Proceedings of the Seventh International Conference on
Railway Technology: Research, Development and Maintenance
Edited by: J. Pombo
Civil-Comp Conferences, Volume 15, Paper 8.10
Civil-Comp Press, Edinburgh, United Kingdom, 2026
ISSN: 2753-3239, doi: 10.4203/ccc.15.8.10

3D-to-Plan: Automatic Generation of CAD Sketch of Retaining Walls from Trainborne LiDAR Point Clouds

G. Dufau, J. Sanchez, I. El Ghazaly, G. Neveu and B. Salavati

SNCF Réseau, France

Abstract

LiDAR has been growing in railway maintenance operations and has started to become a staple of reliable geometric analysis of railway scenes. Despite this flourishing interest, few methods are published on automatic high-level analysis.
In this work, we aim at providing a clear pipeline to showcase the use of trainborne LiDAR data for industrial use and continuous maintenance. More specifically, we propose a method to provide geometry-preserving CAD sketch of retaining walls to alleviate the burden on operation workers needing to freehand draw them.
While methods based on photogrammetry or TLS (terrestrial laser scan) exist, they usually require human intervention in the process, whereas in this paper we propose a human-free method from acquisition to result, with a very cost-effective processing time (~15 minutes per retaining walls).

Keywords: LiDAR, retaining walls, artificial intelligence, CAD, point cloud, segmentation

1 Introduction

Retaining walls are crucial engineering works for safety and reliability of the railway infrastructure, whether they serve as supporting bridges or preventing bedrocks from collapsing. On the French railway system, more than 5000 retaining walls are present, with some of them as old as the system, more than a hundred years old. These walls are highly heterogenous, as depicted in Figure 1., especially in their geometry: they can span from a few meters up to a kilometer, they can measure between 50 centimeters and 20 meters, they can be built out of various materials, and their other features (inclination, distance to the tracks, distance to the ground) vary as much. Overall, these differences make it very time-consuming to inspect and maintain. As a result, retaining wails are typically inspected every 5 years, depending on their criticality and their condition.

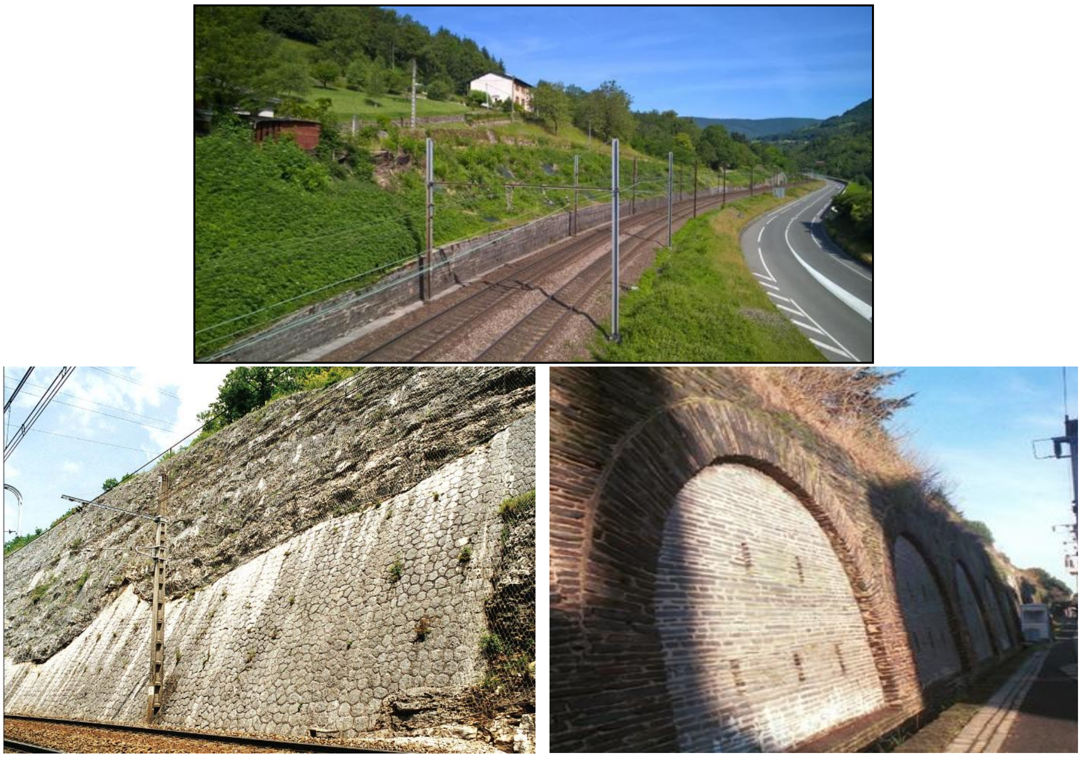

Figure 1. Examples of retaining walls.

Upon inspection, a "radis" (a digital sketch) is produced, corresponding to an outline of the wall from a side view perspective, annotated with all the information relating to the inspection: materials, condition, defaults, vegetation, size, etc. An example of a radis can be found in Figure 2.

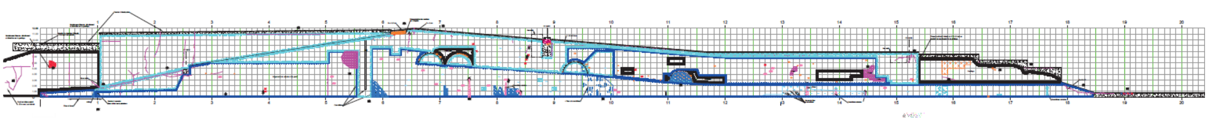

Figure 2. Radis sketch of a retaining wall.

Our contribution is to provide the radis outline automatically, for the maintenance worker to directly annotate a geometry-consistent schema. To achieve this, several methods needed to be developed, and can be summarized as follows:

- Automatic detection of retaining walls

- 3D to 2D projection
- Outline detection from 2D representation

We provide an illustration of the various steps in Figure 3.
Figure 3. The proposed pipeline.
Furthermore, to ensure the usability of this digital sketch, it is computed as a CAD plan, that can be used on existing software. Finally, landmark objects are added to the sketch to provide context, which includes rails and catenary poles.

2 Methods

This section is dedicated to presenting the proposed method in depth, from the input data to the output plan, and to provide the necessary details to understand every step presented in Figure 3. Our data comes from a trainborne LiDAR scanner, specifically a RIEGL VMX-RAIL mounted on an acquisition train, the ESV ("Engin de Surveillance des Voies"). Details about the platform can be found in [1].
The high-resolution point clouds resulting from these acquisitions are the input of our work. These point clouds have a resolution of up to 6000pts/m². The high quality of the data ensures a dense acquisition of retaining walls and a high level of detail, allowing precise segmentation from the bedrocks when present. An example of these point clouds can be found in Figure 4.

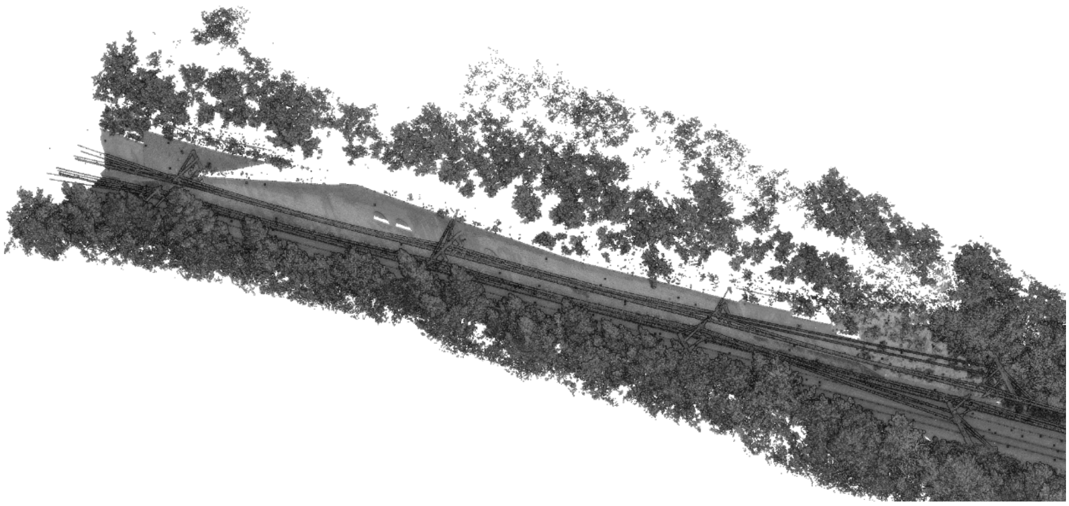

Figure 4. Example of trainborne LiDAR point cloud containing a retaining wall.

Furthermore, to reduce computation time, we leverage existing SNCF databases of the retaining walls to have their approximate localization on the network. This way, we can directly process relevant chunks of the network and have a per-wall approach, rather than processing the full railway network.
Retaining walls tend to be close to the railway tracks, which means the field of view of the LiDAR system can be limited by these structures. In typical scenarios, the proposed method can accurately outline walls up to 12 meters high. For higher

structures, it depends on its inclination and the distance to the tracks, but it can be occluded, as illustrated in Figure 5.

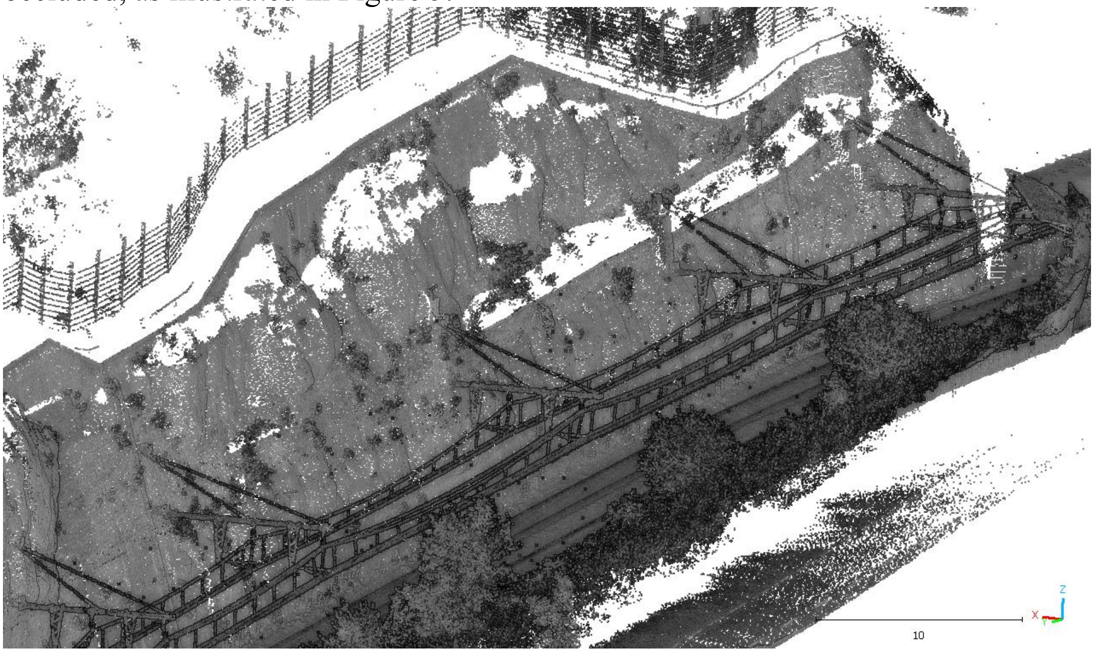

Figure 5. Self-occluded retaining wall.

Typically, the point clouds are extracted to contain the entire retaining wall being processed, with an additional approximate length of 20 meters on each side to make sure we have the full scope of the wall, even in the case of mislocalization, which can happen for tunnel-adjacent retaining walls.

For computation purposes, the ground is systematically removed from point clouds. To achieve that, the DTM (Digital Terrain Model) is computed (with CGAL [relevant reference]), and all points lying near the DTM are removed.

This extracted point cloud goes first through a semantic segmentation pipeline. It requires several steps in order to ensure precise segmentation from the point cloud. The method proposed is machine learning-based and as such relies on the covariance geometric descriptors proposed by [2]. An overall pipeline is presented in Figure 6.

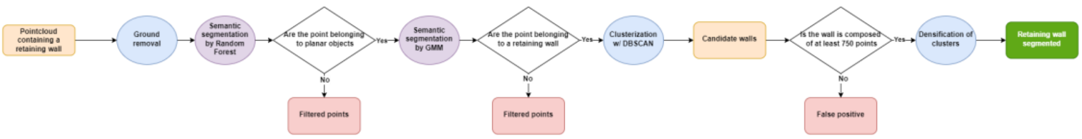

Figure 6. Pipeline of the retaining wall segmentation

To ensure proper featurization we compute: linearity, planarity, sphericity, verticality, omnivariance, surface variation, sum of eigenvalues, anisotropy, and number of neighbors. These features are computed at 3 scales: 15, 25, and 40 cm.

This multi-scale approach is directly inspired by [3], but instead of using XGBoost for semantic segmentation we use a Random Forest. While the Random Forest is discussed as being inferior in previous works, we found it to be suitable for this task as it combines the need for few labelled data and speed of training and inference.

We are doing a binary segmentation: “noise” and “planar structure”. This choice is due to the low expressivity power of the Random Forest combined with the simplicity of this kind of annotation, as it can be hard to properly manually distinguish retaining wall from other planar structures. Planar structure includes several types of objects, mainly: fences, walls, and bedrocks. An example is proposed in Figure 7.

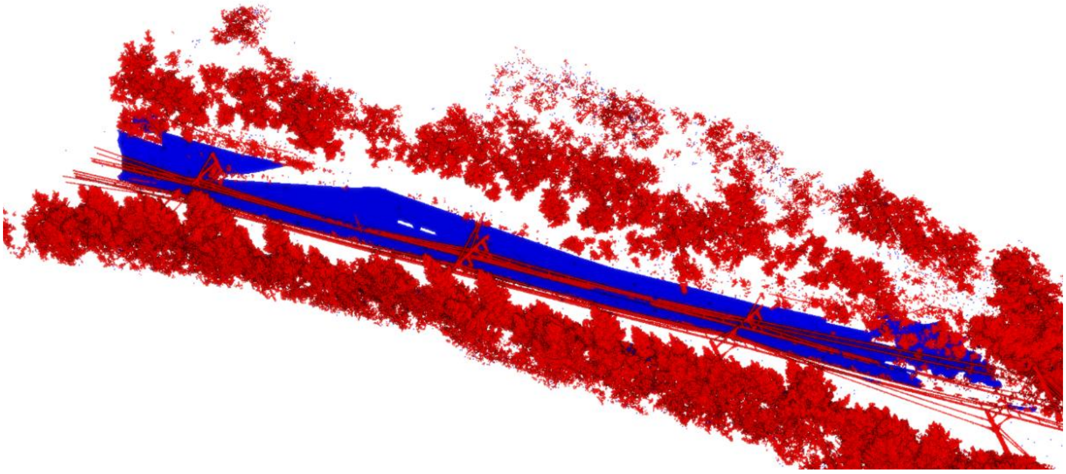

Figure 7. Result of the binary segmentation

To compensate for this simple segmentation, we had a second semi-supervised step to distinguish retaining walls from other planar structures thanks to a GMM (Gaussian Mixture Model). While GMM segmentation is unsupervised, an intermediate step is done to assign the reference cluster of the GMM to the type of object it represents. Specifically, the GMM segmentation works as follows. First, a set of pre-selected retaining walls (about 10) is segmented with the Random Forest, and what is classified as planar structure is used to fit a GMM with 6 classes. Then, these 6 classes are manually assigned to their corresponding labels, i.e. “retaining wall” or “other”. This part consists of the semi-supervised training. Then, at inference time, this fitted GMM is used to segment the planar structures identified by the Random Forest, and only the points assigned to “retaining wall” classes are kept. In the Figure 8. an example of the output.

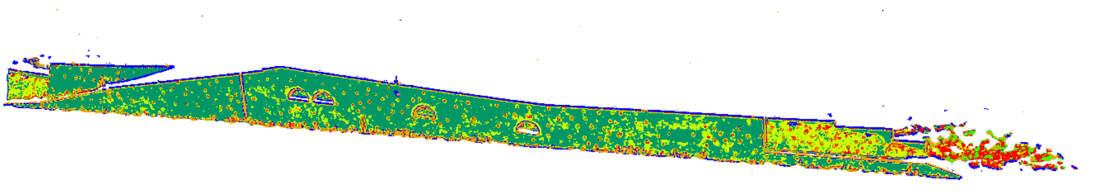

Figure 8. Result of the GMM segmentation, with forest green, lime and blue as retaining wall

In some cases, especially in rocky areas, retaining walls are part of a network of such walls and it is necessary to distinguish between them, as the semantic segmentation

does not permit individually identifying walls. To achieve this, we use DBSCAN on the output of the GMM.
The last step of the wall segmentation is called densification. Due to the cascading segmentation algorithm, and the use of covariance based geometric descriptors, the walls tend to be underestimated. To compensate for this effect, we reintroduce left out points by fetching each point within 20 centimeters of the segmented wall to produce the final output, as presented in Figure 9.

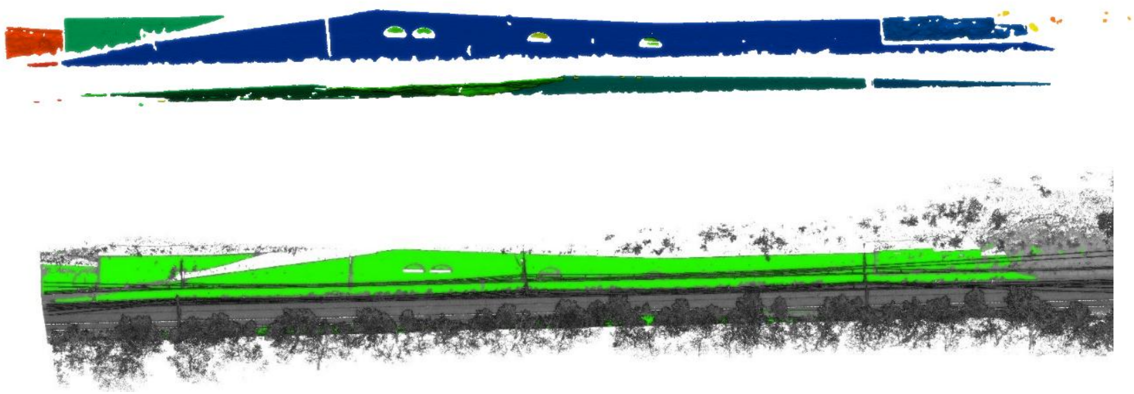

Figure 9. Results from the DBSCAN and subsequent densification.

Once the retaining wall is properly segmented, to reproduce the side view perspective typically used for the outline representation, it is projected to a 2D space.
Let $R = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ be the Euclidean system in which the point cloud is acquired. By convention, we have $\mathbf{z}$ colinear of $\boldsymbol{g}$ the gravity. Let $R_{2D} = (\boldsymbol{x}', \boldsymbol{y}')$ represent the side view of the wall such as that the height of the wall is vertical, meaning $\boldsymbol{y}' = \boldsymbol{z}$. We're looking for $\boldsymbol{x}'$.
Given their use and their material, retaining walls are planar or near-planar structures. Modelling the main plane alongside which are distributed the points of the wall, we can extract $\boldsymbol{n}$ the normal of the plane and define $\boldsymbol{x}''$ as $(\boldsymbol{n} - \langle \boldsymbol{n}, \boldsymbol{z} \rangle) \times \boldsymbol{z}$ creating $R' = (\boldsymbol{x}'', \boldsymbol{z}, \boldsymbol{n} - \langle \boldsymbol{n}, \boldsymbol{z} \rangle)$ an Euclean system with the wall mainly spanning in $(\boldsymbol{x}'', \boldsymbol{z})$. Thus, we can identify $\boldsymbol{x}' = \boldsymbol{x}''$, creating our 2D system representing the side view of the wall, and we have the rotation matrix to transform the wall from R to R_{2D}. The process is illustrated in Figure 10.

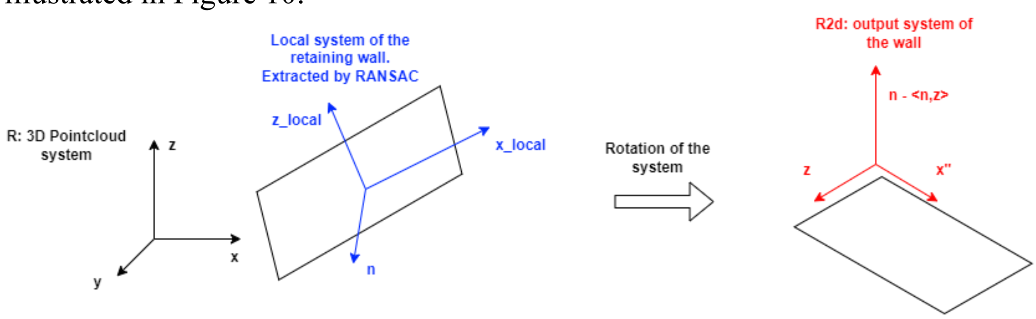

Figure 10. Summary of the 2D system extraction.

Once the wall is rotated into his side-view system, we can rasterize it to create an image of the 2D projection, colorized by its $\boldsymbol{n} - \langle \boldsymbol{n}, \boldsymbol{z} \rangle$ value. Assuming that the wall is alongside the track, this value is an image to the distance to the tracks, as the normal will point towards the rails. An example of the resulting image can be found in Figure 11.

Figure 11. 2D projection of a retaining wall.

Finally, from the image representation of the wall, the outline can be easily extracted with a contour detection algorithm. In this work, we use [4] accordingly tuned. Once the contour is extracted, it is easy to convert it to a CAD-friendly format as the contour is already defined as a $(node, edge)$ ensemble adapted to vectorial representation.

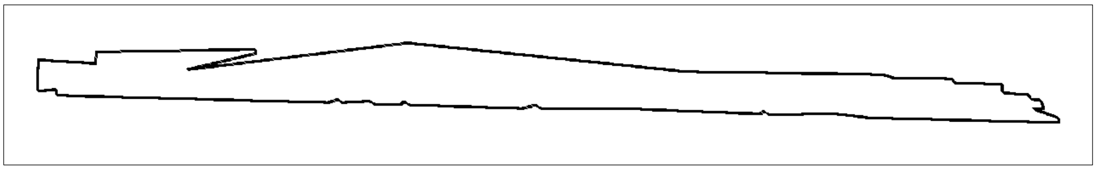

Figure 12. CAD-friendly sketch

3 Results

Due to the nature of the method, it is very hard to provide qualitative evaluation on the segmentation of the retaining walls or on the outline of it. Nonetheless, this method was evaluated in three ways.
First, through exhaustive qualitative evaluation, by manually checking the point clouds. In total 30 walls of various sizes, materials, and locations were used. When possible, the point clouds were compared with priori photographs of the walls to help separate walls from bedrock when needed.

This qualitative analysis resulted in highlighting the limits of our method. The main drawbacks reside in vegetation-covered retaining walls. Due to our binary segmentation, vegetation is removed from the point cloud, but dense vegetation can cause occlusion of the wall and thus make it completely removed during segmentation. This results in unreliable radis sketches, as presented in Figure 13.

Figure 13. Vegetation-covered wall and its output sketch

Secondly, geometric features of the wall extracted from the 3D segmentation were used and compared with the retaining wall database, to ensure that they were consistent, even though both sources could have uncertainties. These features were: geographic coordinates, length, height, surface area, and distance to the ground. Overall, the segmentation over 15 walls achieved up to 98% match with the database, even for walls up to 300 meters long.
Finally, in some cases, prior photogrammetry surveys were used to compare output with our method. While it did not yield quantitative measurements, it highlighted the precision of the output sketch, such as highlighted in Figure 14.

Figure 14. Comparison between photogrammetry survey and our CAD plan. In red manually drawn outline from the photogrammetry, in blue the automatic sketch.

Besides the quality of the results, it is important to know the computational cost to ensure it is viable to deploy it over a large number of structures. We did a full time-analysis of the method, presented in Table 1. It is hard to estimate the cost per wall due to their heterogeneity, so we proposed a time per kilometer. All the processing was done on a laptop with no GPU.

	Length (m)	Time for segmentation (s)	Time for sketch drawing (s)
Wall 1	365	290	200
Wall 2	200	120	80

Table 1. Time-analysis

We can see that compared to human analysis, approximately 4 kilometers per hour, the proposed approach is very fast and can be done prior to any inspection on a local laptop, or run over the entire network in advance.

4 Conclusions and Contributions

In this paper we proposed a novel method to provide automatic CAD-compatible outline of retaining walls. This work is a first step toward automatic surveillance for railway infrastructure. Furthermore, this work can be incorporated into a wider digital twin pipeline.

References

[1] A.-S. Onody, M. Convert, F. Viguier, N. Abdallah, N. Ben Drihem, R. Guerand, J. Sanchez, B. Salavati, "Automated Detection of Vegetation Close to Catenary Wires using LiDAR Data", in J. Pombo, (Editor), "Proceedings of the Sixth International Conference on Railway Technology: Research, Development and Maintenance", Civil-Comp Press, Edinburgh, UK, Online volume: CCC 7, Paper 4.2, 2024, doi:10.4203/ccc.7.4.

[2] Weinmann, M., Jutzi, B., & Mallet, C. (2013). Feature relevance assessment for the semantic interpretation of 3D point cloud data. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2, 313-318.

[3] H. Thomas, F. Goulette, J. -E. Deschaud, B. Marcotegui and Y. LeGall, "Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods," 2018 International Conference on 3D Vision (3DV), Verona, Italy, 2018, pp. 390-398, doi: 10.1109/3DV.2018.00052.

[4] Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32–46, 1985