under Grant 2017NGCM06.

ABSTRACT Line feature constitutes important geometric structure information in image processing and is significant in visual navigation and 3-D structure contour extraction of 3-D objects. The current state-of-the-art line extraction methods exhibit fast extracting speed and good extracting results. However, most line extraction methods face a problem, wherein a long line is easily fragmented. Thus, a long line is broken into several short segments due to local small changes or disturbances. To obtain longer and more useful line segments without the aid of external information, this paper proposes an improved line extraction method called affine-lines, which is based on an affine camera model. First, the affine camera model is used to simulate affine projections from different viewing angles, and a sequence of affine simulated images is obtained via affine transformation matrices. Subsequently, line segments are extracted from the original image and each simulated image. Each set of line segments on the simulated image is back-projected on the original image based on its corresponding inversely affine transformation matrix. Finally, the line segments on the original image are used as references to sequentially purify and optimize subsequent lines. Several defined geometric constraints are used to eliminate pseudo lines and combine short lines. Thus, fragmented lines are connected, and short lines are converted into long lines. Given several sets of close-range and aerial images, experiments are performed and compared via a state-of-the-art line extraction algorithm, called the line segment detector (LSD). The results indicate that the proposed method significantly increases the lengths of the obtained line segments, significantly reduces the fragmentation effect, and obtains more useful line segments. The extracted improved line segments are applied to line matching for building structures, and promising results are obtained.

INDEX TERMS Affine camera model, line extraction, fragmentation effect, LSD.

I. INTRODUCTION

Line and point features constitute important geometric information in computer vision, robotics and photogrammetry communities. Lines exhibit certain special advantages when compared with points. For example, a line does not require two specific end-points in the case of occlusions. Additionally, it is difficult to extract feature points in low textured scenes although it is easier to obtain the lines. Thus, the lines are considered as good alternatives in low textured scenes with few feature points. The geometric relationships between lines are robust, and this is helpful in improving accuracy. The structural information of line features is more abundant and more visualized to express the edge contours of three-dimensional (3D) objects. Extant studies used the advantage of lines in practical engineering projects and achieved good results. Partovi *et al.* [1] used line features to perform contour extraction and regularization of building outlines. Trinh and Jo [2] analyzed the shape of buildings via extracting line segments from building surfaces.

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

Wang *et al.* [3] used line matches to generate better orthophotos. Zhang and Ghosh [4] and Crowley [5] utilized line segments for mapping and navigation. Elqursh and Elgammal [6] added line features into bundle adjustment model to perform the pose estimation of captured images. Gerke [7] and Von Schmude *et al.* [8] used horizontal or vertical line segments as constraints to achieve indirect orientation for cameras. Yu *et al.* [9] detected cracks in pipes via determining whether lines are continuous. Santos *et al.* [10] conducted powerline inspection of unmanned aerial vehicles (UAVs) via the line extraction method.

Several methods based on points are significantly mature, and several extant studies explored the same. The earliest example of a famous line extraction algorithm corresponds to the Hough Transformation [11]. The method is suitable for scenes with a single background texture. Several wrong segments appear when the background texture is complex. The edge extraction and fitting method based on Canny operator involves several parameters [12], and thus it is necessary to set fixed thresholds and results in more false positives or false negatives in the final extracted line segments. Additionally, progressive probabilistic Hough transform (PPHT) [13] is an improved version of the Hough method. This method increases the processing speed via randomly selecting a few edge points. However, the error detection mechanism of this method can only aid in extracting evident long lines, and this is useless for short line segments. Von Gioi *et al.* [14] proposed the multi-segments method that introduced the Helmholtz principle, which calculates and compares all possible segment combinations for acquired initial segments and screens final segments that work well. However, the method exhibits an especially high computational complexity and is difficult to accept in conventional applications. Specifically, the line segment detector (LSD) is an extremely popular line extraction method that was proposed by Von Gioi *et al.* [15] and is available on the Image Processing On Line (IPOL) website. It is designed to work on any digital image without necessitating manual adjustment of parameters and can extract lines in sub-pixel precision. However, the disadvantage of the method is that the extracted long lines are significantly fragmented into short line segments due to the discrepancy in one or several pixels. Akinlar and Topal [16] provided an edge drawing method termed as EDLines that is faster than LSD algorithm. The method is especially suitable for real-time and fast processing scenarios although it exhibits a disadvantage wherein the extracted segments are divided into several lines, thereby resulting in more short lines. To a certain extent, the CannyLines [17], [18] method alleviates the problem of LSD and EDLines by merging and extending line segments and obtains longer and more meaningful line segments. However, the effect of its promotion is not extremely evident, and it is unable to solve the fragmentation and short-line effects. FIGURE 1 shows the results of an aerial image using LSD, EDLines, and CannyLines methods. As shown by the yellow rectangles, each method exhibits the phenomenon of fragmentation.

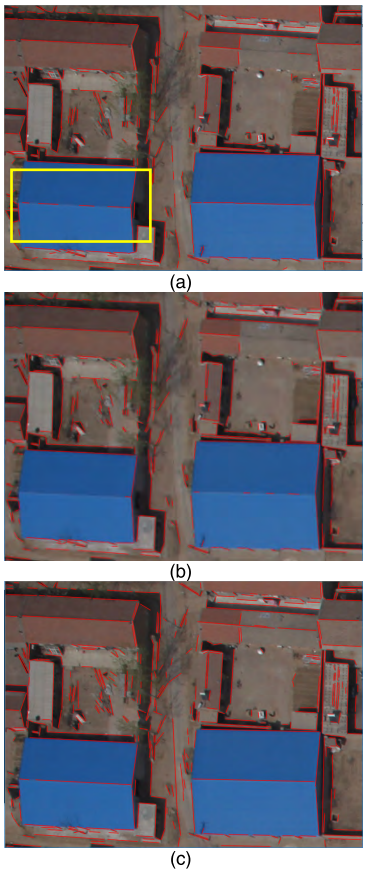


FIGURE 1. Fragmentation effect of three state-of-the-art line extraction methods. (a) LSD. (b) EDLines. (c) CannyLines.

Specifically, several short lines are not required. First, the geometric features of the short segments are not as evident as those of the long segments. In several cases, it is easy to screen the long lines. Second, the positioning accuracy of the short segments is typically worse than that of the long segments. Inaccurate endpoints of the short segment cause more serious errors than those of the long segment. Additionally, several broken segments lead to excessive calculations in the later procedure and exacerbate the ambiguity and misidentification of data such as in line matching and 3D reconstruction. However, the appearance of short lines is related to information on the original image. The photosensitivity difference (for e.g., partial blurring or sharpening) of the camera's charge coupled device (CCD) and appearance of interferences on the object surface (occlusion) results in a non-uniform transition or anomaly in several certain pixels, thereby resulting in the interruption of a long line segment. The original image corresponds to a single data source, and thus it is difficult to solve the problem without requiring external information. Simultaneously, several excellent line extraction algorithms (for e.g., LSD, EDLines, and CannyLines) mainly rely on the gradient strength and gradient direction of edge pixels. Therefore, more useful line information is obtained if it is possible to create

additional observations and change the gradient distribution of the edge pixels without damaging the original image content.

With respect to the fragmentation effect of the aforementioned excellent line extraction methods, Zhang and Koch [19] employed a scale-space pyramid model that consisted of N octave images that were generated via down-sampling the original image with a set of scale factors. However, this led to limited improvement. The present study proposes an improved line extraction method based on an affine camera model that is termed as Affine-Lines. First, the affine camera model is used to simulate different viewing angles, and this is used to construct different affine transformation matrixes. Thus, we obtain additional observation data, namely a series of simulated affine images. Subsequently, line extraction is performed on the original and each simulated image to obtain a series of line segment sets. Finally, line segments on the simulated images are normalized to the original image. The line segments on the original image serve as the reference, and thus redundant line segments are eliminated and merged and useful line segments are retained. Longer and more meaningful line segments are obtained due to optimization and purification. The workflow of Affine-Lines is shown in Figure 2.

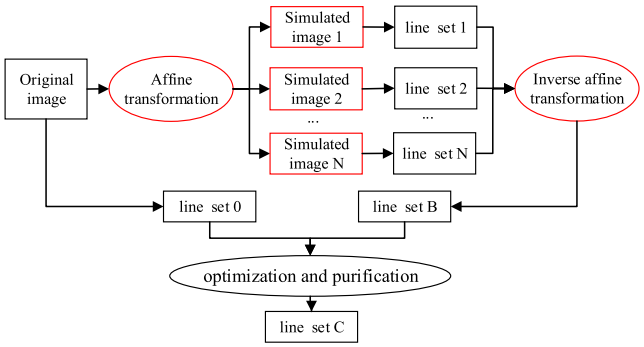


FIGURE 2. Workflow of Affine-Lines.

The study is organized as follows: The second part analyzes the principle and fragmentation problems of the LSD algorithm. The third part introduces the affine camera model and acquisition of the affine transformation matrix and the simulated images. The fourth part proposes a method to purify and optimize sets of line segments. The fifth part details the experiment and analysis. The sixth part discusses the simulation parameters, extraction accuracy, and precision of the proposed method, and application of line matching. The last part provides the conclusion.

II. ANALYSIS OF THE FRAGMENTATION EFFECT OF THE LSD ALGORITHM

Specifically, LSD is a line extraction algorithm that is widely appreciated for its rapidity and effectiveness. However, it is a self-growing method. When a long line segment encounters obstruction, blurring, or noise, it is fragmented into multiple short lines, and this is termed as the fragmentation effect. The LSD line extraction procedure is as follows:

(1) Gaussian sub-sampling is performed on the original image at a scale of 0.8 to solve the aliasing problem (Gaussian kernel is determined by $\sigma = 0.75$, which is a good balance value.

(2) We calculate the Level-Line Angle (LLA)(i.e., the level-line orientation is orthogonal to the gradient orientation) and gradient magnitude for each pixel.

(3) Based on the gradient magnitude of pixels, a pseudo-ordering algorithm is used to sort the values between the smallest and the highest values. Gradient magnitudes are classified into 1024 bins and given the initial state of 'UNUSED'.

(4) The state of pixels with a magnitude of gradient less than $\rho(\rho = 5.2)$ is termed as 'USED'. Thus, the pixels are rejected and not used in the construction of line-support regions.

(5) The highest value in the bins of unused pixels is considered as the seed P0, and a region growing algorithm is applied to form a line-support region. The initial angle of the region θ_{region} denotes the LLA of P0.

a. Thus, P0 is used as the first seed of the region. Subsequently, a search is conducted along its eight- neighborhood for the point with status corresponding to UNUSED and satisfying the condition $|LLA - \theta_{region}| < \tau(\tau = 22.5°)$. The point P1 is incorporated into the region and set as USED.

b. We update P1 as the current point and update θ_{region} based on the current θ_{region} and LLA of P1.

Steps a and b are iterative looped to obtain the final θ_{region}.

(6) Rectangle approximation. We create a rectangle R to cover the line-support region.

(7) We determine the relationship between the density of aligned points and threshold D. If the density does not exceed or equal D, then region R is cut into several smaller regions until the criterion is satisfied.

(8) Rectangle improvement.

(9) We calculate the number of lines that are in fact not lines. This corresponds to the number of false alarms (NFA), and the validity of R is determined. If the NFA is less than or equal to the threshold ε, then it is considered as a meaningful R (ε-meaningful) and R is output. If not, then we continue to improve R until it is satisfied. At this moment, all points in R are denoted as USED.

(10) We continue to process the next seed that is denoted as UNUSED and follow steps 5) to 9) to obtain a new R. The loop is completed when all UNUSED points are used up.

(11) We process all regions to obtain all line segments.

As shown in the aforementioned LSD line extraction process, a point can only belong to a certain line, and thus the extracted lines do not intersect. If two lines are about to intersect, they are broken and correspond to four lines. Additionally, with respect to a specific line segment, it is potentially broken in steps 5) and 7). This is because the region-growing is terminated when the difference between the LLA of the current point and main direction angle of the line segment neighborhood θ_{region} exceeds threshold τ.

This is also because the rectangular frame R is truncated when the density of aligned points is insufficient. The two situations are explored below.

A. LLA

The gradient of each pixel is expressed via a 2×2 template. If the gray value at the pixel (x, y) corresponds to$i(x, y)$, then its gradient calculation formula is as follows:

$$\begin{cases} g_x(x, y) \\ \quad = \dfrac{i(x+1, y) + i(x+1, y+1) - i(x, y) - i(x, y+1)}{2} \\ g_y(x, y) \\ \quad = \dfrac{i(x, y+1) + i(x+1, y+1) - i(x, y) - i(x+1, y)}{2} \end{cases} \tag{1}$$

Thus, LLA is obtained as follows:

$$LLA = \arctan(g_x(x, y)/ - g_y(x, y)) \tag{2}$$

Additionally, the region angle value is as follows:

$$\theta_{region} = \arctan\left(\frac{\sum_j \sin\left(LLA_j\right)}{\sum_j \cos\left(LLA_j\right)}\right) \tag{3}$$

where the index j runs over the pixels in the region.

The gradient magnitude is as follows::

$$G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \tag{4}$$

As shown in Figure 3, for a series of pixels that should be on a line, if one of the pixels exhibits a mutation (indicated by yellow), then its LLA inevitably produces a big difference with other pixels, and the region growing is interrupted at the yellow pixel, and thus the complete area is divided into areas 1 and 2. Hence, the line segment is fragmented and induces the fragmentation effect.

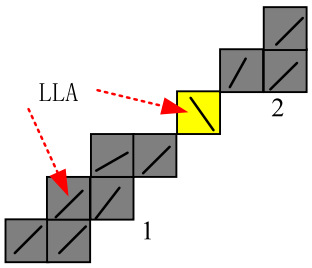


FIGURE 3. Fragmentation caused via pixel mutation.

B. DENSITY OF THE ALIGNED POINTS

After a certain region grows, it is necessary to construct a minimum envelope rectangle for the region and to determine whether the density of aligned points satisfies the threshold requirement. The density of the aligned points is defined as the ratio of the number of all the aligned points in the rectangle to the number of pixels in the area of the rectangle. This is expressed as follows:

$$density = \frac{k}{height(R) \cdot width(R)} \tag{5}$$

where k denotes the number of all the aligned points in the minimum envelope rectangle.

In several cases, two blurred pixels can appear on the edge with LLA similar to those in R2 as shown in Figure 4(a). The LLAs of three gray and three red pixels on the right part satisfy the threshold condition. However, the three gray pixels are closer to R2. Thus, they form a line support region in which the envelope corresponds to R1. However, its density is less than the threshold, and thus R1 is trimmed and adjusted. Hence, the farther pixels are rejected, and R2 that satisfies the density requirement is obtained. Evidently, the length of the line segment decreases.

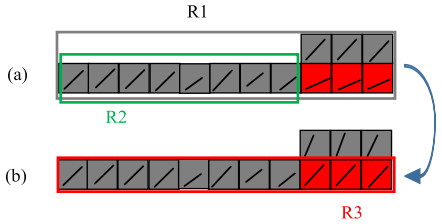


FIGURE 4. Fragmentation caused by insufficient density of the aligned points.

Another excellent line extraction algorithm corresponds to CannyLines. It uses an unconstrained Canny operator to construct an edge map. The key principle is similar to that of LSD and uses gradient information to identify line segments. However, CannyLines exclusively uses a strategy of extending and merging to solve the ''gap'' problem in LSD and obtains a longer and more complete line segment. The extending strategy is as follows: If a pixel suddenly changes, then the LLA does not satisfy the condition of region growing, and it does not directly give up. Conversely, it seeks to determine $k(1 \leq k \leq 3)$ close neighbors in the eight-neighborhoods of the pixel. The vertical distance of the k points to the current line (fitted with the existing line support region) should be within 1 pixel. Subsequently, it determines whether edge pixels exist near the k points. If they exist and if the difference between LLA and θ_{region} is lower than the threshold, then it is considered as in the same direction as the existent region and merges into the area to realize the growing region. Thus, this bridges the mutation pixel and increases the line length. However, the method exhibits a limited improvement in terms of the line length. In several cases, the improvement effect is not extremely evident.

A solution to solve the problems of aforementioned two methods involves changing the gradient difference between the yellow pixel and surrounding pixels in Figure 3. Thus, regions 1 and 2 are connected to solve the fragmentation problem. Simultaneously, with respect to the blurring problem at edges, if the LLA of the two layers of blurred pixels changes, then the LLA of the red pixels is closer to the main direction of

the region. Subsequently, the rectangular region corresponds to R3 (as shown in Figure 4(b)), and the density condition is also satisfied, thereby obtaining longer line segments. Therefore, the next attempt involves determining a transformation method that does not damage the effective content of the image or change the basic geometrical characteristics of the line and only changes the local gradient distribution of the pixels and the LLA. An affine transformation is a simple and effective solution. Simultaneously, more different affine transformations are required to cope with more scenarios. Therefore, an affine camera model that can be used to construct multiple simulations is adopted.

III. AFFINE CAMERA MODEL AND AFFINE TRANSFORMATION MATRIX

A. AFFINE CAMERA MODEL

When an object corresponds to perfect plane and lambert body, it can be observed at any tilt angle. However, several objects in the real world are non-planar and non-Lambertian with occlusion and non-uniformity. An affine camera model observes 'more information' to the maximum possible extent via establishing different viewing angles. The affine camera model was proposed by Yu and Morel [20] to solve the point feature matching problem of large viewing-angle images, and the ASIFT (Affine Scale Invariant Feature Transform) method was created. Simultaneously, the group member Yu proved that ASIFT corresponds to a complete affine invariant feature [21]. Specifically, ASIFT simulates all possible linear deformations caused via changes in the camera's viewing angle. Furthermore, ASIFT [22] adds two simulations of the affine deformation factor (i.e., longitude and latitude) based on SIFT. The SIFT algorithm is invariant to translation, rotation, and zoom. Therefore, ASIFT exhibits translation, rotation, scale, and affine invariance.

The affine camera model is shown in Figure 5. Specifically, u denotes the object's plane, C denotes the current location of camera, $C^{'}$ denotes the zenith point, and θ and ϕ are termed as latitude and longitude, respectively. The hemisphere is divided by longitude and latitude, and the camera moves over the entire hemisphere. The black dots indicate the sampling points (i.e., the intersections of the longitudes and latitudes) that correspond to the positions where the camera is located. With respect to the perspective of camera motion, ϕ denotes the rotation and θ represents the tilt angle that exhibits the relationship $t = 1/\cos\theta$. Additionally, ψ parameterizes the camera spin, and λ corresponds to the zoom based on camera's distance. The longitude ϕ leads to the rotation of camera, and latitudes θ causes the camera to sample in the longitude direction, and this causes the image distortion. The mathematical expression of affine projection is given by combining the aforementioned geometric model as follows:

$$\begin{aligned} A &= H_\lambda R_1(\Psi_1) T_t R_2(\phi) \\ &= \lambda \begin{bmatrix} \cos\Psi & -\sin\Psi \\ \sin\Psi & \cos\Psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \end{aligned} \quad (6)$$

where, $\lambda > 0$, $\theta \in [0, 90)$, $\phi \in [0, \pi)$, $\psi \in [0, 2\pi)$.

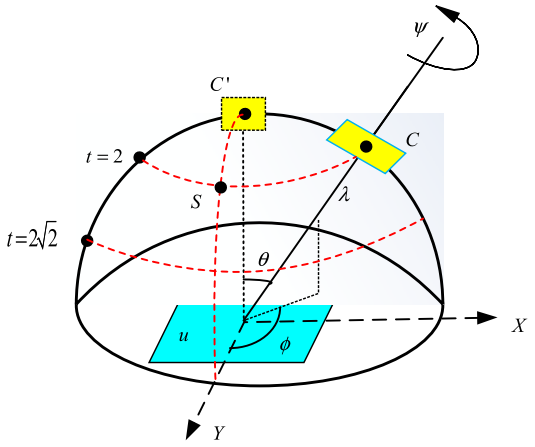


FIGURE 5. Affine camera model.

In (6), an image is first rotated using the longitude angle ϕ, and the rotated image undergoes t-subsampling in the latitudinal direction, thereby causing non-uniform scaling in the x and y directions. Thus, a deformed image is obtained. With respect to a certain line segment, it causes a change in the area of its neighborhood wherein the neighborhood pixels are resampled. Finally, ψ and $\lambda$are used to perform overall rotation and scaling of the deformed image. The final overall rotation and scaling does not affect the internal distribution of the image, and thus it can generally be omitted, i.e., $\psi = 0$, $\lambda = 1$. At this point, the affine transformation matrix under the affine model is reduced as follows:

$$\begin{aligned} A &= \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \\ &= \begin{bmatrix} 1/\cos\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} = A_1 A_2 \end{aligned} \quad (7)$$

The aforementioned affine transformation expression is simple and also does not change the shape of the line segment and relative geometric relationship between line segments. Given this type of transformation, the generation of affine simulated images and line extraction of the simulated images are performed.

B. GENERATION OF SIMULATION IMAGES UNDER THE AFFINE TRANSFORMATION

Based on the aforementioned affine camera model and affine transformation matrix, a series of simulated images is generated. The production process is as follows:

(1) With respect to latitude θ, perform sampling based on geometric sequences $t = 1, a, a^2, \ldots, a^n (a > 1)$ and subsequently obtain different degrees of tilt. Specifically, $a = \sqrt{2}$ corresponds to a good compromise between accuracy and sparsity. The value n can increase to 5 or more.

(2) Under different values of t, longitudeϕ is sampled by following a sequences $0, b/t, \ldots, kb/t (kb/t < 180°)$. Thus, we obtain a series of affine matrices and simulated images, and $b = 72°$ constitutes a good compromise.

Therefore, the generation of simulated images is performed via rotating from the original images based on A_2 and subsequently tilting subsampling based on A_1. The rotated images are obtained via bilinear interpolation and typically accompanied by edge aliasing. In order to eliminate aliasing to obtain better line segments, Gaussian convolution is used prior to t-subsampling. One-dimensional Gaussian filtering is performed on each column of the rotated image along the X direction. The filter function is as follows:

$$G_\sigma^x(x, y) = (1/\sqrt{2\pi}c\sigma)\exp\{-\frac{x^2}{2(c\sigma)^2}\} \tag{8}$$

where, c denotes the scale of Gauss sampling, and its value in the study corresponds to 0.8. $\sigma = c\sqrt{t^2 - 1}$.

A series of simulated images is obtained after rotation, filtering and subsampling,. The conversion formula of the original image and simulated image is as follows:

$$I' = A^* I \tag{9}$$

where, I denotes the original image, and I' denotes the simulated image.

Figure 6 and Figure 7 show an original image and simulated images of UAV, respectively, where $t = \left(\sqrt{2}\right)^3$ $(n = 3)$.

As shown in Figure 7, a series of images at different viewing angles is sequentially simulated via the camera affine model. The images represent almost all observations from all directions of the scene.

FIGURE 6. Original UAV image.

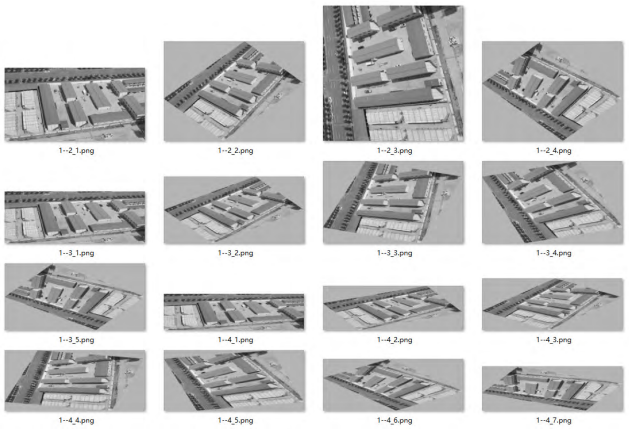


FIGURE 7. Sixteen simulated images.

As shown in the parts of the original image and first simulated image, the distribution of white dots above the house undergoes major changes via the affine transformation. Several white spots are ablated and blurred in the Gaussian resampling via a template operation among pixels, thereby resulting in changes in the LLAs of several pixels. Given the change, the breakage problem of region growth in Figure 3 and Figure 4 is resolved.

IV. PURIFICATION AND OPTIMIZATION OF LINE SETS

A. LINE EXTRACTION AND NORMALIZATION OF SIMULATED IMAGES

Line extraction is performed on the original image and simulated images, and a series of line segment sets is obtained. We assume that there are $N-1$ simulated images based on the original image, and they correspond to affine transformation matrices $\{A_1, \ldots, A_{N-1}\}$. All the lines sets obtained from the N images are expressed as $\varsigma' = \left\{L_0, L_1^{'}, \ldots, L_{N-1}^{'}\right\}$. Specifically, L_0 comes from the original image, and $L_1^{'}, \ldots, L_{N-1}^{'}$ correspond to the sets of segments extracted from each simulated image. The N line sets in ς' do not lie in the same coordinate system, and thus it is necessary to normalize $L_1^{'}, \ldots, L_{N-1}^{'}$ to the original image coordinate system in which L_0 is located. We assume that $L_i^{'} = \left\{l_i^{'1}, l_i^{'2}, \ldots, l_i^{'M_i}\right\}$ $(i = 1, 2, \ldots, N-1)$ where M_i denotes the number of line segments on each simulated image, and then each line $l_i^{'j}$ in $L_i^{'}$ is back-projected to obtain $L_i = \left\{l_i^1, l_i^2, \ldots, l_i^{M_i}\right\}$ via the affine matrix A_i. The transformation relationship is as follows:

$$l_i^j = A_i^{-1} * l_i^{'j} \tag{10}$$

where, $l_i^{'j} = \left\{\left(x1_i^{'j}, y1_i^{'j}\right), \left(x2_i^{'j}, y2_i^{'j}\right)\right\}$, $\left(x1_i^{'j}, y1_i^{'j}\right)$ and $\left(x2_i^{'j}, y2_i^{'j}\right)$ denote the coordinate values of the two ends of $l_i^{'j}$.

Thus, the normalized line segment set is expressed as $\varsigma = \{L_0, L_1, \ldots, L_{N-1}\}$.

B. PURIFICATION AND OPTIMIZATION

With respect to the aforementioned-mentioned line segment set ς, there exist several line segments that overlap. Additionally, given the inevitable point back-projection error, there exists a small deviation between two line segments, thereby resulting in redundancy and clutter. Therefore, it is necessary to merge the collinear line segments and remove the cluttered line segments. Given the possible geometric relationships between two line segments, the study uses the constraint conditions of horizontal distance, vertical distance, angle and intersection to purify and optimize the line segments. The rules are as follows: While processing of the ith line in L_k, we set the previous line segment sets $L_0L_1 \ldots L_{k-1}\left(l_k^1, l_k^2, \ldots, l_k^{i-1}\right)$ as references. This is illustrated in Figure 9: l_2 is processed on the condition that l_1 serves as the base line.

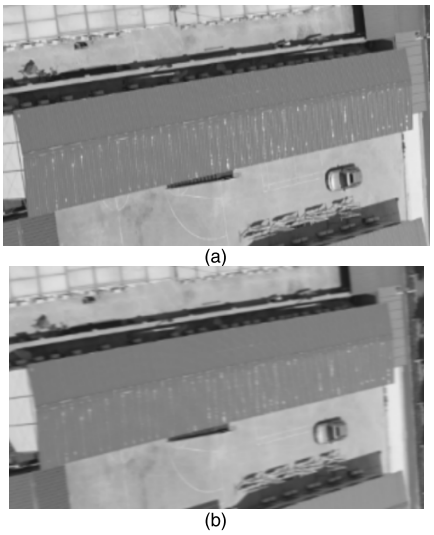


FIGURE 8. Geometric constraints between line segments. (a) Original image. (b) First simulated image.

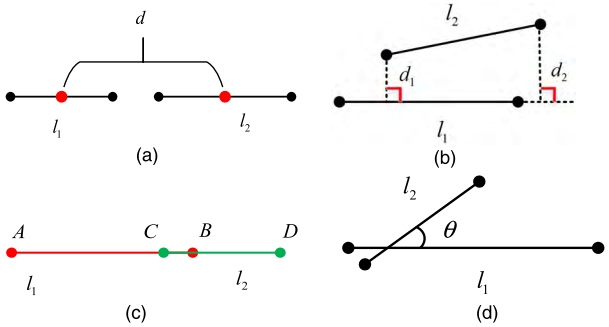


FIGURE 9. Geometric constraints between line segments. (a) Horizontal distance constraint. (b) Vertical distance constraint. (c) Collinear overlap conditions. (d) Angle and intersection constraint.

1) HORIZONTAL DISTANCE CONSTRAINT

If the distance d between the midpoints of two lines exceeds half of the total length of the two lines, i.e., $d > (length(l_1) + length(l_2))/2$, then this indicates that l_1 and l_2 are not related. Thus, l_2 should be denoted as a valid line and retained as shown in Figure 9(a). Additionally, if $d <= (length(l_1) + length(l_2))/2$, then l_2 can be parallel (including collinear) or disjoint or intersecting with l_1. Thus, we continue to determine the vertical distance.

2) VERTICAL DISTANCE CONSTRAINT

As shown in Figure 11(b), under the condition of $d <= (length(l_1) + length(l_2))/2$, if the average of the sum of the distances from the endpoints to the lines is less than the threshold ξ_d (a pixel in the study), i.e., $(d_1 + d_2)/2 < \xi_d$, then it is assumed that l_2 and l_1 are collinear. Subsequently, it is necessary to merge l_2 and l_1 to obtain a long line segment in which the endpoints correspond to A and D as shown in Figure 11(c). If $\xi_d < (d_1 + d_2)/2 < 2.5\xi_d$, then l_2 is considered as the error line and should be removed.

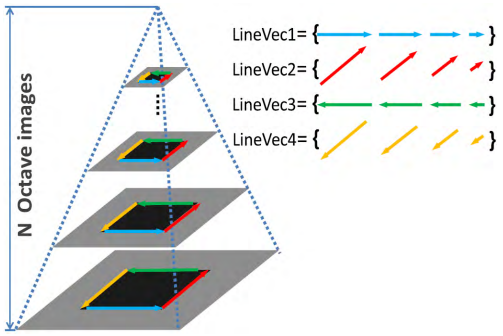


FIGURE 10. Illustration of the line extraction in the scale-space pyramid.

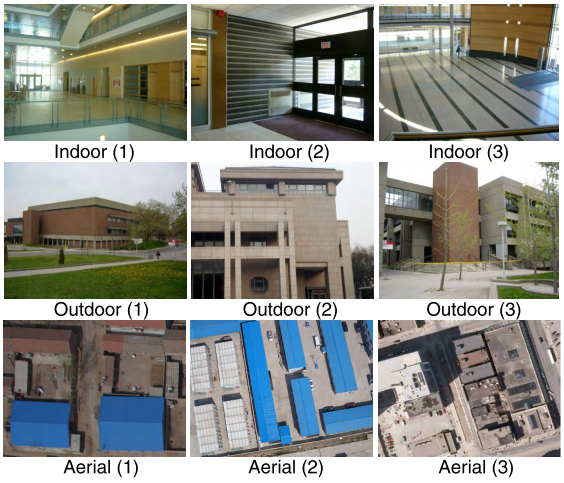


FIGURE 11. Test data.

If $(d_1 + d_2)/2 >= 2.5\xi_d$, then l_2 is considered as disjoint or intersecting l_1, and this should be denoted as a valid line and retained.

Based on the aforementioned-mentioned horizontal and vertical distance constraints, most line segments can be essentially processed. A few areas are excessively sensitive, and thus it is easy to extract a few pseudo-line segments. The line segments can intersect with the reference line segment at a certain angle. Therefore, it is necessary to reject the same via angle and intersection constraints.

3) ANGLE AND INTERSECTION CONSTRAINT

If an intersection exists between the two line segments and the intersection angle θ is between ξ_θ and $8\xi_\theta$ (i.e., ξ_θ corresponds to the angle threshold), namely $\xi_\theta < \arccos(l_1^T l_2 / |l_1^T| |l_2|) < 8\xi_\theta$, then l_2 is considered as a pseudo line and removed as shown in Figure 9(d). With respect to most regularized building structures, the intersection angle between two adjacent intersecting edge lines is not low. In our tests for several frontal close-range images and down-looking aerial images, $\xi_\theta = 5°$ constitutes an optimal choice.

We assume that the final optimized line segment set corresponds to Θ, which is initially empty. Based on the aforementioned three constraints, the processing steps are as follows:

STEP 1: The first line l_0^1 in the set of L_0 that is extracted from the original image is incorporated into Θ.

STEP 2: The second line in L_0 should be compared with l_0^1 under the constraints. If the line is denoted as valid, then it is incorporated into Θ, and Θ is updated. The line is discarded if it does not satisfy constraints.

STEP 3: The third line in L_0 is performed as STEP 2. When all the line segments in L_0 are processed in turn, we obtain the latest Θ.

STEP 4: With respect to each line segments in L_i ($i = 1, 2, 3, \ldots, N-1$), the same operation is performed to screen and optimize via a comparison with Θ until all line segments in ς are processed.

Specifically, the merging method in conjunction with the analysis of accuracy and precision of the extracted lines from Affine-lines is stated in the discussion section.

V. EXPERIMENT AND ANALYSIS

Three indoor images, three outdoor close-range images, and three aerial images are selected as research objects as shown in Figure 11. Outdoor (2) and three aerial images are from our data, and the other images are available at http://cvrs.whu.edu.cn/projects/cannyLines/. As mentioned in the Introduction section, the objective of Zhang's scale-space pyramid involves solving the fragmentation problem of line detection [19]. The method detects lines in different image scale-space and subsequently purifies all lines to obtain longer lines. A theoretical illustration is shown as in Figure 10. For comparison purposes, Zhang's method is hereafter termed as Pyr-LSD.

The proposed method in the study is compared with LSD [23] and Pyr-LSD. For comparison purposes, the proposed method that is compared with the LSD is termed Affine-LSD. In the experiment, the tilt parameter of the affine camera is $t = \left(\sqrt{2}\right)^2$, i.e., $n = 2$, and the number of simulated images corresponds to 9. With respect to Pyr-LSD, the number of layers in the pyramid corresponds to 3.

In order to visually demonstrate several important processes in the method presented in the study, the generation process of the simulated image of Aerial (1) and line segments extracted from the corresponding simulated image is illustrated. As shown in Figure 12, the left column denotes the original gray image and the nine simulated images, and the right column denotes the corresponding extracted line segments via LSD.

As shown in the area of the yellow dashed box in the original gray image, differences exist in the effect of roof extraction from different viewing angles. With respect to the same line structure, the extracted line segment is relatively complete from a certain viewing angle although the fragmentation effect is evident under a few other viewing angles. A richer line segment feature is obtained via complementing the information from different viewing angles.

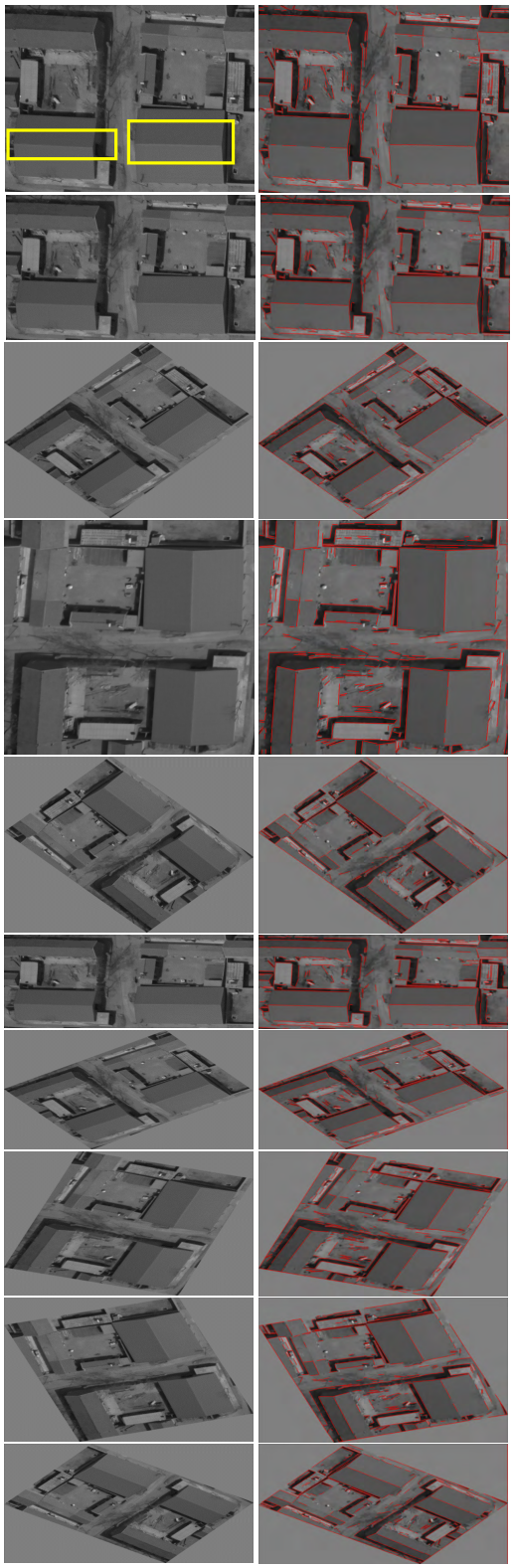
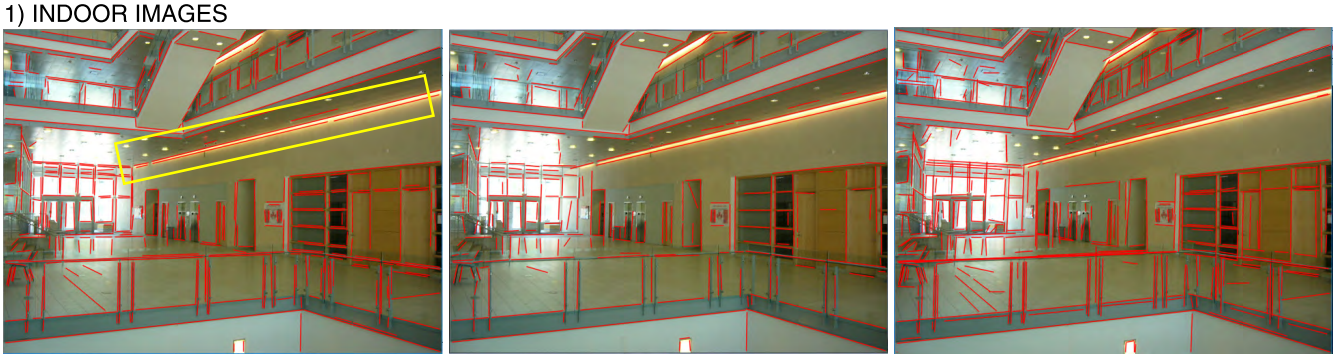


FIGURE 12. Simulated image sequence and line extraction. The left column denotes the original gray image and nine simulated images, and the right column denotes the corresponding results of the line extraction.

Subsequently, the processed results of the nine test images are shown in Figure 13-Figure 21. The number of line segments, average length of the line segments, and total

1) INDOOR IMAGES

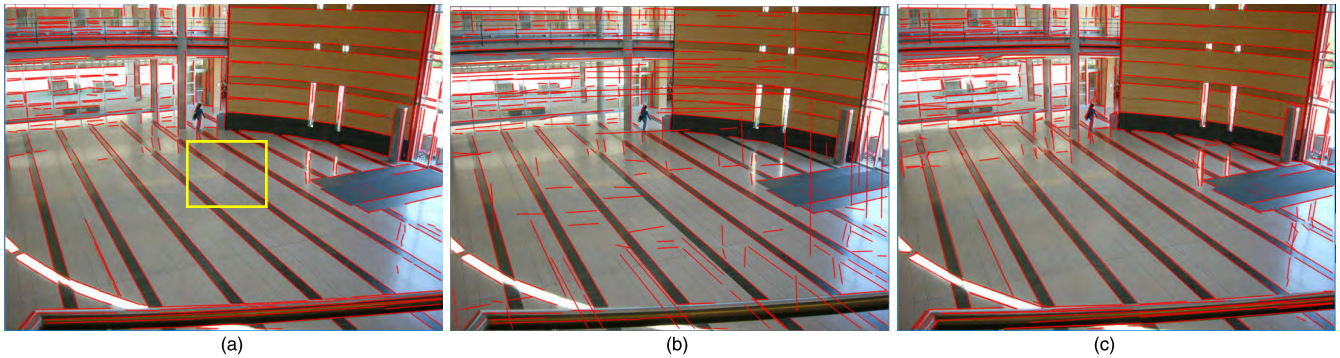


(a) (b) (c)

FIGURE 13. Comparison of the results of the three extraction methods for Indoor (1). (a) LSD (442/39.9/17635.8). (b) Pyr-LSD(303/43.5/13180.5.9). (c) Affine-LSD (504/64.4/32457.6).

(a) (b) (c)

FIGURE 14. Comparison of the results of the three extraction methods for Indoor (2). (a) LSD (487/53.4/26005.8). (b) Pyr-LSD(299/71.1/21258.9). (c) Affine-LSD (445/100.5/44722.5).



(a) (b) (c)

FIGURE 15. Comparison of the results of the three extraction methods for Indoor (3). (a) LSD (479/45.9/21986.1). (b) Pyr-LSD(522/66.5/34713.0). (c) Affine-LSD (528/77.0/40656.0).

length of the line segments are denoted below each image. The unit of length corresponds to pixels. The region of interest where the line segment changes significantly is denoted by a yellow rectangle.

1) INDOOR IMAGES
2) OUTDOOR IMAGES
3) AERIAL IMAGES

As shown in the aforementioned results, the line segments directly extracted via LSD are not complete, and fragmentation occurs in several images. Specifically, Pyr-LSD improves the extraction results to a certain extent via purifying a few redundant lines and merging a few fragmented lines. However, it is not possible to effectively solve a few fragmentation problems (and especially those on structured edges). Nevertheless, the integrity of extracted line segments is significantly improved via the Affine-LSD. The fragmentation effect is significantly attenuated as shown in the yellow rectangles in Figure 13, Figure 15, Figure 16, Figure 18, Figure 19, Figure 20, and Figure 21. Several trivial and insignificant line segments are removed and longer and

2) OUTDOOR IMAGES

(a) (b) (c)

FIGURE 16. Comparison of the results of the three extraction methods for Outdoor (1). (a) LSD (265/33.6/8904.0). (b) Pyr-LSD (162/45.6/7387.2). (c) Affine-LSD (358/82.7/29606.6).

(a) (b) (c)

FIGURE 17. Comparison of the results of the three extraction methods for Outdoor (2). (a) LSD (836/37.2/31099.2). (b) Pyr-LSD(510/48.9/24939.0). (c) Affine-LSD (465/97.4/45291.0).

(a) (b) (c)

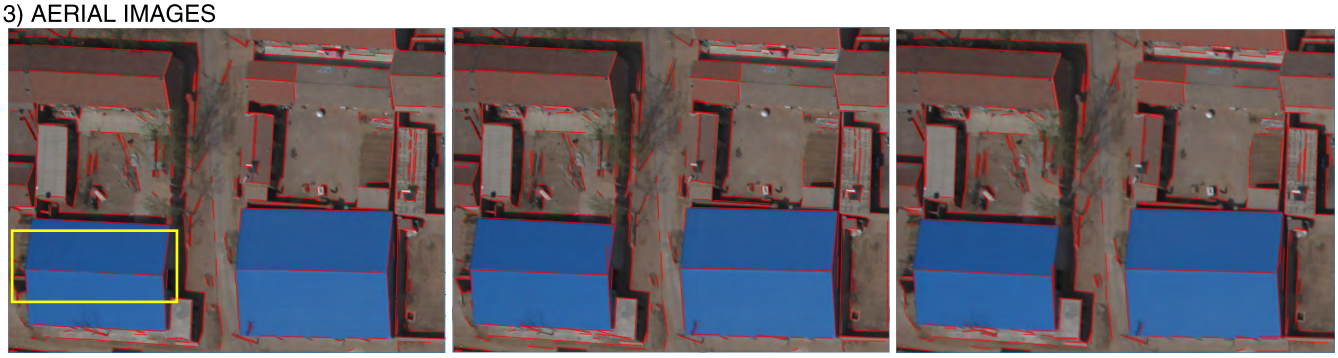
FIGURE 18. Comparison of the results of the three extraction methods for Outdoor (3). (a) LSD (552/26.7/14738.4). (b) Pyr-LSD(350/33.7/11795.0). (c) Affine-LSD (679/50.9/34561.1).

meaningful line segments are retained as shown in the yellow rectangles in Figure 14, Figure 16, and Figure 17. A few line segments that cannot be extracted via LSD are extracted via Pyr-LSD and Affine-LSD as shown by the yellow rectangle in Figure 20. As shown in the number of line segments, average length of the line segments, and total length of the line segments, for certain images, the average length of the line segments increased significantly and the total length of the line segments exceeds that in the original method despite decreases in the number of line segments obtained via the proposed method.

From the aforementioned findings, we conclude that Pyr-LSD and Affine-LSD improve the extraction effect via adding observation information and changing the pixels' gradient distribution. However, Affine-LSD evidently exceeds Pyr-LSD in terms of the average length of lines and visual effects. This is potentially because Pyr-LSD only obtains newly resampled simulated images in the same view direction and because the number of pyramid layers is limited. Thus, the Affine-LSD uniformly resamples an image from each view-angle.

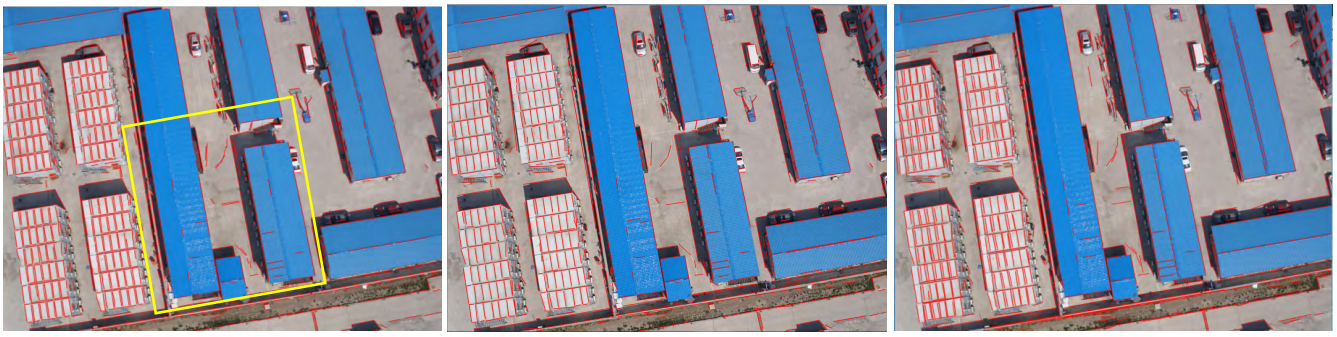
Additionally, Table 1 summarizes the number of initial line segments of each test image (i.e., the total number of line segments extracted from the original image and all simulated images), number of line segments after the optimization, average length of the line segments, total length of the line

3) AERIAL IMAGES



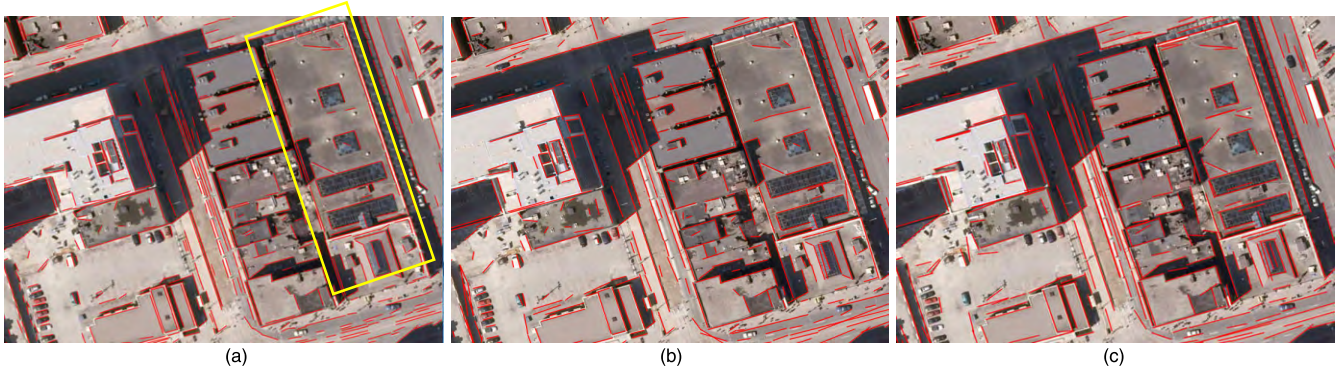
(a) (b) (c)

FIGURE 19. Comparison of the results of the three extraction methods for Aerial (1). (a) LSD (456/30.8/14044.8). (b) Pyr-LSD(330/43.7/14421.0). (c) Affine-LSD (318/101.7/32340.6).



(a) (b) (c)

FIGURE 20. Comparison of the results of the three extraction methods for Aerial (2). (a) LSD (1064/30.0/31920.0). (b) Pyr-LSD(642/40.8/26193.6). (c) Affine-LSD (701/78.5/55028.5).



(a) (b) (c)

FIGURE 21. Comparison of the results of the three extraction methods for Aerial (3). (a) LSD (513/25.5/13081.5). (b) Pyr-LSD(345/32.1/11074.5). (c) Affine-LSD (285/64.9/18496.5).

segments, and time consumption. The average length of line segments and total length of line segments are plotted in Figure 22-Figure 23.

As shown in Figure 22 and Figure 23, following the use of the Affine-Lines method proposed in the study, the length of the line segments is significantly improved, and the average length and total length of the line segments of several images reaches more than twice that of the original method. However, the Affine-Lines method requires a continuous resampling of images, line extraction, back-projection and subsequent purification and optimization, and thus its time cost significantly exceeds that of the original method. The average time consumption is approximately 10 times that of the original method as shown in Table 1. Additionally, a slight error exists due to resampling and back-projection, thereby resulting in slight deviations in the final line segments.

VI. DISCUSSION

The tilt parameter selected in the experimental corresponds to $t = \left(\sqrt{2}\right)^3$, and nine simulations are included. Theoretically, the affine camera model simulates several viewing angles

TABLE 1. Experimental results of different line extraction methods.

Test data	Method	Number of initial line segments	Number of final line segments	Average length of line segments (Pixel)	Total length of line segments (Pixel)	Elapsed Time (s)
	LSD	\	463	35.5	16436.5	0.18
Indoor(1)	Pyr-LSD	\	303	43.5	13180.5	0.73
	Affine- LSD	**3478**	**552**	**62.7**	**34610.4**	**0.96**
Indoor(2)	LSD	\	487	53.4	26005.8	0.11
	Pyr-LSD	\	299	71.1	21258.9	0.83
	Affine- LSD	**3244**	**445**	**100.5**	**44722.5**	**1.10**
	LSD	\	479	45.9	21986.1	0.10
Indoor(3)	Pyr-LSD	\	522	66.5	34713.0	0.85
	Affine- LSD	**3680**	**528**	**77.0**	**40656.0**	**1.05**
	LSD	\	265	33.6	8904.0	0.08
Outdoor (1)	Pyr-LSD	\	162	45.6	7387.2	0.36
	Affine- LSD	**2070**	**358**	**82.7**	**29606.6**	**0.81**
	LSD	\	836	37.2	31099.2	0.09
Outdoor (2)	Pyr-LSD	\	510	48.9	24939.0	3.60
	Affine- LSD	**4829**	**465**	**97.4**	**45291.0**	**3.70**
	LSD	\	552	26.7	14738.4	0.13
Outdoor (3)	Pyr-LSD	\	350	33.7	11795.0	0.88
	Affine- LSD	**3680**	**679**	**50.9**	**34561.1**	**1.16**
	LSD	\	456	30.8	14044.8	0.09
Aerial(1)	Pyr-LSD	\	330	43.7	14421.0	0.92
	Affine- LSD	**3651**	**318**	**101.7**	**32340.6**	**1.15**
	LSD	\	1064	30.0	31920.0	0.16
Aerial(2)	Pyr-LSD	\	642	40.8	26193.6	1.30
	Affine- LSD	**7939**	**701**	**78.5**	**55028.5**	**1.82**
	LSD	\	513	25.5	13081.5	0.10
Aerial(3)	Pyr-LSD	\	345	32.1	11074.5	0.92
	Affine- LSD	**4178**	**285**	**64.9**	**18496.5**	**1.26**

if the sampling interval is set as sufficiently low. However, increases in the number of simulations improves the effect. Thus, the tilt parameter value is discussed. Subsequently, the accuracy and precision of the extracted lines from the proposed method are evaluated. Finally, the Affine-Lines method is applied to line matching.

A. DISCUSSION OF THE NUMBER OF AFFINE SIMULATION

We select a new aerial image and extract line segments from the original image directly (no simulation) and via 9 simulations, 16 simulations, and 26 simulations. Table 2 lists the number of extracted line segments, average length of the line segments, and elapsed times.

As shown in Table 2 and Figure 24, the difference in line extraction between 16 and 26 simulations is not extremely high (special attention focuses on the yellow arrow as indicated to the edge of the building) and especially on the structure extraction of the house. The average length of the

TABLE 2. Comparison of line extraction results under different simulation times.

Affine simulation times	Number of extracted line segments	Average length of line segments (Pixel)	Elapsed time (s)
0	465	43.9	0.14
9	394	99.7	1.70
16	462	113.4	2.32
26	572	119.66	3.03

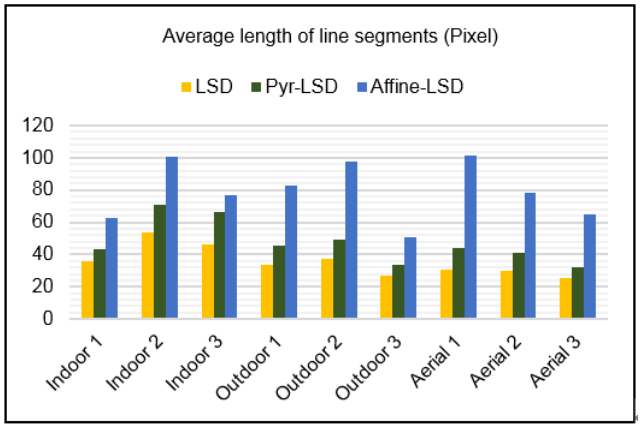


FIGURE 22. Comparison of the average length of line segments obtained under different methods.

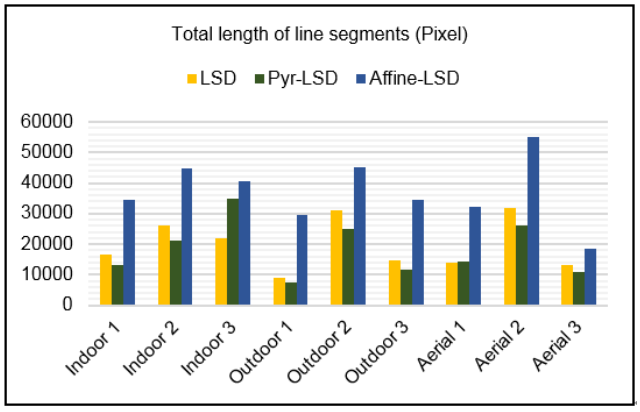


FIGURE 23. Comparison of the total length of line segments obtained under different methods.

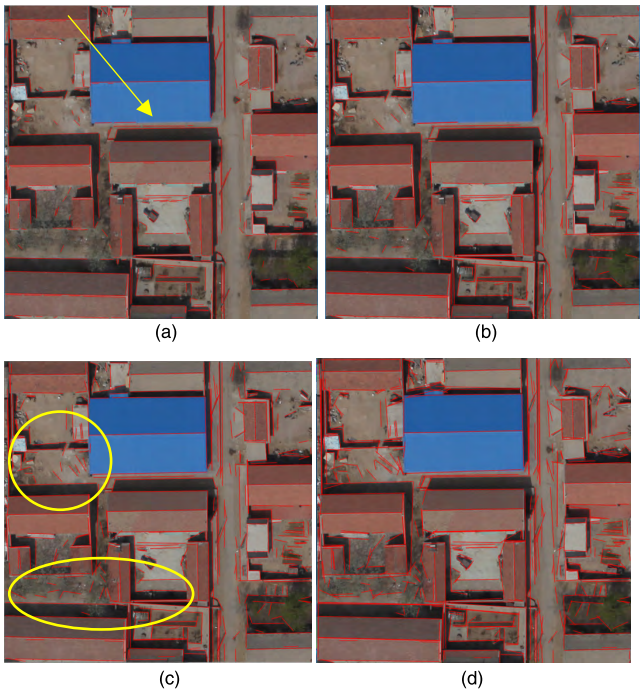


FIGURE 24. Comparison of extraction results under different affine simulation times. (a) $t = \left(\sqrt{2}\right)^0$, no simulation. (b) $t = \left(\sqrt{2}\right)^2$, 9 simulations. (c) $t = \left(\sqrt{2}\right)^3$, 16 simulations. (d) $t = \left(\sqrt{2}\right)^4$, 26 simulations.

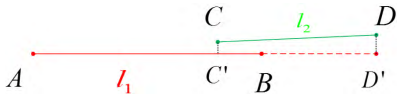


FIGURE 25. Merging of two fragmented lines.

line segments changes from 113.4 to 119.66, and the increase is not significant. Several meaningless segments are added (see the yellow oval box) although the number of line segments increases (462 to 572). Simultaneously, it is more time consuming in terms of time costs. Therefore, in general applications, the number of simulations is not as high as possible and generally assumes a moderate value, i.e., $t = \left(\sqrt{2}\right)^2$ or $t = \left(\sqrt{2}\right)^3$.

B. ACCURACY AND PRECISION OF THE EXTRACTED LINES

In this section, we discuss the accuracy and precision of the extracted lines.

1) PRECISION EVALUATION

First, it is necessary to describe the merging strategy in detail. In the purification and optimization procedure, the short line segments are merged to define a longer line. Figure 25 corresponds to the exaggerated show of Figure 9(c). The merging method of the fragmented lines is as follows: We assume that l_1 (its endpoints are A and B) corresponds to a line segment that is extracted from the original image and that l_2 (i.e., its endpoints are C and D) corresponds to a back-projected line segment from the simulated image. We consider l_1 as the benchmark, and C' and D' correspond to the two projected endpoints of l_2. When an overlapping area exists between l_1 and $\overline{C'D'}$ (this is referred to as the horizontal distance constraint) and the average projection distance from l_2 to l_1 is less than 1 pixel (this is referred to as the vertical distance constraint), then they are merged into a new longer line $\overline{AD'}$ that replaces l_1.

As noted by Von Gioi *et al.* [23], the LSD yields sub-pixel precious results. Thus, the precision ρ_0 corresponds to the sub-pixel level for the extracted referenced lines from the original image. With respect to the other extracted lines from the simulated images, there are two types. The first type corresponds to those fragmented lines that should be merged with the referenced lines as previously described. They are projected on the referenced lines, and thus error is not introduced. The second type corresponds to those lines that are newly produced and independent (i.e., merging is not required), and it is difficult to calculate or precisely directly validated their precision although they are mathematically inferred.

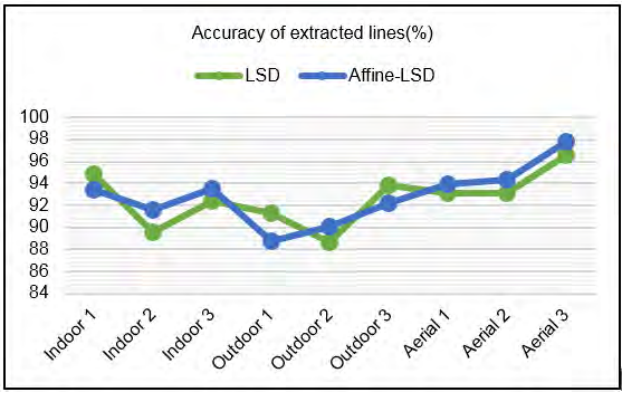


FIGURE 26. Quantitative accuracy evaluations of LSD and Affine-LS.

The simulated images are obtained from the affine transformation matrix A. With respect to a pixel, the relationship between its pixel coordinates on the original image (u, v) and coordinates on the simulated image (u', v') is as follows:

$$(a_0, b_0)^T + A\,(u, v)^T = \left(u', v'\right)^T \tag{11}$$

where a_0 and b_0 denote constants.

With respect to equation (11), the left-hand-side corresponds to the float value and the right-hand-side must correspond to an integer value. Thus, the precision of extracted line on the simulated image decreases (i.e., the precision of the back-projected line segment is simply determined via its two endpoints). Typically, 0.5 pixel is lost for u' and v'. Additionally for a point, the precision loss corresponds to $\sqrt{2} * 0.5 = 0.7$. If a compromised value 0.5 pixel is assigned for ρ_0, then the precision of the second type of lines is expressed as follows:

$$\rho = \sqrt{\rho_0^2 + 2{*}0.7^2} = 1.1$$

It is noted that an error of 1 pixel is visually distinguished.

2) ACCURACY EVALUATION

We adopt the method in [24] to evaluate the performances of the two line segment extraction methods. If the distances between the pixels from an extracted line segment and corresponding pixels from the ground-truth line segment are all less than or equal to 1 pixel, then the extracted line segment is considered as a true result, and otherwise the result is false. The accuracy result is shown in Figure 26. The overall accuracy of Affine-LSD is equal to that of the LSD. On an average, the accuracy of the Affine-LSD is approximately 90%. Evidently, the accuracy slightly decreases with respect to scenes that contain significantly irregular objects (for e.g., Outdoor 1 and Outdoor 3 contain a significant amount of trees and grasses). The accuracy increases with respect to regular scenes that contain mainly structured objects (for e.g., aerial images are dominated by buildings). The Affine-LSD also outperforms LSD in Indoor 2 and Outdoor 2 due to the redundant scenes, and Affine-LSD purifies a few false lines.

C. APPLICATION OF LINE MATCHING

The results of the aforementioned experiment confirm that Affine-Lines increases the length of the line segments and improve the fragmentation effect. Subsequently, line matching is used to further explain the significance of the line extraction improvement.

The basic steps and principles of general line matching are as follows: first, line segments of two images are extracted via a line extraction method, and matching is then performed based on the similarity of line segment neighborhoods. The range of the neighborhood is related to the length and distribution of line segments. Hence, increases in the length of the line segment increase the availability of neighbor information, increase the reliability of similarity matching, and make it easier to successfully match. The characteristic number-line match (CN-LM) [25] corresponds to a state-of-the-art line segment matching method that is invariant to rotation, scale, and affine. The principle of matching involves first defining the neighborhood ranges based on the length and distribution of the to-be-matched line segments, subsequently determining more than three matched points in their neighborhoods, and then constructing a point-to-line invariant via the geometric constraint relationship between the matched point and line segments. Similarity matching is performed based on the invariants. Thus, increases in the length of the line segment increase the number of matched points that exist in its neighborhood, increase the number and probability of constructing invariants, and increase the stability of matching. As shown in Figure 27, the red rectangle dotted box represents the neighborhood of the line segment. When the line segment in the left picture is short, its neighborhood contains only three points. When the length of the line segment increases, it can contain two more points, and thus the geometry of the point-to-line invariant is more stable.

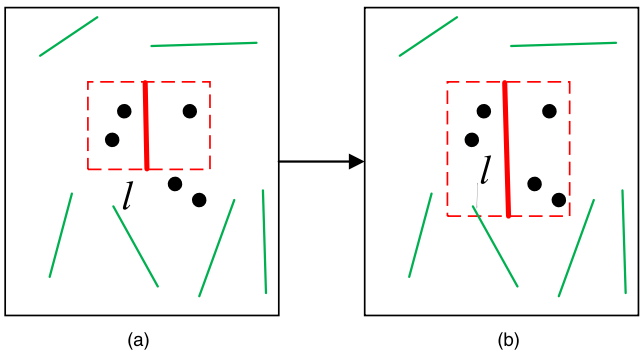


FIGURE 27. Changes in neighborhood information before and after the length of the line segment changes. (a) Neighborhood of the original short segment. (b) Neighborhood after the length of the line segment increases.

The following group of image pairs are selected for comparative experiments, namely the Memorial Hall of Peking University. A part of line extraction of the CN-LM method is used by LSD. Therefore, the comparison experiment uses Affine-LSD to optimize the line extraction, and the improved

longer line segments are used as the line segments that are to be matched. Subsequently, the same matching strategy as CN-LM is adopted. Figure 28 shows the result images, number of extracted line segments, number of correctly matched line segments, correct matching rate, and average length of the matched line segments.

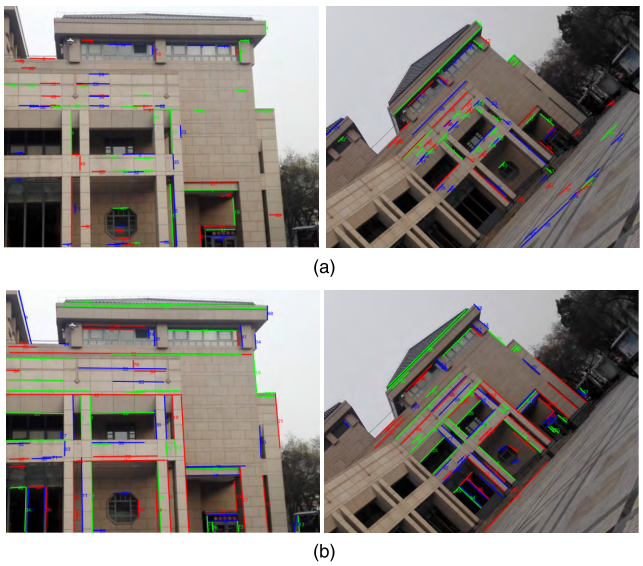


FIGURE 28. Comparison of matching results using CN-LM directly and using CN-LM after the line segment improvement. (a) Matching results of CN-LM. Extracted Lines: (612,560), Correct matched lines: 53, Correct matching rate: 67.1%, Average length: 51.2. (b) Matching results after using Affine-Lines. Extracted Lines: (442,442), Correct matched lines: 76, Correct matching rate: 86.4%, Average length: 102.9.

As shown in Figure 28, increases in the length of the line segment increase the number of correctly matched line segments, total number of matches, correct matching rate, and average length of the matched line segments. With respect to the performance on the building: the broken line segments decrease, and the matched line segments are more complete and more capable of reflecting the main structural skeleton of the buildings. The results reveal that the reconstructed 3D structural line segments are also more complete based on the aforementioned types of matched line segments.

VII. CONCLUSION

The fragmentation effect, length of the line segment, and redundancy in line extraction constitute obstacles for subsequent target detection, recognition, matching, and 3D reconstruction. In the study, the causes of the aforementioned problems were determined via analyzing the LSD algorithm principle, and an improved method of line extraction based on an affine camera model was proposed to compensate for the defects. An affine camera model was used to simulate an image several times to obtain more line segments of different lengths without changing the basic content of the image. Subsequently, purification and optimization were used to obtain longer, more complete, and more streamlined line segments. Additionally, the significance and advantages of the proposed method were verified with respect to line matching. However, a disadvantage of the proposed method corresponds to the time cost problem, and this is unable to satisfy requirements in real-time processing. Subsequently, it is necessary to reduce time consumption to the maximum possible extent and perform fast processing for purification and optimization.

ACKNOWLEDGMENT

The authors would like to thank the editor and anonymous reviewers for their constructive comments and suggestions that improved the quality of the study.

REFERENCES

[1] T. Partovi, R. Bahmanyar, T. Krauß, and P. Reinartz, "Building outline extraction using a heuristic approach based on generalization of line segments," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 3, pp. 933–947, Mar. 2017.

[2] H. H. Trinh and K. H. Jo, "Line segment-based facial appearance analysis for building image," in *Proc. 1st Int. Forum Strategic Technol.* Oct. 2006, pp. 332–335.

[3] Q. Wang, L. Yan, Y. Sun, X. Cui, H. Mortimer, and Y. Lin, "True orthophoto generation using line segment matches," *Photogramm. Rec.*, vol. 33, no. 161, pp. 113–130, 2018.

[4] L. Zhang and B. K. Ghosh, "Line segment based map building and localization using 2d laser rangefinder," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, Apr. 2000, pp. 2538–2543.

[5] J. Crowley, "Navigation for an intelligent mobile robot," *IEEE J. Robot. Autom.*, vol. JRA-1, no. 1, pp. 31–41, Mar. 1985.

[6] A. Elqursh and A. Elgammal, "Line-based relative pose estimation," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3049–3056.

[7] M. Gerke, "Using horizontal and vertical building structure to constrain indirect sensor orientation," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 3, pp. 307–316, 2011.

[8] S. N. Von, P. Lothe, J. Witt, and B. Jähne, "Relative pose estimation from straight lines using optical flow-based line matching and parallel line clustering," in *Proc. Int. Joint Conf. Comput. Vis., Imag. Comput. Graph.* Cham, Switzerland: Springer, Feb. 2016, pp. 329–352.

[9] S.-N. Yu, J.-H. Jang, and C.-S. Han, "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel," *Automat. Construction*, vol. 16, no. 3, pp. 255–261, 2007.

[10] T. Santos *et al.*, "PLineD: Vision-based power lines detection for unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, Apr. 2017, pp. 253–259.

[11] 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.

[12] J. Canny, "A computational approach to edge detection," in *Readings in Computer Vision*. San Mateo, CA, USA: Morgan Kaufmann, 1987, pp. 184–203.

[13] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic Hough transform," *Comput. Vis. Image Understand.* vol. 78, no. 1, pp. 119–137, 2000.

[14] R. G. Von Gioi, J. Jakubowicz, and G. Randall, "Multisegment detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, vol. 2, Sep. 2007, p. II-253.

[15] 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, Jun. 2010.

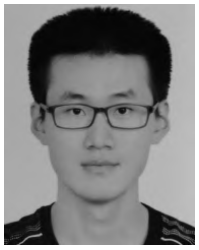
[16] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, 2011.

[17] X. Lu, J. Yao, K. Li, and L. Li, "CannyLines: A parameter-free line segment detector," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 507–511.

[18] *CannyLines: A Parameter-Free Line Segment Detector*. Accessed: 2016. [Online]. Available: http://cvrs.whu.edu.cn/projects/cannyLines/

[19] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *J. Vis. Commun. Image Represent.*, vol. 24, no. 7, pp. 794–805, 2013.

[20] G. Yu and J.-M. Morel, "ASIFT: An algorithm for fully affine invariant comparison," *Image Process. On Line*, vol. 1, pp. 11–38, Feb. 2011.
[21] G. Yu and J. M. Morel, "A fully affine invariant image comparison method," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 1597–1600.
[22] 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.
[23] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector," *Image Process. On Line*, vol. 2, pp. 35–55, Mar. 2012.
[24] F. Wu, S. Li, B. Wang, and J. Ma, "Line segment detection based on probability map," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 10875–10879.
[25] 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.* Cham, Switzerland: Springer, 2016, pp. 599–611.

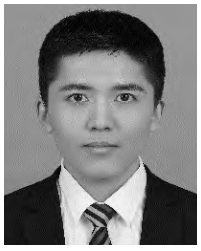


QIANG WANG received the B.S. degree in GIS from Liaoning Technical University, in 2012, and the Ph.D. degree in photogrammetry and remote sensing from the China University of Mining and Technology, Beijing, in 2018.

He is currently a Lecturer with the School of Geographic and Environment Science, Tianjin Normal University, Tianjin, China. His research interests include image processing and close-range and aerial photogrammetry.

YONGSHENG ZHANG received the B.S. degree in applied geophysics from Jilin University, in 1996.

He is currently a Senior Engineer in the level of Researcher with the China Nuclear Industry 23 Construction Co., Ltd. Shenzhen, China. His research interests include data analysis, and intelligent measurement and engineering surveying.



YANYAN LI received the B.S. degree in electronic science and technology from Anhui Polytechnic University, Wuhu, China, in 2013, and the M.S. degree in electronics from Peking University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree in computer vision with the Technical University of Munich, Munich, Germany.

His research interests include visual simultaneous localization and mapping, and 3-D reconstruction based on learning method.

ZHENXIN ZHANG received the Ph.D. degree in geoinformatics from the School of Geography, Beijing Normal University, Beijing, China, in 2016.

He is currently an Associate Professor with the Beijing Advanced Innovation Center for Imaging Theory and Technology, Capital Normal University, Beijing, and also with the Key Laboratory of 3D Information Acquisition and Application, College of Resource Environment and Tourism, Capital Normal University. His research interests include light detection and ranging data processing, quality analysis of geographic information systems, remote-sensing image processing, and algorithm development.

TIEJUN CUI received the Ph.D. degree from the Department of Civil Engineering, Bundeswehr University, Munich, Germany, in 1998.

Since 2000, he has been a Professor with the PLA Institute of Surveying and Mapping. He is currently a specially appointed Professor with Tianjin Normal University, Tianjin, China. His research interests include digital cartography and multi-source spatial data fusion.

LEI YAN received the B.E. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1982, the M.E. degree from the Navy University of Engineering, Nanjing, in 1989, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1993.

From 1998 to 2000, he was a Full Professor and an Academic Leader of the One Hundred Famous Scientists' Plan with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China. Since 2001, he has been a Professor and the Head of the Beijing Key Laboratory of Spatial Information Integration and Its Applications, School of Earth and Space Sciences, Institute of Remote Sensing and Geographic Information System, Peking University, Beijing. He has authored four books and more than 200 articles, and holds 14 patents. His major research interests include digital imaging, airborne remote-sensing systems, and remote-sensing image processing.