ISSN 1392-3730/eISSN 1822-3605

VILNIUS TECH
Vilnius Gediminas Technical University

JOURNAL of CIVIL ENGINEERING and MANAGEMENT

2026
Volume 32
Issue 3
Pages 309–335
https://doi.org/10.3846/jcem.2026.26522

PHYSICS-INFORMED NEURAL NETWORK–INTEGRATED POINT CLOUD DECIMATION AND 3D MESH MODELING FOR THERMO-MECHANICAL SIMULATION OF RAIL FASTENERS IN DIGITAL TWIN SYSTEMS

Jin WANG[1], S Muhammad Ahmed Hassan SHAH[1], Shi QIU[3, 4], Weidong WANG[1], Haleema EHSAN[1], Chengbo AI[2], Yangming LUO[1], Qasim ZAHEER[1]✉

[1]*School of Civil Engineering, Central South University, Changsha 410075, P. R. China*
[2]*Department of Civil and Environmental Engineering, University of Massachusetts, Amherst, MA 01003, USA*
[3]*School of Intelligent Construction and Low-altitude Technology, Guangxi Transport Vocational and Technical College, Nanning Guangxi 530216, P. R. China*
[4]*Guangxi Engineering Research Center for 3D Digital Twin of Transportation Infrastructure, No.1258 Kunlun Avenue, Nanning Guangxi 530216, P. R. China*

Article History:
- received 20 January 2025
- accepted 29 January 2026

Abstract. High-density point cloud data of infrastructure components, such as railway fasteners, often contain excessive noise and redundant points, making them challenging to process for simulation and analysis. This study introduces an integrated framework that transforms raw point clouds into simulation-ready 3D meshes and couples them with Physics-Informed Neural Networks (PINNs) for thermo-mechanical analysis, enabling applications within digital twin environments. The pipeline begins with a RANSAC-based iterative segmentation and outlier removal, achieving an 82.35% reduction in point count while preserving essential geometry. Poisson surface reconstruction and targeted post-processing then produce a high-fidelity, watertight mesh normalized for consistent simulation input. Leveraging this mesh, PINNs solve the steady-state heat equation to model thermal conduction and linear elasticity equations to estimate stress and displacement fields under defined Dirichlet boundary conditions. The resulting temperature and stress distributions are visualized directly on the mesh, providing interpretable, physics-consistent insights into fastener performance. By unifying geometric simplification with data-driven, physics-aware simulation, this approach supports accurate and computationally efficient digital twin development for railway asset monitoring and maintenance.

Keywords: digital twin, physics-informed neural networks, point cloud decimation, 3D mesh generation, rail fasteners, thermo-mechanical simulation.

✉Corresponding author. E-mail: *drqasimzaheer26@csu.edu.cn*

1. Introduction

The development of high-fidelity 3D models has become a cornerstone in modern infrastructure management, serving as accurate digital representations that support a wide range of applications such as scene visualization, disaster response planning, route optimization, and Building Information Modeling (BIM) (Li et al., 2024). The demand for such models continues to grow, driven by their value in enhancing decision-making, improving resource allocation, and enabling predictive maintenance. A key source for constructing these models is point cloud data, often captured using laser scanning or other high-precision sensing technologies (Bello et al., 2020). While these datasets contain detailed geometric information, they also present significant challenges due to their extremely high density, noise, and redundancy. Managing, storing, and processing point clouds with millions of points per component can quickly become computationally prohibitive, particularly when dealing with large-scale infrastructure projects (Mirzaei et al., 2022; Zhang et al., 2015). Traditional approaches, such as spatial indexing, voxel grid decimation, or surface reconstruction, offer partial solutions but often suffer from scalability issues and may discard essential geometric features, rendering them unsuitable for engineering-grade analysis or integration into simulation environments.

Machine learning has introduced powerful tools for processing and analyzing large-scale point cloud data, offering automation and adaptability (Hu et al., 2022; Zhang et al., 2019). Supervised methods like Support Vector Machines and Random Forests have shown strong capabilities in classifying points and extracting features, while unsu-

pervised approaches, such as K-Means or DBSCAN clustering and dimensionality reduction via Principal Component Analysis, uncover latent patterns and reduce computational complexity (Mansour et al., 2024). These advances are further enhanced through feature engineering, where geometric properties, intensity values, and neighborhood relationships are leveraged to improve the representational quality of the data (Li et al., 2019). However, even with these tools, a persistent challenge remains: achieving significant data simplification without compromising the fidelity of key geometric characteristics that are critical for simulation and engineering analysis (Justo et al., 2021; Vassilev et al., 2024). This challenge is particularly acute in digital twin applications, where real-time simulations depend on models that are both computationally efficient and geometrically accurate (Arce Munoz, 2020; Cura et al., 2017).

In railway infrastructure, the problem becomes even more pronounced when dealing with small yet complex components such as rail fasteners (Hinks et al., 2013; Qiu et al., 2024; Walsh et al., 2013). These elements are essential to track stability and safety, but their intricate shapes and functional requirements make them difficult to model and monitor effectively. High-resolution point clouds of fasteners often contain not only fine geometric details but also substantial amounts of irrelevant data and noise (Huang et al., 2024; Xu et al., 2024). Generic simplification algorithms may remove redundant points but risk degrading features vital for mechanical integrity assessment (Bello et al., 2020; Dey et al., 2003; Hidaka et al., 2019). Furthermore, the absence of domain-specific decimation strategies tailored to fastener geometries means that conventional pipelines often fail to produce meshes that are both lightweight and simulation-ready (Taubin, 1995). This gap highlights the need for a specialized approach that reduces dataset size while preserving engineering-critical geometry, enabling integration with advanced simulation frameworks and digital twin platforms (Hajihosseinlou et al., 2024; Hou et al., 2024).

Mesh generation serves as the crucial link between raw point cloud acquisition and numerical simulation. The process transforms unstructured point data into structured surface representations, allowing for more efficient visualization, analysis, and interoperability with engineering tools (Aljumaily et al., 2017; Czerniawski et al., 2018; Raj et al., 2020; Yuan et al., 2021). High-quality meshes require a robust workflow that includes preprocessing to remove noise and outliers (Jing et al., 2021; Rashidi et al., 2013), normal estimation to define surface orientation, reconstruction via methods such as Poisson surface reconstruction to produce watertight geometry, and post-processing steps to refine the mesh, remove defects, and normalize coordinates (Chen et al., 2017; Fathi & Brilakis, 2011). For high-density, noisy datasets, the effectiveness of surface reconstruction depends heavily on the quality of the decimated point cloud, making the preprocessing stage critical (Liu et al., 2023; Weerasinghe, 2023). In recent years, domain-focused decimation and reconstruction strategies have emerged as key enablers for producing meshes that are both computationally manageable and structurally faithful.

The process of designing and extracting informative features from raw data (Dekker et al., 2023; Saovana et al., 2021; Vick & Brilakis, 2018) plays a crucial role in optimizing the performance of machine learning models (Bassier et al., 2020; Chen et al., 2020; Franco-Duran & Mejia, 2016). In the context of point clouds, these features might include geometric properties (Balakrishnan Selvakumaran & Hall, 2022; de la Plata et al., 2021; Rausch & Haas, 2021), intensity values, or local neighborhood information (Cai et al., 2023; Cao et al., 2024b). While it requires domain knowledge and expertise, feature engineering significantly enhances the representation of 3D data for machine learning algorithms (Soilán et al., 2021). By leveraging these techniques, machine learning provides a powerful foundation for processing, analyzing, and unlocking the valuable insights hidden within point cloud data (Wang et al., 2026; Zaheer et al., 2026a, 2026b, 2026c).

Notable research has contributed to the evolution of this field. Bello et al. (2020) have demonstrated the effectiveness of deep learning for point cloud feature extraction, Zhang et al. (2019), Richter and Döllner (2014), Sakr et al. (2011) have reviewed deep learning approaches for large-scale semantic segmentation, Wang et al. (2025a, 2025b), and Dey and Goswami (2003) have introduced a local feature size–based decimation method to preserve detail in complex geometries – concepts closely aligned with the approach adopted in this work (Agrawal & Choudhary, 2023; Zhuang et al., 2023). Advances in Poisson surface reconstruction have further improved mesh quality in the presence of noisy data, which is particularly important for fasteners, where surface features influence load distribution and wear (Boissonnat & Cazals, 2001; Pauly et al., 2002). These developments provide the foundation upon which our proposed methodology builds, enabling a workflow that moves seamlessly from targeted point cloud decimation to simulation-ready mesh generation, and finally to physics-based performance simulation (Alexa et al., 2003; Moenning & Dodgson, 2004).

In this study, we address the dual challenges of excessive data complexity and preservation of structural fidelity in modeling rail fasteners. We present an end-to-end framework that begins with a Random Sample Consensus (RANSAC)–based decimation process tailored to isolate planar regions and remove outliers. This targeted approach achieves an 82.35% reduction in point count while retaining essential geometric features. The decimated point cloud is then processed using Poisson surface reconstruction to produce a watertight mesh, followed by refinement steps such as duplicate vertex removal, degenerate triangle elimination, smoothing, and normalization to ensure simulation compatibility. Unlike existing pipelines that stop at geometric modeling, our framework integrates Physics-Informed Neural Networks (PINNs) to directly

simulate thermo-mechanical behavior over the generated mesh. PINNs embed the governing partial differential equations – specifically, the steady-state heat equation for thermal conduction and the linear elasticity equations for stress and displacement – into their loss functions, enabling physically consistent predictions without the need for extensive labeled data. Mesh vertices serve as the spatial input to the networks, allowing for continuous field predictions across the geometry. This capability supports the generation of interpretable temperature and stress distributions, which can be visualized directly on the mesh to identify hotspots, stress concentrations, and deformation patterns.

By combining targeted geometric simplification with PINN-based physics simulation, the proposed framework produces lightweight yet physically meaningful models that are ideally suited for integration into digital twin systems. Such systems can leverage the efficiency of the decimation process, the fidelity of the mesh reconstruction, and the predictive capabilities of PINNs to enable real-time monitoring, predictive maintenance, and informed decision-making across the lifecycle of railway assets. The main contributions of this study are as follows:

- A domain-specific RANSAC-driven point cloud decimation strategy that isolates planar regions and removes outliers, achieving over 80% reduction in point count while preserving structural detail essential for engineering analysis.
- A systematic 3D mesh generation pipeline incorporating Poisson surface reconstruction, post-processing refinement, and normalization to produce watertight, high-fidelity meshes optimized for simulation use.
- Integration of Physics-Informed Neural Networks for thermo-mechanical simulation, embedding governing PDEs into the network architecture to predict temperature, stress, and displacement fields directly on the reconstructed mesh.
- A unified end-to-end workflow that connects point cloud preprocessing, mesh modeling, and physics-aware simulation into a single framework, enabling scalable and accurate digital twin development for rail infrastructure monitoring.

Benefits and Applications. The proposed end-to-end framework represents a substantial advancement in processing and exploiting point cloud data for infrastructure asset management, particularly for components with intricate geometries such as rail fasteners. By combining targeted point cloud decimation, high-fidelity mesh generation, and physics-informed simulation, it delivers a scalable solution for both visualization and engineering-grade analysis. The decimation step drastically reduces data volume while retaining essential structural features, enabling efficient storage, transmission, and rendering. This makes it feasible to work with large-scale datasets, such as those collected from rail networks, bridges, tunnels, or stations, using standard computational resources. The lightweight datasets also improve interactive visualization performance, allowing stakeholders to explore accurate 3D models without the delays typically associated with raw LiDAR or photogrammetric scans.

Importantly, the generated meshes are both visually accurate and structurally reliable, making them directly compatible with simulation tools such as Finite Element Analysis (FEA) software. This compatibility enables advanced evaluations, including thermal response analysis, stress–strain estimation, and deformation modeling under mechanical loading. The integration of Physics-Informed Neural Networks (PINNs) adds a critical capability – solving governing partial differential equations (PDEs) directly on these meshes to model thermo-mechanical behavior with physical consistency. As a result, the framework not only supports realistic structural analysis but also provides a pathway to fully operational digital twins for railway infrastructure. Such twins enable continuous monitoring, predictive maintenance scheduling, and data-driven decision-making across the asset lifecycle, aligning with Industry 4.0 and 5.0 paradigms where simulation, automation, and intelligent data integration converge.

By addressing both data complexity and structural fidelity, this framework reduces the time, storage, and computational costs associated with high-density point cloud processing while maintaining the precision needed for engineering simulations.

2. Methodology

This study proposes a unified workflow that transforms raw, high-density point clouds of rail fasteners into simulation-ready 3D meshes and subsequently applies PINNs to model thermo-mechanical behavior directly on the reconstructed geometry. The methodology consists of three main phases: (1) RANSAC-driven point cloud decimation, (2) high-quality mesh generation, and (3) physics-informed simulation.

2.1. Point cloud decimation

High-resolution point clouds often contain millions of points, making their processing difficult and costly. To address this, we employ a Random Sample Consensus (RANSAC)–based segmentation (Figure 1) to iteratively identify and isolate planar regions, which are characteristic of many rail fastener surfaces. In each iteration, RANSAC:

- Selects a minimal subset of points.
- Fits a candidate plane model.
- Calculates the distance error for all points relative to this model.
- Classifies points within a specified tolerance as inliers and others as outliers.

The process continues until either the maximum inlier set is obtained or a predefined iteration limit is reached. Outliers, typically representing noise or irrelevant background geometry, are removed. Multiple RANSAC passes allow for finer segmentation, progressively focusing on

the fastener's essential surfaces. This method achieves an 82.35% reduction in point count from an initial dataset of ~1.7 million points (Figure 2) while preserving geometric features critical for structural evaluation.

2.2. Mesh generation pipeline

Once the decimated point cloud is obtained, a structured mesh is generated through the following steps:

- Normal Estimation – Surface normals are calculated for each point to guide subsequent reconstruction.
- Poisson Surface Reconstruction – This algorithm generates a watertight, continuous surface from oriented points, producing an initial mesh that conforms closely to the measured geometry.
- Post-processing – The mesh undergoes refinement, including duplicate vertex removal, degenerate triangle cleanup, and Taubin smoothing to enhance surface quality.
- Quadric Edge Collapse Decimation – The mesh is further simplified while preserving curvature-sensitive features.

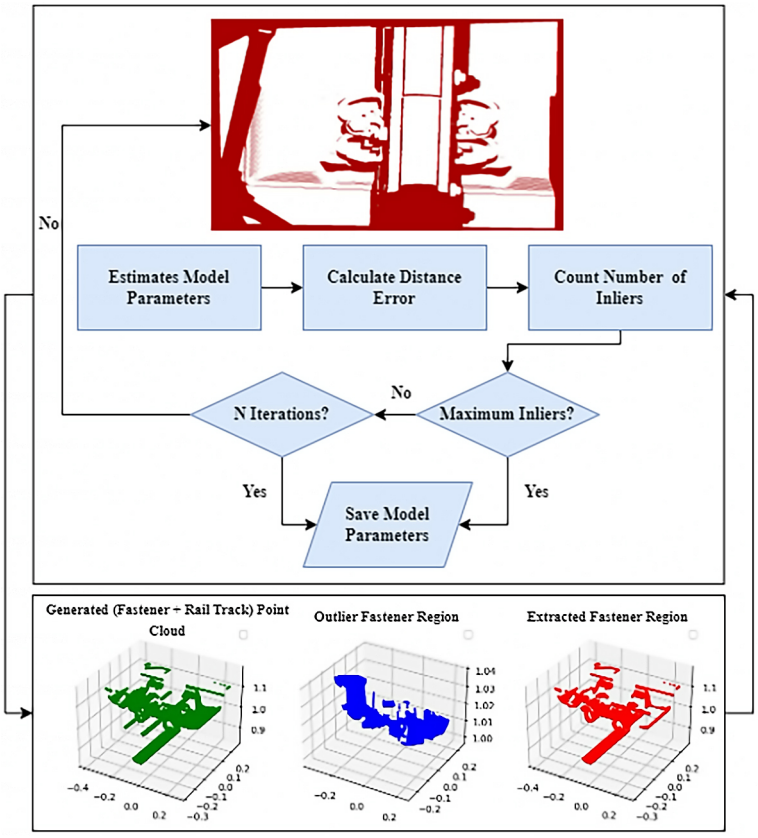


Figure 1. RANSAC segmentation framework

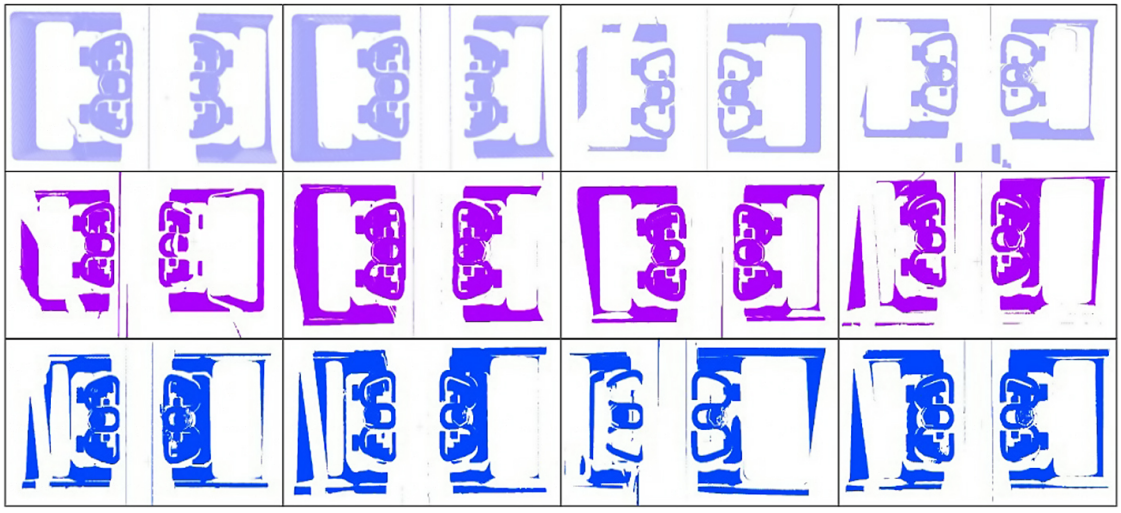


Figure 2. Decimated point cloud

- Triangle Connectivity Analysis and ROI Filtering – Only the mesh regions corresponding to the fastener are retained, ensuring that extraneous geometry is excluded from simulations.
- Normalization – The mesh is centered at the origin and scaled to a standardized coordinate system, removing translation dependencies and ensuring compatibility with numerical solvers.

This process yields a high-fidelity, simulation-ready mesh optimized for both visual inspection and physics-based modeling.

2.3. Physics-informed neural network (PINN) simulation

To simulate physical behaviors on the generated mesh, we implement PINNs, which embed the governing PDEs directly into the network's loss function. Mesh vertices serve as spatial coordinates, and the network learns continuous solutions across the geometry without requiring labeled simulation data. Two PDE-based simulations are performed:

- Steady-State Heat Conduction – Modeled via the equation $\nabla^2 T = f$, representing temperature distribution under synthetic heat loading.
- Linear Elasticity – Governing equations for displacement fields under mechanical loads, with Dirichlet boundary conditions applied to constrained regions.

PINNs predict field values at all mesh vertices, which are visualized as vertex-wise scalar or vector maps, revealing thermal gradients, stress concentrations, and deformation patterns. This physics-informed approach ensures that outputs respect both the data geometry and the underlying physical laws, making results suitable for digital twin integration and predictive maintenance scenarios.

2.4. Model architecture

This research prioritizes fastener health monitoring as a key strategy for ensuring smooth and incident-free operations. The study employs a novel approach that involves transforming and processing point cloud data into 3D meshes. This process consists of several crucial steps that work together seamlessly. These steps ensure an accurate and efficient transformation from raw data into a refined mesh suitable for further analysis. Initially, the system captures the point cloud data representing both the fastener and the portion of the rail track. This data undergoes a segmentation process where model parameters are estimated, and distance errors are calculated to count the number of inliers utilizing the RANSAC algorithm. Through an iterative approach, the algorithm seeks to maximize the number of inliers, repeating the estimation and error calculation until either a set number of iterations is completed, or the maximum number of inliers is achieved. Once the optimal model parameters are identified, they are saved, and the fastener region is extracted from the point cloud, eliminating outliers to produce a clean dataset. Subsequently, the refined point cloud data transitions to the mesh generation phase. Following this, normals are estimated for each point, followed by a surface Poisson reconstruction to generate an initial mesh. This mesh undergoes a series of refinement processes: duplicate vertices are removed, degenerate triangles are eliminated, and the mesh is subjected to Taubin smoothing to enhance its quality. Moreover, quadric decimation is applied to simplify the mesh while preserving significant details. Further, this study performs triangle connectivity analysis to verify the integrity of the 3D mesh. Additionally, region of interest (ROI) triangle filtering is employed to isolate specific sections of the mesh for focused analysis. The culmination of these processes results in a highly detailed and accurate mesh representation of the rail fastener, suitable for further analysis and practical applications. This comprehensive architecture ensures a robust transformation from raw point cloud data to a precise and usable 3D mesh model, meeting the stringent requirements of technical accuracy and efficiency. Mathematical formulation is given below:

$$\frac{|ax+by+cz+d|}{\sqrt{a^2+b^2+c^2}}<\epsilon. \tag{1}$$

The RANSAC algorithm excels at finding large groups of points that are likely to lie on the same flat surface within a fastener point cloud. It achieves this by following a strategic process. The working phenomenon of RANSAC, picking a small handful of points at random from the entire cloud. Using these points, it mathematically creates a model of a flat plane, which can be described by an equation like:

$$ax+by+cz+d=0, \tag{2}$$

here, a, b, and c represent properties of the plane, (x, y, and z) represent the coordinates of any point, and d signifies the distance from the origin to the plane. The next step is to see how well all the other points in the cloud fit this newly created plane. Any point that falls very close to the plane (within a predefined distance limit) is considered an "inlier".

Inliers are points that most likely belong to the flat surface that RANSAC is trying to identify. RANSAC repeats this process – picking points, building a plane model, and classifying inliers – multiple times. The key is to find the largest group of points (called the consensus set) that all fit the same plane model. This iterative approach effectively helps RANSAC isolate and extract the major flat regions within the fastener. The predefined distance limit, represented by the symbol epsilon (ε), determines how strict we are in considering a point to be part of the flat surface, as shown in equation 3. (ε) means a tighter fit to the plane is required for a point to be classified as an inlier. Mathematically, let P_1, P_2, P_3 be three randomly selected data points. The normal vector (a, b, c) of the plane can be estimated as:

$$(a,b,c)=\frac{(P_2-P_1)\times(P_3-P_1)}{(P_2-P_1)\times(P_3-P_1)}. \tag{3}$$

When applying the RANSAC algorithm specifically to point cloud data for plane segmentation, the equations can be adapted as follows:

$$ax_i + by_i + cz_i + d = 0, \tag{4}$$

where (a, b, c) is the normal vector of the plane, x_i, y_i, z_i are the coordinates of a point in the point cloud, and d is the distance from the origin to the plane.

The error between a point $P_i = (x_i, y_i, z_i)$ and the plane can be calculated as the perpendicular distance from the point to the plane:

$$d(P_i) = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}}. \tag{5}$$

A threshold value ε is set to determine inliers. A point P_i is considered an inlier if the distance $d(P_i)$ is less than or equal to ε. The goal is to estimate the parameters (a, b, c, d) of the plane that minimize the error between the model and the data points. This can be formulated as a least-squares problem:

$$\min_{a,b,c,d} \sum_{i=1}^{n} d(P_i)^2, \tag{6}$$

where n is the number of points in the point cloud.

The model is updated based on the inliers, and the parameters are re-estimated. Let I be the set of inliers. The updated parameters are estimated by solving the following least-squares problem:

$$\min_{a,b,c,d} \sum_{i \epsilon I} d(P_i)^2. \tag{7}$$

The probability P of at least one of the random samples being free of outliers after k iterations is given by:

$$P = 1 - \left(1 - (1-\varepsilon)^S\right)^K, \tag{8}$$

where ε is the proportion of outliers in the data, and S is the size of the subset used for model fitting. Solving for K, we get:

$$K = \frac{\log(1-P)}{\log\left(1 - (1-\varepsilon)^S\right)}. \tag{9}$$

2.5. 3D mesh generation

Point clouds, which are 3D points sampled from the surfaces of rail fasteners, are crucial for various geometric processing tasks. One important task is generating meshes from these point clouds, as meshes provide structured surface representations that are useful for advanced analysis and visualization. Figure 3 depicts our novel method for creating meshes from point clouds using open-source tools and techniques. The process begins with obtaining a point cloud dataset, typically collected using sensors that capture detailed spatial information about the fasteners, i.e., LiDAR. These sensors produce dense point clouds as raw data, which is very expensive and complex to process in terms of resources and computational capability. Therefore, decimation is inevitable, which leads us to mesh generation. That we already explained above. Before creating the mesh, the point cloud needs preprocessing to improve data quality and reduce noise. Common preprocessing steps include estimating normals, removing duplicate vertices, and smoothing the mesh to ensure a clean and uniform surface.

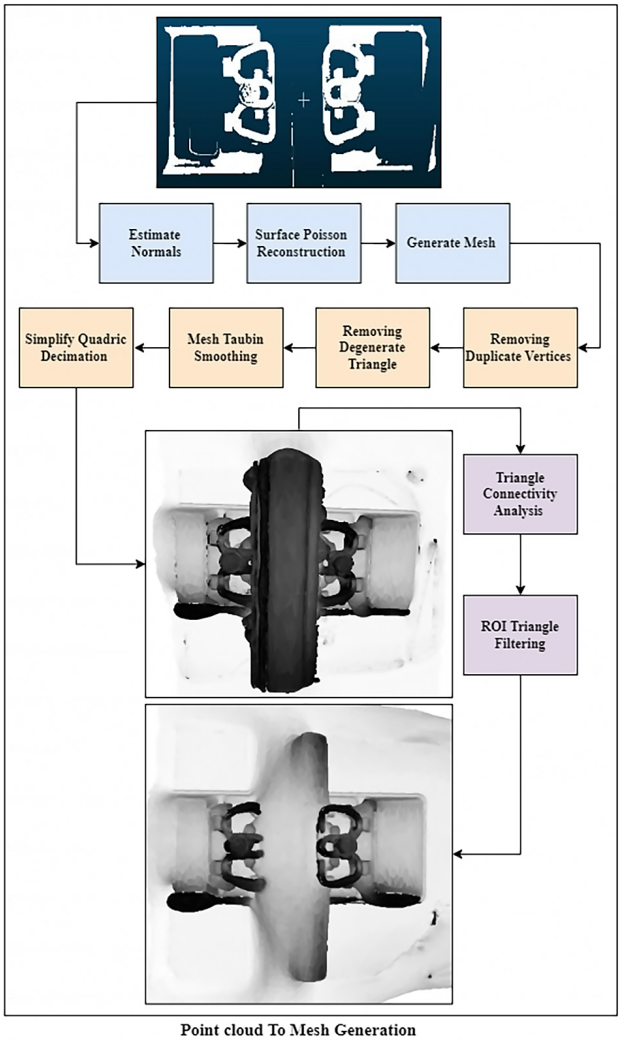


Figure 3. Mesh generation framework

The core of the method is generating a mesh using a single application of the Poisson surface reconstruction algorithm (Kazhdan et al., 2006). This algorithm constructs a mesh that fits the input point cloud accurately, maintaining the original geometry's structural integrity. After generating the mesh, additional post-processing steps can further refine it. These steps often include removing degenerate triangles, normalizing the mesh to a common reference frame, and decimating the mesh. By following these steps, we transformed raw point cloud data into a high-quality mesh that accurately represents the original object, facilitating better analysis and visualization.

By summing all vertex positions across the entire mesh centroid "$\boldsymbol{c}$" is given as:

$$\boldsymbol{c} = \left(\frac{1}{|V|} \sum_{v \in V} x_v, \frac{1}{|V|} \sum_{v \in V} y_v, \frac{1}{|V|} \sum_{v \in V} z_v \right). \quad (10)$$

Normals' estimation is a fundamental step in the mesh generation process. It involves calculating the normal vectors for each point in the point cloud, which are perpendicular to the surface at each point. These normals are crucial for subsequent surface reconstruction steps, as they help define the geometric structure and ensure the smoothness of the generated mesh. The process of normalizing a mesh derived from a point cloud involves a series of deliberate steps. Initially, the symbolic representation ∑ (sigma) hints at a triple summation, typically indicative of iterating through all vertices within the mesh. For every vertex ***v*** within the vertex set ***V*** of the mesh, a centroid ***c*** is computed by summing all vertex positions across the entire mesh. This computation yields the centroid or center of mass for the mesh. Subsequently, each vertex ***v*** undergoes a translation process by subtracting the centroid ***c*** from its position. This transformation effectively repositions every vertex so that the mesh's centroid aligns with the origin (0,0,0) within the coordinate system. Ultimately, this normalization procedure ensures that the mesh is centralized at the origin, facilitating subsequent analyses or operations by eliminating any translation-dependent attributes.

Furthermore, this involved segmenting the mesh into distinct regions using a RANSAC-based geometric segmentation strategy applied at the point-cloud level before surface reconstruction. Specifically, RANSAC is employed to identify and separate dominant geometric primitives associated with the rail fastener from surrounding planar or quasi-planar structures, such as baseplate or track components.

Following this step, the segmented point sets are used to reconstruct the corresponding surfaces, enabling a clear differentiation between the fastener and the surrounding infrastructure, as illustrated in Figure 4. This segmentation is not intended for fine-grained shape decomposition but rather for semantic isolation and outlier suppression, which improves the robustness of subsequent surface reconstruction and physics-based analysis. The resulting segmented mesh facilitates downstream tasks such as object-level interpretation and geometry-aware modeling.

As introduced earlier, the reconstructed mesh is generated using the Poisson surface reconstruction algorithm (Kazhdan et al., 2006), which constitutes the core surface generation step of the proposed pipeline. This paragraph provides additional clarification of that process rather than introducing a separate reconstruction stage. Poisson surface reconstruction is applied once to the segmented and preprocessed point cloud to generate a smooth, watertight surface representation of the rail fastener. The method leverages estimated surface normals to solve a Poisson equation whose solution yields a continuous surface that best fits the input data in a global sense. Owing to its robustness to noise and incomplete sampling, this approach is well-suited for scanned railway components and produces a stable geometric basis for subsequent physics-informed analysis.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = g, \quad (11)$$

where f is the scalar function representing the reconstructed surface, and g is the Laplacian of the input point cloud. The utilization of the Poisson surface reconstruction algorithm ensures the generation of a mesh that accurately conforms to the structural characteristics of the input point cloud. The algorithm cleverly analyzes the point cloud data, focusing on the key features, to build a highly accurate mesh that faithfully represents the original surface. This approach facilitates the creation of meshes that faithfully represent the underlying surfaces, enabling precise geometric processing and analysis as shown in Figure 5.

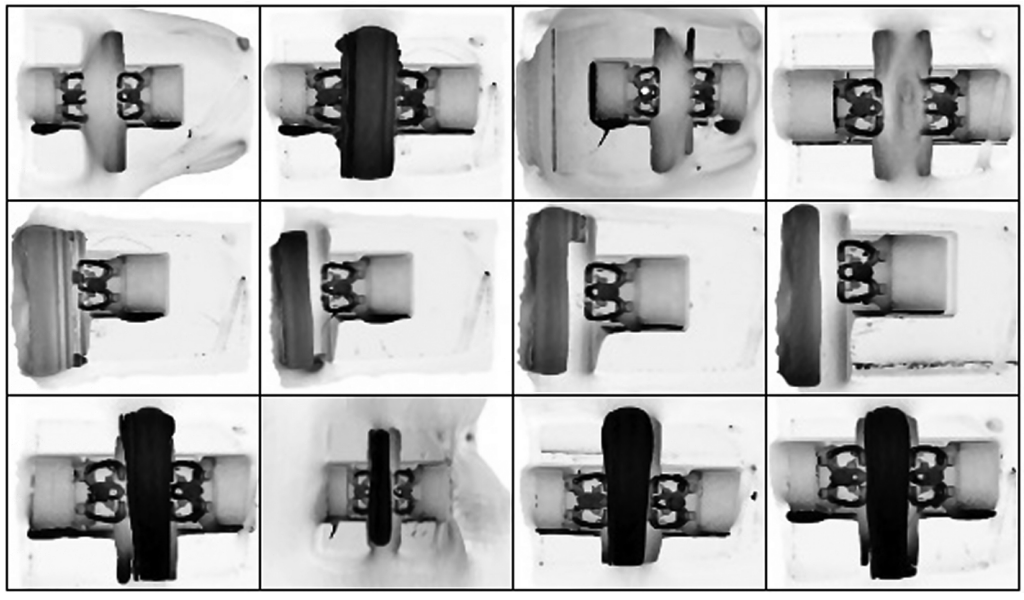


Figure 4. Segmented mesh

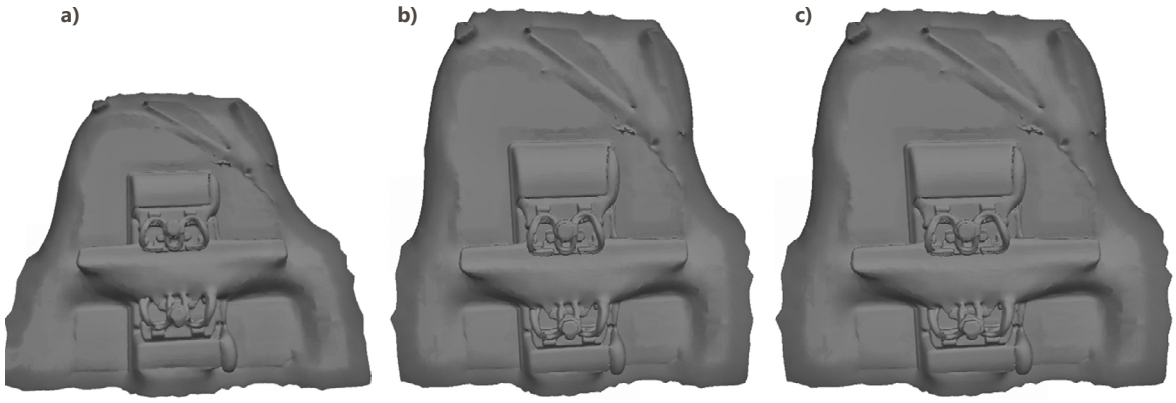


Figure 5. Surface reconstruction

Mathematically, the point cloud is represented as a set of 3D coordinates, $P = \{p_1, p_2, ..., p_n\}$, where $p_i = (x_i, y_i, z_i)$. The normal vector N_i at point p_i is computed as the eigenvector corresponding to the smallest eigenvalue of the covariance matrix of the neighboring points.

$$\text{Covariance matrix} = C = \sum \left(p_i - \overline{p}\right)\left(p_i - \overline{p}\right)^T; \quad (12)$$

$$Cv_i = \lambda_i v_i, \quad (13)$$

where v_i is the eigenvector and λ_i is the eigenvalue. $N_i = v_i$ (corresponding to the smallest eigenvalue λ_i).

The Poisson equation is formulated as:

$$\nabla^2 \Phi = \nabla . N, \quad (14)$$

where Φ is the scalar function, ∇^2 is the Laplacian operator, and N is the vector field of normals. The solution to the Poisson equation is obtained using numerical methods, such as the finite element method or the multigrid method.

Following, the next step involves duplicate vertices removal. The process of removing duplicate vertices is essential for maintaining mesh quality. Duplicate vertices can occur during the reconstruction process and can lead to inaccuracies and complications in the mesh structure. By eliminating these duplicates, the mesh becomes cleaner and more reliable for further processing.

Removing duplicated vertices and degenerate triangles. Degenerate triangles are triangles with very small angles or zero area, which can occur in the mesh due to various numerical issues. Removing these triangles is crucial to avoid mesh artifacts and to ensure the structural integrity of the mesh. This step helps in maintaining a high-quality mesh that accurately represents the intended geometry. This method translated the mesh to center it at the origin:

$$T = I - \frac{1}{n}\sum p_i, \quad (15)$$

where T is the translation matrix, and I is the identity matrix.

Further, Taubin smoothing is used for further enhancement. Taubin smoothing is a mesh refinement technique that enhances the quality of the mesh by reducing noise while preserving essential geometric features (Taubin, 1995).

$$vi' = vi + \lambda \sum_{j \in N(i)} \left(vj - vi\right) - \mu \sum_{j \in N(i)} \left(vj' - vi'\right). \quad (16)$$

Specifically, the new position of vertex vi, denoted vi', is computed as the original position plus a weighted sum of differences between the positions of its neighbors and itself, minus a second weighted sum involving the updated neighbor positions. This is expressed by the equation: vi' equals vi plus lambda times the sum over all neighbors j of vj minus vi, minus mu times the sum over all neighbors j of vj' minus vi'. Here, vi represents the current position of the i-th vertex, vi' is its updated position, $N(i)$ denotes the set of vertices adjacent to vertex i, and lambda and mu are positive parameters that control the strength of the smoothing in two stages. The first summation acts to diffuse the vertex position, similar to Laplacian smoothing, while the second summation compensates for potential shrinkage, allowing the method to preserve sharp features during denoising.

This method uses the graph Laplacian operator to approximate the Laplace-Beltrami operator on the mesh. The Laplacian of a vertex vi is calculated as $\Delta vi = \sum_{j \in N(i)} \left(vj - vi\right)$, which drives the vertex towards the centroid of its neighbors. Taubin's key insight was to apply this smoothing in two sequential steps per iteration: a shrinking phase with a positive parameter λ, followed by an inflation phase with a negative parameter μ. This combination effectively reduces noise while counteracting the shrinkage inherent in a single smoothing pass. Combining these two steps into a single update rule yields the formulation in Eqn (16):

$$\hat{p}_i = p_i + \lambda_i \nabla^2 \left(p_i\right) - \lambda_2 \nabla^4 \left(p_i\right), \quad (17)$$

where $\hat{p}_i$ is the smoothed vertex position, ∇^2 is the Laplacian operator, and ∇^4 is the bi-harmonic operator. The quadric error metric is used to iteratively collapse vertices and edges while minimizing the error.

Quadric decimation simplifies the mesh by reducing the number of vertices while preserving its overall shape.

This process iteratively removes vertices and collapses edges based on a quadric error metric. Instead of associating an error with a vertex, Garland and Heckbert's method associates a 4×4 symmetric error matrix $\boldsymbol{Q}$ with each face. This matrix represents the squared distance to the plane of that face. For a face with a plane equation $n_j^T x + d_j = 0$. where n_i is the unit normal and d_j is the plane's offset), the corresponding error matrix is defined as:

$$Q_j = n_j n_j^T. \tag{18}$$

When collapsing an edge between two vertices, *vi* and *vk*, the optimal position for the new collapsed vertex *vm* is found by minimizing the sum of the quadratic errors from all faces adjacent to either original vertex. The total error for placing a vertex $v = [x, y, z, 1]^T$ at a certain location is given by the quadratic form:

$$Error(v) = v^T Q v, \tag{19}$$

where Q is the sum of the error matrices ($Q = \sum_j Q_j$) from the relevant faces. The vertex position *vm* that minimizes this error is found by solving for $v^T Q = 0$. This approach ensures that vertices are collapsed in a way that locally approximates the original surface geometry, thereby preserving significant details.

Moreover, triangle connectivity analysis ensures the structural integrity of the mesh by examining the relationships between triangles. This analysis helps in identifying and correcting any connectivity issues, such as holes or non-manifold edges, which could compromise the usability of the mesh. Ensuring proper connectivity is crucial for a robust and accurate mesh structure. Region of Interest (RO)I triangle filtering isolates the relevant sections of the mesh by selecting only those triangles that fall within the region of interest. This step ensures that the final mesh focuses on the critical parts of the fastener, eliminating extraneous data and enhancing the efficiency and accuracy of subsequent analyses. By focusing on the ROI, the process ensures that the mesh is both detailed and relevant to the specific application.

Mathematically, Region of Interest (ROI) triangle filtering isolates the relevant sections of the mesh by selecting only those triangles that fall within the region of interest. This step ensures that the final mesh focuses on the critical parts of the fastener, eliminating extraneous data and enhancing the efficiency and accuracy of subsequent analyses. By focusing on the ROI, the process ensures that the mesh is both detailed and relevant to the specific application. Mathematically, a triangle with vertices v_1, v_2, v_3 is included if it lies entirely within the region *R*:

$$v_1 \in R \wedge v_2 \in R \wedge v_3 \in R. \tag{20}$$

This process focuses on the critical parts of the fastener, eliminating extraneous data and enhancing the mesh's relevance and efficiency. Each of the above-mentioned techniques plays a vital role in the overall architecture, contributing to the robust transformation of raw point cloud data into a precise and usable 3D mesh model of the rail fastener.

Before surface reconstruction, a segmentation step based on RANSAC is applied to the raw point cloud. This step is not intended to replace or compete with freeform surface reconstruction methods, but rather to isolate dominant geometric primitives and remove spurious measurements arising from scanning noise, occlusions, or background structures.

Although Poisson surface reconstruction is robust to moderate noise, its performance and stability are sensitive to point density distribution and outlier contamination. The use of RANSAC-based segmentation, therefore, serves as a conditioning step that improves normal consistency and prevents the propagation of artifacts into the reconstructed surface.

2.6. Physics-informed neural network (PINNs)-based modeling of thermo-mechanical behavior in rail fasteners ratio

After the transformation of high-density point clouds into accurate 3D mesh representations of rail fasteners, we implement Physics-Informed Neural Networks (PINNs) to simulate and analyze two critical physical phenomena: steady-state thermal conduction and linear elastic deformation. Unlike traditional numerical methods (e.g., FEM), PINNs integrate physical laws directly into the learning process by embedding governing PDEs as soft constraints into the neural network's loss function. This allows for flexible, mesh-independent simulation over scattered spatial data while preserving the physical integrity of the solution.

While extensive prior work exists on point-cloud processing, surface reconstruction, and physics-based analysis, the primary methodological contribution of this study lies in the structured integration of a physics-informed neural network (PINN) into a geometry-driven digital-twin pipeline. Unlike purely application-oriented implementations, the PINN component in this work is explicitly defined in terms of architecture, governing equations, training strategy, and validation considerations, allowing it to be independently assessed and reproduced.

The PINN is not treated as a generic black-box predictor but as a physics-constrained function approximator operating directly on reconstructed three-dimensional geometry. Its role is to establish a continuous, differentiable field representation that bridges geometric reconstruction and physics-based reasoning, which is not explicitly addressed in conventional point-cloud or mesh-based pipelines. By detailing the design and training of the PINN, this work contributes a reusable modeling block that can be embedded into broader digital-twin frameworks rather than a case-specific application alone.

The heat diffusion problem considered in this study is formulated as a steady-state scalar field problem defined over the reconstructed geometry, rather than as a fully resolved transient heat transfer simulation within a vol-

umetric solid. Specifically, the PINN learns a continuous field $u(x)$ over spatial coordinates sampled from the reconstructed fastener geometry, subject to a Laplacian operator.

The formulation should therefore be interpreted as a geometry-constrained diffusion model intended to demonstrate physics-consistent field learning over complex shapes. It does not explicitly distinguish between surface-only diffusion and full volumetric heat conduction. A fully volumetric heat-transfer formulation, including material properties and transient effects, is beyond the scope of the present work and is left for future investigation.

(i) PINNs architecture

The core idea behind PINNs is to embed governing **Partial Differential Equations (PDEs)** into the neural network training process, as shown in Figure 6. A PINN learns to approximate a solution function $u(x)$ over a spatial domain Ω, such that both the function output and its derivatives satisfy the PDEs and boundary conditions. The network $f_\theta(x)$, parameterized by θ, is trained to minimize a combined loss:

$$\mathcal{L}_{total} = \mathcal{L}_{PDE} + \lambda.\mathcal{L}_{BC}, \tag{21}$$

where: $\mathcal{L}_{PDE}$ enforces the differential equation in the interior points, $\mathcal{L}_{BC}$ enforces boundary conditions on the mesh boundaries, λ is a weighting factor.

The architecture consists of a fully connected feedforward neural network with:

- **Input:** Spatial coordinates $(x, y, z) \epsilon \mathbb{R}^3$;
- **Hidden layers:** Multiple layers with non-linear activation functions (e.g., Tanh);
- **Output:** Scalar field $u \epsilon \mathbb{R}$ (e.g., temperature or displacement component).

Autograd is used to compute gradients with respect to input coordinates, enabling computation of spatial derivatives required by the PDEs.

(ii) Thermal field modeling using PINNs

To simulate steady-state thermal conduction over the rail fastener mesh, we formulate the problem using the Poisson equation for heat transfer. The flowchart of Thermal Field Modeling of Fastener Mesh is given in Figure 7.

Here, $T(x, y, z)$ is the temperature field to be predicted, and $f(x, y, z)$ is a known internal heat source (which can be set to 0 in pure conduction scenarios). The Laplacian operator $\nabla^2 T$ describes the second-order spatial derivatives of temperature in three dimensions.

To model this PDE with PINNs, we construct a neural network $T_\theta(x, y, z)$ and compute the second-order derivatives $\frac{\partial^2 T}{\partial^2 x}, \frac{\partial^2 T}{\partial^2 y}, \frac{\partial^2 T}{\partial^2 z}$ using autograd. The **PDE loss** penalizes deviation from the Laplace equation:

$$\mathcal{L}_{PDE} = \frac{1}{N_i} \sum_{i=1}^{N_i} \left(\nabla^2 T(x_i, y_i, z_i) - f(x_i, y_i, z_i) \right). \tag{22}$$

The **boundary loss** ensures the network adheres to physical conditions, such as fixed temperatures or insulated boundaries:

$$\mathcal{L}_{BC} = \frac{1}{N_b} \sum_{i=1}^{N_b} \left(T(x_i, y_i, z_i) - T_o \right). \tag{23}$$

Once trained, the PINN provides a continuous mapping from 3D spatial coordinates to temperature values, capturing gradients and hotspots within the fastener. This model can simulate thermal behavior under various loading or operational conditions, enabling predictive maintenance, especially where excessive heating may compromise component performance.

The computational domain $\Omega \subset \mathbb{R}^3$ is defined by the reconstructed three-dimensional fastener geometry obtained from the segmented point cloud. The boundary $\partial\Omega$ corresponds to the external surface of the reconstructed mesh. Heat diffusion is modeled inside the solid domain Ω using the steady-state heat equation given in Eqn (22). Dirichlet boundary conditions are imposed on a predefined subset $\partial\Omega_T \subset \partial\Omega$, where the temperature field is prescribed as $T(x) = T_o$. Here, T_o denotes a constant ref-

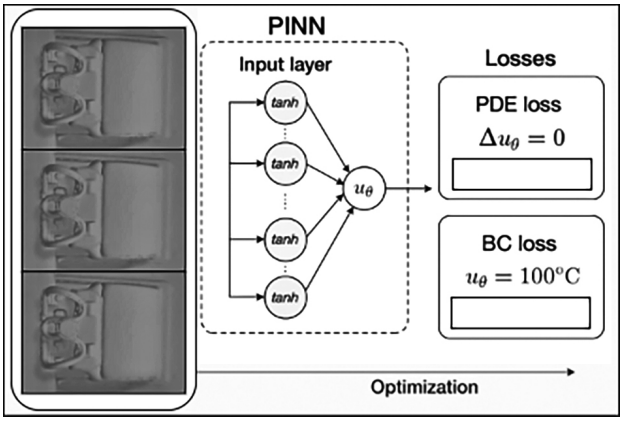


Figure 6. Architecture of physics-informed neural networks

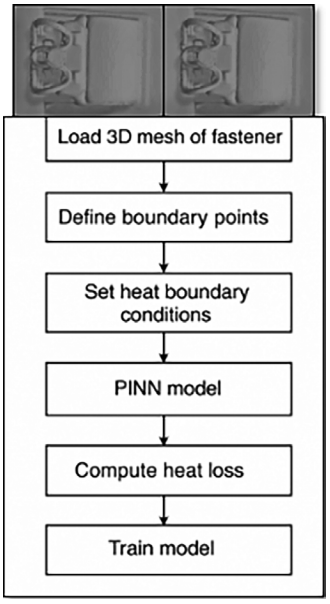


Figure 7. Flow-chart of thermal field modeling of fastener mesh

erence temperature applied on the thermally constrained surface region and does not appear in the interior governing equation.

(iii) Stress field modeling using PINNs

To model the **mechanical stress and deformation** in the rail fastener, we use the framework of **linear elasticity theory.** The computational domain is defined by the reconstructed three-dimensional fastener geometry obtained from the segmented point cloud and subsequent surface reconstruction. The boundary $\partial\Omega$ corresponds to the outer surface of the reconstructed mesh, with boundary points identified geometrically based on extremal surface locations. Dirichlet boundary conditions are imposed on a predefined subset, corresponding to a selected surface region of the fastener.

Assuming steady-state heat conduction and isotropic material behavior, the governing heat diffusion equation within the domain is expressed as:

$$\nabla.\sigma + b = 0, \text{ in } \Omega; \tag{24}$$

$$\sigma = C : \varepsilon, \text{ where } \varepsilon = \frac{1}{2}\left(\nabla u + \nabla u^T\right), \tag{25}$$

where $u\left(x, y, z\right) = \left[u_x, u_y, u_z\right]^T$ is the displacement vector field learned by the PINN. ε is the symmetric strain tensor derived from spatial gradients of displacement. σ is the Cauchy stress tensor, C is the elasticity tensor (determined by Young's modulus EEE and Poisson's ratio ϑ), b is the body force (set to zero in our case).

The PINN model $u_\theta\left(x, y, z\right)$ is trained to minimize a loss function that includes the PDE residual:

$$\mathcal{L}_{PDE} = \frac{1}{N_i}\sum_{i=1}^{N_i}\left(\nabla.\sigma\left(x_i, y_i, z_i\right)\right), \tag{26}$$

here, we calculate the divergence of the stress tensor through multiple levels of autograd (nested gradients). For Dirichlet boundaries (e.g., fixed ends of the fastener), the displacement vector is set to a known value u_o, contributing to a boundary loss:

$$\mathcal{L}_{BC} = \frac{1}{N_b}\sum_{j=1}^{N_b}\left(u_\theta\left(x_i, y_i, z_i\right) - u_o\right). \tag{27}$$

The trained model outputs a 3D displacement field across the mesh. From this, we analytically compute the full strain and stress tensors at each vertex or queried location. This modeling reveals key insights into deformation under load, identifies stress concentration zones, and helps validate the structural safety of the fastener under operational conditions. Additionally, the mesh output from the PINN can be visualized using color-mapped stress magnitude or vector displacement arrows, offering an intuitive understanding of structural behavior and facilitating integration with digital twin environments. Stress Field Modeling of Fastener Mesh is given in Figure 8.

The mechanical analysis is formulated over the same computational domain $\Omega \subset \mathbb{R}^3$, defined by the reconstructed fastener geometry. The boundary $\partial\Omega$ represents the outer surface of the solid fastener. Linear elasticity under small-deformation and static equilibrium assumptions is considered. Displacement Dirichlet boundary conditions are applied on a mechanically constrained subset $\partial\Omega_T \subset \partial\Omega$, corresponding to fixed or supported regions of the fastener, while the remaining boundaries are traction-free.

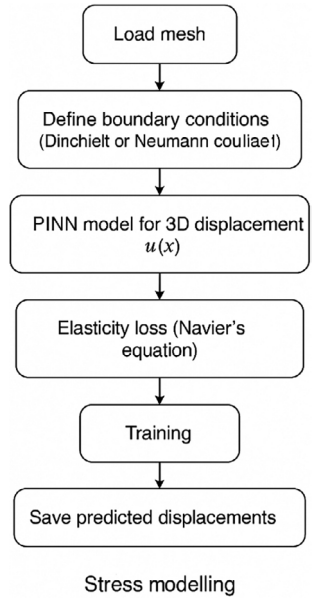


Figure 8. Stress field modeling of fastener mesh

2.7. Motivation for physics-informed learning over reconstructed geometry

Although the finite element method (FEM) remains the standard tool for solving partial differential equations over complex geometries, this work does not seek to replace FEM-based analysis. Instead, physics-informed neural networks (PINNs) are employed as a mesh-independent surrogate modeling framework capable of learning continuous solution fields directly from spatial coordinates under physics constraints.

The reconstructed high-quality mesh is not used for numerical discretization, as in FEM, but rather as a geometric support for sampling interior and boundary points. This distinction is critical: while FEM relies on element-wise integration and stiffness matrix assembly, the PINN formulation enforces the governing equations in a strong form through automatic differentiation, without requiring element connectivity or shape functions.

The use of PINNs in this context is therefore not motivated by efficiency over FEM, but by their ability to (i) provide continuous field representations over complex geometries, (ii) decouple geometry handling from solver formulation, and (iii) enable future extensions toward data–physics hybrid learning where partial observations or uncertain boundary conditions are present. Consequently, the current study should be interpreted as a feasibility analysis rather than a competitive replacement of conventional FEM solvers.

2.8. PINN architecture and training details

The physics-informed neural network (PINN) employed in this study is implemented as a fully connected multilayer perceptron (MLP). The network takes the three-dimensional spatial coordinates (x, y, z) as input and outputs a single scalar field variable representing the predicted physical quantity. The architecture consists of five hidden layers, each containing 64 neurons, with tanh activation functions used throughout the hidden layers to ensure smooth higher-order derivatives. The output layer is linear.

Spatial inputs are normalized to the range [−1, 1] based on the geometric bounds of the reconstructed mesh, while no explicit output normalization is applied, as training is driven exclusively by physics-based constraints. The network is trained using the Adam optimizer with a fixed learning rate of 1×10^{3}. Training is performed in a full-batch manner for 500 epochs, which was found sufficient for loss convergence in all experiments.

The PINN is implemented in PyTorch, and spatial derivatives required for enforcing the governing partial differential equation are computed using PyTorch's automatic differentiation (Autograd) framework. Interior collocation points are sampled directly from the vertices of the reconstructed mesh, while boundary points are identified based on geometric extremities and used where applicable for enforcing boundary conditions. Training is terminated after a fixed number of epochs, as the loss exhibits stable convergence without oscillatory behavior.

Although physics-informed loss terms inherently constrain the hypothesis space, they do not by themselves guarantee generalization. To mitigate overfitting, the proposed PINN is trained under strong physics supervision by enforcing the governing partial differential equation at all collocation points sampled from the reconstructed geometry. This acts as an implicit regularizer by restricting admissible solutions to physically consistent fields.

In addition, training convergence is monitored through the evolution of the physics residual loss, and stable monotonic decay without oscillatory behavior is used as an indicator of adequate generalization. Given the absence of labeled ground-truth field data, a classical data-driven validation split is not applicable. Instead, robustness is evaluated by verifying that the learned solution remains smooth and physically plausible across the entire spatial domain, including regions not explicitly constrained by boundary conditions.

No explicit weight regularization or dropout is employed, as these techniques may interfere with higher-order derivative accuracy in PINNs. The limitations of this validation strategy are acknowledged, and future work will incorporate boundary-aware validation and cross-geometry generalization studies.

The use of a physics-informed neural network (PINN) in this study is not intended to replace conventional finite element methods (FEM), nor to claim superior numerical accuracy. Rather, the motivation lies in exploring a mesh-independent modeling paradigm that can operate directly on reconstructed geometry without requiring volumetric meshing. In the context of complex railway components derived from point-cloud data, generating high-quality volumetric meshes suitable for FEM can be non-trivial and often requires substantial manual intervention.

By contrast, the PINN formulation enforces governing equations in strong form through automatic differentiation, allowing the physical field to be learned directly as a continuous function of spatial coordinates. This makes PINNs attractive as a geometry-aware surrogate modeling block within a digital-twin pipeline, particularly in scenarios where geometry is frequently updated or partially observed. The present work therefore positions PINNs as a complementary tool for exploratory physics-based modeling rather than as a replacement for established FEM solvers.

3. Results and discussion

Here in this section, this study explores point cloud downsizing/decimation and mesh generation in depth. We present the study's outcome, including the original and decimated point clouds. We then analyze the generated meshes, examining their properties in detail. This in-depth analysis delves into the key findings of the research. Following, we'll prove our argument in favor of efficient data handling, 3D simulations, and the development of data-driven digital twins.

3.1. Point cloud decimation analysis

This research focuses on analyzing rail fasteners using point cloud data. To optimize data processing and data utility, data decimation plays a central role. The researchers built a custom dataset specifically for this purpose. The data consists of 3D point clouds, as shown in Figure 9,

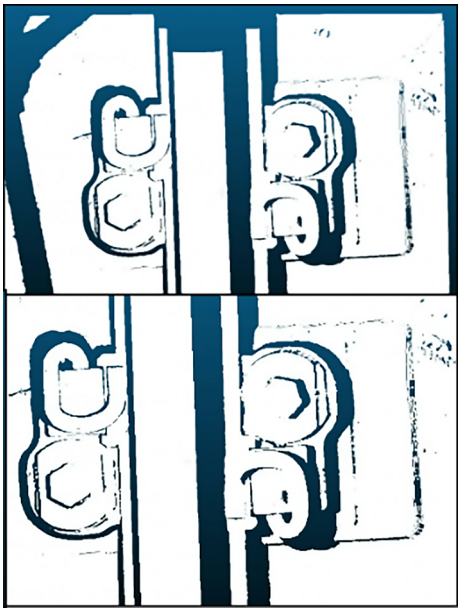


Figure 9. Original point cloud dataset

captured from high-speed railway tracks using an area-structured light camera mounted on an inspection trolley. These point clouds are highly detailed, with approximately 1.7 million points per object. 26 samples were available to perform this research. However, to facilitate efficient analysis and processing, the research employs a decimation technique.

This decimation process effectively reduced the number of points in the point cloud, while still preserving the key features and components of the rail fasteners. This approach not only addressed the challenge of handling large datasets but also achieved a smooth removal of unwanted points. A detailed analysis of the decimation process and its impact on the data is likely presented in Table 1.

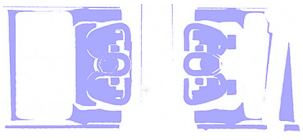
This table explores the impact of decimation on a fastener point cloud. Each row represents a different decimation level, resulting in a significant reduction in points. Analyzing "Average Centroid", "Bounding Box", and "Density" reveals minimal changes in overall shape despite the point count drop, indicating successful decimation that preserves key features. "Clusters" ensures no unintended structural alterations, while "Inliers/Outliers" helps assess data quality by showing the removal of outliers (noise) while keeping valuable data intact. This analysis helps researchers find the optimal balance between point reduction and data integrity for their specific use case. The visual impact of decimation is showcased in Figure 10. This figure highlights the significant reduction in points between the original and decimated point clouds.

The decimation process ultimately led to a more streamlined representation of the fastener point cloud. By removing redundant or irrelevant points, it achieved a cleaner output with improved clarity. This reduction in point count translates to lower computational costs and less model complexity during data processing. Figure 11 illustrates this lighter, decimated point cloud, which serves as the foundation for subsequent mesh generation.

The point cloud data's complexity was significantly reduced, resulting in a final count of 300,000 points per sample as compared to the collected one, i.e., 1.7 million points. This represents an 82.35% reduction compared to the original sample. Importantly, this decimation process achieved a high level of detail, preserving the critical information necessary for monitoring rail fasteners. More details are given in Table 2.

This iterative refinement process demonstrates the effectiveness of the data simplification method. Each iteration significantly reduces the number of data points while maintaining the key details of the fastener. Table 2 shows a clear distinction between inliers, which represent the actual fastener, and outliers, which are irrelevant data points. As the iterations progress, the proportion of inliers remains consistent around 30%, indicating the method focuses on the core data. Crucially, the proportion of outliers steadily decreases, highlighting the method's ability to eliminate extraneous information. This reduction in data points translates to a substantial decrease in processing time, as shown by the significant drop-in time taken from

Table 1. Decimation detail of fastener

Point Cloud	Points	Average Centroid	Bounding Box Diagonal	Bounding Box Volume	Density	Clusters	Inliers	Outliers
	276075	0.42	1.18	0.330	924288.11	14	274077	1998
	238107	0.34	0.80	0.071	797172.76	14	232288	5819
	190153	0.35	0.64	0.039	636624.67	9	182975	7178
	237653	0.33	0.66	0.049	795652.78	8	229056	8597
	337894	0.33	0.91	0.152	1131255.66	19	330377	7517

the first iteration to the fourth. While data size reduction isn't directly displayed, it can be inferred from the inlier ratios. The method likely achieves a significant decrease in data size, potentially exceeding 80%, making storage and analysis significantly more manageable. Overall, this process showcases the method's ability to dramatically simplify point cloud data while preserving crucial information, leading to faster analysis and more efficient inspection of rail fasteners.

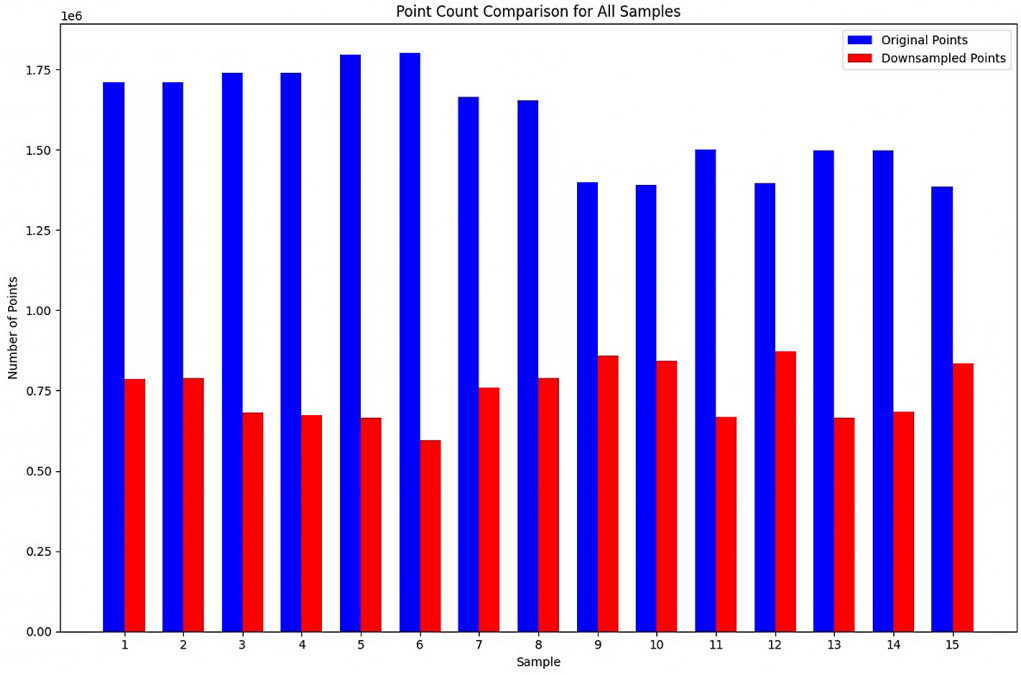


Figure 10. Original and decimated point cloud

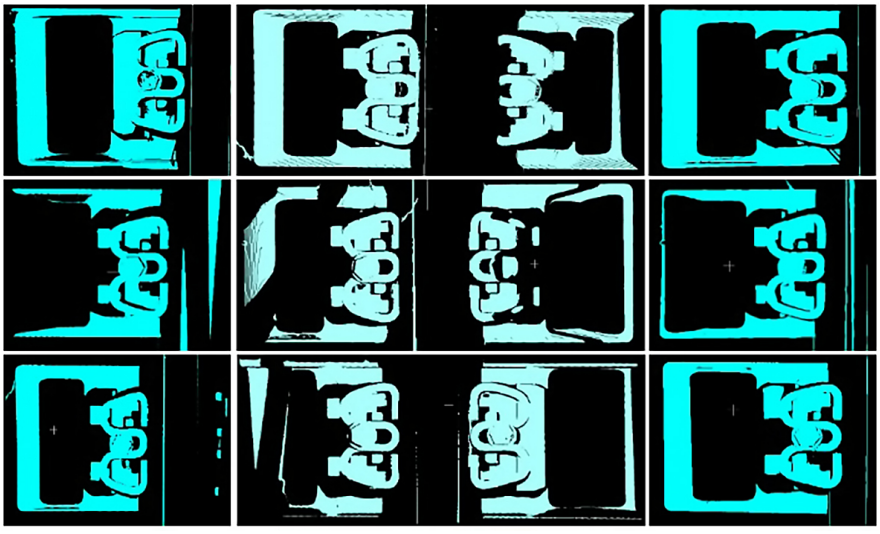


Figure 11. Decimated fastener point cloud

Table 2. Data simplification performance – Reduction in outliers and processing time

Iterations	Inliers	Outlier	Inliers Ratio	Outlier Ratio	Mean PFE	MSE PFE	RMSE PFE	Time taken (seconds)
1	471001	916443	0.34	0.66	0.09	0.02	0.12	5.53
2	339540	576903	0.37	0.63	0.06	0.01	0.09	3.64
3	222174	354729	0.39	0.61	0.11	0.02	0.14	1.24
4	117512	237217	0.33	0.67	0.03	0.00	0.04	1.12

3.2. Mesh generation analysis

This section presents the outcomes of the generation of 3D meshes from point cloud data of rail fasteners. We already addressed the computational challenges associated with dense point clouds by employing decimation. Following decimation, the preprocessed point cloud undergoes mesh generation. This process of generating a 3D mesh effectively constructs a high-fidelity mesh that accurately reflects the underlying geometry of the fastener, as shown in Figure 12.

To refine the generated mesh and enhance its usability, we implement a series of post-processing steps. These techniques resulted in the generation of the 3D mesh of rail fasteners. The detailed analysis is given in Table 3.

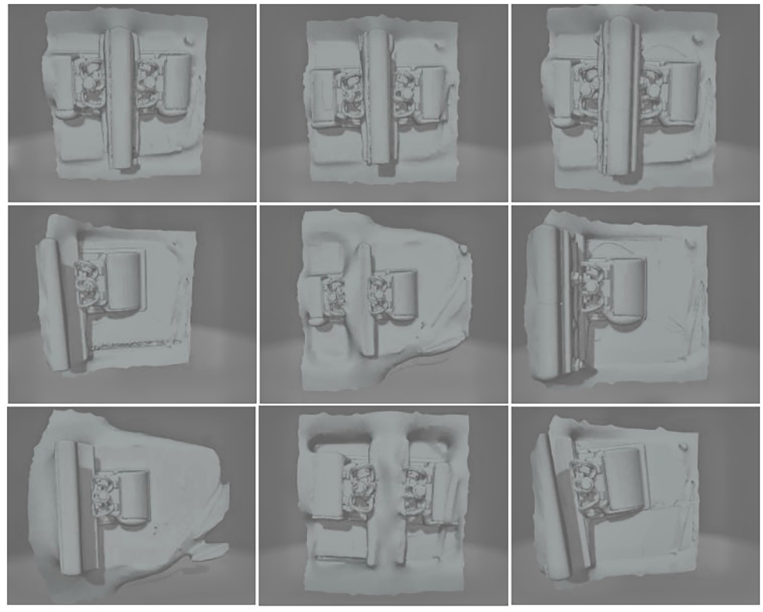


Figure 12. Fastener-generated mesh with rail track

Table 3. Mesh generation analysis

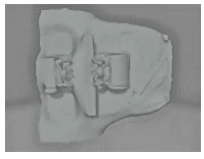
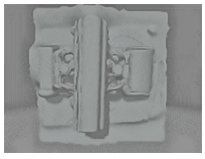
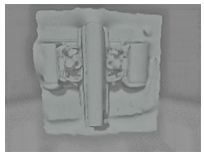
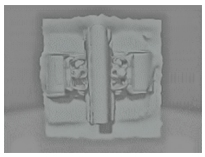
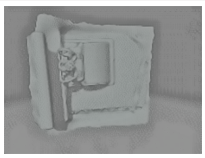
Mesh	Vertex Count	Face Count	Surface Area	Density (mean)
	24965	49999	0.85	9.092
	24838	49999	0.8231	8.946
	24756	50000	0.90	8.85
	24753	49999	0.92	8.84
	24980	50000	0.72	8.66

Table 3 summarizes the key properties of a 3D mesh generated from the decimated point cloud data. The "Vertex Count" column shows the total number of points used to define the mesh's surface. A higher vertex count generally indicates a more detailed and intricate mesh. Paired with the "Face Count", which reflects the number of individual polygons forming the surface, these values provide insights into the overall complexity of the mesh. The "Surface Area" provides the total area of the mesh, typically measured in square units. A larger surface area often corresponds to a more detailed mesh with a higher vertex and face count. Finally, "Density (mean)" might represent the average density of the mesh, calculated as the ratio between the number of vertices and the mesh's. A higher density generally suggests a more intricate mesh with a greater number of points per unit volume.

By analyzing these metrics together, we can gain valuable insights into the level of detail and complexity achieved in the generated mesh. This information is crucial for determining the mesh's suitability for various applications, such as tasks requiring precise geometric representation versus those prioritizing processing efficiencies. Therefore, our results show an exceptional number in terms of analysis and visibility as well. The generated mesh visualized from different angles is given below in Figure 13.

Through optimization of both visualization and complexity, this efficient method has produced a final, clean 3D mesh. After ensuring compatibility with various software programs, this mesh facilitates simulations, real-time maintenance, and monitoring within a digital twin application.

Analyzing surface matching across ten samples (Samples 1-10) reveals the significant improvement achieved through data simplification. Distance metrics like Chamfer Distance, Hausdorff Distance, and Point-to-Surface Distance all show a clear downward trend. This signifies a progressive improvement in how well the simplified data aligns with the original surface – a key indicator of successful matching. Further analysis has been carried out to show the potential of this model. See Table 4 for more details.

Encouragingly, the Surface Normal Consistency metric remains stable across the samples. This suggests that simplification effectively reduces data points while preserving the overall geometric properties crucial for accurate matching. An additional benefit is the improved efficiency – processing time noticeably changes across the samples. This highlights that the simplified data requires less processing power for surface matching tasks, leading to faster analysis. Statistical analysis reinforces these positive observations. The low standard deviation across all metrics indicates consistent improvement with each sample. Additionally, the positive skewness for distance metrics suggests a stronger tendency towards achieving lower discrepancies as simplification progresses.

A graph as shown in Figure 14, where the x-axis represents the simplification process and the y-axis represents the distance metrics. The graph would ideally show a downward trend in these metrics as simplification progresses, visually confirming the improvement in surface matching accuracy.

3.2.1. Comparison with other techniques

Analyzing the efficiency of surface matching techniques is crucial for rail fastener inspection. This section compares the processing time and accuracy achieved by different methods, as shown in Table 5.

While several geometric representations can be derived from point-cloud data, not all are suitable for faithful surface reconstruction of complex, non-convex railway components. In this study, Poisson surface reconstruction is adopted as the primary surface generation technique

Table 4. Surface matching metrics – Improvement with data simplification

Samples	Chamfer Distance	Hausdorff Distance	Point-to-Surface Distance	Surface Normal Consistency	Time Taken (Seconds)
1	2.9916	2.0215	1.1379	0.5116	56.88
2	2.5689	1.7863	0.9250	0.5433	29.92
3	2.1584	1.1642	0.8126	0.5241	33.67
4	2.1467	1.1551	0.8068	0.5209	33.18
5	2.6644	1.4104	1.0335	0.4717	38.14
6	2.6803	1.4241	1.0426	0.4841	35.33
7	2.1357	1.1685	0.7871	0.4867	39.51
8	2.1634	1.1523	0.7910	0.5209	34.18
9	2.6977	1.4393	1.0012	0.3622	36.06
10	2.1924	1.2025	0.8408	0.4769	35.85
Statistical Metrics					
Average	2.4322	1.3411	0.9333	0.4931	35.94
STD	0.3421	0.2345	0.1442	0.0635	4.18
Skewness	0.2341	0.1234	0.0987	0.0678	0.3456
Kurtosis	2.3456	1.5678	1.2345	1.0123	1.6789

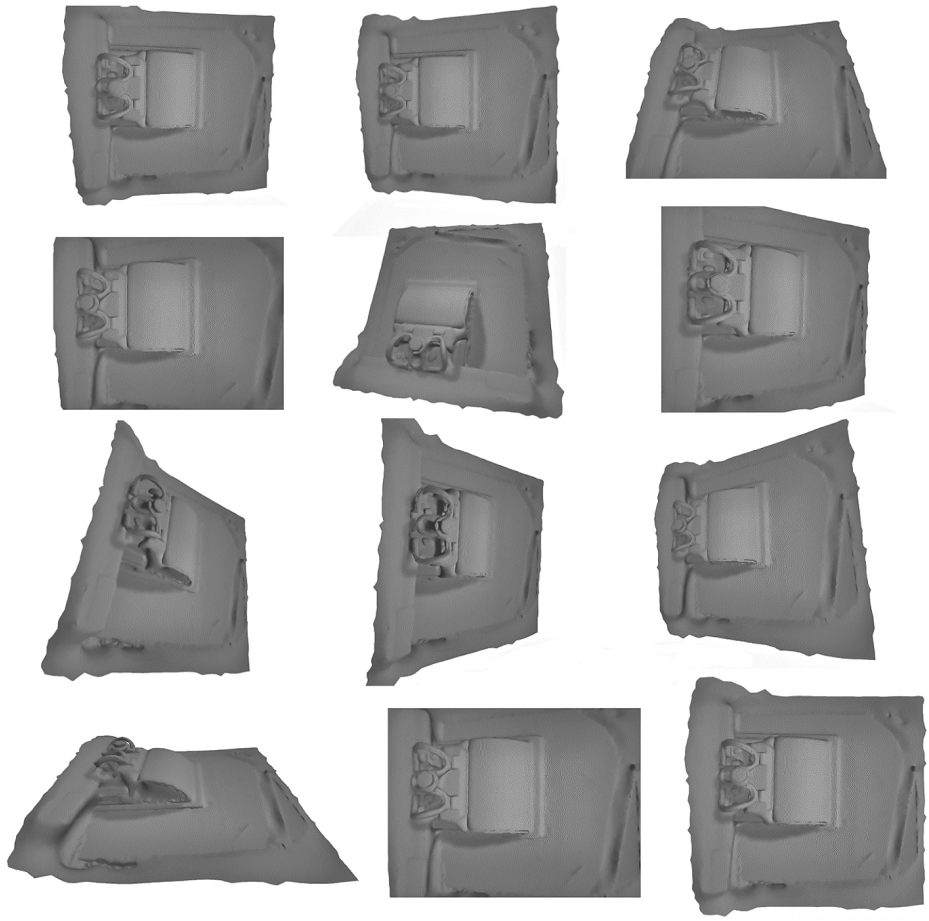


Figure 13. Multiple views of 3D generated mesh

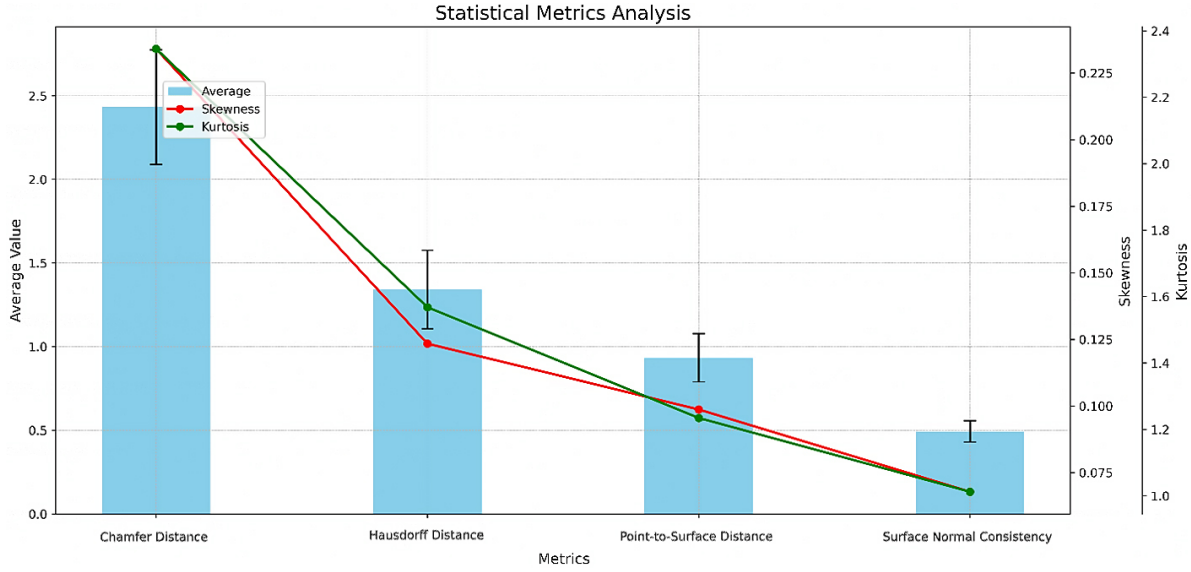


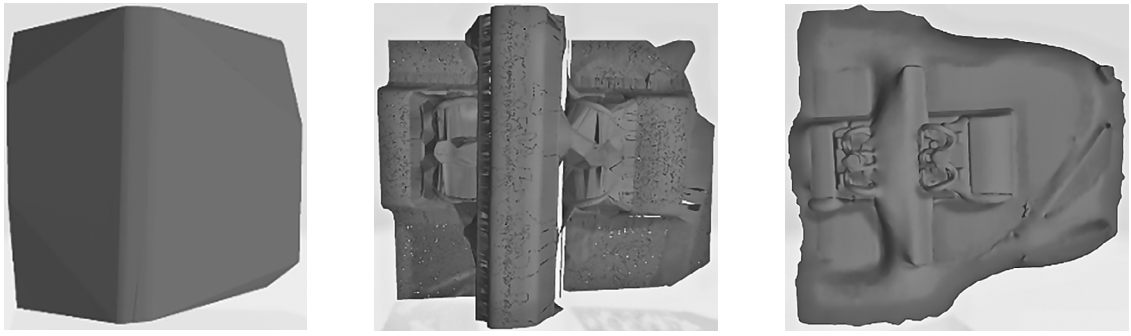
Figure 14. Mesh generation metrics on different samples

due to its ability to produce smooth, watertight meshes that accurately capture the underlying geometry of rail fasteners.

For reference, simpler geometric constructs such as convex hulls (Gao et al., 2011) and alpha shapes (Cao et al., 2024a) are included only as illustrative baselines to highlight the limitations of envelope-based representations when applied to non-convex geometries. In particular, convex hulls provide only an outer bounding approximation and cannot represent concavities or internal geometric features, making them unsuitable as surface reconstruction methods for rail fasteners.

Table 5. Comparison with other techniques on the same dataset

Method	Chamfer Distance	Hausdorff Distance	Point-to-Surface Distance	Surface Normal Consistency	Time Taken (Seconds)
Alpha Shape	0.4712	0.7462	0.0029	//	100+
Convex Hull	0.4877	0.7462	0.0109	//	100+
*Poisson	2.4322	1.3411	0.9333	0.4931	35.94



Convex Hull Alpha Shape Poisson Surface

Figure 15. Comparison of geometric representations derived from the same point cloud

In contrast, Poisson surface reconstruction leverages oriented surface normals and a global implicit formulation to recover detailed surface geometry, preserving curvature and fine-scale features that are critical for subsequent physics-based analysis and digital-twin modeling (Figure 15).

Convex hull and alpha shape are shown as geometric envelopes for reference, while Poisson surface reconstruction provides a detailed, non-convex surface representation suitable for physics-informed modeling.

3.2.2. Segmentation

The segmentation stage is introduced as a preconditioning and semantic-isolation step, rather than as an end task in itself. Its primary purpose is to separate the rail fastener geometry from surrounding infrastructure and spurious structures present in the raw point cloud, thereby improving the robustness of subsequent surface reconstruction and physics-informed modeling. Without segmentation, irrelevant planar components such as the track base or mounting surfaces would be incorporated into the reconstructed mesh, leading to distorted geometry and physically meaningless regions in the learned field.

Segmentation therefore serves three key objectives in this work: (i) reducing geometric clutter and outliers prior to mesh generation, (ii) isolating the fastener as the region of interest for physics-based analysis, and (iii) controlling point density and structural complexity to stabilize Poisson surface reconstruction and downstream PINN training.

The sheer volume of points in such datasets can overwhelm available computational resources. To address this issue, we propose a machine learning-based approach that utilizes mesh generation from the point cloud data. This paves the way for future advancements in automated analysis and monitoring of rail fasteners. A crucial step in this process involves segmentation, which is used to isolate the fastener geometry and remove surrounding planar structures that would otherwise degrade mesh reconstruction and physics-informed field modeling. Specifically, segmentation separates the rail fastener itself from the surrounding infrastructure, such as the base or track. This distinction is valuable for tasks like object recognition and scene understanding within the railway environment. To illustrate the segmentation process in detail, we present a series of histograms representing the X, Y, and Z coordinates of the point cloud data at various stages of segmentation. These histograms depict the distribution of points along each axis throughout the iterative process. The RANSAC algorithm with a distance threshold of 0.01 is employed to isolate planar regions from the initial point cloud. This iterative process progressively removes outliers, leaving behind the inliers (points of interest). The first set of histograms showcases the initial distribution of points across all three axes in the raw point cloud, providing insights into its overall spatial extent and density.

The results of the first segmentation step are visualized in Figure 16. The second column displays histograms of the inliers, points that belong to the detected planar region. These histograms clearly show how the inliers cluster along a specific plane. In contrast, the third column depicts the outliers (remaining points) through their histograms. These outliers exhibit a more scattered distribution, indicating they don't conform to the plane identified during segmentation.

After removing the first set of inliers, segmentation is repeated on the remaining points (outliers from the previous step). Figure 17 (Column 1) shows the initial point distributions for this second segmentation. Column 2 depicts the histograms of the newly identified inliers, representing another planar region within the data. Finally, Col-

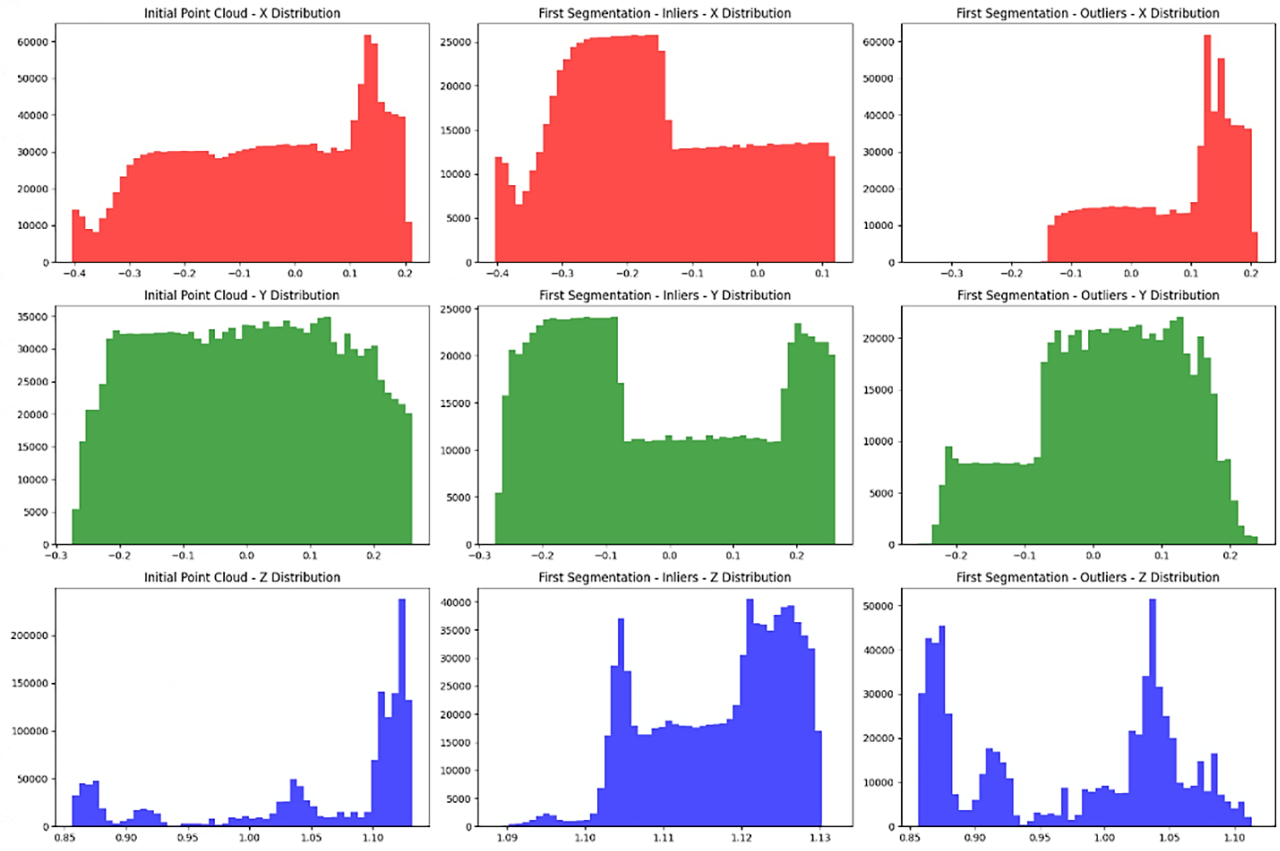


Figure 16. First segmentation

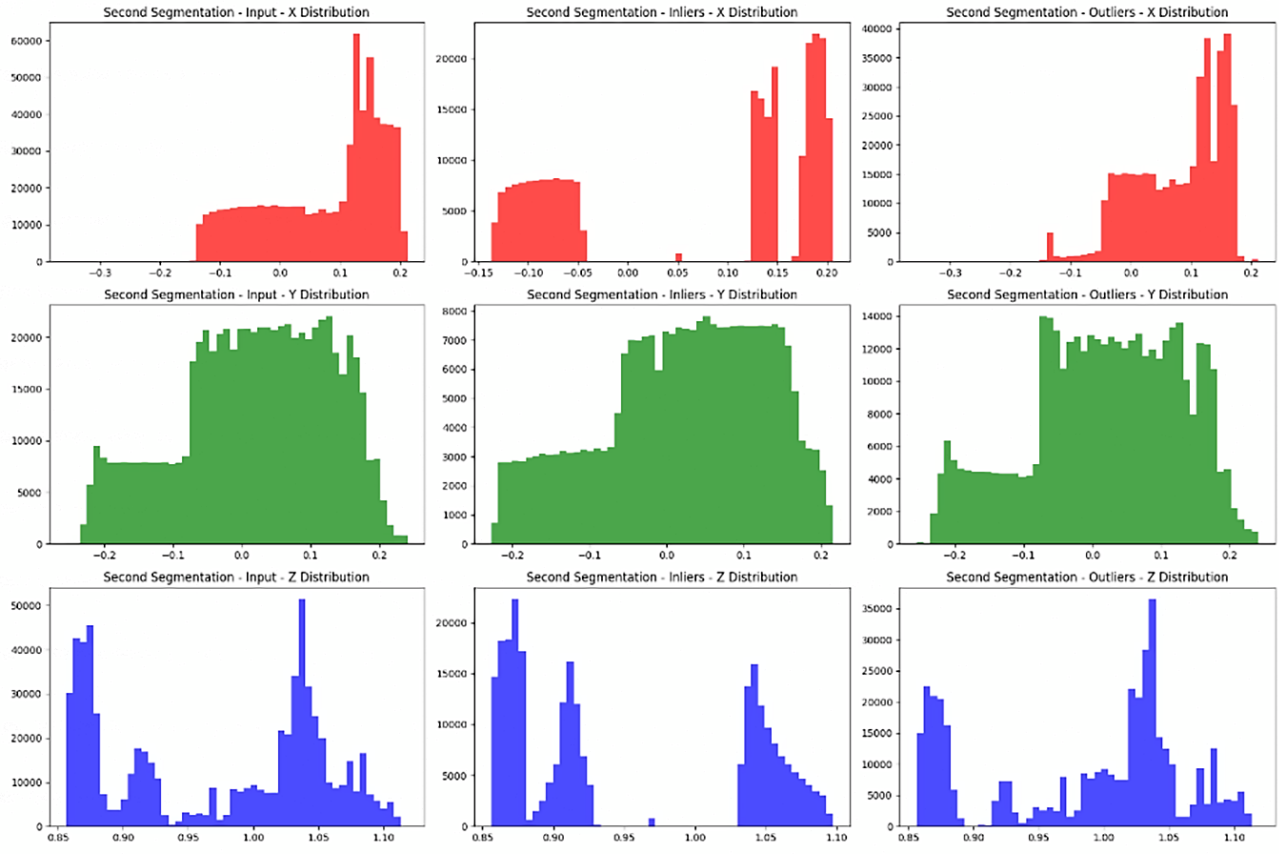


Figure 17. Second segmentation

umn 3 showcases the outliers remaining after this second segmentation, highlighting a further reduction in the total number of points.

The segmentation process iterates further on the remaining outliers from the second step. Figure 18 (Column 1) displays the initial point distributions for this third segmentation. Column 2 depicts the histograms of the newly identified inliers, representing yet another planar region within the data. Finally, Column 3 showcases the final set of outliers, highlighting the points that still haven't been assigned to any detected planes.

A final segmentation step is applied to the remaining outliers from the third iteration (Figure 19, Column 3). Figure 19 – Column 1 displays the inliers identified during this step. Notably, the histograms reveal that even after multiple iterations, a significant number of outliers persist. This highlights the complex composition of the point cloud, likely containing various planar and non-planar structures.

To achieve a high-quality 3D model, the process undergoes several refinement steps after the initial segmentation. First, duplicate vertices and problematic triangles are removed, ensuring a clean and structurally sound mesh. Taubin smoothing then tackles minor imperfections, preserving crucial details. To optimize file size and processing, quadric decimation simplifies the mesh by iteratively removing unnecessary points with minimal shape impact. Following this, a connectivity analysis checks for and fixes any structural issues that could hinder further analysis. Finally, the process concludes by isolating the Region of Interest (ROI) within the fastener. This step removes extraneous data, focusing solely on critical parts like the head or threads, and optimizes the model for its intended use, as shown in Figure 20.

By implementing this comprehensive pipeline of decimation, mesh generation, and post-processing techniques, we successfully transform raw point cloud data into high-quality and informative meshes. These refined meshes serve as a valuable foundation for various geometric processing tasks and detailed visualization of rail fasteners.

3.3. Analysis of thermal and stress distribution field

In this study, Physics-Informed Neural Networks (PINNs) were utilized to model and predict the steady-state thermal and stress fields over a 3D domain defined by mesh geometry. The results were analyzed both qualitatively and quantitatively using predicted field values and training metrics. Visualizations of the predicted temperature and displacement distributions were generated to better understand the physical behavior across the domain. Training convergence was evaluated through loss decomposition into PDE and boundary condition (BC) components, highlighting the model's ability to satisfy physical laws and boundary constraints. Below, we separately discuss the results of the thermal and mechanical stress modeling. The analysis is shown in Figure 21.

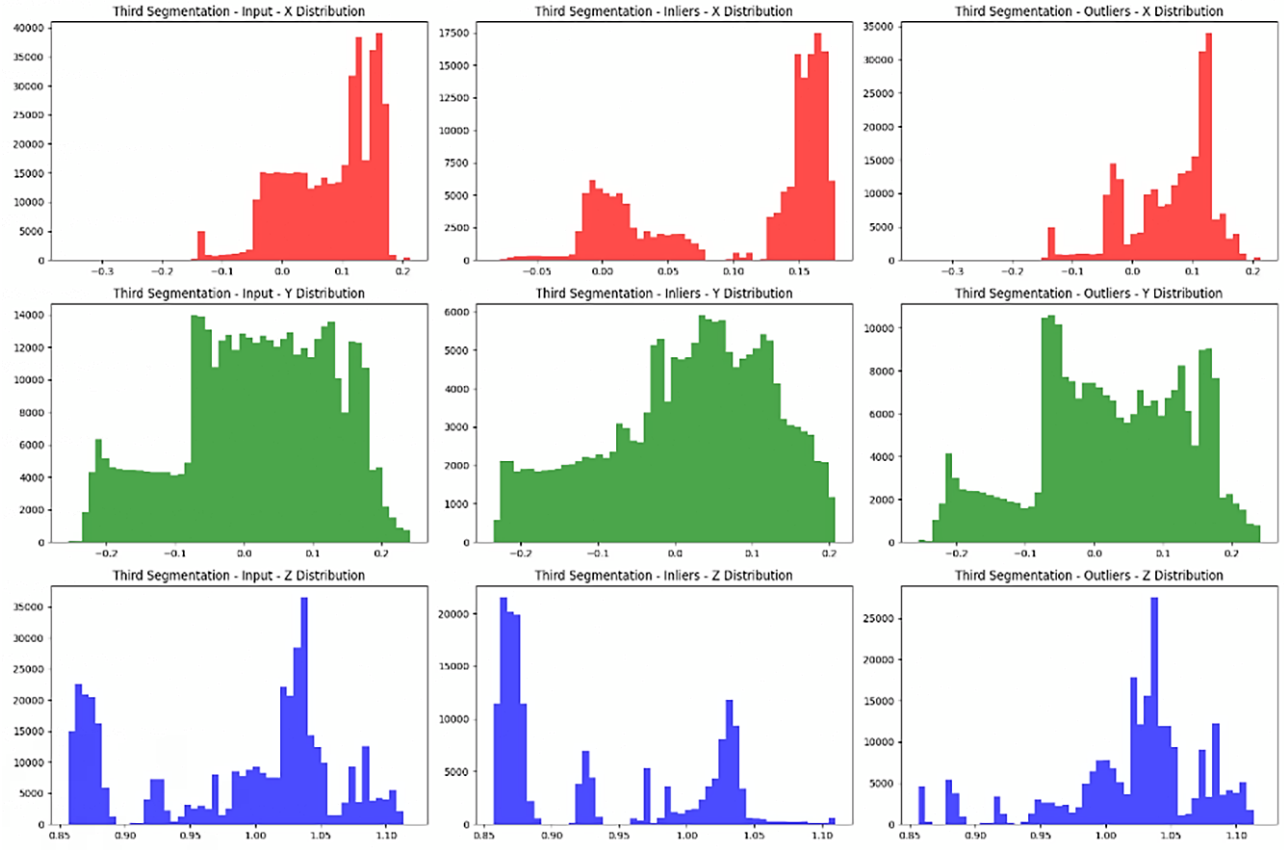


Figure 18. Third segmentation

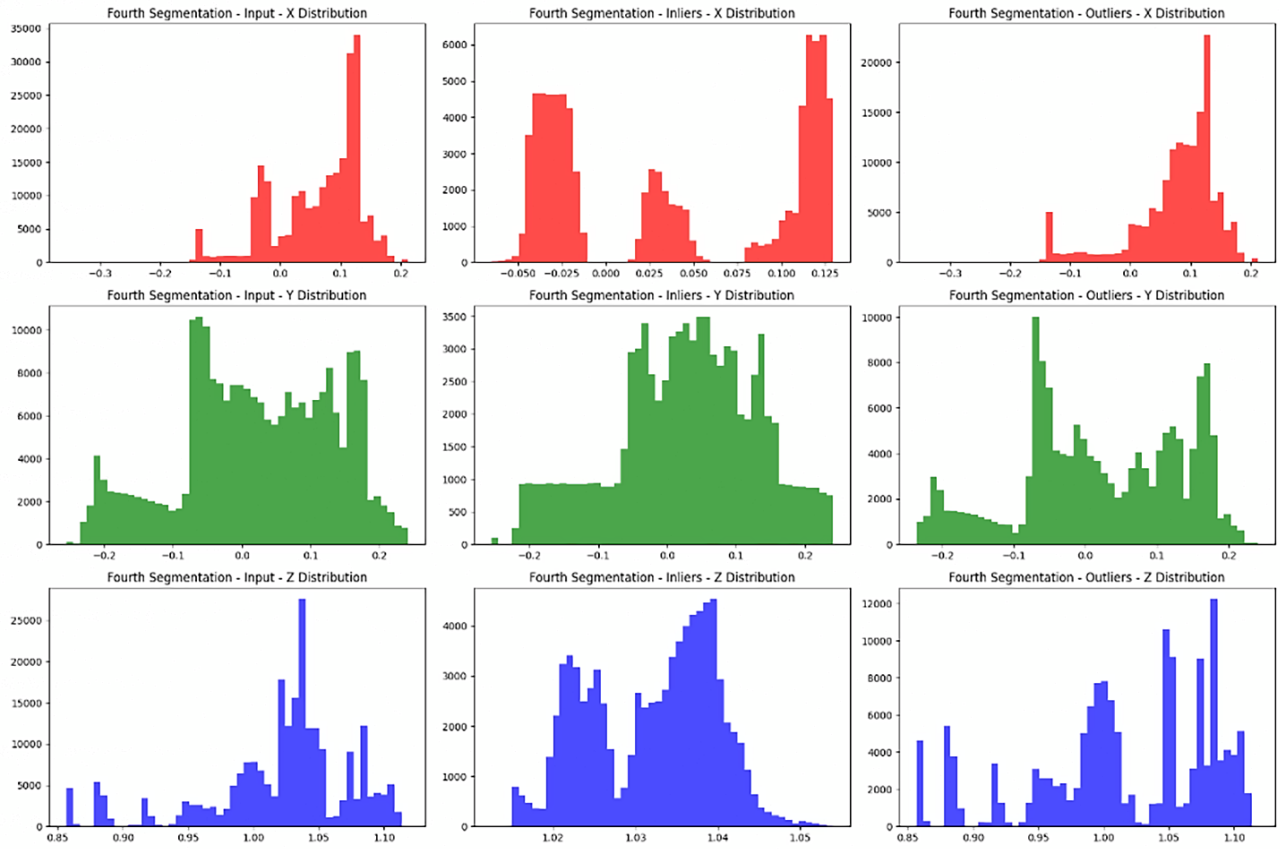


Figure 19. Fourth segmentation

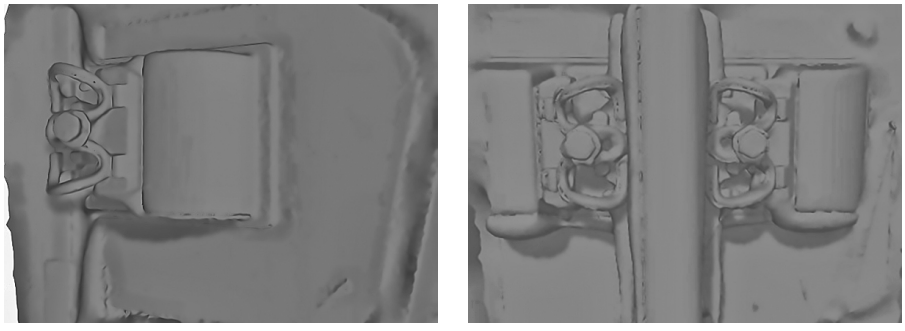


Figure 20. Extracted fastener 3D mesh

3.4. Analysis of thermal field distribution results

The predicted thermal field revealed a consistent temperature distribution across the domain, governed by the imposed boundary conditions and the steady-state heat equation, as shown in Figure 22. Visualization of the temperature magnitude in 3D showed a smooth gradient that aligns with the physical expectations of conductive heat transfer. The computed distribution plot illustrated how temperature values were concentrated in specific regions, with gradual transitions indicating a physically plausible thermal profile.

Further analysis using statistical plots highlighted the spatial temperature variation along key geometric axes. Kernel Density Estimation (KDE) and scatter plots provided insights into how temperature evolved relative to spatial depth (Z-direction), confirming that the model successfully captured directional thermal trends. Additionally, the training loss plots showed that both PDE loss and boundary condition loss decreased over time, validating the model's adherence to the physical constraints and convergence during learning.

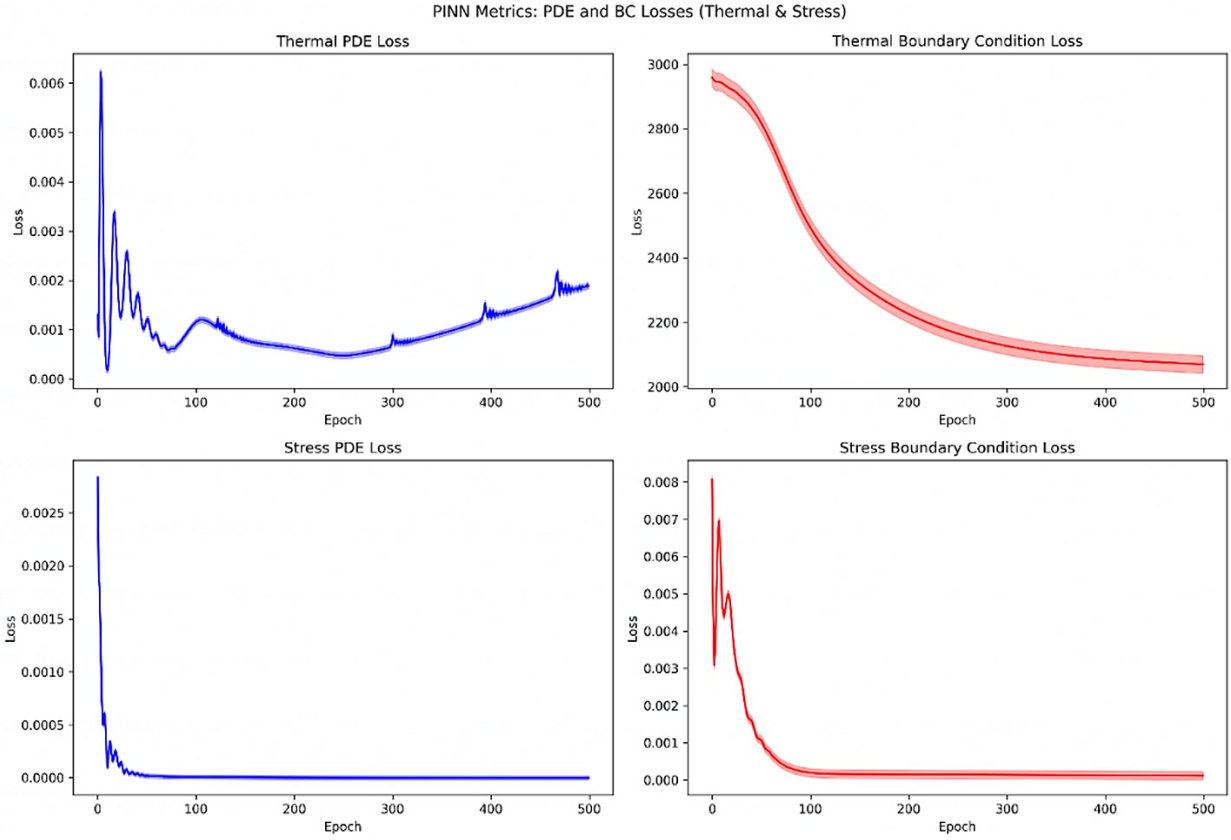


Figure 21. PINN metrics evaluation

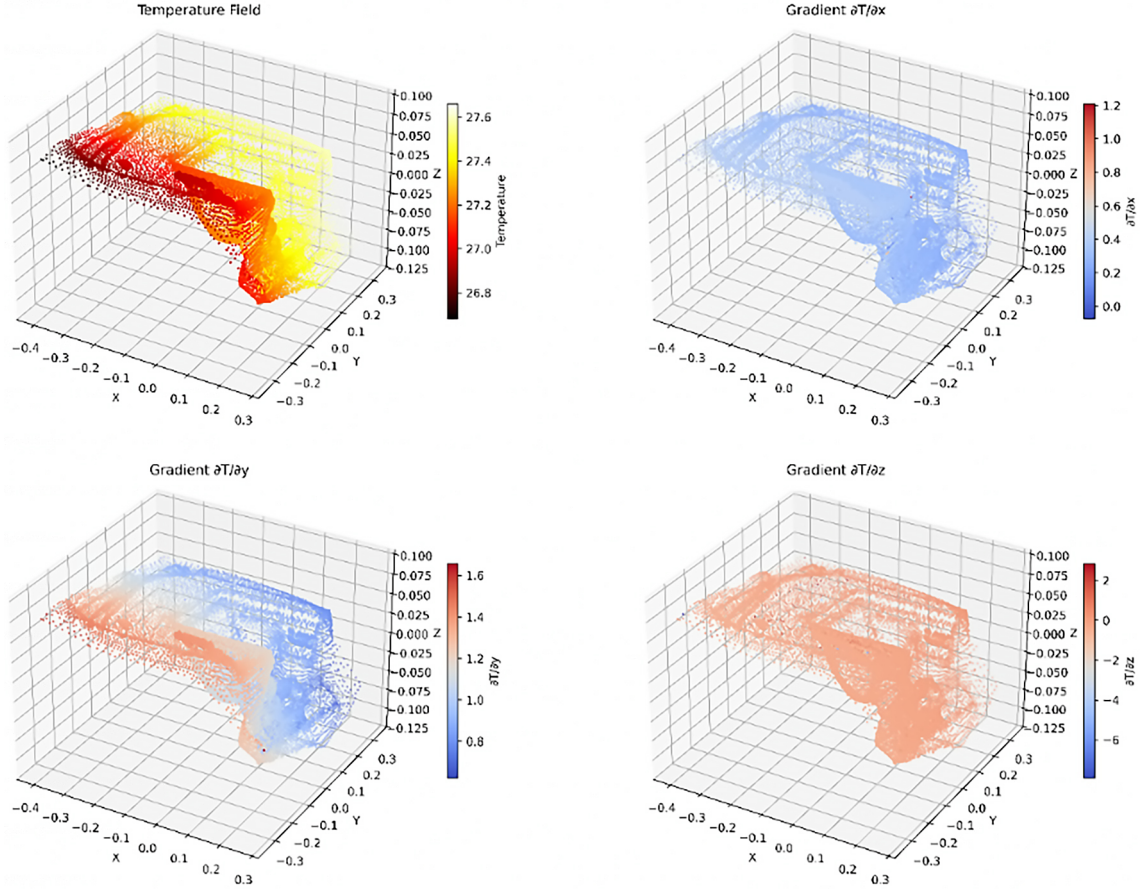


Figure 22. Thermal PINN temperature and gradient field

3.5. Analysis of stress field distribution results

The results for the stress field were obtained by predicting the 3D displacement vector at each vertex of the mesh, which was then analyzed in terms of its components and overall magnitude. The visualization of displacement magnitude and its individual components $\left(u_x, u_y, u_z\right)$ revealed deformation patterns that are characteristic of the material response under the applied conditions. The spatial continuity and smoothness in these plots suggest that the model effectively approximated the elastic behavior of the domain.

Statistical plots, including histograms, KDEs, and correlation heatmaps, provided deeper insight into the relationships between different components of the displacement field. The displacement magnitude distribution indicated a physically consistent deformation range, with higher values near boundary-constrained regions. Training loss analysis confirmed convergence, with a gradual reduction in PDE and BC losses, demonstrating that the PINN model successfully minimized Navier's elasticity residuals while respecting boundary constraints, as shown in Figure 23 and Figure 24.

3.6. Future recommendations

The ability to generate high-quality meshes, as described previously, opens exciting possibilities for the future of structural analysis, particularly within the realm of digital twins. Here are some key recommendations:

- *Real-Time Structural Simulations*: These refined meshes pave the way for real-time simulations of critical structural components like bridges, fasteners, columns, and beams. By incorporating real-time sensor data from the physical structure into the digital twin, engineers can analyze how the structure behaves under various conditions, leading to more informed decisions and improved safety protocols.
- *Cost and Time Reduction*: This technology has the potential to significantly reduce the cost and time associated with traditional structural analysis methods. Modeling in software can be expensive and time-consuming, while simulations using refined meshes offer a faster and more cost-effective alternative for evaluating structural performance.
- *Software Compatibility*: To maximize its usability, it's crucial to ensure compatibility between the generated mesh and widely used structural analysis software like Ansys and Abaqus. This compatibility will allow engineers to seamlessly integrate the mesh into their existing workflows.
- *Advanced Analysis with Multi-Factor Integration*: The future of this technology lies in incorporating additional factors beyond geometry into the mesh. Adding factors like load variations, temperature fluctuations, and material properties will enable even more comprehensive structural analyses. These simulations can predict how a structure responds to real-world conditions, leading to more robust and resilient designs.

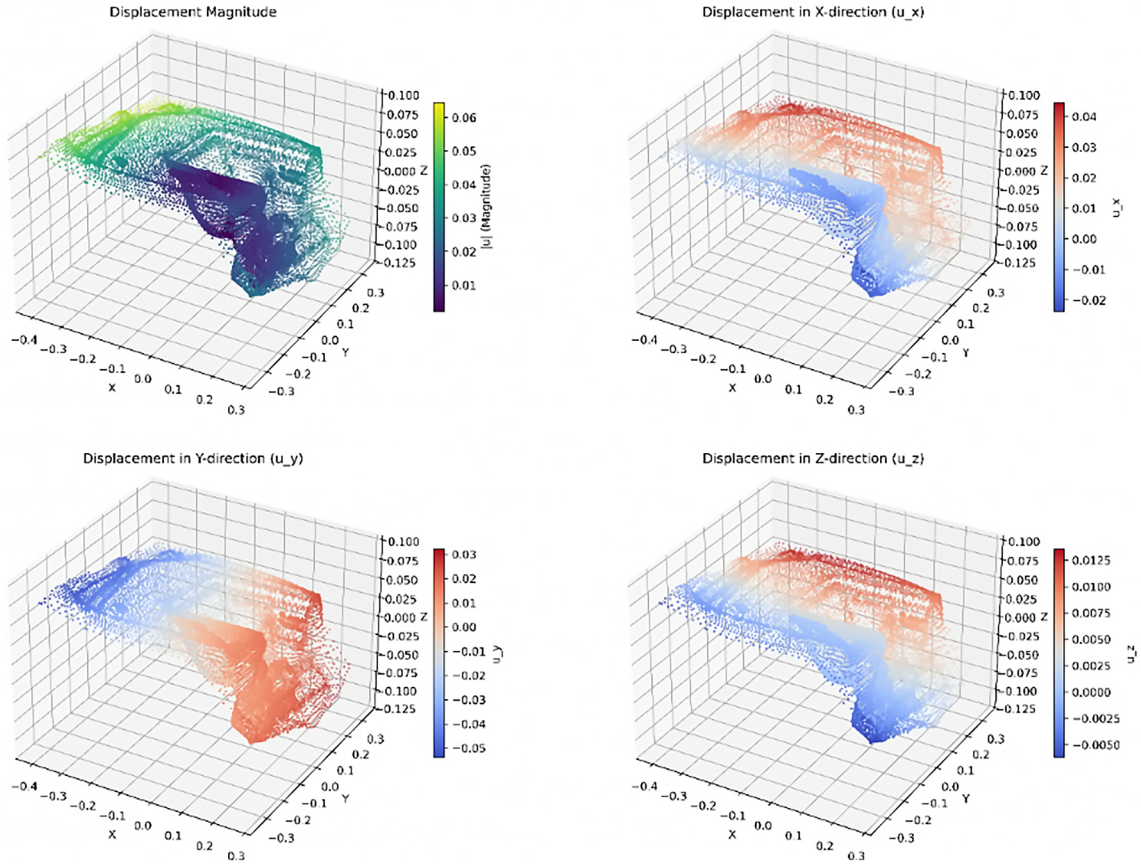


Figure 23. Displacement field visualization

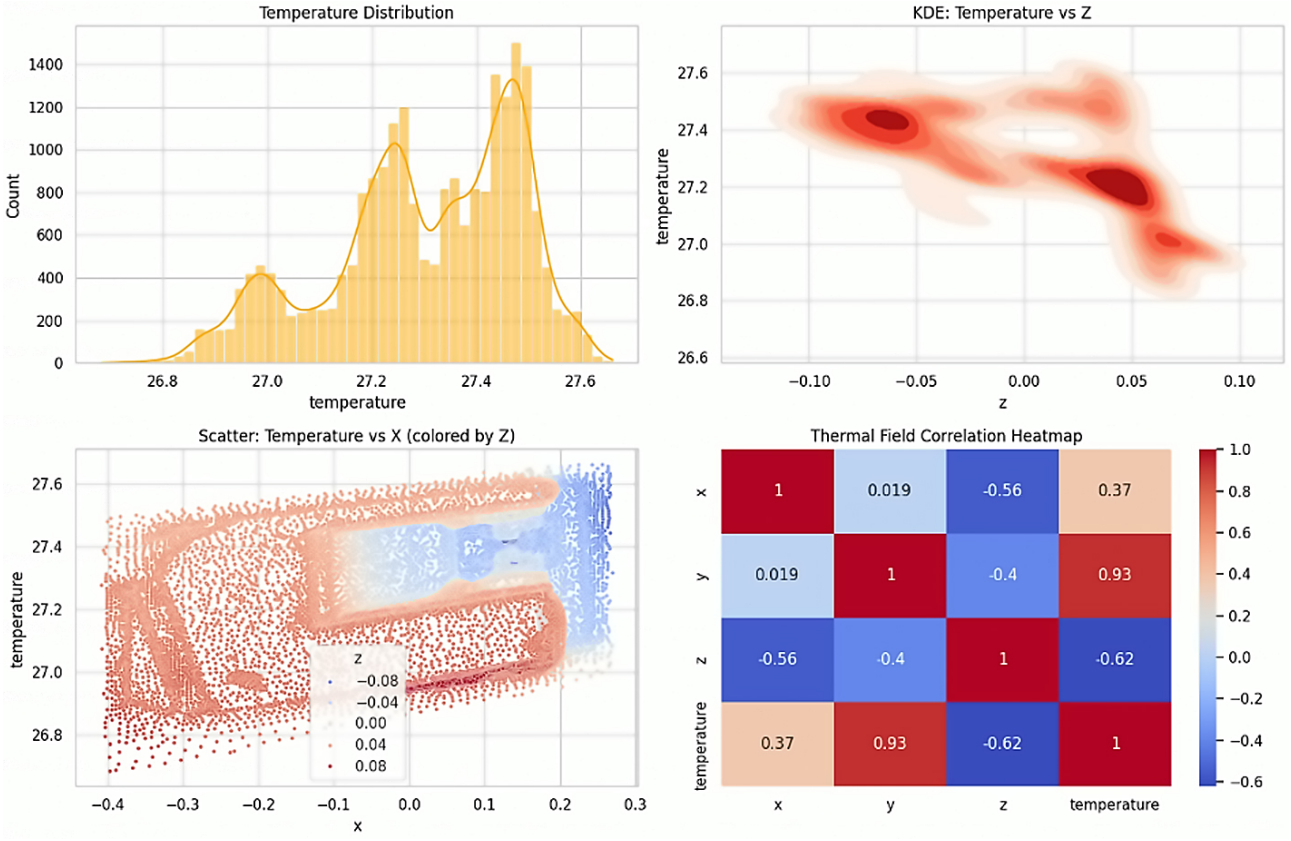


Figure 24. Thermal field visualization

- *Digital Twins for Infrastructure Monitoring*: These refined meshes can be instrumental in creating digital twins for infrastructure monitoring. By integrating the mesh with sensor data from the actual structure, the digital twin can continuously monitor its health and detect potential issues early on. This allows for preventative maintenance and helps to avoid costly infrastructure failures.

By implementing these recommendations, researchers can leverage the power of refined meshes to revolutionize structural analysis, ushering in an era of real-time simulations, reduced costs, and enhanced infrastructure monitoring through digital twins.

The proposed pipeline is implemented and evaluated as an offline modeling framework, and no real-time or online digital-twin deployment is claimed in the present study. Accordingly, the results should be interpreted as a foundational step toward physics-aware digital-twin construction rather than a fully operational digital-twin system.

Nevertheless, the modular design of the pipeline lends itself naturally to future near-real-time digital-twin integration. In a practical deployment, point-cloud updates obtained from periodic inspections or sensing campaigns could be incrementally incorporated into the geometric reconstruction stage, while the PINN could be retrained or fine-tuned using warm-start initialization to avoid full retraining. Given that PINN training is the most computationally intensive stage, update frequencies would be governed by asset-level decision cycles rather than strict real-time constraints.

Issues such as continuous data ingestion, online model updating, and overfitting prevention in streaming scenarios are beyond the scope of the present work and will be addressed in future studies focusing explicitly on operational digital-twin implementations.

4. Conclusions

High-resolution point cloud data of rail fasteners, while rich in geometric detail, often demand substantial computational resources for processing, limiting their practical use in large-scale or commercial applications. This study presents an integrated framework that overcomes these challenges by combining targeted geometric simplification with physics-informed simulation. The pipeline begins with RANSAC-driven iterative segmentation, which efficiently removes irrelevant or noisy points while preserving the fastener's essential geometric features, achieving an 82.35% reduction in point count. The reduced dataset forms the basis for high-fidelity 3D mesh reconstruction using Poisson surface reconstruction, followed by refinement and normalization to ensure compatibility with downstream analysis tools. Building on this geometric foundation, the workflow integrates Physics-Informed Neural Networks (PINNs) to simulate thermo-mechanical behaviors directly on the generated mesh. This enables accurate prediction of temperature fields, stress distributions, and deformation

patterns without the computational overhead of modeling the entire track assembly. By unifying decimation, mesh generation, and PINN-based simulation, the proposed approach supports the creation of lightweight yet structurally accurate digital twins of rail fasteners. These twins can be used for real-time condition monitoring, predictive maintenance, and rapid what-if scenario testing, offering significant time and cost savings for infrastructure management. This research demonstrates that careful integration of geometric processing and physics-based modeling not only optimizes computational efficiency but also enhances the practicality of digital twin systems for railway asset monitoring. The methodology provides a scalable, adaptable foundation for extending similar workflows to other complex infrastructure components, advancing the state of intelligent, data-driven structural health assessment. A direct quantitative comparison with a finite element method (FEM) solution on the same geometry and boundary conditions is not included in the present study. This is primarily due to differences in problem formulation and scope: the proposed PINN framework operates as a mesh-independent, continuous field approximator using strong-form physics constraints, whereas FEM relies on element-wise discretization and stiffness matrix assembly. Consequently, the objective of this work is not to benchmark numerical accuracy against FEM, but to demonstrate the feasibility of embedding a physics-informed learning block within a geometry-driven digital-twin pipeline. Quantitative FEM–PINN error analysis and experimental validation using temperature or displacement measurements are recognized as important next steps and will be addressed in future work focused on validation and deployment. The current study should therefore be interpreted as a methodological exploration rather than a replacement or validation study of established numerical solvers.

Acknowledgements

The research is supported by the National Natural Science Foundation of China (No. 52178442).

Data availability statement

The data used to support the findings of this study are available from the corresponding author upon request.

Conflict of interests

The author declared that there is no conflict of interest.

References

Agrawal, T., & Choudhary, P. (2023). Segmentation and classification on chest radiography: a systematic survey. *Visual Computer*, *39*(3), 875–913. https://doi.org/10.1007/s00371-021-02352-7

Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., & Silva, C. T. (2003). Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, *9*(1), 3–15. https://doi.org/10.1109/TVCG.2003.1175093

Aljumaily, H., Laefer, D. F., & Cuadra, D. (2017). Urban point cloud mining based on density clustering and MapReduce. *Journal of Computing in Civil Engineering*, *31*(5), Article 04017021. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000674

Arce Munoz, S. (2020). *Optimized 3D reconstruction for infrastructure inspection with automated structure from motion and machine learning methods* [Master's thesis]. Brigham Young University.

Balakrishnan Selvakumaran, S., & Hall, D. M. (2022). From crowd to cloud: simplified automatic reconstruction of digital building assets for facility management. *Journal of Facilities Management*, *20*(3), 401–436. https://doi.org/10.1108/JFM-02-2021-0017

Bassier, M., Vergauwen, M., & Poux, F. (2020). Point cloud vs. mesh features for building interior classification. *Remote Sensing*, *12*(14), Article 2224. https://doi.org/10.3390/rs12142224

Bello, S. A., Yu, S., Wang, C., Adam, J. M., & Li, J. (2020). Review: Deep learning on 3D point clouds. *Remote Sensing*, *12*(11), Article 1729. https://doi.org/10.3390/rs12111729

Boissonnat, J. D., & Cazals, F. (2001). Coarse-to-fine surface simplification with geometric guarantees. *Computer Graphics Forum*, *20*(3), 490–499. https://doi.org/10.1111/1467-8659.00542

Cai, G., Zhu, Y., Wu, Y., Jiang, X., Ye, J., & Yang, D. (2023). A multimodal transformer to fuse images and metadata for skin disease classification. *Visual Computer*, *39*(7), 2781–2793. https://doi.org/10.1007/s00371-022-02492-4

Cao, D., Wang, C., Du, M., & Xi, X. (2024a). A multiscale filtering method for airborne LiDAR data using modified 3D alpha shape. *Remote Sensing*, *16*(8), Article 1443. https://doi.org/10.3390/rs16081443

Cao, H., Chen, D., Zhang, Y., Zhou, H., Wen, D., & Cao, C. (2024b). MFINet: a multi-scale feature interaction network for point cloud registration. *The Visual Computer*, *41*(6), 4067–4079. https://doi.org/10.1007/s00371-024-03646-2

Chen, J., Fang, Y., Cho, Y. K., & Kim, C. (2017). Principal axes descriptor for automated construction-equipment classification from point clouds. *Journal of Computing in Civil Engineering*, *31*(2), Article 04016058. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000628

Chen, M., Feng, A., McAlinden, R., & Soibelman, L. (2020). Photogrammetric point cloud segmentation and object information extraction for creating virtual environments and simulations. *Journal of Management in Engineering*, *36*(2), Article 04019046. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000737

Cura, R., Perret, J., & Paparoditis, N. (2017). A scalable and multi-purpose point cloud server (PCS) for easier and faster point cloud data management and processing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *127*, 39–56. https://doi.org/10.1016/j.isprsjprs.2016.06.012

Czerniawski, T., Sankaran, B., Nahangi, M., Haas, C., & Leite, F. (2018). 6D DBSCAN-based segmentation of building point clouds for planar object classification. *Automation in Construction*, *88*, 44–58. https://doi.org/10.1016/j.autcon.2017.12.029

de la Plata, A. R. M., Franco, P. A. C., Franco, J. C., & Bravo, V. G. (2021). Protocol development for point clouds, triangulated meshes and parametric model acquisition and integration in an hbim workflow for change control and management in a unesco's world heritage site. *Sensors*, *21*(4), Article 1083. https://doi.org/10.3390/s21041083

Dekker, B., Ton, B., Meijer, J., Bouali, N., Linssen, J., & Ahmed, F. (2023). Point Cloud analysis of railway infrastructure: A systematic literature review. *IEEE Access*, *11*, 134355–134373. https://doi.org/10.1109/ACCESS.2023.3337049

Dey, T. K., & Goswami, S. (2003). Tight cocone: A water-tight surface reconstructor. *Journal of Computing and Information Science in Engineering*, *3*(4), 302–307. https://doi.org/10.1115/1.1633278

Dey, T. K., Giesen, J., & Goswami, S. (2003). Shape segmentation and matching with flow discretization. In F. Dehne, J. R. Sack, & M. Smid (Eds.), *Lecture notes in computer science*: *Vol. 2748. Algorithms and data structures*. *WADS 2003* (pp. 25–36). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-45078-8_3

Fathi, H., & Brilakis, I. (2011). Automated sparse 3D point cloud generation of infrastructure using its distinctive visual features. *Advanced Engineering Informatics*, *25*(4), 760–770. https://doi.org/10.1016/j.aei.2011.06.001

Franco-Duran, D. M., & Mejia, A. G. (2016). Assessing the cost forecasting performance in construction projects through data envelopment analysis. In *Proceedings of Construction Research Congress 2016* (pp. 2039–2049). ASCE. https://doi.org/10.1061/9780784479827.203

Gao, M., Cao, T.-T., Tan, T., & Huang, Z. (2011). gHull: a three-dimensional convex hull algorithm for graphics hardware. In *I3D '11*: *Symposium on Interactive 3D Graphics and Games* (pp. 204), California, San Francisco. https://doi.org/10.1145/1944745.1944784

Hajihosseinlou, M., Maghsoudi, A., & Ghezelbash, R. (2024). Intelligent mapping of geochemical anomalies: Adaptation of DBSCAN and mean-shift clustering approaches. *Journal of Geochemical Exploration*, *258*, Article 107393. https://doi.org/10.1016/j.gexplo.2024.107393

Hidaka, J., Elfeki, H., Duelund-Jakobsen, J., Laurberg, S., & Lundby, L. (2019). Functional outcome after laparoscopic posterior sutured rectopexy versus ventral mesh rectopexy for rectal prolapse: six-year follow-up of a double-blind, randomized single-center study. *eClinicalMedicine*, *16*(12), 18–22. https://doi.org/10.1016/j.eclinm.2019.08.014

Hinks, T., Carr, H., Truong-Hong, L., & Laefer, D. F. (2013). Point cloud data conversion into solid models via point-based voxelization. *Journal of Surveying Engineering*, *139*(2), 72–83. https://doi.org/10.1061/(ASCE)SU.1943-5428.0000097

Hou, G., Wang, J., & Fan, Y. (2024). Wind power forecasting method of large-scale wind turbine clusters based on DBSCAN clustering and an enhanced hunter-prey optimization algorithm. *Energy Conversion and Management*, *307*, Article 118341. https://doi.org/10.1016/j.enconman.2024.118341

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2022). Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 8338–8354. https://doi.org/10.1109/TPAMI.2021.3083288

Huang, Z., Zheng, H., Li, C., & Che, C. (2024). Application of machine learning-based k-means clustering for financial fraud detection. *Academic Journal of Science and Technology*, *10*(1), 33–39. https://doi.org/10.54097/74414c90

Jing, H., Meng, X., Slatcher, N., & Hunter, G. (2021). Efficient point cloud corrections for mobile monitoring applications using road/rail-side infrastructure. *Survey Review*, *53*(378), 235–251. https://doi.org/10.1080/00396265.2020.1719753

Justo, A., Soilán, M., Sánchez-Rodríguez, A., & Riveiro, B. (2021). Scan-to-BIM for the infrastructure domain: Generation of IFC-complaint models of road infrastructure assets and semantics using 3D point cloud data. *Automation in Construction*, *127*, Article 103703. https://doi.org/10.1016/j.autcon.2021.103703

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. In *Eurographics Symposium on Geometry Processing*, Cagliari, Sardinia, Italy. https://hhoppe.com/poissonrecon.pdf

Li, Z., Cheng, C., Kwan, M. P., Tong, X., & Tian, S. (2019). Identifying asphalt pavement distress using UAV LiDAR point cloud data and random forest classification. *ISPRS International Journal of Geo-Information*, *8*(1), Article 39. https://doi.org/10.3390/ijgi8010039

Li, X., Zhang, Q., Kang, D., Cheng, W., Gao, Y., Zhang, J., Liang, Z., Liao, J., Cao, Y.-P., & Shan, Y. (2024). *Advances in 3D generation*: *A survey*. ArXiv. https://doi.org/10.48550/arXiv.2401.17807

Liu, J., Xiao, Z., Lu, S., Che, D., Dong, M., & Bai, C. (2023). Infrastructure-level support for GPU-enabled deep learning in DATAVIEW. *Future Generation Computer Systems*, *141*, 723–737. https://doi.org/10.1016/j.future.2022.12.014

Mansour, M., Martens, J., & Blankenbach, J. (2024). Hierarchical SVM for semantic segmentation of 3D point clouds for infrastructure scenes. *Infrastructures*, *9*(5), Article 83. https://doi.org/10.3390/infrastructures9050083

Mirzaei, K., Arashpour, M., Asadi, E., Masoumi, H., Bai, Y., & Behnood, A. (2022). 3D point cloud data processing with machine learning for construction and infrastructure applications: A comprehensive review. *Advanced Engineering Informatics*, *51*, Article 101501. https://doi.org/10.1016/j.aei.2021.101501

Moenning, C., & Dodgson, N. a. (2004). Intrinsic point cloud simplification. In *Proceedings of International Conference Graphicon 2004* (pp. 6–10), Moscow, Russia.

Pauly, M., Gross, M., & Kobbelt, L. P. (2002). Efficient simplification of point-sampled surfaces. In *Proceedings of the IEEE Visualization Conference* (*VIS 2002*) (pp. 163–170), Boston, MA, USA. IEEE. https://doi.org/10.1109/VISUAL.2002.1183771

Qiu, S., Zaheer, Q., Ehsan, H., Shah, S. M. A. H., Ai, C., Wang, J., & Zheng, A. A. (2024). Multimodal fusion network for crack segmentation with modified U-Net and transfer learning – based MobileNetV2. *Journal of Infrastructure Systems*, *30*(4), Article 04024029. https://doi.org/10.1061/JITSE4.ISENG-2499

Raj, C., Khular, L., & Raj, G. (2020). Clustering based incident handling for anomaly detection in cloud infrastructures. In *Proceedings of the 2020 10th International Conference on Cloud Computing*, *Data Science* & *Engineering* (*Confluence*) (pp. 611–616), Noida, India. IEEE. https://doi.org/10.1109/Confluence47617.2020.9058314

Rashidi, A., Brilakis, I., & Vela, P. A. (2013). Built infrastructure point cloud data cleaning: an overview of gap filling algorithms. In *Proceedings of the 13th International Conference on Construction Applications of Virtual Reality* (pp. 585–593), London, UK.

Rausch, C., & Haas, C. (2021). Automated shape and pose updating of building information model elements from 3D point clouds. *Automation in Construction*, *124*, Article 103561. https://doi.org/10.1016/j.autcon.2021.103561

Richter, R., & Döllner, J. (2014). Concepts and techniques for integration, analysis and visualization of massive 3D point clouds. *Computers*, *Environment and Urban Systems*, *45*, 114–124. https://doi.org/10.1016/j.compenvurbsys.2013.07.004

Sakr, S., Liu, A., Batista, D. M., & Alomari, M. (2011). A survey of large scale data management approaches in cloud environments. *IEEE Communications Surveys and Tutorials*, *13*(3), 311–336. https://doi.org/10.1109/SURV.2011.032211.00087

Saovana, N., Yabuki, N., & Fukuda, T. (2021). Automated point cloud classification using an image-based instance segmentation for structure from motion. *Automation in Construction*, *12*, Article 103804. https://doi.org/10.1016/j.autcon.2021.103804

Soilán, M., Justo, A., Sánchez-Rodríguez, A., Lamas, D., & Riveiro, B. (2021). 3D point cloud data processing and infrastructure information models: Methods and findings from safeway project. *International Archives of the Photogrammetry*, *Remote Sensing and Spatial Information Sciences*, *43*(B2-2021), 239–246. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-239-2021

Taubin, G. (1995). Curve and surface smoothing without shrinkage. In *Proceedings of IEEE International Conference on Computer Vision* (pp. 852–857), Cambridge, MA, USA. IEEE. https://doi.org/10.1109/ICCV.1995.466848

Vassilev, H., Laska, M., & Blankenbach, J. (2024). Uncertainty-aware point cloud segmentation for infrastructure projects using Bayesian deep learning. *Automation in Construction*, *164*, Article 105419. https://doi.org/10.1016/j.autcon.2024.105419

Vick, S., & Brilakis, I. (2018). Road design layer detection in point cloud data for construction progress monitoring. *Journal of Computing in Civil Engineering*, *32*(5), Article 0401802. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000772

Walsh, S. B., Borello, D. J., Guldur, B., & Hajjar, J. F. (2013). Data processing of point clouds for object detection for structural engineering applications. *Computer-Aided Civil and Infrastructure Engineering*, *28*(7), 495–508. https://doi.org/10.1111/mice.12016

Wang, J., Ahmed, S. M., Shah, H., Atta, Z., & Wang, W. (2025a). An efficient PointNet-based multifaceted autoencoder (PMAE) for denoising rail track fastener point clouds. *Journal of Civil Structural Health Monitoring*, *15*, 3995–4015. https://doi.org/10.1007/s13349-025-01026-5

Wang, J., Ahmed, S. M., Shah, H., Ehsan, H., Faizan, S., Shah, H., Ai, C., Kuang, J., Wang, W., & Qiu, S. (2025b). Self - supervised contrastive anomaly detection in railway fasteners using point clouds and deep metric learning for imbalance dataset. *Journal of Civil Structural Health Monitoring*, *15*, 2861–2886. https://doi.org/10.1007/s13349-025-00960-8

Wang, W., Ehsan, H., Qiu, S., Rahman, T. U., Wang, J., & Zaheer, Q. (2026). Evolution and emerging frontiers in point cloud technology. *Electronics*, *15*(2), Article 341. https://doi.org/10.3390/electronics15020341

Weerasinghe, U. (2023). LIDAR-based 3D object detection for autonomous driving a comprehensive exploration of methods, implementation steps, tools, and challenges in integrating deep learning techniques. *International Journal of Sustainable Infrastructure for Cities and Societies*, *8*(2), 52–64.

Xu, J., Xu, K., Wang, Y., Shen, Q., & Li, R. (2024). *A K-means algorithm for financial market risk forecasting*. ArXiv. https://doi.org/10.48550/arXiv.2405.13076

Yuan, X., Chen, H., & Liu, B. (2021). Point cloud clustering and outlier detection based on spatial neighbor connected region labeling. *Measurement and Control*, *54*(5–6), 835–844. https://doi.org/10.1177/0020294020919869

Zaheer, Q., Ahmed, S. M., Shah, H., Wang, W., Ehsan, H., Ai, C., Wang, J., & Qiu, S. (2026a). Multimodal graph neural network framework for railway fastener tightness assessment from high-resolution point clouds. *Engineering Applications of Artificial Intelligence*, *167*(P2), Article 113829. https://doi.org/10.1016/j.engappai.2026.113829

Zaheer, Q., Ehsan, H., Wang, W., Malik, M., & Wang, J. (2026b). Real-time synthetic data generation and segmentation of railway fasteners using an attention guided dual-phase diffusion-aided model. *Measurement*, *261*, Article 119874. https://doi.org/10.1016/j.measurement.2025.119874

Zaheer, Q., Ehsan, H., Wang, W., Shah, S. F. H., Ai, C., Wang, J., & Qiu, S. (2026c). DA-NGF: A domain-adaptive neurogenerative framework for few-shot railway fastener defect synthesis and transferable representation learning. *Applied Intelligence*, *56*, Article 32. https://doi.org/10.1007/s10489-025-07054-4

Zhang, G., Vela, P. A., Karasev, P., & Brilakis, I. (2015). A sparsity-inducing optimization-based algorithm for planar patches extraction from noisy point-cloud data. *Computer-Aided Civil and Infrastructure Engineering*, *30*(2), 85–102. https://doi.org/10.1111/mice.12063

Zhang, J., Zhao, X., Chen, Z., & Lu, Z. (2019). A review of deep learning-based semantic segmentation for point cloud. *IEEE Access*, *7*, 179118–179133. https://doi.org/10.1109/ACCESS.2019.2958671

Zhuang, H., Zhang, J., & Liao, F. (2023). A systematic review on application of deep learning in digestive system image processing. *Visual Computer*, *39*(6), 2207–2222. https://doi.org/10.1007/s00371-021-02322-z