Intrinsic and Extrinsic Operators for Shape Analysis

Yu Wang* and Justin Solomon
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology (MIT)
*Correspond to wangyu9@mit.edu

May 2019

Abstract

Geometric operators are common objects in surface-based shape analysis and geometry processing. While the intrinsic Laplace–Beltrami operator has been a ubiquitous choice thanks to its intuitive and often desirable properties, it fails to capture the spatial embedding of a shape because it discards extrinsic information; furthermore, it is not always sensitive to the geometric features relevant for a given shape analysis task. To address these challenges, several alternative operators for shape analysis have been proposed in recent work, with an emphasis on operators sensitive to extrinsic features. Many operators appearing in previous work on other problems also encode aspects of extrinsic geometry and are potentially suitable for shape analysis. In this survey, we unify discussion of operators for shape analysis, highlighting key theoretical properties as well as their numerical discretizations. Additionally, we provide numerical experiments on model tasks in the operator-based shape analysis pipeline, including computation of descriptors, distances, and segmentations, to demonstrate the effect of using different operators on the qualitative behavior of algorithms in this space.

Keywords: Shape Analysis, Geometric Operator, Linear Operator, Spectral Geometry, Laplace–Beltrami, Extrinsic Geometry, Intrinsic Geometry.

Contents

To appear in "*Processing, Analyzing and Learning of Images, Shapes, and Forms*," October 2019, Handbook of Numerical Analysis, edited by Ron Kimmel and Xue-Cheng Tai. Primary AMS code: 65M70. Secondary AMS code: 58J50 and 35P05.

1 Introduction

Two viewpoints typify classical measurements for analyzing surface geometry. The *extrinsic* approach considers a surface as it is embedded in three dimensions, relying on distances in the underlying Euclidean space $\mathbb{R}^3$. The *intrinsic* approach, in contrast, restricts consideration to quantities that can be measured without leaving the surface, such as geodesic distances, decoupling surface geometry from its embedding. Historically, the intrinsic approach played a central role in the development of the modern theory of geometry. Largely a byproduct of its prevalence in the mathematical literature as well as a preference for isometry invariance in certain applications, this perspective has become pervasive in computer graphics, geometry processing, and machine learning, leading to many successful algorithms. In practice, that intrinsic methods are inherently invariant to isometry contributes to their popularity: Real-world objects often deform in a nearly-isometric fashion, and intrinsic methods by design do not distinguish poses under these transformations. This is often a desirable feature for applications like shape classification.

The intrinsic approach, however, employs an incomplete description of a surface, in the sense that the embedding as well as critical cues for humans' perception of shape cannot be recovered from intrinsic measurements alone. As a simple illustration, Figure 1 shows cubes with inward or outward bumps; these models are different extrinsically but identical intrinsically. As an extreme example, all origami models are intrinsically equivalent to flat sheets of paper. Hence, shape analysis pipelines that desire to use or learn from all possible geometric cues require reconsideration of their intrinsic building blocks, discarding extrinsic information only if it is truly irrelevant for a given task.

A critical example of the intrinsic–extrinsic divide appears in the operator-based approach to geometry processing. *Linear operators*, such as the intrinsic Laplace–Beltrami operator (or Laplacian for short), play a central role in geometry processing and shape analysis, applied to tasks including segmentation,

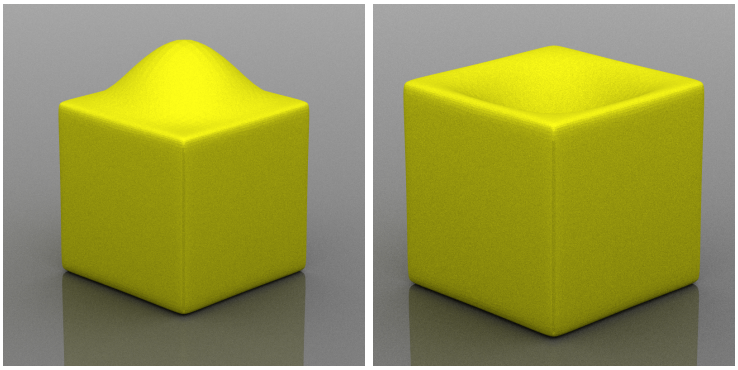


Figure 1: Two *isometric* cubes with an inward or outward bump.

retrieval, parameterization, correspondence, local descriptors, deformation, physical simulation, and deep learning. In particular, *Laplacian-based shape analysis* has achieved remarkable success and popularity for most, if not all, of these tasks. By removing extrinsic information, however, the purely intrinsic surface Laplacian operator can lead to undesirable behavior in shape analysis algorithms. For example, Laplacian-based algorithms cannot distinguish between the two cubes in Figure 1.

Analyzing shapes using alternative operators has attracted recent interest [Liu et al., 2017, Wang et al., 2018b], avoiding the loss of information associated to purely intrinsic methods. These operators can capture aspects of extrinsic geometry and can be sensitive to features other than those typically detected by the Laplacian [Wu and Levine, 1997, Liu and Zhang, 2007, Raviv et al., 2010, Hildebrandt et al., 2012, Au et al., 2012, Wang et al., 2014, Andreux et al., 2014, Ye et al., 2018]. While the Laplacian is the "default" choice in many shape analysis methods, replacing it with another operator can immediately yield algorithms with different—and often improved—behavior. This substitution provides geometric insight that contrasts from that derived using Laplacian-based constructions but yields additional information about shape. For this reason, this survey compares relevant operators, including the Laplacian, the Dirac operator, boundary integral operators, and others, in a unified framework. We provide experiments to highlight similarities and differences among the many options.

2 Preliminaries

2.1 Extrinsic and Intrinsic Geometry

We briefly review basic concepts and theoretical constructions for analyzing intrinsic and extrinsic geometry in the paragraphs below; interested readers can refer to texts in differential geometry [Do Carmo, 2016, Frankel, 2011, Lee, 2013] for expanded mathematical discussion and to [Botsch et al., 2010] for a practical guide to applied geometry.

Intrinsic Geometry Quantities that can be measured without leaving a surface $\mathcal{M}$ are considered intrinsic. The core intrinsic measurement on a surface is its *first fundamental form* (or *metric tensor*), which provides a means of measuring lengths and angles of vectors in the tangent space. Typically, this tensor per point on the surface is expressed as a matrix $g(\mathbf{x}) \in \mathbb{R}^{2\times 2}$ giving the inner products of basis vectors for the tangent space at $\mathbf{x} \in \mathcal{M}$. We denote entries in the metric tensor as g_{ij}, where $i, j \in \{1, 2\}$, and we use $(g^{ij}) = (g^{-1})_{ij}$ to denote the inverse metric. Intrinsic quantities like geodesic distances are fully-determined by the metric of a surface.

The metric allows us to perform calculations about a surface abstractly, without its embedding in $\mathbb{R}^3$. Associating each point with a metric tensor g is already sufficient to define a rich family of mathematical objects. Intrinsic operators like the Laplacian, as we will see later in §5.2, can be defined using the metric g only, remaining invariant if the embedding changes so long as the metric g remains the same.

A crowning result of classical differential geometry is that *Gaussian curvature*, which compares the circumference of an infinitesimal geodesic circle to that of a circle in the plane with the same area, is intrinsic. The better-known definition of Gaussian curvature as $K(\mathbf{x}) = \kappa_1(\mathbf{x})\kappa_2(\mathbf{x})$, i.e., the product of the two *principal curvatures* $\kappa_1(\mathbf{x}), \kappa_2(\mathbf{x})$ at a point $\mathbf{x} \in \mathcal{M}$, involves the eigenvalues of the extrinsic second fundamental form (below).

Extrinsic Geometry Extrinsic geometry considers a surface as an object embedded in $\mathbb{R}^3$. In classical differential geometry, when taking this view, it is typical to treat a surface locally using a parameterization $\mathbf{r}(u,v)\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}^3$ and then apply the analytical tools developed in multivariable calculus to $\mathbf{r}(\cdot,\cdot)$. In addition to the first fundamental form, this embedding enables measurement of the *second fundamental form*, a 2×2 matrix giving the rate of change of the surface normal; the eigenvalues of the second fundamental form are the principal curvatures $\kappa_1(\mathbf{x}), \kappa_2(\mathbf{x})$ mentioned above. The mean curvature at a point $\mathbf{x}\in\mathcal{M}$, defined as average of the two principal curvatures $H(\mathbf{x}) = \big(\kappa_1(\mathbf{x})+\kappa_2(\mathbf{x})\big)/2$, is an extrinsic quantity.

2.2 Operators and Spectra

The operator-based approach to shape analysis constructs an operator (discretely, a matrix) that acts on functions over the surface; for example, the Laplacian roughly maps functions to their second derivatives. Then, the eigenvalues (or, *spectrum*) of the operator and its eigenfunctions encode geometric features. This representation is convenient for a number of reasons. Whereas the list of vertex positions of a triangle mesh is only known up to permutation, rigid motion, and potentially deformation, operators easily can be designed whose list of eigenvalues are invariant to these irrelevant changes.

For many choices of operators including the Laplacian [Zeng et al., 2012, Boscaini et al., 2015, Chern et al., 2018], the Hessian operator [Stein et al., 2018], and the Dirichlet-to-Neumann operator [Wang et al., 2018b], the operator encodes *all* information about the shape up to a class of transformations, in the sense that one can recover an embedding from the operator and/or its spectrum (up to isometry or rigid transformation, depending on the operator). In this case, the operator can be thought as an alternative representation of the shape, providing a *lossless* means of computing geometric information for algorithms downstream. For this reason, it is fairly common for shape analysis methods to require only access to the operator constructed from a shape rather than access to the shape itself.

Spectral methods extract geometric information directly from operator eigenvalues and eigenfunctions [Zhang et al., 2010a]. As we will see later, spectral methods naturally separate features at different scales and extract global features about a shape, making them suitable for design of robust algorithms.

3 Theoretical Aspects and Numerical Analysis

In this section, we present a brief introduction to the theory of linear operators and their applicability to shape analysis; we also introduce general techniques for their discretization.

3.1 Basics of Linear Operators

Let $\mathcal{M}$ be a smooth surface possibly with boundary $\partial\mathcal{M}$, and let $\mathcal{L}^2(\mathcal{M})$ be the space of square (Lebesgue) integrable functions. A linear operator is a map $\mathcal{A}\colon \mathcal{L}^2(\mathcal{M})\to\mathcal{L}^2(\mathcal{M})$ taking in one function on the surface and returning another function, such that $\mathcal{A}(u+v)=\mathcal{A}u+\mathcal{A}v$ and $\mathcal{A}(c\cdot u)=c\cdot\mathcal{A}u$ for $c\in\mathbb{R}$.

For our purposes, we can simply think of the function space $\mathcal{L}^2(\mathcal{M})$ as analogous to the vector space $\mathbb{R}^n$ and the set of operators $\mathcal{L}^2(\mathcal{M})\to\mathcal{L}^2(\mathcal{M})$ as analogous to the set of matrices $\mathbb{R}^{n\times n}$, where n is the number of vertices in a triangle mesh. A complete, rigorous treatment of operators involves functional analysis and linear operator theory, which is beyond the scope of this paper; we refer interested readers to [Brezis, 2010] for discussion.

Equivalent Strong and Weak Forms A linear operator and the PDEs associated with it usually can be expressed using an equivalent integral equation. Let us start with the Laplace equation with Neumann boundary conditions on $\mathcal{M}$:

$$\begin{cases} -\Delta_{\mathcal{M}} u(\mathbf{x}) = f(\mathbf{x}) & \mathbf{x}\in\mathcal{M}\setminus\partial\mathcal{M} \\ \nabla_{\mathbf{n}(\partial\mathcal{M})} u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x}\in\partial\mathcal{M} \end{cases} \tag{1}$$

in which $\Delta_{\mathcal{M}}$ is the Laplacian operator on the curved surface, and $\nabla_{\mathbf{n}(\partial\mathcal{M})}$ is the normal derivative operator, satisfying $\nabla_{\mathbf{n}(\partial\mathcal{M})}u(\mathbf{x}) = \mathbf{n}(\mathbf{x})\cdot\nabla_{\mathcal{M}}u(\mathbf{x})$ where $\nabla_{\mathcal{M}}$ is the intrinsic gradient operator and $\mathbf{n}(\mathbf{x})$ is the normal to the boundary at $\mathbf{x}\in\partial\mathcal{M}$. For the purpose of discussion in this section, readers may simply think of the special case where $\mathcal{M}\subseteq\mathbb{R}^2$, i.e. a planar domain in $\mathbb{R}^2$, for which $\Delta_{\mathcal{M}}$ becomes $\Delta_{\mathbb{R}^2} = \frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}$ and $\nabla_{\mathcal{M}}$ becomes the usual gradient operator in $\mathbb{R}^2$. The general definition of $\Delta_{\mathcal{M}}$ is postponed to §5.2, although our discussion below holds in the general case when $\mathcal{M}$ becomes a curved

surface. To simplify notation, we omit the subscript in $\Delta_{\mathcal{M}}$, $\nabla_{\mathcal{M}}$, and $\nabla_{\mathbf{n}(\partial\mathcal{M})}$ when these operators are used within integral $\int_{\mathcal{M}}$, $\int_{\mathcal{M}}$, and $\int_{\partial\mathcal{M}}$, respectively.

Eq. 1 gives a PDE that is known as the *strong form* of the Laplace equation. If we integrate both sides against an arbitrary function $v(\mathbf{x})$ and sum the results, we obtain the *weak form* of the same problem:

$$-\int_{\mathcal{M}} v(\mathbf{x})\Delta u(\mathbf{x}) + \int_{\partial\mathcal{M}} v(\mathbf{x})\nabla_{\mathbf{n}} u(\mathbf{x}) = \int_{\mathcal{M}} v(\mathbf{x}) f(\mathbf{x}) + \int_{\partial\mathcal{M}} v(\mathbf{x}) g(\mathbf{x}), \quad \forall v(\mathbf{x}). \tag{2}$$

In this formula, the function $v(\mathbf{x})$ is known as a *test function*. Under appropriate assumptions about the function spaces containing $u(\mathbf{x})$ and $v(\mathbf{x})$, one can show that the weak form and the strong form are solved by the same $u(\mathbf{x})$. Roughly speaking, when $v(\mathbf{x})$ is chosen as a delta function, the strong form is recovered: The relationship involving $f(\mathbf{x})$ appears when the delta function is in the interior $\mathcal{M}\backslash\partial\mathcal{M}$, and the relationship involving $g(\mathbf{x})$ appears when the delta function is on the boundary $\partial\mathcal{M}$. Using Green's first identity,

$$\int_{\mathcal{M}} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) = -\int_{\mathcal{M}} v(\mathbf{x})\Delta u(\mathbf{x}) + \int_{\partial\mathcal{M}} v(\mathbf{x})\nabla_{\mathbf{n}} u(\mathbf{x}). \tag{3}$$

The left side of the weak form in Eq. 3 gives a symmetric *bilinear form* $\int_{\mathcal{M}} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x})$, which takes two input functions u, v and outputs a scalar. When $v = u$, the bilinear form becomes $\int_{\mathcal{M}} \|\nabla u(\mathbf{x})\|_2^2$, the *Dirichlet energy* measuring the smoothness of u.

Our discussion so far introduces an equivalence between a PDE (1) and an integral equation (2) specified by a bilinear form $\int_{\mathcal{M}} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x})$. This equivalence can be generalized to a large range of PDEs with natural boundary conditions.

Recall that a bilinear form $a(u, v)$ is linear in the arguments u, v, i.e., $a(u_1+u_2, v) = a(u_1, v)+a(u_2, v)$ and $a(c \cdot u, v) = c \cdot a(u, v)$ $\forall c \in \mathbb{R}$, and analogously for the second argument v. Bilinear forms and linear operators generalize matrix operations in finite-dimensional spaces to infinite-dimensional spaces. Hence, we can make definitions for continuous operators similar to their counterparts in linear algebra, including the following key classes of bilinear operators:

- *Self-adjoint*: We call a bilinear form self-adjoint if $\forall u, v : a(u, v) = a(v, u)$. This generalizes the definition of symmetric matrices $\mathbf{A}$, for which $\mathbf{A} = \mathbf{A}^\intercal$.
- *Positive semidefinite*: We call a self-adjoint bilinear form positive semidefinite (p.s.d.) if we have $a(u, u) \geq 0$ $\forall u$; the form is positive definite (p.d.) if $a(u, u) = 0$ implies $u \equiv 0$. This generalizes the definition of a p.s.d. matrix $\mathbf{A}$, for which $\mathbf{u}^\intercal\mathbf{A}\mathbf{u} \geq 0$ $\forall \mathbf{u} \in \mathbb{R}^n$, and a p.d. matrix $\mathbf{A}$, which additionally satisfies the condition that $\mathbf{u}^\intercal\mathbf{A}\mathbf{u} = 0$ implies $\mathbf{u} \equiv \mathbf{0}$.

Our discussion of the Laplacian follows a template common in functional analysis, in which linear operators are converted to bilinear forms; the linear operator is the strong form and the bilinear form is the weak form. In particular, given a linear operator $\mathcal{A}\colon \mathcal{L}^2(\mathcal{M}) \mapsto \mathcal{L}^2(\mathcal{M})$ we can define a corresponding bilinear form $a(u, v) : \mathcal{L}^2(\mathcal{M}) \times \mathcal{L}^2(\mathcal{M}) \to \mathbb{R}$ as $a(u, v) = \langle u, \mathcal{A}v\rangle_{\mathcal{M}}$, where $u, v \in \mathcal{L}^2(\mathcal{M})$. In the reverse direction, a linear operator can be recovered from a bilinear form, e.g., by applying the Riesz Representation Theorem [Brezis, 2010]. Such equivalence is trivial in finite-dimensional spaces: Both bilinear forms and linear maps are given by matrices.

This equivalence allows us to equate the problem $\mathcal{A}u = f$ with a weak form

$$a(u, v) = \langle f, v\rangle_{\mathcal{M}} \quad \forall v \tag{4}$$

where $\langle u, v\rangle_{\mathcal{M}} := \int_{\mathcal{M}} u(\mathbf{x})v(\mathbf{x})$ is the inner product on $\mathcal{L}^2(\mathcal{M})$, under mild assumptions that u, v, f come from appropriate function spaces—usually a Sobolev space, consisting of functions whose low-order partial derivatives have bounded $\mathcal{L}^2$ integrals. Returning to the Laplacian operator $\Delta_{\mathcal{M}}$, its corresponding bilinear form is $a(u, v) = \int_{\mathcal{M}} \nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x})$; note that fewer derivatives are required of u, v to define the bilinear form than to define the linear operator.

Variational Formulation Given functions $f(\mathbf{x})$ and $g(\mathbf{x})$, consider the variational problem for a function $u(\mathbf{x})$:

$$\min_{u(\cdot)} \int_{\mathcal{M}} \Big(\|\nabla u(\mathbf{x})\|_2^2 - u(\mathbf{x})f(\mathbf{x})\Big) - \int_{\partial\mathcal{M}} u(\mathbf{x})g(\mathbf{x}). \tag{5}$$

Using variational calculus, one can show that problems (1) and (5) are solved by the same function $u(\mathbf{x})$. This relationship justifies calling the Neumann condition $\nabla_{\mathbf{n}} u(\mathbf{x}) = g(\mathbf{x})$ in (1) by its alternative name, the *natural* boundary condition, since it arises from the variational problem (5) without an explicit boundary constraint.

Eigenvalue Problem The eigenvalue problem associated to operator $\mathcal{A} : \mathcal{H} \to \mathcal{H}$ is defined as follows:

$$\mathcal{A}\phi = \lambda\phi, \tag{6}$$

where $\lambda \in \mathbb{R}$ is known as an eigenvalue and $\phi(\cdot)$ is its corresponding eigenfunction. The *spectral theorem* states that in the most common case, namely when $\mathcal{A}$ is a compact self-adjoint operator and $\mathcal{H}$ is a separable Hilbert space [Zhu, 2007], there are countably many eigenvalues and corresponding eigenfunctions. We mainly consider this case in our survey, and hence we use $\{\lambda_i\}_{i=0}^{\infty}$ and $\{\phi_i(\mathbf{x})\}_{i=0}^{\infty}$ to denote the sets of eigenvalues and corresponding eigenfunctions of $\mathcal{A}$, respectively, sorted in ascending order such that $\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots$.

Known as the *Courant–Fischer min-max Theorem*, the strong form (6) can be converted to an equivalent weak form by finding saddle points of the optimization problem

$$\begin{cases} \min\limits_{\phi(\cdot)} & a(\phi, \phi) \\ \text{s.t.} & \langle \phi, \phi \rangle_{\mathcal{M}} = 1. \end{cases} \tag{7}$$

Assuming $a(\cdot, \cdot)$ is symmetric, we can follow the convention that $\{\phi_i(\mathbf{x})\}_{i=0}^{\infty}$ are orthonormal: Eigenfunctions corresponding to different λ's must be orthogonal, and applying Gram–Schmidt orthogonalization to eigenfunctions with the same λ, followed by normalization, ensures that $\{\phi_i(\mathbf{x})\}_{i=0}^{\infty}$ are orthonormal. A consequence of the spectral theorem, for many choices of operators $\mathcal{A}$, is that the ϕ_i's form a *complete* orthonormal basis;[1] in classical mathematics, the completeness of the Laplacian is a consequence of the Sturm-Liouville decomposition [Chavel, 1984, Rosenberg, 1997]. Laplacian eigenfunctions are also known as *manifold harmonics*. When the surface is a sphere, the Laplacian eigenfunctions are called *spherical harmonics*.

The *spectrum* of an operator, $\{\lambda_i\}_{i=0}^{\infty}$, is the generalization of eigenvalues of a matrix. This spectral decomposition of $\mathcal{A}$, as we will see later, extracts information about $\mathcal{M}$, from large- to small-scale.

3.2 PDEs and Green's Functions

A few classic PDEs written in terms of the Laplacian are widely used in shape analysis and provide physical intution for behavior of algorithms in spectral geometry. As we will see later, solutions to these problems closely relate to the spectrum of the Laplacian. For simplicity, starting from this section in the paper we only consider closed surfaces (i.e., ones without boundaries), unless noted otherwise and in which case boundary conditions will be explicitly discussed.

Laplace Equation The Laplacian admits a *Green's function*, also known as the *fundamental solution*, given by a function $G(\mathbf{x}, \mathbf{y})$ satisfying

$$\Delta_{\mathbf{x}} G(\mathbf{x}, \mathbf{y}) = \delta_{\mathbf{y}}(\mathbf{x}), \tag{8}$$

where $\Delta_{\mathbf{x}}$ is still the operator $\Delta_{\mathcal{M}}$ with the notation chosen to emphasize that it operates on $\mathbf{x}$ and $\delta_{\mathbf{y}}(\cdot)$ denotes the Dirac δ-function centered at $\mathbf{y}$. Given G, up to an additive constant function, the solution to the Laplace equation $\Delta_{\mathcal{M}} u(\mathbf{x}) = f(\mathbf{x})$ for "smooth enough" $f(\cdot)$ can be written in closed-form as

$$u(\mathbf{x}) = \int_{\mathcal{M}} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, \mathrm{d}\mathbf{y}.$$

In other words, the linear operator $f(\cdot) \mapsto \int_{\mathcal{M}} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, \mathrm{d}\mathbf{y}$ based on the Green's function "inverts" the original Laplacian operator. The Green's function of a linear operator plays an analogous role to the pseudo-inverse of a (finite-dimensional) matrix: Recall that for a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ whose null space contains only constant vectors, the solution to the linear system $\mathbf{A}\mathbf{u} = \mathbf{f}$ is $\mathbf{u} = \mathbf{A}^{\dagger}\mathbf{f}$ up to an additive constant, when the right-hand side $\mathbf{f} \in \mathbb{R}^{n \times 1}$ sums to zero; here $\mathbf{A}^{\dagger}$ is the (Moore–Penrose) pseudo-inverse of $\mathbf{A}$ that satisfies the relationship $\mathbf{A} \cdot \mathbf{A}^{\dagger} = \mathbf{I} - \frac{1}{n}\mathbf{1}^{\intercal}\mathbf{1}$, similar to the definition in Eq. (8).

Given the spectral decomposition of $\Delta_{\mathcal{M}}$ defined in §3.1, we have $(\Delta_{\mathcal{M}} u)(\mathbf{x}) = \int_{\mathcal{M}} \sum_{k=0}^{\infty} \lambda_k \phi_k(\mathbf{x}) \phi_k(\mathbf{y}) u(\mathbf{y}) \, d\mathbf{y}$. This expression first decomposes $f(\cdot)$ into an orthonormal basis, and multiplies each component with the corresponding eigenvalue: high-frequency components are amplified by a factor of λ_k, so $\Delta_{\mathcal{M}}$ can be thought as a high-pass filter in the signal processing view. The following spectral expansion gives an

[1] Completeness requires that any function in $\mathcal{L}^2(\mathcal{M})$ can be approximated arbitrarily closely by linear combinations of $\{\phi_i(\mathbf{x})\}_{i=0}^{\infty}$.

explicit formula for the Green's function, which can be understood as the pseudo-inverse of the operator $\Delta_{\mathcal{M}}$:

$$G(\mathbf{x}, \mathbf{y}) = \sum_{k=n_0}^{\infty} \frac{1}{\lambda_k} \phi_k(\mathbf{x}) \phi_k(\mathbf{y}). \tag{9}$$

Here, n_0 is the number of zero eigenvalues of $\Delta_{\mathcal{M}}$; $n_0 = 1$ for simply connected surfaces. The corresponding linear operator $f(\cdot) \mapsto \int_{\mathcal{M}} G(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, \mathrm{d}\mathbf{y}$, conversely, is a low-pass filter, in the sense that low-frequency components are exaggerated and emphasized by the multiplicative factor $1/\lambda_k$.

Heat Equation The heat equation (also known as the diffusion equation) describes a time-varying evolution of a function $u(\mathbf{x}, t)$ given its initial distribution $u(\mathbf{x}, 0)$. Physically, this PDE is used to determine the spatial distribution of temperature on a conductive surface after it diffuses for time t:

$$\textit{(Heat equation)} \qquad \frac{\partial}{\partial t} u(\mathbf{x}, t) = \Delta_{\mathbf{x}} u(\mathbf{x}, t) \quad \text{s.t.} \quad u(\mathbf{x}, 0) = f(\mathbf{x}). \tag{10}$$

The *heat kernel* $k_t(x, y)$ specifies the transition density function from $\mathbf{x}$ to $\mathbf{y}$ over a time period t:

$$k_t(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(\mathbf{x}) \phi_i(\mathbf{y}). \tag{11}$$

Figure 2 visualizes (one column of) the heat kernel in log scale, i.e., $\log k_t(\mathbf{x}, \cdot)$, where $\mathbf{x}$ locates at the dot in the center of the isolines. $k_t(\mathbf{x}, \cdot)$ corresponds the temperature distribution at time t, when the initial distribution is $\delta_{\mathbf{x}}(\cdot)$, a Dirac δ-function centered at $\mathbf{x}$. It is not a coincidence that $\log k_t(\mathbf{x}, \cdot)$ looks like $d_g(\mathbf{x}, \cdot)$, the geodesic distance to $\mathbf{x}$: a connection between them is given in §3.4.

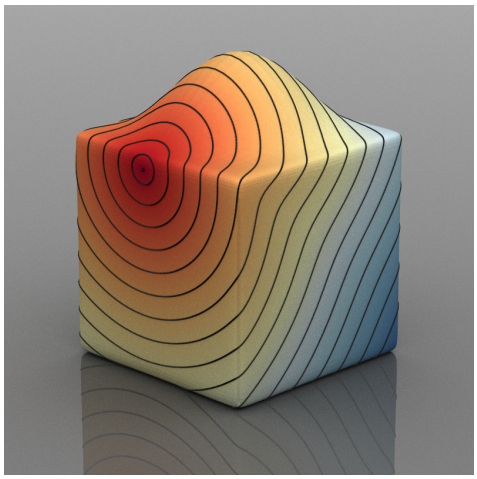


Figure 2: Heat kernel in log scale, $\log k_t(\mathbf{x}, \cdot)$.

Using heat kernel, the solution to the heat equation can be written as

$$u(\mathbf{x}, t) = \int_{\mathcal{M}} k_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, \mathrm{d}\mathbf{y}. \tag{12}$$

Substituting (11), this expression for u first decomposes $f(\cdot)$ into an orthonormal basis associated to the PDE, and then each component decays independently at a rate associated to its eigenvalue. The heat kernel act similarly to Eq. 9: the linear operator $f(\cdot) \mapsto \int_{\mathcal{M}} k_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, \mathrm{d}\mathbf{y}$ can be understood as the exponential of $\Delta_{\mathcal{M}}$. If we interpret $u(\mathbf{x}, t)$ as a probability density function, then $k_t(\mathbf{x}, \mathbf{y})$ in (12) provides the transition density function for Brownian motion [Knight et al., 1981]. Heat diffusion is a Markov process: $k_{t+s}(\mathbf{x}, \mathbf{y}) = \int_{\mathcal{M}} k_t(\mathbf{x}, \mathbf{z}) k_s(\mathbf{y}, \mathbf{z}) \, \mathrm{d}\mathbf{z}$, reflecting the fact that heat starting at $\mathbf{x}$ has to diffuse to $\mathbf{y}$ via some $\mathbf{z}$ in a "memory-less" fashion. These facts are starting points to prove convergence of some discrete approximations to the Laplace–Beltrami operator [Coifman and Lafon, 2006].

Wave and Schrödinger Equations Wave propagation is governed by the wave equation, in which $u(\mathbf{x}, t)$ measures the magnitude of displacement at $\mathbf{x}$ at time t:

$$\textit{(Wave equation)} \qquad \frac{\partial^2}{\partial t^2} u(\mathbf{x}, t) = \Delta_{\mathbf{x}} u(\mathbf{x}, t) \quad \text{s.t.} \quad u(\mathbf{x}, 0) = f(\mathbf{x}) \quad \text{and} \quad \frac{\partial}{\partial t} u(\mathbf{x}, 0) = g(\mathbf{x}). \tag{13}$$

The wave equation, as a second order differential equation, also requires initial conditions on $\frac{\partial}{\partial t} u(\mathbf{x}, 0)$ to yield a unique solution $u(\cdot, \cdot)$, making the discussion complicated; instead, here we examine a similar first-order PDE.

Closely related to the wave equation, the *Schrödinger equation* in Eq. 14 prescribes the evolution of a quantum particle via the wave function $u(\mathbf{x}, t) \colon \mathcal{L}^2(\mathcal{M}) \times \mathbb{R}^+ \to \mathbb{C}$ such that $|u(\mathbf{x}, t)|^2$ is interpreted as the particle's probability density function (p.d.f.) at location $\mathbf{x}$ and time t:

$$\textit{(Schrödinger equation)} \qquad i \frac{\partial}{\partial t} u(\mathbf{x}, t) = -\Delta_{\mathbf{x}} u(\mathbf{x}, t) \quad \text{s.t.} \quad u(\mathbf{x}, 0) = f(\mathbf{x}) \tag{14}$$

where $i = \sqrt{-1}$. This equation is later used to define the wave kernel signature (see §4.3) and the Hamiltonian/Schrödinger operator (see §5.10).

Generalization The PDEs above can be generalized by replacing the operator $-\Delta$ with a general p.s.d. operator $\mathcal{A}$:

$$\textit{(Poisson equation)} \qquad \mathcal{A} u(\mathbf{x}) = f(\mathbf{x}) \tag{15}$$

$$\textit{(Heat equation)} \qquad \frac{\partial}{\partial t} u(\mathbf{x}, t) = -\mathcal{A} u(\mathbf{x}, t) \quad \text{s.t.} \quad u(\mathbf{x}, 0) = f(\mathbf{x}) \tag{16}$$

$$\textit{(Schrödinger equation)} \qquad i \frac{\partial}{\partial t} u(\mathbf{x}, t) = \mathcal{A} u(\mathbf{x}, t) \quad \text{s.t.} \quad u(\mathbf{x}, 0) = f(\mathbf{x}) \tag{17}$$

Solutions to these PDEs can be written in the unified form $u(\mathbf{x}) = \int_{\mathcal{M}} G_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, \mathrm{d}\mathbf{y}$, where

$$G_t(\mathbf{x}, \mathbf{y}) = \begin{cases} \sum_{k=n_0}^{\infty} \frac{1}{\lambda_k} \phi_k(\mathbf{x}) \phi_k(\mathbf{y}) & \text{Poisson equation} \\ \sum_{k=0}^{\infty} e^{-\lambda_k t} \phi_k(\mathbf{x}) \phi_k(\mathbf{y}) & \text{Heat equation} \\ \sum_{k=0}^{\infty} e^{i \lambda_k t} \phi_k(\mathbf{x}) \phi_k(\mathbf{y}) & \text{Schrödinger equation} \end{cases}$$

is the Green's function for each PDE and $\{\phi_k(\cdot)\}_{k=0}^{\infty}$ and $\{\lambda_k(\cdot)\}_{k=0}^{\infty}$ are eigenfunctions and eigenvalues of the operator $\mathcal{A}$, such that $\lambda_0 = 0, \ldots, \lambda_{n_0} = 0, \lambda_{n_0+1} > 0$. For fixed t, the map $f(\cdot) \mapsto \int_{\mathcal{M}} G_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) \, \mathrm{d}\mathbf{y}$ is also a linear operator, which can be thought as the pseudo-inverse or exponential of $\mathcal{A}$, i.e., $\mathcal{A}^{\dagger}, e^{-t\mathcal{A}}, e^{it\mathcal{A}}$, respectively, for these PDEs. Loosely speaking, solving these three PDEs amounts to applying the three operators to $f(\cdot)$. For additional details, we refer the reader to [Evans, 1998].

When $\mathcal{A} = -\Delta_{\mathcal{M}}$, the exponential operator becomes the Laplacian heat kernel $e^{t\Delta_{\mathcal{M}}}$. This relation allows us to compute the heat kernel from the Laplacian operator, bypassing the difficulty of discretizing the "exponentiated" operator directly. When $\mathcal{A}$ is a covariant derivative operator or connection Laplacian operator, the exponentiated operator becomes an operator of parallel transport along vector field or geodesics, respectively, with applications to vector field analysis and processing [Azencot et al., 2015, Sharp et al., 2018] and manifold learning [Singer and Wu, 2012, Wu, 2013].

3.3 Operator Derivation and Discretization

An operator associated to a smooth surface has to be discretized to be used in the computational setting; there are many techniques for operator discretization that are relevant to this task. On regular grids, differentials can be approximated by finite differences, in a consistent manner such that the approximation converges to the differential quantity as the grid resolution increases [Atkinson and Han, 2005]. On irregular domains like meshes, many discretization methods are possible, including the finite element method (FEM) and the finite volume method (FVM).

The goal of operator discretization or construction is to obtain a matrix $\mathbf{A}$, accompanied with a *mass matrix* $\mathbf{M}$, such that the continuous equation $\mathcal{A} u = f$ is discretized as the linear equation $\mathbf{A}\mathbf{u} = \mathbf{M}\mathbf{f}$, where $\mathbf{u}, \mathbf{f}$ are vectors discretizing $u(\cdot)$ and $f(\cdot)$, e.g., $\mathbf{u}, \mathbf{f}$ can be vectors storing the values of $u(\cdot)$ and $f(\cdot)$ at mesh vertices, respectively. In the scenario that function $u(\cdot)$ is given, intuitively the action of an operator $\mathcal{A}$, i.e., applying $\mathcal{A}$ to a scalar function $u \in \mathcal{L}^2(\mathcal{M})$ to yield $f \in \mathcal{L}^2(\mathcal{M})$, is approximated by

matrix multiplication: $\mathbf{f} = \mathbf{M}^{-1}\mathbf{A}\mathbf{u}$; ideally $\mathbf{f}$ should converge to the function $f(\cdot)$ under mesh refinement. In the other scenario when function $f(\cdot)$ is given, usually the major case of interest, ideally $\mathbf{u} = \mathbf{A}^{-1}\mathbf{M}\mathbf{f}$ should converge to the continuous solution $u(\cdot)$ under mesh refinement. The weak form corresponding to $\mathcal{A}$ is approximated by the discrete quadratic form: $a(u, v) \approx \mathbf{u}^\intercal \mathbf{A}\mathbf{v}$; again $\mathbf{u}, \mathbf{v}$ are discretization of functions $u(\cdot), v(\cdot)$. We summarize the typical approaches used to obtain the matrices $\mathbf{A}$ and $\mathbf{M}$ below.

Finite and Boundary Element Methods The *finite element method (FEM)* and *boundary element method (BEM)* [Steinbach, 2007, Śmigaj et al., 2015] are frameworks for discretizing operators in a convergent fashion. The idea of FEM and BEM is to consider the weak form (4), which has to hold for every test function v, and to restrict u, v and f to finite-dimensional linear subspaces $\{\phi_i(\mathbf{x})\}_{i=1}^n$ and $\{\psi_i(\mathbf{x})\}_{i=1}^m$. This finite-dimensional approximation can be written $u = \sum_{i=1}^n \mathbf{u}_i \phi_i$, $f = \sum_{i=1}^m \mathbf{f}_i \psi_i$, where $\mathbf{u} \in \mathbb{R}^n, \mathbf{f} \in \mathbb{R}^m$ stack the coefficients of u and f in the basis. Allowing the test function v to be any basis function in $\{\psi_i(\mathbf{x})\}_{i=1}^m$, Eq. (4) becomes a finite-dimensional linear system $\mathbf{A}\mathbf{u} = \mathbf{M}\mathbf{f}$, where

$$\mathbf{A} \in \mathbb{R}^{m \times n} : \mathbf{A}_{ij} = a(\psi_i, \phi_j) = \langle \psi_i, \mathcal{A}\phi_j \rangle_{\mathcal{M}} \tag{18}$$

$$\mathbf{M} \in \mathbb{R}^{m \times m} : \mathbf{M}_{ij} = \langle \psi_i, \psi_j \rangle_{\mathcal{M}}. \tag{19}$$

The most commonly-used basis is the piecewise-linear basis, i.e., the "hat functions" on a triangle mesh, as well as the piecewise-constant functions, specifying per-vertex and per-triangle data, respectively. Figure 3 shows an example of piecewise-linear basis on a triangle mesh.

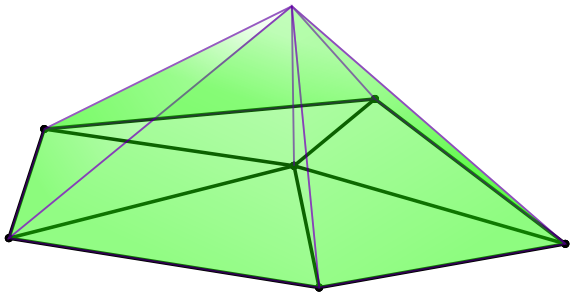


Figure 3: The "hat" basis function corresponding to the vertex in the center.

The *Galerkin method* refers to the case that $\{\phi_i(\mathbf{x})\}_{i=1}^n$ and $\{\psi_i(\mathbf{x})\}_{i=1}^m$ are chosen to be the same basis, so $\mathbf{A}$ becomes a p.s.d. matrix. FEM mostly applies to differential operators whose action is local, leading to a sparse matrix $\mathbf{A}$ with sparsity specified by the vertex adjacency. BEM follows the same procedure above, but applies to boundary integral operators instead; this leads to differences in how the inner product $a(\psi_i, \phi_j)$ is evaluated; we postpone the discussion of BEM until it is used in §5.17.

Discrete Exterior Calculus (DEC) DEC [Desbrun et al., 2005] provides an alternative framework to derive operators, building discrete operators operating on per-element quantities directly. DEC gives discrete equivalents of the exterior derivative, Hodge star, and other operators used as building blocks to construct more complicated differential operators on a surface. Integrated forms and operators are defined analogously to their continuous counterparts but on mesh elements; they are typically designed so that discrete versions of important theorems hold exactly, such as the divergence theorem or other specializations of the generalized Stokes' Theorem for differential forms [Frankel, 2011]. Sometimes the operator derived using DEC coincides with the one derived by linear FEM; the cotangent Laplacian is such an example (see §5.2).

Mixed Methods and Compositions Complicated geometric operators often can be derived as the composition of "simpler" operators, like differential operators of lower orders, reducing the difficulty of discretization. As examples, the bi-Laplacian applies the Laplacian twice [Botsch and Kobbelt, 2004, Jacobson et al., 2011]; the Hessian operator is composed of the gradient operator and the matrix divergence operator [Stein et al., 2018]; and in [Wang et al., 2018b], the Dirichlet-to-Neumann operator is constructed as the composition of a few boundary integral operators that are straightforward to discretize. Such composition usually can be justified by the mixed finite element method.

Graph Affinity Laplacian For a graph with n vertices connected by edges, its graph Laplacian $\mathbf{L} \in \mathbb{R}^{n\times n}$, is a matrix such that $\mathbf{L}_{ij} \neq 0$ for $i \neq j$ if and only if vertices i, j are adjacent and $\mathbf{L}_{ii} = -\sum_{j\neq i} \mathbf{L}_{ij}$. Spectral analyses of graph Laplacians have led to remarkable success for graph partitioning and image segmentation [Shi and Malik, 2000]. Entries in the graph Laplacian are weights measuring the affinity or similarity between vertices; a larger edge weight $\mathbf{L}_{ij}$ indicates a stronger bond between vertex i and j such that, e.g., in graph partitioning the edge ij should be less likely to be cut.

Triangle meshes can be viewed as graphs, and hence graph-based methods can be applied to meshes. Moreover, the discrete operators obtained using other methods, e.g., the cotangent Laplacian (§5.2), can be thought as particular versions of the graph Laplacian with local-geometry-aware weights that typically depend on angles and vertex areas in the triangle mesh. With this connection between discrete operators and graph Laplacians, sometimes the discrete operator matrix $\mathbf{A}$ can be constructed heuristically, as long as matrix entry $\mathbf{L}_{ij}$ provides some measure of affinity between vertex i and vertex j.

Variants and Generalizations The domain $\mathcal{D}$ and image $\mathcal{I}$ of a general linear operator $\mathcal{A}\colon \mathcal{D} \to \mathcal{I}$ are not necessarily the same function space, in which case the discrete operator becomes a rectangular matrix $\mathbb{R}^{m\times n}$ rather than a square matrix. As an example, the gradient operator (see e.g. [Tong et al., 2003]) maps a scalar function to a vector field, which live in different function spaces. We may also obtain a rectangular matrix due to the choice of basis: In Eq. 18, if $\{\phi_i(\mathbf{x})\}_{i=1}^n$ and $\{\psi_i(\mathbf{x})\}_{i=1}^m$ are chosen as spaces of piecewise linear and piecewise constant functions, respectively, then we obtain an operator of size $\mathbb{R}^{f\times n}$, i.e., #faces×#vertices. The function space can go beyond the set of real-valued functions. For example, the Dirac operator [Liu et al., 2017] maps quaternion-valued functions $\mathcal{L}^2(\mathcal{M};\mathbb{H})$ to $\mathcal{L}^2(\mathcal{M};\mathbb{H})$, and the operator becomes a quaternion-valued matrix in $\mathbb{H}^{m\times n}$. Operators can also act on vector fields or forms rather than scalar fields, e.g., divergence and curl operators [Tong et al., 2003].

Operator Properties and Desiderata When discretizing an operator, it is desirable that certain algebraic properties of the continuous operator transfer to the discrete operator. In this case, theorems and properties of some continuous functions can be preserved in a discrete sense. As an example, applying the Laplacian to a constant function yields zero function: $\Delta_{\mathbf{x}} 1(\mathbf{x}) = 0(\mathbf{x})$. Accordingly, for a discrete Laplacian $\tilde{\mathbf{L}}$, we would like $\tilde{\mathbf{L}}\mathbf{1} = \mathbf{0}$. One may ask for a discretization of a discrete operator that preserves as many of the properties of the smooth operator as possible. It is usually not possible for a discrete operator to enjoy all possible discrete analogs of smooth properties; see [Wardetzky et al., 2007] for a "no free lunch" theorem in the case of discrete Laplacians.

3.4 Operators and Geometry

Geometric PDEs [Taylor, 2011, Taylor, 2013] defined on $\mathcal{M}$ as well as geometric operators like $\Delta_{\mathcal{M}}$ encode and reveal information about the geometry and topology of $\mathcal{M}$. Several methods in shape analysis and geometry processing are based on this observation.

Geometric analysis is a branch of mathematics that applies analytical tools to operators and PDEs for purposes of studying the underlying geometry. It also connects the *global geometry* of a surface with its *topology*. As an example, elliptic differential operators on $\mathcal{M}$ satisfy the Atiyah-Singer index Theorem [Booss, 2012], which relates analytical properties of the operator to the topology of $\mathcal{M}$. As a special case, the familiar Gauss–Bonnet Theorem equates the total Gaussian curvature $\int_{\mathcal{M}} K(\mathbf{x})$ on $\mathcal{M}$ with a topological quantity—the Euler characteristic—distinguishing spheres, torii, and other surfaces with varying genus. Equipped with the tools to study geometry and topology quantitatively, Perelman's proof of the Poincaré Conjecture [Perelman, 2002, Perelman, 2003a, Perelman, 2003b] is among the most remarkable achievements of geometric analysis.

Spectral geometry studies the relationship between spectral properties of operators and the underlying geometry. As an example, the (Laplacian) heat kernel $k_t(\mathbf{x}, \mathbf{y})$ is related to the geodesic distance $d_g(\cdot,\cdot)$ via the following formula obtained by [Varadhan, 1967]:

$$\lim_{t\to 0} t \log k_t(\mathbf{x}, \mathbf{y}) = -\frac{1}{4} d_g^2(\mathbf{x}, \mathbf{y})$$

In graphics, this relation is exploited for efficient geodesic distance computation by the *heat method* [Crane et al., 2013b].

Spectrum asymptotics provide another example relating an operator with the geometry. Weyl's Law

gives the following asymptotic formula for Dirichlet eigenvalues of Laplacian on a domain $\Omega \subseteq \mathbb{R}^d$:

$$\lambda_j \sim \frac{(2\pi)^2}{(\mathrm{vol}(\Omega)|B(d)|)^{2/d}} j^{2/d}, \quad j \to \infty$$

where $|B(d)|$ is the volume of the unit ball in $\mathbb{R}^d$, and $\mathrm{vol}(\Omega)$ is the volume of Ω. See [Chavel, 1984, Rosenberg, 1997, Zelditch, 2009, Craioveanu et al., 2013] for Weyl's Law and other results relating the Laplacian spectrum to geometric quantities. The recent survey [Henrot, 2017] summarizes progress in spectral geometry with connections to shape optimization, which studies problems such as determining the shape that maximizes or minimizes λ_j for certain j.

3.5 Inverse Problems

While an operator and its spectrum can be deterministically computed on a given surface $\mathcal{M}$, *inverse problems* study what information about $\mathcal{M}$ can be recovered and extracted from an operator or its spectrum.

Shape-From-Operator A central inverse problem for operator-based geometry is to ask, given a particular operator, whether or not we can recover the shape, possibly up to rigid motion or isometry. Below we list some examples drawn from the smooth and discrete geometry literatures:

- *Laplacian*: In continuous theory, the Laplacian determines the metric of a surface; accordingly, one can recover edge lengths and vertex coordinates (up to isometry) from the discrete Laplacian operator [Zeng et al., 2012, Boscaini et al., 2015, Corman et al., 2017a, Chern et al., 2018].
- The *single layer potential* is another example of an operator for which the inverse problem above is well-posed, since it encodes the inverse distance between pairs of vertices; multi-dimensional scaling can recover the embedding from distances, up to rigid motion, see e.g. [Williams, 2002].
- The *Dirichlet-to-Neumann* operator determines the first and second fundamental forms of the underlying surface, which in turn determines its geometry up to rigid motion due to the Bonnet Theorem [Wang et al., 2018b].
- *Dirac*: Although this property is not explicitly mentioned in the geometry processing literature, it is also straightforward to see that a mesh can be recovered up to translation from the strong form of discrete relative Dirac operator [Crane et al., 2011, Liu et al., 2017, Kostrikov et al., 2018], which stacks edge vectors in an edge–vertex adjacency graph.

Algorithms have been proposed to recover shape from forms [Wang et al., 2012], metrics [Chern et al., 2018], and operators [Boscaini et al., 2015, Corman et al., 2017a], although many of these techniques rely on nonlinear methods without guarantee of success.

Shape-From-Spectrum A more difficult question is whether or not it is possible to uniquely recover a shape from an operator's eigenvalues only, again up to rigid motion or isometry. This is a famous problem in mathematics: "Can one hear the shape of a drum?," as posed by [Kac, 1966]. This question is also of practical interest: Eigenvalues are used as "ShapeDNA," global shape signatures, and "shape2vector" features, using various operators [Reuter et al., 2006, Liu et al., 2017, Wang et al., 2018b], and hence the inverse problem is equivalent to asking whether these signatures are unique/invertible.

For the Laplacian operator, in the general setting the answer is negative: there exist isospectral surfaces [Gordon et al., 1992a, Gordon et al., 1992b]. However, if the shape space is restricted to convex planar regions with analytic boundary, the answer is positive [Zelditch, 2000]. Furthermore, it remains an open question for non-convex analytic planar domains; also, for the case that the shape is a closed 2-manifold or 2-manifold with analytical boundary, which attracts the most practical interest, the question is still open. The recent work [Cosmo et al., 2019] tackles this inverse problem using differentiable eigensolvers, leading to empirical successes recovering surfaces from their Laplacian spectra.

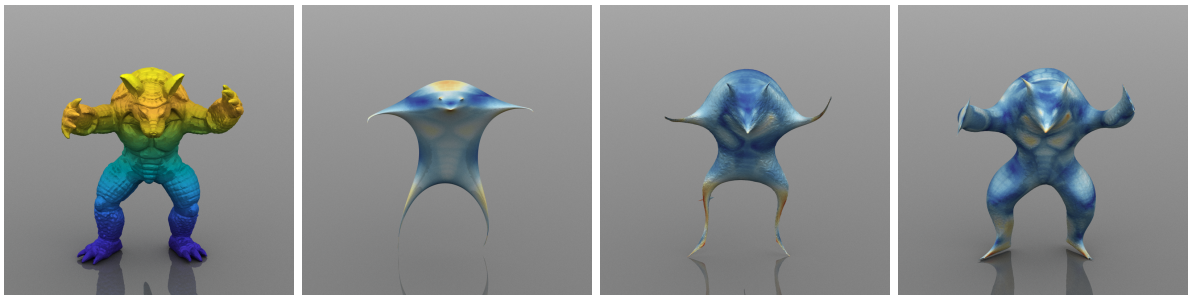
Few results are known for operators other than Laplacian. For the intrinsic Dirac operator, examples of Dirac-isospectral tori are given in [Ammann and Bär, 1998]. For Dirichlet-to-Neumann operator (to the best of our knowledge), whether "hearing the shape" for 3D surfaces is possible is still an open challenge. A recent result indicates that the 2D polyline can be recovered from its Steklov eigenvalues [Levitin et al., 2017].

4 Spectral Shape Analysis and Applications

In this section, we discuss how practical algorithms can benefit from geometric operators and their eigendecompositions. Before we look into particular methods, we first present a few ideas that motivate the use of geometric operators and their eigendecompositions in computational tasks.

4.1 Spectral Analysis: from Euclidean Space to Manifold

A Signal Processing Viewpoint Spectral methods provide bases to represent geometric signals and associated data. Particularly, eigenfunctions $\{\phi_i\}_{i=0}^{\infty}$ of an appropriate operator (see §3.1) generalize the Fourier basis for 1-periodic functions $\mathbb{T}(1) = \{f \in \mathcal{L}^2(\mathbb{R}) : f(x) = f(x+1)\}$, i.e., $\{\sin(2\pi nx)\}_{n=1}^{\infty} \cup \{\cos(2\pi nx)\}_{n=0}^{\infty}$, to a curved manifold $\mathcal{M}$. This provides a basis to represent geometric signals, allowing us to generalize traditional signal processing and image analysis for data associated to points in $\mathbb{R}^n$, to data on a geometric domain. Figure 4 shows an example of reconstructing geometric signals—in particular mesh coordinates—using its Laplacian eigenfunctions. Typically, the Laplacian eigenfunctions are used as the generalization of Fourier bases, but eigenfunctions of any p.s.d. operator $\mathcal{A}$ that form a complete basis can be used as well. Applying this idea to problems in graphics dates back to *curve Fourier descriptors* [Zahn and Roskies, 1972] and *surface signal processing* (via Laplacian) [Taubin, 1995], if not earlier.



(a) Rest Pose. (b) 10 (c) 20 (d) 40

(e) 100 (f) 200 (g) 400 (h) 800

Figure 4: Reconstructing mesh coordinates using increasing numbers of eigenfunctions. The color encodes the error of reconstructed coordinates.

Multi-Resolution and Hierarchical Representation Spectral methods enable multi-resolution analysis on meshes. The multi-resolution approach processes information and features at different scales separately. Inspired by the success of multi-resolution methods in signal processing and image analysis, the importance of multi-resolution analysis on meshes has been recognized since the pioneering work of [Eck et al., 1995, Kobbelt, 1997].

[Guskov et al., 1999] generalize image-based Laplacian pyramids and wavelets to meshes, based on mesh decimation and upsampling procedures. As an alternative, [Mahadevan, 2007] and [Zhong and Qin, 2014] construct wavelets out of Laplacian eigenfunctions. This spectral approach has a natural advantage: It separates geometric signals at increasing frequencies, making it easy to apply multiresolution analysis to geometric data. The multi-resolution consideration is also influential in designing practical algorithms: A recurrent strategy we will see in this section is to apply the multi-scale organization of a spectral basis to obtain a *coarse-to-fine* hierarchy, an idea that appears in hierarchical segmentation [De Goes et al., 2008, Huang et al., 2009] and approximation of multi-scale descriptors [Vaxman et al., 2010].

Spectral Shape Analysis and Processing Motivated by the facts above, geometric operators and their spectral decompositions provide a versatile toolbox for mesh processing, known as *spectral mesh processing*. [Zhang et al., 2010a, Zhang et al., 2010b] survey this topic, focusing on Laplacian-based methods. Although most methods in spectral mesh processing were originally introduced for Laplacian, many remain valid if we simply substitute the Laplacian with an alternative operator, e.g. any of the operators discussed in [Raviv et al., 2011b, Hildebrandt et al., 2012, Aflalo et al., 2013, Liu et al., 2017, Wang et al., 2018b, Ye et al., 2018].

4.2 Spectral Data Analysis

Spectral shape analysis is strongly influenced by developments and advances in spectral data analysis, particularly *spectral graph theory* [Chung and Graham, 1997] and *manifold learning* [Cayton, 2005]. These areas study the geometric structures of graphs and data manifolds, usually sharing similar considerations with problems in analyzing 3D surfaces. As we will see in the section, many spectral or operator-based methods in shape analysis can be thought as adaptions or variants of prior counterparts in these areas; as an example, [Rustamov, 2009] points out an equivalence between the Laplacian mesh editing and semi-supervised learning. For this reason, we highlight relevant methods in spectral data analysis.

Spectral Graph Theory with Applications The analogy between spectral shape analysis and spectral graph theory has been explored since the inception of spectral shape analysis and geometry processing [Gotsman et al., 2003, Gotsman, 2003]. Indeed, many discrete geometric operators can be thought as special instances of the graph Laplacian or adjacency matrix, with geometry-aware entries, i.e., weights determined by geometry of the underlying polygonal mesh like edge lengths or angles. From this view, discrete mesh analysis might be considered a *specialization* of graph analysis.

Earlier work in (spectral) shape analysis uses various versions of graph Laplacians constructed on the mesh graph [Taubin, 1995, Liu and Zhang, 2007], largely inspired by applications of spectral graph theory to image analysis and computer vision [Shi and Malik, 2000]. While graph combinatorial measurements such as the average node degree are of interest for graph analysis, this information is irrelevant to geometry processing, reflecting only e.g. which polygon mesh representation is used or the meshing algorithms generated the data. To alleviate sensitivity to graph combinatorics, graph Laplacians, as well as their spectra [Levy, 2006], were quickly replaced by the cotangent Laplacian, which converges to the smooth Laplace–Beltrami operator and exhibits more robust behavior [Dyer et al., 2007].

Many concepts and algorithms used for mesh processing and analysis are exactly same as those for graphs. As an example, the *Fiedler vector* $\phi_1(\mathbf{x})$ is the eigenvector corresponding to the second-smallest eigenvalue of the graph Laplacian, which has been used for graph partitioning [Fiedler, 1975, Spielman and Teng, 1996]. *Fiedler theory* reveals the number of zero Laplacian eigenvalues equals to the number of connected components in a graph; an identical statement holds for manifolds.

Manifold Learning and Data Science Manifold learning explores geometric structure in clouds of data points. Techniques in this area employ spectral methods for analyzing data manifolds, under the assumption that a dataset is an empirical sample from a low-dimensional manifold on which the data resides. Many of these methods have been adapted to analyze meshes or are incorporated in mesh processing pipelines, as we will see in §4.4.

Two classic examples from the early literature in manifold learning are *IsoMap* [Tenenbaum et al., 2000] and *LLE (locally linear embedding)* [Roweis and Saul, 2000], which use spectral methods to find embeddings of data that preserve global and local distances within a set of data points, respectively. IsoMap uses *multi-dimensional scaling* (MDS) to embed data points, so that Euclidean distances in the embedded space preserve, as much as possible, the geodesic distances from the original manifold. Prior to IsoMap, [Schwartz et al., 1989] propose to flatten a curved surface by minimizing a related distance-based distortion metric, giving a similar solution to what they referred to as the mapmaker's problem. LLE uses *k-nearest neighbors* (kNN) to build a graph over the data points, computes the (row-wise) normalized graph Laplacian $\mathbf{L}$ (which is asymmetric) and its singular value decomposition (SVD) $\mathbf{L} = \mathbf{U}\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{V}^\intercal$, and finally outputs $\mathbf{V}\boldsymbol{\Sigma}^{\frac{1}{2}}$ as the embedding.

Laplacian LLE [Belkin and Niyogi, 2003] is an variant improving LLE by considering the non-normalized graph Laplacian, which is symmetric and whose spectrum provably converges to that of the Laplace–Beltrami operator and applying an eigen-decomposition rather than the SVD; this is identical to how the Laplacian typically is used in shape analysis on meshes. These methods inspire mesh

parameterization methods, as we will see in §4.4. *Hessian LLE* [Donoho and Grimes, 2003] replaces the Dirichlet energy used in Laplacian LLE with the Hessian energy (whose formula is given later in Eq. 32). There are various followups including *local tangent space alignment (LTSA)* [Zhang and Zha, 2004]. LTSA first constructs the approximation of tangent space at each data point, and then aligns all tangent spaces, yielding a global parameterization of the underlying manifold. Later [Rosman et al., 2010] take the topological properties of the embedding into consideration. With spirit similar to LTSA, *spectral affine-kernel embeddings (SAKE)* [Budninskiy et al., 2017] introduces the multi-Laplacian quadratic form $\mathbf{Q}$, which is assembled from local operators $\mathbf{L}_i$ whose null space only contain locally-affine functions around each point i, and replaces the Laplacian matrix previously used in manifold learning with this new matrix $\mathbf{Q}$. Recent *parallel transport unfolding (PTU)* [Budninskiy et al., 2019] replaces the geodesic distances obtained via Dijkstra paths as used in IsoMap with geodesic distances obtained via parallel transport, removing the limitation that Isomap can handle only geodesically convex sampled domains. PTU can generate a quasi-isometric, low-dimensional embedding of a manifold with arbitrary topology.

Diffusion maps [Coifman and Lafon, 2006] draw connections between the row-normalized graph Laplacian $\mathbf{P}$ (defined as follows) and Markov chains. They relate $\mathbf{P}$ to anisotropic diffusion, proving that $\mathbf{P}$ converges in a pointwise fashion to the Laplace–Beltrami operator plus a term depending on the point-wise sampling density, when $t \mapsto 0^+$. In particular, $\mathbf{P}$ is defined by taking

$$\mathbf{W}_{ij} = \exp\left(-\frac{d\left(\mathbf{x}_i, \mathbf{x}_j\right)^2}{t}\right) \quad \text{and} \quad \mathbf{P} = \mathbf{D}^{-1}\mathbf{W},$$

where $d_i = \sum_{j=1}^{n} \mathbf{W}_{ij}$ and $\mathbf{D} = \operatorname{diag}(d_i)$, i.e., $\mathbf{D}$ is a matrix that stacks row sums from $\mathbf{W}$ along the diagonal. After defining the operator $\mathbf{P}$, we can compute the *diffusion embedding*

$$\left(e^{-\lambda_1 t}\phi_1(\mathbf{x}), e^{-\lambda_2 t}\phi_2(\mathbf{x}), \cdots, e^{-\lambda_k t}\phi_k(\mathbf{x})\right) \tag{20}$$

for clustering data points, an embedding designed to map similar points on the manifold also close to each other in the embedded space, while being robust and insensitive to noise.

The $\mathbf{P}$ used above is an example of a *point cloud Laplacian* operator. For applications like diffusion maps, various results on the convergence of different versions of point cloud Laplacians are derived in [Belkin and Niyogi, 2005, Hein et al., 2005, Belkin et al., 2008, Belkin et al., 2009, Liang and Zhao, 2013, Shi and Sun, 2017].

There are related methods in machine learning. Multiple methods [Shi and Malik, 2000, Meila and Shi, 2001, Ng et al., 2002] employ eigenfunctions of a Laplacian or affinity matrix for spectral clustering and segmentation on data points. Harmonic interpolation solves a Laplacian system for label propagation in semi-supervised learning [Zhu et al., 2003]. *Manifold regularization* [Belkin et al., 2006] proposes to add Dirichlet energy as a regularization term to the loss function in classification, yielding Laplacian support vector machines (LapSVM).

4.3 Spectral Analysis: Point Embedding, Signature, and Geometric Descriptors

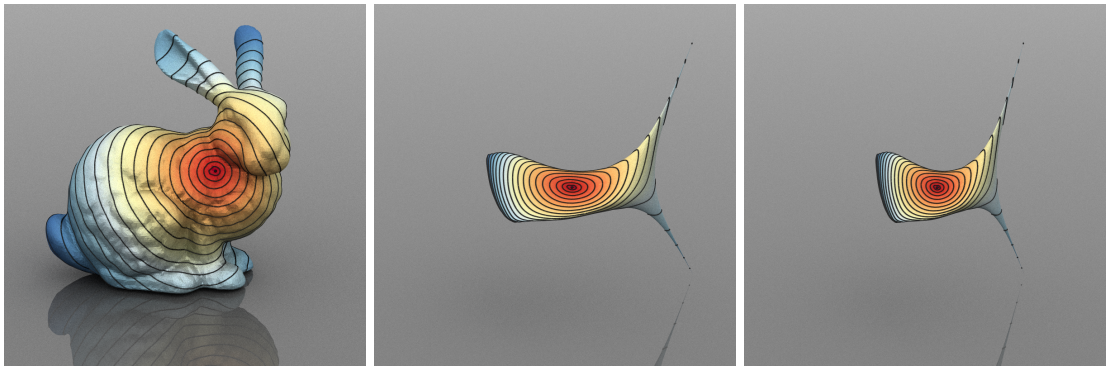
Rather than general clouds of data points, our focus in this survey is spectral analysis of geometric data expressed as a mesh. To this end, in this section we provide a few problems, including segmentation, point signature computation, and distance computation, whose solutions often explicitly involve eigenfunctions of a geometric operator, closely related to the methods we saw in §4.2.

Spectral Embedding and Segmentation Methods for mesh vertex clustering (i.e., segmentation) follow patterns similar to those in clustering data manifolds.

As an example, the *Global Point Signature (GPS)* [Rustamov, 2007] considers the following high-dimensional embedding of points $\mathbf{x}$ on a triangle mesh surface based on the eigenfunctions and eigenvalues of the Laplacian:

$$\left(\frac{\phi_1(\mathbf{x})}{\sqrt{\lambda_1}}, \frac{\phi_2(\mathbf{x})}{\sqrt{\lambda_2}}, \cdots, \frac{\phi_k(\mathbf{x})}{\sqrt{\lambda_k}}\right) \tag{21}$$

This embedding is isometry-invariant, thanks to the use of Laplacian eigenfunctions. [Rustamov, 2007] applies standard k-means algorithms to the embedded vertices, yielding an isometry-invariant segmentation. Figure 5 illustrates the first few dimensions of the diffusion and harmonic embeddings, respectively. Later methods derive analogs of this embedding for different operators, with success for segmentation [Hildebrandt et al., 2012, Liu et al., 2017, Wang et al., 2018b].



(a) (b) (c)

Figure 5: The Stanford bunny model with a log heat kernel $\log k_t(\mathbf{x}, \cdot)$ is shown in (a). The same bunny model but replacing the x, y, z coordinates of a vertex $\mathbf{x}$ with (b) the harmonic embedding $\left(\frac{\phi_1(\mathbf{x})}{\sqrt{\lambda_1}}, \frac{\phi_2(\mathbf{x})}{\sqrt{\lambda_2}}, \frac{\phi_3(\mathbf{x})}{\sqrt{\lambda_3}}\right)$ or (c) the diffusion embedding $\left(e^{-\lambda_1 t}\phi_1(\mathbf{x}), e^{-\lambda_2 t}\phi_2(\mathbf{x}), e^{-\lambda_3 t}\phi_3(\mathbf{x})\right)$.

Spectral Distances Computing the distance between any pair of points on the surface is a fundamental task for shape analysis. While the geodesic distance is a natural choice, it is sensitive to noise and small topology changes; moreover, computing full pairwise geodesic distances is expensive. Spectral distances including diffusion distance [Coifman and Lafon, 2006] and bi-harmonic distance [Lipman et al., 2010] are alternatives with favorable robustness properties.

The diffusion distance $d_D(\cdot, \cdot)$ and bi-harmonic distance $d_B(\cdot, \cdot)$ between points on surface $\mathbf{x}, \mathbf{y} \in \mathcal{M}$ are defined as L^2 distances between the corresponding spectral embeddings:

$$d_D(\mathbf{x}, \mathbf{y})^2 = \sum_{i=1}^{\infty} e^{-2t\lambda_i} \left(\phi_i(\mathbf{x}) - \phi_i(\mathbf{y})\right)^2, \qquad d_B(\mathbf{x}, \mathbf{y})^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i^2} \left(\phi_i(\mathbf{x}) - \phi_i(\mathbf{y})\right)^2.$$

The following equalities for d_D and d_B provide interpretations of these distances: The distance between $\mathbf{x}$ and $\mathbf{y}$ is the integrated difference between the Green's functions or heat kernel functions at $\mathbf{x}$ and $\mathbf{y}$:

$$d_D(\mathbf{x}, \mathbf{y})^2 \equiv k_{2t}(\mathbf{x}, \mathbf{x}) + k_{2t}(\mathbf{y}, \mathbf{y}) - 2k_{2t}(\mathbf{x}, \mathbf{y}) \equiv \int_{\mathcal{M}} \left(k_t(\mathbf{x}, \mathbf{z}) - k_t(\mathbf{y}, \mathbf{z})\right)^2 \mathrm{d}\mathbf{z},$$
$$d_B(\mathbf{x}, \mathbf{y})^2 \equiv G_B(\mathbf{x}, \mathbf{x}) + G_B(\mathbf{y}, \mathbf{y}) - 2G_B(\mathbf{x}, \mathbf{y}) \equiv \int_{\mathcal{M}} \left(G(\mathbf{x}, \mathbf{z}) - G(\mathbf{y}, \mathbf{z})\right)^2 \mathrm{d}\mathbf{z},$$

where $G_B(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{\infty} \frac{1}{\lambda_k^2} \phi_k(\mathbf{x})\phi_k(\mathbf{y})$ is the bi-harmonic Green's function. Note diffusion distance is not a true metric, since $d_D(\mathbf{x}, \mathbf{y}) = 0$ does not necessarily imply $\mathbf{x} = \mathbf{y}$; this issue is addressed by the bi-harmonic distance [Lipman et al., 2010].

The commute-time distance or harmonic distance [Qiu and Hancock, 2007] is popular in graph analysis, due to connections to random walks on graph:

$$d_H(\mathbf{x}, \mathbf{y})^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} \left(\phi_i(\mathbf{x}) - \phi_i(\mathbf{y})\right)^2 \equiv G(\mathbf{x}, \mathbf{x}) + G(\mathbf{y}, \mathbf{y}) - 2G(\mathbf{x}, \mathbf{y}).$$

However, it cannot be defined on continuous surfaces since $G(\mathbf{x}, \mathbf{y})|_{\mathbf{y}=\mathbf{x}}$ is singular.

Point Signatures and Descriptors The goal of shape descriptors or signatures is to compute a vector per point $\mathbf{x}$ on a surface, such that the vector summarizes and encodes local (and sometimes global) geometry near $\mathbf{x}$. After discretization, these descriptors are expressed as per-vertex feature vectors that can be used for higher-level tasks such as shape matching or correspondence:

- The *global point signature* (GPS) [Rustamov, 2007] uses the embedding in Eq. 21 as a signature, which can be ambiguous due to rotations within eigenspaces:

$$\mathrm{GPS}(\mathbf{x}) : \mathcal{M} \to \mathbb{R}^K, \quad \mathrm{GPS}(\mathbf{x}) := \left(\frac{\phi_1(\mathbf{x})}{\sqrt{\lambda_1}}, \frac{\phi_2(\mathbf{x})}{\sqrt{\lambda_2}}, \frac{\phi_3(\mathbf{x})}{\sqrt{\lambda_3}}\right). \tag{22}$$

- The *heat kernel signature (HKS)* [Sun et al., 2009] uses the diagonal of the heat kernel as a signature, summarizing the geometry near a point $\mathbf{x}$:

$$\text{HKS}(\mathbf{x}, t) : \mathcal{M} \times \mathbb{R}^+ \to \mathbb{R}, \quad \text{HKS}(\mathbf{x}, t) := k_t(\mathbf{x}, \mathbf{x}) \equiv \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(\mathbf{x})^2. \tag{23}$$

  The HKS is not subject to the rotation ambiguity of the global point signature (GPS). Additionally, an "informative property" shows that for any $\mathbf{x}$, $\text{HKS}(\mathbf{x}, \cdot)$ determines $\mathcal{M}$ up to isometry. The HKS has been extended using operators beyond the Laplacian, leading to the affine-invariant HKS [Raviv et al., 2011a], the modified Dirichlet HKS [Hildebrandt et al., 2012], the Steklov HKS [Wang et al., 2018b], and the Dirac HKS [Liu et al., 2017]. Figure 6 illustrates a typical heat kernel signature with a fixed small time, computed using the Steklov eigenfunction, combining [Sun et al., 2009] and [Wang et al., 2018b]: we see the signature roughly aligns with the local curvature.

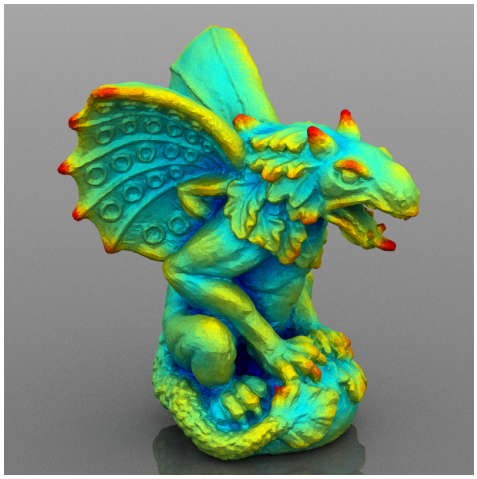


Figure 6: The heat kernel signature computed using the Steklov eigenfunctions.

  Developed in parallel to the HKS, the *auto diffusion function (ADF)* [Gebal et al., 2009] proposes to consider the time-varying scalar field $\text{ADF}_t(\mathbf{x}) := k_{t/\lambda_1}(\mathbf{x}, \mathbf{x})$, which is a time scaled version of HKS, for segmentation and skeletonization.

- The *scale-invariant heat kernel signature (SI-HKS)* [Bronstein and Kokkinos, 2010] modifies the HKS to factor out the effect of scale using operations in the spatial and frequency domains.

- The *wave kernel signature (WKS)* [Aubry et al., 2011] is derived from the Schrödinger equation, Eq. 14, whose solutions can be written in the form $u(\mathbf{x}, t) = \sum_{k=0}^{\infty} e^{i\lambda_k t} \phi_k(\mathbf{x}) f_E(\lambda_k)$. In particular, the WKS is defined as follows:

$$\text{WKS}(\mathbf{x}, E) : \mathcal{M} \times \mathbb{R}^+ \to \mathbb{R}, \quad \text{WKS}(\mathbf{x}, E) := \lim_{T \to \infty} \int_0^T \|u(\mathbf{x}, t)\|^2 \equiv \sum_{k=0}^{\infty} f_E^2(\lambda_k) \phi_k(\mathbf{x})^2,$$

  where $f_E^2(\cdot)$ is chosen as a log-normal distribution function with parameter E:

$$f_E^2(x) = \left[\sum_k e^{\frac{-(\log E - \log \lambda_k)^2}{2\sigma^2}} \right]^{-1} e^{\frac{-(\log E - \log x)^2}{2\sigma^2}}.$$

  The WKS can be interpreted as the average (over time) probability density at $\mathbf{x}$. Compared to the HKS, the WKS is more discriminative to geometric features and information at small scales, usually leading to more accurate shape matching.

- From a signal processing perspective, the HKS and WKS apply different one-parameter filter functions (parameterized by t or E) to λ, i.e., $e^{-\lambda t}$ and $f_E^2(\lambda)$, respectively. *Optimal spectral descriptors* [Litman and Bronstein, 2014] learn this filter function rather than using a pre-designed function.

- The *discrete-Time Evolution Process (DEP)* descriptor [Melzi et al., 2018a] considers a discrete-time process rather than a continuous one as in HKS and WKS. The discrete-time process is fully-specified by a generic pairwise relation function $A_\delta(\mathbf{x}, \mathbf{y})$ depending on a scale parameter δ, similar to the heat kernel $k_t(\mathbf{x}, \mathbf{y})$ which depends on a time parameter t in HKS. Given an initial function $f^{(0)}(\cdot)$, this iterative evolution is very similar to the continuous diffusion in Eq. 12, with time t replaced with discrete step index $l \in \mathbb{Z}$:

$$f_\delta^{(l)}(\mathbf{x}) = \int_{\mathcal{M}} A_\delta(\mathbf{x}, \mathbf{y}) f_\delta^{(l-1)}(\mathbf{y}) \, \mathrm{d}\mathbf{y}.$$

  The DEP point signature is defined as follows:

$$\mathrm{DEP}(\mathbf{x}, \{1, 2, ..., K\}) : \mathcal{M} \times \{1, 2, ..., K\} \to \mathbb{R}, \quad \mathrm{DEP}(\mathbf{x}, k) := s_{\delta_k}(\mathbf{x}).$$

  in which the score function $s_\delta(\mathbf{x})$ is defined as

$$s_\delta(\mathbf{x}) := \sum_{l=1}^{\infty} r_\delta^l f_\delta^{(l)}(\mathbf{x}),$$

  where r_δ is a constant that depends on δ.

- The *intrinsic shape context (ISC)* descriptor [Kokkinos et al., 2012] builds a local geodesic chart system for each $\mathbf{x} \in \mathcal{M}$ and uses the average of some per-vertex descriptor $I(\mathbf{x})$ over intrinsic angular bins and radial bins within the chart. This corresponds to diffusion in the chart; Geodesic CNNs (GCNN) [Masci et al., 2015] apply similar operations in learnable settings.

4.4 Shape Analysis and Geometry Processing

That operators and their spectra encode geometric information has been explored by a variety of applications in geometry processing, shape analysis, and computer graphics. Early work utilizing geometric operators include [Reuter et al., 2005], which proposes to use eigenvalues of the continuous Laplace–Beltrami operator as a shape representation, and [Levy, 2006] which considers the Laplacian eigenfunctions as a means to understand geometry, with applications to surface registration, segmentation and parameterization. Since then a large number of methods—mainly based on the Laplacian—has been developed.

Geometric operators encode geometric information, from which topological properties can also be inferred. We first look at segmentation, skeletonization, and quadrangulation, which leverage the fact that topological structures can be extracted from level sets of functions derived from eigenfunctions and eigenvalues. In particular, the zero level-set of an eigenfunction is called the *nodal set*, which appears in many methods. Geometric operators also encode information such as symmetry. Particularly symmetric patterns in eigenfunctions and derived descriptors reveal symmetry in the shape. [Ovsjanikov et al., 2008] embeds shape vertices to their point-wise spectral signature and reduces the problem of intrinsic symmetry detection to determining Euclidean symmetries in the signature space. [Thomas and Natarajan, 2014] proposes a multi-scale symmetry detection method by examining contours of level sets of first few eigenfunctions.

Segmentation [Liu and Zhang, 2004] apply spectral clustering methods from other fields to mesh segmentation; their method applies the normalized cuts [Shi and Malik, 2000] to a point cloud Laplacian. Extending this work, [Liu and Zhang, 2007] embed meshes to 2D using the first two nonzero eigenfunctions of Laplacian and a related geometric operator, i.e., mapping $\mathbf{x} \in \mathcal{M} \to (\phi_1(\mathbf{x}), \phi_2(\mathbf{x})) \in \mathbb{R}^2$, and apply contour analysis to the projected planar shape for segmentation. [De Goes et al., 2008] use diffusion distances to build a coarse-to-fine hierarchy of segments for articulated bodies. [Huang et al., 2009] use eigenvectors of the Hessian of a nonlinear deformation energy instead of Laplacian, for hierarchically decomposing deformable shapes. Finally, [Reuter et al., 2009] use the Laplacian nodal sets for shape segmentation.

Related, *spectral surface reconstruction* [Kolluri et al., 2004] applies a variant of spectral graph partitioning to a Delaunay tetrahedralization of a point cloud, to identify inside and outside tetrahedra, and outputs triangular faces adjacent to both inside and outside tetrahedra.

Skeletonization (and Segmentation) Skeletonization might be considered as a "dual" problem to segmentation. As an example of this duality, [Shi et al., 2008] consider the Fiedler vector of an anisotropic Laplacian, i.e. the scalar function $\phi_1(\mathbf{x})$, and constructs its *Reeb graph*, intuitively a graph whose edges pass orthogonally to the isolines of $\phi_1(\mathbf{x})$ and whose vertices correspond to regions where the isolines have a particular connectivity; in this method, the vertices of the Reeb graph roughly correspond to segments on the surface determined by the Fiedler vector. Similarly, [Patane et al., 2008] identify critical points (saddle and extreme points) of Laplacian eigenfunctions, applies isoline analysis to these points, and connects them to extract a Reeb graph, from which shape segmentations and skeletonizations can be then obtained.

Nodal sets of the Fiedler vector can be unstable. To address this issue, the Auto Diffusion Function [Gebal et al., 2009] uses diffusion to define a time-varying scalar field $\mathrm{ADF}_t(\mathbf{x}) := k_{t/\lambda_1}(\mathbf{x},\mathbf{x})$ (see §4.3), evaluates $\mathrm{ADF}_t(\mathbf{x})$ at multiple t values, extracts Reeb graphs for each of them, and finally outputs a skeleton combining these Reeb graphs. More broadly, the Reeb graph extracts a topological skeleton for $\mathcal{M}$, and hence can be used as a (graph-valued) shape descriptor.

Parameterization and Remeshing Laplacian eigenfunctions demonstrate patterns that can be exploited and extracted for surface parameterization and quadrangulation. The most famous relevant result, *Courant's theorem* states that the number of partitions by the nodal set for the k-th Laplacian eigenfunction $\phi_k(\cdot)$ is less or equal to k [Courant, 1923].

The *Morse–Smale complex* associated to a real-valued function $f\colon \mathcal{M}\to\mathbb{R}$ is a cellular decomposition of $\mathcal{M}$ into quadrangular patches. The edges of the complex form a graph on $\mathcal{M}$ whose vertices are critical points of f and whose edges roughly align with directions of the gradient $\nabla_{\mathcal{M}} f$. Laplacian eigenfunctions enjoy several properties that make them strong candidates as the generating function f. In particular, their minima and maxima appear alternately, and critical points are evenly distributed with respect to geodesic distance.

Spectral surface quadrangulation [Dong et al., 2006] generates an initial Morse–Smale complex from $\phi_k(\mathbf{x})$, where k is the expected number of nodal sets; in practice, k roughly equals the eigenfunction index thanks to Courant's Theorem above (usually $k<100$). Computation of the complex is followed by topological noise removal, patch boundary adjustment, and refinement. This yields a smooth parameterization that can be used to generate a quadrilateral mesh. Examples of this technique and comparison to other quad meshing algorithms are provided in the survey [Bommes et al., 2013].

Spherical parameterization is the task of mapping genus-0 meshes to the unit sphere. [Gotsman et al., 2003] tackle this problem by optimizing for a graph Laplacian matrix on the mesh graph such that its spectral embedding $\mathbf{x}\in\mathcal{M}\to(\phi_0(\mathbf{x}),\phi_1(\mathbf{x}),\phi_2(\mathbf{x}))\in\mathbb{S}^3$ becomes a valid spherical parameterization.

Iso-charts [Zhou et al., 2004] apply an IsoMap-like procedure for mesh parameterization and spectral clustering, as basic steps in a pipeline to decompose a mesh into low-stretch atlases/charts. Based on the key observation that geodesic distance distortion is closely related to stretch distortion induced by parameterization, IsoMap is used to generate the initial parameterization. A similar idea appears in the earlier work [Zigelman et al., 2002], which applies MDS to preserve the surface geodesic distances and uses the first two dimensions of the optimal MDS embedding as a surface parameterization, leading to a texture map with low distance distortion.

The spectral embedding $(\mathbf{u},\mathbf{v}) = (\phi_1(\mathbf{x}),\phi_2(\mathbf{x}))$ can be used as a parameterization with fairly low distortion [Liu and Zhang, 2007]. *Spectral conformal parameterization* [Mullen et al., 2008] uses a generalized Fiedler vector of the *conformal energy* matrix $\mathbf{C}$, defined via

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}^{\mathsf{T}}\mathbf{C}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix} := \mathbf{u}^{\mathsf{T}}\mathbf{L}\mathbf{u}+\mathbf{v}^{\mathsf{T}}\mathbf{L}\mathbf{v}-\frac{1}{2}\sum_{e_{ij}\in\partial\mathcal{M}}(\mathbf{u}_i\mathbf{v}_j-\mathbf{u}_j\mathbf{v}_i),$$

where $\mathbf{L}$ is the cotangent Laplacian and e_{ij} is an edge along the boundary. The third term in this energy equals the total area of the parameterization. An earlier method coined *intrinsic parameterization* [Desbrun et al., 2002] first proposes this energy and solves a related linear system.

Relatedly, eigenfunctions have been used for frame field design, tangential vector and n-vector field processing on surfaces [Azencot et al., 2015, Azencot et al., 2017, Brandt et al., 2017, Brandt et al., 2018].

Data Compression Since the ϕ_i's generalize the Fourier basis, it is straightforward to use $\{\phi_i\}_{i=0}^k$ to compress geometric data and signals. Spectral compression of mesh geometry [Karni and Gotsman, 2000] projects vertex (x,y,z) coordinate functions onto $\{\phi_i(\cdot)\}_{i=0}^k$. [Ben-Chen and Gotsman, 2005] present theoretical results roughly showing that spectral compression using the Laplacian basis is optimal in the

mean-square error sense, for a random mesh whose vertex positions are drawn from certain probability distributions. [Aflalo et al., 2015] prove the optimality of the Laplacian eigenfunctions for approximating functions with bounded variation on surfaces.

Mesh Smoothing and Fairing As an example of generalizing signal processing to meshes, the *spectral filtering* methods proposed in [Vallet and Lévy, 2008] apply low-pass, high-pass, and band-exaggeration filters to eigenvalues, leading to effects of mesh smoothing, details sharpening, and band-feature enhancement, respectively. Relatedly, locally amplifying surface area according to its Gaussian curvature produces natural exaggeration effects [Sela et al., 2015].

Feature Extraction Descriptors like heat and wave kernel signatures in §4.3 provide powerful tools to find keypoints that can be candidates for landmark correspondences. Similarly, spectral information can be used to detect salient points and regions on a surface. For example, spectral mesh saliency detection [Song et al., 2014] generalizes image spectral saliency detection [Hou and Zhang, 2007] to meshes: Simply speaking, instead of applying a filter function to eigenvalues as in spectral filtering, the method essentially applies a filter to the log of eigenvalues, to amplify and emphasize the low-frequency part of the spectrum.

Shape Matching and Correspondence [Ovsjanikov et al., 2010] apply the HKS to match feature points for the correspondence problem. The influential framework of *functional maps* [Ovsjanikov et al., 2012] directly computes a map F between eigenfunctions on two domains $\mathcal{M}$ and $\mathcal{N}$, i.e., $F : \mathcal{L}^2(\mathcal{M}) \to \mathcal{L}^2(\mathcal{N})$. Many methods have been proposed to improve this framework and extend it to alternative applications, such as the partial shape correspondence [Rodolà et al., 2017]. We refer to the recent course notes [Ovsjanikov et al., 2016] for an extensive survey.

Shape Differences *Shape differences* provide a framework to compare two shapes that are put into correspondence using a functional map [Rustamov et al., 2013]. Given two geometric domains $\mathcal{M}$ and $\mathcal{N}$ as well as a functional map $F : \mathcal{L}^2(\mathcal{M}) \to \mathcal{L}^2(\mathcal{N})$, the shape difference operator $D : \mathcal{L}^2(\mathcal{M}) \to \mathcal{L}^2(\mathcal{M})$ measures the pointwise distortion induced by F on $\mathcal{M}$. In particular, D modifies any function $f \in \mathcal{L}^2(\mathcal{M})$ linearly, i.e. the modification yields $Df \in \mathcal{L}^2(\mathcal{M})$, such that the bilinear form is best preserved under the map F, in the sense that $\forall g \in \mathcal{L}^2(\mathcal{M}) : |a_{\mathcal{M}}(g, Df) - a_{\mathcal{N}}(Fg, Ff)|$ is minimized. Here $a(\cdot,\cdot)$ is a positive semi-definite bilinear form (i.e., inner product), which can be the Dirichlet energy (Laplacian), the area form (identity operator) [Rustamov et al., 2013], or the DtN/Steklov operator in weak form [Wang et al., 2018b]. The resulting operator $D = \mathcal{A}_{\mathcal{M}}^{\dagger} F^{\intercal} \mathcal{A}_{\mathcal{N}} F$ reflects how the local surface geometry of $\mathcal{M}$ and $\mathcal{N}$ differ under map F, where $\mathcal{A}_{\mathcal{M}}, \mathcal{A}_{\mathcal{N}}$ is the operator $\mathcal{A}$ operating on the domain $\mathcal{M}, \mathcal{N}$, respectively, operator $\mathcal{A}$ is the strong form that corresponds to the bilinear form $a(\cdot,\cdot)$ with $a(u, v) = \langle u, \mathcal{A}v \rangle_{\mathcal{M}}$ for $u, v \in \mathcal{L}^2(\mathcal{M})$, and the superscript $\cdot^{\dagger}$ denotes the pseudo-inverse of an operator. Spectral basis reduction can be applied to $\mathcal{A}_{\cdot}$ and F, decreasing the computational cost.

The shape difference framework provides methods for shape analogy tasks and shape dataset exploration. Combined with a shape-from-Laplacian routine like those proposed in [Boscaini et al., 2015, Corman et al., 2017a, Chern et al., 2018], it leads to applications including style and deformation transfer, and computation of an "intrinsic average" of a few shapes by "averaging" their Laplacian eigenspaces. Recent work [Huang et al., 2019] proposes a learning-based framework to reconstruct surfaces from shape difference operators.

Shape Retrieval [Reuter et al., 2006] advocate using the vector of Laplacian eigenvalues as "ShapeDNA," an ID for shape retrieval. In "Shape Google" [Bronstein et al., 2011], descriptors including heat kernel signatures and scale-invariant heat kernel signatures are collectively used to construct a "bag of features" whose distributions are used to represent a shape. [Lian et al., 2013] provide a comprehensive comparison of shape retrieval algorithms including many spectral methods.

Self functional maps [Halimi and Kimmel, 2018] provide an alternative shape representation/signature that can be used for retrieving shapes. Conceptually similar to the construction of shape difference operators, this framework studies the interaction between two different metric spaces on the same domain. It uses functional maps between spaces spanned by the regular and the scale-invariant Laplacian eigenfunctions (see §5.5) as the shape signature.

Geometric (Deep) Learning Geometric operators have also been applied to geometric deep learning [Bronstein et al., 2017], in architectures based on convolutional neural networks (CNNs). *Geodesic CNN (GCNN)* [Masci et al., 2015] and *anisotropic CNN (ACNN)* [Boscaini et al., 2015] consider diffusion and anisotropic diffusion operators as alternative models to planar 2D convolutions on a curved surface. *Surface networks* [Kostrikov et al., 2018] use the Laplacian and Dirac operators in lieu of the graph adjacency matrix used in Graph Neural Networks [Scarselli et al., 2009]. The recent work [Wang et al., 2019] further proposes a framework for learning a family of discretized operators on meshes, including the Laplacian and Dirac operators.

Instead of applying operators in spatial domain, *Spectral graph neural networks* [Bruna et al., 2014, Defferrard et al., 2016] apply the graph Laplacian operator in the spectral domain. *Synchronized spectral CNN* [Yi et al., 2016] applies and extends this approach to 3D meshes, with a setup similar to functional maps. *Deep functional maps* [Litany et al., 2017] learn to refine the point-wise SHOT descriptor [Tombari et al., 2010] to be used in a differentiable functional map layer, minimizing the geodesic error from the ground-truth correspondence. Later unsupervised versions [Halimi et al., 2019, Roufosse and Ovsjanikov, 2018] remove the need of the ground-truth: [Halimi et al., 2019] employ a geodesic distortion loss, and [Roufosse and Ovsjanikov, 2018] define an unsupervised loss function in spectral domain, combined with the functional map framework.

While the deep learning methods above have gained recent popularity, the application of spectral data to learning from geometry emerged prior to geometric deep learning. For example, [Aflalo et al., 2011] learn a diffusion kernel for shape retrieval, and *optimal spectral descriptors* [Litman and Bronstein, 2014] learn spectral filter coefficients in a signal processing fashion.

Structure and Vibration Analysis In mechanics, the eigenfunctions of the stiffness matrix (Hessian) of some nonlinear energy—which become the Laplacian eigenfunctions for a certain energy—correspond to principal vibration modes to which the shape resonates most, under external forces and perturbations without damping. The corresponding eigenvalues are known as the intrinsic frequencies for the shape. Beyond the Laplacian, the bi-harmonic equation, having been considered in geometric modelling [Botsch and Kobbelt, 2004], is another common linearized elastic model, whose eigenfunctions represent modes of vibration of a thin elastic plate.

Acoustics The eigenvalues of the stiffness matrix correspond to the characteristic frequencies of the sound made by the shape, appearing in a large number of works in sound simulation [van de Doel and Pai, 1996, van de Doel and Pai, 1998, O'Brien et al., 2002, Chadwick et al., 2009, Ren et al., 2013, Langlois et al., 2014]. This is the forward problem of computing eigenvalues from the shape. Next we talk about its inverse problem, that is, designing a shape making sound of certain frequencies.

Shape Optimization Shape optimization searches the optimal shape that minimizes a given objective functional—which frequently involve eigenvalues and eigenfunctions—from a class of shapes. In the inverse acoustic problem of instrument design, eigenvalues are directly used as part of an objective function for shape optimization [Yoo et al., 2006, Yu et al., 2010, Bharaj et al., 2015]. In shape analysis, [Cosmo et al., 2019] propose *isospectralization*, optimizing and deforming shapes to match a given spectrum, as a prepossessing step for better correspondence. See [Henrot, 2017] for recent theoretical progress in spectral geometry with connections to shape optimization.

Reduced Simulation Constraining shape deformation to be within the span of top eigenfunctions of some operator reduces the degrees of freedom for physical simulation significantly, achieving interactive speeds for simulating deformations of large meshes. Using eigenfunctions of the stiffness matrix associated to a physical system is known as *modal analysis* [Pentland and Williams, 1989, Hauser et al., 2003, Barbič and James, 2005, Hildebrandt et al., 2011]; Laplacian eigenfunctions are often used for reduced deformation, when the deformation energy is not specified [Rong et al., 2008]. The scalar- and vector-valued Laplacian eigenfunctions are also used for reduced fluid simulation [Liu et al., 2015].

4.5 Other Aspects of Spectral Shape Analysis

Localized Bases Laplacian eigenfunctions, as the generalization of the Fourier basis to manifolds, are globally supported, which can be a limitation in many applications. Reduced simulation is an example: deformation, restricted as a linear combination of eigenfunctions, must be global, leading to counter-intuitive visual effects. Hence, some works promote locality and sparsity in a spectral-type basis. For

instance, *Compressed manifold harmonics (CMH)*, originating in scientific computing [Ozoliņš et al., 2013], are introduced to graphics in [Neumann et al., 2014]. These methods add an additional term, the L^1 norm on eigenfunctions $\phi_i(\cdot)$, to the objective in the original eigenvalue problem (see Eq. 7), to promote sparsity in the solution. Relatedly, [Bronstein et al., 2016] study how to appropriately and consistently discretize the L^1-norm on meshes. *Localized manifold harmonics (LMH)* [Melzi et al., 2018b] propose to modify the standard Laplacian, yielding a new operator whose eigendecomposition leads to localized orthogonal bases. *Multiscale diffusion wavelets* [Mahadevan, 2007] construct hierarchical wavelet trees using Laplacian eigenfunctions.

Bases for Shape Collections When multiple poses of a shape are available, methods like the *singular value decomposition (SVD)* or *proper orthogonal decomposition (POD)* provide modes that are frequently used for modal analysis in engineering sciences, see e.g. [Kunisch and Volkwein, 2002]. Data driven methods, like those used in tensor factorization on shape frames, animation compression [Alexa and Müller, 2000], and inverse skinning [Kry et al., 2002], also involve similar techniques; see the recent survey [Jacobson et al., 2014] for an extensive review. Relatedly, given a collection of 3D shapes, *fuzzy correspondence* [Kim et al., 2012] proposes a framework for robustly computing point correspondence. It first constructs an approximate shape-wise correspondence matrix $\mathbf{C} \in \mathbb{R}^{ns \times ns}$, where s is the number of shapes and n the number of vertices in each shape, and then applies spectral analysis to $\mathbf{C}$. The similarity between two vertices is defined as the distance between their spectral (diffusion) embedding of $\mathbf{C}$.

When there are multiple non-isometric shapes, Laplacian eigenfunctions are not compatible across shapes. To address this issue, *coupled quasi-harmonic bases* [Kovnatsky et al., 2013] study the joint diagonalization problem of Laplacians computed on shape collections. *Laplace–Beltrami basis pursuit* [Schonsheck et al., 2018] aligns the eigenfunctions between two shapes, with applications to non-isometric shape registration. It optimizes for a conformal deformation of the source shape, such that the Laplacian eigenfunctions of the deformed shape match those on the target shape.

4.6 Numerical Aspects

Most of the applications above rely on matrix approximations of an operator designed to capture quantities on a smooth surface. Machinery from numerical analysis is needed to justify the discrete approximations of smooth quantities. Algorithms that are well-founded numerically tend to have stronger invariance to remeshing and other common confounding factors in geometry processing.

Convergence When $\mathcal{M}$ is a flat domain in $\mathbb{R}^n$, the mathematical theory of FEM establishes convergence of the linear problem $\mathbf{Au} = \mathbf{Mf}$, showing that $\mathbf{u}$ converges to $u(\cdot)$ under suitable assumptions [Brenner and Scott, 2007]. Similarly, the continuous eigenvalue problem is approximated by the generalized eigenvalue problem of the discretized matrix system

$$\mathbf{A}\boldsymbol{\phi} = \lambda \mathbf{M}\boldsymbol{\phi} \quad \text{s.t. } \boldsymbol{\phi}^\intercal \mathbf{M}\boldsymbol{\phi} = \mathbf{I}. \tag{24}$$

The discrete spectrum $\{\lambda_i, \boldsymbol{\phi}_i\}_{i=0}^k$ usually converges to the continuous one, proved on a case-by-case basis using similar techniques [Sun and Zhou, 2016].

Surprisingly, for a curved surface $\mathcal{M}$, convergence analysis of linear problems involving the Laplacian only recently has been established [Holst, 2001, Dziuk and Elliott, 2013]. A rigorous analysis of FEM on manifolds inevitably requires theorems and techniques involving geometric analysis on curved surfaces, machinery that is not necessary for FEM on flat domains.

Eigensolvers Most methods like FEM yield sparse linear systems whose eigenvalue problems needs to be solved. Recall that linear systems can be solved using either direct or iterative algorithms; [Botsch et al., 2005] include pointers to many sparse linear solvers and as well as the corresponding preconditioners for systems arising in mesh processing. Similarly, for the eigenvalue problem, direct or iterative algorithms can be employed. Direct solvers output all eigenvectors and eigenvalues, while iterative solvers return the first few ones. In most applications, only the first few eigenfunctions/eigenvalues are needed, in which case iterative solvers are favored.

Iterative eigensolvers, including shifted power method, orthogonal projection methods, and Krylov subspace methods (e.g. Arnoldi method and Lanczos algorithm), further combined with filtering and restarting techniques and the use of a preconditioner, have a large number of variants [Saad, 2011]. The

state-of-the-art methods, including the implicitly restarted Arnoldi method, are widely available in sparse linear algebra packages including ARPACK [Lehoucq et al., 1998]. Some methods, particularly BEM for integral-based operators, lead to dense linear systems. They can be solved by iterative algorithms like LOBPCG (locally optimal block preconditioned conjugate gradient) [Knyazev, 2001], which only asks for an oracle that implements the matrix-vector product, with implementations widely available.

Accelerations and Preconditioners Iterative eigensolvers can be accelerated using *spectral shift* and *preconditioning*. In computer graphics, since [Dong et al., 2006] a shift eigenvalue solver is frequently used, with a properly chosen value of spectral shift σ. σ prescribes the maximum frequency (eigenvalue) of interest and separates the eigenvectors out of the range $[0, \sigma)$, efficiently accelerating the convergence.

The use of a preconditioner can greatly accelerate the convergence of an iterative linear or eigen-solver. Preconditioners proposed first for linear problems can be used for eigensystems as well, including those based on the incomplete Cholesky/LU factorization, the multi-grid methods [Golub and Van Loan, 2012], or more general black-box algebraic multi-grid methods. Inspired by the breakthroughs in theoretical computer science of linear time graph preconditioners [Spielman and Teng, 2004], in [Krishnan et al., 2013] a multi-level preconditioning scheme is efficiently implemented for the Laplacian, with linear time and memory complexity, successful applied to mesh and image editing.

For for dense BEM systems, efficient approximations can be obtained by employing the *hierarchical matrix* [Börm et al., 2003] or *fast multipole* [Coifman et al., 1993] methods.

Multiresolution, Subsampling, and Approximation Methods There are multiple methods to approximate eigenfunctions. Since low-frequency eigenfunctions are stable to sampling rate, one approach to accelerate computations is to reduce the mesh size. This can be done by either explicitly maintaining a smaller downsampled mesh, or implicitly using algebric prolongation and restriction operators like ones in multigrid method [Brenner and Scott, 2007].

The Nyström subsampling method [Williams and Seeger, 2001, Drineas and Mahoney, 2005] samples columns from the operator matrix. [Liu et al., 2006] apply the Nyström subsampling method to efficiently approximate eigenvectors; farthest point sampling is recommended over random point sampling for better accuracy. Multiresolution techniques are used in [Dong et al., 2006] and [Wang et al., 2018b] for Laplacian and Steklov eigenvalue problems, respectively. [Wang et al., 2018a] apply low-rank estimation through the Nyström method to the kernel matrix in kernelized functional maps. [Vaxman et al., 2010] introduce a multiscale approach to evaluate heat kernels hierarchically.

The recent work [Nasikun et al., 2018] proposes a fast approximation scheme for the Laplacian eigenfunctions. It first constructs a subspace, and then solves the eigenvalue problem in variational form (Eq. 7) but restricting the solution within the subspace. The subspace, whose construction is inspired by skinning weights for shape deformation, is fast to compute and capable to well approximate low-frequency functions.

Robustness [Dyer et al., 2007] empirically study the robustness of the Laplacian eigenvalues when evaluated on different discretizations of the same object, showing superiority of the mesh Laplacian over the graph Laplacian. Exploring the fact that lowest eigenvalues are robust to sampling, the recent *spectral coarsening* [Liu et al., 2019] algorithm constructs and optimizes a sparse operator matrix, of a much smaller size than the mesh size, whose lowest eigenvalues well preserve those of the original geometric operator. It first applies a combinatorial coarsening procedure to select key nodes, and then uses the nodes to construct a graph Laplacian. Entries of this Laplacian are determined by minimizing the error of multiplying the Laplacian to the ground truth top k eigenfunctions.

Padé Approximation of Heat Kernel and Distance For some problems, it is possible to avoid computing the sequence $\{\phi_i, \lambda_i\}_{i=0}^k$ altogether. As an example, we have seen that the matrix exponential $e^{-t\mathbf{L}}$ and reciprocal $1/\mathbf{L}^p$ are frequently used in operator-based geometry processing, and it suffices to approximate these matrix functions directly. To this end, [Patané, 2015, Patané, 2017] explores the idea of using the Padé approximant of matrix exponentials and reciprocals. Generically speaking, the Padé approximant of a function is a ratio of two polynomials $P(\cdot), Q(\cdot)$ whose coefficients can be derived using a Taylor expansion. For instance, the following expression gives the Padé approximation of the matrix exponential:

$$e^{-t\mathbf{L}} := \sum_{i=0}^{\infty} \frac{(-t\mathbf{L})^i}{i!} \approx \frac{P(\mathbf{L})}{Q(\mathbf{L})} = c_0\mathbf{I} + \sum_{i=1}^{k} \frac{c_i}{\mathbf{L} + s_i\mathbf{I}}$$

we have the last equality, as the result of algebraic factoring of the polynomials. Applying this operator reduces to solving a few linear systems rather than solving an eigenvalue problem. [Patanè, 2017] reviews Laplacian kernels and distances including numerical aspects of fast approximation.

5 Relevant Geometric Operators

In this section, we will discuss choices of the geometric operator $\mathcal{A}$ to perform spectral shape analysis. Relevant geometric operators roughly can be categorized as coming from two approaches. The first approach considers a shape as a thin shell specified by the surface; all intrinsic and some extrinsic methods fall in this category. The second approach considers a shape as a 3D solid whose boundary is specified by the surface. In this section, we first discuss methods using the first approach, and then those using the second approach.

5.1 Identity Operator, Area Form, and Mass Matrix

When discretizing a general linear operator $\mathcal{A}: \mathcal{L}^2(\mathcal{M}) \to \mathcal{L}^2(\mathcal{M})$ or equivalently the bilinear form $a(\cdot,\cdot): \mathcal{L}^2(\mathcal{M}) \times \mathcal{L}^2(\mathcal{M}) \to \mathbb{R}$, FEM outputs the final discrete operator in the form $\mathbf{M}^{-1}\mathbf{A}$, where $\mathbf{M}$ is the mass matrix.

As an example, when $\mathcal{A}$ is the *identity operator*, the corresponding bilinear form is $a(u,v) = \int_{\mathcal{M}} uv$. If we let $u, v = 1_{\mathcal{R}}$, i.e., the constant function supported on region $\mathcal{R} \in \mathcal{M}$, then $a(u,v)$ evaluates to the area of $\mathcal{R}$. For this reason, $a(u,v) = \int_{\mathcal{M}} uv$ is also frequently referred to as the *area form*. For this simple identity operator, we have $\mathbf{A} = \mathbf{M}$, yielding the discrete operator $\mathbf{M}^{-1}\mathbf{A} = \mathbf{I}$; in other words, the mass matrix $\mathbf{M}$ is the discrete weak form of the identity operator. $\mathbf{M}$ is also a discrete area form, in that $\mathbf{1}^\intercal\mathbf{M}\mathbf{1}$ evaluates to the total area of a triangle mesh.

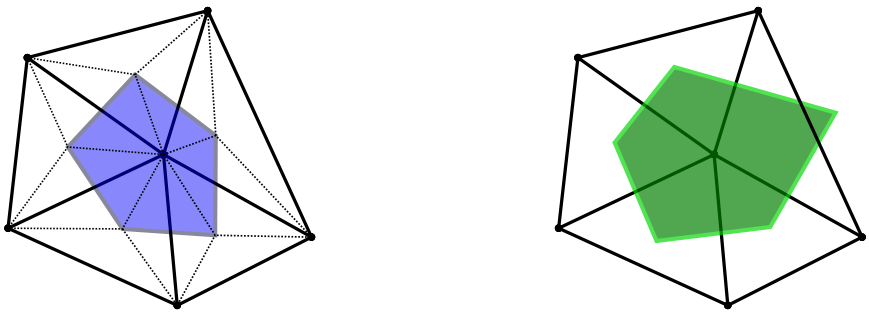
Despite its simplicity, the area form already reveals interesting information about the shape. The shape difference framework [Rustamov et al., 2013] introduces an area-based shape difference, which measures the area distortion between two surfaces under a functional map using measurements derived from the area form.

Discretization: Full Mass Matrix Following exactly the FEM, we have $\mathbf{M}_{ij} = \langle \phi_i, \phi_j \rangle_{\mathcal{M}}$. Using a piecewise linear basis function on a triangle mesh, $\mathbf{M}$ evaluates to the following expression:

$$\mathbf{M}_{ij} = \langle \phi_i, \phi_j \rangle_{\mathcal{M}} = \sum_{T \in \mathcal{T}(i) \cap \mathcal{T}(j)} \int_{\mathcal{M}} \phi_i \phi_j = \sum_{T \in \mathcal{T}(i) \cap \mathcal{T}(j)} \begin{cases} \frac{1}{6}\text{AREA}(T) & \text{if } i = j \\ \frac{1}{12}\text{AREA}(T) & \text{otherwise,} \end{cases}$$

where $\mathcal{T}(i)$ is the set of triangles adjacent to vertex i. Such $\mathbf{M}$ is referred to as the *full mass matrix*, which has the same sparsity pattern as the graph Laplacian induced by the mesh graph.

Discretization: Lumped Mass Matrix The full mass matrix $\mathbf{M}$ is a sparse but non-diagonal matrix, making it expensive to invert $\mathbf{M}$ or solve the linear system $\mathbf{M}\mathbf{x} = \mathbf{b}$, where $\mathbf{b}$ is e.g. the mean curvature normal in Laplacian smoothing [Sorkine et al., 2004]. In practice, instead of using this exact full mass matrix, a *lumped* (diagonalized) mass matrix is usually used instead. This substitution can be justified, in the sense that the asymptotic convergence rate remains the same if a lumped mass matrix is used [Ciarlet, 2002]. The diagonal entries of a lumped mass matrix associate each vertex i with some area A_i surrounding it, such that $\mathbf{M}_{ii} = A_i$. Popular area dividing schemes lead to the barycentric or Voronoi mass matrices.



(a) Barycentric area. (b) Voronoi area.

Figure 7: The barycentric area (a) and the Voronoi area (b) associated to the vertex in the center.

For any triangle T in the mesh, the *barycentric mass matrix* $\mathbf{M}^b$ simply associates $\frac{1}{3}\text{Area}(T)$ to each of T's incident vertex, as shown in Figure 7(a)

$$\mathbf{M}^b_{ii} = \sum_{T \in \mathcal{T}(i)} \frac{1}{3} \text{AREA}(T).$$

The *Voronoi mass matrix* $\mathbf{M}^v$ sets $\mathbf{M}^v_{ii}$ as the Voronoi area around vertex i, i.e., the area of a neighborhood in the Voronoi diagram as shown in Figure 7(b). In [Meyer et al., 2003], the Voronoi mass matrix is recommended, since it leads to a provably tight error bound when used to compute the mean curvature; a modified Voronoi scheme is proposed to handle negative areas due to obtuse triangles.

5.2 Laplace–Beltrami (Intrinsic Laplacian)

In the extrinsic 2-dimensional space (flat planar domain), the Laplacian operator is defined as: $\Delta_{\mathbb{R}^2} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. The manifold Laplacian (Laplace–Beltrami) generalizes this definition to 2-manifolds, including curved surfaces, as follows:

$$\Delta_{\mathcal{M}} = \nabla_{\mathcal{M}} \cdot \nabla_{\mathcal{M}} = \frac{1}{\sqrt{|\det g|}} \partial_i \left(\sqrt{|\det g|} g^{ij} \partial_j \right) \tag{25}$$

where g_{ij} is the metric tensor and $(g^{ij}) = (g_{ij})^{-1}$ is the inverse metric. This definition of Laplacian only involves the metric of $\mathcal{M}$; in other words, the Laplacian is an intrinsic operator, remaining invariant under isometric transformations which does not change the metric g.

Harmonic functions Harmonic functions, i.e., function $u(\cdot)$ such that $\Delta_{\mathcal{M}} u = 0$, enjoy a long list of special properties, including the *Mean Value Theorem*, which roughly states that $u(\mathbf{x})$ equals to the average value of $u(\cdot)$ evaluated in any neighborhood of $\mathbf{x}$, and consequently the *Maximum Principle*, which states that extrema of $u(\cdot)$ must lie on the boundary $\partial\mathcal{M}$. Additionally, harmonic functions are C^∞. This is a remarkable fact: although the definition of harmonic function only involves second-order derivatives, the condition $\Delta_{\mathcal{M}} u = 0$ is so strong that $u(\cdot)$ has to be infinitely differentiable (i.e. smooth).

These properties contribute to the wide applicability of the Laplacian operator. In mathematics and physics, the Laplace equation Eq. 1 is particularly common, describing a wide range of physical phenomena including static electric potential fields, steady-state fluid flow, equilibrium states of diffusion processes and temperature distributions, and others. These models have been introduced to graphics and geometry processing, appearing in a large number of early works: Laplacian mesh editing [Sorkine et al., 2004, Sorkine, 2005], Poisson mesh editing [Yu et al., 2004], Poisson reconstruction [Kazhdan et al., 2006], Harmonic coordinates [Joshi et al., 2007], Green coordinates [Lipman et al., 2008], mesh deformation and fairing [Botsch and Kobbelt, 2004], and so on. Solutions to the Laplacian system are robust to noises and sampling bias, favorable in practical applications.

Discretization On flat or curved triangle meshes, the Laplacian operator has the following celebrated cotangent formula [Pinkall and Polthier, 1993]:

$$\mathbf{L}_{ij} = \begin{cases} \frac{1}{2}\left(\cot \alpha_{ij} + \cot \beta_{ij}\right) & \text{if } \{i,j\} \in \mathbf{E} \\ -\sum_{j \neq i} \mathbf{L}_{ij} & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

where α_{ij}, β_{ij} are the two angles opposite to edge ij, as labeled in Figure 8.

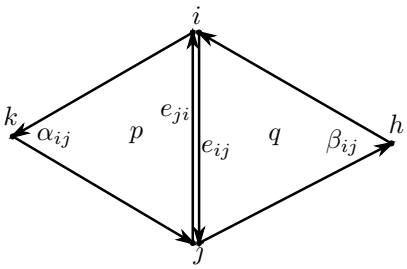


Figure 8: An illustration of notations used in this paper. p, q are two adjacent triangles. Triangle p consists of vertices ikj, and triangle q consists of vertices ijh. α_{ij} and β_{ij} are the two angles opposite to the undirected edge ij in the two triangles, respectively. $e_{ij} = -e_{ji}$ is a pair of opposite directed edges ij and ji.

The contangent formula in Eq. 26 can be derived in many ways. Using FEM, this formula can be derived by evaluating $\mathbf{L}_{ij}$ as the inner product between gradients of hat basis functions $\mathbf{L}_{ij} = -\langle \nabla_{\mathcal{M}} \phi_i, \nabla_{\mathcal{M}} \phi_j \rangle_{\mathcal{M}}$, where ϕ_i is piecewise linear basis (hat basis) centered at vertex i. Evaluating this integral is a standard computation using calculus; the exact steps of calculation can be found in e.g. [Zhang et al., 2010a]. An alternative derivation of the formula Eq. 26 using the finite volume method is given in [Botsch et al., 2010, Chapter 3.3]. A derivation using DEC can be found in e.g. [Hirani, 2003].

The discrete approximants can be understood in the strong sense that $\Delta_{\mathcal{M}} u \approx \mathbf{M}^{-1}\mathbf{L}\mathbf{u}$ or in the weak sense that $\int_{\mathcal{M}} (\nabla_{\mathcal{M}} u)^2 \approx -\mathbf{u}^{\intercal}\mathbf{L}\mathbf{u}$, where $\mathbf{M}$ is the mass matrix. Using an lumped mass matrix leads to the following discrete definition of $\Delta_{\mathcal{M}}$ on a triangulated mesh:

$$\Delta_{\mathcal{M}} u\left(v_i\right) \approx \frac{1}{\mathbf{M}_{ii}} \sum_{v_j \in \mathcal{N}_1(i)} \frac{1}{2} \left(\cot \alpha_{ij} + \cot \beta_{ij}\right) \left(\mathbf{u}_j - \mathbf{u}_i\right)$$

where $\mathcal{N}_1\left(i\right)$ is the set of 1-ring neighbors adjacent to vertex i.

5.3 Combinatorial and Graph Laplacians

The signal processing approach to geometry processing was first introduced in the seminal work [Taubin, 1995], generalizing classical discrete Fourier analysis to discrete surface signal analysis. Prior to the popularity of the cotangent Laplacian, the uniform Laplacian (Tutte Laplacian) or combinatorial Laplacian [Zhang, 2004] was used instead. The uniform Laplacian [Taubin, 1995] is a graph Laplacian with uniform weights:

$$\left(\mathbf{L}\mathbf{u}\right)_i := \frac{1}{\left|\mathcal{N}_1\left(i\right)\right|} \sum_{v_j \in \mathcal{N}_1(i)} \left(\mathbf{u}_j - \mathbf{u}_i\right).$$

This purely combinatorial and topological definition fails to converge to a continuous operator that captures the geometry of the surface. This discrepancy, in practical applications, can lead to undesirable behavior; as an example, the resulting parameterization can have large distortions as observed in [Zhou et al., 2004]. This issue is resolved by the geometry-aware Laplacian, which behaves more consistently compared to the graph Laplacian; a qualitative comparison of spectral behaviors can be found in e.g. [Sorkine, 2006, Dong et al., 2006]. For this reason, graphics methods based on the graph Laplacian have been replaced by those using the cotangent Laplacian. Despite the drawbacks, eigenfunctions of graph Laplacians, favored for their simplicity, have been applied to a variety of geometric deep learning methods recently such as the spectral mesh CNN [Yi et al., 2016].

5.4 Restricted Laplacian

The restricted Laplacian [Chuang et al., 2009] also discretizes the Laplace–Beltrami operator, but instead of using piecewise linear hat basis $\phi_i(\cdot)$, it uses a different basis $b_i(\cdot)$ defined in the extrinsic 3D space:

$$\mathbf{L}_{ij} = \langle \nabla_{\mathcal{M}} b_i, \nabla_{\mathcal{M}} b_j \rangle_{\mathcal{M}}.$$

In particular, the basis $b_i(\cdot)$ is chosen as spatial polynomials (obtained as product of second order B-spline basis) supported on nearby voxels. Although the basis is defined extrinsically, the resulting operator is

close to the cotangent Laplacian in Eq. 26 and converges to the same intrinsic Laplacian under mesh refinement. A major advantage of this restricted Laplacian is its robustness to mesh tessellation: the cotangent Laplacian has undesirable numerics and behavior on meshes with low-quality tessellation since the function $\cot\theta \to \infty$ as $\theta \to 0$.

5.5 Scale Invariant Laplacian

In Eq. 25, the Laplacian can be defined given any metric g, not necessarily the natural metric induced by the embedded space $\mathbb{R}^3$. Following this observation, [Aflalo et al., 2013] proposes to use an alternative metric $\tilde{g}$, the scale and isometry invariant metric, which scales the original metric by $\tilde{g}_{ij} = |K|g_{ij}$, where K is the Gaussian curvature. This modified metric factors out the effect of scaling, allowing scale-invariant analysis. This new metric tensor induces a new Laplacian $\Delta_{\tilde{g}}$, using the same definition in Eq. 25 but replacing g with $\tilde{g}$. The new Laplacian $\Delta_{\tilde{g}}$ and its derived shape descriptors, by design, are scale-invariant. Prior work [Bronstein and Kokkinos, 2010] develops scale-invariant descriptors directly, without introducing a scale-invariant metric. The scale invariant metric is also introduced for volumetric domains [Raviv and Raskar, 2015].

The scale invariant Laplacian $\Delta_{\tilde{g}}$ is discretized as $\mathbf{K}^{-1}\mathbf{M}^{-1}\mathbf{L}$, where $\mathbf{M}, \mathbf{L}$ are the same (lumped) mass matrix and cotangent matrix used for intrinsic Laplacian, and $\mathbf{K}$ is a diagonal matrix such that $\mathbf{K}_{ii} = \sqrt{K_i^2 + \epsilon^2}$, where K_i is the discrete Gaussian curvature at vertex i. K_i can be approximated as $K_i = \frac{2\pi - \sum_j \gamma_j^i}{\frac{1}{3}A_i}$, where γ_j^i is an angle adjacent to vertex i, and A_i again is the area of vertex i.

Self functional maps [Halimi and Kimmel, 2018] propose to use functional maps between eigenfunctions of $\Delta_{\tilde{g}}$ and Δ_g as a shape signature.

5.6 Affine and Equi-Affine Invariant Laplacian

The sequence of works [Raviv et al., 2011b, Spagnuolo et al., 2012, Raviv et al., 2014a, Raviv and Kimmel, 2015] consider the equi-affine and affine invariant geometry for analyzing shapes under non-rigid transformations. *Affine invariance* means that surfaces are considered the same under affine transformations, i.e., linear transformations $\mathbf{x} \mapsto \mathbf{A}\mathbf{x} + \mathbf{b}$, including squeezing and shearing. *Equi-affine invariance* means that surfaces are considered the same under affine transformations that preserve volumes, i.e., $\det(\mathbf{A}) = 1$. Particularly, an equi-affine invariant metric is first defined in [Raviv et al., 2011b], on top of which an *equi-affine invariant Laplacian* can be constructed, again using the same definition in Eq. 25. The later [Raviv and Kimmel, 2015] obtain an affine invariant metric and the derived *affine invariant Laplacian*, by applying the technique of scale invariant metric construction [Aflalo et al., 2013] to the equi-affine invariant metric.

[Raviv et al., 2011a] apply this insight to a combined geometry-color space, constructing shape descriptors fusing geometric and photometric information. [Raviv et al., 2014b] apply the affine invariant approach to medical imaging, evaluating local contractions of soft tissues.

5.7 Anisotropic Laplacian

One implicit assumption in deriving the (regular) Laplacian $\Delta_{\mathcal{M}}$ is the isotropic assumption, that within a local chart (tangent plane) the model assumes there is no difference along any direction. Consequently, the diffusion equation associated with $\Delta_{\mathcal{M}}$ assumes that the heat diffuses at the same speed in all directions, as in Figure 2. This assumption can be removed, leading to a more general model.

In the flat domain $\mathbb{R}^2$, removing isotropy means that $\Delta_{\mathbb{R}^2} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is replaced by the more general formula $\Delta^A_{\mathbb{R}^2} = \sum_{i,j=1}^2 \frac{\partial}{\partial x_j}\left[a_{ji}(\mathbf{x})\frac{\partial}{\partial x_i}\right]$, where $a_{ji}(\mathbf{x})$ is a function of $\mathbf{x}$. On a curved surface, the formula becomes the expression

$$\Delta^A_{\mathcal{M}} = \nabla_{\mathcal{M}} \cdot (A(\mathbf{x})\nabla_{\mathcal{M}})$$

where $A(\mathbf{x}) \in \mathbb{R}^{2\times 2}$ is the thermal conductivity tensor (matrix) that operates on tangent vectors. $\Delta^A_{\mathcal{M}}$ is the *anisotropic Laplacian* operator; the additional degrees of freedom to choose a tensor field $A(\mathbf{x})$ allows the anisotropic Laplacian to be *directionally sensitive*.

[Andreux et al., 2014] propose to use an anisotropic Laplacian for shape analysis, surpassing methods based on the usual isotropic Laplacian by a large margin on tasks including segmentation and region detection. *Anisotropic diffusion descriptors* [Boscaini et al., 2016b] consider spectral feature descriptors derived by the anisotropic Laplacian, and propose to learn anisotropic kernels. Learnable anisotropic kernels have been further applied to geometric deep learning to construct *anisotropic convolutional neural*

networks (ACNN) [Boscaini et al., 2016a]. A more restrictive anisotropic Laplacian has been proposed earlier in [Shi et al., 2008] to extract Reeb graphs and skeletons; this version of operator amounts to setting $a_{12}(\mathbf{x}) = a_{21}(\mathbf{x}) = 0$ and $a_{1,1}(\mathbf{x}) = a_{2,2}(\mathbf{x})$ in the general form.

Discretization The anisotropic diffusion kernel $A(\mathbf{x})$ needs to be prescribed in a pointwise fashion, usually in a local coordinate system; [Andreux et al., 2014] use principal curvature directions for this purpose, and [Boscaini et al., 2016b] propose an alternative kernel parameterization independent of the extrinsic curvature. Following [Andreux et al., 2014], we assume an orthonormal frame field is given $\hat{\mathbf{u}}, \hat{\mathbf{v}} \in \mathbb{R}^3$, which can be e.g. the directions of principal curvatures. The thermal conductivity tensor $A(\mathbf{x})$ is given per-triangle as $\mathbf{U}_{ijk} \in \mathbb{R}^{3\times 3}$, operating w.r.t. the frame field $\hat{\mathbf{u}}, \hat{\mathbf{v}}$. Let $\hat{\mathbf{e}}_{ij} \in \mathbb{R}^3$ denote the normalized edge vector from vertex i to vertex j. Define $\mathbf{H}$ as the *shear matrix*, which depends on $\mathbf{U}$ and $A(\mathbf{x})$ and encodes anisotropic scaling up to an orthogonal basis change; for an exact formula for $\mathbf{H}$, we refer to the original paper [Andreux et al., 2014]. Then, we can write

$$\mathbf{L}_{ij} = \begin{cases} \frac{1}{2}\left(\frac{\langle \hat{\mathbf{e}}_{kj}, \hat{\mathbf{e}}_{ki}\rangle_{\mathbf{H}}}{\sin\alpha_{ij}} + \frac{\langle \hat{\mathbf{e}}_{hj}, \hat{\mathbf{e}}_{hi}\rangle_{\mathbf{H}}}{\sin\beta_{ij}}\right) & \text{if } \{i,j\} \in \mathbf{E} \\ -\sum_{k\neq i} \mathbf{L}_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

where $\langle\cdot,\cdot\rangle_{\mathbf{H}}$ is the $\mathbf{H}$-weighted inner product, α_{ij}, β_{ij} are the two angles opposite to edge ij, and k, h are opposite vertices on which α_{ij}, β_{ij} locate at, respectively, as labeled in Figure 8. In the isotropic case, we have $\mathbf{H} = \mathbf{I}$.

5.8 Hessian and Normal Restricted Hessian: A Family of Linearized Energies

The Hessian matrix of any nonlinear deformation energy at the rest pose can be used as an alternative to the Laplacian. This Hessian is guaranteed to be p.s.d., since the deformation energy is minimized at the rest pose. Discussed in [Zorin, 2005, Hildebrandt et al., 2010], this observation provides an approach to systematically derive new operators by linearizing any nonlinear deformation energy. Analyzing eigenmodes of the Hessian is known as *modal analysis*, a technique that has been applied to reduced physical simulation [Barbič and James, 2005], shape segmentation [Huang et al., 2009], and shape analysis [Hildebrandt et al., 2010].

The resulting Hessian is a matrix in $\mathbb{R}^{3n\times 3n}$ for a mesh with n vertices, and its eigenfunctions in $\mathbb{R}^{3n}$ can be thought of as principal velocity fields along which the energy varies. Analogous to the regular heat kernel signature introduced as the trace of heat kernel of Laplacian, [Hildebrandt et al., 2010] introduce a vector version named *vibration signature*, which uses the principal velocity eigenmodes. Similarly, a vector-based *the vibration signature* is introduced using a similar definition to the diffusion distance.

[Hildebrandt et al., 2010] also propose restricting the velocity to the normal direction, leading to a matrix in $\mathbb{R}^{n\times n}$.

5.9 Modified Dirichlet Energy

Applying the Hessian approach to the thin shell energy leads to the *modified Dirichlet energy* [Hildebrandt et al., 2010, Hildebrandt et al., 2012]. The modified Dirichlet energy is the sum of the Dirichlet energy $\int_{\mathcal{M}} \frac{1}{2}\|\nabla_{\mathcal{M}} u(\mathbf{x})\|_2^2$ and an additional term

$$\int_{\mathcal{M}} \frac{1}{2}(\kappa_1^2(\mathbf{x}) + \kappa_2^2(\mathbf{x}))u(\mathbf{x})^2 \equiv \int_{\mathcal{M}} \left(H(\mathbf{x})^2 - {}^1\!/_2 K(\mathbf{x})\right) u(\mathbf{x})^2$$

where $\kappa_1(\mathbf{x})$ and $\kappa_2(\mathbf{x})$ are the two principal curvatures at $\mathbf{x}$. In geometric modeling, when $u(\mathbf{x}) = 1(\mathbf{x})$ the additional term is often referred to as the total curvature functional and is closely related to the Willmore energy [Zorin, 2005], that is, $\int_{\mathcal{M}} \frac{1}{4}(\kappa_1(\mathbf{x}) - \kappa_2(\mathbf{x}))^2 \equiv \int_{\mathcal{M}} \left(H(\mathbf{x})^2 - K(\mathbf{x})\right)$.

The additional term is extrinsic, penalizing $u(\cdot)$ in regions with large extrinsic bending. Applying the spectral surface quadrangulation method of [Dong et al., 2006], but instead using these new eigenfunctions, yields quadrangulations that better align to the extrinsic curvature of the surface. The descriptors and distances derived using the modified Dirichlet energy are sensitive to extrinsic curvatures.

5.10 Hamiltonian (Schrödinger) Operator

The *Hamiltonian operator* [Choukroun et al., 2018b] $\mathcal{H}$, or the *Schrödinger operator* [Choukroun et al., 2018a], is an elliptic operator of the form

$$\mathcal{H}f = -\Delta_{\mathcal{M}} f + \mathcal{V}f, \tag{28}$$

where $f \in \mathcal{L}^2(\mathcal{M})$, and $\mathcal{V}: \mathcal{L}^2(\mathcal{M}) \to \mathcal{L}^2(\mathcal{M})$ is the *diagonal operator* defined as $[\mathcal{V}f](\mathbf{x}) = V(\mathbf{x})f(\mathbf{x})$, in which $V: \mathcal{M} \to \mathbb{R}$ is a prescribed real-valued scalar function.

The weak form of the Hamiltonian operator amounts to modifying the Dirichlet energy by adding the term

$$\int_{\mathcal{M}} V(\mathbf{x})u(\mathbf{x})^2.$$

This additional term penalizes $u(\cdot)$ differently on different regions on $\mathcal{M}$. In particular, using the variational definition of the eigenvalue problem, the lowest eigenfunctions will avoid regions where $V(\mathbf{x})$ is large and will instead concentrate on regions with low potential. [Choukroun et al., 2018a] study the problem of designing potential functions.

The PDE associated with the operator, the *Schrödinger equation*, can be viewed as replacing the Laplacian in the wave equation with the Hamiltonian operator. The Schrödinger equation in quantum mechanics prescribes the probability wave density function $\Psi(x,t)$ of a particle with mass m under potential field V:

$$i\hbar \frac{\partial \Psi}{\partial t} = \frac{-\hbar^2}{2m} \Delta_{\mathcal{M}} \Psi + \mathcal{V}\Psi, \tag{29}$$

where $\hbar$ is the Planck's constant and $|\Psi(\mathbf{x},t)|^2$ is interpreted as the probability density distribution.

5.11 Curvature Laplacian

Among the earliest work applying a spectral approach to shape analysis, [Liu and Zhang, 2007] introduce a graph Laplacian $\mathbf{L}$ called *curvature Laplacian*. $\mathbf{L} := \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a matrix whose diagonal stacks summations of rows in the weight matrix $\mathbf{W}$, i.e, $d_i = \sum_{j=1}^{n} \mathbf{W}_{ij}, \mathbf{D} = \operatorname{diag}(d_i)$, and $\mathbf{W}$ is defined as

$$\mathbf{W}_{ij} = \begin{cases} \left(\left|\kappa_i^{min}\right| + \left|\kappa_j^{min}\right|\right) \frac{|\mathbf{e}_{ij}|}{h} |\langle \frac{\mathbf{e}_{ij}}{|\mathbf{e}_{ij}|}, \mathbf{z} \rangle| & \text{if } \kappa_i^{min} < 0 \text{ or } \kappa_j^{min} < 0 \\ \varepsilon & \text{otherwise,} \end{cases}$$

where κ_i^{min} is the minimal principal curvature at vertex i, $\mathbf{e}_{ij}$ is the (unnormalized) edge vector of edge ij (as labeled in Figure 8), h is the average edge length in the mesh, and $\mathbf{z}$ is the normalized average of the directions for curvatures κ_i^{min} and κ_j^{min}. When $\kappa_i^{min}, \kappa_j^{min} \geq 0$, which implies *local convexity*, $\mathbf{W}_{ij}$ is set to a small value ϵ merely to maintain mesh connectivity. An edge ij will have a large weight if vertex i or j has a negative minimal principal curvature direction roughly aligning with $\mathbf{e}_{ij}$; this implies that locally the mesh is bent roughly along edge ij, so the edge ij should not be cut in a spectral embedding using this operator. The spectral embedding of this Laplacian, combined with contour analysis, results in segmentation that is aware of the extrinsic bending.

5.12 Concavity-aware Laplacian

With the observation that vertices on concave seams are often part of potential segmentation boundaries, [Au et al., 2012, Wang et al., 2014] propose a *concavity-aware Laplacian* of the following form:

$$\mathbf{L}_{ij} = \begin{cases} \dfrac{w_{ij}}{\sum_{\{i,k\} \in \mathbf{E}} w_{ik}} & \text{if } \{i,j\} \in \mathbf{E} \\ -1 & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{E}$ is the set of edges, and w_{ij} is a concavity-sensitive weights:

$$w_{ij} = \frac{e_{ij}\beta}{|K_i| + |K_j| + \epsilon}.$$

In this expression, K_i is the Gaussian curvature at vertex i, and $\beta = 0.01$ if either i or j is a concave vertex and $\beta = 1$ otherwise. The vertex concavity factor β is determined locally by using the inner product

between edge vector directions and normal difference, or more robustly via PCA within the vertex's 1-ring neighborhood; to further improve robustness, this operator can be constructed on a smoothed mesh.

[Au et al., 2012] define solutions to linear systems involving the concavity-aware Laplacian to be *concavity-aware fields*, whose isolines align with concave seams and serve as candidate cuts. [Wang et al., 2014] use the eigenfunctions of the concavity-aware Laplacian in a framework involving hierarchical spectral analysis and isoline-based boundary detection.

5.13 Extrinsic and Relative Dirac Operators

The Dirac operator D, named after Paul Dirac, is a square root of the negative semi-definite Laplacian operator, whose definition thus has to resort to number systems with imaginary parts. In 1D we have $D = \sqrt{-1}\frac{\partial}{\partial x}$ and $D^2 = -\frac{\partial^2}{\partial x^2} = \Delta_x$; this definition can be generalized to a curved surface. Dirac operators are defined to operate on spinors on abstract manifolds, and in particular on a 3D surface, spinors can be understood as a quaternion-valued fields, suggesting the notation $D\colon \mathcal{L}^2(\mathcal{M};\mathbb{H}) \to \mathcal{L}^2(\mathcal{M};\mathbb{H})$. There are multiple Dirac operators D; we discuss the *extrinsic Dirac operator* [Kamberov et al., 1996] first.

Extrinsic Dirac Operator D_f Identifying $\mathbb{R}^3$ with $\mathrm{Im}(\mathbb{H})$, a 3D surface can be defined as a map $f\colon \mathcal{M} \to \mathbb{H}$, i.e., from a point on the (abstract) surface $\mathcal{M}$ to $\mathbb{R}^3$. The differential of the map, $df\colon T\mathcal{M} \to \mathbb{H}$, maps a tangent vector to $\mathbb{R}^3$ or $\mathrm{Im}(\mathbb{H})$.

Consider a surface deformation that applies a local conformal transformation (i.e., rotation and scaling) to each tangent plane, using a smooth quaternion field $\phi(\cdot)$. In quaternion language, this procedure can be written as a pointwise modification on df to $\tilde{df} = \bar{\phi} \cdot df \cdot \phi$. For $\tilde{df}$ to be integrable, i.e., consistent in the sense that the modification remains the differential of a surface, the following Dirac equation (or *integrability condition*) has to hold:

$$D_f \phi = \rho \phi \tag{30}$$

where $\rho\colon \mathcal{M} \to \mathbb{R}$ is a real-valued function that controls local distortion of the curvature, and D_f is the extrinsic Dirac operator defined as

$$D_f \phi := -\frac{\mathrm{d}f \wedge \mathrm{d}\phi}{|\,\mathrm{d}f|^2}.$$

Here, $|\,\mathrm{d}f|^2$ is the squared length element that corresponds to the unsigned area element for surface f. [Crane et al., 2011] propose to use the Dirac equation Eq. 30 to compute the conformal transformation $\phi(\cdot)$ of a triangle mesh, with prescribed function $\rho(\cdot)$ encoding curvature information. This work proposes the following discrete version of the extrinsic Dirac operator $\mathbf{D} \in \mathbb{H}^{|F|\times|V|}$. $\mathbf{D}_{pi}$ is nonzero as long as vertex i is within triangle p, given by the expression

$$\mathbf{D}_{pi} = -\frac{1}{2A_p} e_p^{(i)},$$

where A_p is the area of triangle p, and $e_p^{(i)}$ is the opposing edge vector of vertex i within triangle p in quaternion value. The right-hand rule has been assumed, such that edge vectors are oriented counter-clockwise around each triangle. Numerically, a quaternion-valued matrix can be equivalently written as a real-valued matrix by expanding each entry into a 4×4 block. The extrinsic Dirac operator also appears in conformal surface flows [Crane et al., 2013a], with applications to surface fairing. *Surface Networks* [Kostrikov et al., 2018] use this version of Dirac operator in deep neural networks, to capture extrinsic geometric information.

Relative Dirac Operator D_N As mentioned earlier, D_f can be thought of as a square root of the Laplacian. This phrase can be understood via the relation

$$D_f^2 \phi = \Delta\phi + \frac{\mathrm{d}N \wedge \mathrm{d}\phi}{|\,\mathrm{d}f|^2}$$

where N is the Gauss map. [Liu et al., 2017] make the observation that the additional term

$$D_N \phi := D_{f,N} \phi = -\frac{\mathrm{d}N \wedge \mathrm{d}\phi}{|\,\mathrm{d}f|^2}$$

defined as the *relative Dirac operator* of a single surface f, is a purely extrinsic operator. The closely related relative Dirac operator *between two surfaces* f_1, f_2 with correspondence is defined as

$$D_{f_1,f_2}\phi := -\frac{\mathrm{d}f_2 \wedge \mathrm{d}\phi}{|\mathrm{d}f_1|^2}.$$

With this theory in place, [Liu et al., 2017] use the following one-parameter family of operators to linearly blend the Laplacian and the relative Dirac operator

$$L(\tau) := (1-\tau)\Delta + \tau D_N.$$

Since the relative Dirac operator vanishes on purely flat regions, this blending with the Laplacian injects ellipticity (positive definiteness) that constrains the large kernel of the operator and stabilizes the lowest eigenfunctions.

By considering ϕ as a face-based function, [Ye et al., 2018] define an alternative discrete extrinsic Dirac operator $\mathbf{D}_E \in \mathbb{H}^{|F|\times|F|}$. We denote the extrinsic Dirac operator as D_E, which comes from the same continuous operator as D_f; this alternative notation is merely to reflect that a different discretization will be used. For each triangle p,

$$(\mathbf{D}_E\boldsymbol{\phi})_p = \frac{1}{2}\sum_q (2H_{pq} + e_{pq})\,(\boldsymbol{\phi}_q - \boldsymbol{\phi}_p)$$

where $\boldsymbol{\phi} \in \mathbb{H}^{|F|\times 1}$, the sum $\sum_q \cdot$ is over all the three triangles adjacent to triangle p, H_{pq} is a quaternion whose real part equals to the integrated mean curvature H_{pq} over the edge, and e_{pq} is a quaternion whose imaginary part equals to the edge vector in $\mathbb{R}^3$; unspecified real or imaginary parts are always zero. Here the integrated mean curvature is $H_{pq} := \frac{1}{2}\,|e_{pq}| \tan\frac{\theta_{pq}}{2}$, where θ_{pq} is the bending angle at edge pq.

This version of the extrinsic Dirac operator connects to a discrete spin transformation, resulting in improved numerical performance. It also leads to a discrete intrinsic Dirac operator derived in a unified framework, discussed as follows.

5.14 Intrinsic Dirac Operator D_I

The recent work [Ye et al., 2018] also proposes to use an *intrinsic Dirac operator*, whose original intrinsic definition requires the languages of spinors and spin manifolds. An equivalent definition of the intrinsic Dirac operator D_I is as follows:

$$D_I = D_f + H$$

where H is the mean curvature (as a diagonal operator). Although it can hardly be seen from this formula, H cancels out the extrinsic part of D_f, yielding an intrinsic operator D_I that only depends on surface metric.

Following the same notation in the previous subsection, an alternative discrete intrinsic Dirac operator $\mathbf{D}_I \in \mathbb{H}^{|F|\times|F|}$ on a mesh can be defined as

$$(\mathbf{D}_I\boldsymbol{\phi})_p = \frac{1}{2}\sum_q (2H_{pq} + e_{pq}) \cos\frac{\theta_{pq}}{2}\boldsymbol{\phi}_q.$$

The cosine factor plays the role of a correction, ensuring exact covariance of the discrete operator to isometric deformations, providing a key theoretical property analogous to the continuous counterpart [Ye et al., 2018]. Empirically this discretization improves performance on relevant applications. See [Hoffmann and Ye, 2018] for more theoretical justifications of this version of discrete Dirac operators, using a definition of the discrete spin structure. The definition of the intrinsic Dirac operator explicitly involves the spin structure, making it more discriminative than the intrinsic Laplacian, e.g., in distinguishing two approximately isometric tori with different spin structures as illustrated in [Ye et al., 2018, Figure 5].

5.15 Volumetric (Extrinsic) Laplacian

So far we have discussed operators taking the thin shell approach; now we move to operators that consider shapes as the boundaries of solids. Namely, starting from this subsection, we consider a volumetric domain Ω whose boundary is specified by a surface $\mathcal{M} = \partial\Omega$.

The Laplace equation with Neumann boundary conditions is defined as follows:

$$\begin{cases} -\Delta_{\mathbb{R}^3} u(\mathbf{x}) = 0 & \mathbf{x} \in \Omega \\ \dfrac{\partial}{\partial n} u(\mathbf{x}) = 0 & \mathbf{x} \in \partial\Omega \end{cases} \tag{31}$$

where $\Delta_{\mathbb{R}^3} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the (extrinsically defined) Laplacian operator in $\mathbb{R}^3$. Critical information about the surface is encoded by parts corresponding to the boundary condition. The volumetric heat kernel signature [Raviv et al., 2010, Wang and Wang, 2015, Patané, 2015] considers the heat kernel of $\Delta_{\mathbb{R}^3}$ and applies it to analyze solid shapes.

This volumetric Laplacian can also be discretized by a cotangent matrix $\mathbf{L}$: $\mathbf{L}_{ij}$ is nonzero if i, j are adjacent vertices, and the weights $\mathbf{L}_{ij}$ are cotangents of dihedral angles opposite to edges within the same tetrahedron; see e.g. [Jacobson, 2013, §2.1] for an explicit formula and derivation.

5.16 Hessian Energy

Hessian energy The Hessian LLE [Donoho and Grimes, 2003] substitutes the Hessian in place of the Laplacian used in LLE, enjoying several theoretical guarantees. For a volumetric domain Ω given by a tetrahedral mesh, [Stein et al., 2018] consider and discretize the Hessian energy defined as:

$$E_{\mathbf{H}^2}(u) := \frac{1}{2} \int_\Omega \|\mathbf{H}(u(\mathbf{x}))\|_F^2 = \frac{1}{2} \sum_{i,j} \int_\Omega \left(\frac{\partial^2}{\partial \mathbf{x}_i \partial \mathbf{x}_j} u(\mathbf{x}) \right)^2 \tag{32}$$

where $\mathbf{H}(u(\mathbf{x})) \in \mathbb{R}^{3\times 3}$ is the symmetric matrix consisting of second-order partial derivatives. Spectral analysis on the Hessian energy can be thought as an application of Hessian LLE to mesh data. An interesting property of the Hessian energy is that its null space contains only affine functions, i.e., $\mathbf{1}, \mathbf{x}, \mathbf{y}, \mathbf{z}$. [Stein et al., 2018] discretize the Hessian energy as a composition of the gradient operator and a matrix divergence operator, using a mixed FEM approach. As pointed out in their paper, such discretized operator sometimes includes spurious modes in its kernel; this issue seems to be empirically avoided by the followup work [Stein et al., 2019]. Furthermore, [Stein et al., 2019] generalize the (planar) Hessian energy to a curved surface, with an additional term correctly taking into account the effect of intrinsic curvature.

The bi-Laplacian energy, used for a variety of tasks [Botsch and Kobbelt, 2004, Zorin, 2005, Jacobson et al., 2010, Jacobson et al., 2011], is closely related to the Hessian energy; they both correspond to the same PDE (bi-harmonic equation $\Delta^2_{\mathbb{R}^3} u = 0$) but with different boundary conditions. [Fisher et al., 2007] present a DEC Laplacian with modified boundary conditions, and the resulting bi-Laplacian energy [Wang et al., 2015] also includes affine functions in the kernel, similarly to the Hessian energy.

5.17 Single Layer Potential Operator and Kernel Method

Starting from this subsection, we look at operators defined using boundary integrals. Such operators can be discretized using boundary element methods (BEM).

One example of a boundary integral operator is the *single layer potential operator* $\mathcal{V} : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$, which is defined on surface Γ as

$$[\mathcal{V}u](\mathbf{x}) := \int_\Gamma G(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) \, \mathrm{d}\Gamma(\mathbf{y}),$$

where $G(\mathbf{x}, \mathbf{y}) := \frac{1}{4\pi} \frac{1}{|\mathbf{x}-\mathbf{y}|}$ is the fundamental solution of the Laplace equation.

[Wu and Levine, 1997] simulate the shape as an electric conductor and exploit the resulting charge distribution for shape segmentation. This amounts to solving a linear system consisting of (the inverse of) the single layer potential operator. The charge density function for an electric conductor with constant electrical potential is $[\mathcal{V}^{-1}1](\mathbf{x})$, where $1(\cdot)$ is the constant function. It is well known in physics that this charge density function is highly correlated with the shape concavity: the charge tends to vanish at most concave regions and accumulate at sharp convexities. Based on this key observation, [Wu and Levine, 1997] use the charge density function as a segmentation clue: it first detects highly concave seams by tracing local minima charge density, and then cuts along these seams to segment the shape.

Boundary Element Method (BEM) The single layer potential is an example of an integral operator for which vertices globally interact with each other, unlike differential operators whose actions are local. In §3.3 we presented a framework applicable to both FEM and BEM; following the discussion there, BEM is a similar technique to FEM, with the difference that the inner product $\langle \psi_i, \phi_j \rangle_a$—which now involves boundary integrals—will be evaluated with different techniques. In the case of the single layer potential operator, the corresponding weak form is

$$a(u,v) = \int_\Gamma \int_\Gamma u(\mathbf{x}) G(\mathbf{x},\mathbf{y}) v(\mathbf{y}) \,\mathrm{d}\Gamma(\mathbf{x}) \,\mathrm{d}\Gamma(\mathbf{y}) \approx \sum_{\mathbf{x}_i} \sum_{\mathbf{y}_i} w(\mathbf{x}_i,\mathbf{y}_j) u(\mathbf{x}_i) G(\mathbf{x}_i,\mathbf{y}_j) v(\mathbf{y}_j), \textsc{Area}(\mathbf{x}_i) \textsc{Area}(\mathbf{y}_j),$$

where $\mathbf{x}_i, \mathbf{y}_j$ are the quadrature points, $\textsc{Area}(\mathbf{x}_i), \textsc{Area}(\mathbf{y}_j)$ are the associated areas, and $w(\mathbf{x}_i, \mathbf{y}_j)$ is the corresponding quadrature weight. Since $G(\mathbf{x},\mathbf{y})|_{\mathbf{y}=\mathbf{x}}$ is singular, special case needs to be taken when designing the quadrature scheme to properly handle the integrable singularity in the inverse distance kernel [Steinbach, 2007].

The strong form of the single layer operator is $\mathbf{V} \in \mathbb{R}^{n \times n}$, given by $\mathbf{V}_{ij} = \int_\Gamma \int_\Gamma \phi_i(\mathbf{x}) G(\mathbf{x},\mathbf{y}) \phi_j(\mathbf{y}) \,\mathrm{d}\Gamma(\mathbf{x}) \,\mathrm{d}\Gamma(\mathbf{y})$, an integrated version of piecewise inverse distances. $\mathbf{V}$ is a dense matrix, which usually admits a low rank approximation for efficient computations; details can be found in [Steinbach, 2007]. Getting back to the method of [Wu and Levine, 1997], after discretization, the resulting charge density is $\mathbf{V}^{-1}\mathbf{M}\mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^n$ is a constant vector.

More generally, it is possible to replace the kernel $\frac{1}{|\mathbf{x}-\mathbf{y}|}$ with $\frac{1}{|\mathbf{x}-\mathbf{y}|^p}$ or Gaussian $e^{-\frac{1}{t}|\mathbf{x}-\mathbf{y}|^2}$: this amounts to applying the kernel method in machine learning for vertex clustering [Shawe-Taylor et al., 2004].

5.18 Dirichlet-to-Neumann Operator (Poincaré-Steklov Operator)

Consider the Laplace equation with Dirichlet boundary conditions:

$$\begin{cases} \Delta_{\mathbb{R}^3} g(\mathbf{x}) = 0 & \mathbf{x} \in \Omega \\ g(\mathbf{x}) = u(\mathbf{x}) & \mathbf{x} \in \Gamma = \partial\Omega. \end{cases} \tag{33}$$

The *Dirichlet-to-Neumann operator (DtN)* $\mathcal{S} \colon \mathcal{L}^2(\Gamma) \to \mathcal{L}^2(\Gamma)$ is defined as the map $u \mapsto g_n = \frac{\partial}{\partial n} g(\Gamma)$. In physics, this operator models small vertical oscillations of an ideal fluid in a container. The eigenvalue problem of the DtN operator is known as the *Steklov eigenvalue problem*. Like the Laplace–Beltrami spectrum encoding intrinsic geometry, the Steklov spectrum encodes extrinsic geometry information: the Steklov heat kernel admits an asymptotic expansion whose coefficients are determined by the mean and the Gaussian curvatures $H(\mathbf{x})$ and $K(\mathbf{x})$ [Polterovich and Sher, 2015]. The Steklov spectral geometry also enjoys many other properties [Girouard and Polterovich, 2017].

The DtN operator can be written as the composition of boundary integral operators:

$$\mathcal{S} = \mathcal{V}^{-1} \left(\frac{1}{2}\mathcal{I} + \mathcal{K} \right) = \mathcal{H} + \left(\frac{1}{2}\mathcal{I} + \mathcal{K}^\intercal \right) \mathcal{V}^{-1} \left(\frac{1}{2}\mathcal{I} + \mathcal{K} \right)$$

where the double layer potential $\mathcal{K} : H^{1/2}(\Gamma) \to H^{1/2}(\Gamma)$ and hypersingular operator $\mathcal{H} : H^{1/2}(\Gamma) \to H^{-1/2}(\Gamma)$ are defined as:

$$[\mathcal{K}\phi](\mathbf{x}) := \int_\Gamma \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} \phi(\mathbf{y}) \,\mathrm{d}\Gamma(\mathbf{y}), \quad [\mathcal{H}\phi](\mathbf{x}) := -\int_\Gamma \frac{\partial^2 G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{x}) \partial n(\mathbf{y})} \phi(\mathbf{y}) \,\mathrm{d}\Gamma(\mathbf{y}).$$

To discretize the operator $\mathcal{S}$, it suffices to discretize these boundary operators $\mathcal{V}, \mathcal{K}, \mathcal{H}$. Although these operators are defined on the boundary, another remarkable property of $\mathcal{S}$ is that its weak form corresponds to the volumetric Dirichlet energy on the harmonic extension of a surface function into the interior. That is, the corresponding strong form is $a(u,v) = \int_\Omega \nabla \mathcal{E}v \cdot \nabla \mathcal{E}u \equiv \int_\mathcal{M} u \mathcal{S} v$, where $\mathcal{E}u$ is the smooth (harmonic) extension of u into the interior, i.e., $\mathcal{E}u$ is the solution to Eq. 33.

[Wang et al., 2018b] use the operator $\mathcal{S}$ for shape analysis, as an extrinsic and volumetric alternative to the Laplace–Beltrami operator. It successfully applies this operator to a large number of tasks, including point signature computation, volume-aware distances, segmentation, and extrinsic shape differences, while being robust to topological changes and noise that affects Laplacian-based computations.

5.19 Other Extrinsic Methods

There are many other methods capturing various aspects of extrinsic geometry, which may not necessarily define extrinsic operators explicitly.

The early spectral method [Pauly and Gross, 2001] decomposes meshes into patches, and for each patch applies Discrete Fourier Transform (DFT) and inverse DFT on extrinsic voxel grids. The Distribution of Distances (D2) [Osada et al., 2002] and the Signature of Histograms of Orientations (SHOT) [Tombari et al., 2010] descriptors are extrinsically defined geometric histograms. Spherical harmonics [Kazhdan et al., 2006] are basis functions in the extrinsic 3D space, independent to the shape to be analyzed. [Wang et al., 2012] define the discrete first and second fundamental forms as edge lengths and dihedral angles, and proposes a linear method to reconstruct the surface from the two discrete fundamental forms. [Corman et al., 2017b] introduce multiple shape difference operators including those encode extrinsic geometry. [Rustamov, 2011] interpolates intrinsic Laplacian eigenfunctions into the interior of a surface using barycentric coordinates, which carry extrinsic information, and builds descriptors using the interpolants.

6 Summary and Experiments

The geometric operators in this paper roughly fall into two categories: operators that view shapes as thin shells and operators that view shapes as solids. There are also other patterns and common wisdom about these operators, summarized as follows.

Relevance of curvature A common machinery occurring in multiple methods is to explicitly bake curvatures into a thin shell operator. Indeed, the modified Dirichlet energy [Hildebrandt et al., 2012] directly adds the scalar function $\frac{1}{2}(\kappa_1^2(\mathbf{x}) + \kappa_2^2(\mathbf{x})) = H(\mathbf{x})^2 - K(\mathbf{x})$ as a diagonal operator; the scale invariant Laplacian [Aflalo et al., 2013] uses a multiplicative factor, the inverse Gaussian curvature $K(\mathbf{x})^{-1}$, to cancel out the scaling effect in metric g; the curvature Laplacian [Liu and Zhang, 2007] is defined using the minimal principal curvature as well as its direction; the concavity-aware Laplacian [Au et al., 2012] constructs a discrete Laplacian whose weights are inverse Gaussian curvatures; and the extrinsic Dirac operator equals to the sum of the mean curvature $H(\mathbf{x})$ (as a diagonal operator) and the intrinsic Dirac operator [Ye et al., 2018].

Diagonal operator and operator blending Both the modified Dirichlet energy [Hildebrandt et al., 2012] and the Hamiltonian operator [Choukroun et al., 2018b] construct a diagonal operator from a scalar (potential) function, such that the two methods coincide if we set the potential function to be $V(\mathbf{x}) = \frac{1}{2}(\kappa_1^2(\mathbf{x}) + \kappa_2^2(\mathbf{x}))$. Dirac operators [Liu et al., 2017] use infinite well boundary conditions, which also amounts to adding a large potential function—as a diagonal operator—only at boundaries. Any non-negative scalar function can be converted into a p.s.d. diagonal operator, and it would be interesting to investigate alternative scalar functions rather than ones based on curvatures.

Also, all these methods effectively blend two eigen spaces by linearly adding one operator—usually the Laplacian—to another operator. The blended operator has a set of "compromised" eigenfunctions adapting to both operators, yields low energies measured by (the weak form of) both operators: take the Hamiltonian operator as an example, the Laplacian, or the Dirichlet energy, favors eigenfunctions that are smooth, while the potential energy $\int_{\mathcal{M}} \frac{1}{2} V(\mathbf{x}) u(\mathbf{x})^2$ encourages eigenfunctions to avoid regions with large $V(\mathbf{x})$, so the blended Hamiltonian operator has eigenfunctions that are smooth except in areas with very large $V(\mathbf{x})$. To sum up, operator blending provides a means of balancing the considerations of both operators.

Extrinsic or Intrinsic We already discuss the pros and cons of being an extrinsic or intrinsic operator in §1. Table 1 summarizes this key property of geometric operators.

Laplacian [Pinkall and Polthier, 1993]	$\Delta_{\mathcal{M}}$	Intrinsic
Scale-invariant [Aflalo et al., 2013]	$\Delta_{\mathcal{M}}^{(si)}$	Intrinsic
Anisotropic Laplacian [Andreux et al., 2014]	$\Delta_{\mathcal{M}}^{A}$	Either
Modified Dirichlet [Hildebrandt et al., 2012]	$\Delta_{\mathcal{M}}^{\kappa}$	Extrinsic
Relative Dirac [Liu et al., 2017]	D_R	Extrinsic
Extrinsic Dirac [Ye et al., 2018]	D_E	Extrinsic
Intrinsic Dirac [Ye et al., 2018]	D_I	Intrinsic
DtN / Steklov [Wang et al., 2018b]	$\mathcal{S}$	Extrinsic
Volumetric Laplacian	$\Delta_{\mathbb{R}^3}$	Extrinsic

Table 1: Summary of intrinsic and extrinsic operators.

6.1 Experiments

In the rest of this section, we provide experiments comparing the spectral properties of a few representative operators. First, we show the eigenfunctions of these operators, which can reveal rich information about geometry. As eigenfunctions are not unique due to sign ambiguities and rotations within eigenspaces formed by repeating eigenvalues, we then visualize quantities derived from eigenfunctions including heat kernel signatures and spectral distances, which remain invariant w.r.t. rotations within eigenspaces.

6.2 Eigenfunctions

First, we show the eigenfunctions computed on a cube with an outward bump in Figure 9, comparing them with those computed on a cube with an inward bump in Figure 10. Conforming with theoretical predictions, we see all intrinsic operators demonstrate identical patterns on both cubes, while the extrinsic operators behave differently on the two cubes. It is also interesting to see distinguishing behaviors among these operators. All intrinsic eigenfunctions, as well as extrinsic ones, are aware of corners of the cube, which are singularities of the Gaussian curvature. This is particularly the case for the scale invariant Laplacian: since the Gaussian curvature approaches infinity at the corners, the corresponding eigenfunctions also have singular patterns at these corners. The anisotropic Laplacian demonstrates "vortious patterns" in its eigenfunctions, thanks to the explicit use of an anisotropic kernel.

In addition to corners, all extrinsic operators are sensitive to edges, where the mean curvature becomes large. For the modified Dirichlet energy, the additional term $\int_{\mathcal{M}} \frac{1}{2}(\kappa_1^2(\mathbf{x})+\kappa_2^2(\mathbf{x}))u(\mathbf{x})^2$ penalizes function $u(\cdot)$ at edges and corners where $(\kappa_1^2(\mathbf{x}) + \kappa_2^2(\mathbf{x}))$ becomes very large, making eigenfunctions vanish at edges and effectively decomposing the cube into six disconnected pieces separated by edges. Similarly, the extrinsic Dirac operator, as the sum of the intrinsic Dirac operator with the mean curvature, also has eigenfunctions whose patterns are separated by edges. The eigenfunctions of the DtN operator also put a particular emphasis on edges of the cube. The relative Dirac operator resonates most in extrinsically flat regions on the shape.

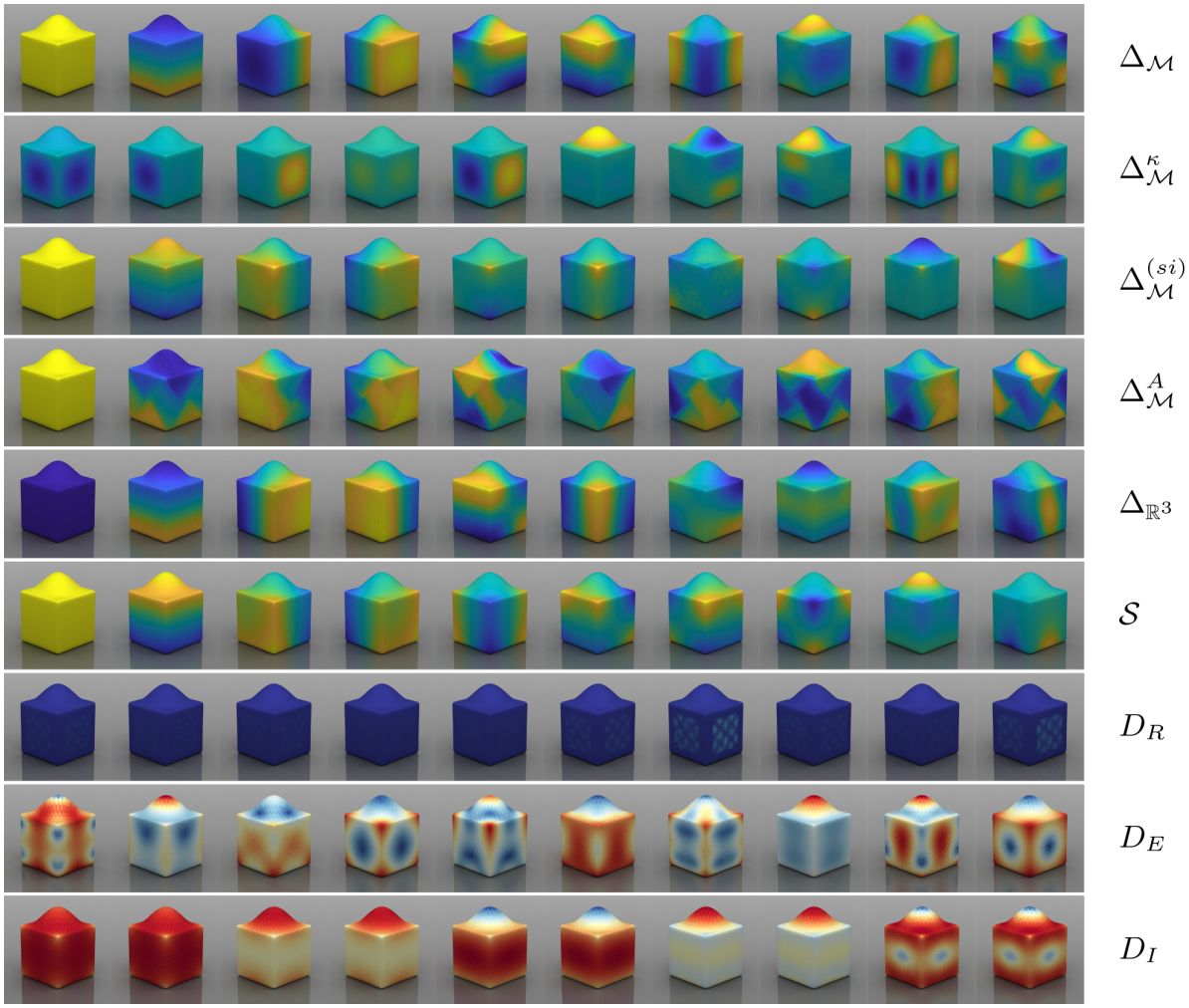


Figure 9: First few eigenfunctions computed on the cube with an outward bump. For the Dirac operators, norms of the (quaternion valued) eigenfunctions are visualized.

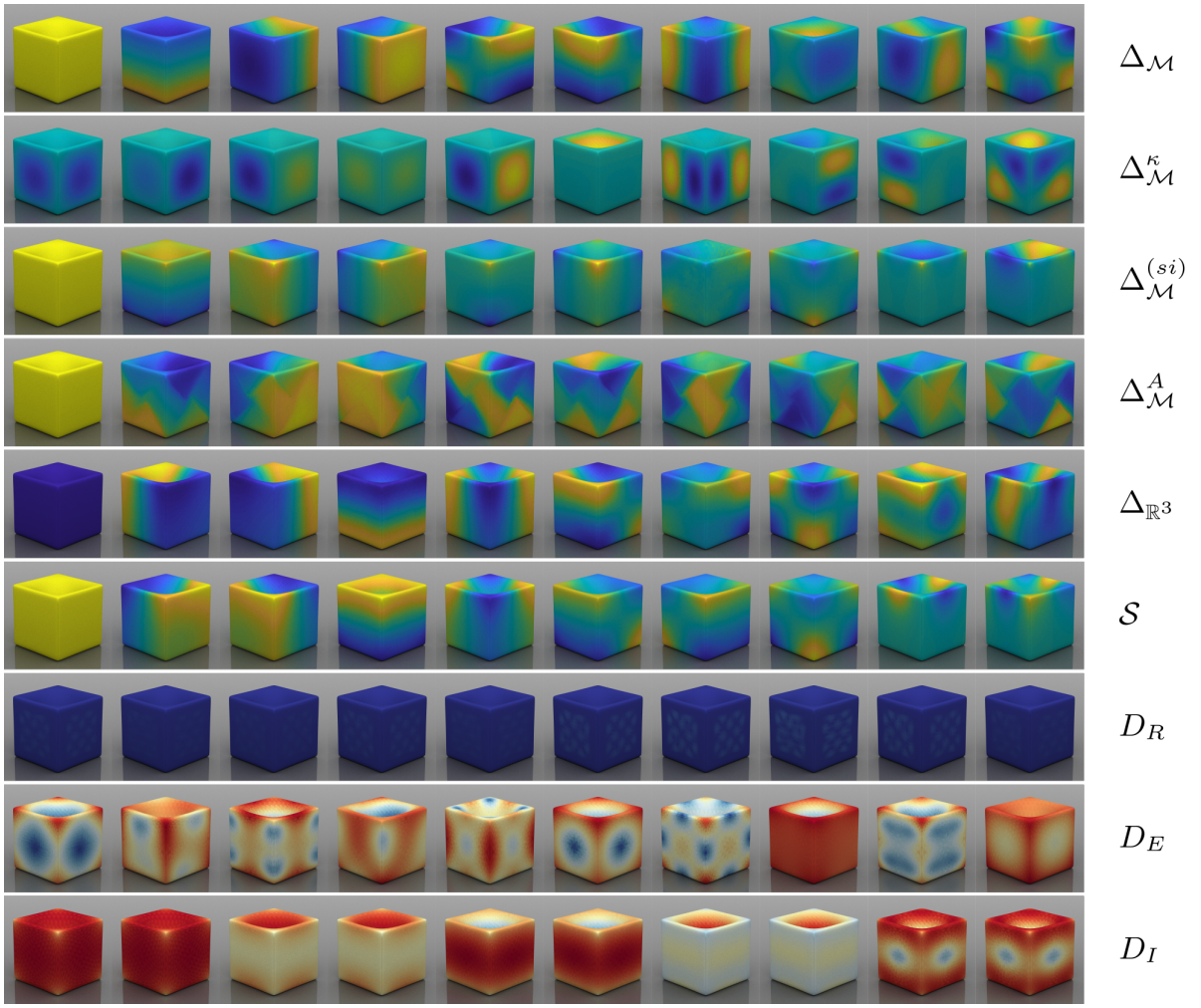


Figure 10: First few eigenfunctions computed on the cube with an inward bump. For the Dirac operators, norms of the (quaternion valued) eigenfunctions are visualized.

Additional exemplar shapes are provided in Figure 11 and 12, in which we illustrate eigenfunctions on a bowl with an inward or outward bumps. Compared to the cube in which the mean curvature changes in an extreme fashion, the bowls have smooth curvature varying to a lesser extent across the shape. Consequently, the bowl's Laplacian eigenfunctions become more similar to eigenfunctions of the modified Dirichlet and the scale invariant Laplacian. The relative Dirac operator vanishes on purely flat regions in the cubes, obstructing a visual comparison; the bowls provide examples that do not contain such regions, and hence we see the relative Dirac eigenfunctions have patterns aligning to ridges of the bumps, resonating most at nearly-flat plateaus.

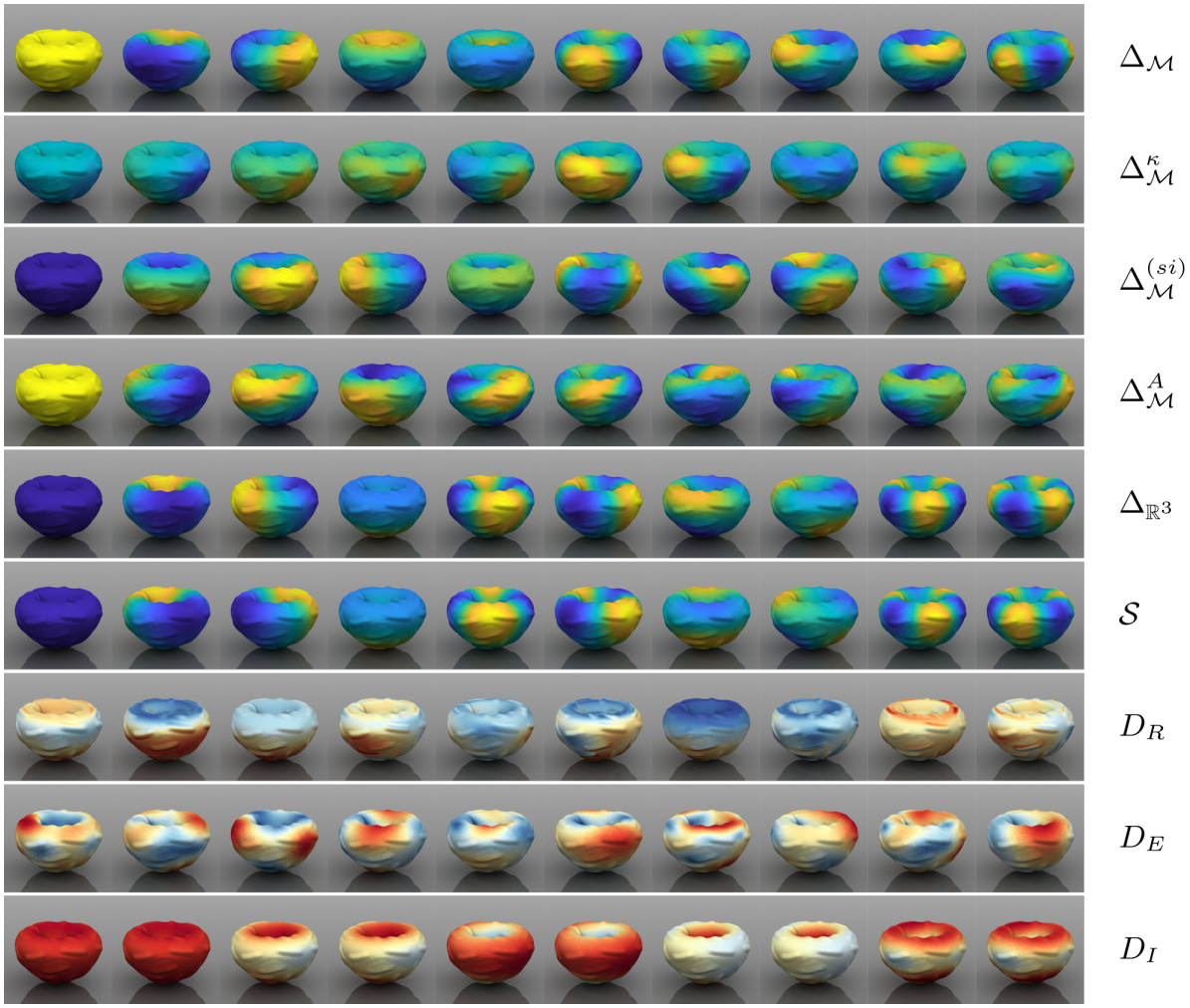


Figure 11: Eigenfunctions computed on a bowl with an inward bump.

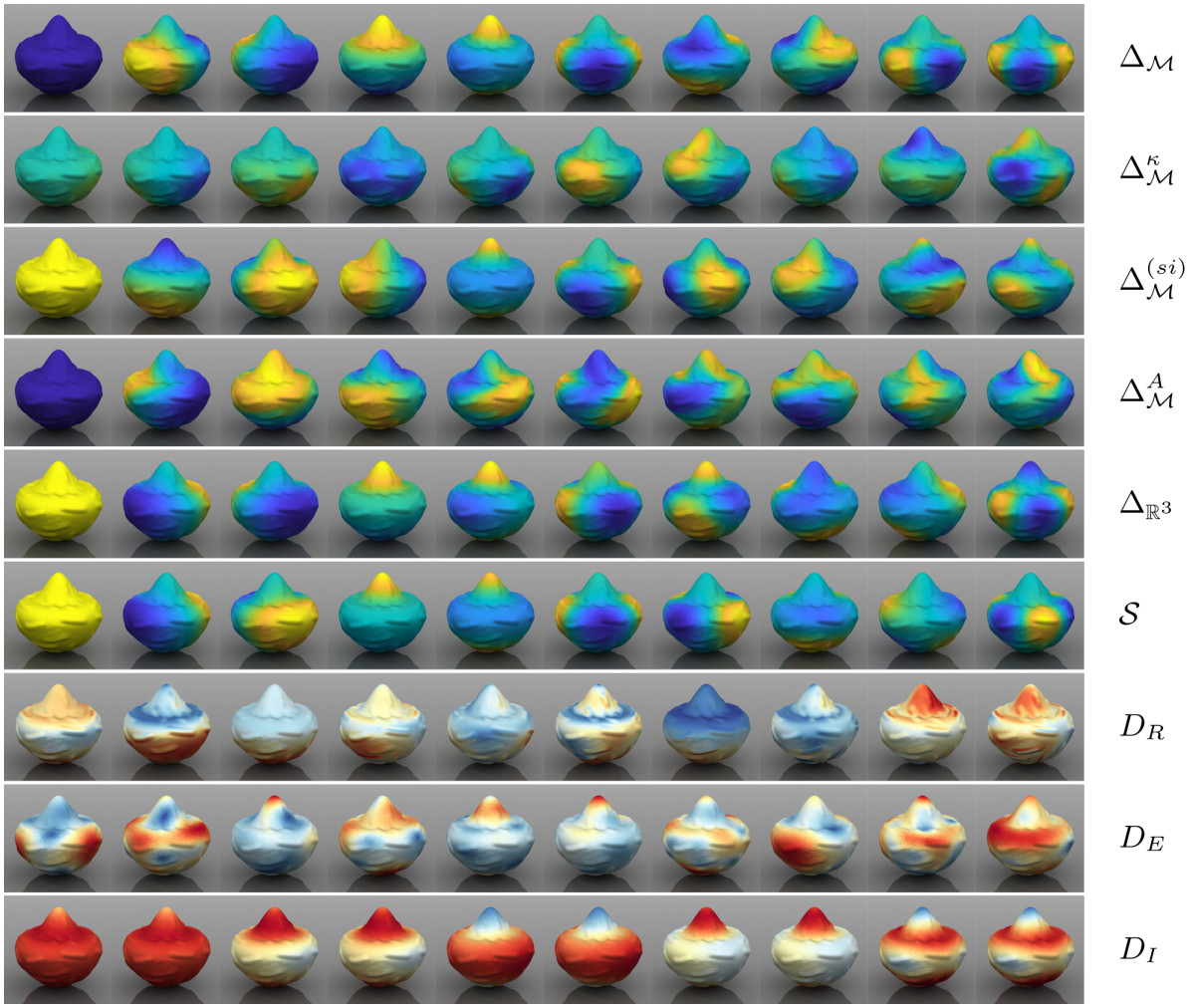


Figure 12: Eigenfunctions computed on a bowl with an outward bump.

Figure 13 visualizes eigenfunctions computed on the gargoyle model, at a few representative frequencies. For all operators, eigenfunctions with larger eigenvalues have denser nodal sets. The gargoyle model also has a relative flat region at the bottom, making relative Dirac eigenfunctions concentrate in the bottom.

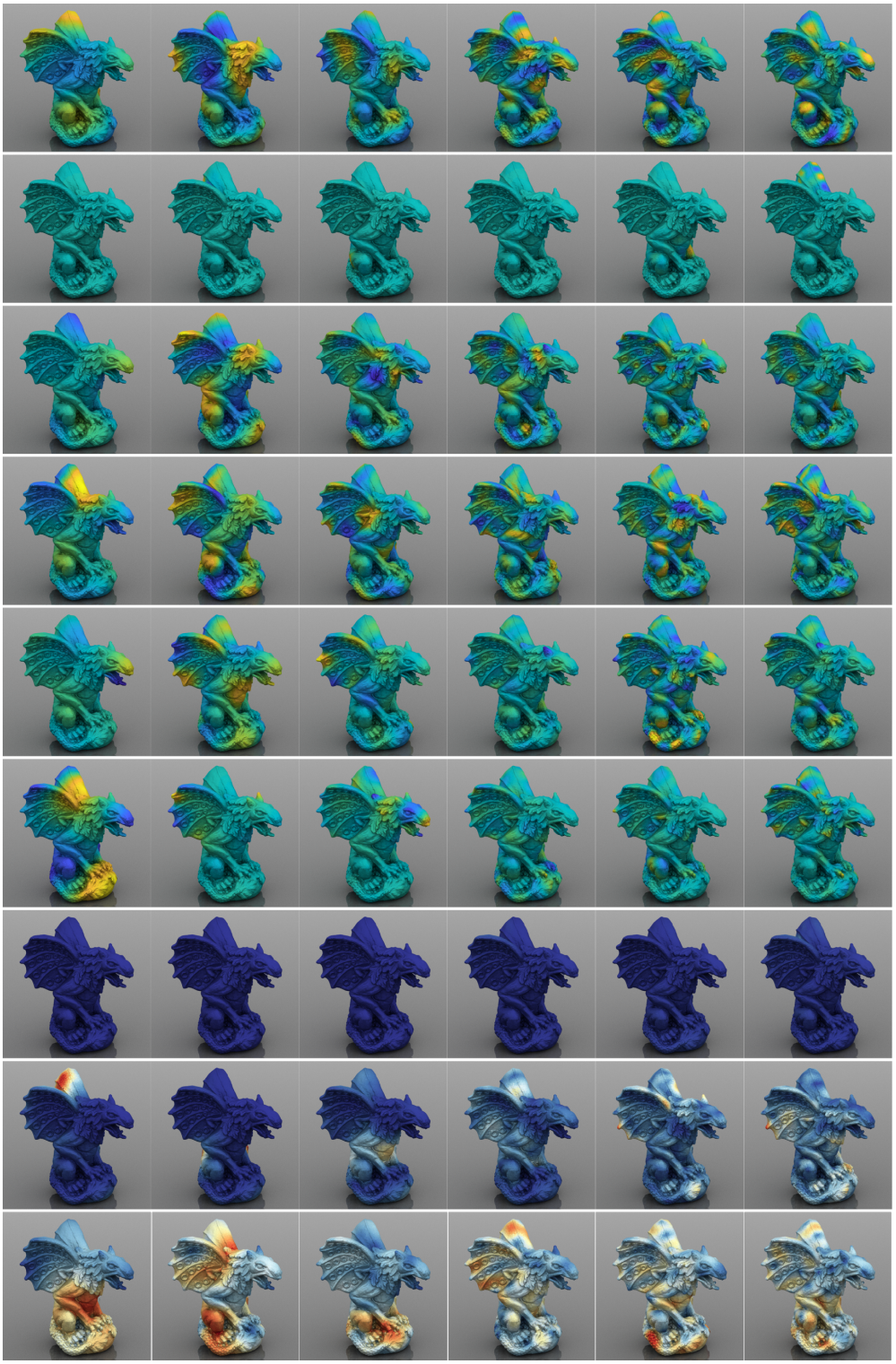


Figure 13: The 5^{th}, 10^{th}, 20^{th}, 40^{th}, 80^{th}, 100^{th}, 120^{th} eigenfunctions computed on the gargoyle model.

Since visually interpreting and extracting information from the eigenfunctions may not always be easy, we now start to look at quantities derived from eigenfunctions and eigenvalues.

6.3 Heat Kernel Signatures

We compute heat kernel signatures (HKS) using the definition in Eq. 23, using either a small time parameter $t = \frac{4 \log 10}{10 \lambda_{120}}$, or a large time parameter $t = \frac{4 \log 10}{\lambda_5}$. We set $k = 120$, i.e., using the first 120 eigenfunctions and eigenvalues. The resulting heat kernel signatures are shown in Figure 14. For HKS

with a small time parameter, the operators commonly put emphasis on regions where there is large Gaussian curvature, such as the horns in the gargoyle model or the fingertips in the hand model. The effect of adding the principal curvature term in the modified Dirichlet energy $\Delta^{\kappa}_{\mathcal{M}}$ is most visible in the gargoyle model: the HKS has a large value at edges of wings in the model, which are places with large mean curvatures. Such places with large mean curvatures are also promoted by the Steklov and volumetric Laplacian HKS with a small time parameter, but less significantly as in the modified Dirichlet HKS.

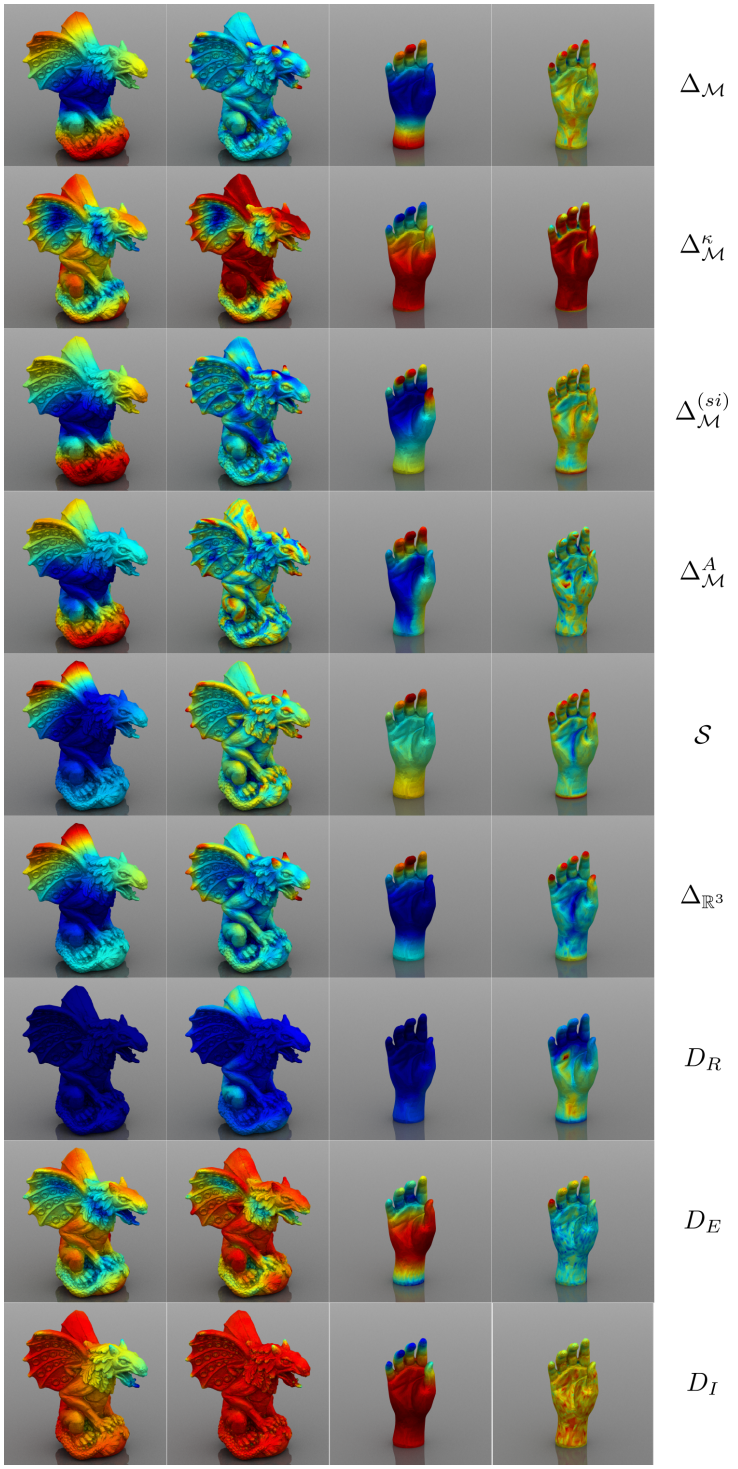


Figure 14: Heat kernel signatures in log scale, computed using either a small time (odd rows) or a large time (even rows) parameter.

6.4 Segmentation

We evaluate the results of spectral segmentation on the Princeton segmentation dataset [Chen et al., 2009] and provide visual comparisons of the results obtained by different operators. We first compute the harmonic embedding $\left(\frac{\phi_1(\mathbf{x})}{\sqrt{\lambda_1}}, \frac{\phi_2(\mathbf{x})}{\sqrt{\lambda_2}}, \cdots, \frac{\phi_k(\mathbf{x})}{\sqrt{\lambda_k}}\right)$, as introduced in Eq. 21, and then apply standard clustering algorithms to the points embedding in $\mathbb{R}^k$. We simply run k-means clustering, with a user-specified number of clusters. For the Dirac operators with quaternion-valued eigenfunctions, we use $\left(\frac{|\phi_1(\mathbf{x})|}{\sqrt{\lambda_1}}, \frac{|\phi_2(\mathbf{x})|}{\sqrt{\lambda_2}}, \cdots, \frac{|\phi_k(\mathbf{x})|}{\sqrt{\lambda_k}}\right)$ following [Ye et al., 2018]. We choose $k = 100$ in all experiments.

Figure 15 shows results of segmentation obtained by following this simple approach. We see that the Laplacian tends to produce segmentation avoiding regions with large Gaussian curvatures, such as ears in the teddy bear model. Volume-based operators tend to decompose shapes into volumetric parts, and the other extrinsic operators including modified Dirichlet energy and Dirac operators cut flat regions from the rest of the shape. An interesting example is the intrinsic Dirac operator, which tends to gather regions with similar geometry (e.g., all fingertips in the hand model) into the same component, different from other intrinsic operators. The relative Dirac operator segments the nearly-flat plateaus from other regions; in [Liu et al., 2017] a more sophisticated clustering algorithm other than the k-means is used to extract a segmentation.

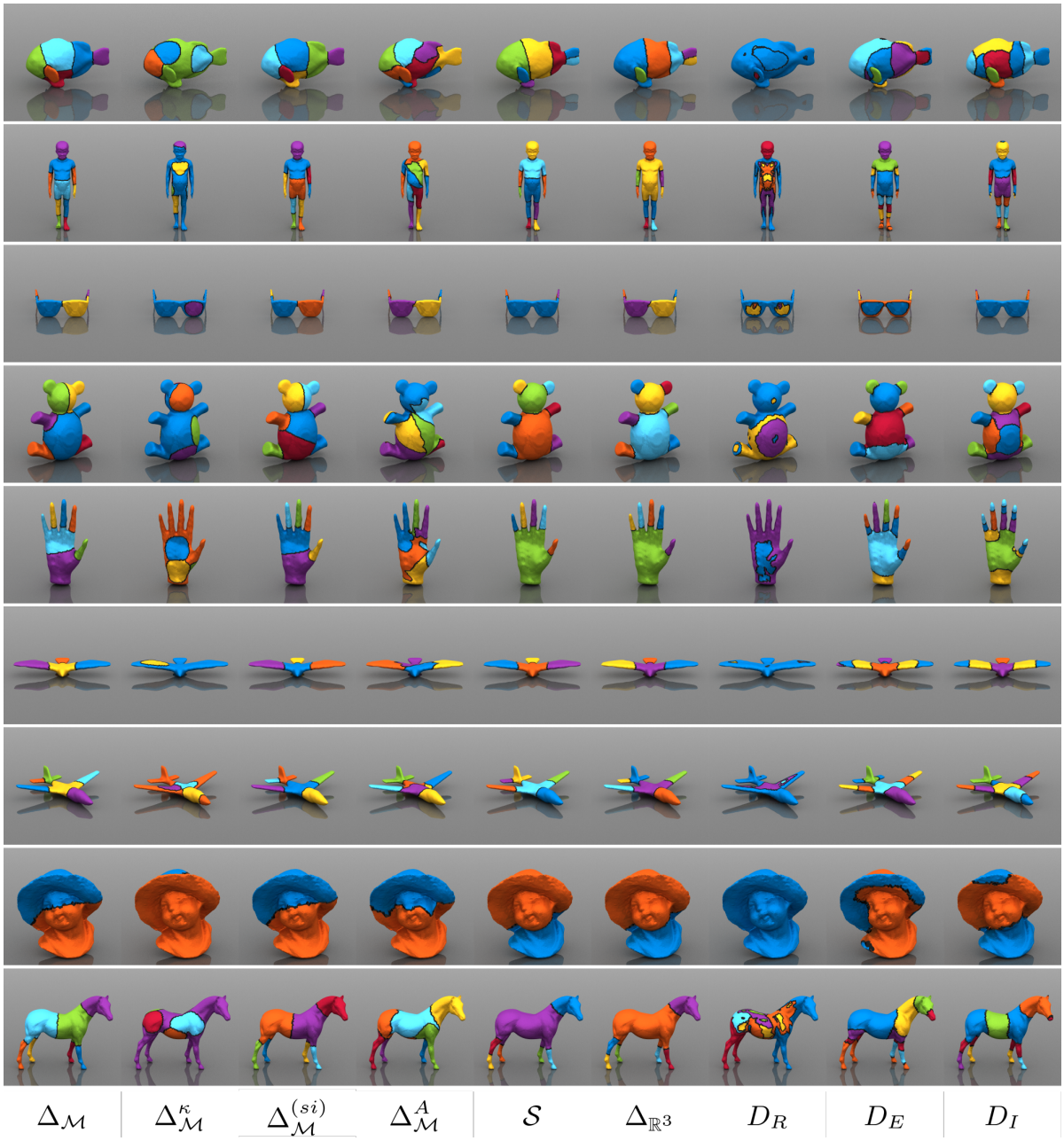


Figure 15: A comparison of segmentation results based on different operators.

6.5 Distance or Dissimilarity

In Figure 16 and 17, we show the diffusion distance and the "bi-operator" distance, which are generalizations of the diffusion and bi-harmonic distance for the Laplacian [Coifman and Lafon, 2006, Lipman et al., 2010] to other operators. Though the generalizations may not make perfect sense as distance measures for other operators, these distances can always be interpreted as dissimilarities of spectral embeddings between points on the surface. Indeed we see that distances in the same category, i.e., Laplacian-like operators (the Laplacian and the scale invariant Laplacian) or volumetric operators (the volumetric Laplacian and the Steklov) etc., tend to have similar patterns. In terms of differences, we can see that the scale invariant bi-distance is less evenly distributed in the geodesic sense, compared to the one for Laplacian, as a consequence of using a modified metric. For the anisotropic diffusion distance, as expected the resulting distances copy the anisotropic nature induced by the eigenfunctions. For the volumetric distances like ones based on the volumetric Laplacian or the DtN/Steklov operator, we see eigenfunctions can have local minimum on the surface. This is the correct behavior due to the volume-geodesic nature of the distances.

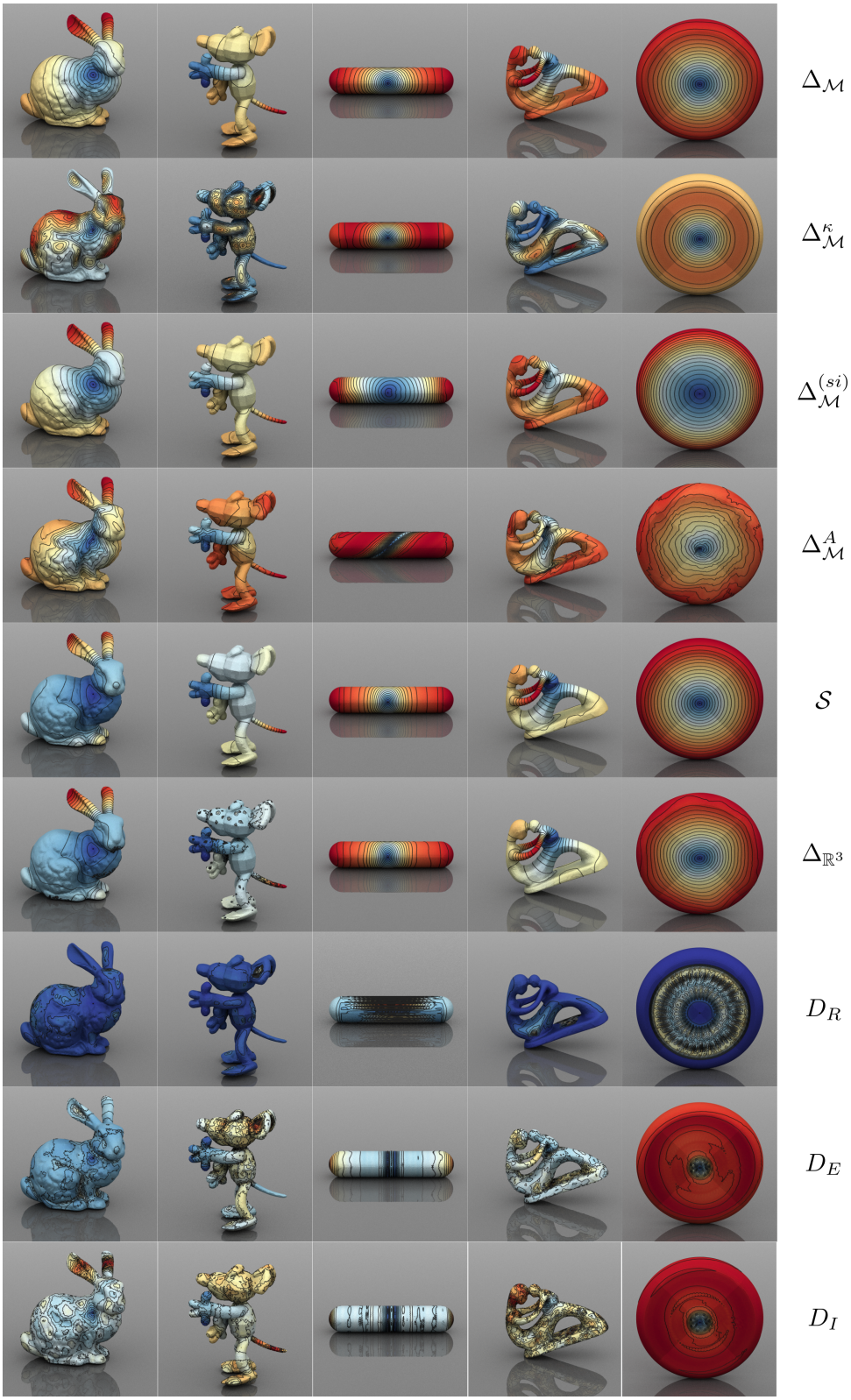


Figure 16: Diffusion distances computed on a variety of operators.

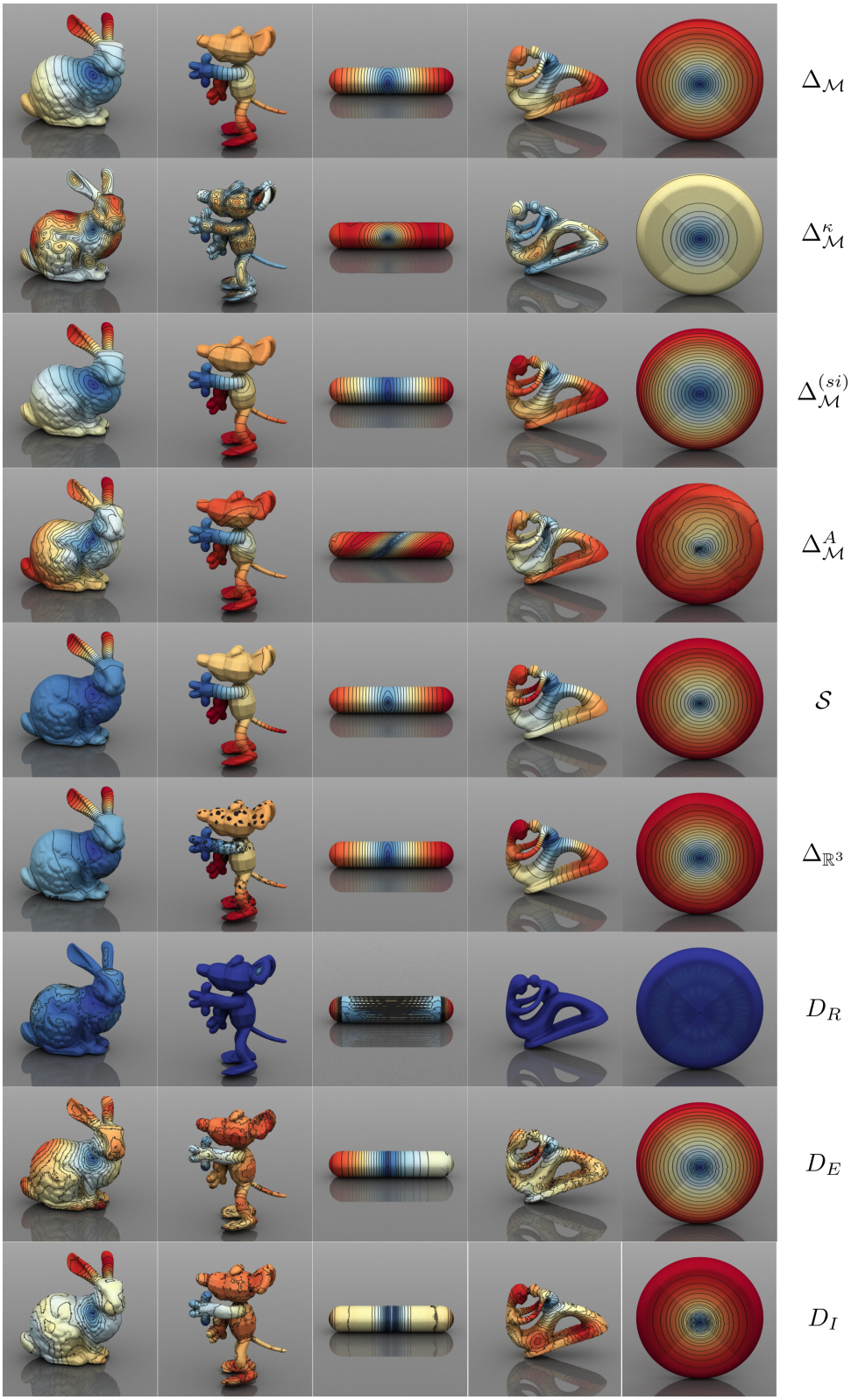


Figure 17: Bi-"operator" distances computed on a variety of operators.

7 Conclusion and Future Work

7.1 Summary

We have seen a few key distinguishing features among these geometric operators, either theoretically or empirically:

- *Laplacian and relatives.* Beyond the Laplacian, the scale invariant Laplacian, the equi-affine and affine invariant Laplacians, and the conformal Laplacian can be thought as the (regular) Laplacian on a "ghost manifold." That is, given an input surface, we first create an abstract manifold that is not necessarily embeddable by deforming the input surface, and then we compute the Laplacian on that manifold. Eigenfunctions of the new Laplacian, are pull backs of the regular Laplacian eigenfunctions, computed on the abstract manifold, onto the original surface.
- *Volumetric operators.* The DtN and volumetric Laplacian are defined on the volumetric domains. Although [Wang et al., 2018b] also propose a generalization of DtN to open surfaces, all experiments in this survey are performed on closed surface with a volumetric interior. Accordingly, the eigenfunctions have similar volume-aware patterns.
- *Dirac operators.* The definitions of Dirac operators involve the spin structure of a surface, making them unique from other operators. The spin structure allows the intrinsic Dirac to detect topological information to which Laplacians are insensitive [Ye et al., 2018].
- *Anisotropic operators.* Most operators we survey implicitly use the isotropic assumption. The anisotropic operator is uniquely characterized by an alternative choice, and its eigenfunctions, descriptors, and segmentation all demonstrate direction-sensitive patterns.

7.2 Future Work

The diverse and versatile set of operators discussed in this paper suggests possible directions for future work, presenting mathematical, algorithmic, and application-oriented challenges.

Better understanding of existing operators Most known theoretical results in spectral geometry are for the Laplacian operator and relatives, and theoretical results on other operators are not as prevalently available. While the Laplacian may be simply replaced by other operators, identifying most relevant substitutions for certain applications with theoretical justification provides many promising opportunities for future work.

Synthesizing new operators We have seen that different operators capture varying aspects of intrinsic and extrinsic geometry. Additively blending operators and multiplicatively composing operators provide a means of constructing new operators from existing ones. It would be interesting to create new operators by combining existing ones inside or outside this survey, and to better characterize the various means of composition. Some techniques used in one operator potentially can be applied to other operators, such as introducing scale invariant metrics, to derive an even larger collection of operators.

Task-dependent and learnable operators Identifying the most relevant operator for a certain task is another promising direction. While this may not be easy, in practice it is also possible to take a "bag-of-operators" approach, similar to the bag of descriptors method proposed in [Bronstein et al., 2011] to further improve the performance of shape analysis algorithms.

When exemplar data or outputs are available, the most effective operators can be learned. We have seen multiple algorithms that add learnable degrees of freedom to operators, and the emerging field of geometric deep learning has inspired considerable attention in this direction [Bronstein et al., 2017].

Acknowledgements

We would like to thank Sebastian Claici and Paul Zhang for proofreading, as well as the anonymous reviewers for their insightful comments. The authors acknowledge the generous support of Army Research Office grant W911NF-12-R-0011, National Science Foundation grant IIS-1838071, Air Force Office of Scientific Research award FA9550-19-1-0319, the MIT–IBM Watson AI Laboratory, the Toyota–CSAIL

Joint Research Center, and a gift from Adobe Systems. Part of this work was performed while the first author was visiting the Institute for Pure and Applied Mathematics (IPAM), supported by the National Science Foundation grant DMS-1440415. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these organizations.

References

[Aflalo et al., 2015] Aflalo, Y., Brezis, H., and Kimmel, R. (2015). On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences*, 8(2):1141–1160.

[Aflalo et al., 2011] Aflalo, Y., Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2011). Deformable shape retrieval by learning diffusion kernels. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 689–700. Springer.

[Aflalo et al., 2013] Aflalo, Y., Kimmel, R., and Raviv, D. (2013). Scale invariant geometry for nonrigid shapes. *SIAM Journal on Imaging Sciences*, 6(3):1579–1597.

[Alexa and Müller, 2000] Alexa, M. and Müller, W. (2000). Representing animations by principal components. In *Computer Graphics Forum*, volume 19, pages 411–418. Wiley Online Library.

[Ammann and Bär, 1998] Ammann, B. and Bär, C. (1998). The dirac operator on nilmanifolds and collapsing circle bundles. *Annals of Global Analysis and Geometry*, 16(3):221–253.

[Andreux et al., 2014] Andreux, M., Rodola, E., Aubry, M., and Cremers, D. (2014). Anisotropic laplace-beltrami operators for shape analysis. In *European Conference on Computer Vision*, pages 299–312. Springer.

[Atkinson and Han, 2005] Atkinson, K. and Han, W. (2005). Theoretical numerical analysis, volume 39. Springer.

[Au et al., 2012] Au, O. K.-C., Zheng, Y., Chen, M., Xu, P., and Tai, C.-L. (2012). Mesh segmentation with concavity-aware fields. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1125–1134.

[Aubry et al., 2011] Aubry, M., Schlickewei, U., and Cremers, D. (2011). The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV Workshops*, pages 1626–1633.

[Azencot et al., 2017] Azencot, O., Corman, E., Ben-Chen, M., and Ovsjanikov, M. (2017). Consistent functional cross field design for mesh quadrangulation. *ACM Transactions on Graphics (TOG)*, 36(4):92.

[Azencot et al., 2015] Azencot, O., Ovsjanikov, M., Chazal, F., and Ben-Chen, M. (2015). Discrete derivatives of vector fields on surfaces–an operator approach. *Transactions on Graphics*, 34(3):29.

[Barbič and James, 2005] Barbič, J. and James, D. L. (2005). Real-time subspace integration for st. venant-kirchhoff deformable models. In *ACM transactions on graphics (TOG)*, volume 24, pages 982–990. ACM.

[Belkin and Niyogi, 2003] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396.

[Belkin and Niyogi, 2005] Belkin, M. and Niyogi, P. (2005). Towards a theoretical foundation for laplacian-based manifold methods. In *International Conference on Computational Learning Theory*, pages 486–500. Springer.

[Belkin et al., 2006] Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434.

[Belkin et al., 2008] Belkin, M., Sun, J., and Wang, Y. (2008). Discrete Laplace operator on meshed surfaces. In *Proc. SODA*, pages 278–287.

[Belkin et al., 2009] Belkin, M., Sun, J., and Wang, Y. (2009). Constructing Laplace operator from point clouds in $\mathbb{R}^d$. In *Proc. SODA*, pages 1031–1040.

[Ben-Chen and Gotsman, 2005] Ben-Chen, M. and Gotsman, C. (2005). On the optimality of spectral compression of mesh data. *ACM Transactions on Graphics (TOG)*, 24(1):60–80.

[Bharaj et al., 2015] Bharaj, G., Levin, D. I., Tompkin, J., Fei, Y., Pfister, H., Matusik, W., and Zheng, C. (2015). Computational design of metallophone contact sounds. *ACM Transactions on Graphics (TOG)*, 34(6):223.

[Bommes et al., 2013] Bommes, D., Lévy, B., Pietroni, N., Puppo, E., Silva, C., Tarini, M., and Zorin, D. (2013). Quad-mesh generation and processing: A survey. In *Computer Graphics Forum*, volume 32, pages 51–76. Wiley Online Library.

[Booss, 2012] Booss, B. (2012). Topology and analysis: the atiyah-singer index formula and gauge-theoretic physics. Springer Science & Business Media.

[Börm et al., 2003] Börm, S., Grasedyck, L., and Hackbusch, W. (2003). Hierarchical matrices. Technical report, Max Planck Institute.

[Boscaini et al., 2015] Boscaini, D., Eynard, D., Kourounis, D., and Bronstein, M. M. (2015). Shape-from-operator: Recovering shapes from intrinsic operators. In *Computer Graphics Forum*, volume 34, pages 265–274.

[Boscaini et al., 2016a] Boscaini, D., Masci, J., Rodolà, E., and Bronstein, M. (2016a). Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197.

[Boscaini et al., 2016b] Boscaini, D., Masci, J., Rodolà, E., Bronstein, M. M., and Cremers, D. (2016b). Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441. Wiley Online Library.

[Botsch et al., 2005] Botsch, M., Bommes, D., and Kobbelt, L. (2005). Efficient linear system solvers for mesh processing. In *Mathematics of Surfaces XI*, pages 62–83. Springer.

[Botsch and Kobbelt, 2004] Botsch, M. and Kobbelt, L. (2004). An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics (TOG)*, 23(3):630–634.

[Botsch et al., 2010] Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., and Lévy, B. (2010). Polygon mesh processing. AK Peters/CRC Press.

[Brandt et al., 2017] Brandt, C., Scandolo, L., Eisemann, E., and Hildebrandt, K. (2017). Spectral processing of tangential vector fields. In *Computer Graphics Forum*, volume 36, pages 338–353. Wiley Online Library.

[Brandt et al., 2018] Brandt, C., Scandolo, L., Eisemann, E., and Hildebrandt, K. (2018). Modeling n-symmetry vector fields using higher-order energies. *ACM Transactions on Graphics (TOG)*, 37(2):18.

[Brenner and Scott, 2007] Brenner, S. and Scott, R. (2007). The mathematical theory of finite element methods, volume 15. Springer Science & Business Media.

[Brezis, 2010] Brezis, H. (2010). Functional analysis, sobolev spaces and partial differential equations. Springer Science & Business Media.

[Bronstein et al., 2016] Bronstein, A., Choukroun, Y., Kimmel, R., and Sela, M. (2016). Consistent discretization and minimization of the l1 norm on manifolds. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 435–440. IEEE.

[Bronstein et al., 2011] Bronstein, A. M., Bronstein, M. M., Guibas, L. J., and Ovsjanikov, M. (2011). Shape Google: Geometric words and expressions for invariant shape retrieval. *Transactions on Graphics*, 30(1):1.

[Bronstein et al., 2017] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

[Bronstein and Kokkinos, 2010] Bronstein, M. M. and Kokkinos, I. (2010). Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1704–1711. IEEE.

[Bruna et al., 2014] Bruna, J., Zaremba, W., Szlam, A., and Lecun, Y. (2014). Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*.

[Budninskiy et al., 2017] Budninskiy, M., Liu, B., Tong, Y., and Desbrun, M. (2017). Spectral affine-kernel embeddings. In *Computer Graphics Forum*, volume 36, pages 117–129. Wiley Online Library.

[Budninskiy et al., 2019] Budninskiy, M., Yin, G., Feng, L., Tong, Y., and Desbrun, M. (2019). Parallel transport unfolding: A connection-based manifold learning approach. *SIAM Journal on Applied Algebra and Geometry*, 3(2):266–291.

[Cayton, 2005] Cayton, L. (2005). Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17):1.

[Chadwick et al., 2009] Chadwick, J. N., An, S. S., and James, D. L. (2009). Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. *ACM Trans. Graph.*, 28(5):119–1.

[Chavel, 1984] Chavel, I. (1984). Eigenvalues in riemannian geometry, volume 115. Academic press.

[Chen et al., 2009] Chen, X., Golovinskiy, A., and Funkhouser, T. (2009). A benchmark for 3d mesh segmentation. In *Transactions on Graphics*, volume 28, page 73. ACM.

[Chern et al., 2018] Chern, A., Knöppel, F., Pinkall, U., and Schröder, P. (2018). Shape from metric. *ACM Transactions on Graphics (TOG)*, 37(4):63.

[Choukroun et al., 2018a] Choukroun, Y., Pai, G., and Kimmel, R. (2018a). Sparse approximation of 3d meshes using the spectral geometry of the hamiltonian operator. *Journal of Mathematical Imaging and Vision*, 60(6):941–952.

[Choukroun et al., 2018b] Choukroun, Y., Shtern, A., Bronstein, A. M., and Kimmel, R. (2018b). Hamiltonian operator for spectral shape analysis. *IEEE transactions on visualization and computer graphics*.

[Chuang et al., 2009] Chuang, M., Luo, L., Brown, B. J., Rusinkiewicz, S., and Kazhdan, M. (2009). Estimating the Laplace–Beltrami operator by restricting 3d functions. In *Computer Graphics Forum*, volume 28, pages 1475–1484.

[Chung and Graham, 1997] Chung, F. R. and Graham, F. C. (1997). Spectral graph theory. Number 92. American Mathematical Soc.

[Ciarlet, 2002] Ciarlet, P. G. (2002). The finite element method for elliptic problems, volume 40. Siam.

[Coifman et al., 1993] Coifman, R., Rokhlin, V., and Wandzura, S. (1993). The fast multipole method for the wave equation: A pedestrian prescription. *IEEE Antennas and Propagation Magazine*, 35(3):7–12.

[Coifman and Lafon, 2006] Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30.

[Corman et al., 2017a] Corman, E., Solomon, J., Ben-Chen, M., Guibas, L., and Ovsjanikov, M. (2017a). Functional characterization of intrinsic and extrinsic geometry. *ACM Transactions on Graphics (TOG)*, 36(2):14.

[Corman et al., 2017b] Corman, E., Solomon, J., Ben-Chen, M., Guibas, L., and Ovsjanikov, M. (2017b). Functional characterization of intrinsic and extrinsic geometry. *Transactions on Graphics*, 36(2):14:1–14:17.

[Cosmo et al., 2019] Cosmo, L., Panine, M., Rampini, A., Ovsjanikov, M., Bronstein, M. M., and Rodolà, E. (2019). Isospectralization, or how to hear shape, style, and correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7529–7538.

[Courant, 1923] Courant, R. (1923). Ein allgemeiner satzt zur theorie der eigenfunktionen selbsadjungierter differentialausdrücke. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1923:81–84.

[Craioveanu et al., 2013] Craioveanu, M.-E., Puta, M., and RASSIAS, T. (2013). Old and new aspects in spectral geometry, volume 534. Springer Science & Business Media.

[Crane et al., 2011] Crane, K., Pinkall, U., and Schröder, P. (2011). Spin transformations of discrete surfaces. In *ACM Transactions on Graphics (TOG)*, volume 30, page 104. ACM.

[Crane et al., 2013a] Crane, K., Pinkall, U., and Schröder, P. (2013a). Robust fairing via conformal curvature flow. *ACM Transactions on Graphics (TOG)*, 32(4):61.

[Crane et al., 2013b] Crane, K., Weischedel, C., and Wardetzky, M. (2013b). Geodesics in heat: A new approach to computing distance based on heat flow. *Transactions on Graphics*, 32(5):152.

[De Goes et al., 2008] De Goes, F., Goldenstein, S., and Velho, L. (2008). A hierarchical segmentation of articulated bodies. In *Computer graphics forum*, volume 27, pages 1349–1356. Wiley Online Library.

[Defferrard et al., 2016] Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852.

[Desbrun et al., 2005] Desbrun, M., Hirani, A. N., Leok, M., and Marsden, J. E. (2005). Discrete exterior calculus. *arXiv math/0508341*.

[Desbrun et al., 2002] Desbrun, M., Meyer, M., and Alliez, P. (2002). Intrinsic parameterizations of surface meshes. In *Computer graphics forum*, volume 21, pages 209–218. Wiley Online Library.

[Do Carmo, 2016] Do Carmo, M. P. (2016). Differential geometry of curves and surfaces: Revised and updated second edition. Courier Dover Publications.

[Dong et al., 2006] Dong, S., Bremer, P.-T., Garland, M., Pascucci, V., and Hart, J. C. (2006). Spectral surface quadrangulation. *Acm transactions on graphics (tog)*, 25(3):1057–1066.

[Donoho and Grimes, 2003] Donoho, D. L. and Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596.

[Drineas and Mahoney, 2005] Drineas, P. and Mahoney, M. W. (2005). On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(Dec):2153–2175.

[Dyer et al., 2007] Dyer, R., Zhang, H., Möller, T., and Clements, A. (2007). An investigation of the spectral robustness of mesh laplacians. Technical report, Simon Fraser University.

[Dziuk and Elliott, 2013] Dziuk, G. and Elliott, C. M. (2013). Finite element methods for surface pdes. *Acta Numerica*, 22:289–396.

[Eck et al., 1995] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. (1995). Multiresolution analysis of arbitrary meshes. In *Siggraph*, volume 95, pages 173–182.

[Evans, 1998] Evans, L. C. (1998). Partial differential equations. *Graduate studies in mathematics*, 19(4):7.

[Fiedler, 1975] Fiedler, M. (1975). A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633.

[Fisher et al., 2007] Fisher, M., Schröder, P., Desbrun, M., and Hoppe, H. (2007). Design of tangent vector fields. In *ACM transactions on graphics (TOG)*, volume 26, page 56. ACM.

[Frankel, 2011] Frankel, T. (2011). The geometry of physics: an introduction. Cambridge university press.

[Gebal et al., 2009] Gebal, K., Bærentzen, J. A., Aanæs, H., and Larsen, R. (2009). Shape analysis using the auto diffusion function. In *Computer Graphics Forum*, volume 28, pages 1405–1413. Wiley Online Library.

[Girouard and Polterovich, 2017] Girouard, A. and Polterovich, I. (2017). Spectral geometry of the Steklov problem. *J. Spectral Theory*, 7(2):321–359.

[Golub and Van Loan, 2012] Golub, G. H. and Van Loan, C. F. (2012). Matrix computations, volume 3. JHU press.

[Gordon et al., 1992a] Gordon, C., Webb, D., and Wolpert, S. (1992a). Isospectral plane domains and surfaces via riemannian orbifolds. *Inventiones mathematicae*, 110(1):1–22.

[Gordon et al., 1992b] Gordon, C., Webb, D. L., and Wolpert, S. (1992b). One cannot hear the shape of a drum. *Bulletin of the American Mathematical Society*, 27(1):134–138.

[Gotsman, 2003] Gotsman, C. (2003). On graph partitioning, spectral analysis, and digital mesh processing. In *2003 Shape Modeling International.*, pages 165–171. IEEE.

[Gotsman et al., 2003] Gotsman, C., Gu, X., and Sheffer, A. (2003). Fundamentals of spherical parameterization for 3d meshes. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 358–363. ACM.

[Guskov et al., 1999] Guskov, I., Sweldens, W., and Schröder, P. (1999). Multiresolution signal processing for meshes. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 325–334. Citeseer.

[Halimi and Kimmel, 2018] Halimi, O. and Kimmel, R. (2018). Self functional maps. In *2018 International Conference on 3D Vision (3DV)*, pages 710–718. IEEE.

[Halimi et al., 2019] Halimi, O., Litany, O., Rodola, E., Bronstein, A. M., and Kimmel, R. (2019). Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379.

[Hauser et al., 2003] Hauser, K. K., Shen, C., and O'Brien, J. F. (2003). Interactive deformation using modal analysis with constraints. In *Graphics Interface*, volume 3, pages 16–17.

[Hein et al., 2005] Hein, M., Audibert, J.-Y., and Von Luxburg, U. (2005). From graphs to manifolds–weak and strong pointwise consistency of graph laplacians. In *International Conference on Computational Learning Theory*, pages 470–485. Springer.

[Henrot, 2017] Henrot, A. (2017). Shape optimization and spectral theory. De Gruyter Open.

[Hildebrandt et al., 2011] Hildebrandt, K., Schulz, C., Tycowicz, C. V., and Polthier, K. (2011). Interactive surface modeling using modal analysis. *ACM Transactions on Graphics (TOG)*, 30(5):119.

[Hildebrandt et al., 2010] Hildebrandt, K., Schulz, C., von Tycowicz, C., and Polthier, K. (2010). Eigenmodes of surface energies for shape analysis. In *GMP*, pages 296–314. Springer.

[Hildebrandt et al., 2012] Hildebrandt, K., Schulz, C., von Tycowicz, C., and Polthier, K. (2012). Modal shape analysis beyond laplacian. *Computer Aided Geometric Design*, 29(5):204–218.

[Hirani, 2003] Hirani, A. N. (2003). Discrete exterior calculus. PhD thesis, California Institute of Technology.

[Hoffmann and Ye, 2018] Hoffmann, T. and Ye, Z. (2018). A discrete extrinsic and intrinsic dirac operator. *arXiv preprint arXiv:1802.06278*.

[Holst, 2001] Holst, M. (2001). Adaptive numerical treatment of elliptic systems on manifolds. *Advances in Computational Mathematics*, 15(1-4):139–191.

[Hou and Zhang, 2007] Hou, X. and Zhang, L. (2007). Saliency detection: A spectral residual approach. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. Ieee.

[Huang et al., 2009] Huang, Q.-X., Wicke, M., Adams, B., and Guibas, L. (2009). Shape decomposition using modal analysis. In *Computer Graphics Forum*, volume 28, pages 407–416. Wiley Online Library.

[Huang et al., 2019] Huang, R., Rakotosaona, M.-J., Achlioptas, P., Guibas, L., and Ovsjanikov, M. (2019). Operatornet: Recovering 3d shapes from difference operators. *arXiv preprint arXiv:1904.10754*.

[Jacobson, 2013] Jacobson, A. (2013). Algorithms and interfaces for real-time deformation of 2d and 3d shapes. PhD thesis, ETH Zurich.

[Jacobson et al., 2011] Jacobson, A., Baran, I., Popovic, J., and Sorkine, O. (2011). Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4):78–1.

[Jacobson et al., 2014] Jacobson, A., Deng, Z., Kavan, L., and Lewis, J. (2014). Skinning: real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*, page 24. ACM.

[Jacobson et al., 2010] Jacobson, A., Tosun, E., Sorkine, O., and Zorin, D. (2010). Mixed finite elements for variational surface modeling. In *Computer Graphics Forum*, volume 29, pages 1565–1574.

[Joshi et al., 2007] Joshi, P., Meyer, M., DeRose, T., Green, B., and Sanocki, T. (2007). Harmonic coordinates for character articulation. *ACM Transactions on Graphics (TOG)*, 26(3):71.

[Kac, 1966] Kac, M. (1966). Can one hear the shape of a drum? *American Mathematical Monthly*, 73(4):1–23.

[Kamberov et al., 1996] Kamberov, G., Pedit, F., and Pinkall, U. (1996). Bonnet pairs and isothermic surfaces. *arXiv preprint dg-ga/9610006*.

[Karni and Gotsman, 2000] Karni, Z. and Gotsman, C. (2000). Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286. ACM Press/Addison-Wesley Publishing Co.

[Kazhdan et al., 2006] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7.

[Kim et al., 2012] Kim, V. G., Li, W., Mitra, N. J., DiVerdi, S., and Funkhouser, T. A. (2012). Exploring collections of 3d models using fuzzy correspondences. *ACM Trans. Graph.*, 31(4):54–1.

[Knight et al., 1981] Knight, F. B. et al. (1981). Essentials of brownian motion and diffusion. Number 18. American Mathematical Soc.

[Knyazev, 2001] Knyazev, A. V. (2001). Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541.

[Kobbelt, 1997] Kobbelt, L. (1997). Discrete fairing. In *Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces*, volume 97, pages 101–131.

[Kokkinos et al., 2012] Kokkinos, I., Bronstein, M. M., Litman, R., and Bronstein, A. M. (2012). Intrinsic shape context descriptors for deformable shapes. In *Proc. CVPR*, pages 159–166.

[Kolluri et al., 2004] Kolluri, R., Shewchuk, J. R., and O'Brien, J. F. (2004). Spectral surface reconstruction from noisy point clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 11–21. ACM.

[Kostrikov et al., 2018] Kostrikov, I., Jiang, Z., Panozzo, D., Zorin, D., and Joan, B. (2018). Surface networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*.

[Kovnatsky et al., 2013] Kovnatsky, A., Bronstein, M. M., Bronstein, A. M., Glashoff, K., and Kimmel, R. (2013). Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448. Wiley Online Library.

[Krishnan et al., 2013] Krishnan, D., Fattal, R., and Szeliski, R. (2013). Efficient preconditioning of laplacian matrices for computer graphics. *ACM Transactions on Graphics (TOG)*, 32(4):142.

[Kry et al., 2002] Kry, P. G., James, D. L., and Pai, D. K. (2002). Eigenskin: real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 153–159. ACM.

[Kunisch and Volkwein, 2002] Kunisch, K. and Volkwein, S. (2002). Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical analysis*, 40(2):492–515.

[Langlois et al., 2014] Langlois, T. R., An, S. S., Jin, K. K., and James, D. L. (2014). Eigenmode compression for modal sound models. *ACM Transactions on Graphics (TOG)*, 33(4):40.

[Lee, 2013] Lee, J. M. (2013). Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–31. Springer.

[Lehoucq et al., 1998] Lehoucq, R. B., Sorensen, D. C., and Yang, C. (1998). Arpack users' guide: solution of large-scale eigenvalue problems with implicitly restarted arnoldi methods, volume 6. Siam.

[Levitin et al., 2017] Levitin, M., Parnovski, L., Polterovich, I., and Sher, D. A. (2017). Sloshing, Steklov and corners i: Asymptotics of sloshing eigenvalues. *arXiv:1709.01891*.

[Levy, 2006] Levy, B. (2006). Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13. IEEE.

[Lian et al., 2013] Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoué, G., Van Nguyen, H., Ohbuchi, R., et al. (2013). A comparison of methods for non-rigid 3d shape retrieval. *Pattern Recognition*, 46(1):449–461.

[Liang and Zhao, 2013] Liang, J. and Zhao, H. (2013). Solving partial differential equations on point clouds. *SIAM Journal on Scientific Computing*, 35(3):A1461–A1486.

[Lipman et al., 2008] Lipman, Y., Levin, D., and Cohen-Or, D. (2008). Green coordinates. *ACM Transactions on Graphics (TOG)*, 27(3):78.

[Lipman et al., 2010] Lipman, Y., Rustamov, R. M., and Funkhouser, T. A. (2010). Biharmonic distance. *Transactions on Graphics*, 29(3):27.

[Litany et al., 2017] Litany, O., Remez, T., Rodolà, E., Bronstein, A. M., and Bronstein, M. M. (2017). Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, pages 5660–5668.

[Litman and Bronstein, 2014] Litman, R. and Bronstein, A. M. (2014). Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180.

[Liu et al., 2015] Liu, B., Mason, G., Hodgson, J., Tong, Y., and Desbrun, M. (2015). Model-reduced variational fluid simulation. *ACM Transactions on Graphics (TOG)*, 34(6):244.

[Liu et al., 2017] Liu, H.-T. D., Jacobson, A., and Crane, K. (2017). A Dirac operator for extrinsic shape analysis. In *Computer Graphics Forum*, volume 36, pages 139–149. Wiley Online Library.

[Liu et al., 2019] Liu, H.-T. D., Jacobson, A., and Ovsjanikov, M. (2019). Spectral coarsening for geometric operators. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019, To Appear)*.

[Liu et al., 2006] Liu, R., Jain, V., and Zhang, H. (2006). Sub-sampling for efficient spectral mesh processing. In *Computer Graphics International Conference*, pages 172–184. Springer.

[Liu and Zhang, 2004] Liu, R. and Zhang, H. (2004). Segmentation of 3d meshes through spectral clustering. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, pages 298–305. IEEE.

[Liu and Zhang, 2007] Liu, R. and Zhang, H. (2007). Mesh segmentation via spectral embedding and contour analysis. In *Computer Graphics Forum*, volume 26, pages 385–394. Wiley Online Library.

[Mahadevan, 2007] Mahadevan, S. (2007). Adaptive mesh compression in 3d computer graphics using multiscale manifold learning. In *Proceedings of the 24th international conference on Machine learning*, pages 585–592. ACM.

[Masci et al., 2015] Masci, J., Boscaini, D., Bronstein, M., and Vandergheynst, P. (2015). Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45.

[Meila and Shi, 2001] Meila, M. and Shi, J. (2001). Learning segmentation by random walks. In *Advances in neural information processing systems*, pages 873–879.

[Melzi et al., 2018a] Melzi, S., Ovsjanikov, M., Roffo, G., Cristani, M., and Castellani, U. (2018a). Discrete time evolution process descriptor for shape analysis and matching. *ACM Transactions on Graphics (TOG)*, 37(1):4.

[Melzi et al., 2018b] Melzi, S., Rodolà, E., Castellani, U., and Bronstein, M. M. (2018b). Localized manifold harmonics for spectral shape analysis. In *Computer Graphics Forum*, volume 37, pages 20–34. Wiley Online Library.

[Meyer et al., 2003] Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. (2003). Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer.

[Mullen et al., 2008] Mullen, P., Tong, Y., Alliez, P., and Desbrun, M. (2008). Spectral conformal parameterization. In *Computer Graphics Forum*, volume 27, pages 1487–1494. Wiley Online Library.

[Nasikun et al., 2018] Nasikun, A., Brandt, C., and Hildebrandt, K. (2018). Fast approximation of laplace-beltrami eigenproblems. In *Computer Graphics Forum*, volume 37, pages 121–134. Wiley Online Library.

[Neumann et al., 2014] Neumann, T., Varanasi, K., Theobalt, C., Magnor, M., and Wacker, M. (2014). Compressed manifold modes for mesh processing. In *Computer Graphics Forum*, volume 33, pages 35–44. Wiley Online Library.

[Ng et al., 2002] Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856.

[O'Brien et al., 2002] O'Brien, J. F., Shen, C., and Gatchalian, C. M. (2002). Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 175–181. ACM.

[Osada et al., 2002] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2002). Shape distributions. *Transactions on Graphics*, 21(4):807–832.

[Ovsjanikov et al., 2012] Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., and Guibas, L. (2012). Functional maps: a flexible representation of maps between shapes. *Transactions on Graphics*, 31(4):30.

[Ovsjanikov et al., 2016] Ovsjanikov, M., Corman, E., Bronstein, M., Rodolà, E., Ben-Chen, M., Guibas, L., Chazal, F., and Bronstein, A. (2016). Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, page 9. ACM.

[Ovsjanikov et al., 2010] Ovsjanikov, M., Mérigot, Q., Mémoli, F., and Guibas, L. (2010). One point isometric matching with the heat kernel. In *Computer Graphics Forum*, volume 29, pages 1555–1564. Wiley Online Library.

[Ovsjanikov et al., 2008] Ovsjanikov, M., Sun, J., and Guibas, L. (2008). Global intrinsic symmetries of shapes. In *Computer graphics forum*, volume 27, pages 1341–1348. Wiley Online Library.

[Ozoliņš et al., 2013] Ozoliņš, V., Lai, R., Caflisch, R., and Osher, S. (2013). Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373.

[Patané, 2015] Patané, G. (2015). Volumetric heat kernel: Padé-Chebyshev approximation, convergence, and computation. *Computers & Graphics*, 46:64–71.

[Patané, 2017] Patané, G. (2017). Accurate and efficient computation of laplacian spectral distances and kernels. In *Computer Graphics Forum*, volume 36, pages 184–196. Wiley Online Library.

[Patanè, 2017] Patanè, G. (2017). An introduction to laplacian spectral distances and kernels: theory, computation, and applications. *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging*, 9(2):1–139.

[Patane et al., 2008] Patane, G., Spagnuolo, M., and Falcidieno, B. (2008). Reeb graph computation based on a minimal contouring. In *2008 IEEE International Conference on Shape Modeling and Applications*, pages 73–82. IEEE.

[Pauly and Gross, 2001] Pauly, M. and Gross, M. (2001). Spectral processing of point-sampled geometry. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 379–386. ACM.

[Pentland and Williams, 1989] Pentland, A. and Williams, J. (1989). Good vibrations: Modal dynamics for graphics and animation.

[Perelman, 2002] Perelman, G. (2002). The entropy formula for the ricci flow and its geometric applications. *arXiv preprint math/0211159*.

[Perelman, 2003a] Perelman, G. (2003a). Finite extinction time for the solutions to the ricci flow on certain three-manifolds. *arXiv preprint math.DG/0307245*.

[Perelman, 2003b] Perelman, G. (2003b). Ricci flow with surgery on three-manifolds, arxiv: math. *D6/0303109*.

[Pinkall and Polthier, 1993] Pinkall, U. and Polthier, K. (1993). Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36.

[Polterovich and Sher, 2015] Polterovich, I. and Sher, D. A. (2015). Heat invariants of the Steklov problem. *The Journal of Geometric Analysis*, 25(2):924–950.

[Qiu and Hancock, 2007] Qiu, H. and Hancock, E. R. (2007). Clustering and embedding using commute times. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1873–1890.

[Raviv et al., 2011a] Raviv, D., Bronstein, A. M., Bronstein, M. M., Kimmel, R., and Sochen, N. (2011a). Affine-invariant geodesic geometry of deformable 3d shapes. *Computers & Graphics*, 35(3):692–697.

[Raviv et al., 2014a] Raviv, D., Bronstein, A. M., Bronstein, M. M., Waisman, D., Sochen, N., and Kimmel, R. (2014a). Equi-affine invariant geometry for shape analysis. *Journal of mathematical imaging and vision*, 50(1-2):144–163.

[Raviv et al., 2010] Raviv, D., Bronstein, M. M., Bronstein, A. M., and Kimmel, R. (2010). Volumetric heat kernel signatures. In *Proc. 3DOR*, pages 39–44.

[Raviv et al., 2011b] Raviv, D., Bronstein, M. M., Bronstein, A. M., Kimmel, R., and Sochen, N. (2011b). Affine-invariant diffusion geometry for the analysis of deformable 3d shapes. In *CVPR 2011*, pages 2361–2367. IEEE.

[Raviv and Kimmel, 2015] Raviv, D. and Kimmel, R. (2015). Affine invariant geometry for non-rigid shapes. *International Journal of Computer Vision*, 111(1):1–11.

[Raviv et al., 2014b] Raviv, D., Lessick, J., and Raskar, R. (2014b). Evaluating local contractions from large deformations using affine invariant spectral geometry. In *International Workshop on Statistical Atlases and Computational Models of the Heart*, pages 147–157. Springer.

[Raviv and Raskar, 2015] Raviv, D. and Raskar, R. (2015). Scale invariant metrics of volumetric datasets. *SIAM Journal on Imaging Sciences*, 8(1):403–425.

[Ren et al., 2013] Ren, Z., Yeh, H., and Lin, M. C. (2013). Example-guided physically based modal sound synthesis. *ACM Transactions on Graphics (TOG)*, 32(1):1.

[Reuter et al., 2009] Reuter, M., Biasotti, S., Giorgi, D., Patanè, G., and Spagnuolo, M. (2009). Discrete Laplace–Beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390.

[Reuter et al., 2005] Reuter, M., Wolter, F.-E., and Peinecke, N. (2005). Laplace-spectra as fingerprints for shape matching. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 101–106. ACM.

[Reuter et al., 2006] Reuter, M., Wolter, F.-E., and Peinecke, N. (2006). Laplace–Beltrami spectra as 'shape-DNA' of surfaces and solids. *Computer-Aided Design*, 38(4):342–366.

[Rodolà et al., 2017] Rodolà, E., Cosmo, L., Bronstein, M. M., Torsello, A., and Cremers, D. (2017). Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library.

[Rong et al., 2008] Rong, G., Cao, Y., and Guo, X. (2008). Spectral mesh deformation. *The Visual Computer*, 24(7-9):787–796.

[Rosenberg, 1997] Rosenberg, S. (1997). The laplacian on a riemannian manifold: an introduction to analysis on manifolds, volume 31. Cambridge University Press.

[Rosman et al., 2010] Rosman, G., Bronstein, M. M., Bronstein, A. M., and Kimmel, R. (2010). Nonlinear dimensionality reduction by topologically constrained isometric embedding. *International Journal of Computer Vision*, 89(1):56–68.

[Roufosse and Ovsjanikov, 2018] Roufosse, J.-M. and Ovsjanikov, M. (2018). Unsupervised deep learning for structured shape matching. *arXiv preprint arXiv:1812.03794*.

[Roweis and Saul, 2000] Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

[Rustamov, 2007] Rustamov, R. M. (2007). Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233. Eurographics Association.

[Rustamov, 2009] Rustamov, R. M. (2009). On mesh editing, manifold learning, and diffusion wavelets. In *IMA International Conference on Mathematics of Surfaces*, pages 307–321. Springer.

[Rustamov, 2011] Rustamov, R. M. (2011). Interpolated eigenfunctions for volumetric shape processing. *The Visual Computer*, 27(11):951–961.

[Rustamov et al., 2013] Rustamov, R. M., Ovsjanikov, M., Azencot, O., Ben-Chen, M., Chazal, F., and Guibas, L. (2013). Map-based exploration of intrinsic shape differences and variability. *Transactions on Graphics*, 32(4):72.

[Saad, 2011] Saad, Y. (2011). Numerical methods for large eigenvalue problems: revised edition, volume 66. Siam.

[Scarselli et al., 2009] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

[Schonsheck et al., 2018] Schonsheck, S. C., Bronstein, M. M., and Lai, R. (2018). Nonisometric surface registration via conformal laplace-beltrami basis pursuit. *arXiv preprint arXiv:1809.07399*.

[Schwartz et al., 1989] Schwartz, E. L., Shaw, A., and Wolfson, E. (1989). A numerical solution to the generalized mapmaker. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9):1005–1008.

[Sela et al., 2015] Sela, M., Aflalo, Y., and Kimmel, R. (2015). Computational caricaturization of surfaces. *Computer Vision and Image Understanding*, 141:1–17.

[Sharp et al., 2018] Sharp, N., Soliman, Y., and Crane, K. (2018). The vector heat method. *arXiv preprint arXiv:1805.09170*.

[Shawe-Taylor et al., 2004] Shawe-Taylor, J., Cristianini, N., et al. (2004). Kernel methods for pattern analysis. Cambridge university press.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

[Shi et al., 2008] Shi, Y., Lai, R., Krishna, S., Sicotte, N., Dinov, I., and Toga, A. W. (2008). Anisotropic laplace-beltrami eigenmaps: Bridging reeb graphs and skeletons. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7. IEEE.

[Shi and Sun, 2017] Shi, Z. and Sun, J. (2017). Convergence of the point integral method for laplace–beltrami equation on point cloud. *Research in the Mathematical Sciences*, 4(1):22.

[Singer and Wu, 2012] Singer, A. and Wu, H.-T. (2012). Vector diffusion maps and the connection laplacian. *Communications on pure and applied mathematics*, 65(8):1067–1144.

[Śmigaj et al., 2015] Śmigaj, W., Betcke, T., Arridge, S., Phillips, J., and Schweiger, M. (2015). Solving boundary integral problems with BEM++. *Transactions on Mathematical Software*, 41(2):6.

[Song et al., 2014] Song, R., Liu, Y., Martin, R. R., and Rosin, P. L. (2014). Mesh saliency via spectral processing. *ACM Transactions on Graphics (TOG)*, 33(1):6.

[Sorkine, 2005] Sorkine, O. (2005). Laplacian mesh processing. In *Eurographics State of the Art Report*.

[Sorkine, 2006] Sorkine, O. (2006). Differential representations for mesh processing. In *Computer Graphics Forum*, volume 25, pages 789–807. Wiley Online Library.

[Sorkine et al., 2004] Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., and Seidel, H.-P. (2004). Laplacian surface editing. In *Proc. SGP*, pages 175–184.

[Spagnuolo et al., 2012] Spagnuolo, M., Bronstein, M., Bronstein, A., and Ferreira, A. (2012). Affine-invariant photometric heat kernel signatures. *The Eurographics Association*, pages 39–46.

[Spielman and Teng, 1996] Spielman, D. A. and Teng, S.-H. (1996). Spectral partitioning works: Planar graphs and finite element meshes. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 96–105. IEEE.

[Spielman and Teng, 2004] Spielman, D. A. and Teng, S.-H. (2004). Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the STOC*, volume 4.

[Stein et al., 2018] Stein, O., Grinspun, E., Wardetzky, M., and Jacobson, A. (2018). Natural boundary conditions for smoothing in geometry processing. *ACM Transactions on Graphics (TOG)*, 37(2):23.

[Stein et al., 2019] Stein, O., Jacobson, A., Wardetzky, M., and Grinspun, E. (2019). A smoothness energy without boundary distortion for curved surfaces. *arXiv preprint arXiv:1905.09777*.

[Steinbach, 2007] Steinbach, O. (2007). Numerical approximation methods for elliptic boundary value problems: finite and boundary elements. Springer.

[Sun et al., 2009] Sun, J., Ovsjanikov, M., and Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum*, volume 28, pages 1383–1392.

[Sun and Zhou, 2016] Sun, J. and Zhou, A. (2016). Finite element methods for eigenvalue problems. Chapman and Hall/CRC.

[Taubin, 1995] Taubin, G. (1995). A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM.

[Taylor, 2013] Taylor, M. (2013). Partial differential equations ii: Qualitative studies of linear equations, volume 116. Springer Science & Business Media.

[Taylor, 2011] Taylor, M. E. (2011). Partial differential equations i: Basic theory (applied mathematical sciences).

[Tenenbaum et al., 2000] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

[Thomas and Natarajan, 2014] Thomas, D. M. and Natarajan, V. (2014). Multiscale symmetry detection in scalar fields by clustering contours. *IEEE transactions on visualization and computer graphics*, 20(12):2427–2436.

[Tombari et al., 2010] Tombari, F., Salti, S., and Di Stefano, L. (2010). Unique signatures of histograms for local surface description. In *Proc. ECCV*, pages 356–369.

[Tong et al., 2003] Tong, Y., Lombeyda, S., Hirani, A. N., and Desbrun, M. (2003). Discrete multiscale vector field decomposition. In *ACM transactions on graphics (TOG)*, volume 22, pages 445–452. ACM.

[Vallet and Lévy, 2008] Vallet, B. and Lévy, B. (2008). Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260.

[van de Doel and Pai, 1996] van de Doel, K. and Pai, D. K. (1996). Synthesis of shape dependent sounds with physical modeling. Georgia Institute of Technology.

[van de Doel and Pai, 1998] van de Doel, K. and Pai, D. K. (1998). The sounds of physical shapes. *Presence*, 7(4):382–395.

[Varadhan, 1967] Varadhan, S. R. S. (1967). On the behavior of the fundamental solution of the heat equation with variable coefficients. *Communications on Pure and Applied Mathematics*, 20(2):431–455.

[Vaxman et al., 2010] Vaxman, A., Ben-Chen, M., and Gotsman, C. (2010). A multi-resolution approach to heat kernels on discrete surfaces. In *ACM Transactions on Graphics (TOG)*, volume 29, page 121. ACM.

[Wang and Wang, 2015] Wang, G. and Wang, Y. (2015). Multi-scale heat kernel based volumetric morphology signature. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 751–759.

[Wang et al., 2014] Wang, H., Lu, T., Au, O. K.-C., and Tai, C.-L. (2014). Spectral 3d mesh segmentation with a novel single segmentation field. *Graphical Models*, 76(5):440–456.

[Wang et al., 2018a] Wang, L., Gehre, A., Bronstein, M. M., and Solomon, J. (2018a). Kernel functional maps. In *Computer Graphics Forum*, volume 37, pages 27–36. Wiley Online Library.

[Wang et al., 2018b] Wang, Y., Ben-Chen, M., Polterovich, I., and Solomon, J. (2018b). Steklov spectral geometry for extrinsic shape analysis. *ACM Transactions on Graphics (TOG)*, 37(7):21.

[Wang et al., 2015] Wang, Y., Jacobson, A., Barbič, J., and Kavan, L. (2015). Linear subspace design for real-time shape deformation. *ACM Transactions on Graphics (TOG)*, 34(4):57.

[Wang et al., 2019] Wang, Y., Kim, V., Bronstein, M., and Solomon, J. (2019). Learning geometric operators on meshes. In *7th International Conference on Learning Representations (ICLR 2019) Workshop on Representation Learning on Graphs and Manifolds.*

[Wang et al., 2012] Wang, Y., Liu, B., and Tong, Y. (2012). Linear surface reconstruction from discrete fundamental forms on triangle meshes. In *Computer Graphics Forum*, volume 31, pages 2277–2287. Wiley Online Library.

[Wardetzky et al., 2007] Wardetzky, M., Mathur, S., Kälberer, F., and Grinspun, E. (2007). Discrete Laplace operators: no free lunch. In *Symposium on Geometry processing*, pages 33–37.

[Williams, 2002] Williams, C. K. (2002). On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19.

[Williams and Seeger, 2001] Williams, C. K. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688.

[Wu, 2013] Wu, H.-T. (2013). Embedding riemannian manifolds by the heat kernel of the connection laplacian. *arXiv preprint arXiv:1305.4232*.

[Wu and Levine, 1997] Wu, K. and Levine, M. D. (1997). 3d part segmentation using simulated electrical charge distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1223–1235.

[Ye et al., 2018] Ye, Z., Diamanti, O., Tang, C., Guibas, L., and Hoffmann, T. (2018). A unified discrete framework for intrinsic and extrinsic Dirac operators for geometry processing. In *Computer Graphics Forum*, volume 37, pages 93–106. Wiley Online Library.

[Yi et al., 2016] Yi, L., Su, H., Guo, X., and Guibas, L. (2016). Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *arXiv preprint arXiv:1612.00606*.

[Yoo et al., 2006] Yoo, H. H., Cho, J. E., and Chung, J. (2006). Modal analysis and shape optimization of rotating cantilever beams. *Journal of Sound and vibration*, 290(1-2):223–241.

[Yu et al., 2010] Yu, Y., Jang, I. G., Kim, I. K., and Kwak, B. M. (2010). Nodal line optimization and its application to violin top plate design. *Journal of Sound and Vibration*, 329(22):4785–4796.

[Yu et al., 2004] Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., and Shum, H.-Y. (2004). Mesh editing with poisson-based gradient field manipulation. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 644–651. ACM.

[Zahn and Roskies, 1972] Zahn, C. T. and Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on computers*, 100(3):269–281.

[Zelditch, 2000] Zelditch, S. (2000). Spectral determination of analytic bi-axisymmetric plane domains. *Geometric & Functional Analysis GAFA*, 10(3):628–677.

[Zelditch, 2009] Zelditch, S. (2009). Local and global analysis of eigenfunctions. *arXiv preprint arXiv:0903.3420*.

[Zeng et al., 2012] Zeng, W., Guo, R., Luo, F., and Gu, X. (2012). Discrete heat kernel determines discrete riemannian metric. *Graphical Models*, 74(4):121–129.

[Zhang, 2004] Zhang, H. (2004). Discrete combinatorial laplacian operators for digital geometry processing. In *Proceedings of SIAM Conference on Geometric Design and Computing. Nashboro Press*, pages 575–592.

[Zhang et al., 2010a] Zhang, H., Van Kaick, O., and Dyer, R. (2010a). Spectral mesh processing. In *Computer Graphics Forum*, volume 29, pages 1865–1894.

[Zhang et al., 2010b] Zhang, H., Van Kaick, O., and Dyer, R. (2010b). Spectral mesh processing. In *Computer graphics forum*, volume 29, pages 1865–1894. Wiley Online Library.

[Zhang and Zha, 2004] Zhang, Z. and Zha, H. (2004). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing*, 26(1):313–338.

[Zhong and Qin, 2014] Zhong, M. and Qin, H. (2014). Sparse approximation of 3d shapes via spectral graph wavelets. *The Visual Computer*, 30(6-8):751–761.

[Zhou et al., 2004] Zhou, K., Synder, J., Guo, B., and Shum, H.-Y. (2004). Iso-charts: stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 45–54. ACM.

[Zhu, 2007] Zhu, K. (2007). Operator theory in function spaces. Number 138. American Mathematical Soc.

[Zhu et al., 2003] Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.

[Zigelman et al., 2002] Zigelman, G., Kimmel, R., and Kiryati, N. (2002). Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):198–207.

[Zorin, 2005] Zorin, D. (2005). Curvature-based energy for simulation and variational modeling. In *International Conference on Shape Modeling and Applications 2005 (SMI'05)*, pages 196–204. IEEE.