EUROGRAPHICS Workshop on ... '0x
N.N. and N.N.
(Editors)

Volume 0 (200x), Number 0

Mesh-free Discrete Laplace–Beltrami Operator

F. Petronetto[1], A. Paiva[2], E. S. Helou[2], D. E. Stewart[3] and L. G. Nonato[2]

[1]Universidade Federal do Espírito Santo, Brazil
[2]Universidade de São Paulo, ICMC, Brazil
[3]University of Iowa, USA

Abstract

In this work we propose a new discretization method for the Laplace–Beltrami operator defined on point-based surfaces. In contrast to the existing point-based discretization techniques, our approach does not rely on any triangle mesh structure, turning out truly mesh-free. Based on a combination of Smoothed Particle Hydrodynamics and an optimization procedure to estimate area elements, our discretization method results in accurate solutions while still being robust when facing abrupt changes in the density of points. Moreover, the proposed scheme results in numerically stable discrete operators. The effectiveness of the proposed technique is brought to bear in many practical applications. In particular, we use the eigenstructure of the discrete operator for filtering and shape segmentation. Point-based surface deformation is another application that can be easily carried out from the proposed discretization method.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computer Graphics/Computational Geometry and Object Modeling—Curve, surface, solid and object representations

1. Introduction

The Laplace–Beltrami (LB) operator is a basic tool for many geometry processing tasks such as mesh editing [SCOL*04], mesh segmentation [LZ07], noise filtering, and remeshing [BNPS10]. Discretizing the Laplace–Beltrami operator is a fundamental step in all of those applications and much effort has been made towards developing robust mechanisms to carry out such a discretization efficiently. In fact, reliable methods exist to discretize the LB operator on triangular surfaces, allowing for estimation of differential quantities [MDSB02] as well as exploiting the LB spectral structure to address many practical problems [LZ10].

In contrast to triangular meshes, few methods have been proposed to discretize the LB operator on point-based surfaces. Although having valuable properties such as convergence, those few methods still bear weaknesses that motivate further investigation. For example, in order to estimate local areas and weights involved in the LB discretization, existing point-based techniques resort to local triangulations constructed in the neighborhood (tangent space) of each sample point [BSW09, LSW09, LPG12]. Besides contradicting the point-based paradigm of avoiding meshes as much as possible, the local triangulations are sensitive to abrupt changes in the density of samples, negatively impacting the accuracy while introducing numerical instabilities.

In this work we present a novel point-based discretization method for the Laplace–Beltrami operator that, differently from existing methods, does not require any mesh construction. The mesh-free nature of the proposed methodology follows from the mathematical foundation it relies on, which combines Smoothed Particle Hydrodynamics (SPH) and an optimization procedure to estimate local area elements. As we show in Section 4, the new discretization scheme turns out to be more accurate than previous point-based methods, more robust when facing changes in the density of points, and, as stated in Section 3, it results in stable discrete operators. We show the effectiveness of the proposed discretization method through a set of comparisons against state-of-the-art discretization techniques. The good performance of the proposed discrete operator is confirmed in a set of practical applications such as point-based surface deformation, filtering and surface area estimation.

In summary, the main contributions of this work are:

- A novel discretization method for the Laplace–Beltrami operator (Section 3) that avoids triangular meshes altogether while still being more accurate and robust than existing point-based techniques.
- An optimization mechanism to estimate surface area elements (Section 3.3). This new mechanism is less sensitive to variations on the density of points, rendering the discretization more stable.

To the best of our knowledge, the SPH theory has never been applied to approximate differential operators on surfaces embedded in $\mathbb{R}^3$. Therefore, this work brings the SPH framework to the context of geometry processing applications.

2. Related Work

In order to better contextualize our approach and highlight its properties we organize the existing methods for discretizing the Laplace–Beltrami operator into two main groups, mesh-based and point-based methods.

Mesh-based The *graph Laplacian* is one of the simplest mechanisms for discretizing the LB operator, resulting in a matrix with off-diagonal entry (i, j) equal -1 if the i^{th} and j^{th} vertices are connected by an edge, 0 otherwise, and diagonal entries set to the degree of the vertex. The lack of geometric information, though, makes the graph Laplacian highly dependent on the underlying mesh [Lev06], which hinders its use in many applications. Mesh dependence can be mitigated if Gaussian weights are used to set off-diagonal entries. Belkin and Niyogi [BN03] showed that, under certain conditions, such a weighting scheme converges to a weighted Laplacian. Lafon [Laf04] generalized Belkin and Niyogi's proof for general isotropic weights. Convergence of discrete graph Laplacian to its continuous counterpart was addressed by Hein *et al.* [HAVL05].

Most mesh-based discretization techniques derive from finite element method using distinct assumptions, resulting in variations of the so called *cotangent scheme* [DMSB99, MDSB02, PP93]. Depending on the area element used, the cotangent scheme may produce non-symmetric matrices with complex eigenvalues and eigenvectors, which impair its use in applications that rely on the eigenstructure of the operator to accomplish geometrical operations. Symmetrization [Lev06, VL08] and generalized eigenvalues [RBG*09] are typical mechanisms employed to tackle the lack of symmetry. Except in special cases, convergence cannot be established for the cotangent scheme [Xu04]. An even more strict result has been proved by Wardetzky *et al.* [WMKG07], which shows that a discrete LB operator satisfying all the properties of its continuous counterpart cannot exist on general meshes. Recently, Hildebrandt and Polthier [HP11] show that consistency can be ensured by combining local functions and the cotangent-based discretization, although the resulting discrete operator is still asymmetric.

Chuang *et al.* [CLB*09] presented a different discretization scheme that also relies on the finite element method. Chuang's discrete LB operator is obtained by restricting 3D basis functions to the surface, resulting in a construction that is invariant to mesh topology while allowing a natural multiresolution structure on the function space. Belkin *et al.* [BSW08] proposed a convergent discretization scheme that makes use of Gaussian kernel defined in the neighborhood of each vertex of the mesh. Although symmetry cannot be ensured, the Belkin's discrete operator does not depend much on the shape of the triangles and it can be extended to surfaces embedded in higher dimensional spaces. Moreover, as shown by Dey *et al.* [DRW10], the spectra of Belkin's operator also converge to the continuous one. Liu *et al.* [LXZ08] presented a method to discretize the LB operator on quadrilateral meshes that is convergent under the particular condition that no vertex has valence other than four. A discretization scheme on general polygonal meshes has been presented by Alexa and Wandetzky [AW11], although a formal proof of convergence for the resulting operator still remains an open issue.

Point Based Methods able to discretize the LB operator directly from point sets are not so abundant, being the scheme proposed by Belkin *et al.* [BSW09] one of the few representatives. This method is indeed a point set version of Belkin's previous work [BSW08], replacing the global mesh structure by a Delaunay triangulation built in the tangent space of each point. While the point based version of Belkin's method exhibits convergence, the resulting matrix is not symmetrizable in general, although a symmetrization scheme has recently been proposed by Liu *et al.* [LPG12]. Luo *et al.* [LSW09] presented a variation of Belkin's discretization scheme that is also convergent and allows for handling the lack of symmetry through generalized eigenvalues. However, both extensions of Belkin's approach rely on Voronoi meshes built on the tangent plane of each point, so, as for Belkin's scheme, they are not true mesh-free methods.

In contrast to the techniques described above, the discretization method proposed in this work does not make use of any auxiliary mesh to accomplish the discretization. Moreover, the resulting matrix can be manipulated so as to carry out spectral decompositions as an eigenvalue problem, thus ensuring a spectrum comprised of real eigenvalues and eigenvectors. We also show that under reasonable conditions the SPH Laplacian operator is numerically stable. Numerical experiments show that our approach converges while still being more accurate and robust to changes in the density of the sample points.

3. Mesh-free Approximation for the Laplace–Beltrami Operator

As already mentioned, the proposed discretization method is based on the Smoothed Particle Hydrodynamics framework. Therefore, before presenting the details of our approach, we provide a brief overview of the SPH discretization method.

3.1. The SPH Approximation Method

The Smoothed Particle Hydrodynamics method is a mesh-free numerical scheme used to discretize differential operators appearing in the mathematical modeling of physical problems. Due to its simplicity and effectiveness, SPH became a very popular discretization method in Computer Graphics, being widely employed in physically-based animation [AW09].

In order to approximate a real-valued or vector-valued function $f \in C^k(\Omega)$ at each point $\mathbf{x}$ of a domain $\Omega \subset \mathbb{R}^d$, the SPH method relies on the following smoothing approximation via convolution

$$f * W_h(\mathbf{x}) = \int_\Omega f(\mathbf{x}')W_h(\|\mathbf{x}-\mathbf{x}'\|)d\mathbf{x}', \qquad (1)$$

where the *kernel* W_h is a symmetric, positive, smooth, compactly supported function whose integral over domain Ω is equal to one. The value h defines the region of influence of W_h. The family of functions $\{W_h\}$ that holds those properties is called an *approximation of identity* [EG92]. Moreover, such a family satisfies $f * W_h(\mathbf{x}) \to f(\mathbf{x})$, when $h \to 0$.

Given a finite set of sample points $\mathcal{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in Ω and the values $f_i = f(\mathbf{x}_i)$ at each point of $\mathcal{P}$, the SPH discretization is accomplished by replacing the continuous integral (1) by its corresponding Riemann sum, resulting in

$$\langle f(\mathbf{x})\rangle = \sum_{j\in N_\mathbf{x}} f_j W_h\left(\|\mathbf{x}-\mathbf{x}_j\|\right) V_j, \qquad (2)$$

where j indexes the points $\mathbf{x}_j \in \mathcal{P}$ lying in the neighborhood $N_\mathbf{x}$ of $\mathbf{x}$ and V_j is the volume element associated to $\mathbf{x}_j$. Differential operators can be discretized in a similar manner. In particular, Equation (1) can be mathematically manipulated to produce the following SPH discretization for the Laplace operator in each point $\mathbf{x}_i \in \mathcal{P}$ (see [PPL*10]):

$$\langle \Delta f_i\rangle = 2\sum_{j\in N_i} f_{ij}\frac{\mathbf{x}_{ij}}{\|\mathbf{x}_{ij}\|^2}\cdot\nabla W_h\left(\|\mathbf{x}_{ij}\|\right)V_j, \qquad (3)$$

where $f_{ij} = f_i - f_j$, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and N_i is the neighborhood of $\mathbf{x}_i$. The discrete form of the Laplace operator (3) involves only first-order derivatives, which makes the discretization simple and computationally efficient.

3.2. SPH Discretization of Laplace–Beltrami Operator

Let $f \in C^2$ be a real-valued function defined on a smooth surface $\mathcal{M}$ embedded in $\mathbb{R}^3$, and let $\hat{f}$ be the *normal extension* of f outside $\mathcal{M}$, that is, $\hat{f}(\mathbf{y}) = f(\mathbf{x})$ for all $\mathbf{y}$ in a narrow band of $\mathcal{M}$ and lying on the line normal to $\mathcal{M}$ passing through $\mathbf{x}$. A well-known result (see Demanet [Dem06]) states that the Laplace–Beltrami operator $\Delta_\mathcal{M}$ applied to a function f defined on $\mathcal{M}$ is the same as the conventional Laplace operator applied to the normal extension $\hat{f}$ of f. Since the derivatives of $\hat{f}$ vanish along the normal ($\hat{f}$ is constant along the normals), we can write

$$\Delta_\mathcal{M} f(\mathbf{x}) = \Delta_{T_\mathbf{x}\mathcal{M}}\hat{f}(\mathbf{x}), \qquad (4)$$

where $T_\mathbf{x}\mathcal{M}$ denotes the tangent plane in $\mathbf{x}$.

In less mathematical terms, Equation (4) states that computing the LB operator of a function f is equivalent to computing the Laplace operator of $\hat{f}$ restricted to the tangent plane $T_\mathbf{x}\mathcal{M}$ in each point $\mathbf{x} \in \mathcal{M}$.

Equation (4) provides a straightforward mechanism to bring the SPH approximation (3) to the context of surfaces. More specifically, let $\mathcal{P}_\mathcal{M}$ be a finite set of sample points on a surface $\mathcal{M}$ and suppose that the values f_i of a function f are known in each point $x_i \in \mathcal{P}_\mathcal{M}$. By combining (3) and (4) we end up with the following SPH discretization for the Laplace–Beltrami operator in a point $\mathbf{x}_i$:

$$\langle \Delta_\mathcal{M} f_i\rangle = 2\sum_{j\in N_i} f_{ij}\frac{\hat{\mathbf{x}}_{ij}}{\|\hat{\mathbf{x}}_{ij}\|^2}\cdot\nabla W_h\left(\|\hat{\mathbf{x}}_{ij}\|\right)V_j, \qquad (5)$$

where $\hat{\mathbf{x}}_{ij} = \hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j$ is the difference between $\mathbf{x}_i$ and the normal projection $\hat{\mathbf{x}}_j$ of $\mathbf{x}_j$ onto the tangent plane $T_{\mathbf{x}_i}\mathcal{M}$. Recalling that W_h is a radial basis function, the chain rule gives:

$$\nabla W_h\left(\|\hat{\mathbf{x}}_{ij}\|\right) = \frac{\hat{\mathbf{x}}_{ij}}{\|\hat{\mathbf{x}}_{ij}\|}w'(q), \quad \text{with } q = \|\hat{\mathbf{x}}_{ij}\|/h\,.$$

In other words, $\nabla W_h\left(\|\hat{\mathbf{x}}_{ij}\|\right)$ lies on the tangent plane $T_{\mathbf{x}_i}\mathcal{M}$ (in this work, w is the quintic spline function described in the Appendix).

We close this subsection with a theoretical result showing that, under reasonable conditions, the SPH approximation of the LB operator results in a numerically stable discrete operator (the proof is provided as supplementary material).

Theorem 1 Suppose that the following assumptions hold:

- **(A1)** The ratios V_i/V_j are bounded, and bounded away from zero;
- **(A2)** Every point $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ is within distance $\widehat{h}$ of some point $\mathbf{x}_j$ in the set $\{\mathbf{x}_i\}_{i=1}^n$;
- **(A3)** The number of points $\mathbf{x}_j$ within distance h of any given point $\mathbf{x} \in \Omega$ is never more than a positive constant M;
- **(A4)** There is a number $r > 0$ (which is specified in the proof) where $h/\widehat{h} \geq r$;
- **(A5)** $W_h(s) = h^{-d}w(s/h)$ for all s, where w is symmetric ($w(-s) = w(+s)$), has compact support $[-1,+1]$, $w'(s) < 0$ for $0 < s < 1$, and $w''(0) < 0$;
- **(A6)** The region Ω has a Lipschitz boundary.

Let $\mathbf{e} = [1, 1, \ldots, 1]^T$ and $\widehat{\mathbf{V}} = [\widehat{V}_1, \widehat{V}_2, \ldots, \widehat{V}_n]^T$, where $\widehat{V}_i = h^{-d}V_i$. Let Δ_h be the matrix resulting from (3). Then the

matrices $\Delta_h + \widehat{\mathbf{V}}\mathbf{e}^T$, have inverses that are bounded independently of h, for sufficiently small $h > 0$. Thus the resulting discrete operator is numerically stable. □

Note that the matrices Δ_h are singular, just as the Laplacian operator is not invertible. However, the kernel of Δ_h is span$\{\mathbf{e}\}$, and the kernel of Δ_h^T is span$\{\widehat{\mathbf{V}}\}$. This is why we show the boundedness of the inverses of $\Delta_h + \widehat{\mathbf{V}}\mathbf{e}^T$.

Assumption (A1) is perhaps the most difficult to ensure, since $\mathbf{V}$ depends on the solution of a linear system. However, negative and zero V_i clearly violate the intuition behind the method. Assumptions (A2–A4) ensure that the point set is neither too concentrated nor too sparse, at least on distances of order h. There is, however, a certain overall density that needs to be achieved to ensure the stability result holds (see (A4)). Assumption (A5) is a mild assumption on W_h; although some implementations of SPH methods use weight functions such as $W(s) = c\,e^{-\alpha s^2}$ which do not have compact support, in practice they decay so rapidly that they are treated as having compact support. Assumption (A6) is a standard one for partial differential equations.

To conclude, we point out that Theorem 1 also holds for the LB operator presented in (5) if the surface $\mathcal{M}$ is C^2. This extra assumption ensures that $\|\mathbf{x}_i - \widehat{\mathbf{x}}_i\|$ is $O(h^2)$, preserving the geometric conditions. As stated, Theorem 1 also ensures stability for SPH approximation of the Laplace operator (3).

3.3. Pointwise Area Elements

The volume elements in the conventional SPH discretization are defined from physical attributes (e.g., mass, and density) associated with the underlying problem. In our context, though, only geometric information is available, thus volume elements (area elements in our case) must be estimated so as to make the discretization process feasible. However, estimating area elements is a problematic issue, mainly in the context of point-based discretization methods. In fact, existing point-based LB discretization techniques resort to local triangulations as basic tool to estimate local areas.

Our approach, in contrast, does not require any triangulation to compute the area elements V_i in each point $\mathbf{x}_i$, which renders it truly mesh-free. The reasoning is to estimate V_i, $i = 1,\ldots n$, by solving an optimization problem derived from the unitary property of W_h given by

$$\int_\Omega W_h\left(\|\mathbf{x} - \mathbf{x}'\|\right) d\mathbf{x}' = 1\,.$$

More specifically, this property leads to a linear system

$$A\mathrm{v} = \mathrm{b}\,, \tag{6}$$

where $a_{ij} = W_h(\|\mathbf{x}_{ij}\|) \geq 0$, $b_i = 1$ and $v_i = V_i$.

The usual Tykhonov-Philips [EHN00] approach amounts to solve the minimization problem:

$$\min \quad F^\rho(\mathrm{v}) := \|A\mathrm{v} - \mathrm{b}\|^2 + \rho\|\mathrm{v}\|^2\,, \tag{7}$$

with $\rho \geq 0$. The term $\rho\|\mathrm{v}\|^2$ is imposed so as to regularize the linear system (6).

Besides smoothing the solution so as to produce evenly distributed area elements, the regularization term also helps to enforce a positive solution for the system. Let v^ρ be the solution of (7). Notice that

$$\rho\|\mathrm{v}^\rho\|^2 \leq F^\rho(\mathrm{v}^\rho) \leq F^\rho(0) = \|\mathrm{b}\|^2\,;$$

therefore

$$\|\mathrm{v}^\rho\| \leq \frac{\|\mathrm{b}\|}{\sqrt{\rho}}\,,$$

i.e., $\lim_{\rho\to\infty} \mathrm{v}^\rho = 0$. Now let us consider the equation $\nabla F^\rho(\mathrm{v}^\rho) = 0$, which is true since v^ρ is an optimizer for F^ρ:

$$A^T(A\mathrm{v}^\rho - \mathrm{b}) = -\rho\mathrm{v}^\rho.$$

From the last two expressions one can see that $A^T\mathrm{b} \approx \rho\mathrm{v}^\rho$ as $\rho \to \infty$. Since A and b are positive, v_i should also be positive for large values of ρ.

In practice, we notice that when the density of points does not change considerably a value for ρ as small as 10^{-3} is enough to ensure positivity. However, evenly distributed area elements is not appropriated when the density of the samples changes substantially throughout the surface. When this is the case, we switch to the following constrained optimization to compute the area elements

$$\begin{cases} \min & F^\rho(\mathrm{v}) \\ \text{s.t.:} & v_i \geq \tau\,, \quad i = 1,\ldots,n \end{cases} \tag{8}$$

where the values $\rho = 10^{-3}$ and $\tau = 2^{-52}$ (MATLAB® machine precision) turned out to be suitable for all experimented data sets.

Figure 1 shows the histograms of area elements in a tori with radius from the center of the hole to the center of the tori tube be 0.8, and the radius of the tube be 0.2. The histogram on the right was generated using the optimization process while the histogram on the left corresponds to the same problem but without the optimization. Notice that area elements vary greatly when the regularization is not imposed, resulting in many negative elements that impair the discretization matrix of being positive semi-definite.

The algorithm used to solve (8) is a simple projected gradient step [Gol64] with an Armijo-like rule [Arm66] for the step size (see convergence analysis in the Appendix). In Algorithm 1 we denote $\mathrm{v}^k(\mu) := P\big(\mathrm{v}^k - \mu\nabla F^\rho(\mathrm{v}^k)\big)$, with P the projection operator over the feasible set. The parameters ε: the stopping criterion, is set as 10^{-3}; υ: the relative tolerance of the Armijo search is chosen as 10^{-4}; β: the factor used to reduce the step size is 10^{-1}; $\omega = 10^{-10}$ is a technical safeguard to ensure convergence. The initial approximation v^0 is given by the direct solution of the minimization problem (7).

Figure 2 shows comparisons between the surface area approximated using Algorithm 1 (simply called by COAE al-

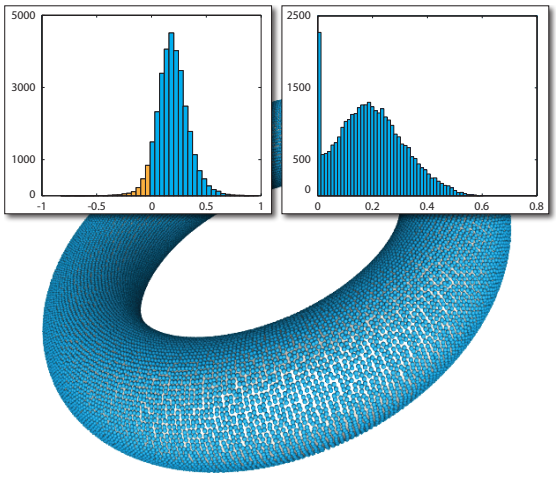


Figure 1: *Histogram of area elements ($\times 10^{-3}$) with (right) and without (left) the optimization process in a ring tori surface with 32k sampling points.*

	model				
	# points	8k	12k	15k	20k
area	triangles	12.56	3.98	5.56	5.52
area	COAE	12.57	4.03	5.50	5.37
area	error	0.08%	1.26%	1.08%	2.72%
	# iter.	✗	6k	23k	63k
	time	✗	31 sec	151 sec	505 sec

Figure 2: *Point-based surface area estimation: the values below each model correspond to the area estimated from the triangle mesh and using the COAE algorithm. The number of iterations performed by the algorithm and corresponding computational times are also shown.*

Algorithm 1 Constrained Optimization for Area Elements.

Require: v^0 // Initial approximation
  $k = 0$
  // Main loop: ε = tolerance
  repeat
    // Maximum feasible step size μ
    $\Lambda = \{i : v_i^k > \tau \text{ and } [\nabla F^\rho(\mathrm{v}^k)]_i > 0\}$
    if $\Lambda = \emptyset$ **then**
      $\mu = 1$;
    else
      $\mu = \max\left\{\omega, \min_{i\in I}\left\{\left(v_i^k - \tau\right)/[\nabla F^\rho(\mathrm{v}^k)]_i\right\}\right\}$
    end if
    // Armijo search
    while $F^\rho(\mathrm{v}^k) - F^\rho\left(\mathrm{v}^k(\mu)\right) \leq \frac{\upsilon}{\mu}\|\mathrm{v}^k - \mathrm{v}^k(\mu)\|^2$ **do**
      $\mu = \mu\beta$
    end while
    // Update
    $\mathrm{v}^{k+1} = \mathrm{v}^k(\mu)$
    $k = k+1$
  until $\frac{1}{\mu}\|\mathrm{v}^k - \mathrm{v}^{k-1}\|_\infty > \varepsilon$

gorithm) and computed directly from the corresponding triangle mesh. Notice that in the uniform unit sphere, the initial approximation v^0 contains only positive area elements, then the COAE algorithm returns v^0 without any iteration. All results were generated on a 3.2 GHz Intel® Core i7 with 16 GB of RAM.

It is worth mentioning that the approach described above for local area element estimation can be employed in SPH-based discretization schemes other than the one presented in this paper.

3.4. Normal Vectors and Symmetrization

Normal Vectors If normal vectors are not available, they must be estimated in order to define the tangent planes [MN03]. Principal Component Analysis (PCA) is the typical tool employed to compute normal vectors. However, it is well known that PCA is sensitive to noise and outliers. In order to get around this drawback we adopt a more robust approach called *Weighted Principal Component Analysis* (WPCA) [KC03]. In each point $\mathbf{x}_i$ the WPCA computes a weighted covariance matrix E_i as

$$E_i = \sum_{j\in N_i} \theta_{ij}(\mathbf{x}_j - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_i)^T, \qquad (9)$$

where $\bar{\mathbf{x}}_i = \sum_{j\in N_i} \theta_{ij}\mathbf{x}_j$ and $\theta_{ij} = W_h(\|\mathbf{x}_{ij}\|)/\sum_{j\in N_i} W_h(\|\mathbf{x}_{ij}\|)$. The normal vector $\mathbf{n}_i$ in $\mathbf{x}_i$ is given by the cross product $\mathbf{n}_i = \mathbf{v}_i^1 \times \mathbf{v}_i^2/\|\mathbf{v}_i^1 \times \mathbf{v}_i^2\|$, where $\mathbf{v}_i^1$ and $\mathbf{v}_i^2$ are the eigenvectors associated to the two largest eigenvalues of E_i.

Symmetrization The LB operator $\Delta_\mathcal{M}$ is symmetric (self-adjoint) on a compact Riemannian manifold $\mathcal{M}$ [Hel10], that is:

$$\langle f, \Delta_\mathcal{M} g\rangle_{L^2(\mathcal{M})} = \langle \Delta_\mathcal{M} f, g\rangle_{L^2(\mathcal{M})}.$$

In the discrete case, one expects that a LB operator L is represented by a *symmetrizable matrix*, i.e., the matrix L can be factored as the product of a symmetric matrix and a symmetric positive definite (SPD) matrix [LPG12]. Such a property is important because symmetrizable matrix has real eigenvalues and their eigenvectors are orthogonal w.r.t. the inner product defined by $\langle \mathbf{u}, \mathbf{v}\rangle_M = \mathbf{u}\cdot M\mathbf{v}$, where M is a SPD matrix [Hus78]. In order to build a symmetrizable matrix we evaluate $\hat{\mathbf{x}}_{ij}$ and $\hat{\mathbf{x}}_{ji}$ using the same plane orthogonal to $(\mathbf{n}_i + \mathbf{n}_j)/\|(\mathbf{n}_i + \mathbf{n}_j)\|$, considering the average normal to project both $\mathbf{x}_i$ and $\mathbf{x}_j$ onto the plane. Therefore,

distinct planes are used to evaluate each term of the summation in (5). Since average normals do not deviate too much from the tangent planes, the error introduced by the use of multiple planes in each point $\mathbf{x}_i$ is not large (see Section 4). Another positive aspect of the multiple planes scheme is that normal vectors must be consistently oriented only when computing the average normal vectors, which can be ensured with a simple dot product.

Assuming the multiple planes scheme above, the SPH-based LB operator discretization produces a symmetrizable matrix of order $n \times n$ (n is the number of points) given by the product $L = GD$, where G is a symmetric matrix with

$$G_{ij} = \begin{cases} -2\dfrac{\hat{\mathbf{x}}_{ij}}{\|\hat{\mathbf{x}}_{ij}\|^2} \cdot \nabla W_h\left(\|\hat{\mathbf{x}}_{ij}\|\right), & \text{if} \quad i \neq j \\ -\dfrac{1}{V_i}\displaystyle\sum_{k=1}^{n} G_{ik} V_k, & \text{otherwise} \end{cases} \tag{10}$$

and D is a diagonal matrix with entries $D_{ii} = V_i > 0$, thus SPD. Moreover, it is easy to see that the nonsymmetric matrix L is self-adjoint w.r.t. the inner product $\langle \mathbf{u}, \mathbf{v} \rangle_D$.

4. Results and Comparisons

We start this section showing how the estimation of normal vectors affects the accuracy of the proposed LB discretization method. The table in Figure 3 shows ℓ^2 and ℓ^∞ errors (first and second numbers in each entry of the table, respectively) of our LB operator when applied to the functions depicted in the first column. Notice that, in a uniformly sampled unit sphere, the WPCA-based normal estimation resulted in approximations as accurate as the ones obtained with the true normals (computed analytically). The WPCA scheme also presents excellent results when facing non-uniform sample distribution and abrupt variation in the density of samples, clearly overcoming the PCA-based approach. These results have motivated us to use WPCA as the normal estimation mechanism. In particular, the non-uniform unit sphere was generated using the marching cubes algorithm; the density varying sphere was created by applying the quadric edge simplification algorithm [GH97] on an uniformly sampled unit sphere; and all uniformly sampled unit spheres were generated using [HSS00].

In order to confirm the quality of the proposed discretization method we provide comparisons against two other techniques, namely, the well-known cotangent scheme [PP93] (with weights given by $1/3$ of the area of the triangles incident to each vertex [BSW08]), denoted by COT, and the discrete point-cloud operator proposed by Luo *et al.* [LSW09], which we denote by PCL. The three methods are assessed in terms of approximation error and robustness. The unit sphere is used as ground truth in the comparisons, as analytical solutions can be explicitly computed on the sphere.

	PCA	WPCA	analytic
	uniform		
f(x,y,z) = x	14.8 / 0.66	2.93 / 0.22	2.92 / 0.22
f(x,y,z) = x^2	23.7 / 1.14	24.5 / 1.03	24.5 / 1.03
f(x,y,z) = e^x	21.5 / 1.39	19.4 / 1.39	19.4 / 1.39
	non-uniform		
f(x,y,z) = x	52.1 / 2.40	33.2 / 1.75	13.4 / 1.11
f(x,y,z) = x^2	51.3 / 2.42	53.6 / 2.26	54.0 / 2.26
f(x,y,z) = e^x	58.2 / 3.62	48.8 / 2.99	43.3 / 3.00
	density varying		
f(x,y,z) = x	31.4 / 1.45	11.2 / 0.83	11.1 / 0.78
f(x,y,z) = x^2	40.3 / 2.22	39.9 / 1.65	39.9 / 1.67
f(x,y,z) = e^x	38.4 / 2.67	32.2 / 2.19	32.0 / 2.25

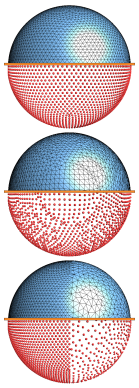


Figure 3: *Normal estimation error using PCA and WPCA. Tables show error ($\times 10^{-3}$) in ℓ^2 / ℓ^∞ norms (left and right values in each entry of the table, respectively), for unit spheres with 8k samples. Top table contains errors obtained in an uniformly sampled sphere while errors for non-uniform and density varying sampled spheres are presented in the last two tables.*

Figure 4 shows asymptotic behavior (consistency analysis) in semi-log scale for the three techniques when discretizing the LB operator in uniformly sampled spheres with increasing density. Notice that the error resulting from the SPH discretization decreases faster than the one obtained with COT and PCL. The top right plots in Figure 4 show the errors in semi-log scale for the ℓ^∞ norm. The semi-log scale makes the lack of convergence of the cotangent scheme more evident. It is worth mentioning that, for the function $f(x,y,z) = x$, the proposed scheme is the only one that results in decreasing error in ℓ^∞ norm.

The robustness of the SPH approach when facing non-uniform samples and abrupt changes in the density of samples can be observed in Figure 5. Notice that, the error produced by Luo's method and cotangent scheme (second and third rows in Figure 5) are, in most cases, at least one order of magnitude higher than the error produced by the proposed SPH-based discretization scheme.

The spectrum of the SPH-based LB operator can be obtained by solving the (nonsymmetric) eigenvalue problem:

$$-L\phi = \lambda\phi, \tag{11}$$

where the eigenvalues λ_i are non-negative real numbers (we are assuming $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$) with pairwise orthogonal eigenvectors $\{\phi_1, \phi_2, \ldots, \phi_n\}$. However, Krylov based methods become costly, slowly converging, and numerically unstable when dealing with large nonsymmetric eigenvalue problems. This drawback can be circumvented by converting the problem in a symmetric eigenvalue problem as follows:

$$-C\mathbf{y} = \lambda\mathbf{y}, \tag{12}$$

with $C = D^{\frac{1}{2}} G D^{\frac{1}{2}}$ and $\mathbf{y} = D^{\frac{1}{2}}\phi$. The orthogonality be-

tween normalized eigenvectors $\mathbf{y}_i$ ensures that the eigenvectors ϕ_i (11) are pairwise D-orthonormal as follows:

$$\langle \phi_i, \phi_j \rangle_D = \phi_i \cdot D\phi_j = D^{\frac{1}{2}}\phi_i \cdot D^{\frac{1}{2}}\phi_j = \mathbf{y}_i \cdot \mathbf{y}_j = \delta_{ij},$$

where δ_{ij} is the Kronecker's delta.

Figure 6 shows that the SPH-based technique also performs well in terms of eigenvalue approximation. The eigenvalues of the SPH-based LB operator tend to the analytical eigenvalues when the number of sample points increases. Moreover, the multiplicity of the eigenvalues matches closely the multiplicity of their analytical counterpart. Moreover, the SPH-based approach turns out to be more robust and accurate when dealing with non-uniform and density varying sample distribution, as depicted in Figure 7.

Convergence rates for the three methods are shown in Figure 8. More specifically, the plots show the percent error obtained by approximating the solution of the Poisson equation $-\Delta_{\mathcal{M}} u = f$ with the same constraints for the three discrete operators. A set of uniformly sampled torus with increasing density of points were used in this comparison. As one can see, the SPH-based LB operator presents a good performance in terms of convergence, outperforming COT and PCL.

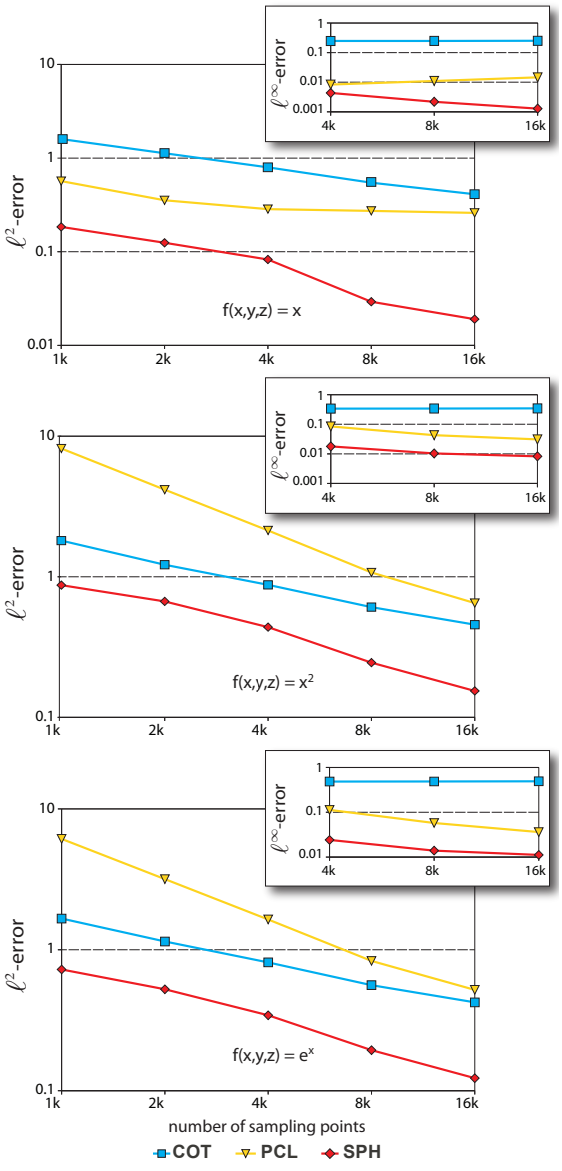


Figure 4: *Consistency analysis of $\Delta_{\mathcal{M}} f$ on uniformly sampled unit sphere. The curves allow to compare the SPH-based method against the cotangent (COT) and Luo's (PCL) schemes, using percent error in ℓ^2 (larger graphs) and ℓ^∞ (top right) norms.*

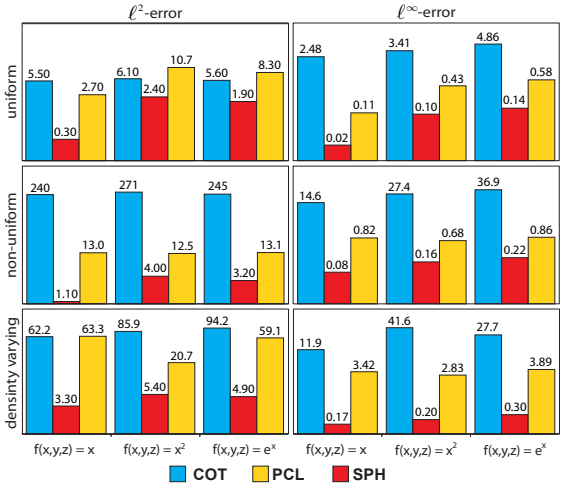


Figure 5: *Approximation error of $\Delta_{\mathcal{M}} f$ on uniform, non-uniform and density varying unit spheres sampled with 8k points using percent error ($\times 10^{-1}$) in ℓ^2 (left) and ℓ^∞ (right).*

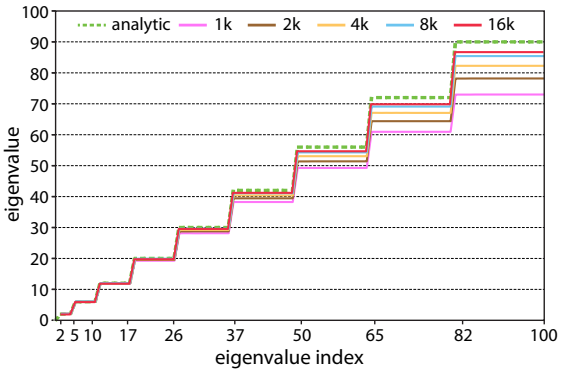


Figure 6: *Convergence of the eigenvalues of the SPH-based LB operator on unit sphere.*

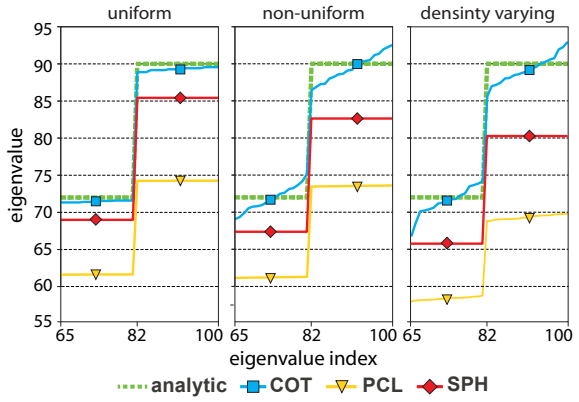


Figure 7: *Comparisons of the eigenvalues of the LB operator on a 8k point sampled unit sphere with uniform (left), non-uniform (center) and density varying (right) sampling. The eigenvalues of the SPH-based LB operator are closer to their analytical counterpart than the ones computed from symmetric COT (proposed by Levy [Lev06]) and PCL, even for non-uniform and density varying samples.*

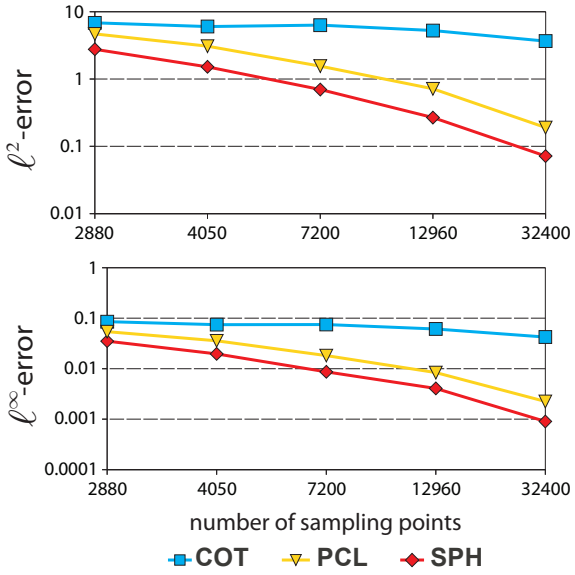


Figure 8: *Convergence analysis for the Poisson equation $-\Delta_{\mathcal{M}} u = f$ on uniformly sampled torus. The semi-log graph shows the percent error in ℓ^2 (top) and ℓ^∞ (bottom) norms for the function $f(x,y,z) = -x/(x^2+y^2)^{1/2}$ approximated by the SPH-based LB operator, the symmetric COT and PCL.*

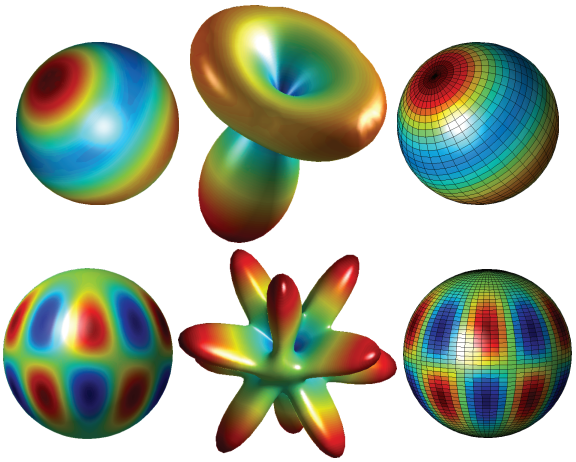


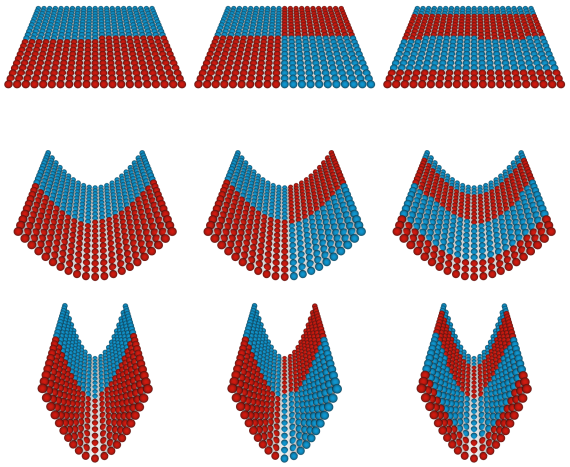
Figure 9: *Two spherical harmonics Y_3^0 (top) and Y_5^4 (bottom) computed using the proposed discrete LB operator (left) and their analytical counterpart (right) on uniformly sampled unit sphere with 8k points. Deformations of the unit sphere (middle) according to the eigenvectors obtained with our discrete operator.*

Figure 9 shows spherical harmonics Y_m^l of degree m and order l computed using the SPH discrete operator. Analytical solutions are shown on the right. Notice that the harmonics computed from the SPH-based LB operator closely resemble the analytical ones. The two shapes in the middle of Figure 9 have been generated by displacing the samples from the surface towards the center of the unit sphere according to the values of the harmonics (points with negative values were pushed to the center). The resulting deformation provides a better visualization of analyze the vibration modes.

Figure 10 shows that the eigenvectors of the SPH-based LB operator are invariant under isometries, that is, the eigenvectors depend only on Riemannian distances on the surface. Therefore, nodal domains are well preserved under isometric transformations. The planar surface in the first row in Figure 10 is an isometric mapping of the parabolic cylinder $\mathbf{r}(u,v) = (u, au^2, v)$ ($a = 0.5$ middle and $a = 1$ bottom) onto the plane. Notice that nodal regions are fairly preserved after the isometric mapping.

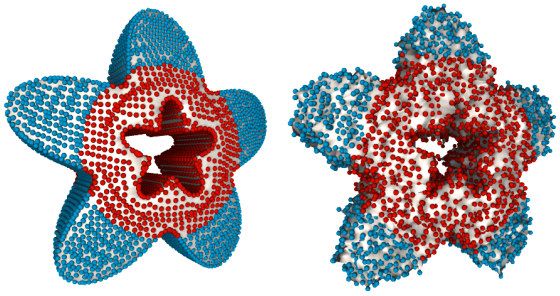
Figure 11 illustrates the stability of the spectrum of the proposed operator when facing noisy data. Figure 11(a) shows the segmentation induced by nodal domains of the eigenvector ϕ_6 on a *trimmed star* model, which contains sharp features. The nodal domains segment the model as expected, capturing symmetries. From Figure 11(b) one can notice that the nodal domains are well preserved when 3% of pseudorandom noise is added to the normal and tangential components of the sample points, showing the stability of the spectrum of the SPH-based LB operator.

Regarding computational times, the SPH technique is similar to the Luo's method if constrained optimization is not needed. If the regularization results in negative area elements, the constrained optimization must be invoked, which may increase computational times (our code was implemented in MATLAB®).



(a) (b) (c)

Figure 10: *Nodal domains are invariant under isometric transformations: eigenvectors (a) ϕ_2 (b) ϕ_4 and (c) ϕ_{10} on a plane (top) and two parabolic cylinders with low (middle) and high (bottom) mean curvature sampled with 441 points.*



(a) (b)

Figure 11: *Spectral segmentation on a trimmed star model with 5k sampling points: (a) original and (b) noisy models.*

5. Applications

The proposed discretization approach can be employed in many geometry processing applications, some of which illustrated in this section.

As a first application we show the use of the SPH-based LB operator in point-based surface filtering. Given the eigenvalues λ_i and the eigenvectors ϕ_i making up the eigenspace of the SPH discrete operator, a smooth function $f : \mathcal{M} \to \mathbb{R}$ can be decomposed as $f = \sum_i \langle f, \phi_i \rangle_D \phi_i$. The coefficients $\langle f, \phi_i \rangle_D$ and the basis functions ϕ_i (also called *manifold harmonics*) behaves like a Fourier spectral decomposition, allowing for the design of filters to process the function f on $\mathcal{M}$ [VL08]. Figure 12 shows the result of applying a low-pass filter to the coordinates (seen as functions) of the sampling points the Armadillo model.

Surface deformation is another application that benefits from our discrete operator. Following the methodology of differential coordinates [Sor06], point-based surfaces can be deformed by solving the linear system $L\mathrm{x} = \mathrm{b}$, where L is the discrete LB operator, $\mathrm{b} = (2I - L)\mathrm{x}_o$, x_o is the original coordinates of the vertices, and I is the identity matrix. Constraining the system with the coordinates of points on the surface as well as points moved away to steer the deformation one obtains results as illustrated in Figure 13.

Figure 14 illustrates the behaviour of the proposed operator when solving the classical Plateau's problem, which consists in fixing two curves Γ (circles with equal radius in our case) and then find the surface of minimal area whose boundary is Γ. We use the method suggested by Pinkall and Polthier in [PP93], where the minimal surface M is approximated by a sequence of surfaces that converges to M. More specifically, the algorithm works as follows: (1) given the fixed polygonal boundary curves Γ, determine a surface $M^{(0)}$ such that $\partial M^{(0)} = \Gamma$ (∂M accounts for the boundary of M); (2) let $M^{(k)}$ be a surface satisfying $\partial M^{(k)} = \Gamma$. Compute the surface $M^{(k+1)}$ such that $\partial M^{(k+1)} = \Gamma$ and the points on $M^{(k+1)}$ satisfy the homogeneous system $L^{(k)}x = 0$, where $L^{(k)}$ is the Laplacian matrix given by the SPH-based LB operator applied to $M^{(k)}$. (3) If $|area(M^{(k)}) - area(M^{(k+1)})| < \varepsilon$ then $M^{(k+1)}$ is the desired approximation of the minimal surface, otherwise, return to the step (2) using $M^{(k+1)}$. In our example, $M^{(0)}$ is given by a cylinder with radius and height equal to 2 and sampled with 7840 points (top-left in Figure 14). The parameter $\varepsilon = 10^{-4}$ has been used as stop criterion. Notice that the area (computed from Algorithm 1) of the approximate surface converges to the area of the analytical catenoid.

6. Discussion and Limitation

The comparisons presented in Section 4 clearly show the effectiveness of the proposed SPH discretization method, surpassing state-of-art methods in accuracy and robustness. The superior performance of our scheme is a consequence of combining the SPH discretization method with the new optimization mechanism to estimate area elements. Simplicity is another strength of our approach, since it does not require the construction of any local mesh. The optimization scheme for volume elements estimation is more flexible and general than the local construction of Voronoi/Delaunay diagrams employed by previous approaches, mainly if a generalization to high-dimensional point clouds data is of interest.

The parameter h used by the kernel W_h may affect the accuracy and the computational cost of our method. The results

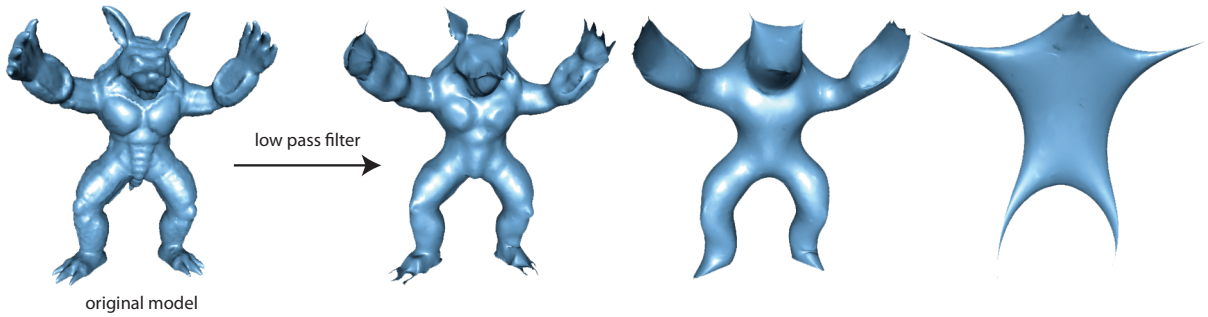
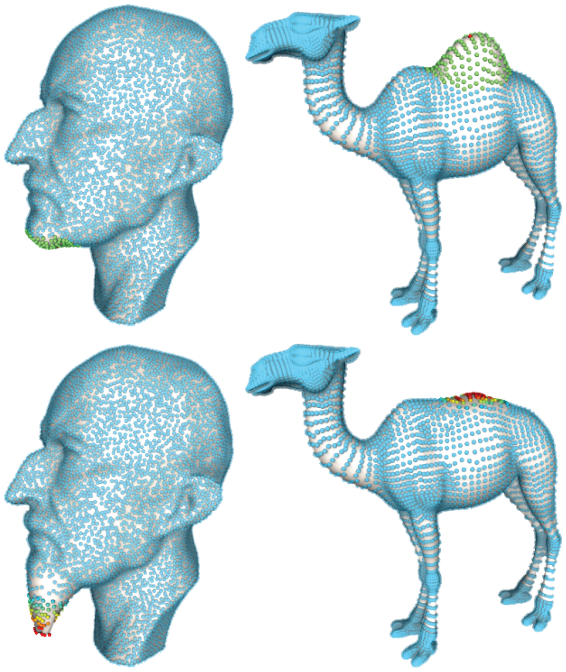


Figure 12: *Low-pass filtering of the Armadillo model sampled with 22k points. From left to right: original model, smoothed reconstruction with* 1000*,* 100 *and* 10 *eigenvectors.*



(a) (b)

Figure 13: *Deformations carried out taking as basis the proposed discrete LB operator: (a) outward deformation in Max Planck model with 12k sampling points and (b) inward deformation in camel model with 10k sampling points. The colormaps represent the point displacement constraints (top) and the magnitude of the deformation (bottom).*

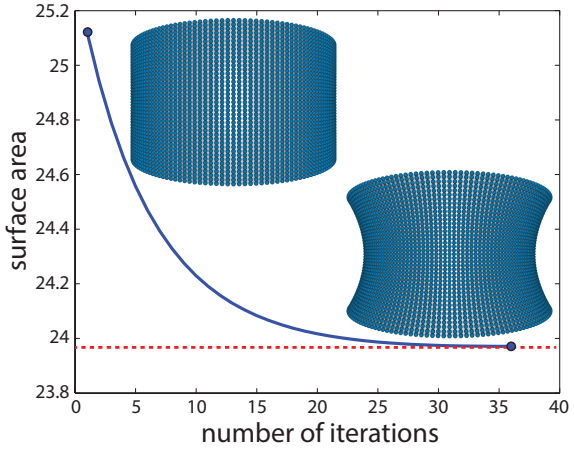


Figure 14: *Approximating a catenoid minimal surface (bottom-right) starting from a cylinder (top-left). The area of the discrete surface (blue) converges to the area of the catenoid (red).*

shown in Section 4 have been obtained with h as follows

$$h = \frac{1}{n}\sum_{i=1}^{n} \max_{j \in N_i}\{\|\mathbf{x}_i - \mathbf{x}_j\|\},$$

where N_i is made up of the 100 nearest neighbors of $\mathbf{x}_i$. On average, this value of h provides just one-third of the number of samples used by PCL algorithm to build the neighborhood of each sample. Since the spectrum is less sensitive to h, good results can be reached by using $\#N_i = 50$.

Moreover, we can define constant area elements by taking the average of the solution of the linear system (6) without performing the COAE algorithm. Although accuracy is affected when area elements are evenly distributed among the sample points, the overall behavior of the operator is preserved, mainly when the density of samples does not change considerably on the surface. The middle image in Figure 15 shows the solution of $\Delta_{\mathcal{M}} f = 0$, with constraints $f = 1$ and $f = 0$ imposed at the bottom part of the arms of the Bimba model.

Notice that the solution behaves exactly as expected, with values varying smoothly from one side to the other of the model. The same is true for eigenvectors, as depicted in the

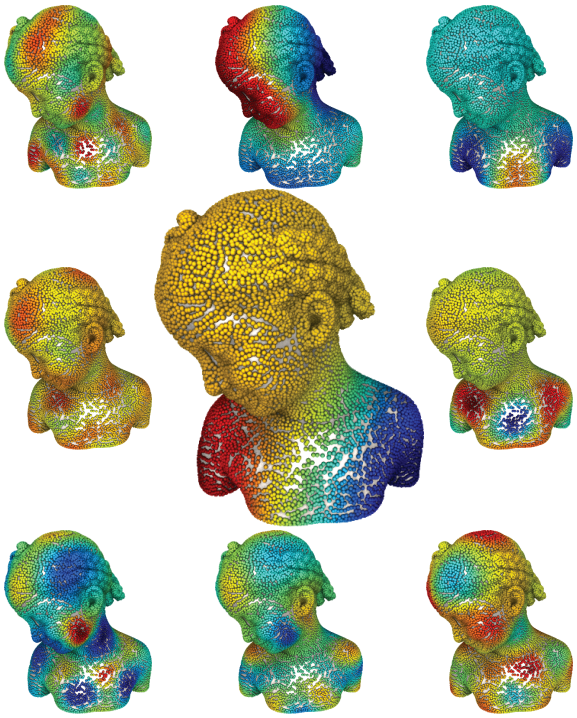


Figure 15: *The discrete operator behaves as expected when local areas are set constant in the Bimba model sampled with 16k points. Middle is the solution of $\Delta_{\mathcal{M}} f = 0$ and the surrounding models are some eigenvectors.*

surrounding images of Figure 15. For geometry processing and graphics applications that do not require very accurate results, such uniform values for area elements turns out to be a good option.

One of the main issues with point set surfaces is to determine the neighborhood relation of points on surfaces with close sheets. The image on the top of Figure 16 shows the nodal set of the 2nd eigenvector of the SPH-based LB operator on a non-convex surface when the neighborhood of points are not properly defined. If surface normals are consistently oriented one can cluster them so as to better define neighborhoods, as proposed by Kobbelt *et al.* [KBSS01]. The botton image in Figure 16 shows the nodal sets of the eigenvector ϕ_2 of the SPH-based LB operator when the neighborhood of each point is computed using normal clusters. Notice that the model is split correctly when neighborhoods are computed properly.

As shown in Figure 11, the nodal sets of the SPH-based LB operator is stable under noise perturbations. However, as depicted in Figure 17, the accuracy of the proposed SPH-based LB operator is affected when noise is added (5% of noise is added to the normal and tangential components of the points), since the COAE algorithm does not employ any

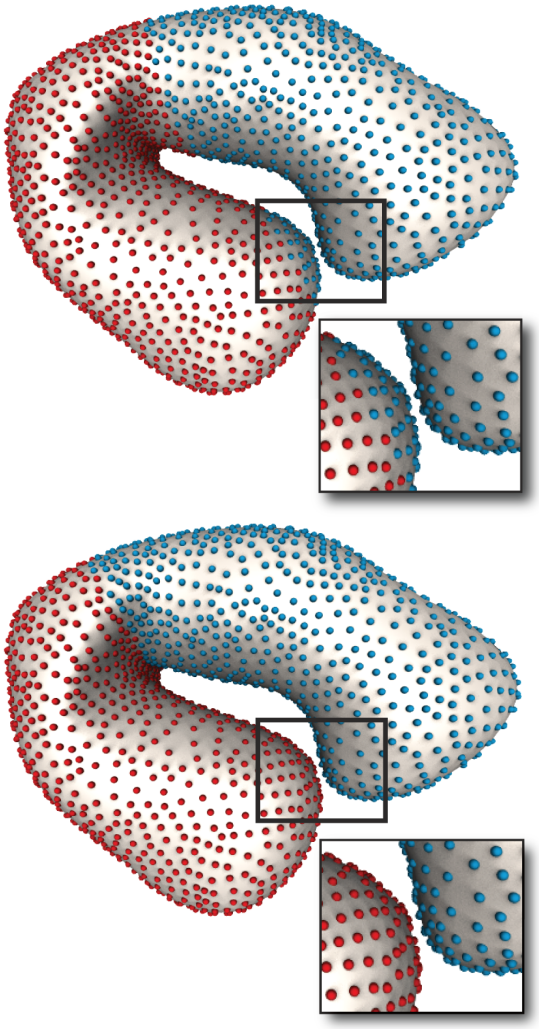


Figure 16: *Nodal domains generated by the eigenvector ϕ_2 on sausage model with 3k sampling points: without (top) and with (bottom) clustering of normals.*

point cloud denoising process. In order to improve accuracy when computing the spectrum, we can perform a denoising process similar to the strategy used by the PCL operator. The process consists in projecting the neighborhood of each point onto its corresponding tangent plane computed via WPCA. As one can observe from the Figure 17, the COAE algorithm combined with such a denoising strategy results in a quite accurate spectrum computation, outperforming COT and PCL techniques.

7. Conclusion

In this work we proposed a novel SPH-based discretization method for the Laplace–Beltrami operator which does not

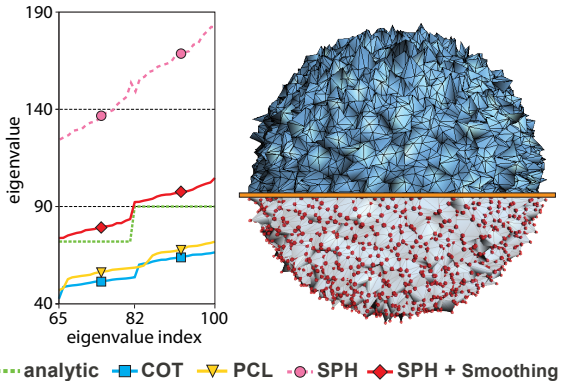


Figure 17: *The accuracy of the SPH-based LB operator when approximating the corresponding spectral space on noisy unit sphere sampled with 8k points. SPH-based LB operator outperforms the symmetric COT and PCL when a simple denoising step is added to the COAE algorithm.*

make use of any triangular mesh in its formulation. The evaluation we provided shows that our approach outperforms existing techniques in terms of accuracy as well as robustness. The new mechanism for estimating area elements turned out very effective, ensuring accurate results. Moreover, the new optimization procedure to estimate area elements can also be employed to improve SPH-discretization methods in a more general context, benefiting other applications such as fluid flow simulation. In summary, accuracy and robustness render the proposed discretization method one of the most attractive for discretizing the Laplace–Beltrami operator on point-based surfaces. To finish, we pointed out that while our new LB operator is empirically more accurate, it does not provide any convergence guarantee, being this the main point for future work.

Acknowledgements

The authors are grateful to Josï£¡ Miguel Malacarne for helpful discussions. The authors also wish to thank the reviewers for their careful reading and comments about earlier drafts of this paper. This work was partially supported by grants from CNPq, FAPES and FAPESP.

References

[Arm66] ARMIJO L.: Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics 16* (1966), 1–3. 4

[AW09] ADAMS B., WICKE M.: Meshless approximation methods and applications in physics based modeling and animation. In *Eurographics Tutorials* (2009), pp. 213–239. 3

[AW11] ALEXA M., WARDETZKY M.: Discrete Laplacians on general polygonal meshes. *ACM T. Graph. 30* (2011), 102:1–102:10. 2

[BN03] BELKIN M., NIYOGI P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput. 15*, 6 (2003), 1373–1396. 2

[BNPS10] BERGER M., NONATO L. G., PASCUCCI V., SILVA C. T.: Fiedler trees for multiscale surface analysis. *Comput. Graph. 34* (2010), 272–281. 1

[BSW08] BELKIN M., SUN J., WANG Y.: Discrete Laplace operator on meshed surfaces. In *Sympos. on Comput. Geom.* (2008), pp. 278–287. 2, 6

[BSW09] BELKIN M., SUN J., WANG Y.: Constructing Laplace operator from point clouds in R^d. In *Sympos. on Discr. Algorith.* (2009), pp. 1031–1040. 1, 2

[CLB*09] CHUANG M., LUO L., BROWN B. J., RUSINKIEWICZ S., KAZHDAN M.: Estimating the Laplace-Beltrami operator by restricting 3D functions. In *Sympos. on Geom. Proc* (2009), pp. 1475–1484. 2

[Dem06] DEMANET L.: *Painless, highly accurate discretizations of the Laplacian on a smooth manifold*. Tech. rep., Stanford University, 2006. 3

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH* (1999), pp. 317–324. 2

[DRW10] DEY T. K., RANJAN P., WANG Y.: Convergence, stability, and discrete approximation of Laplace spectra. In *Sympos. on Discr. Algorith.* (2010), pp. 650–663. 2

[EG92] EVANS L., GARIEPY R.: *Measure theory and fine properties of functions*. CRC, 1992. 3

[EHN00] ENGL H. W., HANKE M., NEUBAUER A.: *Regularization of Inverse Problems*. Kluwer Academic Publishers, 2000. 4

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *SIGGRAPH '97* (1997), pp. 209–216. 6

[Gol64] GOLDSTEIN A.: Convex programming in Hilbert spaces. *B. Am. Math. Soc. 5*, 70 (1964), 709–710. 4

[HAVL05] HEIN M., AUDIBERT J., VON LUXBURG U.: From graphs to manifolds–weak and strong pointwise consistency of graph laplacians. *Learning theory* (2005), 470–485. 2

[Hel10] HELFFER B.: *Spectral theory and applications: an elementary introductory course*. Tech. rep., Université Paris-Sud, 2010. 5

[HP11] HILDEBRANDT K., POLTHIER K.: On approximation of the Laplace-Beltrami operator and the Willmore energy of surfaces. *Comput. Graph. Forum 30*, 5 (2011), 1513–1520. 2

[HSS00] HARDIN R. H., SLOANE N. J. A., SMITH W. D.: *Tables of Spherical Codes with Icosahedral Symmetry*. Tech. rep., AT&T Lab, 2000. 6

[Hus78] HUSEYIN K.: *Vibrations and stability of multiple parameter systems*, vol. 6. Springer, 1978. 5

[KBSS01] KOBBELT L. P., BOTSCH M., SCHWANECKE U., SEIDEL H.-P.: Feature sensitive surface extraction from volume data. In *SIGGRAPH '01* (2001), pp. 57–66. 11

[KC03] KOREN Y., CARMEL L.: Visualization of labeled data using linear transformations. In *IEEE InfoVis* (2003), pp. 121–128. 5

[Laf04] LAFON S.: *Diffusion maps and geometric harmonics*. PhD thesis, Yale University, 2004. 2

[Lev06] LEVY B.: Laplace-Beltrami eigenfunctions towards an algorithm that "understands" geometry. In *IEEE Intern. Conf. on Shape Model. and Applic.* (2006), pp. 13–13. 2, 8

[LPG12] LIU Y., PRABHAKARAN B., GUO X.: Point-based manifold harmonics. *IEEE T. Vis. Comput. Gr. 18*, 10 (2012), 1693–1703. 1, 2, 5

[LSW09] LUO C., SAFA I., WANG Y.: Approximating gradients for meshes and point clouds via diffusion metric. In *Sympos. on Geom. Proc* (2009), pp. 1497–1508. 1, 2, 6

[LXZ08] LIU D., XU G., ZHANG Q.: A discrete scheme of Laplace-Beltrami operator and its convergence over quadrilateral meshes. *Comput. Math. Appl. 55* (2008), 1081–1093. 2

[LZ07] LIU R., ZHANG H.: Mesh segmentation via spectral embedding and contour analysis. *Comput. Graph. Forum 26* (2007), 385–394. 1

[LZ10] LÉVY B., ZHANG H. R.: Spectral mesh processing. In *SIGGRAPH Courses* (2010), pp. 8:1–8:312. 1

[MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visual Math.* (2002), pp. 1–26. 1, 2

[MN03] MITRA N., NGUYEN A.: Estimating surface normals in noisy point cloud data. In *Sympos. on Comput. Geom.* (2003), ACM, pp. 322–328. 5

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2* (1993), 15–36. 2, 6, 9

[PPL*10] PETRONETTO F., PAIVA A., LAGE M., TAVARES G., LOPES H., LEWINER T.: Meshless Helmholtz-Hodge decomposition. *IEEE T. Vis. Comput. Gr. 16*, 2 (2010), 338–342. 3

[RBG*09] REUTER M., BIASOTTI S., GIORGI D., PATANÈ G., SPAGNUOLO M.: Discrete Laplace-Beltrami operators for shape analysis and segmentation. *Comp. Graph. 33* (2009), 381–390. 2

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., H.-P.SEIDEL: Laplacian surface editing. In *Sympos. on Geom. Proc* (2004), pp. 175–184. 1

[Sor06] SORKINE O.: Differential representations for mesh processing. *Comput. Graph. Forum 25* (2006), 789–807. 9

[VL08] VALLET B., LÉVY B.: Spectral geometry processing with manifold harmonics. *Comput. Graph. Forum 22*, 2 (2008), 251–260. 2, 9

[WMKG07] WARDETZKY M., MATHUR S., KÄLBERER F., GRINSPUN E.: Discrete Laplace operators: No free lunch. In *Sympos. on Geom. Proc.* (2007). 2

[Xu04] XU G.: Discrete Laplace-Beltrami operators and their convergence. *Comput. Aided Geom. D. 21*, 8 (2004), 767–784. 2

Appendix

Kernel function W_h

$$W_h\left(\|\mathbf{x}-\mathbf{x}_j\|\right) = \frac{7}{478\pi h^2}\, w\left(\frac{\|\mathbf{x}-\mathbf{x}_j\|}{h}\right) \qquad \text{with}$$

$$w(q) = \begin{cases} (3-q)^5 - 6(2-q)^5 + 15(1-q)^5\,, & 0 \le q < 1 \\ (3-q)^5 - 6(2-q)^5\,, & 1 \le q < 2 \\ (3-q)^5\,, & 2 \le q < 3 \\ 0\,, & q \ge 3 \end{cases}$$

Convergence of the optimization algorithm

We first show that the Armijo criterion is actually attained in a finite number of steps of the search. Let us, for simplicity, write $f = F^{\rho}$ and $\mathrm{v}(\mu) = P\big(\mathrm{v} - \mu\nabla f(\mathrm{v})\big)$. We now consider the vector $\delta\mathrm{v}(\mu) := \frac{1}{\mu}\big(\mathrm{v} - \mathrm{v}(\mu)\big)$:

$$[\delta\mathrm{v}(\mu)]_i = \begin{cases} [\nabla f(\mathrm{v})]_i\,, & \text{if} \quad [\nabla f(\mathrm{v})]_i \le 0\,; \\ [\nabla f(\mathrm{v})]_i\,, & \text{if} \quad [\nabla f(\mathrm{v})]_i > 0 \\ & \text{and} \quad v_i - \mu[\nabla f(\mathrm{v})]_i \ge \tau\,; \\ \dfrac{v_i - \tau}{\mu}\,, & \text{otherwise.} \end{cases}$$

Notice that in the latter case $(v_i - \tau)/\mu \in \big[0, [\nabla f(\mathrm{v})]_i\big)$. Given that, it is easy to see that $\delta\mathrm{v}(\mu)\cdot\nabla f(\mathrm{v}) \ge \|\delta\mathrm{v}(\mu)\|^2$ and therefore, since f is quadratic we have

$$\begin{aligned} f(\mathrm{v}) - f\big(\delta\mathrm{v}(\mu)\big) &= \mu\delta\mathrm{v}(\mu)\cdot\nabla f(\mathrm{v}) - \frac{1}{2}\mu^2\delta\mathrm{v}(\mu)\cdot\nabla^2 f(\mathrm{v})\delta\mathrm{v}(\mu) \\ &\ge \mu\|\delta\mathrm{v}(\mu)\|^2 - \frac{1}{2}\mu^2 L\|\delta\mathrm{v}(\mu)\|^2 \\ &= \mu\left(1 - \frac{\mu L}{2}\right)\|\delta\mathrm{v}(\mu)\|^2\,, \end{aligned}$$

where L is a superior bound on $\|\nabla^2 f\|$. Thus we see that with $\mu \le \frac{2(1-\upsilon)}{L}$ the criterion is satisfied. This shows that the algorithm is well defined, never getting stuck in the Armijo search procedure. It also implies the following result:

Lemma 1 Let the starting step size for the Armijo search at iteration k be $\bar{\mu}_k$ and the step size after the search be μ_k, then

$$\mu_k \ge \min\left\{\bar{\mu}_k, \frac{2\beta(1-\upsilon)}{L}\right\}.$$

The proposed algorithm satisfies

$$f(\mathrm{v}^0) \ge f(\mathrm{v}^k) + \upsilon\sum_{i=0}^{k}\mu_i\|\delta\mathrm{v}^k(\mu_k)\|^2,$$

which in turn implies that $\sum_{k=0}^{\infty}\mu_k\|\delta\mathrm{v}^k(\mu_k)\|^2 < \infty$. Given the above lemma, this summation clearly means that $\delta\mathrm{v}^k(\mu_k) \to 0$ because $\bar{\mu}_k \ge \omega > 0$.

We can now show that the sequence generated by the algorithm converges. We first notice that $\{\mathrm{v}^k\}$ is bounded since the algorithm decreases $f(\mathrm{v}^k)$ at each step and f is such that $\|\mathrm{x}\| \to \infty \Rightarrow f(\mathrm{x}) \to \infty$. Let us then take any convergent subsequence $\{\mathrm{v}^{l_k}\}$, say $\mathrm{v}^{l_k} \to \mathrm{v}^*$. It is easy to note that $\delta\mathrm{v}^k(\mu_k) \to 0$ implies that either $[\nabla f(\mathrm{v}^*)]_i = 0$ or $[\nabla f(\mathrm{v}^*)]_i > 0$ and $v_i^* = \tau$, i.e., v^* is an optimal point. On the other hand, because the algorithm is non-decreasing, it implies that every accumulation point is optimal. Since there is only one optimal point, full convergence is proven. □