Steklov Geometry Processing: An Extrinsic Approach to Spectral Shape Analysis

YU WANG, Massachusetts Institute of Technology
MIRELA BEN-CHEN, Technion – Israel Institute of Technology
IOSIF POLTEROVICH, Université de Montréal
JUSTIN SOLOMON, Massachusetts Institute of Technology

We propose Steklov geometry processing, an extrinsic approach to spectral geometry processing and shape analysis. Intrinsic approaches, usually based on the Laplace–Beltrami operator, cannot capture the spatial embedding of a shape up to rigid motion, while many previous extrinsic methods lack theoretical justification. Instead, we propose a systematic approach by considering the Steklov eigenvalue problem, computing the spectrum of the Dirichlet-to-Neumann operator of a surface bounding a volume. A remarkable property of this operator is that it encodes the volumetric geometry. We use the boundary element method (BEM) to discretize the operator, accelerated by hierarchical numerical schemes and preconditioning; this pipeline allows us to solve eigenvalue and linear problems on large-scale meshes despite the density of the Dirichlet-to-Neumann discretization. We further demonstrate that our operators naturally fit into existing frameworks for geometry processing, making a shift from intrinsic to extrinsic geometry as simple as substituting the Laplace–Beltrami operator with the Dirichlet-to-Neumann operator.

1 INTRODUCTION

Geometry processing and shape analysis tools for computer graphics typically draw from two complementary theories of geometry. To distinguish these, consider a closed surface embedded in three dimensions. From the viewpoint of extrinsic geometry, we might examine the surface as the outer boundary of a volume. This approach relies on distances and other measurements taken from the space surrounding the surface to understand its shape. Contrastingly, many techniques in differential geometry decouple extrinsic shape from intrinsic geometry, which is concerned with quantities like geodesic distances that can be measured without leaving the outer surface. A crowning achievement of classical differential geometry shows that certain quantities like Gaussian curvature can be measured intrinsically; this basic observation inspired theoretical exploration of purely intrinsic techniques. Computational geometry processing embraced the intrinsic perspective early on, leading to numerous applications of intrinsic computations.

Intrinsic geometry, however, is an ineffective description of shape for many applications. First, the spatial embedding information is lost. The variability in embedding can be essential. As an extreme example, consider searching a database of origami models: From an intrinsic perspective, all origami is equivalent to the same piece of flat paper. Second, the intrinsic perspective is a counterintuitive way to describe the shapes of many real-world objects, e.g., identifying the inward and outward bumps on the cubes in Figure 1.

A naïve approach to extrinsic geometry could be to use the \((x, y, z)\) coordinates of the embedding. This incorporates information about the embedding but is not invariant to rigid motion. Alternatively, rotation-invariant shape descriptors (e.g., built from spherical harmonic power spectra) usually involve extrinsic information about a surface as a shell rather than as the boundary of a volume.

In this paper, we provide a practical and mathematically-justified spectral approach to extrinsic geometry for geometry processing. We propose Steklov geometry processing, an extrinsic alternative to geometry processing using the intrinsic Laplace–Beltrami operator. Steklov geometry processing is a surface-only approach for volume-aware shape analysis, by using a boundary operator that takes the interaction between non-adjacent vertices into consideration. In particular, we consider the Dirichlet-to-Neumann (DtN) operator and its spectrum (the Steklov spectrum). A physical interpretation of the DtN operator is that it maps from a temperature distribution on the surface to the resulting heat flux through the surface. As we will prove, DtN and its spectrum encode complete extrinsic geometry. We also show that DtN and some of its peers can be applied efficiently using the boundary element method (BEM), accelerated using preconditioning and hierarchical techniques that scale linearly in the size of a boundary triangle mesh.

We generalize our method to additional geometric primitives, notably including surfaces with open boundary. Since it is built using the Laplace equation, our discretized Dirichlet-to-Neumann operator naturally fits into existing frameworks for geometry processing and analysis. We show that in applications including compression, measurement of shape differences, and spectral distance computation, shifting from intrinsic geometry to extrinsic geometry is as simple as substituting the cotangent Laplacian with a discretized Steklov operator. This simple change leads to qualitatively different behavior for geometric algorithms.
2 RELATED WORK

2.1 Intrinsic geometry

Intrinsic computations are ubiquitous in geometry processing, applied to problems including distance computation (Crane et al. 2013; Lipman et al. 2010), segmentation (Reuter et al. 2009), shape description (Kokkinos et al. 2012), shape retrieval (Bronstein et al. 2011), and correspondence (Ovsjanikov et al. 2012); these citations are a small sampling of the literature.

Many intrinsic algorithms use the Laplace–Beltrami operator, which can be approximated e.g. with the celebrated cotangent formula (Pinkall and Polthier 1993) or with convergent approximations on simplicial complexes (Belkin et al. 2008) and point clouds (Belkin et al. 2009; Liang et al. 2012). Laplace–Beltrami eigenfunctions are multiscale and can characterize shape up to isometry (Aubry et al. 2011; Sun et al. 2009); truncation in this basis also provides favorable approximation properties (Aflalo et al. 2015). See the surveys by Zhang et al. (2010), Sorkine (2005), and Patané (2016) for discussion of this technique and applications.

Functional maps, proposed by Ovsjanikov et al. (2012), have inspired interest in a broader ‘operator-based’ approach to geometry. Operator-based geometry processing now includes intrinsic techniques for shape exploration (Rustamov et al. 2013), vector field processing (Azenct et al. 2013, 2015), interpolation (Azenct et al. 2016), simulation (Azenct et al. 2014), and deformation (Boscaini et al. 2015)—among other tasks.

The intrinsic approach is desirable in many scenarios. Many real-world objects deform near-isometrically, that is, without affecting intrinsic structure. Isometry invariance then creates an advantage: An object can be bent to another pose, while still being considered the same intrinsically. For example, intrinsic invariance can be favorable in shape retrieval since the same object in different poses will be clustered together by design.

2.2 Extrinsic geometry

Extrinsic techniques appear in geometry processing, usually without the multiscale and stability advantages of spectral geometry. The most common expression of discrete extrinsic geometry is in computation of mean curvature, the extrinsic counterpart of Gaussian curvature. Extrinsic methods appear in computer vision, e.g. the SHOT (Tombari et al. 2010) and D2 (Osada et al. 2002) descriptors, typically without completeness guarantees.

A few methods attempt to extend operator-based methods to incorporate extrinsic information. The volumetric heat kernel signature (Raviv et al. 2010) utilizes a meshing of the volume enclosed by a surface, applied to segmentation by Litman et al. (2012); Wang and Wang (2015) presents a similar construction on tetrahedral meshes. This is computationally expensive and has the property that results depend on the resolution of the interior mesh; note that methods like (Patané 2015) may accelerate heat kernel evaluation but still require meshing of the interior. Rustamov (2011) proposes extending surface-based Laplace–Beltrami eigenfunctions to the interior of a domain using generalized barycentric coordinates. This extends descriptors like the heat kernel signature (HKS) from the outer surface to the interior but is still intrinsic in the sense that when it is restricted to the outer surface it coincides with the surface-based computation. Wang et al. (2014) modify the Laplace–Beltrami operator to include concavity information, aiding in applications like segmentation without distinguishing between the interior and exterior of a closed surface. Similarly, Corman et al. (2017) use the Laplacian of a meshed shell around a triangulated surface, incorporating mean curvature but treating the surface as a thin shell rather than the boundary of a volume. Techniques like (Chuang et al. 2009) use extrinsic calculations to approximate the Laplace–Beltrami operator; these methods aim to capture intrinsic geometry.

We use the Dirichlet-to-Neumann (DtN) operator as an extrinsic analog to the Laplace–Beltrami operator. In graphics, Gao et al. (2014) use DtN for skinning; it also appears in a recent pipeline for parameterization (Sawhney and Crane 2017). In simulation and PDEs, it appears as a Schur complement eliminating interior degrees of freedom (Bertozuza 2003; Liu et al. 2016; Quarteroni and Valli 1999; Smith et al. 2004; Toselli and Widlund 2005).

Very recent concurrent work proposes using the Dirac operator to capture extrinsic shape (Liu et al. 2017). This method is among the first to extend spectral geometry processing to include extrinsic information, but it still treats the surface as a shell rather than as the boundary of a volume. While this work opens intriguing theoretical questions regarding the informativeness of the Dirac operator (see their §7), the Dirichlet-to-Neumann operator we propose for shape analysis enjoys strong grounding in existing theory; this allows us to prove in §3.1 that the Dirichlet-to-Neumann operator fully captures extrinsic geometry up to rigid motion (Proposition 3.1). On the other hand, unlike the operators we consider in this paper, their Dirac operator benefits from the efficiency of sparse linear algebra.

2.3 Numerical PDE

Two typical numerical methods for approximating Dirichlet-to-Neumann operators are the finite element method (FEM) and the boundary element method (BEM). FEM yields a sparse linear system on a discretization of the volume bounded by a surface, while BEM yields a dense linear system on just the boundary. Efficiency-wise, the added number of elements required for FEM makes the two methods similar asymptotically. We use BEM to avoid dependence on volumetric meshing.

Fast numerical methods can be used to accelerate BEM, including the fast multipole method (FMM) and the hierarchical matrix (H-matrix) method. FMM (Engheta et al. 1992; Rokhlin 1985) expands the Green’s function using a multipole approximation, grouping close points as a single source. The H-matrix method represents BEM operators hierarchically, reducing the cost of assembly and matrix-vector products to $O(n \log n)$ time (Börm et al. 2003; Hackbusch 1999). Either method reduces evaluation cost to $O(n \log 1/\varepsilon)$ time, where $\varepsilon$ is a prescribed accuracy, making it possible to solve large problems on a single workstation.

3 MATHEMATICAL PRELIMINARIES

3.1 Steklov eigenproblem
Denote a volumetric domain as $\Omega \subseteq \mathbb{R}^3$ with boundary surface $\Gamma = \partial \Omega$. The Steklov eigenvalue problem is defined as:

$$
\begin{aligned}
\Delta \psi(x) &= 0 & x \in \Omega \\
\nabla_n \psi(x) &= \lambda \psi(x) & x \in \Gamma
\end{aligned}
$$

where $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2$ denotes the Laplacian operator and $\nabla_n$ the normal derivative at the boundary. The spectrum of the Steklov problem is discrete and given by a sequence of eigenvalues $\lambda = 0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \to \infty$.

$(\lambda_n)_{n=0}^\infty$ can be interpreted as the eigenvalues of the Dirichlet-to-Neumann (DtN) operator $S : H^{1/2}(\Gamma) \to H^{-1/2}(\Gamma)$ defined as $Sf := \nabla_n(\bar{E}f)$, where $\bar{E}$ denotes the harmonic extension into the interior; denote $\phi_n = \psi_n|_\Gamma : \Gamma \to \mathbb{R}$ as the corresponding eigenfunctions. $S$ is known as the Dirichlet-to-Neumann operator because it maps the boundary Dirichlet data to the Neumann data (e.g., voltage-to-current, or temperature-to-flux).

Just as the Laplace–Beltrami spectrum encodes intrinsic geometry, the spectrum of $S$ encodes extrinsic information. In particular, for smooth domains in $\mathbb{R}^3$, the Steklov heat kernel admits the asymptotic expansion as $t \to 0^+$:

$$
e^{-tS}(x, x) = \sum_{i=0}^{\infty} a_i(x) t^{\frac{i}{2}} + \sum_{i=1}^{\infty} b_i(x) t^{-\frac{i}{2}} \log t,
$$

where the coefficients $a_i(x)$ for $k = 0, 1, 2$ and $b_1(x)$ are local geometric invariants, in the sense that they are determined by the local geometry of $\Gamma$ in a neighborhood of the point $x \in \Gamma$ (Polterovich and Sher 2015). See (Duistermaat and Guillemin 1975) where such an expansion is proved for a general elliptic pseudodifferential operator.

The following expressions were obtained by Polterovich and Sher (2015):

$$
a_0(x) = \frac{1}{2\pi} \quad a_1(x) = \frac{H(x)}{4\pi} \quad a_2(x) = \frac{1}{16\pi} \left( \frac{H(x)^2 + K(x)}{3} \right),
$$

where $H(x)$ and $K(x)$ are, respectively, the mean and the Gaussian curvatures of $\Gamma$. While the heat expansion for the Laplace–Beltrami operator on the boundary captures only intrinsic geometric invariants of $\Gamma$, the first few terms of the Steklov heat expansion (1) contain the mean curvature of the boundary, which is an extrinsic geometric quantity. We refer to the recent survey by Girouard and Polterovich (2017) for other results on spectral geometry of $S$.

Recall that the Laplace–Beltrami operator is invariant under isometries. For example, the bumpy cubes in Figure 1 have isometric boundaries, and therefore their boundary Laplacians coincide. At the same time, the boundaries of the bumpy cubes have different extrinsic geometries, and one could check that the DtN operators on these cubes are also different. Moreover, in striking contrast with the boundary Laplacian, under some assumptions the only maps preserving the DtN operator for domains in $\mathbb{R}^3$ are rigid motions.

**Proposition 3.1.** Suppose $\Omega_1, \Omega_2 \subseteq \mathbb{R}^3$ are compact domains with $C^\infty$ boundaries $\Gamma_1, \Gamma_2$ and Dirichlet-to-Neumann operators $S_1$ and $S_2$, respectively. Let $a : \Omega_1 \to \Omega_2$ be a bijection which is $C^\infty$ up to the boundary, and let $\tilde{a} : \Gamma_1 \to \Gamma_2$ be the induced mapping between the boundaries. Suppose that the operators $S_1$ and $S_2$ coincide up to composition with $a$, i.e., $S_2f = \tilde{a}_*S_1^*a^*f$, for any $f \in C^\infty(\Gamma_2)$, where $\tilde{a}^*f = f \circ \tilde{a}$, $\tilde{a}_*g = g \circ \tilde{a}^{-1}$ denote the pull-backs by $\tilde{a}$ and $\tilde{a}^{-1}$, respectively. Then $a$ is a rigid motion.

**Proof.** The following argument was communicated to us by M. Karpukhin: By Proposition 1.3 and the discussion above it in (Lee and Uhmann 1989), the Dirichlet-to-Neumann map of a smooth surface determines the full Taylor series of the metric $g$ at the boundary in the boundary normal coordinates. In particular, it determines the metric itself—which in our case is the first fundamental form of the boundary surface—and its first derivative, which gives us the second fundamental form (see e.g., (Kachalov et al. 2001, §2.1.18)). At the same time, it follows from Bonnet’s theorem that the first and the second fundamental form determine a surface in $\mathbb{R}^3$ up to rigid motions. \hfill $\square$

Proving an analogous result for non-smooth boundaries remains a significant challenge for future work.

### 3.2 Boundary representation and operators

To derive a boundary representation of the DtN operator, first consider the Laplace equation with Dirichlet boundary conditions:

$$
\begin{aligned}
\Delta u(x) &= 0 & x \in \Omega \\
u(x) &= g(x) & x \in \Gamma
\end{aligned}
$$

where $g(\Gamma)$ is Dirichlet data on the boundary. A basic fact from elliptic PDE is that (5) uniquely determines $u(\Omega)$ and hence its Neumann data $\bar{g}_n = \frac{\partial}{\partial n}u(\Gamma)$. By definition, the Dirichlet-to-Neumann operator $S$ is the map $g \mapsto \bar{g}_n$.

To avoid discretizing the interior of the domain $\Omega$, we use integral operators to bypass solving the Laplace equation (5). Here, we introduce relevant boundary operators and notation that will be used throughout the paper. We refer readers to (Steinbach 2007) for a comprehensive introduction to these operators in the context of the boundary element method.

**Single layer potential.** The single layer potential $V : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$ is defined via

$$
[V\phi](x) := \int_{\Gamma} G(x, y) \phi(y) d\Gamma(y),
$$

where $G(x, y) = \frac{1}{4\pi} \frac{1}{|x-y|}$ is the fundamental solution of the Laplace equation. Physically, $\mathcal{V}$ maps an input electric charge distribution $\phi$ to the resulting electric potential distribution.

**Double layer potential.** The double layer potential $\mathcal{K} : H^{1/2}(\Gamma) \to H^{1/2}(\Gamma)$ is defined via

$$
[\mathcal{K}\psi](x) := \int_{\Gamma} \frac{\partial G(x, y)}{\partial n(y)} \psi(y) d\Gamma(y),
$$

where the integral is understood in the sense of the Cauchy principal value (Kanwal 2013). Physically, $\mathcal{K}$ maps an input electric dipole density distribution $\phi$ to the resulting electric potential distribution.
Adjoint double layer potential. The adjoint double layer potential \( \mathcal{T} : H^{-1/2}(\Gamma) \to H^{-1/2}(\Gamma) \) is defined as the conormal derivative of \( \mathcal{V} \):

\[
[\mathcal{T} \phi](x) := \int_{\Gamma} \frac{\partial G(x, y)}{\partial n(x)} \phi(y) \, d\Gamma(y),
\]

where the integral is understood in the sense of the Cauchy principal value. Physically, \( \mathcal{T} \) maps an input electric dipole density distribution \( \phi \) to the normal derivatives of the resulting electric potential distribution.

Hypersingular operator. The hypersingular operator \( \mathcal{H} : H^{1/2}(\Gamma) \to H^{-1/2}(\Gamma) \) is defined as minus the conormal derivative of \( \mathcal{V} \):

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

Physically, \( \mathcal{H} \) maps an input electric dipole density distribution \( \phi \) to normal derivatives of the resulting electric potential distribution.

Operator properties. \( \mathcal{V} \) and \( \mathcal{H} \) are self-adjoint operators, i.e.

\[
\langle \nu, \mathcal{V} \mu \rangle_T = \langle \mathcal{V} \nu, \mu \rangle_T \quad \text{and} \quad \langle \nu, \mathcal{H} \mu \rangle_T = \langle \mathcal{H} \nu, \mu \rangle_T,
\]

where the inner product \( \langle \cdot, \cdot \rangle_T \) is given by

\[
\langle \phi, \psi \rangle_T := \int_{\Gamma} \phi(y) \psi(y) \, d\Gamma(y).
\]

\( \mathcal{K}, \mathcal{T} \) are adjoint to each other, i.e. \( \langle \nu, \mathcal{K} \mu \rangle_T = \langle \mathcal{T} \nu, \mu \rangle_T \). Furthermore, \( \mathcal{V} \) is positive definite, and \( \mathcal{H} \) is positive semi-definite.

3.3 Calderón projection & Dirichlet-to-Neumann operator

The Calderón projection relates \( \mathcal{V} \) and \( \mathcal{H} \) through the relationship

\[
\begin{bmatrix}
g_1 \\
g_n
\end{bmatrix} = \begin{bmatrix}
\frac{1}{2} I - \mathcal{K} \\
\mathcal{V} \mathcal{H} - \frac{1}{2} I + \mathcal{T}
\end{bmatrix} \begin{bmatrix}
g_1 \\
g_n
\end{bmatrix}
\]

where \( I \) is the identity operator (Grubb 2009). From the first row of this expression, we can derive the Dirichlet-to-Neumann operator \( \mathcal{S} : g_1 \mapsto g_n \) as the composition

\[
\mathcal{S} = \mathcal{V}^{-1} \left( \frac{1}{2} I + \mathcal{K} \right).
\]

Combining both rows, however, reveals an alternative symmetric expression for the same operator:

\[
\mathcal{S} = \mathcal{H} + \left( \frac{1}{2} I + \mathcal{T} \right) \mathcal{V}^{-1} \left( \frac{1}{2} I + \mathcal{K} \right).
\]

Since \( \mathcal{H} \) is self-adjoint and positive semi-definite, \( \mathcal{V} \) is positive definite, and \( (\mathcal{K}, \mathcal{T}) \) are adjoint pairs, this alternative form symbolically shows that \( \mathcal{S} \) is self-adjoint and positive semi-definite, with \( \langle u, \mathcal{S} u \rangle_T = 0 \) if and only if \( u \) is a constant function. In our discretization, we work with this second formula because its discretization will become symmetric and positive semidefinite by construction.

Note that interior Steklov eigenfunctions are known in closed form by the representation formula

\[
u(x) = \int_{\Gamma} G(x, y) g_n(y) \, d\Gamma(y) - \frac{\partial G(x, y)}{\partial n(y)} g(y) \, d\Gamma(y)
\]

for any \( x \in \Omega \). Here, \( u(\cdot) \) is the \( i \)-th eigenfunction in the interior when we let \( g(\cdot) \) and \( g_n(\cdot) \) to be the \( i \)-th eigenfunction on the surface and its normal derivative, respectively; later in equation (14) they are denoted as \( u \) and \( t \). Our algorithm will compute both \( u \) and \( t \), so interior eigenfunctions can be easily evaluated if needed.

4 DISCRETE DIRICHLET-TO-NEUMANN OPERATOR

4.1 Weak form boundary operators

We discretize all operators discussed above. Each corresponds to a weak-form discretized operator matrix:

\[
\mathcal{V} \mapsto V, \quad \mathcal{K} \mapsto K, \quad \mathcal{T} \mapsto T, \quad \mathcal{H} \mapsto H, \quad \text{and} \quad I \mapsto M.
\]

Take the single layer integral as an example. Assume \( u(\Gamma) \) is solution to the single layer integral equation \( Vu(x) = f(x) \). Then for any test function \( v(\Gamma) \) we have the weak form \( \langle v, Vu \rangle_T = \langle v, f \rangle_T \). Restricting \( u, v, f \) to the piecewise-linear subspace \( S^1_h(\Gamma) \) (the so-called “hat functions” on a triangulated surface) leads to the finite-dimensional linear system \( Vu = Mf \), where \( V, M \) are Galerkin matrices

\[
V \in \mathbb{R}^{n \times n} : \langle V v, u \rangle_T = \langle \phi, V \phi \rangle_T \quad (8)
\]

\[
M \in \mathbb{R}^{n \times n} : \langle M v, u \rangle_T = \langle \phi, \phi \rangle_T. \quad (9)
\]

In this linear system, \( u, f \) contain the coefficients of \( u \) and \( f \) in the piecewise linear basis, resp.

Similarly, we have

\[
K \in \mathbb{R}^{n \times n} : \langle K v, u \rangle_T = \langle \phi, K \phi \rangle_T \quad (10)
\]

\[
T \in \mathbb{R}^{n \times n} : \langle T v, u \rangle_T = \langle \phi, T \phi \rangle_T \quad (11)
\]

\[
H \in \mathbb{R}^{n \times n} : \langle H v, u \rangle_T = \langle \phi, H \phi \rangle_T. \quad (12)
\]

It follows directly from properties in §3.2 that \( V = VT, H = HT \), \( T = KT, V > 0, \) and \( H \geq 0 \). We refer interested readers to (Steinbach 2007) for a detailed discussion of these discretized operators.

4.2 Discretized Dirichlet-to-Neumann operator

We approximate the DtN operator by substituting the continuous functions \( v \) and \( f \) for their piecewise-linear approximations \( v^h \) and \( f^h \). Further, using the properties of the discrete DtN operator, we have

\[
\mathcal{S} \sim M^{-1}(H + (0.5M + T)V^{-1}(0.5M + K)).
\]

Define the (symmetric) weak form of this operator as

\[
\mathcal{S} = H + (0.5M + T)V^{-1}(0.5M + K).
\]

5 MATRIX-FREE FORMULATION

Many applications require solving the linear system \( Su = Mf \) or the generalized eigenvalue problem \( Sx = \lambda Mx \). A naïve approach is to assemble all the operators as dense matrices via (13). Without acceleration, however, assembling these dense matrices would take \( O(n^3) \) time, and matrix-vector products would require \( O(n^2) \) operations. Furthermore, formula (13) involves inverting the dense matrix \( V \), which is expensive. Instead we apply a reformulation that avoids explicit matrix inversion.

5.1 Expansion

Define an auxiliary variable \( t \) as \( t = V^{-1}(0.5M + K)u \). Notice that \( t \) represents the Neumann data, i.e. the normal derivative at boundary.
The linear system $Su = Mf$ then can be expanded as a saddle point system
\[
\begin{bmatrix}
V & -Q \\
Q^T & H
\end{bmatrix}
\begin{bmatrix}
t \\
u
\end{bmatrix} =
\begin{bmatrix}
0 \\
Mf
\end{bmatrix},
\]
where $Q := 0.5M + K$.

The same technique also applies to reformulating the eigenvalue problem $Su = \lambda Mu$ as the system:
\[
\begin{bmatrix}
V & -Q \\
Q^T & H
\end{bmatrix}
\begin{bmatrix}
t \\
u
\end{bmatrix} = \lambda
\begin{bmatrix}
0 \\
M
\end{bmatrix}
\begin{bmatrix}
t \\
u
\end{bmatrix}
\]
(14)

The left-hand sides of these expressions do not contain matrix inversions, allowing us to use iterative linear system/eigenvalue solvers that only require matrix-vector products. See §5.4 for details of our solver and §5.6 for how we use the hierarchical techniques to apply $V$, $Q$, and other matrices without storing their elements.

5.2 Symmetrization

Numerical solution to saddle point systems like the ones in the previous subsection is a basic task in numerical PDE; see e.g. (Benzi et al. 2005). To use the conjugate gradient (CG) algorithm and other techniques requiring positive definiteness, we apply a Bramble–Pasciak transformation (1988):
\[
\begin{bmatrix}
\alpha VP^{-1} - I & 0 \\
-\alpha Q^TP^{-1} & I
\end{bmatrix}
\begin{bmatrix}
V & -Q \\
Q^T & H
\end{bmatrix}
\begin{bmatrix}
t \\
u
\end{bmatrix} = \begin{bmatrix} 0 \\
Mf \end{bmatrix},
\]
where $P_V$ is a symmetric preconditioner for the single layer potential (our choice is defined in §5.3). The constant $\alpha = \frac{1}{\gamma}$ is chosen to ensure the positive-definiteness of the system matrix as $\gamma = 0.9\sigma_{\min}(P_V^{-1}V) < 0.9 < 1$, where $\sigma_{\min}(\cdot)$ denotes the smallest singular value. The matrix of this system is symmetric and positive semi-definite:
\[
A = \begin{bmatrix}
\alpha VP^{-1} - I & 0 \\
-\alpha Q^TP^{-1} & I
\end{bmatrix}
\begin{bmatrix}
V & -Q \\
Q^T & H
\end{bmatrix}
\begin{bmatrix}
t \\
u
\end{bmatrix} = A \begin{bmatrix}
0 \\
M
\end{bmatrix}
\begin{bmatrix}
t \\
u
\end{bmatrix}
\]
(15)

Under the Bramble–Pasciak transformation, the eigenvalue problem can be written as
\[
A \begin{bmatrix}
t \\
u
\end{bmatrix} = \lambda 
\begin{bmatrix}
0 \\
M
\end{bmatrix}
\begin{bmatrix}
t \\
u
\end{bmatrix}
\text{ or more simply } Ax = \lambda Mx.
\]

Although $A$ and $M$ are positive semi-definite, $\alpha A + \beta M$ is strictly positive definite, for any $\alpha, \beta > 0$. So it can be solved by a shifted generalized eigenvalue method.

5.3 Preconditioning

Steinbach and Wendland (1998) proved that a modified hypersingular operator is a good preconditioner for the layer potential operator, and that the singular layer potential is a good preconditioner for the Dirichlet-to-Neumann operator (13). Especially it was proved that the spectral condition numbers of these two preconditioners are $O(1)$, in the sense that the condition number remains the same when the domain is upsampled. Their choice of preconditioners for $S$ and $V$ are:
\[
\begin{align*}
P^{-1}_S & := 4M^{-1}(H + \frac{\beta}{\gamma}11^T)M^{-1} \\
P^{-1}_V & := M^{-1}VM^{-1}
\end{align*}
\]
(16)

where we choose $\beta = 1/(1^T M 1)^{3/2}$ to ensure scale invariance. Since $M$ is sparse and diagonally dominant, Jacobi iteration could compute $M^{-1}x$ in $O(n)$ time. In our implementation, however, we use a diagonal lumped mass matrix that can be inverted in closed form.

For saddle point system $Ax = \lambda Mx$, Bramble and Pasciak (1988) recommend the following preconditioner to (15) which is spectrally equivalent to $A$:
\[
P := \begin{bmatrix}
V - \gamma P_V & P_S \\
\gamma P_V & P_S
\end{bmatrix}
\]

The upper left block of this matrix cannot be inverted efficiently, so it has to be combined with a modified Bramble-Pasciak CG solver. We found that the following alternative and much simpler preconditioner works well and is compatible with the formulas in (16):
\[
P := \frac{1}{\gamma I} P_V
\]

See (Stoll and Wathen 2007) for a survey on other preconditioning options for saddle point system.

5.4 Iterative solvers

We use the preconditioned conjugate gradient (CG) algorithm to solve linear systems of equations involving our positive definite operators. As noted above, PCG requires matrix-vector products rather than storing our operators explicitly, as would be required by Gaussian elimination.

We employ the locally optimal block preconditioned conjugate gradient (LOBPCG) solver (Knyazev 2001) to compute the spectrum efficiently. The convergence speed of this method depends on the condition number $\text{cond}(P_A^{-1}A)$. LOBPCG allows for an initial guess of the eigenvectors. We apply a multi-scale idea, computing "progressive spectra" from progressively simplified meshes obtained using (Hoppe 1996). We first compute eigenvalues and eigenfunctions on a low-resolution mesh, and upsample the low resolution eigenfunctions as initialization for high-resolution meshes. Typically we use 2–3 levels of progressively simplified meshes, where at each level the mesh is simplified by a factor of $1/4$; we use the nearest neighborhood rule to upsample eigenfunctions. Note Vaxman et al. (2010) use a related approach to evaluate the heat kernel.

5.5 Generalization

We have introduced the Dirichlet-to-Neumann operator for compact manifolds with closed boundaries. As long as we are given a normal vector field at every point, however, the integral-based definitions of inner products $\langle \mathbf{v}, \mathbf{Vu} \rangle_T$, $\langle \mathbf{v}, \mathbf{Ku} \rangle_T$, $\langle \mathbf{v}, \mathbf{Hu} \rangle_T$, and $\langle \mathbf{v}, \mathbf{T u} \rangle_T$ as well as the resulting operator $S$ remain valid even if $\Gamma$ does not bound some region in $\mathbb{R}^3$. Given this observation, to process oriented surfaces with boundary we simply use the same DtN operator (13), although without theoretical justification via PDE.

We experimentally verify that this natural generalization demonstrates reasonable and robust behavior. Figure 2 illustrates our "generalized Steklov" eigenfunctions computed on a hemisphere from these integral operators. Figure 3 demonstrates that the resulting spectrum blends smoothly as the shape changes from a sphere to a hemisphere by moving the cutting plane linearly.
5.6 Implementation details

Our rephrasing of the DtN operator in terms of boundary integrals is an example of the boundary element method (BEM) from numerical PDE. Hence, our implementation uses the BEM++ library (Śmigaj et al. 2015) to evaluate the matrices above, whose elements are obtained using numerical quadrature. BEM++ uses hierarchical matrices ("H-matrices") to compute matrix-vector products efficiently, effectively approximating the integrals with hierarchical expressions that group together small contributions from far-away vertices (Börn et al. 2003). \( \sigma_{\text{min}}(P^2V) \) can be approximated by solving the minimum eigenvalue of the squared system, without the need for a preconditioner.

6 VALIDATION

6.1 Robustness to topological change

In addition to the hemisphere test in Figure 2, Figure 4 gives an example of robustness of our operator to topological change. Here, we compare the Steklov eigenvalues and eigenfunctions of a torus with zero, one, and two rings cut out. These topological changes—which drastically affect the Laplace–Beltrami operator—have little effect on the Steklov spectrum.

Figure 5 shows an example of robustness to "topological noise." Here, we remove triangles from the mesh with uniform probability and show the resulting Steklov spectrum. Once again, our operator is remarkably stable to these changes, yielding a stable spectrum even when 50% of the triangles are removed.
6.2 Robustness to surface sampling

Figure 6 shows an example of robustness to the sampling rate. Here we simplify the dragon mesh and observe that the resulting Steklov spectrum does not change significantly.

6.3 Robustness to vertex noise

Figure 7 illustrates stability of our BEM discretization when noise is added to the boundary surface. Although it is defined in §3.1 via a second order differential equation, we observed that the lower Steklov eigenfunctions are particularly robust to noise, similarly to those of the Laplace–Beltrami operator. Both Laplacian and Steklov spectra scale down globally, since adding noise increases the edge lengths and thus the surface area of the mesh.

As a small note for future work, the BEMPP library occasionally fails when the mesh self-intersects. Typically this is not a problem, but in this noise test nearby triangles with opposing normals can cause issues for numerical integration.

6.4 Stability test for volume isometries

The Steklov spectrum is invariant to volumetric isometry, and our experiments demonstrate that it remains stable when the volume enclosed by the outer surface deforms near-isometrically. In the first row of Figure 9, when the rectangular prisms are deformed using a near-volume-isometric bending map, the Steklov and Laplacian eigenvalues and eigenfunctions remain almost unchanged. In the second row, however, the thin box is deformed subject to a near-surface-isometry but non-volume-isometry, by shearing the interior of the box. Only the Steklov spectrum captures the volumetric change as the shear increases. This example illustrates how the Steklov spectrum discriminates nuanced non-isometric volume changes, while the Laplacian spectrum fails to capture the difference.
6.6 Convergence to analytical eigenvalues

The analytical Steklov eigenvalues and eigenfunctions are known for the sphere. We compare the numerical solution of our method on the sphere for verification purposes. Figure 11 verifies that our discretized operators are faithful to these ground-truth quantities.

Fig. 11. Relative error measured in between spherical harmonics and the eigenspaces of our Steklov operator. Relative error decreases as the mesh resolution increases. The error is measured for a fixed number of 25 iterations, computed from random initialization.

6.7 Timing

For the cube with 6146 vertices and 12288 triangles, BEMPP takes 60 seconds to assemble all boundary element operators, one LOBPCG iteration takes 24 seconds on average, and 20 iterations are usually sufficient for random initialization. Usually less than 10 iterations are needed for initialization from low-resolution eigenfunctions. We observed that all the timings roughly scale linearly with size of the mesh. Our implementation is in Python, and results are collected on an Alienware laptop with an Intel i7 4800MQ CPU and Linux operating system. As many graphics applications do not require a very high precision, we believe that there is large room for future work to improve the computational efficiency (see §8).

6.8 Comparison with FEM

An alternative approach for extrinsic spectral geometry uses the volumetric Laplacian. Spectral computations with this operator do not have a boundary integral formulation, and hence we would need a volumetric mesh of the interior of the domain. This limits applicability, since tet meshing typically requires a watertight surface as input, and the results depend on the algorithm used for generating a tet mesh. Furthermore, applications such as computation of shape differences (§7.4) would require correspondence of not just the outer surface but the interior as well, which may not be available. As we will show in §7.2, we do not observe a case in which an explicit mesh of the interior leads to significantly better results for geometry processing tasks.

These high-level concerns aside, one could reasonably discretize the DtN operator using the finite element method (FEM) on a tet mesh. In particular, the DtN operator can be derived as the Shur complement of the volumetric Laplacian. On a tetrahedral mesh with volumetric FEM-based Laplacian $L^{(vol)}$ and mass matrix $M^{(vol)}$, divide this matrix into blocks corresponding to boundary vertices $b$ and interior vertices $i$:

$$L^{(vol)} = \begin{bmatrix} L_{bb} & L_{bi} \\ L_{ib} & L_{ii} \end{bmatrix}.$$
Figure 12 illustrates a drawback of using the tet-based DtN operator. Here, we compute Steklov eigenvalues on two different tetrahedral meshes of the same unit sphere using first-order FEM. This figure shows that Steklov eigenvalues depend on the choice of tetrahedral meshing of the boundary. In particular, faster sparse computations on coarse FEM meshes come at the cost of inaccurate Steklov approximation. Contrastingly, BEM faithfully approximates Steklov eigenvalues for the sphere beyond the range tested in this experiment.

We further observe that BEM with the correct full mass matrix (our method) yields the closest spectrum to ground truth by a significant margin; even BEM with a lumped diagonal mass matrix does not yield accurate results. Hence, our implementation always uses the full mass matrix in the generalized eigenvalue problem. We do use lumped mass matrices to construct preconditioners for the iterative solvers, since they can be inverted quickly; this affects only the number of iterations for the solver rather than the quality of the final output.

7 EXPERIMENTS AND APPLICATIONS

We highlight two properties of the DtN operator $S$ distinguishing it from the Laplace–Beltrami operator, motivated in the theoretical discussion in §3:

- $S$ captures the volumetric structure of the interior.
- $S$ incorporates the mean curvature of the outer surface.

Accordingly, the algorithms we derive based on the Steklov spectrum enjoy these properties.

Many existing frameworks for intrinsic geometry processing can be extended to take extrinsic geometry into consideration by substituting the Laplacian operator with the DtN operator. There are many ways to justify this simple substitution:

- Many intrinsic geometry tasks involve the surface-based Laplace–Beltrami operator, whose weak form corresponds to a Dirichlet energy on the surface:
  \[
  \langle \nabla_f u, \nabla_f u \rangle = u^T L u.
  \]

The weak form of the DtN operator provides the closely-related volumetric Dirichlet energy of the harmonic extension $E$ of a surface-based function:

\[
\langle \nabla E u, \nabla E u \rangle_{\Omega} = u^T S u.
\]

- Both operators’ associated eigenproblems involve Rayleigh quotients with similar numerators and identical denominators (Girouard and Polterovich 2017).
- For $n$ disconnected/closed pieces, both operators have $n$-dimensional null spaces.
- Laplace–Beltrami and DtN operators both can be understood as subtracting the value of a function at a point from its average over a neighborhood. The DtN operator first interpolates the function harmonically to the volume, while Laplace–Beltrami restricts to the surface.

The close analogies above hint that shifting from intrinsic geometry to extrinsic geometry can be accomplished by substituting the cotangent Laplacian operator with a discretized DtN operator.
Fig. 13. Selected Steklov eigenfunctions on the dragon model, with comparison to Laplacian eigenfunctions. Although the Steklov eigenfunctions are computed using the boundary geometry, these eigenfunctions are aware of the geometry of the volume enclosed by the dragon surface, in particular the S-shaped bend of its body.

We demonstrate this substitution in a few contexts. In particular, we experiment and study properties of Steklov-based kernel signatures, distances, and shape differences, which are fundamental components of many high-level geometry processing and shape analysis algorithms.

7.1 Steklov spectrum

Figures 13 and 14 illustrate Steklov eigenfunctions on more complex models; surface-based Laplace–Beltrami eigenfunctions are shown for comparison. Both models have more vertices than can be feasibly handled using dense matrix computations, highlighting the necessity for iterative solvers introduced in §5.4.

A few qualitative differences between the intrinsic and extrinsic eigenfunctions are worth pointing out. On the gargoyles model, the Steklov eigenfunctions exhibit more localization on the wings; the wings are distinctive volumetric features, whereas the outer surface blends the wings into the body/base. On the dragon model, eigenfunctions are localized in individual folds of the dragon, while the Laplace–Beltrami eigenfunctions extend along the entire surface. This effect is reflective of the fact that intrinsically the dragon model is a long tube in which the 180° fold of the body between the front and hind legs is insignificant; volumetrically, however, the up/down bending of the dragon is a prominent geometric feature.

Figure 15 shows an experiment similar to the ones in (Vallet and Lévy 2008), in which the eigenfunctions of the DtN operator are used to compress and subsequently reconstruct the $xyz$ coordinate functions of a triangle mesh. Interestingly, fewer Steklov eigenfunctions are needed to capture key geometric features relative to the Laplace–Beltrami eigenfunctions. That is, low-frequencies in DtN space appear to better capture geometric variation.

7.2 Kernel-based descriptors

The heat equation associated to the DtN operator is given by

$$\frac{du}{dt} = -Su,$$

whose solution can be written

$$u(x, t) = \int_{\Gamma} k_t(x, y) f(y) dy.$$  \hspace{1cm} (18)

Here, $k_t(x, y)$ is the heat kernel

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

where $\lambda_i$ and $\phi_i$ are the $i^{th}$ eigenvalue and eigenfunction of $\mathcal{S}$, resp.

If we replace DtN with the Laplace–Beltrami operator, the heat kernel signature (HKS) of a point $x \in \Gamma$ is defined as $h_t(x) = k_t(x, x)$ (Sun et al. 2009), the diagonal of the heat kernel. Considered as a function of $t$, $h_t(x)$ provides a multi-scale characterization of intrinsic geometry near $x$; it is a basic descriptor used in many shape matching and correspondence algorithms. The wave kernel signature (WKS) (Aubry et al. 2011), also defined using the Laplace–Beltrami operator, outperforms the heat kernel signature for shape matching tasks using a related eigenvalue formula.

The heat and wave kernel signatures can be naturally generalized to the DtN operator by replacing the Laplace–Beltrami spectrum/eigenfunctions with Steklov spectrum/eigenfunctions. A similar construction is considered by Raviv et al. (2010), who construct a volumetric HKS via a coarse discretization of the Laplacian in the interior of a volume bounded by a watertight surface. Without this interior meshing, our DtN-based HKS and WKS capture similar extrinsic shape properties.

Figure 16 compares the Laplace–Beltrami, DtN, and volumetric HKS functions on a triangulated surface; the volumetric HKS is approximated using a tetrahedral mesh of the interior, as explained in §6.8. Following (Sun et al. 2009), we consider the time interval $[\frac{4 \ln 10}{\lambda_{30}}, \frac{4 \ln 10}{\lambda_{2}}]$ in all of our examples, and Figure 16 shows typical patterns of the signature using either a small or large time from the range. The DtN and volumetric kernel signatures capture the mean curvature in the palm of the hand, ignored by the intrinsic HKS. Similarly, Figure 17 illustrates the point signatures at the apexes of the bumps in Figure 1. Since the two cubes are isometric, Laplace–Beltrami-based signatures does not discriminate the two points, while the DtN-based signature distinguishes the two models.

Figure 18 illustrates how DtN-derived descriptors can be more discriminative than their Laplace–Beltrami counterparts. Here, we mark a point on the foot of a human model and show the distance between its descriptor and those of other surface points. Fewer points have descriptors close to the descriptor of the foot, showing that the DtN WKS embedding is less ambiguous. In the second example, DtN WKS is better aware of the left-right symmetry breaking, and tends to discriminate left and right feet. This property may or may
7.3 Spectral distance

Several spectral distances are defined between points on a surface in terms of the spectrum of the Laplace-Beltrami operator. These distances enjoy certain stability and smoothness properties not satisfied by the geodesic distance and are computable using linear algebra machinery. These distances naturally generalize to the Steklov spectrum, providing spectral volume-aware distances that require computation only on the boundary.

Working in analogy to previous work (Coifman and Lafon 2006; Lipman et al. 2010), we define the Steklov diffusion distance and the bi-Steklov distance as follows

\[ d_D(x, y)^2 = \sum_{i=1}^{\infty} e^{-2\lambda_i} (\phi_i(x) - \phi_i(y))^2 \]

\[ d_B(x, y)^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i^2} (\phi_i(x) - \phi_i(y))^2 . \]

These distances are aware of relationships between points that reach across the volume.

Figures 19, 20, and 21 show several examples of these distances on meshed surfaces. A few key examples point out the special properties of our distance compared to its intrinsic and/or more naïve extrinsic counterparts:
under the map $\mathcal{F}$ the Dirichlet energy $h(f, f)$ of $f$'s harmonic extension, will be best preserved in the sense $|b_M^f(f, f) - b_M^f(Ff, Ff)|$ is minimized.

The action of $\mathcal{D}$ reflects how different the two geometric domains are. Inherited from the DtN operator, $\mathcal{D}$ defines a more rigid notion of shape differences that captures extrinsic shape changes not captured by intrinsic differences.

For example, Figure 22 shows a shape difference between a rectangular sheet and a bent one, Figure 23 shows a shape difference between a cigar and a bent cigar. The point-wise distortion is measured by the metric

$$\text{(distortion)}_i = \frac{e_i^T \mathcal{M} e_i}{\epsilon_i^T \mathcal{M} \epsilon_i},$$

as in (Rustamov et al. 2013). The Steklov-based shape difference is far less noisy and distinguishes the bending direction.

Figure 24 illustrates the effect of principal component analysis (PCA) on a collection of shape differences measured through correspondence to a base mesh; see (Rustamov et al. 2013) for details of the technique. Our new difference operator captures the extrinsic changes between surfaces that are nearly identical intrinsically, most notably the "bumpy cubes" as an extreme example. In general, our Steklov-based shape difference matrices capture more variability in the shape collections than the Laplace–Beltrami counterparts, our Steklov-based shape difference matrices capture more variability in the shape collections than the Laplace–Beltrami counterparts, even when using the generalized DtN operator introduced in §5.5 to study differences between open surfaces (the bending sheet and falling cloth sequences).

7.5 Shape exploration and retrieval
In shape retrieval applications, we may wish to distinguish the same piece of geometry in different “poses,” e.g. for purposes of organizing a database of human scans or for processing an animated sequence of an articulated character. In these instances, the additional discriminative features provided by DtN-based computation may be desirable. As an example, Figure 25 shows an eigenspace embedding of a collection of shapes from the TOSCA dataset (Bronstein et al. 2008) using the Steklov spectrum, similar to (Reuter et al. 2006).

7.6 Volume-aware segmentation
Equation 2 suggests that Steklov eigenfunctions encode point-wise mean curvature, providing geometric clues for surface segmentation useful in descriptor-based algorithms (Chen et al. 2009); see Figure 26 for an example. To demonstrate, in Figure 27 we apply a naïve strategy for segmentation. For the input mesh, we first compute the Steklov spectrum embedding as

$$\begin{pmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_k(x) \end{pmatrix},$$

where $x \in \Omega$. Then, we apply the $k$-means clustering to this embedding with a user-specified number of clusters; to avoid local optima, we restart $k$-means ten times with random initialization and keep the clustering with lowest objective value. This simple strategy yields consistent segmentations: While the Laplacian embedding tends to segment the surface into flat patches, the Steklov embedding tends to segment the shape into volumetric parts.

• Flat disk (Figure 19): The DtN distance is small between points at the center of the top and the bottom of the disk, which are close if you cut through the interior of the disk and far along the surface.

• Mouse (Figure 20): This example shows the opposite effect. The two hands of the mouse are close in Euclidean distance, but in Steklov distance they are far. This is because our distances are interior-aware; unlike the disk, the shortest path between the two hands through the interior is large.

Note that computation of shortest-path distances in restricted to the interior of a non-convex triangulated surface is NP-hard (Canny and Reif 1987). Though the Steklov family of distances is computed without interior discretization, it behaves similarly to distances computed using the volumetric Laplacian.

7.4 Shape differences and variability
Rustamov et al. (2013) introduce a notion of shape differences based on an area-based inner product $b^a(u, v) = \langle u, v \rangle_{\Gamma}$ and a conformal inner product $b^c(u, v) = \langle \nabla u_1, \nabla v_2 \rangle_{\Gamma}$ between surface functions $u$ and $v$.

Similarly, we introduce a volume-aware inner product $b^v(u, v) := \langle \nabla E u, \nabla E v \rangle_{\Omega}$. Superficially this again could be viewed as the a substitution of the Laplacian $\mathcal{L}$ used for intrinsic shape differences with the DtN operator $\mathcal{S}$. After discretization, this new inner product induces a shape difference operator $\mathcal{D} = \mathcal{S}^T \mathcal{M} \mathcal{F} \mathcal{S}$ giving a functional map taking functions on the source $\mathcal{M}$ to functions on the target $\mathcal{N}$. This shape difference operator modifies the function $f$ such that,
8 CONCLUSION

The Dirichlet-to-Neumann operator provides an intuitive way to transfer the successes of intrinsic geometry processing to applications needing volumetric information. As we demonstrate using assorted applications in §7,DtN operators can substitute for Laplace–Beltrami operators mutatis mutandis to incorporate extrinsic geometry into existing algorithms and pipelines. The end result is an easily-applied and relatively intuitive extrinsic operator backed by theoretical understanding of its behavior on smooth domains. Our experiments further demonstrate that the boundary integral formulation of the DtN operator appears to encode meaningful geometric data on surfaces with boundary that may not enclose a volumetric region.

Our technique as it currently is implemented exhibits some limitations whose resolution will expand the applicability of our work.
Most prominently, basic BEM algorithms can fail when triangle meshes self-overlap, e.g., if two triangles intersect. This drawback potentially can be resolved using a more complex implementation that detects and resolves self-overlap before evaluating boundary integrals, e.g., using techniques recently introduced to the graphics community by Zhou et al. (2016). Another smaller drawback of our work is the assumption that volumes enclosed by surfaces are homogeneous. In physically-motivated contexts, it may be useful to incorporate anisotropy via modified Laplacian operators or by warping of the interior of the domain. This can be challenging using the boundary element method, which requires closed-form Green’s functions for the differential operators involved.

These potential extensions aside, future work on the computational side might focus on the efficiency of our technique. While asymptotically BEM matches if not surpasses the efficiency of FEM, our current BEM implementation could benefit from accelerated schemes for numerical quadrature and hierarchical matrix evaluation to bring this efficiency to practice. Additional speed could be gained by considering GPU-based implementations of BEM and eigenvalue algorithms, including the GPU-accelerated version of LOBPCG proposed in (Anzt et al. 2015); their blocked matrix-vector product has some rough similarity to the hierarchical matrix method used in our BEM implementation. GPU-accelerated BEM has been considered in several recent works including (Hamada 2011; Stock and Gharakhani 2010; Takahashi and Hamada 2009; Yokota et al. 2011), a developing topic in numerical analysis that will improve the practical aspects of our method as a side effect.

More broadly, our experiments with DtN operators and the Steklov eigenvalue problem reveal the value of considering extrinsic shape in geometry processing pipelines. This and other future approaches will bring a complete geometric characterization to the constellation of spectral and operator-based shape processing techniques.

ACKNOWLEDGEMENTS
The authors thank Mikhail Karpukhin for communicating the proof of Proposition 3.1. J. Solomon acknowledges funding from an MIT Skoltech Seed Fund grant (“Boundary Element Methods for Shape...
Fig. 22. Point-wise distortion, as well as principal distortion (top singular vector of shape difference operator) measured by the Steklov-based and Laplacian-based shape difference operators for a bending paper.

Fig. 23. Point-wise distortion measured by the Steklov-based and Laplacian-based shape difference operators. Our Steklov-based operator distinguishes bending direction of the cigar.

Fig. 24. PCA on collections of shape differences. Steklov-based shape difference reveals the extrinsic shape difference that conformal shape difference (Laplacian-based) fails to capture. For the bumpy cube array, Steklov-based shape difference operator successfully classifies all shapes into four clusters: inward and inward bumps, inward and outward bumps, outward and inward bumps, as well as outward and outward bumps.

REFERENCES

Yonathan Aflalo, Haim Brezis, and Ron Kimmel. 2015. On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences* 8, 2 (2015), 1141–1160.

Hartwig Anzt, Stanimire Tomov, and Jack Dongarra. 2015. Accelerating the LOBPCG method on GPUs using a blocked Sparse Matrix Vector Product. In *Proc. Symposium on High Performance Computing*. 75–82.

Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV Workshops*. 1626–1633.

Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Maks Ovsjanikov. 2013. An operator approach to tangent vector field processing. In *Computer Graphics Forum*, Vol. 32. 73–82.

Omri Azencot, Maks Ovsjanikov, Frédéric Chazal, and Mirela Ben-Chen. 2015. Discrete Derivatives of Vector Fields on Surfaces—An Operator Approach. *Transactions on Graphics* 34, 3 (2015), 29.

Omri Azencot, Orestis Vantzos, and Mirela Ben-Chen. 2016. Advection-Based Function Matching on Surfaces. In *Computer Graphics Forum*, Vol. 35. 55–64.

Omri Azencot, Steffen Weidemann, Maks Ovsjanikov, Max Wardetzky, and Mirela Ben-Chen. 2014. Functional fluids on surfaces. In *Computer Graphics Forum*, Vol. 33. 237–246.

Mikhail Belkin, Jian Sun, and Yusu Wang. 2008. Discrete Laplace operator on meshed surfaces. In *Proc. SODA*. 278–287.

Mikhail Belkin, Jian Sun, and Yusu Wang. 2009. Constructing Laplace operator from point clouds in $\mathbb{R}^d$. In *Proc. SODA*. 1031–1040.

Michele Benzi, Gene H Golub, and Jörg Liesen. 2005. Numerical solution of saddle point problems. *Acta Numerica* 14 (2005), 1–137.

S Bertoluzza. 2003. Non conforming domain decomposition: the Steklov–Poincaré operator point of view. In *Domain Decomposition Methods in Science and Engineering*. ACM Transactions on Graphics, Vol. 0, No. 0, Article 0. Publication date: 20XX.
Fig. 25. (a) 2D PCA plot of “ShapeDNAs” computed using Steklov and Laplacian eigenvalues. Shapes from the same category tend to cluster together. (b) For both Steklov and Laplacian eigenvalues, shapes from the same category have similar eigenvalues; the Steklov spectra cluster correctly but also distinguish between shapes within the same class.
Fig. 26. The Steklov eigenfunctions corresponding to the smallest five non-zero Steklov eigenvalues, with comparison to surface Laplacian eigenfunctions. Note that the Steklov eigenfunctions align with the local extrema of mean-curvature, suggesting its use as a segmentation clue.
Fig. 27. Results of segmentation by Steklov and Laplacian embedding.