Automatic and Accurate 3D Railway Line Extraction and Reconstruction in Multiple Aerial Images with Kalman Filter

Dong Wei[a,1], Xiaotong Li[a,1], Yongjun Zhang[a,*], Chang Li[b] and Ziqian Huang[a,1]

[a]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, 430072, P.R.China

[c]College of Urban and Environmental Science, Central China Normal University, Wuhan, 430072, P.R.China

ARTICLE INFO

Keywords:
3D line segments
line clustering
line triangulation
3D line evaluation

ABSTRACT

Three-dimensional (3D) lines require further enhancement in both clustering and triangulation. Line clustering assigns multiple image lines to a single 3D line to eliminate redundant 3D lines. Currently, it depends on the fixed and empirical parameter. However, a loose parameter could lead to over-clustering, while a strict one may cause redundant 3D lines. Due to the absence of the ground truth, the assessment of line clustering remains unexplored. Additionally, 3D line triangulation, which determines the 3D line segment in object space, is prone to failure due to its sensitivity to positional and camera errors.

This paper aims to improve the clustering and triangulation of 3D lines and to offer a reliable evaluation method. (1) To achieve accurate clustering, we introduce a probability model, which uses the prior error of the structure from the motion, to determine adaptive thresholds;

1. Introduction

Currently, the length of the railway has exceeded 1.3 million kilometers on the earth, for which the maintenance and development of railways have a significant impact on safe operations. As the preliminary stage of extracting 3D railway track (RT) accurately and efficiently, to support engineering design, monitor construction quality, and ensure operational safety, has become one of the basic components in the maintenance of existing railways.

The extraction of RT can be achieved by real-time kinematics, LiDAR, and multiple images. The real-time kinematic is generally mounted on a railway measurement vehicle and obtains the RT by moving along the rail track. In general, it has a satisfactory accuracy while requiring operations on the track, thus demanding the cooperation of railway departments, and there are issues related to both safety and efficiency. LiDAR sensors can be mounted on a drone, which is more convenient and secure than real-time kinematic. Because a further process, like point segmentation or classification, is required for RT extraction, the drone must maintain a low flight altitude to satisfy the standards of the point-cloud density, which would impact the efficiency. A drone with cameras can capture aerial images efficiently with a safe distance from the railway area. But RT extraction is challenging in aerial images: (1) The dense points reconstructed with aerial images are inaccurate around the railway track because of the occlusion and matching problems caused by the parallax variation. (2) Joining image semantics to obtain RT might be workable; However, how to detect the semantics of RT accurately and completely in aerial images remains to be studied.

If we reconstruct the dense points from multiple aerial images and detect the RT from point clouds, the overall method of finding RT is similar to deal with the point clouds that obtained from mobile laser scanning (MLS) or airborne laser scanning (ALS). Generally, the RT can be detected with semantic segmentation, while the significant noise, inaccurate edge localization, and large density variations of point clouds bring about great challenges to the robust semantic segmentation. Thus, most general segmentation algorithm cannot be used directly in RT segmentation; instead, the carefully designed geometric priors was used to guide the segmentation and the grouping of RT: such as constructing the shape features and density data on the basis of railway bed extraction. However, these methods relies heavily on the quality and density of the point cloud, thus requiring the drone to maintain a low flight path to improve point cloud quality and reduce the processing range. Compared with point clouds, images contains rich

*Corresponding author
weidong@whu.edu.cn (D. Wei); zhangyj@whu.edu.cn (Y. Zhang); lichang@ccnu.edu.cn (C. Li)
[1]Co-first authors.

semantic informations. Thus, several studies exploited the deep learning method that design the network for training and detect the RT from aerial images, which demonstrated the effectiveness of deep learning technology in RT extraction. Moreover, the deep learning method relies heavily on training samples and considering the texture of railway regions varies greatly across the world, it may require an increased number of training samples to obtain a more generalizable detection network. In addition, these methods just deal with single frame and lacked the strategy of processing multiple aerial images.

This paper propose the accurate RT extraction for multiple aerial images, which fully exploits the contexture and geometry informations across multiple images:

- To exploit the geometry constraint of RT across multiple images, we extract the straight line from images as the basic geometry cell, for which we propose the robust clustering and triangulation methods. We first propose the noise-resistant clustering across multiple images to obtain the complete and non-redundant 3D line; Then, we propose the novel and accurate triangulation algorithm to refine the 3D line position of the RT.
- To exploit the rich texture information in images, our clustering method exploit the deep features of existing network trianed from millions of images, rather than a new network specifically designed for RT extraction; thus requiring non pre-training, which generaly needs expensive samples.

Compared to LiDAR based methods, we use more affordable imaging drones to conduct an efficient and safer railway aera maping than ALS drones or MLS equipments, and the rich contexture is exploited to compensate for issues caused by point cloud quality. Compared to the former image-based methods, we propsose the complete clustering and reconstruction strategies that obtain the accurate and non-redundant 3D RT from multiple aerial images; and non pre-training is required due to the utilization of geometry guidance in multiple images.

2. Related works

3. Methodology

The flow of our methodology is presented in Figure 1. We take aerial images of the railway area as input, for which SfM and point clouds are used in advanced with existing software. We first extract the image line and convert it to a space line with the locally optimal plane of the point cloud. Then, we cluster the single 3D line to RT candidates with the frame work derived from DBSCAN, during which the texture information of multiple images extracted from ResNet is used as one of the inlier distance. Having obtained the RT cluster, we then trace and reconstruct the vector-based RT in the Kalman framwork, which fully exploits the RT structure and the multi-view geometries to resolve the uncertainty caused by initial image line segment extraction and the point cloud error. The railway-track pair is the start seed of our Kalman method. We first convert image lines to 3D lines with the local optimal plane of the point cloud. Then, we cluster the single 3D line to RT pairs with DBSCAN frame work, during which the deep feature of multiple images is used as the inlier distance.

3.1. Railway track with Kalman filter

For a point $\mathbf{p} = (x, y, z)$ on the *RL*, we can calculate its normalized local direction $\mathbf{d} = (dx, dy, dz)$, and we use two points and directions to represent the state of the local *RLP*. Denote the two points as $\mathbf{p}_1$ and $\mathbf{p}_2$, and denote the directions of the two points as $\mathbf{d}_1$ and $\mathbf{d}_2$. Obviously, the *RLP* has fixed geometry patterns we introduce two geometry constraints for them in the filter process: (1) $\mathbf{d}_1$ and $\mathbf{d}_2$ should be as close as possible; and (2) the change in distance between $\mathbf{p}_1$ and $\mathbf{p}_2$ is as small as possible during the filter process. Then, the state to be estimated in Kalman filter is

$$\mathbf{x_k} = \begin{bmatrix}\mathbf{p}_{k,1} & \mathbf{d}_{k,1} & \mathbf{p}_{k,2} & \mathbf{d}_{k,2} & \mathbf{p}_{k,1} - \mathbf{p}_{k,2} & \mathbf{d}_{k,1} - \mathbf{d}_{k,2}\end{bmatrix}^\top \in R^{18}. \tag{1}$$

In the following contents, we use the superscript *pre* like $\mathbf{x}^{pre}$ or $\mathbf{p}_1^{pre}$ to denote the prediction variable and use superscript *obs* like $\mathbf{x}^{obs}$ or $\mathbf{p}_1^{obs}$ to show the observation variable. We use a scalar t to control the prediction during the transition along the *RL*:

$$\mathbf{x}_k^{pre} = \mathrm{F}\mathbf{x}_{k-1}, \quad \mathrm{F} = \mathrm{diag}\,(\mathrm{I}, t\cdot\mathrm{I}, \mathrm{I}, t\cdot\mathrm{I}, \mathrm{I}, \mathrm{I}) \tag{2}$$

where I is 3×3 identity matrix. In each transition, the observation state is acquired as follows: for $\mathbf{p}_{k,i}^{obs}$ and $\mathbf{d}_{k,i}^{obs}$, we reconstruct the 3D line with multiple images (Section 3.2), with which the actual position and direction can be

calculated; $\mathbf{p}_{k,1}^{obs} - \mathbf{p}_{k,2}^{obs}$ is set as $\mathbf{0}$ to make the direction consistent; for $\mathbf{d}_{k,1}^{obs} - \mathbf{d}_{k,2}^{obs}$, we want it to be equal during the transition, thus setting it as the last state directly. Thus, the observation state is summarized by

$$\mathbf{x}_k^{obs} = \begin{bmatrix} \mathbf{p}_{k,1}^{obs} & \mathbf{d}_{k,1}^{obs} & \mathbf{p}_{k,2}^{obs} & \mathbf{d}_{k,2}^{obs} & \mathbf{p}_{k-1,1} - \mathbf{p}_{k-1,2} & \mathbf{0} \end{bmatrix}^\top \in R^{18}. \tag{3}$$

Thus, the measurement can be the same as $\mathbf{x}_k$, and the general Kalman filter to track the *RPL* is

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathrm{K}\left(\mathbf{z}_k - \hat{\mathbf{x}}_k^-\right), \quad \mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k, \tag{4}$$

where $p\left(\mathbf{v}_k\right) \sim N(0, R)$, $\hat{\mathbf{x}}_k^-$ is the prediction with Eq. (2), and K is the Kalman gain that iteratively calculated from the estimate error. Please refer to (ref) for the details of Kalman gain.

Now

3.2. Accurate railway line reconstruction

3.3. The seed generation for railway track

We construct the rough 3D line based on the dense point cloud. Given the end point $\mathbf{p}$ of a 2D line segment, we randomly sample three space points around $\mathbf{p}$ to construct the plane π, and we cast a ray $\mathbf{r}$ passing through $\mathbf{p}$ from the camera center. Then, the 3D point candidate $\mathrm{P} \in R^{3\times1}$ for $\mathbf{p}$ can be obtained by $\mathbf{r}$-to-π intersection, and the candidate 3D line L can be represented by the two 3D point:

$$L = \left\{ inter\left(\pi, \mathbf{r}_1\right), inter\left(\pi, \mathbf{r}_2\right) \right\}, \tag{5}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the rays of the two endpoints, and *inter* calculates the ray-to-plane intersection. After random sampling of n times, we obtain a set of candidate 3D lines $\{L\}_{i=1}^{n}$ and use the LMEDS algorithm, which does not require an inlier threshold, to confirm the best 3D line for a 2D line:

$$L^* = \arg\min_{L_i} \operatorname{median} \left\{ d_{ij} \right\}_{j=1}^{n}, \tag{6}$$

where d_{ij} is the projection distance between L_i and L_j.

We group two 3D lines as a RT pair based on their angle $\theta_{i,j}$, overlap $o_{i,j}$, and projection distance $d_{i,j}$:

$$\left\{ RT = \left(L_i, L_j\right) \mid \theta_{i,j} < t_\theta, o_{i,j} > t_o, d_{i,j} \in I \right\}, \tag{7}$$

$\theta_{i,j}$ and $o_{i,j}$are easy to choose, e.g., 5° and 60%, because the RT pair is parallel and highly overlapped; while the interval I needs the rough width ω between the two RT, which can be acquired from construction standards or point clouds. We recommend setting $I = \left[2/3\omega, 4/3\omega\right]$ that uses one-third of ω as the margin of error. Because a 3D line may satisfy Eq. (7) with many others, the greedy algorithm is used to assign the candidate pair, which uses the sum of the overlap rate as the maximum score.

We sort the RT based on their scores of the geometry alignment and select the top 10% RT and use contextual information to further validate the RT pair. In detail, if the RT's central line is within 1° and t projection distance with another RT, its score is increased by $\mathcal{N}\left(\mu, (t/3)^2\right)$. we use the global average pooling layer in ResNet50 to describe the feature of the RT pair. Because it has been trained on massive amounts of data and can capture texture information for classification in the absence of labels. Also, we re-transform the image blocks to reduce the ambiguity caused by scale and rotation. After extraction of RT features, we use DBSCAN to group them with the cosine distance, and retain the group with the highest number as the seeds of RT.

References