JCFI: a composite index for RMLS-based shield tunnel segment joint recognition

Liying Wang*[a], Ze You[a], Yiwei Yu[a], Yong Feng[b], Chunxi Xie[c]

[a] School of Geomatics, Liaoning Technical University, Fuxin, China - (lwang, zyou, yyu)
[b] Division of Geoinformation Management, Department of Natural Resources of Liaoning Province, Shenyang, China
[c] Institute of Surveying, Mapping and Geographic Information, China Railway Design Group Co., LTD., Tianjin, China

Keywords: Shield Tunnel, Segment Joint, Rail-borne Mobile Laser Scanning, Composite Index

Abstract

The accurate recognition of segment joints serves as a critical step for capturing joint anomaly information, evaluating segment assembly quality, diagnosing structural health status, and determining the loosening of connecting bolts. It holds significant importance for the operation and maintenance of shield tunnels. However, existing studies on joint recognition based on Rail-borne Mobile Laser Scanning (RMLS) suffer from insufficient comprehensiveness in feature representation, leading to notably poor accuracy and robustness under complex scenarios such as noise interference, data loss due to object occlusion, and uneven point cloud density. To address this issue, this study proposes a shield tunnel segment joint recognition method based on the Joint Composite Feature Index (JCFI). The proposed method first employs a cross-sectional ellipse fitting approach to filter out obvious non-lining points. Subsequently, a composite index JCFI, which integrates curvature, left-right density ratio, and relative depth, is designed to quantitatively characterize the feature differences of segment joints. Finally, based on the constructed JCFI indicator, the recognition of circumferential and longitudinal joints is sequentially achieved. Validation tests using RMLS point cloud data from the Guangzhou Metro Line 8 tunnel demonstrate that the proposed method, by constructing the JCFI that comprehensively characterizes joint features, effectively handles complex scenarios including noise interference, joint missing, and uneven point cloud density. The joint recognition achieves a recall rate of 90.14%, a precision rate of 99.04%, and an IoU of 89.36%, providing a reliable technical solution for the accurate identification of shield tunnel segment joints.

1. Introduction

The subway, as the core component of urban rail transit systems, imposes stringent requirements on the safety, efficiency, and environmental adaptability for its underground tunnels. The shield tunneling method, owing to its high degree of automation, operational safety, broad geological adaptability, and minimal impact on surrounding environments, has been widely adopted in subway tunnel construction (Li et al., 2022). The main structure of a subway shield tunnel is formed by assembling numerous precast concrete segments connected via bolts. The joints between these segments (hereinafter referred to as “segment joints”) consequently represent inherent weak parts of the tunnel structure. These areas are prone to structural defects such as joint opening, segment dislocation, and spalling, which may lead to apparent issues like water leakage (Feng et al., 2024, Liu et al., 2021, Niu et al., 2022). Therefore, achieving efficient and accurate recognition of segment joints and digital reconstruction of their 3D spatial structures is of paramount importance for promptly detecting anomalies at joint locations, evaluating the quality of segment assembly, diagnosing structural health status, and judging the loosening of connecting bolts. This technical process enables effective identification and prevention of long-term threats posed by various defects to the tunnel structure, thereby ensuring the operational safety of subway tunnels and demonstrating significant engineering application value.

Rail-borne Mobile Laser Scanning (RMLS) technology can directly acquire high-precision 3D point clouds and their laser reflection intensity data of the entire tunnels. It possesses distinct digital acquisition advantages, providing a reliable data foundation for the automated and high-precision recognition of segment joints, and has thus been widely adopted by numerous research and practical institutions. However, the complex internal environment of subway tunnels poses challenges to the high-precision recognition of segment joints. First, the dense pipelines and intricate equipment arrangement in the tunnel directly introduce environmental noise, which interferes with the accurate identification of joints. Second, segment joints are often obscured by auxiliary facilities, resulting in incomplete point clouds in the corresponding areas and increasing the difficulty of capturing joint features. Third, the inherently uneven spatial distribution of point clouds disrupts the continuity characteristics that joints should exhibit, further complicating the overall task of joint identification. Currently, methods for recognizing segment joints in shield tunnels based on RMLS can be categorized into three types:

(1) Image processing-based methods (Yu et al., 2021, Feng et al., 2024, Li et al., 2024, Cui et al., 2024, Ye et al., 2025).These methods typically project 3D tunnel point clouds onto 2D representations (e.g., grayscale/intensity maps, depth images, or unwrapped tunnel panoramas), and then detect joint-related linear structures using classical operators (edge/line detectors, morphological filtering, etc.) or learning-based detectors on the 2D domain. Recent advances have focused on improving the robustness of the projection-and-detection pipeline, for example by jointly exploiting multi-modal cues (intensity + depth) to enhance joint saliency, adopting more stable unwrapping/projection strategies to reduce geometric distortion, and incorporating structural priors (e.g., the spatial relationship between bolt holes and joints) to constrain the search space and suppress false positives. Moreover, with the rapid development of 2D vision models, several studies have explored two-stage or hybrid frameworks on projected tunnel images for detecting tunnel components and related anomalies, which also provides transferable technical insights for joint localization. Neverthe-

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-783-2026

less, the inherent projection step makes performance sensitive to resolution, occlusion-induced gaps, and local surface irregularities, and the dimension reduction inevitably weakens the expressiveness of 3D joint geometry.

(2) Deep learning-based semantic segmentation methods (Huang et al., 2024, Li et al., 2023, Ji et al., 2023, Zhou et al., 2023). These methods employ supervised or semi-supervised point cloud networks (e.g., PointNet, STS-D, GL-Net) to learn semantic representations of tunnel elements from annotated data and directly output segment-joint results in an end-to-end segmentation manner. Recent progress has been driven by multi-scale feature modeling and attention-based aggregation to better capture contextual cues under varying point densities (Zhou et al., 2023; Huang et al., 2024), by combining global feature learning with locally discriminative refinement to improve boundary awareness and structural continuity (Li et al., 2023), and by introducing semi-supervised strategies to reduce the cost of dense manual annotation in large-scale tunnel scenes (Ji et al., 2023). In addition, the release of large-scale benchmarks and datasets has promoted more reliable training and evaluation for subway tunnel point cloud segmentation (Cui et al., 2024), while emerging “no-training / foundation-model-inspired” attempts for tunnel component segmentation further suggest a promising direction for reducing data dependency (Ye et al., 2025). Despite these advances, segment joints remain typical “slender and small targets” with limited contribution to overall segmentation accuracy; they are therefore prone to being dominated by larger background categories during optimization, which often leads to insufficient learning for the joint class and makes it difficult to simultaneously achieve engineering-level completeness and centimeter-level localization accuracy.

(3) Local extremum-based methods (Zhi et al., 2025, Cui et al., 2024, Lin et al., 2023, Fu, 2023, Yi et al., 2025, Wang et al., 2025).These methods extract and locate joints by exploiting local mutation patterns around joint regions reflected in indicators such as laser intensity, point density, geometric distance, and curvature. Recent developments have attempted to improve robustness from multiple angles, including enhancing data quality and normalization (e.g., intensity correction to mitigate scanning-condition effects) (Fu, 2023) and proposing advanced filtering strategies tailored for complex tunnel point clouds (Zhi et al., 2025). Meanwhile, several studies incorporate domain priors—such as bolt-hole patterns or ring seam/ring number cues—to constrain joint candidates and reduce ambiguity (Lin et al., 2023; Yi et al., 2025), and more targeted pipelines have also been designed for extracting and locating segment joints from RMLS point clouds (Wang et al., 2025). However, many existing approaches still rely heavily on a single metric or fixed thresholds, resulting in limited sensitivity and accuracy under occlusion, noise contamination, and uneven point density, and making it difficult to stably distinguish true joints from joint-like interferences in real tunnel environments.

Despite the above progress, significant challenges still remain in complex tunnel environments, which restrict the practical deployment of existing methods in engineering applications: (i) For image processing-based pipelines, projection/unwrapping inevitably introduces 3D information loss and may amplify errors under occlusion-induced missing data and resolution-dependent artifacts, leading to degraded joint completeness and positioning accuracy. (ii) For deep learning-based segmentation, joint regions are small and slender, often suffering from class imbalance and insufficient feature learning, while strong dependence on annotated data and the risk of cross-tunnel domain shift hinder stable generalization and engineering-grade reliability. (iii) For local extremum-based approaches, handcrafted indicators are frequently parameter-sensitive and vulnerable to noise, density variation, and joint-like disturbances, and single-metric designs in particular tend to produce unstable responses in real RMLS scenarios.

To address these limitations, this paper proposes a shield tunnel segment joint recognition method based on a composite index, namely the Joint Composite Feature Index (JCFI). The method first removes obvious non-lining points via cross-sectional ellipse fitting, and then constructs JCFI by fusing multiple complementary geometric cues (curvature, left-right density ratio, and relative depth) to comprehensively characterize joint features. With the JCFI-driven sequential recognition of circumferential and longitudinal joints, the proposed method enhances robustness against noise, occlusion, and uneven point density, thereby directly addressing the key pain points of existing methods in terms of engineering practicality.

2. Methodology

The technical workflow of the proposed JCFI-based method for recognizing segment joints in shield tunnels is illustrated in Figure 1. First, the cross-section ellipse fitting method is employed to remove obvious non-lining points. Subsequently, JCFI is formulated by integrating curvature, left-right density ratio, and a relative depth metrics, to quantify the feature differences of the joints. Finally, leveraging the constructed JCFI, the successive recognition of circumferential and longitudinal joints is accomplished.

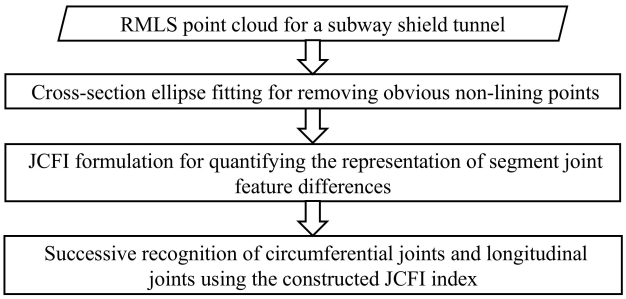

Figure 1. Flowchart of the proposed method.

2.1 Cross-section ellipse fitting for removing obvious non-lining points

The raw point cloud of the RMLS shield tunnel is designated as

$$P_{raw} = \{p_{s,i}(x_{s,i}, y_{s,i}, z_{s,i}, I_{s,i}) \mid s \in [1, S], i \in [1, N_0]\} \quad (1)$$

where s is the cross-section index, S is the total number of cross-sections, i is the index of a laser point within a cross-section, and N_0 is the total number of points in the raw point cloud. The tuple ($x_{s,i}$, $y_{s,i}$, $z_{s,i}$) represents the 3D coordinates of the laser point, and $I_{s,i}$ is its reflectance intensity. Since segment joints are located on the tunnel lining surface, the presence of noise and non-lining points from ancillary facilities (e.g., cables, pipes, and brackets) in the raw data can interfere with accurate joint identification. Therefore, the scheme based on cross-sectional ellipse fitting proposed by Wang et al. (2025) was employed to eliminate obvious non-lining points that de-

viate significantly from the fitted ellipse of the tunnel cross-section, resulting in a preprocessed lining point cloudP_{lining}.

$$P_{lining} = \{p_{s,j}(x_{s,j}, y_{s,j}, z_{s,j}, I_{s,j}) \mid s \in [1, S], j \in [1, N_1]\} \tag{2}$$

where j is the index of a laser point, andN_1 is the total number of points in the preprocessed lining point cloud.

2.2 JCFI formulation for quantifying the characterization of segment joint feature differences

Traditional recognition methods, typically dependent on a single metric, suffer from instability and inaccuracy due to their susceptibility to noise, missing data, and uneven point density. To address these limitations and to deliver a robust representation of joint characteristics, a composite index named JCFI (Equation (3)) is formulated by integrating three distinct metrics: curvature (c), left-right density ratio (q), and relative depth ($\bar{d}$).

$$JCFI_{p_{j,r}} = c_{p_{j,r}} \times q_{p_{j,r}} \times \bar{d}_{p_{j,r}} \tag{3}$$

where r is the search radius, the value of which is correlated with the physical width of the joint.

The multiplicative fusion is adopted for two main reasons. First, the three metrics describe complementary joint characteristics from different perspectives; therefore, a true joint point is expected to simultaneously exhibit “high curvature”, “balanced left-right density” (i.e., q close to 1), and “protrusion” (i.e., positive $\bar{d}$). Using multiplication acts as an implicit “AND” constraint: if any metric is close to zero, the composite index will be suppressed, which helps reduce false positives caused by single-feature noise. Second, compared with an additive form, the product is less sensitive to the absolute scale of each metric and avoids introducing extra weighting parameters that would require additional tuning across different tunnels.

A detailed explanation of each of the three metrics will be provided in the following subsection.

2.2.1 Curvature metric The surface of tunnel segments is continuous and smooth, whereas the joint regions between adjacent segments exhibit local convex protrusions, resulting in significant differences in local geometric sharpness. To quantify this disparity for effective discrimination between joints and segments, a curvature metric describing the degree of local geometric variation is postulated (Equation (4)).

$$c_{p_{j,r}} = \frac{1}{|N_{p_{j,r}}|} \sum_{k=1}^{|N_{p_{j,r}}|} \left\| \mathbf{n}_{p_j} - \mathbf{n}_{p_k} \right\| \tag{4}$$

where $N_{p_j,r}$ refers to the point set within a spherical neighborhood of radius r centered at point p_j, and $|N_{p_j,r}|$ is the number of points in the set;$p_k$is the k-th laser point in the set, k=1,2,…,$|N_{p_j,r}|$, with $n_{p_j}$and n_{p_k} being the normal vectors at points p_j and p_k, respectively. This curvature metric, computed from a fixed-radius spherical neighborhood, is exemplified in Figure 2(a). The comparative results of a traditional curvature metric based on a fixed number of neighboring points (The number of points inside the sphere is set to 15) are visualized in Figure 2(b), serving to highlight the superiority of the proposed curvature metric.

A comparative analysis of Figures 2(a) and (b) reveals that the proposed curvature metric, by employing a fixed neighborhood

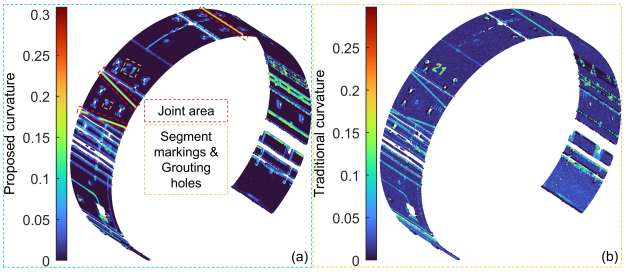

Figure 2. Calculation results for different curvature metrics. (a) The proposed curvature; (b) Traditional curvature.

radius r, ensures a consistent spatial scope for geometric characterization. Even in regions with uneven point density, it robustly captures geometric sharpness by adaptively varying the number of points within the neighborhood. In contrast, the traditional curvature calculation method with a fixed number of neighborhood points has significant limitations: in dense point cloud areas, the spatial range of the neighborhood becomes excessively small, while in sparse areas, the spatial range of the neighborhood becomes overly large. Both scenarios tend to cause drastic fluctuations in curvature calculation results. Consequently, the proposed metric demonstrates superior stability, particularly in scenarios with pronounced spatial variations in point density. However, as visible in the detailed view of Figure 2(a), this designed curvature metric, while effectively distinguishing segments from joints, exhibits residual noise. Specific non-joint regions (e.g., segment markings and grouting holes) produce high curvature values similar to true joints, underscoring the inherent limitation of relying on a single metric for achieving joint extraction.

2.2.2 Left-right density ratio metric As shown in Figure 3, the point cloud densities on both sides of a segment joint are relatively close.

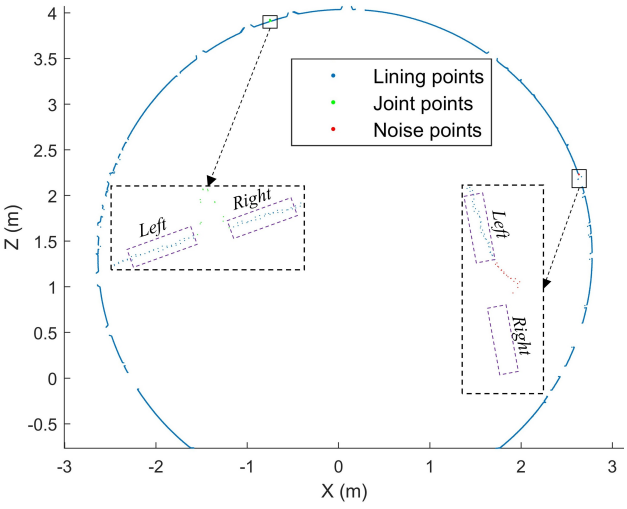

Figure 3. Comparison of the left-right point density differences between joint and noise points.

However, in noisy areas where the curvature features resemble those of joints, there is a significant difference in the point densities between their two sides. Therefore, a left-right density ratio metric (Equation (5)) is constructed to quantify this differential characteristic and to further improve the discrimination

between joints and non-joint areas.

$$q_{p_{j,r}} = \frac{\min(n_{p_{j,r}}^{s,Left}, n_{p_{j,r}}^{s,Right})}{\max(n_{p_{j,r}}^{s,Left}, n_{p_{j,r}}^{s,Right})} \tag{5}$$

where $n_{p_{j,r}}^{s,Left}$ and $n_{p_{j,r}}^{s,Right}$denote the number of laser points within the left-side and right-side neighborhoods, respectively, of point p_j on cross-section s within a search radius r.

2.2.3 Relative depth metric As shown in Figure 2(a), non-joint interferences such as segment markings and grouting holes share key characteristic commonalities with true joint regions: both exhibit high curvature and similar point density distributions on both sides. This makes it difficult to effectively distinguish them using only the curvature and left-right density ratio metrics, easily leading to misidentification. To break through this recognition bottleneck, a third core metric “relative depth” is designed. As illustrated in Figure 4(a), from the perspective of geometric morphology, joint points appear as local protrusions due to structural characteristics, whereas interference points like grouting holes manifest as local depressions. This creates a significant difference in the relative depth dimension. Leveraging this disparity, this paper constructs the relative depth metric (Equation (6)). By quantifying the “protrusion-depression” morphological difference, it achieves precise discrimination between joint and non-joint regions that both exhibit “high curvature and a similar left-right density ratio”, as illustrated in Figure 4(b).

$$\bar{d}_{p_{j,r}} = d_j - \frac{1}{|N_{p_{j,r}}|} \sum_{k=1}^{|N_{p_{j,r}}|} d_k \tag{6}$$

where d_j and d_k denote the depth values of points p_j and p_k, respectively, on their respective cross-sections.

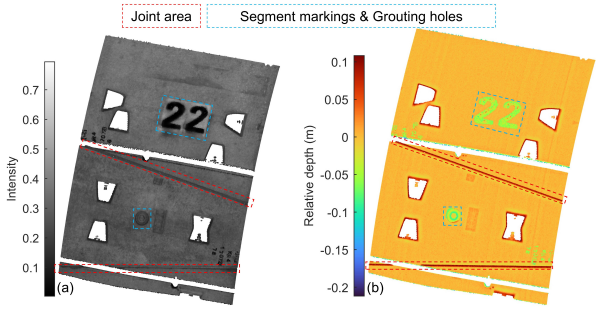

Figure 4. Protrusion-depression morphological difference. (a) The structural surface. (b) Quantification of the bulge-depression morphological differences based on relative depth.

2.3 Segment joints recognition based on the constructed JCFI

To address issues such as incomplete representation of joint features and susceptibility to interference in complex tunnel environments with traditional methods, this study proposes a segment joint recognition method based on the constructed JCFI. By quantifying the feature differences in curvature, left-right density ratio, and relative depth between segments and their joints, and combining adaptive threshold segmentation with clustering algorithms, this method sequentially achieves feature representation and instance segmentation of circumferential and longitudinal joints.

2.3.1 Circumferential joints recognition Significant geometric and spatial distribution differences exist between circumferential joints and segments. To accurately capture this disparity, this method uses the JCFI index defined in Section 2.2 to perform point-by-point feature calculation on the tunnel lining point cloud P_{lining}. Specifically, for any point $p_{s,i}$ in the P_{lining}, its JCFI value is calculated according to Equation (3), thereby constructing the circumferential joint feature set $JCFI\_Circumferential$. Among them, the search radius r in the feature calculation process is related to the physical width r_1 of the circumferential joint, i.e., $r = r_1$,a value that can be directly obtained from as built drawings or field measurement records of the tunnel. The effectiveness and robustness of this heuristic configuration will be further validated through a sensitivity analysis presented in the experiments section.

To intuitively demonstrate the ability of JCFI to distinguish between circumferential joints and segments, Figure 5(b) visualizes the JCFI values of a section of tunnel point cloud (corresponding to Figure 5(a)). The results show that the JCFI values of circumferential joint regions are significantly higher than those of segment regions. This difference can effectively represent the essential distinctions between joints and segments in terms of curvature, left-right density ratio, and relative depth, forming clear characteristic boundaries and confirming the validity and reliability of JCFI as a quantitative basis for identifying circumferential joints.

$$JCFI\_Circumferential = \{\text{JCFI}(p_{s,i}) \mid p_{s,i} \in P_{\text{lining}}\} \tag{7}$$

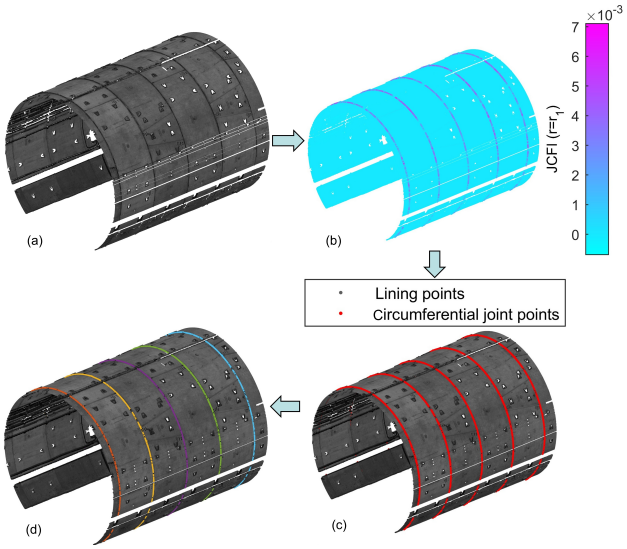

Figure 5. Circumferential joint recognition process based on the constructed JCFI. (a) Lining point cloud; (b) Calculated JCFI; (c) Detected circumferential joint points; (d) Instance segmentation of circumferential joints.

To extract circumferential joint points from the feature set $JCFI\_Circumferential$, an adaptive threshold segmentation scheme is designed to achieve binary segmentation between circumferential joint and non-circumferential joint. Specifically, the σ principle in statistics is used to set the segmentation threshold T:

$$T = \mu + 3\sigma \tag{8}$$

where μ is the mean value of the feature set $JCFI\_Circumferential$, and σ is its standard deviation.

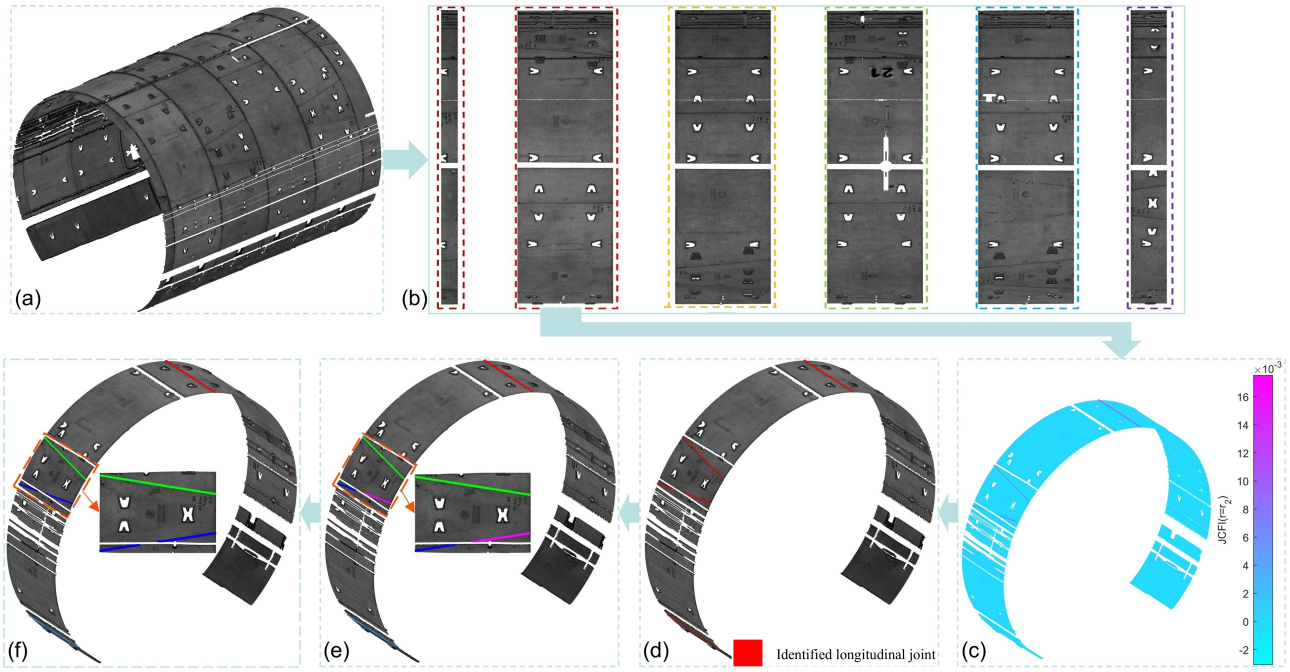

Figure 6. Longitudinal joint recognition process based on the constructed JCFI. (a) Lining point cloud; (b) Calculated JCFI; (c) Detected longitudinal joint points; (d) Instance segmentation of longitudinal joints; (e) DBSCAN clustering result. (f) Optimized instance segmentation of longitudinal joints, each in a unique color.

Points satisfying $JCFI\_Circumferential > T$ are identified as circumferential joint points, forming the circumferential joint point set. The recognition result is shown in Figure 5(c). It can be seen from Figure 5(c) that the circumferential joint points are successfully separated from the background point cloud. However, they do not yet form individual circumferential joints.

Considering that circumferential joints exhibit a band-like distribution in space with distinct gaps between different joints, the DBSCAN clustering algorithm is adopted for the instance segmentation of the joint point set. The segmentation result is depicted in Figure 5(d), where each cluster represents an individual circumferential joint. Its point set can be expressed as:

$$\{P_c \mid c = 1, 2, \ldots, C\} \tag{9}$$

where c is the circumferential joint index, C is the total number of circumferential joints, and P_c denotes the point set corresponding to the c-th circumferential joint. This completes the precise and automated identification of independent circumferential joint instances from the original tunnel point cloud.

2.3.2 Longitudinal joints recognition Longitudinal joints refer to the seams for longitudinal splicing of tunnel segments. The objective of longitudinal joint recognition is to achieve accurate instance segmentation of individual longitudinal joints within the same shield tunnel ring. Therefore, based on the results of circumferential joint recognition, this study establishes a longitudinal joint recognition technical workflow consisting of “shield ring division, JCFI feature construction and visualization, 3σ threshold segmentation, DBSCAN clustering, and post-processing optimization” to realize precise instance segmentation of longitudinal joints.

In accordance with the circumferential joint recognition method described in Section 2.3.1 above, circumferential joint detection is performed on the lining point cloud P_{lining} (Figure 6(a)), thereby dividing it into multiple independent shield tunnel rings (Figure 6(b)). Furthermore, each shield tunnel ring is treated as an independent processing unit, and schemes similar to circumferential joint recognition—including JCFI feature construction and visualization (Figure 6(c)), 3σ threshold segmentation (Figure 6(d)), and DBSCAN clustering (Figure 6(e))—are adopted to achieve instance segmentation of longitudinal joints. Similarly, the parameter of search radius r is set according to the physical width r_2 of the longitudinal joint (i.e., $r = r_2$), a parameter that can likewise be derived from engineering records or as-built survey data. This configuration is also subject to the same sensitivity analysis to verify its validity and robustness.

It is important to note that due to data missing caused by factors such as obstruction from auxiliary facilities, the over-segmentation phenomenon—where a single longitudinal joint is split into multiple instances—is common in the results of longitudinal joint instance segmentation, as shown in the locally enlarged view of Figure 6(e). To address this issue, this study proposes a post-processing method based on straight-line feature fusion. This method takes the initial segmentation results obtained from DBSCAN clustering as instances and uses the geometric property that longitudinal joints appear as approximate straight lines in the tunnel unfolding plane to merge and correct the initial instances. The specific implementation steps are as follows: (1) Acquisition of initial instance segmentation. The DBSCAN clustering algorithm is used to perform instance segmentation on the longitudinal joint point set, obtaining an initial instance segmentation sequence $C_{\text{initial}} = \{C_1, \ldots, C_w, \ldots, C_W\}$, where C_w represents the point set of the W-th initial segment instance, W is the total number of initial instances, and $W \geq L$ (where L is the true number of longitudinal joints). (2) Straight-line fitting. Based on the prior knowledge that “longitudinal joints appear as approximate straight lines in the unfolding plane”, C_w is mapped onto a two-dimensional plane (hereinafter referred to as the “arc-length vs. vertical axis” plane) with the arc length ($v = R{\cdot}\theta$, where θ is the

polar angle of the laser point in the polar coordinate system) as the horizontal axis and the longitudinal coordinate y as the vertical axis, forming a 2D point set $Q_w = \{(v, y)_1, (v, y)_2, \ldots\}$. The RANSAC line fitting algorithm is then used to fit a straight line to Q_w, yielding the straight-line equation $y = a_w \cdot v + b_w$ for each instance C_w, where a_w is the slope and b_w is the intercept. (3) Instance merging. Using the prior knowledge that “point sets belonging to the same longitudinal joint should be approximately distributed on a single straight line”, all pairs of initial segmented instances (C_w, C_g) (where $C_g \in C_{\text{initial}}$ and $g \neq w$) are traversed. If their corresponding straight-line parameters satisfy Equation (8), C_w and $C_g$are determined to belong to the same actual longitudinal joint and merged. This process is repeated until no more instances meet the merging condition. The recognized longitudinal joints are shown in Figure 6(f).

$$\text{Distance}((a_w, a_g), (b_w, b_g)) < \tau \tag{10}$$

To improve readability, the proposed JCFI-based segment joint recognition procedure is presented in concise pseudocode form in Algorithm 1.

Algorithm 1 JCFI-based segment joint recognition

Input: Raw tunnel point cloud P_{raw}; circumferential neighborhood radius r_1; longitudinal neighborhood radius r_2; threshold rule $T = \mu + 3\sigma$; DBSCAN parameters $(\varepsilon_1, \text{minPts}_1)$ and $(\varepsilon_2, \text{minPts}_2)$; merging threshold τ
Output: Circumferential joint instances $\{P_c\}_{c=1}^{C}$ and longitudinal joint instances $\{P_\ell\}_{\ell=1}^{L}$
$P_{\text{lining}} \leftarrow$ EllipseFitFilter(P_{raw}) ▷ remove non-lining points
$\{P_c\}_{c=1}^{C} \leftarrow$ JointDetect$(P_{\text{lining}}, r_1, \varepsilon_1, \text{minPts}_1)$ ▷ circumferential joints
$\{\mathcal{R}_k\}_{k=1}^{K} \leftarrow$ DivideIntoRings$(P_{\text{lining}}, \{P_c\})$
$\mathcal{C}_{\text{init}} \leftarrow \emptyset$
for $k = 1$ **to** K **do**
    $\mathcal{C}_{\text{init}} \leftarrow \mathcal{C}_{\text{init}} \cup$ JointDetect$(\mathcal{R}_k, r_2, \varepsilon_2, \text{minPts}_2)$ ▷ longitudinal candidates
end for
$\{P_\ell\}_{\ell=1}^{L} \leftarrow$ LineFusionMerge$(\mathcal{C}_{\text{init}}, \tau)$ ▷ merge split instances
return $\{P_c\}_{c=1}^{C}, \{P_\ell\}_{\ell=1}^{L}$
function JointDetect$(P, r, \varepsilon, \text{minPts})$
    Compute JCFI $J(p; r)$ for each $p \in P$ using Eqs. (3)–(6)
    $\mu \leftarrow \text{mean}(J)$; $\sigma \leftarrow \text{std}(J)$; $T \leftarrow \mu + 3\sigma$
    $P^+ \leftarrow \{p \in P \mid J(p; r) > T\}$ ▷ candidate joint points
    $\{C_i\} \leftarrow \text{DBSCAN}(P^+, \varepsilon, \text{minPts})$ ▷ instance clustering
    return $\{C_i\}$
end function
function LineFusionMerge$(\mathcal{C}_{\text{init}}, \tau)$
    Fit a line for each instance in the unfolded plane via RANSAC (Eq. (10))
    Merge instances with line-parameter distance $< \tau$
    return merged instances $\{P_\ell\}$
end function

3. Experiments

3.1 Experimental data

To verify the effectiveness and engineering applicability of the proposed method, the RMLS point cloud data of the tunnel on Guangzhou Metro Line 8 was selected as the test data in this experiment. Collected from actual engineering scenarios, this data falls into the category of typical tunnel point cloud data, exhibiting strong scene representativeness and data validity.

As shown in Figure 7, this data covers a 54.83 m-long shield tunnel, including 19,738 cross-sections and 192,434,561 laser points. Its data density and spatial coverage are sufficient to meet the requirements of the fine-grained identification task for segment joints. From the analysis of tunnel structural parameters, this tunnel has an inner diameter of 5.4 m and is assembled from 38 shield rings, with a single ring width of approximately 1.5 m. The segments adopt the staggered assembly process, a mainstream technique in the engineering field; each shield ring consists of 6 segments, specifically including 1 top block, 2 adjacent blocks, and 3 standard blocks.

This data contains typical issues such as noise interference, occlusion by auxiliary facilities, incomplete point clouds of segment joints, and uneven distribution of point cloud density. These characteristics fully reflect the typical challenges encountered in segment joint recognition in actual tunnel environments, providing reliable data support for evaluating the robustness and generalizability of the proposed method in practical engineering scenarios.

To achieve quantitative evaluation of the recognition accuracy of the proposed method, a professional interactive segmentation tool was used in this study to perform fine-grained annotation of segment joints in the experimental point cloud. The generated Ground Truth serves as a benchmark for the subsequent quantitative comparison and analysis of method performance.

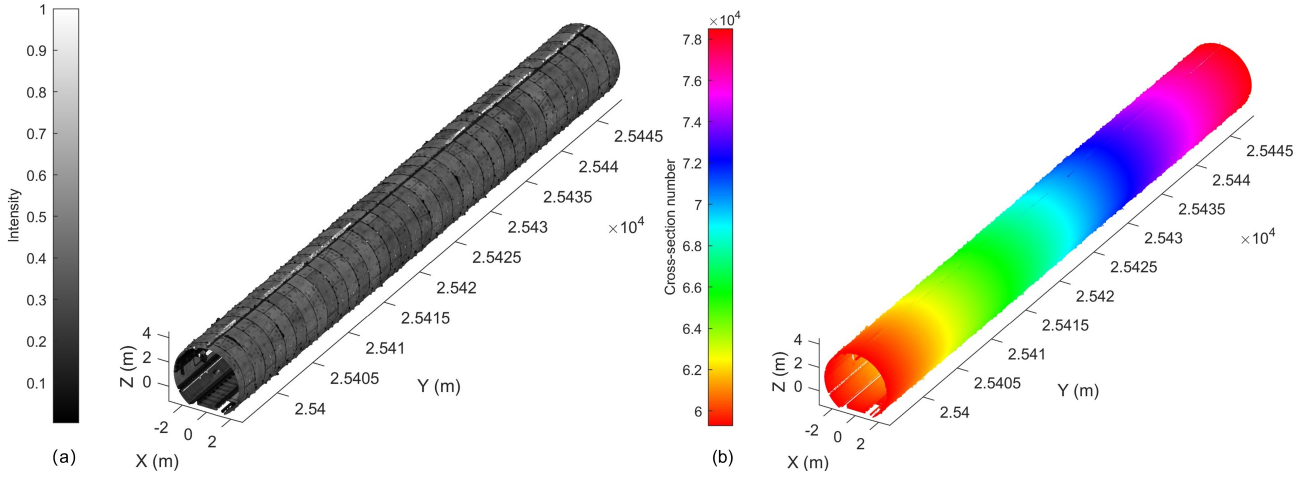

Figure 7. Experimental data. (a) The intensity rendering diagram; (b) Cross-sectional number rendering diagram.

3.2 Experimental validation

To reduce the impact of interfering factors such as noise, cables, and bolt holes on the subsequent recognition results of segment joints, the obvious non-lining point removal algorithm described in Section 2.1 was adopted to identify and eliminate laser points in the experimental data that deviate from the elliptical characteristics of the cross-section. The local details of the visualization results are shown in Figure 8.

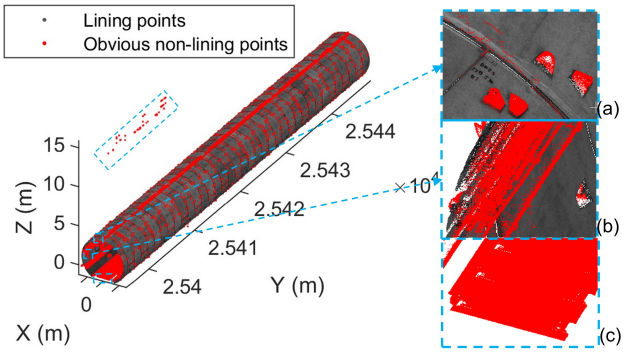

Figure 8. Obvious non-lining points removal results. (a) The detail of joints and bolt holes; (b) The detail of cables; (c) The detail of rail.

As can be seen from Figure 8, the algorithm can completely retain the lining and joint structures while eliminating interfering structures, which is specifically reflected in the following four details: The outliers in the upper-left area of the main graph in Figure 8 were accurately identified and eliminated; The locally enlarged view Figure 8(a) shows that both the outliers above the joints and the bolt hole points were effectively eliminated; The locally enlarged view Figure 8(b) shows that linear structures such as pipelines attached to the lining surface were successfully identified and eliminated; The locally enlarged view Figure 8(c) shows that the large structure at the bottom of the tunnel—the bottom rail—was completely identified and eliminated. In addition, statistics show that after the removal of obvious non-lining points, the number of experimental point clouds decreased from the original 192,434,561 to 106,865,852. A total of approximately 44.47% of obvious non-lining points were removed. The above results demonstrate that the proposed algorithm can not only efficiently eliminate various types of interfering points but also well preserve the key information of the tunnel lining main structure and joint areas, thus achieving an effective balance between denoising and shape preservation.

Based on the tunnel lining point cloud shown in Figure 8, and with reference to the JCFI index calculation method described in Section 2.2 and the JCFI-based segment joint recognition method outlined in Section 2.3, the circumferential and longitudinal joints of segments were automatically identified in sequence (Parameter settings based on the average in-situ measured joint widths: r_1=0.07 m, r_2=0.04 m). The recognition results are presented in Figure 9.

From the overall visualization results of Figure 9, the algorithm achieved complete recognition of all circumferential and longitudinal joints, with no redundant noise points or false joints in the recognition results. Meanwhile, the extracted widths of circumferential and longitudinal joints exhibited good consistency, with no local over-width or under-width phenomena. This fully demonstrates the algorithm’s high accuracy in representing the geometric morphology of joints.

Specifically, Figure 9(a) focuses on the details of circumferential joint extraction for a single shield ring. It can be observed that the circumferential joints exhibit excellent instance segmentation performance: each circumferential joint has clear boundaries and a definite attribution, and there is no over-merging issue where multiple circumferential joints are incorrectly clustered into a single instance. This verifies the algorithm’s accuracy in distinguishing individual circumferential joints.

In contrast, Figure 9(b) presents the local scene of a recognized longitudinal joint. Even though local point cloud missing occurred due to occlusions from auxiliary facilities in the tunnel (e.g., pipelines, brackets)—which further resulted in geometric discontinuities in the joint—the algorithm still did not incorrectly segment it into two independent instances. This performance is attributed to the post-processing strategy of straight-line feature fusion proposed in this study. This strategy leverages the near-straight geometric characteristics of longitudinal joints in the tunnel unfolding plane and merges longitudinal joint segments with similar straight-line slopes and intercepts, thereby ensuring the geometric continuity and correctness of the recognition results.

To verify the accuracy of the method proposed in this study, the Intersection over Union (IoU), recall, and precision metrics proposed by Li et al. (2023) were employed for the quantitative analysis of the method’s results, as detailed in Table 1.

Quantitative evaluation results presented in Table 1 demonstrate that the method proposed in this study exhibits excellent performance. Specifically, the recall rate, precision rate, and IoU

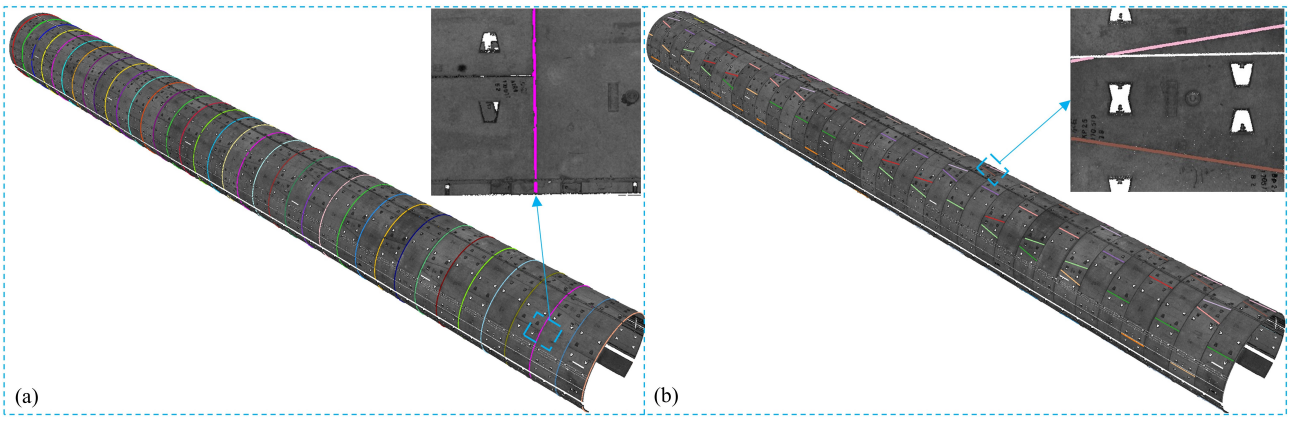

Figure 9. Segment joints recognition results: unique colors for individual instances. (a) Circumferential joints; (b) Longitudinal joints.

Joint type	Recall	Precision	IoU
Circumferential joints	85.59%	99.82%	85.46%
Longitudinal joints	95.77%	98.19%	94.11%
Segment joints	90.14%	99.04%	89.36%

Table 1. Segment joints recognition results.

of circumferential joints reach 85.59%, 99.82%, and 85.46%, respectively; for longitudinal joints, the corresponding metrics are 95.77%, 98.19%, and 94.11%, respectively; for segment joints, the corresponding metrics are 90.14%, 99.04%, and 89.36%, respectively. All evaluation indicators exceed 85%, which fully confirms the effectiveness and reliability of the proposed method.

To conduct an in-depth analysis of the key factors affecting the recognition accuracy and integrity of the proposed method, a visualized analysis of the segmentation error distribution was performed, with the results shown in Figure 10.

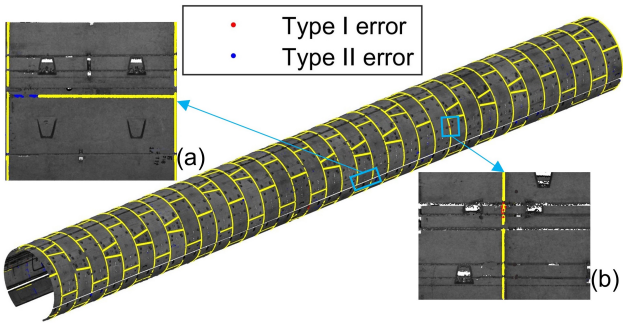

Figure 10. Error distribution in segment joint recognition. (a) Type II error (missed detection, false negative, FN); (b) Type I error (false detection, false positive, FP).

The following observations can be made from Figure 10: The point cloud at the boundary of longitudinal joints is affected by the shield segment dislocation, leading to a significant reduction in local point cloud density. This causes deviations in the calculation of the “left-right density ratio” measure within the JCFI index, resulting in the JCFI values of some points in the boundary area failing to meet the segmentation threshold and ultimately leading to incomplete boundary extraction (as shown in Figure 10(a)). However, such missing regions are confined to the extremely narrow area at the edge of longitudinal joints and do not affect the effective recognition of the main structure of longitudinal joints.

In high-interference areas (e.g., circumferential joint regions adjacent to cable brackets), a very small number of circumferential joints exhibit local data missing. This is attributed to two main reasons: first, the point cloud in this region has inherent data missing due to target occlusion; second, the high reflectivity of the brackets causes distortion in the calculation of the “relative depth” of the surrounding point cloud, which results in the JCFI values of some circumferential joint points being lower than the 3σ segmentation threshold and thus leads to local missing (as shown in Figure 10(b)). Given that the occlusion range of the brackets is limited and circumferential joints inherently feature a continuous annular distribution, such missing regions do not compromise the overall recognition integrity of circumferential joints.

To evaluate the influence of the neighborhood radius on joint identification accuracy, comparative experiments were conducted for both circumferential joints and longitudinal joints using

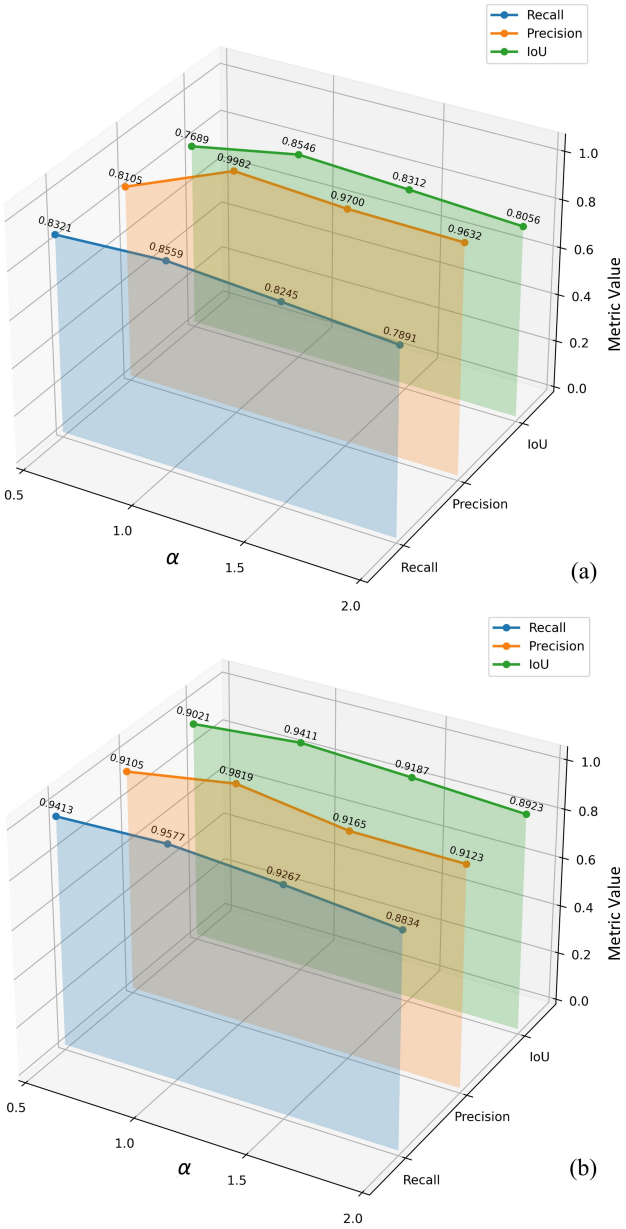

Figure 11. Effect of the neighborhood radius scaling factor α on joint identification performance (Recall, Precision, and IoU). (a) Circumferential joint; (b) Longitudinal joint. The neighborhood radii are set to $r_1 = \alpha \times 0.07$ m and $r_2 = \alpha \times 0.03$ m.

different neighborhood scales. Specifically, the neighborhood radius for circumferential joints was set as $r_1 = \alpha \times 0.07$ m, and that for longitudinal joints was set as $r_2 = \alpha \times 0.03$ m, where $\alpha = 0.5$, 1.0, 1.5, and 2.0. The corresponding accuracy evaluation results are shown in Figure 11, where the results for circumferential joints are presented in Figure 11(a) and those for longitudinal joints are presented in Figure 11(b).

When $\alpha = 0.5$ ($r_1 = 0.035$ m, $r_2 = 0.015$ m), the neighborhood is too small to provide sufficiently stable local statistics, causing unstable responses at some joint boundary points and leading to boundary discontinuities or missed detections; the IoU values are relatively low (76.89% for circumferential joints and 90.21% for longitudinal joints). As the radius increases to $\alpha = 1.0$ ($r_1 = 0.07$ m, $r_2 = 0.03$ m), the neighborhood scale better matches the geometric width of the joints, providing more reliable local statistics while preserving bound-

ary details, and the identified joint boundaries become clearer and more continuous. Under this setting, the performance is optimal, with IoU increasing to 85.46% for circumferential joints and 94.11% for longitudinal joints. When α is further increased to 1.5 and 2.0, the neighborhood statistics are more likely to be influenced by points outside the joint regions, which can cause slight overexpansion of the detected joint width and blurred boundaries; consequently, IoU gradually decreases (for circumferential joints: 83.12% and 80.56%; for longitudinal joints: 91.87% and 89.23%).

Overall, these results indicate that the neighborhood radius has a significant effect on the identification accuracy of both circumferential and longitudinal joints. Setting $\alpha = 1.0$ (i.e., $r_1 = 0.07$ m and $r_2 = 0.03$ m) achieves a favorable balance between statistical stability and boundary preservation, yielding the best performance.

3.3 Comparative experiments

To verify the effectiveness and advantages of the proposed method, comparative experiments were conducted for segment joint point cloud segmentation in shield tunnels. We compared our method with three representative open source deep learning models: PointNet(Qi et al., 2017), Cylinder3D(Zhu et al., 2021), and MinkUNet(Choy et al., 2019). Quantitative results of segmentation accuracy are presented in Table 2, where our method achieves the highest IoU on both circumferential and longitudinal joints, as well as the best mIoU, recall, and precision. The inference time of each method is reported separately in Table 3. Although our approach requires slightly longer inference time (107 s) than the baselines, it yields substantial improvements in segmentation quality, particularly for longitudinal joints (+34.23% over PointNet).

Table 2. Quantitative comparison of segment joint segmentation accuracy

Method	Circumferential joint IoU	Longitudinal joint IoU	mIoU	Recall	Precision
PointNet	70.88%	64.67%	67.78%	76.69%	81.03%
Cylinder3D	77.52%	44.57%	61.05%	76.03%	76.53%
MinkUNet	79.31%	59.88%	69.59%	81.24%	85.02%
Ours	85.46%	94.11%	89.79%	88.14%	99.37%

Table 3. Comparison of inference time

Method	PointNet	Cylinder3D	MinkUNet	Ours
Inference time (s)	80	96	84	107

Furthermore, to provide an intuitive comparison, Figure 12 visualizes the segmentation results produced by different methods on the same tunnel section, highlighting the differences in boundary completeness and noise robustness.

As shown in Figure 12, PointNet performs poorly in recognizing circumferential joint points affected by interference and also struggles to identify staggered longitudinal joints. Cylinder3D shows weak performance in the lower part of the tunnel section, where both circumferential joint points and longitudinal joint points are frequently missed. MinkUNet achieves better overall segmentation; however, similar to the previous two methods, it still produces a small number of false positives and false negatives near broken joints and at circumferential and longitudinal joint intersections. In contrast, the proposed method produces clean and consistent results, and no obvious false positives or missed detections can be observed from the visualization.

These differences suggest that learning based baselines are more sensitive to local point cloud degradation, such as occlusion, nonuniform density, and interference. By relying on local geometric statistics, the proposed method remains stable near discontinuities and intersections, which helps preserve joint boundaries while suppressing noise.

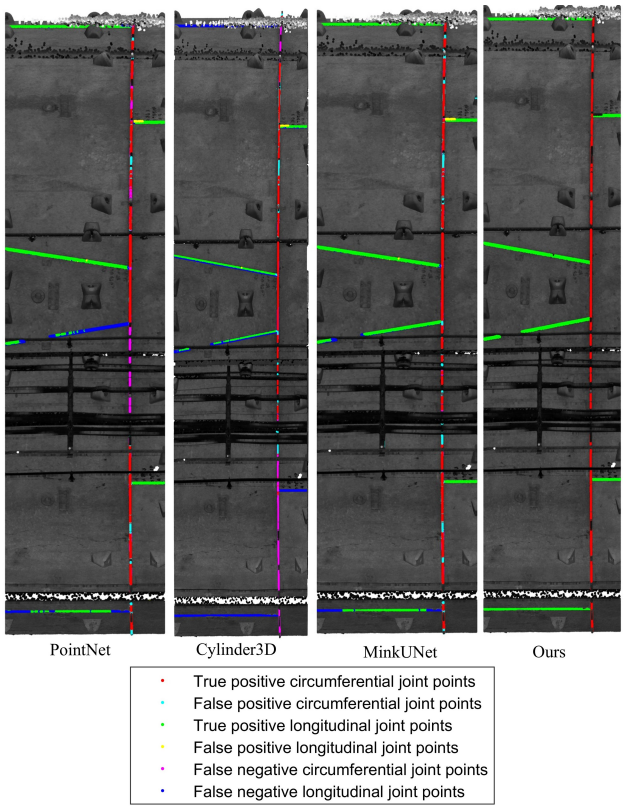

Figure 12. Qualitative comparison of segment joint segmentation results produced by different methods on the same tunnel section.

4. Conclusion

This study presents a method for segment joint recognition and instance segmentation in shield tunnels based on the constructed JCFI index.The JCFI is constructed by integrating three complementary measures: curvature, the left-right density ratio, and relative depth to detect joint points. On this basis, circumferential joints and longitudinal joints are further separated into instances using different clustering algorithms.

The proposed method is validated on in-situ RMLS measurements from Guangzhou Metro Line 8, demonstrating both feasibility and high accuracy. Quantitative evaluation shows that the circumferential-joint recognition achieves 85.59% recall, 99.82% precision, and 85.46% IoU, while the longitudinal-joint recognition achieves 95.77% recall, 98.19% precision, and 94.11% IoU, indicating reliable performance in realistic tunnel environments.

The main advantage comes from the composite design of JCFI: integrating multiple cues reduces sensitivity to any single feature and improves robustness under noise, occlusion, and uneven point density, because reliable evidence can still be recovered from the remaining cues to suppress false detections and maintain stable recognition. The overall pipeline is also template-free and adaptive. Although joint patterns exhibit certain regularities, their layouts vary across projects and seldom

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-783-2026 | © Author(s) 2026. CC BY 4.0 License.

match a fixed template, limiting the transferability of template-matching approaches without redesign. In addition, key parameters are determined directly from the data via statistical characteristics (e.g., a 3σ-style adaptive thresholding strategy), which reduces repeated manual tuning of neighborhood settings and thresholds and improves cross-scene reusability.

Future work will focus on extending the current approach to a wider range of tunnel types and joint geometries. In particular, general-purpose denoising and feature construction methods will be studied for non-protruding joints and non-circular tunnels (e.g., horseshoe-shaped, semi-circular, and rectangular tunnels).

References

Choy, C., Gwak, J., Savarese, S., 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3075–3084.

Cui, H., Li, J., Mao, Q., Hu, Q., Dong, C., Tao, Y., 2024. STSD:a large-scale benchmark for semantic segmentation of subway tunnel point cloud. *Tunnelling and Underground Space Technology*, 150, 105829. https://www.sciencedirect.com/science/article/abs/pii/S0886779824002475.

Feng, Y., Feng, S., Zhang, X., Zhang, D., Zhao, Y., 2024. A two-step deep learning-based framework for metro tunnel lining defect recognition. *Tunnelling and Underground Space Technology*, 150, 105832. https://www.sciencedirect.com/science/article/abs/pii/S0886779824002505.

Fu, J., 2023. Research on intensity correction and application of mobile lidar point cloud for shield tunnels. Master’s thesis, Jiangxi University of Science and Technology, Ganzhou, China.

Huang, X., Zhang, X., Li, Q., Zhou, B., 2024. A method for semantic segmentation of tunnel point clouds with multi-scale and attention aggregation. *Science of Surveying and Mapping*, 49(7), 56–65. https://doi.org/10.16251/j.cnki.1009-2307.2024.07.007.

Ji, A., Zhou, Y., Zhang, L., Tiong, R. L., Xue, X., 2023. Semi-supervised learning-based point cloud network for segmentation of 3D tunnel scenes. *Automation in Construction*, 146, 104668. https://doi.org/10.1016/j.autcon.2022.104668.

Li, J., Zhang, Z., Sun, H., Xie, S., Zou, J., Ji, C., Lu, Y., Ren, X., Wang, L., 2023. GL-Net: semantic segmentation for point clouds of shield tunnel via global feature learning and local feature discriminative aggregation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 199, 335-349. https://www.sciencedirect.com/science/article/abs/pii/S0924271623001004.

Li, X., Zhang, D., Hou, Y., 2022. Analysis of shield tunnel ground deformation characteristics and affecting factors in water-rich soft stratum: a case study on the section tunnel of tianjin metro Line 6. *Applied Sciences*, 12(12). https://www.mdpi.com/2076-3417/12/12/6208.

Li, Z., Ma, E., Lai, J., Su, X., 2024. Tunnel deformation prediction during construction: an explainable hybrid model considering temporal and static factors. *Computers & Structures*, 294, 107276. https://doi.org/10.1016/j.compstruc.2024.107276.

Lin, K., Shi, B., Yang, M., Wang, C., Shang, C., 2023. A disease localization method for shield tunnels based on laser point cloud for ring seam and ring number recognition. *Journal of Railway Science and Engineering*, 20(12), 4835–4847. https://doi.org/10.19713/j.cnki.43-1423/u.T20230082.

Liu, X. G., Chen, Y. Y., Liu, X., 2021. Laser scanning-based rapid detection of deformation of shield tunnel section. *Journal of Traffic and Transportation Engineering*, 21(2), 107–116. https://doi.org/10.19818/j.cnki.1671-1637.2021.02.009.

Niu, Y., Xie, Y., Zhang, H., Yue, X., Han, Y., Hu, Z., 2022. Study on deformation characteristics of segment joints of the immersed tunnel in Hong Kong-Zhuhai-Macau Bridge. *Advances in Materials Science and Engineering*, 2022(1), 7152132. https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/7152132.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 652–660.

Wang, L., You, Z., Feng, Y., Xie, C., Camara, M., 2025. Extraction and location of subway shield tunnel segment joints from RMLS point clouds. *International Journal of Digital Earth*, 18(1). https://doi.org/10.1080/17538947.2025.2528627.

Ye, Z., Lin, W., Faramarzi, A., Xie, X., Ninić, J., 2025. SAM4Tun: no-training model for tunnel lining point cloud component segmentation. *Tunnelling and Underground Space Technology*, 158, 106401. https://www.sciencedirect.com/science/article/pii/S0886779825000392.

Yi, L., Yin, X., Zhang, T., 2025. Segment joint recognition of shield tunnels based on bolt hole features. *Railway Standard Design*, 69(2), 111–118. https://doi.org/10.13238/j.issn.1004-2954.202407180003.

Yu, J. W., Liao, J. H., Chen, Z. P., Xiong, Y. G., Xiong, Z. M., 2021. A tunnel segment segmentation and section extraction based on point cloud model of cross-section datum plane. *Bulletin of Surveying and Mapping*, 159–162. https://doi.org/10.13474/j.cnki.11-2246.2021.326.

Zhi, Z., Chang, B., Li, Y., Du, Z., Zhao, Y., Cui, X., Ran, J., Li, A., Zhang, W., 2025. P-CSF: polar coordinate cloth simulation filtering algorithm for multi-type tunnel point clouds. *Tunnelling and Underground Space Technology*, 155, 106144. https://www.sciencedirect.com/science/article/abs/pii/S0886779824005625.

Zhou, Y., Ji, A., Zhang, L., Xue, X., 2023. Attention-enhanced sampling point cloud network (ASP-CNet) for efficient 3D tunnel semantic segmentation. *Automation in Construction*, 146, 104667. https://www.sciencedirect.com/science/article/abs/pii/S0926580522005374.

Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Li, Z., 2021. Cylinder3D: An Effective 3D Framework for Driving-Scene LiDAR Semantic Segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4470–4479.