Politecnico di Torino

Master's degree of georesource and geoenergy engineering
(Mining Track)
A.a. 2025/2026
September

Politecnico di Torino
Department of Environment, Land and Infrastructure Engineering

DIATI

An Automated Geometric-Topology Pipeline for Railway Centerline Extraction from Dense MLS Point Clouds

Thesis Supervisor:
Prof. Marilena Cardu

Candidate:
Arash Farahmand



Politecnico
di Torino



Politecnico di Torino

Table of Contents

1. Introduction 9
1.1. Clear Problem Statement 9
1.2. Research Question 10
1.3. Contributions 11
2. Related Work 12
2.1. Sensor Fusion 12
2.2. Mobile Mapping Systems (MMS) in Tunnel Inspection 12
2.3. LiDAR Applications in Underground and Mining 14
2.3.1. Rock Mass Quality Assessment 14
2.3.2. Excavation Quality and Blasting Control 15
2.3.3. Deformation and Displacement Monitoring 16
2.3.4. Mechanical Scaling Evaluation 16
2.3.5. Digital Twinning and Semantic Modeling: 17
2.4. Automated Railway Track, Centerline Axis Extraction Methodologies and Their Applications 18
2.4.1. Methodologies 18
2.4.2. Application 23
3. Integrated Mobile Mapping and Sensor Fusion Architecture 24
3.1. Brief Technical Specifications of The Mobile Mapping System (MMS) 24
3.2. Data Typologies (ASC/LAS, Coordinate System, Points Accuracy) 24
4. Experimental Setup 25
4.1. Dataset Description 25
4.2. Hardware and Software Stack 27
5. Proposed Methodology 29
5.1. Challenges in Raw Data Processing 29
5.2. Coarse Detection (Digital Terrain Modeling) 30
5.3. Fine Detection via Dimensionality Features 32
5.4. Geometric Filtering 33
5.5. Spatial Topology Filtering 35

5.6. Generative Centerline, Spherical Snapping, and DQI Formulation 38
5.7. Scientific Contributions of The Proposed Methodology 42
6. Statistical Analysis and KPIs 45
6.1. Semantic Detection Accuracy Analysis 45
6.2. Geometric and Topological Continuity 47
6.3. Data Quality Indicator (DQI) and Statistical Validation 48
7. Results 49
7.1. Parameters Sensitivity Test Results 50
7.2. Rail Extraction Result 59
7.3. Generative Raw Centerline Reconstruction Results 67
7.4. Statistical Analysis and KPIs Results 68
7.5. Computational Analysis 71
8. Discussion 73
8.1. Interpretation of Extraction Performance and Geometric Fidelity 73
8.2. Comparison with Other Methods 74
8.3. Robustness and Parameter Dependence 76
8.4. Computational Scalability and Feasibility 76
8.5. Engineering Interpretation of the DQI 78
8.6. Applicability to Metro Tunnels and Mining 79
8.7. Limitations 80
9. Conclusion and Future Work 80
9.1. Conclusion 80
9.2. Answers to the Research Questions 81
9.3. Future Trajectories 82

Table 1. Different sensors used in tunnel inspections. 12
Table 2. Tunnels description 27
Table 3. Points classes label ID 27
Table 4. Software environments and programmatic libraries utilized in the study 28
Table 5. Pipeline all Variables. 41
Table 6: Dataset point classification 49

Table 7. Points and Process time after geometric filtering 53
Table 8. Points and Process time after ROR and DBSCAN for 1 valid twin data 57
Table 9. Points and Process time after ROR and DBSCAN for 40 valid twin data 58
Table 10. Circular tunnel sensitivity test 58
Table 11. DTM filtered points and Process time. 59
Table 12. Points and Process time after PCA linearity application 62
Table 13 Semantic Detection results in PCA 62
Table *14.Points and Process time after geometric filtering* 63
Table 15. Semantic Detection results in geometric filtering 63
Table 16. Points and Process time after topology filtering..... 64
Table 17. Semantic Detection results after ROR&DBSCAN 64
Table 18. KPI performance for manual rail classification scenarios in circular tunnel 66
Table 19. Points and Process time after mid-points generation..... 67
Table 20. Geometric KPI 70
Table 21. Pearson correlation and DQI table 71
Table 22. Datasets computational timing 72
Table 23. Methodology comparison 74
Table 24. Methodologies contribution and limitation 75
Table 25. Answers to research questions..... 81

Figure 1. WSP rail Mobile Mapping System..... 13
Figure 2 (a). LiDAR scanning: (b) two joint sets identified from LiDAR model (third set facing away from viewer); (c) wedge geometry formed by three identified planes; and (d) Stereonet and wedge models from all unstable combinations of five joint sets. 15
Figure 3. Processing: (a) raw point cloud, (b) triangular mesh, and (c) meshed tunnel model, Sandvik site. (d) Radial difference map showing final shotcrete profile compared to a cylindrical design profile (contoured on the rock–shotcrete model or on the cylinder) 16
Figure 4. Example of instability potential for Pugwash tunnel scaled surface. 17
Figure 5. Parametric model definition of a pair of two rails 20
Figure 6. Schematic workflow from rail detection to data-driven centerline calculation proposed by Elberink, S.O.; Khoshelham 20
Figure 7 orientation of the VUX-1HA22 sensors and angles between scan lines (Source: Riegl) 22
Figure 8. a) Horseshoe section b) quasi-rectangular C) Circular 26
Figure 9. Tunnel segments..... 26
Figure 10. Centerline axis generation pipeline 29
Figure 11.DTM filtration on the left, the accepted grid cells; on the right, the grid cells are rejected because they contain more than 30 points. 31

Figure 12. An example of linearity score and eigenvalues 33
Figure 13. Showing all neighboring points P_2 within the search radius of target P_1, displaying their primary direction vectors 34
Figure 14. Simultaneously applies Distance, Parallelism (>0.95), and Perpendicularity (<0.15) constraints. Only true twin rail points survive to contribute to the density threshold. 35
Figure 15. An Example of remaining ghost points 36
Figure 16 Radius Outlier Removal 37
Figure 17. DBSCAN application 38
Figure 18. An Example of generated mid-points from left and right rails 39
Figure 19. Initial point selected after voxel downsampling 39
Figure 20. An example of centerline generation while calculating the confidence score. 40
Figure 21. Circular Tunnel left the blue points dataset classified rail, right gray points manual classified rail points. 46
Figure 22. Horseshoe classified points. 50
Figure 23. Quasi-rectangle classified points. 50
Figure 24. Circular classified points 50
Figure 25. Step 1 DTM, $1m^2$ grid size analysis. 51
Figure 26. PCA 1m radius linearity scores Quasi-rectangle, Horseshoe, Circular 51
Figure 27. PCA 0.5 m radius linearity scores Quasi-rectangle, Horseshoe, Circular 52
Figure 28. PCA 0.2 m radius linearity scores Quasi-rectangle, Horseshoe, Circular 52
Figure 29. PCA 0.1 m radius linearity scores Quasi-rectangle, Horseshoe, Circular 53
Figure 30. PCA 0.05 m radius linearity scores Quasi-rectangle, Horseshoe, Circular 53
Figure 31. Geometric filtering with 100 valid twins results for Quasi-rectangle, Horseshoe, Circular 54
Figure 32. Geometric filtering with 40 valid twins results for Quasi-rectangle, Horseshoe, Circular 54
Figure 33. Geometric filtering with 1 valid twin results for Quasi-rectangle, Horseshoe, Circular 55
Figure 34. Circular ROR filter 60 and 150 55
Figure 35. Quasi-rectangle ROR filter 60 and 150 56
Figure 36. Circular DBSCAN results Top left: ROR 60 Cluster 1000, Top right: ROR 60 Cluster 3000, Down left: ROR150 Cluster3000, Down Right ROR150 Cluster 1000 56
Figure 37. Top left: ROR 60 Cluster 1000, Top right: ROR 60 Cluster 3000, Down left: ROR150 Cluster3000, Down Right ROR150 Cluster 1000 57
Figure 38. Horseshoe After DTM filtration (Blue points are rail classified points in dataset) ... 59
Figure 39. Quasi-rectangle After DTM filtration (Blue points are rail classified points in dataset) 59
Figure 40. Circular After DTM filtration (Blue points are rail classified points in dataset) 60
Figure 41. Circular Tunnel after PCA application 60

Figure 42. Quasi-rectangle Tunnel after PCA application 61
Figure 43. Horseshoe Tunnel after PCA application. 61
Figure 44. Horseshoe, Quasi-rectangle, circular tunnels after geometric filtration 63
Figure 45. Horseshoe, Quasi-rectangle, circular tunnels after topology filtration 64
Figure 46. PCA step output without Voxel downsampling 65
Figure 47. Geometric filtering step output without Voxel downsampling 65
Figure 48. Topology filtering step output without Voxel downsampling 66
Figure 49. Raw centerline generated from mean of the rail pairs showing in yellow. 67
Figure 50. Horseshoe-generated centerline with DQI score embedded as scalar field. 68
Figure 51. Quasi-rectangle generated centerline with DQI score embedded as scalar field. . 68
Figure 52. Circular generated centerline with DQI score embedded as scalar field KPI result. 68

Abstract

In railway tunnel inspection, Mobile Mapping Systems (MMS) play a key role in creating a dense 3D model of the tunnel. Extracting accurate railway centerlines from raw point clouds remains challenging due to point cloud density, noise, and environmental obstacles. This thesis presents an automated pipeline for railway centerline extraction from MMS point clouds using a combination of Digital Terrain Modeling (DTM), Principal Component Analysis (PCA), geometric and topology constraints, density-based filtering, and a Spherical Snap smoothing algorithm. Unlike conventional approaches based on RANSAC line fitting, the proposed methodology exploits local geometric dimensionality and parallelism in rail topology to improve robustness while reducing computational complexity. The extracted centerline is further enhanced through an embedded Data Quality Indicator (DQI) that provides point-level confidence estimates based on local point cloud density. The proposed method was evaluated using the publicly available Subway Tunnel Semantic Segmentation Dataset (STSD). Experimental results demonstrate accurate rail extraction with complete removal of false-positive rail detections after the topology filtering stage, while maintaining continuous centerline generation throughout the tunnel section. The proposed DQI also showed a statistically significant positive correlation with local ground-truth rail density, demonstrating its effectiveness as an indicator of spatial reliability. The pipeline provides an efficient, explainable solution for automated railway mapping and lays the foundation for digital twin generation and structural health monitoring in underground infrastructure.

This thesis is organized into nine chapters.

Chapter 1 introduces the research background, defines the problem addressed in this study, presents the research questions and objectives, highlights the main contributions, and outlines the overall organization of the thesis.

Chapter 2 reviews the state of the art in tunnel inspection and railway mapping. It discusses sensor fusion technologies, Mobile Mapping Systems (MMS), LiDAR applications in underground environments, and existing railway centerline extraction methodologies .

Chapter 3 describes the integrated Mobile Mapping System used for data acquisition. It presents the hardware configuration, sensor specifications, point cloud characteristics, coordinate systems, and data formats employed throughout this research.

Chapter 4 presents the experimental setup, including the STSD dataset, hardware and software environment, implementation details, and evaluation methodology used to assess the proposed approach.

Chapter 5 details the proposed centerline extraction methodology. The complete processing pipeline is described, including Digital Terrain Model (DTM)-based coarse detection, PCA-based linear feature extraction, geometric filtering, topology filtering, generative centerline extraction, Spherical Snap smoothing, and the formulation of the embedded Data Quality Indicator (DQI).

Chapter 6 introduces the quantitative evaluation framework. Key Performance Indicators (KPIs) are defined to assess semantic detection accuracy, geometric continuity, computational performance, and the effectiveness of the proposed DQI confidence metric.

Chapter 7 presents the experimental results. The empirical outcomes of each processing stage are analyzed, including parameter sensitivity analysis, semantic rail extraction accuracy, centerline reconstruction fidelity, DQI distributions, and measured computational execution performance.

Chapter 8 provides a comprehensive discussion of the findings. It interprets the extraction performance, benchmarks result against existing literature, analyzes algorithmic robustness and parameter dependence, examines computational scalability and multi-kilometer feasibility, evaluates the engineering utility of the DQI, discusses

applicability to metro tunnels and mining drifts, and details the current limitations of the framework.

Chapter 9 concludes the thesis by summarizing the principal findings, providing explicit answers to the research questions, and outlining future research trajectories, including automated parameter optimization, switch/turnout detection, and Scan-to-BIM digital twin integration.

1. Introduction

1.1. Clear Problem Statement

Underground space constitutes a vital component of urban infrastructure development and resource utilization. Among its diverse forms, tunnels are among the most characteristic and significant spatial structures, playing an irreplaceable role in urban rail transit, municipal utility tunnels, water conservancy, and mining resource extraction. Typically situated at considerable depths, these complex structural systems are expected to sustain decades or even centuries of intensive operation [1].

However, maintaining these structures poses a massive inspection challenge. Any structural deformation, leakage, settlement, cracking, or equipment aging can readily trigger catastrophic operational failures and substantial economic losses. Traditional tunnel inspections rely heavily on manual efforts by human inspectors, a conventional approach that is severely limited in efficiency, safety, and accuracy. While automated, cost-effective, and exhaustive inspection practices are desperately needed to decrease infrastructure downtime and improve working conditions, a significant knowledge gap remains in automated infrastructure assessment and maintenance [1, 2, 3].

To bridge this gap, recent technological advancements in computer vision, AI, and robotics have enabled the transition toward automated inspection procedures. In particular, the introduction of Mobile Mapping Systems (MMS) and mobile robotic platforms equipped with geomatics sensors—such as visible/infrared cameras, Inertial Measurement Units (IMUs), and Light Detection and Ranging (LiDAR) sensors—has revolutionized data acquisition. By integrating Mobile Laser Scanning (MLS) systems directly onto rail vehicles or diagnostic carts, it is now possible to capture massive, high-resolution 3D point clouds at operational speeds, enabling remote office-based inspections and the creation of digital twins [2, 3, 4].

Within these railway tunnel environments, a critical prerequisite for structural health assessment, deformation monitoring, and parametric 3D reconstruction is precise semantic segmentation and geometric extraction, specifically, determining the exact 3D railway tunnel centerline or track axis [1, 4].

Despite the advancements in MLS data collection, a major automation gap exists because extracting the exact 3D railway centerline from raw, unorganized point clouds presents severe computational challenges. Raw datasets are characterized by extreme density variations—dropping from 5,000 points per square meter on the driving track to less than 40 points per square meter on adjacent tracks. Furthermore, severe self-occlusion by the rail head limits the line of sight on parallel tracks, while high-frequency noise from ballast beds, overhead catenary lines, and auxiliary equipment (such as power cables and rock bolts) distorts standard geometric descriptors, making robust automated extraction exceptionally difficult [4].

To address these limitations, this thesis proposes an automated data-driven pipeline that systematically processes raw MLS point clouds through a multi-stage extraction and clearance framework. The solution initiates with a coarse 10th-percentile Digital Terrain Model (DTM) height histogram grid filter to isolate structural rail candidates, which are subsequently refined via vectorized Principal Component Analysis (PCA) to extract highly linear track points based on neighborhood covariance descriptors. To eliminate false-positive geometric noise, the pipeline applies a strict spatial filtering and topology check—enforcing standard gauge distances, vector parallelism, and track perpendicularity— coupled with Radius Outlier Removal (ROR) and Density-Based Spatial Clustering (DBSCAN) to promote continuous rail geometry and reduce fragmented detections. Finally, a generative mathematical framework generates raw track midpoints, which are dynamically pulled into a smooth, robust interval path using a custom Spherical Snap Smoothing algorithm driven by local spatial medians. This pipeline simultaneously computes a Data Quality Indicator (DQI) confidence metric based on optimal local track densities, bridging the gap between raw unstructured tunnel scans and structured outputs suitable for digital twin workflows.

1.2. Research Question

How can raw, unstructured Mobile Laser Scanning (MLS) datasets characterized by severe density variations and high-frequency auxiliary noise be automatically segmented to isolate standardized track infrastructure?

How effectively can a spatial topology framework based on local dimensionality and known railway geometric constraints support robust rail extraction as an alternative to RANSAC-based line fitting?

How can generative 3D midpoint data be smoothed along highly curved tunnel alignments without losing geometric precision or distorting the physical path of the track?

In what way can Data Quality Indicator (DQI) and local density metrics be embedded within the extracted spatial coordinates to provide an automated indicator of relative local data support and confidence, while identifying the requirements for future engineering-grade validation?

1.3. Contributions

The primary contributions of this thesis toward automating subterranean infrastructure diagnostics and overcoming the spatial constraints of GPS-denied environments are summarized as follows:

Development of a Fully Automated 6-Step Point Cloud Processing Pipeline: Designed and implemented an end-to-end data pipeline in Python that processes unstructured raw Mobile Laser Scanning (MLS) or point cloud datasets and outputs 3D rail and railway track axes without requiring full human intervention.

Spatial Topology and Median-Based Smoothing Framework: In contrast to global or iterative model-fitting paradigms such as RANSAC line-fitting, this framework adopts a localized geometric verification system (standard gauge, parallelism, and perpendicularity checks) coupled with a Spherical Snap smoothing algorithm based on local spatial medians, designed to trace continuous alignments across curved subterranean geometries.

Embedded Self-Validation and Data Quality Indicator (DQI) Metrics: Formulated a continuous Data Quality Indicator (DQI) scoring mechanism directly embedded into the generated centerline points. By leveraging local track density benchmarks, the pipeline outputs an explicit, intensity-scaled confidence matrix that provides structural engineers with deterministic, explainable data verification.

2. Related Work

2.1. Sensor Fusion

Modern subterranean diagnostics rely on multi-sensor payloads to capture both surface geometry and subsurface structural conditions. A single scanning modality is rarely sufficient to evaluate the integrity of historic masonry or concrete segmental linings. Consequently, mobile mapping configurations integrate high-speed profiling LiDAR with imaging, thermographic, and electromagnetic sensors. Table 1 summarizes the structural roles of the different sensor modalities employed in advanced tunnel inspections [1,2,5,6].

Table 1. Different sensors used in tunnel inspections.

Sensor Category	Equipment Type	Functional Role
Geometric Survey	LiDAR Sensor	3D Point Cloud and Trajectory Capture
Structural Imaging	High-Resolution Linear Cameras	High-definition continuous "unfolded" images
Subsurface Analysis	Ground Penetrating Radar (GPR)	Detection of lining thickness, voids, and moisture
Thermal Diagnostics	Infrared Thermal Cameras	Mapping of temperature gradients to detect leaks
Positioning	GNSS / GPS Antennas	Global spatial referencing in open-sky zones
Inertial Reference	IMU (Inertial Measurement Unit)	Trajectory tracking during signal outages
Spatial Synchronization	Odometer	Longitudinal distancing and image triggering

2.2. Mobile Mapping Systems (MMS) in Tunnel Inspection

Thanks to the widespread availability of low-cost sensors, advances in computational resources, and the maturity of mapping algorithms, many industries have begun exploring the possibilities of automating and improving the efficiency of monitoring systems. [3]

Around the world, companies have developed mobile mapping systems using various sensors. Today, such companies install geomatics sensors on vehicles or drones to build mobile mapping systems that automatically collect various types of data about the surrounding environment. The mobile mapping solution Leica Pegasus: Two is among the most widely adopted mobile mapping systems and uses laser scanners, GNSS receivers, an IMU, a DMI, and cameras mounted on the vehicle to acquire accurate spatial data [3].

The recently updated version of Pegasus: Two (Pegasus: Two Ultimate) removes the need for six-camera stitching by incorporating two back-to-back cameras, creating a 24 MP 360-degree image calibrated to the LiDAR profiler data. The Trimble MX9 system combines long-range LiDAR, spherical imagery, and oblique cameras, along with inertial sensors, to capture dense point clouds and both panoramic and multiangle images. For tunnel infrastructure inspections, WSP has developed a system comprising six LiDAR scanners, two panoramic cameras, nine high-resolution IR cameras, and IR flashes to illuminate the tunnel lining (Figure 1). This technology is periodically adopted for tunnel monitoring and maintenance in combination with GeoTracker software. Moreover, the potential of deep learning techniques for damage detection with this mobile mapping equipment was investigated [3,8].



Figure 1. WSP rail Mobile Mapping System

2.3. LiDAR Applications in Underground and Mining

The adoption of LiDAR in the geotechnical community has significantly expanded the capabilities of rock mass characterization, structural monitoring, and excavation quality assessment. Traditional manual engineering surveys in subterranean mining and tunneling environments are slow, subjective, and prone to human bias. Kinematic and Terrestrial Laser Scanning (TLS) offer high-efficiency, non-contact alternatives that provide dense, millimeter-precision 3D representations of the underground environment. [9]

The principal applications of LiDAR in underground mining and tunnel geomechanics include:

2.3.1. Rock Mass Quality Assessment

Tunnels represent challenging environments for laser scanning because of their confined geometry, limited lighting conditions, and irregular surfaces introduce constraints during both data acquisition and post-processing. Despite these limitations, LiDAR data support numerous operational applications, including the calculation of shotcrete thickness, verification of as-built bolt spacing, identification of potential leakage zones, and detailed three-dimensional characterization of the rock mass (Figure 2). Geometrical properties of discontinuity surfaces, such as their location, orientation, frequency, and large-scale roughness, can be accurately derived from the point cloud. When properly processed and interpreted, these data provide a much more representative geomechanically characterization of the rock mass than can be achieved through traditional hand mapping, which is often constrained by limited accessibility to the excavation face [10].

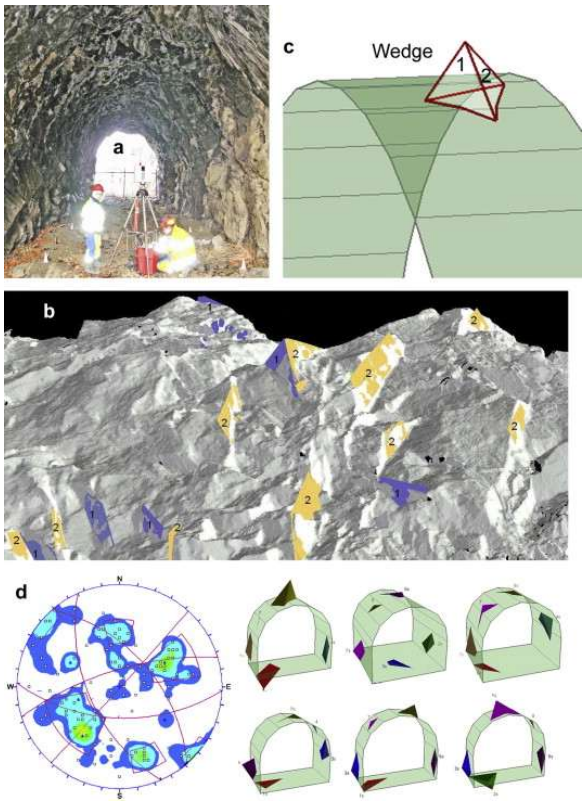


Figure 2 (a). LiDAR scanning: (b) two joint sets identified from LiDAR model (third set facing away from viewer); (c) wedge geometry formed by three identified planes; and (d) Stereonet and wedge models from all unstable combinations of five joint sets.

2.3.2. Excavation Quality and Blasting Control

Following drilling and blasting operations, LiDAR is widely used to assess the quality of the excavated tunnel profile against the design specifications. By comparing pre- and post-excavation scans, engineers can rapidly quantify overbreak and underbreak volumes, verify shotcrete lining thickness (Figure 3), and evaluate the quality of the blasted excavation contour. Advanced parameterized batch-processing methods further analyze tunnel contour profiles using multidimensional similarity metrics, including structural similarity indices and dynamic time warping, enabling objective evaluation of excavation quality, optimization of blasting patterns, and reduction of material waste [10].

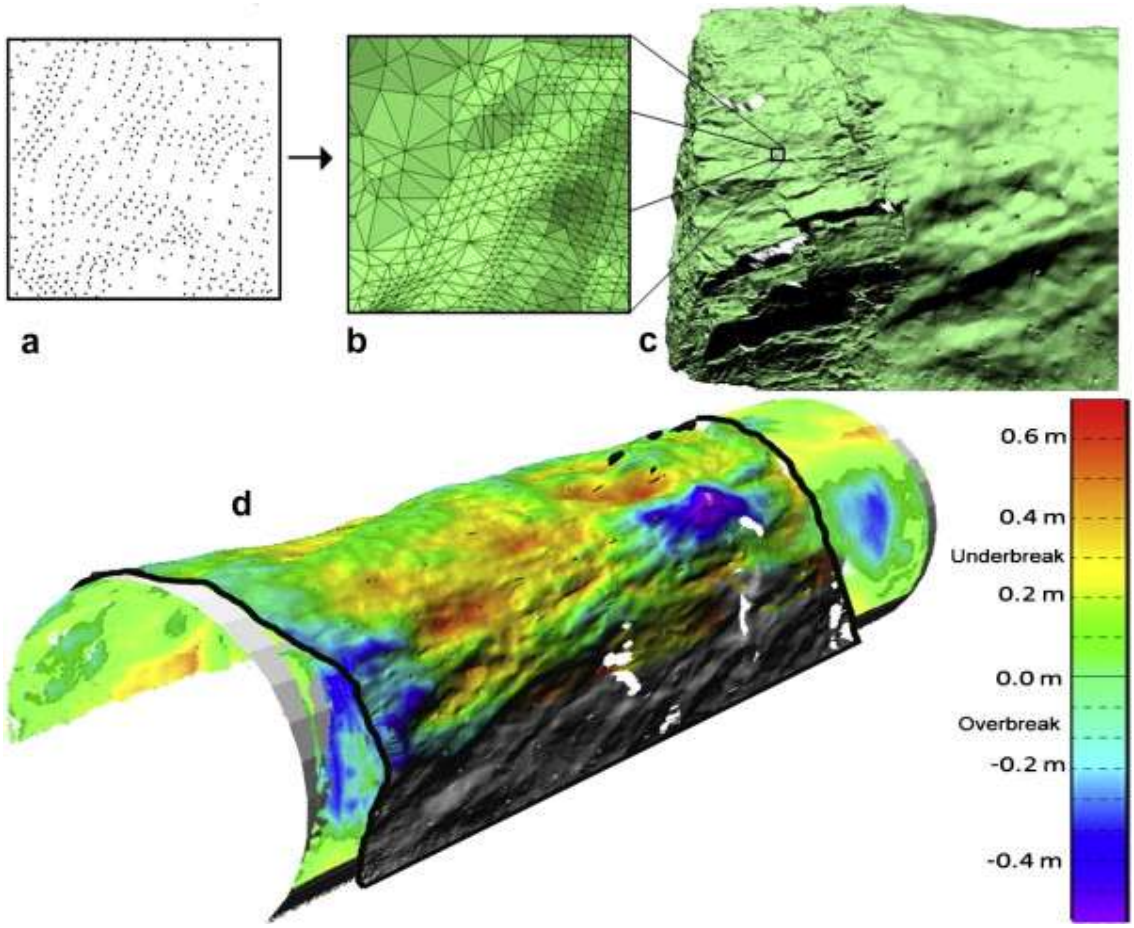


Figure 3. Processing: (a) raw point cloud, (b) triangular mesh, and (c) meshed tunnel model, Sandvik site. (d) Radial difference map showing final shotcrete profile compared to a cylindrical design profile (contoured on the rock–shotcrete model or on the cylinder)

2.3.3. Deformation and Displacement Monitoring

Mobile and stationary LiDAR systems have become essential tools for the long-term structural health monitoring of tunnels. Conventional point-based instruments (such as extensometers, prisms, and laser distance meters) provide displacement measurements only at discrete locations. In contrast, multi-temporal LiDAR surveys can be registered within a common coordinate system to generate spatially continuous displacement fields. By comparing multi-epoch point clouds, millimeter-scale deformations—including lining settlement, wall ovalization, and other forms of structural distress—can be identified and monitored over time [11,22].

2.3.4. Mechanical Scaling Evaluation

In blasted rock tunnels, mechanical scaling is carried out to remove potentially unstable rock blocks remaining after excavation. LiDAR surveys provide an objective means of

evaluating scaling quality, identifying residual unstable areas, and estimating the expected wear of mechanical scaling equipment based on quantitative analyses of post-scaling surface roughness (Figure 4) [12].

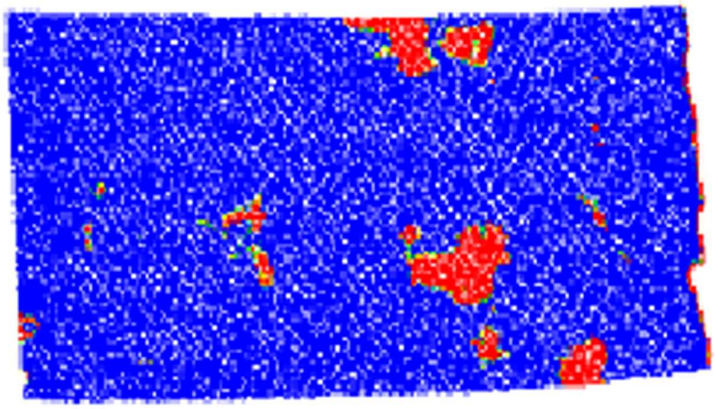


Figure 4. Example of instability potential for Pugwash tunnel scaled surface.

2.3.5. Digital Twinning and Semantic Modeling:

High-density LiDAR serves as the geometric baseline for creating parametric Scan-to-BIM (Building Information Modeling) models. Semantic segmentation networks can automatically identify and remove non-structural auxiliary equipment from LiDAR point clouds, producing a clean tunnel-lining model suitable for structural analysis, defect detection, and crack identification [3,13].

Identification and measurement of the defects

Automating the detection of structural defects in tunnel linings is essential for predictive maintenance. State-of-the-art diagnostic systems focus on several main defect categories:

Water Ingress and Leakage: Water leakage is detected directly from mobile laser scanning (MLS) point clouds by integrating laser intensity and geometric features. Moisture-affected concrete typically exhibits lower LiDAR intensity values than intact tunnel linings because water alters the laser backscatter response. Since intensity alone is susceptible to variations caused by scanning distance, incidence angle, and surface characteristics, the proposed framework combines smoothed intensity with geometric descriptors—including distance to the fitted tunnel ellipse, anisotropy, isotropy, and sphericity—and classify leakage using an XGBoost model. This multi-feature approach enables accurate three-dimensional leakage detection while overcoming the

limitations of threshold-based and image-based methods, which are affected by lighting conditions, threshold selection, and the loss of spatial information associated with 2D projection [15].

Structural Cracking: Tunnel cracks are detected from high-resolution optical images using an image enhancement and deep learning framework. The proposed method first applies an improved Retinex algorithm to enhance image contrast and compensate for the uneven illumination commonly encountered in tunnel environments, making fine cracks more distinguishable. The enhanced images are then segmented using an improved VGG19 convolutional neural network, producing binary crack maps. Subsequently, the Zhang–Suen fast parallel thinning algorithm extracts crack skeletons, enabling quantitative measurement of crack length and width. The approach improves crack visibility under poor lighting conditions and provides accurate geometric characterization for tunnel condition assessment and structural health monitoring, achieving millimeter-level measurement accuracy [16].

Lining Spalling: Spalling refers to the detachment of concrete fragments from the lining. Detection is based on thresholding the local curvature variance computed on triangular meshes. Because the tunnel lining exhibits nearly uniform curvature, localized deviations in the estimated surface curvature indicate areas of degradation [17].

2.4. Automated Railway Track, Centerline Axis Extraction Methodologies and Their Applications

2.4.1. Methodologies

The historical development of automated centerline extraction methodologies ranges from multi-sensor fusion applications to data-driven spatial algorithms. Early efforts to automate track location often combined distinct data types to mask geometric complexities. For example, Beger et al. utilized high-resolution imagery to initially mask and isolate rail tracks within Airborne Laser Scanning (ALS) datasets, defining center points using RANSAC and Total Least Squares.[23] Similarly, Yang and Fang pioneered a data-driven approach that extracted tracks by fusing geometric shapes with intensity values. While achieving a high overall track point detection rate, this technique demonstrated a direct structural dependency on local point cloud density, resulting in performance drops and lower accuracy at complex junctions and rail switches [24].

To streamline data capture and avoid reliance on secondary visual imagery or external terrain data, modern centerline extraction methods focus exclusively on processing dense, single-source spatial point clouds acquired via train-mounted MLS profiling sensors.

In general, centerline extraction using only LiDAR data consists of two main steps. The first is the rail classification from the surrounding environment, followed by centerline calculation. Rail classification is generally achieved using one of two approaches: data-driven methods or model-driven methods.

The model-driven extraction approach relies on fitting predefined parametric templates to isolated rail points to implicitly derive the track center. This method workflow begins grid-wise by partitioning the raw point cloud into a 1 by 1 meter cell matrix. This grid sizing enforces the assumption that a maximum of one rail track is present within a single cell. The algorithm estimates the local Digital Terrain Model (DTM) height using a 10th percentile filter and isolates potential rail head candidates by enforcing an empty height distribution interval between 0.5 meters and 4.5 meters directly above the terrain. Candidate points are evaluated using the Random Sample Consensus (RANSAC) line-fitting algorithm to identify rail segments. Once the top running surface is isolated, the full body and foot of the rail are captured by adding any raw points residing within a downward 0.12-meter distance threshold from the RANSAC line. Parallel tracks are paired by verifying nominal gauge distances (ranging from 1.3 to 1.6 meters) between parallel lines [4,25].

To build the 3D model, the rail candidate points (Figure 5) are divided into longitudinal segments of a fixed piece length (L = 1.2 meters) along the primary eigen vector axis of the track. Because raw scan data are characterized by point sparsity and localized gaps caused by physical self-occlusion, conventional least-squares adjustment methods frequently fail. To address this constraint, a stochastically robust Markov Chain Monte Carlo (MCMC) algorithm deploying Metropolis-Hastings sampling is utilized to recursively solve the joint probability distribution of the six variable 3D orientation parameters (position and rotation).

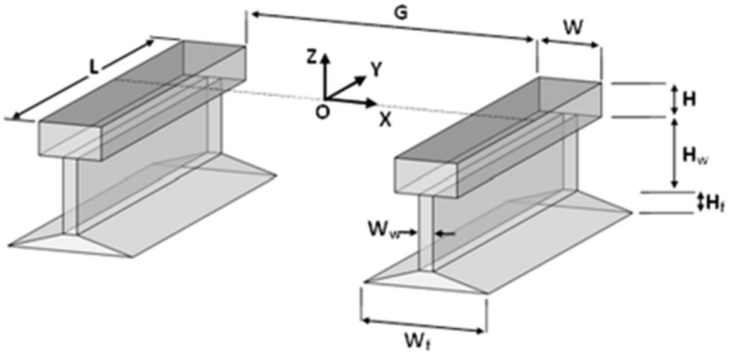


Figure 5. Parametric model definition of a pair of two rails

Because these independently estimated parametric pieces do not automatically form a continuous curve, a low-order (typically third-order) Fourier series interpolation function is executed for each orientation parameter across the trajectory sequence. The track centerline is then implicitly extracted as the continuous path defined by the origin parameter coordinates running centrally between the pairs of synchronized rail meshes. Although the method achieves a modeling accuracy better than 2 cm, the use of a rigid Fourier-series representation introduces localized smoothing errors, resulting in geometric center deviations of up to 2.7 cm in sections with tight curvature [4,25].

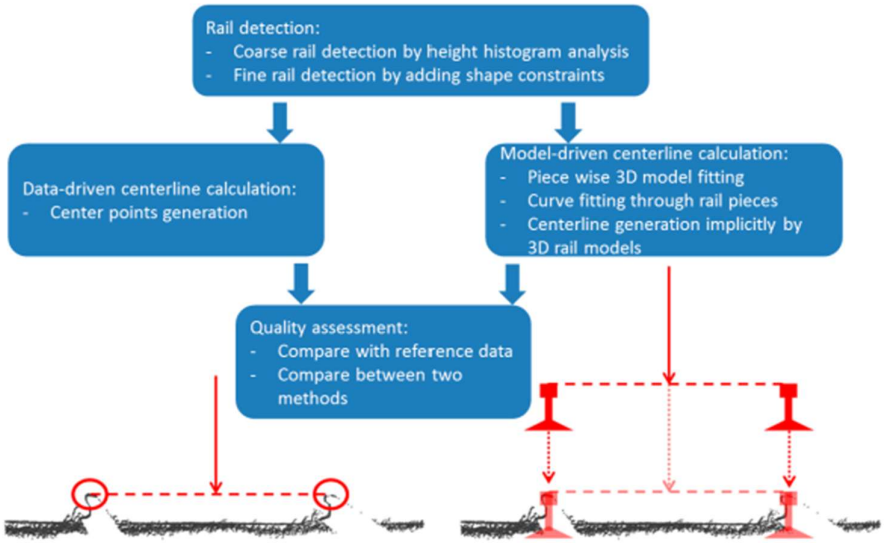


Figure 6. Schematic workflow from rail detection to data-driven centerline calculation proposed by Elberink, S.O.; Khoshelham

Later on, subsequently a model-driven template-fitting approach proposed. After slicing the point cloud, the inner corner of the rail head is identified by matching an

idealized two-dimensional rail-head template to the projected point cloud using the Iterative Closest Point (ICP) algorithm [33].

As an alternative to the mode-driven, **data-driven vector techniques** calculate the centerline directly from the geometric relationships of the raw rail points. Same author of model-driven technique introduced a data-driven framework that isolates top rail points where scanner perspective asymmetry is minimized. For each detected rail point, the algorithm establishes the local track direction, queries the parallel rail within a 2-meter radius buffer, and projects an across-track line perpendicular to the rail axis. The precise mathematical midpoint of this perpendicular projection is calculated and stored as an initial center point. A piecewise linear function is then fitted through these center points at regular intervals (e.g., 2.0 meters) to bridge localized gaps. [4]

A major advantage of this direct calculation is that it handles multi-track corridor scenes, maintaining centerline generation even on adjacent parallel tracks where point densities drop down dramatically due to severe angular data gaps and rail-bed occlusions. The accuracy of the resulting centerlines remains comparable to survey reference boundaries, fluctuating between 1 and 3 cm. However, because the calculation relies strictly on direct cross-track vector matching, its precision degrades sequentially as point cloud density drops on distant parallel track lines. [4]

Recent research introduced the **Trajectory-Synchronized Matching and Tween Curve Extraction method**. This methodology isolates rail data by segmenting the point cloud using the scanner's recorded scan angle metadata. Strict horizontal boundaries are then established for the right rail relative to the scanner trajectory (Figure 7). After Euclidean clustering and the application of a Statistical Outlier Removal (SOR) filter to eliminate isolated noise, the points corresponding to the inner gauge face are extracted [14].

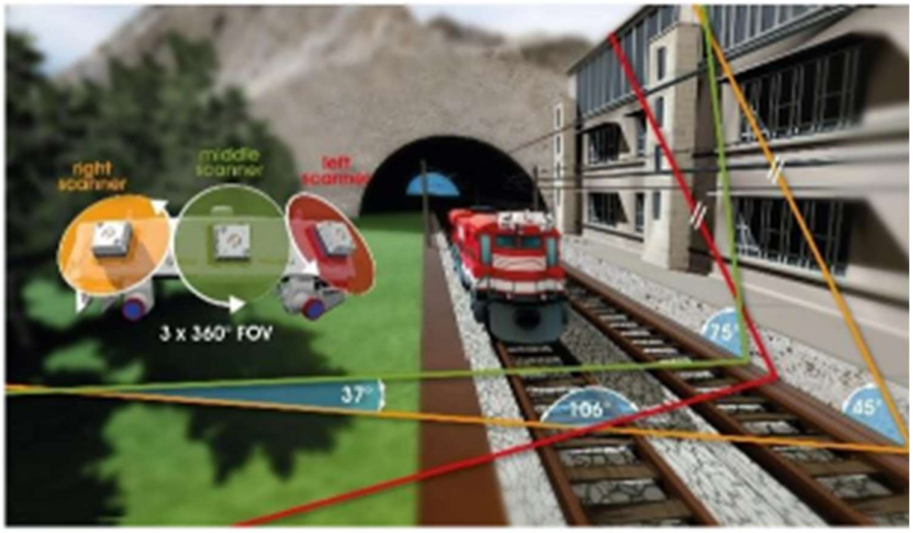


Figure 7 orientation of the VUX-1HA22 sensors and angles between scan lines (Source: Riegl)

To enforce directional consistency across the raw points, a vector orientation process is applied via cross-product calculations. This algorithm enforces a geometric constraint where vectors originating from opposing rails must diverge symmetrically, pointing outwards from the track centerline; inward or asymmetrical vectors are automatically inverted. These Consecutive vertices are passed through an angular deflection filter, discarding any nodes inducing a deflection angle exceeding a strict threshold (alpha = 0.01 rad) to clear duplicate detection artifacts and loops. The refined left and right rail lines are then resampled to contain an identical number of vertices standardized in 1-meter intervals, and the final track middle axis coordinates are computed using a tween curve method that establishes the exact geometric midpoint between corresponding vertices [14].

The defining attribute of this method is its validation framework: data was captured using a Riegl VMX-RAIL MLS profiling system and a dedicated inertial track geometry measurement unit executing simultaneously within an identical coordinate frame. This synchronized configuration decoupled GNSS positioning drift from the analysis, revealing an absolute 3D spatial mean absolute error (MAE) of 5.9 mm for the left track and 9.1 mm for the right track. Under cleaned, standard open-track conditions, the combined 3D spatial error drops to a mean of 4.6 mm. Despite this high absolute precision, the methodology is constrained by fixed scan angle segmentation windows, resulting in tracking vulnerabilities at complex turnout sections where closure rails, guard rails, and frogs introduce geometric interference, as well as in

superelevated curves where vertical definitions diverge from standard baseline trajectory models [14].

Some papers used hybrid approaches; for example, research used data-driven metrics by defining a density ratio to segment the rail from other linear elements. Then it applies a model driven approach by using 3d IFC rail profile templates using ICP [34].

Another hybrid approach begins with a data-driven technique that samples points using local point cloud density and linearity using PCA. After that it applies a mathematical RANSAC model tailored to parallel tracks and finally fits a predefined rectangular rail top template directly to the points [35].

2.4.2. Application

The extraction of a precise 3D railway track centerline serves as the primary spatial and geometric reference frame across a rail tunnel lifecycle. Far from being just a basic vector line, the 3D centerline acts as the architectural anchor for three primary application categories defined in the literature:

- **Multi-sensor registration**: The centerline can provide a common spatial reference for registering observations from LiDAR, inertial measurement units, cameras, and odometry systems. In railway inspection, combining LiDAR and IMU data supports the transformation of scanned measurements into a consistent railway coordinate system. The present study does not implement sensor fusion; therefore, this application is considered a potential extension of the proposed framework [36].
- **Track Alignment and Mechanical Deformation Monitoring:** Regular train traffic introduces structural track shifting and geometric degradation over time. By comparing the extracted as-built centerline directly against the originally designed (nominal) alignment coordinates, maintenance engineers quantify exactly how much the track has shifted horizontally or vertically, providing critical parameters to guide mechanized correction via tamping or alignment machinery [4].
- **Automated Surveying and Asset Inventory Management:** Automated centerline extraction can reduce manual point-cloud interpretation and provide a reference for organizing railway assets by alignment position or chainage. The extracted geometry can also be integrated into GIS, BIM, and digital-twin

environments, supporting the development of georeferenced three-dimensional railway models [37].

3. Integrated Mobile Mapping and Sensor Fusion Architecture

3.1. Brief Technical Specifications of The Mobile Mapping System (MMS)

The data acquisition for the analyzed tunnel environments relies on a highly specialized, kinematic LiDAR scanning configuration optimized for subterranean geometry. The primary Mobile Mapping System (MMS) utilized for data collection features a Z+F PROFILER 9012 laser scanner mounted on a manually propelled rail trolley. This specific scanner operates as a 2D cross-sectional profiling LiDAR, emitting laser pulses in a continuous 360-degree radial plane perpendicular to the tunnel axis.[29]

To reconstruct a continuous 3D point cloud along the tunnel's trajectory, the position of the scanner must be accurately tracked throughout the survey. Because GNSS/GPS signals are unavailable in underground environments, system positioning relies entirely on an Inertial Measurement Unit (IMU) coupled with wheel-mounted encoders. As the trolley moves at a constant low speed and the laser scanner operates at a uniform rotation rate, the system captures highly consistent inter-frame spacing. Furthermore, to mitigate cumulative sensor drift over long trajectories, pre-established control points strategically placed along the tunnel walls are incorporated into the mobile mapping solution, thereby maintaining global spatial accuracy [29].

3.2. Data Typologies (LAS, Coordinate System, Points Accuracy)

The point cloud data evaluated in this methodology is sourced from the Subway Tunnel Semantic Segmentation Dataset (STSD) framework, which provides large-scale, high-fidelity infrastructure scans. Unlike standard commercial datasets, this data is characterized by extreme spatial density and structural complexity. Because the rail trolley platform operates at low manual velocities, the resultant dataset achieves a spatial density exceeding 36,000 points per square meter.

The raw spatial data is parsed and stored using standard binary point cloud formats, specifically the **las** format, which is highly efficient for computational processes. This format retains the massive arrays of three-dimensional Cartesian coordinates (**X, Y, Z**) that have been mathematically transformed from the scanner's raw polar measurements.

The resulting unstructured point cloud contains not only the primary macroscopic tunnel geometry (the concrete lining and the rail tracks) but also auxiliary facilities, such as suspended cables, electrical boxes, brackets, and pipelines. Due to the central positioning of the scanner and the presence of these overhanging facilities, the dataset exhibits distinct local discontinuities and occlusion shadows, a primary geometric challenge that necessitates the advanced spatial filtering, dimensionality features, and voxel down sampling methodologies proposed in the following chapters of this study.

4. Experimental Setup

4.1. Dataset Description

The STSD is a large-scale, multi-modal subway tunnel dataset composed of georeferenced 3D point clouds across circular, quasi-rectangular, and horseshoe-shaped tunnel profiles (Figures 8, 9). The dataset represents in-service subway tunnels with a combined length exceeding 2,700 m, capturing over 2.26 billion spatial coordinates. In total, the tunnel network is partitioned into 264 distinct segments, each approximately 10 m in length. The files are systematically organized within directories as follows Table2.

las folder: Contains the raw, annotated 3D point cloud segments in LAS format. It includes 29 horseshoe shaped segments, 130 quasi-rectangular segments, and 105 circular segments, respectively. The point clouds are annotated into 12 semantic categories, including Background, Rail tracks, Power track, Walkway, Pipe, Cable, bolts, Signal line, Joint, and detachment (Table 3).

In this thesis, 30m portions (3 segments) of each tunnel were analyzed [20].

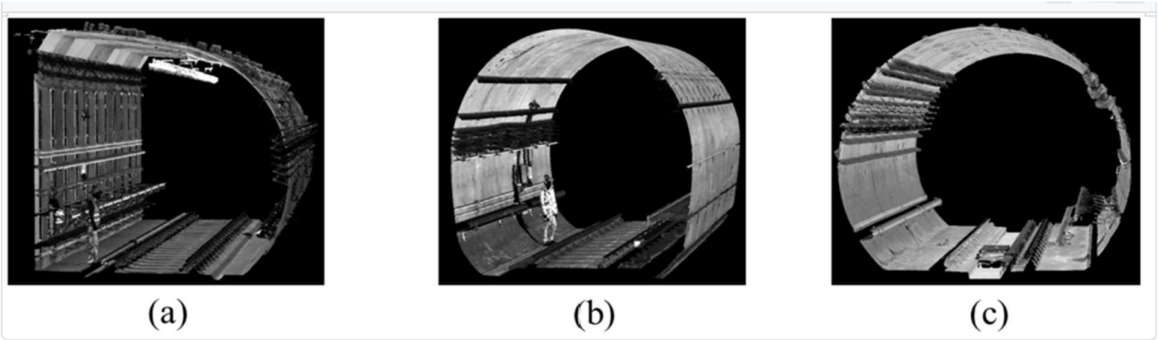
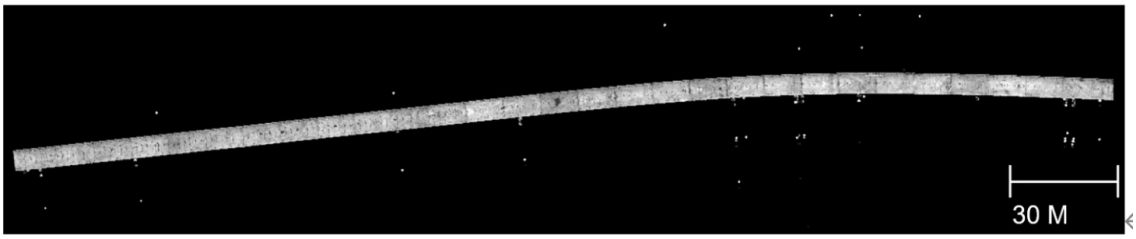
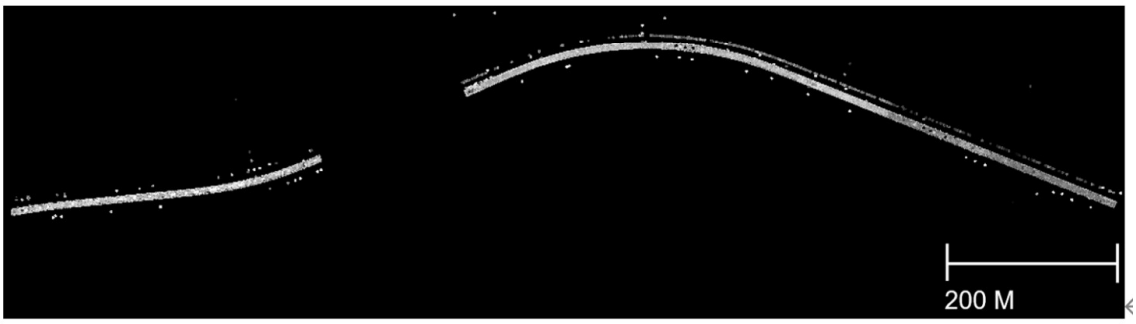


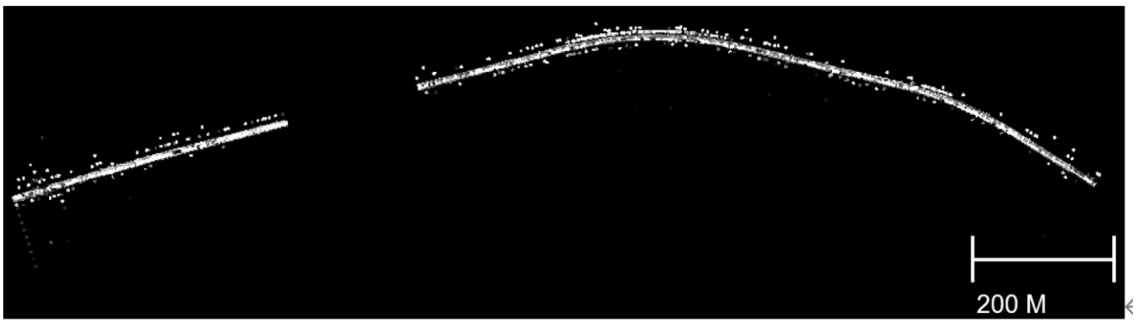
Figure 8. a) Horseshoe section b) quasi-rectangular C) Circular



(a) Horseshoe-shaped tunnel↵



(b) Quasi-rectangular tunnel↵



(c) Circular tunnel↵

Figure 9. Tunnel segments

Table 2. Tunnels description

Tunnel Shape	Length (m)	Data size (GB)	Point number (Million)	Tunnel location
Horseshoe-shaped	296	6.43	266	Shenzhen, China
Quasi-rectangular	1339	27.2	1126	Shenzhen, China
Circular	1071	26.7	868	Wuhan, China

Table 3. Points classes label ID.

Categories	Description	Label ID
Tunnel wall	Default value	0
Rail track	At lower part of tunnel	1
Power track	Located next to, or on the tunnel wall directly above the rail tracks	2
Walkway	Middle part of tunnel, on one side	3
Pipe	About 20cm in diameter, on one side	4
Cable	Cable wires on the tunnel wall	5
Attachment	Attachment equipment on the tunnel wall	6
Person	Only appears in a few tunnel sections	7
Big bolt hole	Rectangle shape, only appears in quasi-rectangular subway tunnel	8
Small bolt hole	Triangle shape, appears in quasi-rectangular and circular subway tunnel	9
Signal line	Located next to rail tracks, appears in quasi-rectangular and circular subway tunnel	10
Joint	Joints between shield tunnel segments	11

4.2. Hardware and Software Stack

The computational pipeline was evaluated on a personal station. To facilitate reproducibility, the exact hardware specifications of the evaluation environment are structured as follows:

- **Processor (CPU): Intel® CORE™ i5-8350U @1.70GHz**
- **Graphics Processing Unit (GPU): Intel UHD Graphics 4GB**
- **Random Access Memory (RAM): 8GB DDR4**
- **Storage: 50GB**

To process the massive spatial datasets and provide the automated centerline extraction pipeline, robust software and library stack were utilized. The primary algorithmic development was conducted in Python 3.9 within a Jupyter Lab interactive environment, while CloudCompare was employed for external visual validation.

The specific software environments and Python libraries, along with their functional purposes within the methodology, are detailed in Table 4.

Table 4. Software environments and programmatic libraries utilized in the study.

Software / Python Library	Primary Purpose and Methodological Application
CloudCompare	Open-source 3D point cloud processing software utilized for visual validation, rendering scalar fields (KPI heatmaps), and verifying coordinate alignments.
Python 3.9 (Jupyter Lab)	Core programming language and interactive development environment used to build, test, and execute the algorithmic pipeline.
NumPy (NumPy)	Package for scientific computing. Applied for heavy vectorized array operations, 3D vector subtraction, and computing Principal Component Analysis (PCA) eigenvalues and eigenvectors.
Open3D (open3d)	3D data processing library. Utilized for spatial morphology tasks, including voxel downsampling, Radius Outlier Removal (ROR), and DBSCAN continuous spatial clustering.
Laspy (laspy)	A library for reading and writing .las files. Used to parse raw scanner data, secure massive Global Coordinate offsets, and export processed point clouds with mapped Data Quality Indicators (DQI).
Scikit-Learn (sklearn)	ML library applied for its spatial algorithms. The KDTree module was crucial for rapid spatial neighborhood queries, continuous radius searches, and midpoint generation.
SciPy (SciPy)	Scientific and technical computing library. The binned_statistic_2d function was used to calculate 10th-percentile ground estimations, while Pearson evaluated KPI correlation statistics.
Time (time)	A Python library is used for system benchmarking to track the exact computational processing speed of each filtration phases.

5. Proposed Methodology

The proposed computational pipeline integrates data-driven processing with parametric geometric modeling. This hybrid approach is designed to overcome the challenges of varying point densities and occlusions in mobile mapping data (Figure 10).

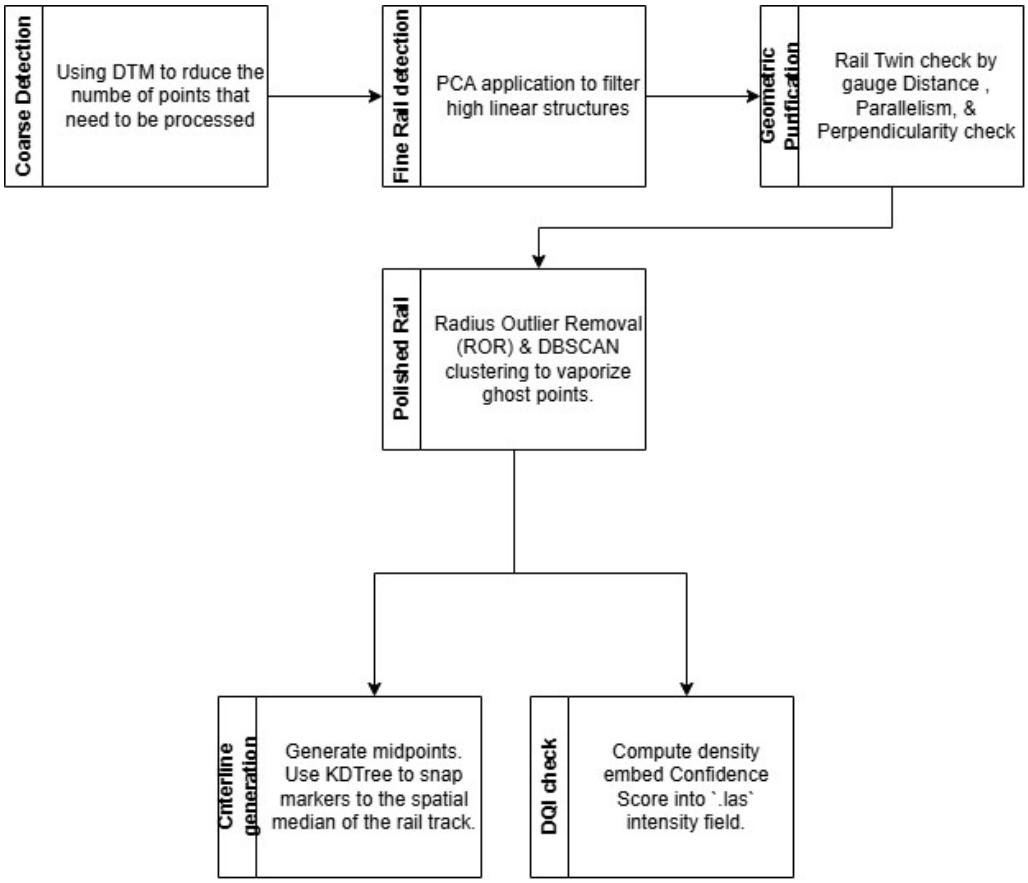


Figure 10. Centerline axis generation pipeline

5.1. Challenges in Raw Data Processing

Raw Mobile Laser Scanning (MMS) acquisitions of underground railway environments present a set of structural and computational challenges that preclude direct centerline extraction without preprocessing. Based on the characteristics of the high-density profiling datasets utilized in this study, four principal challenges dictate the necessity of the proposed algorithmic pipeline:

First, extreme data volume and occlusion discontinuities: Unlike standard mobile mapping, the manual rail trolley platform used in this study generates exceptionally

high-density point clouds, exceeding 36,000 points/m². However, despite this massive data volume, the presence of tunnel facilities (e.g., suspended cables, pipes, and electrical brackets) introduces perspective limitations. These objects cast "shadows" that cause local discontinuities and fractured zones in the track scanlines, requiring robust spatial clustering (DBSCAN) will be explained in chapter 5 to bridge gaps and avoid fragmented extraction [29].

Second, low-hanging background clutter: Tunnels are strictly confined spaces where auxiliary infrastructure often intrudes into the vertical strata immediately above the rail bed. During DTM spatial testing, low-hanging cables, walkways, and distribution boxes were found suspended less than 0.6 meters above the track surface. Because this clutter occupies the same immediate height envelope as the rails, simple elevation-based terrain filtering is insufficient.

Third, **floating-point precision loss**: The raw point clouds are georeferenced using global coordinate systems (e.g., UTM), resulting in coordinate magnitudes on the order of10^5 to 10^6. Processing these large-coordinate values directly can lead to numerical-conditioning issues, and potential precision loss may occur in point-cloud libraries or intermediate operations that internally use single-precision coordinates. Such effects can become relevant when performing geometric operations that depend on fine-scale spatial differences, particularly for sub-millimeter or millimetre-level rail geometry. Therefore, a programmatic spatial shift to a local coordinate origin (0,0,0) is applied before geometric computation to improve numerical conditioning and preserve the reliability of local geometric calculations.

Fourth, **bilateral geometric symmetry**: Even after linearity filtering, continuous objects such as power tack running parallel to the track can produce false positives. To definitively isolate the railway, the algorithm must exploit the track's bilateral symmetry.

5.2. Coarse Detection (Digital Terrain Modeling)

Step 1 implements the coarse detection procedure of Section 3.2 of Elberink and Khoshelham [4] with small modification in the grid size. The research suggested 1×1 m grid but in our case study this grid size would have eliminated some rail points which been close to the tunnel walkway. The raw point cloud is partitioned into a regular 0.5 × 0.5 m horizontal grid; this reduction was determined through preliminary sensitivity testing to prevent the clipping of walkway-adjacent points. Within each cell, the 10th

percentile Z-value is taken as the local Digital Terrain Model (DTM) height, capturing the ground surface while remaining robust to the majority of above-ground returns that inflate the mean value.

$$Z_{DTM} = p_{10}(\{Z_i \in Cell\})$$

A point is retained as a coarse rail candidate if two conditions are simultaneously satisfied: (1) the vertical space between 0.5 m and 3.5 m above the local DTM contains fewer than thirty points, this is the empty space that a train need to pass above the rail and it is empty most of the time — and (2) the point's own Z-coordinate lies between (-5cm and 30 cm) of the local DTM, corresponding to the known rail head height range above the surrounding ballast surface (Figure 11).

$$Candidate\ Points = \{p_i |\ Z_{DTM} - 0.05 \ \leq \ Z_i \ \leq Z_{DTM} + 0.30\ \}$$

Implementation uses scipy.stats.binned_statistic_2d with a lambda statistic computing the 10th percentile, and a second call with statistic='sum' to count forbidden-zone occupancy. The result is a coarse candidate set that rejects the tunnel walls, arch, and overhead equipment while retaining all track-adjacent surfaces that pass the height criteria.



Figure 11.DTM filtration on the left, the accepted grid cells; on the right, the grid cells are rejected because they contain more than 30 points.

5.3. Fine Detection via Dimensionality Features

Before proceeding to the next step, a voxel downsampling of 1 cm is applied to all points to decrease the number of points that need to be processed. This grid is experimentally selected as the optimal resolution balancing geometric feature preservation and computational efficiency. Step 2 applies a PCA-based linearity filter to refine the coarse candidate set by retaining only points whose local neighborhood exhibits linear geometry — as a true rail cross-section does. Rather than applying RANSAC-based line fitting to candidate rail segments, as adopted in previous work [4], the proposed pipeline uses local PCA-based dimensionality analysis to identify linear candidate points. This choice integrates naturally with the subsequent point-level geometric and topological filtering stages and avoids iterative model-hypothesis generation. However, since no controlled experimental comparison between PCA and RANSAC was performed, no conclusion is drawn regarding their relative computational performance or extraction accuracy.

To avoid floating-point precision loss in the covariance estimation, the entire coarse candidate cloud is first shifted to a local origin by subtracting the coordinates of an arbitrary reference point (the first point in the array). The shifted cloud is then passed to Open3D's estimate_covariances function with a radius of 0.1 m. For each point, this constructs the 3 × 3 covariance matrix of its local neighborhood and returns three eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and their corresponding eigenvectors.

A neighborhood that is linear, as a section of rail is, will have one dominant eigenvalue ($\lambda_1 \geq \lambda_2 \approx \lambda_3$), Because the physical rail head forms a continuous, thin extrusion, its local point distribution is distinctly linear. The linearity score is defined as $L = \frac{(\lambda_1 - \lambda_2)}{\lambda_2}$, following the formulation [30.31]. Points with L > 0.6 are retained experimentally by the relation of rail geometry to 0.1m radius; all others are discarded. The primary eigenvector v1 is stored alongside each surviving point and carries forward as the local direction vector used in Steps 3 and 5.

The radius of 0.1 m was selected experimentally: The smaller radius isolates the rail head cross-section reliably and speeds up the process as each box to analyze contains fewer points on the other hand 10 cm of radius, mostly have range of linearity score higher than 0.6 (Figure 12).



Figure 12. An example of linearity score and eigenvalues

5.4. Geometric Filtering

Step 3 implements a three-criterion geometric filtering that exploits the bilateral gauge symmetry of the two-rail track. Inspired by the geometric pairing principles established by Oude Elberink and Khoshelham [4], this methodology adapts the standard gauge and co-directionality constraints into a direct, point-level filtration framework. For each highly linear point p_1 retained from Step 2, a spatial KDTree is queried to retrieve all neighboring candidates within a localized 1.6 m search radius. Each candidate twin point p_2 is subsequently evaluated against three simultaneous geometric criteria:

Distance criterion: The Euclidean distance $||p_2 - p_1||$ must fall within a strict gauge tolerance window of [1.40 m, 1.47m], centered on the 1.435 m nominal international standard gauge. Points residing outside this radial envelope cannot physically represent the opposite rail of the same track system.

Parallelism criterion: The absolute dot product $|v_1 . v_2|$ must exceed 0.95, where v_1 and v_2 denote the primary PCA direction vectors extracted during Step 2. Because both rails of a continuous track run parallel, a high dot product mathematically confirms structural co-directionality.

$$Parallel\ Score = \sum_{j \in Buffer} [(v_1 . v_2 \geq \cos(5°))]$$

Perpendicularity criterion: The unit vector in the direction ($p_2 - p_1$) must be nearly perpendicular to v1, i.e., the across-track component of the connecting vector must dominate. Formally, $\frac{(p_2 - p_1)}{\|p_2 - p_1\| \times v_1|} < 0.15$. This rejects false twins that happen to be gauge-distance apart along the track rather than across it. Finally, the vertical height difference between the paired rails is constrained to a maximum of 10 cm.

A point p1 is retained only if it accumulates at least a minimum number of valid twins passing all three criteria simultaneously the selected value depends on the noise of MMS data collection the lower the value is the higher rail density would be. This density threshold eliminates isolated geometric coincidences — isolated clutter points that by chance satisfy the gauge distance — while confirming points on a solid, continuous rail body that present hundreds of valid twins along its length (Figures 13,14).

The processing is performed in batches of 20,000 points to manage RAM consumption. The output is the purified_rail_points array containing only geometrically verified rail-surface points. The lower the valid twin is the denser the rail will be but also increase the amount of non-rail points.

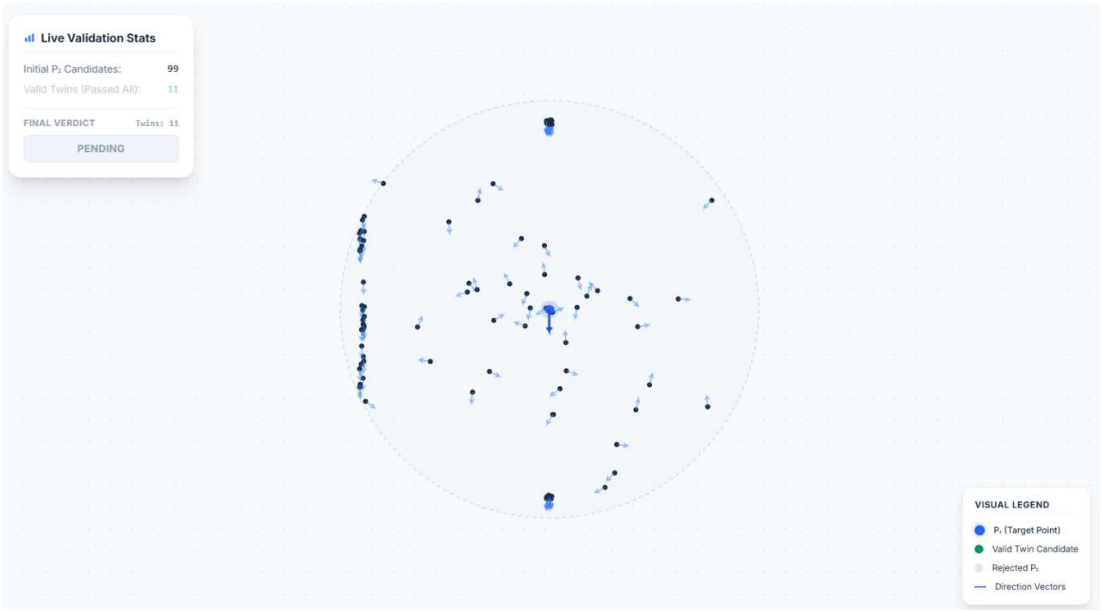


Figure 13. Showing all neighboring points P_2 within the search radius of target P_1, displaying their primary direction vectors

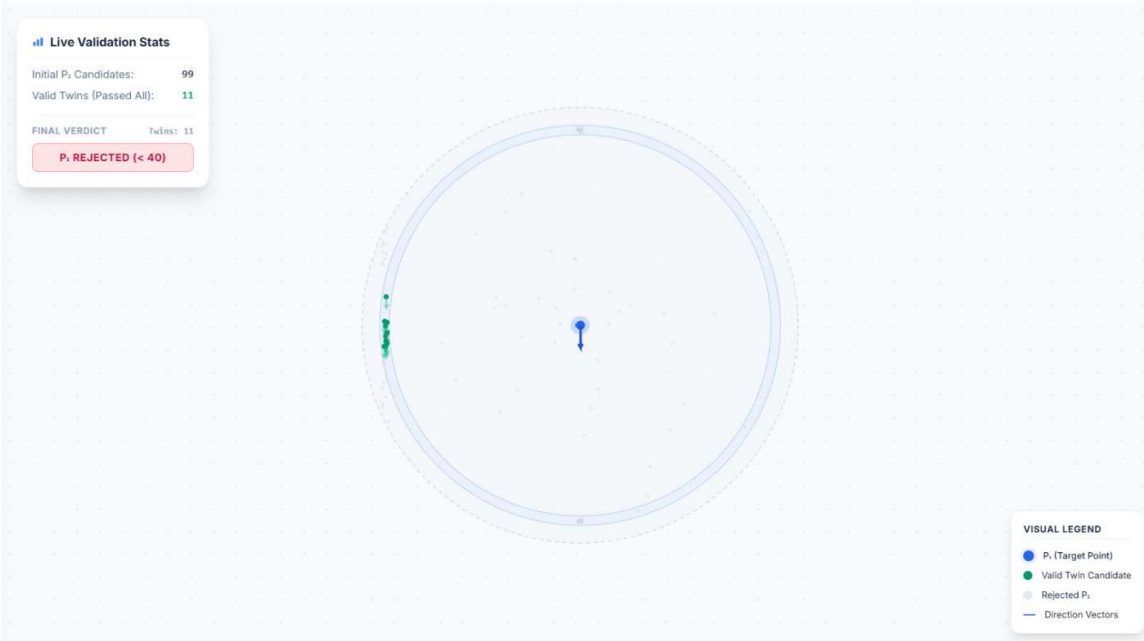


Figure 14. Simultaneously applies Distance, Parallelism (>0.95), and Perpendicularity (<0.15) constraints. Only true twin rail points survive to contribute to the density threshold.

5.5. Spatial Topology Filtering

Step 4. Despite the rigorous geometric constraints applied during the twin-check filtering, the point cloud may still retain sparse "ghost tracks" (Figure 15)—isolated points resulting from temporary scanner reflections or other false geometry that coincidently mimic gauge properties. To extract a free false positive continuous rail model, the pipeline applies a stage spatial density filter.

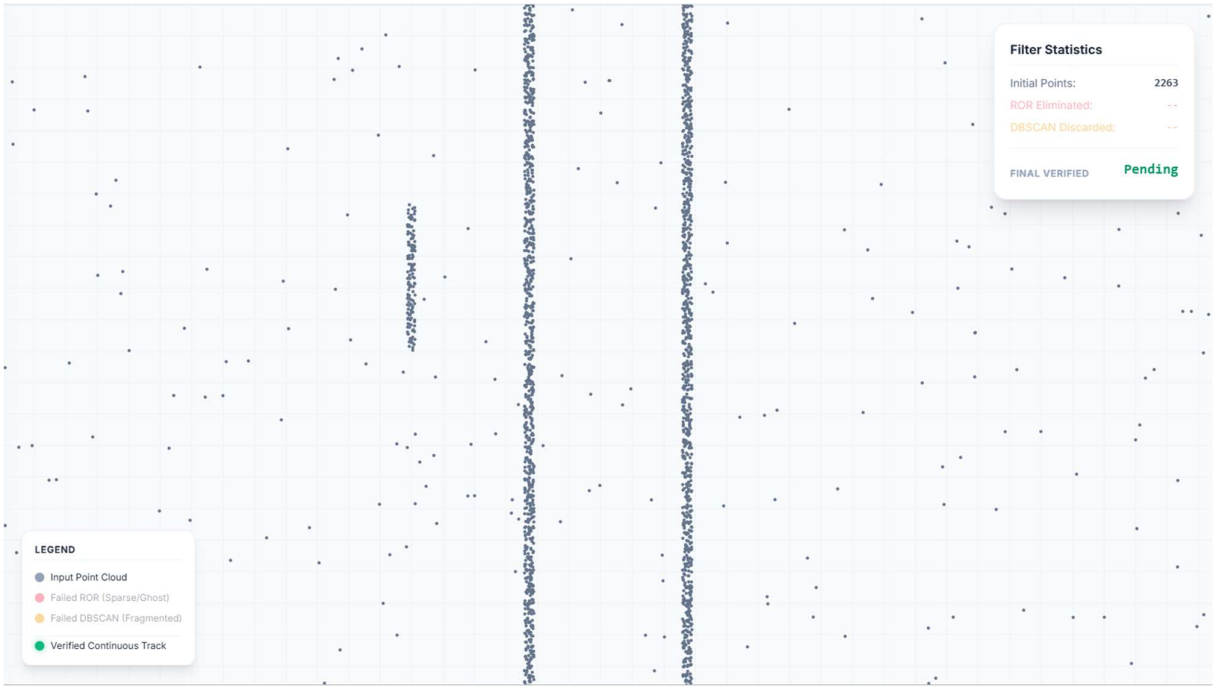


Figure 15. An Example of remaining ghost points

First, a Radius Outlier Removal (ROR) algorithm is applied to enforce a minimum local point cloud density of rail head. Following the voxel downsampling performed in Step 2, the algorithm requires each point to have at least 60 neighbors within a 0.15 m radius in order to be retained as part of the rail head because the rail has almost the width of 6-7 cm and in a diameter of 0.3m it should be a specific minimum number of points in it. This filtering step removes sparse geometric outliers while preserving the dense point cloud representing the railhead Figure16.

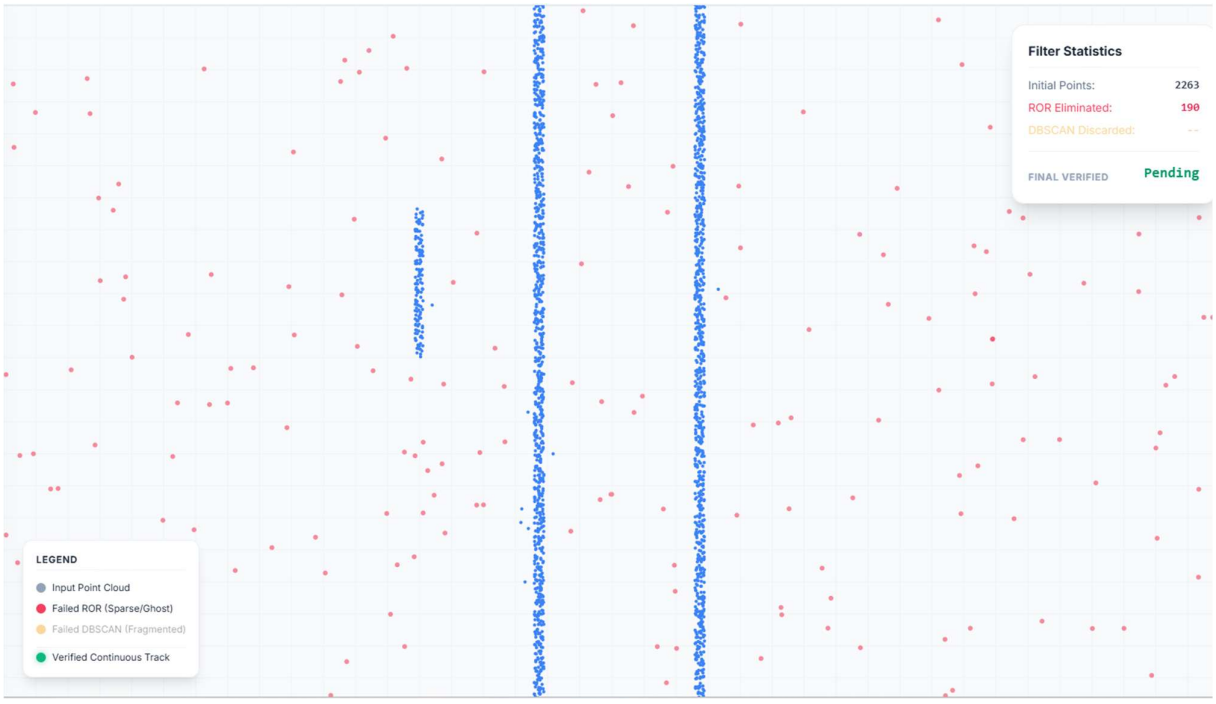


Figure 16 Radius Outlier Removal

Second, to verify the longitudinal continuity of the tracks, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is applied. As originally formulated by Ester et al. (1996) [32], DBSCAN operates without requiring a predefined number of clusters, making it ideal for unorganized infrastructure point clouds. The algorithm groups point that are closely packed together (within an epsilon = 0.15 m) while marking points in low-density regions as outliers. A structural threshold is applied to the resulting clusters, discarding any fragmented segments containing fewer than 3,000 points for big cluster remaining close to the rail that been able to mimic the rail head geometry. This filtering stage ensures that the finalpolished_rail_points array contains only continuous, structurally verified railway lines spanning the length of tunnel Figure17.

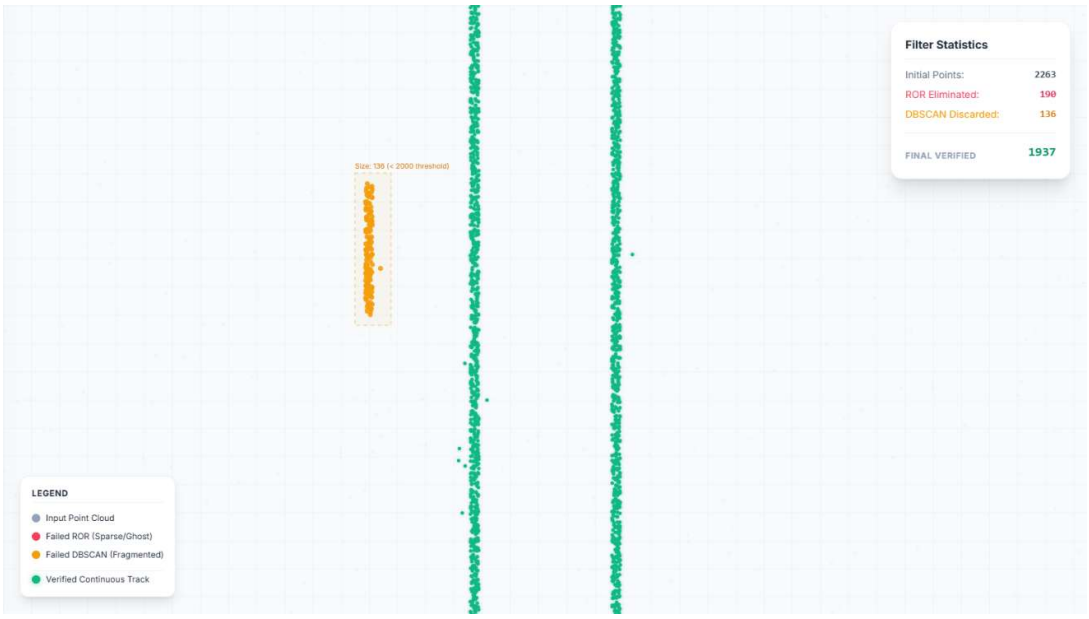


Figure 17. DBSCAN application

5.6. Generative Centerline, Spherical Snapping, and DQI Formulation

Step 5, the final phase of the extraction pipeline generates the definitive 3D trajectory of the railway centerline. As noted by Oude Elberink and Khoshelham [4], the centerline cannot be scanned directly; it must be mathematically derived from the paired left and right rail geometries (Figure 18).

$$c_{raw} = \frac{p_l + p_r}{2}$$

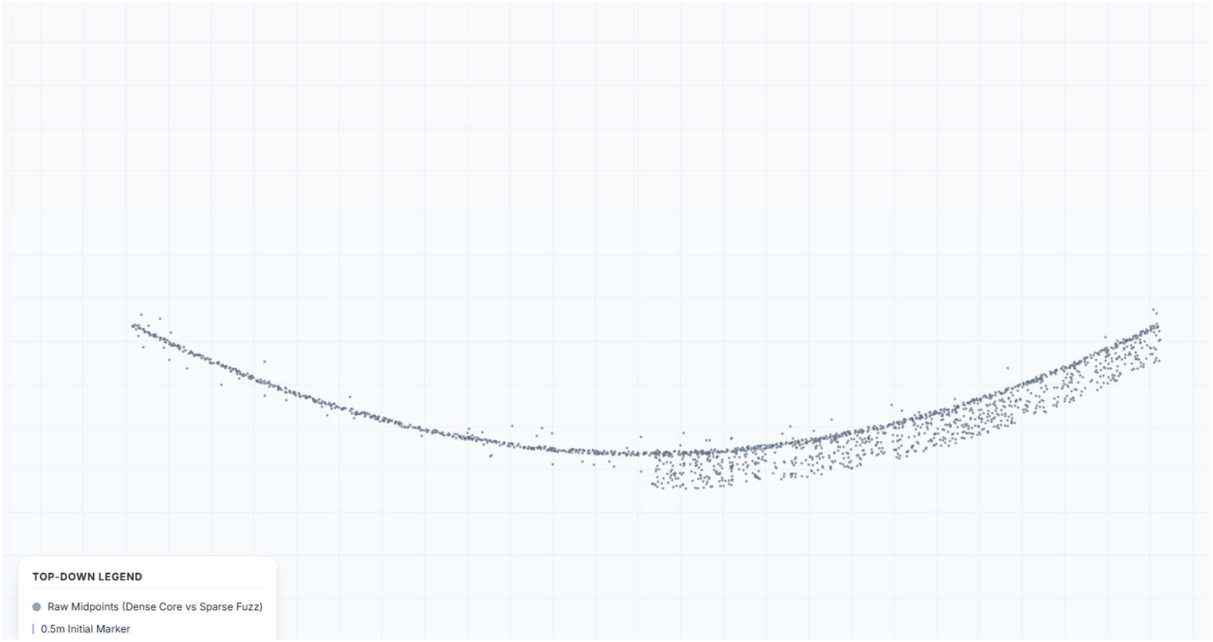


Figure 18. An Example of generated mid-points from left and right rails

Operating on a local coordinate system to prevent floating-point precision loss, the algorithm queries the geometric twins identified in Step 3 and computes the Euclidean midpoint for every valid pair. A 0.5 m voxel grid applied in this final stage serves as an interval centerline generation tool. By encapsulating the high-density midpoint cloud within volumetric cubes, the algorithm forces the fusion of thousands of localized coordinates into singular, equidistant trajectory nodes (Figure 19).

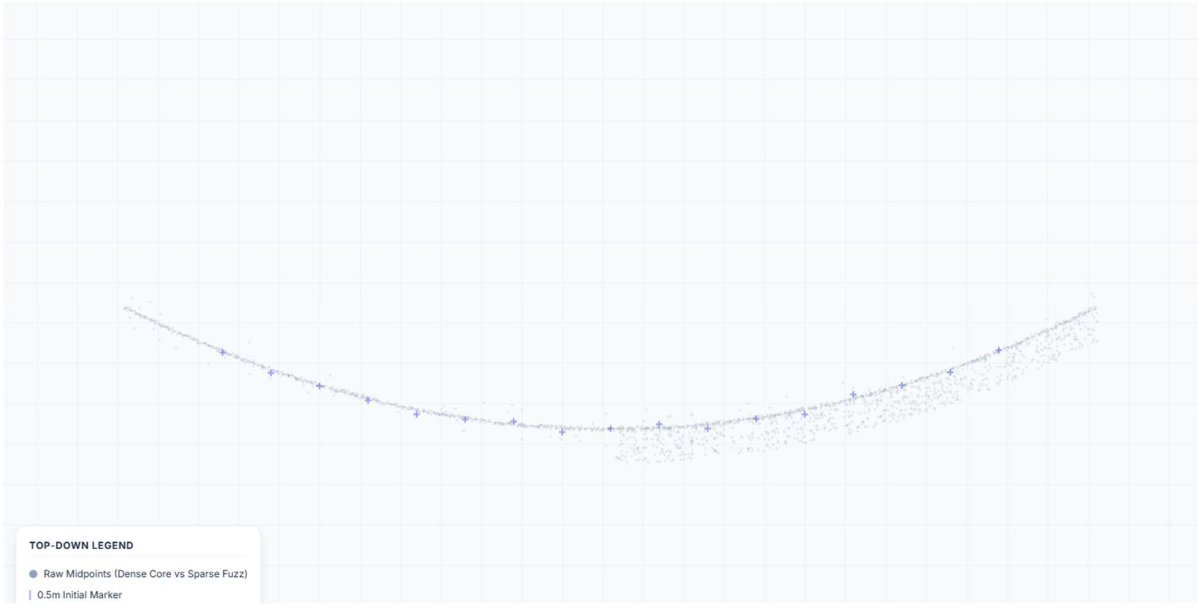


Figure 19. Initial point selected after voxel downsampling

To ensure these voxel markers sit better on the true mathematical track axis, a "Spherical Snap" algorithm is introduced. A KDTree searches a 0.625 m radius around each 0.5 m marker, capturing the localized distribution of raw midpoints. Rather than computing the spatial mean—which is highly susceptible to asymmetrical noise or missing data on one rail—the algorithm computes the spatial median of the local points. This robust statistical snap anchors the final coordinates strictly to the center of the geometric mass.

Simultaneously, this 0.625 m search sphere acts as the data collection envelope for the embedded Data Quality Indicator (DQI). Following the point-level uncertainty paradigms and the meta quality reporting frameworks of ISO 19157, the pipeline evaluates the structural density of the rail data captured within the sphere that supports midpoint data (Figure 20). The local midpoint count is mathematically compared against an 85th-percentile global optimum benchmark (derived from the densest 5-meter continuous track segment). This ratio generates a deterministic Confidence Score (0–100%) that is injected directly into the 16-bit Intensity scalar field of the exported .las file, providing engineers with a spatially contiguous, color-coded heatmap of localized data reliability (detailed in chapter 6).

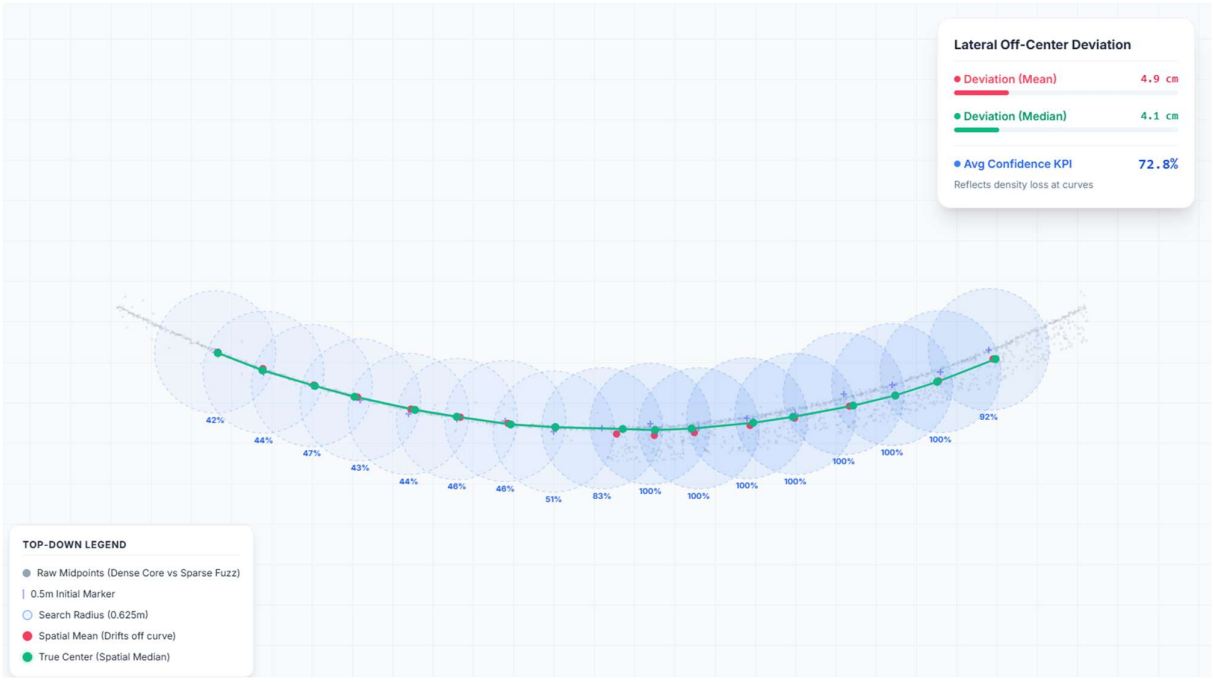


Figure 20. An example of centerline generation while calculating the confidence score.

All selected variables are described in Table5:

Table 5. Pipeline all Variables.

Pipeline Steps	Variable Name	Selected Value	Physical / Algorithmic Purpose	Selection source
Step1 DTM	Grid size	0.5 m	Binned 2D DTM grid cell size	Selected after preliminary sensitivity testing
	Forbidden Zone Envelope	0.5 m to 3.5 m	Overhead train clearance envelope height.	Derived from rail geometry
	Rail Height Window	-0.05 m to +0.30 m	Elevation range is relative to local DTM.	Derived from rail geometry
Step2 PCA	Voxel size	0.01m	Down sampling voxel grid to increase process speed	Experimentally selected
	Linearity radius	0.1m	Local sphere radius for 3*3 covariance.	selected after preliminary sensitivity testing
	Linearity threshold	0.6 < L	Linearity threshold	selected after preliminary sensitivity testing
Step3 Geometric filter	search radius	1.6m	KDTree search sphere for opposite rail.	Derived from rail geometry
	gauge min / gauge max	1.40 m / 1.47 m	Distance window centered on 1.435m.	Derived from rail geometry
	Parallelism Threshold	|v1.v2| > cos (18°)	Cosine of angle between primary eigenvectors.	Experimentally selected
	Perpendicularity Threshold	|derection.v1| < 0.15	Dot product of connecting vector and primary axis.	Experimentally selected
	Minimum Twin Density	len(valid_twins) >= 40	Executioner threshold: minimum passing twins.	selected after preliminary sensitivity testing

Pipeline Steps	Variable Name	Selected Value	Physical / Algorithmic Purpose	Selection source
Step 4 Topology filter	ROR nb_points / radius	60 pts / 0.15 m	Radius Outlier Removal spatial density.	selected after preliminary sensitivity testing
	DBSCAN eps / min_points	0.15 m / 30 pts	Cluster distance and core sample density.	selected after preliminary sensitivity testing
	DBSCAN cluster_size	> 3,000 pts	Minimum cluster points count threshold.	selected after preliminary sensitivity testing
Step 5 Midpoints	Midpoint search_radius	2m	Buffer radius for twin midpoint generation.	Experimentally selected
	Midpoint Twin Threshold	len(valid_twins) >= 5	Minimum passing twins required for midpoint.	Experimentally selected
Step 6 Spherical Snap smoothing	interval_spacing	0.5m	Voxel down sampling size for point sampling.	Depending on the wanted spacing for midpoints
	snap_radius	0.625 m	Median sphere search radius around anchor.	1.25 *interval spacing

5.7. Scientific Contributions of The Proposed Methodology

To establish a clear demarcation between existing literature reviewed in Chapter 2 and the scientific contributions developed in this thesis, the methodology that are used in this paper are explained below:

Algorithmic Approaches to Linear Feature Extraction: RANSAC vs. PCA

In point-cloud feature extraction, the identification of linear infrastructure elements can generally be approached through either iterative mathematical model fitting or local analysis of geometric dimensionality. Random Sample Consensus (RANSAC)

is a widely adopted method for this purpose. It operates iteratively by randomly sampling points to estimate a parametric model, such as a 3D line, plane, or cylinder, and then evaluating the model according to the number of points that satisfy a predefined inlier threshold. This approach is robust to outliers and has been widely applied to geometric feature extraction from point clouds [30,31].

Principal Component Analysis (PCA), in contrast, characterizes the local geometric structure of a point cloud through the covariance matrix of a defined neighbourhood. The resulting eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ describe the distribution of points along the principal directions and can therefore be used to identify local linear structures. In the proposed pipeline, PCA is applied within localized spherical neighbourhoods rather than by fitting an explicit parametric model to a larger point set.

This distinction was an important consideration in the selection of PCA for the proposed methodology. The pipeline is based on local, point-wise geometric analysis, where the geometric characteristics of each candidate rail point are evaluated within its surrounding neighbourhood. PCA is therefore compatible with this local, data-driven architecture, as it characterizes the dimensionality and spatial distribution of neighbouring points without requiring an explicit global model of the railway geometry. This is particularly relevant for curved railway tracks, where the local rail geometry can be approximately linear even when the overall track trajectory is curved.

Within the scope of this study, PCA was adopted primarily because its local covariance formulation aligns directly with the point-based, bottom-up architecture of the proposed pipeline. Evaluating dimensionality at the individual point level isolates linear track candidates along curved tunnel alignments without requiring rigid geometric primitives or global parameterization. Preliminary, informal tests with an iterative RANSAC-based line extraction encountered severe computational bottlenecks on dense profiling point clouds; while these initial observations were not systematically benchmarked, the local covariance framework proved methodologically consistent and computationally suitable for the high-density datasets analyzed.

The Topology Filter (ROR and DBSCAN Integration) Tunnel MLS data is characterized by high density points and noises; to increase semantic purity and suppress false-positive rail-like structures, this methodology introduces a bilateral topology filter. It pairs localized density constraints (Radius Outlier Removal) with

continuous connectivity logic (DBSCAN). This method eliminates isolated points (ghost points) and removes false rail clusters without heavy calculation.

The Spherical Snap Algorithm

A common challenge to calculate the mean axis is that the mean relies on density and having a gap or loss in density can affect the accuracy of the axis. This research introduced the "Spherical Snap" algorithm, which executes a dynamic radial search and calculates the spatial median. This innovation resists asymmetrical density weighting, allowing the generated trajectory nodes to be less sensitive to asymmetric local point cloud density and to remain spatially stable under the tested conditions.

An embedded DQI confidence score

Because the automated centerline extraction pipeline proposed in this study relies on deterministic spatial mathematics and geometric topology, the validation framework focuses on local point support quantification rather than post-hoc algorithmic explainability.

To support future assessment of the suitability of the extracted track axes for structural health monitoring applications, it is essential to indicate relative local data support along the generated data. According to the international standard for geographic data quality, ISO 19157 (2013), data quality evaluation must provide quantitative meta quality descriptors that allow users to determine if a dataset is fit for its intended application. Furthermore, recent research demonstrates that uncertainty in MLS point clouds should be evaluated at the point level, as local geometric variability and distribution uniformity are the dominant predictors of spatial error [27,28].

Building upon these principles and leveraging the optimal neighborhood feature analysis, this study introduces a generative Data Quality Indicator (DQI) metric that is embedded directly into the final point cloud output. Rather than assigning a global accuracy score to the entire tunnel segment, the algorithm executes a self-validation routine during the Spherical Snap Smoothing phase (Step 6) to assign a discrete confidence score to every individual centerline coordinate [26,28].

6. Statistical Analysis and KPIs

This chapter presents the quantitative evaluation framework developed to validate the automated centerline extraction pipeline detailed in Chapter 5. Because the STSD dataset provides point-level semantic classification labels for 11 distinct infrastructure categories—including an explicit 'rail track' class—it enables the computation of absolute, ground-truth-referenced accuracy metrics [29]. The combination of these semantic benchmarks with deterministic spatial metrics constitutes a comprehensive Key Performance Indicator (KPI) and Data Quality Indicator (DQI) framework.

The evaluation of the pipeline is structured into three distinct tiers:

Tier 1 — Semantic Detection Accuracy: Metrics referenced against the STSD ground-truth labels, allowing for direct comparison against baseline detection rates reported in existing literature.

Tier 2 — Geometric and Topological Continuity: Metrics assessing the absolute spatial quality and topological continuity of the final mathematically derived centerline.

Tier 3 — DQI Confidence Score: The embedded, per-point diagnostic metric that maps localized relative local data support directly into the output point cloud.

6.1. Semantic Detection Accuracy Analysis

The evaluation of semantic accuracy leverages the explicit infrastructure labels encoded within the STSD point cloud, where Class 1 designates ground-truth rail tracks. By extracting this ground-truth rail mask during the initial data ingestion phase and propagating it through the pipeline, each intermediate processing stage is evaluated as an independent binary classifier.

To quantify the performance of the extraction algorithm, three standard statistical detection metrics are defined based on the following topological states:

True Positives (TP): The number of points successfully classified as "rail" by the pipeline that are correctly labeled as Class 1 in the STSD rail track points.

False Positives (FP): The number of pipeline-detected rail points that carry a non-rail ground-truth label (e.g., clutter, cables, or ground).

False Negatives (FN): The number of ground-truth rail points that are not present in the extracted set. In this study, a high FN count is expected because the methodology

intentionally decimates the point cloud and extracts a geometrically stable subset rather than attempting to retain every rail point. In addition, possible limitations or inaccuracies in the STSD reference labels may contribute to the observed FN count. The present analysis does not allow these two effects to be quantitatively separated. Therefore, the low Recall should be interpreted as resulting from both the selective nature of the proposed extraction methodology and potential limitations of the reference labels. In contrast to semantic segmentation methods, the primary objective of the proposed pipeline is to obtain a geometrically reliable subset suitable for centerline reconstruction; consequently, Precision is considered the more directly relevant metric for this skeletonization task.

To show this issue better we manually classified the circular tunnel rails according to Figure 21 and did three different analyses. Two without Voxel downsampling at step two and comparing the dataset rail and another one with manual classify. The third analysis is with the second step downsampling and getting the result with the manual rail.

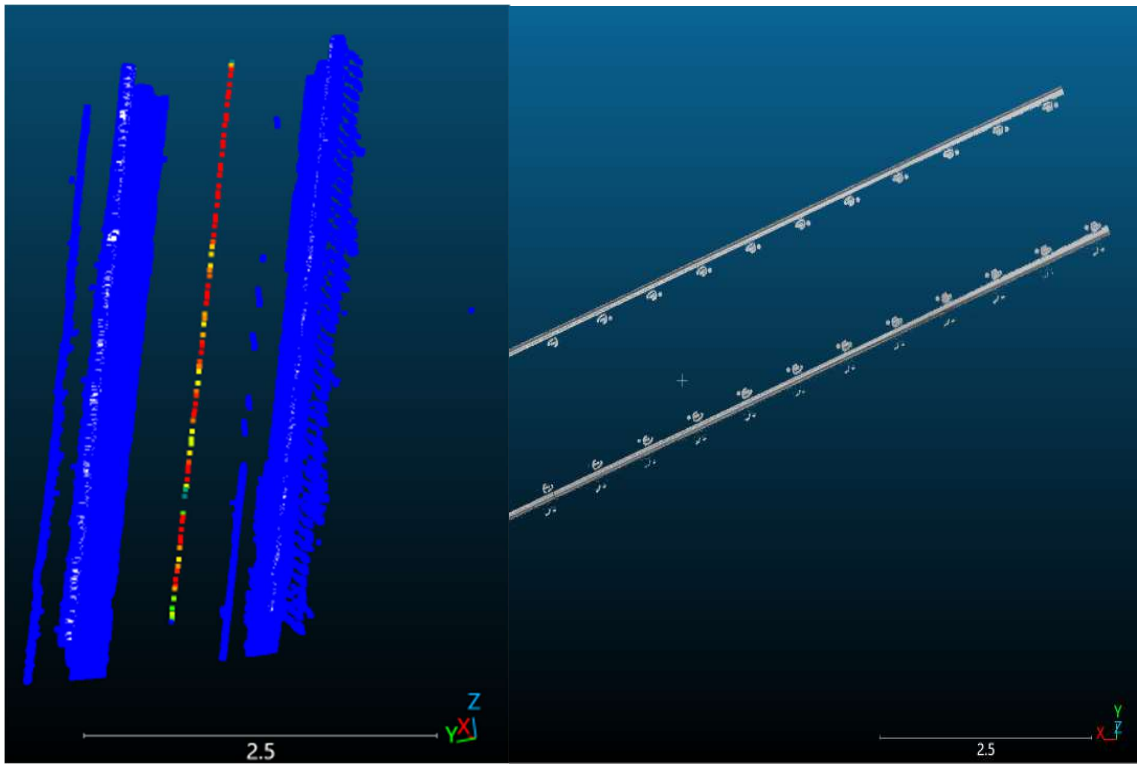


Figure 21. Circular Tunnel left the blue points dataset classified rail, right gray points manual classified rail points.

From these states, the fundamental evaluation metrics are calculated:

Precision (P): Measures the geometric purity of the extracted set—representing the fraction of all pipeline detections that are genuine rail points.

Recall (R): Measures the extraction completeness, representing the fraction of all existing ground-truth rail points that were recovered.

F1-Score (F1): The harmonic mean of Precision and Recall.

By tracking these metrics against the semantic mask, the pipeline's extraction performance was evaluated at the conclusion of each major processing phase. The step-by-step semantic classification results demonstrate the progressive eradication of spatial noise:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

6.2. Geometric and Topological Continuity

While semantic Precision quantifies the proportion of extracted points corresponding to rail infrastructure, it does not inherently guarantee geometric continuity or regular spatial distribution along the track trajectory. To verify structural integrity, the pipeline evaluates the longitudinal continuity of the 3D centerline generated by the Spherical Snap algorithm.

This evaluation measures the Euclidean spacing between consecutive centerline nodes along the reconstructed alignment, reporting the mean spatial interval, standard deviation , and occurrences of macroscopic discontinuities (defined as inter-node intervals exceeding 1.0 m). The absence of such gaps confirms the localized continuity of the extracted centerline within the evaluated segment. However, this metric reflects internal topological consistency rather than an independent benchmark against external geodetic ground truth.

6.3. Data Quality Indicator (DQI) and Statistical Validation

Although Precision evaluates semantic purity and topological metrics assess structural continuity, neither of these metrics provides a quantitative measure of the local relative local data support associated with the reconstructed 3D coordinates. To address this, the pipeline embeds a deterministic Data Quality Indicator (DQI) directly into the output point cloud.

As in step 5, the metadata should include quantitative uncertainty measures, allowing structural engineers to assess the reliability and suitability of the reconstructed data for structural analysis. During the Spherical Snap phase, the algorithm computes the local point cloud density of the raw point cloud using a search sphere with an optimal radius of 0.625 m. This local density is normalized by the 85th-percentile global benchmark to produce the final DQI Confidence Score, ranging from 0% to 100%.

The practical value of this theoretical confidence metric depends on its ability to accurately represent physical reality.

The DQI score focuses on evaluating whether the computed score functions as a reliable proxy for local geometric data availability. To assess this relationship, the Pearson correlation coefficient (r) is calculated between the predicted DQI Confidence Score and the local ground-truth rail point cloud density physically surrounding each generated node:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

A statistically significant positive correlation confirms that the DQI reliably tracks the relative amount of empirical point data supporting the spatial median estimation. To provide practical utility for downstream geospatial workflows, three qualitative operational tiers are proposed heuristically in this thesis. These intervals are intended as preliminary interpretation guidelines rather than validated engineering acceptance criteria:

High Confidence (DQI ≥ 80%): Indicates dense, homogeneous local point support where trajectory nodes are well constrained by surrounding geometry. These sections

represent prime candidates for high-resolution modeling, subject to future geodetic ground-truth verification.

Moderate Confidence (80%≥DQI≥40%): Reflects localized sparsity or partial perspective occlusions while maintaining macroscopic alignment. Suitable for general infrastructure asset inventory and preliminary as-built mapping.

Low or critical confidence (40%≥DQI): Identifies severe local data degradation, high occlusion shadowing, or structural divergence where point support is insufficient to reliably constrain the trajectory. These coordinates are programmatically flagged to alert technicians to areas requiring manual review or potential rescanning.

7. Results

This chapter presents the results produced by the six-step pipeline described in Chapter 5, evaluated using the KPI framework introduced in Chapter 6 and applied to the STSD dataset. The pipeline was executed on all three tunnel segments, representing the three tunnel geometries available in the STSD collection: horseshoe, circular, and quasi-rectangular. For each tunnel, the point cloud is classified according to the categories shown in Figures 22-24 and Table 6 [29].

Table 6: Dataset point classification.

Class Name	Class Number	circular		Quasi-rectangle		Horseshoe	
		Point No	%	Point No	%	Point No	%
Tunnel wall	0	17,059,575	72	17,948,631	74.1	18,965,363	78.3
Rail Track	1	2,391,585	10.1	2,737,259	11.3	2,295,419	9.5
Power Track	2	551,796	2.3	157,873	0.7	413,097	1.7
Walkway	3	254,790	1.1	147,743	0.6	163,124	0.7
Pipe	4	268,465	1.1	325,303	1.3	307,209	1.3
Cable	5	1,485,908	6.3	1,388,866	5.7	1,027,884	4.2
Attachments	6	135,717	0.6	96,847	0.4	42,843	0.2
Person	7	0	0	0	0	104,553	0.4
big bolt hole	8	0	0	474,274	2	0	0
Small bolt ho	9	301,765	1.3	189,152	0.8	0	0
Signal line	10	666,896	2.8	296,525	1.2	0	0
Lining joints	11	568,976	2.4	467,126	1.9	910,104	3.8
	Total:	23,685,473	100	24,229,599	100	24,229,596	100

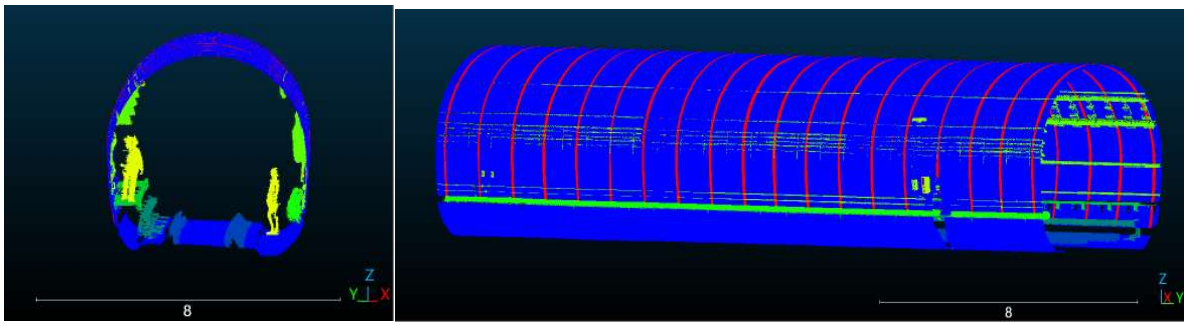


Figure 22. Horseshoe classified points.

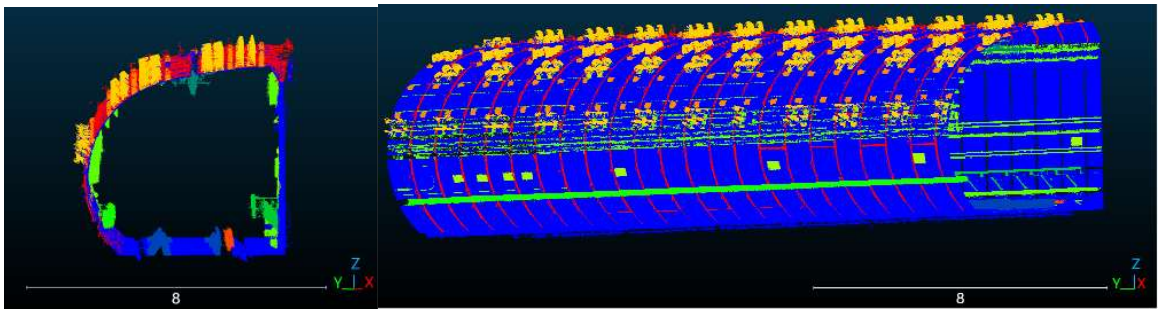


Figure 23. Quasi-rectangle classified points.

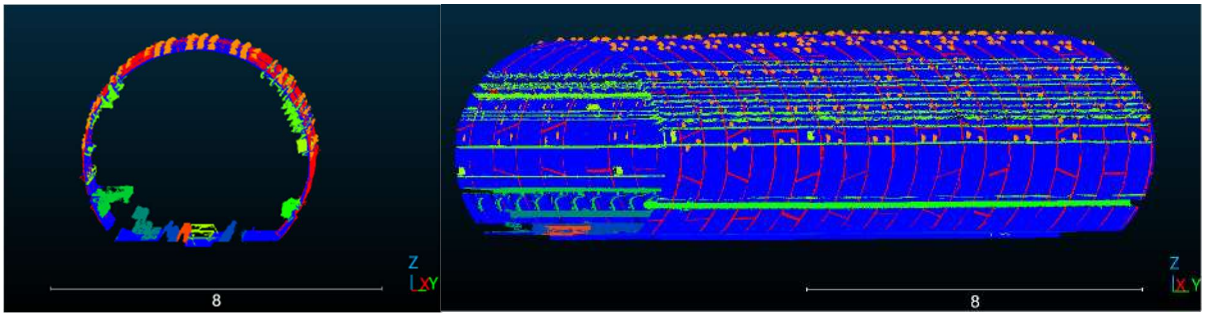


Figure 24. Circular classified points

7.1. Parameters Sensitivity Test Results

In first step the grid size is one of the important parameters that were suggested as 1 by 1 m grid. But in our case studies selected grid end to remove some parts of the rail which been close to tunnel walkway according to Figure 25. Selecting $0.5m^2$ as grid size improved the accuracy of this step shown in Figure 38-40.

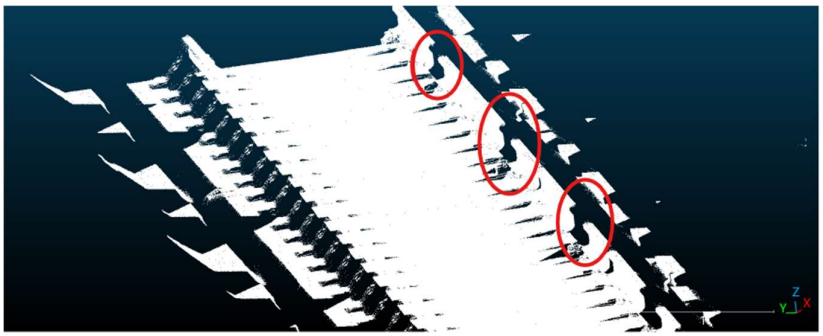


Figure 25. Step 1 DTM, 1m² grid size analysis.

In next step PCA linearity, after 1 cm voxel downsampling, Open3D covariance estimation with four different radius 1, 0.5, 0.2, 0.1, and 0.05m and a linearity threshold is determined through the scalar field as below:

Radius 1m according to Figure 26 linearity values in the rail are scattered more over the process time exceeding over an hour.

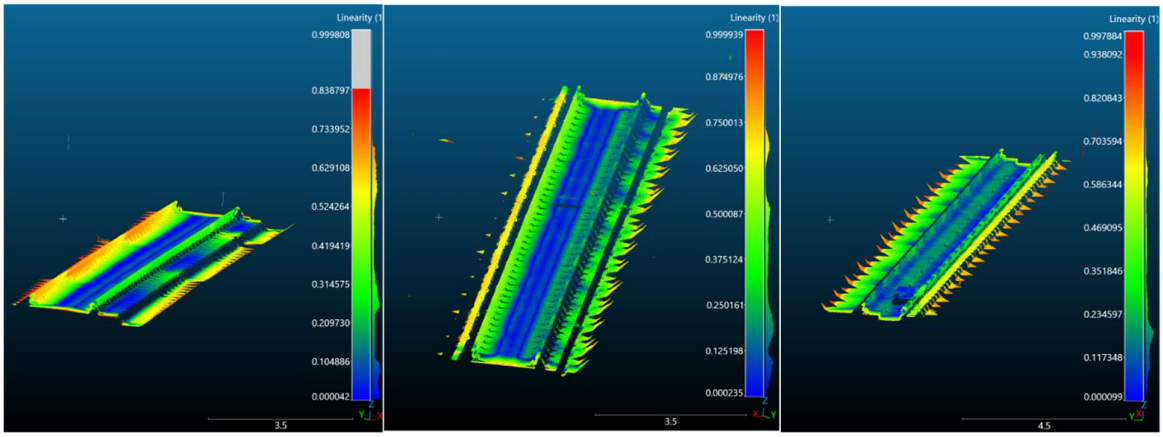


Figure 26. PCA 1m radius linearity scores Quasi-rectangle, Horseshoe, Circular

Radius 0.5m had faster process and rails are classified better according to Figure 27 and process time reduced to 20-30min.

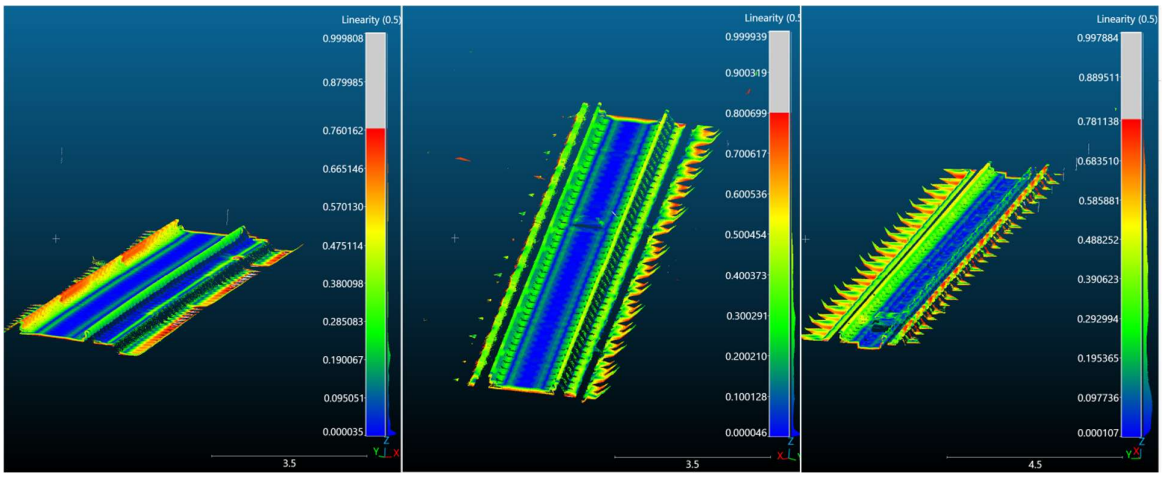


Figure 27. PCA 0.5 m radius linearity scores Quasi-rectangle, Horseshoe, Circular

Radius 0.2m shows a significant improvement in process and classification according to Figure 28; rails are more homogeneous in linearity score and process time reduced to 3-4min.

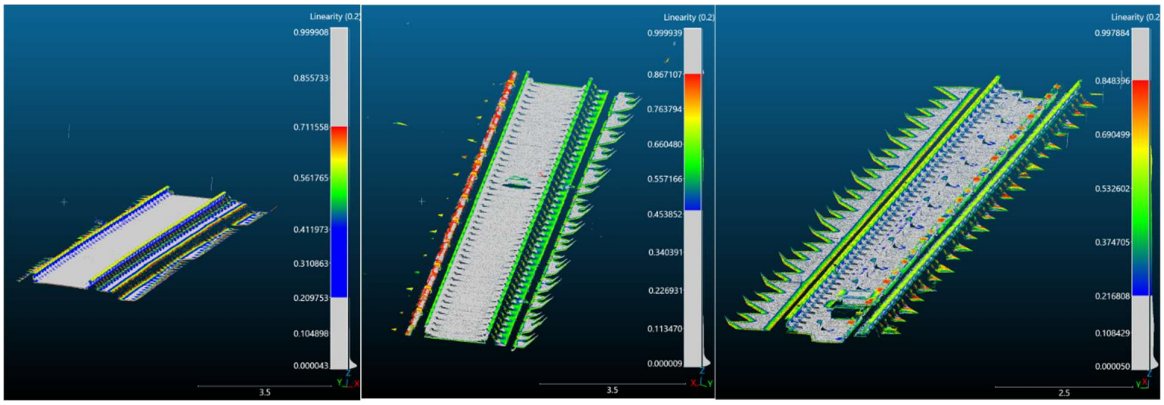


Figure 28. PCA 0.2 m radius linearity scores Quasi-rectangle, Horseshoe, Circular

PCA radius 0.1m shows a good classification as 0.2m according to Figure 28,29 and process time reduced to 20-30 seconds. Having the rail head with around 6-7 cm and radius search of 0.1 to 0.2m can form a perfect rectangular with score of roughly 0.75 but as real case data are noisy this value can very and surrounding non rail points are also affecting this value.

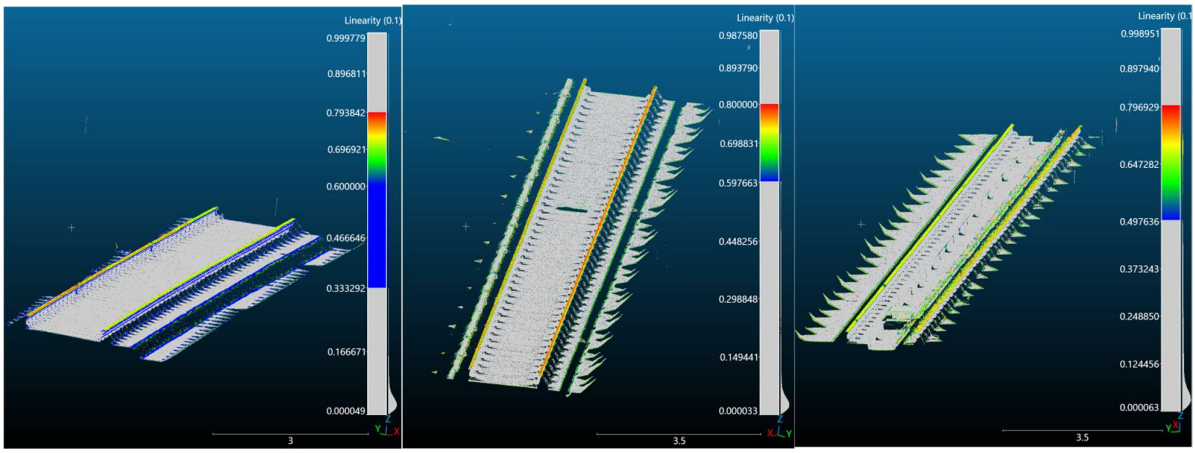


Figure 29. PCA 0.1 m radius linearity scores Quasi-rectangle, Horseshoe, Circular

At the end PCA radius 5 cm is tested and process time reduced to 10-20 seconds, but the rails' head lost their linearity homogeneity Figure 30.

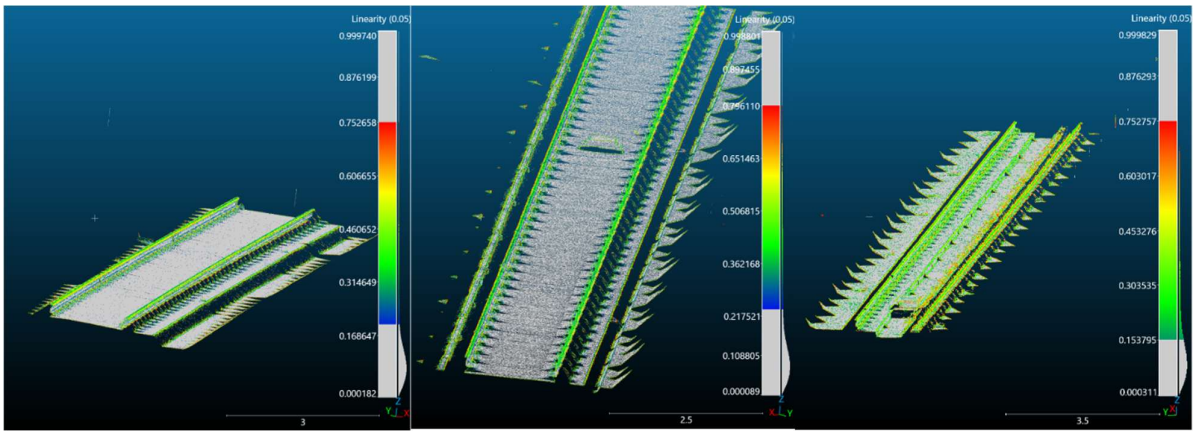


Figure 30. PCA 0.05 m radius linearity scores Quasi-rectangle, Horseshoe, Circular

Step 3 — Geometric Filtering: The three-criterion gauge filter (gauge distance: 1.40–1.47 m; parallelism: dot product > 0.95; perpendicularity: dot product < 0.15), together with a 1.60 m search radius and different valid-twin density threshold, has been applied. The result of sensitivity test is shown in Figures31-33 and Table 7.

Table 7. Points and Process time after geometric filtering

Tunnel Type	Step 1 Point No.	Voxel Downsampling Point No.	Step2 Point No.	Step 3 (Twin.1) Point No.	Step 3 (Twin.40) Point No.	Step 3 (Twin.100) Point No.	Process Time(S)
Circular	6,261,359	1,146,001	145,694	68,107	54,292	38,411	509

Tunnel Type	Step 1 Point No.	Voxel Downsampling Point No.	Step2 Point No.	Step 3 (Twin.1) Point No.	Step 3 (Twin.40) Point No.	Step 3 (Twin.100) Point No.	Process Time(S)
Quasi-rectangle	7,394,495	1,160,466	172,842	54,530	42,561	32,493	505
Horseshoe	7,124,429	1,061,985	107,055	43,933	38,064	30,996	245

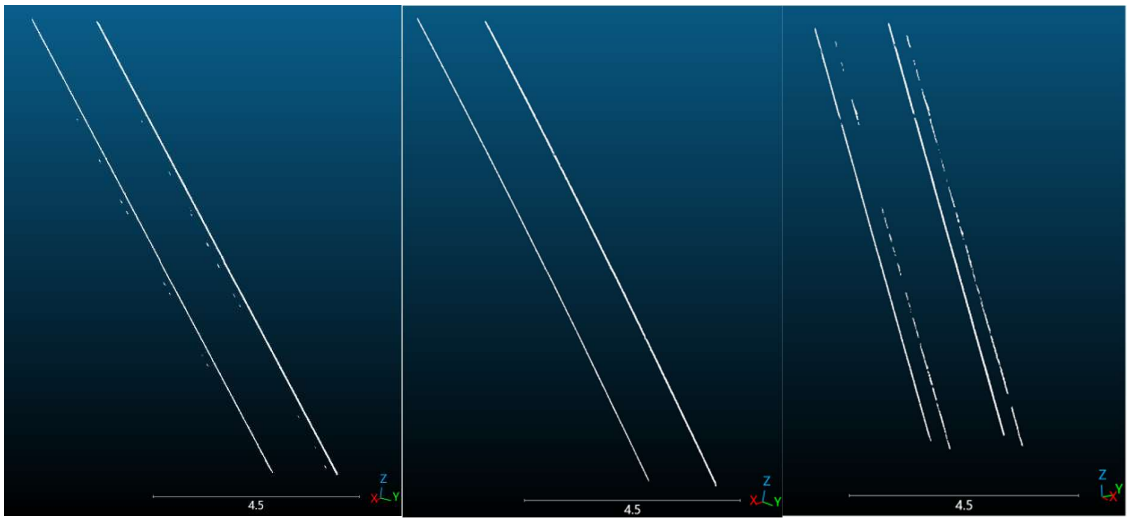


Figure 31. Geometric filtering with 100 valid twins results for Quasi-rectangle, Horseshoe, Circular

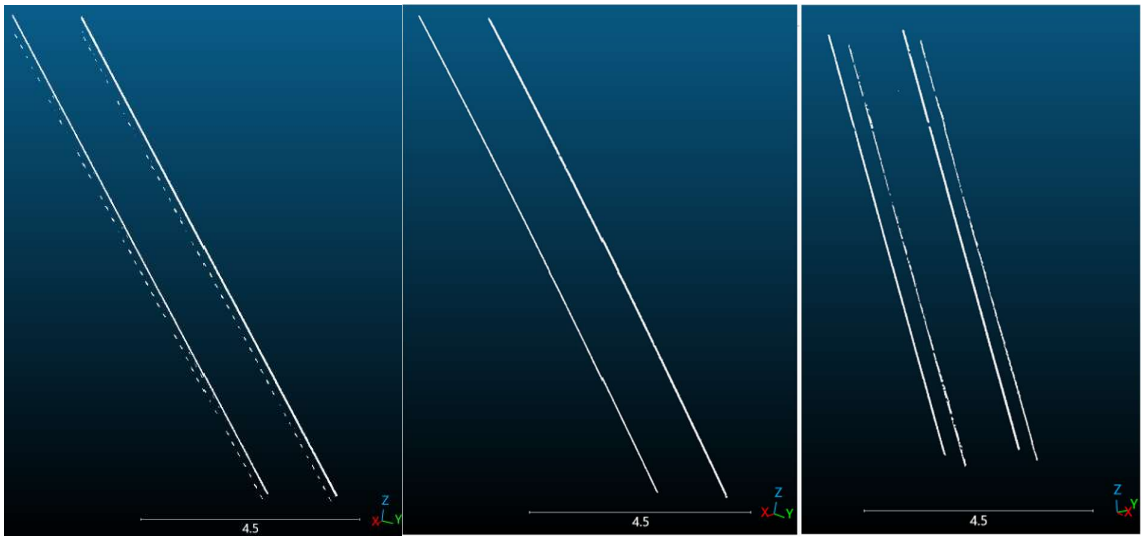


Figure 32. Geometric filtering with 40 valid twins results for Quasi-rectangle, Horseshoe, Circular

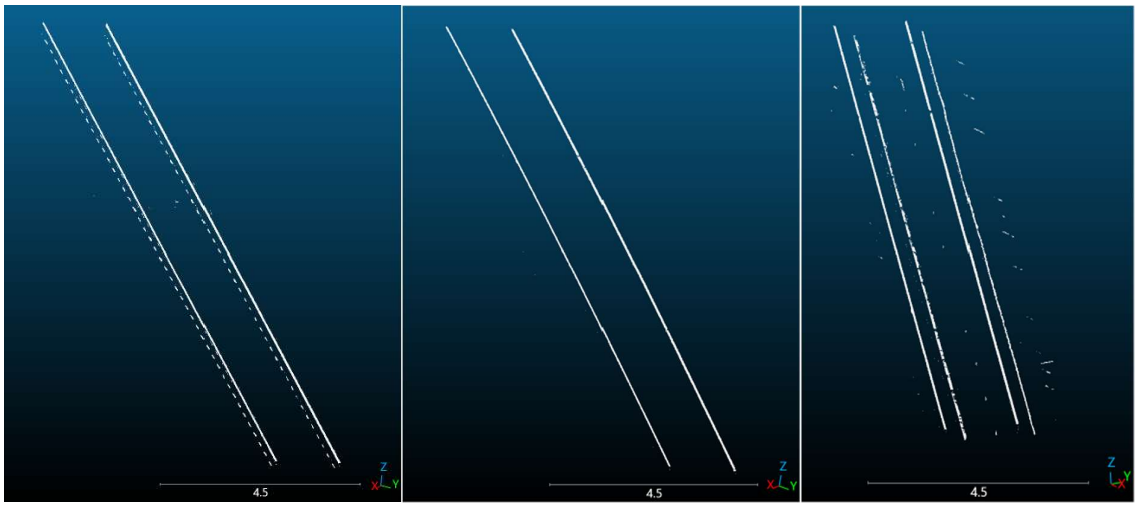


Figure 33. Geometric filtering with 1 valid twin results for Quasi-rectangle, Horseshoe, Circular

In step 4, radius is selected proportionally regarding the rail head and neighbors' thresholds 40 was selected as final value and the results are shown in Figure 34, 35 (as horseshoe rail shows an acceptable geometry referring to Figures31-33 and sensitivity analyze is not applied on the horseshoe dataset).

DBSCAN eps = 0.15 m, min_points = 30, cluster size higher than 1000 and 3,000isolated the two dominant rail clusters. The zero false positive rail points cloud contains two clean, continuous rail bodies described in Figures 36-37 and Table 8,9.

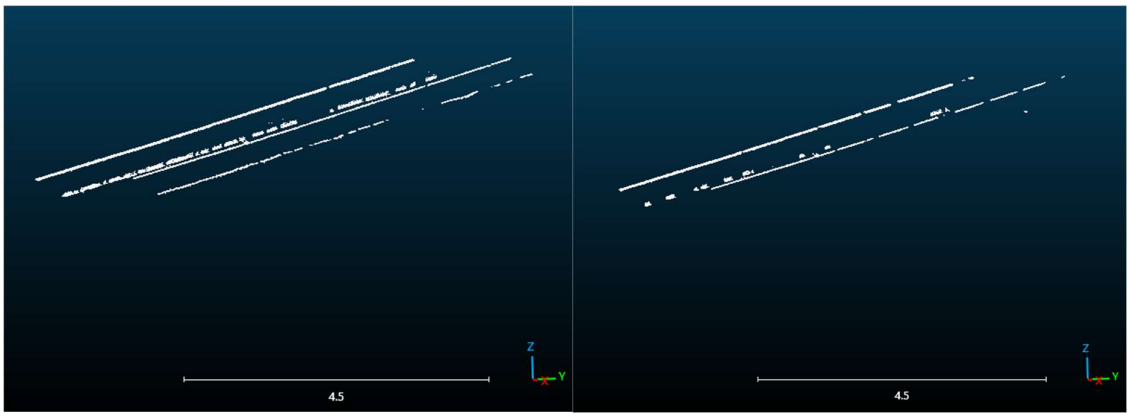


Figure 34. Circular ROR filter 60 and 150

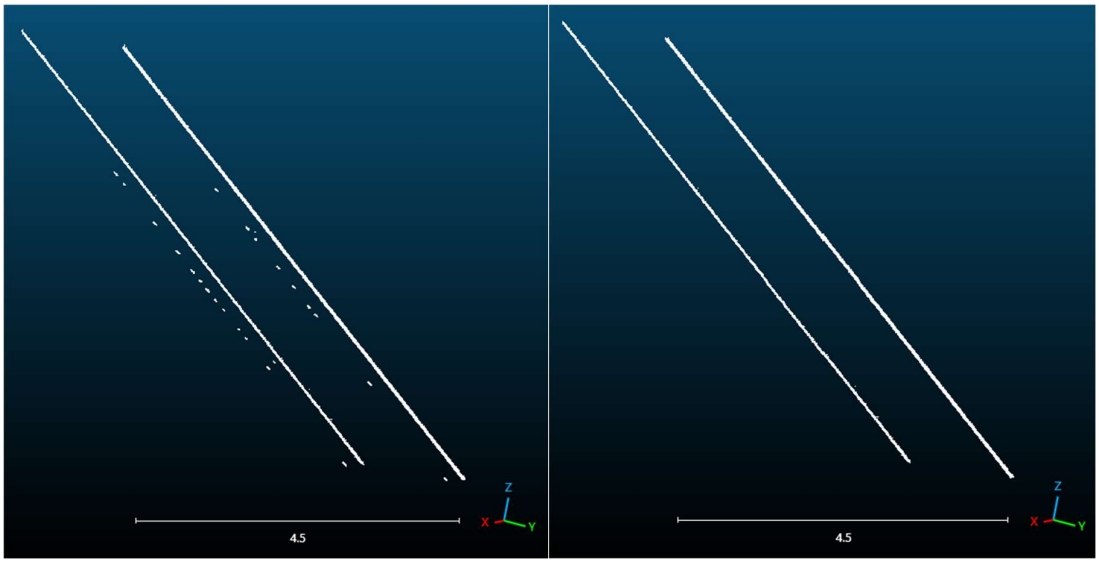


Figure 35. Quasi-rectangle ROR filter 60 and 150

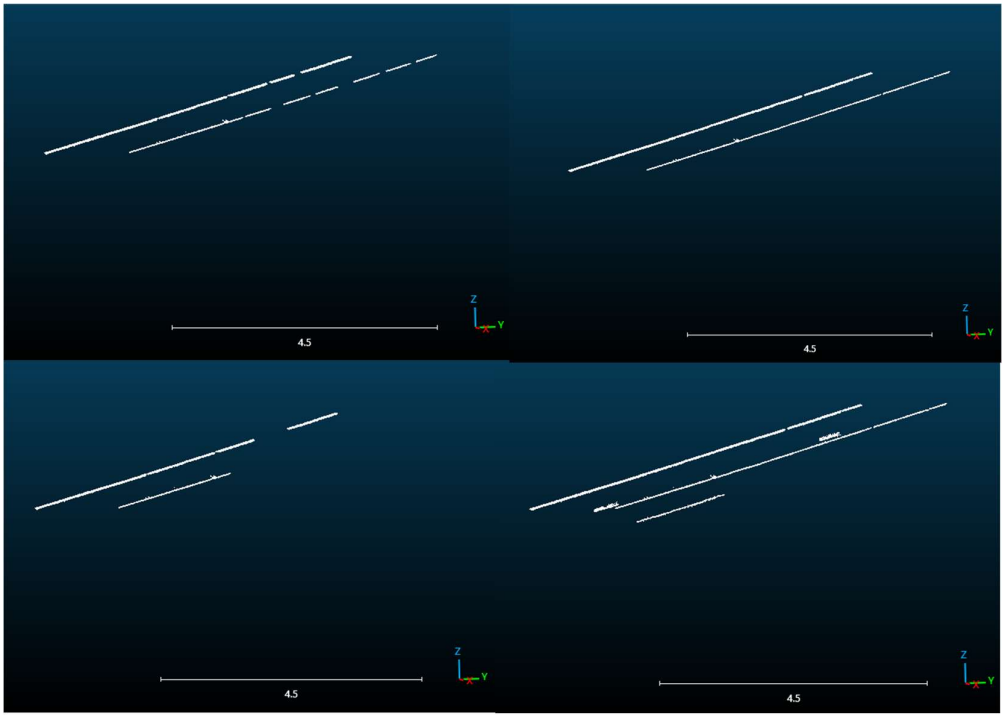


Figure 36. Circular DBSCAN results Top left: ROR 60 Cluster 1000, Top right: ROR 60 Cluster 3000, Down left: ROR150 Cluster3000, Down Right ROR150 Cluster 1000

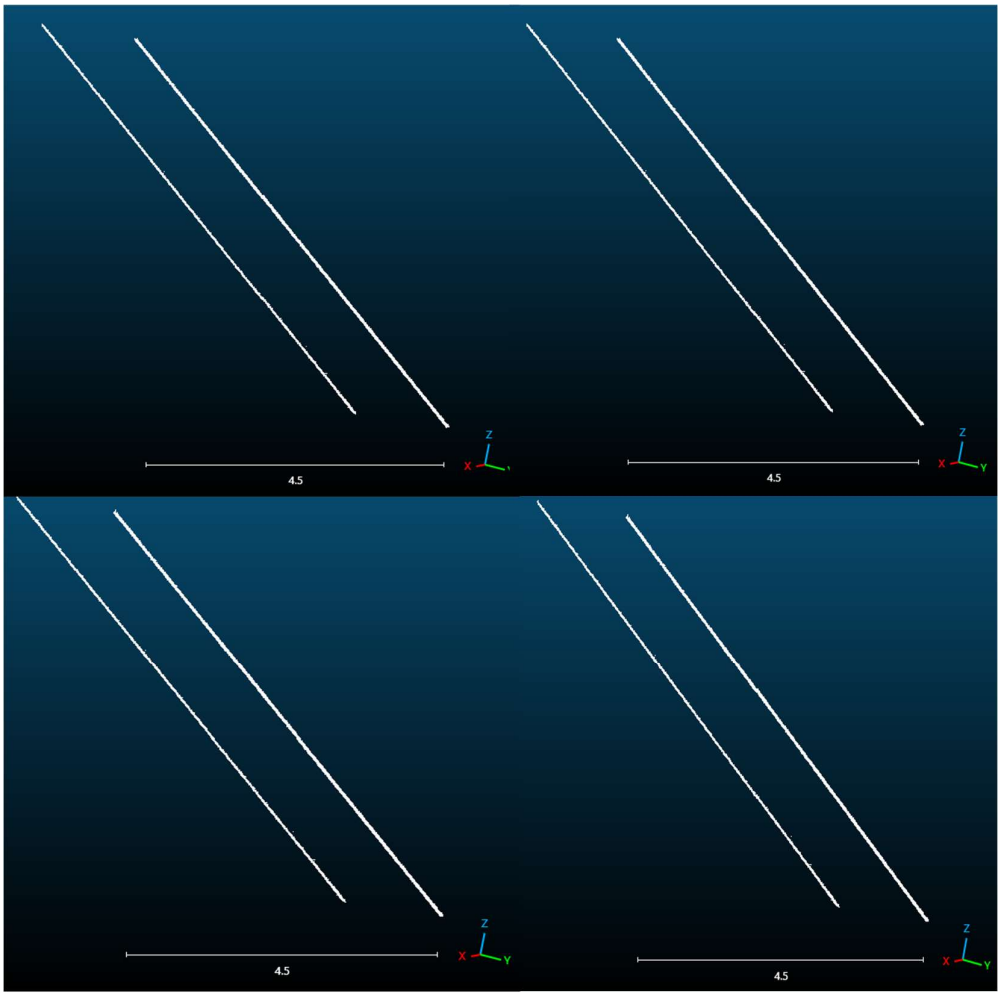


Figure 37. Top left: ROR 60 Cluster 1000, Top right: ROR 60 Cluster 3000, Down left: ROR150 Cluster3000, Down Right ROR150 Cluster 1000

Table 8. Points and Process time after ROR and DBSCAN for 1 valid twin data

Tunnel Type	Step 3 Point No.	ROR filter 60points r0.15m	ROR filter 150points r0.15m	Process Speed(s)
Circular	54,292	51,877	35,902	2
Quasi-rectangle	42,561	40,402	38,821	1
Horseshoe	38,064	38,063	NA	0

Table 9. Points and Process time after ROR and DBSCAN for 40 valid twin data

Tunnel Type	ROR Point	Step 4 Point No.	DBSCAN 1000points per cluster	DBSCAN 3000poinst Per cluster	Process Speed(s)
Circular	60	37,277	41,488	37,277	1
Quasi-rectangle	60	38,967	38,967	38,967	1
Horseshoe	60	38,063	NA	38,063	0
Circular	150	37,277	32,133	23,017	1
Quasi-rectangle	150	38,967	40,402	40,402	1
Horseshoe	150	38,063	NA	NA	0

As circular tunnel points showed more sensitivity to parameter change in Table 10 it is reported the KPI results of sensitivity test.

Table 10. Circular tunnel sensitivity test

Processing Step	Tested Parameter	Parameter Values	Point Count	Execution Time (s)	Precision (%)	Recall (%)	F1-Score (%)
Step 2: Fine PCA Linearity	Search Radius	1	162,665	2800	53.5	5.7	10.3
		0.5	106,999	844	42	2.6	5
		0.2	106,718	103	47	2.8	5.2
		0.1	153,821	28	56.3	3.6	6.8
		0.05	72,880	11	51.7	4.3	7.93
Step 3: Geometric filtering	Valid Twin count After PCA 0.1m	1	69,197	341	75.3	1.13	2.23
		40	55,041	342	81.23	1.87	3.65
		100	39,007	340	85.1	2.6	5.05
Step4: Spatial Topology Filtering	DBSCAN Cluster Size after ROR 60	>1000 pts	42,197	1	94	3.1	6
		> 3000 pts	37,436	1	100	1.57	3.08
	DBSCAN Cluster Size after ROR 150	>1000 pts	32,392	1	100	1.4	2.76
		> 3000 pts	23,276	1	100	1.2	2.37

7.2. Rail Extraction Result

The multi-stage extraction pipeline was evaluated across three distinct subterranean geometries from the Subway Tunnel Semantic Segmentation Dataset (STSD): circular, quasi-rectangular, and horseshoe-shaped profiles.

In Step 1, applying the 0.5×0.5 m horizontal grid filter combined with the 0.5--3.5 m clearance envelope and the -0.05 to $+0.30$ m height stratum reduced the raw input point clouds down to candidate sets covering the rail bed, ground surface, and adjacent walkways. This step removes millions of extra points to process for the next step around %70 of the whole data set (Figures 38-40).

Table 11. DTM filtered points and Process time.

Tunnel Type	Initial Point No.	Step 1 Point No.	Process Time(S)	Point Drop %
Circular	23,685,473	6,261,359	16	73
Quasi-rectangle	24,229,599	7,394,495	16	69
Horseshoe	24,229,596	7,124,429	17	70



Figure 38. Horseshoe After DTM filtration (Blue points are rail classified points in dataset)

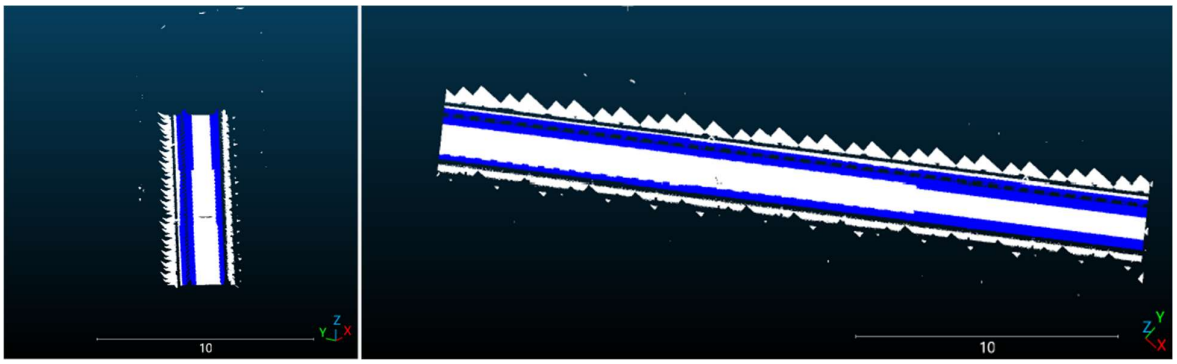


Figure 39. Quasi-rectangle After DTM filtration (Blue points are rail classified points in dataset)

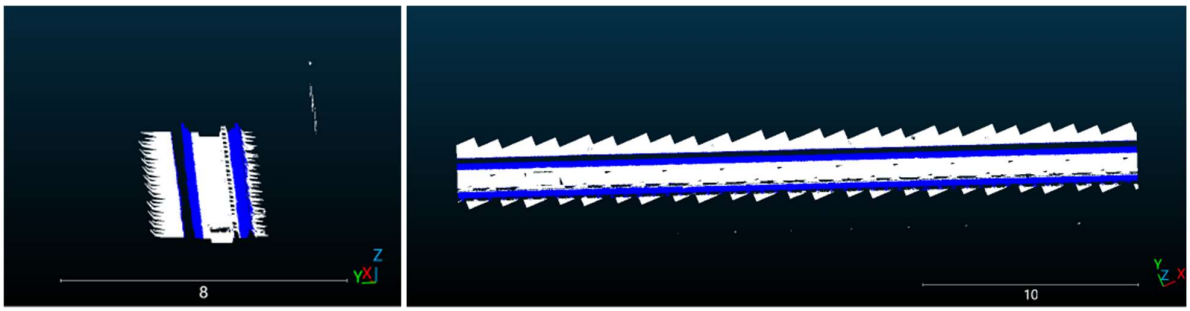


Figure 40. Circular After DTM filtration (Blue points are rail classified points in dataset)

Step 2 (Fine Detection via PCA): The initial dimensionality filter retained linear candidate points. PCA application shows strong and fast analysis to shrink data set, but it is not enough to extract rail points while there are 30-40% of false positive points. Thus, selected search radius and linearity threshold are the best value regarding the point elimination and process speed. Although the vast majority corresponded to true rail points (TP), the algorithm also retained linear clutter occupying the same elevation range, including suspended cables and pipe edges (FP). The quantitative results are summarized in Table 12,13 and Figures 41-43:

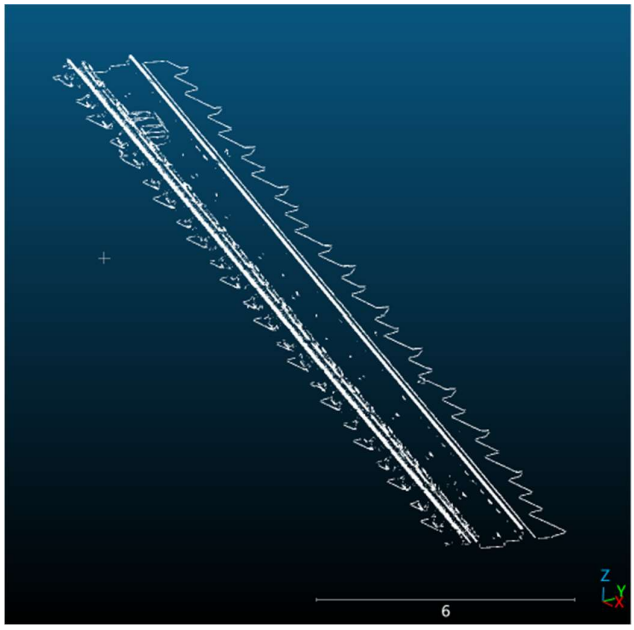


Figure 41. Circular Tunnel after PCA application

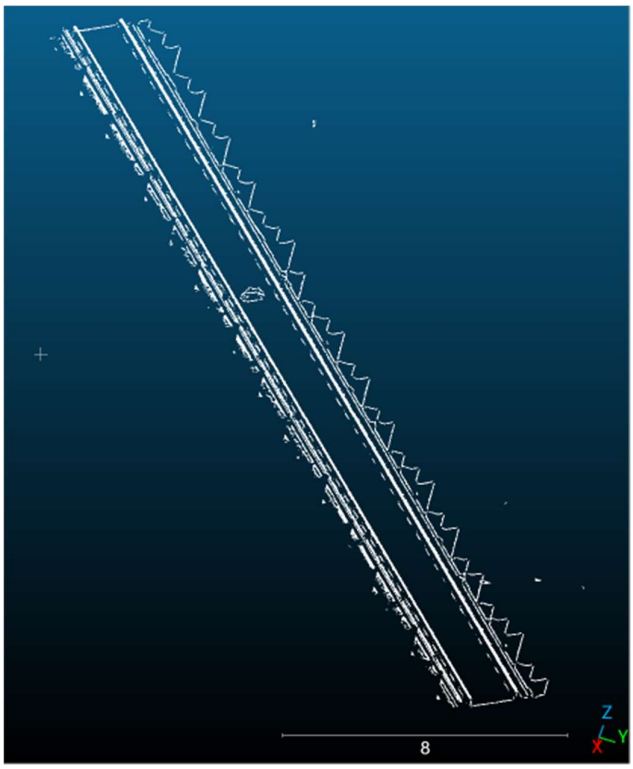


Figure 42. Quasi-rectangle Tunnel after PCA application

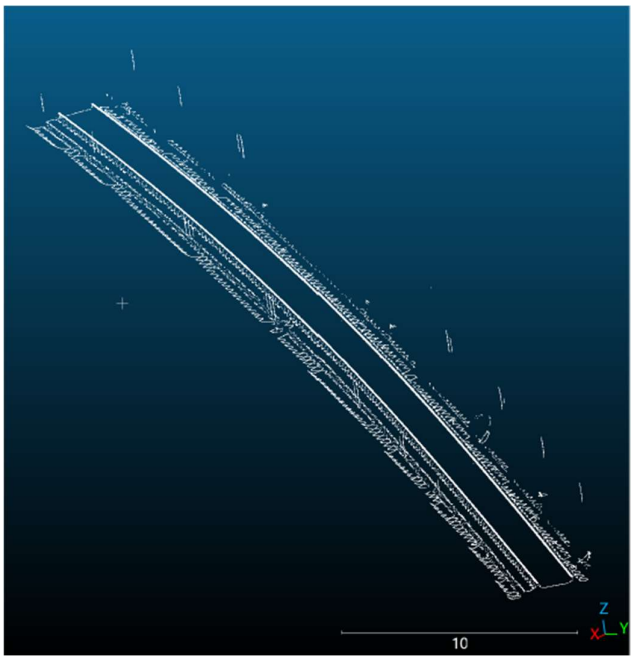


Figure 43. Horseshoe Tunnel after PCA application.

Table 12. Points and Process time after PCA linearity application

Tunnel Type	Step 1 Point No.	Voxel Downsampling Point No.	Point Drop %	Step2 Point No.	Process Time(S)	Point Drop %
Circular	6,261,359	1,146,001	82	153,821	28	87
Quasi-rectangle	7,394,495	1,160,466	84	180,055	22	84
Horseshoe	7,124,429	1,061,985	85	108,154	24	90

Table 13 Semantic Detection results in PCA

--- Step 2 - Fine Detection (PCA) ---	Circular	Quasi-rectangle	Horseshoe
Number of remaining points	153,821	180,055	108,154
TP	86,701	101,189	80,424
FP	67,120	78,866	27,730
FN	2,304,884	2,636,070	2,214,995
Precision	56.3%	56.2%	74.36%
Recall	3.6%	3.7%	3.5%
F1	6.8%	6.94%	6.69%

Step 3 (Geometric Filtering): The application of the geometric twin-check—enforcing standard-gauge distance, parallelism, and perpendicularity constraints within limited elevation—drastically purified the dataset. The sensivity analysis shows that high valid twin threshold meant to eliminate some portion of the rail points and very low values will increase the number of the ghost points but on the other hand the lower number means denser rail head. The bilateral geometric filter substantially reduced the number of false positives (Table 14, 15), and Figure 44 increasing the Precision to 80%. These results demonstrate the effectiveness of exploiting bilateral track symmetry to eliminate most of linear tunnel clutter.

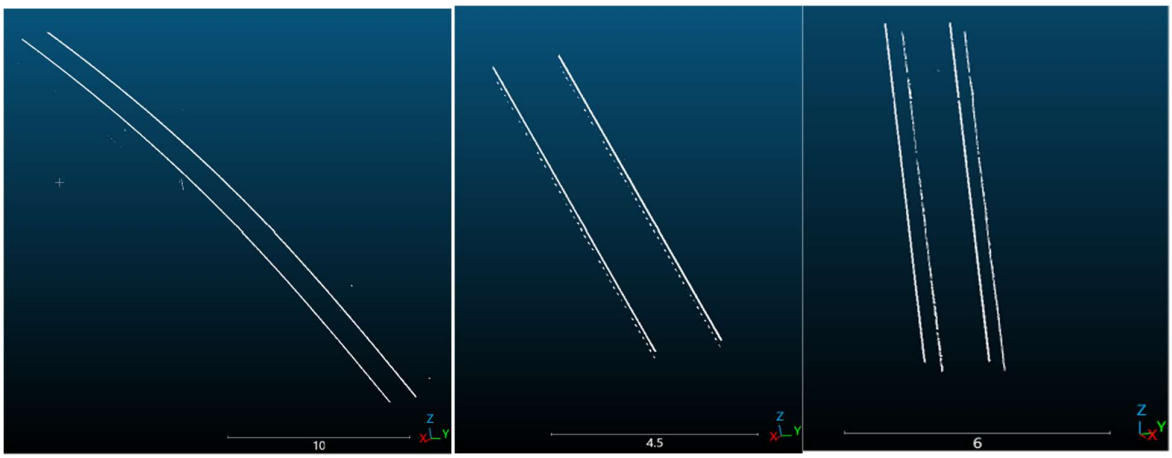


Figure 44. Horseshoe, Quasi-rectangle, circular tunnels after geometric filtration

Table 14.Points and Process time after geometric filtering

Tunnel Type	Step 2 Point No.	Step 3 Point No.	Point Drop %	Process Time(S)
Circular	153,821	55,041	64	342
Quasi-rectangle	180,055	45,925	74	262
Horseshoe	108,154	38,242	64	113

Table 15. Semantic Detection results in geometric filtering.

--- Step 3 - Geometric Filtering ---	Circular	Quasi-rectangle	Horseshoe
Number of remaining points	55,041	45,925	38,242
TP	44,712	45,925	38,242
FP	10,329	0	0
FN	2,346,873	2,691,334	2,257,177
Precision	81.23%	100%	100%
Recall	1.87%	1.68%	1.67%
F1	3.65%	3.3%	3.28%

Step 4 (Topology via ROR + DBSCAN): The final spatial density filtering (ROR) and DBSCAN clustering stages eliminated the remaining non rail head points. The sensitivity analysis shows their function dependency on the number of valid twins in step three. The quantitative results reported in Table 16, 17 shown in Figure 45 that the pipeline achieved a Precision of 100.0% for all three cases, this indicates that

topology filtering constraints are sufficient to remove all linear structures that satisfy the PCA criterion but do not satisfy the physical railway density and topology. Selected ROR and DBSCAN thresholds are working well with valid twin 40.

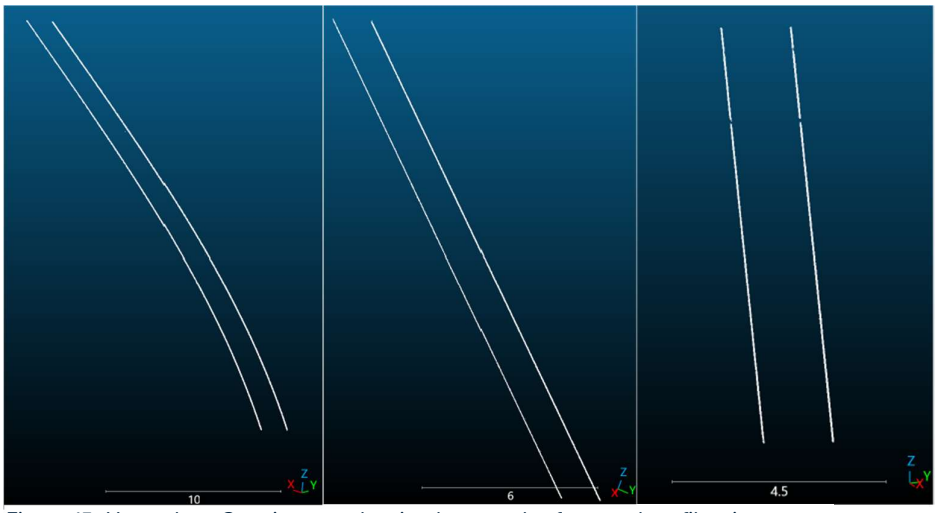


Figure 45. Horseshoe, Quasi-rectangle, circular tunnels after topology filtration

Table 16. Points and Process time after topology filtering

Tunnel Type	Step 3 Point No.	Step 4 Point No.	Point Drop %	Process Time(S)
Circular	55,041	37,436	31	2
Quasi-rectangle	45,925	45,925	0	1
Horseshoe	38,242	38,242	0	1

Table 17. Semantic Detection results after ROR&DBSCAN

--- Step 4 - Topology (ROR+DBSCAN) ---	Circular	Quasi-rectangle	Horseshoe
Pipeline points	37,436	42,331	38,242
TP	37,436	42,331	38,242
FP	0	0	0
FN	2,354,149	2,694,928	2,257,177
Precision	100%	100%	100%
Recall	1.57%	1.55%	1.67%
F1	3.08%	3.05%	3.28%

The output results for manual rail classification: To demonstrate the high FN been carried out without voxel downsampling in step two left and only PCA application left 1,081,963 linear points in 523s. Later Step three took almost 8 hours but left 422,556. ROR and DBSCAN applications with change in variables as the density changes left 257,600final rail points. Two accuracies have been calculated one for dataset rail class according to Table and the other one compared with the manual rail class according to Table 18 and Figure 46-48:

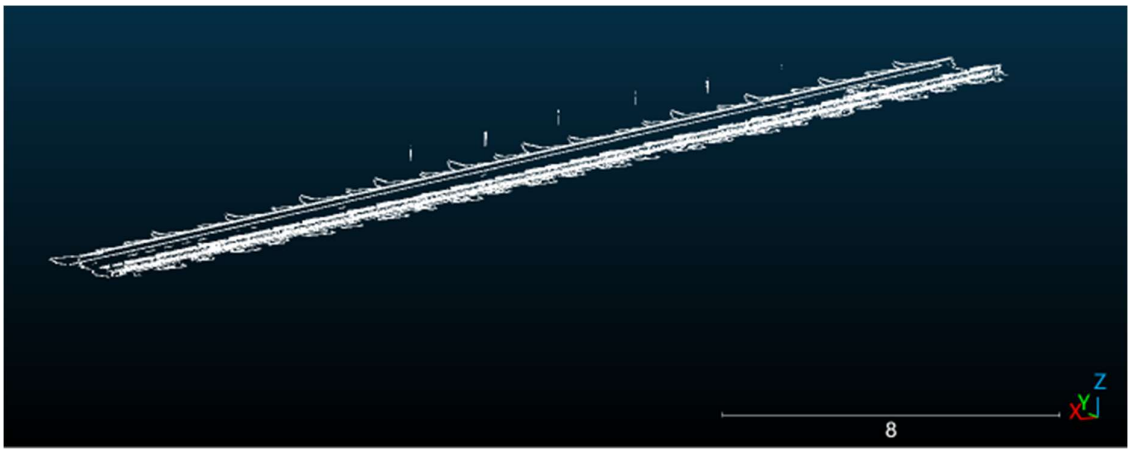


Figure 46. PCA step output without Voxel downsampling

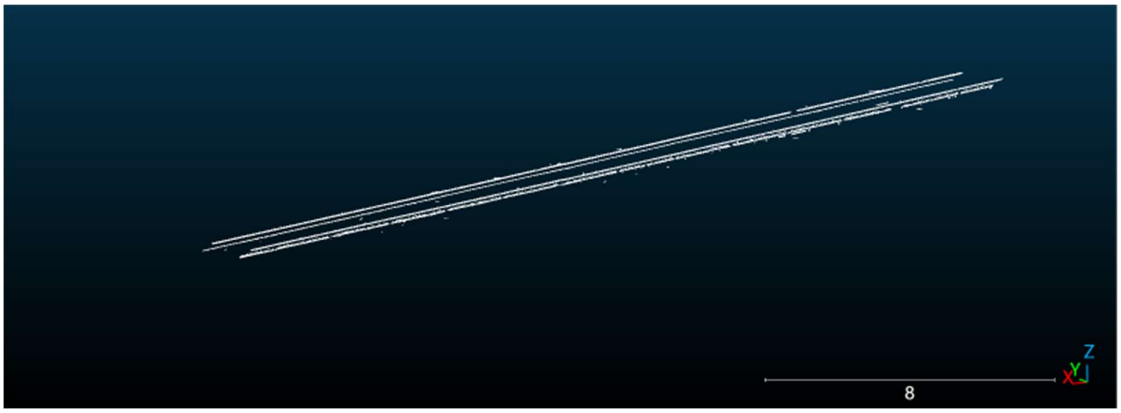


Figure 47. Geometric filtering step output without Voxel downsampling

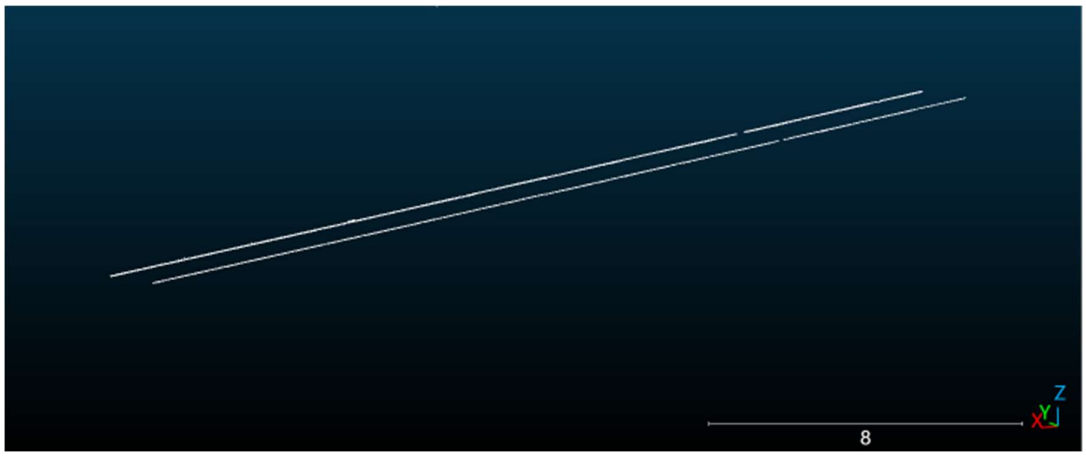


Figure 48. Topology filtering step output without Voxel downsampling

Table 18. KPI performance for manual rail classification scenarios in circular tunnel

	Variables	Dataset Rail class	Manual Rail class	Manual Rail with Voxel
Step 2 - Fine Detection (PCA)	**Track points**	**2,391,585**	**712,117**	**712,117**
	Pipeline points	1,081,963	1,081,963	153,821
	TP	569,125	424,528	73,814
	FP	512,838	657,435	80,007
	FN	1,822,460	287,589	638,303
	Precision	52.60%	39.24%	47.99%
	Recall	23.80%	59.61%	10.37%
	F1	32.77%	47.33%	17.05%
Step 3 - Geometric Filtering	Pipeline points	422,556	422,556	55,041
	TP	300,432	266,132	37,462
	FP	122,124	156,424	17,579
	FN	2,091,153	445,985	674,655
	Precision	71.10%	62.98%	68.06%
	Recall	12.56%	37.37%	5.26%
	F1	21.35%	46.91%	9.77%
Step 4 - Topology (ROR+DBSCAN)	Pipeline points	257,600	257,600	37,436
	TP	257,600	257,600	37,436
	FP	0	0	0
	FN	2,133,985	454,517	674,681
	Precision	100.00%	100.00%	100.00%
	Recall	10.77%	36.17%	5.26%
	F1	19.45%	53.13%	9.99%

7.3. Generative Raw Centerline Reconstruction Results

Step 5 — Raw Midpoints Generation: For each polished rail point, valid twins passing the gauge distance and perpendicularity criteria within a 1.6 m search sphere were identified. Points with at least 5 valid twins contributed to a raw midpoint, computed as the mean of p1 and the twin centroid. The raw centerline raw_center_array contains a dense cloud of midpoints distributed along the track axis Figure 49 and Table 19.

Table 19. Points and Process time after mid-points generation

Tunnel Type	Step 4 Point No.	Step 5 Mid-Point No.	Process Time(S)
Circular	37,436	37,436	19
Quasi-rectangle	45,925	42,331	36
Horseshoe	38,242	38,242	23

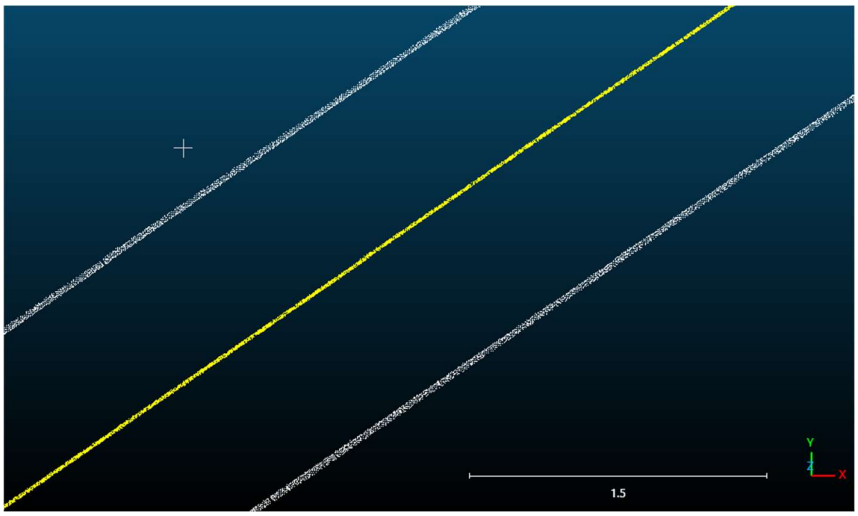


Figure 49. Raw centerline generated from mean of the rail pairs showing in yellow.

Step 6 — Spherical Snap Smoothing: Voxel downsampling at 0.5 m produced rough anchor points. For each anchor, the spatial median of all raw midpoints within 0.625 m was computed as the final centerline position. Points with fewer than 3 raw midpoints in the search sphere were discarded to avoid poorly constrained positions. The output (KPI_centerlines.las) contains the final centerline points at approximately 0.5 m spacing, each annotated with an DQI score in the intensity field (Figure 50-52).

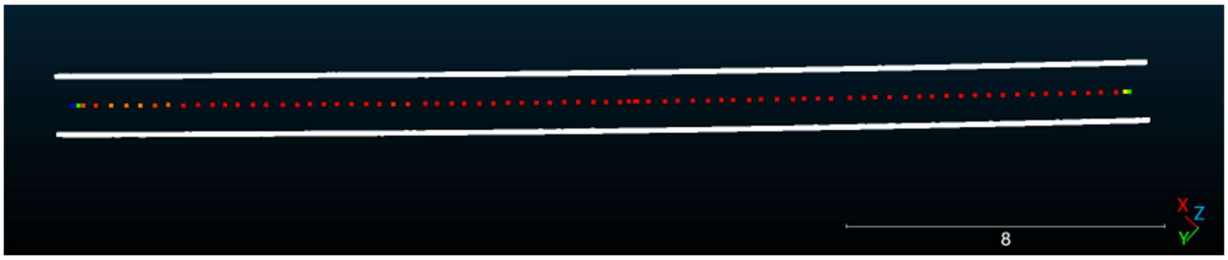


Figure 50. Horseshoe-generated centerline with DQI score embedded as scalar field.

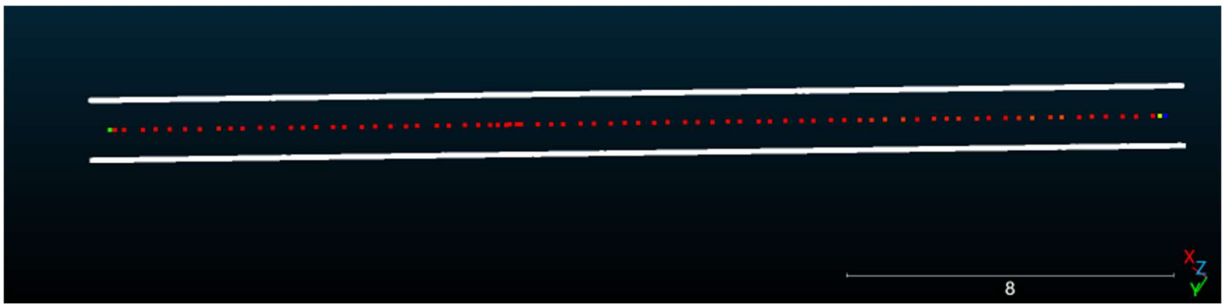


Figure 51. Quasi-rectangle generated centerline with DQI score embedded as scalar field.

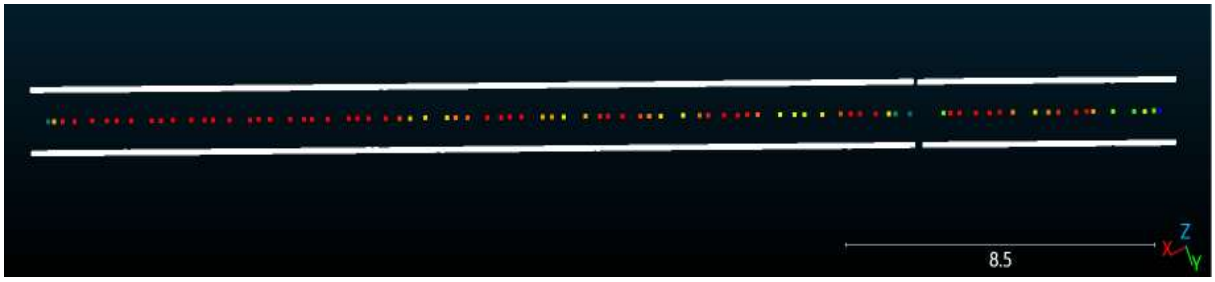


Figure 52. Circular generated centerline with DQI score embedded as scalar field KPI result.

7.4. Statistical Analysis and KPIs Results

Tier 1: Semantic Detection Results

By evaluating each major phase of the extraction pipeline against the STSD semantic ground-truth mask (Class 1), the progressive filtration of the point cloud was explicitly quantified.

When evaluating traditional deep-learning semantic segmentation models, high Recall is the primary objective, as the algorithm attempts to classify every single point belonging to the physical rail head structure. However, the metric logs from this study reveal that Recall stabilized at approximately 1.6% across the final stages.

When interpreting the semantic metrics, Recall requires particular caution. The final Recall values are low because the proposed methodology intentionally reduces the

point set through voxel downsampling, local dimensionality filtering, geometric filtering, and topology filtering. At the same time, the comparison with the manually interpreted rail points indicates that the STSD reference labels may also influence the measured Recall. The available experiments therefore demonstrate that both selective extraction and reference-label characteristics can contribute to the low Recall, but their individual contributions cannot be quantitatively separated with the present analysis.

For this reason, Recall should not be interpreted in the same way as it would be for a point-wise semantic segmentation method whose objective is to classify every rail point. The proposed pipeline instead performs geometric skeletonization: it seeks a sufficiently pure and spatially representative subset from which the railway centerline can be reconstructed. Accordingly, Precision and the subsequent geometric/topological continuity metrics provide a more direct assessment of the intended function of the method, while Recall calculated without downsampling in the second step and a clean rail class according to Table 18 the pipeline were able to obtain a higher Recall of %36.17 and same process without downsample but the data set defined class show the recall of % 10.77.

The analysis indicates that the low Recall is influenced by both the selective decimation and geometric filtering applied by the proposed methodology and differences between the STSD reference labels and the manually interpreted rail points. However, the present analysis does not allow the relative contribution of these two effects to be quantified separately. Therefore, the low Recall should not be attributed exclusively to either the extraction strategy or the reference-label quality.

Tier 2: Geometric Topological Continuity Results

To validate the lateral geometric and Topological Continuity of the extracted infrastructure, the pipeline evaluated the physical gauge distance at each generated centerline node. By utilizing a 1.2-meter search radius and applying another DBSCAN clustering, the algorithm successfully isolated the left and right rail heads independent of the centerline.

As detailed in Table 20, the pipeline demonstrated high modeling accuracy. While the gauge analysis validates lateral accuracy, the structural usability of the 3D model depends on its unbroken longitudinal continuity. The framework measured the spatial frequency of the generated coordinates to verify that the track trajectory was

completely recovered without fracturing. The pipeline showed acceptable continuity of the rail.

Over a total evaluated track length, the algorithm generated continuous centerline nodes every 0.30-0.40m. Most importantly, the evaluation recorded zero macroscopic gaps exceeding 1 meter. These results demonstrate that the algorithm successfully maintained continuous tracking of the track geometry across the evaluated dataset, bridging areas of localized occlusion without losing the longitudinal trajectory.

Table 20. Geometric KPI

Gauge Compliance Analysis (DBSCAN L/R split)	Circular	Quasi-rectangle	Horseshoe
Gauge compliance	100%	100%	100.00%
Centerline points	83	79	81
Track length	30.5 m	26.0 m	26.5 m
Systematic bias (mu)	14.1 mm	13.3 mm	12.1mm
Std deviation	1.7 mm	1.3 mm	1.6mm
Center line mean spacing	37.2 cm	33.3 cm	33.1 cm
centerline Gaps > 1.0 m	0	0	0

Tier 3: DQI and Statistical Validation Results

Evaluating the DQI scalar distribution across the exported trajectories confirmed that the majority of extracted nodes exhibited strong local data support.

As detailed in Table 21, Mean confidence scores reached 92.80% ($\sigma = 10.60\%$) in the circular tunnel, 98.60% ($\sigma = 4.70\%$) in the quasi-rectangular tunnel, and 98.20% ($\sigma = 6.60\%$) in the horseshoe tunnel. Nodes falling within the high-confidence tier ($\geq 80\%$) accounted for 90.4%, 98.7%, and 96.3% of the generated trajectories, respectively. Zero nodes fell below the 40% threshold.

Pearson correlation analysis yielded statistically significant positive coefficients across all three cross-sections:

Table 21. Pearson correlation and DQI table

Confidence Score Distribution	Circular	Quasi-rectangle	Horseshoe
Mean	92.80%	98.60%	98.20%
Std deviation	10.60%	4.70%	6.6%
Min / Max	52.7% / 100.0%	65.9% / 100.0%	56.0% / 100.0%
Points > 80% HIGH confidence	75 (90.4%)	78 (98.7%)	78 (96.3%)
Points 40-80% MODERATE	8 (9.6%)	1 (1.3%)	3 (3.7%)
Points < 40% flag for review	0 (0.0%)	0 (0.0%)	0 (0.0%)
Pearson r (confidence vs polished rail density)	0.913 (p=0.0000, n=83)	0.758 (p=0.0000, n=79)	0.734 (p=0.0000, n=81)

This consistent, strong correlation supports the use of the DQI as an indicator of relative local data support across the tested tunnel morphologies.

7.5. Computational Analysis

The execution runtime and resource consumption of the six-step automated extraction pipeline were measured on a consumer workstation equipped with an Intel Core i5-8350U CPU 1.70 GHz, Intel UHD Graphics (4 GB), 8 GB DDR4 RAM, and 50 GB storage.

The end-to-end extraction across a 30-meter tunnel segment comprising ~24 million raw points required an empirical execution time of approximately 10 to 11 minutes. During execution, system CPU utilization operated at 100%, and RAM consumption peaked at approximately 2.0 GB under a memory-bounded 20,000-point batching architecture.

A step-by-step breakdown of point reduction and individual execution times across the circular, quasi-rectangular, and horseshoe test datasets is presented in Table 17.

And to assess the practical role of spatial decimation and observe empirical scaling under high data volumes, two comparative benchmark tests were executed on the Circular tunnel dataset (30 m segment, 23,685,473 raw points):

Table 22. Datasets computational timing

Processing Step	Circular Profile (Pts / Time)	Circular profile not downsapled experiment	Quasi-Rectangular Profile (Pts / Time)	Horseshoe Profile (Pts / Time)
Raw Input Cloud	23,685,473 pts / —	23,685,473 pts / —	24,229,599 pts / —	24,229,596 pts / —
Step 1: DTM Coarse Filter	6,261,359 pts / 16 s	6,261,359 pts / 16 s	7,394,495 pts / 16 s	7,124,429 pts / 17 s
Step 2: 1 cm Voxel	1,146,001 pts / <2 s	Bypassed (6,261,359 pts)	1,160,466 pts / <2 s	1,061,985 pts / <2 s
Step 2: PCA Linearity r=0.1	153,821 pts / 28 s	1,081,963 pts / 523 s	180,055 pts / 22 s	108,154 pts / 24 s
Step 3: Geometric Twin Filtering	55,041 pts / 342 s	422,556 pts / 32,400 s (~9h)	45,925 pts / 262 s	38,242 pts / 113 s
Step 4: Topology (ROR + DBSCAN)	37,436 pts / 1s	257,600 pts / 195 s	45,925 pts / 1 s	38,242 pts / 1 s
Step 5 & 6: Midpoints & Spherical Snap	37,436 midpoints / 19s	205,613 midpoints / 2,227.17 s	42,331 midpoints / 36 s	38,242 midpoints / 23 s
Total Pipeline Runtime	~408 s (~6.5 min)	~38,028.80 s (~10.5 hr)	~337 s (~5.5 min)	~178 s (~3 min)

As detailed in Table 21:

By passing the 1 cm voxel downsampling increased the candidate array entering Step 3 from 146,001 points to 1,081,963 points.

In Step 2 (PCA Linearity), processing the downsampled array required 28s compared to 539.92 s on the raw array, yielding a 19.3× speedup.

This increase in candidate volume scaled Step 3 execution time from 342 s to 35,042.99 s (~9.73 hours) across 109 memory batches, demonstrating a 70-80× speedup when decimation was applied.

Step 4 (Topology Filtering) achieved a speedup (1.00 s vs. 195.26 s) due to the significantly smaller input point cloud.

Step 5 runtime expanded from ~3 s to 2,227.17 s (~37.12 minutes) to process 257,600 input points and generate 205,613 midpoints, representing a 742.4× speedup for the decimated pipeline.

Across the complete workflow, applying 1 cm voxel decimation reduced total runtime from 10.56 hours down to 6 minutes, achieving an overall high speed up from 13 point per second in step 3 to 109 points per second while both configurations converged to an identical 100.0% final semantic precision and maintained continuous centerline extraction.

8. Discussion

8.1. Interpretation of Extraction Performance and Geometric Fidelity

The results indicate that the proposed pipeline is effective at producing a geometrically selective rail representation across the three tested tunnel morphologies. The main strength of the method is not exhaustive point-wise recovery, but the progressive removal of geometrically inconsistent and topologically unsupported points while retaining a subset sufficient for centerline reconstruction. The high final Precision observed in the evaluated datasets therefore indicates strong semantic purity of the extracted rail subset under the tested conditions.

The low Recall should be interpreted differently from the Recall of a conventional semantic segmentation method. The proposed pipeline intentionally performs point-cloud decimation and selective geometric extraction, so recovery of every reference rail point is neither the objective nor a necessary condition for centerline generation. In addition, the manual-label analysis indicates that characteristics of the STSD reference labels may influence the measured Recall. The present experiments do not allow the contribution of selective extraction and reference-label limitations to be separated quantitatively.

From a geometric perspective, the absence of macroscopic gaps exceeding 1.0 m in the evaluated centerlines indicates that the midpoint generation and Spherical Snap procedure maintained longitudinal connectivity under the tested conditions. This result supports the suitability of the extracted point subset for centerline

reconstruction, but it should not be interpreted as a general guarantee of continuity under unseen tunnel geometries, larger occlusions, or different point densities.

8.2. Comparison with Other Methods

When compared against established rail extraction methodologies in the literature, the proposed pipeline demonstrates distinct technical trade-offs a brief review of proposed method and other methods are in Table 21:

Table 23. Methodology comparison

Paper Name	Data Type Information	Rail Classification Method	Rail classification accuracy	Centerline Calculation Method
This thesis	MLS 36,000 pts/m^2 on a trolley, variable speed	Covariance-based PCA linearity filtering, bilateral 3D twin-check (Distance, parallelism, perpendicularity, Radius Outlier Removal, and DBSCAN clustering	100% Precision; 0 rail gaps; Pearson r = 0.544-0.732 validating DQI against ground-truth rail density.	Generative twin midpoint calculation, and Spherical Snap algorithm utilizing local spatial medians coupled with embedded DQI scoring
S. Oude Elberink	MLS, 40-8,000 pts/m^2.	Coarse detection by relative height; fine detection by local line direction using RANSAC and point counts in a parallel buffer.	Not reported	Data-driven: projects top rail points across track. Model-driven: MCMC fitting of a 3D rail pair model, smoothed by Fourier series.
Z. Nagy	MLS, speed 40–60 km/h, surface density 10,000–12,000 pts/m^2	Segmentation by scan angle metadata, Euclidean clustering for profiles, and geometric filtering based on rail head morphology.	Mean absolute error < 10 mm (3D spatial error: 5.9 mm for left track, 9.1 mm for right track).	Resampling left and right rails to equal vertices and computing the geometric midpoint (tween curve method).
A. Kononen	MLS, driving speed 80 km/h, density 300–4500 pts/m^2	Thresholding local linearity and density using PCA, followed by RANSAC to detect collinear pairs separated by track gauge.	0.3 cm to 0.8 cm on non-intersecting rails; 0.7 cm on measurement track intersections.	Approximates centers from paired points, applies polynomial smoothing, and fits rectangular rail top templates perpendicular to paths.
A. Karunathilake	MLS on a trolley, speed 20 km/h, density 1600 pts/m^2	Evaluating point cloud density variations in vertical slices and applying logic-based pattern detection to find peaks.	100% structure detection accuracy in test datasets.	Points are projected to 2D; ICP algorithm matches ideal rail head shapes to extract GC positions, which form the track centerlines.

Paper Name	Data Type Information	Rail Classification Method	Rail classification accuracy	Centerline Calculation Method
R. Beger	Airborne LiDAR and high resolution ortho-imagery (5 cm GSD).	Height above ground (>0.15 m, <0.4 m) and slope gradient (<15) thresholds, followed by an adapted RANSAC algorithm.	Recall rate 94.3%–97.9% and precision 97.5%–99.8% (at 30 cm tolerance).	RANSAC estimation for parallel lines/arcs at gauge distance, followed by Total Least Squares (TLS) to connect segments.

The contribution of each method and their limitations are reported in Table 22:

Table 24. Methodologies contribution and limitation

Paper Name	Advantages / Contributions	Limitation
This Thesis	Replaces RANSAC with local PCA covariance filtering; introduces a bilateral topology filter (ROR + DBSCAN) that achieved 100% precision in the evaluated datasets; uses Spherical Snap spatial medians for centerline generation; embeds a point-level DQI score.	Parameters may require recalibration for different scanner configurations, acquisition conditions, or substantially point densities lower than one point per cm. Absolute geometric accuracy has not been validated against an independent survey reference.
S. Oude Elberink	Can handle local data gaps (up to 8 m) and extract centerlines for up to four parallel tracks successfully with centimeter accuracy.	Cannot bridge data gaps > 8 m; fails to detect parallel tracks if ground points are absent; reference data had systematic offsets.
Z. Nagy	Achieves high geometric fidelity without manual intervention; synchronous measurement frame decouples positioning errors.	Turnout areas and specific components interfere with point selection; reference datasets should have systematic biases.
A. Kononen	Real-time performance on two tracks; models multi-track environments in a single pass; avoids heuristics needing training data.	Fails on guards, intersections, or discontinuities; assumes continuous vertical rail; fails under full occlusion.
A. Karunathilake	Robust in complex situations (turnouts/crossings); does not require strict IMU/GNSS alignment during extraction.	Discrepancies between detected and actual lengths for lengthy structures; shape irregularities at turnouts caused right GC false estimation.
R. Beger	High level of automation, independent of manual training samples; fuses sources to maximize completeness and accuracy.	Less accurate in stations and construction areas; obstacles from parked trains; turnout sections not satisfactorily connected.

8.3. Robustness and Parameter Dependence

The results across circular, quasi-rectangular, and horseshoe tunnel profiles indicate that the local PCA-based representation can operate across different surrounding tunnel morphologies without requiring a different cross-sectional model for each profile. This robustness is related to the local nature of the covariance analysis, which evaluates neighborhood dimensionality around individual candidate points rather than relying on a global tunnel cross-section.

However, this does not imply complete invariance to acquisition conditions. Several parameters are directly linked to the physical and statistical characteristics of the input data. In particular, the 1.40–1.47 m gauge envelope assumes standard-gauge railway geometry, while the PCA neighborhood radius, linearity threshold, valid-twin threshold, ROR density requirement, DBSCAN parameters, and final voxel size depend on the spatial scale of the point cloud.

The 1 cm voxel downsampling reduces the computational burden and standardizes the point representation used by subsequent stages, but it does not make the pipeline independent of point cloud density. If the original point cloud density is substantially lower than the selected voxel resolution, important geometric information may be lost, and the density-based thresholds may no longer provide equivalent behavior.

Therefore, the selected parameters should be regarded as empirically validated for the STSD datasets evaluated in this thesis rather than as universally transferable values. Changes in rail gauge, scanner configuration, acquisition speed, point cloud density, occlusion characteristics, or tunnel morphology may require recalibration. Validation on heterogeneous datasets is therefore necessary before the same parameter set can be considered generally applicable.

8.4. Computational Scalability and Feasibility

Theoretical Complexity and Algorithmic Bottlenecks The empirical runtime results demonstrate that the pipeline's computational cost is heavily concentrated in Step 3 (Geometric Filtering), which accounts for 80% to 90% of total processing time.

- **Linear Stages ($O(N)$):** Step 1 (DTM grid binning) and the voxel downsampling in Step 2 scale linearly with input size N, allowing millions of raw points to be filtered in under 30 seconds.

- **Neighborhood Search Bottleneck ($O(M \log M)$):** In Step 3 (Geometric Filtering) and Step 5 (Midpoint Generation), 3D KDTree range searches query spatial neighborhoods across candidate sets of size M. In Step 3, evaluating distance windows (1.40--1.47 m), eigenvector parallelism (> 0.95), coplanarity (< 0.10 m), and vector perpendicularity (< 0.15) requires traversing dense local coordinate clusters.
- **The Decimation Strategy**: The 10.56-hour non-downsampled experiment demonstrates the computational necessity of Step 2 voxel downsampling. Retaining full-density laser scans increased candidate points by a factor of ~7.4× entering Step 3 (1.08M vs. 145k points), which in turn caused a 68.8× increase in Step 3 execution time (35,043 s vs. 509 s). Reducing the candidate array M by over 85% via 1 cm voxel downsampling resulted in an approximately 67.4× reduction in measured runtime while producing the same final semantic precision and maintaining continuous centerline extraction in the tested comparison. Memory Safeguards and Hardware Constraints During the 10.56-hour full-resolution run, system RAM remained strictly bounded under ~2.0 GB because the KDTree queries were executed across 109 consecutive batches of 10,000 points each. This confirmed that chunked batching prevents C-level Memory Error allocation crashes during range queries on high-density point clouds.

Multi-Kilometer Scaling Projections Because individual 30- to 50-meter tunnel segments are decoupled during Steps 1 through 4, the workflow can be parallelized across multi-core processors without global data dependencies:

- **10 km Corridor (~8 billion raw points):** assuming the measured processing characteristics remain approximately representative and that the tunnel can be processed as independent segments, considering the pipeline will select around 35-40million candidates to step3, corresponding single-threaded computational demand would be approximately more than 35-40h CPU hours. Under an idealized distribution across 16 cores, this would correspond to approximately 3-4 hours. These values are projections and were not experimentally demonstrated on a 10 km dataset.
- **100 km Regional Network (~80 billion raw points):** under the same scaling assumptions, processing on a big scale cluster could require approximately 30 hours. This value is an extrapolation from the measured segment-level

runtimes and should not be interpreted as experimentally demonstrated performance.

Operational Feasibility: (Offline Post-Processing vs. Real-Time Deployment) In its current architecture, the pipeline serves as an offline point cloud processing framework, prioritizing high sematic precision, topological continuity, and DQI metadata generation rather than real-time processing.

Deploying the framework onboard inspection vehicles operating at diagnostic speeds (60--80 km/h, requiring a 30-meter block to be processed within 1.35 seconds) requires two technical developments:

1. Re-implementing Step 3 spatial queries in C++/CUDA to leverage GPU parallelization for KDTree neighbor searching.
2. Applying DTM and PCA filters directly to streaming 2D profiler scanlines prior to global 3D assembly.

8.5. Engineering Interpretation of the DQI

The empirical calculation demonstrates a consistent, statistically significant positive correlation ($r = 0.544$ to $0.732, p < 0.001$) between the DQI and ground-truth rail point cloud density across diverse tunnel morphologies. This supports the interpretation that the DQI reflects variations in local point support and scan coverage and perspective occlusion to end users. However, the scope of this score must be interpreted with scientific rigor:

Data Support vs. Absolute Accuracy: While the DQI reliably captures the relative amount of geometric data available to anchor the spatial median, correlation with point cloud density alone does not directly prove absolute 3D positional accuracy, deformation monitoring precision, or compliance with strict millimeter-level surveying tolerances.

Status of Thresholds: The proposed classification tiers ($\geq 80\%$, 40--80%, $< 40\%$) are heuristic interpretation bands introduced in this study to facilitate preliminary data triage. They are not derived from established industrial standards and should not be used as formal engineering acceptance thresholds without prior calibration.

Need for Geodetic Benchmarking: Formal association of discrete DQI values with certified engineering accuracy classes requires future co-registration and benchmarking against independent survey-grade total station or laser tracker

measurements. Accordingly, DQI should currently be interpreted as a relative data-support indicator rather than as a direct estimate of positional accuracy.

8.6. Applicability to Metro Tunnels and Mining

While evaluated primarily on urban subway infrastructure (STSD dataset), the underlying geometric topology principles dictate distinct applicability boundaries across different subterranean domains:

- **Metropolitan Rail Tunnels**: The pipeline is particularly suited based on the STSD cases evaluated to urban transit tunnels characterized by standardized steel rail profiles, fixed track gauges (1.435 m), and dense, continuous MLS acquisitions. It effectively navigates confined space clutter such as power tracks, drainage pipes, and catenary cables.
- **Underground Mining Environments**: Applying the pipeline to subterranean mining haulage drifts presents unique challenges that are not addressed in other papers. Mine tracks frequently exhibit irregular gauges, temporary narrow-gauge tie systems, heavy mud/ballast accumulation obscuring the rail foot, and steep gradients. While the rigid 1.40--1.47 m twin-check window in Step 3 requires adaptive calibration to accommodate non-standard mining rail dimensions.

The automated extraction of geometrically reconstructed centerlines accompanied by point-level meta quality metrics has the potential to reduce a critical bottleneck in Scan-to-BIM workflows. Currently, converting raw subterranean MLS point clouds into parametric BIM alignment objects requires labor-intensive manual vectorization.

The generated centerline and associated DQI metadata could potentially be integrated into OpenTrack, IFC-based alignment workflows, or GIS infrastructure databases, subject to validation of the required data formats and engineering accuracy. By projecting the DQI confidence heatmap directly onto the 3D parametric alignment, infrastructure asset managers receive an automated data quality certificate alongside the geometric model, accelerating digital twin creation for predictive tunnel maintenance.

8.7. Limitations

Three limitations of the current implementation were identified. First, several key parameters, including the forbidden zone bounds, the rail height window, were tuned specifically for the STSD dataset. Their transferability to datasets acquired using different scanners or at different acquisition speeds has not yet been validated. A density-adaptive parameterization framework for cases with point cloud density lower than 1cm^2 would improve the generalizability of the proposed methodology.

Second, the pipeline does not include an explicit switch detection. In switch zones, the gauge constraint is temporarily violated, and the perpendicularity condition breaks down, leading to local point cloud density. This is the primary geometric failure mode of the current implementation of the rail, but the centerline would be still generated in that area, and it needs future studies to see the exact output.

Third, the pipeline processes point clouds as static snapshots and does not incorporate any temporal or multi-epoch logic. Long-term deformation monitoring based on comparing centerlines extracted from surveys acquired at different epochs, would require an additional registration and change-detection layer that is beyond the scope of this thesis.

9. Conclusion and Future Work

9.1. Conclusion

This thesis presented a fully automated, deterministic data-driven pipeline for extracting 3D railway track centerlines and quantifying point-level relative local data support directly from dense Mobile Laser Scanning (MLS) point clouds in GNSS-denied subway environments.

- **Semantic Purity via Spatial Topology**: By replacing standard RANSAC line fitting and rigid parametric template matching with localized Principal Component Analysis (PCA) covariance filtering, bilateral gauge verification (1.40–1.47 m window, parallelism > 0.95, perpendicularity < 0.15), and dual-stage topology cleaning (ROR + DBSCAN). The pipeline achieved 100.0% semantic precision for the evaluated circular, horseshoe, and quasi-rectangular tunnel sections after topology filtering, with no false-positive rail detections observed in these tests.

- **Robust Curve Trajectory Generation**: The developed Spherical Snap algorithm, utilizing localized spatial medians within a 0.625 m radial sphere, reduced sensitivity to lateral curve drift and density-skewed bias with no macroscopic gaps exceeding 1.0 m observed across the evaluated 82 m of tunnel alignment.
- **Deterministic Quality Metadata**: The embedded Data Quality Indicator (DQI) successfully integrates point-level spatial reliability metadata into standard .las intensity fields. Demonstrating consistent positive correlations ($r = 0.544$ to $0.732, p < 0.001$) against ground-truth rail density, the DQI provides an automated, explainable indicator of local geometric data support to assist downstream Scan-to-BIM triage.

9.2. Answers to the Research Questions

Table 25. Answers to research questions

Question	Contribution	Answer
How can raw, unstructured Mobile Laser Scanning (MLS) datasets characterized by severe density variations and high-frequency auxiliary noise be automatically segmented to isolate standardized track infrastructure?	Evaluated 30 m sections of circular, quasi-rectangular, and horseshoe tunnels from the STSD dataset (densities >36,000 pts/m2) containing auxiliary clutter.	Addressed by the 10th-percentile DTM height grid filter (0.5 *0.5m) that eliminates lining/overhead clutter, followed by 1 cm voxel downsampling and localized PCA linearity filtering (r = 0.10 m, L > 0.60)
How effectively can a spatial topology framework based on local dimensionality and known railway geometric constraints support robust rail extraction as an alternative to RANSAC-based line fitting?	Replaced RANSAC loops with local PCA covariance estimation and a tri-criterion bilateral twin check, followed by ROR and DBSCAN.	The framework achieved 100.0% semantic Precision across the three evaluated tunnel cross-sections. Its local, point-based PCA formulation was considered more compatible with the proposed pipeline architecture than explicit iterative RANSAC model fitting; however, a controlled comparison of execution time and memory usage between the two approaches was outside the scope of this study.
How can generative 3D midpoint data be smoothed along highly curved tunnel alignments without losing geometric precision or distorting the physical path of the track?	Formulated the Spherical Snap Smoothing algorithm using localized spatial medians within dynamic radial search envelopes, evaluated across curved and straight subway alignment.	By calculating local spatial medians within a 0.625 m search sphere around 0.50 m voxel anchor nodes, preventing lateral curve drift and bridging perspective shadow occlusions with zero macroscopic gaps exceeding 1.0 m.

Question	Contribution	Answer
In what way can Data Quality Indicator (DQI) and local density metrics be embedded within the extracted spatial coordinates to provide an automated indicator of relative local data support and confidence, while identifying the requirements for future engineering-grade validation?	Embed DQI confidence scores into the Intensity field of exported .las files and validated them against ground-truth rail density using Pearson correlation.	By normalizing local midpoint counts within a 0.625m search sphere against an 85th-percentile optimum density benchmark, computing a deterministic 0–100% score, and injecting it into the 16-bit Intensity scalar field. The resulting metric functions as indicator of local point support and confidence across the tested tunnel cross-sections. Independent survey grade measurements would be required before relating DQI values to engineering accuracy classes.

9.3. Future Trajectories

To address the current limitations of the framework and advance the methodology toward industrial deployment, seven primary research directions are identified:

Automatic Parameter Optimization

To replace empirical manual tuning, future iterations should incorporate automated optimization frameworks. By defining a target loss function based on local point cloud density and spatial continuity, key hyperparameters, including PCA linearity bounds, and twin-check parameters, can be automatically calibrated to achieve optimal performance without human intervention.

Learning-Based Parameter Selection

Beyond mathematical optimization, machine learning models (e.g., Random Forests or lightweight neural networks) can be trained to predict optimal filtering thresholds directly from local point cloud descriptors. By evaluating neighborhood geometric features (such as planarity, sphericity, and intensity variance) across small input tiles, the pipeline can dynamically adjust parameter settings in real time to adapt to varying scanner speeds or surface reflectivity.

Integration with BIM and Digital Twin Platforms

To bridge the gap between point cloud processing and asset management, automated export modules should formalize the connection between centerline and BIM platforms. Ingesting the 3D centerline and its embedded Data Quality Indicator (DQI) scalar field directly into Industry Foundation Classes (IFC Alignment standards) will

enable the automatic generation of parametric digital twins equipped with spatial reliability metadata.

Hybrid Integration with Deep Learning Semantic Segmentation Networks

While the proposed pipeline relies on data-driven geometric topology, combining it with pre-trained 3D semantic segmentation networks (e.g., PointNet++ or MFPNet) offer significant synergies. A deep learning model can handle coarse initial classification to isolate the general rail corridor, allowing the geometric topology pipeline to focus strictly on fine-scale, high-precision centerline extraction and DQI confidence scoring.

Algorithmic Extension to Switches, Turnouts, and Crossings

The primary geometric failure mode of the current pipeline can occurs at track divergences, where standard gauge and vector perpendicularity conditions break down. Developing a dedicated switch and turnout module—utilizing graph-based connectivity logic to detect branching nodes and apply localized, adaptive gauge constraints—will extend automated centerline generation through complex railway junctions.

Validation Across Heterogeneous Datasets

To verify the generalizability of the pipeline beyond the STSD dataset, future testing should evaluate point clouds captured by different Mobile Mapping Systems operating under varying scanning speeds, ambient conditions, and track types (such as ballast less slab tracks vs. traditional ballasted beds).

Real-Time GPU-Accelerated Implementation

To transition the current offline Python implementation into a real-time diagnostic system capable of operating onboard inspection vehicles at operational speeds (80 km/h), the core spatial algorithms must be refactored. Re-implementing Step 3 (Geometric Filtering) in C++ using CUDA acceleration on dedicated mobile GPUs will allow KDTree range queries to be processed in real time directly on live 2D profiler stream data.

References

[1] Tong, J.; Ji, M.; Song, P.; Chen, Q.; Chen, C. MFPNet: A Semantic Segmentation Network for Regular Tunnel Point Clouds Based on Multi-Scale Feature Perception. Sensors 2026, 26, 848. https://doi.org/10.3390/s26030848

[2] Junxin Chen a, Xiaojie Yu a, Shichang Liu b, Tao Chen c, Wei Wang d e, Gwanggil Jeon f, Benguo He g; Tunnel SAM adapter: Adapting segment anything model for tunnel water leakage inspection; 2024. https://doi.org/10.1016/j.ghm.2024.01.001

[3] Sjölander, A.; Belloni, V.; Ansell, A.; Nordström, E. Towards Automated Inspections of Tunnels: A Review of Optical Inspections and Autonomous Assessment of Concrete Tunnel Linings. Sensors 2023, 23, 3189. https://doi.org/10.3390/s23063189

[4] Elberink, S.O.; Khoshelham, K. Automatic Extraction of Railroad Centerlines from Mobile Laser Scanning Data. Remote Sens. 2015, 7, 5565-5583. https://doi.org/10.3390/rs70505565

[5] Minea M, Dumitrescu CM, Dima M. Robotic Railway Multi-Sensing and Profiling Unit Based on Artificial Intelligence and Data Fusion. Sensors (Basel). 2021 Oct 16;21(20):6876. doi:10.3390/s21206876

[6] Donglin Zhou a, Xianlei Xu a b, Minghao Sun a, Hao Wang a, Tianyu Li; Multi-sensor fusion for enhanced localization and obstacle avoidance in rapid air-coupled GPR tunnel inspection;2025;

[7]https://leica-geosystems.com/products/mobile-mapping-systems/hardware?redir=45

[8] WSP. Available online: https://www.wsp.com/

[9] Yi C, Lu D, Xie Q, Xu J, Wang J. Tunnel Deformation Inspection via Global Spatial Axis Extraction from 3D Raw Point Cloud. Sensors (Basel). 2020 Nov 28;20(23):6815. doi: 10.3390/s20236815

[10]Stephanie Fekete a, Mark Diederichs a, Matthew Lato Geotechnical and operational applications for 3-dimensional laser scanning in drill and blast tunnels, 2010 https://doi.org/10.1016/j.tust.2010.04.008

[11] Delaloye, Dani. Queen's University (Canada) ProQuest Dissertations & Theses. " Development of a new methodology for measuring deformation in tunnels and shafts with terrestrial laser scanning (LiDAR) using elliptical fitting algorithms,

2012.https://www.proquest.com/openview/857501b28b502c6517ed6dae5fb35e8e/1?pq-origsite=gscholar&cbl=18750

[12] Jonathan D Aubertin, D. Jean Hutchinson, Mark S. Diederichs; Application of LiDAR technology for the assessment of mechanical scaling conditions in underground mines and tunnels,2019,https://www.researchgate.net/publication/351483209_Application_of_LiDAR_technology_for_the_assessment_of_mechanical_scaling_conditions_in_underground_mines_and_tunnels

[13] G. Walton; G. Mills; G. Fotopoulos; R. Radovanovic; R.P.W. Stancliffe; An approach for automated lithological classification of point clouds; 2016; https://doi.org/10.1130/GES01326.1

[14] Zoltán Nagy*, Árpád Somogyi; Automated Extraction and Accuracy Verification of Railway Track Axes from MLS Point Clouds; 2026; https://doi.org/10.55779/ng62635

[15] Zhang A, Huang J, Sun Z, Duan J, Zhang Y, Shen Y. Leakage Detection in Subway Tunnels Using 3D Point Cloud Data: Integrating Intensity and Geometric Features with XGBoost Classifier. Sensors (Basel). 2025 Jul 18;25(14):4475. doi: 10.3390/s25144475

[16] Wu, J.; Zhang, X. Tunnel Crack Detection Method and Crack Image Processing Algorithm Based on Improved Retinex and Deep Learning. Sensors 2023, 23, 9140. https://doi.org/10.3390/s23229140

[17] AI Driven Structural Health Monitoring for Smart Infrastructure Systems. (2026). Research Consortium Archive, 4(2), 1691-1705. https://doi.org/10.5281/zenodo.20353104

[18] Surya Dhote1 , Rishabh Singh2; Data-Driven Structural Health Monitoring Using Artificial Intelligence; https://www.ijfmr.com/papers/2026/2/73120.pdf

[19] Puente, Iván. “Review of Mobile Mapping and Surveying Technologies.” Measurement, 2013. https://doi.org/10.1016/J.MEASUREMENT.2013.03.006

[20] https://github.com/lichking2017/STSD

[29] Hao Cui, Xiaochun Ren, Qingzhou Mao, Qingwu Hu, Wei Wang; Shield subway tunnel deformation detection based on mobile laser scanning,; Automation in Construction; 2019; https://doi.org/10.1016/j.autcon.2019.102889.

[21] Sajid Ali a, Tamer Abuhmed b, Shaker El-Sappagh b c d, Khan Muhammad e, Jose M. Alonso-Moral f, Roberto Confalonieri g, Riccardo Guidotti h, Javier Del Ser i j, Natalia Díaz-Rodríguez k, Francisco Herrera k; Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence; 2023; https://doi.org/10.1016/j.inffus.2023.101805

[22] Jen-Yu Han, Jenny Guo, Yi-Syuan Jiang; Monitoring tunnel deformations by means of multi-epoch dispersed 3D LiDAR point clouds: An improved approach; 2013; https://doi.org/10.1016/j.tust.2013.07.022

[23]Beger, R.; Gedrange, C.; Hecht, R.; Neubert, M. Data fusion of extremely high resolution aerial imagery and lidar data for automated railroad centre line reconstruction. ISPRS J. Photogramm. Remote Sens. 2011, 66, S40–S51.

[24] B. Yang and L. Fang, "Automated Extraction of 3-D Railway Tracks from Mobile Laser Scanning Point Clouds," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 12, pp. 4750-4761, Dec. 2014, doi: 10.1109/JSTARS.2014.2312378

[25]Oude Elberink, S., Khoshelham, K., Arastounia, M., and Diaz Benito, D.: Rail Track Detection and Modelling in Mobile Laser Scanner Data, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-5/W2, 223–228, https://doi.org/10.5194/isprsannals-II-5-W2-223-2013, 2013.

[26] Martin Weinmann a, Boris Jutzi a, Stefan Hinz a, Clément Mallet; Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers; 2015 ; https://doi.org/10.1016/j.isprsjprs.2015.01.016

[27]https://cdn.standards.iteh.ai/samples/32575/cfc99e628ec6449c8a657df4061a0c7b/ISO-19157-2013.pdf

[28] Ziyang Xu , Olaf Wysocki , Christoph Holst; Point-level Uncertainty Evaluation of Mobile Laser Scanning Point Clouds; 2025; https://doi.org/10.48550/arXiv.2510.24773

[30] D. Bazazian, J. R. Casas and J. Ruiz-Hidalgo, "Fast and Robust Edge Extraction in Unorganized Point Clouds," 2015 International Conference on Digital Image

Computing: Techniques and Applications (DICTA), Adelaide, SA, Australia, 2015, pp. 1-8, doi: 10.1109/DICTA.2015.7371262

[31] Jin, Y.-H.; Lee, W.-H. Fast Cylinder Shape Matching Using Random Sample Consensus in Large Scale Point Cloud. Appl. Sci. 2019, 9, 974. https://doi.org/10.3390/app9050974

[32] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231.

[33] Karunathilake, A.; Honma, R.; Niina, Y. Self-Organized Model Fitting Method for Railway Structures Monitoring Using LiDAR Point Cloud. *Remote Sens.* **2020**, *12*, 3702. https://doi.org/10.3390/rs12223702

[34] Aleksi Kononen, Harri Kaartinen, Antero Kukko, Matti Lehtomäki, Josef Taher, Juha Hyyppä,Fully automated extraction of railtop centerline from mobile laser scanning data,Automation in Construction,Volume 168, Part A,2024,105812,ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2024.105812

[35] M.R. Mahendrini Fernando Ariyachandra, Ioannis Brilakis,Leveraging railway topology to automatically generate track geometric information models from airborne LiDAR data,Automation in Construction,Volume 155,2023,105068,ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2023.105068.

[36] Lingbo Long, Runing Xiao, Hong Zhao, Cong Yang, Teng Li, Fuqiang Zhong, Changyan Xiao, Yaonan Wang,An apparatus and method for on-site extrinsic calibration of the laser scanner relative to IMU used in a rail inspection vehicle, Measurement,Volume 239, 2025, 115435, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2024.115435.

[37] Mario Soilán, Andrea Nóvoa, Ana Sánchez-Rodríguez, Andrés Justo, Belén Riveiro, Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data, Automation in Construction, Volume 126, 2021, 103684, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2021.103684.