Journal for Geometry and Graphics
Volume 29 (2025), No. 1, 107–123

Fitting Curves to Point Clouds: Targeted Approximation Methods for HBIM

Larysa Ivanova

Dresden University of Technology, Dresden, Germany
Kyiv National University of Construction and Architecture, Kyiv, Ukraine
larysa.ivanova@mailbox.tu-dresden.de, ivanova.ls@knuba.edu.ua

Abstract. This study focuses on the reconstruction and fitting of curves to 2D and 3D scanned point clouds in the context of HBIM applications. The paper begins by analyzing the distinctive characteristics of the input data, such as inherent noise, outliers, non-uniform sampling, and variable point density. Recognizing the critical impact of these factors on the choice of approximation methods, a classification system is proposed to evaluate both input data attributes and modeling objectives.

The key aspect of selecting a method is defined by the modeling objectives, namely the required accuracy and level of detail of the reconstructed curves. The author emphasizes that aligning method selection with this classification enables optimization of computational resources and processing time.

The methods proposed in this study varies depending on the specified criteria and includes the Normal Vector method, the Crawling method, the Trend method, and the Gravity method. Each technique is briefly described in terms of its core principles and applicability based on the proposed classification.

This task-oriented approach aims to achieve a balance between reconstruction quality and computational efficiency. To validate the proposed methods, a comparative evaluation is conducted. The results demonstrate satisfactory accuracy in curve reconstruction combined with optimal processing time. Overall, the proposed approximation methods significantly enhance the processing and analysis of 2D and 3D scanned point clouds. Their ability to handle noise, outliers, and variable point densities – along with their computational efficiency – makes them valuable tools for applications in architecture and industrial design.

Key Words: point cloud approximation, curve Fitting, HBIM (Heritage Building Information Modeling), approximation types, data features

MSC 2020: 68W25 (primary), 65D10, 65D17, 68W40

ISSN 1433-8157/

1 Introduction

Spatial forms represented by point clouds are commonly encountered across a wide range of applications where such data is collected using scanning devices. However, point clouds are not well-suited for direct geometric processing. Therefore, converting point clouds into parametric representations, also known as approximation methods, is a highly relevant task in computer graphics, image processing, and several applied fields. One of the most prominent application areas of these methods is the creation of digital building models in architecture, especially due to the growing importance of Building Information Modeling (BIM). Furthermore, in [13], the author highlights the effectiveness of mathematical and geometric methods for processing discrete point data in forecasting and optimization of the processes. A well-known case study [14] describes the virtual reconstruction of the war-damaged St. Catherine's Church (Katharinenkirche) in Nuremberg as a means to support historical interpretation.

The research presented in this article is part of the "freeform4BIM" project [9], which involves the development of algorithms for modeling free-form surfaces from point clouds in order to generate BIM-compatible models. The aim of this research project is to create models of structural elements of historical buildings based on 3D scanned point clouds in the context of HBIM (Heritage Building Information Modeling). To achieve this goal, it was necessary to create mathematical models of geometric elements derived from point clouds, with particular attention to free-form features.

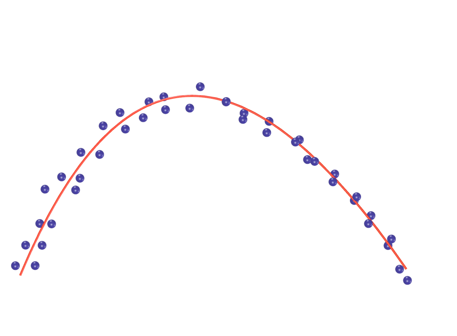
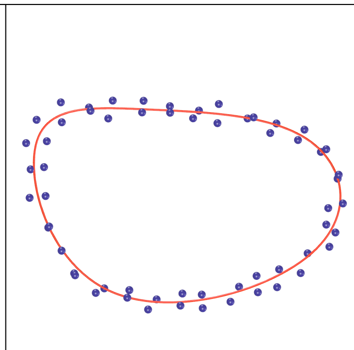
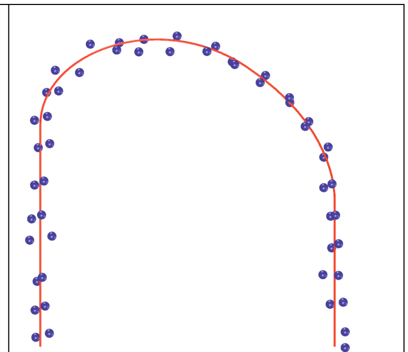
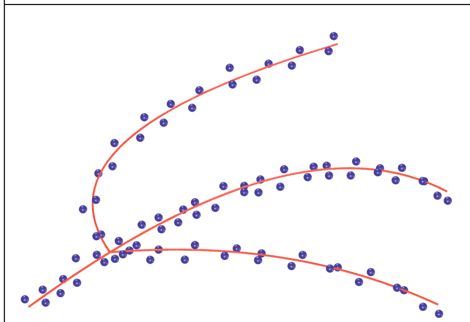
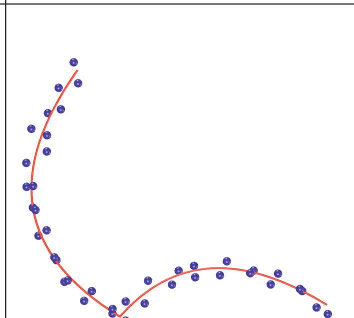
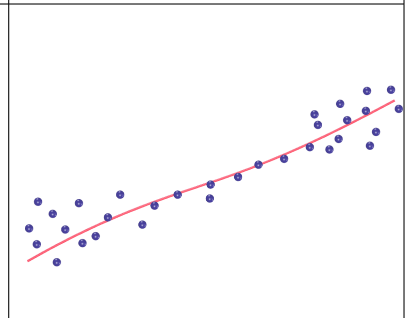
A key aspect of point cloud processing is selecting the most appropriate method for a given dataset and application. Each method has its own advantages and limitations, and the optimal choice depends on factors such as data characteristics (e.g., noise levels, data density), the desired smoothness of the result, and available computational resources. The author introduces the curve approximation techniques tailored for processing scanned point cloud data, developed in the context of the current research project. The employment of multiple methods is grounded in the conceptual framework proposed by the study, which emphasizes a task-oriented selection of approximation strategies. This approach necessitates a detailed analysis of the characteristics of the input data, including noise levels, point cloud density, and the geometric nature of the target curve: whether it is linear, branching, or complex in topology. Such a data-driven evaluation enables the identification of the most suitable approximation method in terms of computational efficiency, time expenditure, and reconstruction accuracy. The primary objective was to determine and implement effective approximation methods applicable to HBIM workflows, specifically for modeling point clouds generated from scans of architectural features during the restoration of historical structures. In typical HBIM practice, parametric object libraries are combined with point cloud processing algorithms derived from laser scanning or photogrammetric techniques, forming a comprehensive digital heritage modeling framework. Recent research increasingly explores geometry-based HBIM workflows, emphasizing procedural modeling [4, 8], classification [4, 5], and algorithmic reconstruction [7]. This section briefly reviews the most relevant studies related to curve reconstruction from point-based data, specifically in the context of HBIM.

In recent years, several efficient algorithms have been proposed for reconstructing curves from point clouds. In [1] the authors use a truncated Fourier series to detect feature points from unstructured point clouds, and then grow curves from these points. A skeletonization algorithm proposed in [3], based on graph theory, has been successfully applied to extract structural parameters of botanical trees and is valuable for shape analysis. It is robust against sparse sampling and variable point density. In [6] a method is presented that identifies geometric features in a point cloud and outputs a set of smoothed curves. The feature detection process is multi-stage and refinement-based, with demonstrated advantages for surface meshing and point-based geometry densification. The main advantage of the medial skeleton extraction method in [10] lies in its low sensitivity to the quality of the input point cloud. An interesting smoothing approach is proposed in [15], which emphasizes preserving curve characteristics while avoiding distortion. It is applicable to both vectors and point lists. The study in [17] investigates B-spline fitting to planar point clouds based on Delaunay triangulation, which allows handling intersecting curves. A skeletal representation is used to encode the topological structure, guiding the merging of curve segments. The stepwise process allows iterative validation and optimization of computation. The authors point out that the intersection areas, often affected by noise, pose significant challenges during reconstruction. In [18] a curve evolution method is proposed for smoothing 3D curves, where a curve is represented by an ordered set of 3D points or as a medial axis of a volume. In [19] the authors highlight that while skeleton extraction from 2D shapes is relatively well-studied, 3D skeletonization remains significantly more complex. The authors provide a detailed analysis of the properties of various 3D skeleton types. In [21] a semi-automatic method is proposed for generating polygonal models of trees from sparse point clouds. A multi-step skeleton and branching reconstruction process is applied. In [20], a multi-stage algorithm is introduced for reconstructing 2D curves from unorganized noisy data, preserving fine details and curvature variations through its layered refinement.

The reviewed algorithms undoubtedly exhibit unique strengths and are suited for specific use cases. However, many still face limitations, especially when high noise levels or outliers affect input data. In cases where sharp features such as corners, intersections, or branches are present, noise-robustness becomes particularly challenging: sharp features may be misinterpreted as noise, while most modern methods tend to smooth or even amplify such distortions. These challenges are tackled to varying degrees through iterative processes, staged corrections, or the synthesis of new data. This ambiguity and variability in input data drives the need for continued research in this field. A universal approach remains an open challenge, one that requires a detailed classification of point cloud characteristics and the goals of reconstruction. Depending on the objective, whether it's skeleton extraction or generating smooth parametric curves, different methods are employed. The specific nature of the data and the problem always leads to a diversity of approaches. Despite the variety of available techniques for fitting curves and surfaces to point clouds, a crucial task remains: analyzing, classifying, and organizing both the properties of the input data and the capabilities of the algorithms. Such an approach is highly practical and functional for researchers and practitioners alike.

The author emphasizes the need for a classification system for these methods to streamline research efforts and support practitioners in identifying suitable algorithms. This article aims to initiate such a framework and define principles for method classification. Advancing this initiative will require collective effort from the research community. Due to the specific nature of the input data and desired outcomes in spatial point cloud processing, particularly for architectural elements, the author developed original methods. In addition, the author focuses on the concept of targeted method selection, which is applicable not only to HBIM but to point cloud approximation in general. The aim of this study is to define a framework for identifying goal-oriented, efficient approximation methods for 2D and 3D point cloud data. In support of this concept, the article presents several of the author's own approximation methods.

Table 1: Schemes of distribution types of input point clouds with characteristic curves

		
a) Open linear distribution	b) Closed linear distribution	c) Smoothly connected segments
		
d) Branching structures	e) Angular junctions	f) Variable point width or density

2 Integrated Features of the Smoothing Approximation Methods

To select an appropriate curve approximation method for a given application, one must first define the key characteristics of the point cloud data and the objectives of the task. These factors include properties of the input data, dimensionality, the desired representation of the resulting curve, technical and software constraints, interactive response speed, and more. Such clarification allows for the selection of the most efficient method with optimal resource consumption.

Another specific feature of these tasks lies in the discrete nature of digitized data. This calls for the application and development of discrete differential geometry attributes [2, 13] to compute and assess approximation results.

The author proposes a classification of input and output attributes, as well as areas of application, to justify the target selection of the approximation method:

1. *Input Data Organization*
2. *Input Point Distribution Types* (Table 1 and Table 2)
   To enable comparative testing, synthetic point clouds were generated based on randomized samples along known target curves. These curves serve as structural skeletons for the generated datasets.
3. *Spatial Properties of Input Data*
4. *Output Curve Representation*
5. *Technical and Software Considerations*

To systematize the approaches, the author has developed a classification table (Table 3).

Table 2: Classification of Point Cloud Distributions

№	Type	Description	Typical Applications
1	Open linear distribution	Sequential points along an open curve	Wall contours, open arches, section profiles
2	Closed linear distribution	Points form a closed curve or polygonal loop	Room perimeters, column cross-sections
3	Smooth segment connections	Segments connected smoothly with varying curvature	Architectural envelopes, blended geometry
4	Branching routes	One curve branches into two or more directions	Cracks, path networks, drainage channels
5	Angular junctions	Segments join at sharp or obtuse angles	Building corners, joints, complex profiles
6	Variable width/density	Irregular spacing or changing curve thickness	Worn surfaces, erosion zones, sensor noise

3 Spatial Point Set Approximation Methods

The integrated features of the input data and the objectives of the problem determine the preferred method for processing a set of spatial points. In this section, the author focuses on the input data processing procedure, which divides methods into two groups: methods of data averaging and methods of geometric relations between groups of original data points. A similar classification of smoothing approximation methods can be found in [16].

The creation of geometric objects was done in Rhinoceros 3D and Grasshopper. The following Grasshopper plugins were used for algorithm, development, and testing: Cockroach and Leafvein. The choice of programs was determined by the project objectives.

3.1 Methods of Geometric Relationship Between Points

Methods based on various geometric relationships between groups of points are widely used in digitized data systems. Depending on user-defined tolerances and the algorithmic approach, points may be adjusted, ”extraneous” points removed, and new output points generated. The outcome is obtained as a novel, discrete set of output points. Additional data processing algorithms are required to produce a continuous curve or to change the density of discrete points. Usually, these procedures operate within localized clusters of points, although a global procedure is also possible.

3.1.1 The Normal Vector Method

The Normal Vector method is specifically developed for processing 2D point sets representing planar sections such as arches, floor plans, cross-sections, and other architectural outlines. A key objective of this method is to provide intuitive control over the resulting curve, especially in areas of high curvature or sharp corners, while minimizing computational effort to support interactive processing.

This method is particularly suited for ordered point series with minimal noise, where input points can be meaningfully sequenced. The core idea is based on geometric relationships among neighboring points, allowing for curve smoothing through localized adjustments using

Table 3: Classification of Approximation Attributes and Input Data Characteristics

№	**Category**	**Subtype / Type**	**Description / Purpose**
1	Input Data Organization	a) Unorganized point cloud	No structure; only spatial positions are known
		b) Organized point cloud	Structured set; may include weights, tangents, curvature
2	Point Distribution	a) Linear (open)	Sequential points along an open path
		b) Linear (closed)	Points form a closed contour
		c) Smoothly joined	Segments joined smoothly with varying curvature
		d) Branching	Multiple directional paths from a common point
		e) Angular joints	Sharp or obtuse connections between segments
		f) Density/width variation	Uneven point spacing; worn/damaged structures
3	Spatial Dimension	a) 2D	Flat sections, architectural plans
		b) 3D	Volumetric point sets, spatial distributions
4	Output Form	a) Parametric	B-splines, Bézier curves etc.
		b) Implicit	Defined by an equation $R(X_1, \ldots, X_n) = 0$
		c) Discrete	Meshes, edges, point sequences
		d) Attributes inside domain	Tangents, curvature, torsion
		e) Attributes outside domain	Extrapolated shape info
		f) Approximate best-fit	Not all points used, but curve fits the majority
5	Technical Aspects	a) Control point influence	Local/global influence on curve shape
		b) Robustness	Resistance to noise, outliers
		c) Computational load	Suitability for real-time applications

normal vectors.

The principle of processing point clouds using the Normal Vector method allows for effective smoothing of both three-dimensional and two-dimensional point clouds. However, a special algorithm for controlling the resulting smoothing curve of three-dimensional point clouds, especially in areas with high curvature or sharp angles, has not yet been developed. The author has identified the search for such an algorithm as a priority task for future research.

Principle of the Method Given an input point cloud, the method extends the point set by extrapolating two additional points at the beginning and end of the sequence to allow free movement of the boundary points. For each point in the set, its immediate neighbors (preceding and succeeding points) are used to define a local circle. The displacement vector for each point is directed toward the center of this circle (Figure 1:(a)).

This displacement represents an imaginary normal to the desired smoothed curve at the current point. Each point is moved along this normal direction by a distance X, optionally

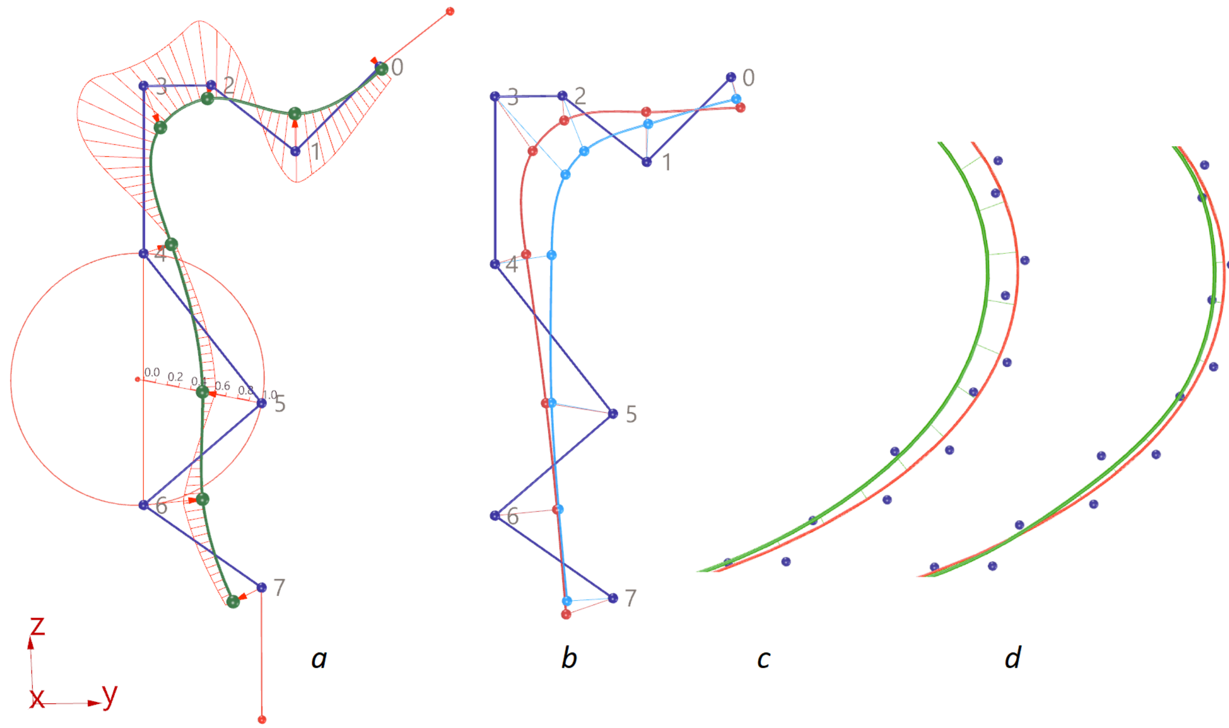


Figure 1: (a) Schematic representation of the initial iteration of the Normal Vector method applied to an ordered 2D point cloud. (b) Comparison between uniform (light blue dots) and adaptive (red dots) weighted points after five iterations. (c) Result using uniform weighted points; (d) Result using curvature-adaptive weighted points. The red colored curve is the target curve.

scaled by a user-defined weighting coefficient. After a few iterations, the discrete curve reaches a state of equilibrium, where further movement is minimal.

Handling Sharp Features and Curve Shrinkage A common issue in smoothing-based approximation methods is curve shrinkage in regions of high curvature or at corners. This phenomenon occurs when the approximated curve deviates significantly from the original point set. The Normal Vector method effectively mitigates this effect by adjusting the weight coefficients locally. In areas requiring tighter adherence to the original geometry, such as sharp corners or high-curvature zones, the weight coefficients can be applied to reduce unwanted deviation (see Figure 1:(b), (d)).

At the initial stage, a uniform weighting coefficient is assigned to all points in the input point cloud: $(w_i = 0.5, \quad \forall i)$. This value ensures an acceptable rate of convergence during the initial iterations of the algorithm.

Following each iteration, a procedure is performed to check the spatial relationship between the input point cloud and the current discrete approximating curve (Figure 1:(a), (c)). Specifically, each original point is evaluated based on its relative position, whether it lies to the left or right of the curve in the local context. If a group of three or more consecutive original points is found on the same side of the curve, this is treated as an indicator of potential over-contraction in that region. Continued iteration in such zones may lead to the approximating curve deviating significantly from the original point distribution due to excessive smoothing.

To mitigate this effect, an adaptive weighting scheme is introduced for the affected points, based on the local curvature of the approximating curve at the current iteration (Figure 1:(a)). The weight w_i assigned to point i is defined as:

$$w_i = \frac{1}{1 + a \cdot k_i}$$

where:

- w_i is the adaptive weight for point i,
- k_i is the local curvature at point i,
- $a \in \mathbb{R}_+$ is a tunable smoothing parameter that controls sensitivity to curvature.

The formulation ensures that points located in regions of high curvature exert less influence on the subsequent iterations of the smoothing process. As a result, sharp features and areas with high directional variation are better preserved, enhancing the geometric fidelity of the final approximation.

This weighting strategy grants designers fine control over the influence of individual points on the resulting curve, allowing for responsive adjustment without sacrificing computational efficiency. The simplicity of the method ensures that only basic geometric operations are required, making it especially suitable for real-time or semi-automated processing during the sketch-level evaluation of scanned data.

3.1.2 The Crawling Method

The Crawling method is designed for efficient curve approximation of 3D unstructured point clouds with moderate noise density. Its primary purpose is to provide a rapid and visually interpretable smoothing technique during the sketch-level assessment of point cloud data.

This method relies on sequential local fitting of line segments, referred to as FitLines, computed from randomly selected groups of N neighboring points (Figure 2). These groups are determined through a stochastic traversal of the point set, ensuring flexible coverage of the entire cloud.

Each FitLine is calculated for a given local neighborhood, and its midpoint is extracted as a discrete point on the approximating curve. The sequence of such midpoints forms the backbone of the resulting discrete curve.

It is important to note that the number of output points forming the curve may not match the number of input points (Figure 3). Consequently, the resulting curve may be partially sparse, and a supplementary interpolation algorithm may be required to fill large gaps between points and improve continuity.

This approach offers a practical balance between computational efficiency and sufficient geometric fidelity, especially suited for early-stage geometric reconstruction in HBIM workflows.

3.1.3 The Trend Method

The Trend method is designed to operate reliably on unordered, unoriented, and branched 3D point clouds. A distinguishing feature of such input data is the presence of complex topologies, including branches, intersections, and varying densities. Among the techniques discussed, this method is the most versatile, capable of handling nearly any point distribution. However, this generality comes at the cost of increased computational complexity and runtime.

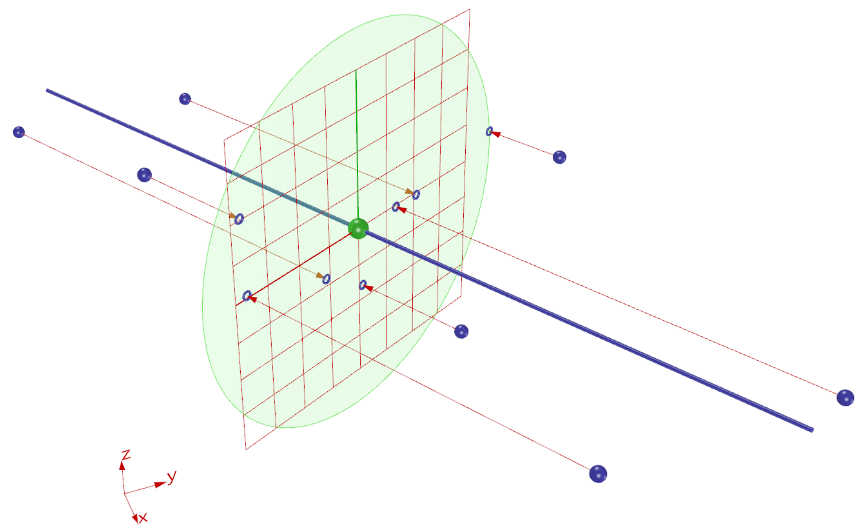


Figure 2: Schematic representation of processing a random group of N neighbors (blue dots) using the Crawling method. The green dot is the desired middle point of the FitLine.

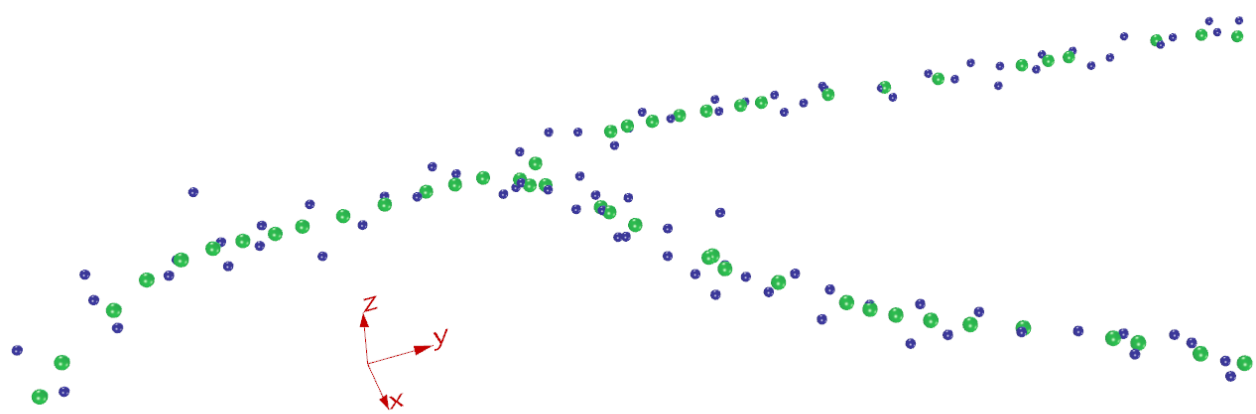


Figure 3: The Crawling Method compresses the input data points (blue dots) to the desired points (green dots) of the Fitting curve in just two iterations.

The most important parameter guiding the selection of the discretization step is the resolution of the point cloud, typically defined as the mean distance between neighboring points. This resolution helps determine the size of local neighborhoods used in the trend estimation process.

Purpose and Applicability The primary objective of the Trend Method is to extract approximate curve skeletons from spatially complex point sets—particularly those representing deteriorated, damaged, or eroded surfaces in historic architectural structures. Such data is often affected by high levels of noise and irregular geometry. For these use cases, the initial approximation must at least identify directional trends in the cloud, serving as a foundation for further refinement by other methods.

Despite its relatively high computational demands, the method demonstrates robust noise suppression while effectively handling branched or intersecting data structures.

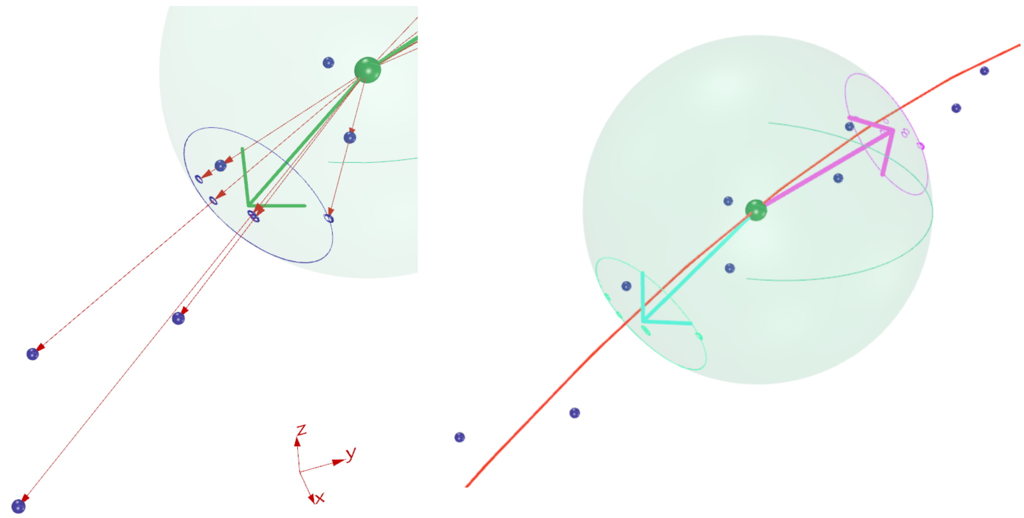


Figure 4: Left: The Trend direction is defined as a vector originating from the projection center and pointing toward the centroid of the point cluster. Right: Points projected onto the sphere are grouped into clusters based on nearest-neighbor distances, which are determined by the spatial resolution of the input dataset.

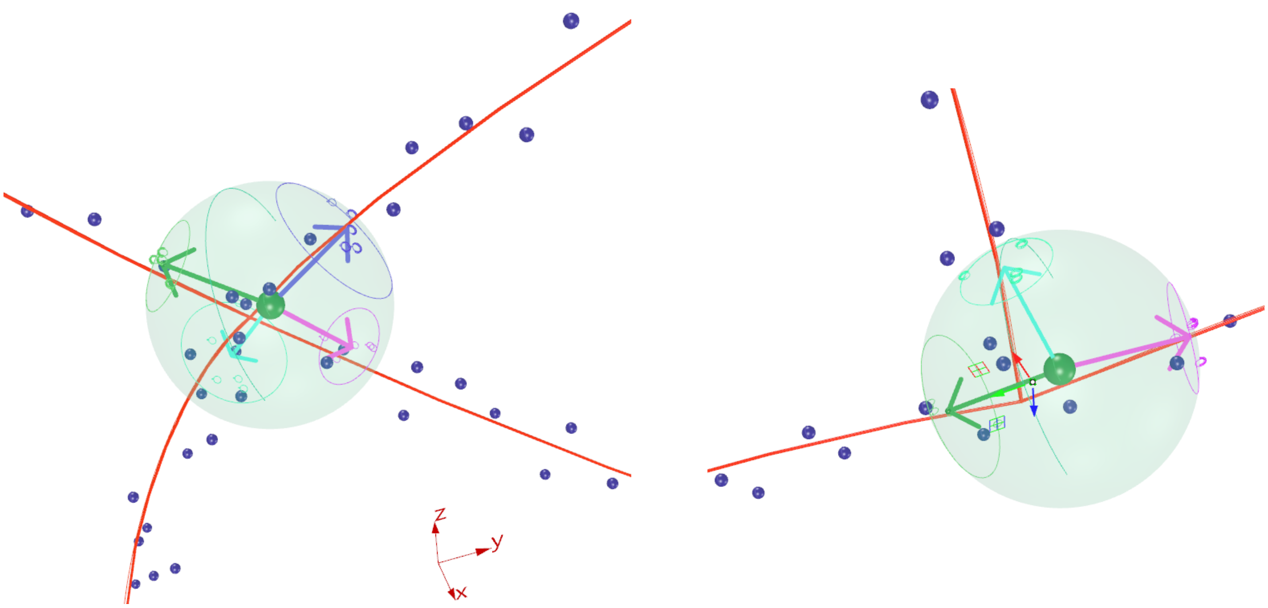


Figure 5: The Trend method is effective for processing point clouds with branching routes.

Core Algorithm The method works by identifying local trend directions within spherical neighborhoods. For each selected point in the cloud, a local group of neighbors is defined within a radius based on the point cloud's resolution. These neighboring points are then projected onto a virtual sphere centered at the arithmetic mean of the local group using central projection (see Figure 4: Left).

Clusters of projected points on the sphere, each containing no fewer than N neighbors, indicate stable directions of local Trends (Figure 4: Right and Figure 5). By iteratively advancing along these trend directions, the method can trace entire branches of the point cloud, effectively constructing a trend line that approximates the skeleton of the spatial data.

It is important to note that the trend line itself is not a final approximating curve. Rather, it serves as a preliminary structure that can be refined using one of the additional approximation methods proposed in this study. Notably, the Gravity method is well suited for further refinement following Trend extraction.

Application Scope The Trend method is well suited for processing closed-loop distributions, varying-density point sets, and complex hybrid structures involving interconnected cells, branches, or intersections. Its general applicability makes it a cornerstone technique in the proposed suite of approximation tools, particularly when working with point clouds obtained from cultural heritage or architectural scanning environments.

3.2 Point-averaging Methods

The essence of these methods is rooted in generating an output sequence of points by calculating the average position of local point groups. The output is a new set of discrete points. Supplementary data processing iterations can change the density of output points and then obtain a set of points on the approximation curve.

3.2.1 The Gravity Method

Initially developed for refining results from Trend method extraction, the Gravity method also performs effectively as a standalone tool for sketch-level approximation and rapid form estimation. The Gravity method constructs a skeletal approximation by computing Centroids of local neighborhoods within the point cloud. It acts as a precise refinement tool, complementing the Trend method by better anchoring the curve within the cloud's spatial structure. This hybrid approach overcomes a common shortcoming of many approximation techniques: curve shrinkage in areas of high curvature or sharp turns. By combining Trends with Centroid-based smoothing, the method improves both accuracy and geometric fidelity.

Principle of the Method The Gravity method is based on the definition of the desired point as a weighted mean of neighboring points (Figure 6: Left). The number of neighbors depends on the width or density of the input set of points.

$$c_0 = \frac{\sum_i^N c_i k_i}{\sum_i^N k_i}$$

c_i – coordinates of points, k_i – weight coefficient , N – number of neighboring points.

The weight coefficients are subjective and are assigned by the expert of the studied problem or by the user of the method.

Applicability and Flexibility The Gravity method applies in diverse contexts:

- Ordered point sets.
- Unordered sets with known movement direction (e.g., from prior Trend method analysis).
- Fully unordered sets by using stochastic sampling of local neighborhoods.
- Sets with variable width or density of points.

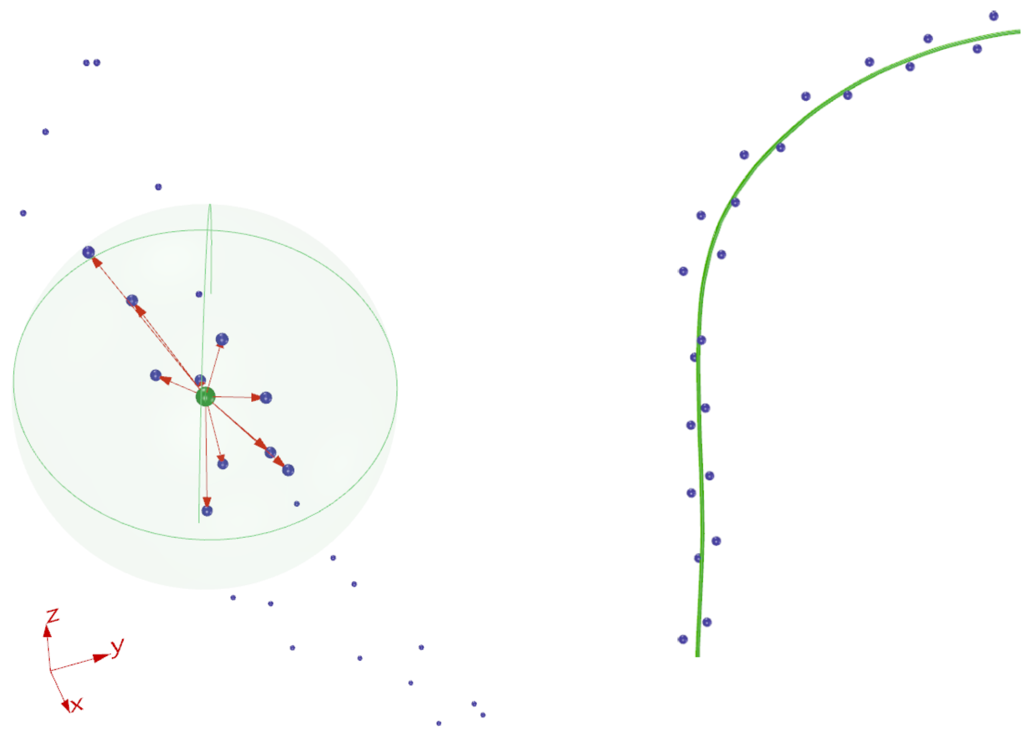


Figure 6: Left: The Gravity method computes the desired point as the weighted centroid of nearby points. Right: The resulting curve traces the centerline of the point cloud.

Algorithmic Principle Each point is computed as the weighted mean of its neighbors (Figure 6: Left). The number of neighbors is determined by local density or the estimated width of the point cloud. The method generalizes to N-dimensional data [11, 12], making it applicable across various domains.

Limitations and Mitigation The method's main limitation is boundary degradation: endpoints tend to vanish with repeated iterations. This is mitigated by extrapolating and reinserting them after each step. While this adds some computational overhead, the problem is controllable. Furthermore, applying spline interpolation to the obtained sequence of points results in smooth curves with high geometric accuracy, since gravity points are typically located along a naturally smoothed trajectory (Figure 6: Right).

4 Comparative Analysis of Curve Approximation Methods

The effectiveness of the approximation methods presented in this study was evaluated using experiments on synthetically generated point clouds (Section 2), enabling a controlled comparison of their geometric approximation performance (see Figure 7). To quantitatively assess method efficiency, the noise reduction percentage $d\%$ was calculated. The point clouds were randomly generated based on a known target curve, which served as the ground truth for comparison. The noise reduction percentage was computed using the following expression:

$$d = \left(1 - \frac{G}{B}\right) \times 100$$

where:

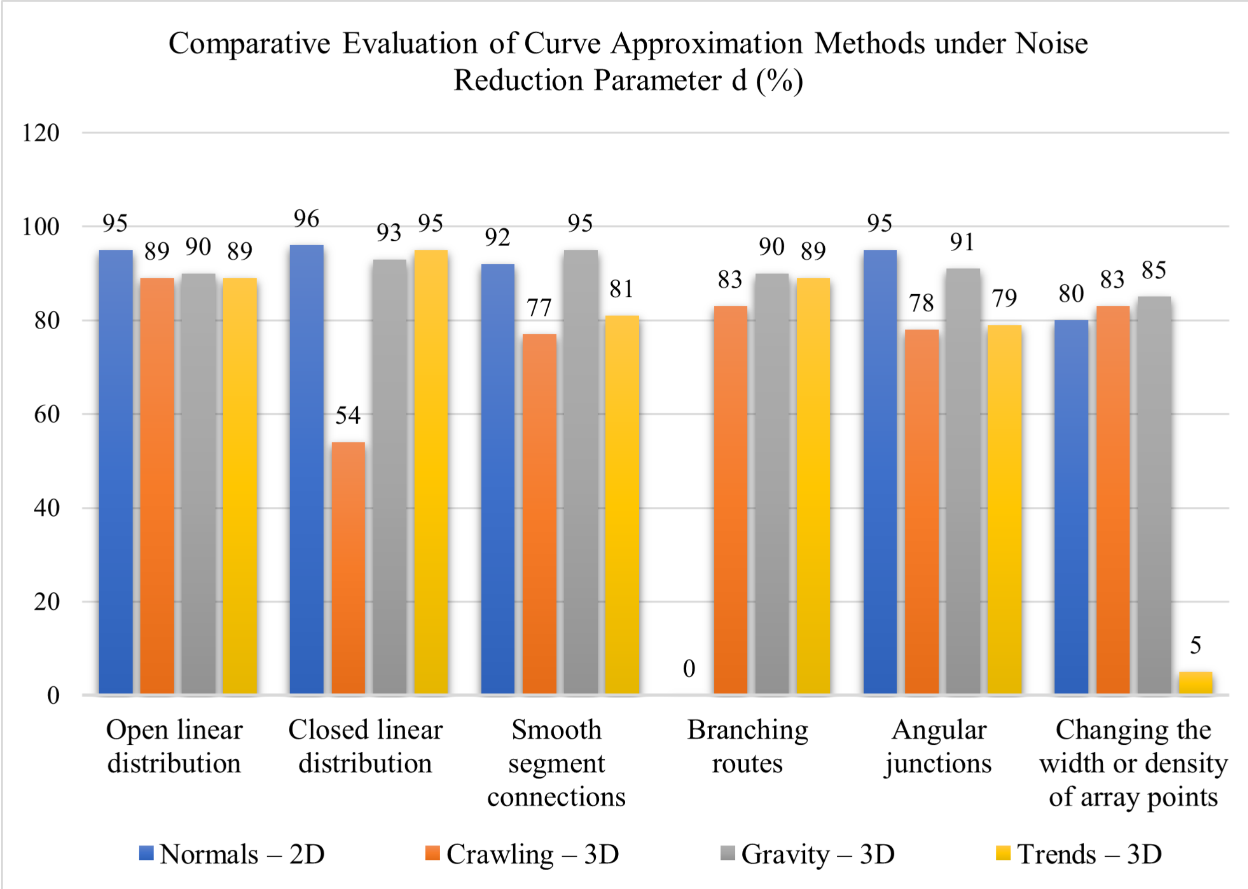


Figure 7: Comparative Study of Curve Approximation Techniques:
Normals – 2D Normal Vector Method (5 iterations);
Crawling – 3D Crawling Method (2 iteration);
Gravity – 3D Gravity Method (3 iterations, unordered set of points);
Trends – 3D Trend Method (1 iteration for each local point group)

- G – is the mean absolute deviation of the processed points (obtained after applying the smoothing/fitting method) from the reference curve;
- B – is the mean absolute deviation of the initial (noisy) randomly generated points from the same reference curve.

A higher value of $d\%$ indicates a more effective noise reduction relative to the original noisy input.

All procedures were implemented to operate in real-time. The comparative analysis confirms that the proposed methods achieve a satisfactory level of reconstruction accuracy while maintaining optimal computational efficiency and resource consumption. All methods were implemented and tested within the Rhino-Grasshopper software environment. This platform supports the effective and precise modeling of complex geometric forms and proved well-suited for the development and visualization of the approximation techniques described.

To assess the performance and applicability of the proposed curve approximation methods, a comparative analysis was conducted across different types of point cloud distributions typically encountered in HBIM workflows. The evaluation considered the following criteria:

- Topological complexity (e.g., branching, intersections).

- Geometric characteristics (e.g., high curvature, varying density).
- Computational efficiency.
- Resulting curve smoothness and fidelity.

* The Normal Vector method assumes an ordered input sequence. The author is currently developing a topology-driven extension to support complex nonlinear distributions, including three-dimensional ones.

** The Crawling method struggles with closed, high-curvature shapes due to local compression artifacts.

*** Gravity method yields 54% smoothing by default, improved to 95% with weighting coefficient adjustment.

**** Gravity method underperforms (5%) for clouds with highly variable point density unless paired with pre-structuring techniques (e.g., Trend method).

***** Trend method delivers consistently high-quality smoothing for both simple and topologically complex point sets.

5 Conclusions and Discussion

Some general points about smoothing approximation methods are important:

Trade-off between accuracy and computational efficiency: Smoothing approximation methods often involve finding a balance between the features of the estimated function or surface and the computational resources required. For example, to find 2D curves approximating plane sections, such as the outline of an arch or other planar section, it is sufficient to apply the computationally simple and highly interactive method of normals.

Choice of smoothing parameters: Most smoothing methods involve the selection of parameters that control the amount of smoothing applied to the data. These parameters determine the balance between fitting the data closely and capturing the trends (e.g., the trends method).

Impact of noise and outliers: Point data sets are often subject to noise or contain outliers, which can distort the underlying patterns or introduce distortions in the smoothed approximation. To reduce these problems, a combination of two smoothing methods should be considered, e.g., in the first step, processing the data to identify a trend in the distribution of the set of points. In the second step, processing the obtained smoothing results using the gravity method, which can significantly reduce the noise of the data but requires a stable density set of input points. Furthermore, the gravity method has a higher interactivity rate for organized sets of points.

Visualization and interpretation: Smoothing approximation methods are often employed to generate visually appealing representations of the data or to facilitate data interpretation. However, the interpretation of the smoothed results should consider the limitations of the chosen method and the assumptions made during the smoothing process.

Application-specific considerations: The choice of a smoothing approximation method should be guided by the specific requirements and characteristics of the application domain. Various methods may be more suitable for certain types of data or specific objectives.

In conclusion, the choice of smoothing approximation methods should consider the trade-off between accuracy and computational efficiency, the impact of noise and outliers, visualization requirements, and application-specific needs. Selecting appropriate smoothing parameters and assessing the quality of the approximation are crucial steps in the process. Ultimately, the effectiveness of the smoothing method will depend on its ability to capture the underlying patterns in the data.

Finally, there are a few points about the perspective branches' application of smoothing approximation methods: smoothing in image processing, geospatial analysis and mapping, financial modelling and forecasting, data visualisation and exploratory analysis, signal processing and sensor data analysis. In summary, the ability of smoothing approximation methods to reveal underlying trends, reduce noise, interpolate data, and facilitate data exploration makes them invaluable tools in various domains. At the same time, the acceptability of the result depends on the validly selected method. To solve this problem, the author proposes this research.

Acknowledgment

The author gratefully acknowledges the Volkswagen Foundation for supporting this research as part of a fellowship granted to Larysa Ivanova. Sincere thanks are extended to Prof. Dr.-Ing. Daniel Lordick for supervising the internship and for providing valuable suggestions that significantly improved this paper. The author also thanks Dipl.-Ing. Daria Dordina for her consultation on the use of Grasshopper software.

References

[1] E. Altantsetseg, Y. Muraki, K. Matsuyama, and K. Konno: *Feature line extraction from unorganized noisy point clouds using truncated Fourier series*. The Visual Computer **29**, 617 – 626, 2013. doi: 10.1007/s00371-013-0800-x.

[2] A. Bobenko: *Advances in Discrete Differential Geometry*. 2016. ISBN 978-3-662-50446-8. doi: 10.1007/978-3-662-50447-5.

[3] A. Bucksch, R. Lindenbergh, and M. Menenti: *SkelTre: Robust skeleton extraction from imperfect point clouds*. The Visual Computer **26**, 1283–1300, 2012. doi: 10.1007/s00371-010-0520-4.

[4] M. Capone and E. Lanzara: *Scan-to-BIM vs 3D ideal model HBIM: parametric tools to study domes geometry*. ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XLII-2/W9**, 219–226, 2019. doi: 10.5194/isprs-archives-XLII-2-W9-219-2019.

[5] V. Croce, G. Caroti, A. Piemonte, and M. G. Bevilacqua: *From survey to semantic representation for Cultural Heritage: the 3D modeling of recurring architectural elements*. ACTA IMEKO **10**, 98, 2021. doi: 10.21014/acta imeko.v10i1.842.

[6] J. Daniels, L. Ha, T. Ochotta, and C. Silva: *Robust Smooth Feature Extraction from Point Clouds*. 123–136. 2007. doi: 10.1109/SMI.2007.32.

[7] F. Diara and F. Rinaudo: *Open source HBIM for Cultural Heritage: A project proposal*. The International Archives of the Photogrammetry, Re-

mote Sensing and Spatial Information Sciences **XLII-2**, 303–309, 2018. doi: 10.5194/isprs-archives-XLII-2-303-2018.

[8] C. Dore: *Procedural Historic Building Information Modelling (HBIM) For Recording and Documenting European Classical Architecture*. 2017. doi: 10.21427/D7045G. https://api.semanticscholar.org/CorpusID:64688779.

[9] fokus GmbH Leipzig and Scan3D GmbH Berlin and TU Dresden AG Geometric Modelling and Visualization and ZAFT: *freeform4BIM: Modeling of freeform surfaces in point clouds to generate BIM models of existing objects*. Project in the GEOkomm cooperation network twin4bim, 2022-2025. https://www.zaft-dresden.de/news/76/freeform4BIM/.

[10] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen: *L1-medial skeleton of point cloud* **32**(4), 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461913.

[11] L. Ivanova: *General principles for analyzing multidimensional data models*. Science, engineering, and technology: global and current trends 113, 2019. https://lib.udau.edu.ua:8443/server/api/core/bitstreams/af3a6974-a83e-47af-91e0-f1d0b08cfc46/content.

[12] L. Ivanova: *Geometric Models for Multidimensional Databases*. Applied geometry and engineering graphics (97), 51–59, 2020. doi: 10.32347/0131-579x.2020.97.51-59.

[13] L. Ivanova: *Investigation of economic processes by methods of discrete differential geometry*. Applied geometry and engineering graphics (99), 101–109, 2020. doi: 10.32347/0131-579x.2020.99.101-109.

[14] M. Ludwig, G. Herbst, D. Rieke-Zapp, R. Rosenbauer, S. Rutishauser, and A. Zellweger: *The Advantages of Parametric Modelling for the Reconstruction of Historic Buildings. The Example of the in War Destroyed Church of St. Catherine (Katharinenkirche) in Nuremberg*. vol. XL-5/W1. 2013. doi: 10.5194/isprsarchives-XL-5-W1-161-2013.

[15] M. Mansouryar and A. Hedayati: *Smoothing Via Iterative Averaging (SIA)A Basic Technique for Line Smoothing*. International Journal of Computer and Electrical Engineering 307–311, 2012. doi: 10.7763/IJCEE.2012.V4.501.

[16] M. Mansouryar and A. Hedayati: *Smoothing Via Iterative Averaging (SIA)A Basic Technique for Line Smoothing*. International Journal of Computer and Electrical Engineering 307–311, 2012. doi: 10.7763/IJCEE.2012.V4.501.

[17] B. Pengbo, L. Gongning, and W. Kuanquan: *A graph-based method for fitting planar B-spline curves with intersections*. Journal of Computational Design and Engineering **3**(1), 14–23, 2016. ISSN 2288-4300. doi: 10.1016/j.jcde.2015.05.001.

[18] D. Santana-Cedrés, N. Monzón, and L. Álvarez: *An Algorithm for 3D Curve Smoothing*. Image Processing On Line **11**, 37–55, 2021. doi: 10.5201/ipol.2021.292.

[19] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea: *3D Skeletons: A State-of-the-Art Report*. Computer Graphics Forum 2016. ISSN 1467-8659. doi: `10.1111/cgf.12865`.

[20] J. Wang, Z. Yu, W. Zhang, M. Wei, C. Tan, N. Dai, and X. Zhang: *Robust reconstruction of 2D curves from scattered noisy point data*. Computer-Aided Design **50**, 27–40, 2014. ISSN 0010-4485. doi: `10.1016/j.cad.2014.01.003`.

[21] H. Xu, N. Gossett, and B. Chen: *Knowledge and heuristic-based modeling of laser-scanned trees*. ACM Trans. Graph. **26**, 2007. doi: `10.1145/1289603.1289610`.

Received June 18, 2025; final form August 4, 2025.