

# Railway Line Reconstruction in Images via Kalman Filter

Dong Wei<sup>a,1</sup>, Xiaotong Li<sup>a,1</sup>, Yongjun Zhang<sup>a,\*</sup>, Bin Zhang<sup>a,1</sup>, Guangshuai Wang<sup>a,1</sup> and Ziqian Huang<sup>a,1</sup>

<sup>a</sup>School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, 430072, P.R.China

<sup>c</sup>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

The length of the railway has exceeded 1.3 million kilometers on earth, for which maintenance and development have a significant impact on safe operations. Thus, extracting vectorized 3D railway line (*RL*) accurately and efficiently, to support engineering design, monitor construction quality, and ensure operational safety, has become one of the basic preliminary tasks in the maintenance of the *RL*.

*RL* extraction 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 *RL* 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 further process is required for extraction of *RL* with LiDAR points, such as point segmentation or classification, the drone must maintain a low flight altitude to satisfy the requirement of point-cloud density. This would also impact safety and efficiency. A drone with cameras can capture high-resolution aerial images efficiently with a safe distance from the railway. But extraction of *RL* is challenging in aerial images: (1) The dense points of aerial images are inaccurate around the railway track due to occlusion and matching problems caused by the parallax variation. (2) Joining image semantics could be workable; but how to detect and reconstruct *RL* accurately and completely in multiple aerial images remains to be studied.

Point segmentation is the core method for detecting *RL* with point clouds, either from aerial images or LiDAR points from mobile laser scanning (MLS) or airborne laser scanning (ALS). Traditional algorithms carefully design geometric priors to guide segmentation to find *RL* and learning-based algorithms generally train the segmentation model on the basis of PointNet. Significant noise, which requires the drone to maintain a low flight path to improve the quality of the point cloud and reduce the processing range, because inaccurate edge localization, and large density variations of point clouds will present great challenges for robust semantic segmentation. Compared with point clouds, images contain rich semantic information. Thus, some studies used the deep learning method to detect *RL* from aerial images, but may require an increased number of training samples to obtain a more generalizable detection network; also, most of these methods only dealt with a single image block and lacked the strategy for processing multiple aerial images. In recent years, significant progress has been made in 3D line detection, but there are still no specific algorithms

\*Corresponding author

weidong@whu.edu.cn (D. Wei); zhangyj@whu.edu.cn (Y. Zhang)

<sup>1</sup>Co-first authors.

47 tailored for railway line detection. As a result, the reconstructed 3D lines often represent railway edges and are prone  
48 to fragmentation and missing segments.

49 This work is inspired by autonomous driving technology. If railway tracks are considered analogous to roads, lane-  
50 following techniques can be used to simulate the train autopilot. In straight sections of the railway, where the direction of  
51 travel is locally linear, trajectories can be predicted directly. For curved sections, multiview reconstruction is utilized to  
52 refine the prediction, enabling precise curve navigation. By integrating multi-view reconstruction with local prediction  
53 through a Kalman Filter, this approach effectively addresses the challenges of track reconstruction, which is often  
54 vulnerable to noise and errors. To determine the starting point of the train, we employed a joint clustering method that  
55 combines the features of the deep network from the images with the geometric features of the 3D line segments. To  
56 reconstruct a precise and detailed 3D railway track, we used a gradient descent approach to optimize the local track  
57 segments in images based on the structural characteristics of the railway. In summary:

- 58 • We propose the Kalman Filter framework, which combines both multiview geometry and semantics for automatic  
59 and accurate 3D *RL* extraction from aerial images.
- 60 • We propose an iterative optimization method with Gradient descend algorithm to find the accurate *RL* in images,  
61 which fully considers the *RL* various in different images.

62 Compared to LiDAR-based methods, we use more affordable imaging drones to conduct an efficient and safer railway  
63 map than ALS drones or MLS equipment. Compared to previous image-based methods, we propose complete tracking  
64 and reconstruction strategies that obtain accurate and vectorized 3D *RL* from multiple aerial images; also, no pre-  
65 training is required in our method.

## 66 2. Related works

67 Railway detection and line reconstruction are essential components of accurately modeling railway infrastructure.  
68 This study reviews three key aspects: point cloud-based methods for railway detection, image-based approaches for  
69 enhanced recognition, and 3D line reconstruction techniques for precise segment modeling.

### 70 2.1. Point Cloud-Based railway detection

71 3D point clouds, owing to their efficiency and precision, are widely used for railway segment detection, with  
72 both traditional and modern methods contributing to advancements in the field. Traditional approaches primarily rely  
73 on data fusion and geometric models. Beger et al. (2011) demonstrated that fusing aerial imagery with LiDAR data  
74 significantly enhances railway centerline reconstruction, highlighting the benefits of multi-source data integration.  
75 Similarly, Yang and Fang (2014) utilized shape and intensity features to model railway tracks, while Sánchez-Rodríguez  
76 et al. (2019) adopted a coarse-to-fine RANSAC strategy to refine railway tunnel power line detection. These methods  
77 showcase the potential of tailored geometric models in specific applications but suffer from sensitivity to data quality  
78 and acquisition conditions. Infrastructure-focused approaches, such as those by Cserép et al. (2022), emphasize detailed  
79 railway fragmentation and infrastructure recognition using dense LiDAR data, enabling detailed mapping of complex  
80 structures. Ye et al. (2022) extended this to high-definition lane extraction in curved road scenarios, underlining the  
81 adaptability of point cloud data for intricate transportation features. Ariyachandra and Brilakis (2023) further enhanced  
82 railway topology analysis by integrating geometric constraints, paving the way for more reliable detection and modeling  
83 of railway components, including catenary structures (Zhang et al., 2023).

84 Although these methods provide valuable information, they rely heavily on manually defined features, which limits  
85 their scalability across diverse environments. To address these challenges, deep learning approaches have emerged as  
86 a promising solution. Ma et al. (2022) introduced BoundaryNet, leveraging MLS point clouds and satellite imagery  
87 to extract road boundaries with high accuracy. Zhang et al. (2022) tackle complex geometries, such as overhead wires,  
88 by learning intricate patterns in airborne LiDAR data. Simulation frameworks like TrainSim (D'Amico et al., 2023)  
89 provide a synthetic system to train deep learning models, enhancing their robustness in real-world applications. Real-  
90 time detection systems, such as the fusion-based approach (Tang et al., 2024), combine LiDAR and camera data to  
91 improve efficiency and accuracy, and the anomaly detection techniques by Ge et al. (2024) integrate semi-supervised  
92 learning and decision fusion for robust railway inspection. Despite these advances, deep learning methods demand  
93 substantial computational resources and rely on extensive annotated datasets, as exemplified by WHU-Railway3D  
94 (Qiu et al., 2024), which serves as a benchmark for railway semantic segmentation.

## 95 2.2. Image-Based railway detection

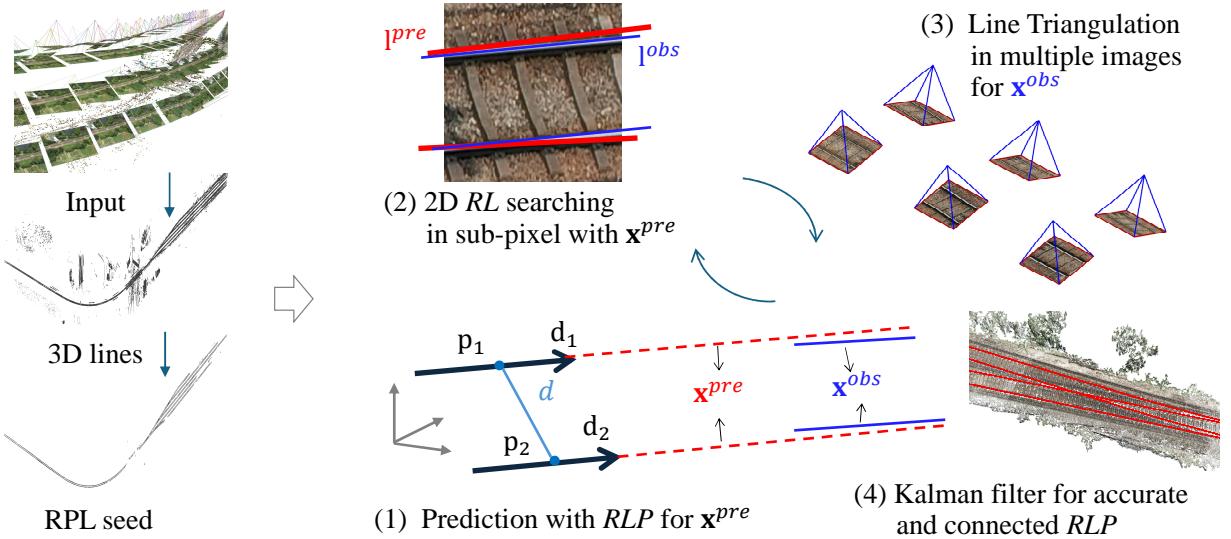
96 Image-based railway detection has been revolutionized by deep learning, leveraging recent advances in pattern  
97 recognition to improve performance across diverse scenarios. Unlike point cloud-based methods, image-based  
98 techniques benefit from the ubiquity of cameras and the rich contextual information in images, making them ideal  
99 for railway recognition. Early approaches, such as those by Yang et al. (2022a), combined discretization, filtering,  
100 and reconstruction to detect railway tracks in UAV images, demonstrating their applicability for inspection purposes.  
101 Zheng et al. (2022) improved feature integration through cross-layer refinement, enabling robust lane detection under  
102 varying environmental conditions. These methods laid the foundation for incorporating deeper geometric modeling  
103 and advanced architectures. Recent methods have focused on improving the representation of complex structures. For  
104 instance, Chae et al. (2023) reconstructed lane lines as polylines, emphasizing the importance of geometric modeling  
105 for autonomous driving. Transformer-based methods (Chen et al., 2023; Luo et al., 2023; Yao et al., 2023) introduced  
106 a paradigm shift by capturing long-range dependencies, enabling accurate detection even in complex or sparse data  
107 scenarios. Flexible anchoring mechanisms like FLAMNet (Ran et al., 2023) adapt to challenges such as lane curvature  
108 and occlusions, addressing limitations of fixed-detection models.

109 Semantic segmentation remains a cornerstone for image-based railway detection. Weng et al. (2023) enhanced tra-  
110 ditional architectures like DeepLabV3+, providing precise track extraction for automated maintenance. Furthermore,  
111 Zhang et al. (2024) advanced curved lane detection with ECPNet, a method that combines global spatial understanding  
112 with local precision, showcasing the role of adaptive techniques in refining detection accuracy. Geometry-driven  
113 methods, such as the topology-guided approach by Yang et al. (2022b), bridge the gap between traditional and deep  
114 learning methods by leveraging structural constraints to enhance detection robustness. This highlights the enduring  
115 relevance of geometric insights in complementing data-driven techniques.

## 116 2.3. 3D line reconstruction

117 The forward intersection of multi-view line segments is performed based on segment association, where multiple  
118 planes constructed by the imaging center and image segments are intersected to solve for accurate 3D line segments.  
119 The overall framework can be broadly categorized into two types: Strict association results: These methods directly fit  
120 line segments in 3D space under strict threshold constraints (Jain et al., 2010), or iteratively adjust and optimize 3D line  
121 segments through more rigorous back-projection models (Hofer et al., 2017; Liu et al., 2021; Schmid and Zisserman,  
122 1997). Reconstruction-based methods: These approaches reconstruct 3D line segments from matched two-view line  
123 segments and then select the optimal representative line segment based on the geometric consistency of the segments  
124 (Wei et al., 2022). Although these methods typically achieve suboptimal positioning accuracy compared to the first  
125 type, they are less sensitive to gross errors in matching and feature association, balancing reconstruction efficiency and  
126 quality, and are suitable for scenarios with more relaxed matching and association constraints. The forward intersection  
127 of multi-view line segments is more complex than that of point features. When line segments are close to the epipolar  
128 plane, significant intersection errors may occur, resulting in inaccurate segment positioning (Hartley and Zisserman,  
129 2003). Although various solutions have been proposed to address this issue, their fundamental idea remains consistent:  
130 constructing an optimization function by integrating geometric information and back-projection errors, rather than  
131 solely focusing on the error model of the line segments themselves. Among these, coplanarity between line segments  
132 serves as an effective geometric constraint. For distant adjacent line segments, 3D line segments are solved jointly  
133 based on coplanarity constraints (Ok et al., 2012). Other studies reconstruct 3D line segments, group them into planes  
134 through fitting, and project the grouped segments onto the fitted planes to obtain 3D line segments.

135 Recently, deep learning methods have been explored for plane detection and grouping based on the orientation  
136 of 3D lines, using plane associations for forward intersection (Wang et al., 2020). This method produces visually  
137 regularized line clouds, but its actual positioning accuracy is significantly affected by coplanarity thresholds and scene  
138 structure. In the latest studies (Liu et al., 2021), two approaches were proposed to optimize the forward intersection of  
139 line segments: Vanishing point-based intersection constraints: This approach corrects segment errors using vanishing  
140 point constraints. It remains based on the coplanarity of line segments but allows the use of existing vanishing point  
141 detection algorithms for segment grouping. Deep learning-based matching of homologous points on line segments:  
142 Since the intersection of homologous points is only invalid near the epipolar line, this method provides robust positional  
143 constraints. These advancements demonstrate promising directions for improving both the accuracy and robustness of  
144 line segment forward intersection.



**Figure 1:** The flow of our method. It uses the aerial images as the input and produces accurate and connected 3D railway lines.

### 145 3. Methodology

146 The basic cell of our reconstruction is the railway line pair (*RLP*) consisting of two railway lines (*RL*) (Fig. 1 (1)).  
 147 We use a center point  $(x, y, z)$ , the position angle  $(\phi, \omega, \kappa)$ , and the width  $w$  of the *RLP* to shape the local *RLP*:

$$\mathbf{x} = [x, y, z, \rho, \kappa, \phi, \omega]^\top \in R^7. \quad (1)$$

148 The flow is presented in Fig. 1. Our method takes the SfM result of the aerial images as input. We first obtain the  
 149 3D line with our 2D line extraction (ref) and reconstruction (ref) algorithm. Then we cluster the single 3D line to find  
 150 the seed of *RLP* with the alignment in geometry and deep features. Starting with each seed, we trace and reconstruct  
 151 the *RLP* in the framework of Kalman filter, in which the prediction and the observation comes from the geometry  
 152 property of the *RLP* and the multiview reconstruction, respectively.

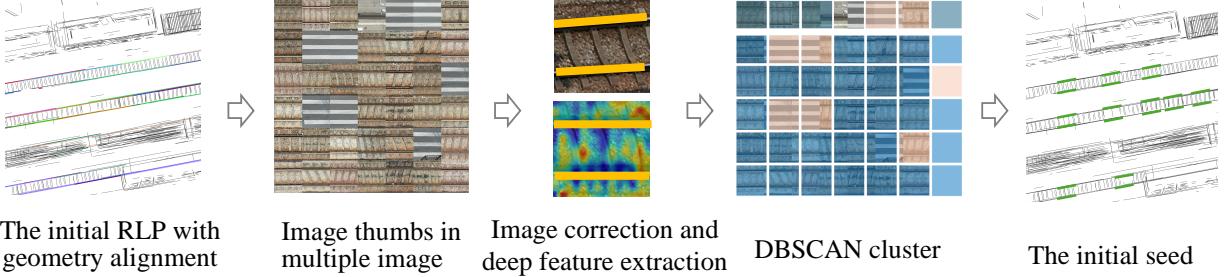
#### 153 3.1. Initial seed generation

154 With the 3D lines  $\{L\}$  reconstructed from images, we first group double lines as a seed candidate of the *RLP* based  
 155 on their angle  $\theta_{a,b}$ , overlap  $o_{a,b}$ , and projection distance  $d_{a,b}$ :

$$\left\{ (L_a, L_b) \mid \theta_{a,b} < t_\theta, o_{a,b} > t_o, d_{a,b} \in \left[ \frac{2\omega}{3}, \frac{4\omega}{3} \right] \right\}. \quad (2)$$

156  $\theta_{a,b}$  and  $o_{a,b}$  are easy to choose because the lines in *RLP* is parallel and is highly overlapped; while  $d_{a,b}$  needs the  
 157 rough width  $\omega$  of *RLP*, which can be acquired from construction standards or point clouds. We use one-third of  $\omega$  as  
 158 the margin error to reduce reliance on initial values. Because  $L_i$  may satisfy Eq. (2) with others, we score  $(L_a, L_b)$   
 159 with its geometry alignments to other candidates: we fit the central line of  $L_a$  and  $L_b$  as  $L_{ci}$ ; if  $L_{c1}$  and  $L_{c2}$  are within  
 160  $t_p$ , the scores of both  $(L_{a1}, L_{b1})$  are increased by  $\mathcal{N}(\mu, (t_p/3)^2)$ . Then, we sort the candidate based on their scores  
 161 and eliminate the pair whose 3D line has been grouped in the former pairs.

162 As illustrated in Fig. 2, considering that the texture along the *RLP* should be roughly the same, we use the global  
 163 average pooling layer in ResNet as the basic feature, which has been trained on massive amounts of data and can capture  
 164 texture patterns for classification in the absence of labels, to cluster the initial seed. Denoting  $\mathbf{f} \in R^n$  as the *RLP* feature,  
 165 we acquire the set of features  $\{\mathbf{f}_i\}_{i=1}^m$  from the support images, which can be confirmed our line reconstruction method  
 166 (ref). Before the feature extraction, the image block is corrected to ensure the center line of *RLP* passes through the  
 167 image center horizontally and the width of *RLP* is half of the image. it reduces the ambiguity of the deep feature



**Figure 2:** The initial seed with deep features. Given the line pair with geometry alignment, we extract the deep feature for their image block in multiple image, with which we use the DBSCAN to confirm the initial seed of *RLP*.

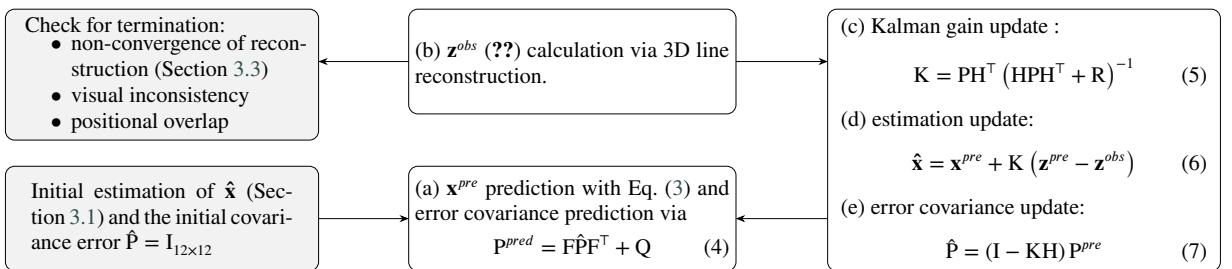
168 caused by scale and rotation. Then, we use DBSCAN to cluster the features with the cosine distance and retain the  
 169 largest group as the seeds of the *RLP*.

### 170 3.2. Railway track with Kalman filter

171 Given the initial seed of *RLP*, we take it as the start place of the train and the predict position for the running train  
 172 in the next time  $\mathbf{x}$  is controlled by  $\Delta t$  with the train's attitude:

$$\begin{aligned} x_k &= x_{k-1} + \Delta t \cdot \cos(\phi_{k-1}) \cdot \cos(\omega_{k-1}), \\ y_k &= y_{k-1} + \Delta t \cdot \sin(\phi_{k-1}) \cdot \cos(\omega_{k-1}), \\ z_k &= z_{k-1} + \Delta t \cdot \sin(\omega_{k-1}), \\ \rho_k &= \rho_{k-1}, \kappa_k = \kappa_{k-1}, \phi_k = \phi_{k-1}, \omega_k = \omega_{k-1}. \end{aligned} \quad (3)$$

173 Note the train's attitude is unchanged during the prediction. For each state, there is an actual observation  $\mathbf{x}^{obs}$  arising  
 174 from the line reconstruction in multiple images (Section 3.3). Generally, the observation may contain errors induced  
 175 from wrong 2D *rl* detection and the corresponding 3D *rl* reconstruction; while the prediction is destined to fail when the  
 176 train is passing through a bend. Thus, as shown in Fig. 3, we update the state of the train by combining the observation  
 and prediction state with the discrete Kalman filter.

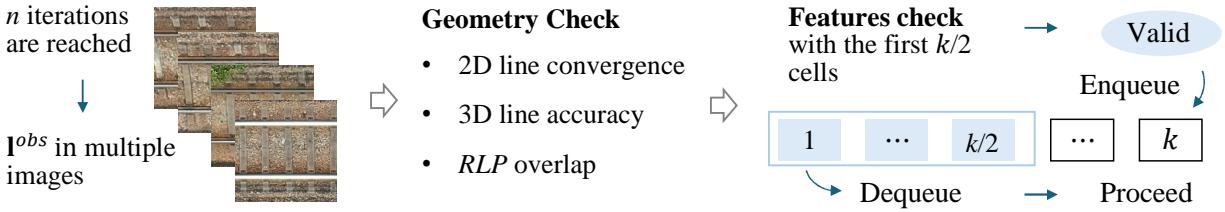


**Figure 3:** The flow of kalman filter for the *RLPestimation.Q* in Eq. (4) and *R* in Eq. (5) represents the covariance matrix of observation noise and process noise, respectively.

177 Because the triansation is a nonlinear process, we need the Jacobian matrix  $F_k \in R^{7 \times 7}$  for the linearization:  
 178

$$\begin{aligned} F_k(1, 6) &= -\Delta t \sin(\phi_{k-1}) \cos(\omega_{k-1}), \\ F_k(1, 7) &= -\Delta t \cos(\phi_{k-1}) \sin(\omega_{k-1}), \\ F_k(2, 6) &= \Delta t \cos(\phi_{k-1}) \cos(\omega_{k-1}), \\ F_k(2, 7) &= \Delta t \sin(\phi_{k-1}) \sin(\omega_{k-1}), \\ F_k(i, i) &= 1, i \in [1, 7]. \end{aligned} \quad (8)$$

179 With Eq. (8),

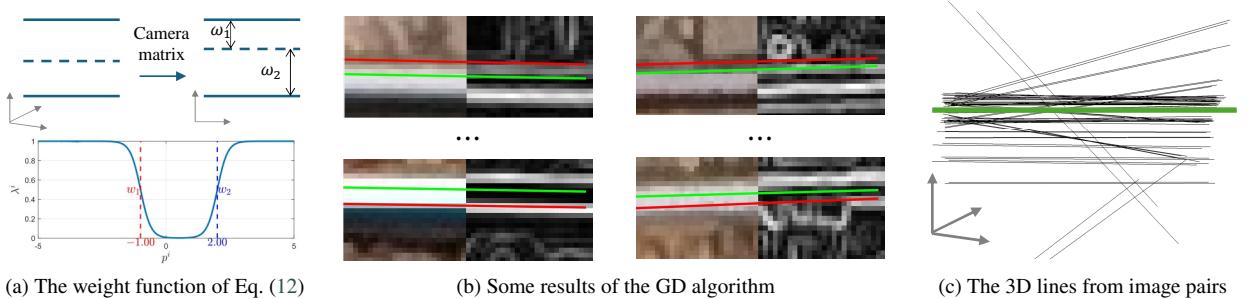


**Figure 4:** Visual alignment check in Kalman Filter. We project the estimation of  $RLP$  in multiple images and extract the deep feature with pre-trained network. Then, we check their cosine distance with the first  $n/2$  cells in the queue, where  $n$  is obtained by dividing the  $RLP$  width by the step size  $t$ .

For each seed of  $RLP$ , we will track it with the Kalman filter and terminate the track when one of the following conditions is met. (1) *Overlap*: with each of the last  $k$   $RLP$  in the current filter process, we find the nearest  $RLP$  set that have been reconstructed in the previous filter process. If half of the  $RLP$  can find a nearest one within  $\frac{1}{10}d$ , we say that the overlap has occurred. (2) *Inaligned deep features*: the texture is not aligned with the previous state, We save the set of deep features that has passed the texture validation in a queue and use the first half of the elements in the queue to validate the new  $\{\mathbf{f}'\}_{i=1}^{m_1}$ :

$$\sum_{i=1}^{m_1} \mathbb{I} \left( \exists j \in [m_2], \cos(\mathbf{f}_i, \mathbf{f}'_j) > \theta \right) \geq \frac{m_1}{2} \quad (9)$$

where  $m_2$  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  $\{\mathbf{f}'\}_{i=1}^{m_1}$  satisfies Eq. (9), it is pushed to the queue, and then the queue is dequeued if its cell passed *cal*. (3) *Degenerate reconstruction*: there is no correct 3D line that can be reconstructed from multiple images in Section 3.3. Note that it is unnecessary to check for termination in every estimate, especially when we use a small  $t$  for robustness; instead, we employ this check when  $k$  iterations are reached and roll back to the state of the previous check when the overlap or feature inconsistency occurs.



**Figure 5:** Illustration of the line reconstruction.

### 3.3. Accurate railway line position with gradient descending

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

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

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

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

197 where  $h_i$  and  $p_i$  is the parallel and horizontal distance to  $\mathbf{l}^{obs}$ , respectively. Because the  $RL$  has a width that is different  
198 at with positions and images, we use  $\lambda_i$  in Eq. (10) to weight the gradients:

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

199 where  $w_1$  and  $w_2$  are the distance calculated from the the prior width of  $RL$ .

200 We use the gradient ascend to find  $\mathbf{l}^{obs}$  with Eq. (10):

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

201 where  $\alpha$  is the learning rate.  $\Delta \mathcal{L}$  is the gradient from Eq. (10) and Eq. (11):

$$\Delta \mathcal{L} = - \sum_{i=1}^N \lambda_i \mathbf{J}_i \begin{bmatrix} G_x(x_i, y_i) & G_y(x_i, y_i) \end{bmatrix}^\top, \quad (14)$$

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

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

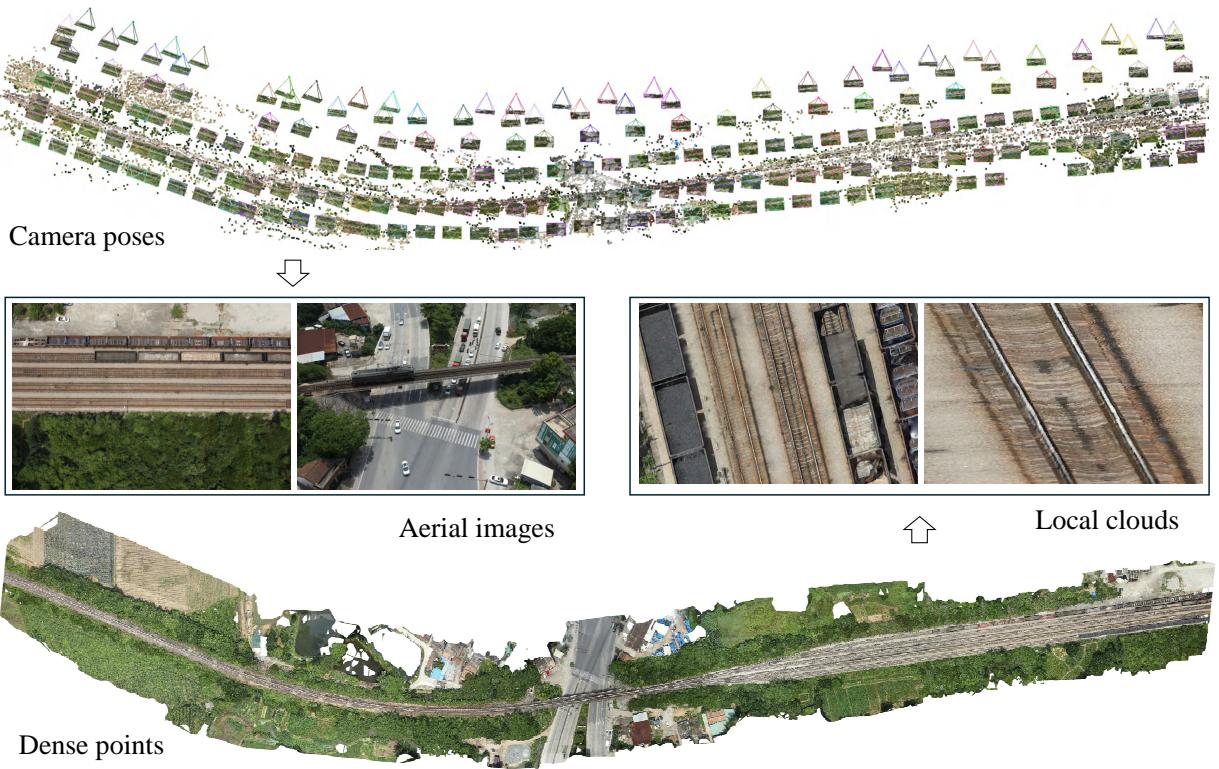
203 Eq. (13) takes  $\mathbf{l}^{pre}$  as input and iterates until the parameters converge, i.e., when the change in the parameters is smaller  
204 than a predefined threshold.

205 In the task of reconstructing the optimal 3D line segment from multiple  $\mathbf{l}^{obs}$ , we propose an efficient strategy.  
206 Specifically, for  $n$  images with  $n$  observed line segments, we first perform pairwise 3D line segment reconstructions  
207 for all image pairs, generating  $n \times (n - 1)/2$  candidate 3D line segments. Next, a score is computed for each 3D line  
208 segment by calculating the weighted sum of distances to other 3D line segments within a specified distance threshold.  
209 Finally, the 3D line segment with the highest score is selected as the optimal reconstruction. We say the degenerate  
210 reconstruction occurs when the highest score is 0 and the Kalman filter will be terminated as illustrated in Figs. 3 and 4.  
211 Using the known RLP structure, we can easily use the approximate RLP width range to define an appropriate inlier  
212 threshold, enhancing the robustness and accuracy of the method.

## 213 4. Experiments

214 We used five datasets to test our proposed algorithm. The details of the data are shown in Table 1. As shown  
215 in the figure, multiple-view images from two perspectives, one showing an aerial drone overhead operation and the  
216 other showing a vehicle operation, are rendered from a manually constructed 3D railway model. We directly generate  
217 dense point clouds from the 3D model for use by other algorithms. The other two sets of data were obtained through  
218 drones, and we manually drew the 3D railway track. Currently, there is no publicly available code for automatically  
219 reconstructing railway segments from multi-view images. Therefore, we compare our approach with deep learning  
220 algorithms based on point cloud semantics.

221 Since our algorithm does not require training with samples, when compared to a 3D semantic segmentation  
222 algorithm, it would be unfair to train a deep segmentation model and test it on the same dataset. Therefore, we use  
223 two approaches for evaluation. The first uses the The second splits a portion of the test dataset for training, with the  
224 remaining part used for validation. Because the ground truth of the  $RL$  is the vector structure, we have to deal with  
225 the cloud segmentation result for quantitative evaluation. As illustrated in figure.1, we projected the  $RL$  cloud to the  
226 ground truth and retained the points within a distance threshold, when the nearest two points are beyond the distance  
227 threshold, we say the false negative has occurred.



**Figure 6:** The initial seed with deep features. Given the line pair with geometry alignment, we extract the deep feature for their image block in multiple image, with which we use the DBSCAN to confirm the initial seed of *RLP*.

228 We use recall, accuracy, and F-score to evaluate all methods. (1) Recall measures the ability of the method to  
 229 correctly identify the true railway straight segments. It is the proportion of correctly identified straight segments (true  
 230 positives) out of all the actual straight segments in the ground truth. (2) Accuracy evaluates the overall correctness of  
 231 the method by comparing the total number of correct predictions (both true positives and true negatives) to the total  
 232 number of predictions made. (3) F-score is the harmonic mean of precision and recall, offering a balanced measure  
 233 between them. It is particularly useful when there is an uneven class distribution, giving a single metric to assess both  
 234 the precision and recall of the method.

## 235 References

- 236 M. R. M. F. Ariyachandra and I. Brilakis. Leveraging railway topology to automatically generate track geometric information models from airborne  
 237 LiDAR data. *Automation in Construction*, 155:105068, Nov. 2023. doi: 10.1016/j.autcon.2023.105068.
- 238 R. Beger, C. Gedrange, R. Hecht, and M. Neubert. Data fusion of extremely high resolution aerial imagery and LiDAR data for automated railroad  
 239 centre line reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6):S40–S51, Dec. 2011. doi: 10.1016/j.isprsjprs.2011.  
 240 09.012.
- 241 Y. J. Chae, S. J. Park, E. S. Kang, M. J. Chae, B. H. Ngo, and S. I. Cho. Point2Lane: Polyline-Based Reconstruction With Principal Points for Lane  
 242 Detection. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):14813–14829, Dec. 2023. doi: 10.1109/TITS.2023.3295807.
- 243 Z. Chen et al. Generating Dynamic Kernels via Transformers for Lane Detection. *Proceedings of the IEEE/CVF International Conference on  
 244 Computer Vision*, pages 6812–6821, 2023. doi: 10.1109/ICCV51070.2023.00629.
- 245 M. Cserép, A. Demján, F. Mayer, B. Tábori, and P. Hudoba. Effective Railroad Fragmentation and Infrastructure Recognition Based on Dense  
 246 LiDAR Point Clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2–2022:103–109, May 2022. doi:  
 247 10.5194/isprs-annals-V-2-2022-103-2022.
- 248 G. D’Amico et al. TrainSim: A Railway Simulation Framework for LiDAR and Camera Dataset Generation. *IEEE Transactions on Intelligent  
 249 Transportation Systems*, 24(12):15006–15017, Dec. 2023. doi: 10.1109/TITS.2023.3297728.
- 250 X. Ge et al. An Anomaly Detection Method for Railway Track Using Semisupervised Learning and Vision-Lidar Decision Fusion. *IEEE Transactions  
 251 on Instrumentation and Measurement*, 73:1–15, 2024. doi: 10.1109/TIM.2024.3417537.
- 252 R. Hartley and A. Zisserman. Multiple view geometry in computer vision. 2003.

- 253 Manuel Hofer, Michael Maurer, and Horst Bischof. Efficient 3d scene abstraction using line segments. *Computer Vision and Image Understanding*,  
254 157:167–178, 2017. ISSN 1077-3142. doi: <https://doi.org/10.1016/j.cviu.2016.03.017>. URL <https://www.sciencedirect.com/science/article/pii/S1077314216300236>.
- 255 Arjun Jain, Christian Kurz, Thorsten Thormählen, and Hans-Peter Seidel. Exploiting global connectivity constraints for reconstruction of 3d line  
256 segments from images. In *CVPR*, pages 1586–1593, 2010. doi: 10.1109/CVPR.2010.5539781.
- 257 Liu Liu, Hongdong Li, Haodong Yao, and Ruyi Zha. Plückernet: Learn to register 3d line reconstructions”. In *2021 IEEE/CVF Conference on*  
258 *Computer Vision and Pattern Recognition (CVPR)*, pages 1842–1852, 2021. doi: 10.1109/CVPR46437.2021.00188.
- 259 Y. Luo et al. LATR: 3D Lane Detection from Monocular Images with Transformer. In *Proceedings of the IEEE/CVF International Conference on*  
260 *Computer Vision*, pages 7907–7918, 2023. doi: 10.1109/ICCV51070.2023.00730.
- 261 L. Ma, Y. Li, J. Li, J. M. Junior, W. N. Goncalves, and M. A. Chapman. BoundaryNet: Extraction and Completion of Road Boundaries With Deep  
262 Learning Using Mobile Laser Scanning Point Clouds and Satellite Imagery. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):  
263 5638–5654, Jun. 2022. doi: 10.1109/TITS.2021.3055366.
- 264 Ali Ozgun Ok, Jan Dirk Wegner, Christian Heipke, Franz Rottensteiner, Uwe Soergel, and Vedat Toprak. Matching of straight line segments  
265 from aerial stereo images of urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 74:133–152, 2012. ISSN 0924-2716. doi:  
266 <https://doi.org/10.1016/j.isprsjprs.2012.09.003>. URL <https://www.sciencedirect.com/science/article/pii/S0924271612001591>.
- 267 B. Qiu et al. WHU-Railway3D: A Diverse Dataset and Benchmark for Railway Point Cloud Semantic Segmentation. *IEEE Transactions on*  
268 *Intelligent Transportation Systems*, pages 1–17, 2024. doi: 10.1109/TITS.2024.3469546.
- 269 H. Ran, Y. Yin, F. Huang, and X. Bao. FLAMNet: A Flexible Line Anchor Mechanism Network for Lane Detection. *IEEE Transactions on Intelligent*  
270 *Transportation Systems*, 24(11):12767–12778, Nov. 2023. doi: 10.1109/TITS.2023.3290991.
- 271 C. Schmid and A. Zisserman. Automatic line matching across views. In *CVPR*, pages 666–671, 1997. doi: 10.1109/CVPR.1997.609397.
- 272 A. Sánchez-Rodríguez, M. Soilán, M. Cabaleiro, and P. Arias. Automated Inspection of Railway Tunnels’ Power Line Using LiDAR Point Clouds.  
273 *Remote Sensing*, 11(21):2567, Nov. 2019. doi: 10.3390/rs11212567.
- 274 T. Tang et al. A Real-Time Method for Railway Track Detection and 3D Fitting Based on Camera and LiDAR Fusion Sensing. *Remote Sensing*, 16  
275 (8):1441, Apr. 2024. doi: 10.3390/rs16081441.
- 276 Wei Wang, Wei Gao, Hainan Cui, and Zhanyi Hu. Reconstruction of lines and planes of urban buildings with angle regularization. *ISPRS Journal*  
277 *of Photogrammetry and Remote Sensing*, 165, 2020.
- 278 Dong Wei, Yi Wan, Yongjun Zhang, Xinyi Liu, Bin Zhang, and Xiqi Wang. Elsr: Efficient line segment reconstruction with planes and  
279 points guidance. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15786–15794, 2022. doi:  
280 10.1109/CVPR52688.2022.01535.
- 281 Y. Weng et al. A Railway Track Extraction Method Based on Improved DeepLabV3+. *Electronics*, 12(16):3500, Aug. 2023. doi: 10.3390/  
282 electronics12163500.
- 283 B. Yang and L. Fang. Automated Extraction of 3-D Railway Tracks from Mobile Laser Scanning Point Clouds. *IEEE Journal of Selected Topics in*  
284 *Applied Earth Observations and Remote Sensing*, 7(12):4750–4761, Dec. 2014. doi: 10.1109/JSTARS.2014.2312378.
- 285 H. Yang, X. Li, Y. Guo, and L. Jia. Discretization–Filtering–Reconstruction: Railway Detection in Images for Navigation of Inspection UAV. *IEEE*  
286 *Transactions on Instrumentation and Measurement*, 71:1–13, 2022a. doi: 10.1109/TIM.2022.3220295.
- 287 S. Yang, G. Yu, Z. Wang, B. Zhou, P. Chen, and Q. Zhang. A Topology Guided Method for Rail-Track Detection. *IEEE Transactions on Vehicular*  
288 *Technology*, 71(2):1426–1438, Feb. 2022b. doi: 10.1109/TVT.2021.3133327.
- 289 C. Yao, L. Yu, Y. Wu, and Y. Jia. Sparse Point Guided 3D Lane Detection. *Proceedings of the IEEE/CVF International Conference on Computer*  
290 *Vision*, pages 8329–8338, 2023. doi: 10.1109/ICCV51070.2023.00768.
- 291 C. Ye et al. Robust Lane Extraction From MLS Point Clouds Towards HD Maps Especially in Curve Road. *IEEE Transactions on Intelligent*  
292 *Transportation Systems*, 23(2):1505–1518, Feb. 2022. doi: 10.1109/TITS.2020.3028033.
- 293 L. Zhang et al. A Deep Learning Based Method for Railway Overhead Wire Reconstruction from Airborne LiDAR Data. *Remote Sensing*, 14(20):  
294 5272, Oct. 2022. doi: 10.3390/rs14205272.
- 295 Y. Zhang, Y. Yang, X. Gao, L. Xu, B. Liu, and X. Liang. Robust Extraction of Multiple-Type Support Positioning Devices in the Catenary  
296 System of Railway Dataset Based on MLS Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–14, 2023. doi:  
297 10.1109/TGRS.2023.3280208.
- 298 Y. Zhang, Y. Zheng, C. Wu, and T. Zhang. Enhanced curve perception in lane detection via adaptive guided techniques. *IEEE Transactions on*  
299 *Vehicular Technology*, 73(10):14450–14461, Oct 2024. doi: 10.1109/TVT.2024.3408162.
- 300 T. Zheng et al. CLRNet: Cross Layer Refinement Network for Lane Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision*  
301 *and Pattern Recognition*, pages 888–897, 2022. doi: 10.1109/CVPR52688.2022.00097.