

STAR-Edge: Structure-aware Local Spherical Curve Representation for Thin-walled Edge Extraction from Unstructured Point Clouds

Zikuan Li^{*}, Honghua Chen^{*}, Yuecheng Wang, Sibow Wu, Mingqiang Wei, Jun Wang[†]

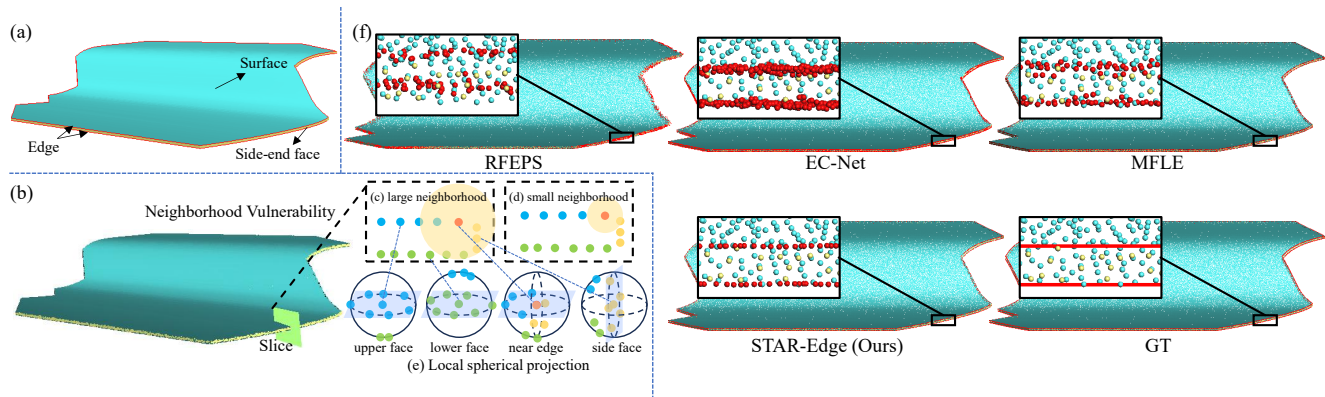


Figure 1. The characteristics of our method and visual comparison of edge extraction results in thin-walled structures. (a) defines the concept of a thin-walled edge. (b) highlights that the primary challenge in thin-walled edge extraction is the sensitivity to local neighborhood selection. (c) illustrates how a larger neighborhood may include points from both upper and lower surfaces as well as side-end faces, blurring boundary information. (d) shows that a smaller neighborhood may suffer from noise and insufficient sampling on side faces, lacking the contextual information needed for accurate edge point recognition. (e) depicts the spherical projection of the local neighborhood of four points, where points of the same color lie on the same underlying surface. This illustrates our key observation that points co-planar with the neighborhood center tend to align along a great circle arc after spherical projection. (f) provides a qualitative comparison of our STAR-Edge with RFEPS [29], EC-Net [32], and MFLE [4].

Abstract

Extracting geometric edges from unstructured point clouds remains a significant challenge, particularly in thin-walled structures that are commonly found in everyday objects. Traditional geometric methods and recent learning-based approaches frequently struggle with these structures, as both rely heavily on sufficient contextual information from local point neighborhoods. However, 3D measurement data of thin-walled structures often lack the accurate, dense, and regular neighborhood sampling required for reliable edge extraction, resulting in degraded performance.

In this work, we introduce STAR-Edge, a novel approach designed for detecting and refining edge points in thin-walled structures. Our method leverages a unique representation—the local spherical curve—to create structure-aware neighborhoods that emphasize co-planar points while reducing interference from close-by, non-co-planar surfaces. This representation is transformed into a rotation-invariant descriptor, which, combined with a lightweight

multi-layer perceptron, enables robust edge point classification even in the presence of noise and sparse or irregular sampling. Besides, we also use the local spherical curve representation to estimate more precise normals and introduce an optimization function to project initially identified edge points exactly on the true edges. Experiments conducted on the ABC dataset and thin-walled structure-specific datasets demonstrate that STAR-Edge outperforms existing edge detection methods, showcasing better robustness under various challenging conditions. The source code is available at <https://github.com/Miraclelzk/STAR-Edge>.

1. Introduction

Thin-walled structures are essential in our daily lives, seen in items like mechanical parts, furniture, and various manufactured objects. These structures often contain thin-walled edges, defined as boundary lines where surfaces intersect at their side-end faces, characterized by a minimal thickness relative to other dimensions (see Fig. 1 (a)). As geometric

^{*} Equal contribution [†] Corresponding author

edges provide an abstraction of complex 3D shapes, extracting edges from 3D measurements (e.g. unstructured point clouds) in these structures is crucial for downstream tasks.

Extracting geometric edges from point clouds has been a long-standing problem. Traditional techniques generally rely on local geometric properties, such as eigenvalues of the covariance matrix [2, 28], normals [27], curvature [12], or elaborately-designed descriptors [4]. While effective to some extent, these methods depend on first- or second-order derivatives estimated from local neighborhoods, making them particularly sensitive to point disturbances and sampling sparsity or irregularities, especially on thick side-end faces. More recently, some deep neural networks [1, 8, 13, 13, 18, 19, 32] have been proposed to enhance edge accuracy by leveraging deep features [26], neural edge representations [35], or novel point representations (e.g., Scale Space Matrix [18]). However, these methods still struggle with thin-walled structures (as seen in Fig. 1 (f)), as they rely on extracting useful information from *valid* local neighborhoods.

To be more specific, the local neighborhood of points around thin-walled edges is often unreliable. A larger neighborhood may include points from both upper and lower surfaces as well as side-end faces, blurring the boundary information (see Fig. 1 (c)). Conversely, a smaller neighborhood may suffer from noise and insufficient sampling on side-end faces, lacking the contextual information needed for accurate edge point recognition by those above networks (see Fig. 1 (d)). Although multi-scale schemes [26] can partially mitigate this issue, we argue that this approach is sub-optimal, as it requires hard-coding several fixed neighborhood scales.

Ideally, each point’s local representation should be *structure-aware*. This means it should reliably emphasize points on the same underlying surface (e.g., the same plane) while minimizing interference from points on different structures (e.g., another plane). Interestingly, we observe that points co-planar with the neighborhood center tend to align along a great circle arc (dashed circle on the blue plane in Fig. 1 (e)) after spherical projection, effectively reflecting the underlying surface of the thin-walled structure. Based on this observation, naturally, we can construct a local spherical curve to incorporate as many points on the great circle arc as possible, thereby enhancing points on the same underlying surface while reducing the influence of points from other surfaces.

To this end, we present STAR-Edge, a novel approach for detecting and refining edge points in thin-walled structures. Our method begins with a new representation, the local spherical curve, to explicitly build structure-aware local neighborhoods. This design enhances information aggregation from nearby co-planar points, even in noisy, sparse, or irregularly point distribution settings. Next, we construct

a rotation-invariant descriptor based on the local spherical curve and integrate it with a multi-layer perceptron (MLP) to accurately classify edge and non-edge points. Beyond edge classification, we also use the local spherical curve representation to estimate more precise point normals and introduce an optimization function to project points closer to the true edge. To validate our approach, we conduct a series of experiments on the well-known ABC dataset [17] and specialized datasets for thin-walled structures [6]. Extensive experimental results demonstrate that our approach outperforms existing edge detection methods. Our contributions are summarized as follows:

- We introduce STAR-Edge, a new method for detecting and refining edge points in thin-walled structures.
- We develop a new representation, the local spherical curve, to construct structure-aware local neighborhoods, enhancing information aggregation from nearby co-planar points, even under challenging conditions like noise, sparsity, or irregular point distribution.
- Using the local spherical curve representation, we construct a rotation-invariant local descriptor to achieve more accurate edge point classification. Additionally, this representation allows us to accurately calculate point cloud normals, enabling further refinement of edge points.
- We perform extensive experiments on specialized datasets for thin-walled structures and the ABC dataset, showing that our method outperforms existing edge extraction techniques.

2. Related Works

2.1. Traditional method for edge extraction

Early edge extraction techniques primarily targeted mesh models [15, 25], with subsequent advancements focusing on direct edge feature extraction from point clouds. Oztireli et al. [23] apply robust kernel regression combined with Moving Least Squares (MLS) to preserve fine details and sharp features in point-based surface representations. Daniels et al. [9] further advance this by introducing a robust method that identifies sharp features in point clouds by generating smooth edge-aligned curves through multi-step refinement and surface fitting. They both rely on iterative refinement processes to progressively extract and enhance sharp edge features in point cloud data.

Mérigot et al. [20] propose a robust method using Voronoi-based covariance matrices to robustly extract curvature, sharp features, and normals from point clouds. Bazazian et al. [2] present a fast method for sharp edge extraction from unorganized point clouds using eigenvalues of covariance matrices from k-nearest neighbors. Building on these approaches, other researchers have explored various techniques for identifying and preserving critical features in point clouds. For instance, Huang et al. [14] present an

edge-aware resampling method that computes reliable normals away from edges while effectively preserving sharp features. Ni et al. [22] propose AGPN, a method for detecting edges and tracing feature lines in 3D point clouds using neighborhood geometry analysis with RANSAC and hybrid region growing. Mineo et al. [21] present a boundary point detection algorithm with fast Fourier transform (FFT)-based filtering to improve edge reconstruction and reduce noise in tessellated surfaces from point clouds. Chen et al. [4] propose a two-phase algorithm for extracting line-type features from unstructured point clouds, using a statistical metric to detect feature points and an anisotropic contracting scheme to reconstruct feature lines against noise.

In summary, most traditional point cloud edge extraction methods rely heavily on neighborhood reliability, which includes dense sampling and minimal noise. However, in thin-walled structures, neighborhood information is often ambiguous, making conventional methods generally unsuitable for such structures.

2.2. Learning-based methods for edge extraction

With the emergence of learning techniques, many data-driven approaches have been proposed to improve the robustness of edge extraction and the preservation of sharp edges [5, 34]. EC-Net [32] is introduced as an edge-aware point set consolidation network designed to handle sparse, irregular, and noisy point clouds by both minimizing distances to 3D meshes and edges. PIE-NET [26] uses PointNet++ [24] as the backbone for classification and regression to extract edges from 3D point clouds as parametric curves, and has been extensively validated on synthetic data. CPE-Net [7] is designed in a multitask learning paradigm and uses a learnable weighted least squares fitting approach to accurately extract countersink edges under noisy conditions. Generally, all of these methods initially classify edge points using deep learning techniques, followed by edge generation through upsampling, clustering, or fitting smooth curves to capture edge structures in point clouds.

There are also some methods that focus on learning edge features directly from point clouds. Bode et al. [3] proposed BoundED, a neural network that uses local neighborhood statistics to classify non-edge, sharp-edge, and boundary points in 3D point clouds. Yao et al. [30] propose HGNet, a point cloud network that aggregates hierarchical features from points to super-surfaces. Zhao et al. [33] propose a deep learning approach to detect and consolidate sharp feature edges in 3D point clouds using a multi-task network that predicts displacement vectors for edge refinement. APEE-Net [6] is a dual-task neural network designed for extracting precise edges on aircraft panels by identifying nearby points and also using displacement vectors for repositioning. In addition, NerVE [35] defines a neural volumetric edge representation to extract parametric edge

curves through graph search, leveraging neural fields rather than traditional point cloud processing. Ye et al. [31] propose a Neural Edge Field method for reconstructing 3D feature curves from multi-view images, optimizing a neural implicit field with a rendering loss to extract 3D edges without 3D supervision. Although these learning-based methods achieve better performance than previous geometric-based methods, they still rely heavily on sufficient contextual information from local point neighborhoods, making them challenging to apply to thin-walled structures.

3. Method

The proposed method extracts edges from thin-walled structures, as illustrated in Fig. 2, through three main steps: local spherical curve construction, edge point identification, and edge optimization. First, for each point, we project its sparse neighborhood onto a unit sphere, constructing a closed spherical curve to represent the local neighborhood (Sec 3.1). Next, we use spherical harmonics (SH) to encode this spherical curve as a rotation-invariant local descriptor, which is then passed through an MLP layer to classify the target point as an edge or non-edge point (Sec 3.2). Finally, we leverage this descriptor again to estimate the normal vector, allowing us to formulate a projection function that refines the edge point positions (Sec 3.3).

3.1. Local Spherical Curve Construction

Our key observation is that neighboring points co-planar with a target point align along great circle arcs after spherical projection, allowing the overall shape to be approximated as a closed, continuous curve. Thus, given a point cloud $P = \{p_i \in \mathbb{R}^3 | i = 1, 2, \dots, N\}$, we first perform spherical normalization on each point p_i 's neighborhood point to obtain the local spherical point set Q_i , and then fit a closed, continuous curve.

Specifically, by taking p_i as the center of the sphere, we normalize the radius of its neighboring points to 1, projecting them onto the unit sphere. The normalization process is defined as follows:

$$Q_i = \{q_j = \frac{p_j - p_i}{\|p_j - p_i\|} \mid p_j \in KNN(p_i)\}, \quad (1)$$

where q_j represents the local spherical point and p_j is the KNN neighboring points of p_i . Next, we apply Principal Component Analysis (PCA) to Q_i , projecting it onto a local plane. We then compute the convex hull of the projected points and select the vertices of this convex hull as the key point set K_i for curve fitting:

$$K_i = \text{ConvexHull}(\text{PCA}(Q_i)). \quad (2)$$

Finally, we perform spline fitting on K_i to obtain a spherical curve, effectively representing the local neighborhood.

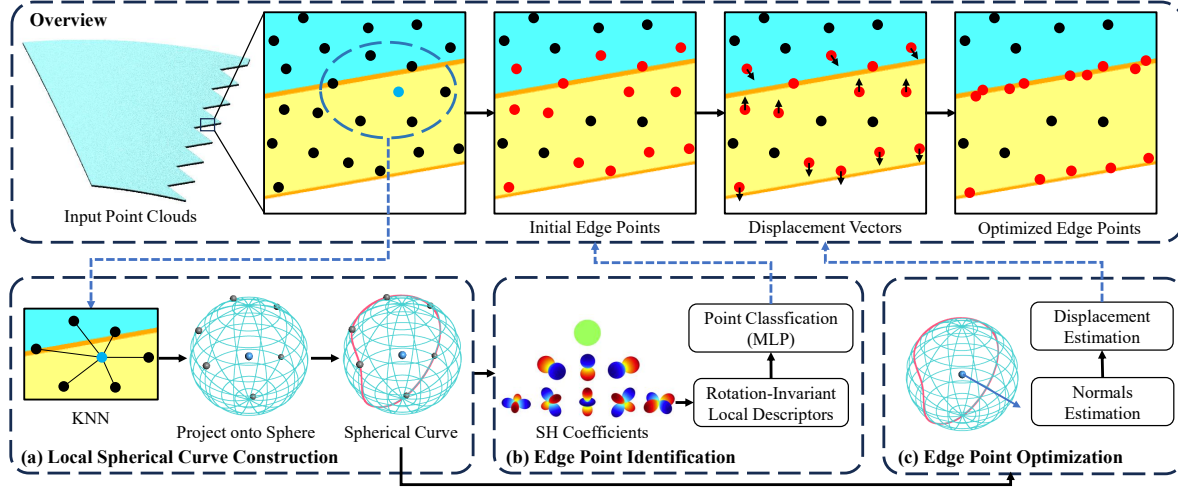


Figure 2. Overview of the proposed method. STAR-edge comprises three main steps: (a) constructing a local spherical curve; (b) identifying edge points by integrating rotation-invariant local descriptors with a light-weight MLP layer; and (c) optimizing edge points by adjusting their positions based on the estimated normal vector from the local spherical curve.

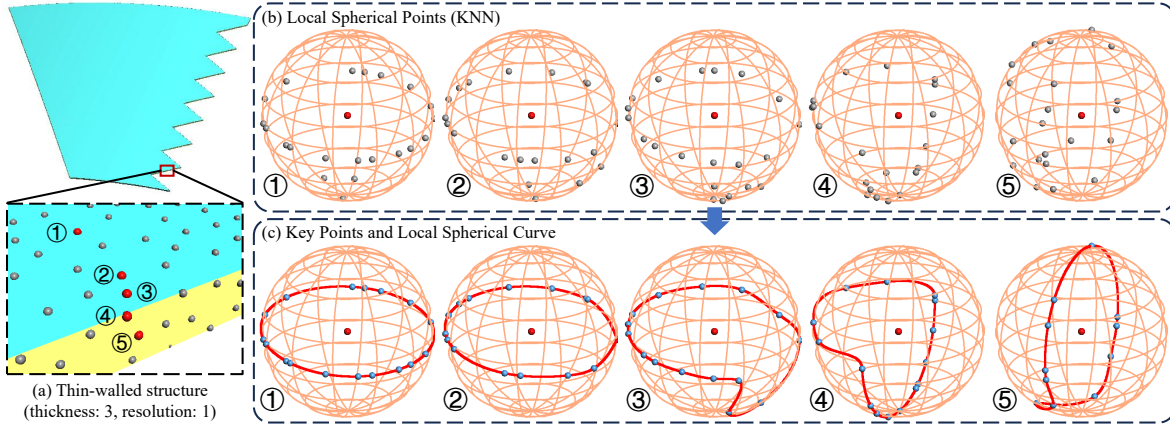


Figure 3. Neighborhood distribution of five surface points on the unit sphere. Local spherical points are shown in grey, sampled key points on the fitted spherical Curve in blue, and local spherical curves in red. We observe that the neighborhoods of edge points resemble two semicircles (e.g. ④), while those of non-edge points maintain a complete great circle (e.g. ①).

Fig. 3 shows the distribution of local spherical points (grey), key points (blue), and the local spherical curve (red) for five target points across a thin-walled structure. Points ① and ②, both located on the top surface, their neighborhoods are close to a complete great circle despite interference from points on other surfaces. Point ⑤, positioned on a side-end face with sparse sampling, is influenced by both upper and lower surfaces of the thin-walled structure; however, its local spherical curve still largely retains a great circle form. In contrast, points ③ and ④ are near a sharp edge, with ④ closer to the edge than ③. The local spherical curves of ③ and ④ are roughly formed by two semicircles, presenting a different pattern compared to points ① and ②. These results demonstrate that the proposed local spherical

curve can robustly and sensitively capture edge features.

3.2. Edge Point Identification

After obtaining the spherical curve for each point, we use SH, an effective time-frequency domain analysis technique on a sphere, to construct rotation-invariant descriptors. This SH-encoded curve serves as a rotation-invariant local descriptor, which is then fed into an MLP to predict whether each point is an edge point.

In detail, we discretize the local spherical curve into a set of scattered points. Each points (x, y, z) is mapped to spherical coordinates $f(\theta, \phi)$ on the unit sphere, where the mapping is defined as $(\theta, \phi) = (\arccos z, \arctan y/x)$. Here, θ and ϕ represent the zenith and azimuth angles, respec-

tively. Let $f_1, f_2, \dots, f_n$ be a series of mapping on spherical space. We employ kernel density estimation [10] to construct a smooth, continuous function based on these sample points:

$$f(\theta, \phi) = \frac{1}{nh} \sum_{i=1}^n k\left(\frac{\delta_i(\theta, \phi)}{h}\right), \quad (3)$$

where k is a kernel function, defined as a normal probability density function:

$$k(x) = 1/\sqrt{2\pi} \exp(-0.5x^2). \quad (4)$$

The term $\delta_i(\theta, \phi)$ denotes the geodesic distance between $f_i(\theta_i, \phi_i)$ and (θ, ϕ) . The parameter h , known as the window width or smoothing parameter, modulates the influence of mapping points f_i to the density estimation at (θ, ϕ) . It is defined as $h = \pi/B$, where B represents the spherical function bandwidth.

To compute the SH coefficients for the band-limited function, we apply the discrete SH transform. Following the Shannon-Nyquist sampling theorem [11], we sample the model equiangularly at twice the bandwidth B in both latitude and longitude, resulting in $2B \times 2B$ sampling points $(\theta_j, \phi_k) \in (0, \pi) \times [0, 2\pi)$, with

$$\begin{cases} \theta_j = \pi(2j+1)/4B & , j = 0, 1, \dots, 2B-1 \\ \phi_k = \pi k/B & , k = 0, 1, \dots, 2B-1 \end{cases} \quad (5)$$

The SH coefficients of the function $f(\theta, \phi)$ are obtained through the discrete SH transform:

$$\hat{f}_l^m = \frac{\sqrt{2\pi}}{2B} \sum_{j=0}^{2B-1} \sum_{k=0}^{2B-1} w_j f(\theta_j, \phi_k) e^{-im\phi_k} P_l^m(\cos \theta_j) \quad (6)$$

for degree $0 < l < B$ and order $|m| \leq l$. Here P_l^m denotes the associated Legendre functions defined on $[-1, 1]$. The weights w_j are computed using the method outlined in [11].

The SH bases are indexed by two integers, l and m , where $-l \leq m \leq l$, resulting in $2l+1$ coefficients for each degree l . The parameter l determines the frequency of the basis [16]. Consequently, the $2l+1$ coefficients corresponding to the same degree l are combined into a vector:

$$\beta_l = \{\hat{f}_l^m : m = -l, \dots, 0, \dots, l\}, \quad (7)$$

and the ℓ_2 norm $\|\beta_l\|$ is taken as the energy of the function f in degree l . At last, the rotation-invariant local descriptor D of function $f(\theta, \phi)$ with all B degrees is defined as

$$D(f, B) = \{\|\beta_l\| : l = 0, \dots, B-1\}. \quad (8)$$

After computing this descriptor, we calculate the average descriptor for ground-truth edge points and measure the feature difference between each point and this average descriptor. By applying a threshold to this difference, we can

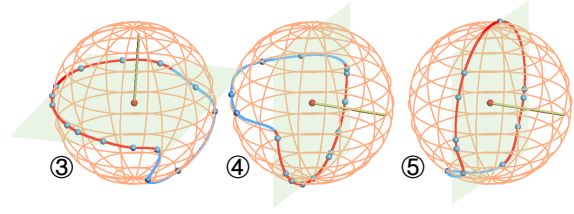


Figure 4. Normal estimation results for points ③, ④, and ⑤ in Fig. 3 (a). In each sub-figure, the red curve is the great circle arc extracted by RANSAC, the blue curve is the non-great circle segment, and the yellow arrow is the estimated normal vector.

determine whether a point is classified as an edge point. However, in our initial testing, we found that the threshold-based method struggles to effectively handle different edge styles. To address this, we use an MLP to adaptively learn from the rotation-invariant local descriptor, enabling accurate edge point identification. Detailed network structure is provided in the supplementary material.

3.3. Edge Point Optimization

We observe that the initially identified edge points are roughly scattered around the true edge. While these points are linearly distributed overall, they form a band-like structure cross two intersecting surfaces when viewed locally. To improve accuracy, we propose constraining edge points to surface intersection regions. This approach is motivated by the fact that an edge point at the intersection of two surfaces has a zero projection distance to both surfaces.

Let p_i represent an initially-identified edge point. Inspired by [29] and [6], we introduce the following optimization function:

$$\min_{z_i} \sum_{p_j \in KNN(p_i)} ((z_i - p_j) \cdot n_j)^2 + \mu \|z_i - p_i\|^2, \quad (9)$$

where n_j is the normal vector of p_j , and μ is a balance parameter. The first term encourages z_i to stay close to the tangent planes of neighboring points, while the second term penalizes the deviation of z_i from its initial position p_i . In all our experiments, μ is set to a default value of 0.1.

Normal estimation. Note that the edge point optimization method described above relies on accurate normal vectors; however, conventional normal estimation methods is unreliable in noisy or sparse neighborhood conditions. To resolve it, we propose a normal estimation algorithm based on the proposed local spherical curves, which can enhance robustness due to its underlying structure awareness.

As illustrated in Fig. 4, we observe that points co-planar with p_i (shown on the red curve) tend to align on the same plane as the great circle arc after projection onto a sphere. In contrast, other points (shown on the blue curve) lie on a different plane. Therefore, we use this great circle arc

(red) on the local spherical curve to approximate the local plane and derive the normal vector. Specifically, we employ random sampling consensus (RANSAC) to identify the great circle arc on the local spherical curve to determine the normal plane, and use the rest (blue) to determine the normal orientation. Fig. 4 shows the normal estimation of the neighborhood ③, ④, and ⑤ in Fig. 3 (a).

4. Experiments and Discussion

In this section, we evaluate the effectiveness of our method by comparing it to several learning-based approaches [26, 32, 35] and two traditional methods [4, 29]. We then assess its robustness under varying noise intensities and point cloud resolutions and conduct an ablation study to examine the impact of specific design choices.

4.1. Experiment Setup

Dataset. Given the characteristics of thin-walled structures (small dimensions in the thickness direction), we selected a dataset [6] specifically focused on thin-walled structures for our experiments. This dataset includes 42 distinct 3D models with curved edges of varying shapes, sizes, and curvatures, and thicknesses. For a more detailed description of this dataset, please refer to the supplemental file. Note that a subset of 32 shapes is used for training, while 10 shapes are reserved for testing. For each shape, we sample point clouds at resolutions of 0.3, 0.5, and 0.8. Additionally, three levels of random Gaussian noise are added to the point clouds at a resolution of 0.5. Each sampled point cloud contains approximately 20,000 to 15 million points.

We also perform experiments on the well-known ABC dataset [17], which contains over one million CAD models with edge annotations. To evaluate the generalizability of our method, we select sharp-edged CAD models from the ABC dataset, following the setup in [35]. This subset comprises 2,364 models, which are randomly divided into a training set (80%) and a test set (20%). For each shape, 50,000 points are sampled.

Implementation. All experiments are conducted on an Intel i7-11700 CPU and an NVIDIA RTX 4090 GPU. For constructing the local rotation-invariant descriptor, we use a *KNN* query with $k = 26$ and a spherical harmonics transform bandwidth of 10. The proposed approach is implemented in PyTorch and trained over 40 epochs using the Adam optimizer, with a batch size of 1024 and a learning rate of 0.02.

Metrics. We quantitatively evaluate all methods across two aspects. First, point classification performance is assessed using Recall, Precision, F1-score, and Accuracy. Second, to evaluate the quality of the final edge point extraction, we introduce a new metric: Edge Chamfer Distance (ECD). This metric measures the similarity between the optimized edges and the ground-truth edges. The introduction of ECD

is necessary because the ground-truth sampled point cloud consists of dense points, while the optimized edges may vary in sparsity depending on the chosen resolution. Standard Chamfer Distance, without adjustment, could disproportionately penalize sparse edges, leading to inaccurate similarity assessments. ECD addresses this limitation by providing a more reliable comparison for edge extraction quality. Let X represents the point cloud of the optimized edges, and Y denote the ground-truth edges. The ECD is defined as follows:

$$\text{ECD} = \frac{1}{|X|} \sum_{p \in X} \min_{q \in Y} \|p - q\|_2^2 \quad (10)$$

Note that for each shape, we label sampled points within a specified distance threshold from the nearest edge as ground-truth edge points. We set empirically this threshold to 0.3 for the dataset in [6] and 0.025 for the ABC dataset.

4.2. Evaluation on Thin-walled Dataset

In this section, we compare our STAR-Edge method with state-of-the-art approaches, including EC-Net [32], PIE-Net [26], NerVE [35], MFLE [4], and RFEPS [29], on the thin-walled dataset. All methods use point cloud data with a resolution of 0.5 as input. To accommodate network size limitations, PIE-Net is provided with 8,096 uniformly sampled points obtained through farthest point sampling, while NerVE converts the point cloud into a voxel grid with a resolution of 64^3 . All networks are retrained on the same dataset for a fair comparison.

Fig. 5 provides visual comparisons of all methods, while quantitative results are presented in Tab. 1. Both results demonstrate that our approach outperforms others on thin-walled structures, achieving better accuracy and a more uniform edge point distribution. Specifically, EC-Net, which up-samples point clouds in edge regions, tends to generate redundant points around true edges, leading to higher Edge Chamfer Distance (ECD). PIE-Net and NerVE perform less effectively on this dataset; PIE-Net requires dense sampling to ensure its effectiveness in edge extraction, while NerVE’s voxel size exceeds the thin-wall thickness, making it struggle to perceive edges. A potential improvement for these methods is to segment small patches for re-processing, as discussed in the supplemental file. The geometry-based methods, MFLE and RFEPS, also show limitations: MFLE requires frequent parameter adjustments, while RFEPS produces noisier edge results. Contrast, our method leverages the local spherical curve to achieve more precise edge points and displacement vectors, resulting in superior performance for edge detection in thin-walled structures.

Running time. The running times of various methods on point clouds containing approximately 320,000 points are as follows: our method required 214.0 seconds, MFLE completed in 1.14 seconds, RFEPS took 19 minutes, ECNet

Table 1. Quantitative comparison of edge point extraction on the thin-walled structure dataset. ‘-’ denotes a failed calculation.

	EC-Net [32]	PIE-Net [26]	MFLE [4]	RFEPS [29]	STAR-Edge (Ours)
ECD↓	0.2936	52.1204	0.4839	1.4392	0.0587
Recall↑	-	0.0503	0.9755	0.9997	0.9952
Precision↑	-	0.0295	0.3028	0.1512	0.5418
F1↑	-	0.0365	0.4621	0.2625	0.7013
Accuracy↑	-	0.9349	0.9470	0.9544	0.9931

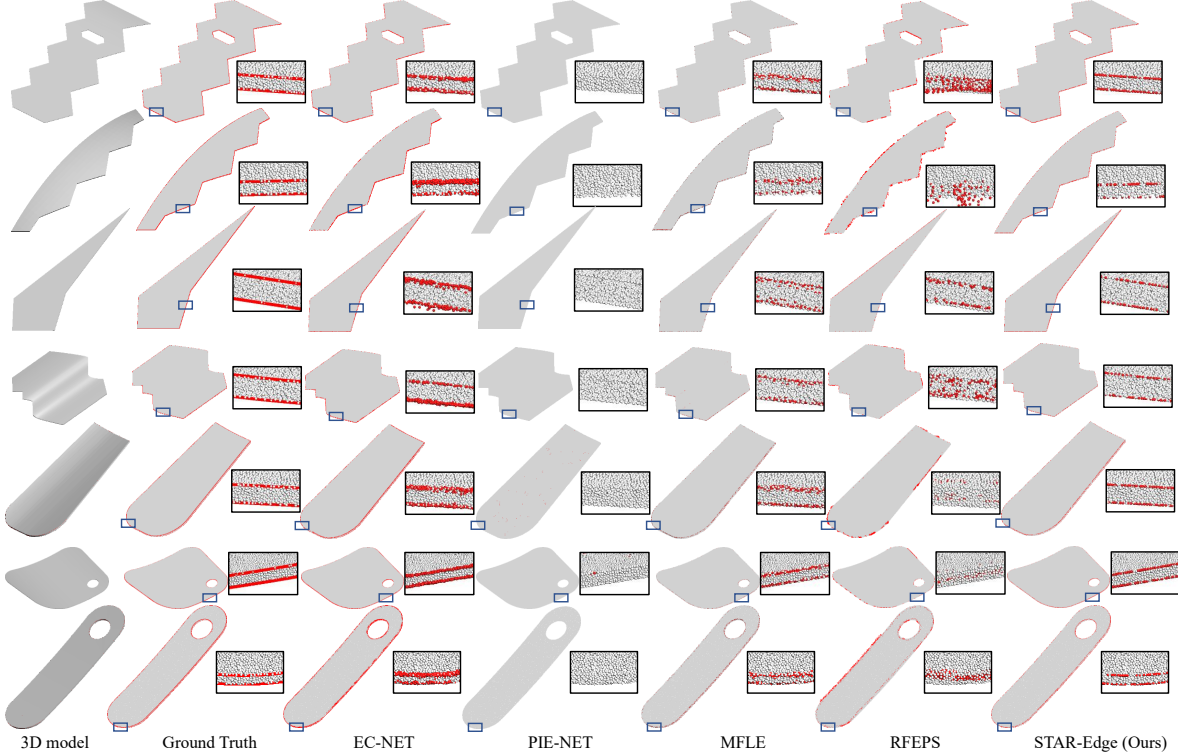


Figure 5. Visual comparison of state-of-the-art methods on the thin-walled structure dataset. In these close-up views, our method demonstrates superior accuracy in edge point extraction. Edge points are in red color.

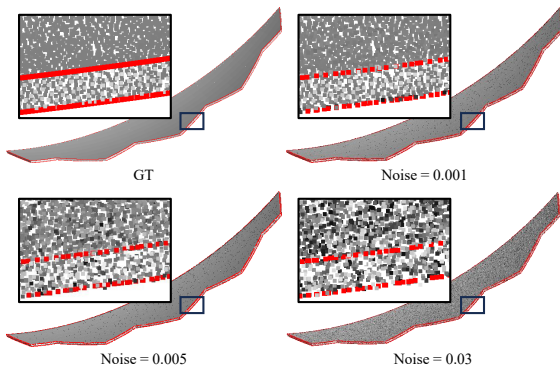


Figure 6. Visual comparison under different noise levels.

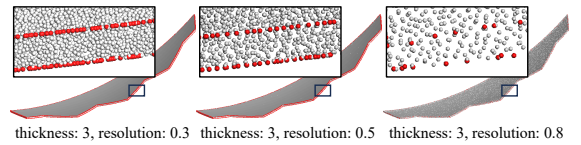


Figure 7. Visual comparison under different resolutions.

suming less than 1.0 second. We attribute PIE-Net’s exceptional speed to its down-sampling process. More detailed statistics can be found in the supplementary materials.

4.3. Robustness Study

We conducted robustness testing for the proposed STAR-Edge by introducing varying levels of noise and resolution. In our experiments, the thin-walled shapes have a thickness

finished in 83 seconds, and PIE-Net was the fastest, con-

Table 2. Quantitative classification results under different noise levels and resolutions.

	Noise			Resolution		
	0.001	0.005	0.03	0.3	0.5	0.8
ECD↓	0.059	0.059	0.199	0.048	0.059	0.261
Recall↑	0.995	0.995	0.994	0.960	0.995	0.988
Precision↑	0.542	0.541	0.523	0.784	0.542	0.582
F1↑	0.701	0.701	0.685	0.863	0.701	0.731
Accuracy↑	0.993	0.993	0.993	0.998	0.993	0.993

Table 3. Quantitative ECD comparisons under different noise and resolution. ‘-’ indicates that the calculation failed.

Methods	Noise			Resolution		
	0.001	0.005	0.03	0.3	0.5	0.8
EC-Net	0.295	0.293	0.294	-	0.294	0.518
PIE-Net	50.285	52.332	49.156	50.791	52.120	52.222
RFEPS	1.440	1.436	1.442	0.391	1.439	10.341
MFLE	0.484	0.453	0.409	0.406	0.484	0.742
Ours	0.059	0.059	0.199	0.048	0.059	0.260

of 2–3 (which is very small relative to other shape dimension). At a resolution of 0.3, approximately 9 points can be sampled on the side-end surface, whereas at a resolution of 0.8, it becomes challenging to obtain even 4 uniformly distributed points. Testing at a resolution of 0.8 poses a significant robustness challenge, as the limited number of sampling points reduces the reliability of projections onto spherical curves.

Tab. 2 and Tab. 3 present the effects of noise intensity and resolution on the predictions of our STAR-Edge method. We observe that, even under high noise intensity or significantly reduced resolution, our method consistently produces reasonable edge point predictions with low Edge Chamfer Distance (ECD). As shown in Fig. 6, when the noise level reaches 0.003, the detection performance remains largely unaffected. As shown in Fig. 7, demonstrates that even in an extreme case with a thickness of 3 and a resolution of 0.8, the upper and lower edges can still be clearly distinguished. Additionally, in Tab. 3, we compare the performance of STAR-Edge with state-of-the-art methods across inputs with varying noise levels and resolutions. Our method achieves the best numerical performance.

4.4. Ablation Study

Tab. 4 reports the performance of our method under varying SH bandwidths, which determine the precision and detail of the local feature representation. The results indicate that performance improves with increasing bandwidth; however, higher bandwidth settings also result in increased computational cost. Notably, the performance at a bandwidth of 10 is comparable to that at 15, making a bandwidth of 10 the optimal choice for balancing accuracy and efficiency.

Table 4. Ablation study about SH bandwidths.

Bandwidth	Recall↑	Precision↑	F1↑	Accuracy↑	ECD↓
5	0.988	0.523	0.685	0.991	0.122
10	0.995	0.542	0.701	0.993	0.059
15	0.995	0.583	0.785	0.998	0.048

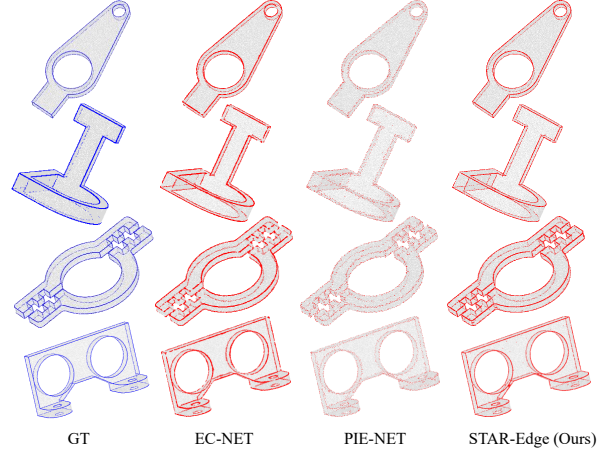


Figure 8. Visual comparison of edge extraction results on 3D shapes from the ABC dataset. Our method achieves more precise edge extraction, particularly in the sharp corner regions.

4.5. Evaluation on ABC dataset

We evaluate our method on the ABC dataset following the experimental setup in [35]. Quantitative results are presented in Tab. 5, which demonstrate that STAR-Edge consistently outperforms all baseline methods in terms of edge extraction accuracy. Visual comparisons in Fig. 8 further highlight the superior performance of our method, as it achieves more precise edge extraction, particularly in the sharp corner regions.

Table 5. Quantitative comparison on the ABC dataset.

	EC-NET	PIE-NET	MFLE	RFEPS	NerVE	Ours
ECD↓	0.0037	0.0090	0.0049	0.0026	0.0026	0.0024

5. Conclusion

In this work, we presented STAR-Edge, a novel point cloud edge extraction approach specifically designed for thin-walled structures. The core of our method is the introduction of a new structure-aware neighborhood representation—the local spherical curve. This representation effectively emphasizes co-planar points while minimizing interference from unrelated surfaces, enabling robust and accurate edge detection in complex thin-walled geometries.

6. Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.92367301, No.52425506, No.92267201, No.52275493, No.T2322012).

References

- [1] Dena Bazazian and M Eulàlia Parés. Edc-net: Edge detection capsule network for 3d point clouds. *Applied Sciences*, 11(4):1833, 2021. 2
- [2] Dena Bazazian, Josep R Casas, and Javier Ruiz-Hidalgo. Fast and robust edge extraction in unorganized point clouds. In *2015 international conference on digital image computing: techniques and applications (DICTA)*, pages 1–8. IEEE, 2015. 2
- [3] Lukas Bode, Michael Weinmann, and Reinhard Klein. Bounded: Neural boundary and edge detection in 3d point clouds via local neighborhood statistics. *ISPRS Journal of Photogrammetry and Remote Sensing*, 205:334–351, 2023. 3
- [4] Honghua Chen, Yaoran Huang, Qian Xie, Yuanpeng Liu, Yuan Zhang, Mingqiang Wei, and Jun Wang. Multiscale feature line extraction from raw point clouds based on local surface variation and anisotropic contraction. *IEEE Transactions on Automation Science and Engineering*, 19(2):1003–1016, 2021. 1, 2, 3, 6, 7
- [5] Honghua Chen, Zeyong Wei, Xianzhi Li, Yabin Xu, Mingqiang Wei, and Jun Wang. Repcd-net: Feature-aware recurrent point cloud denoising network. *International Journal of Computer Vision*, 130(3):615–629, 2022. 3
- [6] Mengqi Chen, Laishui Zhou, Honghua Chen, and Jun Wang. Thin-walled aircraft panel edge extraction from 3d measurement surfaces via feature-aware displacement learning. *IEEE Transactions on Instrumentation and Measurement*, 2024. 2, 3, 5, 6
- [7] Mengqi Chen, Laishui Zhou, Yongming Zhang, Honghua Chen, and Jun Wang. Aircraft skin countersink primitive extraction from 3-d measurement point clouds via deep clustering and fitting. *IEEE Transactions on Instrumentation and Measurement*, 2024. 3
- [8] Kseniya Cherenkova, Elona Dupont, Anis Kacem, Ilya Arzhannikov, Gleb Gusev, and Djamila Aouada. Sepicnet: Sharp edges recovery by parametric inference of curves in 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2727–2735, 2023. 2
- [9] Joel II Daniels, Linh K Ha, Tilo Ochotta, and Claudio T Silva. Robust smooth feature extraction from point clouds. In *IEEE International Conference on Shape Modeling and Applications 2007 (SMI'07)*, pages 123–136. IEEE, 2007. 2
- [10] Richard A Davis, Keh-Shin Lii, and Dimitris N Politis. Remarks on some nonparametric estimates of a density function. *Selected Works of Murray Rosenblatt*, pages 95–100, 2011. 5
- [11] James R Driscoll and Dennis M Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in applied mathematics*, 15(2):202–250, 1994. 5
- [12] Timo Hackel, Jan D Wegner, and Konrad Schindler. Contour detection in unstructured 3d point clouds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1610–1618, 2016. 2
- [13] Chems-Eddine Himeur, Thibault Lejemble, Thomas Pellegrini, Mathias Paulin, Loic Barthe, and Nicolas Mellado. Pcednet: A lightweight neural network for fast and interactive edge detection in 3d point clouds. *ACM Transactions on Graphics (TOG)*, 41(1):1–21, 2021. 2
- [14] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. Edge-aware point set resampling. *ACM transactions on graphics (TOG)*, 32(1):1–12, 2013. 2
- [15] Andreas Hubeli and Markus Gross. Multiresolution feature extraction for unstructured meshes. In *Proceedings Visualization, 2001. VIS'01.*, pages 287–294. IEEE, 2001. 2
- [16] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing*, pages 156–164, 2003. 5
- [17] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019. 2, 6
- [18] Yujia Liu, Stefano D'Aronco, Konrad Schindler, and Jan Dirk Wegner. Pc2wf: 3d wireframe reconstruction from raw point clouds. In *International Conference on Learning Representations*, 2021. 2
- [19] Albert Matveev, Ruslan Rakhimov, Alexey Artemov, Gleb Bobrovskikh, Vage Egiazarian, Emil Bogomolov, Daniele Panozzo, Denis Zorin, and Evgeny Burnaev. Def: Deep estimation of sharp geometric features in 3d shapes. *ACM Transactions on Graphics*, 41(4), 2022. 2
- [20] Quentin Mérigot, Maks Ovsjanikov, and Leonidas J Guibas. Voronoi-based curvature and feature estimation from point clouds. *IEEE Transactions on Visualization and Computer Graphics*, 17(6):743–756, 2010. 2
- [21] Carmelo Mineo, Stephen Gareth Pierce, and Rahul Summan. Novel algorithms for 3d surface point cloud boundary detection and edge reconstruction. *Journal of Computational Design and Engineering*, 6(1):81–91, 2019. 3
- [22] Huan Ni, Xiangguo Lin, Xiaogang Ning, and Jixian Zhang. Edge detection and feature line tracing in 3d-point clouds by analyzing geometric properties of neighborhoods. *Remote Sensing*, 8(9):710, 2016. 3
- [23] A Cengiz Öztireli, Gael Guennebaud, and Markus Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer graphics forum*, pages 493–501. Wiley Online Library, 2009. 2
- [24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3
- [25] Charlie CL Wang. Incremental reconstruction of sharp edges on mesh surfaces. *Computer-Aided Design*, 38(6):689–702, 2006. 2

- [26] Xiaogang Wang, Yuelang Xu, Kai Xu, Andrea Tagliasacchi, Bin Zhou, Ali Mahdavi-Amiri, and Hao Zhang. Pie-net: Parametric inference of point cloud edges. *Advances in neural information processing systems*, 33:20167–20178, 2020. [2](#), [3](#), [6](#), [7](#)
- [27] Christopher Weber, Stefanie Hahmann, and Hans Hagen. Sharp feature detection in point clouds. In *2010 shape modeling international conference*, pages 175–186. IEEE, 2010. [2](#)
- [28] Shaobo Xia and Ruisheng Wang. A fast edge extraction method for mobile lidar point clouds. *IEEE Geoscience and Remote Sensing Letters*, 14(8):1288–1292, 2017. [2](#)
- [29] Rui Xu, Zixiong Wang, Zhiyang Dou, Chen Zong, Shiqing Xin, Mingyan Jiang, Tao Ju, and Changhe Tu. Rfeps: Reconstructing feature-line equipped polygonal surface. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022. [1](#), [5](#), [6](#), [7](#)
- [30] Ting Yao, Yehao Li, Yingwei Pan, and Tao Mei. Hgnet: Learning hierarchical geometry from points, edges, and surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21846–21855, 2023. [3](#)
- [31] Yunfan Ye, Renjiao Yi, Zhirui Gao, Chenyang Zhu, Zhiping Cai, and Kai Xu. Nef: Neural edge fields for 3d parametric curve reconstruction from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8486–8495, 2023. [3](#)
- [32] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 386–402, 2018. [1](#), [2](#), [3](#), [6](#), [7](#)
- [33] Tong Zhao, Mulin Yu, Pierre Alliez, and Florent Lafarge. Sharp feature consolidation from raw 3d point clouds via displacement learning. *Computer Aided Geometric Design*, 103:102204, 2023. [3](#)
- [34] Haoran Zhou, Honghua Chen, Yidan Feng, Qiong Wang, Jing Qin, Haoran Xie, Fu Lee Wang, Mingqiang Wei, and Jun Wang. Geometry and learning co-supported normal estimation for unstructured point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13238–13247, 2020. [3](#)
- [35] Xiangyu Zhu, Dong Du, Weikai Chen, Zhiyou Zhao, Yinyu Nie, and Xiaoguang Han. Nerve: Neural volumetric edges for parametric curve extraction from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13601–13610, 2023. [2](#), [3](#), [6](#), [8](#)