AIMS Mathematics, 9(11): 32491–32513.
DOI: 10.3934/math.20241557
Received: 30 July 2024
Revised: 24 September 2024
Accepted: 16 October 2024
Published: 15 November 2024

https://www.aimspress.com/journal/Math

Research article

Curvature estimation for point cloud 2-manifolds based on the heat kernel

Kai Wang[1], **Xiheng Wang**[2] **and Xiaoping Wang**[1,*]

[1] College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China

[2] Wuhan Research Institute of Posts and Telecommunications, Wuhan 430074, China

* **Correspondence:** Email: levine@nuaa.edu.cn.

Abstract: The geometry processing of a point cloud 2-manifold (or point cloud surface) heavily depends on the discretization of differential geometry properties such as Gaussian curvature, mean curvature, principal curvature, and principal directions. Most of the existing algorithms indirectly compute these differential geometry properties by seeking a local approximation surface or fitting point clouds with certain polynomial functions and then applying the curvature formulas in classical differential geometry. This paper initially proposed a new discretized Laplace-Beltrami operator by applying an inherent distance parameter, which acts as the foundation for precisely estimating the mean curvature. Subsequently, the estimated mean curvature was taken as a strong constraint condition for estimating the Gaussian curvatures, principal curvatures, and principal directions by determining an optimal ellipse. The proposed methods are mainly based on the heat kernel function and do not require local surface reconstruction, thus belonging to truly mesh-free methods. We demonstrated the correctness of the estimated curvatures in both analytic and non-analytic models. Various experiments indicated that the proposed methods have high accuracy. As an exemplary application, we utilized the mean curvature for detecting features of point clouds.

Keywords: automatic fiber placement; variable stiffness trajectory; point representation; meshless method; first-order shear deformation theory
Mathematics Subject Classification: 53A45, 53A70

1. Introduction

With the advancement of 3D acquisition technology, the point-based representation of complex objects and environmental point clouds has been extensively utilized in various shape modeling, graphical rendering, and engineering design and manufacturing applications, as cited in [1]. With the growth of available data, significant attention has been given to analyzing the original point clouds.

Due to the uncertainty of the exact shape represented by point clouds, curvature estimation, including the two principal curvatures and their related directions, mean curvature and Gaussian curvature, has numerous applications in computer aided design, computer graphics, and other associated fields, and it involves many applications such as surface segmentation (as mentioned in [2–4]), point cloud simplification [5], point cloud registration [6], surface reconstruction [7, 8], and feature extraction [9, 10]. More generally, a simple and accurate curvature estimation method for point cloud surfaces would ensure reliable numerical behavior for a large number of applications, which is a hot and challenging issue.

In this paper, we presented a series of differential geometry properties for point cloud surfaces based on the heat kernel function. At the core of them is a novel mean curvature normal operator that is based on the discrete Laplace-Beltrami operator derived from the heat kernel function. The proposed method does not need to fit and interpolate point clouds as a local surface to indirectly approximate the geometry properties of point clouds, and it does not need to rely on any local mesh or Voronoi-based map. They only involve the coordinates of point cloud surfaces and are truly mesh-free methods.

1.1. Related works

Before delving into our contributions, we initially reviewed some previous work, confining our exposition to discretization methods rather than surveying the numerous applications in which they are employed. Surprisingly, though, these study topics have witnessed very distinct developments and we reviewed the existing ones separately.

1.1.1. Curvature estimation on meshes

The estimation of curvature properties for the mesh surface have been studied for many years and is a crucial topic [11]. The method commonly applied is to define a concept of curvature for meshes and to study how well they approximate the actual curvature of the underlying surface. Taubin [12] estimated the principal curvatures and directions of a triangular mesh surface based on the Euler formulas. Chen [13] defined some intrinsic surface properties from triangular surface, which is based on the Meusnier and Euler theorem. Dong [14] presented an algorithm to estimate principal curvatures by simplifying the Chen's method. A classical approximating strategy was presented by Meyer [15] and they proposed a unified and consistent set of flexible operators to approximate important geometric attributes, including normal vectors and curvatures (Gaussian, mean, and principal) on any triangular mesh. Goldfeather [16] gave a cubic algorithm to approximate principal direction vectors on mesh surfaces. Goes [17] constructed a class of discrete differential operators for arbitrary polygon mesh. Based on the local approximation of mesh vertices and associated normals by a quadratic surface, Makovník [18] introduced a method to estimate the mean and Gaussian curvature and several related quantities for a polygonal mesh.

1.1.2. Curvature estimation on point clouds

Recently, several methods have been put forward to estimate curvatures and feature information on point cloud surfaces. These schemes can roughly be divided into two categories: Continuous approximation and discrete estimates.

Continuous approximation: These methods are the simplest and give rise to plenty of references.

Their first step typically is to reconstruct the point cloud surface into a continuous implicit or parametric form and then compute curvature attributes with classical curvature formulas in differential geometry. Douros [19] locally fitted the point cloud surface with an analytic representation described by quadric surface patches and then assessed curvature properties at each point. Chen [20] first made use of an implicit quadric surface to fit the point cloud data and then calculated the curvature according to the curvature formulas of the implicit surface. Spek [21] presented a fast method by iteratively fitting a parabolic quadric surface to compute principal curvatures from range images. Cheng [22] fitted the point clouds into a parametric surface to study the point cloud simplification method. Based on principal component analysis, Asao [23] fitted a quadratic hypersurface by minimizing the square error. [24] selected 8 neighbors ring from a point cloud for each point and then they estimated the surface curvatures by fitting the point cloud surface into a quadratic surface.

Discrete estimates: These schemes generally utilize a series of local neighbors of point clouds, such as ball neighbors, K-neighbors, or Voronoi cell, to approximate the original point clouds. Based on an analytic energy function in [25,26], Yang [27] directly computed Gasussian and mean curvatures with the curvature formulas in [28] derived by an implicit function. Similarly, Tian [29] adopted the same energy function and drew a similar conclusion as Yang [27]. Moreover, this energy function referred to in [25,26] was also used by Miao [30] to roughly estimate the curvatures of discrete point clouds. Zhang [31] presented a robust algorithm to estimate principal curvatures and his basic idea was to locally fit each normal section circle. A more detailed introduction can be found in his other study [32]. Wang [33] proposed two different curvature estimation methods with local differential properties called the Voronoi method and MLS projection. Then, Yao [6] applied Wang's method to develop a point cloud registration algorithm. Based on the Voronoi cells, Quentin [34] presented an efficient and robust method for extracting curvature information of point clouds. His method is integral in nature and adopts convolved covariance matrices of Voronoi cells. Lachaud [35] estimated curvature tensor information by generating random triangles. Based on the Weiangarten map, Cao [36] proposed a direct and efficient method to estimate the curvature. He automatically acquired curvatures from the Weingarten map with a least square fitting method. Lange [37] also presented a method for anisotropic fairing of a point-sampled surface by applying an anisotropic geometric mean curvature flow.

1.2. Contributions

Our main intention is to deduce the geometrical properties of point cloud surfaces, including the mean κ_M, Gaussian κ_G, two principal curvatures $\kappa_{1,2}$, and two principal directions $\mathcal{T}_{1,2}$. Overall, our main contributions can be subdivided into three aspects:

(i) A new discretized Laplace-Beltrami operator applying an inherent distance parameter for point cloud surfaces is proposed. This operator serves as the basis for accurately estimating the mean curvature.
(ii) We introduce a new computational method aimed at the area of point cloud surfaces. The provided scheme does not rely on any mesh or Voronoi-based map and can simply transform the complex problem of computing the area into solving a series of linear equations.
(iii) A strong constraint condition is introduced to estimate the principal curvatures and directions by finding an optimal ellipse. The results are better than those in [31, 32] which do not have any constraints.

1.3. Outlines

The remainder of this paper is structured as follows: Based on the work in Appendix A, Section 2 illustrates the concrete contexts for estimating mean curvatures over a sampled point cloud surface, see Algorithm 1. Section 3 mainly discusses a simple method to find two principal curvatures and related principal directions by finding an optimal ellipse. The corresponding Gaussian curvature is acquired instantly. Some pratical experimental cases are presented in Section 4 to demonstrate the efficiency and accuracy of the proposed methods. Some important conclusions are drawn in Section 5.

2. Mean curvature

Our method for estimating the mean curvature can be purely depicted in term of the operations carried out on the smooth point cloud surface $\mathcal{M}$ which is sampled by a series of points $\mathrm{P} = \{\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_n\}$ and embedded within the Euclidean space $\mathbb{R}^3$. Let Δ be the negative semi-definite Laplace-Beltrami operator that acts on the differential real-valued functions $f : \mathcal{M} \to \mathbb{R}$ over the point cloud surface $\mathcal{M}$. The estimation of mean curvatures with the heat kernel functions involves the following fundamental steps.

Algorithm 1 Mean curvature with heat kernel

1: Deduce the Laplace-Beltrami operator Δ_{pc} with h.
2: Solve a linear system $\mathbf{H}\mathcal{A} = \mathbf{b}$ to compute the area $\mathcal{A}$.
3: Acquire the mean curvature $\kappa_M(\mathrm{x}_i) = \frac{1}{2} \cdot \left\| \mathrm{K}_{\mathrm{pc}}(\mathrm{x}_i) \right\|$.

Let us commence by considering the approximation of the Laplace-Beltrami operator Δ_{pc} for the fixed heat diffuse time t over the point cloud surface $\mathcal{M}$. A sufficiently differentiable function $u(\mathrm{x}, t)$ satisfies the heat equation if

$$\frac{\partial}{\partial t} u(\mathrm{x}, t) + \Delta u(\mathrm{x}, t) = 0. \tag{2.1}$$

Given an initial heat distribution $u(\mathrm{x}, 0) = f(\mathrm{x})$, the heat distribution [38–40]

$$u(\mathrm{x}, t) = \mathbf{H}^t f(\mathrm{x}) = \int_{\mathbb{R}^2} f(\mathrm{y}) H^t(\mathrm{x}, \mathrm{y}) d\mathrm{y}$$

is the solution of the heat equation at heat diffuse time t, where the heat kernel function is usually taken as a Gaussian function [41]:

$$H^t(\mathrm{x}, \mathrm{y}) = (4\pi t)^{-1} \cdot e^{-\frac{\|\mathrm{x}-\mathrm{y}\|^2}{4t}}. \tag{2.2}$$

Subsequently, an approximating operator which has been proved to approximate the actual Laplace-Beltrami operator is constructed [42]:

$$\Delta f(\mathrm{x}) = \lim_{t \to 0} -\frac{1}{t} \left((4\pi t)^{-1} \int_{\mathbb{R}^2} e^{-\frac{\|\mathrm{x}-\mathrm{y}\|^2}{4t}} f(\mathrm{y}) d\mathrm{y} - f(\mathrm{x})(4\pi t)^{-1} \int_{\mathbb{R}^2} e^{-\frac{\|\mathrm{x}-\mathrm{y}\|^2}{4t}} d\mathrm{y} \right). \tag{2.3}$$

Based on the formula $d\mathrm{x}_j = \frac{1}{\mathcal{A}(\mathcal{M})} \mathcal{A}(\mathrm{x}_j)$ and Theorem 3.1 in [42], we discretized the above expression using a series of neighbor points over a point cloud surface $\mathcal{M}$ with the empirical version of the

integrals involved:

$$\Delta_{pc} f(\mathrm{x}) = \frac{1}{4\pi t^2} \sum_{j \in B(\mathrm{x})} e^{-\frac{\left\|\mathrm{x}-\mathrm{x}_j\right\|^2}{4t}} \Big(f(\mathrm{x}) - f(\mathrm{x}_j)\Big) \mathcal{A}(\mathrm{x}_j), \tag{2.4}$$

where $B(\mathrm{x})$ is the Ball neighbor of the point $\mathrm{x} \in \mathrm{P}$, it satisfies

$$B_r(\mathrm{x}) = \{\mathrm{q} \in \mathrm{P} \mid \big\|\mathrm{x} - \mathrm{q}\big\| \leq r\},$$

where $r = \alpha \cdot h (3.5 \leq \alpha \leq 4.5)$ is the ball radius (see Figure 1) and it usually remains constant for all points $\mathrm{p} \in \mathrm{P}$ to ensure that the Laplace-Beltrami operator Δ_{pc} is symmetric. Additionally, h is an inherent distance parameter for a given point cloud surface and it will be further discussed in the subsequent subsection. The above approximating operator also can be rewritten into a matrix form

$$\Delta_{pc} f(\mathrm{x}) = \mathbf{L}_{pc}^{t} \cdot \mathbf{f}(\mathrm{x}), \tag{2.5}$$

where the matrix $\mathbf{L}_{pc}^{t} \in \mathbb{R}^{n \times n}$ is a Laplace matrix relative to the heat diffuse time t over a point cloud surface $\mathcal{M}$.

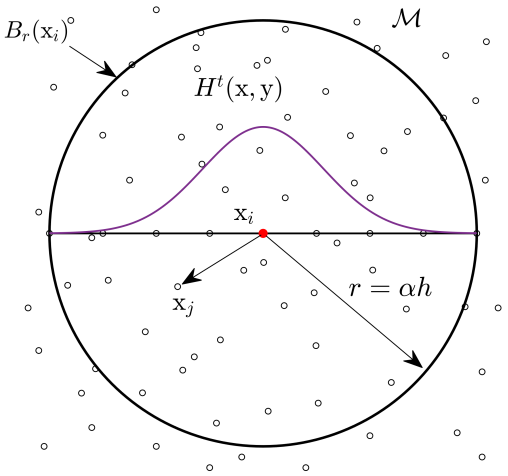


Figure 1. The heat kernel function $H^t(\mathrm{x},\mathrm{y})$ within the two-dimensional problem domain $B_r(\mathrm{x}_i)$ where there is a point cloud surface $\mathcal{M}$.

2.1. Redefining the Laplace-Beltrami operator Δ_{pc} with h

The key to the success of Algorithm 1 lies in finding a stable heat diffuse time t. Eq 2.3 indicates that a smaller heat diffuse time t will yield a better approximation with respect to the Laplace-Beltrami operator Δ_{pc} and area $\mathcal{A}$. However, decreasing the quantity of t does not necessarily enhance the approximating accuracy for a fixed sampled point cloud surface $\mathcal{M}$. Therefore, the motivation for step 1 of Algorithm 1 is to search an optimal heat diffuse time t that is neither too large nor too small. In practice, determining an optimal and stable t is difficult and we here propose a simple searching case that works well, that is, $t = mh^2$, where $m > 0$ is a constant and h is a distance parameter that is highly sensitive to the distribution of the sampled point clouds P.

Typically, to acquire a symmetric Laplace matrix $\mathbf{L}_{pc}^{t}$, h is taken as the mean spacing distance between all points and their nearest ones, which works well in performance for a point cloud surface sampled uniformly and it can be seen as a global distance parameter, that is

$$h = \frac{1}{n}\sum_{i=1}^{n}\min_{j\in B(\mathrm{x})}\{\left\|\mathrm{x}-\mathrm{x}_j\right\|\}. \tag{2.6}$$

But in practice, most of the sampled point cloud surfaces are non-uniform. In this situation, a series of local and self-adaption parameters h for any $\mathrm{x} \in \mathrm{P}$ are available, which are taken as the linear combination of the spacing distance between their nearest point and farthest point among their neighbors, that is

$$h = (1-\gamma)\min_{j\in B(\mathrm{x})}\left\{\left\|\mathrm{x}-\mathrm{x}_j\right\|\right\} + \gamma\max_{j\in B(\mathrm{x})}\left\{\left\|\mathrm{x}-\mathrm{x}_j\right\|\right\}, \tag{2.7}$$

where $\gamma \in [0,1]$ is a linear parameter. That means that the distance parameter h for different points $\mathrm{x}_i \in \mathrm{P}$ does not necessarily have to been taken as the same value as in Eq 2.6. At this time, the K-nearest neighbor is available and it surely causes a non-symmetric Laplace matrix $\mathbf{L}_{pc}^{t}$. Fortunately, this non-symmetric property does not affect the accuracy of Algorithm 1. Moreover, the experimented and convincing constant $m = \frac{1}{4}$ with a high accuracy is adopted (see Appendix A) and then the identity concerning the heat diffuse time t and distance parameter h,

$$\boxed{t = \frac{1}{4}h^2,} \tag{2.8}$$

is available in all examples. Therefore, by discarding the unstable heat diffuse time t, we redefined the approximating Laplace-Beltrami operator $\Delta_{pc}f(\mathrm{x})$ in Eq 2.4 relative to the inherent and consistent distance parameter h referred to in Eqs 2.6 and 2.7 for a given point cloud surface $\mathcal{M}$:

$$\Delta_{pc}^{h}f(\mathrm{x}) = \frac{4}{\pi h^4}\sum_{j\in B(\mathrm{x})} e^{-\frac{\left\|\mathrm{x}-\mathrm{x}_j\right\|^2}{h^2}}\left(f(\mathrm{x})-f(\mathrm{x}_j)\right)\mathcal{A}_j. \tag{2.9}$$

Meanwhile, the rewritten form Δ_{pc}^{h} generates a new Laplace matrix $\mathbf{L}_{pc}^{h} \in \mathbb{R}^{n\times n}$ and its symmetric property depends on weather the global neighbors $B_r(\mathrm{x})$ and distance parameter h are searched.

2.2. Computation of area

The area $\mathcal{A}$, an essential property in the discretized Laplace-Beltrami operator [15], is defined based on the physical attributes associated with the underlying problem. In this paper, only the coordinates of the sampled point clouds P are available and thus area must be estimated to make the discretization process feasible. However, existing point-based Laplace-Beltrami operator techniques, such as [43], resort to the local Voronoi map or tangent spaces at the sampled point clouds P as the basic tool to acquire local areas. In contract, our method from step 2 of Algorithm 1 does not require any auxiliary tool to compute each local area $\mathcal{A}_i$ aiming at point $\mathrm{x}_i \in \mathrm{P}$, which makes it truly meshless. The reasoning is to estimate $\mathcal{A}_i (i = 1, 2, \cdots, n)$ by solving an optimization problem of a linear system derived from the unitary property [41] of heat kernel functions in Eq 2.2, that is

$$\int_{\Omega} H^t(\mathrm{x},\mathrm{y})\mathrm{d}\mathrm{y} = 1.$$

By utilizing the identity in Eq 2.8 and Appendix A, the above integral can be empirically discretized as

$$\frac{1}{\pi h^2}\sum_{j\in B(\mathrm{x})} e^{-\frac{\left\|\mathrm{x}-\mathrm{x}_j\right\|^2}{h^2}}\mathcal{A}_j = 1.$$

More specifically, this property yields a linear system

$$\mathbf{H}\mathcal{A} = \mathbf{b},$$

for all points $\mathrm{x}_i \in \mathrm{P}$, where $H_{ij} = \frac{1}{\pi h^2}e^{-\frac{\left\|x_i-x_j\right\|^2}{h^2}} \geq 0$ is an element of the matrix $\mathbf{H}$ relative to the neighbor point x_j of x_i and $b_i = 1$ is an element of the vector $\mathbf{b}$. The rationality of applying this linear system to approximate the actual local area can be referred to in Appendix A.

2.3. Mean curvature normal

In this subsection, we compute the integral of the mean curvature normal over the point cloud surface $\mathcal{M}$. From step 3 of Algorithm 1, we define an operator that maps a point $\mathrm{x} \in \mathrm{P}$ to the vector

$$\mathrm{K}_{pc}(\mathrm{x}) = 2\kappa_M(\mathrm{x})\cdot\mathbf{n}(\mathrm{x}),$$

where K_{pc} is the mean curvature normal operator and is also an approximation of the Laplace-Beltrami operator for the point cloud surface $\mathcal{M}$, and κ_M is the mean curvature. Based on the Laplacian $\Delta_{u,v}\mathrm{x} = -\mathrm{x}_{uu} - \mathrm{x}_{vv}$ [44], it follows that

$$\int_{\mathcal{A}_\mathcal{M}} \mathrm{K}_{pc}(\mathrm{x})d\mathcal{A} = \int_{\mathcal{A}_\mathcal{M}} \Delta_{u,v}\mathrm{x}d\mathcal{A}.$$

We intended to discretize the interval of the mean curvature normal operator K_{pc}. By considering Eq 2.9, a pratical discretization at a point $\mathrm{x}_i \in \mathrm{P}$ is defined by

$$\mathrm{K}_{pc}(\mathrm{x}_i) = \frac{4}{\pi h^4}\sum_{j\in B(\mathrm{x}_i)} e^{-\frac{\left\|\mathrm{x}_i-\mathrm{x}_j\right\|^2}{h^2}}(\mathrm{x}_i - \mathrm{x}_j)\mathcal{A}_j. \tag{2.10}$$

Based on this formula, we can easily compute the mean curvature κ_M by taking half of the magnitude of this last expression, which is

$$\boxed{\kappa_M(\mathrm{x}_i) = \frac{1}{2}\cdot\left\|\mathrm{K}_{pc}(\mathrm{x}_i)\right\|,} \tag{2.11}$$

where the sign of mean curvature κ_M is determined by the directions of the mean curvature normal K_{pc} and normal vector $\mathbf{n}$. If $\mathrm{K}_{\mathrm{pc}}\cdot\mathbf{n} > \mathbf{0}$, the κ_M is taken as positive. On the contrary, it takes as negative. For the normal vector $\mathbf{n}$, principal component analysis (PCA) [45] is adopted in this paper.

2.4. Mean curvature as a quadrature

To determine the two principal curvatures and directions at a point $\mathrm{x}_i \in \mathrm{P}$, we first show the mean curvature κ_M from Eqs 2.10 and 2.11, which can be interpreted as a quadrature of normal curvatures:

$$\begin{aligned}
\kappa_M(\mathrm{x}_i) &= \frac{1}{2}(2\kappa_M(\mathrm{x}_i)\mathbf{n}) \cdot \mathbf{n} \\
&= \frac{1}{2}\mathrm{K}_{pc}(\mathrm{x}_i) \cdot \mathbf{n} \\
&= \frac{2}{\pi h^4} \sum_{j \in B(\mathrm{x}_i)} e^{-\frac{\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2}{h^2}} (\mathrm{x}_i - \mathrm{x}_j)\mathcal{A}_j \cdot \mathbf{n} \\
&= \frac{1}{\pi h^4} \sum_{j \in B(\mathrm{x}_i)} \left[e^{-\frac{\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2}{h^2}} \left\|(\mathrm{x}_i - \mathrm{x}_j)\right\|^2 \mathcal{A}_j \right] \cdot \kappa_{i,j}^N,
\end{aligned} \tag{2.12}$$

where $\kappa_{i,j}^N$ is defined as

$$\kappa_{i,j}^N = 2\frac{(\mathrm{x}_i - \mathrm{x}_j) \cdot \mathbf{n}}{\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2}.$$

The radius R of the osculating circle passing through two points x_i and x_j can be easily found with the estimated mean curvature in Eq 2.12, see Figure 2. Since we were obliged to have a right angle at the neighbor point x_j of x_i, we have $(\mathrm{x}_i - \mathrm{x}_j) \cdot (\mathrm{x}_i - \mathrm{x}_j - 2R\mathbf{n}) = \mathbf{0}$. This implies

$$R = \frac{\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2}{2(\mathrm{x}_i - \mathrm{x}_j) \cdot \mathbf{n}}.$$

It further proves that $\kappa_{i,j}^N$ is a normal curvature estimated in the direction of edge $\mathrm{x}_i\mathrm{x}_j$. Therefore, Eq 2.11 can be interpreted as a quadrature of normal curvature with weights w_{ij}:

$$\kappa_M(\mathrm{x}_i) = \sum_{j \in B(\mathrm{x}_i)} w_{ij}\kappa_{ij}^N,$$

where $w_{ij} = \frac{1}{\pi h^4} e^{-\frac{\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2}{h^2}} \left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2 \mathcal{A}_j$ is a weight function relative to the edge $\mathrm{x}_i\mathrm{x}_j$.

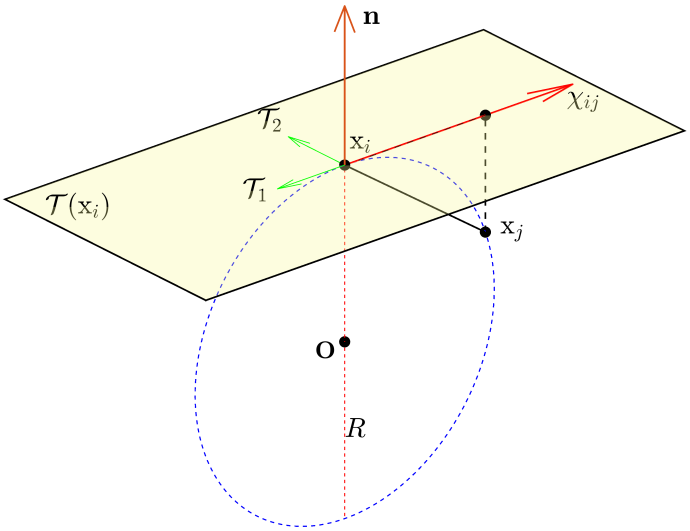


Figure 2. Osculating circle of edge $x_i x_j$.

3. Principal curvature and Gaussian curvature

To calculate the two principal curvatures $\kappa_{1,2}$ and their related principal directions $\mathcal{T}_{1,2}$ on a sampled point cloud surface $\mathcal{M}$, finding the optimal fitting ellipse is adopted in this paper, which has been referred to in [12, 31, 32]. Since the mean curvatures κ_M have been accurately estimated in the previous subsection, we can utilize the property of the mean curvatures involved, that is, $\kappa_1 + \kappa_2 = 2\kappa_M$, as a strong constraint condition to solve the two eigenvalues and eigenvectors of a symmetric curvature tensor matrix $\mathbf{B}$. This matrix is composed of three unknown coefficients a, b, and c:

$$\mathbf{B} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}.$$

According to the eigenvalue properties of matrices as shown in [46], it can be seen that $\kappa_1 + \kappa_2 = a + c$ and $\kappa_1 \cdot \kappa_2 = ac - b^2$. Moreover, the matrix $\mathbf{B}$ can be employed to estimate the normal curvatures in any direction on the tangent plane $\mathcal{T}(x_i)$, as seen in Figure 2. In other words, we have that

$$\chi_{i,j}^T \mathbf{B} \chi_{i,j} = \kappa_{i,j}^N,$$

where $\chi_{i,j} \in \mathbb{R}^2$ is a unit vector relative to the projection $\mathcal{T}_{i,j} \in \mathbb{R}^3$ of the edge $x_i x_j$ on the tangent plane $\mathcal{T}(x_i)$, which is usually acquired by projecting all the points in $B(x)$ onto the tangent plane $\mathcal{T}(x_i)$:

$$\mathcal{T}_{i,j} = \frac{(x_j - x_i) - [(x_j - x_i) \cdot \mathbf{n}] \cdot \mathbf{n}}{\left\| (x_j - x_i) - [(x_j - x_i) \cdot \mathbf{n}] \cdot \mathbf{n} \right\|}.$$

The three unknown coefficients a, b, and c in the matrix $\mathbf{B}$ can be found by minimizing the error function

$$E(a, b, c) = \sum_j w_{ij} \left(\chi_{ij}^T \mathbf{B} \chi_{ij} - \kappa_{ij}^N \right)^2,$$

along with three constraint conditions:

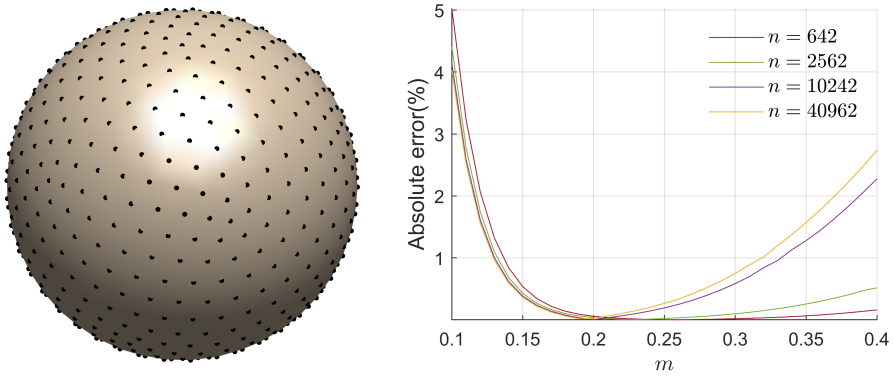
$$\begin{cases} a + c = 2\kappa_M, \\ ac - b^2 \leq \kappa_M^2, \\ ac - b^2 \geq -((\min\{\kappa_{ij}^N\} - \varepsilon) - \kappa_M)^2 + \kappa_M^2, \end{cases} \tag{3.1}$$

where $\varepsilon > 0$ is a tiny constant. Subsequently, the two principal curvatures $\kappa_{1,2}$ of a point $\mathrm{x}_i \in \mathrm{P}$ can be directly acquired through the two eigenvalues of the tensor matrix $\mathbf{B}$. Its two eigenvectors in 2D typically correspond to the actual principal directions $\mathcal{T}_{1,2}$. Eventually, one can directly compute the Gaussian curvature κ_G with the conventional differential geometry formula:

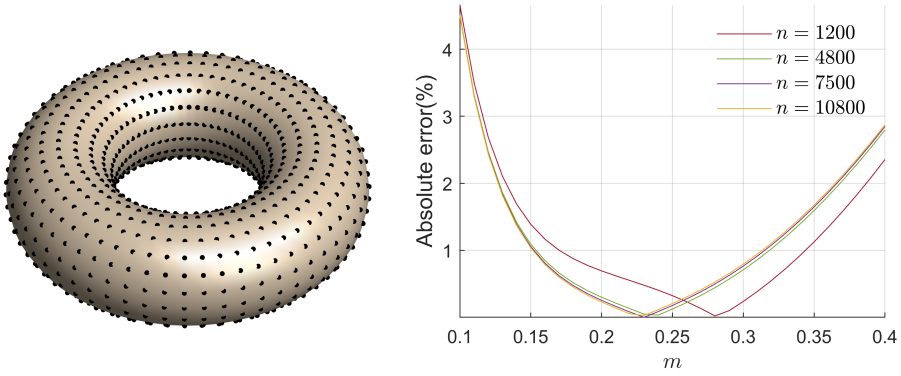
$$\kappa_G = \kappa_1 \cdot \kappa_2.$$

4. Results and applications

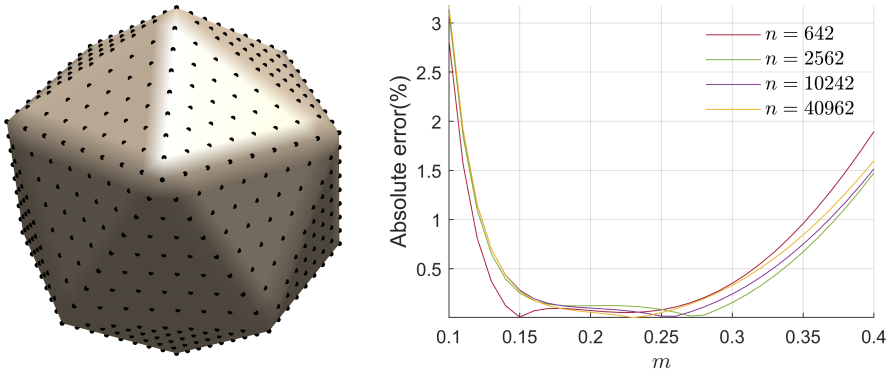
Appendix A has already verified the validity of the mean curvatures with the heat kernel function in some analytic models (like the Saddle, Arch, and Paraboloid) and all the results in Figures 3–6 show that the absolute errors of the mean curvatures κ_M and the areas $\mathcal{A}$ of the point cloud surfaces $\mathcal{M}$ are lower than 1% in the case of $m = 1/4$. Therefore, unless otherwise specified, $m = 1/4$ will be applied in all examples. Moreover, the linear parameter γ in Eq 2.7 belonging to $0.1 \sim 0.5$ will work well in practice in terms of performance. The proposed method can also be adopted to compute mean curvatures on some non-analytic models discretized by large-scale scan point clouds, as seen in Figure 7.



(a) Sphere

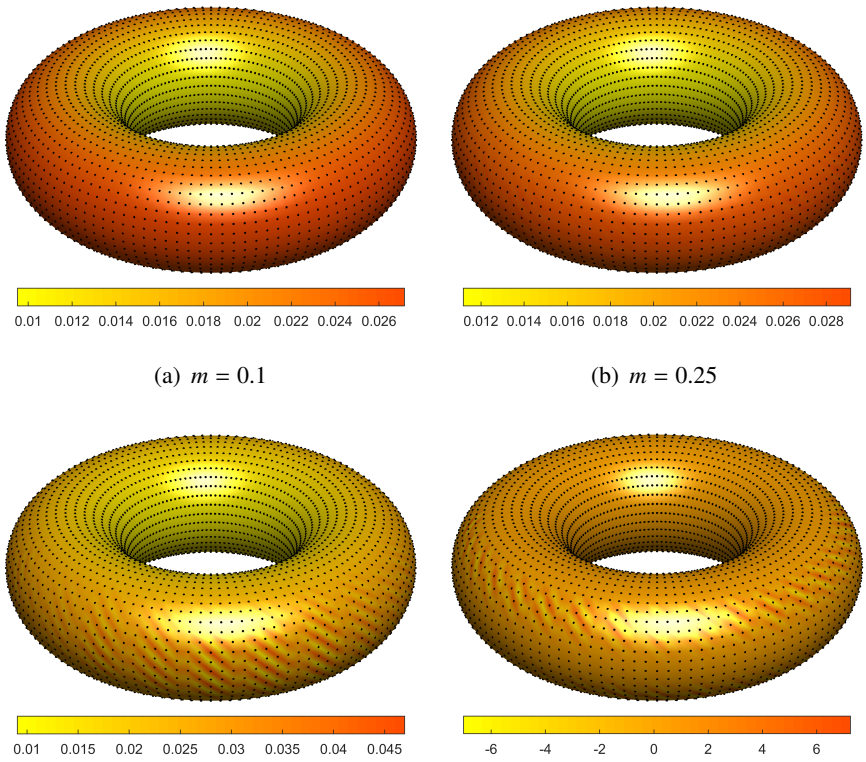


(b) Torus



(c) Icosahedron

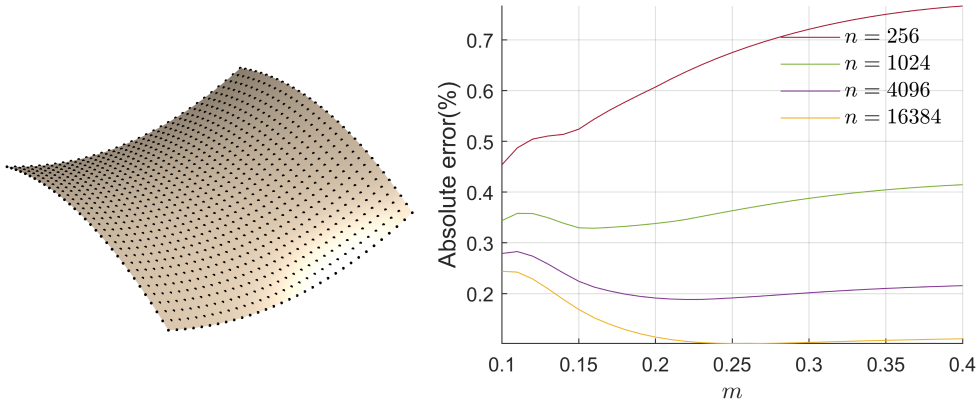
Figure 3. The area absolute errors of the point cloud surfaces $\mathcal{M}$ as a function of the parameter m, where $t = mh^2$. Each error curve corresponds to the model with a different sampled density n. Note that in most cases, $m = 1/4$ approximates the optimal parameter values and possesses an absolute error less than 1%.



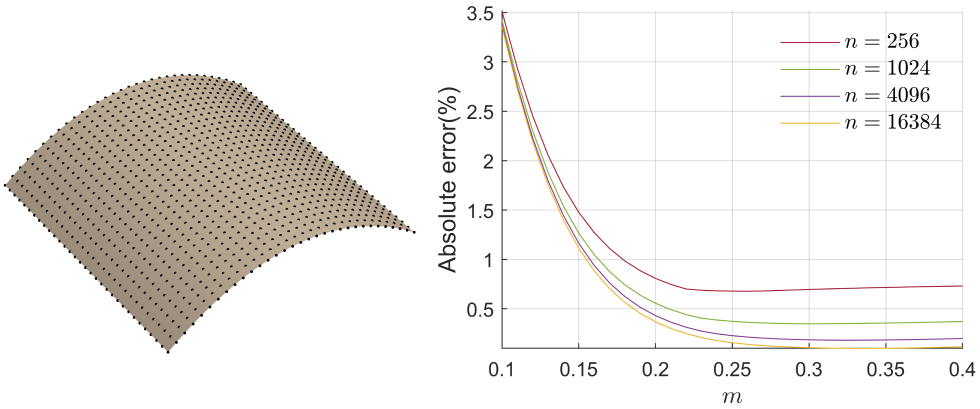
(a) $m = 0.1$ (b) $m = 0.25$

(c) $m = 0.3$ (d) $m = 0.45$

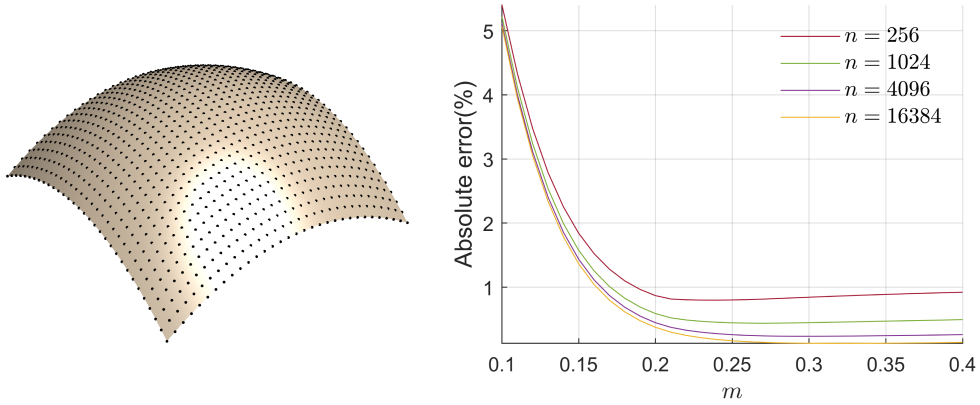
Figure 4. The visualization of the area $\mathcal{A}$ on the Torus model with different parameters m.



(a) Saddle: $z = \frac{x^2-y^2}{16}(-4 \le x, y \le 4)$

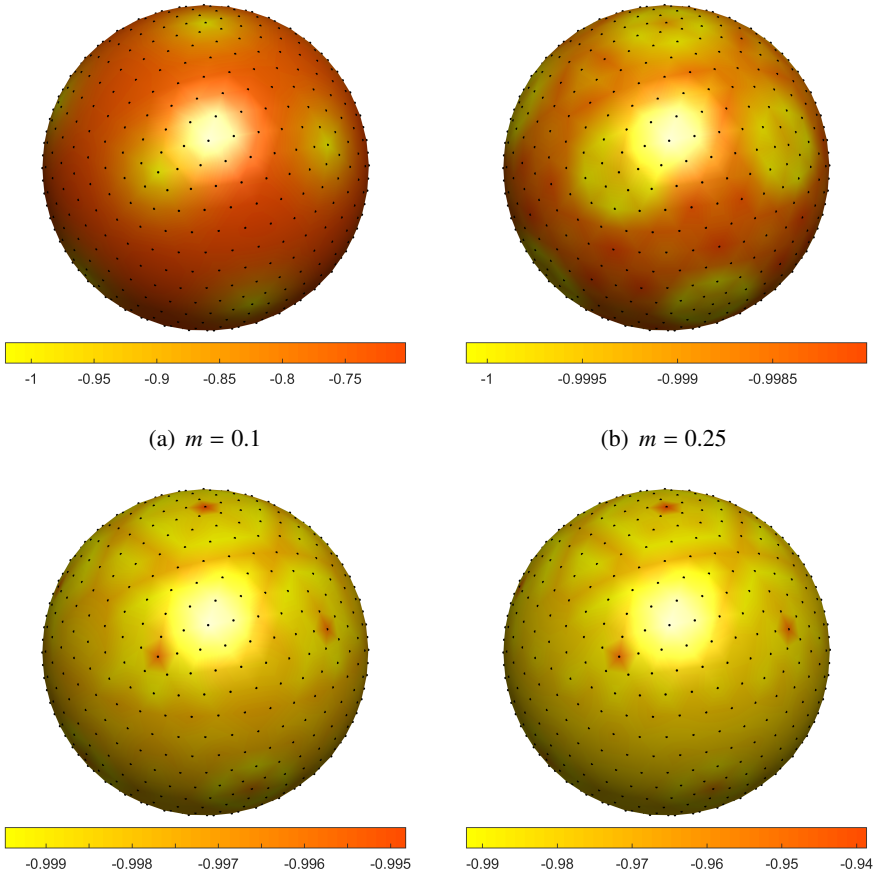


(b) Arch: $z = -\frac{x^2}{10}(-4 \le x, y \le 4)$



(c) Paraboloid: $z = \frac{-x^2-y^2}{10}(-4 \le x, y \le 4)$

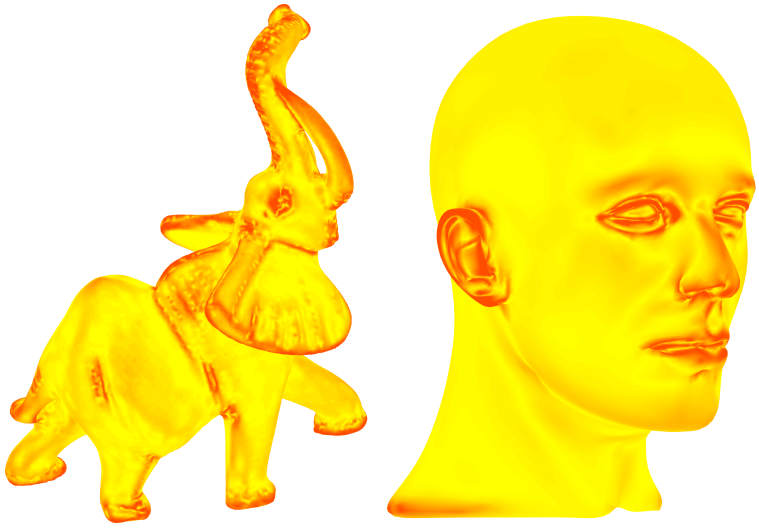
Figure 5. The absolute errors of the mean curvatures of point cloud surfaces $\mathcal{M}$ as a function relative to the parameter m, where $t = mh^2$. Similar to Figure 3, $m = 1/4$ approximates the optimal parameter values and possesses an absolute error below 1% in each error curve.



(a) $m = 0.1$ (b) $m = 0.25$

(c) $m = 0.3$ (d) $m = 0.45$

Figure 6. The visualization of the mean curvature κ_M on the Sphere model with different parameters m. Note that in a unit sphere case, the exact solutions of mean curvatures at every points are -1 (see Example 4.1 in [28]).



(a) Elephant (b) Head

Figure 7. Drawing mean curvatures on non-uniform models.

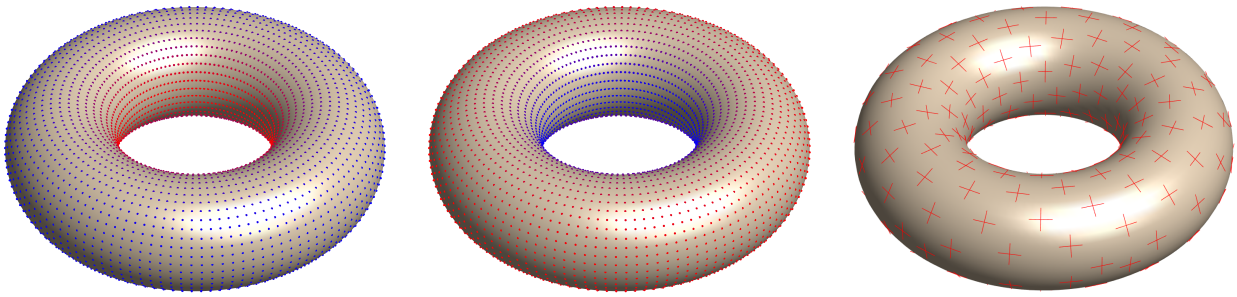
To further prove the validity of the proposed method with the heat kernel function, we tested the Gaussian curvature κ_G, principal curvatures $\kappa_{1,2}$, and their directions $\mathcal{T}_{1,2}$ on some analytic models whose differential properties can be computed exactly with some ready-made curvature formulas. We first sampled a torus function ($\sqrt{x^2+y^2}-r_2)^2+z^2=r_1^2$ with $r_1=1$ and $r_2=2.3701$. It cannot be denied that finding an optimal fitting ellipse directly to acquire the Gaussian curvature κ_G, principal curvatures $\kappa_{1,2}$, and principal directions $\mathcal{T}_{1,2}$, such as [12, 31, 32], is fairly robust. But we have provided the mean curvatures κ_M with a relatively low absolute error (see Figures 5 and 6), which will be adopted to serve as a strong constraint, that is $\kappa_1+\kappa_2=2\kappa_M$ (see Eq 3.1). This constraint is perfectly valid for smooth surfaces, especially closed ones described by point clouds. Figures 8(b) and (c) show the visualized cases for the Gaussian curvature and principal directions in the above sampled Torus model. For the case in this research without the constraint in Eq 3.1, the principle is similar to [12, 31, 32]. Therefore, the comparisions for non-constrained examples are ignored here. But for open ones, we sampled three implicit surfaces and tested the principal directions in Figure 9. As expected, the computed and exact directions match very closely away from the boundary of the point cloud surfaces. Near the boundary of point cloud surfaces, the principal directions estimated by applying the heat kernel function follow the edges of a surface. We here provide another error mechanism aimed at the principal directions $\mathcal{T}_{1,2}$:

$$E=\frac{\sum_{i=1}^{n}\arccos\left(\frac{|\chi_i\cdot\chi_i^*|}{\|\chi_i\|\cdot\|\chi_i^*\|}\right)}{n}. \tag{4.1}$$

This mechanism is mainly obtained from the average included angle between the exact principal direction χ^* and the computed ones χ. Table 1 provides the estimating errors of the principal directions $\chi_{1,2}$ concerning the methods that find an optimal fitting ellipse, which will infer our advantages of possessing a strong constraint $\kappa_1+\kappa_2=2\kappa_M$.

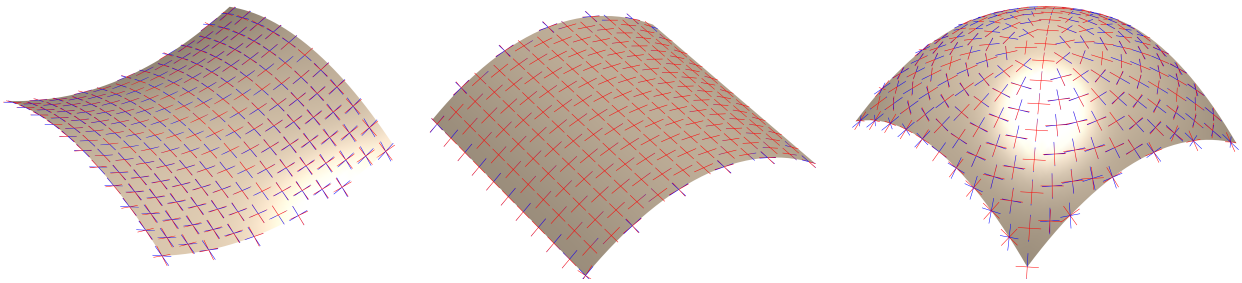
Table 1. Average absolute errors (%) of principal directions using Eq 4.1.

Model Types	n	[31, 32]	Ours
Arch	256	0.7854	0.0080
	1024	0.7854	0.0041
	4096	0.7854	0.0035
	16384	0.7732	0.0031
Saddle	256	0.7795	0.0114
	1024	0.7810	0.0098
	4096	0.7796	0.0081
	16384	0.7788	0.0068
Paraboloid	256	0.8594	0.3735
	1024	0.8541	0.3203
	4096	0.8578	0.3071
	16384	0.8551	0.3038



(a) Mean curvatures κ_M (b) Gaussian curvatures κ_G (c) Principal directions $\mathcal{T}_{1,2}$

Figure 8. The visualization of curvatures with the heat kernel function on a point cloud surface $\mathcal{M}$.

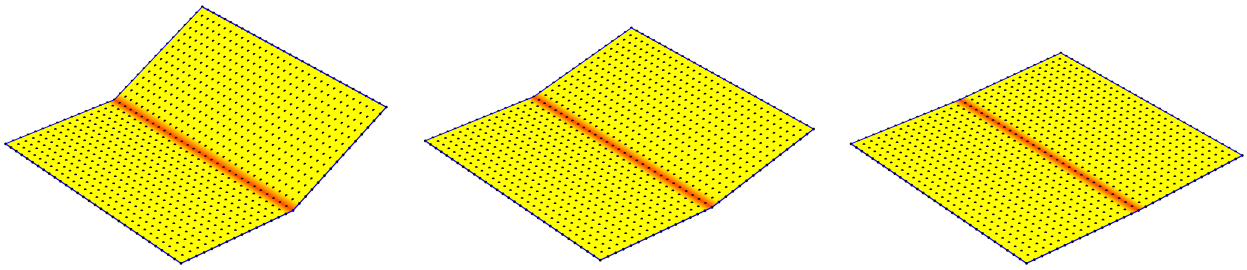


(a) $z = \frac{x^2-y^2}{16}(-4 \le x, y \le 4)$ (b) $z = -\frac{x^2}{10}(-4 \le x, y \le 4)$ (c) $z = \frac{-x^2-y^2}{10}(-4 \le x, y \le 4)$

Figure 9. Implicit surfaces with exact (in blue) and estimated (in red) principal curvature directions $\mathcal{T}_{1,2}$.

Moreover, to demonstrate the advantages of the proposed methods, some classical schemes, classified as continuous approximation and discrete estimates, respectively, are selected as our comparators. Table 2 has already demonstrated with detail the comparisons of the average absolute error for κ_M, κ_1, and κ_G on implicit open cases referred to in Figure 9, such as the Arch, Saddle, and Paraboloid models with different density n. Here, Eq 5.1 is applied. It is not difficult for us to find that the proposed method in the heat kernel has a higher accuracy than the compared references. This is because the references [27, 29] highly depend on a series of precise normal vectors and they should constantly search for an "optimal" parameter μ in approximating the actual model curvatures. Usually, finding a suitable μ is difficult. Besides, the references [31, 32] estimated the point cloud surface properties by finding a fitting ellipse without any constraint. Therefore, they have a bad performance even on some analytic models.

Since the heat kernel method in Algorithm 1 provides a mean curvature normal operator, it basically provides a precise estimating method for mean curvatures, see Eq 2.11. This strategy can also be used to detect sharp edges. In order to better understand the effect of edge sharpness on the quality of feature detection, we sampled a folded square surface made of two planar rectangular patches joined by a common edge, where the density of the point cloud is $n = 961$. Figure 10 illustrates that feature estimation with the mean curvature is available in detecting sharp edges whose interior angle is equal to 1°. Our method is also applicable to large-scale scanning point clouds. As shown in Figure 11, a comparison of different mean curvature methods for feature detection on a large-scale scanning model (block model) is presented. It is not difficult to find that the proposed method also has a good performance in the application of feature detection.

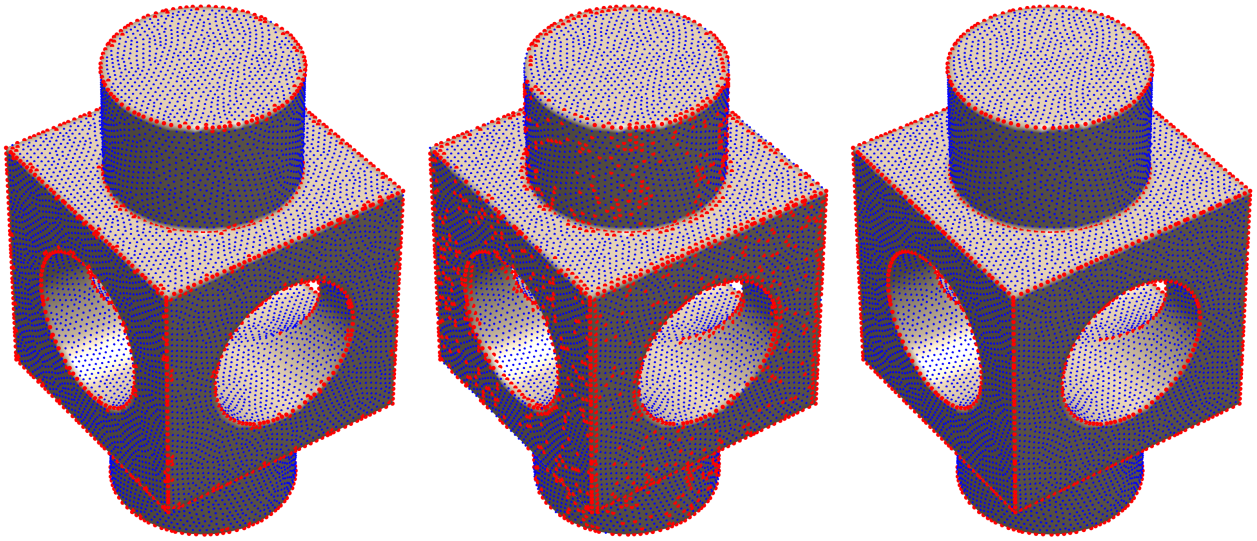


(a) 135° (b) 165° (c) 179°

Figure 10. Estimated curvature feature with the mean curvature in Eq 2.11 on a folded square for different interior angles.

Table 2. Comparisons of average absolute error for κ_M, κ_1, and κ_G on Arch, Saddle, and Paraboloid models with different density n.

Model Types	n	Mean curvature $\kappa_M(\%)$			Principal curvature $\kappa_1(\%)$			Gaussian curvature $\kappa_G(\%)$		
		[27,29]	[31,32]	Ours	[27,29]	[31,32]	Ours	[27,29]	[31,32]	Ours
Arch	256	9.3575	7.9516	0.4009	10.5543	11.2017	0.7997	0.5238	0.3699	0.0052
	1025	9.4890	10.2769	0.2367	10.6986	12.6093	0.4769	0.4926	2.1727	0.0043
	4096	9.6554	14.6644	0.1603	10.9951	18.2447	0.3355	0.4379	15.6367	0.0023
	16384	9.7485	13.2974	0.1228	11.1866	16.6100	0.3214	0.4035	10.1592	0.0011
Saddle	256	0.6642	0.4572	0.6750	0.7908	10.2575	1.4015	0.1480	1.1369	0.1541
	1024	0.3976	0.6222	0.3633	0.4394	10.0183	0.8320	0.0846	1.1558	0.0960
	4096	0.8709	2.5710	0.1914	0.9418	7.2490	0.4368	0.0461	1.1341	0.0500
	16384	0.8333	11.6928	0.1017	0.8681	17.9741	0.2421	0.0254	6.6147	0.0294
Paraboloid	256	16.2655	14.3173	0.8024	14.4826	16.4250	1.6114	2.2576	2.0542	0.1433
	1024	16.7059	14.5352	0.4456	15.7177	15.5423	1.2397	2.2601	2.1326	0.0999
	4096	14.9447	14.7209	0.2605	16.1801	12.1010	0.5660	2.1653	2.7441	0.0564
	16384	14.7564	26.7635	0.1657	16.6413	36.3490	0.3477	2.1666	24.9964	0.0315



(a) [27,29] (b) [31,32] (c) Ours

Figure 11. Comparison of different mean curvature methods used for feature detection on a large-scale scanning model (block model).

5. Conclusions

Based on the heat kernel function, geometry properties such as the Gaussian curvature κ_G, mean curvature κ_M, principal curvature $\kappa_{1,2}$, and their principal directions $\mathcal{T}_{1,2}$ for sampled point cloud surfaces $\mathcal{M}$ are estimated. The proposed method does not need to locally approximate and fit the point clouds and it should not apply any mesh or Voronoi-based map, which is a truly mesh-free method. Experiments show that the provided curvature estimation method has higher accuracy than the existing methods and has great potential in practical applications. In particular, the mean curvature can be used to detect the features of point clouds even for very small interior angles. Although the mean curvatures using the heat kernel function are accurate, the Gaussian curvature is still acquired by finding an optimal ellipse. In the future, we would like to develop another new operator using the Gauss-Bonnet theorem to derive the Gaussian curvature on point cloud surfaces.

Author contributions

Kai Wang: Designed project, wrote the first draft, provided figures and revised the manuscript; Xiheng Wang: Designed the project and revised the manuscript; Xiaoping Wang: Conceptualized and supervised the project, revised the manuscript, and acquired funding. All authors have read and approved the final version of the manuscript for publication.

Acknowledgments

This research was supported by National Natural Science Foundation of China (No. 51575266 and No. 52075258).

Conflict of interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

1. K. Wang, X. Wang, J. Gan, A general method of trajectory generation based on point-cloud structures in automatic fibre placement, *Comp. Stru.*, **314** (2023), 116976. https://doi.org/10.1016/j.compstruct.2023.116976
2. X. Ji, X. Zhang, H. Hu, Point cloud segmentation for complex micro surfaces based on feature line fitting, *Mult. Tools Appl.*, **80** (2021), 4553–4578. https://doi.org/10.1007/s11042-020-09910-6
3. L. Zhu, W. Chen, X. Lin, L. He, Curvature variation inspired sampling for point cloud classification and segmentation, *IEEE Signal. Process. Lett.*, **29** (2022), 1868–1872. https://doi.org/10.1109/LSP.2022.3200585
4. A. Vo, L. Truong-Hong, Octree-based region growing for point cloud segmentation, *ISPRS J. Photogramm. Remote Sens.*, **104** (2015), 88–100. https://doi.org/10.1016/j.isprsjprs.2015.01.011
5. Y. Gao, C. Ping, A simplification method for point cloud of t-profile steel plate for shipbuilding, *Algorithms*, **14** (2002), 202. https://doi.org/10.3390/a14070202
6. Z. Yao, Q. Zhao, X. Li, Point cloud registration algorithm based on curvature feature similarity, *Measurement*, **177** (2021), 109274. https://doi.org/10.1016/j.measurement.2021.109274
7. Y. He, S. Kang, H. Liu, Curvature regularized surface reconstruction from point clouds, *SIAM J. Imaging Sci.*, **13** (2020), 1834–1859. https://doi.org/10.1137/20M1314525
8. K. Wang, X. Wang, D. Zhang, Constructing continuous curves on point-cloud surfaces with directed projection operator, *J. Comput. Sci.*, **79** (2020), 1877–7503. https://doi.org/10.1016/j.jocs.2023.102028
9. X. Wang, H. Chen, L. Wu, Feature extraction of point clouds based on region clustering segmentation, *Multimedia Tools Appl.*, **79** (2020), 1877–7503. https://doi.org/10.1007/s11042-019-08512-1
10. J. Nie, Extracting feature lines from point clouds based on smooth shrink and iterative thinning, *Graph. Models*, **84** (2016), 38–49. https://doi.org/10.1016/j.gmod.2016.04.001
11. T. Surazhsky, E. Magid, O. Soldea, A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data, *Comput. Vis. Image Underst.*, **107** (2007), 139–159. https://doi.org/10.1109/ROBOT.2003.1241726
12. G. Taubin, Estimating the tensor of curvature of a surface from a polyhedral approximation, *Proc. IEEE Int. Conf. Comput. Vis.*, **1** (1995), 902–907. https://doi.org/10.1109/ICCV.1995.466840
13. X. Chen, F. Schmitt, Intrinsic surface properties from surface triangulation, *Lect. Notes Comput. Sci.*, **588** (1992). https://doi.org/10.1007/3-540-55426-2_83
14. C. Dong, G. Wang, Curvatures estimation on triangular mesh, *Proc. IEEE Int. Conf. Comput. Vis.*, **6** (2005), 128–136. https://doi.org/10.1631/jzus.2005.AS0128
15. M. Meyer, M. Desbrun, P. Schröder, Discrete differential-geometry operators for triangulated 2-manifolds, **1** (2003), 35–57. https://doi.org/10.1007/978-3-662-05105-4_2

16. J. Goldfeather, V. Interrante, A novel cubic-order algorithm for approximating principal direction vectors, *ACM Trans. Graph.*, **23** (2004), 45–63. https://doi.org/10.1145/966131.966134

17. D. Fernando, B. Andrew, Discrete differential operators on polygonal meshes, *ACM Trans. Graph.*, **39** (2020), 1–14. https://doi.org/10.1145/3386569.3392389

18. M. Makovník, P. Chalmoviansky, Curvature estimation for meshes via algebraic quadric fitting, *Arxiv*, 2023. https://doi.org/10.48550/arXiv.2304.08909

19. I. Douros, B. Buxton, Three-dimensional surface curvature estimation using quadric surface patches, *Scanning*, **44** (2002). Available from: https://api.semanticscholar.org/CorpusID:14748373

20. S. Cheng, Y. Zhou, J. Shi, Curvature estimation to scattered point cloud, *Appl. Mech. Mater.*, **29** (2010), 1263–1267. Available from: https://www.scientific.net/AMM.29-32.1263

21. A. Spek, W. Li, T. Drummond, A fast method for computing principal curvatures from range images, *Arxiv*, **1** (2017), 33–41. https://doi.org/10.48550/arXiv.1707.00385

22. Y. Cheng, W. Li, C. Jiang, A novel point cloud simplification method using local conditional information, *Meas. Sci. Technol.*, **33** (2022), 125203. https://doi.org/10.1088/1361-6501/ac8ac1

23. A. Yasuhiko, I. Yuichi, Curvature of point clouds through principal component analysis, *CoRR*, **1** (2021), 1–23. Available from: https://dblp.org/rec/journals/corr/abs-2106-09972

24. A. Foorginejad, K. Khalili, Umbrella curvature: A new curvature estimation method for point clouds, *Procedia Tech.*, **12** (2014), 347–352. https://doi.org/10.1016/j.protcy.2013.12.497

25. N. Amenta, Defining point-set surfaces, *ACM Trans. Graph.*, **23** (2004), 264–270. https://doi.org/10.1145/1015706.1015713

26. N. Amenta, The domain of a point set surfaces, *EG SPG*, **1** (2004), 139–147. https://doi.org/10.2312/SPBG/SPBG04/139-147

27. P. Yang, X. Qian, Direct computing of surface curvatures for point-set surfaces, *SPBG*, **1** (2007), 139–147. https://doi.org/10.2312/SPBG/SPBG07/029-036

28. R. Goldman, Curvature formulas for implicit curves and surfaces, *Comput. Aided Geom. Des.*, **22** (2005), 632–658. https://doi.org/10.1016/j.cagd.2005.06.005

29. M. Tian, J. Lu, L. Zhang, Curvature estimation on point cloud using an indicator function, *HPCCT & BDAI '20*, **1** (2020), 24–28. https://doi.org/10.1145/3409501.3409505

30. Y. Miao, Curvature estimation of Point-Sampled surfaces and its applications, *Computat. Sci. Appl.*, **3482** (1971), 1023–1032. https://doi.org/10.1007/11424857_110

31. X. Zhang, H. Li, Robust curvature estimation and geometry analysis of 3D point cloud surfaces, *J. Inf. Comput. Sci.*, **6** (2009), 1983–1990. Available from: https://api.semanticscholar.org/CorpusID:54773600

32. X. Zhang, Curvature estimation of 3D point cloud surfaces through the fitting of normal section curvatures, *ASIAGRAPH 2008 Proc.*, **1** (2008), 1–10. Available from: https://api.semanticscholar.org/CorpusID:10468338

33. K. Wang, F. Chen, Y. Chen, Directly compute curvatures on Point-based surface, *Mini-Micro Syst.*, **26** (2005), 813–817. Available from: https://api.semanticscholar.org/CorpusID:123831294

34. Q. Mérigot, M. Ovsjanikov, L. Guibas, Voronoi-Based curvature and feature estimation from point clouds, *IEEE Trans. Vis. Comput. Graph.*, **26** (2005), 743–756. https://doi.org/10.1109/TVCG.2010.261

35. J. Lachaud, D. Coeurjolly, C. Labart, Lightweight curvature estimation on point clouds with randomized corrected curvature measures, *Comput. Graph. Forum*, **42** (2023), e14910. https://doi.org/10.1111/cgf.14910

36. Y. Cao, D. Li, H. Sun, Efficient weingarten map and curvature estimation on manifolds, *Mach. Learn.*, **110** (2021), 1319–1344. https://doi.org/10.1007/s10994-021-05953-4

37. C. Lange, K. Polthier, Anisotropic smoothing of point sets, *Comput. Aided Geom. Des.*, **22** (2005), 680–692. https://doi.org/10.1016/j.cagd.2005.06.010

38. S. Rosenberg, *The laplacian on a riemannian manifold: An introduction to analysis on manifolds*, Cambridge: Rosenberg, 1997. https://doi.org/10.1017/cbo9780511623783

39. M. Ahmed, I. Eldesoky, The profound effect of heat transfer on magnetic peristaltic flow of a couple stress fluid in an inclined annular tube, *Mod. Phys. Lett. B*, **38** (2024), 2450233. https://doi.org/10.1142/S0217984924502336

40. P. Kumar, S. Balakrishnan, A. Magesh, Numerical treatment of entropy generation and Bejan number into an electroosmotically-driven flow of Sutterby nanofluid in an asymmetric microchannel, *Numer. Heat Tr. B-fund.*, **1** (2024), 1–20. https://doi.org/10.1080/10407790.2024.2329773

41. A. Grigoryan, *Heat kernel and analysis on manifolds*, Rhode Island: American mathematical society and international press, 2009. https://doi.org/10.1090/amsip/047

42. M. Belkin, P. Niyogi, Towards a theoretical foundation for Laplacian-based manifold methods, *J. Comput. Syst. Sci.*, **74** (2008), 1289–1308. https://doi.org/10.1016/j.jcss.2007.08.006

43. M. Belkin, J. Sun, Y. Wang, Constructing Laplace operator from point clouds in Rd, *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2009, 1031–1040. Available from: `https://dl.acm.org/doi/abs/10.5555/1496770.1496882`

44. D. Ulrich, H. Stefan, K. Albrecht, *Minimal surfaces*, Berlin: Springer Berlin Heidelberg, 2010. https://doi.org/10.1007/978-3-642-11698-8

45. M. Pauly, *Point primitives for interactive modeling and processing of 3D geometry*, Germany: Hartung-Gorre, 2003. https://doi.org/10.3929/ETHZ-A-004612876

46. S. Gilbert, *Introduction to linear algebra*, 5 Eds., Wellesley: Wellesley-Cambridge Press, 2003. https://doi.org/10.1007/978-1-4612

47. C. Keenan, W. Clarisse, W. Max, Geodesics in heat: A new approach to computing distance based on heat flow, *ACM Trans. Graph.*, **32** (2013), 1–13. https://doi.org/10.1145/2516971.2516977

Appendix A: Choice of parameter m

To find an optimal heat diffuse time t for the point cloud surface $\mathcal{M}$, we suggested representing the heat diffuse time t as a quadratic function $t = mh^2$ in relation to the distance parameter h that is closely related to the sampled distribution of the point clouds themselves. Then, the problem of choosing a

parameter t can be transformed into evaluating the constant m. This assessment criterion can also be discovered in computing geodesic map with the heat method [47], which performs extremely well in practice.

Influence of the constant m on area $\mathcal{A}$

To estimate an ideal m, an error evaluating mechanism with respect to illustrating absolute errors of the area $\mathcal{A}$ is defined first:

$$E = \frac{|\mathcal{A} - \mathcal{A}^*|}{\mathcal{A}^*} \times 100\%,$$

where the symbol $\mathcal{A}^*$ represents the exact solution and $\mathcal{A}$ is the area sum of the points in $\mathcal{M}$. This mechanism is suitable for point cloud surfaces $\mathcal{M}$ where the superficial area can be easily acquired, which means that the given models have analytic equations, as see in Figure 4. Moreover, Figure 5 shows the visualization for the influences on the estimated area for different parameters m and it reveals that $m = 1/4$ is applicable in computing the area for the point cloud surface $\mathcal{M}$.

Influence of m on mean curvature κ_M

Our goal is to accurately compute the mean curvatures κ_M on the point cloud surfaces $\mathcal{M}$ first and then we need to expand these results to estimate some extra key properties, such as the principal curvatures $\kappa_{1,2}$, the principal directions $\mathcal{T}_{1,2}$, and even the Gaussian curvature κ_G. Similar to the area $\mathcal{A}$, the mean curvatures κ_M are sensitive to m. Therefore, we develop another error evaluating mechanism aimed at the mean curvatures κ_M of the point cloud surfaces:

$$E = \frac{\sum_{i=1}^{n} \left| |\kappa_i| - \left|\kappa_i^*\right| \right|}{n} \times 100\%, \tag{5.1}$$

where κ_i represents the estimated curvature of the point $\mathrm{x}_i \in \mathrm{P}$ and κ^* is its related exact solution. This mechanism can also be adopted to evaluate the Gaussian curvature κ_G, mean curvature κ_M, and two principal curvature $\kappa_{1,2}$.

Using some formulas in Example 4.1 in [28], some models with analytic mean curvatures, such as the Saddle, Arch, and Paraboloid, are adopted to analyze the influence of the parameter m. Figure 5 illustrates some absolute error curves with respect to m and Figure 6 shows the visualization of mean curvature κ_M with a different m on a unit sphere. All the experiments in this section suggest that $m = 1/4$ is a continuously stable parameter for the pending function $t = mh^2$ and it generates lower errors in approximating the actual area sum and mean curvature, as seen in Figures 3–6. To sum up, $m = 1/4$ is available and the inherent distant parameter h will replace the heat diffuse time t in subsequent computations.