Segmentation of Railway tracks, Contact cables and Catenary cables from a LiDAR Point Cloud

JEAN LOUIS CLEMENTE, University of Twente, The Netherlands

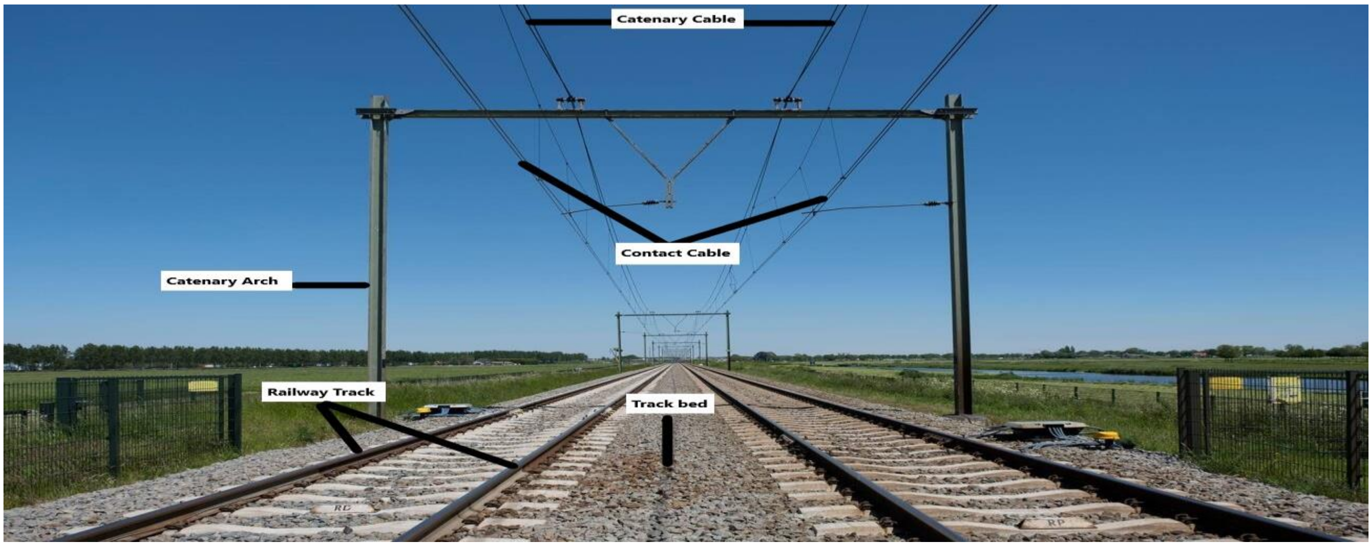


Figure 1 Labelled Railway Track Corridor.

Additional Key Words and Phrases: Segmentation, railway track, contact cable, catenary cable, point cloud data, maintenance, safety

1 Abstract

This study focuses on the segmentation of railway tracks, contact cables and catenary cables from LiDAR data of railway infrastructure. This is achieved by taking advantage of known railway characteristics, using linear algebraic techniques, and applying algorithms such as RANSAC and DBSCAN. The developed algorithm is applied to a few different railway segments from a larger dataset. The findings show that all objects of interest were correctly recognized, however the methodologies applied have their limitations.

2 Introduction

Object detection using LiDAR is a field of growing importance, it is being applied to fields such as robotics, autonomous vehicles, mapping, surveying, forensics, crime scene investigation, environmental monitoring, and infrastructure, just to name a few. This dependence on LiDAR technology is expected to grow dramatically in the coming years as the demand for automated technology continues to grow [3]. LiDAR technology enables the creation of extremely precise and thorough maps of the environment, allowing for detailed object extraction. These extracted objects can be used for many purposes, such as infrastructure monitoring, non-invasive archaeological mapping, elevation mapping of remote places, and much more.

LiDAR technology presents an opportunity to automate railway maintenance and safety analysis. By using LiDAR sensors mounted on inspection vehicles or trains, accurate 3D measurements of infrastructure can be collected. Automated algorithms can then analyze the data to detect defects, track geometry deviations, structural issues, and vegetation encroachment. LiDAR enables predictive maintenance, asset management, and data integration. Implementing LiDAR technology in railway infrastructure monitoring provides an opportunity to improve efficiency, reduce costs, and enhance railway safety.

Railways in the European Union are considered safe, but improvements are needed to become a global leader in railway safety [9]. There are significant variations in safety performance among EU Member States, with over 2,000 accidents occurring each year, costing EUR 1.4 billion and resulting in numerous fatalities and injuries [9]. While collisions and derailments represent a small portion of accidents, incidents involving rolling stock in motion and level crossings constitute the majority. Progress in reducing accidents has been limited, particularly in train collisions and derailments. More effort is needed to enhance infrastructure safety, investigation processes, and occurrence reporting for a comprehensive understanding of risks and better risk management [9].

In this study, our primary objective is to accurately segment railway tracks, contact cables, and catenary cables, as in Fig 1, enabling effective object detection. By achieving precise segmentation, one can establish a benchmark for future investigations, facilitating comparisons to identify any deviations from operational specifications and improve safety measures accordingly. Furthermore, we can use this type of object detection to detect variations in railway standards used and based on this information we can identify the compatibility of railway systems with each other. Object detection plays a vital role in various aspects of railway operations, including obstacle detection, continuous infrastructure monitoring, unauthorized intrusion detection, collision avoidance, and optimizing performance. Implementing these capabilities enhances the overall safety and efficiency of railway systems, ensuring smoother operations and reduced risks to passengers and personnel.

This leads us to the following research question:

"How can the segmentation of railway tracks, contact cables, and catenary cables from LiDAR point cloud data be achieved?"

To address this question effectively, it is required to break down the research question into further sub research questions, provided below:

- "What techniques can be utilized to differentiate railway tracks from other objects in LiDAR data?
- "How can the LiDAR data be processed to accurately extract the contact cables?
- "What methods can be employed to differentiate catenary cables from other objects in LiDAR data?"

To answer the research questions posed above, the approach involves reviewing literature, collecting LiDAR data, developing segmentation algorithms, evaluating the performance of the designed algorithm, refining the developed algorithms, conducting testing, comparing with existing approaches, and providing implementation recommendations. These points will be discussed further in the upcoming sections, beginning with the literature review, dataset, methodology, results, discussions, and future work sections.

3 Literature Review

There have been a lot of studies done when it comes to object extraction from LiDAR data. However, most of these studies focus primarily on a few sectors, such as object detection in autonomous driving, road sign detection, vegetation detection and urban infrastructure detection (buildings, roads, etc.). Zheng et al. [8] implements a vegetation removal algorithm. Engels [10] propose a more robust method towards detecting objects at different distance in regards to autonomous driving. Hui et al. [5] implements a new method of object-based building extraction.

The next studies reviewed are specific towards object detection in railway infrastructure. Damjan [4] proposes a methodology for extracting the railway tracks from LiDAR data using height classification, eigen decomposition and a region growing algorithm. Lou et al. [6] proposes a different approach compared to the previous author for railway track detection using the physical shape of the railway track, geometrical properties, and the reflection intensity feature. Zhang et al. [7] proposes a method of detecting solely the power lines, using a self-adaptive region growing method to detect the power line parallel with the rials. Arastounia [2] proposes a method for both template matching and region growing for detecting rail and wire points based on known railway architecture characteristics. Additionally Arastounia [1] proposes a new faster method for his region growing implementation from his previous work by implementing eigen decomposition and making it more data-driven.

4 Dataset Overview

The datasets used in this research was provided by Strukton Rail, and is a laser point cloud file, which is one of the file formatting standards used during the data collection step. The first dataset contains over 80 million data points. The dataset covers an aera of 16,474,959 m^2 around the Delft central station located in the south of the Netherlands. In addition to the railway infrastructure the dataset also includes buildings, tram lines, roadways, water ways and bicycle paths, just to name a few. The data contains the X, Y, Z coordinates, the scale factor of each coordinate axis, the unscaled x, y, z coordinates, the offsets, the max and min values, and much more. On the other hand, the dataset is missing some crucial information specifically the EPSG codes. The EPSG code also known as the European Petroleum Survey Group codes are a standardized system used to identify and reference coordinate systems and geospatial parameters. Thus, leaving the units of the coordinate open to interpretation, however, most of the LiDAR in the Netherlands is recorded in the RD-coordinate system. Through testing it was identified that this is the case for the dataset being used. Fig 2 below provides an overview of what the data looks like when visualized from a top-down perspective, while Fig 3 provides a cross sectional view of the data that is focused on the railway segment.

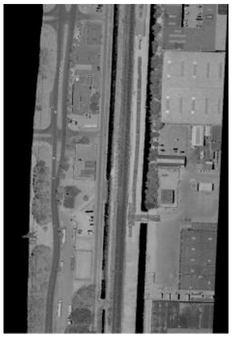


Figure 2 Top-down view of data.

Figure 3 Cross section view of data.

The data in Fig 3 at looks like it is missing data values specifically those representing the catenary archways and overhanging cables, this is true as the data was collected using airborne laser scanning also know as ALS. On further inspection it is identify that the due to the top-down scan and overlapping nature of the cables and catenary arch way top cause the contact cables and catenary arch way mast to be less point dense then the rest of the objects of interest in the dataset. Furthermore, the dataset is initially classified in just two classes ground and unclassified. As the data is passed through the algorithm, the points are reclassified as rail track, contact cable and catenary cable.

The second dataset used is a collection of smaller railway segments, each one of these segments contains approximately about 600,000 datapoints and have an approximate aera of 2500m^2, in total there are 86 of these segments which vary in both shape and size, however, we will only work with a few pre-determined number of data segments. This dataset, like the previous one contains all the relevant information but once again misses out on the EPSG codes.

4 Software's used for Programming and Visualization

4.1 CloudCompare

CloudCompare is used to visualize, segment, transform, etc. the data. In this study CloudCompare is used to segment out a small part of the data set, as well as visualize the results based on point classification.

4.2 Python

Python was the coding language of choice to create the algorithm for the segmentation task, specifically Jupyter notebook was used to run and test the code in small blocks without having to iterate over all the code again during debugging. Libraries such as laspy, numpy, pandas, sklearn, scipy, and plotly were used to implement the necessary algorithms to complete the segmentation task.

5 Methodology

This section covers the processes involved in segmenting the railway tracks, contact cables and catenary cables from a laser point cloud file. Section 5.1 covers steps involved in data preprocessing. Section 5.2 covers steps involved with the seed point selections of each component respectively. Moving forward we will primarily cover the methodologies used for the first dataset.

5.1 Data Preprocessing

Given the original dataset is too large to effectively segment out the identified railway components, the original dataset is segmented into a much smaller dataset, primarily containing a small railway corridor with dimensions of 225 meters in length, 10 meters in width and 9 meters in height. Furthermore, the ground points are filtered out, leaving only the unclassified points to work with. This results in the total number of points being 110,452 which is already a large reduction from 80 million points and should help with computation times. Furthermore, the new data segment is also fitted to the X axis to allow for easier distance calculations in later sections. Fig 4 presents a horizontal view of the new data segment; from this one can clearly see the track bed as a bright blue line while the catenary cables are represented as a light blue line and the contact cables are barely visible. The brightness of the lines is directly proportional to the density of the points in that space. The same hold true for Fig 5 which displays a cross sectional view of the new dataset, here the track bed and rail track points are clearly visible while the overhanging cables are barely noticeable.

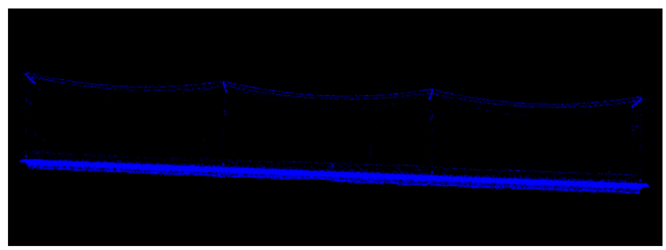


Figure 4 Horizontal view of the new data.

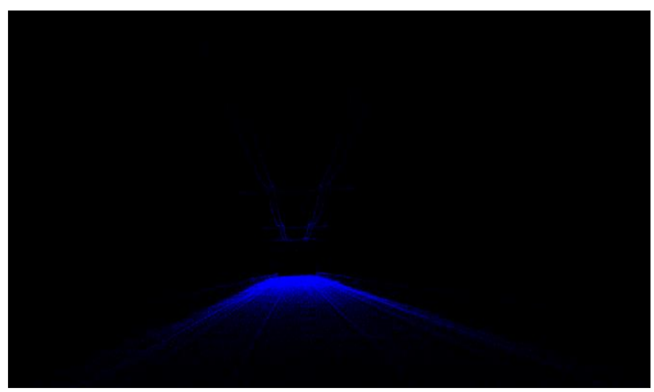


Figure 5 Cross sectional view of the new data.

Now that a rather useable data segment is obtained, the data preprocessing can begin by applying a height based coarse classification on the data points. This is possible as the railway characteristics for the Netherlands is publicly available, and these facts can be used to break the dataset down into four different clusters. Primarily the track bed cluster, contact cable cluster, catenary cable cluster, and another cluster containing points that are no of interest and are thus disregarded.

The track bed cluster includes points belonging to the track bed and rail track, this is obtained by calculating the median height of all points in the data segment. It can be argued that the median height of the all the points is approximately the track bed height since the highest cloud point density occurs on the track bed. Once the median height is obtained, one can create a window to look for points that are either 0.5 meters above or below the median height. This results in the minimum and maximum possible track bed height as in equation 1 and 2 below. All points that fall within this window are reclassified accordingly.

$$\text{Maximum Track Bed height} = \text{median height} + 0.5 \quad (1)$$

$$\text{Minimum Track Bed height} = \text{median height} - 0.5 \quad (2)$$

The contact cable cluster includes points that belong to the contact cable as well as possible points that belong to the catenary arch mast. Now that the value of maximum track bed height is available, one can use their knowledge of the height of the contact cable above the maximum track bed height to identify the contact cable points, in this case the contact cable lies between 6 and 6.7 meters above the maximum track bed height. The catenary cable cluster includes points belonging to the catenary cable, as well points that belong to the catenary arch, using the same concept as earlier it was identified that the catenary cables lie above 6.7 meters above the maximum track bed height

With the height characteristics of each cluster known we can iterate over all the points of the point cloud in the Z axis and classify the points according to their respective cluster. Found below is the associated pseudocode for the coarse classification:

- **Notation:** *LasFile: preprocessed data segment; coH: contact cable height from track bed; caH: catenary cable height from track bed; tbH: track bed height; mitbH: minimum track bed height; matbH: maximum track bed height; tbC: track bed cluster; uC: unwanted cluster; coC: contact cable cluster; caC: catenary cable cluster; micoH: minimum contact cable height relative to track bed; micaH: minimum catenary cable height relative to track bed*
- **Input:** *LasFile, coH and caH*
- *Calculate tbH as the median height of all the points*
- *Set the mitbH and matbH*
- *Set the micoH and micaH*
- For ∀Point ∈ LasFile
    - if point >= mitbH and point < matbH
    - reclassify the point as tbC
    - else if point >= matbH and point < micoH
    - reclassify the point as uC
    - else if point >= micoH and point < micaH
    - reclassify the point as coC
    - else if point >= micaH
    - reclassify the point as caC
- **end for** *loop on point*

5.2 Object detection

5.2.1 Railway Track extraction

To detect the railway track points from the track bed segment obtained in the previous section, Eigen decomposition along with the RANSAC algorithm is utilized.

For each point in the track bed segment a 3D spherical object is created, this object contains the neighbors of the selected point for which a covariance matrix is constructed, through eigen decomposition the eigenvalues and eigenvectors are obtained. For this case we are most concerned with the 3rd eigen value, also known as λ_3, since λ_3 looks at the variance between points in the Z direction. The value of the variance in the Z direction is important, since the rail track sits above the track bed and thus points belonging to the track are most likely to have a high variance when compared with its neighbors.

Through visual testing it was found that a variance threshold of 0.07 was the best fit as it resulted in the least information loss. The λ_3 is then compared against the variance constraint, and points that meet this criterion are then compared with the 90th height percentile within each point neighborhood, if the point neighborhood is larger than the 90th height percentile the point is then labelled as being part of the railway track. These calculations resulted in a point reduction from over 100,00 points to 46,000 points for the railway segment under consideration.

Passing the points labelled as being part of the railway track of into the RANSAC algorithm allow us to find all the points that lie in a line with the highest Z space point density. RANSAC is used to fit lines to the extracted points from the previous step. The algorithm iteratively identifies inliers (points on the detected line) and removes them based on an inlier threshold value. Additionally, we calculate the average y value of each line and find lines that lie 1.435 meters apart from each other as this is the track gauge, with some level of tolerance to account for misclassified points. The railway track points are then reclassified accordingly, thus allowing for multiple rail track detection. Found below is the associated pseudocode for the railway track detection, the first pseudocode is for the eigen decomposition step used to identify candidate railway track points and the second pseudocode is used to detect the railway track lines using RANSAC:

- **Notation:** *tbC: track bed cluster; np: neighboring points; r: radius around which to construct the 3D sphere; ec: the eigen constant threshold; cp: candidate rail track points; cM: covariance matrix; λ3: eigenvalue; hp: 90th height percentile*
- **Input:** *tbC, r, ec*
- **Set** *cp = [ ]*
- **For** ∀Point ∈ tbC

- *Construct np as the 3D spherical point neighborhood using the given r*
- *Calculate cM as the covariance matrix of np*
- *Calculate the λ_3 eigenvalue of pN using PCA on cM*
- **If** λ_3 *>= ec*
    - *Calculate hp of np*
    - **For** ∀point ∈ np
        - ***If*** *point > hp*
        - Add point to cp
    - **End for** on point
- **End if** on λ_3

- **End for** on Point

- **Notation:** cp: candidate rail track points; mlp: minimum points in line; dt: detected lines from RANSAC; rt: railway track points; iT: inlier threshold value; ay: average y value; li, lj: lines detected from RANSAC; tg: track gauge in meters; tol; tolerance value
- **Input:** *cp, mlp, iT and tol*
- **Set** *dt = [ ]*
- **Set** *rt = [ ]*
- *Calculate dt as all possible lines using RANSAC with the iT and mlp values*
- **For** ∀ li ∈ dt
    - **For** ∀ lj∈ dt
        - *Calculate ay of li*
        - *Calculate ay of lj*
        - *If absolute value of (li -lj) < tg – tol and > tg + tol*
            - *Add li and lj to rt*
        - **End if**
    - **End for** *on lj*
- **End for** *on li*

5.2.1 Contact cable extraction

To detect the contact cables, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is used. DBSCAN is a density-based clustering algorithm that groups together data points that are close to each other in terms of a specified distance metric. In this case, the Euclidean distance metric is used. The algorithm takes two parameters: epsilon, which defines the maximum distance between two points for them to be considered neighbors, and minimum number of samples, which specifies the minimum number of points required to form a dense region or cluster. The algorithm assigns labels to each point based on their cluster membership. Points that are not assigned to any cluster are considered noise points. For each unique cluster, the equation of the curve is calculated are cross checked against the parabolic curve equation and if the cluster closely resembles the equation, then it is considered as a contact cable. Found below is the associated pseudocode for the contact cable detection:

- **Notation:** *coC: contact cable cluster; ep: epsilon value; mpC: minimum points to be considered a possible contact cable; cocDb:* contact cables clusters identified by DBSCAN; ce: equation of the curve for comparison; dce: detected curve equation
- **Input:** *coC, ep, mpC*
- **Set** *co = [ ]*
- Calculate the cocDb using the ep and mpC values
- For ∀cluster ∈ cocDb
    - Calculate the dce of cluster
    - If dce = ce
        - Add cluster to co
    - **End if**
- **End for** *on cluster*

6 Results

Using the methodology described in section 5, it was possible to detect both the railway tracks and contact cables. However, there are some limitations that should be noted, which will be addressed in the section 8.

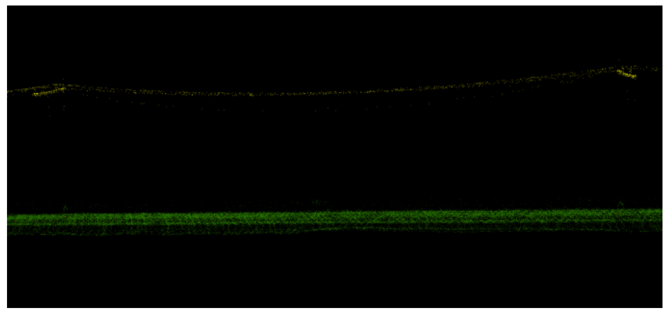


Figure 6 Horizontal view of the applied coarse classification on the dataset

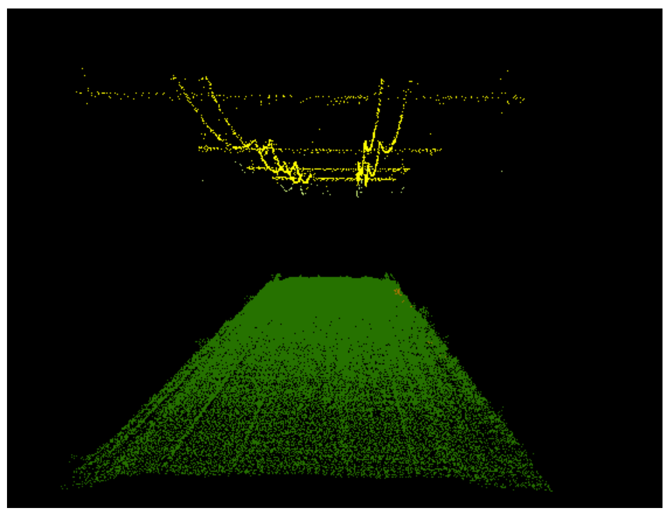


Figure 7 Cross sectional view of the applied coarse classification on the dataset

Figure 6 and Figure 7 visualize the output of the coarse classification step that was used in section 5.1. The bright yellow points belong to the catenary cable cluster, the light-yellow points which are a bit difficult to see belong to the contact cable cluster and the green points belong to the tack bed cluster. The intermediary points between the track bed and the contact cables have been removed as we are not interested in them.

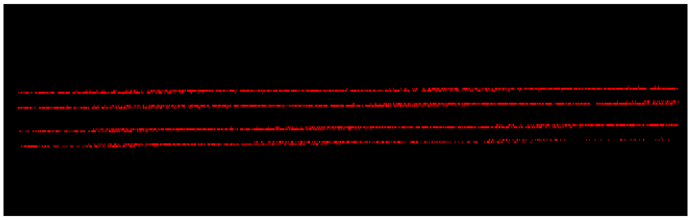


Figure 8 Line detection using RANSAC

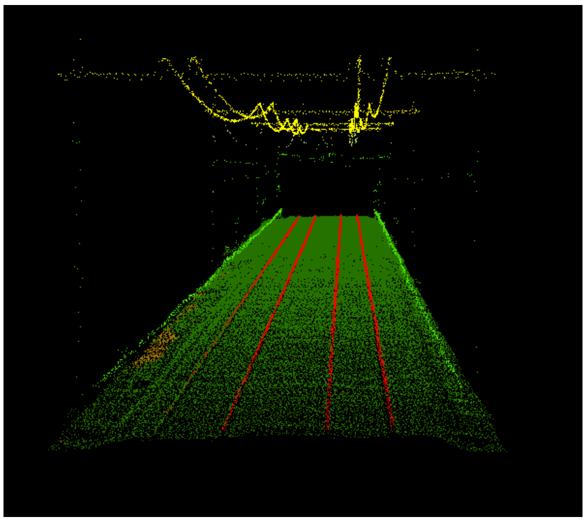


Figure 9 The rail track points overlayed on the existing points

Figure 8 and Figure 9 visualize the output of the Railway Track extraction as described in section 5.2.1. We can see that the RANSAC algorithm was able to successfully detect the railway track points, which are highlighted in red. These points have been overlayed over the point classification obtained from the heigh based coarse classification.

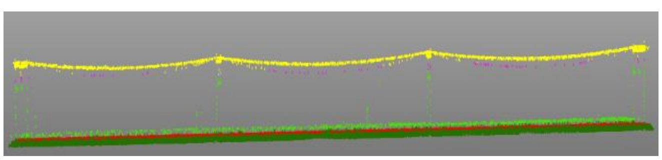


Figure 10 Contact cable detection using DBSCAN highlighted as purple points

Figure 10 visualize the contact cables that have been detected using the DBSCAN algorithm described in section 5.2.2. The purple points indicate the points belonging to the contact cable.

Dataset segments	IOU on track	IOU on contact cable
1 (straight railway corridor)	0.45	0.81
2 (straight railway corridor)	0.52	0.63

Table 1 Results of IOU on each tested segment

Table 1 provides the results of the algorithms object detection compared with a manually segmented version of the same data using the IOU (intersection over union) metric.

7 Discussions

The obtained results are rather difficult to evaluate at an object and point cloud level, due to the lack of a true ground truth value. However, we decide to use IOU (intersection over union), by comparing the results obtained from the algorithm with a manually classified version of the same dataset as seen in Table 1 above.

For the first data segment we see in IOU value of 0.45 for the track and 0.81 for the contact cable. Regarding the track detection we can attribute the low IOU value to the manual classification as well as to the fact that the lines detected mainly detected the center of the railway track itself and not its edges. Regarding the catenary cable we can attribute the much higher result to the fact that the point cloud density in the dataset for the contact cables were low and thus most of the points detected were done successfully.

For the second data segment which had much higher contact cable point density we can start to see a drop in the IOU value to 0.63 which can be attribute to the fact that the cables had a much smaller curve to them and thus there was some misclassification occurring in the detection step, as well as there being a slight curve on y axis for the contact cables .

Based on a visual analysis of the above figures we can see that the results were rather successful in answering two of the three sub research questions that were posed. Referring to Figure 6 and Figure 7 we can see that almost all the points were correctly classified according to their respective clusters. There is a presence of outliers when it comes to separation of the contact cable cluster and catenary cable cluster, since the lowest point on the catenary cable lies at height that is just slightly higher than some of the points that belong to the contact cable cluster and thus, we can say that these points have been misclassified in the coarse classification step.

Additionally in Figure 8 and Figure 9 we can see that some of the points that belong to the railway track were misclassified as being part of the track bed. This could be a result of the elevation in the dataset and scan angle of the LiDAR scanner. Points belonging to the left rail pair, that sat at the lowest elevation points in the dataset were not classified properly.

Regarding the contact cable detection, we already highlighted a few issues in the coarse classification, where some points were classified as being part of the catenary cable segment and thus were not considered when running the curve detection algorithm using DBSCAN and thus only partial curves were detected. Mainly losing information about where the contact cables connect to on the catenary arch ways. Furthermore, we can see the low point density in the contact cable segmentation, that was a result of the ALS equipment that was used to collect the data.

8 Future work

For future research we must considered a few limitations of the dataset regarding the contact cables, and methodologies used for rail track detection. As well as propose a method for the segmentation of the catenary cables based on some of the literature review done earlier. Since this was not successfully implemented in this study and thus was left out of the methodology section.

8.1 Dataset Limitations and Segmentation Challenges

The dataset used for the segmentation of LiDAR data for railway infrastructure had several limitations that posed challenges for accurate segmentation. Firstly, the dataset lacked thorough documentation, which made it difficult to gather comprehensive information about the data. In particular, the absence of metadata, such as EPSG codes, made it challenging to properly interpret and analyze the spatial coordinates of the LiDAR points.

Furthermore, the dataset presented challenges in segmenting the contact cables due to the scan angle. The scan angle of the LiDAR sensors resulted in a lack of sufficient points for effectively identifying and segmenting the contact cables. This limitation impacted the accuracy and completeness of the segmentation results, as the insufficient data hindered the ability to precisely delineate the contact cables in the LiDAR point cloud.

These dataset limitations highlight the importance of thorough documentation and comprehensive metadata for LiDAR datasets, especially in the context of railway infrastructure segmentation. Access to accurate and complete metadata, including EPSG codes, is crucial for correctly interpreting the spatial information and ensuring the accurate segmentation of railway infrastructure components, such as contact cables. Additionally, careful consideration of the scan angle during data collection is essential to capture sufficient points for effective segmentation and ensure accurate results for railway infrastructure analysis and monitoring.

8.2 Railway Track Detection Limitations and the Need for Robust Methods

The detection of railway tracks in the data also faced certain limitations that necessitate the development of more robust methods to enhance accuracy and reliability. One significant limitation pertains to the challenges posed by elevation changes and curves along the railway track. These elevation changes can introduce complexities in accurately detecting and segmenting railway tracks.

To address this limitation, it is essential to develop a more advanced data driven algorithm and technique that can effectively handle elevation changes as well as curved data segments. These methods should account for variations in the height of the railway track and adjust the detection process accordingly. By incorporating elevation information and considering the unique characteristics of railway infrastructure, such as track slopes and inclines, more accurate and reliable detection results can be obtained.

To address the limitation brought about by segmenting curved railway track segments we need to develop the algorithm further to get more conclusive results. The main challenge faced in this limitation is correctly identify the railway track points from the RANSAC algorithm, we would need to first identify some metric or constraint value that can help narrow down the number of possible railway track lines output by RANSAC.

8.3 Proposed methodology for the segmentation of the Catenary Cables

Based on the literature review done at the start of this study, one possible method of detecting the catenary cable would be to use a region growing algorithm based on the detected railway tracks and using certain criteria such as point density, vertical height variation, or geometric properties of the points within the growing region. Implementing this based on the methodologies proposed by Arastounia [1], should allow for accurate segmentation of the catenary cables.

9 Conclusion

This study can detect the railway tacks and contact cables with a certain level of confidence. Using eigen decomposition and the RANSAC algorithm the railway track was successfully segmented. Using the DBSCAN algorithm and curve fitting the contact cable were also successfully extracted. However, this study failed in the extraction of the catenary cables, but propose a methodology in section 8, that provides a small bit of insight on how the methods implemented in this study can be grown to incorporate this feature as well.

“How can the segmentation of railway tracks, contact cables, and catenary cables from LiDAR point cloud data be achieved?”

- “What techniques can be utilized to differentiate railway tracks from other objects in LiDAR data?
- “How can the LiDAR data be processed to accurately extract the contact cables?
- “What methods can be employed to differentiate catenary cables from other objects in LiDAR data?”

With regards to the research questions above we can conclude that we were able to successfully achieve two of the three research questions that have been posed, with a high level of confidence. The third sub research question was not able to be answered due to the approach used for the other two research questions and a proposed methodology can be found in section 8.3 Through visual analysis we were able to get a good amount of data segmented correctly.

REFERENCES

[1] Mostafa Arastounia. 2017. An Enhanced Algorithm for Concurrent Recognition of Rail Tracks and Power Cables from Terrestrial and Airborne LiDAR Point Clouds. *Infrastructures* 2, 2 (June 2017), 8. DOI:https://doi.org/10.3390/infrastructures2020008

[2] Mostafa Arastounia. AUTOMATIC CLASSIFICATION OF LIDAR POINT CLOUDS IN A RAILWAY ENVIRONMENT.

[3] Robert Bogue. 2022. The growing importance of lidar technology. *Industrial Robot: the international journal of robotics research and application* 49, 6 (January 2022), 1025–1031. DOI:https://doi.org/10.1108/IR-05-2022-0138

[4] Máté Cserép and Adalbert Demján. Object extraction of rail track from VLS LiDAR data.

[5] Zhenyang Hui, Zhuoxuan Li, Penggen Cheng, Yao Yevenyo Ziggah, and JunLin Fan. 2021. Building Extraction from Airborne LiDAR Data Based on Multi-Constraints Graph Segmentation. *Remote Sensing* 13, 18 (January 2021), 3766. DOI:https://doi.org/10.3390/rs13183766

[6] Yidong Lou, Tian Zhang, Jian Tang, Weiwei Song, Yi Zhang, and Liang Chen. 2018. A Fast Algorithm for Rail Extraction Using Mobile Laser Scanning Data. *Remote Sensing* 10, 12 (December 2018), 1998. DOI:https://doi.org/10.3390/rs10121998

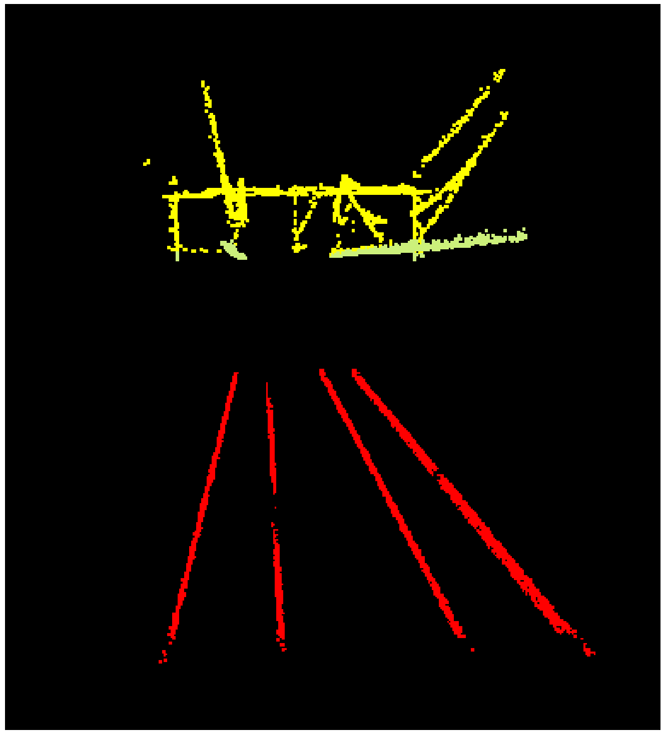
[7] Shanxin Zhang, Cheng Wang, Zhuang Yang, Yiping Chen, and Jonathan Li. 2016. AUTOMATIC RAILWAY POWER LINE EXTRACTION USING MOBILE LASER SCANNING DATA. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLI-B5, (June 2016), 615–619. DOI:https://doi.org/10.5194/isprsarchives-XLI-B5-615-2016

[8] Han Zheng, Feitong Tan, and Ruisheng Wang. 2016. POLE-LIKE OBJECT EXTRACTION FROM MOBILE LIDAR DATA. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLI-B1, (June 2016), 729–734. DOI:https://doi.org/10.5194/isprs-archives-XLI-B1-729-2016

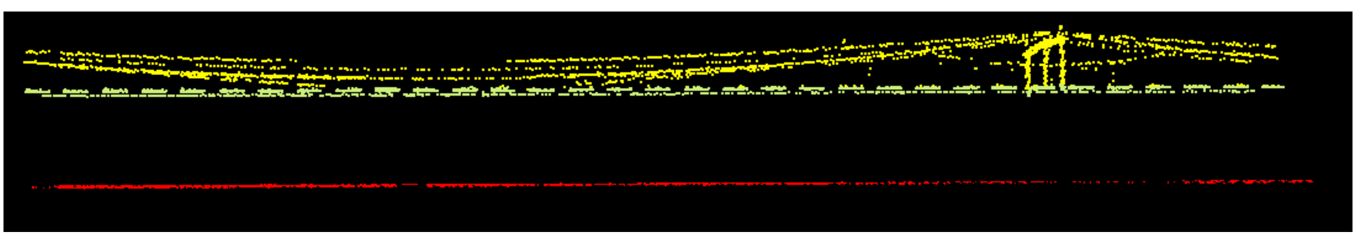
[9] 2018. Analysis and Monitoring | European Union Agency for Railways. Retrieved July 2, 2023 from https://www.era.europa.eu/domains/analysis-and-monitoring_en

[10] 3D Object Detection From LiDAR data using Distance Depended Feature Extraction. *Eindhoven University of Technology research portal*. Retrieved July 2, 2023 from https://research.tue.nl/en/studentTheses/3d-object-detection-from-lidar-data-using-distance-depended-featu

APPENDIX



The above image is the result of the algorithm on the second data segment from a cross sectional view.



The above image is the result of the algorithm on the second data segment from a horizontal view.