

<https://doi.org/10.1038/s40494-025-01678-y>

# Automated generation of archeological line drawings from sculpture point cloud based on weighted centroid projection

Check for updates

Jian Li<sup>1,2</sup>, Juntao Yang<sup>1</sup>, Guohe Han<sup>2,3</sup>, Hao Wu<sup>4</sup>, Jin Zhu<sup>2,3</sup> & Hao Cui<sup>1,2</sup> ✉

Archeological line drawings are essential in archeological research, providing visual representations of artifact morphology and precise geometric measurements. Traditionally, these drawings were meticulously created by hand, a process that is time-consuming and labor-intensive. To overcome these limitations, this paper presents an algorithm for extracting feature lines from sculpture point cloud, enabling the automated generation of archeological line drawings. The method introduces a weighted centroid-based geometric metric to identify surface feature points and classify them based on their concavity or convexity. An iterative refinement process is then applied to ensure that feature points align correctly along the feature lines, and boundary points are extracted using an angular criterion. Finally, an enhanced curve-growing algorithm connects the feature and boundary points, producing a 3D line drawing. Experiments conducted on various stone sculptures demonstrate the practicality and accuracy of the proposed approach, with results showing significant improvements over existing methods.

Large stone sculptures represent a unique category of cultural relics. Compared to other types of artifacts, they are larger in size, heavier in weight, more difficult to transport, and more susceptible to damage, which makes their preservation particularly challenging. In the study of ancient craftsmanship and culture, these sculptures hold significant value. Their monumental scale and intricate craftsmanship reflect the historical, cultural, and artistic characteristics of ancient societies. Throughout history, they have withstood natural disasters and human destruction, making those that have survived especially valuable. The preservation of these surviving sculptures must be prioritized to prevent further deterioration, underscoring the broader issue of stone sculpture conservation. Among these preservation efforts, the creation of archeological line drawings is a fundamental task in documenting and preserving cultural relics<sup>1</sup>. Such drawings are utilized in excavation reports, academic papers, monographs, textbooks, and exhibition charts, serving both archival and cultural dissemination purposes<sup>2</sup>. They play a crucial role in archeology.

Archeological line drawings incorporate cartographic principles into archeological research, utilizing relevant theories and methods to accurately depict and represent the original characteristics of artifacts, while also recording precise geometric data. These drawings not only directly reflect the size and shape of artifacts but also provide indirect insights into their materials and craftsmanship. Furthermore, as a display tool, line drawings

can emphasize specific areas of interest, highlighting features and evolutionary patterns of artifacts across different periods. They serve as valuable resources for the subsequent preservation and restoration of cultural relics, making them an essential tool in archeological research.

Before the advent of advanced photographic technology, archeological line drawings were primarily created manually by specialists. The process involved establishing baselines, setting grids, and selecting measurement points using rulers, followed by rendering the drawings based on the collected data. This method was complex, required significant expertise from the draftsmen, and was relatively inefficient. The concept and practice of stereoscopic photogrammetry date back to the late 19th and early 20th centuries, with its initial applications primarily focused on topographic surveying<sup>3,4</sup>. As digital photography and computer-aided drawing technologies advanced, this technique increasingly found use in documenting and analyzing complex objects such as architecture, sculpture, and archeological artifacts. Early applications were largely centered on architectural surveying, including the documentation of cathedrals and bridges, which required detailed line drawings to aid in design and restoration. Rosenfeld et al.<sup>5</sup> described simple sets of parallel operations that can be used to detect texture edges. Canny<sup>6</sup> developed an edge detector designed for arbitrary edge contours and expanded it with operators of varying widths to address different signal-to-noise ratios within images. Aldsworth<sup>7</sup> employed an

<sup>1</sup>School of Geoscience and Technology, Zhengzhou University, Zhengzhou, China. <sup>2</sup>Archaeological Innovation Center, Zhengzhou University, Zhengzhou, China.

<sup>3</sup>School of Archaeology and Cultural Heritage, Zhengzhou University, Zhengzhou, China. <sup>4</sup>School of Urban and Environmental Sciences, Central China Normal University, Wuhan, China. ✉e-mail: [cuihao@zzu.edu.cn](mailto:cuihao@zzu.edu.cn)

integrated photogrammetry and CAD system to produce accurate, cost-effective, and efficient reconstruction drawings for historical buildings. Almagro<sup>8</sup> established a simplified method for architectural photogrammetry using semi-measuring cameras and analytical plotters, achieving higher efficiency in acquiring architectural drawings while maintaining sufficient accuracy. Streilein et al.<sup>9</sup> outlined the digital system for architectural photogrammetry under development at the Swiss Federal Institute of Technology, describing basic measurement methods for determining geometric primitives and emphasizing the need for interaction between digital photogrammetry systems and CAAD systems. Cooper<sup>10</sup> discussed the issues of excessive simplification in models of real-world objects and the idealization problems encountered when deriving line drawings from realistic images of complex objects. Grompone et al.<sup>11</sup> proposed a linear-time line segment detector that delivers accurate results with a controlled number of false detections and requires no parameter tuning. The algorithm was tested on a large set of natural images, yielding precise results. Akinlar et al.<sup>12</sup> proposed a linear-time line segment detector based on the edge segment chains produced by a novel edge detector, Edge Drawing (ED). It delivers accurate results without the need for parameter tuning, achieving this at exceptionally fast speeds. Dhankhar et al.<sup>13</sup> presented a review of various approaches for image segmentation based on edge detection techniques, comparing and analyzing methods such as Sobel, Prewitt, Roberts, Canny, and Laplacian Gaussian (LoG). Dolapsaki et al.<sup>14</sup> presented an effective, semi-automated method for detecting 3D edges in point clouds using high-resolution digital images. This work aimed to address the ongoing challenge of automating the generation of vector drawings from 3D point clouds of cultural heritage objects. Digital photogrammetry enables the rapid and cost-effective acquisition of an artifact's geometric and texture data<sup>15</sup>, which is then used to generate a 3D textured model and create orthophotos<sup>16,17</sup>. Orthophotos accurately reflect the true geometric shape and dimensions of the ground or objects, making them directly applicable for measuring and calculating spatial parameters such as the location, area, and length of features<sup>18</sup>. With the advancement of remote sensing technology, particularly the widespread use of drone imagery, orthophotos now provide high-resolution spatial data, offering significant advantages in representing intricate details. They can clearly capture subtle variations, complex shapes, and fine features of objects<sup>19,20</sup>. In archeology, orthophotos enable precise documentation of artifacts and archeological sites, serving as a reliable foundation for subsequent analysis, measurement, and research. Agrafiotis et al.<sup>21</sup> used conditional adversarial networks to generate accurate and realistic 2D drawings from orthoimages of Cultural Heritage masonry buildings. However, some minor flaws remain, particularly in areas with plaster, where distinguishing the edges of stones is challenging. By combining orthophotos with software such as Photoshop and AutoCAD, researchers can create high-precision archeological line drawings. However, for artifacts with complex geometric forms or 3D characteristics, the planar line drawings derived from orthophotos only represent surface details, failing to capture the complete 3D structure.

With the rapid advancement of 3D laser scanning technology, point cloud technology has become increasingly prevalent in the generation of archeological line drawings, yielding significant results<sup>22</sup>. By using high-precision laser scanning equipment, archeologists can comprehensively capture the 3D information of artifacts and obtain detailed point cloud data. Through the application of 3D point cloud feature extraction techniques and computer graphics processing technologies, various methods can be employed to extract feature lines from artifacts<sup>23</sup>. These methods can automatically identify key features, such as surface undulations, protrusions, and depressions, from point cloud, generating lines of archeological significance. The extraction of feature lines not only provides an abstract description of the artifact's surface morphology but also plays a crucial role in artifact research. First, feature lines clearly convey the geometric structure of the artifact, enabling an accurate restoration of its morphological characteristics and facilitating scholarly research and analysis. Second, feature lines can reveal details about the manufacturing techniques and stylistic features of artifacts, such as patterns and carving techniques on ancient

stone sculptures and bronze objects, offering valuable insights for historical and cultural studies. Finally, as a key component of archeological records, feature lines contribute to the digital preservation and display of artifacts. They can be used in virtual restoration and also provide a scientific basis for future restoration efforts. Scholars worldwide have conducted extensive research on feature recognition and extraction from 3D point cloud data, which can be broadly classified into two approaches: those based on mesh models and those based on unorganized point cloud.

In mesh-based model research, Hubeli et al.<sup>24</sup> proposed a framework for extracting mesh features from unstructured two-manifold surfaces, which includes a classification phase and a detection phase. This process is not fully automatic and still requires manual adjustments to achieve optimal performance. Decarlo et al.<sup>25</sup> introduced the concept of suggestive contours and provided two computational methods to combine actual contours with suggestive contours, depicting richer shape information. The generated results, however, heavily depend on geometric information such as curvature and normals. Ohtake et al.<sup>26</sup> proposed a method based on implicit surface fitting, where local mesh regions are fitted with implicit surfaces, and ridges and valleys are generated based on curvature extrema. Although accurate, the computational cost is high. Hildebrandt et al.<sup>27</sup> proposed a computational solution based on discrete differential geometry, using a filtering method for high-order surface derivatives to improve the stability and smoothness of feature line extraction, while avoiding the high computational cost of high-order surface fitting. Yoshizawa et al.<sup>28</sup> estimated curvature tensors and derivatives through local polynomial fitting to detect ridges, but manual filtering is required for complex geometric surfaces. Judd et al.<sup>29</sup> proposed the concept of apparent ridges, defined as the point trajectories that maximize view-dependent curvature. By considering the observer's perspective, this method more accurately conveys the visual features of 3D shapes, but it is not suitable for models with high noise or intricate details. Luo et al.<sup>30</sup> proposed a multi-scale method for detecting ridges and valleys in 3D models based on the minimum description length (MDL) principle. The resulting line drawings eliminate redundant lines, but the computational complexity is high. Lu et al.<sup>31</sup> studied various types of lines, such as occlusion contours, suggestive contours, ridges and valleys, and apparent ridges, analyzing their advantages in depicting different facial features. They generated optimized line drawings results for different facial models. Li et al.<sup>32</sup> combined the skeletal information of 3D models with feature lines, using line optimization to render the final drawings. They applied an enhancement-based method that integrates visual perception to create artistic effects, but this approach is not suitable for large mesh models. Wang et al.<sup>33</sup> proposed an algorithm for generating archeological line drawings based on explicit ridges. It performed well in line drawings of artifacts like the Terracotta Warriors, but neglected some finer details of the artifacts. Uchida et al.<sup>34</sup> used a fully convolutional neural network to extract lines from 3D models and encoded them into line attribute images, where line thickness was determined by line strength. The resulting stylized line drawings depended on the quantity and type of training data. Liu et al.<sup>35</sup> introduced a model for generating stylized line drawings from 3D shapes. The model takes 3D shapes and viewpoints as inputs and produces drawings with textured strokes. However, when there is a significant mismatch between the input geometric curves and the training drawings, the network may fail to accurately reproduce the stroke thickness and displacement. Ren et al.<sup>36</sup> proposed a line drawing extraction method for traditional Chinese ornaments, automatically generating pattern elements by computing 3D model line drawings. Jin et al.<sup>37</sup> proposed a semi-automatic framework that can extract expressive strokes from 3D models and render them in the style of traditional Eastern ink painting using a robotic arm. The framework is capable of producing visually pleasing ink paintings with expressive brushstrokes. While the framework is not fully automated, as the selection of expressive contours depends on individual users. Tang et al.<sup>38</sup> designed a framework for optimizing the model structure to enhance the generation of suggestive contours, thereby enabling the extraction of these contours from any model. Nguyen et al.<sup>39</sup> proposed a method for the joint simplification of lines and regions that penalizes large changes in region structure while

maintaining region closure. This feature enables region stylization that is consistent with the contour curves and underlying 3D geometry.

In research on unorganized point cloud, Lee et al.<sup>40</sup> proposed an improved moving least squares technique, using the Euclidean minimum spanning tree and region-growing iteration to refine the point cloud and reconstruct smooth curves. This method is limited to relatively simple-shaped point cloud. Gumhold et al.<sup>41</sup> directly extracted feature points from point cloud through covariance analysis and used eigenvectors to classify these points into border points, crease points, and corner points. Pauly et al.<sup>42</sup> used a multi-scale classification operator to analyze features across multiple scales, with the size of the local neighborhood serving as a discrete scale parameter to gradually complete feature extraction. Demarsin et al.<sup>43</sup> employed a first-order segmentation approach to extract candidate feature points and processed them as graphs to recover distinct feature lines in the point cloud. Daniels et al.<sup>44</sup> proposed a multi-step feature extraction refinement method, using robust moving least squares to fit the surface of vertex neighborhoods, ultimately generating smooth and complete feature lines. However, the method was slow when processing large point cloud datasets. Lu et al.<sup>45</sup> introduced a robust method for extracting 3D lines from stereo point cloud by integrating 2D image information with 3D point cloud from stereo cameras. 2D lines are first extracted from the image in the stereo pair, followed by 3D line regression from the back-projected 3D point set of the images points in the detected 2D lines. Rodríguez et al.<sup>46</sup> described the interaction between users and 3D point clouds through a 2D computer display, providing detailed equations for both 3D-to-2D projection and 2D-to-3D back-projection. Briese et al.<sup>47</sup> presented a fully automatic procedure for determining feature lines from TLS data. This approach consists of two main steps: the automatic detection of feature line segments and the modeling of complete feature lines. Weber et al.<sup>48</sup> calculated Gaussian clustering in local neighborhoods to detect sharp features on sampled surfaces and improved the accuracy of feature points through iterative selection. Chen et al.<sup>49</sup> proposed a fast and reliable approach for detecting 3D line segment on the unorganized point cloud from multi-view stereo. The core idea is to discover weak matching of line segments by reprojecting 3D point to 2D image plane and infer 3D line segment by spatial constraints. Mérigot et al.<sup>50</sup> proposed a method for computing the curvature and normals of locally smooth surfaces, which they used for feature extraction, but it was sensitive to outliers at moderate distances from the point cloud. Park et al.<sup>51</sup> extended the tensor voting (TV) theory to extract clear features from unorganized point cloud, enabling the handling of point cloud containing noise and outliers. Pang et al.<sup>52</sup> developed a method to directly extract ridges and valleys from 3D fingerprint model point cloud, using moving least squares to fit local point cloud surfaces and calculate curvature and curvature tensors. Kim et al.<sup>53</sup> used moving least squares to approximate surfaces, estimating the local curvature and derivatives at target points. They employed Delaunay surface subdivision to compute neighborhood information and detect ridges and valleys. Altantsetseg et al.<sup>54</sup> used one-dimensional truncated Fourier series to detect feature points and refined potential feature points through curvature-weighted Laplacian smoothing. Lin et al.<sup>55</sup> presented a novel method that is capable of accurately extracting plane intersection line segments from large-scale raw scan points. The 3D line-support region, namely, a point set near a straight linear structure, is extracted simultaneously. Bazazian et al.<sup>56</sup> proposed a fast and accurate method for detecting sharp edge features by analyzing the eigenvalues of the covariance matrix computed from each point's k-nearest neighbors. Nie et al.<sup>57</sup> proposed a method for measuring local surface variation—Smooth Shrinkage Index (SSI)—which combines Laplacian smoothing and refinement techniques to accurately extract feature points. Yuan et al.<sup>58</sup> derived point cloud curvature through polynomial fitting, classified feature points based on mean curvature, and used the Euclidean minimum spanning tree to construct feature lines. Zhang et al.<sup>59</sup> introduced a statistical model based on the Poisson distribution to adaptively extract feature points from point cloud and calculated the threshold of local features using the natural properties of the model surface. Lin et al.<sup>60</sup> proposed a new framework for extracting line segments from large-scale point cloud by efficiently

segmenting the input into a collection of facets, which provide sufficient information to identify linear features within local planar regions, thus facilitating line segment extraction. Wang et al.<sup>61</sup> described the creation of a highly accurate building façade feature extraction method from 3D PCD with a focus on structural information. Their approach involves three major steps: image feature extraction, exploration of the mapping method between the image features and 3D PCD, and optimization of the initial 3D PCD facade features considering structural information. Ahmed et al.<sup>62</sup> proposed a novel edge and corner detection algorithm for unorganized point clouds. The method uses an adaptive density-based threshold to distinguish 3D edge points, clusters curvature vectors, and uses their geometric statistics to classify a point as a corner. Mitropoulou et al.<sup>63</sup> introduced an automated and efficient method for detecting planes and extracting their intersection edges from unorganized point clouds. Lu et al.<sup>64</sup> proposed a straightforward 3D line segment detection algorithm that combines point cloud segmentation with 2D line detection to extract 3D line segments from large-scale, unorganized point clouds. Chen et al.<sup>65</sup> proposed a novel method for generating and regularizing point cloud feature lines, which consists of two main steps: extraction of the outline points according to the property of vectors distribution and cluster, feature points are sorted according to the vector deflection angle and distance and they are fitted using the improved cubic b-spline curve fitting algorithm. Zhang et al.<sup>66</sup> pioneered the development of 3D guided multi-conditional GAN (3D-GMcGAN), a deep neural network-based contour extraction network for large-scale point cloud. Cosgrove et al.<sup>67</sup> highlighted a security vulnerability in CNN-based edge detection models, demonstrating that they are susceptible to adversarial examples, which can not only cause failures in edge detection but also transfer to other CNN models. Abdellali et al.<sup>68</sup> proposed a robust, learnable line detector and descriptor (L2D2) that enables the efficient extraction and matching of 2D lines using the angular distance between 128-dimensional unit descriptor vectors. Hu et al.<sup>69</sup> introduced a geometric feature enhanced line extraction method with Hierarchical Topological Optimization for extracting line segments from large-scale point cloud. Their approach effectively filters major spurious lines and extracts more complete line segments. Chen et al.<sup>70</sup> proposed a method based on the Plane Fitting Residual-Local Shape Variation (PFR-LSV) metric to detect potential feature points, using an anisotropic shrinkage scheme to approximate the feature points to the actual feature lines. Liu et al.<sup>71</sup> proposed a “structure-to-detail” feature-aware algorithm that accurately locates geometric features of different sizes and extracts boundary features through tensor analysis. Liu et al.<sup>72</sup> proposed a local centroid computation algorithm based on neighbor reweighting to identify geometric features in point cloud, simultaneously recognizing convex, concave, and boundary points, and optimized feature points through assimilation and dissimilation operators. Bode et al.<sup>73</sup> proposed a novel set of features describing the local neighborhood on a per-point basis via first and second order statistics as input for a simple and compact classification network to distinguish between non-edge, sharp-edge, and boundary points in the given data. Cao et al.<sup>74</sup> proposed a deep neural network, WireframeNet, for converting point cloud into wireframes. This network inputs a set of disordered points and outputs a complete wireframe structure. Zang et al.<sup>75</sup> developed a large-scale 3D point cloud contour extraction network (LCE-NET) to generate contours consistent with human perception of outdoor scenes. It is the first time that an end-to-end learning-based framework has been proposed for contour extraction on point cloud at this scale. Zong et al.<sup>76</sup> introduced an efficient method based on plane segmentation and projection, transforming 3D planar point cloud into 2D images through graphical projection to avoid point-based nearest neighbors' searching. Zhu et al.<sup>77</sup> proposed NerVE, a neural volumetric edge representation that directly detects structured edges for extracting parametric curves from point cloud, thereby overcoming the limitations of previous point-by-point methods. One limitation of this approach is that it may produce unintended connection points. Jiao et al.<sup>78</sup> proposed a Multi-scale Laplace Network (MSL-Net), a new deep-learning-based method based on an intrinsic neighbor shape descriptor, to detect sharp features from 3D point cloud. Yu et al.<sup>79</sup> presented a feature line extraction

algorithm for building roof point cloud based on the linear distribution characteristics of neighborhood points. The proposed algorithm could directly extract high-quality roof feature lines from point cloud for both single buildings and multiple buildings. Wu et al.<sup>80</sup> achieved accurate extraction of contours from 3D target point cloud by integrating high-resolution images and sparse point cloud data. This method significantly improves the precision and accuracy of contour extraction for large objects, offering relatively accurate completion for partially defective contours. Si et al.<sup>81</sup> proposed a novel variational feature component extraction method called PointFEA, which effectively represents the underlying features of local point cloud regions. Wu et al.<sup>82</sup> proposed a fold point extraction method based on histogram detection, which identifies folds through corner point features and extracts fold points by measuring the distance to the furthest bar in the histogram. Betsas et al.<sup>83</sup> proposed the first implementation of a 3D edge detection method that exploits four-channel images and compared the detected 3D edges with their corresponding edges in a textured 3D model. Makka et al.<sup>84</sup> proposed a method that automates 3D edge detection in point clouds by exploiting differences in the directions of normal vectors to detect finite edges. These edges are then pruned, grouped into edge segments, and fitted to identify 3D edges. Aggarwal et al.<sup>85</sup> proposed a 3D edge detection approach using unsupervised classification, which learned features from depth maps at three different scales through an encoder-decoder network. Edge-specific features were then extracted and clustered to classify each point as an edge or non-edge. Xin et al.<sup>86</sup> proposed a two-stage method for extracting high-quality line segments from large-scale point cloud, based on a weighted centroid displacement scheme and three geometric operators. This approach ensures the simplicity, accuracy, and completeness of the generated line segments.

Building on the aforementioned methods, this paper introduces a novel feature line extraction approach designed to automatically generate archeological line drawings from noisy, unorganized point cloud of various stone sculptures. First, a projection of the neighbor-weighted centroid in the normal direction is employed as a metric for feature detection. This metric effectively mitigates noise interference and accurately quantifies surface variation in the point cloud. Next, to construct smooth and complete feature lines, the feature points are refined by projecting them onto the true feature line and detecting boundary points, ensuring the integrity of the feature lines. Finally, an improved curve growth algorithm is utilized to connect the feature and boundary points, generating smooth, continuous feature lines that form the archeological line drawings. The method has been tested on a variety of stone sculptures' point cloud, yielding results that outperform existing approaches. The primary contributions of this paper are as follows:

(1) A surface variation metric based on weighted centroid projection is proposed, which integrates roughness weight and normal difference weight to calculate the weighted centroid of the target point. The geometric saliency

of the target point is quantified through the projection of the feature vector, composed of the target point and its weighted centroid, onto the normal direction.

(2) A separate refinement strategy for ridgeline and valley points is introduced, enhancing the distinction between ridges and valleys.

(3) An approach for feature line construction based on both distance and angle is proposed, wherein the distance and angular factors of neighboring points are simultaneously considered during the feature line construction process, thereby generating complete and smooth feature lines.

The structure of this paper is as follows: Section “Methods” overviews the proposed method and explains the specific principles of each process; Section “Result” presents the experimental data, results, and a comparative analysis with other methods; and finally, Section “Discussion” summarizes the characteristics of the proposed method and discusses future research directions.

## Methods

The automatic generation method for archeological line drawings proposed in this study consists of four primary steps:

### (1) Identification of Feature Points

Feature points are typically located in regions where surface changes are most pronounced. To quantify these changes, the centroid projection distance is employed as a metric for detecting feature points. Since the input point cloud may contain noise, some noise points could be mistakenly identified as feature points. To mitigate this, the centroid is weighted by a combination of surface roughness and normal difference, resulting in more robust feature point detection.

### (2) Refinement of Feature Points

The detected feature points are generally distributed near the true feature line, forming a band-like pattern of a certain width. To ensure proper connectivity, these feature points must be refined to align linearly along the true feature line. In this study, principal component analysis (PCA) is utilized to calculate the principal direction of the feature points, projecting them onto the principal direction line for refinement.

### (3) Detection of Boundary Points

Boundary point detection is essential for constructing a complete feature line for the point cloud model. This study adopts the angle criterion method, which accurately extracts boundary points by analyzing the angular relationships between the target point and its neighboring points.

### (4) Construction of Feature Lines

After obtaining all feature points and boundary points, voxel downsampling is performed to generate a sparse and uniformly distributed point set. Subsequently, an improved curve growth algorithm is applied to construct smooth and complete feature lines.

The overall methodological workflow is illustrated in Fig. 1.

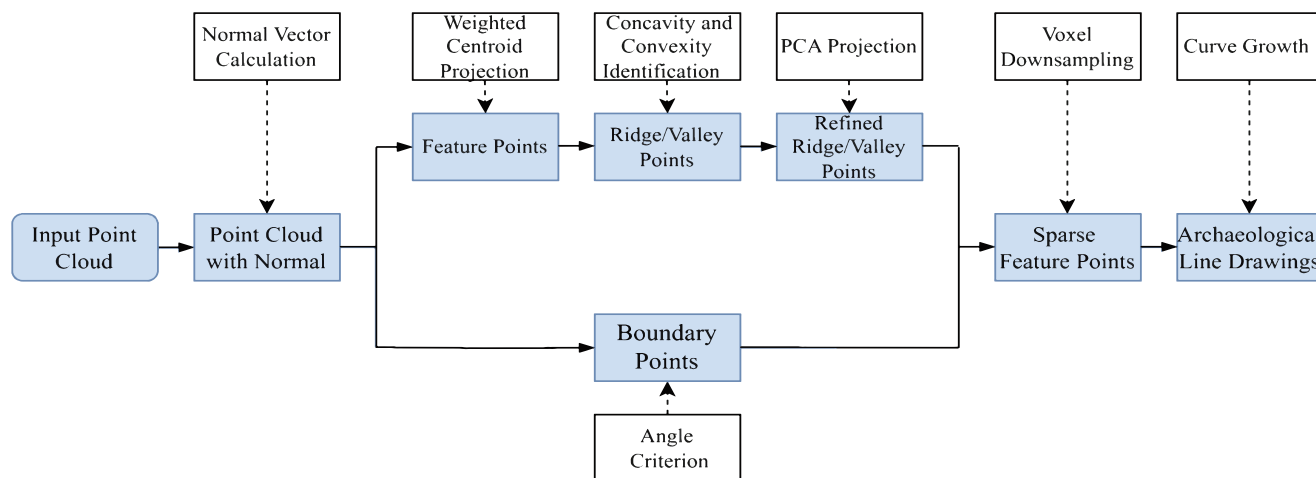


Fig. 1 | Algorithm Flowchart.

**Feature point detection**

(1) Calculation and Reorientation of Normal Vectors

Given an initial point cloud  $P = \{p_i\}_{i=1}^N$ , assume that  $P$  is an unorganized and unstructured point cloud, where  $N$  represents the number of points. For a target point  $p_i$ , its neighborhood is defined as:

$$\Omega(p_i) = \{p_j \mid \|p_j - p_i\| < r\} \tag{1}$$

where  $r$  is the neighborhood radius. To obtain the normal vectors of the point cloud  $P$ , PCA can be used to compute the initial normal vectors. These normals are then reoriented to ensure consistent directions using direction propagation and viewpoint-based correction.

The covariance matrix for each point  $p_i$  is defined as:

$$M_i = \frac{1}{|\Omega(p_i)|} \sum_{j \in \Omega(p_i)} (p_j - \bar{p})(p_j - \bar{p})^T \tag{2}$$

where  $|\Omega(p_i)|$  is the number of neighboring points of  $p_i$  and  $\bar{p}$  is the centroid of the neighborhood. The covariance matrix  $M_i$  in Eq. (2) is decomposed to obtain its eigenvalues, with the eigenvector corresponding to the smallest eigenvalue being the normal vector  $n_i$  of point  $p_i$ .

Due to the inherent ambiguity of normal vectors, where the calculation determines only the line along which the normal vector lies without specifying its direction, this paper employs a minimum spanning tree (MST) method to reorient the normal vectors. This approach transforms the challenge of achieving a globally consistent orientation into a graph optimization problem. Initially, a seed point is selected, with its normal vector oriented outward. Then, using the minimum spanning tree of the Riemannian graph, the normal vectors along the traversal path are adjusted to ensure that the dot product between the normal vectors of adjacent points is positive, resulting in a consistent outward orientation across the normal vector field.

(2) Surface Variation Metric

Feature points are generally found in areas of significant surface variation, necessitating an intuitive metric to quantify this change. In this study, the centroid projection distance is used as the metric, defined as follows:

$$D_i = |(c_i - p_i)^T n_i| \tag{3}$$

where  $c_i$  is the centroid of the neighborhood  $\Omega(p_i)$  of the target point  $p_i$ .  $D_i$  is the projection of the vector  $\overrightarrow{p_i c_i}$  in the direction of the normal vector  $n_i$ . As shown in Fig. 2, points with significant surface variation generally exhibit large centroid projection distances, whereas points in smooth regions have smaller projection distances. However, irregular noise can cause certain points on smooth surfaces to display large projection distances, leading to their misclassification as feature points. Moreover, the small offset of a target

point relative to the centroid reduces the metric’s sensitivity to actual surface variations.

To address these challenges, this paper introduces a weighted centroid method as a replacement for the original centroid. The weights are computed based on two factors: surface roughness and variations in normal vectors. This approach enhances the noise resistance of the centroid projection distance measurement and improves the differentiation between feature and non-feature points, thereby increasing the robustness of the metric. Consequently, it effectively mitigates the impact of noise and enables more accurate identification of regions with significant surface variation.

(3) Roughness Weight

In this paper, roughness is defined as the distance between each point  $p_i$  and the best-fit plane of its neighborhood  $\Omega(p_i)$ . To this end, let the coordinates of the target point  $p_i$  be  $(x_i, y_i, z_i)$ , the centroid coordinates of its neighborhood  $\Omega(p_i)$  be  $(x_c, y_c, z_c)$ , and the normal vector of that neighborhood be  $n_i = (a, b, c)$ . Then, the equation of the fitted plane for the neighborhood is:

$$aX + bY + cZ + d = 0 \tag{4}$$

where  $d = -(ax_c + by_c + cz_c)$ . Then, by calculating the distance from point  $p_i$  to the plane, the roughness  $R_i$  is defined as:

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

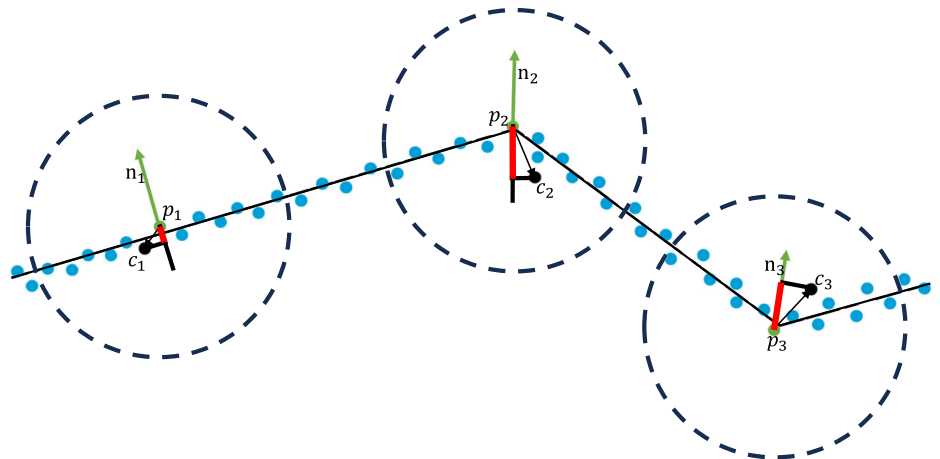
Roughness reflects the smoothness of the local region of the point cloud. Regions that are more irregular or rich in surface details have greater roughness, while smoother or flatter areas have lower roughness. To enhance the robustness of the centroid, weights are assigned to the neighboring points when calculating the centroid. Neighboring points with greater roughness are given higher weights, while those with lower roughness receive lower weights. To reduce the influence of noise, this paper defines the weight  $W_{r,i}$  based on the average roughness of the neighboring points  $p_j \in \Omega(p_i)$  of the target point  $p_i$ . Specifically,  $W_{r,i}$  is defined as:

$$W_{r,i} = e^{\frac{r_i}{\mu_r} + 1} \tag{6}$$

$$r_i = \frac{1}{|\Omega(p_i)|} \sum_{j \in \Omega(p_i)} R_j \tag{7}$$

where  $r_i$  is the average roughness within the neighborhood  $\Omega(p_i)$ , and  $\mu_r$  is a scaling factor. The value of  $W_{r,i}$  is constrained to the interval  $(0, e)$  and is negatively correlated with  $r_i$ .

**Fig. 2 | Centroid projection distance.** The red line segments represent the projection distances.



(4) Normal Difference Weight

Assuming that the normal vectors of the point cloud surface are consistently oriented, the magnitude of the normal differences within a neighborhood can reflect the degree of surface variation in that neighborhood. Areas with more distinct features exhibit greater normal differences, and vice versa. The angle  $\alpha_{ij}$  between the normals of two adjacent points  $p_i$  and  $p_j$  reflects the normal difference, thereby indicating the degree of surface variation within the neighborhood. To mitigate the effects of uneven sampling in the point cloud, the angle  $\beta_{ij}$  of the normals within a unit distance is used as a measure of normal difference:

$$\beta_{ij} = \frac{\alpha_{ij}}{|p_i - p_j|} \tag{8}$$

where  $|p_i - p_j|$  is the Euclidean distance between points  $p_i$  and  $p_j$ .

The normal difference weight  $W_{n,i}$  for the target point  $p_i$  is defined as:

$$W_{n,i} = e^{-\frac{\theta_i}{\mu_\theta} + 1} \tag{9}$$

$$\theta_i = \frac{1}{|\Omega(p_i)|} \sum_{j \in \Omega(p_i)} (\beta_{ij} - \bar{\beta}_i)^2 \tag{10}$$

where  $\theta_i$  is the variance of the unit distance normal angles within the neighborhood  $\Omega(p_i)$ .  $\mu_\theta$  is a scaling factor, and  $\bar{\beta}_i$  is the average of all normal angles within the neighborhood of point  $p_i$ . Similarly, the value of  $W_{n,i}$  is constrained to the interval  $(0, e)$  and is negatively correlated with  $\theta_i$ .

Combining the above two weighting factors, the comprehensive weight  $W_i$  is defined as:

$$W_i = \frac{1}{W_{r,i} * W_{n,i}} \tag{11}$$

To calculate the weighted centroid projection distance for each point, the weighted centroid  $W_i c_i$  is used instead of the ordinary centroid:

$$D_i = |(W_i c_i - p_i)^T n_i| \tag{12}$$

The weighted centroid projection distance  $D_i$  in Eq. (12) is used to quantitatively represent the degree of surface variation. The offset between the target point  $p_i$  in the feature area and the weighted centroid is larger, resulting in a greater projection distance; conversely, in non-feature areas, the projection distance is smaller. This approach effectively mitigates the interference of noise points and enhances the difference between feature and non-feature points. To identify feature points, this paper ranks all points based on the weighted centroid projection distance  $D_i$  and uses the 80th percentile as the threshold  $\xi$ . If the current point's  $D_i$  is greater than  $\xi$ , it is considered a feature point.

Additionally, the sign of the result from  $(W_i c_i - p_i)^T n_i$  in Eq. (12) can be used to determine the concavity or convexity of the feature points. If the sign is positive, the angle between the vector formed by the target point and the weighted centroid  $\overrightarrow{p_i c_i}$  and the normal vector  $n_i$  is less than  $90^\circ$ , as shown by point  $p_3$  in Fig. 2; if the sign is negative, the angle is greater than  $90^\circ$ , as shown by point  $p_2$  in Fig. 2. The normal vectors have been oriented outward, so the sign of  $(W_i c_i - p_i)^T n_i$  indicates whether the feature point is on a concave or convex surface; thus,  $p_2$  can be identified as a ridge point and  $p_3$  as a valley point.

**Refinement of feature points**

The detected feature points are roughly distributed around the actual feature lines, forming a band-like structure with a certain width. To achieve the connection of feature points, they need to be refined to linearly contract onto the actual feature lines. Inspired by Lee et al. <sup>40</sup>, this paper approximates the refinement process as a projection of feature points. The input consists of the extracted feature point set. By performing PCA on the local neighborhood of

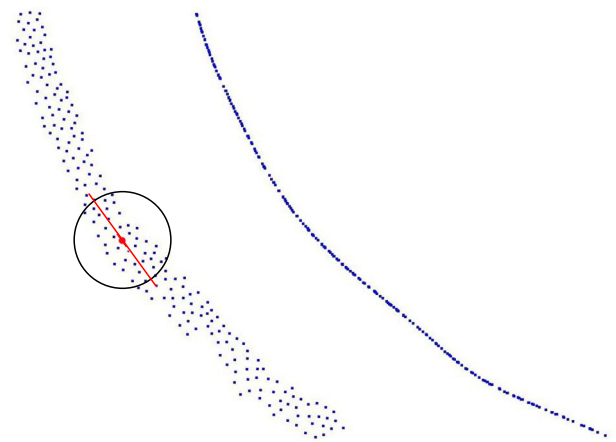


Fig. 3 | Refinement of feature points.

any feature point, the main directional line of the neighborhood is obtained, and the point is projected onto this main directional line for refinement.

Let the set of feature points be  $F$ , and given a point  $p \in F$ , the neighborhood point set is defined as:

$$\Omega'(p) = \{p_i \in F \mid \|p_i - p\| < r'\} \tag{13}$$

where  $r'$  is the neighborhood radius. Referring to Eq. (2), the 3x3 covariance matrix is calculated as follows:

$$C = \frac{1}{|\Omega'(p)|} \sum_{i \in \Omega'(p)} (p_i - \bar{p})(p_i - \bar{p})^T \tag{14}$$

where  $\bar{p}$  is the centroid of the neighborhood,  $\bar{p} = \frac{1}{|\Omega'(p)|} \sum_{i \in \Omega'(p)} p_i$ . By solving the eigenvalue equation  $Cx_i = \lambda_i x_i, i \in \{0, 1, 2\}$  and assuming  $\lambda_0 \geq \lambda_1 \geq \lambda_2$ , we obtain the eigenvector  $x_0$  corresponding to the maximum eigenvalue  $\lambda_0$ . The direction of this vector represents the principal direction of the neighborhood  $\Omega'(p)$ . The feature point  $p$  is then projected onto the line defined by the centroid  $\bar{p}$  and the eigenvector  $x_0$ , completing the refinement.

The refinement process involves multiple iterations. Since the initial feature points are band-like and relatively wide, a larger initial neighborhood radius is required to ensure coverage of all feature points on the same feature line, as shown in Fig. 3. As the iterations progress, the width of the band gradually decreases, allowing for a reduction in the neighborhood radius for subsequent iterations to improve algorithm efficiency. Let the initial neighborhood radius be  $r_1$  and the number of iterations be  $\eta$ . Then,  $r_i = \frac{1}{2^{i-1}} r_1, i = 1, 2, \dots, \eta$ .

Typically, two spatially adjacent feature lines, such as paired ridges and valleys, may lead to erroneous results if not distinguished during the refinement process. For example, when refining feature points on a ridge, a larger initial neighborhood may include nearby valley feature points, leading to refinement errors. Therefore, based on the concavity and convexity of the feature points described in Section 3.1, they are classified into valley points  $F_1$ :

$$F_1 = \{p_i \in F \mid (W_i c_i - p_i)^T n_i > 0\} \tag{15}$$

and ridge points  $F_2$ :

$$F_2 = \{p_i \in F \mid (W_i c_i - p_i)^T n_i < 0\} \tag{16}$$

Then refinement is performed separately on  $F_1$  and  $F_2$ .

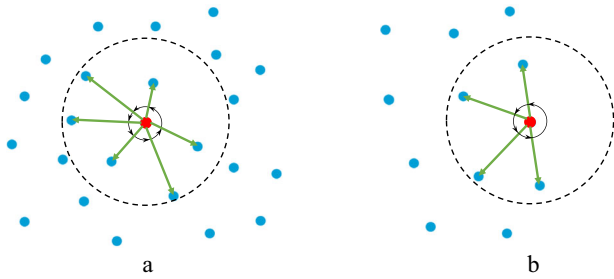


Fig. 4 | Angle criterion. a Non-boundary point. b Boundary point.

### Detection of boundary points

Previous work has identified the geometric features of the sampled surface, excluding boundary points. To generate a complete archeological line drawings, it is essential to extract boundary points. This paper employs the Angle Criterion (AC) proposed by Bendels et al.<sup>87</sup> to extract boundary points.

For a given point  $p$ , the angle criterion projects all neighboring points  $p_i (i = 1, 2, \dots, n)$  in the neighborhood  $\Omega(p)$  onto the tangent plane of  $p$ , resulting in projection points  $p'_i$ . Then, using  $p$  as the corner point, the projection points are connected in pairs to  $p$  in a clockwise (or counterclockwise) manner, forming a series of angles  $\Theta = \{\theta_1, \theta_2, \dots, \theta_{n-1}\}$ . The maximum angle  $\theta_{max}$  in  $\Theta$  is found; the larger  $\theta_{max}$  is, the higher the likelihood that  $p$  is a boundary point. Therefore, an angle threshold  $\varepsilon$  can be set, and when  $\theta_{max} > \varepsilon$ ,  $p$  is classified as a boundary point, as shown in Fig. 4.

### Feature line construction

The previous work has identified the complete set of feature points on the sampled point cloud surface, including ridge-valley feature points and boundary points. To construct the feature lines of the sampled point cloud, all feature points need to be connected following certain rules. Since the density of the refined feature points may be much higher than the average sampling density of the point cloud and unevenly distributed, uniform downsampling of the feature points is required to generate sparsely distributed points with uniform density, which serve as the input for feature line construction.

This paper proposes an improved curve growing method based on Lee et al.<sup>40</sup> to connect the feature points. When constructing the feature lines, a neighborhood radius needs to be defined as follows:

$$R = n \cdot \bar{d} \tag{17}$$

where  $\bar{d}$  is the average distance between each point and its nearest neighbor, and  $n$  is a constant, with an empirical value of  $n = 3$ .

The basic idea of the algorithm for feature line construction is as follows:

(1) Randomly select any point  $p_0$  as the seed point. In the neighborhood with a radius  $R$ , use PCA to compute the principal direction line  $l_0$ , defined by  $p_0$  and the eigenvector corresponding to the largest eigenvalue in the neighborhood. Find the nearest neighbor point  $p_1$  of  $p_0$ . If the angle between the line passing through  $p_0p_1$  and  $l_0$  is smaller than the threshold  $\tau$ , then  $p_0$  and  $p_1$  form the initial vertex set of the feature line,  $V = \{p_0, p_1\}$ . Otherwise, continue searching for the next nearest neighbor until a point meeting the condition is found.

(2) Taking  $p_1$  as an example, if the feature line grows from  $p_1$ , it is necessary to construct the neighborhood centered on  $p_1$  and compute its principal direction line  $l_1$ . Then, for all neighboring points except those in the vertex set  $V$ , calculate the angle between the line formed by  $p_1$  and each neighboring point and  $l_1$ , and find the point  $p_i$  with the smallest angle. To control the direction of curve growth, ensure that the angle  $\angle p_0p_1p_2$  is greater than  $90^\circ$ . Otherwise, continue searching for a

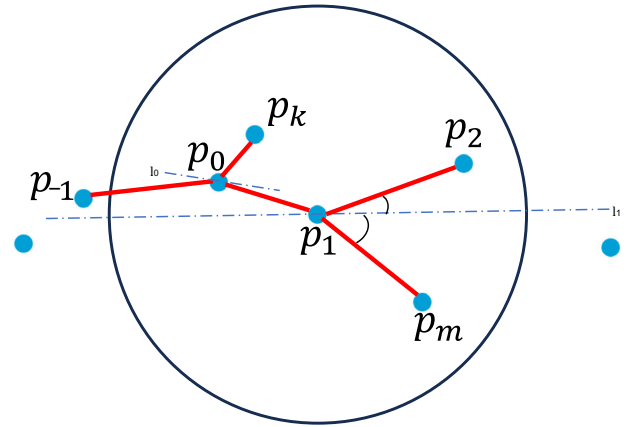


Fig. 5 | Curve growth.

point that meets the condition. Add the found point to  $V$  as the new endpoint and name it  $p_2$ . To allow the curve to grow in both directions, apply the same rule to endpoint  $p_0$  and find endpoint  $p_{-1}$ , resulting in the vertex set  $V = \{p_{-1}, p_0, p_1, p_2\}$ .

(3) Repeat step (2) until no further valid endpoints can be found, ultimately obtaining the vertex set  $V = \{p_j, \dots, p_0, p_1, \dots, p_i\}$ . Connect the points in  $V$  sequentially to form a feature curve. Then, mark the points in  $V$  to exclude them from subsequent curve growth processes.

(4) Repeat the above steps until all points are marked, completing the construction of feature curves.

Figure 5 illustrates the curve growth process. Although  $p_k$  is the nearest point to  $p_0$ , the angle between the line  $p_0p_k$  and  $l_0$  exceeds the threshold  $\tau$ , so the next nearest point  $p_1$  is selected. When continuing to grow the curve at the endpoint  $p_1$ , the line  $p_1p_2$  has the smallest angle with  $l_1$ , so  $p_2$  is chosen and  $p_m$  is excluded.

By applying the above method to ridge points, valley points, and boundary points, the corresponding ridges, valleys, and boundary contour lines can be obtained.

However, when this method is applied to complex sampled surfaces, feature line breaks may occur. To address this, this paper proposes additional connection schemes between feature curves. Typically, feature line breaks occur at the junctions between different types of feature lines. Ridges and valleys usually do not intersect, so only the connections between ridges and boundary contour lines, as well as valleys and boundary contour lines, need to be considered. At the endpoint of a ridge or valley, the nearest point on the boundary contour line is found. If the distance between the two points is less than a certain threshold, the points are connected to form a complete, closed structural feature line.

Figure 6 illustrates the overall process for the automatic generation of archeological line drawings.

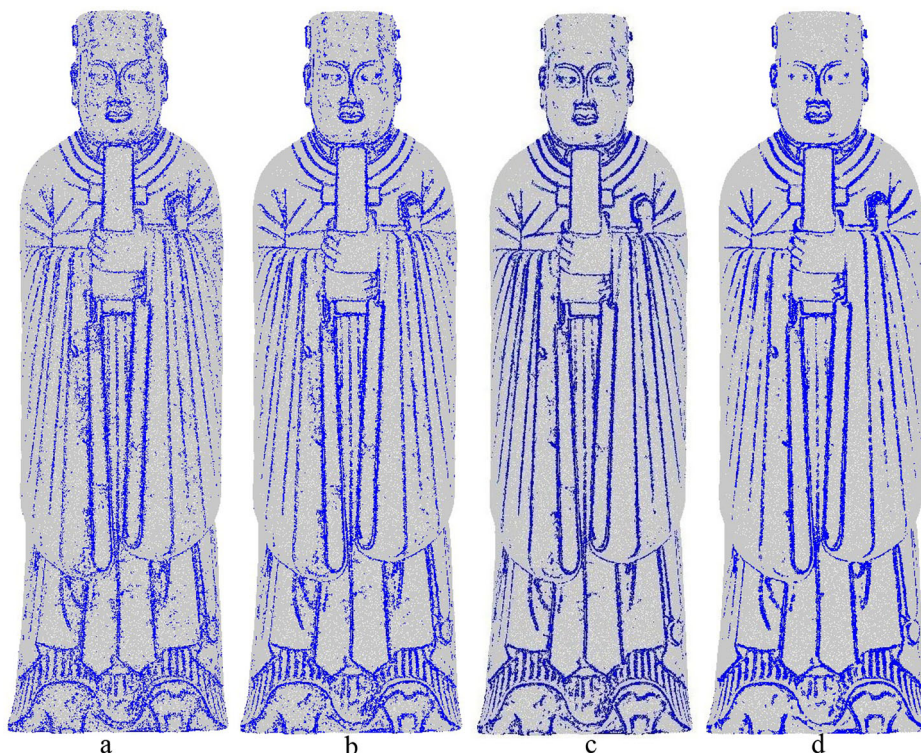
### Results

This section presents the use of various point cloud types as experimental subjects to assess the performance of the proposed method. Specifically, the experimental data primarily consists of point cloud from the stone sculptures of the Northern Song Dynasty royal tombs, including those of civil officials, auspicious birds, mounting stones, and stone elephants. The point cloud data for these sculptures was obtained using Structure from Motion (SfM) technology. Additionally, several point cloud models from the Stanford public dataset were selected for comparative experiments. The proposed method is compared with several advanced techniques, including Curvature<sup>50</sup>, Smoothness Shrinkage Index (SSI)<sup>57</sup>, Plane Fitting Residual-Local Shape Variation (PFR-LSV)<sup>70</sup>, and Structure to Detail (StD) feature perception<sup>72</sup>. The algorithm was implemented in C++, with the following hardware specifications: AMD R7-5800H CPU and 16 GB RAM.

**Fig. 6 | Overall process.** a Input point cloud. b Feature points, with red indicating ridge points and blue indicating valley points. c Refined feature points. d Sparse points. e Boundary points. f Archeological line drawings.



**Fig. 7 | Visualization of different expected values.** a  $n = 8$ . b  $n = 12$ . c  $n = 16$ . d  $n = 20$ .



### Parameter sensitivity analysis

To achieve better experimental results, the proposed method requires adjustments to several key parameters. These parameters include the neighborhood radius  $r$  for computing normal vectors and local centroids, the scale factors  $\mu_r$  and  $\mu_\theta$ , the number of iterations  $\eta$  for refinement, the angle threshold  $\varepsilon$  for boundary point determination, and the angle threshold  $\tau$  for feature line construction.

The neighborhood radius  $r$  is a critical parameter that directly affects the accuracy of feature extraction. The value of  $r$  needs to be determined based on the sample density, typically estimated as a multiple of the average sampling density of the point cloud; however, this requires calculating the density of the entire point cloud. To improve the efficiency of radius estimation, an initial radius can be estimated based on the bounding box of the point cloud. Then, random

sampling can be performed on the point cloud to calculate the number of neighboring points within the initial radius for each sampled point, and to gather statistics on the minimum, maximum, mean, and standard deviation of the point counts. Based on the statistical data, the radius is adjusted through an iterative optimization process to find the most suitable value until the number of neighboring points approaches the desired value. In this paper, experiments were conducted using the point cloud of civil official stone sculpture from the Northern Song Dynasty royal tombs, with desired values set at 8, 12, 16, and 20, respectively. Comparative experiments showed that the best results were achieved when the expected number of neighboring points was set to 16, as illustrated in Fig. 7.

$\mu_r$  and  $\mu_\theta$  are scaling factors that control the weights of roughness and normal difference, respectively; they can adjust the contributions of

**Fig. 8 | Visualization of weight value  $\mu_r$ , a  $10^{-1}$  (b)  $10^{-2}$  (c)  $10^{-3}$  (d)  $10^{-4}$**



both factors when computing the weighted centroid. Depending on the specific characteristics of the sample, the range for  $\mu_r$  is  $[10^{-4}, 10^{-1}]$ . In the experiments, values of  $\mu_r$  were set to  $10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ , with the results shown in Fig. 8, indicating that the best performance was achieved when  $\mu_r = 10^{-3}$ . To ensure that the contributions of the two weights to the centroid calculation are comparable, the value of  $\mu_\theta$  must satisfy the condition  $\frac{\bar{r}}{\mu_r} = \frac{\bar{\theta}}{\mu_\theta}$ , where  $\bar{r}$  and  $\bar{\theta}$  are the mean values of  $r_i$  and  $\theta_i$  for the randomly sampled points in the sample. Generally, an iteration count  $\eta$  in the range of 3 to 5 can yield good refinement results. Based on experience, the angle threshold  $\varepsilon$  is set to  $20^\circ$ , while the range for  $\tau$  is  $[10^\circ, 20^\circ]$ .

### Qualitative evaluation

This paper tests the proposed method on four different types of stone sculpture point cloud: civil official, auspicious bird, mounting stone, and stone elephant. These sculptures exhibit significant differences in their shape characteristics, and due to prolonged weathering, their surfaces are uneven, resulting in considerable noise in the corresponding point cloud data. Figure 9 shows the feature extraction results for the civil official. Due to the presence of irregular noise in planar areas, the Curvature method fails to correctly distinguish these noise points, resulting in their incorrect identification as feature points, as shown in Fig. 9(b). Although SSI can distinguish these noise points, there is a certain degree of feature loss, as seen in Fig. 9(c). PFR-LSV demonstrates strong resistance to noise, yielding more refined feature extraction results. However, the local surface variance used by PFR-LSV cannot accommodate features of different scales, resulting in non-feature points near deep features being incorrectly identified as feature points, as seen in Fig. 9(d). StD can recognize multi-scale features, providing a more comprehensive feature extraction result, but some noise still exists, as shown in Fig. 9(e). As shown in Fig. 9(f), (g), the proposed method produces the best feature extraction results, generating comprehensive and detailed feature lines.

The auspicious bird in Fig. 10 has multi-level features and a relatively complex structure. SSI and PFR-LSV fail to identify shallow features,

resulting in feature breakage. SSI achieves a relatively comprehensive set of features but also contains considerable noise. As shown in Fig. 10(f), the proposed method can extract the most comprehensive set of feature points, and the corresponding feature line construction results are visually appealing, as seen in Fig. 10(g).

Figure 11 presents a point cloud model of a mounting stone, with a highly complex surface pattern composed of numerous fine and dense lines, significantly increasing the difficulty of feature extraction. The Curvature and PFR-LSV methods mistakenly identify points in uneven areas as feature points, leading to excessive noise points in the results. The feature points identified by SSI are very fragmented and evidently contain many errors. StD effectively removes the influence of noise points; however, there are still instances of feature loss, primarily due to the small feature scale of this model, while StD uses a larger initial scale that is not suitable for such models. Figure 11(g) displays the results of the feature line construction, showcasing most of the model's features.

Figure 12 depicts a large stone elephant. Compared to other methods, the proposed method still achieves the best feature extraction results. The final constructed feature lines clearly reflect the facial features of the stone elephant and the patterns on its side, as shown in Fig. 12(g).

To demonstrate the applicability of the proposed method, we selected two models, the armadillo and the Happy Buddha, from the Stanford dataset for testing. Figures 13 and 14 show the results of feature extraction from the armadillo model and the Happy Buddha model using the proposed method, respectively. The point cloud data for both models was sampled from their respective meshes. The proposed method achieved superior results in feature extraction compared to other methods.

To objectively assess the performance of the proposed method, we compared the lines generated by the proposed method with suggestive contours from the method proposed by DeCarlo et al.<sup>25</sup>, as well as with lines extracted by experts. The generation of suggestive contours requires performing a 3D reconstruction of the original point cloud to create a 3D mesh model, followed by the adjustment of various parameters to achieve optimal results. Additionally, precise



Fig. 9 | Feature extraction and line drawings for civil official. a Input (b) Curvature (c) SSI (d) PFR-LSV (e) StD (f) Ours (g) Lines.

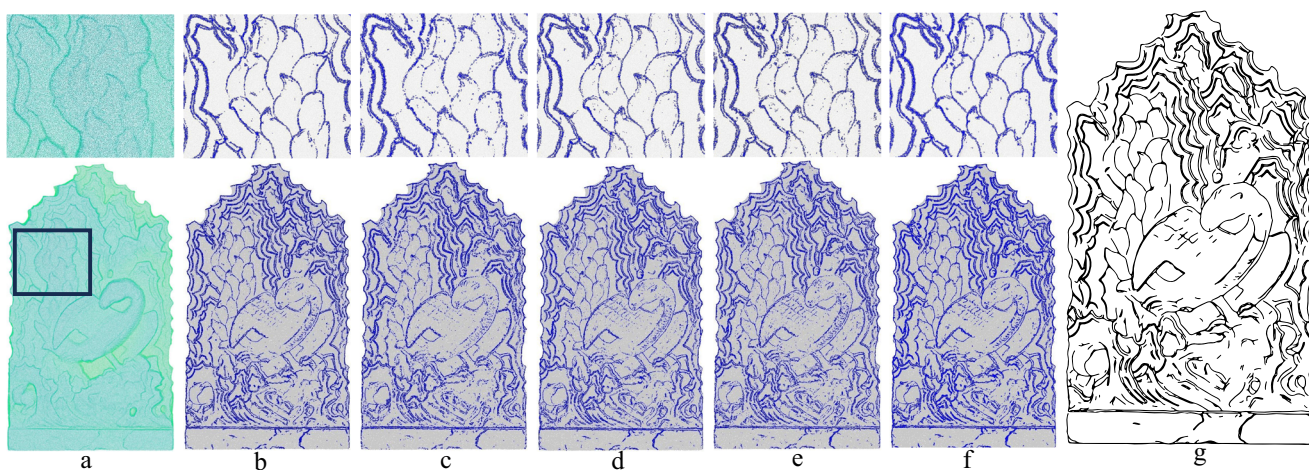


Fig. 10 | Feature Extraction and Line Drawings for Auspicious Bird. a Input (b) Curvature (c) SSI (d) PFR-LSV (e) StD (f) Ours (g) Lines.

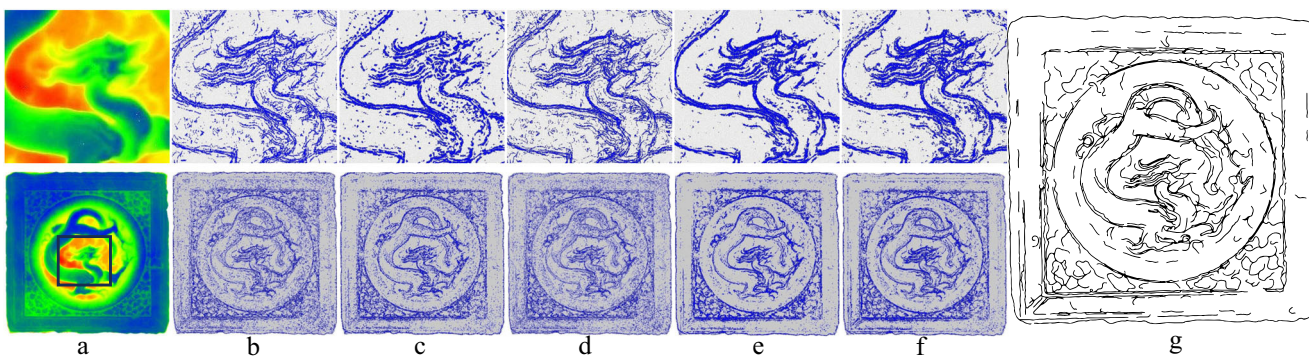


Fig. 11 | Feature extraction and line drawings for mounting stone. a Input (b) Curvature (c) SSI (d) PFR-LSV (e) StD (f) Ours (g) Lines.

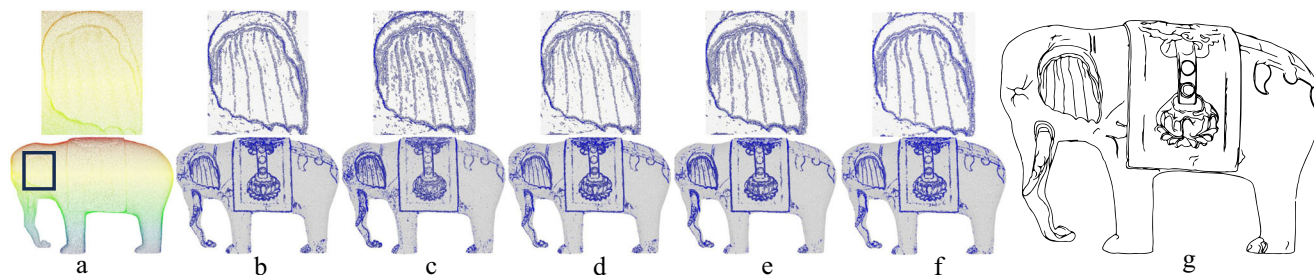


Fig. 12 | Feature extraction and line drawings for stone elephant. a Input (b) Curvature (c) SSI (d) PFR-LSV (e) StD (f) Ours (g) Lines.

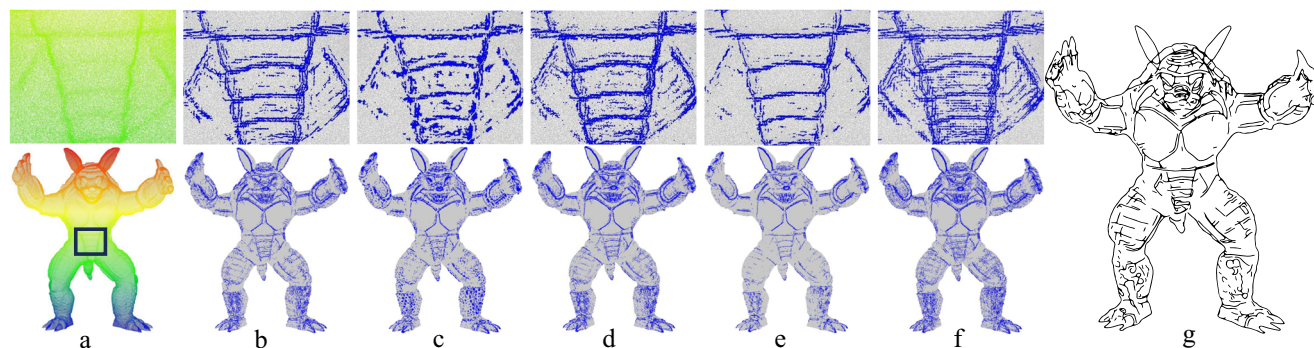


Fig. 13 | Feature Extraction and Line Drawings for Armadillo. a Input (b) Curvature (c) SSI (d) PFR-LSV (e) StD (f) Ours (g) Lines.

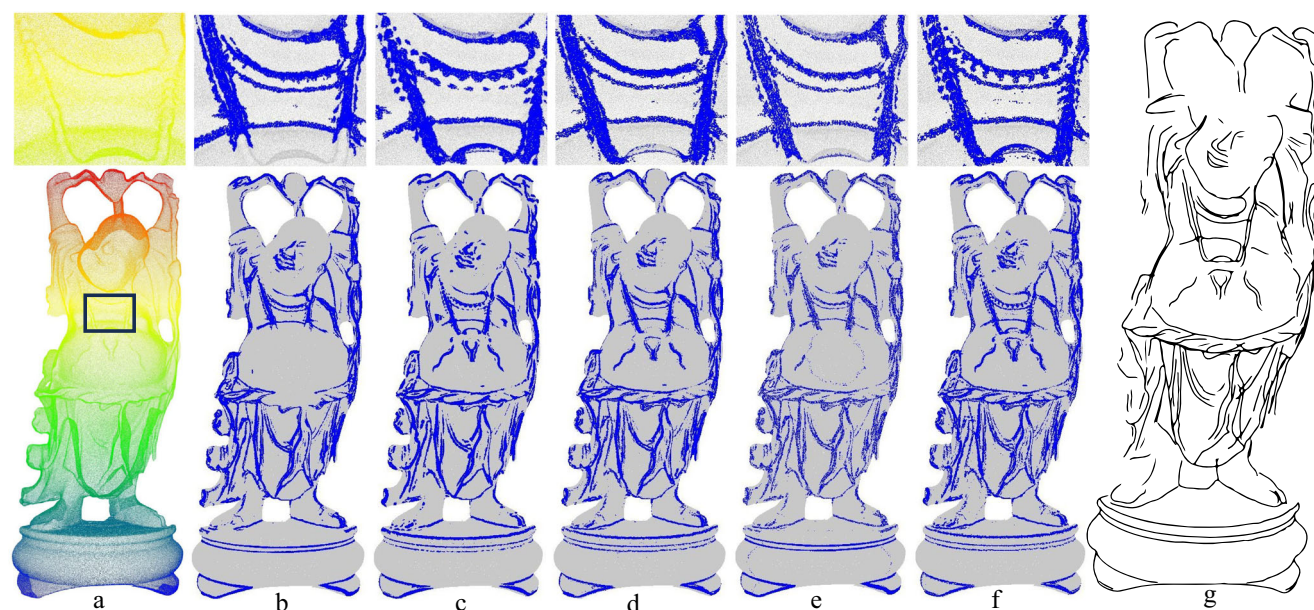
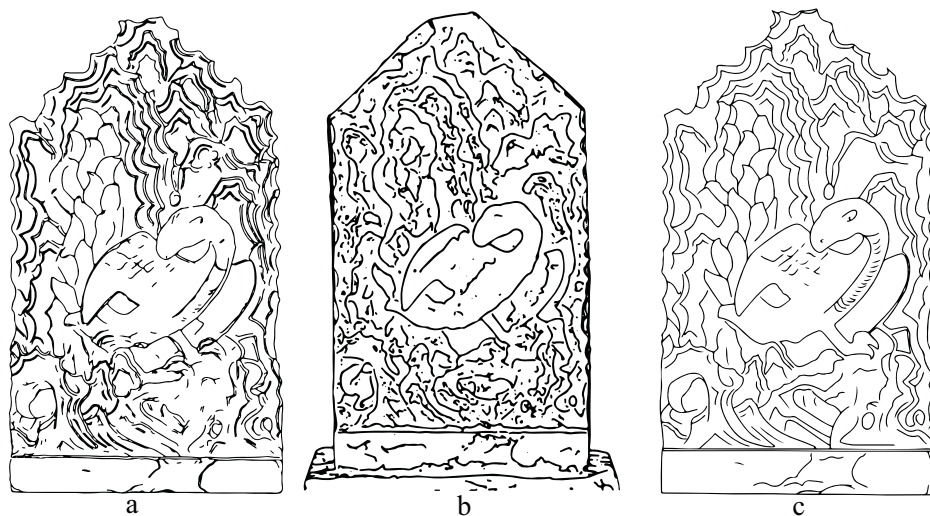


Fig. 14 | Feature extraction and line drawings for happy Buddha. a Input (b) Curvature (c) SSI (d) PFR-LSV (e) StD (f) Ours (g) Lines.

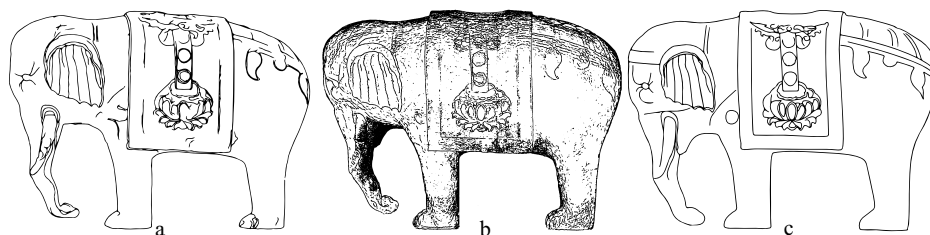
orthophotos can be derived from the 3D mesh model, and experts were invited to extract lines using AutoCAD based on these orthophotos. Figures 15–17 illustrate the lines extracted by different methods. Compared to the suggestive contours, those generated by the proposed method are more comprehensive and continuous. The lines extracted by experts are more esthetically pleasing and accurate, as the extraction process involves manual optimization and refinement, such as

filling in missing lines and removing redundant ones. However, the lines extracted by experts from orthophotos are 2D, and the entire process is time-consuming. In contrast, the proposed method is more efficient and generates 3D lines that contain more information. This makes the proposed method applicable to fields such as reconstruction, registration, and object detection, demonstrating its broader potential for practical applications.

**Fig. 15 | Extracted lines of auspicious bird. a Ours. b Suggestive contours. c Expert-extracted lines.**



**Fig. 16 | Extracted lines of stone elephant. a Ours. b Suggestive contours. c Expert-extracted lines.**



**Fig. 17 | Extracted lines of civil official. a Ours. b Suggestive contours. c Expert-extracted lines.**

**Quantitative evaluation**

To quantitatively assess the quality of feature point extraction, we compared the proposed method with four other methods, and the quantitative results

are presented in Table 1. Due to the lack of true feature information, we asked professional illustrators to manually draw and select feature points on these surfaces as the ground truth feature point set. This paper uses three metrics to evaluate the performance of each method: Precision (Pre), Recall (Rec), and F1-score. Pre and Rec are defined as  $Pre = \frac{TP}{TP+FP}$  and  $Rec = \frac{TP}{TP+FN}$  respectively. TP (True Positives) represents the number of correctly detected points, FP (False Positives) represents the number of incorrectly detected points, and FN (False Negatives) represents the number of points that were incorrectly rejected. Pre indicates the proportion of correctly detected points among all detected points, while Rec reflects the completeness of the correctly detected feature points. The F1-score is defined as  $2 \times \frac{Pre \times Rec}{Pre+Rec}$  representing the trade-off between precision and recall.

As shown in Table 1, the proposed method generally achieved relatively better results across the three metrics, producing results that are closest to the true features.

Table 2 records the running times of all methods. Different methods may employ different feature point refinement and connection strategies, so only the running time of the feature point detection phase is recorded. We implemented all algorithms in C++ under the Windows 10 operating system, using Visual Studio 2019 and PCL 1.12 as the development environment. As shown in Table 2, Curvature has the highest efficiency because this method does not require the calculation of normal vectors. On the other hand, StD takes the longest time because it requires feature point detection at multiple scales. The proposed method is only slightly less efficient than Curvature, with a shorter running time than the other three methods, demonstrating a satisfactory level of efficiency.

**Robustness test**

We assessed the robustness of the proposed method by examining the impact of point cloud density and accuracy on the final results. The

**Table 1 | Quantitative evaluation results**

Model	Methods	Pre	Rec	F1
Civil official (373k)	Curvature	0.4928	0.6942	0.5764
	SSI	0.4466	0.6291	0.5224
	PFR-LSV	0.3689	0.516	0.4302
	StD	0.532	0.7344	0.6170
	Ours	<b>0.5465</b>	<b>0.7535</b>	<b>0.6335</b>
Auspicious Bird (638k)	Curvature	0.6678	<b>0.6849</b>	0.6762
	SSI	0.5868	0.6019	0.5943
	PFR-LSV	0.6513	0.6681	0.6596
	StD	0.6438	0.6471	0.6454
	Ours	<b>0.6936</b>	0.6802	<b>0.6868</b>
Mounting Stone (1416k)	Curvature	0.4658	0.6849	0.5545
	SSI	0.4896	0.6723	0.5666
	PFR-LSV	0.5136	0.7025	0.5934
	StD	0.5026	0.6631	0.5718
	Ours	<b>0.5369</b>	<b>0.7432</b>	<b>0.6234</b>
Stone Elephant (323k)	Curvature	0.4555	0.807	0.5823
	SSI	0.4082	0.7231	0.5218
	PFR-LSV	0.4547	0.8055	0.5812
	StD	<b>0.4845</b>	0.7988	<b>0.6032</b>
	Ours	0.4755	<b>0.8293</b>	0.5695
Armadillo (2410k)	Curvature	0.7053	0.7935	0.7468
	SSI	0.6502	0.7682	0.7043
	PFR-LSV	0.7186	0.8109	0.7620
	StD	0.7296	0.8091	0.7673
	Ours	<b>0.7452</b>	<b>0.8524</b>	<b>0.7952</b>
Happy Buddha (1184k)	Curvature	0.7349	0.8157	0.7732
	SSI	0.7163	0.8059	0.7585
	PFR-LSV	0.7928	0.8194	0.8059
	StD	0.7824	0.8361	0.8084
	Ours	<b>0.8091</b>	<b>0.8436</b>	<b>0.8260</b>

The bold values in Table 1 indicate the best performance for each metric across the methods compared.

Auspicious Bird was chosen as the experimental object, with a height of 3.12 m, a width of 1.81 m, and a surface point density of 104,512 points/m<sup>2</sup>. The original point cloud was downsampled to generate datasets with varying densities, from which feature points and line drawings were extracted, as shown in Fig. 18. The results demonstrate that the proposed method performs well in high-density areas and areas with uneven density distribution, producing high-quality detection outcomes. However, its performance decreases in low-density regions.

Accuracy refers to the uncertainty in the position of each point within 3D space, and it is a critical factor influencing the quality of feature extraction results. To assess the impact of varying levels of accuracy on the proposed method, we introduced Gaussian noise to simulate different degrees of error. Gaussian noise, a widely used error simulation technique, assumes that the positional error of each point follows a normal distribution. By adjusting the standard deviation, we

**Table 2 | Running times of related methods (Unit: s)**

Model	Curvature	SSI	PFR-LSV	StD	Ours
Civil official	0.87	7.74	5.67	10.55	7.53
Auspicious Bird	1.41	11.04	9.72	20.86	8.22
Mounting Stone	0.74	7.77	7.00	9.80	6.97
Stone Elephant	3.34	19.77	17.44	35.75	6.56
Armadillo	5.22	24.98	24.91	40.64	22.55
Happy Buddha	2.55	24.06	19.10	29.68	12.39

generated errors that simulate point clouds with varying levels of accuracy. Specifically, Gaussian noise with standard deviations of 1 mm, 3 mm, 5 mm, and 7 mm was added to the original point cloud to simulate corresponding levels of accuracy.

The proposed feature point detection method operates within the neighborhood of the target point; thus, errors in neighboring points inevitably influence the detection results. To mitigate this impact, we expanded the neighborhood radius during feature point detection, thereby including a larger number of neighboring points in the detection process. This approach is based on the principle that a larger sample size provides a more comprehensive representation of the overall characteristics, thereby reducing the influence of errors on the estimated results. However, as the neighborhood radius increases, smaller-scale features may be omitted, leading to feature loss. Moreover, as the point cloud error increases, a larger neighborhood radius is required to reduce the impact of the error, which consequently exacerbates the issue of feature loss.

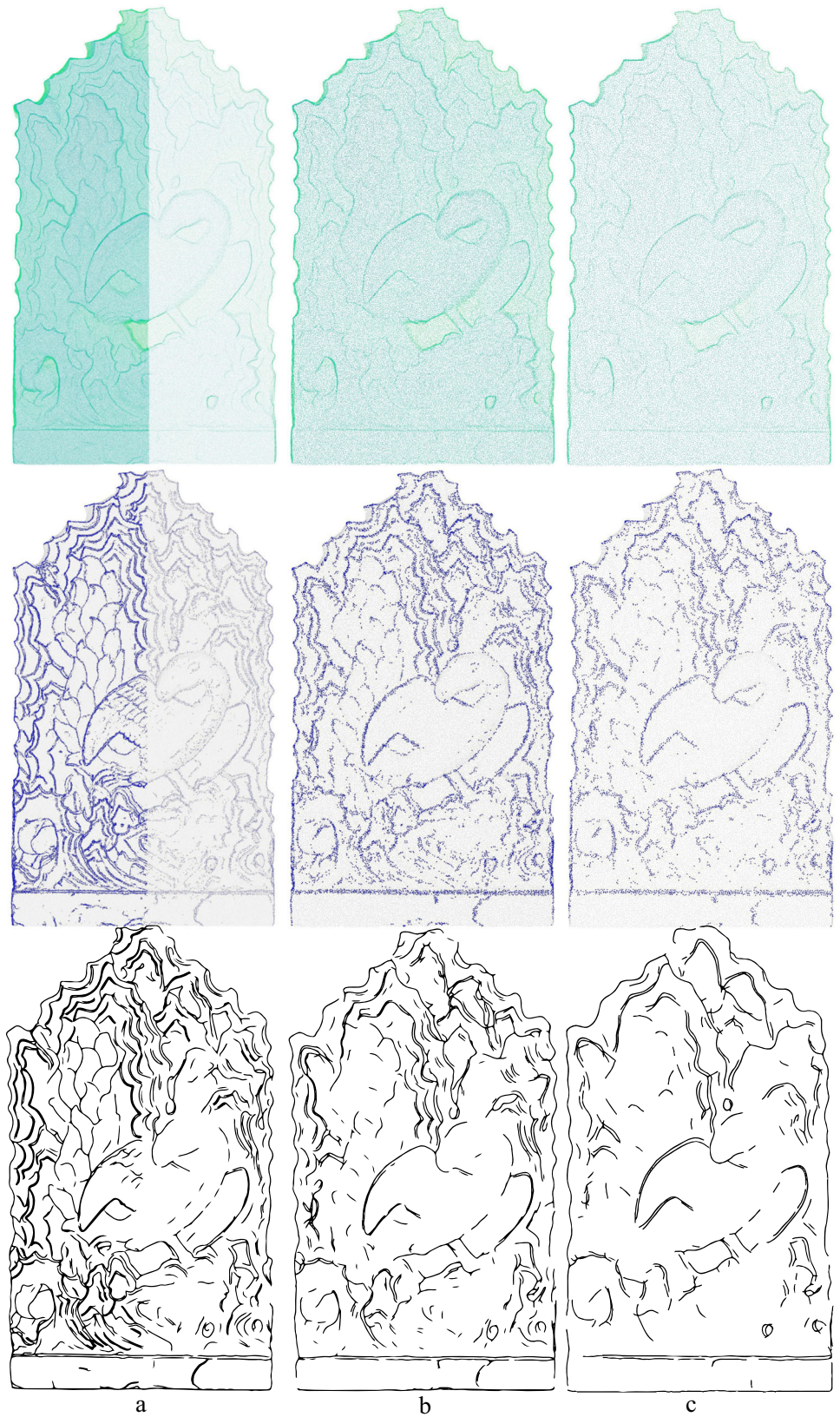
As shown in Fig. 19, as point cloud accuracy decreases, the line drawings generated by the proposed method gradually exhibit feature line breaks and omissions. When the point cloud accuracy reaches 5 mm, the quality of the resulting line drawings deteriorates. This is attributed to the complex surface of the object under study—the stone sculpture—which necessitates a high level of point cloud accuracy. When point cloud accuracy continues to decrease to 7 mm, the proposed method fails to extract correct feature points.

## Discussion

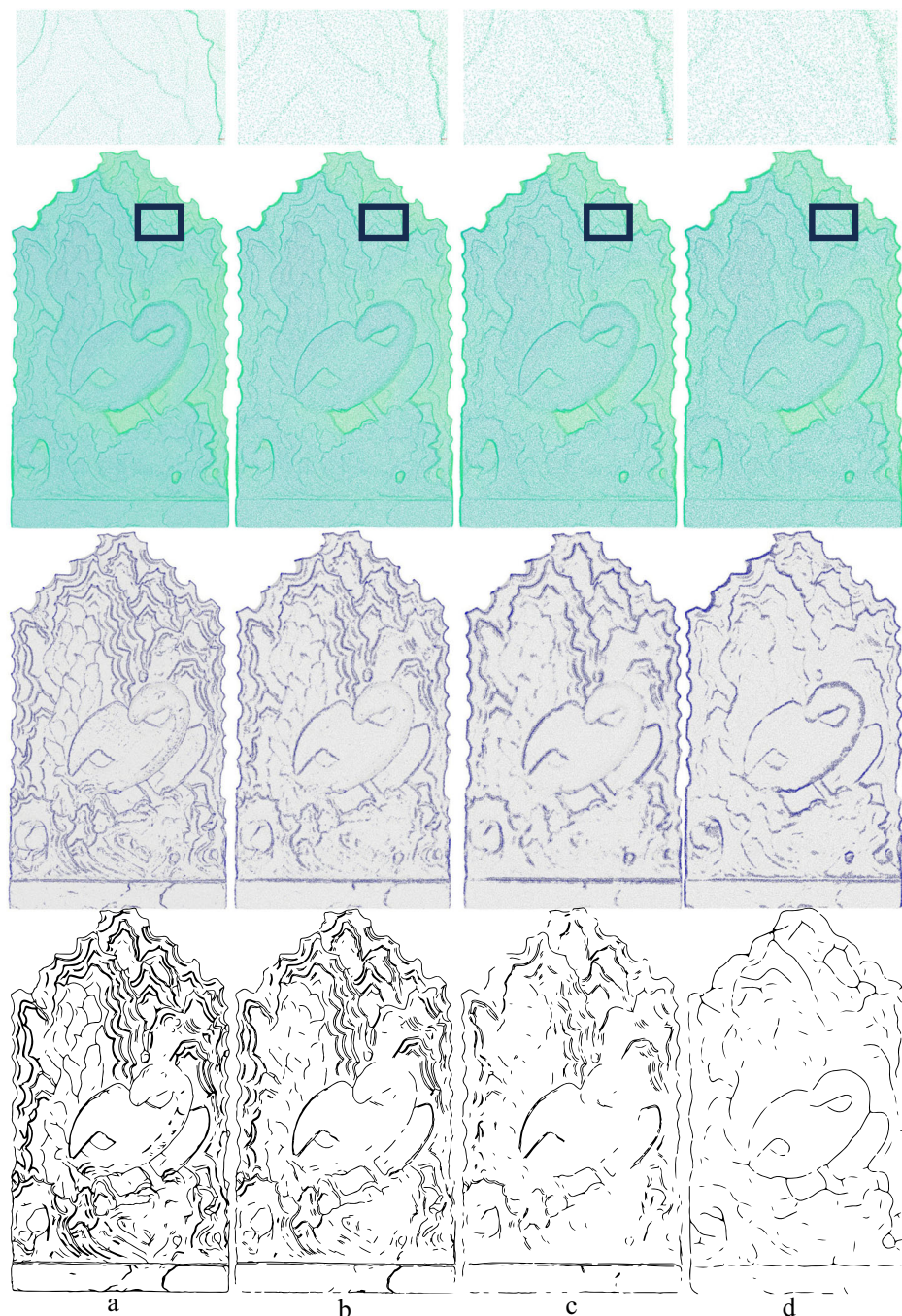
This paper investigates the automatic generation of archeological line drawings from point cloud of large stone sculptures and introduces an effective method for feature line extraction. The method leverages local weighted centroid projection as an innovative geometric metric for detecting feature points on the surface. In this approach, the computation of the local weighted centroid integrates two key weighting factors—surface roughness and normal difference—to bolster the metric’s noise resistance and robustness. Subsequently, the convexity and concavity of the feature points are accurately identified by analyzing the geometric relationship between the feature vectors, which are formed by the feature points and their weighted centroids, and the reoriented normal vectors. Furthermore, this paper proposes an enhanced curve growth method that simultaneously balances the distance and angular factors of neighboring points during the curve expansion process. By processing from both ends of the curve, this method leads to a substantial improvement in algorithmic efficiency. The proposed approach is capable of generating complete and smooth feature curves and has demonstrated exceptional performance across various types of stone sculpture point cloud, showcasing broad potential for practical applications.

In future work, we will explore rendering and stylization techniques for feature lines to create more esthetically pleasing archeological line drawings.

**Fig. 18 | Feature extraction results at different densities.** Surface points density (unit: points/ $m^2$ ): **a** 104512 (left half) and 2240 (right half) **(b)** 1088 **(c)** 640.



**Fig. 19 |** Line drawings of point cloud with different levels of accuracy. a 1 mm. b 3 mm. c 5 mm. d 7 mm.



**Data availability**

Part of the data supporting the findings of this study is available from Zhengzhou University, but restrictions apply to its availability. These data were used under license for the current study and are therefore not publicly accessible. However, data may be obtained from the authors upon reasonable request and with permission from Zhengzhou University. The remaining datasets generated during the study are available in the Stanford 3D Scanning repository, [<https://graphics.stanford.edu/data/3Dscanrep/>].

**Abbreviations**

CAD computer-aided design  
 CAAD computer-aided architectural design

PCD point cloud data  
 MDL minimum description length  
 TV tensor voting  
 SSI Smooth Shrinkage Index  
 PFR-LSV Plane Fitting Residual-Local Shape Variation  
 PCA principal component analysis  
 MST minimum spanning tree  
 AC Angle Criterion  
 SfM Structure from Motion  
 StD Structure to Detail  
 TP True Positives  
 FP False Positives  
 FN False Negatives

Received: 25 October 2024; Accepted: 31 March 2025;

Published online: 13 May 2025

## References

- Lewuillon, S. Archaeological illustrations: a new development in 19th century science. *Antiquity* **76**, 223–234 (2002).
- Furferi, R. et al. Enhancing traditional museum fruition: current state and emerging tendencies. *Herit. Sci.* **12**, 20 (2024).
- Eliel, L. T. One hundred years of photogrammetry. *Photogramm. Eng.* **25**, 359–363 (1959).
- Thompson, M. M. & Whitmore, G. D. Improvements in photogrammetry through a century. *Trans. Am. Soc. Civ. Eng.* **118**, 845–856 (1953).
- Rosenfeld, A. & Thurston, M. Edge and curve detection for visual scene analysis. *IEEE Trans. Comput.* **C-20**, 562–569 (1971).
- Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 679–698 (1986).
- Aldsworth F. An integrated photogrammetry and CAD system applied to the restoration of a seventeenth century house. In: *Proc. of the CIPA XIV International Symposium: Architectural photogrammetry & information systems, Delphi, Greece*. Athens: Technical Chamber of Greece; p. 25–31, (1991)
- Almagro A. Simplified methods in architectural photogrammetry. In: *Proc. of the CIPA XIV International Symposium: Architectural photogrammetry & information systems, Delphi, Greece*. Athens: Technical Chamber of Greece; p. 209–225, (1991)
- Streilein A., Beyer H. A. Development of a digital system for architectural photogrammetry. In: *Proc. of the CIPA XIV International Symposium: Architectural photogrammetry & information systems, 1991, Delphi, Greece*. Athens: Technical Chamber of Greece; p. 55–61, (1991)
- Cooper, M. C. Interpretation of line drawings of complex objects. *Image Vis. Comput.* **11**, 82–90 (1993).
- Grompone von Gioi, R., Jakubowicz, J., Morel, J. M. & Randall, G. LSD: a fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 722–732 (2010).
- Akinlar, C. & Topal, C. EDLines: a real-time line segment detector with a false detection control. *Pattern Recogn. Lett.* **32**, 1633–1642 (2011).
- Dhankhar, P. & Sahu, N. A review and research of edge detection techniques for image segmentation. *Int J. Comput. Sci. Mob. Comput.* **2**, 86–92 (2013).
- Dolapsaki, M. M. & Georgopoulos, A. Edge detection in 3D point clouds using digital images. *ISPRS Int J. Geo-Inf.* **10**, 229 (2021).
- Arias, F. et al. Use of 3D models as a didactic resource in archaeology. A case study analysis. *Herit. Sci.* **10**, 112 (2022).
- Davis, A., Belton, D., Helmholz, P., Bourke, P. & McDonald, J. Pilbara rock art: laser scanning, photogrammetry and 3D photographic reconstruction as heritage management tools. *Herit. Sci.* **5**, 25 (2017).
- Tong, Y., Cai, Y., Nevin, A. & Ma, Q. Digital technology virtual restoration of the colours and textures of polychrome Bodhidharma statue from the Lingyan Temple, Shandong, China. *Herit. Sci.* **11**, 12 (2023).
- Barazzetti, L., Brumana, R., Oreni, D., Previtali, M. & Roncoroni, F. True-orthophoto generation from UAV images: Implementation of a combined photogrammetric and computer vision approach. *ISPRS Ann. Photogramm., Remote Sens. Spat. Inf. Sci.* **II-5**, 57–63 (2014).
- Liu, Y., Zheng, X., Ai, G., Zhang, Y. & Zuo, Y. Generating a high-precision true digital orthophoto map based on UAV images. *ISPRS Int J. Geo-Inf.* **7**, 333 (2018).
- Lv, J., Jiang, G., Ding, W. & Zhao, Z. Fast digital orthophoto generation: a comparative study of explicit and implicit methods. *Remote Sens.* **16**, 786 (2024).
- Agrafiotis, P., Talaveros, G., Georgopoulos, A. Orthoimage-to-2D Architectural Drawing with Conditional Adversarial Networks. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci.* X-M-1–2023, 11–18 (2023).
- Alshwabkeh, Y. & Baik, A. Integration of photogrammetry and laser scanning for enhancing scan-to-HBIM modeling of Al Ula heritage site. *Herit. Sci.* **11**, 147 (2023).
- Yang, C. et al. Digital characterization of the surface texture of Chinese classical garden rockery based on point cloud visualization: small-rock mountain retreat. *Herit. Sci.* **11**, 13 (2023)
- Hubeli, A., Gross, M. Multiresolution feature extraction for unstructured meshes. In: *Proc.* p. 287–294; (IEEE, 2001).
- DeCarlo, D., Finkelstein, A., Rusinkiewicz, S. & Santella, A. Suggestive contours for conveying shape. *ACM Trans. Graph* **22**, 848–855 (2003).
- Ohtake, Y., Belyaev, A. & Seidel, H.-P. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph* **23**, 609–612 (2004).
- Hildebrandt, K. Polthier, K., Wardetzky, M. Smooth feature lines on surface meshes. In: *Proc. of the third Eurographics symposium on Geometry processing: Vienna, Austria; 04–06 July 2005*. ACM International Conference Proceeding Series; 255:85–es. (2005).
- Yoshizawa, S., Belyaev, A., Seidel, H.-P. Fast and robust detection of crest lines on meshes. In: *Proc. of the 2005 ACM Symposium on Solid and Physical Modeling*. p. 227–232 (ACM, 2005).
- Judd, T., Durand, F. & Adelson, E. Apparent ridges for line drawing. *ACM Trans. Graph* **26**, 19 (2007).
- Luo, T., Li, R. & Zha, H. 3D line drawing for archaeological illustration. *Int J. Comput. Vis.* **94**, 23–35 (2011).
- Lu, Q., Wang, L., Lou, L., Zheng, H. Evaluating line drawings of human face. In: *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. p. 1–9; (2013)
- Li, Z., Qin, S., Jin, X., Yu, Z. & Lin, J. Skeleton-enhanced line drawings for 3D models. *Graph Models* **76**, 620–632 (2014).
- Wang, X., Geng, G., Li, X., Zhou, J. A cultural relic line drawings generation algorithm based on explicit ridge line. In: *2015 International Conference on Virtual Reality and Visualization (ICVRV)*. p. 173–176 (IEEE, 2015).
- Uchida, M. & Saito, S. Stylized line-drawing of 3D models using CNN with line property encoding. *Comput. Graph.* **91**, 252–264 (2020).
- Liu, D., Fisher, M., Hertzmann, A., Kalogerakis, E. Neural strokes: Stylized line drawing of 3D shapes. In: *2021 IEEE/CVF Int Conf Comput Vis ICCV*. 2021. p. 14184–14193 (IEEE, 2021).
- Ren, P., Yang, M. A line-drawing extraction method for traditional ornament based on 3D models. In: *2022 15th International Symposium on Computational Intelligence and Design (ISCID)*. p. 245–248 (IEEE, 2022).
- Jin, H. et al. A semi-automatic oriental ink painting framework for robotic drawing from 3D models. *IEEE Robot Autom. Lett.* **8**, 6667–6674 (2023).
- Tang, Y., Dong, F., Zhang, J. An optimization framework for suggestive contours. In: *2024 7th International Conference on Computer Information Science and Application Technology (CISAT)* p. 628–634 (IEEE, 2024).
- Nguyen, V., Fisher, M., Hertzmann, A. & Rusinkiewicz, S. Region-aware simplification and stylization of 3D line drawings. *Comput Graph Forum* **43**, e15042 (2024).
- Lee, I.-K. Curve reconstruction from unorganized points. *Comput. Aided Geom. Des.* **17**, 161–177 (2000).
- Gumhold, S., Kami, Z., Isenburg, M., Seidel, H.-P. Predictive point-cloud compression. In: *ACM SIGGRAPH 2005 Sketches*. p. 137–1 (ACM, 2005).
- Pauly, M., Keiser, R. & Gross, M. Multi-scale feature extraction on point-sampled surfaces. *Comput. Graph. Forum* **22**, 281–289 (2003).
- Demarsin, K., Vanderstraeten, D., Volodine, T. & Roose, D. Detection of closed sharp edges in point clouds using normal estimation and graph theory. *Comput.-Aided Des.* **39**, 276–283 (2007).
- Daniels, J. I., Ha, L. K., Ochotta, T., Silva, C. T. Robust smooth feature extraction from point clouds. In: *International Conference on Shape Modeling and Applications (SMI 07)*. p. 123–136 (IEEE, 2007).
- Lu, Z., Baek, S., Lee, S. Robust 3D line extraction from stereo point clouds. In: *2008 IEEE Conference on Robotics, Automation and Mechatronics*; 2008. p. 1–5 (IEEE, 2008).

46. Rodríguez Miranda, Á., Valle Melón, J. M., Martínez Montiel, J. M. 3D Line Drawing from Point Clouds using Chromatic Stereo and Shading. In: *Proc. of the 14th International Conference on Virtual Systems and Multimedia (VSMM 2008)*. p. 77–84 (Digital Heritage, 2008).
47. Briese, C. & Pfeifer, N. Towards automatic feature line modelling from terrestrial laser scanner data. *Int Arch. Photogramm. Remote Sens Spat. Inf. Sci.* **XXXVII**, 463–468 (2008).
48. Weber, C., Hahmann, S., Hagen, H. Sharp feature detection in point clouds. In: 2010 Shape Modeling International Conference. p. 175–186 (IEEE, 2010).
49. Chen, T., Wang, Q. 3D line segment detection for unorganized point clouds from multi-view stereo. In: Kimmel R., Klette R., Sugimoto A., editors. *Computer vision—ACCV 2010. Lecture Notes in Computer Science*, vol 6493. p. 400–411 (Springer, 2011).
50. Mérigot, Q., Ovsjanikov, M. & Guibas, L. J. Voronoi-based curvature and feature estimation from point clouds. *IEEE Trans. Vis. Comput. Graph* **17**, 743–756 (2011).
51. Park, M. K., Lee, S. J. & Lee, K. H. Multi-scale tensor voting for feature extraction from unstructured point clouds. *Graph. Models* **74**, 197–208 (2012).
52. Pang, X., Song, Z. & Xie, W. Extracting valley-ridge lines from point-cloud-based 3D fingerprint models. *IEEE Comput. Graph. Appl.* **33**, 73–81 (2013).
53. Kim, S.-K. Extraction of ridge and valley lines from unorganized points. *Multimed. Tools Appl.* **63**, 265–279 (2013).
54. Altantsetseg, E., Muraki, Y., Matsuyama, K. & Konno, K. Feature line extraction from unorganized noisy point clouds using truncated Fourier series. *Vis. Comput.* **29**, 617–626 (2013).
55. Lin, Y. et al. Line segment extraction for large scale unorganized point clouds. *ISPRS J. Photogramm. Remote Sens* **102**, 172–183. (2015).
56. Bazazian, D., Casas, J. R., Ruiz-Hidalgo, J. Fast and Robust Edge Extraction in Unorganized Point Clouds. In: 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA). 2015. p. 1–8 (IEEE, 2015).
57. Nie, J. Extracting feature lines from point clouds based on smooth shrink and iterative thinning. *Graph. Models* **84**, 38–49 (2016).
58. Yuan, L., Matsuyama, K., Chiba, F., Konno, K. A study of feature line extraction and closed frame structure of a stone tool from measured point cloud. In: *Nicograph Int Nicolnt*. 2016. p. 44–51 (IEEE, 2016).
59. Zhang, Y., Geng, G., Wei, X., Zhang, S. & Li, S. A statistical approach for extraction of feature lines from point clouds. *Comput. Graph.* **56**, 31–45 (2016).
60. Lin, Y., Wang, C., Chen, B., Zai, D. & Li, J. Facet segmentation-based line segment extraction for large-scale point clouds. *IEEE Trans. Geosci. Remote Sens* **55**, 4839–4854 (2017).
61. Wang, Y., Ma, Y., Zhu, A., Zhao, H. & Liao, L. Accurate facade feature extraction method for buildings from three-dimensional point cloud data considering structural information. *ISPRS J. Photogramm. Remote Sens* **139**, 146–153 (2018).
62. Ahmed, S., Tan, Y., Chew, C., Mamun, A., Wong, F. Edge and Corner Detection for Unorganized 3D Point Clouds with Application to Robotic Welding. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). p. 7350–7355 (IEEE, 2018).
63. Mitropoulou, A. & Georgopoulos, A. An automated process to detect edges in unorganized point clouds. *ISPRS Ann. Photogramm. Remote Sens Spat. Inf. Sci.* **IV-2/W6**, 99–105 (2019).
64. Lu, X., Liu, Y., Li, K. Fast 3D Line Segment Detection from Unorganized Point Cloud. ArXiv. 2019; abs/1901.02532.
65. Chen, X. & Yu, K. Feature line generation and regularization from point clouds. *IEEE Trans. Geosci. Remote Sens* **57**, 9779–9790 (2019).
66. Zhang, W. et al. Large-scale point cloud contour extraction via 3D guided multi-conditional generative adversarial network. *ISPRS J. Photogramm. Remote Sens* **164**, 97–105 (2020).
67. Cosgrove, C., Yuille, A. L. Adversarial Examples for Edge Detection: They Exist, and They Transfer. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). p. 1059–1068. (IEEE, 2020).
68. Abdellali, H., Frohlich, R., Vilagos, V., Kato, Z. L2D2: Learnable Line Detector and Descriptor. In: 2021 International Conference on 3D Vision (3DV). p. 442–452 (IEEE, 2021).
69. Hu, Z. et al. Geometric feature enhanced line segment extraction from large-scale point clouds with hierarchical topological optimization. *Int. J. Appl. Earth Obs Geoinf.* **112**, 102858 (2022).
70. Chen, H. et al. Multiscale feature line extraction from raw point clouds based on local surface variation and anisotropic contraction. *IEEE Trans. Autom. Sci. Eng.* **19**, 1003–1016 (2022).
71. Liu, T. et al. Neighbor reweighted local centroid for geometric feature identification. *IEEE Trans. Vis. Comput Graph* **29**, 1545–1558 (2023).
72. Liu, Z. et al. Robust and accurate feature detection on point clouds. *Comput.-Aided Des.* **164**, 103592 (2023).
73. Bode, L., Weinmann, M. & Klein, R. BoundED: Neural boundary and edge detection in 3D point clouds via local neighborhood statistics. *ISPRS J. Photogramm. Remote Sens* **205**, 334–351 (2023).
74. Cao, L., Xu, Y., Guo, J. & Liu, X. WireframeNet: a novel method for wireframe generation from point cloud. *Comput. Graph.* **115**, 226–235 (2023).
75. Zang, Y. et al. LCE-NET: Contour extraction for large-scale 3-D point clouds. *IEEE Trans. Geosci. Remote Sens* **61**, 1–13 (2023).
76. Zong, W. et al. Toward efficient and complete line segment extraction for large-scale point clouds via plane segmentation and projection. *IEEE Sens J.* **23**, 7217–7232 (2023).
77. Zhu, X. et al. NerVE: Neural volumetric edges for parametric curve extraction from point cloud. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). p. 13601–13610 (IEEE, 2023).
78. Jiao, X. et al. MSL-Net: Sharp feature detection network for 3D point clouds. *IEEE Trans. Vis. Comput. Graph.* **30**, 6433–6446 (2024).
79. Yu, J., Wang, J., Zang, D. & Xie, X. A feature line extraction method for building roof point clouds considering the grid center of gravity distribution. *Remote Sens.* **16**, 2969 (2024).
80. Wu, C. et al. UAV building point cloud contour extraction based on the feature recognition of adjacent points distribution. *Meas* **230**, 114519 (2024).
81. Si, H. & Wei, X. Feature extraction and representation learning of 3D point cloud data. *Image Vis. Comput.* **142**, 104890 (2024).
82. Wu, J. et al. Three-dimensional contour detection method based on fusion of machine vision and laser radar. *Meas. Sci. Technol.* **35**, 105203 (2024).
83. Betsas, T., Georgopoulos, A. 3D edge detection and comparison using four-channel images. *Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **J1-ISPRES-Archives**, XLVIII-2/W2-2022 (2024).
84. Makka, A., Pateraki, M., Betsas, T. & Georgopoulos, A. 3D edge detection based on normal vectors. *Int Arch. Photogramm. Remote Sens Spat. Inf. Sci.* **XLVIII-2/W4-2024**, 295–300 (2024).
85. Aggarwal, A., Stolkin, R. & Marturi, N. Unsupervised learning-based approach for detecting 3D edges in depth maps. *Sci. Rep.* **14**, 796 (2024).
86. Xin, X. et al. Accurate and complete line segment extraction for large-scale point clouds. *Int J. Appl. Earth Obs Geoinf.* **128**, 103728 (2024).
87. Bendels, G., Schnabel, R. & Klein, R. Detecting holes in point set surfaces. *J. WSCG* **14**, 89–96 (2006).

## Acknowledgements

The work is supported by the National Natural Science Foundation of China (No. 42241759, No. 42001405), Key R&D and promotion project of Henan Province (CN) (No. 232102240017), the Natural Science Foundation of Henan Province (CN) (No. 242300420212), China Postdoctoral Science Foundation (No. 2024M752938), Archaeological Innovation and Enhancement Plan 2024 of the Archaeological Innovation Center of Zhengzhou University.

### Author contributions

L.J.: Supervision, Project administration, Funding acquisition. Y.J.T.: Writing—review & editing, Writing—original draft, Validation, Methodology. H.G.H.: Cultural heritage resources provision. W.H.: Data organization. Z.J.: Cultural heritage resources provision. C.H.: Writing—review & editing, Validation, Funding acquisition.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to Hao Cui.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025