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 first define

We take aerial images along with the camera matrix of the railway area as input. We first reconstruct the 3D line with our proposed algorithm that are publicly available in. Then, we cluster the single 3D line to initial RT seed with the alignment in both geometry and deep features. Start with each initial seed, we trace and reconstruct the 3D RLP in the Kalman framwork, which fully exploits the RLP structure and the multi-view geometries to achieve accurate and robust RLP reconstruction.

3.1. Initial seed generation

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

$$\left\{ RT = \left(L_i, L_j\right) \mid \theta_{i,j} < t_\theta, o_{i,j} > t_o, d_{i,j} \in I \right\}, \tag{1}$$

$\theta_{i,j}$ 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. (1) 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 $\mathcal{N}\left(\mu, (t/3)^2\right)$.

Considering the texture along the *RLP* should be roughly the same, as illustrated in Fig. 3, we use the global average pooling layer in ResNet101 as the basic feature, which has been trained on massive amounts of data and can capture texture patterns for classification in the absence of labels, to confirm the initial seed and check for termination. Denoting $\mathbf{f} \in R^n$ as the *RLP* feature, we acquire the set of features $\left\{\mathbf{f}_i\right\}_{i=1}^m$ from the m support images. To reduce the ambiguity of the deep feature caused by scale and rotation, the image block is transformed to ensure that the center line of *RLP* passes through the image center horizontally and the width of *RLP* is half of the image. 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.

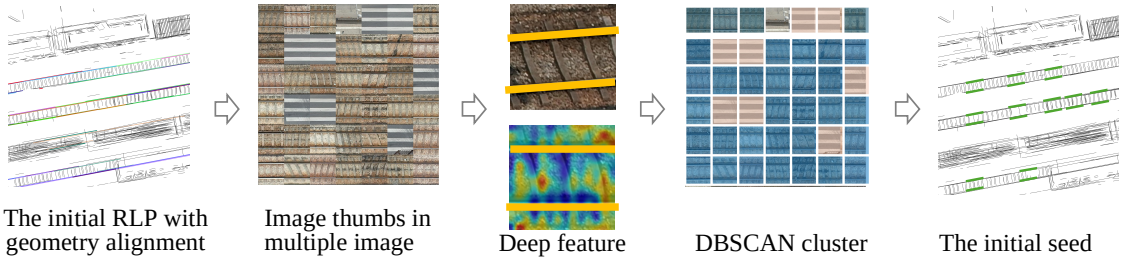

Figure 1: Visual alignment check in Kalman Filter. We project the estimation of *RLP* in multiple images and extract the deep feature with pre-trained network. Then, we check their cosine distance with the first $n/2$ cells in the queue, where n is obtained by dividing the *RLP* width by the step size t.

3.2. Railway track with Kalman filter

As shown in, we use two points and directions to represent the state of the local *RLP*. Denote the point and direction as $\mathbf{p} = (x, y, z)$ and $\mathbf{d} = (dx, dy, dz)$, respectively. The vector to be estimated for the *RLP* is

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{d}_1 & \mathbf{d}_2 \end{bmatrix}^\top \in R^{12}. \tag{2}$$

The prediction of $\mathbf{x}$ is controlled by a scalar t:

$$\mathbf{x}^{pre} = \mathrm{F}\mathbf{x}^{-}, \in R^{12}, \quad \mathrm{F} = \mathrm{diag}\left(t \cdot \mathrm{I}_{6\times6}, \mathrm{I}_{6\times6}\right) \in R^{12\times12}, \tag{3}$$

where the superscript $^-$ marks the previous state. For each prediction $\mathbf{x}^{pre}$, There is an actual observation $\mathbf{x}^{obs}$ arising from line reconstruction in multiple images (Section 3.3). The *RLP* has fixed geometry patterns, i.e., $\mathbf{d}_1$ and $\mathbf{d}_2$ should be as close as possible, and the distance change between $\mathbf{p}_1$ and $\mathbf{p}_2$ is as small as possible. We achieve these two constraints by extending the observation vector:

$$\mathbf{z}^{obs} = \begin{bmatrix} \mathbf{x}^{obs} & \mathbf{p}_1^- - \mathbf{p}_2^- & \mathbf{0}_{1\times3} \end{bmatrix}^\top \in R^{18}. \tag{4}$$

Correspondingly, the observation matrix that translate $\mathbf{x}^{pre}$ to the observation form is

$$\mathbf{z}^{pre} = \mathrm{H}\mathbf{x}^{pre}, \quad \mathrm{H} = \begin{bmatrix} & \mathrm{I}_{12\times12} & \\ \mathrm{I}_{3\times3} & -\mathrm{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathrm{I}_{3\times3} & -\mathrm{I}_{3\times3} \end{bmatrix} \in R^{18\times12}. \tag{5}$$

Then, as shown in Fig. 2, we use the general discrete Kalman filter to update the state.

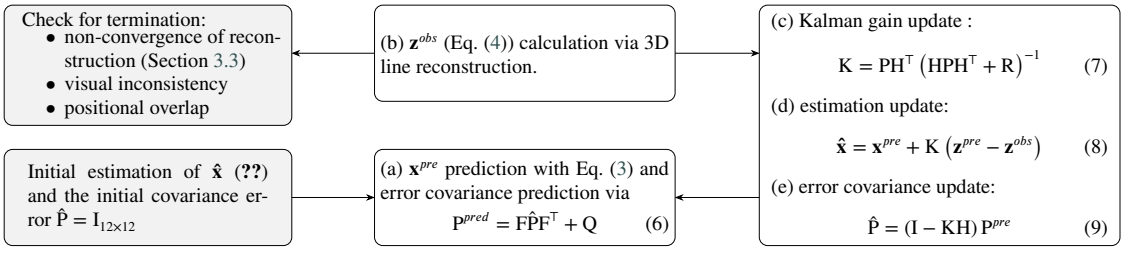

Figure 2: The flow of kalman filter for the *RLP* estimation.Q in Eq. (6) and R in Eq. (7) represents the covariance matrix of observation noise and process noise,respectively.

3.3. Accurate railway line measurement

For a 3D line in $\mathbf{x}^{pre}$ (Eq. (3)), we convert it into image with camera matrix and obtain the 2D line segment $\mathbf{l}^{pre} = \left[x_c, y_c, \theta\right]$; then we search around $\mathbf{l}^{pre}$ for the observation $\mathbf{l}^{obs}$, which should have the maximum gradient

response:

$$\mathcal{L} = \sum_{i=1}^{N} \lambda_i \cdot \| G_x\left(x_i, y_i\right), G_y\left(x_i, y_i\right) \|^2, \tag{10}$$

where G_x and G_y is the gradient magnitude in two dimensions; the sample point is calculated by

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} h_i \\ p_i \end{bmatrix}, \tag{11}$$

where h_i and p_i is the parallel and horizontal distance of $\mathbf{l}^{obs}$, respectively. Because the *RL* has a width that is different at different positions in different images, we use λ_i to weight the gradients:

$$\lambda_i = 1 - \frac{1}{1 + e^{10(w_1 - p_i)}} + \frac{1}{1 + e^{10(w_2 - p_i)}} \tag{12}$$

where w_1 and w_2 are the distance calculated from the the prior width of *RL*. We use the gradient ascend,

$$\mathbf{l}_i^{obs} = \mathbf{l}_{i-1}^{obs} - \alpha \Delta \mathcal{L}, \tag{13}$$

to find $\mathbf{l}^{obs}$ with Eq. (10), where α is the learning rate. $\Delta\mathcal{L}$ is the gradient from Eq. (10) and Eq. (11):

$$\Delta \mathcal{L} = -\sum_{i=1}^{N} \lambda_i \mathbf{J}_i \left[G_x(x_i, y_i) \quad G_y(x_i, y_i) \right]^\top, \tag{14}$$

where $\mathbf{J}_i$ is the Jacobian matrix

$$\mathbf{J}_i = \begin{bmatrix} \partial x_i/\partial x_c & \partial x_i/\partial y_c & \partial x_i/\partial \theta \\ \partial y_i/\partial x_c & \partial y_i/\partial y_c & \partial y_i/\partial \theta \end{bmatrix}^\top \tag{15}$$

Eq. (13) takes $\mathbf{l}^{pre}$ as input and iterates until the parameters converge, i.e., when the change in the parameters is smaller than a predefined threshold. After the measurement in multiple images, we obtain a set of image lines and the accurate 3D line is reconstructed with the method in (ref).

3.4. Initial seed and termination with deep features

Check for termination. We save the set that has passed texture validation in a queue and use the first half of the elements in the queue to validate the new $\left\{\mathbf{f}'\right\}_{i=1}^{m1}$:

$$\sum_{i=1}^{m_1} \mathbb{I}\left(\exists j \in \left[m_2\right], \cos\left(\mathbf{f}_i, \mathbf{f}_j'\right) > \theta\right) \geq \frac{m_1}{2} \tag{16}$$

where $m2$ is the total number of $\mathbf{f}$ in the first half of the feature set in the queue; $\mathbb{I}$ represents an indicator function which takes the value of 1 when a certain condition is true and 0 otherwise. If $\left\{\mathbf{f}'\right\}_{i=1}^{m1}$ satisfy Eq. (16), it is pushed to the queue, and then the queue is dequeued if its cell passed cal.

References

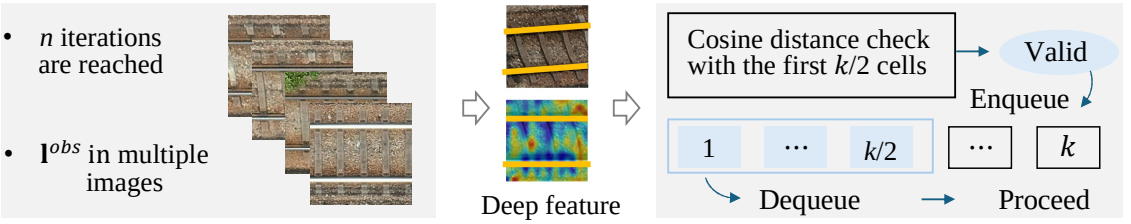

Figure 3: Visual alignment check in Kalman Filter. We project the estimation of *RLP* in multiple images and extract the deep feature with pre-trained network. Then, we check their cosine distance with the first $n/2$ cells in the queue, where n is obtained by dividing the *RLP* width by the step size t.