Noname manuscript No.
(will be inserted by the editor)

Discretizing Laplace–Beltrami Operator from Differential Quantities

**Ruimin Wang · Zhouwang Yang* ·
Ligang Liu · Qing Chen**

Received: date / Accepted: date

Abstract The Laplace–Beltrami operator (LBO) is the fundamental geometric object associated with manifold surfaces and has been widely used in various tasks in geometric processing. By understanding that the LBO can be computed by differential quantities, we propose an approach for discretizing the LBO on manifolds by estimating differential quantities. For a point on the manifold, we first fit a quadratic surface to this point and its neighborhood by minimizing the least-square energy function. Then we compute the first- and second-order differential quantities by the approximated quadratic surface. Finally the discrete LBO at this point is computed from the estimated differential quantities and thus the Laplacian matrix over the discrete manifold is constructed. Our approach has several advantages: it is simple and efficient and insensitive to noise and boundaries. Experimental results have shown that our approach performs better than most of the current approaches. We also propose a feature-aware scheme for modifying the Laplacian matrix. The modified Laplacian matrix can be used in other feature preserving geometric processing applications.

Keywords Laplace–Beltrami Operator · Discretization · Differential Quantities

1 Introduction

The Laplace–Beltrami Operator (LBO), also called manifold Laplacian, is a fundamental geometric object associated with a Riemannian manifold. Discrete LBO, which is also called Laplacian matrix, has been quite widely used in spectral analysis on discrete surfaces [3,14,36] and various tasks of geometric processing [7,9,17,25,26].

A number of discretizations of the Laplacian for discrete surfaces have been proposed. Most existing discretization methods are based on the definition of LBO which

R. Wang · Z. Yang* · L. Liu · Q. Chen
University of Science and Technology of China, Hefei, Anhui 230026, P. R. China
E-mail: wruimin@mail.ustc.edu.cn, {yangzw, lgliu, qchen}@ustc.edu.cn

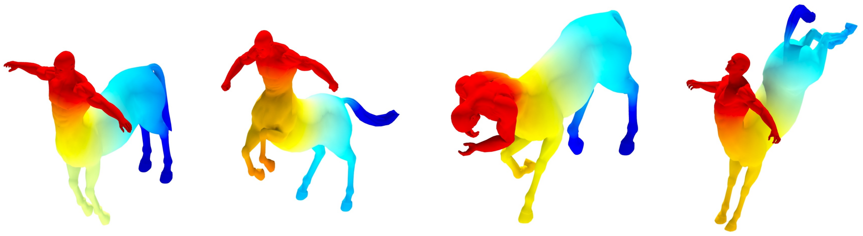


Fig. 1 The Fiedler vectors of our method on four centaurs with different poses. Fiedler vector is the eigenvector associated with the first non-zero eigenvalue.

is defined as the solution of an integral equation. Thus they have similar forms of cotangent scheme [22] over meshes based on the Finite Element Method (FEM) with different assumptions. The works of [5] and [18] define the Laplacian matrices over point clouds by an integration process via an edge weighting matrix.

However, existing discretization methods suffer from a few problems. Graph-based or FEM-based methods are based on the computation of finite differences of the LBO and the convergence of these methods can be established for only special classes of meshes. These methods may be sensitive to mesh density and noise and have poor performance on the boundary of the non-closed meshes. Other methods adopt some integral forms [5, 18], which rely on sampling rates to produce meaningful discrete operator and are generally computationally expensive.

We have observed that in differential geometry [8] the LBO can be defined as a more essential form (see Equation (1)) in which only the differential quantities of no more than second order are involved. That means the LBO at a point can be characterized by the first- and second-order differential quantities at that point. This inspires us to believe that the LBO can be computed by estimating the differential quantities which captures the essence of the manifold.

We propose an approach for discretizing the LBO on manifolds from the differential quantities. For a given point on the discrete manifold, we fit a quadratic surface over the point and its neighborhood in a least-square sense. First- and second-order differential quantities are computed by those of the approximated quadratic surface. Then the discrete LBO at this point is computed from the estimated differential quantities and thus the Laplacian matrix over the discrete manifold is constructed.

Our approach is easy to implement and is fast as it involves solving only 5×5 linear systems. It has almost all the desired properties on the Laplacian matrix. Due to the local fitting process, our approach can handle either mesh surfaces or point clouds and is insensitive to data noise. Furthermore, it handles boundaries and non-uniform sampling well. Detailed analysis and comparisons to previous approaches are found in Section 4 and 5. Figure 1 illustrates the Filder vectors of our method on four Centaurs models. We also propose a feature-aware scheme for modifying the Laplacian matrix which is quite useful for feature preserving geometric processing.

The contribution of our work is summarized as follows:

- An approach for discretizing the LBO from differential quantities is proposed.
- The proposed approach performs better than most existing methods.

- A feature-aware scheme is proposed for modifying the Laplacian matrix which is very useful for feature preserving geometric processing.

In this paper, we also comprehensively put forward a C^0 modification and compare our method with previous ones comprehensively. The LBO obtained by our approach performs better than most existing methods. We believe that this fundamental work would benefit various applications in geometric processing and would inspire other works in this field.

2 Basic Notation and Previous Methods

There have been a number of works on spectral analysis on discrete surfaces during the last two decades [7, 14, 26, 31, 35, 36]. First, we review some basic notions that will be employed in this work:

2.1 Laplace–Beltrami operator

It is well known that differential geometry provides a convenient basis for describing the behavior of a shape. In differential geometry, the Laplace operator can be generalized to operate on functions defined on surfaces in Euclidean space and more generally on Riemannian manifolds. This general operator is Laplace–Beltrami operator. Like the Laplacian, the Laplace–Beltrami operator is defined as the divergence of the gradient, and is a linear operator taking functions into functions.

Definition 1 Let S be a surface $\mathrm{r}(u,v)$ of parameters $(u^1 = u, u^2 = v)$ with metric matrix $g_{\alpha\beta} =< \frac{\partial \mathbf{r}}{\partial u^\alpha}, \frac{\partial \mathbf{r}}{\partial u^\beta} >$, $g = \det(g_{\alpha\beta})$, $(g^{\alpha\beta}) = (g_{\alpha\beta})^{-1}$. The Laplace–Beltrami operator on a function f is defined as:

$$\Delta_S f = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u^\beta} (\sqrt{g} g^{\alpha\beta} \frac{\partial f}{\partial u^\alpha}) \tag{1}$$

Note that (1) is in the form of an Einstein summation convention and can be expanded as follows:

$$\begin{aligned}\Delta_S f =& \frac{1}{2g} (\frac{\partial g}{\partial u}, \frac{\partial g}{\partial v}) (\begin{smallmatrix} g^{11} & g^{12} \\ g^{21} & g^{22} \end{smallmatrix}) \begin{pmatrix} \frac{\partial f}{\partial u} \\ \frac{\partial f}{\partial v} \end{pmatrix} \\ &+ [(\frac{\partial}{\partial u}, \frac{\partial}{\partial v}) (\begin{smallmatrix} g^{11} & g^{12} \\ g^{21} & g^{22} \end{smallmatrix})] \begin{pmatrix} \frac{\partial f}{\partial u} \\ \frac{\partial f}{\partial v} \end{pmatrix} \\ &+ g^{11} \frac{\partial^2 f}{\partial u^2} + 2g^{12} \frac{\partial^2 f}{\partial u \partial v} + g^{22} \frac{\partial^2 f}{\partial v^2}.\end{aligned} \tag{2}$$

2.2 Previous Methods

Geometric shapes are often represented as 2D curved surfaces, 2D surface meshes (usually triangle meshes with geometric positions), 2D surface point clouds (sets of points without connectivity) or 3D solid objects (e.g. using voxels or tetrahedra meshes). Numerically, discretizing LBO operator represents constructing a Laplace matrix L with $\Delta_S f_{\mathbf{p}} \approx L\mathbf{f}|_{\mathbf{p}}$ where $\mathbf{f}$ is the discrete sampling of the function f over the discrete surface and $\mathbf{p}$ is one point on the discrete surface. Laplacian matrix is an analog of the continuous Laplace operator, defined so that it has meaning on a graph or a discrete grid. Several discretizations of the Laplace–Beltrami operator exist for the different types of geometric representations as mentioned below.

Generally, the most common Laplacian matrix on 2D surface meshes is the weighted graph Laplacian $L = (a_{ij})$ defined by

$$a_{ij} = \begin{cases} \omega_{ij} > 0, & \text{if } (i,j) \text{ is an edge} \\ -\sum_{j\neq i} \omega_{ij}, & i = j \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

There are various graph Laplacians defined in the literature. These include normalized graph Laplacian (GL), Kirchhoff Laplacian (KL), and Tutte Laplacian (TL) [20, 24, 28, 35]. KL simply uses the uniform weighting $\omega_{ij} = 1$. GL has the entry as $GL_{ij} = 1/\sqrt{d_i d_j}$ and TL has the entry as $TL_{ij} = 1/d_i$. Here d_i stands for the degree of ith vertex on the mesh. These methods are simple since they only use connectivity information of the given mesh, thus they do not reveal the geometry of the mesh. And they are quite sensitive to the density of the points on mesh.

The works of [3, 6, 7, 19, 23, 27, 30] apply edge or cotangent weighting Graph Laplacian to various applications. [1] discusses the consistency of these methods using Finite Element Methods (FEM) and Discrete Exterior Calculate (DEC). [32] and [34] attemp to prove convergence of these methods under some assumptions. Only [7] and [19] guarantee the convergence when their methods are applied to some special classes of meshes such as meshes with all valence 6. [4] proposes an approximating algorithm with point-wise convergence guarantees. [33] gets the discrete approximation of differential operators via local quadratic fitting. However, most of these methods are sensitive to noise and can hardly handle the boundaries.

The graph-based methods cannot define the Laplacian on point cloud due to the lack of connectivity information. The works of [5] and [18] study the generation of the discrete Laplace–Beltrami operator over point clouds. They construct the connectivity in the local neighborhood of each point by projecting the neighborhood onto the tangent plane and use edge weighting matrix to stand for the Laplacian matrix. These methods adopt an integral equivalent definition which might be sensitive to the sampling rates of the data and the boundaries. Moreover, they are computationally expensive [18].

It is worth noting that our work has much similarity with two very recent papers [16, 33] on constructing the discrete Laplace-Beltrami operator. All three works share the similar idea to construct differential operators by computing differential quantities through local quadratic approximation and have almost same pipeline in the discretizing process that consists of local parametrization, local approximation, and matrix construction. However, these three works are coincidentally developed from different motivations and applications. [16] uses the discrete differential operators, which extends the discretization method in [15] by taking the discrete Laplace–Beltrami operator as a tool to understand the geometry, to solve partial differential equations on manifolds represented by meshless points. [33] adopts local quadratic fitting to discretize differential operators on manifold and analyzes the convergence of these operators. [16] focuses on applications and behavior in solving partial differential equations while [33] concentrates on mathematical properties of these discrete differential operators.

Our work addresses some fundamental problems in computer graphics including compression, smoothing, and spectral analysis. In our work, we further propose a feature-aware modification of our Laplacian matrix which is applicable for well handling the 3D meshes with sharp features in the applications of computer graphics.

3 Laplacian Matrix from Differential Quantities

It is observed from (2) that the LBO operator can be computed utilizing differential quantities, of no more than a second-order computation of the geometry and the derivatives of the function f. Thus being able to numerically compute the differential quantities and the derivatives of f is essential to our method.

3.1 Local quadratic approximation

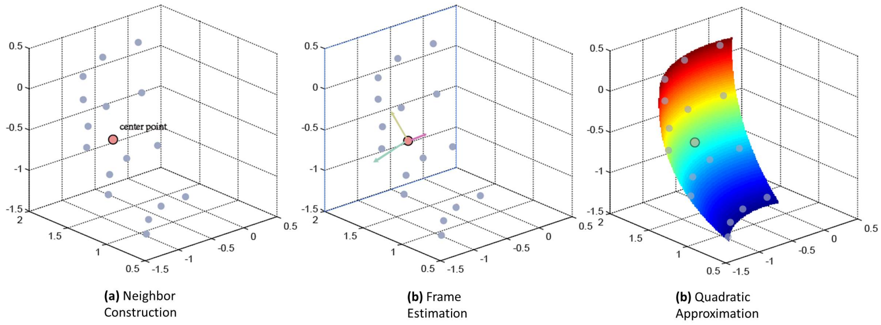
Let $\mathbf{p} = (x_p, y_p, z_p)^T$ be a point on a surface S embedded in $\mathbb{R}^3$. The surface can be described by a certain parametrization of two variables $\mathbf{r}(u,v)$ which is smooth in the vicinity of point $\mathbf{p}$. A second-order asymptotic expansion of the surface in the neighborhood of $\mathbf{p}$ then yields:

$$\begin{aligned}\mathbf{r}(u,v)-\mathbf{p} &= \mathbf{r}(u,v)-\mathbf{r}(0,0)\\ &= \mathbf{r}_u(0,0)u+\mathbf{r}_u(0,0)v\\ &\quad +\tfrac{1}{2}\mathbf{r}_{uu}(0,0)u^2+\mathbf{r}_{uv}(0,0)uv\\ &\quad +\tfrac{1}{2}\mathbf{r}_{vv}(0,0)v^2+o(u^2+v^2).\end{aligned} \tag{4}$$

Here we assume $\mathbf{r}(0,0) = \mathbf{p}$ without loss of generality.

The main idea of our approach is to fit points in the vicinity of a reference $\mathbf{p}$ using a parabolic approximation (i.e., quadratic polynomial surface), and then compute differential quantities of the underlying surface. The parabolic surface can be used for approximating the derivatives of a certain function on the surface. And the obtained differential quantities are used for discretizing the Laplace–Beltrami operator according to equation (2). The approximation procedure is illustrated in Figure 2.

Suppose that the discrete surface $M = \{\mathbf{p}_i\}_{i=1}^N$ is a sampling from the underlying surface S. The input data $\{\mathbf{p}_i\}_{i=1}^N$ can be a mesh or point cloud. Let $\mathscr{B}_k(\mathbf{p}_i) = \{\mathbf{x}_1, \cdots, \mathbf{x}_k\}$ be the k-nearest neighbors of a fixed $\mathbf{p}_i$. The k-nearest neighborhood can be obtained via the Approximate Nearest Neighbor algorithm [2].



(a) Neighbor Construction

(b) Frame Estimation

(b) Quadratic Approximation

Fig. 2 The procedure of local quadratic approximation: (a). the neighborhood construction; (b). the local frame estimation; (c). the approximated underlying surface.

First, we apply a simple Principle Component Analysis(PCA) [11] scheme to estimate a local frame of surface at $\mathbf{p}_i$, and then get a local parametrization (u,v) in its neighborhood. Denote the positional matrix $C_{3\times k}$ as

$$\mathbf{c}_j = \mathbf{x}_j - \mathbf{p}_i,$$

where $\mathbf{c}_j$ is the jth column of C. Let $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ be the three eigenvectors with their corresponding eigenvalues in descending order of the correlation matrix CC^T. According to the theory of PCA, $\mathbf{e}_1$ and $\mathbf{e}_2$ can be considered as the direction of local parameters while $\mathbf{e}_3$ can be considered as the rough normal. Here the rough normal means the normal is not the exact normal of the surface. Actually $\{\mathbf{p}_i; \mathbf{u} = \mathbf{e}_1, \mathbf{v} = \mathbf{e}_2, \mathbf{n} = \mathbf{e}_3\}$ forms a local orthogonal coordinate system. The coordinates of $\mathbf{x}_j$ under this local system are given by

$$u_j = (\mathbf{x}_j - \mathbf{x}_0)^T \mathbf{u}, v_j = (\mathbf{x}_j - \mathbf{x}_0)^T \mathbf{v}, h_j = (\mathbf{x}_j - \mathbf{x}_0)^T \mathbf{n}.$$

Then, according to the asymptotic expansion (4), the underlying surface can be second-order approximated by a paraboloid:

$$\mathbf{q}(u,v) = \begin{pmatrix} u \\ v \\ q(u,v) \end{pmatrix} \tag{5}$$

where $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5)^T$ and $q(u,v) = (u, v, \frac{1}{2}u^2, uv, \frac{1}{2}v^2)\mathbf{c}$. The local quadratic approximation model in least-square sense can now be set up:

$$\min \sum_{j=1}^{k} (q(u_j, v_j) - h_j)^2. \tag{6}$$

The optimal solution of (6) is easily obtained as

$$\mathbf{c}^* = (Q^T Q)^{-1} Q^T \mathbf{h}, \tag{7}$$

where $\mathbf{h} = (h_1, \cdots, h_k)^T$ and

$$Q = \begin{pmatrix} u_1 & v_1 & \frac{1}{2}u_1^2 & u_1v_1 & \frac{1}{2}v_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_k & v_k & \frac{1}{2}u_k^2 & u_kv_k & \frac{1}{2}v_k^2 \end{pmatrix}. \tag{8}$$

Let f be a function defined on the surface S and $\{f(\mathbf{p}_i)\}_{i=1}^N$ is the discretization of f on $\mathscr{M} = \{\mathbf{p}_i\}_{i=1}^N$. Denote

$$E_k = (-\mathbf{1}, I_k) = \begin{pmatrix} -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{k\times(k+1)}.$$

and

$$A = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \mathbf{a}_4 \\ \mathbf{a}_5 \end{pmatrix}_{5\times(k+1)} = (Q^TQ)^{-1}Q^TE_k. \tag{9}$$

With these row vectors $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5\}$, we can use the discrete function values on M to estimate its derivatives. As a result, we have

$$\begin{cases} \frac{\partial f}{\partial u} \approx \mathbf{a}_1\mathbf{f}_i, \ \frac{\partial f}{\partial v} \approx \mathbf{a}_2\mathbf{f}_i, \\ \frac{\partial^2 f}{\partial u^2} \approx \mathbf{a}_3\mathbf{f}_i, \ \frac{\partial^2 f}{\partial u \partial v} \approx \mathbf{a}_4\mathbf{f}_i, \ \frac{\partial^2 f}{\partial v^2} \approx \mathbf{a}_5\mathbf{f}_i \end{cases} \tag{10}$$

where $\mathbf{x}_0 = \mathbf{p}_i$ and $\mathbf{f}_i = (f(\mathbf{x}_0), f(\mathbf{x}_1), \cdots, f(\mathbf{x}_k))^T$.

Remark. The construction of local quadratic surface would fail if Matrix Q defined in (8) becomes singular. In this case, one could use the Moore–Penrose inverse $(Q^TQ)^+$ instead of $(Q^TQ)^{-1}$ like [33]. Then a least-square solution with minimal l^2 norm is found. But in our implementation we use a simpler but practical way to get rid of singularity which is to expand the size of the current neighborhood. Actually the default size of neighborhood is set as 18 and the singularity barely occurs.

3.2 Construction of the Laplace–Beltrami Matrix

At point $\mathbf{p}_i$ the surface is locally approximated by

$$\mathbf{q}(u,v) = \big(u, v, q(u,v)\big)^T, \tag{11}$$

where $q(u,v) = c_1u + c_2v + \frac{1}{2}c_3u^2 + c_4uv + \frac{1}{2}c_5v^2$.

Theorem 1 *If the surface has the form of* (11)*, the differential quantities used in* (2) *at* $(0,0)$ *can be computed as:*

$$\begin{cases} \left(g_{\alpha\beta}\right)|_{\mathbf{p}_i} = \begin{pmatrix} 1+c_1^2 & c_1c_2 \\ c_1c_2 & 1+c_2^2 \end{pmatrix}, \\ g = \det(g_{\alpha\beta})|_{\mathbf{p}_i} = 1+c_1^2+c_2^2, \\ \left(g^{\alpha\beta}\right)|_{\mathbf{p}_i} = \begin{pmatrix} \frac{1+c_2^2}{1+c_1^2+c_2^2} & \frac{-c_1c_2}{1+c_1^2+c_2^2} \\ \frac{-c_1c_2}{1+c_1^2+c_2^2} & \frac{1+c_1^2}{1+c_1^2+c_2^2} \end{pmatrix}, \end{cases} \tag{12}$$

and

$$\begin{cases} g_{11u} = 2c_1c_3, \\ g_{12u} = g_{21u} = c_1c_4 + c_2c_3, \\ g_{22u} = 2c_2c_4, \\ g_{11v} = 2c_1c_4, \\ g_{12v} = g_{21v} = c_1c_5 + c_2c_4, \\ g_{22v} = 2c_2c_5; \\ g_u = 2c_1c_3 + 2c_2c_4, \\ g_v = 2c_1c_4 + 2c_2c_5; \\ g_u^{11} = (g\cdot g_{22u} - g_{22}g_u)/g^2, \\ g_u^{12} = g_u^{21} = -(g\cdot g_{12u} - g_{12}g_u)/g^2, \\ g_v^{21} = g_v^{12} = -(g\cdot g_{12v} - g_{12}g_v)/g^2, \\ g_v^{22} = (g\cdot g_{11v} - g_{11}g_v)/g^2. \end{cases} \tag{13}$$

Here we denote $\frac{\partial g}{\partial u}, \frac{\partial g}{\partial v}, \frac{\partial g_{\alpha\beta}}{\partial u}, \frac{\partial g_{\alpha\beta}}{\partial v}, \frac{\partial g^{\alpha\beta}}{\partial u}, \frac{\partial g^{\alpha\beta}}{\partial v}$ *by* g_u, g_v, $g_{\alpha\beta u}, g_{\alpha\beta v}$, $g_u^{\alpha\beta}, g_v^{\alpha\beta}$, *respectively.*

Proof Since the surface has the form of (11), the first-order derivatives are:

$$\begin{cases} \mathbf{q}_u = (1, 0, c_1 + c_3u + c_4v)^T, \\ \mathbf{q}_v = (0, 1, c_2 + c_4u + c_5v)^T. \end{cases} \tag{14}$$

So, we have

$$\begin{cases} g_{11} & =<\mathbf{q}_u, \mathbf{q}_u>= 1 + (c_1 + c_3u + c_4v)^2, \\ g_{12} = g_{21} & =<\mathbf{q}_u, \mathbf{q}_v>= (c_1 + c_3u + c_4v)(c_2 + c_4u + c_5v), \\ g_{22} & =<\mathbf{q}_v, \mathbf{q}_v>= 1 + (c_2 + c_4u + c_5v)^2. \end{cases} \tag{15}$$

According to their definition, the metric matrix, its determinant and the inverse are given as

$$\begin{cases} \left(g_{\alpha\beta}\right) = \begin{pmatrix} 1 + (c_1 + c_3u + c_4v)^2 & (c_1 + c_3u + c_4v)(c_2 + c_4u + c_5v) \\ (c_1 + c_3u + c_4v)(c_2 + c_4u + c_5v) & 1 + (c_2 + c_4u + c_5v)^2 \end{pmatrix}, \\ g = \det(g_{\alpha\beta}) = 1 + (c_1 + c_3u + c_4v)^2 + (c_2 + c_4u + c_5v)^2, \\ \left(g^{\alpha\beta}\right) = \begin{pmatrix} \frac{1+(c_2+c_4u+c_5v)^2}{1+g} & \frac{-(c_1+c_3u+c_4v)(c_2+c_4u+c_5v)}{1+g} \\ \frac{-(c_1+c_3u+c_4v)(c_2+c_4u+c_5v)}{1+g} & \frac{1+(c_1+c_3u+c_4v)^2}{1+g} \end{pmatrix}. \end{cases} \tag{16}$$

Let $(u,v)=(0,0)$, it is obvious that (12) is established. Referring to the first equation in (15), we have

$$\begin{aligned} g_{11u} &= \tfrac{\partial}{\partial u}(1+(c_1+c_3u+c_4v)^2) = 2c_3(c_1+c_3u+c_4v), \\ g_{11v} &= \tfrac{\partial}{\partial v}(1+(c_1+c_3u+c_4v)^2) = 2c_4(c_1+c_3u+c_4v). \end{aligned}$$

At $(0,0)$, it is reduced to

$$g_{11u} = 2c_1c_3,\; g_{11v} = 2c_1c_4.$$

Similarly, the first six equations in (13) can be established. Meanwhile,

$$\begin{aligned} g_u &= 2c_3(c_1+c_3u+c_4v)+2c_4(c_2+c_4u+c_5v), \\ g_v &= 2c_4(c_1+c_3u+c_4v)+2c_5(c_2+c_4u+c_5v). \end{aligned}$$

This stands for $g_u = 2c_1c_3+2c_2c_4$, $g_v = 2c_1c_4+2c_2c_5$ at (0,0). Since $(g^{\alpha\beta}) = (g_{\alpha\beta})^{-1}$, $g^{11} = g_{22}/g$. According to the chain rule, we obtain

$$\begin{aligned} g^{11}_u &= (g\cdot g_{22u} - g_{22}g_u)/g^2, \\ g^{12}_u &= g^{21}_u = -(g\cdot g_{12u} - g_{12}g_u)/g^2, \\ g^{21}_v &= g^{12}_v = -(g\cdot g_{12v} - g_{12}g_v)/g^2, \\ g^{22}_v &= (g\cdot g_{11v} - g_{11}g_v)/g^2. \end{aligned}$$

So the rest of the theorem is proved.

□

So far we have computed all the differential quantities involved in (2) and are ready to discretize the Laplace–Beltrami operator. With the help of these geometric quantities and the derivatives in (10), it is straightforward for us to attain an estimator of $\Delta_S f|_{\mathbf{p}_i}$ in closed form

$$\Delta_S f|_{\mathbf{p}_i} \approx \mathbf{b}_i \mathbf{f}_i,$$

where the row vector $\mathbf{b}_i = [\frac{1}{2g}(g_u g^{11}+g_v g^{21}) + (g^{11}_u + g^{21}_v)]\mathbf{a}_1 + [\frac{1}{2g}(g_u g^{12}+g_v g^{22}) + (g^{12}_u + g^{22}_v)]\mathbf{a}_2 + g^{11}\mathbf{a}_3 + 2g^{12}\mathbf{a}_4 + g^{22}\mathbf{a}_5 \in \mathbb{R}^{k+1}$.

For each $i\in\{1,\cdots,N\}$, we bring in a selection matrix $S_i \in \mathbb{R}^{(k+1)\times N}$, which holds 0 or 1 as possible components and satisfies

$$(\mathbf{p}_i, \mathbf{x}_1, \cdots, \mathbf{x}_k) = (\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N) S_i^T.$$

It is apparent that $\mathbf{f}_i = S_i\mathbf{f}$ with $\mathbf{f} = (f(\mathbf{p}_1),\cdots,f(\mathbf{p}_N))^T \in \mathbb{R}^N$. Let $L_M \in \mathbb{R}^{N\times N}$ be the discrete Laplacian matrix. Then we have

$$(L_M\mathbf{f})_i = L_i\mathbf{f} = \mathbf{b}_i\mathbf{f}_i,$$

where obviously $L_i = \mathbf{b}_i S_i$. Eventually we get a discretization of Laplace–Beltrami operator and the corresponding discrete Laplacian matrix

$$L_M = \begin{pmatrix} L_1 \\ \vdots \\ L_N \end{pmatrix}. \tag{17}$$

4 Discussion

Since the Laplacian matrix constructed in Section 3 is a discretization of Laplace–Beltrami Operator on 3D objects, we want the matrix to own some desired properties. [31] discusses a few properties previous Laplacian matrices had/did not have. Here we list and prove the properties our Laplacian matrix satisfies.

4.1 Properties of the Laplacian matrix

Property 1 **(NULL)**: $L_M\mathbf{f} = 0$ whenever f is a constant function on the surface.

Note that $(L_M\mathbf{f})_i = \mathbf{b}_i\mathbf{f}_i$ automatically implies that L_M satisfies (NULL) since $\mathbf{f}_i$ is a constant vector and $E_k\mathbf{f}_i = 0$. □

Property 2 **(LIN).** Linear precision: $L_M\mathbf{f} = 0$ whenever M is part of the Euclidean plane, and $\mathbf{f}$ is a linear function on the plane.

The characteristic of least-square method guarantees that the obtained local surface is part of a plane if the sampling points all lie on one certain Euclidean plane. Meanwhile the estimated coordinates $\{c_i\}$ and row vectors $\{\mathbf{a}_i\}$ are also all precise since least square can reconstruct linear signals accurately. So $L_M\mathbf{f}$ equals 0 at any sampling point since the real LBO of a plane equals 0. □

Property 3 **(LOC).** Locality: Altering the function value of a distant point will not affect the action of the Laplacian locally.

The desired property is that for any pair $\mathbf{p} \neq \mathbf{q}$, $\Delta f(\mathbf{p})$ is independent of $f(\mathbf{q})$. But it is unable to be obtained numerically. So often as not, locality means the Laplacian at one point is associated with a small number of points. This is easy to explain while noting that the number of non-zero elements in one row of L_M is k. □

There is another important property of a desired discrete Laplacian matrix: **(Symmetrization).** Consider a smooth surface S, possibly with a boundary, equipped with a Riemannian metric. Denote the intrinsic L^2 inner product of functions ϕ and ψ on S by $<\phi,\psi>_{L^2} = \int_S \phi\psi d\sigma$. The Laplace–Beltrami operator Δ_S is self-adjoint, i.e., $<\Delta_S\phi,\psi>_{L^2} = <\phi,\Delta_S\psi>_{L^2}$ whenever ϕ and ψ are sufficiently smooth and vanish along the boundary of S.

When discretizing the surface S into M, the inner product in $L^2(S)$ does not correspond to the Euclidean inner product in $\mathbb{R}^N$. So the self-adjoint property of Laplace–Beltrami operator cannot launch in the symmetry of the discrete Laplacian matrix. Therefore, the requirement for the symmetrical Laplacian matrix L_M is unreasonable. Largely, the desire for symmetry is based on the following motivation: Real symmetric matrices exhibit real eigenvalues and orthogonal eigenvectors. One alternative is to construct a symmetrizable Laplacian matrix, i.e., the matrix can be symmetrized under similarity transformation.

Of course, our method does not provide the symmetric Laplacian matrix. Generally, it is difficult to prove that the Laplacian matrix is symmetrizable. In the numerical

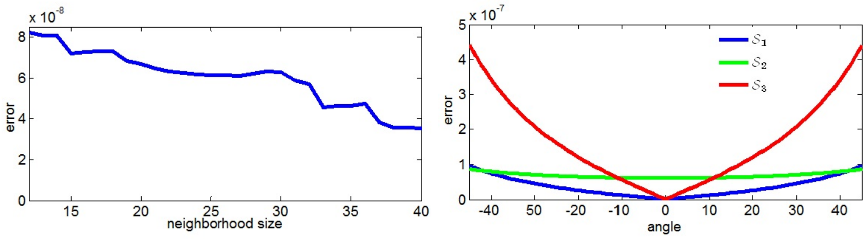


Fig. 3 Left figure demonstrates the relationship between approximation and size of neighborhood. Right figure demonstrates the influence of deflected normals on approximation quality. Approximation error vs size of neighborhood.

experiment, we find that most eigenvalues and eigenvectors of our Laplacian matrix are real. This might indicate that our Laplacian matrix is probably symmetrizable, but we cannot prove it yet.

Remark. [33] proposes a comprehensive proof of point-wise convergence of various discrete differential operators. And the convergence of our operator can be considered as a special case whose results can be summarized in the following manner: If the matrix Q defined in formula (8) is nonsingular meaning that the parametric neighbor is well-posed for quadratic fitting, the value of discrete LBO at $\mathbf{x}_0$ has linear convergence. Additionally we would like to refer the interested reader to Xu's work for further discussion about the convergence.

4.2 Parameter settings

Size of neighborhood k is a parameter related to our algorithm which determines the number of points used in the approximation process. It is necessary to discuss how much k would influence the final result. Consider the surface $z = x^2 + y^2$ and its discretization around point $\mathbf{p} = (-1,-1,2)^T$. Denote the n-nearest neighbors of $\mathbf{p}$ by $\{\mathbf{x}_j\}_{j=1}^n$ where $n = 40$. Let $\{\mathbf{q}_j^{(k)}\}_{j=1}^n$ be the points corresponding to $\{\mathbf{x}_j\}$ on the quadratic surface approximated by k-nearest neighbors ($12 \le k \le 40$). The quality of approximation is measured by error $\sqrt{\frac{1}{n}\sum \|p_i - q_i\|^2}$ in the sense of $\|.\|_2$. It is obvious that the result is already good enough with small k, and quality improvement of fitting is limited, as shown in Figure 3 (right side). So we choose 18 as default in our algorithm.

Here we also show the evidence and claim that very rough normal vectors can be used to parameterize the points in local quadratic approximation. For three surfaces: $z = x^2 + y^2$, $x^2 + y^2 + (z-1)^2 = 1$, $z = \sin(x)\sin(y)$, with an exact normal vector $(0,0,1)^T$ at the point $\mathbf{p} = (0,0,0)^T$, we observe the approximation error with noisy normals (no more than 45 degrees random rotational deviation in the origin normal direction). Even when the normal deviation climbs up to 45 degrees, the error is still small, as shown in the right figure in Figure 3 (right side).

5 Applications and Comparison

To date, discrete spectral analysis is a popular tool in digital geometric processing. Here we briefly introduce some common applications taking advantage of the Laplacian matrix.

5.1 Applications

The Laplacian matrix could be regarded as a filter of signals on a surface and has been used to smooth and optimize the meshes [7, 10, 20, 26]. They also have been used in mesh editing [24] and mesh parameterization [6, 13]. The fundamental Laplace smoothing model for these methods can be represented as:

$$\min_{X} \|LX\|^2 + \lambda \|X - Y\|^2. \tag{18}$$

Here L denotes the Laplacian matrix, X denotes the optimized coordinate matrix of surface and Y donates the input coordinate matrix. λ is called a tuning parameter since its value determines whether the optimized surface is to be smooth or be close to the input surface. Figure 6 shows the result of Laplace smoothing using different kinds of Laplacian matrices.

The eigendecomposition of the Laplacian matrix provides a set of eigenpairs which are widely used in many fields. The eigenpairs are obtained via the solving traditional eigenproblem:

$$(L - \mu_i I)\mathbf{e}_i = 0,\ i = 1, \cdots, N. \tag{19}$$

Here, I is identity matrix, μ_i is the eigenvalue and $\mathbf{e}_i$ is the corresponding eigenvector. The eigenvectors $\{\mathbf{e}_i\}$ can be considered as a set of bases onto which functions on manifolds are projected. The spectral transform coefficients could be further analyzed or manipulated. In [12], they project the manifold itself onto the spectral space spanned by a small number of eigenvectors corresponding to the eigenvalues with s-mallest magnitudes. The coefficients associated with larger eigenvalues are removed to reduce the storage space required for mesh geometry. Figure 5 shows how this method is used to compress a mesh surface by 150 eigenvectors with different Laplacian matrices.

Spectral shape analysis that relies on the spectrum (eigenvalues and eigenfunctions) is one technique in geometric processing since the spectrum is invariant under isometries. [29] shows a recent survey about spectral clustering and [3] provides a classic method. [17] performs mesh segmentation through spectral clustering. [21] introduces a way to improve the result of shape correspondence through eigenvectors of Laplacian matrix.

5.2 Comparison with Previous Methods

We compare our approach with four representative previous methods in this section, i.e., the KL Laplacian [35], the Cotan Laplacian [22], the FEM Laplacian [19],

Model information and running time (in millisecond)						
Moldel	# Vertices	Ours	KL Graph	Cot Graph	FEM	PB-MHB
Half-Sphere (Figure 6)	5,102	219	116	130	145	168,000
Centaur (Figure 1)	15,768	687	206	280	311	831,000
Un-uniform Bear (Figure 5)	28,643	1,098	546	590	671	1,121,000
Un-uniform Ant (Figure 5)	28,562	1,023	512	574	611	1,134,000

Table 1 In this table, we indicate the number of points and timing for constructing different kinds of Laplacian matrix.

and the PB-MHB Laplacian [18]. The programming code of the PB-MHB method was provided by the cited authors. We supplemented this information with our own method and the other methods. Since [33] and our method both discretize Laplacian operator via local quadratic fitting, the performance of these methods are quite similar. Additionally, [33] is a simultaneous work, here we do not compare our method with his.

5.2.1 Timing

Graph-based methods, KL Graph, Cot Graph and FEM methods are very fast since the graph has already been determined and traversal over vertices and edges is easy to achieve. Our method has more specific details since we need to construct the k-nearest neighborhood, approximate quadratic surface in the sense of least square and compute differential quantities. Fortunately, this procedure is not time consuming. Table 1 shows the time cost of all five methods. The time cost of our method is acceptable comparing to graph-based methods and much lesser than PB-MHB. PB-MHB costs too much time in their Voronoi Cell Estimation step.

5.2.2 Precision of Laplacian matrix

To discuss the convergence of each method, we test several functions on two different domains with known analytical solutions of LBO on it. Scale-independent methods, such as KL Graph and Cot Graph, are not able to reflect LBO values on functions. So here we only compare our method with FEM and PB-MHB. The first domain consists of random sampling points from a bounded plane. Test functions are $f = x + y$, $f = x^2 + y^2$ and $f = e^x + e^y$ with ground truth under LBO, i.e., $L_S f = 0$, $L_S f = 4$, $L_S f = e^x + e^y$. Error is computed in both the entire and interior region. Here entire region means we also compute the error on the boundary of the domain. Next, we select a regular unit sphere as our second domain. Two spherical harmonic functions $f = z$ and $f = 3z^2 - 1$ are tested. These functions under LBO are $L_{\mathscr{S}} f = -2z$ and $L_{\mathscr{S}} f = -18z^2 + 6$ since they are eigenfunctions with eigenvalue -2 and -6, respectively. Error is then measured in the sense of both $\|L_M \mathbf{f} - (\Delta_S f)_M\|_2$ and $\|L_M \mathbf{f} - (\Delta_S f)_M\|_\infty$, where $(\Delta_S f)_M$ is the discretization of function $\Delta_S f$ on M. Table 2 shows the result.

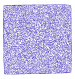
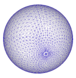
Precision Comparison								
Domain	Test Function		Ours		FEM		PB-MHB	
			Entire	Interior	Entire	Interior	Entire	Interior
	$x+y$	$\|.\|_2$	$1.6(-7)$	$1.4(-7)$	$4.8(1)$	$7.2(-4)$	$2.1(3)$	$1.6(2)$
		$\|.\|_\infty$	$2.8(-6)$	$2.6(-6)$	$3.5(3)$	$6.5(-2)$	$8.0(4)$	$1.2(4)$
	x^2+y^2	$\|.\|_2$	$1.3(-8)$	$1.0(-8)$	$1.7(3)$	$8.6(-1)$	$2.4(3)$	$2.1(2)$
		$\|.\|_\infty$	$4.0(-7)$	$3.3(-7)$	$1.3(5)$	$2.9(1)$	$1.0(5)$	$1.6(4)$
	e^x+e^y	$\|.\|_2$	$7.8(-4)$	$5.7(-4)$	$1.4(3)$	$6.2(-1)$	$1.1(3)$	$3.6(2)$
		$\|.\|_\infty$	$1.4(-2)$	$8.0(-3)$	$9.8(4)$	$1.7(1)$	$7.4(4)$	$1.2(4)$
	z	$\|.\|_2$	$8.0(-5)$	-	$1.5(-3)$	-	$9.2(-5)$	-
		$\|.\|_\infty$	$1.4(-3)$	-	$5.1(-2)$	-	$2.3(-3)$	-
	$3z^2-1$	$\|.\|_2$	$9.3(-5)$	-	$2.4(-3)$	-	$1.2(-4)$	-
		$\|.\|_\infty$	$2.5(-3)$	-	$8.8(-1)$	-	$3.5(-3)$	-

Table 2 The error between functions under real Laplace–Beltrami Operator and discrete Laplacian matrix. The error is measured under both l_2 and l_∞ norm. Three Laplacian matrix, two domains and five functions are involved. The element a(b) in the table stands for $a \times 10^b$.

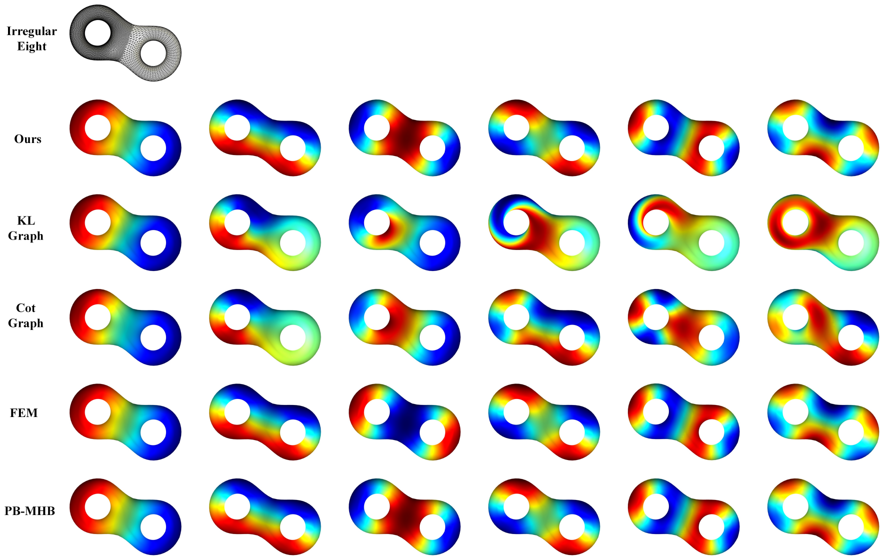


Fig. 4 Eigenvectors associated with first 6 non-zero eigenvectors of all five kinds of Laplacian matrix. From top to bottom, it is the result of our method, KL graph, cot graph, FEM, PB-MHB.

Our method has the best precision; the FEM method is quite sensitive to the boundary; PB-MHB saw the worst result on plane and a much better result on sphere. This is probably why they have claimed in their paper that it would be best if the given surface is close.

5.2.3 Robust to density

Ideally, the Laplacian matrix should not be sensitive to the sampling of the given discrete surface. But unfortunately, KL Graph and Cot Graph are quite sensitive to

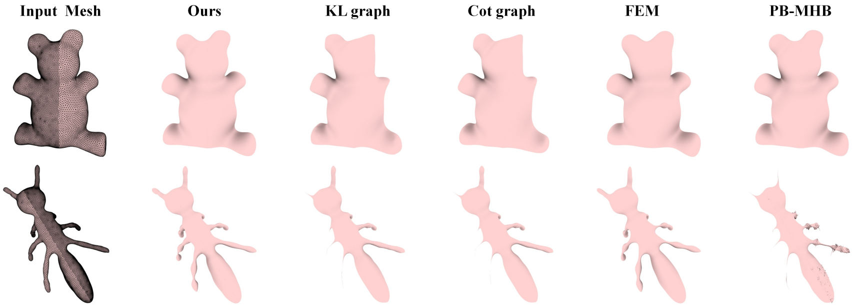


Fig. 5 Compression results for irregular mesh with 150 eigenvectors. From left to right: our method, KL graph, Cot graph, FEM and PB-MHB.

density since they do not contain the information about local stiffness. Figure 4 shows the six leading eigenvectors (corresponding to the six smallest eigenvalues) of a non-uniform Model Eight of different methods. Our method, FEM and PB-MHB are able to stay good; KL Graph and Cot Graph get rather bad and unreliable results. Figure 5 shows the result of a compression algorithm we introduced above with 150 eigenvectors on a non-uniform mesh. Our method and FEM preserves the shape of both irregular Bear and Ant; PB-MHB produces bad results with the ant because there are some ridges on it; KL Graph and Cot Graph fail in both situations.

5.2.4 Robust to Boundary

Graph-based methods are all very sensitive to the boundary because of the loss of graph information at the boundary. So usually one has to modify the Laplacian matrix before using it if the given mesh contains boundaries. On the contrary, our method which is based on an approximation scheme stays stable to the boundary. Table 2 already shows that FEM and PB-MHB get singularity at the boundary numerically. To illustrate what this would bring, we apply classic Laplace global smoothing introduced above on a noisy half-sphere. As shown in Figure 6, only our method achieves a good result. Other graph-based methods all degenerate at the boundary. PB-MHB even gets a singular result in **(f)**.

5.2.5 Conclusion

We have compared our method with four representative methods. Table 3 gives a brief conclusion. First, we compare the properties mentioned in [31]. Notice that we get different conclusions about **(LIN)**, because we believe that FEM and PB-MHB just own **(LIN)** in the interior region. Other conclusions are the same as [31]. The reason we say that PB-MHB does not have the property of **(LOC)**, is that the number of non-zero elements of their matrix is too big. In conclusion, our method owns most of the properties such as with higher precision, robust to boundary which is the problem of all the graph-based methods, and robust to density. What's more, our method does

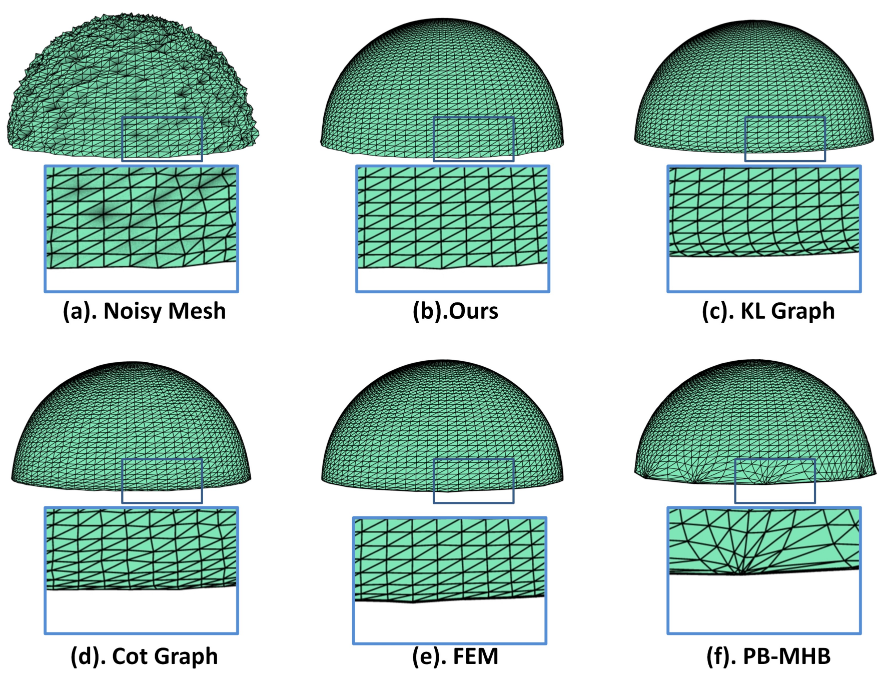


Fig. 6 Global Laplace smoothing on a noisy half-sphere. (**a**) Given noisy mesh; (**b**)-(**f**) the smoothing results of Our method, KL Graph, Cot Graph, FEM and PB-MHB.

Comparison Results					
Property	Ours	KL	Cot	FEM	PB-MHB
(NULL.)	✓	✓	✓	✓	✓
(LIN.)	✓	×	×	×	×
(LOC.)	✓	✓	✓	✓	×
(SYM.)	×	✓	✓	×	×
Work for Point Cloud	✓	×	×	×	✓
Fast to Compute	✓	✓	✓	✓	×
Robust to Density	✓	×	×	✓	✓
Robust to Boundary	✓	×	×	×	×
Scale Dependent	✓	×	×	✓	✓
Easy to Implement	✓	✓	✓	✓	×

Table 3 Final comparison of previous methods. It is marked with checkmark or X-mark for having or not having the property. Our method performs the best.

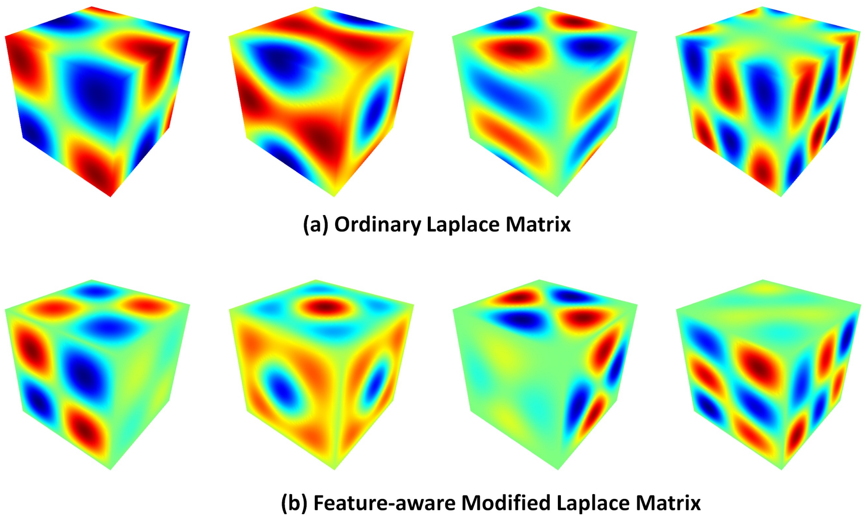

(a) Ordinary Laplace Matrix
(b) Feature-aware Modified Laplace Matrix

Fig. 7 Four eigenvectors of our Laplacian matrix without (**a**) and (**b**) with modify.

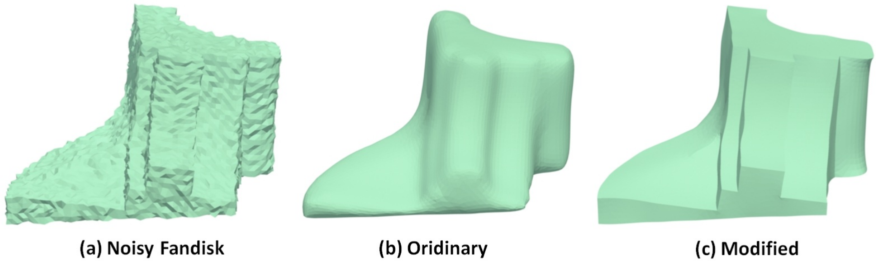


Fig. 8 The result of global smoothing on a noisy Fandisk (**a**), with ordinary (**b**), and modified (**c**) Laplacian matrix.

not care about the form of input, is almost as fast as graph-based methods and quite easy to implement.

6 Feature-aware Laplacian Matrix

Generally, LBO is defined on C^2 surface. Directly using our method on surface with sharp features is unable to reflect C^0 information, as illustrated in Figure 7(**a**). So here we present a strategy to modify the ordinary Laplacian matrix with prior of feature location.

Suppose there are two types of sharp features: crease curve, which brings the discontinuities of first derivatives across itself; corner point, at which the tangent of any passing curve is discontinuous. Corners can also be described as the intersections of several creases. If point $\mathbf{p}$ is identified as a corner, we remove the Laplacian penalty by setting the corresponding row of $L_{\mathscr{M}}$ to be zero. If $\mathbf{p}$ is on a crease, the corresponding row of Laplacian matrix is replaced by the second-order central difference of **p** (using its adjacent neighbors) along the crease.

The eigenvectors of modified Laplacian matrix are illustrated in Figure 7(**b**). These eigenvectors embody the feature information much better than ordinary ones in Figure 7(**a**). As a handy application, we use both ordinary and modified Laplacian matrix on a noisy Fandisk applying the global smoothing algorithm. The result is illustrated in Figure 8. The same smooth parameter λ is set for both cases. The modified (feature-aware) Laplacian matrix preserves the sharp features that the ordinary Laplacian matrix cannot achieve.

7 Conclusion

There are plenty of methods to construct the Laplace–Beltrami operator on a discrete surface. In this paper, we discretize the operator from essential differential quantities. Then we compare our method with four representative previous ones. Numerical experiments show that our Laplacian matrix has much better precision than other methods in most cases. Our Laplacian matrix is robust to boundary which all graph-based ones cannot achieve. At last, we present a strategy to modify the Laplacian matrix into a more feature-aware matrix. This process helps the C^2 operator better understand C^0 information on the surface.

There are also some limitations with our approach, which motivates our future research. Our method still performs a little slower than graph-based methods. We are currently considering the implementation of our method on a GPU to accelerate this process and release a spectral analysis tool for researchers.

Acknowledgements We would like to thank the anonymous reviewers for their constructive comments and Dr. Hector Steele for his help on the paper. We are grateful for one of the anonymous reviewers to point out the paper [33] and for Prof. Hongkai Zhao from University of California at Irvine to point out his papers [15, 16]. The work is supported by the NSF of China (Nos. 11171322, 11271343, 61222206), One Hundred Talent Project of the Chinese Academy of Sciences, and the Program for New Century Excellent Talents in University (No. NCET-11-0881).

References

1. Arnold, D., Falk, R., Winther, R.: Finite element exterior calculus, homological techniques, and applications. Acta Numerica **15**(1), 1–156 (2006)
2. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J. ACM **45**(6), 891–923 (1998)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems 14, pp. 585–591 (2001)
4. Belkin, M., Sun, J., Wang, Y.: Discrete laplace operator on meshed surfaces. In: Proceedings of the twenty-fourth annual symposium on Computational geometry, pp. 278–287. ACM (2008)
5. Belkin, M., Sun, J., Wang, Y.: Constructing laplace operator from point clouds in R^d. In: Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1031–1040 (2009)
6. Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. Computer Graphics Forum (Proc. Eurographics) **21**(3), 209–218 (2002)
7. Desbrun, M., Meyer, M., Schröder, P., Barr, A.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proc. SIGGRAPH, pp. 317–324 (1999)

8. Do Carmo, M.: Differential geometry of curves and surfaces, vol. 2. Prentice-Hall Englewood Cliffs, NJ (1976)
9. Dong, S., Bremer, P., Garland, M., Pascucci, V., Hart, J.: Spectral surface quadrangulation. ACM Transactions on Graphics (Proc. SIGGRAPH) **25**(3), 1057–1066 (2006)
10. Hildebrandt, K., Polthier, K.: Anisotropic filtering of non-linear surface features. Computer Graphics Forum **23**(3), 391–400 (2004)
11. Jolliffe, I.T.: Principal Component Analysis, vol. 487. Springer-Verlag New York (1986)
12. Karni, Z., Gotsman, C.: Spectral compression of mesh geometry. In: Proc. SIGGRAPH, pp. 279 – 286 (2000)
13. Lévy, B., Petitjean, S., Ray, N., Maillot, J., et al.: Least squares conformal maps for automatic texture atlas generation. ACM Transactions on Graphics (Proc. SIGGRAPH) **21**(3), 362–371 (2002)
14. Lévy, B., Zhang, H.: Spectral mesh processing. In: ACM SIGGRAPH 2010 Courses, p. 8. ACM (2010)
15. Liang, J., Lai, R., Wong, T.W., Zhao, H.: Geometric understanding of point clouds using Laplace–Beltrami operator. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 214–221. IEEE (2012)
16. Liang, J., Zhao, H.: Solving partial differential equations on point clouds. SIAM Journal on Scientific Computing **35**(3), A1461–A1486 (2013)
17. Liu, R., Zhang, H.: Mesh segmentation via spectral embedding and contour analysis. Computer Graphics Forum (Proc. Eurographics) **26**(3), 385–394 (2007)
18. Liu, Y., Prabhakaran, B., Guo, X.: Point-based manifold harmonics. IEEE Transactions on Visualization and Computer Graphics **18**(10), 1693–1703 (2011)
19. Meyer, M., Desbrun, M., Schröder, P., Barr, A.: Discrete differential-geometry operators for triangulated 2-manifolds. Visualization and Mathematics **3**(7), 34–57 (2002)
20. Nealen, A., Igarashi, T., Sorkine, O., Alexa, M.: Laplacian mesh optimization. In: Proceedings of ACM GRAPHITE, pp. 381–389 (2006)
21. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: A flexible representation of maps between shapes. ACM Transactions on Graphics (Proc. SIGGRAPH) **31**(4), 30 (2012)
22. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Experimental Mathematics **2**(1), 15–36 (1993)
23. Reuter, M., Biasotti, S., Giorgi, D., Patanè, G., Spagnuolo, M.: Discrete Laplace–Beltrami operators for shape analysis and segmentation. Computers & Graphics **33**(3), 381–390 (2009)
24. Sorkine, O.: Differential representations for mesh processing. Computer Graphics Forum **25**(4), 789–807 (2006)
25. Sorkine, O., Cohen-Or, D.: Least-squares meshes. In: Proceedings of Shape Modeling International, pp. 191–199 (2004)
26. Taubin, G.: A signal processing approach to fair surface design. In: Proc. SIGGRAPH, pp. 351–358. ACM (1995)
27. Vallet, B., Lévy, B.: Spectral geometry processing with manifold harmonics. Computer Graphics Forum (Proc. Eurographics) **27**(2), 251–260 (2008)
28. Vollmer, J., Mencl, R., Mueller, H.: Improved laplacian smoothing of noisy surface meshes. Computer Graphics Forum (Proc. Eurographics) **18**(3), 131–138 (1999)
29. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**(4), 395–416 (2007)
30. Wardetzky, M., Bergou, M., Harmon, D., Zorin, D., Grinspun, E.: Discrete quadratic curvature energies. Computer Aided Geometric Design **24**(8-9), 499–518 (2007)
31. Wardetzky, M., Mathur, S., Kälberer, F., Grinspun, E.: Discrete laplace operators: no free lunch. In: Proc. SGP, pp. 33–37. ACM (2007)
32. Xu, G.: Discrete Laplace–Beltrami operators and their convergence. Computer Aided Geometric Design **21**(8), 767–784 (2004)
33. Xu, G.: Consistent approximations of several geometric differential operators and their convergence. Applied Numerical Mathematics (2013)
34. Xu, Z., Xu, G., Sun, J.G.: Convergence analysis of discrete differential geometry operators over surfaces. In: Mathematics of Surfaces XI, pp. 448–457. Springer (2005)
35. Zhang., H.: Discrete combinatorial laplacian operators for digital geometry processing. In: Proc. SIAM Conference on Geometric Design and Computing, pp. 575 – 592 (2004)
36. Zhang, H., Van Kaick, O., Dyer, R.: Spectral methods for mesh processing and analysis. In: Proceedings of Eurographics State-of-the-art Report, pp. 1–22 (2007)