

## Article

# Towards 3D Reconstruction of Multi-Shaped Tunnels Utilizing Mobile Laser Scanning Data

Xuan Ding <sup>1,2</sup>, Shen Chen <sup>1,\*</sup> , Mu Duan <sup>2</sup> , Jinchang Shan <sup>3</sup>, Chao Liu <sup>4</sup> and Chuli Hu <sup>1,2</sup>

- <sup>1</sup> National Engineering Research Center for Geographic Information System, School of Geography and Information Engineering, Engineering Research Center of Natural Resource Information Management and Digital Twin Engineering Software, Ministry of Education, China University of Geosciences (Wuhan), Wuhan 430074, China; dingxuan@cug.edu.cn (X.D.); huchl@cug.edu.cn (C.H.)
- <sup>2</sup> Wuhan CUG Smart City Research Institute Co., Ltd., Wuhan 430080, China; duanmu@cug.edu.cn
- <sup>3</sup> Wuhan Hanyang Municipal Construction Group Co., Ltd., Wuhan 430050, China; shanjinchang@focus-in.cn
- <sup>4</sup> The Institute of Geological Survey, China University of Geosciences, Wuhan 430074, China; Liuchao2482@163.com
- \* Correspondence: chenshen@cug.edu.cn

**Abstract:** Using digital twin models of tunnels has become critical to their efficient maintenance and management. A high-precision 3D tunnel model is the prerequisite for a successful digital twin model of tunnel applications. However, constructing high-precision 3D tunnel models with high-quality textures and structural integrity based on mobile laser scanning data remains a challenge, particularly for tunnels of different shapes. This study addresses this problem by developing a novel method for the 3D reconstruction of multi-shaped tunnels based on mobile laser scanning data. This method does not require any predefined mathematical models or projection parameters to convert point clouds into 2D intensity images that conform to the geometric features of tunnel linings. This method also improves the accuracy of 3D tunnel mesh models by applying an adaptive threshold approach that reduces the number of pseudo-surfaces generated during the Poisson surface reconstruction of tunnels. This method was experimentally verified by conducting 3D reconstruction tasks involving tunnel point clouds of four different shapes. The superiority of this method was further confirmed through qualitative and quantitative comparisons with related approaches. By automatically and efficiently constructing a high-precision 3D tunnel model, the proposed method offers an important model foundation for digital twin engineering and a valuable reference for future tunnel model construction projects.

**Keywords:** mobile laser scanning; point cloud; intensity image; tunnel 3D reconstruction



**Citation:** Ding, X.; Chen, S.; Duan, M.; Shan, J.; Liu, C.; Hu, C. Towards 3D Reconstruction of Multi-Shaped Tunnels Utilizing Mobile Laser Scanning Data. *Remote Sens.* **2024**, *16*, 4329. <https://doi.org/10.3390/rs16224329>

Academic Editor: Giuseppe Casula

Received: 8 October 2024

Revised: 13 November 2024

Accepted: 15 November 2024

Published: 20 November 2024

**Correction Statement:** This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

By offering digital representations of tunnels, digital twin models are particularly useful in the long-term maintenance and operations of tunnels. The basis of digital twin modeling is accurately building a high-precision 3D tunnel model based on the physical world [1]. The construction of 3D models is largely supported by mobile 3D laser scanning, which is widely employed in obtaining not only the detailed features of tunnel lining surfaces but also high-precision spatial coordinate 3D point clouds [2,3]. However, using mobile laser scanning data to build 3D tunnel models with structural integrity and high-quality textures remains a challenge, particularly for tunnels of different shapes.

Many scholars have thus investigated the use of these data in constructing 3D tunnel models that represent high-quality and high-accuracy information about the interior texture and 3D geometry of tunnels [4]. These data are traditionally collected via camera imaging, but poor lighting and spatial constraints inside tunnels negatively affect the quality and ability of the resulting 2D images to capture the complexity in the internal structure of tunnels [5–7]. To address this problem, scholars have investigated the application of

algorithms in converting point clouds into 2D images [8,9]. For instance, Xu et al. [10] generated a 2D image by unfolding the point cloud along the top line parallel to the central axis of a tunnel, while Qiu et al. [11] unfolded point clouds by using a standard design cross-section as reference. However, these two methods have their respective setbacks. Specifically, the image obtained by the former approach may demonstrate some length deformation and overlapping of cracks at the top of the tunnel, while the image obtained by the latter approach may generate a distorted view of the interior surface of the tunnel. Sun et al. [12] generated high-quality 2D images via cylindrical projection based on point cloud data. Zhang et al. [13] and Chen et al. [14] further improved cylindrical projection in order to improve image accuracy and reduce computational complexity. However, tunnels in practical engineering environments often have horseshoe, arch, and rectangular shapes, and the aforementioned methods specifically focus on tunnels with circular or elliptical shapes and need to have their projection parameters preset based on these shapes [15,16]. Liu et al. [17] addressed this limitation by developing a tunnel surface parameterization algorithm based on harmonic mapping that uses the mapping of 3D models to generate 2D images, while Deng et al. [18] obtained a 2D seamless panoramic image of tunnel linings by unwrapping the 3D tunnel lining mesh into straight 2D meshes. While these two methods can effectively preserve the lining information of tunnels regardless of shape, they show certain disadvantages, such as their requirements for complex workflows and high-quality 3D mesh models. Therefore, a method that can convert point clouds of various tunnel shapes into high-quality and high-accuracy 2D images should be proposed.

Although image processing algorithms based on 2D images can efficiently detect tunnel defects, they do not provide information about the spatial location of such defects [19–21]. Therefore, 3D models need to be constructed to achieve tunnel digitization [22]. Photogrammetry methods require significantly cheaper and probably smaller equipment while providing image data. A 3D model of the tunnel surface can be processed automatically using a single camera to take images of the surface from multiple angles. However, the reconstruction process requires some degree of overlap among these images [23–25]. Xue et al. [26] used SfM technology to achieve image overlaps and construct 3D models. However, the image texture used for localization is prone to dust, dirt, and water leakage buildup, hence leading to algorithm failure. Huang et al. [27], Song et al. [28], and Zhao et al. [29] aimed for a rapid 3D tunnel reconstruction by generating triangular meshes from point clouds. However, this method is inapplicable to large-scale point cloud reconstruction and is highly sensitive to point cloud noise.

As a classical implicit-function-based point cloud reconstruction method, Poisson reconstruction addresses the potential hole and ambiguity issues and demonstrates strong tolerance to incomplete point clouds and noise. Given these advantages, this method has been widely used in 3D tunnel modeling. For instance, Ao et al. [30] and Qi et al. [31] built high-precision, smooth 3D tunnel models using Poisson reconstruction to achieve efficient defect monitoring. Guo et al. [32], Jiang et al. [33], and Xiang et al. [34] tested the feasibility of using Poisson reconstruction in 3D tunnel modeling.

However, the Poisson reconstruction algorithm has several limitations, such as generating pseudo-surfaces on the resulting 3D models, affecting their accuracy. Liu et al. [35] proposed an improved Poisson surface reconstruction algorithm based on boundary constraints, which further enhances the accuracy and reconstruction efficiency of the model compared to the traditional Poisson reconstruction algorithms. However, in reconstructing non-enclosed tunnels, misconnecting those regions that do not align with the morphological features at the edges can easily generate pseudo-surfaces.

In sum, most of the available methods for obtaining 2D intensity images from 3D point clouds are only applicable to shielded circular tunnels and require the projection parameters and mathematical models to be predefined according to the shape of these tunnels. In addition, pseudo-surfaces can easily influence the accuracy of Poisson reconstruction models for non-enclosed tunnels. To address these limitations, this study introduces a novel method that uses mobile laser scanning data for the 3D reconstruction of multi-shaped

tunnels. This method offers high-quality textures for digital 3D models by converting point clouds into 2D images that match the geometric features of tunnel linings. The pseudo-surfaces in the Poisson reconstruction model are also divided by this method through a dynamic adjustment of the threshold. Aside from enabling the rapid construction of high-precision digital 3D tunnel models with various shapes, the proposed method also eliminates the need for specific projection parameters and mathematical models, thus offering a valuable methodological reference for digital tunnel modeling projects. With its excellent flexibility and robustness, the proposed method also provides a solid model foundation for the automatic construction of digital tunnels in the future.

The rest of this paper is structured as follows. Section 2 discusses the processes and principles involved in generating 2D images and in the 3D modeling of multi-shaped tunnels. Section 3 describes the implementation of the prototype system along with the experimental results. Section 4 compares the proposed method with other reconstruction methods presented in the literature. Conclusions are given in Section 5.

## 2. Methods

Figure 1 shows the workflow of the proposed method, which has two key components and provides inspectors with an intuitive overall 3D view of the information of the tunnel. First, when generating images for multi-shaped tunnel point clouds, the proposed method converts the point cloud data of multi-shaped tunnels into 2D images based on a novel projection strategy and then rectifies these images based on the tunnel lining's geometric features. Second, when modeling the 3D tunnel point cloud, the proposed method reduces the number of pseudo-surfaces in the Poisson reconstruction model by using an adaptive threshold approach, thus facilitating the development of a high-precision 3D tunnel model.

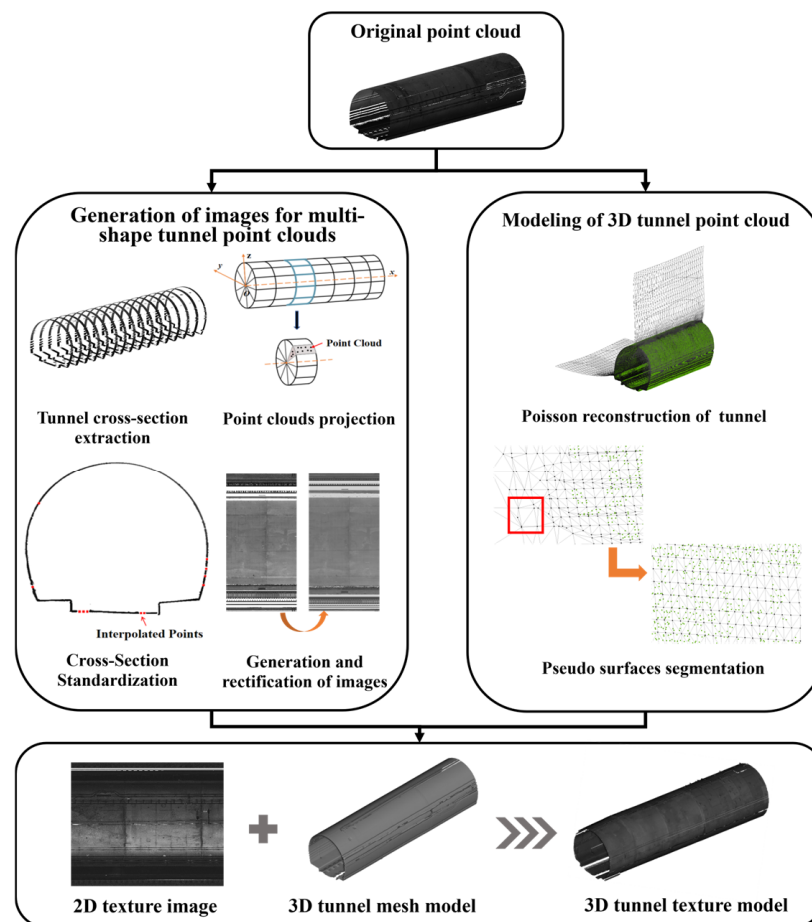


Figure 1. Workflow of the proposed method.

### 2.1. Generation of Images for Multi-Shaped Tunnel Point Clouds

An important step in the 2D image reconstruction of tunnel interiors is projecting point clouds onto a developable surface or plane. These point clouds reflect intensity or color information and contain 3D coordinates. However, tunnels of varying shapes can significantly amplify the projection distortions inherent in this conversion process, especially when a uniform projection scheme is applied. Although customized projection strategies can help reduce such distortions, the parameters involved in the process are too complex and significantly reduce the processing efficiency. In consideration of these issues, this study adopts an innovative “unfold-first, rectify-after” strategy, where the tunnel cross-sections are initially extracted, and then the minimum bounding cylinder parameters are calculated. The point cloud is then projected onto a cylindrical surface, and the cross-sectional point sets are normalized afterward. The projected cylinder is eventually unrolled along the track centerline, and the intensity data from the point cloud are used to obtain a 2D intensity image from the 2D point cloud. This 2D intensity image is subsequently rectified while taking the geometric features of the tunnel lining into account.

#### 2.1.1. Extraction of Tunnel Cross-Sections

To begin extracting the tunnel cross-section, the tunnel should be initially converted from the Cartesian to the tunnel stationing coordinate system, as shown in Figure 2, where the  $x'$ -axis denotes the tunnel stations. Extracting the tunnel cross-section requires fitting a mathematical shape to several scattered points and then obtaining the fitted points. Some of the most commonly used fitting methods include the least squares fitting method, which is particularly effective in scenarios with high data quality and small errors. The RANSAC algorithm is also widely used, being ideal for scenarios with significant noise or outliers. However, both of these methods require the fitting model specifications to be predefined, hence reducing their application value in extracting tunnel cross-sections from point clouds of different shapes.

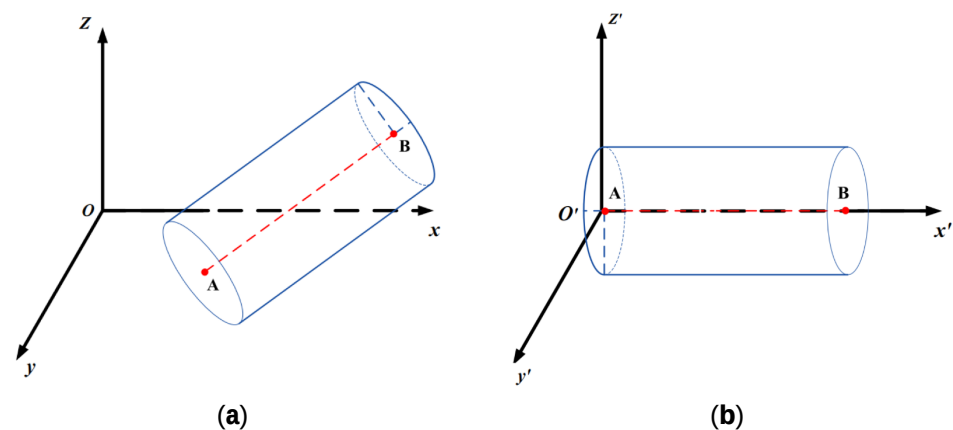


**Figure 2.** Location of the tunnel point cloud in the 3D coordinate system: (a) The Cartesian coordinate system; (b) The tunnel stationing coordinate system.

Therefore, the tunnel boundary point cloud in this study is extracted using the Alpha Shape algorithm, which uses two circles with a common chord to determine whether a scatter point is a boundary point or not. Due to this simple process, this algorithm can efficiently extract the boundaries of projected point clouds.

Considering the complexity of tunnel environments and the diversity of equipment inside tunnels, errors may occur when acquiring laser scanner data and segmenting the tunnel into sections. Therefore, the mileage values of the tunnel ring data within a certain mileage interval around the current mileage  $x_0$  are unified, as shown in Figure 3, and then the tunnel cross-section is extracted from the 2D points on the  $yoz$  plane along this tunnel segment by using the Alpha Shape algorithm.

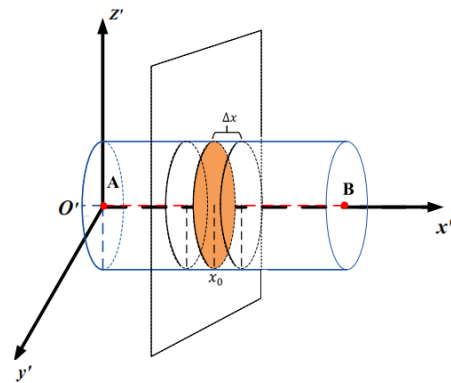


**Figure 3.** Intuitive illustration of extracting tunnel ring sections and unifying the mileage.

### 2.1.2. Projection of Tunnel Point Clouds

During the tunnel cross-section extraction process, the outer contour of the tunnel wall is reflected by the cross-sectional point set. Due to the approximately cylindrical distribution of the tunnel point cloud in a 3D space, a projection cylinder can be constructed for coordinate mapping. Therefore, this study only searches for a minimum radius circle that contains all 2D cross-sectional points instead of calculating the base circle of the tunnel projection cylinder. The search process proceeds as follows (Figure 4):

1. The center of the circle is set to the coordinates of the first point  $P_1$  in the initial cross-sectional point set, and then the radius is set to a minimal value. This circle is denoted by  $C_1$ .
2. The second point  $P_2$  is selected. If circle  $C_1$  does not contain  $P_2$ , then this circle is updated by setting the line segment between points  $P_1$  and  $P_2$  as its diameter. The updated circle is denoted by  $C_2$ .
3. The third point  $P_3$  is selected. If this point is located inside circle  $C_2$ , then this circle does not need to be updated. Otherwise,  $P_1$ ,  $P_2$ , and  $P_3$  are used to build the new circle  $C_3$  to ensure that this circle includes the new point.
4. All points are sequentially traversed. Circle  $C_j$  is not updated if it contains the remaining point  $P_i$ . Otherwise, steps 2 and 3 are repeated to recalculate the updated circle.
5. After traversing all points, the coordinates of the three points and the equation of the circle can be used to obtain the center and radius of the minimum bounding circle for the cross-sectional point set.

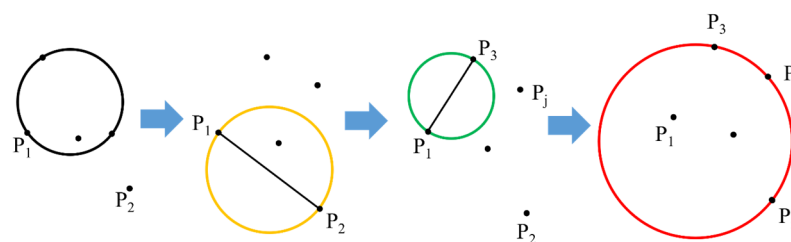


**Figure 4.** Diagram of calculating the minimum bounding circle.

The center and radius parameters of the minimum enclosing circle for the cross-sectional point set are determined. Then, the mileage length of the tunnel segment is set as the projected cylinder height to ensure that the denoised point cloud is fully enclosed in a unique cylinder. The coordinate mapping of the tunnel point cloud based on the projection cylinder is presented in Figure 5, where the minimum bounding circle and the tunnel cross-section are denoted by the blue dashed line and the gray curve, respectively.

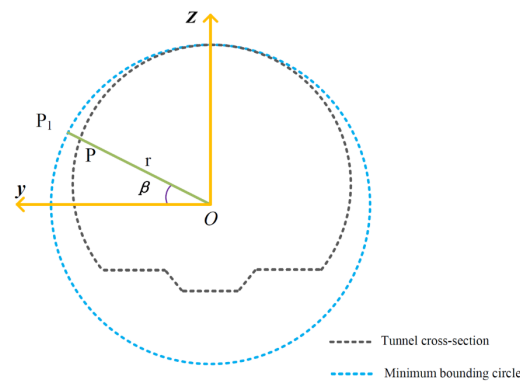


**Figure 5.** Diagram of Coordinate Mapping.

The clockwise angle  $\beta_i$  between any point  $P(y_i, z_i)$  in the point cloud and the  $y$ -axis is obtained as follows:

$$\beta_i = \begin{cases} \arctan\left(\frac{z_i - z_0}{y_i - y_0}\right) + \frac{\pi}{2}, & y_i > 0 \\ \arctan\left(\frac{z_i - z_0}{y_i - y_0}\right) + \frac{3\pi}{2}, & y_i \leq 0 \end{cases} \quad (1)$$

The mapped coordinates  $(\beta_i, x_i)$  are then obtained by associating the point with its corresponding tunnel mileage. The original 3D coordinates are then successfully converted into their 2D versions.

### 2.1.3. Standardization of Tunnel Cross-Section Point Sets

The extracted initial cross-section has a dense set of points given the large number of points used for its extraction. If the original image is cropped at different angles using the coordinates of these points after mapping, then the pixel count in sub-images may be too low. This can result in stretching or compression distortion in the subsequent processing. To achieve an effective interpolation in these sub-images, the segmentation threshold is set to 1 degree, at which point each standard cross-section set has 360 points. The tunnel cross-section is standardized as follows:

1. After coordinate mapping, set the point cloud according to the clockwise angle between each point and the  $y$ -axis.
2. Repeat the process at each angular interval to check for the presence of points. In case multiple points are present, the point whose angle is closest to the midpoint of the given interval is selected. Otherwise, this interval is represented by a linearly interpolated point.
3. If the initial cross-section contains abnormalities, such as depressions, protrusions, or missing sections, then apply linear interpolation to optimize these abnormal cross-sectional points.

The above standardization process is illustrated in Figure 6, and the case of missing initial cross-sectional points is illustrated in Figure 6a. Two adjacent points in the initial cross-section are said to have missing points between them when the straight-line distance  $d$  separating them exceeds a certain threshold. A depression in the tunnel cross-section can be seen in Figure 6b. An abnormal concavity is detected in the presence of significant differences among the slopes of those straight lines that connect adjacent points (represented by  $\beta$ ,  $\alpha$ , and  $\gamma$ ). A protrusion in the tunnel cross-section can be observed in Figure 6c. The vertical distance from the point to the horizontal line through the projection center is determined, and an abnormal protrusion is detected when several adjacent points show significant variations in their vertical distance  $d$  and when the vertical or horizontal coordinates show the corresponding changes. The use of linear interpolation to normalize the tunnel cross-sections is illustrated in Figure 6d. The angle  $\alpha$  between the  $y$ -axis and the line that connects the adjacent points  $A(y_1, z_1)$  and  $B(y_2, z_2)$  is calculated, the length of  $AB$  is denoted by  $L_{AB}$ ,  $N$  denotes the number of additional points required, and  $\Delta L = L_{AB}/N$

represents the length of each segment in the evenly divided intervals. The interpolation points are then obtained as follows:

$$\begin{cases} y_i = y_1 + \Delta L * i * \cos\alpha \\ z_i = z_1 + \Delta L * i * \sin\alpha \end{cases} \quad (2)$$

where  $y_i$  and  $z_i$  denote the horizontal and vertical coordinates of the interpolated point, respectively.

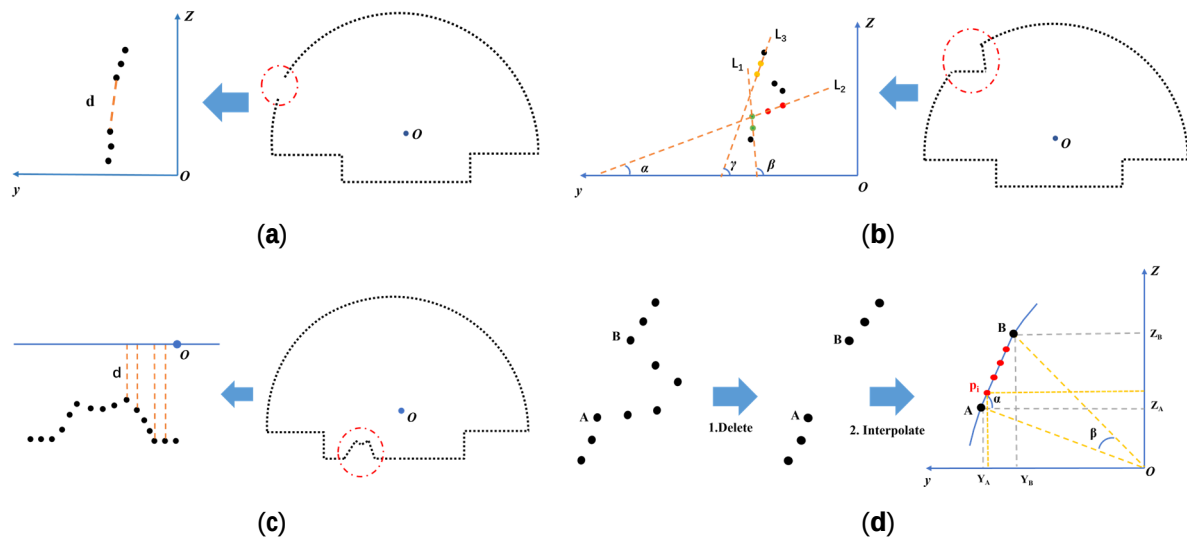


**Figure 6.** Intuitive illustration of the standardization of tunnel cross-sectional point sets: (a) Missing points in the cross-section; (b) Abnormal depression in the cross-section; (c) Protrusion in the cross-section; (d) Linear interpolation for standardizing the tunnel cross-section.

#### 2.1.4. Generation and Rectification of Images

The point cloud on the cylindrical surface is unfolded along the track centerline, and the point cloud intensity data are used to convert the 2D point cloud into a 2D intensity image, as shown in Figure 7.

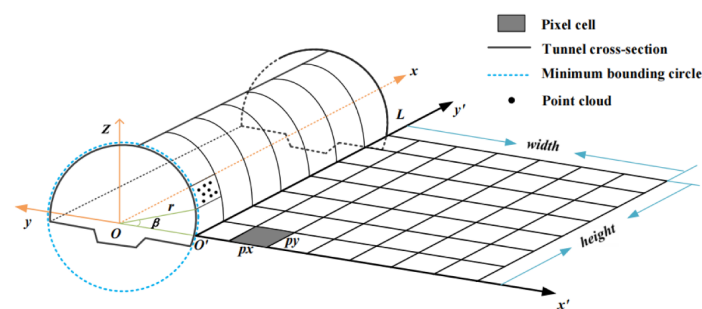


**Figure 7.** Diagram of cylindrical projection unwrapping.

In Figure 7, the tunnel stationing and 2D image coordinate systems are represented by  $o - xyz$  and the  $x'o'y'$  plane, respectively, while the grid resolution is represented by parameters  $px$  and  $py$ , which indicate the angular range corresponding to each pixel. Given the tunnel length  $L$  and the minimum bounding circle radius  $r$ , the image dimensions are expressed as:

$$\begin{cases} height = \left\lceil \frac{L}{py * r} \right\rceil \\ width = \left\lceil \frac{2\pi}{px} \right\rceil \end{cases} \quad (3)$$

To locate the point cloud corresponding to each pixel in the image, assume that the coordinates of an arbitrary pixel are  $(x_i, y_i)$ , and the point cloud points within this pixel are  $(\beta_i, x_i)$ . The points within the pixel and the pixel coordinates should satisfy the following range relationships.

To identify the point cloud that corresponds to each pixel in the image, the coordinates of an arbitrary pixel are assumed as  $(x_i, y_i)$ , and the point cloud points within this pixel are assumed as  $(\beta_i, x_i)$ . The following range relationships should be satisfied by the points within the pixel and the pixel coordinates:

$$\{(\beta_i, x_i) | (x_i - 1) * px \leq \beta_i \leq x_i * px, (y_i - 1) * py * r \leq x_i \leq y_i * py * r\} \quad (4)$$

where  $i = 1, 2, 3, \dots, k$ , with  $k$  denoting the number of points in the pixel. The point cloud within each pixel is determined, and then the average reflection intensity of these points, which serves as the grayscale value  $C$  of the pixel, is computed as:

$$C = \frac{\sum_{i=1}^k I_i}{k} \quad (5)$$

where  $I_i$  represents the reflected intensity value of the point cloud. An initial high-precision 2D image containing all detailed information about the tunnel interior is generated after obtaining the reflected intensity value for all pixels in the image.

As shown in Figure 8, the 2D image is divided into 359 sub-images using the angle of each point in the cross-section. Dividing the unfolded cylindrical projection image at equal angular intervals will result in sub-images having the same size, which prevents the 2D image from accurately reflecting the tunnel lining features.

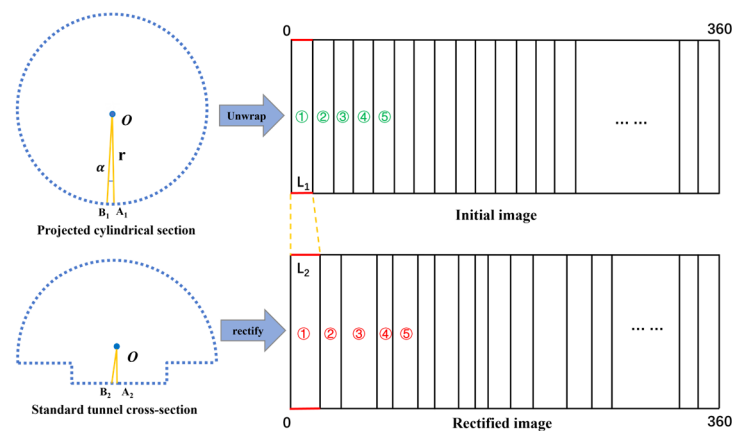


**Figure 8.** Diagram of Image Rectification.

In this case, the initial image should be rectified based on the actual tunnel lining geometric features. The straight-line distance between two points at a specified angular interval on the standard cross-section is initially calculated along with the arc length at the same angular interval on the projected cylindrical cross-section. The sub-images are then stretched or compressed by using the ratio of the two aforementioned distances as the scaling factor.

The position of the pixel on the initial image that corresponds to each point on the projection cylinder cross-section is defined as follows:

$$pos = \left[ \frac{\alpha}{px} \right] \quad (6)$$

where  $px$  represents the image resolution, and  $\alpha$  denotes the central angle corresponding to arc  $A_1B_1$ . The original image can then be divided into several sub-images based on the position  $pos$ .  $L_1$  denotes the arc length at the specified angular interval on the projection

cross-section, while  $L_2$  denotes the straight-line distance between points on the standard cross-section at the same angular interval. If both points on the standard cross-section are  $A_2(y_{A2}, z_{A2})$  and  $B_2(y_{B2}, z_{B2})$ , with  $r$  denoting the radius of the minimum bounding circle, then the geometric rectification ratio  $K$  for the corresponding sub-images can be obtained based on Equation (7).

$$K = \frac{L_1}{L_2} = \frac{\alpha * r}{\sqrt{(y_{A2} - y_{B2})^2 + (z_{A2} - z_{B2})^2}} \quad (7)$$

The sub-images are compressed when  $K$  is less than 1 and stretched otherwise. Nearest-neighbor interpolation, linear interpolation, or cubic spline interpolation can be used to rectify the sub-images. After being rectified according to ratio  $K$ , these sub-images are sequentially stitched together to generate a 2D image that accurately highlights the tunnel lining geometric features.

## 2.2. Modeling of 3D Tunnel Point Cloud

The coordinate information of the point cloud can be used to reconstruct its 3D mesh model. However, the Poisson surface reconstruction algorithm often produces models containing many pseudo-surfaces, which are unrealistic or erroneous parts of the reconstructed model surface that arise due to the limitations of the algorithm or the incompleteness of the input data. These pseudo-surfaces typically manifest as unreasonable connections, additional surfaces, or holes, reducing the accuracy of the model. Analyzing the properties of point cloud data and the characteristics of pseudo-surfaces reveals that an automated adjustment of the segmentation threshold preserves the true structure of the model. This approach simultaneously optimizes the removal of pseudo-surfaces. The 2D grayscale tunnel lining images are also mapped as texture maps onto the 3D mesh model to generate a high-precision 3D tunnel model with high-quality textures.

### 2.2.1. The Poisson Surface Reconstruction Method

Coordinates and normal vectors should be assigned to points within the dataset when using the Poisson equation to construct surface models. However, using 3D laser scanning to obtain point clouds may be unable to return normal vector information for each point. Therefore, the normal vectors should be calculated using suitable methodologies. In consideration of the tunnel condition, the normal vector of each point  $\vec{n} = (x_n, y_n, z_n)$  is formulated as  $x_n = -x$ ,  $y_n = 0$ ,  $z_n = -z$  as shown in Figure 9, and the normal vector field is subsequently calculated. The normal vector is normalized, and a virtual border is constructed to obtain a limited boundary. This approach reduces reconstruction time and ensures that the precision of the reconstructed area can be controlled.

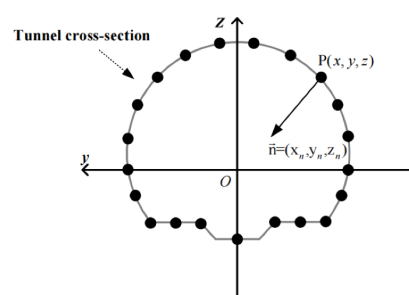


**Figure 9.** Calculate the normal vector of every point.

The directed point set and segmentation indication function are presented in Figure 10a and 10b, respectively. The smoothing filtering indication function is initially applied. Then, smoothing functions are used to calculate the directed gradient fields and block the emergence of infinite values. The relation of the smoothing indication function gradient

and the surface normal vector field calculated by the divergence theory is shown as follows.

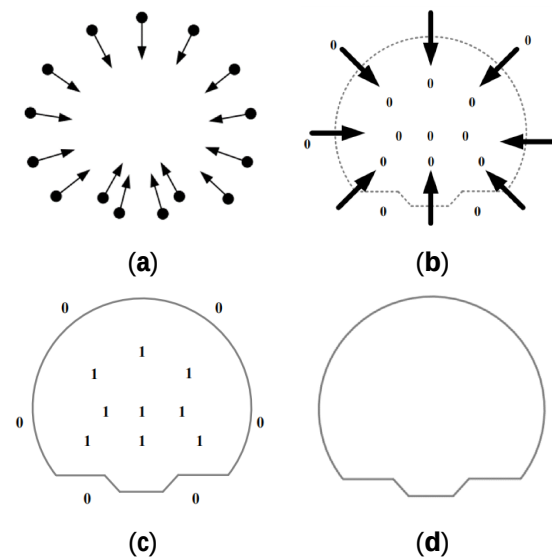


**Figure 10.** Intuitive illustration of Poisson reconstruction tunnel cross-section: (a) Oriented points  $\vec{V}$ ; (b) Indicator gradient  $\nabla\chi_M$ ; (c) Indicator function  $\chi_M$ ; (d) Model entity  $\partial M$ .

The boundary of an object was expressed as  $\partial M$  (Figure 10d), and the indication of  $M$  was expressed as  $\chi_M$  (Figure 10c), the normal vector of every point of the  $\partial M$  was expressed as  $\vec{N}_{\partial M}(p)$ , and the smoothing filtering was expressed as  $F(q)$ . According to the Gaussian divergence theory, the gradient of the smoothing indication function was defined as the vector field generated by smoothing the normal vector of the surface. The calculation could be expressed as Equation (8):

$$\nabla(\chi M * F)(q_0) = \int_{\partial M} F_p(q_0) N_{\partial M}(p) dp \quad (8)$$

The integral of the surface cannot be directly calculated because the surface is unknown. The sample points are then divided into smaller blocks  $P_s (P_s \in \partial M)$ , and the summed integral of the points in each of these blocks is used to replace the integral of the surface.

$$\nabla(\chi M * F)(q) = \sum_{s \in S} \int_{P_s} F_p(q) N_{\partial m}(p) dp \approx \sum_{s \in S} |P_s| F_{s,p}(q) s \cdot \vec{N} = \vec{V}(q) \quad (9)$$

If the vector field  $\vec{V}$  is known, then  $\nabla\chi = \vec{V}$  can be used to calculate the function  $\chi$ . However, given that the vector field  $\vec{V}$  is non-integrable, a precise calculation method cannot be obtained. Therefore, the solution is retrieved by  $\min_{\chi} \|\nabla\chi - \vec{V}\|$  based on the theory of minimum square approximation, where the divergence operator is  $\nabla \cdot (\nabla\chi) = \nabla \cdot \vec{V}$ .

$\Delta\chi = \nabla \cdot \vec{V}$  denotes the Poisson equation during the calculation. Apart from solving this equation, using the marching cube algorithm to extract the iso-surface ultimately results in the generation of the surface model.

### 2.2.2. Adaptive Thresholding for Segmenting Pseudo-Surfaces

The 3D mesh model built using the traditional Poisson reconstruction algorithm is shown in Figure 11, along with its pseudo-surfaces. Due to the partial differential nature of the Poisson equation, the presence of errors will shift the indicator function and produce many pseudo-closed surfaces at the edges of non-watertight surface connections, as can be seen in Figure 11a.

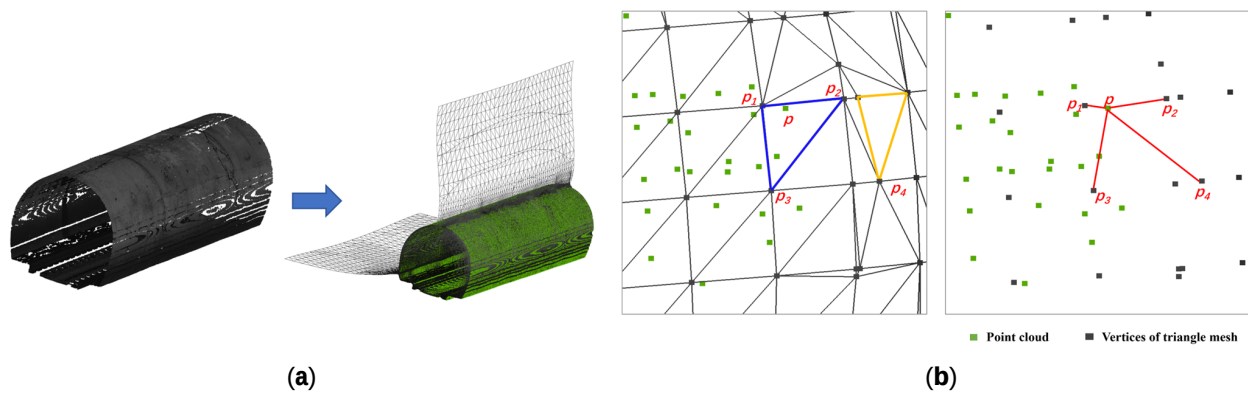


**Figure 11.** The tunnel model constructed by the traditional Poisson reconstruction algorithm: (a) 3D tunnel mesh model in wireframe mode; (b) Pseudo-surfaces and triangular mesh on the edges of the model.

In the tunnel model, the triangular mesh has vertices located around the input point cloud, as can be seen in Figure 11b, where the triangular mesh and pseudo-surfaces are denoted by the blue and yellow triangles, respectively. The distances from input point  $p$  to the blue triangular mesh vertices ( $p_1$ ,  $p_2$  and  $p_3$ ) are significantly shorter than the distance between this input point and vertex  $p_4$  of the yellow pseudo-surface, thereby highlighting a relatively short distance between the input point cloud and the vertices of the model and a relatively large distance between the input point cloud and the pseudo-surface vertices. This distance threshold can be used to divide the pseudo-surfaces. However, this process consumes much time, especially with large datasets, given that it needs to be iterated several times across all input point cloud data and model vertices.

The triangular mesh of the model contains many triangles with small perimeter variations and a small number of pseudo-surfaces with huge perimeter variations. These triangles and pseudo-surfaces are then used to determine the distribution probability of triangular mesh elements with similar perimeters. The vertices of triangular mesh elements with the highest perimeter distribution probabilities are selected as the sampling points. Then, the segmentation threshold for pseudo-surfaces in the entire model is determined by calculating the distance between these sampling points and the input point cloud. Through this approach, the time consumed in selecting the segmentation threshold can be significantly reduced.

Triangles with similar perimeters may also be encountered frequently, given that the triangular mesh of the model largely comprises the entire Poisson model mesh. The set of edges of the triangular mesh in the Poisson model is  $A = a_{1i}, a_{2i}, a_{3i}, i = 1, 2, \dots, n$ , where  $n$  represents the number of triangular meshes in the Poisson model. Meanwhile, each triangle has a perimeter of  $L_i = a_{1i} + a_{2i} + a_{3i}$ , where  $L_i$  can be transformed into an integer before processing by scaling this variable by a factor.  $L_{min}$  and  $L_{max}$ , which denote the minimum and maximum parameters, respectively, should also be transformed into integers. If  $v_j$  denotes the number of triangles with perimeter  $L_j$ , then its distribution probability  $F(v_j)$  can be calculated as follows:

$$F(v_j) = \frac{v_j}{n}, j \in [L_{min}, L_{max}] \quad (10)$$

$F(v_j)$  is sorted in an ascending order, the sampling points are defined as the vertices of those triangles with the highest perimeter distribution probabilities, and the distance from these sampling points to the original point cloud is calculated.  $k$  represents the number of input points,  $P = \{p_1, p_2, \dots, p_k\}$  denotes the input point cloud set,  $(x_p, y_p, z_p)$  and  $(x_q, y_q, z_q)$  denote the input point and sampling point coordinates, and  $Q = \{q_1, q_2, \dots, q_m\}$

represents the sampling point set. The Euclidean distance between the input point cloud and the sampling point vertices is then calculated as:

$$d = \sqrt{(x_{pi} - x_{qj})^2 + (y_{pi} - y_{qj})^2 + (z_{pi} - z_{qj})^2} \quad (11)$$

The triangular mesh vertices are arranged in rows, while the input point cloud is arranged in columns. Afterward, the Euclidean distance between the input points and the triangular mesh vertices is arranged in an ascending order.

The maximum distance between the model triangle vertices and the input point cloud is then determined based on the density of the model vertices, which is expressed as:

$$\rho' = \rho * \frac{k'}{k} \quad (12)$$

where  $\rho$  denotes the input point cloud density,  $k'$  represents the number of vertices in the triangular mesh of the Poisson model, and  $\bar{d}$  denotes the maximum distance between the input point cloud and the triangular model vertices, which is defined as:

$$\bar{d} = \frac{\sum_{j=1}^m d_{[\rho']}}{m} \quad (13)$$

The vertices of the Poisson model and the input point cloud are then traversed, and the corresponding distances are calculated. A point is marked as 0 if such distance exceeds  $\bar{d}$ ; otherwise, this point is marked as 1. The non-closed tunnel 3D model is eventually obtained by removing those triangular mesh elements that are marked as 0.

### 2.2.3. Automatic Texture Mapping

The generated model surfaces still show the best fit for the local points even though the Poisson reconstruction algorithm modifies the positions of the original point cloud. In this case, one can assume that the model vertices are distributed around the projection cylinder and the original point cloud. Those pixel units that correspond to the model vertices are then determined following the point cloud unfolding process, and the texture mapping coordinates are subsequently obtained by normalizing the positional coordinates. These texture mapping coordinates are defined as:

$$\begin{cases} u = \left[ \frac{x_i / px}{width} \right], 0 \leq u \leq 1 \\ v = 1 - \left[ \frac{y_i / py}{height} \right], 0 \leq v \leq 1 \end{cases} \quad (14)$$

where  $px$  and  $py$  denote the image resolution,  $p_i(x_i, y_i)$  represents the coordinates of the model vertices, and  $height$  and  $width$  are calculated using Equation (3).

## 3. System Implementation and Experimental Results

All experiments were carried out on a personal computer with an Intel Core i7-14700HX 5.50GHz CPU and 16GB RAM. The prototype system implements the method proposed in this paper, which can effectively reconstruct three-dimensional tunnel texture models based on point cloud data. To validate the effectiveness of the method, we performed tunnel reconstruction experiments using tunnel point cloud datasets of different shapes.

### 3.1. System Implementation

The 3D Tunnel Management System, developed using Visual Studio 2022 C++ and the Qt platform, facilitates the rapid 3D reconstruction of tunnel point cloud data and enables spatial browsing of the tunnel model and point cloud, thereby playing a crucial role in tunnel maintenance and safety assessment.

As illustrated in Figure 12a, the system is structured into three main layers: the data layer, the business layer, and the presentation layer. The data layer encompasses various data resources, including point cloud, model, and image data, essential for 3D reconstruction. The business layer, serving as the core of the system, includes functional modules for the entire process of 3D reconstruction, such as data loading and browsing, preprocessing, point cloud unrolling, and modeling. The presentation layer provides graphical user interfaces for system-user interactions, enabling users to perform operations defined in the business layer.

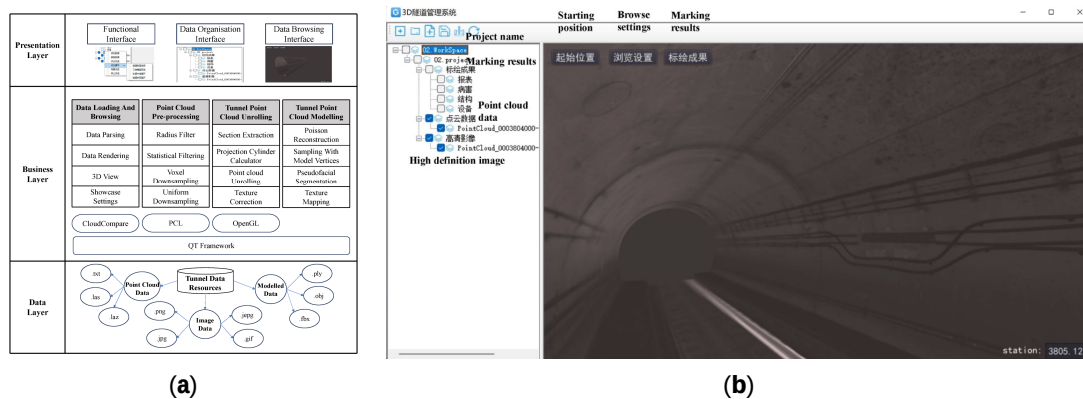


**Figure 12.** Prototype system architecture and interface diagram: (a) Architecture of the 3D tunnel management system; (b) System interface.

The data loading and browsing module parses different formats of tunnel point cloud and model data, displaying the main information in a tree structure within the data directory window (Figure 12b). The tunnel point cloud preprocessing module performs the initial processing of point clouds using common denoising and sampling methods. The point cloud unrolling module extracts spatial and color information from point cloud data to convert it into 2D images, which aids in tunnel disease management and 3D modeling. The point cloud modeling module allows users to select point cloud data for reconstruction, calculates the normal information of the point cloud, and builds high-precision tunnel models.

### 3.2. Description of Experimental Data

To assess the effectiveness and reliability of the proposed method, an experiment was conducted using four sets of point cloud data representing rectangular, horseshoe, circular, and elliptical tunnels, as shown in Figure 13.

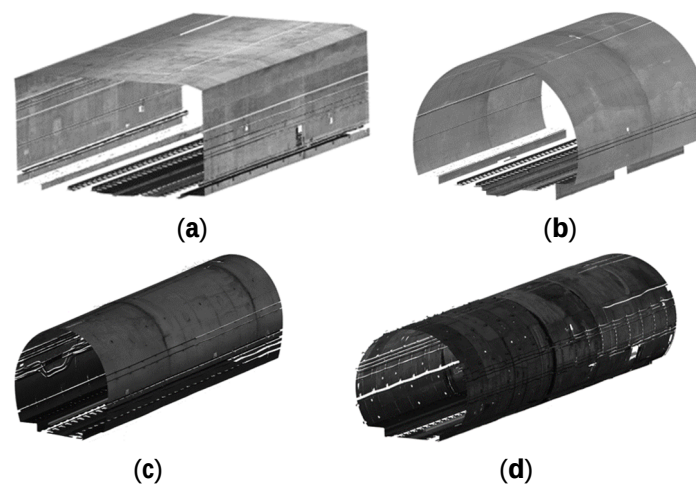


**Figure 13.** Tunnel point cloud data: (a) Rectangular tunnel; (b) Elliptical tunnel; (c) Horseshoe tunnel; (d) Circular tunnel.

### 3.3. Tunnel Reconstruction

#### 3.3.1. Converting Multi-Shaped Tunnel Point Cloud into 2D Intensity Image

The selected point cloud data were extracted from tunnel point cloud files, with each file covering approximately 20 m in length. The statistical outlier removal algorithm and uniform sampling were applied to remove the noise points and refine the point clouds. The point cloud information for the four sets of point cloud data before and after denoising and sampling is presented in a .txt format, as shown in Table 1.

**Table 1.** Preprocessing before and after the number of point clouds for different shaped tunnels.

| Types of Tunnels   | Original Point Cloud | Sampled Point Cloud | Denoised Point Cloud |
|--------------------|----------------------|---------------------|----------------------|
| Rectangular tunnel | 46,475,240           | 5,979,191           | 5,661,672            |
| Elliptical tunnel  | 45,056,421           | 5,809,372           | 5,492,585            |
| Horseshoe tunnel   | 38,107,115           | 5,073,745           | 4,761,269            |
| Circular tunnel    | 46,484,564           | 7,147,574           | 7,052,502            |

The point cloud data before and after preprocessing are compared in Figure 14. Comparison results reveal that many noise points have been removed and that the density of the original data points has decreased.

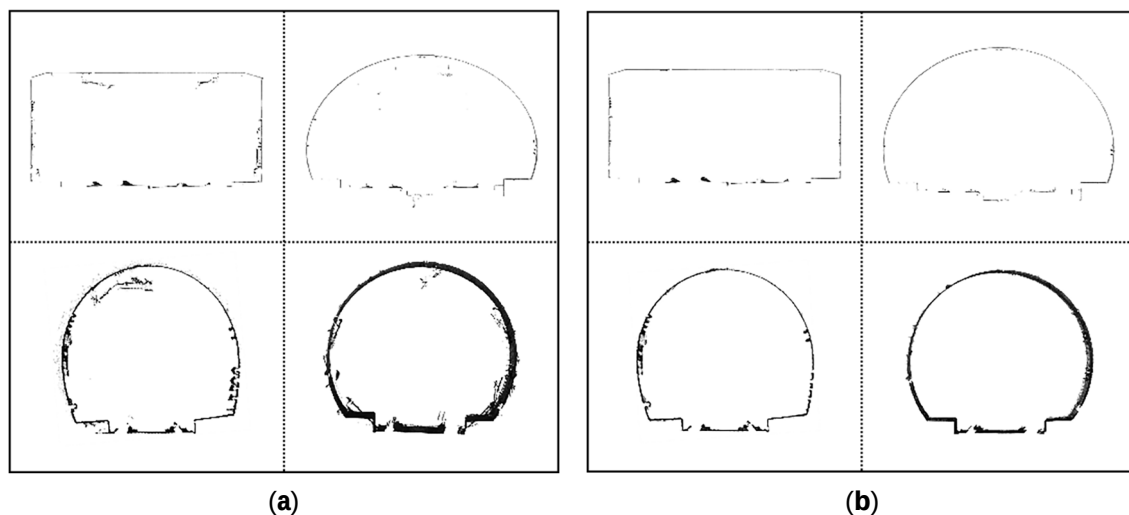


**Figure 14.** Preprocessing before and after cross-sections of different shaped tunnels: (a) Original tunnel point cloud cross-section.; (b) Preprocessed tunnel point cloud cross-section.

The method described in Section 2.1.1 was applied to extract the tunnel cross-sections based on the preprocessed point cloud data. An experimental analysis reveals that point sets can be efficiently extracted from tunnel cross-sections of different shapes when parameter  $\alpha$  is set to 0.5. The minimum bounding circle of the tunnel cross-section was then calculated and combined with the tunnel point cloud mileage length to obtain the parameters of the minimum bounding cylinder.

The initial cross-sections of point clouds for tunnels of different shapes are presented in Figure 15, along with their minimum bounding circles. Regardless of the cross-sectional shape of these tunnels, all points are covered by a minimum bounding circle. The proposed tunnel cross-section point sets standardization method was then applied, and the standardized cross-section of tunnels of any shape shows a point at every angle.

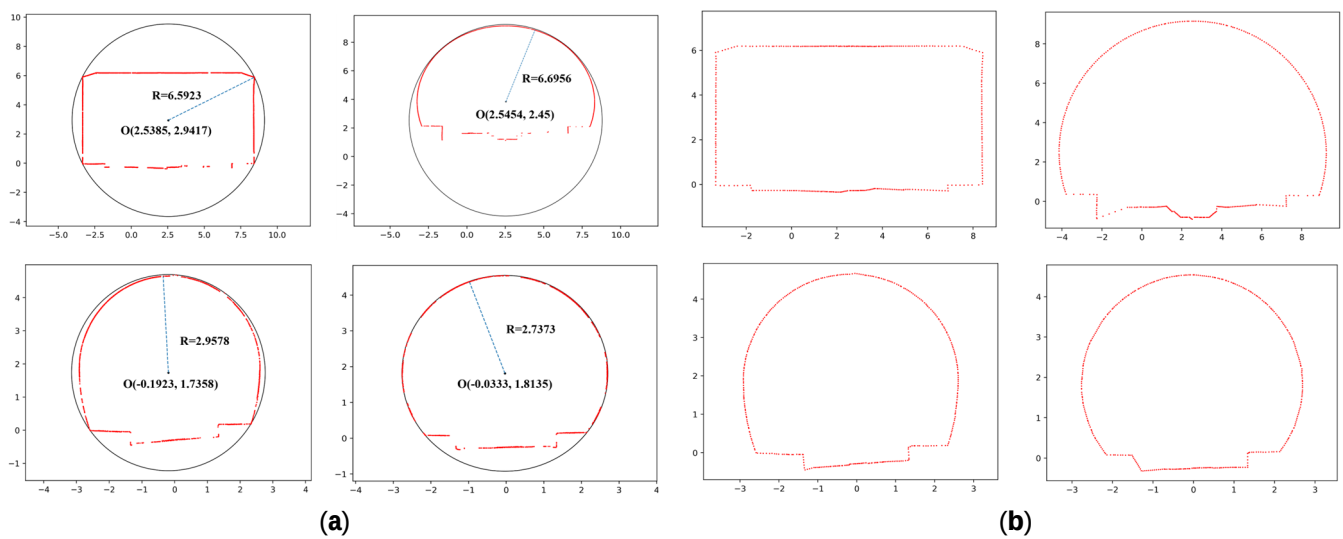


**Figure 15.** The results of tunnel point cloud projection calculations and cross-section standardization: (a) Minimum bounding circle of the tunnel cross-section; (b) Standardized tunnel cross-section.

Considering the accuracy of the point cloud and image, it is essential to determine an appropriate grid resolution when generating an intensity image from the unfolded point cloud. A very low resolution may greatly limit the point cloud data of pixel cells in the image and ultimately affect the quality of the image. However, a very high resolution may limit the precision of the image. The intensity map in this study is created using the grid resolutions of 0.0010, 0.0012, 0.0014, and 0.0018 as parameters. Figure 16 shows the intensity images created using different resolution parameters. The calculation time for creating intensity images at the four different grid resolutions is 823.62s, 308.14s, 243.33s, and 200.85s, respectively.

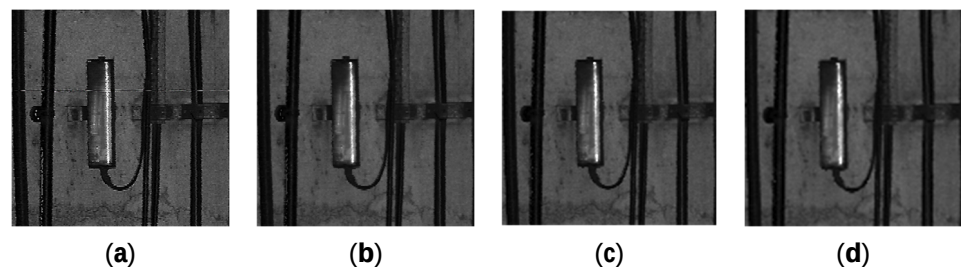


**Figure 16.** Intensity images with different grid resolutions: (a) Grid resolution = 0.0010; (b) Grid resolution = 0.0012; (c) Grid resolution = 0.0014; (d) Grid resolution = 0.0018.

When the grid resolution does not exceed 0.0012, some pixel information at the seams is missing. However, such information becomes blurred when the resolution is increased further to 0.0014. Based on these observations, the grid resolution is set to 0.0012 in this study.

Based on the projected cylinder and standard cross-sectional point sets obtained above, the tunnel point cloud was unrolled into an initial image and subsequently rectified based on the actual tunnel lining geometric features.

Magnified views of the original point cloud, initial image, and rectified image at the same location are presented in Figure 17, where labels 1 and 2 point to the tunnel crown and track surface, respectively. For example, Result-1 presents a magnified view of the crown position in the rectified image. A significant deformation can be observed at the tunnel crown in the initial image of the rectangular tunnel, but the rectified image of the same location closely resembles the original data. These results can be explained by the significant deformation at the crown and the track surface when projecting the rectangular

point cloud onto a cylindrical surface. The point cloud for the tunnel crown in the elliptical tunnel was marked by a square label, which remains almost unchanged before and after rectification. The track surface in the initial image shows a greater deformation when compared with the original point cloud, whereas the rectified image presents the true track information.

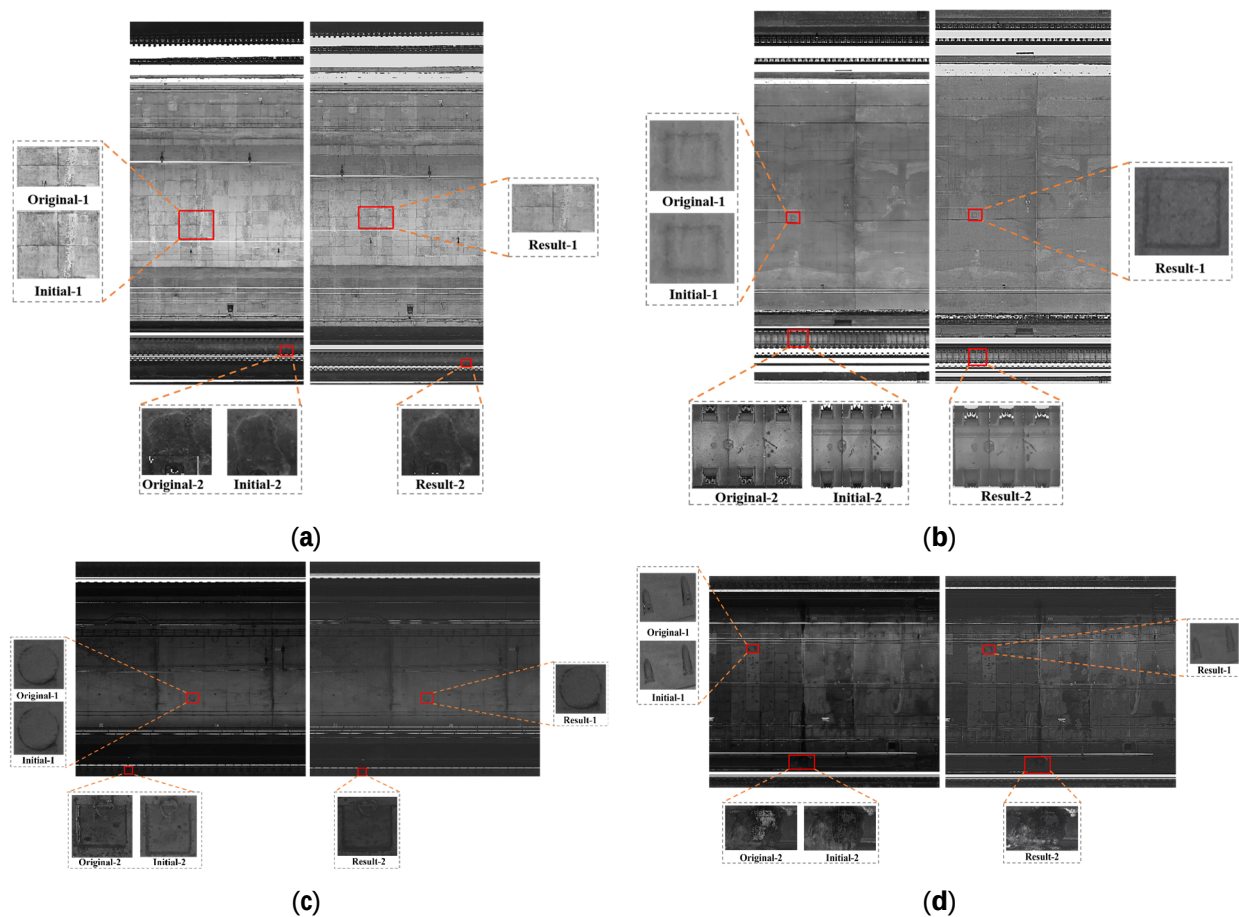


**Figure 17.** Intuitive illustration of 2D intensity images: (a) Rectangular tunnel; (b) Elliptical tunnel; (c) Horseshoe tunnel; (d) Circular tunnel.

In the horseshoe-shaped tunnel, the circular symbol at the tunnel crown and the rail presser on the track surface were marked. The circular symbol does not show any significant changes before and after correction, but the shape of the rail presser in the original image demonstrates significant compression. After rectification, the shape of the rail presser closely resembles that in the original point cloud. The point cloud of the circular tunnel shows water leakages in the pedestrian passage and screws on the tunnel crown lining. The shape of the screws does not change much after rectification, but the water leakages are compressed in the original image. This compression is effectively addressed after rectification, with the shape of water leakages closely resembling that in the original point cloud.

The variation in correction values with respect to angle is presented in Figure 18. The projection cylinder is unfolded from the tunnel track centerline. A greater distance between the point cloud and the projection cylinder surface corresponds to a greater deformation in the converted image and greater requirements for stretching or compressing during correction. Due to the great distance between the projection cylinder surface and the point clouds at the crown and track bottom of the rectangular tunnel, the converted image shows significant deformations, with the corresponding correction values largely deviating from 1. Meanwhile, the point clouds at the crown for the other three tunnel shapes are close to the

projection cylinder surface. Therefore, the rectification ratio of the sub-images fluctuates around 1, conforming to the deformation of the cylindrical projection of the tunnel point cloud. The final result is a high-quality 2D image that matches the tunnel lining structure.

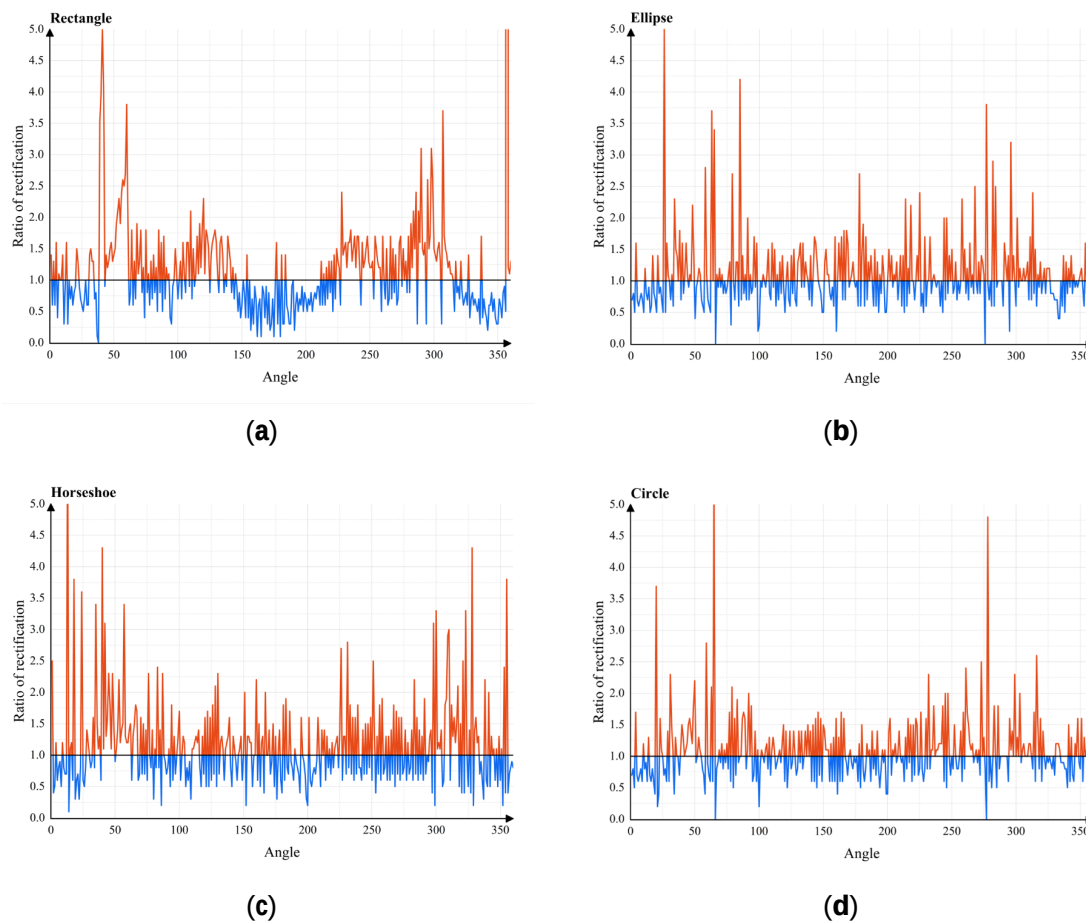


**Figure 18.** Rectification ratio of sub-images: (a) Rectangular tunnel; (b) Elliptical tunnel; (c) Horseshoe tunnel; (d) Circular tunnel.

### 3.3.2. Constructing 3D Model of Tunnel

The initial tunnel model was constructed by the Poisson surface reconstruction method, where the octree defined in the algorithm improves the search speed for spatial elements. As the octree depth increases, mesh resolution increases. Therefore, the memory consumption also increases drastically. The default value of octree depth used in Poisson reconstruction is eight. As shown in Figure 19, these models can preliminarily represent the spatial structure of the tunnel. However, the surface features of the traditional Poisson algorithm have poor sealing property, and it is easy to generate pseudo surfaces by misconnecting the regions that do not belong to morphological features at the edges, among which Figure 19a is the most obvious.

Therefore, to guarantee accuracy, the triangular mesh vertices were sampled using the method described in Section 2.2.2, and the pseudo-surface segmentation thresholds for the tunnel models were subsequently calculated based on these sampled points. The segmentation thresholds are presented in Table 2, along with the number of triangular meshes and vertices. As shown in the Table 2, the clipped model has approximately 4.3% fewer vertices and triangular meshes.

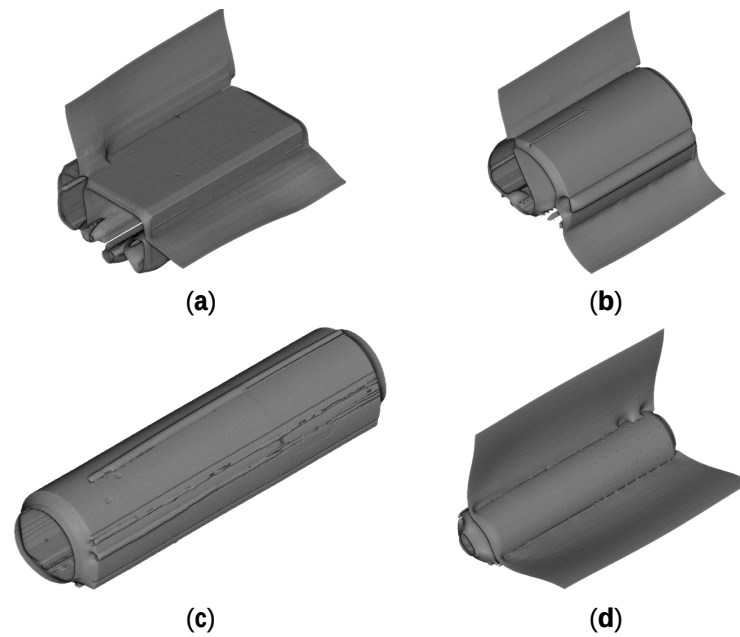


**Figure 19.** Models constructed by Poisson surface reconstruction: (a) Rectangular tunnel; (b) Elliptical tunnel; (c) Horseshoe tunnel; (d) Circular tunnel.

**Table 2.** Results of segmenting pseudo-surfaces with adaptive thresholds.

| Types of Tunnels   | The Initial Model |                   | Sampling Points | Segmentation Threshold | Clipped Model |                   |
|--------------------|-------------------|-------------------|-----------------|------------------------|---------------|-------------------|
|                    | Vertices          | Triangular Meshes |                 |                        | Vertices      | Triangular Meshes |
| Rectangular tunnel | 1,719,180         | 3,438,077         | 48,376          | 0.4233                 | 1,666,180     | 3,292,459         |
| Elliptical tunnel  | 2,135,805         | 4,270,928         | 59,623          | 0.4058                 | 1,997,842     | 3,930,464         |
| Horseshoe tunnel   | 1,148,629         | 2,297,680         | 42,081          | 0.1390                 | 1,131,461     | 2,244,642         |
| Circular tunnel    | 992,053           | 1,983,676         | 39,722          | 0.1271                 | 968,323       | 1,915,498         |

After generating the clipped 3D mesh model, the 2D image can be used as a high-definition texture for the 3D mesh tunnel model according to the automatic texture mapping method. The 3D mesh models after clipping the pseudo-surfaces are presented in Figure 20, along with the 3D textured model. The unfolded tunnel image is accurately mapped onto the model, thereby suggesting that these 3D tunnel models can accurately and comprehensively represent the structure of the tunnel along with its facilities, including platforms, pipes, and tracks, hence facilitating an accurate visualization and analysis of this structure. These 3D tunnel models are, therefore, invaluable for many applications, including tunnel maintenance, simulation, and renovation planning.

To evaluate the accuracy of the constructed tunnel model, this article takes the original tunnel point cloud as the true value and calculates the absolute distance from the point cloud to the model as the quantitative evaluation index of modeling accuracy.

Figure 21 delineates the accuracy of the two reconstruction algorithms for the four types of data. Additional distance metrics are provided in Table 3, where  $D_{max}$ ,  $D_{min}$ , and  $D_{Avg}$  represent the maximum, minimum, and average distances from the original point cloud to the model, respectively. A smaller distance corresponds to a better match between the model and the original point cloud data and a greater model accuracy. A huge distance between the original point cloud and the model can be observed in only a small number of points; these points are therefore excluded from Figure 21. This exclusion also explains why the maximum distance on the horizontal axis of Figure 21 slightly differs from the corresponding  $D_{max}$  presented in Table 3.

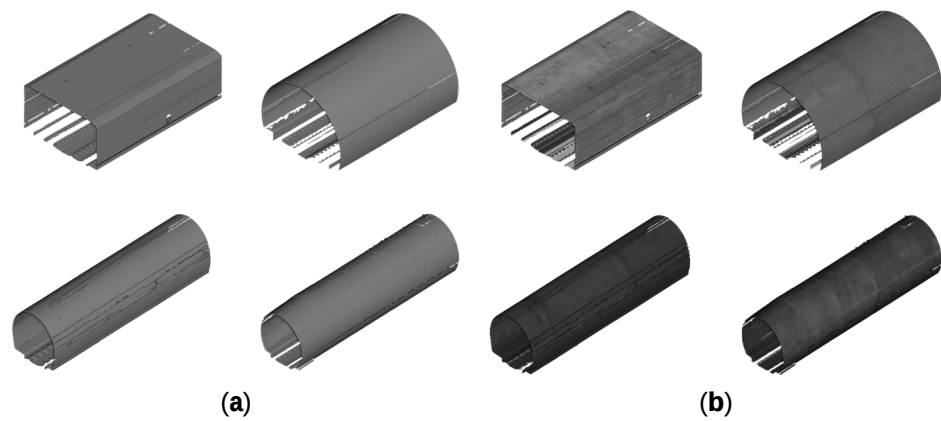


**Figure 20.** Intuitive illustration of 3D tunnel model: (a) Clipped 3D mesh model; (b) 3D texture model.

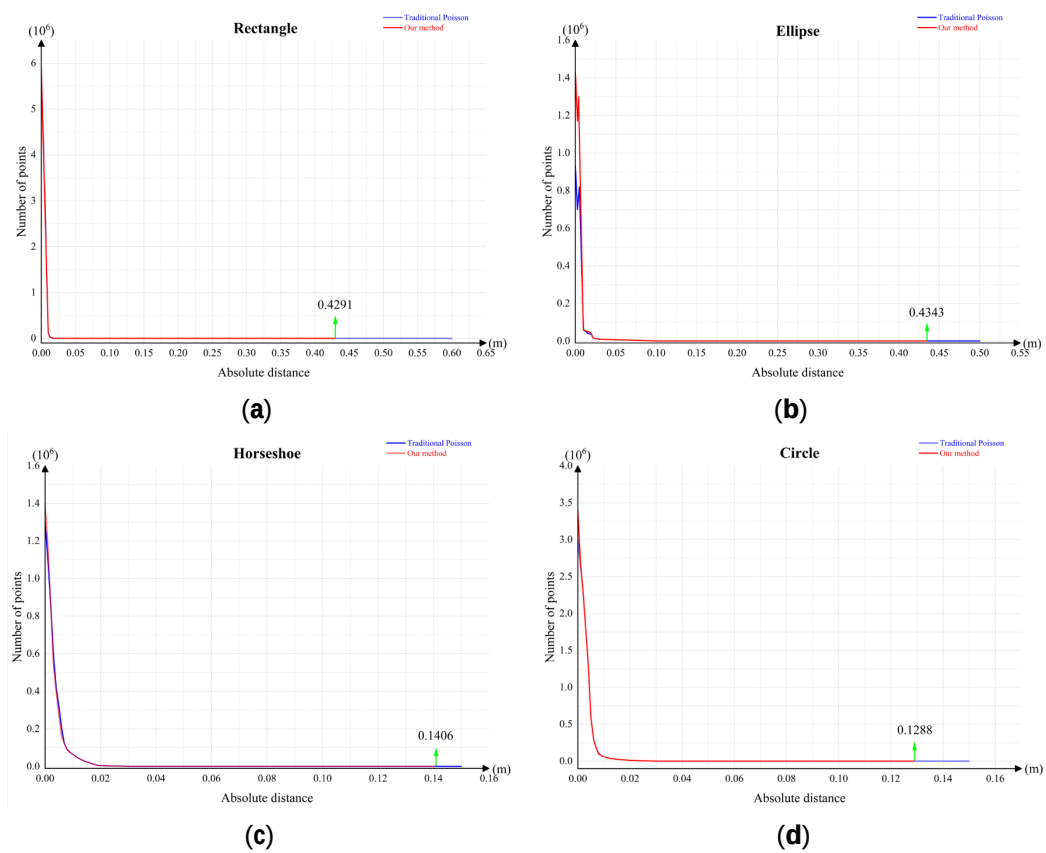


**Figure 21.** The variation in the number of the points with the absolute distance difference between the reconstructed model and the input points: (a) Rectangular tunnel; (b) Elliptical tunnel; (c) Horseshoe tunnel; (d) Circular tunnel.

**Table 3.** Comparison of distance metrics from the input point cloud to the model constructed by two algorithms.

| Types of Tunnels   | Traditional Poisson Reconstruction |               |               | Proposed Method |               |               |
|--------------------|------------------------------------|---------------|---------------|-----------------|---------------|---------------|
|                    | $D_{max}$ (m)                      | $D_{min}$ (m) | $D_{avg}$ (m) | $D_{max}$ (m)   | $D_{min}$ (m) | $D_{avg}$ (m) |
| Rectangular tunnel | 3.1952                             | 0             | 0.00078       | 1.9687          | 0             | 0.00035       |
| Elliptical tunnel  | 2.5869                             | 0             | 0.00038       | 1.5214          | 0             | 0.00023       |
| Horseshoe tunnel   | 0.7093                             | 0             | 0.00081       | 0.5488          | 0             | 0.00076       |
| Circular tunnel    | 2.1720                             | 0             | 0.00067       | 0.8574          | 0             | 0.00058       |

The point clouds of the four types of data have maximum distances of 0.4291, 0.4343, 0.1406, and 0.1288 m from the constructed models, respectively. All of these values are less than the maximum distances recorded by traditional Poisson models. In conclusion, the method proposed in this paper is more effective than the traditional Poisson surface reconstruction algorithm, significantly reducing the number of pseudo-surfaces in the 3D tunnel models and thereby improving both the accuracy and the quality of the models.

#### 4. Discussion

The proposed method for 3D reconstruction of multi-shaped tunnels using mobile laser scanning data involves two critical processes: the generation of 2D texture images from tunnel point clouds and the construction of 3D tunnel mesh models. These processes are essential for creating high-precision and high-quality digital tunnel models. Consequently, we conducted evaluations from two perspectives to assess the effectiveness of the proposed method: the quality of the generated 2D images and the accuracy of the 3D mesh models.

##### 4.1. Comparison of 3D Point Clouds Conversion Results

To further verify the feasibility of the multi-shaped tunnel point cloud image generation method, we compared the results generated by the proposed method (Figure 22b,d) with those of the currently commonly used cylindrical projection method (Figure 22a,c). The cylindrical projection method is usually based on the cylindrical surface equation to expand the 3D shield tunnel point clouds into a 2D point cloud, which is then converted into a 2D image and assigned intensity values to generate 2D intensity images. However, due to the disordered nature of the 3D point clouds, the 2D intensity images generated based on the cylindrical projection method are prone to deformation, especially when dealing with non-circular tunnel point clouds. Therefore, in this section, the horseshoe-shaped and rectangular tunnel point clouds are used as experimental data to evaluate the 2D intensity images generated by the aforementioned methods from both qualitative and quantitative perspectives. The evaluation metrics include image deformation and texture quality.

**Qualitative evaluation:** Image deformation and texture differences were used as qualitative evaluation metrics. To highlight these features, regions of interest from the above four images were plotted based on the ArcGIS 10.2 mapping software. No significant differences can be observed between the 2D intensity images generated by the proposed method and the cylindrical projection method for the horseshoe-shaped point cloud. The 2D intensity images generated using the cylindrical projection method for rectangular tunnel point clouds exhibit significant deformation. In contrast, the method proposed in this paper better preserves the global shape and local details of the tunnel lining, indicating higher stability and accuracy in handling various geometric shapes.

**Quantitative evaluation:** Deformation quantification and texture quality serve as quantitative evaluation metrics for assessing the performance of 2D intensity images generated by two methods based on cylindrical projection and our method. Regarding the deformation quantification metric, two regions of interest, namely Rectangle 1 and Rectangle 2, were extracted at the same positions in each image in Figure 22. We quantified the extent of image deformation using area distortion, which indicates the ratio of the true area of a region of interest to the actual area calculated based on the original point cloud. The extent of two area deformations is presented in Table 4.

The regions of interest for the horseshoe-shaped tunnel images obtained using the two methods have an area ratio close to 1, which suggests that these regions have minimal to almost no deformation. However, the region of interest for those images obtained using the standard cylindrical projection method has an area ratio significantly deviating from 1. This deviation reflects significant deformation in the 2D intensity image of the rectangular tunnel obtained using this method. Therefore, in terms of deformation quantification, the proposed method outperforms the standard cylindrical projection method, especially for non-circular tunnel point clouds.

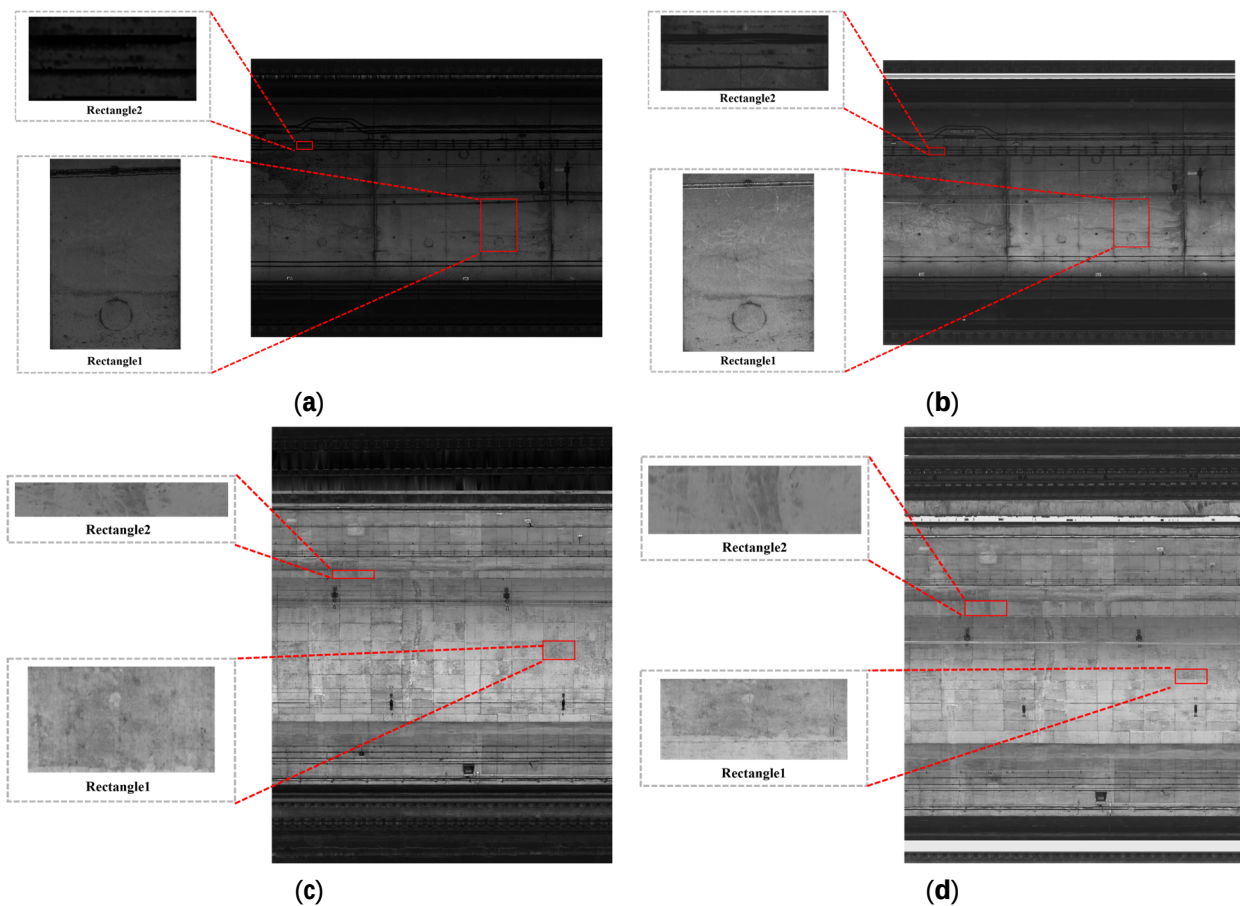


**Figure 22.** Qualitative image comparison and Regions of Interest: (a) Rectangle 1 and Rectangle 2 in tunnel images generated by cylindrical projection from horseshoe tunnel point cloud; (b) Rectangle 1 and Rectangle 2 in tunnel images generated by the proposed method from horseshoe tunnel point cloud; (c) Rectangle 1 and Rectangle 2 in tunnel images generated by cylindrical projection from rectangular tunnel point cloud; (d) Rectangle 1 and Rectangle 2 in tunnel images generated by the proposed method from rectangular tunnel point cloud.

**Table 4.** Area distortion of two methods applied to tunnels of different shapes.

|                        | ROI 1 of Horseshoe Tunnel | ROI 2 of Horseshoe Tunnel | ROI 1 of Rectangular Tunnel | ROI 2 of Rectangular Tunnel |
|------------------------|---------------------------|---------------------------|-----------------------------|-----------------------------|
| Cylindrical projection | 0.984                     | 1.0134                    | 2.389                       | 0.544                       |
| Our method             | 0.997                     | 1.0147                    | 0.901                       | 1.06789                     |

To gain a deeper understanding of the image quality and detail performance shown in Figure 22, we evaluated the image quality based on the mean value, standard deviation, and average gradient. The mean is the average value of the image pixels. It represents the average brightness of the image. The higher the average brightness, the better the image quality. Standard deviation refers to the degree of dispersion of the grayscale value of an image pixel relative to the mean. A higher standard deviation means that the grayscale levels in the image are more dispersed, and the image quality is better. The average gradient expresses the contrast of details and texture transformation in the image, and to a certain extent, it can represent the clarity of the image. Generally, the higher the average gradient, the more changeable, richer, and clearer the image. By comparing the quality metrics of images generated by the two methods, we can more comprehensively assess their

differences in detail performance and overall image quality. The results of these metrics are summarized in Table 5.

**Table 5.** Quality indicators for tunnel intensity images.

|                        | Images of Horseshoe Tunnel |                    |                  | Images of Rectangular Tunnel |                    |                  |
|------------------------|----------------------------|--------------------|------------------|------------------------------|--------------------|------------------|
|                        | Mean                       | Standard Deviation | Average Gradient | Mean                         | Standard Deviation | Average Gradient |
| Cylindrical projection | 40.77                      | 27.18              | 3.72             | 93.67                        | 52.13              | 4.01             |
| Our method             | 66.71                      | 39.92              | 3.87             | 117.97                       | 57.40              | 6.52             |

Table 5 shows that the proposed method exhibits greater standard deviation and average gradient than the method based on cylindrical projection, highlighting its significant advantage in preserving image details.

After the above analysis, we conclude that the 2D intensity image generated by the proposed method in this paper outperforms traditional cylindrical projection methods in multiple ways. This method not only improves image quality but also provides us with richer and more accurate information.

#### 4.2. Comparison of Reconstruction Results

In order to evaluate the reconstruction result, the proposed method was compared with two widely accepted 3D surface reconstruction algorithms: the Delaunay triangulation algorithm and the Alpha Shape reconstruction algorithm. Considering that the accuracy of model reconstruction is our interest, all comparisons here mainly focus on the distance errors between the modeling results and the raw scan point data, which are plotted in Figures 23 and 24.

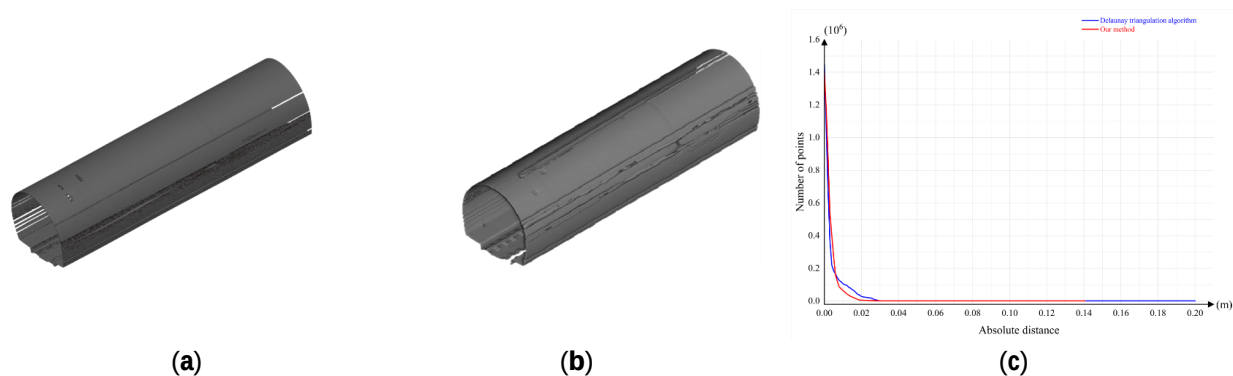


**Figure 23.** Qualitative modeling comparison from (a) Delaunay triangulation and (b) ours. The quantitative statistics are plotted in (c), where the horizontal axis stands for the absolute distance error from the reconstructed model to the input LiDAR scan, and the vertical axis is the number of points.

Despite achieving a uniform and regular surface using the Delaunay triangulation algorithm, a significant number of holes can be observed in the generated model due to the presence of noise, including pipelines and light strips, in the point clouds. Meanwhile, even if the fineness of the triangulated mesh generation can be controlled using a different parameter  $\alpha$  in the Alpha Shape reconstruction algorithm, some unnecessary details are generated on the tunnel surface through this process. By building on the traditional Poisson surface reconstruction algorithm, the proposed method identifies the optimal surface representation across the entire point cloud dataset. This approach creates a smooth and high-quality 3D mesh model while effectively constraining the generation of pseudo-surfaces.

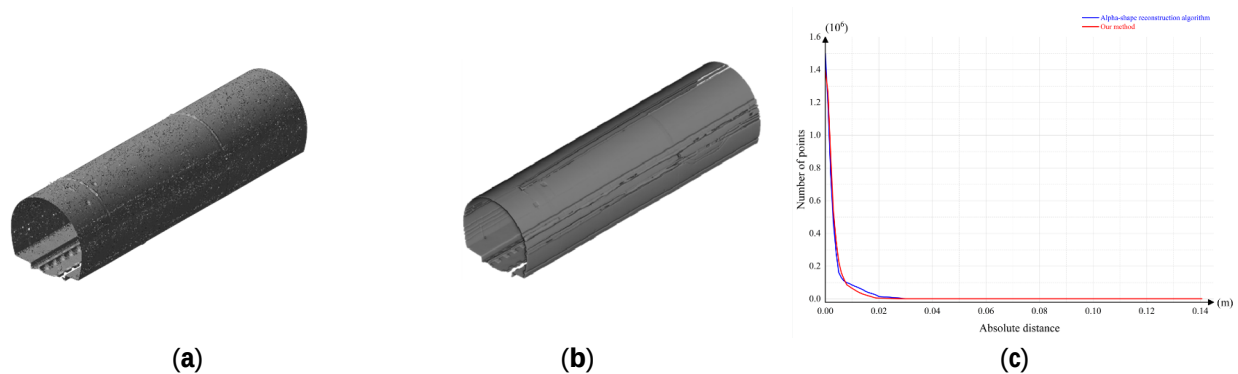


**Figure 24.** Qualitative modeling comparison from (a) Alpha Shape reconstruction and (b) ours. The quantitative statistics are plotted in (c).

While the distance errors in models constructed by the Delaunay triangulation algorithm and Alpha Shape reconstruction algorithm are primarily concentrated in smaller ranges, some errors are still distributed over larger distances. This is due to the non-uniform density of the original point cloud along the tunnel sides, which leads to the creation of inaccurate triangular facets in areas with sparse point data. Despite demonstrating superior accuracy compared with the models generated by the above algorithms, the model generated by the proposed method has a smoother surface due to the global optimization characteristic of the Poisson reconstruction, thus reducing the detailed features of the tunnel lining surfaces. In sum, the proposed method demonstrates its superiority in generating smooth surfaces and handling noisy data, thereby generating highly accurate 3D tunnel models.

## 5. Conclusions

Based on the analysis of the geometric structure characteristics of tunnel linings and the Poisson reconstruction process, this study proposes a method for the 3D reconstruction of multi-shaped tunnels using mobile laser scanning data. The proposed method does not require predefined mathematical models or projection parameters to convert point cloud data into 2D images and generate high-quality texture information for the construction of 3D tunnel models. This method also adopts an adaptive thresholding approach that enhances the precision of the 3D tunnel models by effectively segmenting the pseudo-surfaces in the Poisson reconstruction model. Using point cloud data to generate the 3D tunnel model and texture images eliminates the need to acquire secondary images. This further improves the efficiency of tunnel modeling and provides high-precision 3D models and high-quality textures for digital tunnel engineering and information management. The efficacy of the proposed method is tested by using a 3D Tunnel Management System and performing an experiment using point cloud data from four tunnels of different shapes. Experimental results confirm that the 2D intensity images generated by the proposed method can accurately reflect the local details and internal environment of tunnels. These images also have a rectification ratio that aligns with the projection deformation characteristics. The proposed method also obtains a higher reconstruction accuracy compared with the traditional Poisson reconstruction algorithm. This method is then qualitatively and quantitatively compared with other related methods to highlight its superior flexibility and efficacy in the 3D reconstruction of multi-shaped tunnel point clouds.

Despite generating high-quality textures and high-precision 3D models for digital tunnels, the proposed method may compromise the realism and visualization quality due to the jagged edges resulting from pseudo-surface segmentation. Additionally, the standardization method for tunnel cross-sections performs well when handling straight segments, generating relatively smooth sectional profiles quickly and effectively, which meets the practical needs of tunnel management engineering. However, for complex curved segments, this method may not provide sufficient smoothness, leading to jagged edges

in the model. Therefore, future research can further explore and improve interpolation methods, such as cubic spline interpolation or B-spline interpolation, to enhance the smoothness of curved segments and the overall visual quality of the models.

**Author Contributions:** Conceptualization, S.C. and M.D.; methodology, X.D., S.C., J.S. and C.H.; software, S.C., M.D., C.L. and J.S.; validation, S.C. and C.L.; formal analysis, X.D. and M.D.; investigation, X.D. and S.C.; resources, M.D. and J.S.; data curation, M.D., C.L. and J.S.; writing—original draft preparation, X.D.; writing—review and editing, S.C.; visualization, M.D.; supervision, J.S. and C.H.; project administration, J.S. and C.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by grants from the National Nature Science Foundation of China (NSFC) Program (No. 42401555), Hubei Provincial Natural Science Foundation (Grant No. JCZRQN202400143), Wuhan Municipal Science and Technology Bureau Scientific and Technological Achievement Transformation Project (No.2023030103010726).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to their integration within the team's internal database, which is structured for internal access and use. Direct public access to this database is not feasible due to operational, privacy, and security protocols.

**Conflicts of Interest:** Xuan Ding, Mu Duan and Chuli Hu were employed by Wuhan CUG Smart City Research Institute Co., Ltd. Jinchang Shan was employed by Wuhan Hanyang Municipal Construction Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors declare no conflicts of interest.

## References

- Li, Y.; Xiao, Z.; Li, J.; Shen, T. Integrating Vision and Laser Point Cloud Data for Shield Tunnel Digital Twin Modeling. *Autom. Constr.* **2024**, *157*, 105180. [[CrossRef](#)]
- Han, Y.; Sun, H.; Zhong, R. Three-Dimensional Linear Restoration of a Tunnel Based on Measured Track and Uncontrolled Mobile Laser Scanning. *Sensors* **2021**, *21*, 3815. [[CrossRef](#)] [[PubMed](#)]
- Ji, C.; Sun, H.; Zhong, R.; Li, J.; Han, Y. Precise Positioning Method of Moving Laser Point Cloud in Shield Tunnel Based on Bolt Hole Extraction. *Remote Sens.* **2022**, *14*, 4791. [[CrossRef](#)]
- Liu, C.; Liu, Y.; Chen, Y.; Zhao, C.; Qiu, J.; Wu, D.; Liu, T.; Fan, H.; Qin, Y.; Tang, K. A State-of-the-Practice Review of Three-Dimensional Laser Scanning Technology for Tunnel Distress Monitoring. *J. Perform. Constr. Facil.* **2023**, *37*, 03123001. [[CrossRef](#)]
- Yi, C.; Lu, D.; Xie, Q.; Xu, J.; Wang, J. Tunnel Deformation Inspection via Global Spatial Axis Extraction from 3D Raw Point Cloud. *Sensors* **2020**, *20*, 6815. [[CrossRef](#)]
- Xia, M.; Mao, Q.; Wang, G.; Fan, T. A Subway Tunnel Image Stitching Method Based on Point Cloud Mapping Relationships and High-Resolution Image. *Eng. Res. Express* **2024**, *6*, 025102. [[CrossRef](#)]
- Huang, Z.; Fu, H.; Fan, X.; Meng, J.; Chen, W.; Zheng, X.; Wang, F.; Zhang, J. Rapid Surface Damage Detection Equipment for Subway Tunnels Based on Machine Vision System. *J. Infrastruct. Syst.* **2021**, *27*, 04020047. [[CrossRef](#)]
- Sun, H.; Liu, S.; Zhong, R.; Du, L. Cross-Section Deformation Analysis and Visualization of Shield Tunnel Based on Mobile Tunnel Monitoring System. *Sensors* **2020**, *20*, 1006. [[CrossRef](#)]
- Li, P.; Wang, Q.; Li, J.; Pei, Y.; He, P. Automated Extraction of Tunnel Leakage Location and Area from 3D Laser Scanning Point Clouds. *Opt. Lasers Eng.* **2024**, *178*, 108217. [[CrossRef](#)]
- Xu, X.; Yang, H. Intelligent Crack Extraction and Analysis for Tunnel Structures with Terrestrial Laser Scanning Measurement. *Adv. Mech. Eng.* **2019**, *11*, 1687814019872650. [[CrossRef](#)]
- Qiu, W.; Cheng, Y.-J. High-Resolution DEM Generation of Railway Tunnel Surface Using Terrestrial Laser Scanning Data for Clearance Inspection. *J. Comput. Civ. Eng.* **2017**, *31*, 04016045. [[CrossRef](#)]
- Sun, H.; Xu, Z.; Yao, L.; Zhong, R.; Du, L.; Wu, H. Tunnel Monitoring and Measuring System Using Mobile Laser Scanning: Design and Deployment. *Remote Sens.* **2020**, *12*, 730. [[CrossRef](#)]
- Zhang, Z.; Ji, A.; Wang, K.; Zhang, L. UnrollingNet: An Attention-Based Deep Learning Approach for the Segmentation of Large-Scale Point Clouds of Tunnels. *Autom. Constr.* **2022**, *142*, 104456. [[CrossRef](#)]
- Chen, Q.; Kang, Z.; Cao, Z.; Xie, X.; Guan, B.; Pan, Y.; Chang, J. Combining Cylindrical Voxel and Mask R-CNN for Automatic Detection of Water Leakages in Shield Tunnel Point Clouds. *Remote Sens.* **2024**, *16*, 896. [[CrossRef](#)]

15. Alsabhan, A.H. The effect of opening shapes on the stability of underground tunnels: A finite element analysis. *Int. J. Geomate* **2021**, *21*, 19–27. [\[CrossRef\]](#)
16. Zhao, Y.; Li, A.; Du, Z.; Chen, Y.; Sun, H.; Zhi, Z. Joint Structure Detection and Multi-Scale Clustering Filtering for Tunnel Lining Extraction From Point Clouds. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 11214–11226. [\[CrossRef\]](#)
17. Liu, Y.; Zhong, R.; Chen, W.; Sun, H.; Ren, Y.; Lei, N. Study of Tunnel Surface Parameterization of 3-D Laser Point Cloud Based on Harmonic Map. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1623–1627. [\[CrossRef\]](#)
18. Deng, F.; Yang, J. Panoramic Image Generation Using Centerline- Constrained Mesh Parameterization for Arbitrarily Shaped Tunnel Lining. *IEEE Access* **2020**, *8*, 7969–7980. [\[CrossRef\]](#)
19. Cheng, X.; Hu, X.; Tan, K.; Wang, L.; Yang, L. Automatic Detection of Shield Tunnel Leakages Based on Terrestrial Mobile LiDAR Intensity Images Using Deep Learning. *IEEE Access* **2021**, *9*, 55300–55310. [\[CrossRef\]](#)
20. Liao, J.; Yue, Y.; Zhang, D.; Tu, W.; Cao, R.; Zou, Q.; Li, Q. Automatic Tunnel Crack Inspection Using an Efficient Mobile Imaging Module and a Lightweight CNN. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 15190–15203. [\[CrossRef\]](#)
21. Jia, D.; Zhang, W.; Liu, Y. Systematic Approach for Tunnel Deformation Monitoring with Terrestrial Laser Scanning. *Remote Sens.* **2021**, *13*, 3519. [\[CrossRef\]](#)
22. Zhang, Z.; Ji, A.; Zhang, L.; Xu, Y.; Zhou, Q. Deep Learning for Large-Scale Point Cloud Segmentation in Tunnels Considering Causal Inference. *Autom. Constr.* **2023**, *152*, 104915. [\[CrossRef\]](#)
23. Fang, K.; Dong, A.; Tang, H.; Miao, M.; An, P.; Zhang, B.; Jia, S. 3D Tunnel Reconstruction and Visualization through Multi-Smartphone Photogrammetry. *Measurement* **2023**, *223*, 113764. [\[CrossRef\]](#)
24. Janiszewski, M.; Torkan, M.; Uotinen, L.; Rinne, M. Rapid Photogrammetry with a 360-Degree Camera for Tunnel Mapping. *Remote Sens.* **2022**, *14*, 5494. [\[CrossRef\]](#)
25. Liu, X.; Zhu, L.; Wang, Y.; Yu, Z. A Crack Detection System of Subway Tunnel Based on Image Processing. *Meas. Control* **2022**, *55*, 164–177. [\[CrossRef\]](#)
26. Xue, Y.; Shi, P.; Jia, F.; Huang, H. 3D Reconstruction and Automatic Leakage Defect Quantification of Metro Tunnel Based on SfM-Deep Learning Method. *Undergr. Space* **2022**, *7*, 311–323. [\[CrossRef\]](#)
27. Huang, H.; Cheng, W.; Zhou, M.; Chen, J.; Zhao, S. Towards Automated 3D Inspection of Water Leakages in Shield Tunnel Linings Using Mobile Laser Scanning Data. *Sensors* **2020**, *20*, 6669. [\[CrossRef\]](#)
28. Song, Q.; Xu, D.; Fang, S. 3D Tunnel Surface Reconstruction Method in Complex Scenarios. In Proceedings of the 2021 China Automation Congress (CAC), Beijing, China, 22–24 October 2021; pp. 1369–1374.
29. Zhao, H.; Hu, Q. 3D Reconstruction and Rendering Visualization of Tunnel Point Cloud-Web of Science Core Collection. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *48*, 93–101. [\[CrossRef\]](#)
30. Ao, X.; Wu, H.; Xu, Z.; Gao, Z. Damage Extraction of Metro Tunnel Surface from Roughness Map Generated by Point Cloud. In Proceedings of the 2018 26th International Conference on Geoinformatics, Kunming, China, 28–30 June 2018; pp. 1–4.
31. Qi, Z.; He, R.; Yang, M.; Zhang, X.; Ai, B.; Li, W.; Wang, S.; Sun, W.; Li, L. Point Cloud-Based Environment Reconstruction and Ray Tracing Simulations for Railway Tunnel Channels. *High-Speed Railw.* **2023**, *1*, 241–247. [\[CrossRef\]](#)
32. Guo, J.; Jiang, J.; Wu, L.; Zhou, W.; Wei, L. 3D Modeling for Mine Roadway from Laser Scanning Point Cloud. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; IEEE: Beijing, China, 2016; pp. 4452–4455.
33. Jiang, T.; Lin, X.; Fang, L.; Yao, T.; Zhao, Y.; Wei, Z. Research on Fitting Algorithm of Scattered Point Cloud Based on 3D Laser Scanning. *Mob. Inf. Syst.* **2022**, *2022*, 8432308. [\[CrossRef\]](#)
34. Xiang, L.; Ding, Y.; Wei, Z.; Zhang, H.; Li, Z. Research on the Detection Method of Tunnel Surface Flatness Based on Point Cloud Data. *Symmetry* **2021**, *13*, 2239. [\[CrossRef\]](#)
35. Liu, Z.; Wang, L.; Tahir, M.; Huang, J.; Cheng, T.; Guo, X.; Wang, Y.; Liu, C. An Improved Poisson Surface Reconstruction Algorithm Based on the Boundary Constraints. *Int. J. Adv. Comput. Sci. Appl. IJACSA* **2023**, *14*, 225–232. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.