

Received: 03 October, 2025

Accepted: 10 October, 2025

Published: 11 October, 2025

***Corresponding author:** Houman Borouchaki, GAMMA3, CTAM, University of Technology of Troyes, 12 rue Marie Curie - CS 42060, Troyes, 10004, France, E-mail: houman.borouchaki@utt.fr

Keywords: Implicit function; 3D reconstruction; Lidar data; Point clouds; Mesh

Copyright License: © 2025 Li Y, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

<https://www.mathematicsgroup.us>



Check for updates

Review Article

Implicit Function-based 3D Reconstruction for Point Cloud Data

Yusen Li^{1,2}, Houman Borouchaki^{1,2*}, Hanlin Miao¹ and Jie Zhang²

¹GAMMA3, University of Technology of Troyes, 12 rue Marie Curie - CS 42060, Troyes, 10004, France

²CTAM, University of Technology of Troyes, 12 rue Marie Curie - CS 42060, Troyes, 10004, France

Abstract

3D reconstruction from point cloud data has become a key component in various domains such as computer graphics, medical imaging, industrial design, and virtual reality. Among the many available approaches, implicit function methods have attracted significant attention due to their robustness and their ability to generate high-quality surface meshes from sparse and noisy data. This paper investigates the fundamental principles of 3D reconstruction, with a particular focus on the application of space-partitioned local fitting methods in implicit surface reconstruction from point clouds. Furthermore, we introduce a hybrid normal estimation and orientation technique to enhance global surface consistency. Experimental results on LiDAR point clouds demonstrate the accuracy and efficiency of the proposed reconstruction pipeline, validating its effectiveness.

Introduction

With the rapid advancement of 3D scanning technologies, point cloud data [1,2] has emerged as a crucial medium for representing three-dimensional spatial information. They are widely used in various domains, including computer graphics, medical imaging, industrial design, and virtual reality [3,4]. Due to their high precision, density, and rich geometric details, point clouds constitute a fundamental resource for 3D reconstruction research [5-8]. Nevertheless, the efficient and accurate reconstruction of surfaces from point cloud data remains a central challenge [9-11]. Over the years, a variety of reconstruction techniques have been developed to address the difficulties associated with structured, unstructured, noisy, or incomplete point clouds. These techniques are commonly classified according to their underlying representations and reconstruction strategies. The principal categories include mesh-based, explicit surface-based, volumetric, and point-based approaches, each characterized by distinct advantages and trade-offs, depending on the properties of the input data and the specific application requirements.

Mesh-based reconstruction methods

Mesh-based approaches reconstruct 3D surfaces by connecting discrete points into polygonal meshes. Among the most widely adopted techniques are the Delaunay triangulation and the marching cubes algorithm. The Delaunay triangulation method generates a triangulated surface by maximizing the minimum angle of triangles, thereby reducing numerical instability. It is particularly well-suited for structured and dense point clouds and ensures the production of manifold surfaces. However, it performs poorly with noisy, sparse, or incomplete data and is highly sensitive to outliers. The marching cubes algorithm is a voxel-based method that extracts isosurfaces from volumetric data. It is computationally efficient for structured datasets, such as CT or MRI scans, and produces smooth surfaces. Nevertheless, it requires the construction of a volumetric grid, which substantially increases memory consumption and limits scalability.

Explicit surface-based methods

These approaches represent surfaces directly through parametric functions or piecewise polynomial approximations

[12,13]. Two widely used methods are Non-Uniform Rational B-Splines (NURBS) and the Bézier method. NURBS construct smooth, continuous surfaces using control points and basis functions, offering precise shape control. NURBS are widely applied in computer-aided design and industrial design; however, they require a predefined topology and involve complex fitting procedures, particularly for large-scale or noisy datasets. Bézier and B-Spline Surfaces provide mathematical representations for smooth surface modeling, yielding high-quality results in computer-aided design. However, their adaptability is limited when applied to unstructured point clouds or datasets with missing information.

Volumetric methods

These approaches define surfaces within a volumetric domain, typically employing implicit functions or level-set formulations. Two widely used methods are Poisson Surface Reconstruction and Level Set Methods. Poisson Surface Reconstruction formulates surface reconstruction as the solution to a Poisson equation, generating an implicit function that approximates the input point cloud. It is robust to noise and incomplete data and produces smooth, watertight surfaces; however, it is computationally intensive and requires accurately defined point normals. Level Set Methods represent evolving surfaces using a signed distance function, naturally accommodating topological changes. Despite this flexibility, these methods are computationally intensive and sensitive to initialization.

Point-based methods

Unlike explicit surface construction, point-based methods represent and render 3D models directly from point clouds. This method defines a smooth surface by computing a locally weighted polynomial approximation. It is robust to noise and well-suited for scattered data; however, it becomes computationally expensive when applied to large datasets.

Traditional surface reconstruction [14] methods face limitations in managing complex topologies, noisy data, and large-scale point clouds [15–17]. Therefore, the development of approaches that can effectively capture intricate geometric structures while remaining robust to noise is of considerable importance.

In recent years, implicit function-based 3D reconstruction methods have gained increasing attention [3,18,19]. These methods represent 3D surfaces implicitly through mathematical functions (such as signed distance functions and radial basis functions), eliminating the need for explicit mesh topology [5]. This allows them to naturally handle complex topological structures and partially missing point cloud data. Additionally, implicit representations can inherently smooth out noise effects, improving reconstruction quality and making them well-suited for various application scenarios, such as computer graphics, medical imaging, industrial design, virtual reality, and augmented reality [3,20].

In this study, we propose a novel approach for geometric reconstruction based on implicit functions. The methodology

consists of two main components. First, surface normals are estimated at each point to ensure consistent surface orientation. Subsequently, a smooth implicit function is constructed in the local neighborhood of each point using the estimated normals. By combining these local implicit functions, voxel data at multiple levels can be utilized to reconstruct the overall geometry. Compared with existing methods, the proposed approach produces smoother geometries across varying levels of detail and demonstrates strong robustness to noise.

Section 2 presents the methodology for normal estimation, while Section 3 details the formulation of local implicit functions. Section 4 provides illustrative examples demonstrating the efficiency of the proposed method. Finally, Section 5 concludes the study and discusses potential directions for future research.

Normal estimation of point clouds and normal orientation consistency

Normal estimation of point clouds based on covariance matrix

Normal estimation is a critical component of point cloud processing, playing a fundamental role in 3D reconstruction, surface fitting, feature extraction, and classification. Normal information describes the local geometric properties of point clouds and is essential for applications such as surface rendering, mesh reconstruction, and point cloud registration.

The covariance matrix-based method is a classical and efficient approach to normal estimation, relying on the statistical distribution of local neighborhoods. By applying Principal Component Analysis (PCA) to the covariance matrix, the normal direction is identified as the eigenvector associated with the smallest eigenvalue. Owing to its mathematical simplicity and computational efficiency, this method has been widely adopted in point cloud processing.

(1) Local Neighborhood Search: To estimate the normal at a point p_i , its local neighborhood $N(p_i)$ must be identified. Common strategies for neighborhood selection include:

k-Nearest Neighbors (k-NN): Identifies the k closest points to p_i , based on Euclidean distance.

Fixed Radius Search: Selects all points within a given radius r centered at p_i .

Let the neighborhood of p_i contain n points, denoted as $N(p_i) = \{p_1, p_2, \dots, p_n\}$.

(2) Covariance Matrix Computation: In point cloud normal estimation, the covariance matrix is typically constructed with respect to a reference point. This reference can be either the current point under consideration or the centroid (mean) of its neighborhood, computed as :

$$\bar{q} = \frac{1}{n} \sum_{i=1}^n q_i = \left(\frac{1}{n} \sum_{i=1}^n q_{xi}, \frac{1}{n} \sum_{i=1}^n q_{yi}, \frac{1}{n} \sum_{i=1}^n q_{zi} \right) \quad (1)$$

The covariance matrix is then computed as:

$$C = \frac{1}{n} \sum_{i=1}^n q_i (q_i - \bar{q})(q_i - \bar{q})^T \quad (2)$$

Where q_i represents the coordinates of the neighboring points. The covariance matrix C is a 3×3 symmetric matrix:

$$C = \begin{bmatrix} C_{xx} & C_{xy} & C_{xz} \\ C_{yx} & C_{yy} & C_{yz} \\ C_{zx} & C_{zy} & C_{zz} \end{bmatrix} \quad (3)$$

Each element C_{uv} is calculated as:

$$C_{uv} = \frac{1}{n} \sum_i^n (u_i - \bar{u})(v_i - \bar{v}) \quad (4)$$

where u, v represents the x, y, z coordinates.

(3) Eigenvector computation: Perform Eigen Decomposition on the covariance matrix C , obtaining eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and their corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, where:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \quad (5)$$

The smallest eigenvalue λ_3 corresponds to the eigenvector $\mathbf{v}_3$, which represents the least variance direction in the local neighborhood. This vector is chosen as the estimated normal:

$$\mathbf{n}_i = \mathbf{v}_3 \quad (6)$$

Normal orientation consistency

In point cloud normal estimation, particularly with covariance matrix-based methods, the computed normals inherently exhibit directional ambiguity, both $\mathbf{n}_i$ and $-\mathbf{n}_i$ are valid normal at a point $\mathbf{p}_i$. Without resolving this ambiguity, inconsistent normal directions can cause significant errors in surface reconstruction, rendering, or registration. Therefore, enforcing consistent normal orientation across the point cloud is essential.

This paper adopts a hybrid strategy that combines local consistency with neighborhood-based propagation. Specifically, a set of seed points is selected from the point cloud, these are typically representative points. The normal directions of these seed points serve as the initial references for propagation. From each seed point, normal directions are locally propagated to neighboring points based on a directional consistency criterion.

To ensure the initial consistency of normal orientations at the seed points, which serve as the starting nodes for orientation propagation, this paper adopts a method based on quadratic surface fitting to refine and correct their normal directions.

Local quadratic surface fitting method: To accurately capture the local geometric characteristics of a point cloud, we adopt a local coordinate-based quadratic surface fitting approach centered at a given point. This method involves establishing a local frame aligned with the surface normal and fitting a second-order surface to the nearest neighboring points in that frame, as shown in Figure 1. The procedure is detailed as follows:

1) Selection of Neighboring Points

Given a query point $\mathbf{P} \in \mathbb{R}^3$ with a corresponding unit surface normal vector $\mathbf{N} \in \mathbb{R}^3$, we identify its k nearest neighbors from a global point set $Q = \{q_i\}_{i=1}^n \in \mathbb{R}^3$, using Euclidean distance as the metric. In this paper, we set $k = 6$.

2) Construction of Local Coordinate System

A local right-handed orthonormal coordinate system $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ is established at $\mathbf{P}$, where:

$\mathbf{w} = \mathbf{N}$ defines the local z -axis direction;

$\mathbf{u}$ is obtained by normalizing the cross product of $\mathbf{N}$ with an arbitrary non-collinear vector $\mathbf{t}$, $\mathbf{u} = \frac{\mathbf{t} \times \mathbf{N}}{\|\mathbf{t} \times \mathbf{N}\|}$, $\mathbf{v} = \mathbf{u} \times \mathbf{w}$. The resulting rotation matrix $\mathbf{R} = [\mathbf{u}^T \ \mathbf{v}^T \ \mathbf{w}^T]^T \in \mathbb{R}^{3 \times 3}$ transforms global coordinates into the local frame. For each neighboring point q_j , the local coordinate q_j^L is computed as:

$$q_j^L = \mathbf{R}(q_j - \mathbf{p}) \quad (7)$$

3) Quadratic Surface Fitting in the Local Frame

After transformation, the neighboring points are represented in the local frame as $q_j^L = (x_j, y_j, z_j)$. We assume the local surface (as shown in Figure 1) near $\mathbf{P}$ can be approximated by a second-order polynomial of the form:

$$z = ax^2 + 2bxy + cy^2 \quad (8)$$

To estimate the coefficients a, b, c we construct a linear system using the transformed coordinates:

$$\mathbf{Z} = \mathbf{A}\mathbf{X} \quad (9)$$

where:

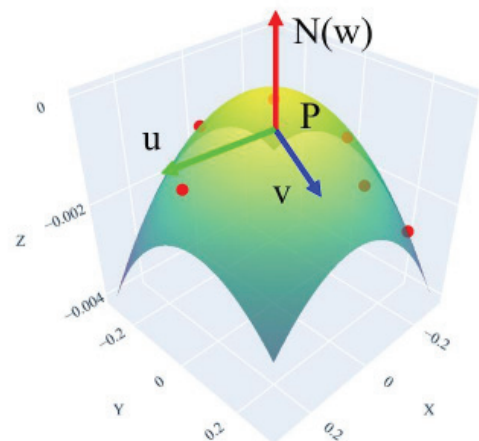


Figure 1: Local Coordinate-Based Quadratic Surface Fitting.

$$\mathbf{Z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_6 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} x_1^2 & 2x_1y_1 & y_1^2 \\ x_2^2 & 2x_2y_2 & y_2^2 \\ \vdots & \vdots & \vdots \\ x_6^2 & 2x_6y_6 & y_6^2 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \quad (10)$$

The coefficients are then estimated via the least-squares solution:

$$\mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Z} \quad (11)$$

Normal consistency by local quadratic surface: To ensure the initial consistency of the normal orientations at the seed points, the entire point cloud is first enclosed within a uniform volumetric grid of resolution $50 \times 50 \times 50$. Each grid vertex is then classified as lying either outside or inside the surface: vertices outside are assigned a positive sign (“+”), while those inside are assigned with a negative sign (“-”), as shown in Figure 2a). This initial classification provides a coarse signed distance field used for orientation refinement in subsequent steps.

For each grid vertex P located near the surface boundary, the closest point Q on the input point cloud is identified, as illustrated in Figure 2b). A local coordinate system is constructed at Q , with the local z -axis is aligned to the normal vector N at that point. Within this local frame, the quadratic surface defined in Equation 8, is fitted to the neighboring points. This fitted surface serves as a smooth approximation of the local geometry near Q . Using the fitted coefficients a, b, c , the function is evaluated at the offset points P^+ and P^- , which are positioned slightly along the positive and negative directions of the normal vector, respectively:

$$\psi(P) = ax_p^2 + 2bx_py_p + cy_p^2 \quad (12)$$

If the evaluated function Ψ at either offset point yields a sign inconsistent with the expected orientation $\psi(P^+) < 0$ or $\psi(P^-) > 0$, then the normal vector is flipped as $\mathbf{N} = -\mathbf{N}$, and the coefficients are adjusted accordingly $a = -a, b = b, c = -c$.

This correction ensures that the fitted surface's normal direction points consistently outward from the shape, resolving ambiguities in surface orientation near thin or concave regions. After obtaining a reliable and consistently oriented normal at the selected seed points using quadratic surface fitting, the method proceeds to propagate the corrected normal directions to the rest of the point cloud using a neighborhood-based orientation propagation algorithm.

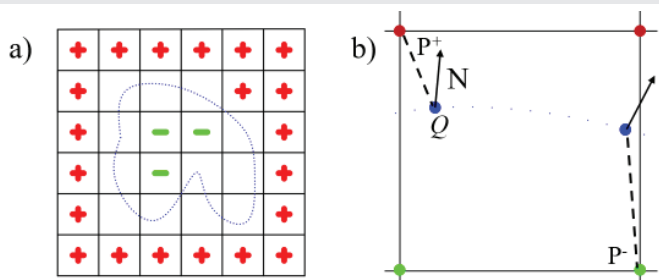


Figure 2: a) Inside and outside vertex classification in a volumetric grid. b) Closest point method for normal orientation.

In this process, each seed point acts as a local root, and its normal direction is propagated to neighboring points in a breadth-first manner. At each propagation step, if the dot product between the current point's normal and that of its neighbor is negative, the neighbor's normal is flipped to align with the propagation direction.

By employing multiple seed points and performing propagation in parallel across the dataset, this method ensures global orientation consistency while avoiding issues such as direction drift or error accumulation that may arise from using a single-root traversal.

The results of the normal orientation correction are shown in Figure 3. As shown in Figure 3a), the initial normal vectors are inconsistent, with some pointing inward and others outward. After applying the proposed method, a globally consistent normal field is obtained. In Figure 3b), the red vectors represent the normal that were already correctly oriented, while the green vectors denote those that were originally inverted but have been successfully corrected to point outward. Figure 3c), shows the result obtained using a traditional method. It can be observed that in regions with complex geometric structures, such as corners and edges, the normal vectors remain noticeably disordered and lack global consistency. Specifically, the normals in the upper part of the model tend to point inward, while those in the lower part point outward, further highlighting the limitations of traditional methods in achieving globally consistent normal orientation. This demonstrates the effectiveness of the proposed strategy in producing coherent and meaningful normal directions across the entire point cloud.

Implicit function-based 3D reconstruction techniques

The space-partitioned local fitting approach provides an effective compromise between efficiency and accuracy in implicit surface reconstruction from point clouds. By dividing the domain into local regions and fitting implicit functions independently, it reduces computational complexity, supports parallelization, and adapts well to non-uniform and noisy data. This enables the method to capture fine geometric details while maintaining global surface consistency. Consequently, we adopt this approach to achieve robust and efficient reconstruction in large-scale point cloud scenarios.

An implicit surface is defined by an analytical equation $F(x, y, z) = C$ where (x, y, z) belongs to a volume domain and C is a constant called “level” and the corresponding surface is a level set. A discrete representation of this surface is the data values of the underlying function, $F(x_i, y_j, z_k)$, at the vertices (x_i, y_j, z_k) of a 3D (Cartesian) grid, $1 \leq i \leq n_i$, $1 \leq j \leq n_j$ and $1 \leq k \leq n_k$ where (n_i, n_j, n_k) are three integers defining the resolution of the grid, also implicitly representing the boundaries and so the (discrete) definition volume of the surface. Such a grid is often called a “voxel” grid, as shown in Figure 4.

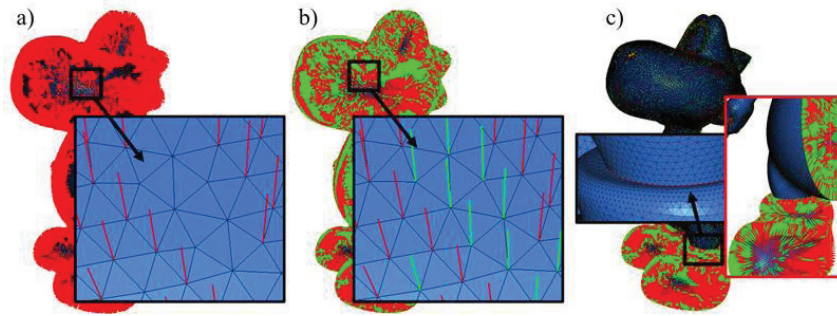


Figure 3: Normal reorientation: a) Inconsistent Initial Normals; b) Reoriented Normals; c) Traditional Method Result.

To compute function values $F(x_i, y_j, z_k)$ at each grid vertex P , we adopt a local approximation strategy based on point cloud data. For each vertex P , the m nearest neighbors $\{Q_i\}_{i=1}^m$ in the point cloud are identified (in this paper $m = 3$). Around each Q_i , a local quadratic surface is fitted:

$$\varphi_i(x, y) = a_i x^2 + b_i xy + c_i y^2 \quad (13)$$

where x and y are coordinates of P in a local coordinate system centered at Q_i . The distance from vertex $P = (x, y, z)$ to the surface Q_i is computed as:

$$d(P, \varphi_i) = z - a_i x^2 - b_i xy - c_i y^2 \quad (14)$$

The function value at P is then obtained through a weighted combination of these distances:

$$F(x_i, y_j, z_k) = \sum_{i=1}^m \omega_i \cdot d(P, \varphi_i) \quad (15)$$

The weight ω_i for each surface is determined by the inverse distance magnitude, normalized by a harmonic normalization factor H :

$$\omega_i = \frac{1}{|d(P, \varphi_i)| \cdot H}, \quad H = \sum_{i=1}^m \frac{1}{|d(P, \varphi_i)|} \quad (16)$$

This formulation gives greater influence to surfaces closer to the point P , and ensures smooth interpolation across the grid.

Figures 5,6 illustrate this process. Figure 5 shows the topological relation between a grid vertex P and its neighboring point samples $\{Q_i\}$, while Figure 6 visualizes the local quadratic surface φ_i constructed at a point Q_i along with the projection of P onto the surface.

Applications

In this section, the proposed 3D reconstruction algorithm is applied to several point cloud datasets. For each case, we present the point cloud, the estimated normals, and the 3D reconstructed model. In certain examples, fine-resolution grids are employed to better capture geometric details during reconstruction.

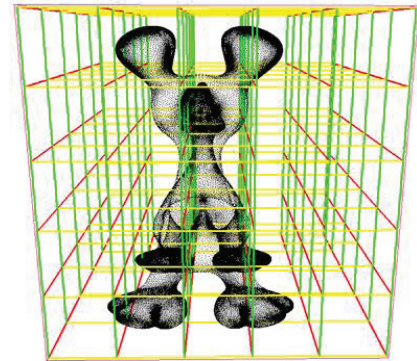


Figure 4: A 3D Cartesian grid enclosing the point cloud model.

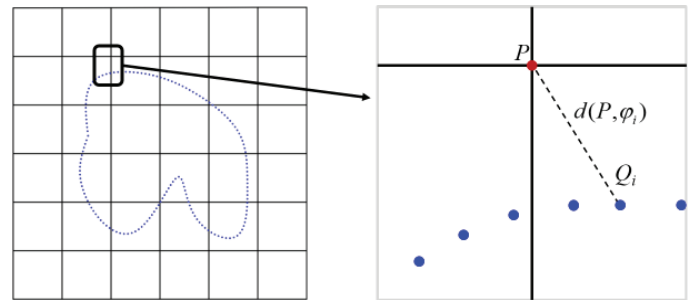


Figure 5: Schematic of voxel grid and local point sample geometry.

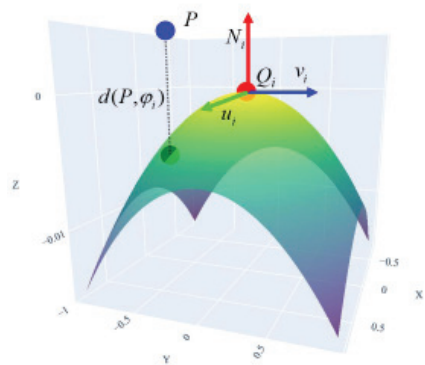


Figure 6: Visualization of the local quadratic surface φ_i near Q_i and orthogonal distance from grid point P .

The implicit surfaces are converted into 3D meshes using the PLOUGH3D software. The PLOUGH3D software generates the triangulation of a given level set of the implicit surface

from a voxel grid. In addition, the software also generates the corresponding level curves in each plane $z = z_k$. Several features are also available for optimization of the shape quality, simplification, and smoothing (roughness removal) of the resulting triangulation. The software also produces a closed surface of a given thickness around the imposed level. This kind of triangulation is particularly useful for surface 3D printing. The software includes several processing phases:

- Analysis of the grid (minimum and maximum values and initialization of the parameter thresholds deduced from these values).
- Generation of the vertices of the triangulation on the edges of a 3D grid triangulation.
- Generation of the triangles of the triangulation.
- Extraction of the topology of the triangulation.
- Optimization of the shape quality of the triangulation.
- Simplification of the triangulation.
- Extraction of the connected components of the triangulation.
- Low frequency smoothing.
- Loading into memory and writing the resulting triangulation.

Figure 7 shows the point cloud data (from 18,667 points to 1,198,896 points), Figure 8 shows the corresponding normal estimation, and Figure 9 shows the reconstructed models.

Using fine grids allows us to capture details on the surface (Figures 10–12).

As demonstrated by the examples above, the proposed approach produces smoother geometric reconstructions across different levels of detail compared to existing methodologies. For illustration, Figure 13 presents a comparison between a reconstructions obtained using the Poisson method and one generated by our proposed approach.

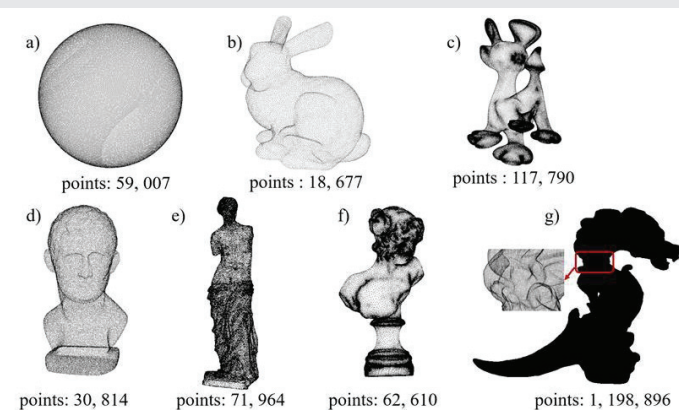


Figure 7: Point clouds.



Figure 8: Normal estimation.

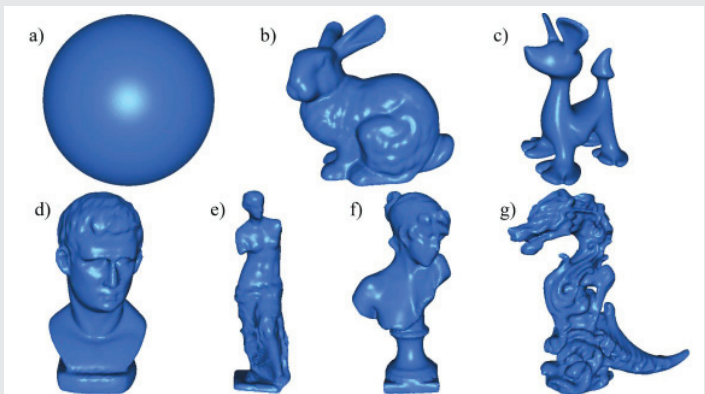


Figure 9: 3D reconstructed models.

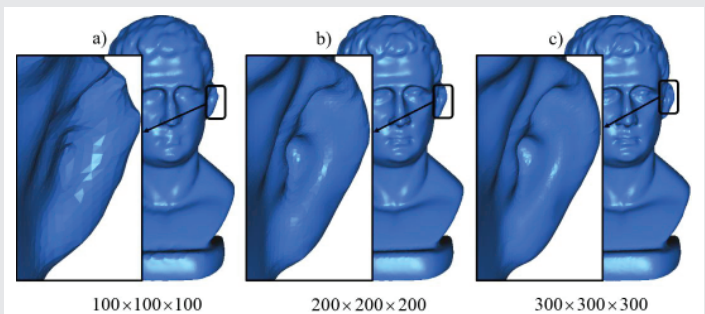


Figure 10: Reconstructed models for 100×100×100, 200×200×200, and 300×300×300 grids.

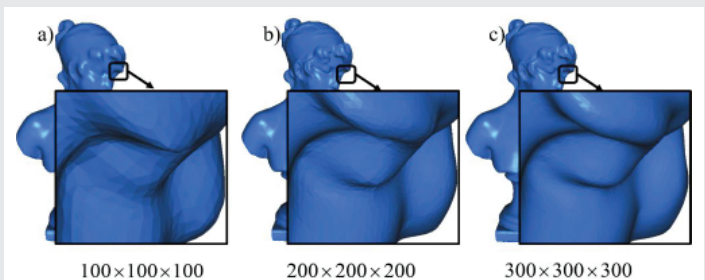


Figure 11: Reconstructed models for 100×100×100, 200×200×200, and 300×300×300 grids.

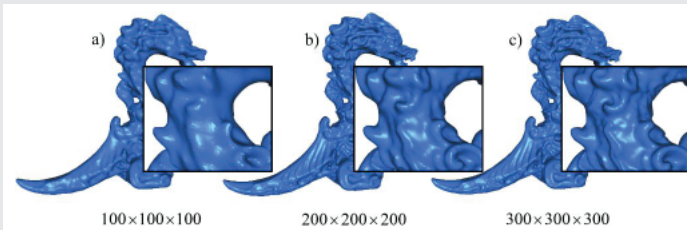


Figure 12: Reconstructed models for 100×100×100, 200×200×200, and 300×300×300 grids.

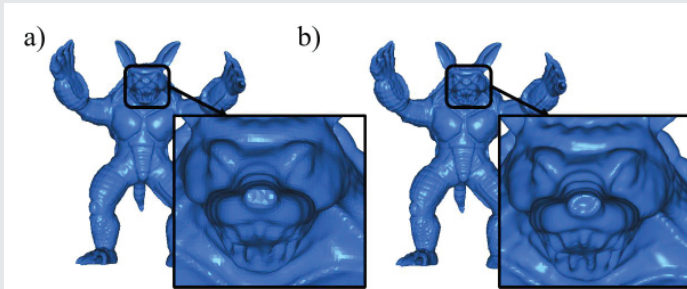


Figure 13: Comparison with Poisson reconstruction.

Conclusion

In this paper, we propose a novel methodology for constructing discrete 3D models from point cloud data, based on implicit surface reconstruction. The approach addresses two key challenges: normal estimation and implicit surface definition. Several examples were presented to demonstrate the effectiveness and efficiency of the method.

Future extensions of this work include the incorporation of sharp edge features, where multiple normals must be considered, and the use of adaptive 3D grids for model construction. The latter would enable the capture of fine geometric details independently of dataset size, further enhancing the scalability and accuracy of the reconstruction process.

References

- Huang X, Mei G, Zhang J, Abbas R. A comprehensive survey on point cloud registration. *arXiv [Preprint]*. 2021; arXiv:2103.02690. Available from: <https://doi.org/10.48550/arXiv.2103.02690>
- Zhang T, Yuan H, Qi L, Zhang J, Zhou Q, Ji S. Point cloud Mamba: Point cloud learning via state space model. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2025;39:10121–10130. Available from: <https://doi.org/10.1609/aaai.v39i10.33098>
- Chibane J, Alldieck T, Pons-Moll G. Implicit functions in feature space for 3D shape reconstruction and completion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020;6970–6981. Available from: <https://doi.org/10.1109/CVPR42600.2020.00700>
- Choy CB, Xu D, Gwak J, Chen K, Savarese S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In: *Computer Vision – ECCV 2016: 14th European Conference on Computer Vision; 2016 Nov 11–14; Amsterdam, The Netherlands. Proceedings, Part VIII*. Springer; 2016;628–644. Available from: <https://arxiv.org/abs/1604.00449>

- Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, et al. Reconstruction and representation of 3D objects with radial basis functions. In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 2001; p. 67–76. Available from: <https://doi.org/10.1145/383259.383266>
- Nguyen TT, Nguyen QM, Liu XG, Ziggah YY. 3D object model reconstruction based on laser scanning point cloud data. In: *International Symposium on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences*. 2012. Available from: https://www.researchgate.net/publication/306278163_3D_OBJECT_MODEL_RECONSTRUCTION_BASED_ON_LASER_SC...
- Li X, Wang Y, Liu Y, Gao X, Song B. Qiyun Pagoda 3D model reconstruction based on laser cloud data. *Surv Mapp*. 2011;9:11–14.
- Boulch A. ConvPoint: Continuous convolutions for point cloud processing. *Comput Graph*. 2020;88:24–34. Available from: <https://doi.org/10.48550/arXiv.1904.02375>
- Brunetaud X, De Luca L, Janvier-Badosa S, Beck K, Al-Mukhtar M. Application of digital techniques in monument preservation. *Eur J Environ Civ Eng*. 2012;16(5):543–556. Available from: <https://doi.org/10.1080/19648189.2012.676365>
- Kwak E, Datchev I, Habib A, El-Badry M, Hughes C. Precise photogrammetric reconstruction using model-based image fitting for 3D beam deformation monitoring. *J Surv Eng*. 2013;139(3):143–155. Available from: [https://doi.org/10.1061/\(ASCE\)SU.1943-5428.0000105](https://doi.org/10.1061/(ASCE)SU.1943-5428.0000105)
- Ren P, Wang W, Bai X. Reconstruction of Imperial University of Shanxi based on virtual reality technology. *Comput Simul*. 2012;29(11):20–23.
- Wang P, Wang Z, Xin S, Gao X, Wang W, Tu C. Restricted Delaunay triangulation for explicit surface reconstruction. *ACM Trans Graph*. 2022;41(5):1–20. Available from: <https://doi.org/10.1145/3533768>
- Lei H. OffsetOPT: Explicit surface reconstruction without normals. *arXiv [Preprint]*. 2025; arXiv:2503.15763. Available from: <https://doi.org/10.48550/arXiv.2503.15763>
- Berger M, Tagliasacchi A, Seversky LM, Alliez P, Guennebaud G, Levine JA, et al. A survey of surface reconstruction from point clouds. *Comput Graph Forum*. 2017;36(1):301–329. Available from: <https://doi.org/10.1111/cgf.12802>
- Ham H, Wesley J, Hendra H. Computer vision-based 3D reconstruction: A review. *Int J Electr Comput Eng*. 2019;9(4):2394–2402. Available from: <https://doi.org/10.11591/ijece.v9i4.pp2394-2402>
- Huang Z, Wen Y, Wang Z, Ren J, Jia K. Surface reconstruction from point clouds: A survey and a benchmark. *IEEE Trans Pattern Anal Mach Intell*. 2024. Available from: <https://doi.org/10.1109/tpami.2024.3429209>
- Lim SP, Haron H. Surface reconstruction techniques: A review. *Artif Intell Rev*. 2014;42(1):59–78. Available from: <https://link.springer.com/article/10.1007/s10462-012-9329-z>
- Zhao HK, Osher S, Fedkiw R. Fast surface reconstruction using the level set method. In: *Proceedings of the IEEE Workshop on Variational and Level Set Methods in Computer Vision*. 2001; p. 194–201. Available from: <https://doi.org/10.1109/VLSM.2001.938900>
- Kazhdan M, Hoppe H. Screened Poisson surface reconstruction. *ACM Trans Graph*. 2013;32(3):1–13. Available from: <https://doi.org/10.1145/2487228.2487237>
- Liu P, Zeng Z, Wang Q, Chen M, Zhang G. Geometry-enhanced implicit function for detailed clothed human reconstruction with RGB-D. *arXiv [Preprint]*. 2025; arXiv:2503.17921. Available from: <https://doi.org/10.1049/cit2.70009>