Geometric Understanding of Point Clouds Using Laplace-Beltrami Operator

Jian Liang
UC Irvine
Irvine, CA
jianl@uci.edu

Rongjie Lai
USC
Los Angeles, CA
rongjiel@usc.edu

Tsz Wai Wong
UC Irvine
Irvine, CA
tszww@uci.edu

Hongkai Zhao
UC Irvine
Irvine, CA
zhao@math.uci.edu

Abstract

In this paper, we propose a general framework for approximating differential operator directly on point clouds and use it for geometric understanding on them. The discrete approximation of differential operator on the underlying manifold represented by point clouds is based only on local approximation using nearest neighbors, which is simple, efficient and accurate. This allows us to extract the complete local geometry, solve partial differential equations and perform intrinsic calculations on surfaces. Since no mesh or parametrization is needed, our method can work with point clouds in any dimensions or co-dimensions or even with variable dimensions. The computation complexity scaled well with the number of points and the intrinsic dimensions (rather than the embedded dimensions). We use this method to define the Laplace-Beltrami (LB) operator on point clouds, which links local and global information together. With this operator, we propose a few key applications essential to geometric understanding for point clouds, including the computation of LB eigenvalues and eigenfunctions, the extraction of skeletons from point clouds, and the extraction of conformal structures from point clouds.

1. Introduction

The representation of geometric entities, such as shapes and surfaces, has been a central problem in 3D modeling. In practice, the majority of these entities are represented by triangular meshes specifying both points and connectivity. In reality, the raw data of these objects are mostly point clouds. For example, geometric data of a solid object in 3D are often obtained by a 3D camera, where the position of a dense array of points are determined using a laser scanner. As such, it is important and desirable to analyze and understand the intrinsic geometry directly on point clouds in many applications. Geometric understanding of point clouds is essential in 3D modeling to determine the geometric quantities on shapes and surfaces, such as the mean and Gaussian curvatures, the surface metric, conformal factors and distortions of surface mappings. From a high level perspective, a correct interpretation of the surface geometry is the key for constructing skeleton from shapes, or computing segmentation on surfaces.

Although this problem is fundamental, most literature in 3D modeling starts by working on given triangular meshes. This is literally taking the geometry from point clouds for granted. Since the connectivity in triangular meshes dictates the complete geometry of shapes and surfaces, a suboptimal triangulation could cause the inaccuracy in the computed geometric quantities. The error is more serious than it may seem, since many geometric quantities of interest are of second or higher order, such as the mean and Gaussian curvatures. More seriously, the topology of the triangulation could be completely different from reality, which seriously affects the high level understanding of shapes.

In applications such as data mining or machine learning, data are usually represented in high dimensions for which laying down a mesh or grid is often impossible. Hence point cloud is the only feasible way to represent the data. However, it is believed that these data points may actually live on a manifold with much lower intrinsic dimensions. Hence, it is an important but challenging task to extract global information and structure directly from point clouds. Mathematically and computationally one can obtain a lot of intrinsic information by studying the behavior of differential equations, such as heat equation, or eigenvalue problem for differential operators, such as Laplace-Beltrami operator, on manifolds [6, 17, 26, 22, 30, 3, 11, 4].

In this paper we first propose a framework for approximating differential operator directly on point clouds based on local surface reconstruction developed in the work by Leung *et al*. [20] for solving PDEs on moving interfaces using grid based particle method. However, a major challenge is to extract global information from the given point clouds due to the lack of connectivity. A possible way to tackle this problem is to create certain special designed tools to recover point clouds' global geometric information from their local information. In this paper, we propose a novel use of the Lalpace-Beltrami (LB) operator as a bridge linking local information and global information. First, generic LB eigenfunctions of surfaces are Morse functions [31], which can tell us the global topological information of surfaces. As a high level understanding of surface's global information, surface skeletons can be constructed from LB eigenfunctions [29]. Second, the LB operator is closely related to surface conformal mapping onto standard domains such as the plane, the unit sphere and the n-torus under the uniformization theorem. Thus the LB operator can be used to construct conformal maps from the given point clouds to standard domains, where their known geometries plus the constructed conformal maps can recover the geometries of the point clouds.

In this paper, we first carefully describe our method to approximate the LB operator on point clouds based on local surface reconstruction by moving least square approximation, which is simple and accurate. By testing the operator on surfaces with known geometry and/or eigenvalues, such as the sphere and the torus, we show that our approximation is more accurate than both method on point clouds and triangular mesh based method. After that, skeleton structures of point clouds are constructed using the first nontrivial LB eigenfunction. Moreover, we illustrate how we can "conformally" map point clouds onto domains with completely known geometry, and hence recover their intrinsic geometry. To demonstrate this, we further test our method on point clouds reconstruction and texture mapping.

2. Theoretical Background

To clearly explain our idea of using the LB operator as bridge to connect surface local and global geometric information, we first introduce some theoretical background of the LB operator and demonstrate how surface global information can be obtained from it. For simplicity, we only consider two-dimensional manifold in $\mathbb{R}^3$, we refer [8] to readers for more details and definitions of the derivatives for more general manifolds.

Let $(\mathcal{M}, g)$ be a smooth surface in $\mathbb{R}^3$ and (s_1, s_2) be its local parametrization near some point $p \in \mathcal{M}$. For a smooth function $f \colon \mathcal{M} \to \mathbb{R}$, the Laplace-Beltrami (LB) operator $\Delta_{\mathcal{M}}$ acting on f near p is given by

$$\Delta_{\mathcal{M}} f = \sum_{i,j=1}^{2} \frac{1}{\sqrt{g}} \frac{\partial}{\partial s_i} \left(\sqrt{g} g^{ij} \frac{\partial f}{\partial s_j} \right) \tag{1}$$

where the coefficients g^{ij} are the components of the inverse of the metric tensor $G = [g_{ij}]$ and $g = \det(G)$. The above intrinsic LB operator is self adjoint and elliptic, hence its spectrum is discrete. Denote the eigenvalues of $\Delta_{\mathcal{M}}$ as $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots$ and the corresponding eigenfunctions as $\phi_0, \phi_1, \phi_2, \cdots$ such that $-\Delta_{\mathcal{M}} \phi_n = \lambda_n \phi_n, n = 0, 1, 2, \cdots$. We call $\{\lambda_i, \phi_i\}|_{i=0}^{\infty}$ an eigen-system of the LB operator on $(\mathcal{M}, g)$ [10].

According to the above formula (1), the action of LB operator on any function is only determined by the local geometric information of $\mathcal{M}$. However, the eigenfunctions of the LB operator can be utilized to extract surface global information. One of evidences is a famous theorem proved by Unlenbeck [31], generic LB eigenfunctions of surfaces are morse functions, which enable LB eigenfunctions as descriptors for the topologies of the surfaces. As a demonstration of this theorem, surface skeleton, which is an intuitive graph representation and high level realization of surface global information [5], can be intrinsically constructed using the Reeb graph of the first nontrivial LB eigenfunction due to the recent work in [29]. In addition, a new surface quadrangulation can be also obtained using the Morse-Smale complex of certain LB eigenfunction [13]. Moreover, LB eigenfunctions can be viewed as either global or local embedding to analyze surface geometric structures [30, 7, 28, 17, 18].

Furthermore, the LB operator is also closely related to harmonic maps between two surfaces. In particular, it is the key part to construct global conformal structures for genus-0 surfaces by minimizing the harmonic energy function [27].

Given two Riemann surfaces $\mathcal{M}$ and $\mathcal{N}$ with metrics g and $\widetilde{g}$ respectively. A diffeomorphism $f \colon \mathcal{M} \to \mathcal{N}$ introduces a new Riemannian metric $f^*(\widetilde{g})$ on $\mathcal{M}$, induced by f and $\widetilde{g}$, called the **pull back metric**. We say that the map f is **conformal** if $f^*(\widetilde{g}) = e^{2u} g$, where $u : \mathcal{M} \to \mathbb{R}$ is a smooth function on $\mathcal{M}$. A parametrization $\varphi \colon \mathbb{R}^2 \to \mathcal{M}$ is a **conformal parameterization** if φ is a conformal map. Intuitively, a map is conformal if it preserves the inner product of the tangent vectors up to a scaling factor, called the conformal factor e^{2u}. An immediate consequence is that every conformal map preserves angles.

For genus-0 surfaces, conformal map is closely related to harmonic map, which is defined to be a critical point of the **harmonic energy**:

$$\mathcal{E}(f) = \frac{1}{2} \int_{\mathcal{M}} e(f) \mathrm{d}\mathcal{M} \tag{2}$$

where $e(f) = \|\mathrm{d}f\|^2 = \sum_{i,j=1,2} g^{ij} \langle f_* \partial_{x^i}, f_* \partial_{x^j} \rangle_{\widetilde{g}}$ is the energy density and $f_* \partial_{x^i}$ is the standard push-forward map.

For a map $f : \mathcal{M} \to \mathcal{N}$ between two genus-0 surfaces $\mathcal{M}$ and $\mathcal{N}$, f is conformal if and only if it is a harmonic map [27]. Therefore, the global conformal structure of genus-0 surface $\mathcal{M}$ can be obtained by computing a conformal map between $\mathcal{M}$ and the unite sphere S^2. More importantly, the optimizer of the above harmonic energy can be essentially obtained by gradient descent approaches, which are highly related to computation of the LB operator of $\mathcal{M}$ [15, 14, 19]. Therefore, in our case of point cloud representations of surfaces, we can expect to use LB operators to extract the global conformal structures for point clouds of genus-0 surfaces.

In summary, we illustrate that LB operator can be viewed as a bridge between local structures and global structures for point clouds. In the rest of this paper, we will first show our numerical approximation of LB operator for point cloud representations of manifolds, and then demonstrate its capability to obtain global information of the given point cloud by constructing its global skeleton structure and conformal structure.

3. Approximation of Laplace-Beltrami Operator Based on Local Reconstruction

Given a point cloud $P = \{p_i | i = 1, \dots, N\}$ sampled from a smooth surface/manifold, the first step of geometric understanding is the construction of local geometry, which allows intrinsic computations on surfaces, such as differentiation. In particular, we are interested in defining the LB operator on P. There are several different ways to construct the LB operator due to different representations of surfaces. Mesh based approaches are proposed in [25, 12, 33]. However, these methods typically assume a good triangulation, which may be a challenging and costly task itself. Closest point method is proposed to construct the LB operator in [24]. The requirements of surface implicit representations and non-intrinsic uniform underlying volumetric grid make their approximation inefficient. Point Cloud Data (PCD) Laplace based on an integral approximation of the LB operator is introduced in [4] . Although PCD Laplace is the first provable reconstruction of the LB operator for point clouds, it requires quite strict sampling conditions on the point clouds and is not very accurate in practice. In this section, we construct our LB operator based on local moving least square approximation. For simplicity, we use two-dimensional manifold in $\mathbb{R}^3$ to illustrate our approach, it is straightforward to generalize the approach to higher dimensions, more details can be found in [23]. Comparison of our approach to other methods will be discussed in section 5.1.

As shown in equation (1), there are two necessary requirements, namely a local parametrization (s_1, s_2) near every point $p_i \in P$, and the derivatives of smooth functions near p_i. Since there is no connectivity information on P, for each point, we rely on its K nearest neighbors (KNN, including itself) for local geometric understanding. LB operator takes a function $f : \mathcal{M} \to \mathbb{R}$ as input and produces another function $\Delta_{\mathcal{M}} f : \mathcal{M} \to \mathbb{R}$ as output. Numerically, a function f defined on the manifold can be represented as a N-dimensional vector $F = [f(p_1), \dots, f(p_N)]^T$. Since LB operator is a linear operator, the discrete LB operator can be represented by an $N \times N$ matrix L.

In this section, we propose an algorithm which constructs our discrete LB operator for point clouds without connectivity information. Our method proceeds in three steps. First, we define at each point p_i a local coordinate system. Then we use moving least square (MLS) to calculate a bivariate polynomial which best approximates the surface locally. In this paper we use quadratic polynomial although higher order polynomial can be used if necessary. In the last step, we modify the classical MLS by introducing a special weight function to locally approximate any function f defined on $\mathcal{M}$.

3.1. Computation of Local Coordinate System

Denote the indices set of KNN of each point $p_i \in P$ by $N(i)$, it is widely accepted and justified that the normal and the local coordinates of $\mathcal{M}$ at p can be approximated well using principal component analysis (PCA) [16, 9]. Using the covariance matrix P_i of $N(i)$, defined by:

$$P_i = \sum_{k \in N(i)} (p_k - c_i)^T (p_k - c_i) \tag{3}$$

we can estimate some local geometric information near p_i. Here, c_i is the local barycenter $c_i = \frac{1}{K} \sum_{k \in N(i)} p_k$. The eigenvectors (e_1^i, e_2^i, e_3^i) of P_i form an orthogonal frame associated with eigenvalues $(\lambda_1^i, \lambda_2^i, \lambda_3^i)$ with $\lambda_1^i \geq \lambda_2^i \geq \lambda_3^i \geq 0$. For ease of computation, p_i is always taken as the origin of the local coordinate system, and vectors (e_1^i, e_2^i, e_3^i) form the orthogonal axes near p_i. In this way, we have defined a local coordinate system $\langle p_i; e_1^i, e_2^i, e_3^i \rangle$ at each point in P. KNN of p_i have local coordinates (x_k^i, y_k^i, z_k^i), which will be used for surface and function approximations.

3.2. Surface Approximation Using Moving Least Square

Approximation of surfaces using MLS has been proposed in [21, 1], and is shown to be effective in modeling local surface geometry. With important geometric quantities such as the mean and Gaussian curvatures being of second order, it suffices to compute a degree 2 polynomial which fits best to the KNN at each point. Once a local coordinate system for a point p_i is constructed, a local degree 2 bivariate polynomial $z_i(x, y)$ is approximated by minimizing the following weighted sum:

$$\sum_{k \in N(i)} w(\|p_k - p_i\|) \left(z_i(x_k^i, y_k^i) - z_k^i \right)^2 \tag{4}$$

where $w(\cdot)$ is some positive weight function and (x_k^i, y_k^i, z_k^i) are local coordinates of point p_k in the KNN of p_i. A typical choice is $w(d) = \exp(-\frac{d^2}{h^2})$ and we choose $h = \max_{k \in N(i)} \|p_k - p_i\|$. $\Gamma_i = (x, y, z_i(x, y))$ is thus a smooth representation of the surface near the point p_i under local coordinate system $\langle p_i; e_1^i, e_2^i, e_3^i \rangle$.

With this local parametric approximation, we can easily compute the metric tensor and other important quantities.

For example, the LB operator (1) can be written as a linear combination of derivatives on the surface, given by

$$\Delta_{\mathcal{M}} f = \alpha_1 \frac{\partial f}{\partial x} + \alpha_2 \frac{\partial f}{\partial y} + \alpha_3 \frac{\partial^2 f}{\partial x^2} + \alpha_4 \frac{\partial^2 f}{\partial x \partial y} + \alpha_5 \frac{\partial^2 f}{\partial y^2} \quad (5)$$

where α_i's are obtained by expanding and simplifying equation (1) and they only depend on coefficients of local surface approximation $z_i(x, y)$. In the next subsection, we show how this can be combined with the local approximation of functions on $\mathcal{M}$ to give a linear operator representing the LB operator.

3.3. Function Approximation Using Moving Least Square

In order to perform intrinsic computations on $\mathcal{M}$, such as differentiation or computing the LB operator acting on a function on $\mathcal{M}$, it is necessary to approximate functions on $\mathcal{M}$. Again, we use MLS to locally approximate a function f on $\mathcal{M}$ by constructing a degree 2 bivariate polynomial approximation $F_i(x, y)$ which minimizes the following weighted sum:

$$\sum_{k \in N(i)} w(\|p_k - p_i\|) \left(F_i(x_k^i, y_k^i) - f_k\right)^2 \quad (6)$$

where $f_k = f(p_k)$. Once we have the approximation $F_i(x, y)$, computing derivatives becomes straight forward.

Since F_i is a degree 2 bivariate polynomial, it takes the form $F_i(x, y) = c_1^i + c_2^i x + c_3^i y + c_4^i x^2 + c_5^i xy + c_6^i y^2$. Finding the minimizer of (6) amounts to setting its partial derivatives to zero. This gives the following linear system:

$$\sum w_k V_k^i (V_k^i)^T C^i = \sum w_k V_k^i f_k \quad (7)$$

where $w_k = w(\|p_k - p_i\|)$, $C^i = \left[c_1^i, c_2^i, c_3^i, c_4^i, c_5^i, c_6^i\right]^T$ and $V_k^i = \left[1, x_k^i, y_k^i, (x_k^i)^2, x_k^i y_k^i, (y_k^i)^2\right]^T$. We can write solution to this system as $C^i = M^i F$, where M^i is some $6 \times N$ matrix. Since we set p_i to be the origin of the local coordinate system, partial derivatives of f can be easily computed. For example, $\frac{\partial f}{\partial x}(p_i) = c_2^i$. Now, using equation (5), we can easily write the approximation of the LB operator as $\Delta_{\mathcal{M}} f(p_i) = L_i F$, where L_i is some row vector. Therefore, the i-th row of our MLS LB operator is simply L_i.

The choice of weight $w(d)$ in equation (6) is crucial to make the discrete LB operator accurate and robust with respect to noise and non-uniform data especially since the eigenvalue problem is a non-local problem. After extensive testing, we find that some commonly used weight functions, such as $w(d) \equiv 1$, $w(d) = \exp(-\frac{d^2}{h^2})$, $w(d) = (1 - d/h)^4 (4d/h + 1)$ or $w(d) = \frac{1}{d^2 + \varepsilon^2}$, do not work very well for our problem. For $w \equiv 1$, it does not give good accuracy since the construction is not localized enough. The other three give accurate local approximation for the LB operator. However, they lack enough overlap among local reconstructions to enforce communication among data points especially when the point cloud is non-uniform. This makes them fail when we use them to compute global geometric information, such as eigenvalues and eigenfunctions. According to our experiments, we find that best results are achieved by choosing a special weight function as the following:

$$w(d) = \begin{cases} 1, & \text{if } d = 0 \\ 1/K, & \text{if } d \neq 0 \end{cases} \quad (8)$$

which keeps a balance between accurate local approximation and global communication.

Remark 1 *We refer readers to [23] for convergence of our MLS LB operator and more details in high-dimensional manifold case. Also, [23] gives a constraint quadratic optimization approach for approximating LB operator which guarantees diagonal dominant property for the resulting matrix, which preserves some desired property of the continuous operator.*

4. Geometric Understanding of Point Clouds

In this section, we use our discrete LB operator to solve some interesting problems related to geometric understanding of point clouds. We first solve the well-studied LB eigenvalue problem. Then, we use the first non-trivial LB eigenfunction to construct global skeleton structures for point clouds. Finally, based on the recent algorithm proposed in [19], we adapt it to find conformal mappings from point clouds of simply connected closed surfaces onto spheres. This allows us to find the conformal structures on these surfaces, which give the complete geometric information for point clouds.

4.1. Solving Eigenvalue Problem

The eigenvalue problem for a smooth manifold is defined as finding eigenvalues and eigenfunctions of the corresponding LB operator $-\Delta_{\mathcal{M}} \phi = \lambda \phi$. Using our MLS LB operator defined above, eigenvalue problem for point clouds sampled from manifolds becomes

$$-L\Phi = \lambda \Phi \quad (9)$$

where $\Phi = [\phi(p_1), \cdots, \phi(p_N)]^T$ is the N-vector representation of ϕ. Solving the above spectral decomposition problem is a well-known problem in numerical linear algebra. For example, in MATLAB we can use function *eig()* to compute the complete decomposition or the function *eigs()* to compute part of the spectrum. Comparison of our approach with Point Cloud Data (PCD) Laplace [4] and finite element method [12] will be discussed in section 5.1.

4.2. Construction of Skeletons from Point Clouds

Based on the first nontrivial eigenfunction, a novel approach of computing surface skeletons is proposed in [29], where the construction of skeleton is realized by constructing Reeb graphs from the first nontrivial LB eigenfunction. Their idea is to define the Reeb graph $R(\phi_1)$ of ϕ_1 as the quotient space defined through the equivalent relation $x \sim y$ if $\phi_1(x) = \phi_1(y)$ for $x, y \in \mathcal{M}$. We adapt their method to construct skeletons for point clouds without connectivity information. We solve eigenvalue problem (9) to obtain the first non-trivial eigenfunction ϕ_1. By connecting barycenters of level sets of ϕ_1, one can find skeleton of the point cloud.

Given the first nontrivial LB eigenfunction ϕ_1 and its local polynomial approximation Φ_1 using approach proposed in section 3.3, the level contour $\phi_1 = c$ can be computed. For each p_i such that $\phi_1(p_i) \in [c-\delta, c+\delta]$, we find a point p that is the minima of the following:

$$\min_p \|p - p_i\|^2 + \beta(\Phi_1(p) - c)^2 \tag{10}$$

By choosing δ small and β large, we find a point that is close to p_i and its function value is close to c. Once all level contours of ϕ_1 are computed, we take their barycenter as the skeleton point of that level set. By connecting these barycenters, the skeleton structure for the given point cloud can be obtained.

4.3. Construction of Conformal Mappings from Point Clouds

As long as the LB operator can be locally approximated for a given point cloud, we can extract global geometric structures of the given point cloud. As an example, we adapt the efficient harmonic energy minimization algorithm in [19] to study global conformal structures for point clouds of genus-0 surfaces.

Given a genus-0 surface $\mathcal{M}$, its conformal structure can be obtained by conformal map from $\mathcal{M}$ to the sphere S^2. Gu *et al*. [15, 14] propose realizing conformal map between genus-0 surfaces as a harmonic energy minimization problem, which is implemented by gradient descent of harmonic energy and projection back to the sphere. More recently, a more efficient algorithm for harmonic energy minimization problem from genus-0 surfaces to sphere is proposed in [19], where authors consider the harmonic energy minimization problem from $\mathcal{M}$ to S^2 to the following optimization problem with spherical constraints:

$$\begin{aligned} \min_{\vec{F}=(f_1,f_2,f_3)} \; & \mathcal{E}(\vec{F}) = \frac{1}{2}\int_{\mathcal{M}} \sum_{i=1,2,3} \|\nabla_{\mathcal{M}} f_i\|^2 \mathrm{d}\mathcal{M} \\ \text{s.t.} \quad & \|\vec{F}(x)\| = \sqrt{f_1^2 + f_2^2 + f_3^2} = 1, \qquad \forall x \in \mathcal{M} \end{aligned} \tag{11}$$

Write $H = \nabla \mathcal{E}(\vec{F}) = -(\Delta_{\mathcal{M}} f_1, \Delta_{\mathcal{M}} f_2, \Delta_{\mathcal{M}} f_3)$ as the Fréchet derivative of $\mathcal{E}(\vec{F})$ with respect to $\vec{F}$ and define a skew-symmetric matrix $A := H\vec{F}^* - \vec{F}H^*$. Similar as the algorithm for orthogonality constraint proposed in [32], a update path $\vec{Y}[\tau] := \vec{F} - \frac{\tau}{2}A(\vec{F}+\vec{Y}[\tau])$ is proposed in [19], which has the following property:

Proposition 1 *For every τ, $Y[\tau]$ satisfies $\|\vec{Y}[\tau](x)\| = 1$ point-wise. In addition, it is given in the closed-form*

$$\vec{Y}[\tau] = \left(I + \frac{\tau}{2}A\right)^{-1}\left(I - \frac{\tau}{2}A\right)\vec{F} \tag{12}$$

which can be computed as $\vec{Y}[\tau] = \alpha[\tau]\vec{F} + \beta[\tau]H$, where

$$\alpha[\tau] = \frac{\left(1 + \frac{\tau}{2}\langle \vec{F}(x), H(x)\rangle\right)^2 - \left(\frac{\tau}{2}\right)^2 \|\vec{F}(x)\|^2 \|H(x)\|^2}{1 - \left(\frac{\tau}{2}\right)^2 \langle \vec{F}(x), H(x)\rangle^2 + \left(\frac{\tau}{2}\right)^2 \|\vec{F}(x)\|^2 \|H(x)\|^2}$$

$$\beta[\tau] = \frac{-\tau\|\vec{F}(x)\|^2}{1 - \left(\frac{\tau}{2}\right)^2 \langle \vec{F}(x), H(x)\rangle^2 + \left(\frac{\tau}{2}\right)^2 \|\vec{F}(x)\|^2 \|H(x)\|^2}$$

Note that the key of the update path $\vec{Y}[\tau]$ is computation of H. Our problem of computing conformal map from point clouds of genus-0 surfaces to the sphere can be view as the discretization of the continuous problem (11). Using our approximation of LB operator, the above update path can be fully adapted to compute conformal map of point clouds. Furthermore, nonmonotone curvilinear search with an initial step size determined by the Barzilai-Borwein formula [2, 34] can be used to speed up the whole process, which is exact the same as the algorithm used in [19]. Moreover, a conformal correction technique based on weighted LB eigensystem is introduced in [19] to remove folding issues for mapping surfaces with long and sharp features. Their technique can also be adapted here to compute conformal map of point clouds for genus-0 surfaces with complicated structures.

5. Experimental Results

In this section, we present the results of our algorithm on various point clouds data. First, we compare our approximation of LB operator with the Point Cloud Data (PCD) Laplacian [4] and mesh based method [12], and show that our method obtains better accuracy compared to these approaches. The eigenfunctions of several point clouds are presented to show the robustness of our method. Next, we demonstrate that our numerically computed LB eigenfunctions can be used to extract skeletons from point clouds. Finally, we show that our method can be used to compute conformal mapping effectively, and obtain triangular meshes and texture mapping directly from point clouds. This shows that our method can be applied to extract both geometric information and high level understanding of point clouds.

sample size	1002	1962	4002	7842	16002
MLS Laplacian					
$\lambda = 20$	0.0516	0.0306	0.0135	0.0046	0.0020
$\lambda = 72$	0.1541	0.0774	0.0476	0.0298	0.0138
PCD Laplacian					
$\lambda = 20$	0.0773	0.0487	0.0431	0.0411	0.0403
$\lambda = 72$	0.1391	0.1174	0.1128	0.1108	0.1100
finite element method					
$\lambda = 20$	0.0165	0.0085	0.0042	0.0021	0.0010
$\lambda = 72$	0.0660	0.0342	0.0169	0.0087	0.0043

Table 1. $E_{\max}$ errors for uniform sampling on unit sphere.

5.1. Comparison of Laplace-Beltrami Operator Approximated by Different Methods

We first test our MLS LB operator for LB eigenvalue problems on point clouds of sphere and torus, where the exact closed form solutions are known. For sphere, we test point clouds uniformly sampled from unit sphere. For the regular point cloud of the torus, points are sampled regularly and evenly from its standard parametric form using two parameters.

On the unit sphere, the exact value of the n-th eigenvalue is given by $\lambda_n = n(n+1)$, with multiplicity $2n+1$. To measure the error of the MLS LB operator in computing eigenvalues, we compute the normalized error $E_{\max,n} = \max(\frac{|\tilde{\lambda}_{n,i} - \lambda_n|}{\lambda_n})$, where $\tilde{\lambda}_{n,i}$'s are the eigenvalues computed from the MLS LB matrix L for eigenvalue λ_n, and i runs over each multiplicity. $E_{\max,n}$ represents the worst possible error in computing λ_n. We show $E_{\max,n}$ for $\lambda = 20$ and 72 in Table 1 of our MLS method, the method in Belkin *et al*. [4], and mesh based method [12]. It can be seen that the error of our approach is of the same order as the mesh based method, while we are more accurate by at least an order compared to the PCD Laplacian method. The results for non-uniform point clouds sampled on the unit sphere are similar.

We also test our method by looking at how accurately our method computes the Laplacian of functions on point clouds of the torus with major radius 1 and minor radius 0.2. "Ground truth" eigenvalues are not available, instead we compute the L_∞ error for the functions $f = z, z^2, e^z$ on the torus, where z is the third coordinate in 3D, and the closed forms of their surface Laplacians are known. The L_∞ error is defined as $E_\infty = \frac{\|\widehat{U} - U\|_\infty}{\|U\|_\infty}$, where U is the known value of Δf in its closed form, and $\widehat{U}$ is the result by applying different discrete LB operators on f. In this test, both errors in the MLS LB and the PCD Laplacian operator decrease at a rate of around $1/N$. However, our method achieves consistently better results than the PCD Laplacian in this test. The error for mesh-based method, however, blows up since some triangles in the mesh are degenerate. The actual values for the errors are shown in Table 2. This shows that our method is the best among the three.

sample size	1080	1920	4320	8892	17280
MLS Laplacian					
$f = z$	0.2054	0.1108	0.0457	0.0257	0.0109
$f = z^2$	0.1039	0.0842	0.0344	0.0388	0.0209
$f = e^z$	0.1699	0.0926	0.0383	0.0212	0.0104
PCD Laplacian					
$f = z$	0.1163	0.1558	0.0872	0.0961	0.0808
$f = z^2$	0.0993	0.1239	0.1150	0.1240	0.1002
$f = e^z$	0.1070	0.1512	0.0829	0.0841	0.0775
finite element method					
$f = z$	0.4761	0.4996	0.9354	9.3957	7.0114
$f = z^2$	0.4659	0.4791	0.9965	5.8476	6.4285
$f = e^z$	0.4270	0.4948	0.8735	6.8495	4.8258

Table 2. E_∞ errors for uniform sampling on torus.

sample size	3375	8000	15625	27000	46656
MLS Laplacian					
$f = x$	0.0448	0.0249	0.0158	0.0109	0.0076
$f = x^2$	0.00095	0.00043	0.00023	0.00014	0.00009
$f = e^x$	0.0263	0.0146	0.0096	0.0069	0.0046

Table 3. E_∞ errors for uniform sampling on T^3.

We conclude this subsection by presenting a high dimension example to show the applicability of our approach to high dimensional point clouds. Consider eigenvalue problem for flat 3-torus T^3, a three dimensional manifold embedded in $\mathbb{R}^6$ parameterized as

$$\Gamma = (\cos\alpha, \sin\alpha, \cos\beta, \sin\beta, \cos\theta, \sin\theta) \qquad (13)$$

with $\alpha, \beta, \theta \in [0, 2\pi]$. Using our approach, the manifold is locally approximated as

$$\begin{aligned}\Gamma = \quad & (x_1, x_2, x_3, y_1(x_1, x_2, x_3), \\ & y_2(x_1, x_2, x_3), y_3(x_1, x_2, x_3))\end{aligned} \qquad (14)$$

and the eigenfunction ϕ is locally approximated as $\Phi(x_1, x_2, x_3)$, where y_1, y_2, y_3 and Φ are 3-dimensional degree 2 polynomials. Our method applies to this 3D manifold in $\mathbb{R}^6$ in a straight forward way. "Ground truth" eigenvalues and their multiplicities are not available, instead we use some test functions to measure the error. We compute the L_∞ error for $\Delta_{\mathcal{M}} f$ for the functions $f = x, x^2, e^x$ on T^3, where x is the first coordinate in $\mathbb{R}^6$, and the closed forms of their surface Laplacians are known. The results are reported in Table 3.

5.2. Applications of Our Method in Geometric Understanding of Point Clouds

In this subsection, we present applications in geometric understanding of point clouds, namely the computation of the LB eigenfunctions from point clouds, the extraction of skeletons from point clouds, and computation of conformal structures from point clouds.

5.2.1 Computation of Laplace-Beltrami Eigenfunctions from Point Clouds

Since the MLS LB operator we defined has the same size as the discrete LB operator using finite element method, computation of eigenfunctions on point clouds using our method is as efficient as the finite element method, with the additional advantage that connectivity is unnecessary and the accuracies are comparable. We computed some sample eigenfunctions from point clouds of different geometries, including the sphere, the torus, a complicated knot, the Stanford Bunny, the 2-torus and the Armadillo. As shown in Figure 1, the computed eigenfunctions color-coded on these shapes are smooth and our method works without knowing connectivity information.

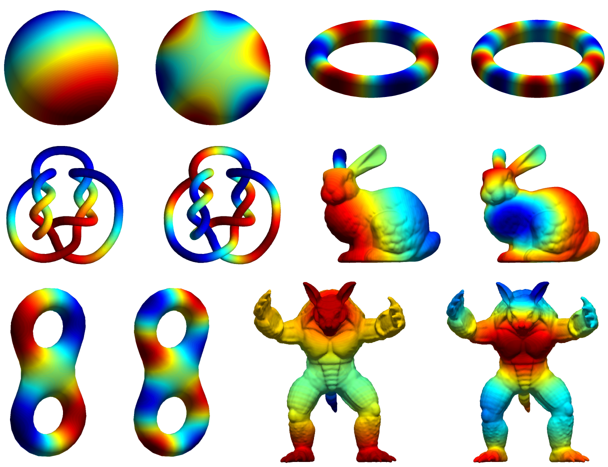

Figure 1. Eigenfunction examples. The above figures show 2 non-trivial eigenfunctions computed by our MLS LB operator for several point clouds data.

5.2.2 Computation of Skeletons from Point Clouds

Using the first non-trivial eigenfunction computed by the MLS LB operator, we compute skeletons for several point clouds using algorithm proposed in section 4.3. As shown in Figure 2, the extracted skeletons agree well with our intuition of the most essential feature, or a "backbone" constituting a shape. This shows that our method is able to achieve high level understanding of the raw data of point clouds, which are only myriads of points in the 3D space in a low level sense.

5.2.3 Computation of Conformal Structures from Point Clouds

Finally, we demonstrate the use of the MLS LB operator to recover the geometry from point clouds. This is achieved by computing a conformal mapping from a point cloud onto a

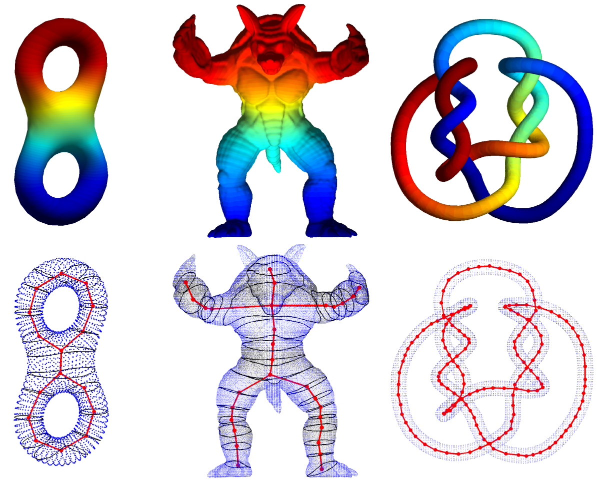

Figure 2. Skeletons examples. The first row: surfaces are color-coded by their first non-trivial eigenfunctions ϕ_1 computed by MLS LB operator. The second row: skeletons computed by our proposed algorithm. Blue dots are data points, black curves are the level contours of ϕ_1, red dots are centers of the corresponding level set curves and red lines are the resulting skeletons.

sphere using the algorithm in Section 4.3. Once a conformal mapping is computed, the point cloud directly inherits its full connectivity and conformality information from the unit sphere. As shown in Figure 3, we apply our algorithm to the point clouds of a fish and the Stanford bunny. After both connectivity and conformality are found, we apply a texture mapping on the point clouds to show their conformal structures. Locally, the checkerboard patterns are very close to squares, indicating that our algorithm successfully captures the complete geometric information on these point clouds.

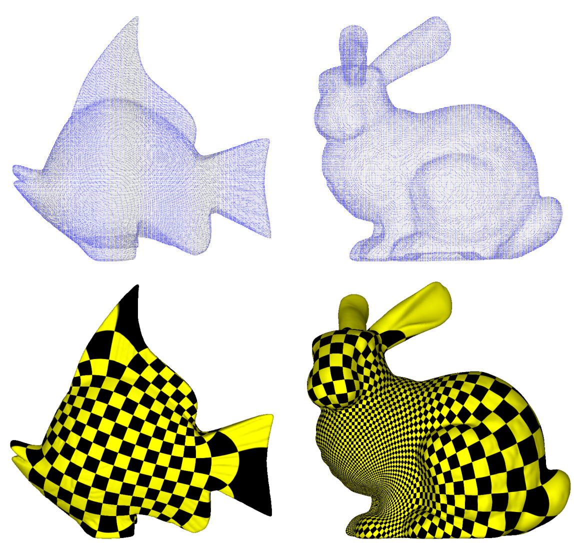

Figure 3. Texture mapping examples. The first row: point clouds of fish and bunny. The second row: texture mappings computed by our proposed algorithm.

6. Conclusion

In this paper, we develop a simple, efficient and accurate algorithm for geometric understanding for point clouds using Lalplace-Beltrami (LB) operator. Our approximation of differential operator is based on local moving least square reconstruction of the surface. We show its advantages over two other methods in the literature. Using LB operator, we demonstrate several applications on point clouds, where traditionally, global triangulations are required, including the computation of LB eigenfunctions, the extraction of skeletons from point clouds, and the computation of conformal structures and complete geometries from point clouds. These promising results represent a novel approach of 3D modeling and geometric understanding without triangular meshes. In the future, we will look into more applications of this approach, such as solving PDEs on point clouds, shape classification and analysis for point clouds.

Acknowledgement

Jian Liang, Tsz Wai Wong and Hongkai Zhao's work are supported by ONR grant N00014-11-1-0602, ARO/MURI W911NF-07-1-0185, and NGA NURI HM1582-10-1-0012. Rongjie Lai's work is supported by Zumberge Individual Award from USC's James H. Zumberge Faculty Research and Innovation Fund.

References

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *Proceedings of the Conference on Visualization '01*, pages 21–28, Washington, DC, USA, 2001. 3

[2] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988. 5

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2002. 1

[4] M. Belkin, J. Sun, and Y. Wang. Constructing Laplace operator from point clouds in $\mathbb{R}^d$. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1031–1040, Philadelphia, PA, USA, 2009. 1, 3, 4, 5, 6

[5] H. Blum and R. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180, 1978. 2

[6] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro. A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, 89(2-3):266–286, 2010. 1

[7] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. *CVPR*, 2010. 2

[8] M. P. D. Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976. 2

[9] E. Castillo and H. Zhao. Point cloud segmentation via constrained nonlinear least squares surface normal estimates. Technical report, UCLA, 2009. 3

[10] I. Chavel. *Eigenvalues in Riemannian Geometry*. Academic press. INC, 1984. 2

[11] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21:6–30, 2006. 1

[12] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Discrete differential geometry operators in nd. *In Proc. VisMath'02 Berlin Germany*, 2002. 3, 4, 5, 6

[13] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Spectral surface quadrangulation. *ACM SIGGRAPH*, 2006. 2

[14] X. Gu, Y. Wang, T. F. Chan, P. Thompson, and S. T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. on Medical Imaging*, 23:949–958, 2004. 2, 5

[15] X. Gu and S. T. Yau. Global conformal surface parameterization. *Symposium on Geometry Processing*, pages 127–137, 2003. 2, 5

[16] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, pages 71–78, New York, USA, 1992. 3

[17] R. Lai, Y. Shi, K. Scheibel, S. Fears, R. Woods, A. W. Toga, and T. F. Chan. Metric-induced optimal embedding for intrinsic 3D shape analysis. *CVPR*, 2010. 1, 2

[18] R. Lai, Y. Shi, N. Sicotte, and A. W. Toga. Automated corpus callosum extraction via laplace-beltrami nodalâ parcellation and intrinsic geodesic curvature flows on surfaces. *ICCV*, 2011. 2

[19] R. Lai, Z. Wen, W. Yin, X. Gu, and L. M. Lui. Fast and robust algorithms for harmonic energy minimization on genus-0 surfaces. *UCLA CAM Reports*, 60, 2011. 2, 4, 5

[20] S. Leung, J. Lowengrub, and H. Zhao. A grid based particle method for solving partial differential equations on evolving surfaces and modeling high order geometrical motion. *J. Comput. Phys.*, 230:2540–2561, April 2011. 1

[21] D. Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67:1517–1531, 1998. 3

[22] B. Levy. Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. In *IEEE International Conference on Shape Modeling and Applications, invited talk*, 2006. 1

[23] J. Liang and H. Zhao. Solving partial differential equations on point clouds. Technical report, UCLA, CAM Reports 2012. 3, 4

[24] C. B. Macdonald, J. Brandman, and S. J. Ruuth. Solving eigenvalue problems on curved surfaces using the closest point method. *Journal of Computational Physics*, 2011. 3

[25] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III. (H.C. Hege and K. Polthier, Ed.) Springer Verlag*, pages 35–57, 2003. 3

[26] D. Raviv, A. M. Bronstein, M. M. Bronstein, R. Kimmel, and N. Sochen. Affine-invariant diffusion geometry for the analysis of deformable 3d shapes. *CVPR*, pages 2361–2367, 2011. 1

[27] R. Schoen and S.-T. Yau. *Lectures on Harmonic Maps*. International Press, 1994. 2

[28] Y. Shi, R. Lai, R. Gill, D. Pelletier, D. Mohr, N. Sicotte, and A. W. Toga. Conformal metric optimization on surface (CMOS) for deformation and mapping in laplace-beltrami embedding space. *MICCAI*, 2011. 2

[29] Y. Shi, R. Lai, S. Krishna, N. Sicotte, I. Dinov, and A. W. Toga. Anisotropic Laplace-Beltrami eigenmaps: Bridging Reeb graphs and skeletons. In *Proc. MMBIA*, 2008. 2, 5

[30] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provabley informative multi-scale signature baded on heat diffusion. *Eurographics Symposium on Geometry Processing*, 2009. 1, 2

[31] K. Uhlenbeck. Generic properties of eigenfunctions. *Amer. J. of Math.*, 98(4):1059–1078, 1976. 2

[32] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *UCLA CAM Reports*, 77, 2010. 5

[33] G. Xu. Convergent discrete laplace-beltrami operator over triangular surfaces. *Proceedings of Geometric Modelling and Processing*, pages 195–204, 2004. 3

[34] H. Zhang and W. W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. on Optimization*, 14:1043–1056, April 2004. 5