thrive SIGGRAPH2019
LOS ANGELES • 28 JULY – 1 AUGUST

STEKLOV SPECTRAL GEOMETRY FOR EXTRINSIC SHAPE ANALYSIS

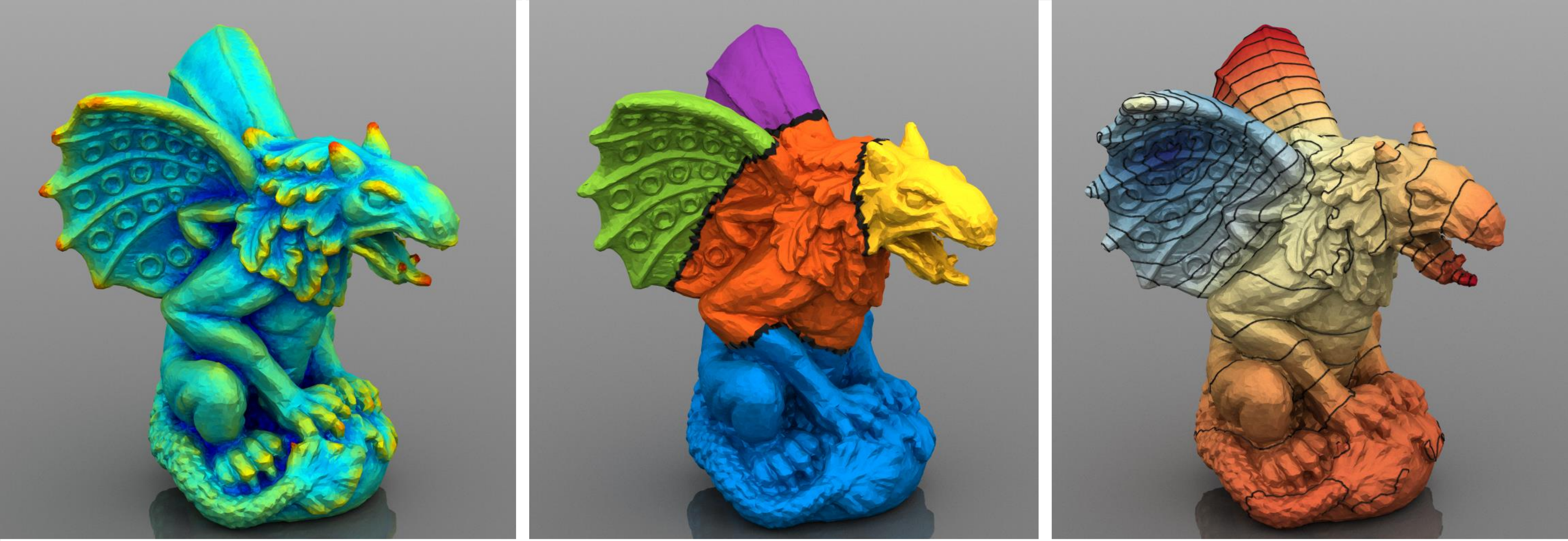
Yu Wang

Mirela Ben-Chen

Iosif Polterovich

Justin Solomon

MIT
Technion Israel Institute of Technology
Université de Montréal
CSAIL



Steklov Spectral Geometry for Extrinsic Shape Analysis

Yu Wang   Mirela Ben-Chen   Iosif Polterovich   Justin Solomon

MIT   Technion Israel Institute of Technology   Université de Montréal   CSAIL

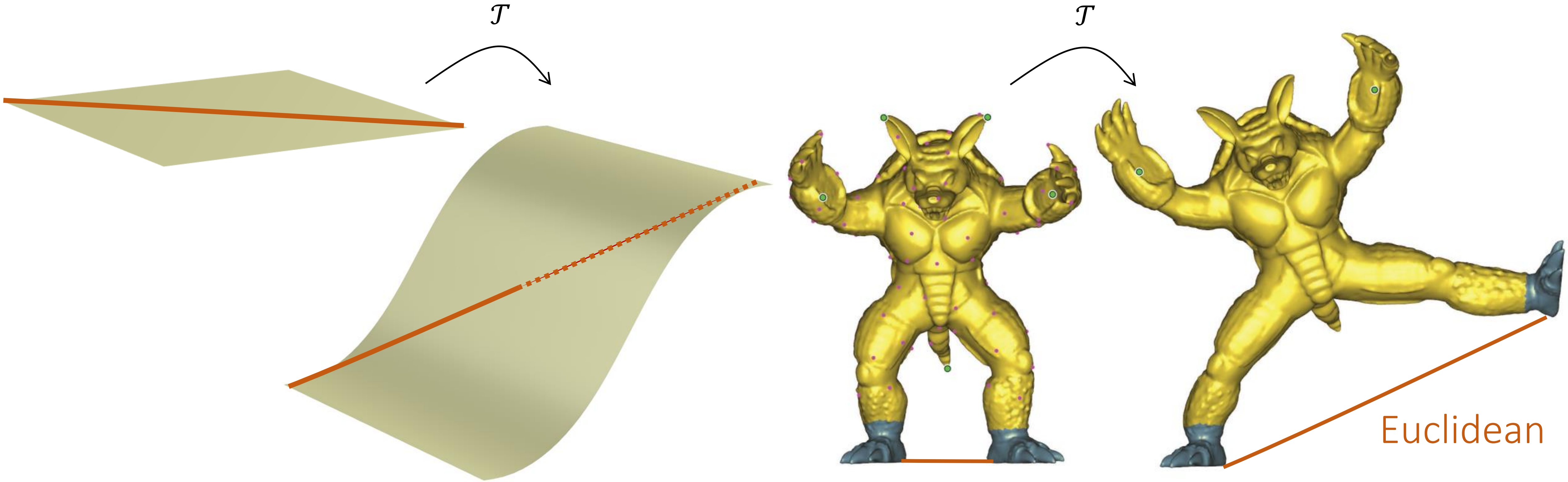
ACM Transactions on Graphics 38(1)
SIGGRAPH 2019
arXiv:1707.07070

Outline: Steklov Spectral Geometry for Extrinsic Shape Analysis

- Background in extrinsic and intrinsic geometry.
- Our solution: an extrinsic geometric operator.
- Theoretical properties and empirical behaviors of our operator.
- A brief look at the implementation details.

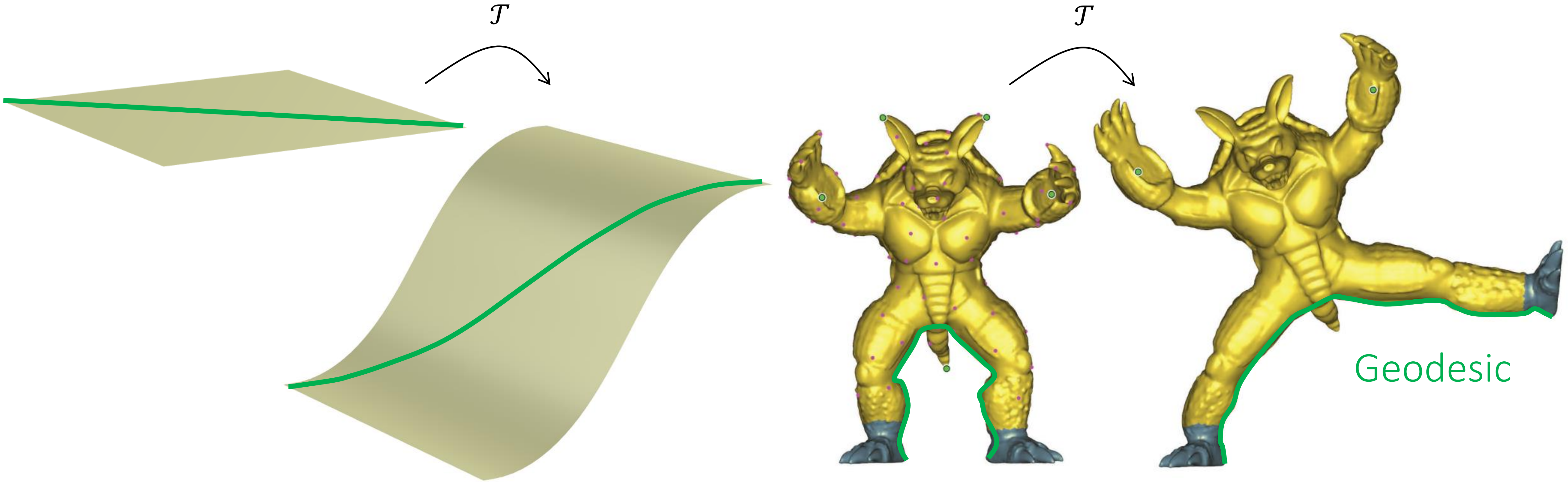
Extrinsic Geometry vs. Intrinsic Geometry

- Extrinsic geometry cares about spatial embedding of the shape.



Extrinsic Geometry vs. Intrinsic Geometry

- Extrinsic geometry cares about spatial embedding of the shape.
- Intrinsic geometry studies properties that can be measured without leaving the surface, e.g. geodesic distances.

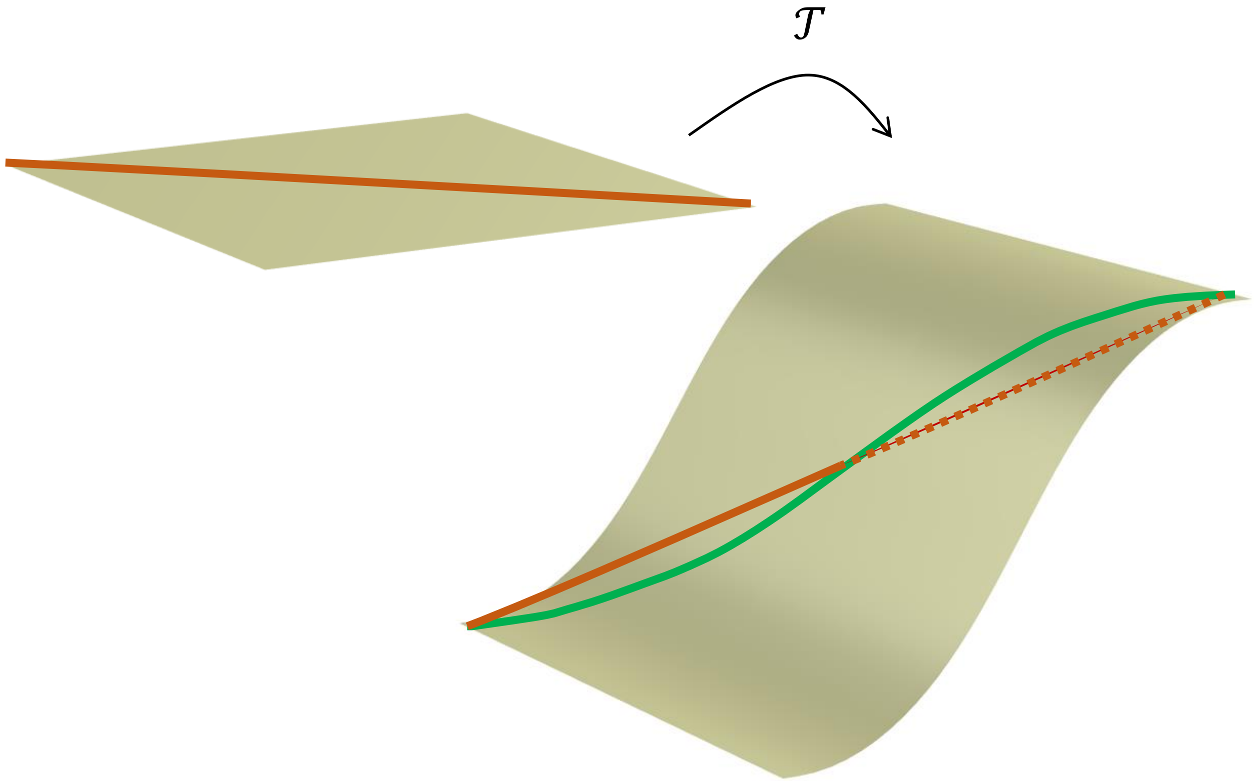


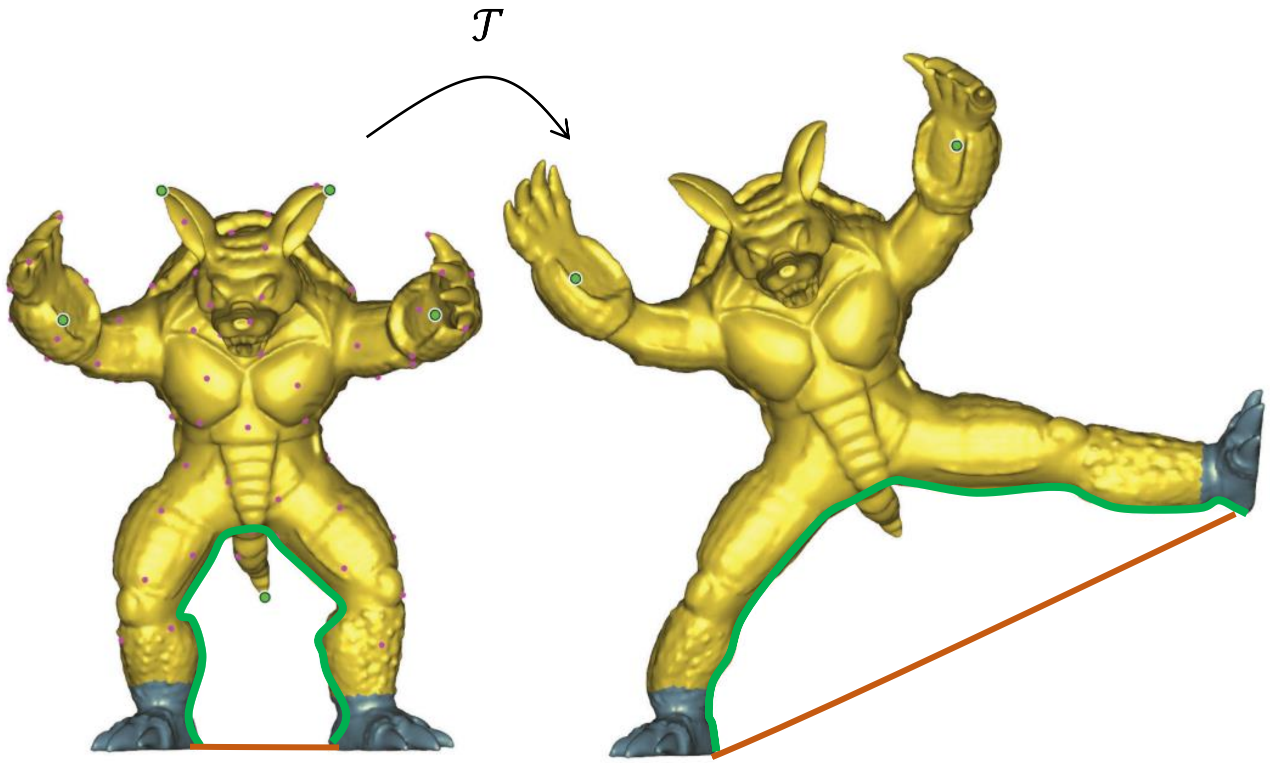
Review: Motivations in Intrinsic Shape Analysis

Intrinsic approaches are invariant to isometry ("pose invariant").

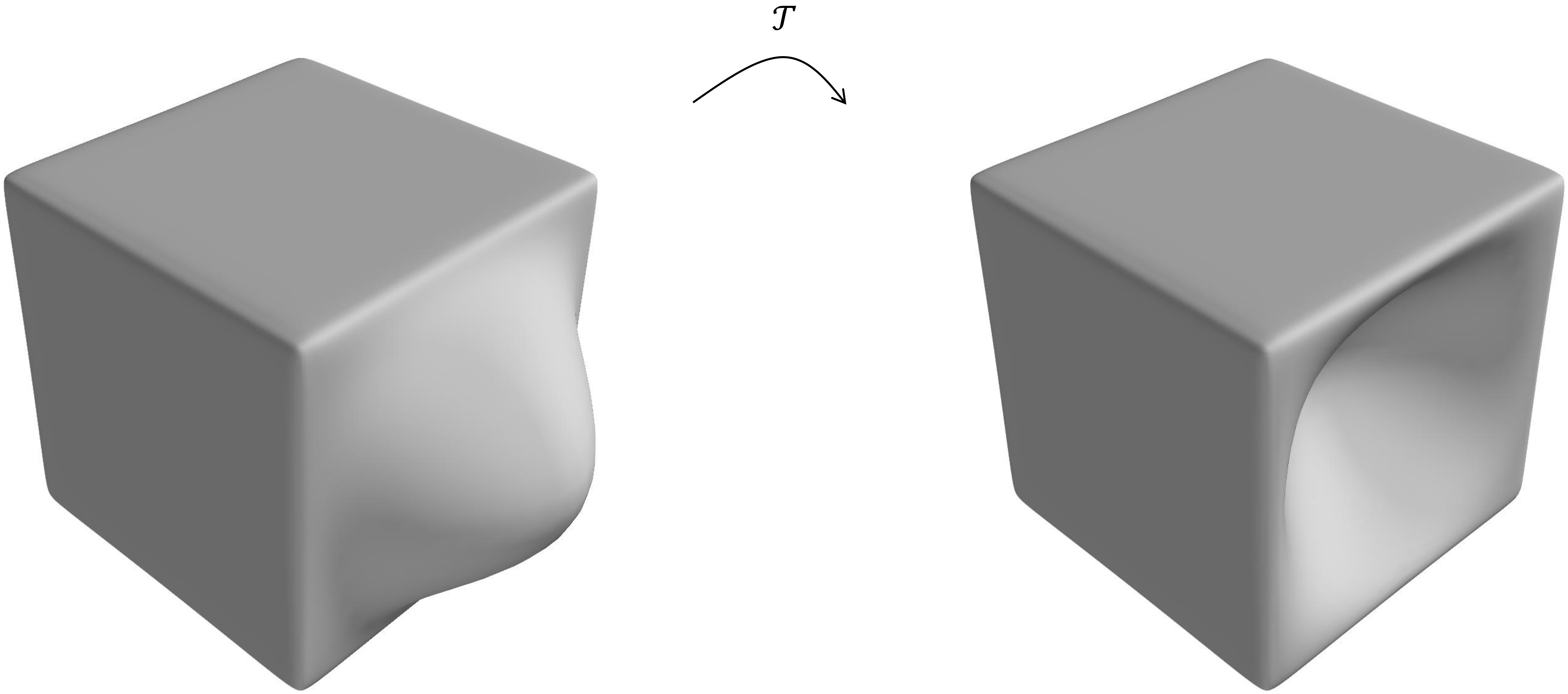
Real-world objects are usually subject to (near-) isometries.

Isometry $\mathcal{T}$: length-preserving map

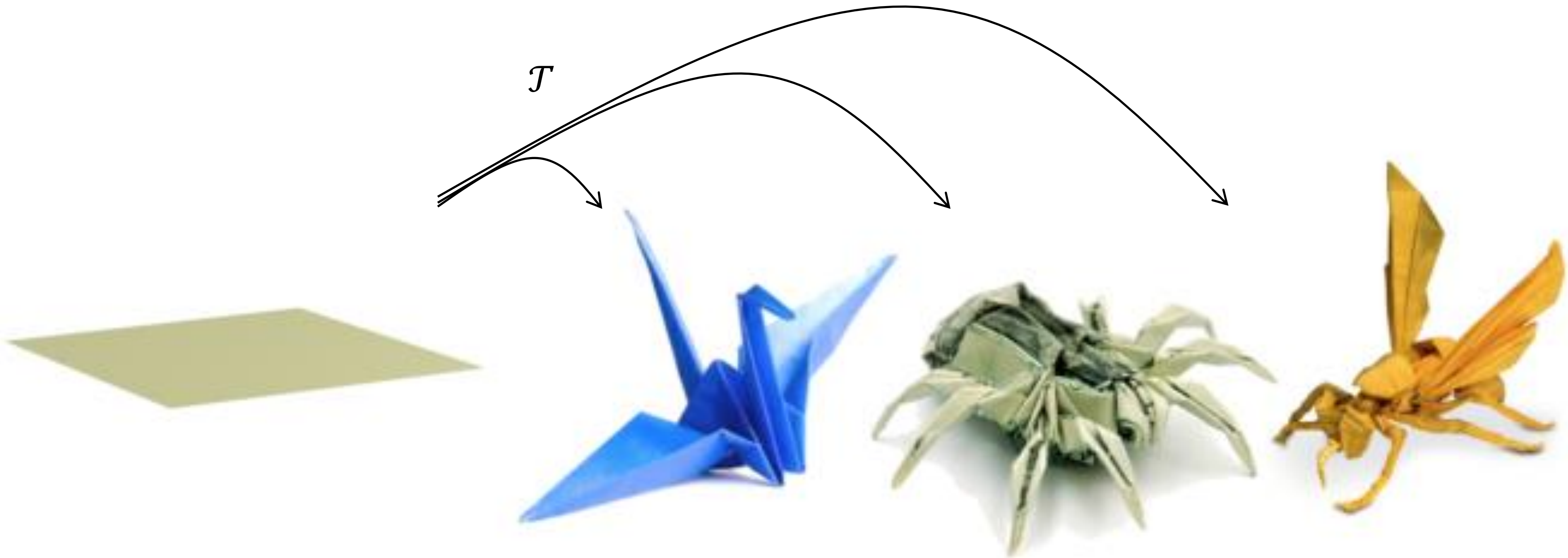




Notion of Intrinsic Geometry can be Counterintuitive



Intrinsic Information is Incomplete

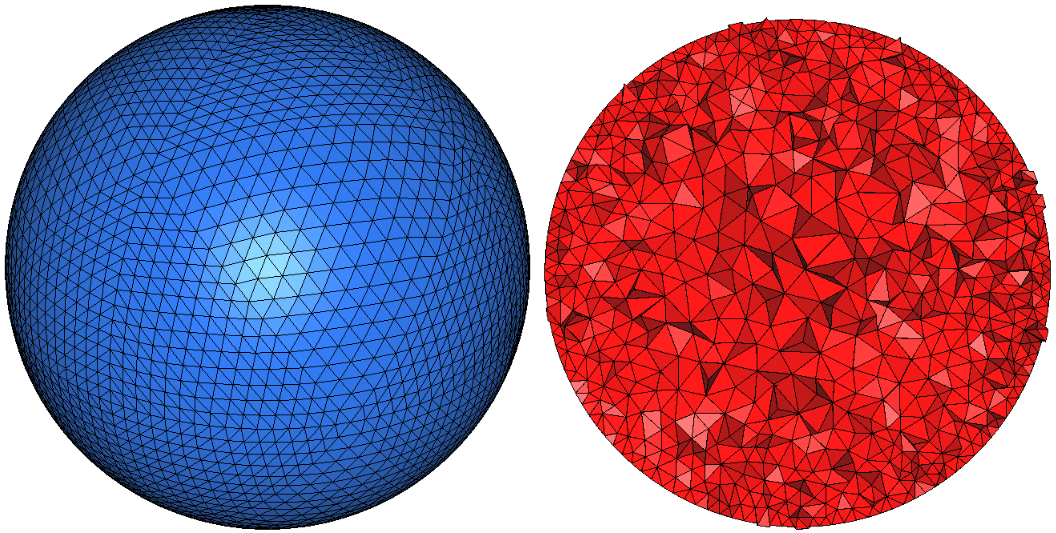


Intrinsic geometry: any origami equivalent to a piece of flat paper!

Origami images from http://image.google.com/

Existing Work in Extrinsic Geometry

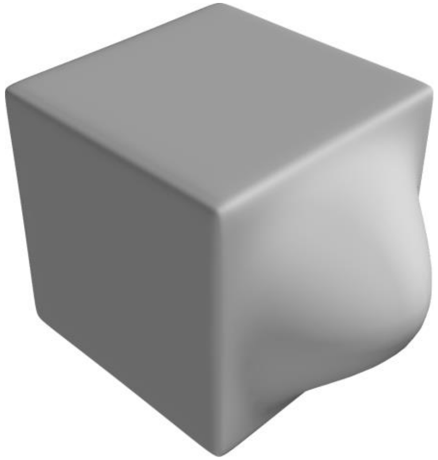
- Extrinsic histogram:
  - SHOT [Tombari et al. 2010]
  - D2 descriptors [Osada et al. 2002]
- Offset surface: [Corman et al. 2017]
- Volume–based:
  - [Raviv et al. 2010]
  - [Litman et al. 2012]
  - [Wang and Wang 2015]
  - [Patane 2015]
  - [Rustamov 2011]

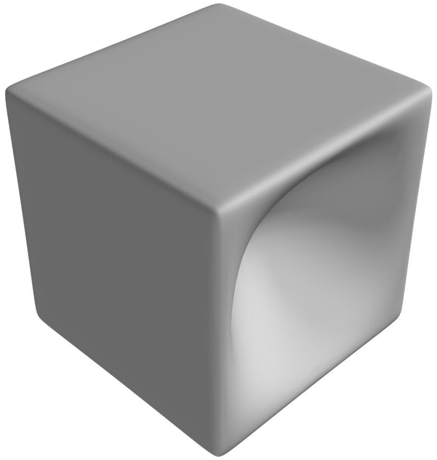


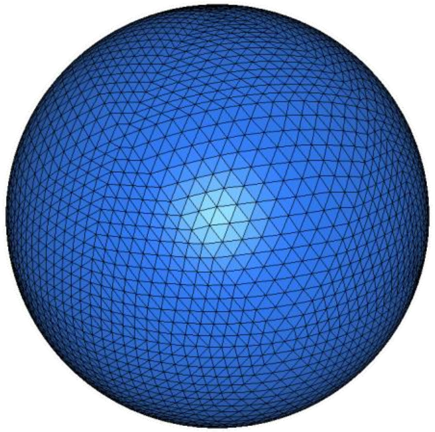
$$\Delta_{\mathbb{R}^3} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

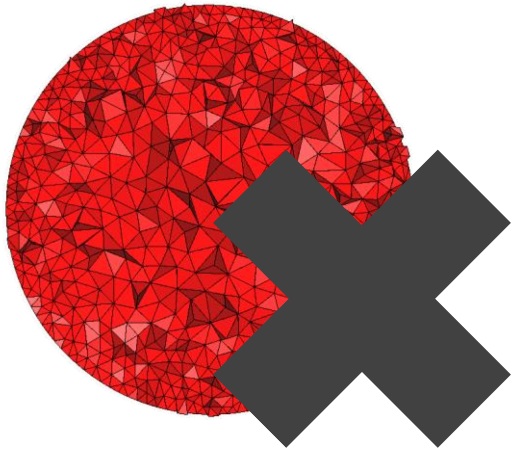
- "Volumetrization" is hard: clean input mesh
- Inability to handle open surfaces/triangle soups
- Inconsistency unless super dense volume mesh
- Cannot attribute shape difference onto surface

Our Problem: Extrinsic Geometry Analysis from a Boundary Representation



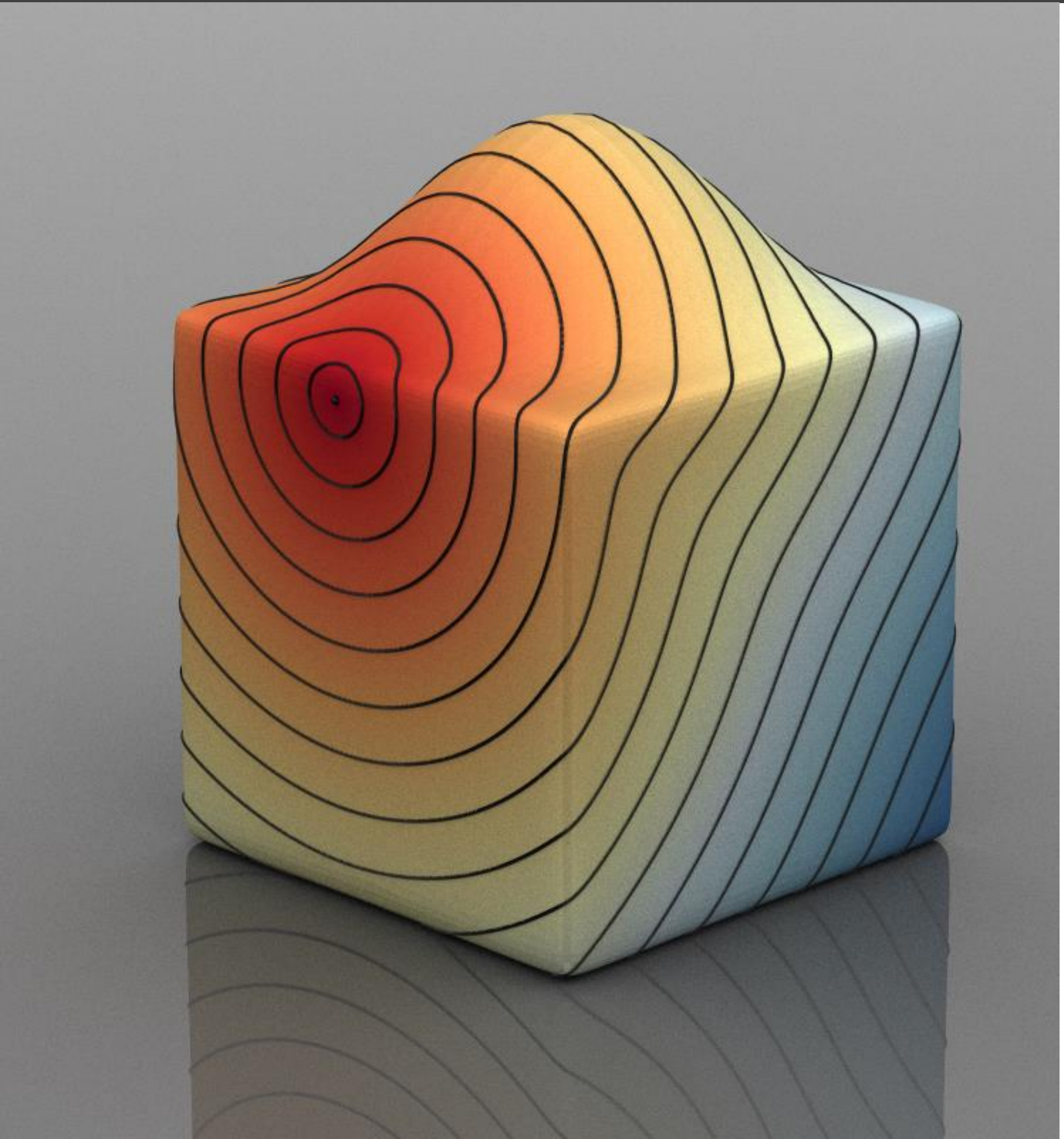






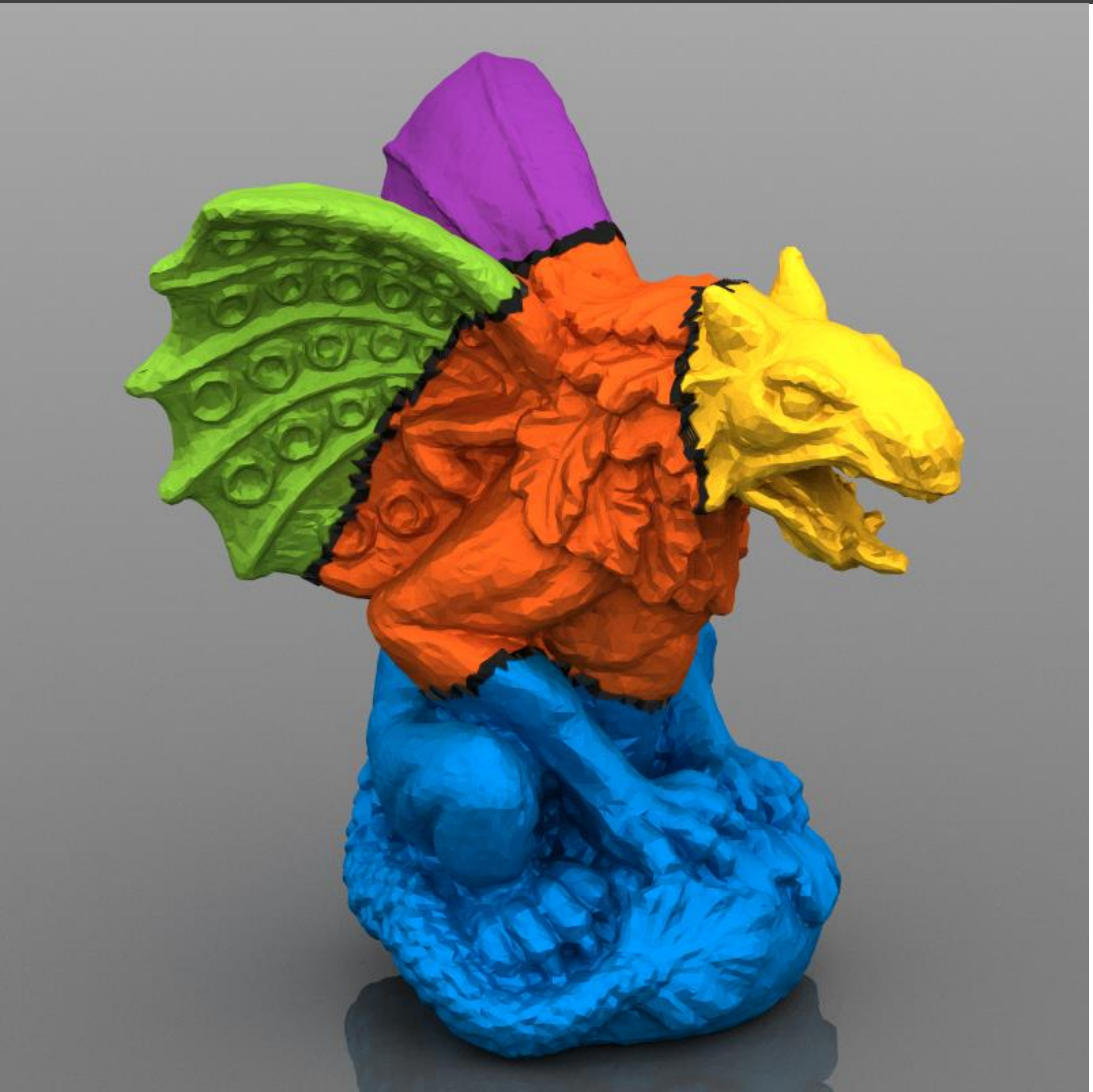
Origami images from http://image.google.com/

Review: (Intrinsic) Shape Analysis



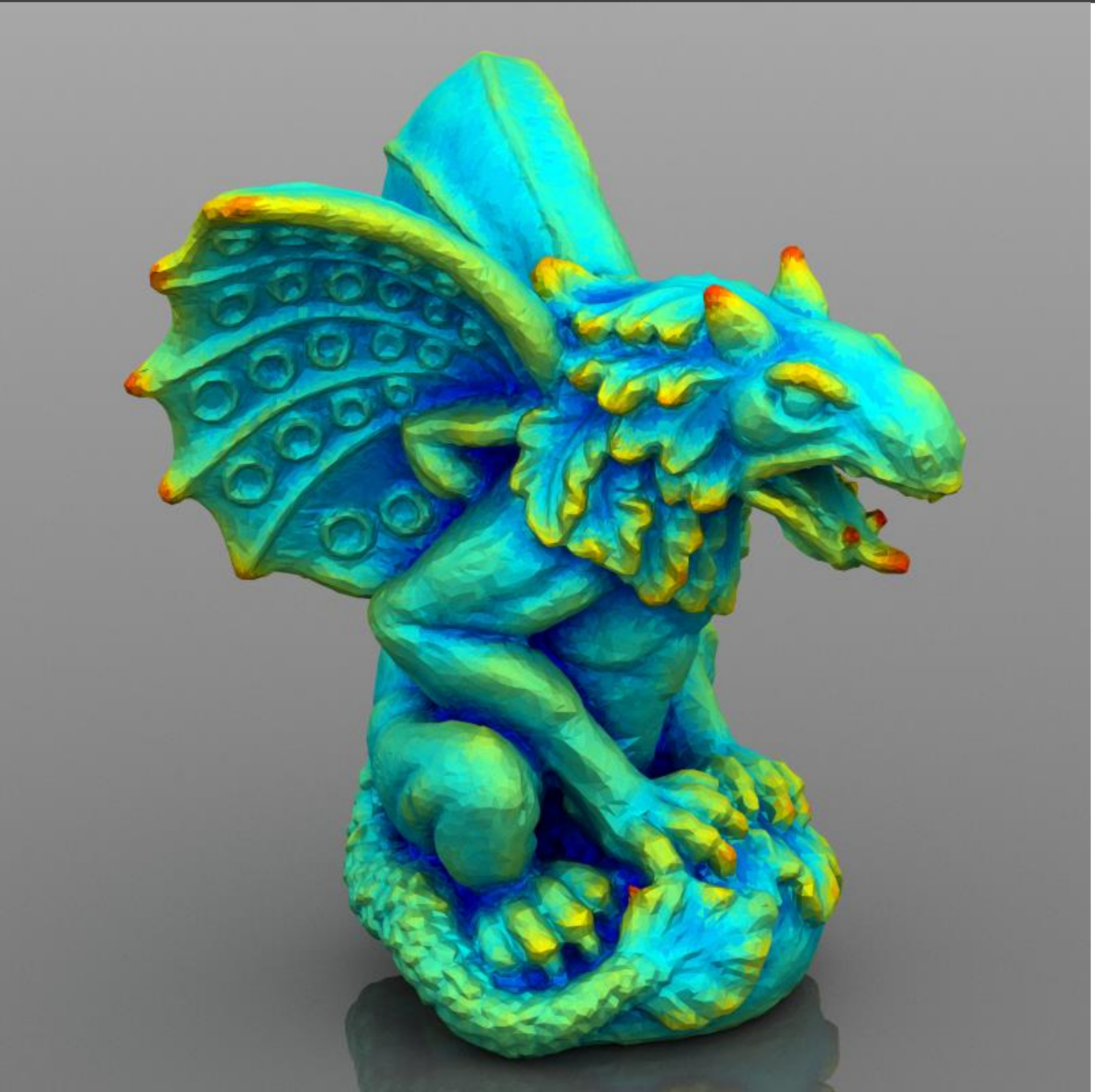
- **Distance** [Lipman et al. 2010; Crane et al. 2013]

Review: (Intrinsic) Shape Analysis



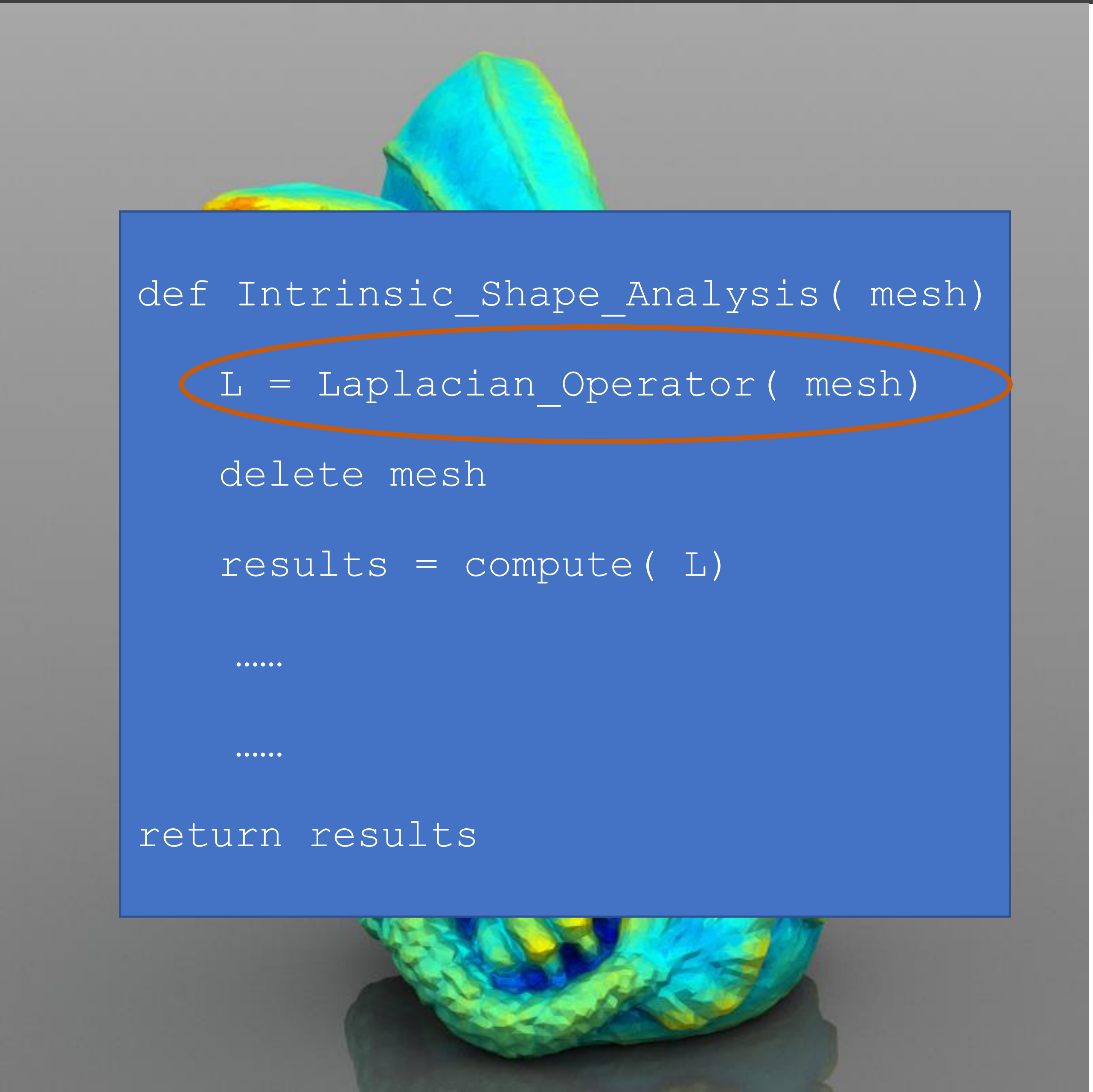
- Distance [Lipman et al. 2010; Crane et al. 2013]
- **Segmentation** [Reuter et al. 2009]

Review: (Intrinsic) Shape Analysis



- Distance [Lipman et al. 2010; Crane et al. 2013]
- Segmentation [Reuter et al. 2009]
- **Shape description** [Sun et al. 2009]

Review: (Intrinsic) Shape Analysis



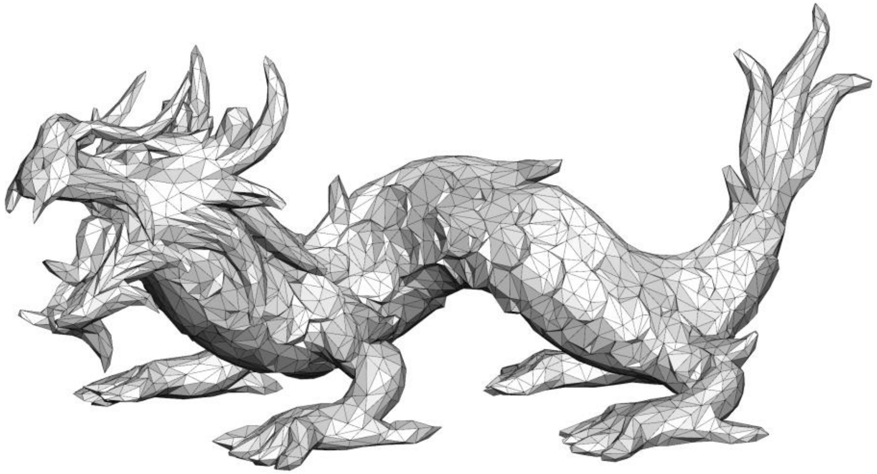
```
def Intrinsic_Shape_Analysis( mesh)
   L = Laplacian_Operator( mesh)
   delete mesh
   results = compute( L)
    ……
    ……
return results
```

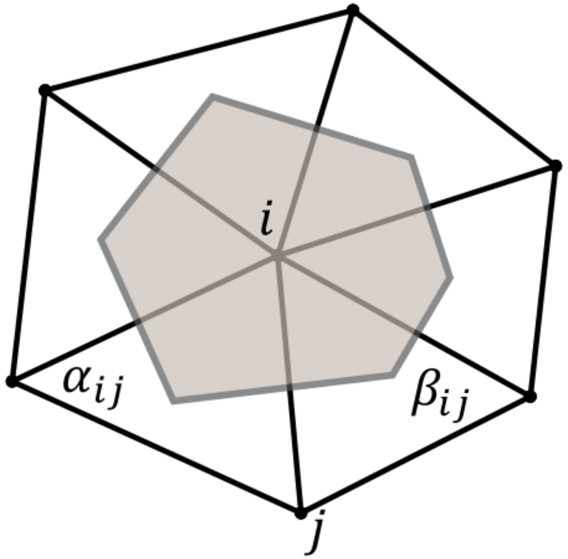
- Distance [Lipman et al. 2010; Crane et al. 2013]
- Segmentation [Reuter et al. 2009]
- Shape description [Sun et al. 2009]
- Shape retrieval [Bronstein et al. 2011]
- Correspondence [Ovsjanikov et al. 2012]
- Shape exploration [Rustamov et al. 2013]
- Vector field processing [Azencot et al. 2013]
- Simulation [Azencot et al. 2014]
- Deformation [Boscaini et al. 2015]

Laplacian Operator/Matrix Capturing Intrinsic Geometry

$$\mathbf{L} \in \mathbb{R}^{n \times n}$$

- The Laplacian operator (i.e. matrix) encodes the surface up to isometry.
- n: #vertices.





$$\mathbf{L}_{ij} = \begin{cases} \frac{1}{2\mathbf{A}_i}\left(\cot \alpha_{ij} + \cot \beta_{ij}\right) & \text{if } \{i, j\} \text{ is an edge} \\ -\sum_{j \neq i} \mathbf{L}_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{A}_i$: the area associated to vertex i.

Laplacian: Definition

Laplace operators in the Euclidean space.

- $\mathbb{R}^2$: $\Delta_{\mathbb{R}^2} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ (for images and 2D graphics)

Generalization from planar domain to a curved domain

Laplace-Beltrami operator (Laplacian) on manifold $\mathcal{M}$.

- $\mathcal{M}$: $\Delta_{\mathcal{M}} = \frac{1}{\sqrt{|\det g|}} \partial_i \left(\sqrt{|\det g|} g^{ij} \partial_j \right)$ (for surface $\mathcal{M}$)

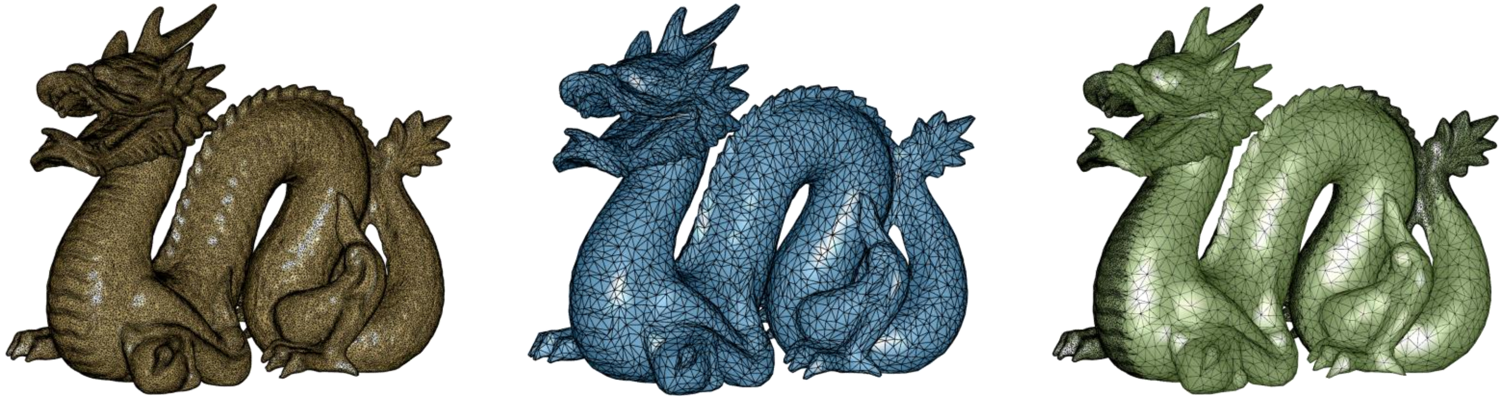
Laplacian

where $g \in \mathbb{R}^{2\times 2}$ is the metric.

Why the (Laplacian) Operator Approach?

$$\mathrm{L} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$$

- The Laplacian operator (i.e. matrix) encodes the surface up to isometry.
- n: #vertices.



Original mesh Coarse mesh Unbalanced mesh

- Invariance to shape representation
  - Triangle meshes
  - Quad meshes
  - Polygon meshes
  - Point clouds
  - Triangle soups
- As the discretization of a *continuous* operator

Our Goal: An Operator/Matrix Capturing Extrinsic Geometry

$$S \in \mathbb{R}^{n \times n}$$

- We are looking for Some operator (i.e. matrix) S encodes extrinsic geometry.
- n: #vertices.

An operator-based approach to systematically introduce extrinsic geometry to many tasks

```
     Extrinsic
def ~~Intrinsic~~_Shape_Analysis( mesh)

    L = ~~Laplacian~~_Operator( mesh)
        Some-Extrinsic-Operator
    delete mesh

    results = compute( L)

     ……

     ……

return results
```

Dirichlet-to-Neumann (DtN) Operator $\mathcal{S}$

Consider a volume Ω bounded by the surface $\Gamma = \partial\Omega$.

$g(\partial\Omega)$

Dirichlet-to-Neumann (DtN) Operator $\mathcal{S}$

Consider a volume Ω bounded by the surface $\Gamma = \partial\Omega$.

$$\begin{cases} \Delta u(\mathbf{x}) = 0 & \mathbf{x} \in \Omega \\ u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x} \in \partial\Omega \end{cases}$$

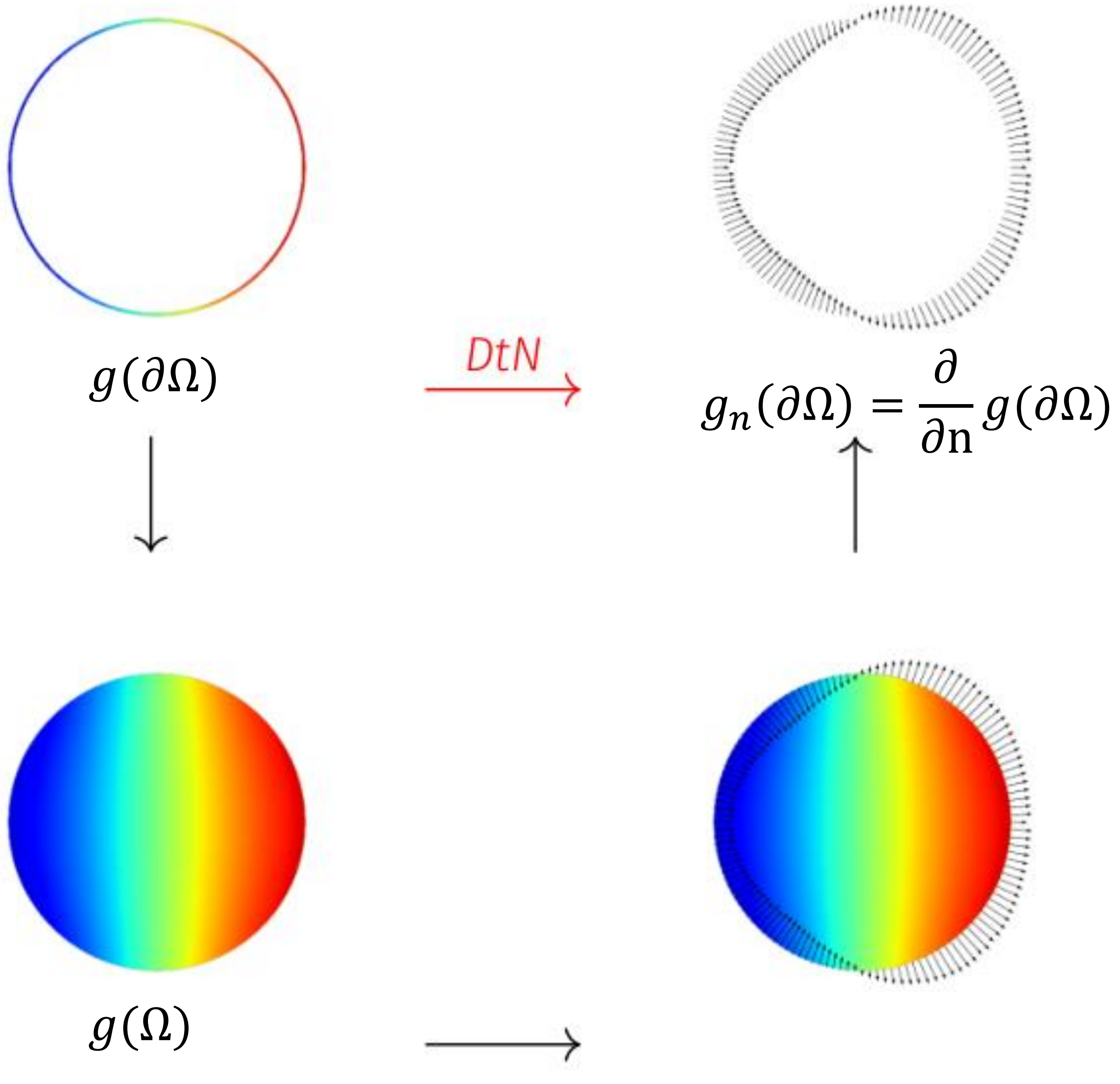
where $g(\Gamma)$ is Dirichlet data

Neumann data $g_n = \frac{\partial}{\partial n} u(\Gamma)$

Dirichlet-to-Neumann (DtN) operator:

$$\mathcal{S} := g \mapsto g_n$$

Also known as the Steklov-Poincaré operator.

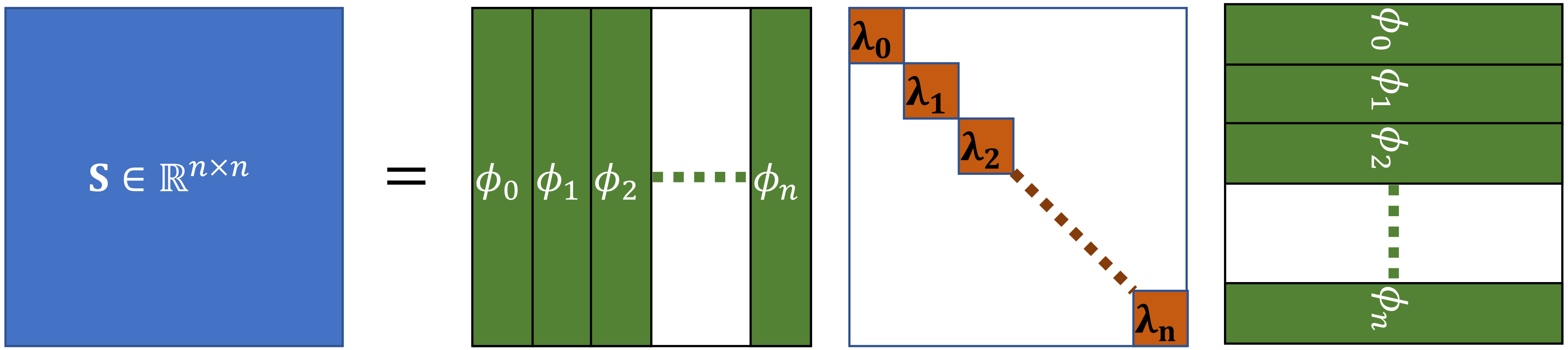


DtN operator and Steklov eigenvalue problem

- Discrete Dirichlet-to-Neumann operator: $\boldsymbol{S} \in \mathbb{R}^{n \times n}$.

n: number of vertices

- $\boldsymbol{S}$ is symmetric and positive semidefinite.



- This eigenvalue problem of $\boldsymbol{S}$ is known as the **Steklov eigenvalue problem.**

DtN Operator $\mathcal{S}$ and Extrinsic Geometry

- The DtN operator $\mathcal{S}$ encodes extrinsic geometry.
  - The surface can be recovered from its DtN operator up to rigid motion.

Theorem

Denote $\Omega_1, \Omega_2 \subseteq \mathbb{R}^3$ as two domains, and $\alpha : \Omega_1 \to \Omega_2$ is a bijection. Under proper assumptions, if the two domains have the same Dirichlet-to-Neumann operators (under map composition), then α must be a rigid motion.[5]

[5] M. Lassas and G. Uhlmann (2001). "On determining a Riemannian manifold from the Dirichlet-to-Neumann map". In: Annales scientifiques de l'Ecole normale supérieure. Vol. 34. 5, pp. 771–787.

DtN Operator $\mathcal{S}$ and Extrinsic Geometry

- The DtN operator $\mathcal{S}$ encodes extrinsic geometry.
  - The surface can be recovered from its DtN operator up to rigid motion.
  - The DtN operator captures critical extrinsic quantities like the mean curvature.

For smooth domains in $\mathbb{R}^3$, the Steklov heat kernel admits the asymptotic expansion as $t \to 0^+$ [Polterovich and Sher 2015]

$$e^{-t\mathcal{S}}(x,x) = \sum_{i=0}^{\infty} e^{-t\lambda_i}\phi_i(x)^2 \sim \sum_{k=0}^{\infty} a_k(x)t^{k-2} + \sum_{l=1}^{\infty} b_l(x)t^l \log t,$$

$$a_0(x) \equiv \frac{1}{2\pi}$$

$$a_1(x) = \frac{H(x)}{4\pi}$$

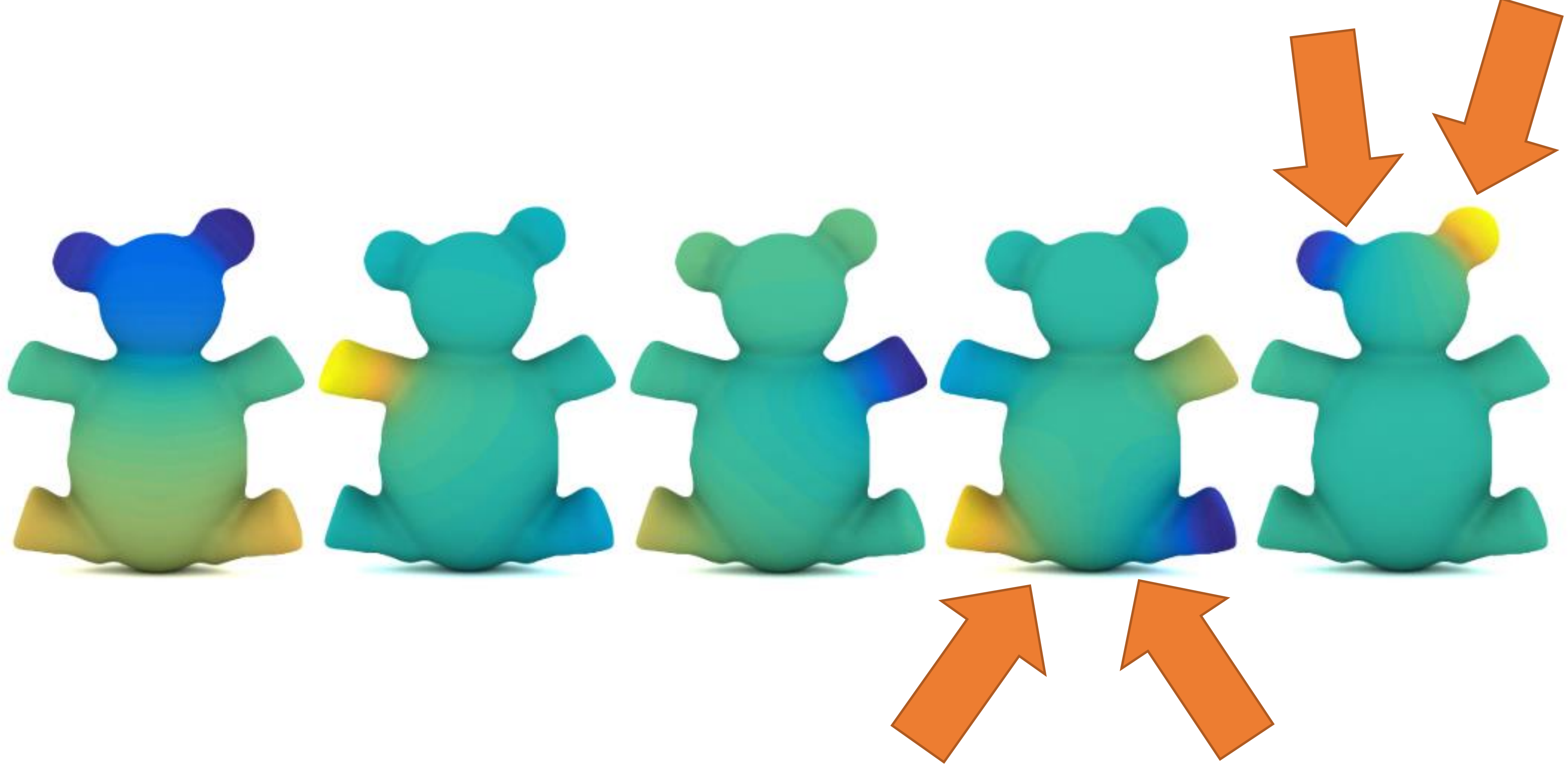
$$a_2(x) = \frac{1}{16\pi}\left(H(x)^2 + \frac{K(x)}{3}\right)$$

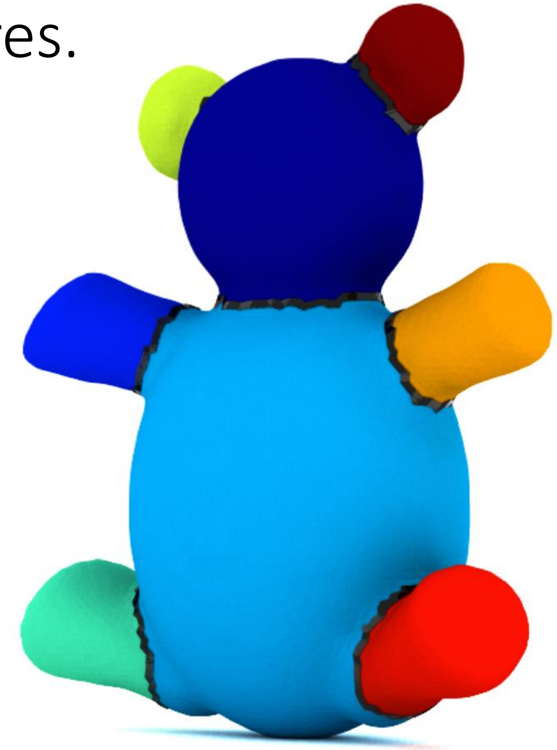
$H(x)$: mean curvature

$K(x)$: Gaussian curvature

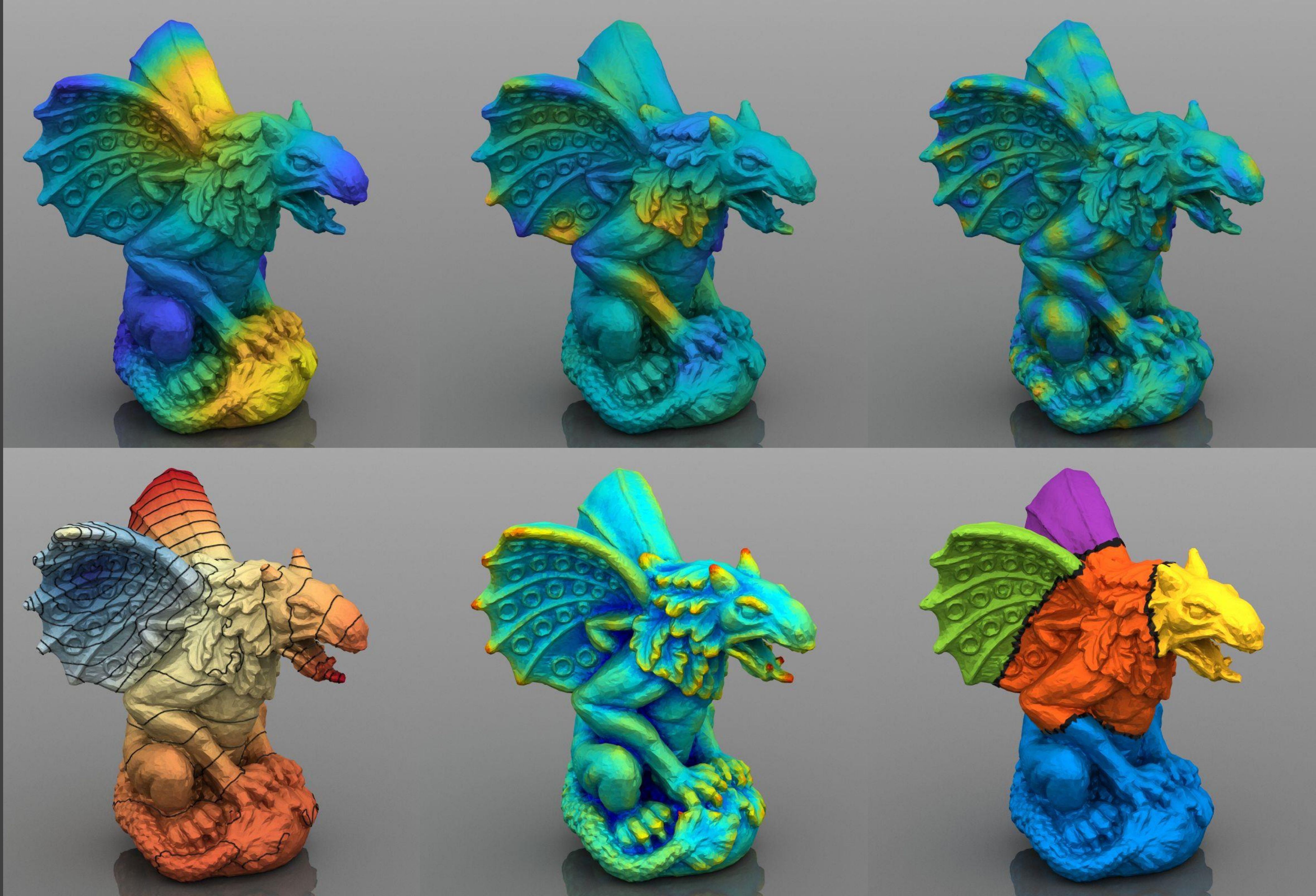
Steklov Eigenfunctions and Extrinsic Geometry

Level sets of Steklov eigenfunctions conform to mean curvatures.





Quantities Defined using Eigenfunctions



Kernel-based Descriptors

Steklov

~~Laplacian~~ eigenfunctions/eigenvalues

Heat kernel:

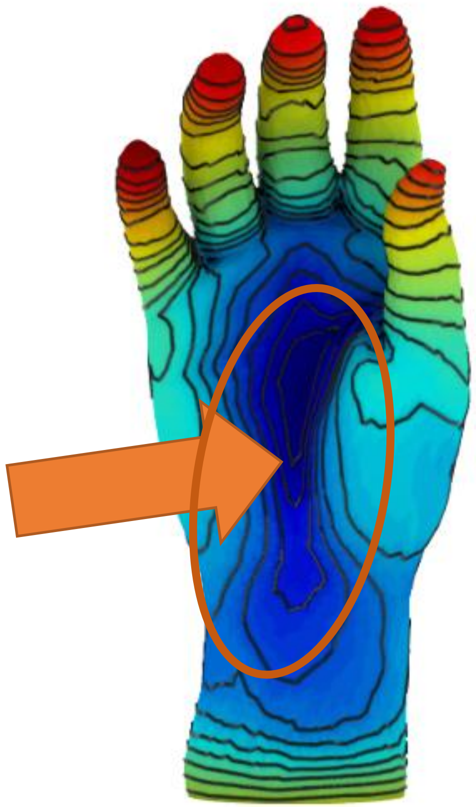
$$k_t(x,y) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(x) \phi_j(y)$$

Heat kernel signature [Sun et al. 2009]:

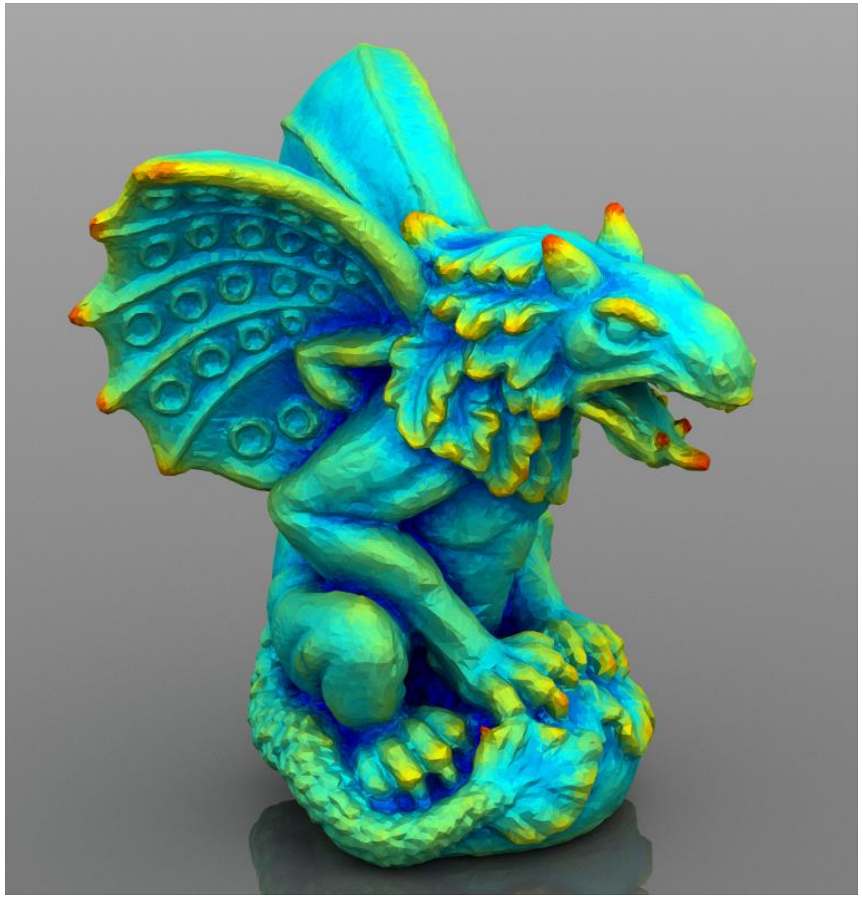
$$h_t(x) = k_t(x,x)$$

Heat Kernel Signature

Laplacian

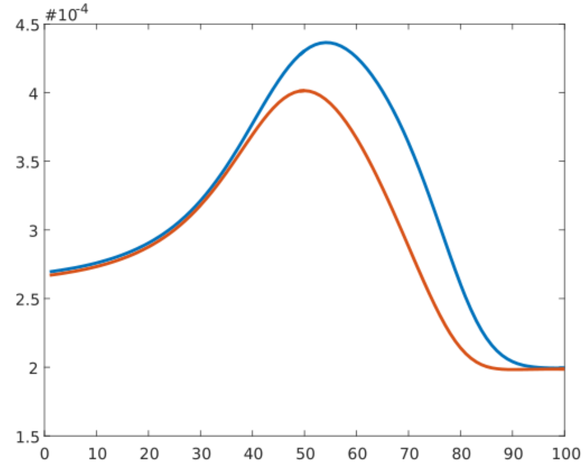


Steklov

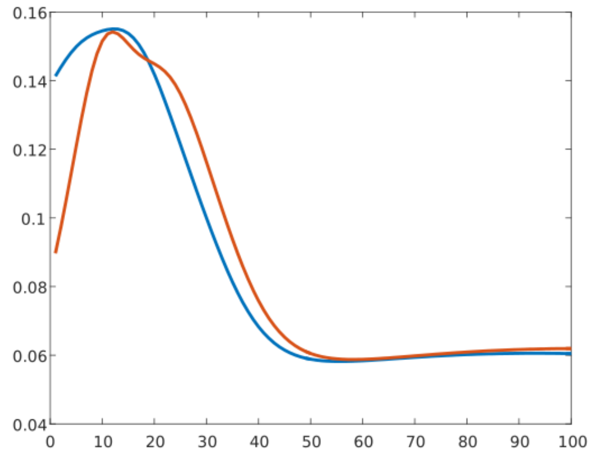


Steklov heat kernel signature as a “multi-scale mean curvature”

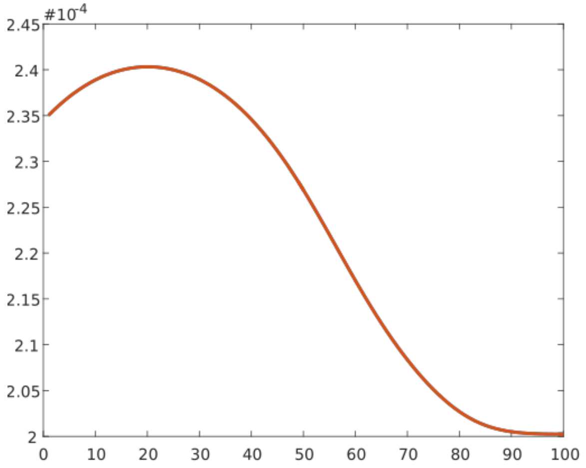
Heat Kernel Signature $h_t(x)$



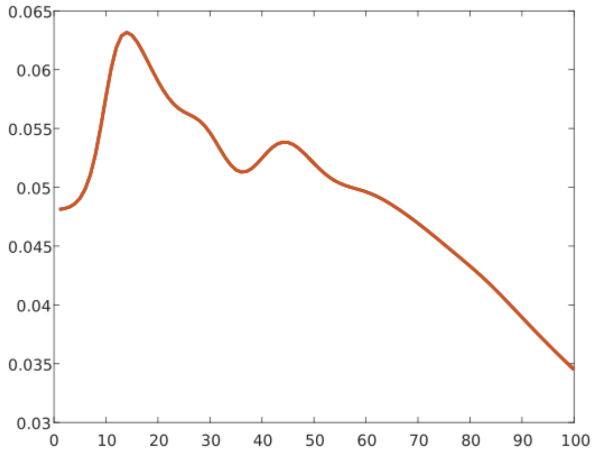
(a) Steklov Scaled-HKS



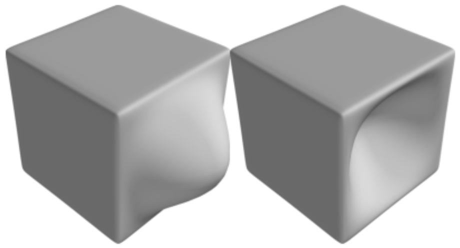
(b) Steklov WKS



(a) Laplacian Scaled-HKS

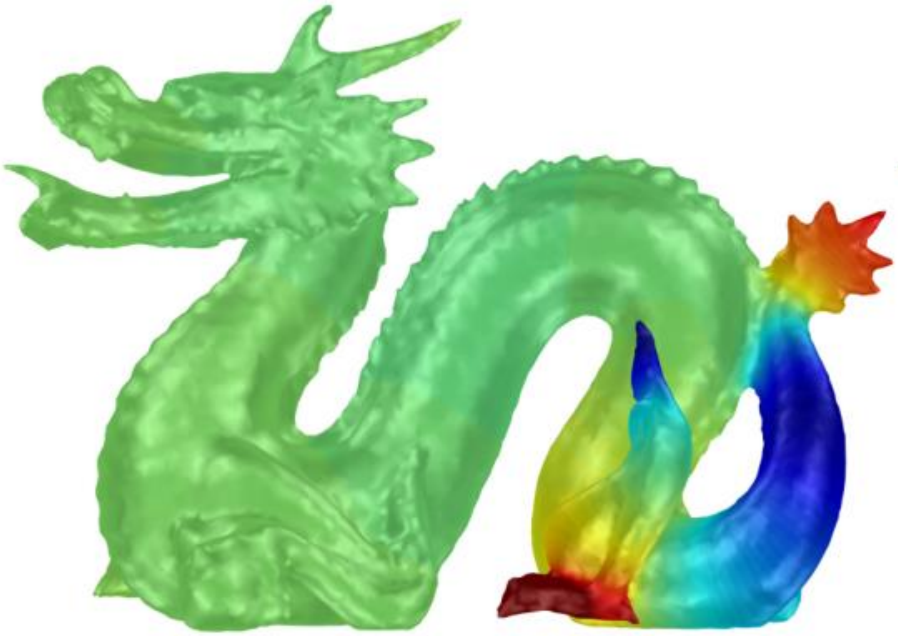


(b) Laplacian WKS

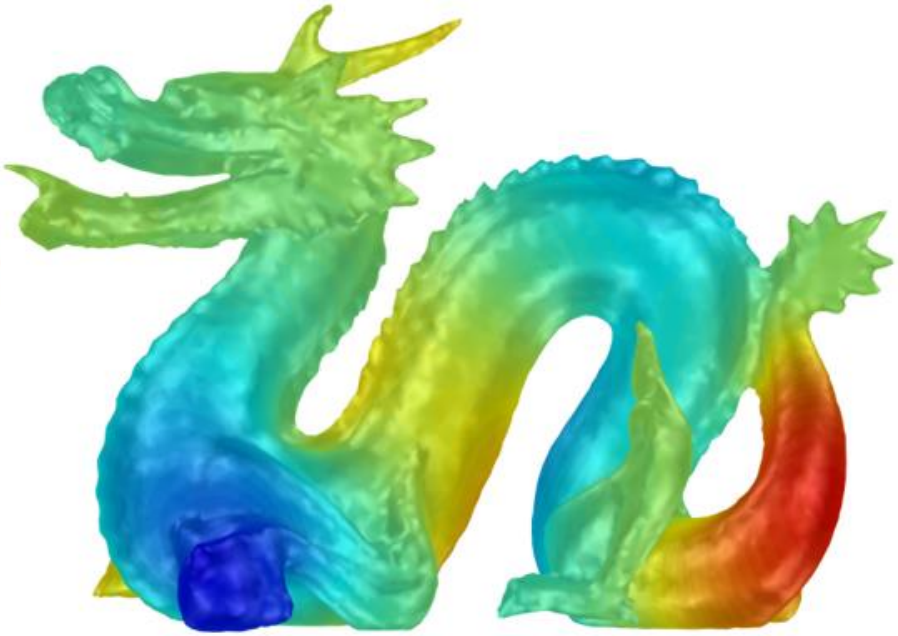


Steklov Spectrum

The 10th eigenfunction.



Steklov

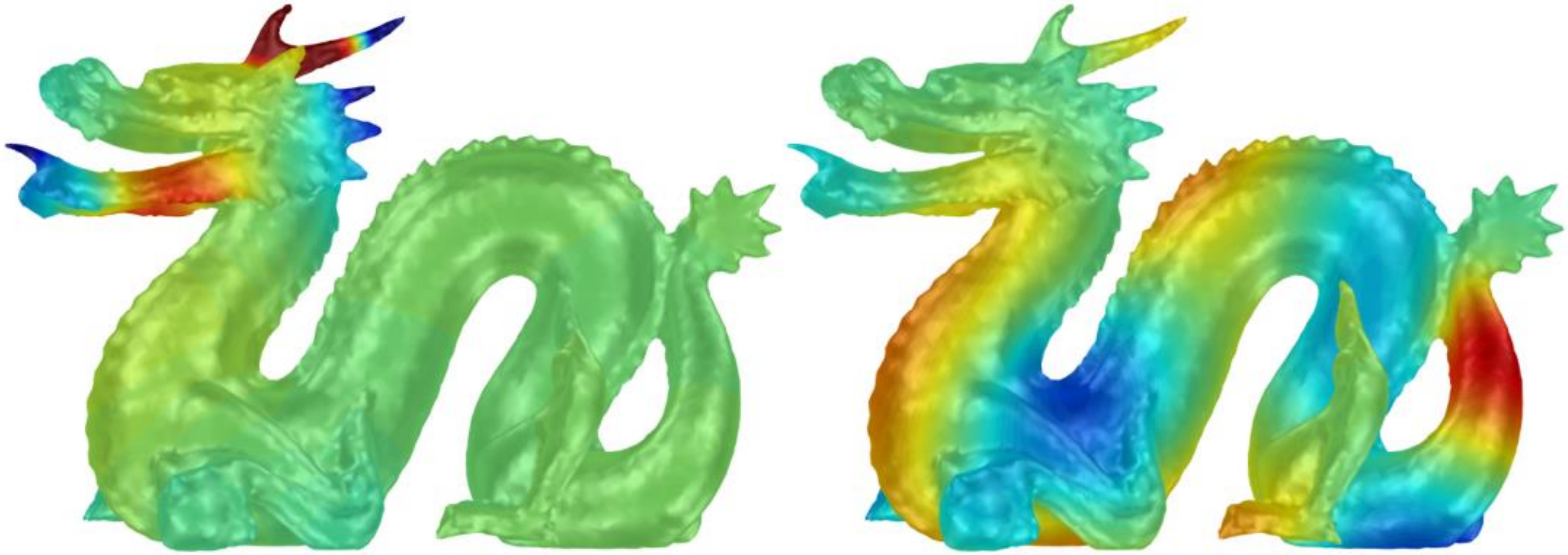


Laplacian

Comparison of eigenfunctions.

Steklov Spectrum

The 20th eigenfunction.

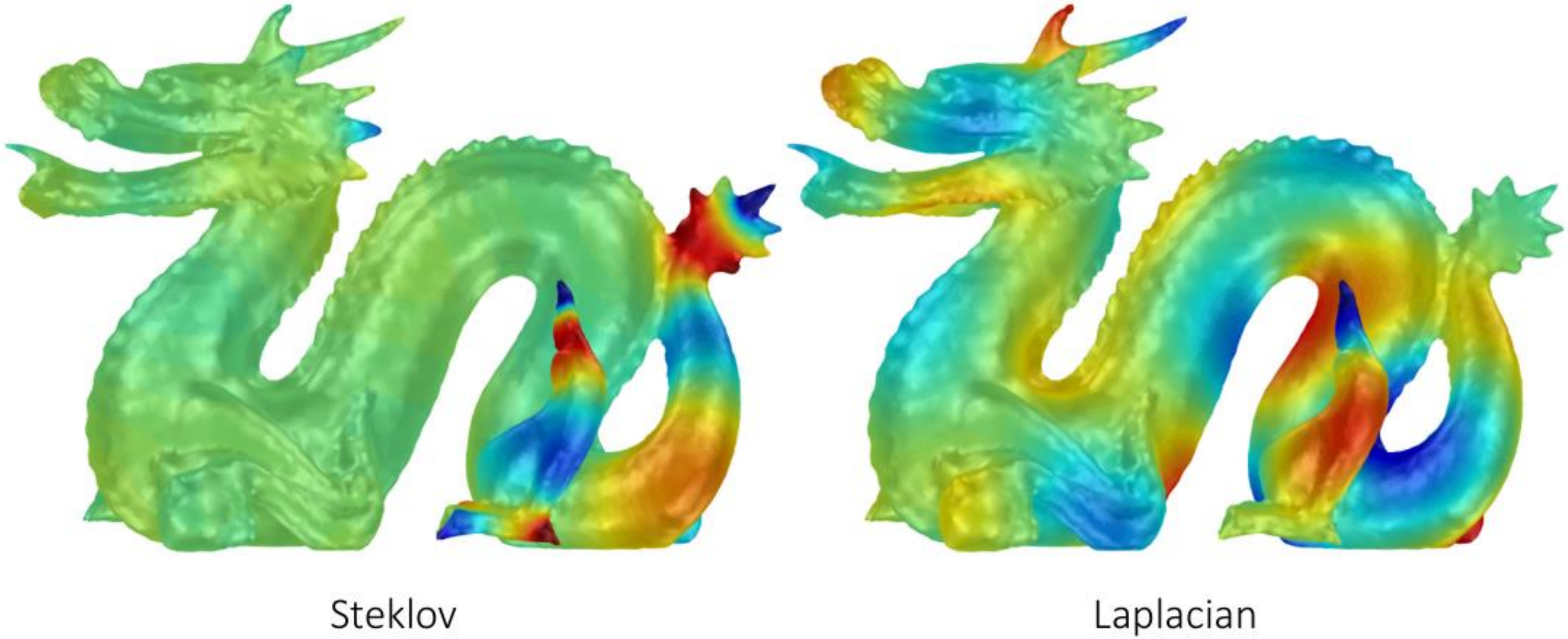


Steklov Laplacian

Comparison of eigenfunctions.

Steklov Spectrum

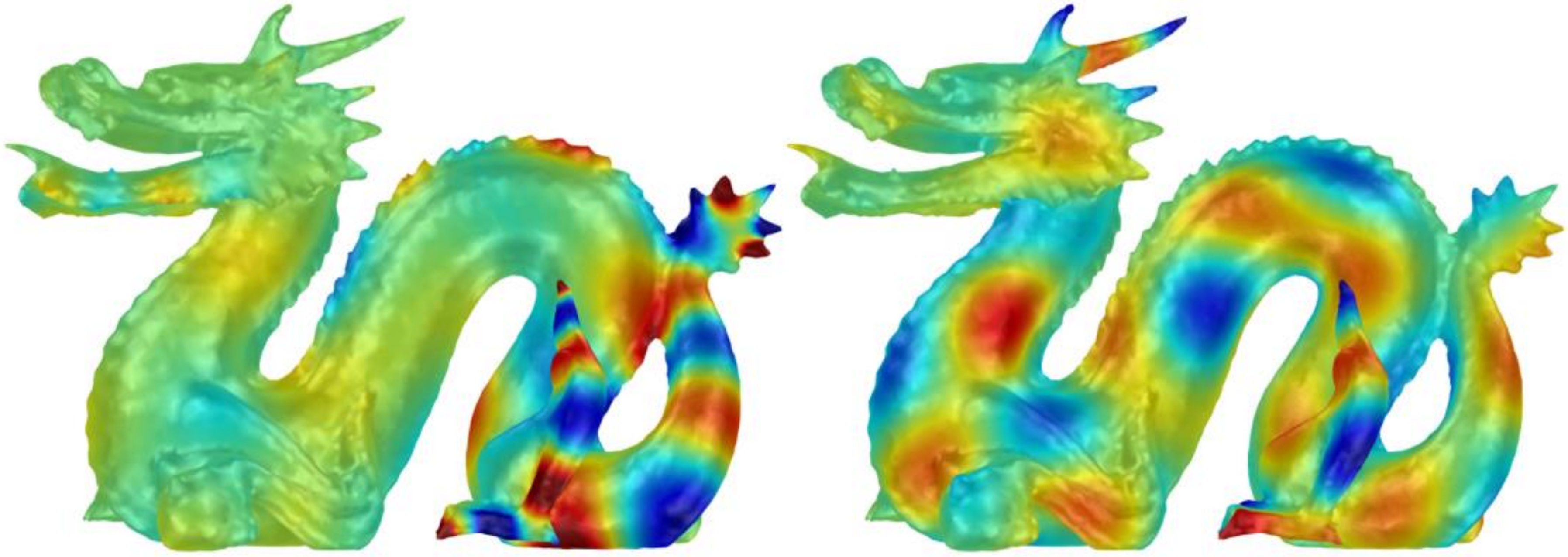
The 40th eigenfunction.



Comparison of eigenfunctions.

Steklov Spectrum

The 100th eigenfunction.

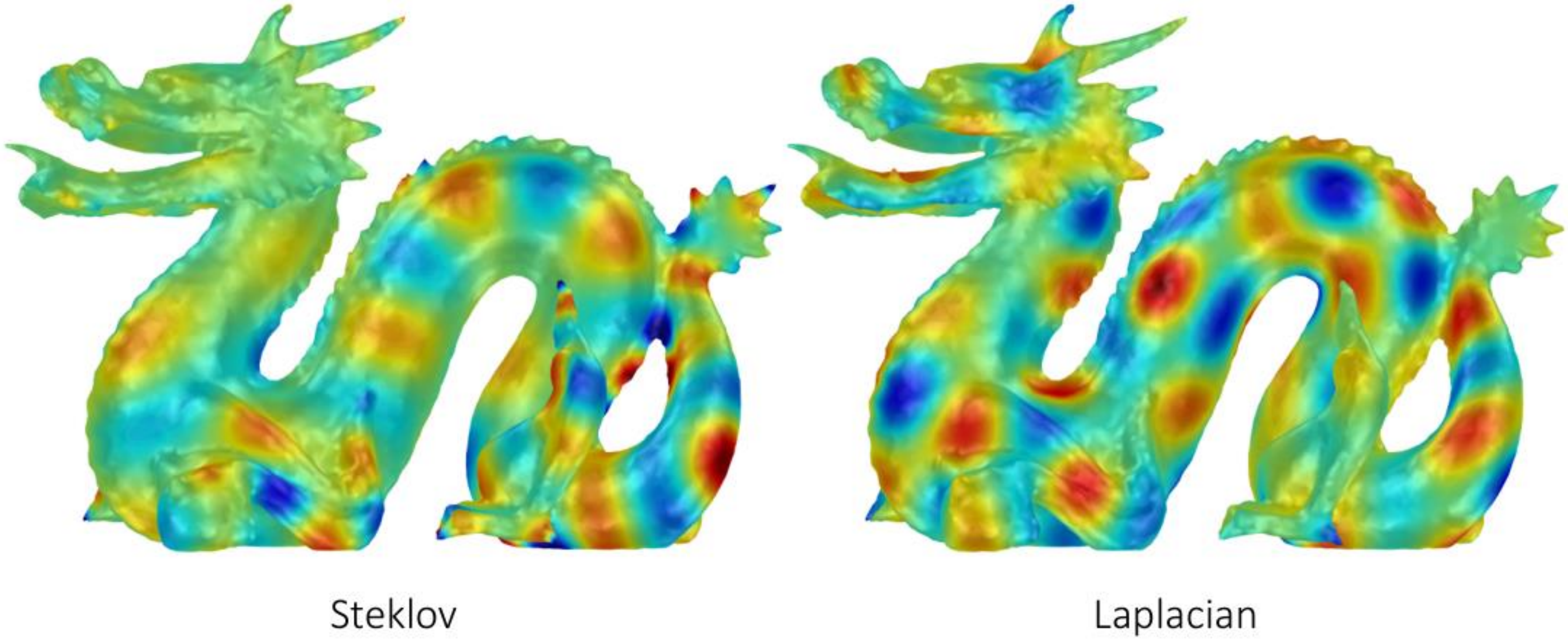


Steklov Laplacian

Comparison of eigenfunctions.

Steklov Spectrum

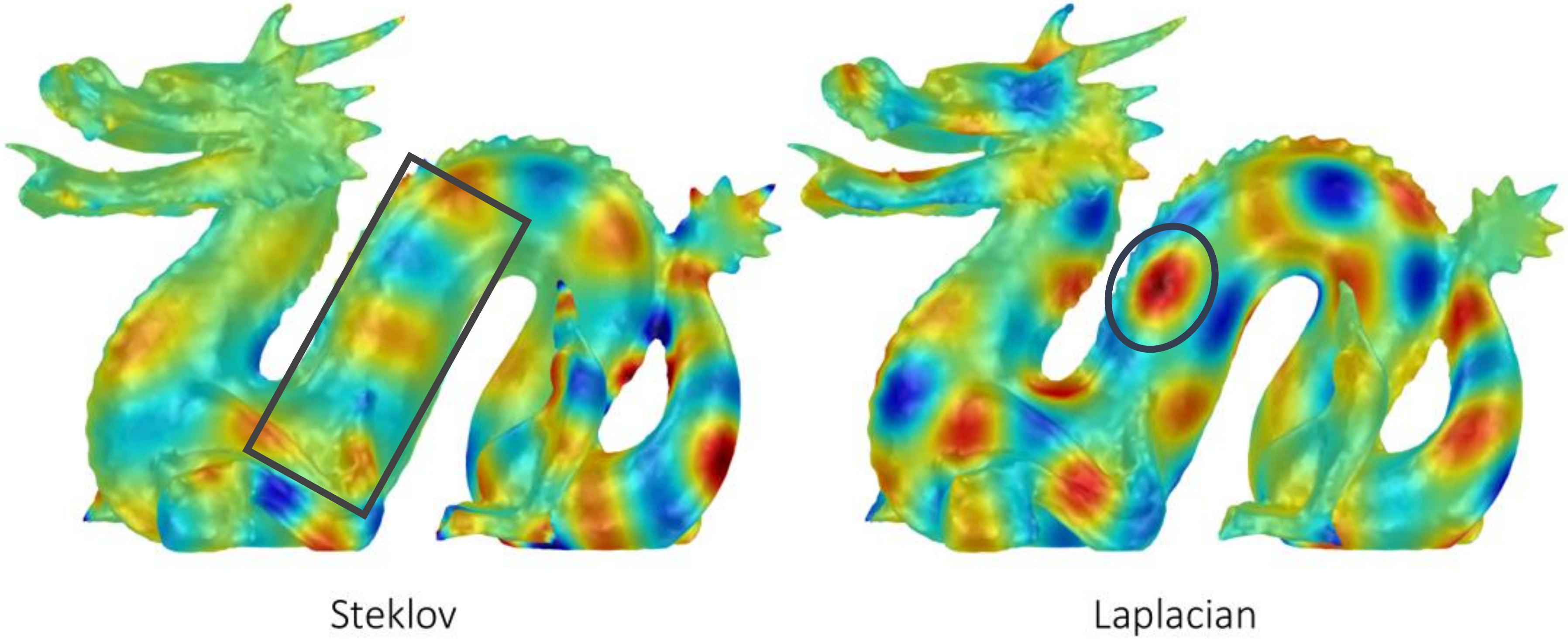
The 200th eigenfunction.



Comparison of eigenfunctions.

Steklov Spectrum

The 200th eigenfunction.



Comparison of eigenfunctions.

Steklov eigenfunction: “cylinder”-like pattern (volume behavior).

Laplacian eigenfunction: “disk”-like pattern (surface behavior).

Spectral Distance

Steklov

~~Laplacian~~ eigenfunctions/eigenvalues

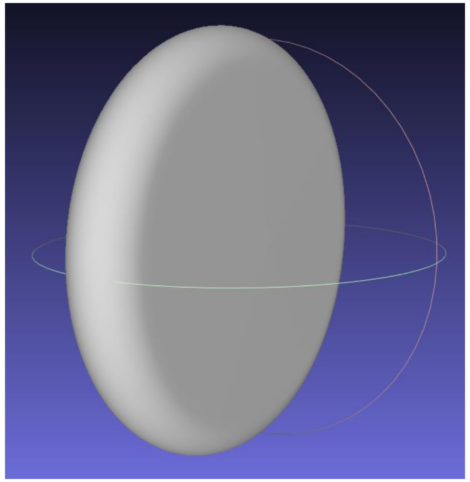
Spectral distance [Lipman et al. 2010]:

$$d_B(x,y)^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i^2} \left(\phi_i(x) - \phi_i(y)\right)^2$$

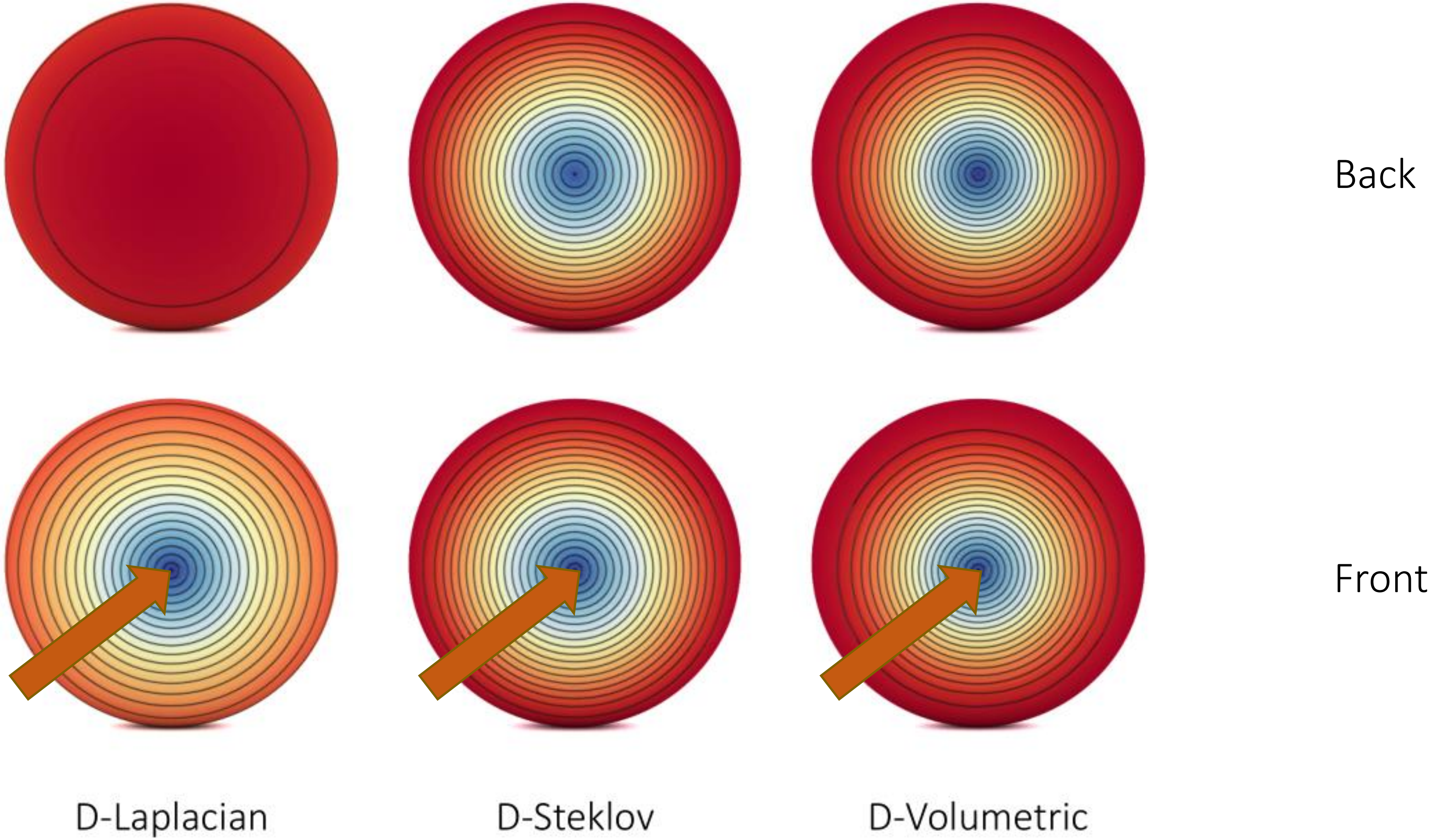
Diffusion distance [Coifman and Lafon 2006]:

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

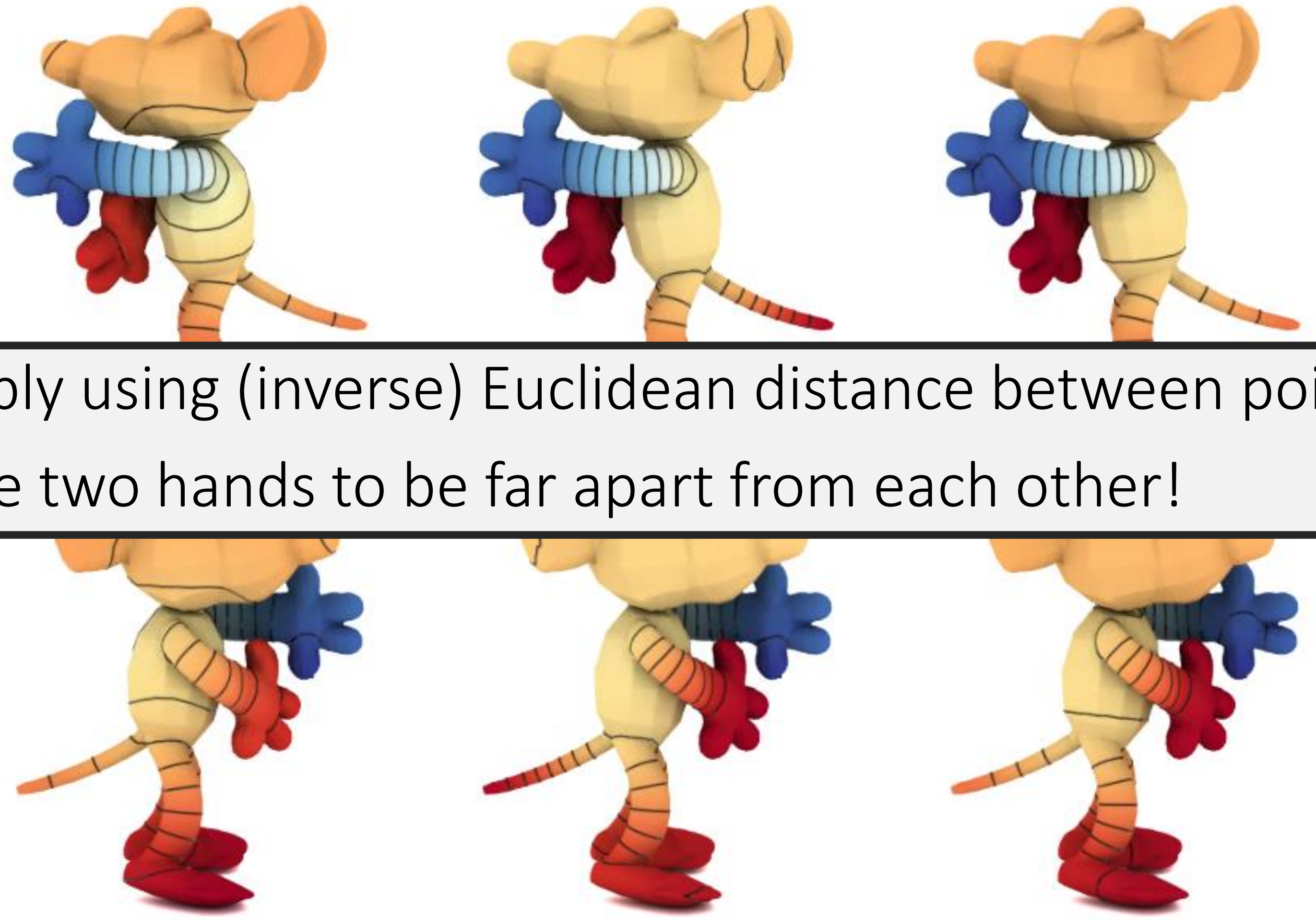
Diffusion Distance



Pancake-like shape



Spectral Distance



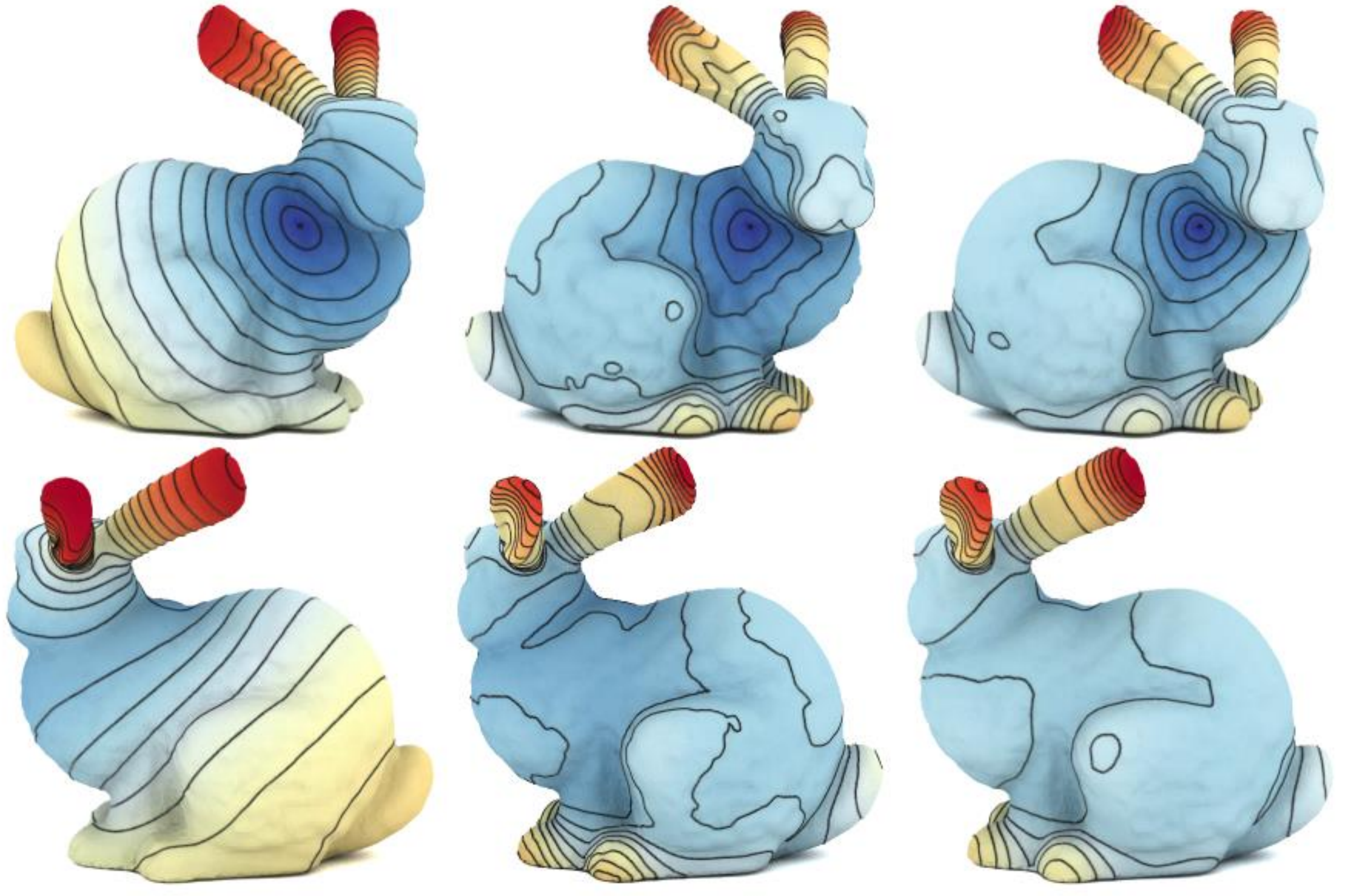
Back

Front

Bi-Laplacian Bi-Steklov Bi-Volumetric

- Why not simply using (inverse) Euclidean distance between points as the metric?
- We would like two hands to be far apart from each other!

Diffusion Distance



Back

Front

D-Laplacian D-Steklov D-Volumetric

Dirichlet-to-Neumann (DtN) Operator $\mathcal{S}$

Consider a volume Ω bounded by the surface $\Gamma = \partial\Omega$

$$\begin{cases} \Delta u(\mathbf{x}) = 0 & \mathbf{x} \in \Omega \\ u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x} \in \partial\Omega \end{cases}$$

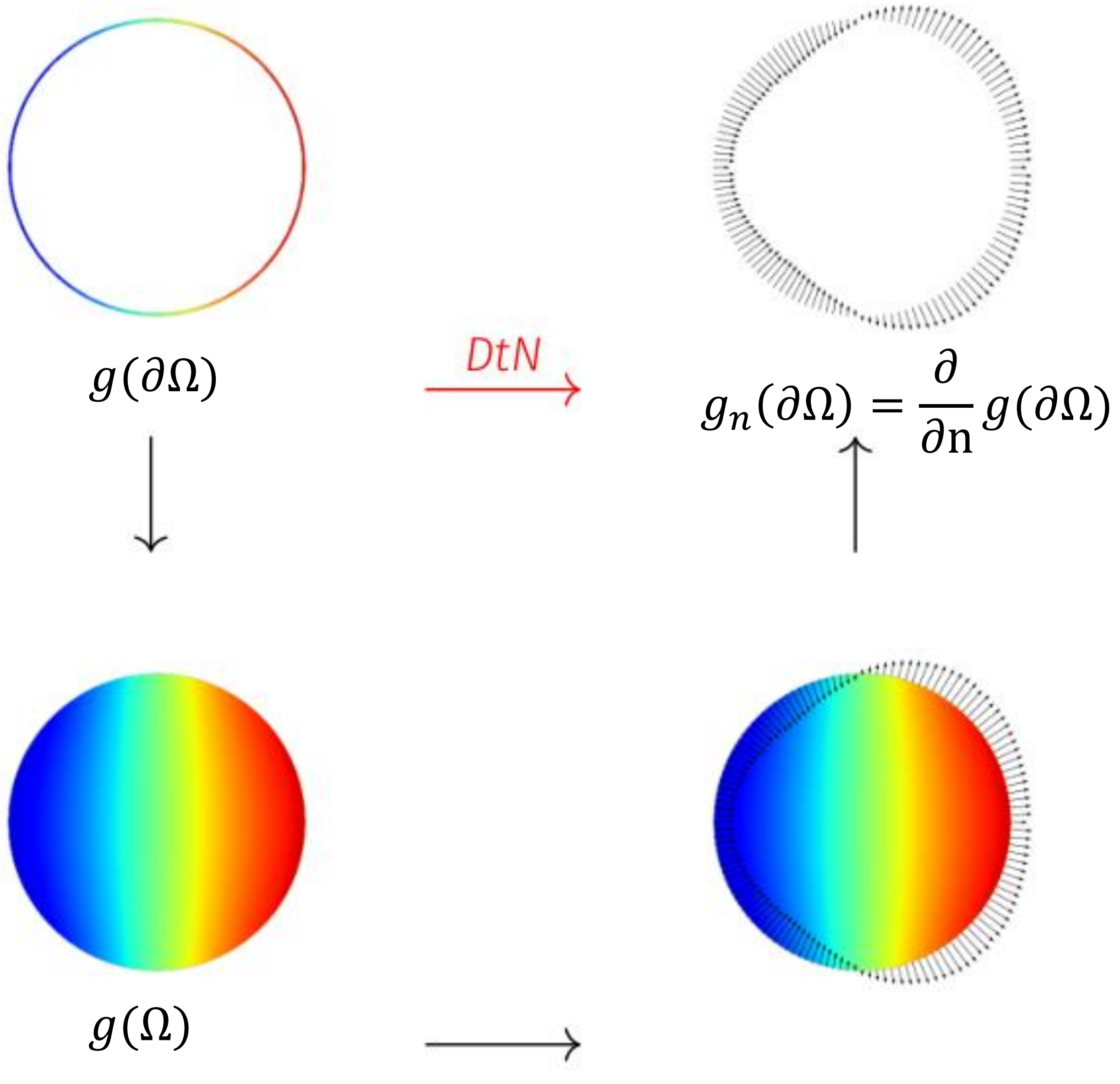
where $g(\Gamma)$ is Dirichlet data

Neumann data $g_n = \frac{\partial}{\partial n} u(\Gamma)$

Dirichlet-to-Neumann (DtN) operator:

$$\mathcal{S} := g \mapsto g_n$$

Also known as the Steklov-Poincaré operator.



DtN as the Composition of Boundary Operators

The DtN operator $\mathcal{S}$ can be written as the composition of operators:

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

$\mathcal{V}, \mathcal{K}, \mathcal{T}, \mathcal{H}$: boundary integral operators. $\mathcal{I}$: identity operator.

- Boundary integral operators that are straightforward to discretize.
- Can be generalized to open surfaces.

Boundary Integral Operators

The single layer potential $\mathcal{V}$ is defined as

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

where $G(\mathbf{x}, \mathbf{y}) := \frac{1}{4\pi}\frac{1}{|\mathbf{x}-\mathbf{y}|}$.

The discretization of $\mathcal{V}$ is $\mathbf{V} \in \mathbb{R}^{n\times n}$ such that $\mathbf{V}_{ij}$ is roughly the (weighted) inverse distance between vertex i and j.

Boundary Integral Operators

The single layer potential $\mathcal{V}$ is defined as

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

where $G(\mathbf{x},\mathbf{y}) := \frac{1}{4\pi}\frac{1}{|\mathbf{x}-\mathbf{y}|}$.

The discretization of $\mathcal{V}$ is $\mathbf{V} \in \mathbb{R}^{n\times n}$ such that $\mathbf{V}_{ij}$ is roughly the (weighted) inverse distance between vertex i and j.

$\mathcal{K}, \mathcal{T}, \mathcal{H}$ have similar definitions to $\mathcal{V}$ but using different kernels rather than $\frac{1}{|\mathbf{x}-\mathbf{y}|}$.

Discrete Boundary Integral Operators

The discretization of $\mathcal{V}$ is $\mathbf{V} \in \mathbb{R}^{n \times n}$ such that $\mathbf{V}_{ij}$ is roughly the inverse distance between vertex i and j. $\mathbf{V}, \mathbf{K}, \mathbf{T}, \mathbf{H} \in \mathbb{R}^{n \times n}$ are similar but using different kernels:

$$v(\mathbf{x}, \mathbf{y}) := \frac{1}{|\mathbf{x} - \mathbf{y}|},$$

$$k(\mathbf{x}, \mathbf{y}) := \frac{(\mathbf{x} - \mathbf{y}) \cdot n(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^3}, \quad t(\mathbf{x}, \mathbf{y}) := \frac{(\mathbf{y} - \mathbf{x}) \cdot n(\mathbf{x})}{|\mathbf{x} - \mathbf{y}|^3} = k(\mathbf{y}, \mathbf{x}),$$

$$h(\mathbf{x}, \mathbf{y}) := -\frac{n(\mathbf{x}) \cdot n(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^3} - \frac{3\,[(\mathbf{x}-\mathbf{y}) \cdot n(\mathbf{y})]\,[(\mathbf{y}-\mathbf{x}) \cdot n(\mathbf{x})]}{|\mathbf{x} - \mathbf{y}|^5},$$

$|\mathbf{x} - \mathbf{y}|$: the distance between points $\mathbf{x}, \mathbf{y}$.

$n(\mathbf{x}), n(\mathbf{y})$: the normal directions at points $\mathbf{x}, \mathbf{y}$ on the surface, resp.

More Boundary Integral Operators

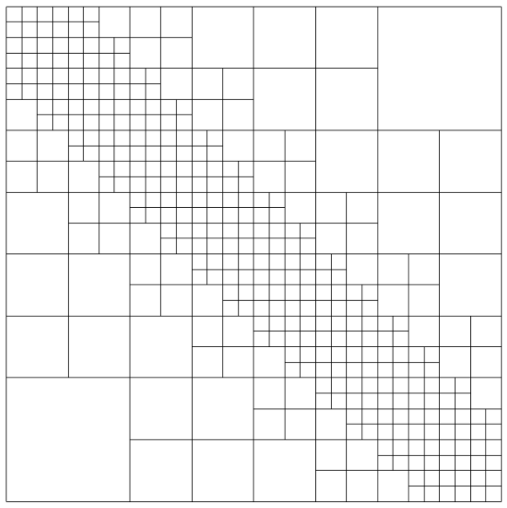
$$[\mathcal{V}\phi](\mathbf{x}) := \int_\Gamma G(\mathbf{x},\mathbf{y})\phi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{y}), \quad [\mathcal{K}\phi](\mathbf{x}) := \int_\Gamma \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})}\phi(\mathbf{y})\,\mathrm{d}\Gamma(\mathbf{y}),$$

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

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

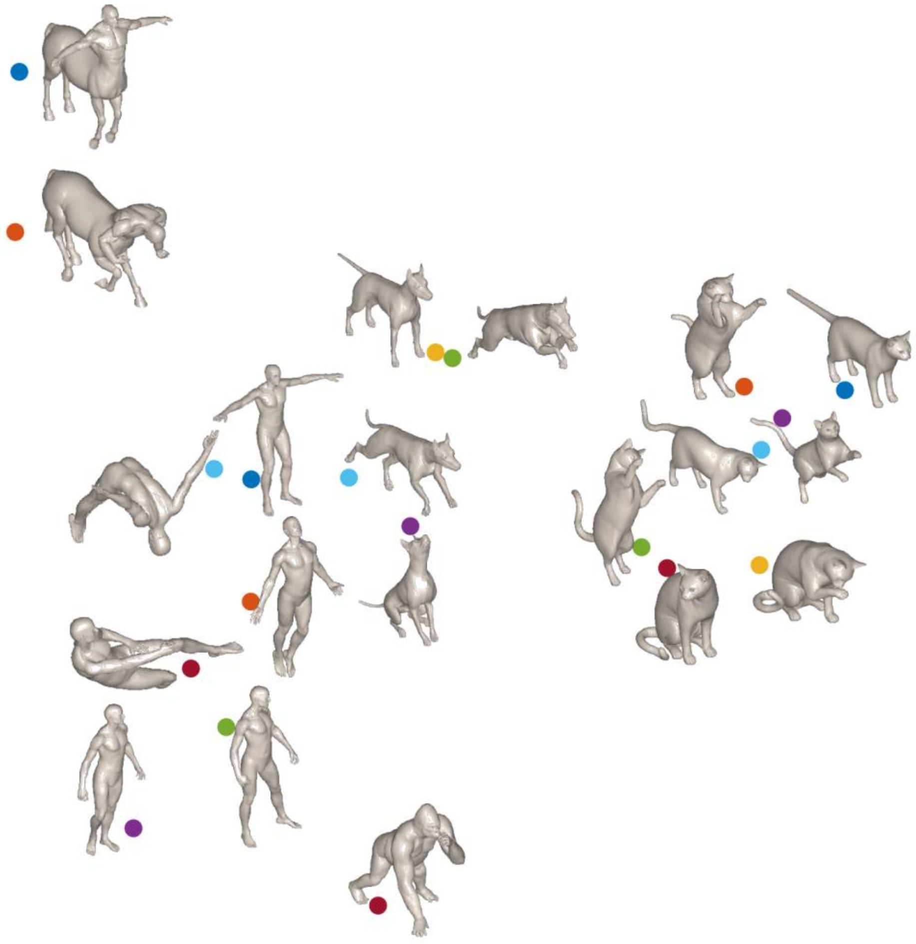
Boundary Element Method (BEM) with Fast Computations

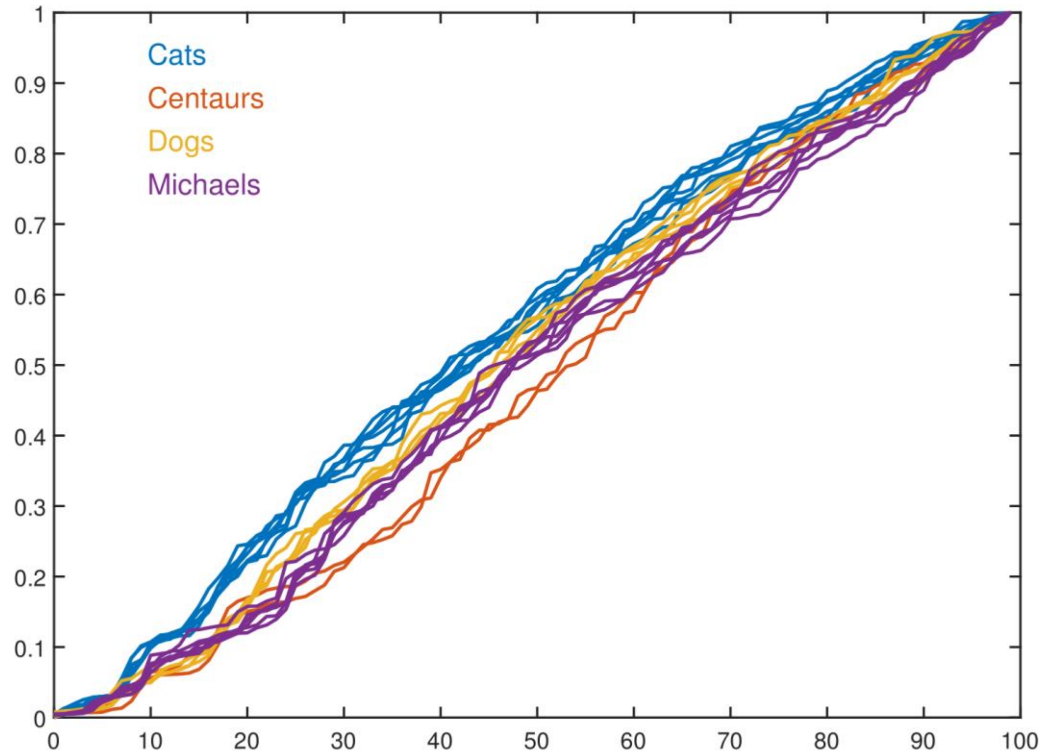
- Hierarchical matrix (H-matrix): reduces complexity to $\mathcal{O}(n \log 1/\epsilon)$.
- Iterative eigensolver for top k eigenfunctions.
- Provably optimal preconditioners are available.



H-matrix

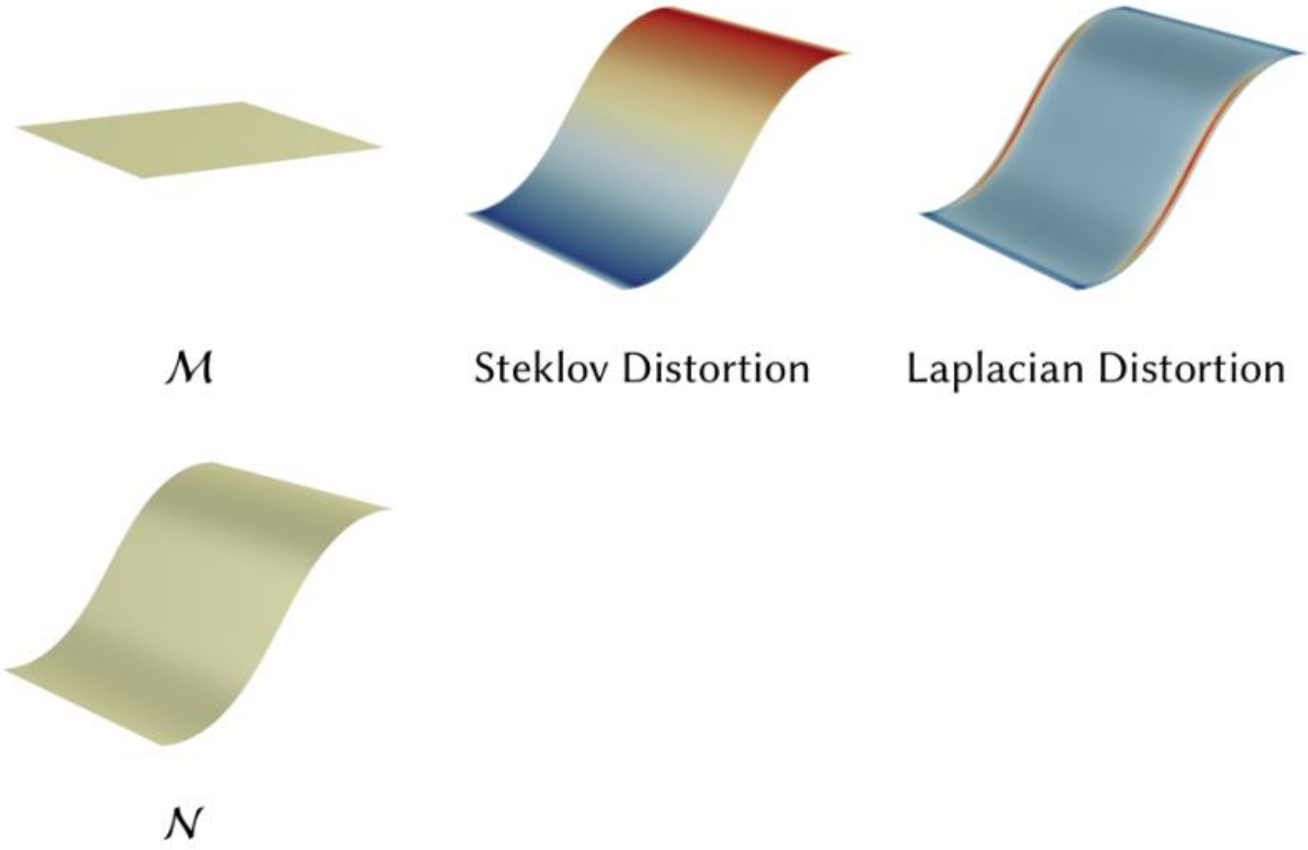
Steklov Eigenvalues as the “ShapeDNA” (i.e. Shape2Vec)





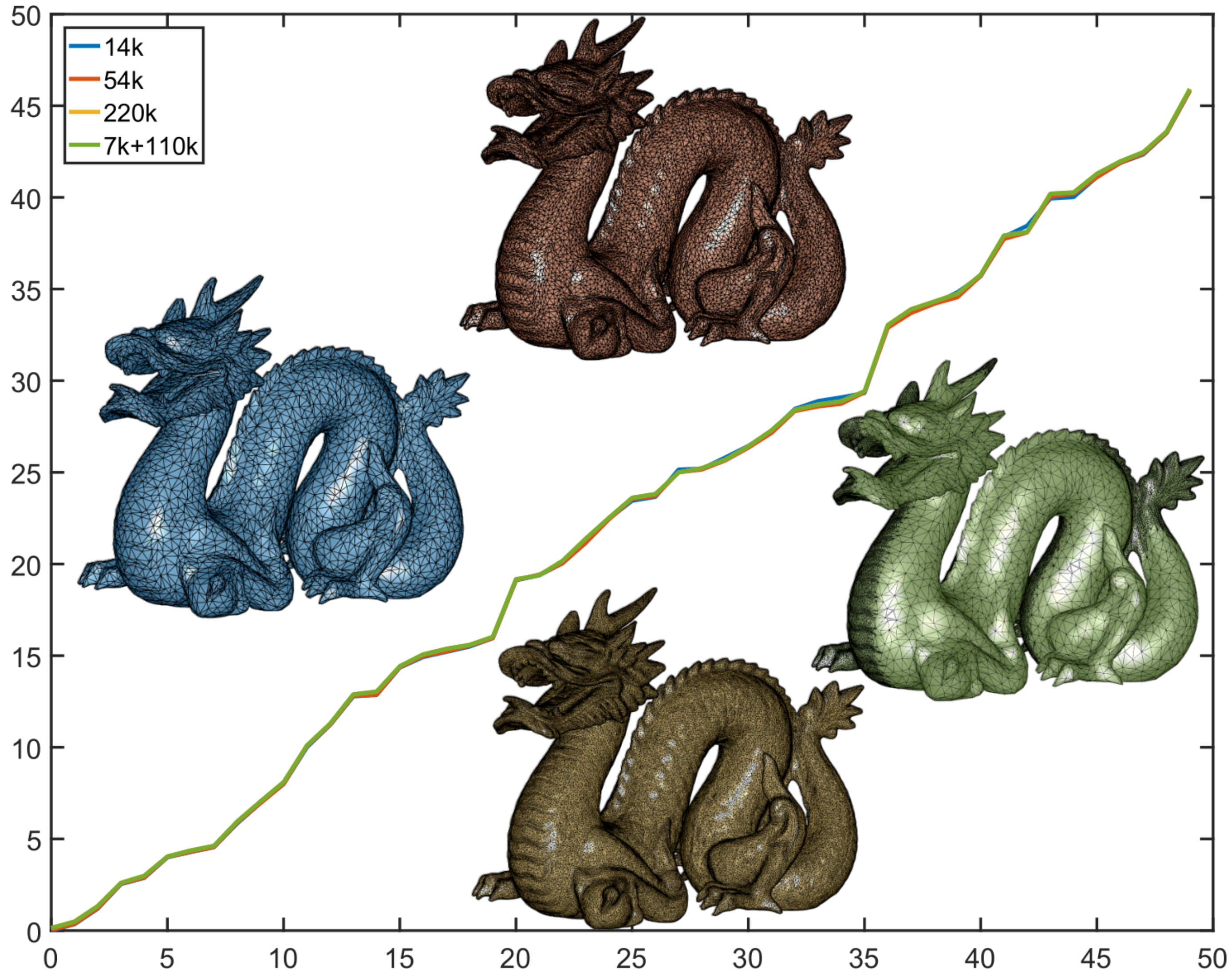
Shape Difference

Generalizing operator approach for shape difference [Rustamov et al. 2013]

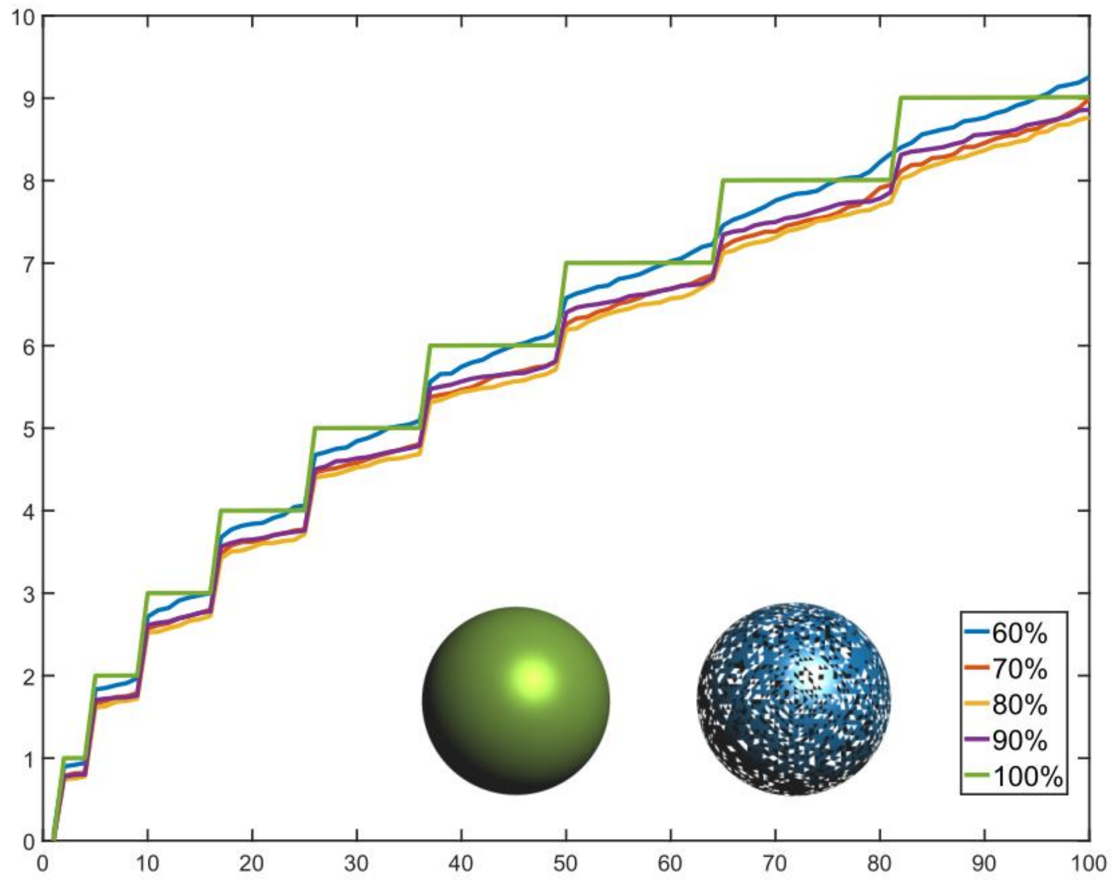


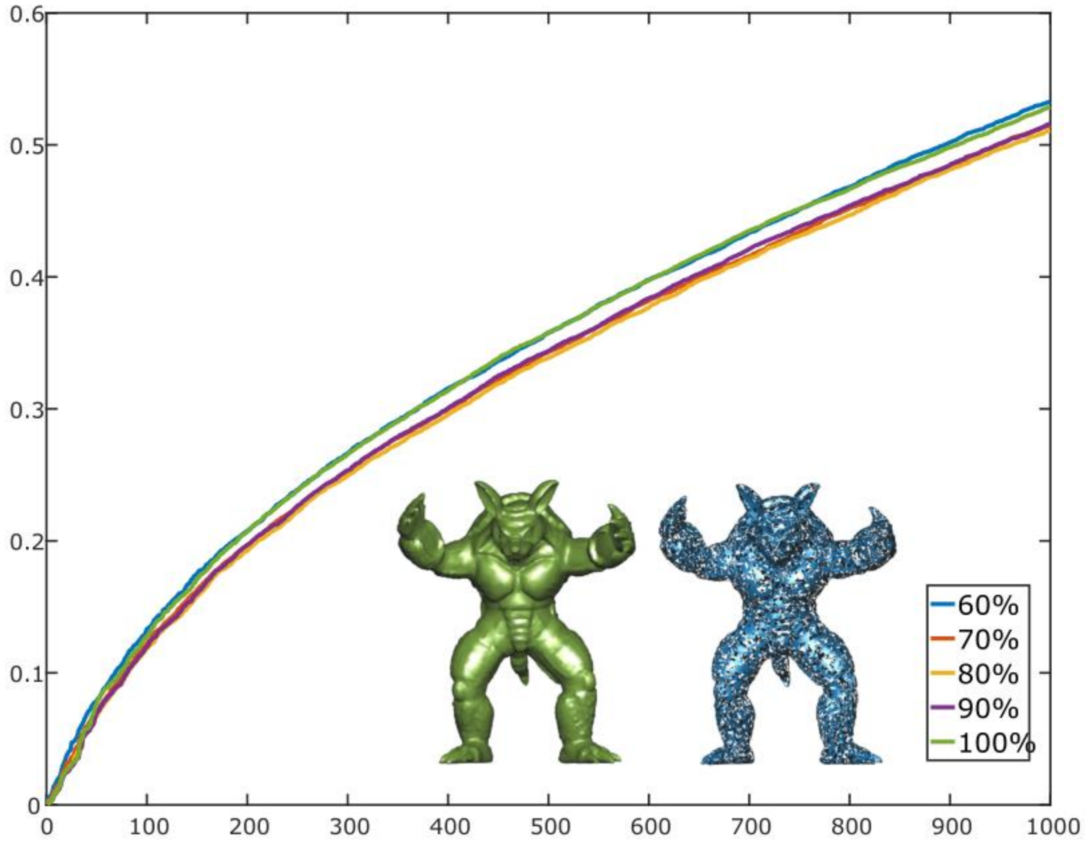
Convergence and Robustness to Irregular Meshing

- Low 14k
- Medium 54k
- High 220k
- Unbalanced 7k+110k

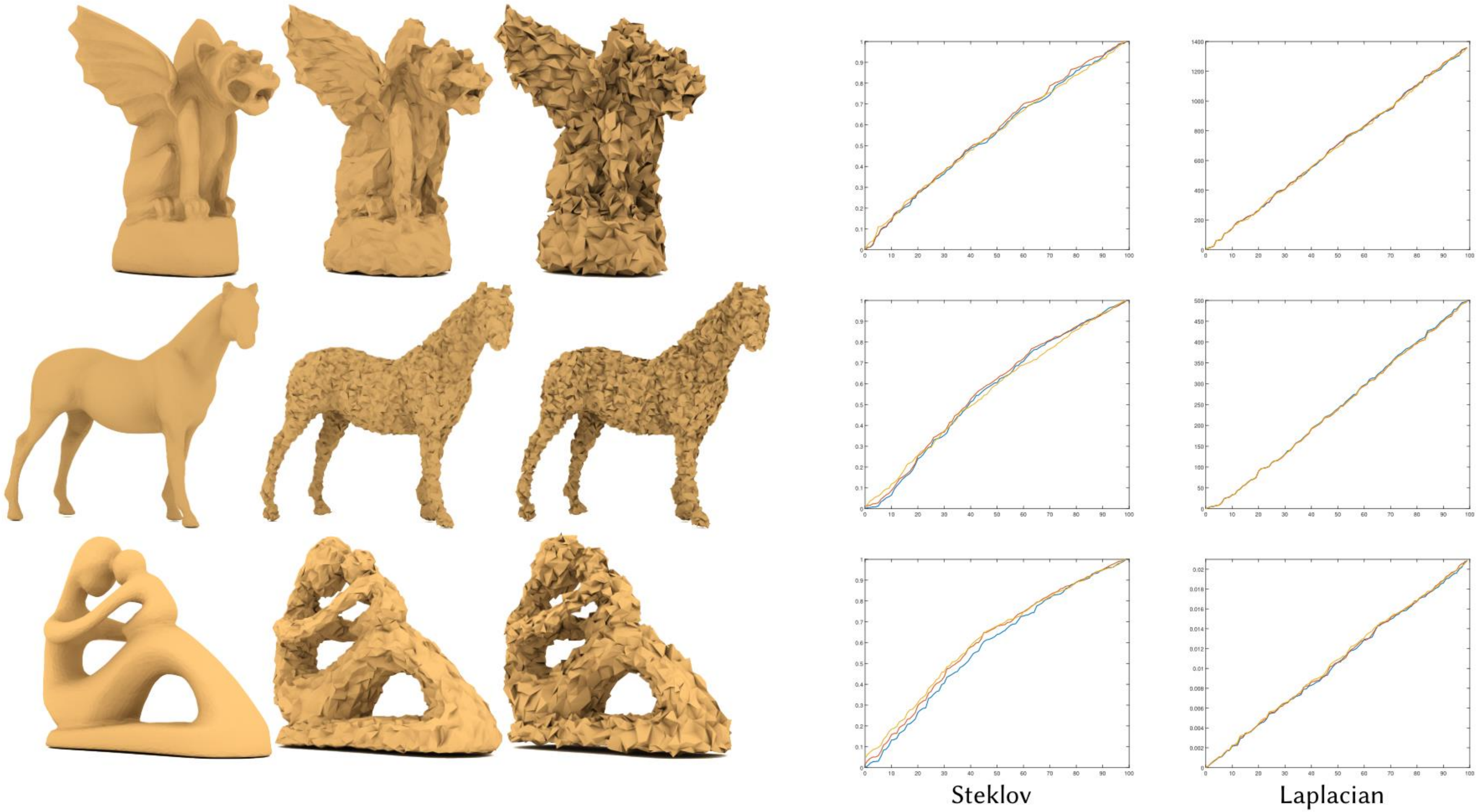


Robustness

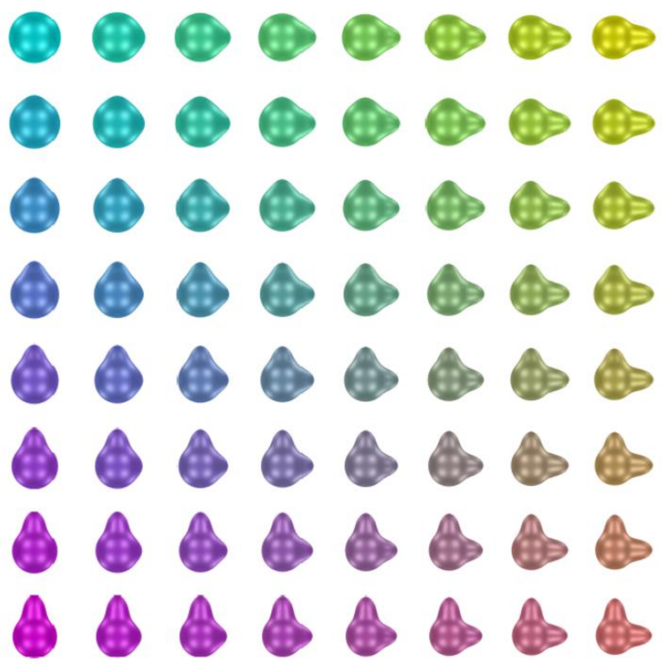

10
9
8
7
6
5
4
3
2
1
0
0
10
20
30
40
50
60
70
80
90
100
60%
70%
80%
90%
100%


0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
400
500
600
700
800
900
1000
60%
70%
80%
90%
100%

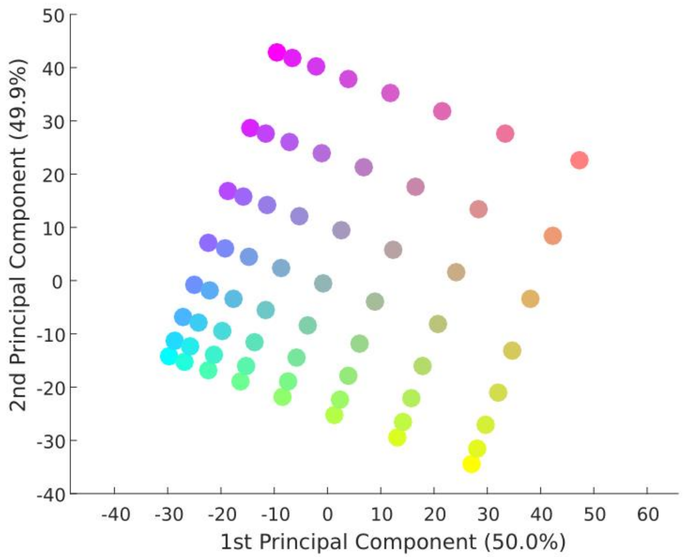
Robustness



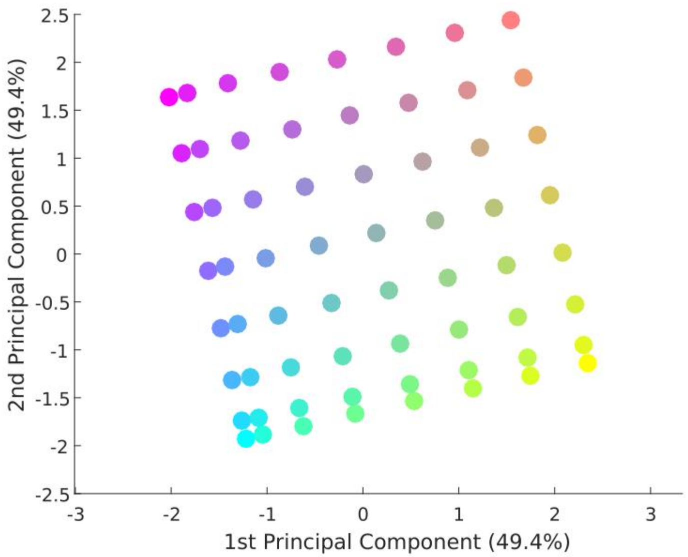
Exploring Shape Variability



Shapes



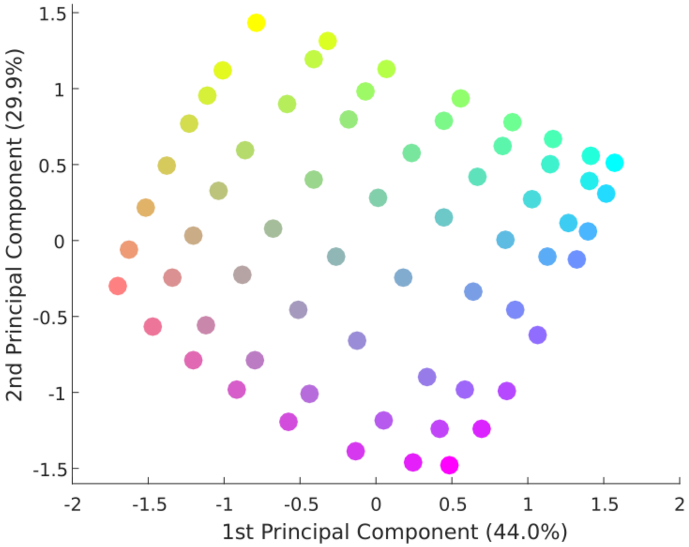
Steklov



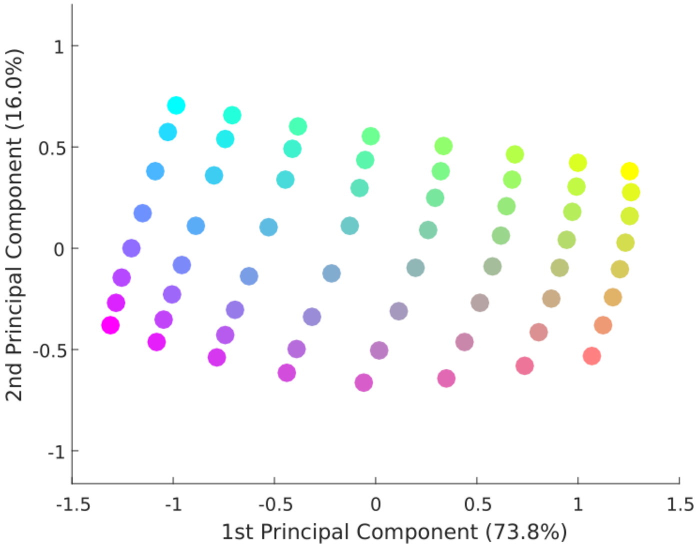
Laplacian

Exploring Shape Variability

Shapes

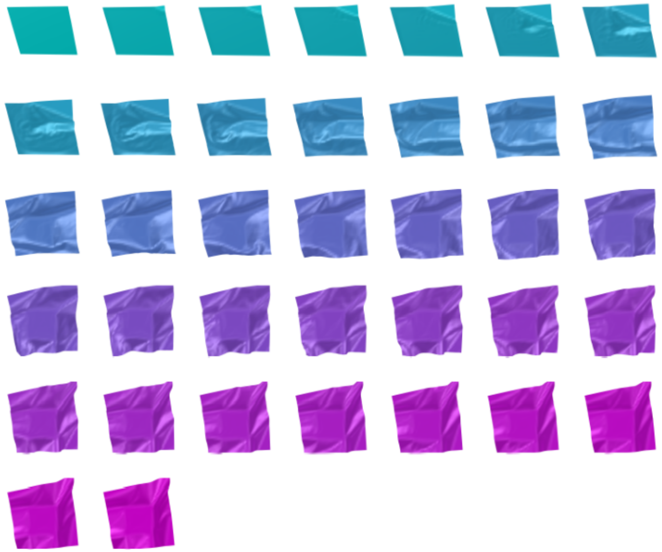


Steklov

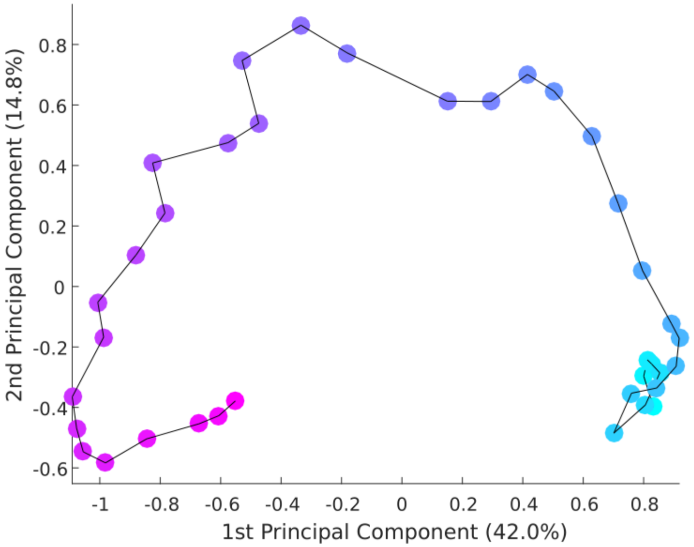


Laplacian

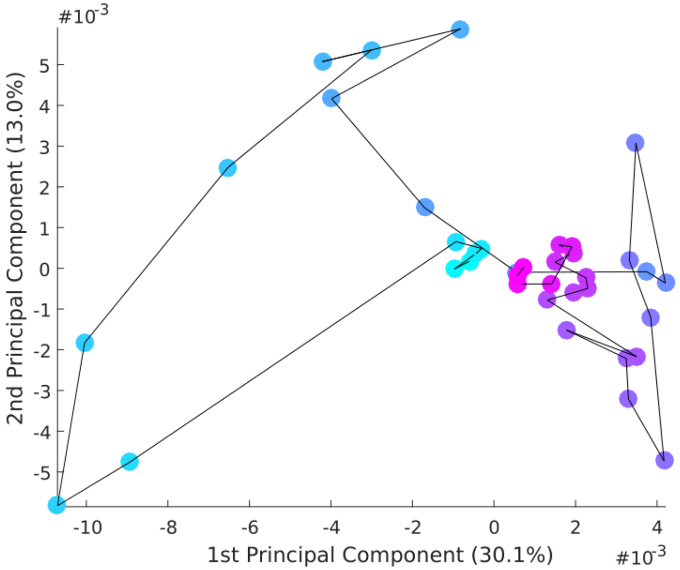
Exploring Shape Variability



Shapes



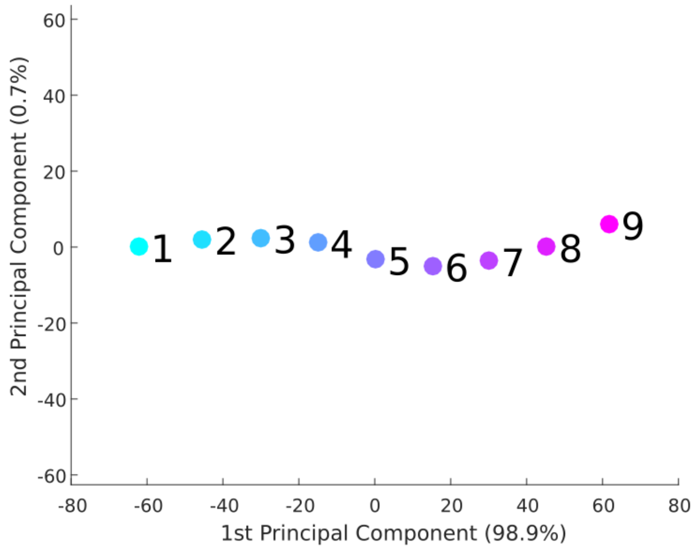
Steklov



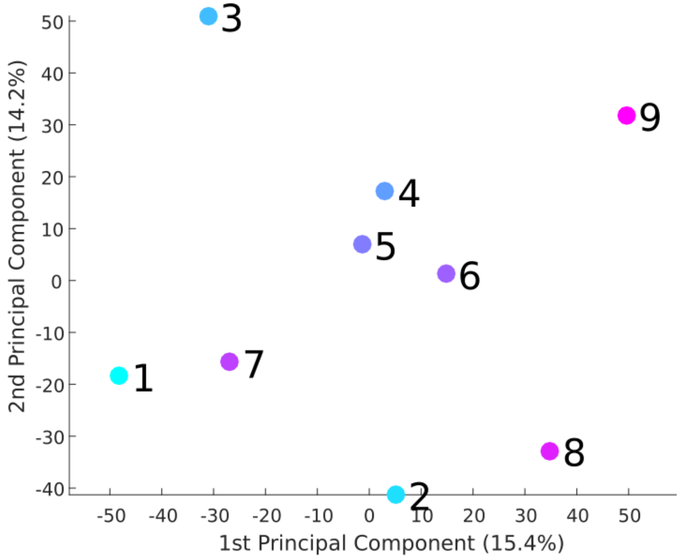
Laplacian

Exploring Shape Variability

Shapes

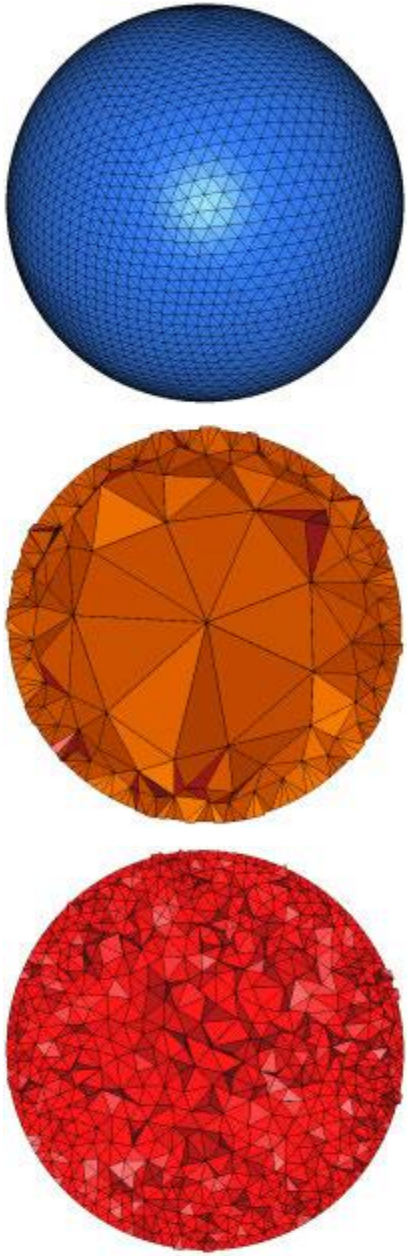


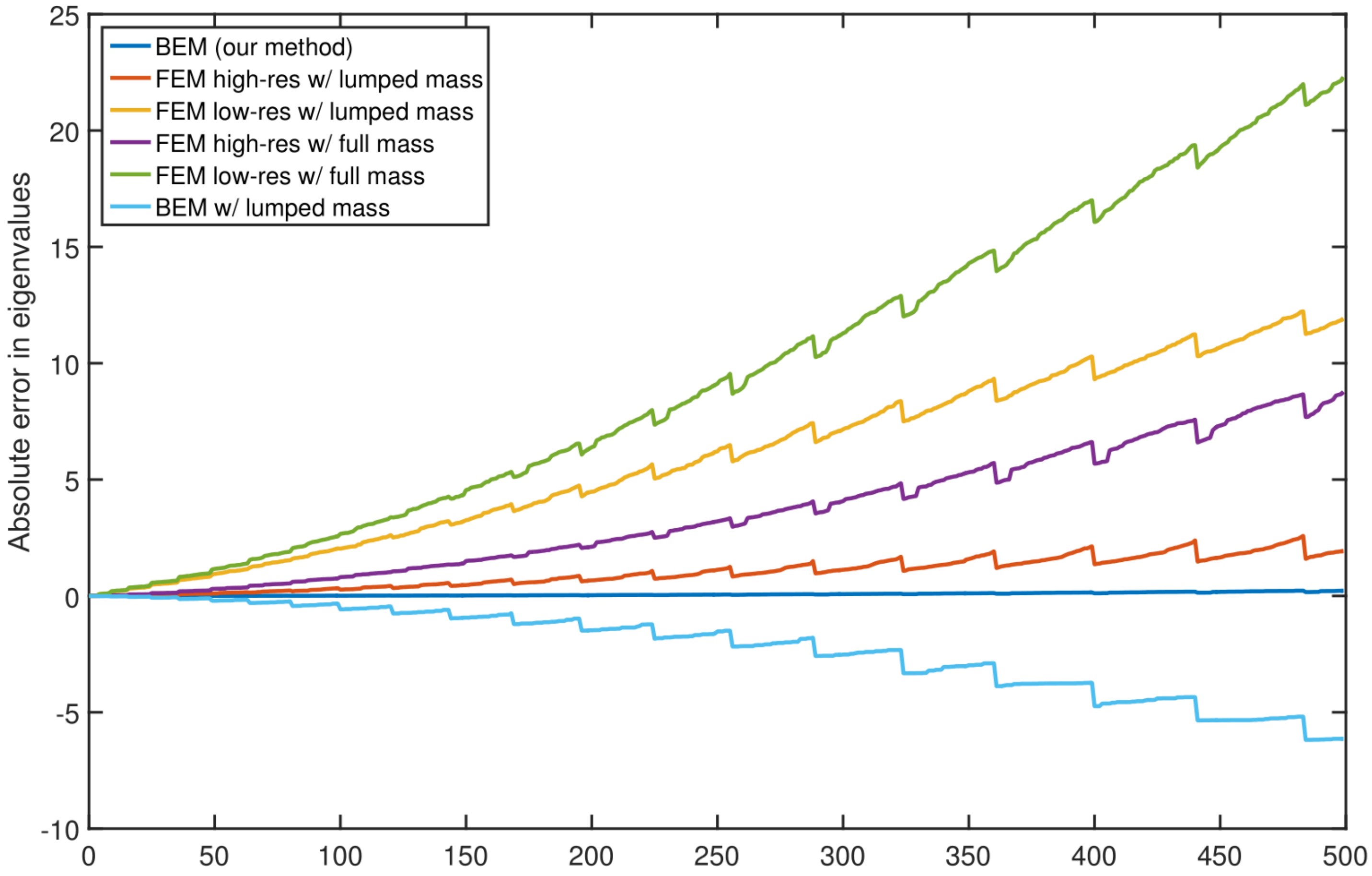
Steklov



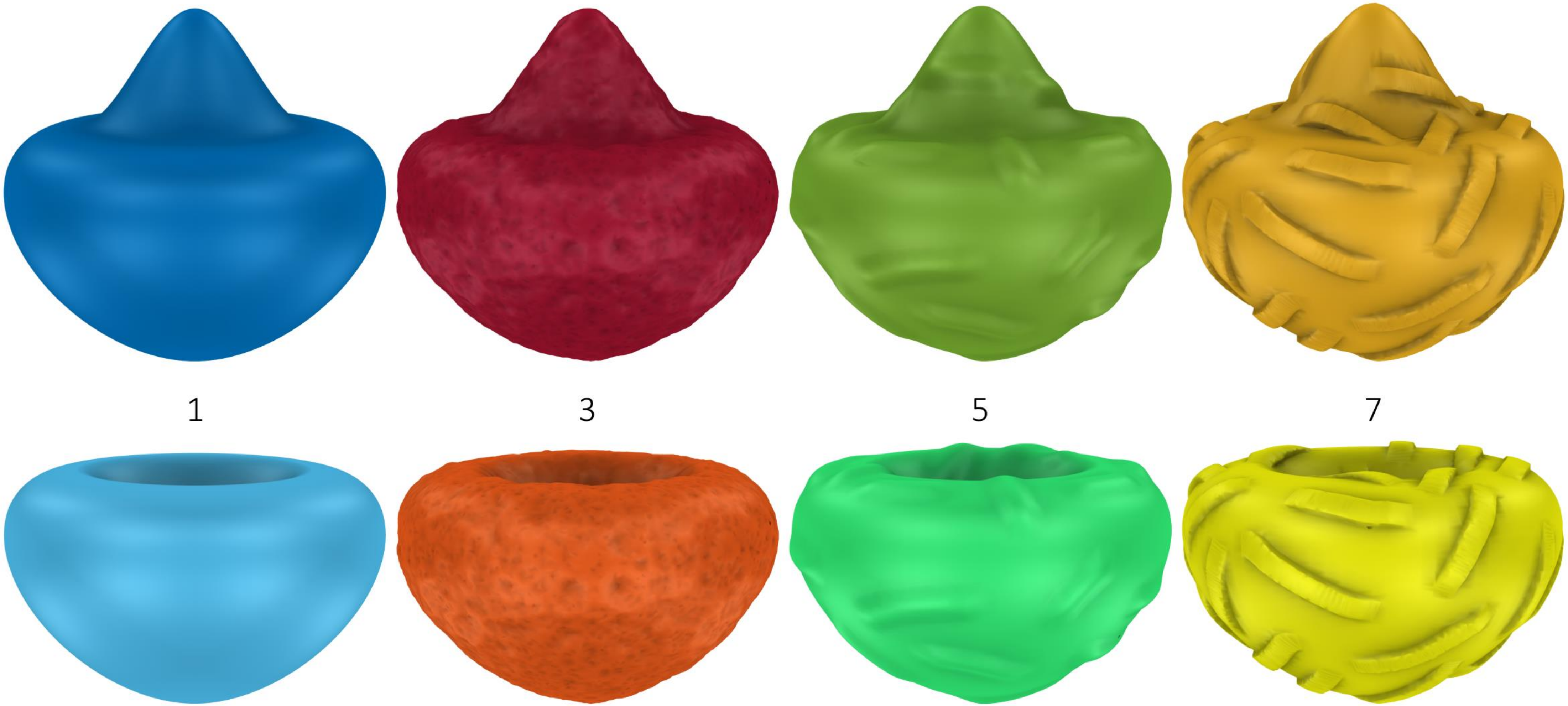
Laplacian

Our BEM Formulation v.s. A Possible FEM Operator

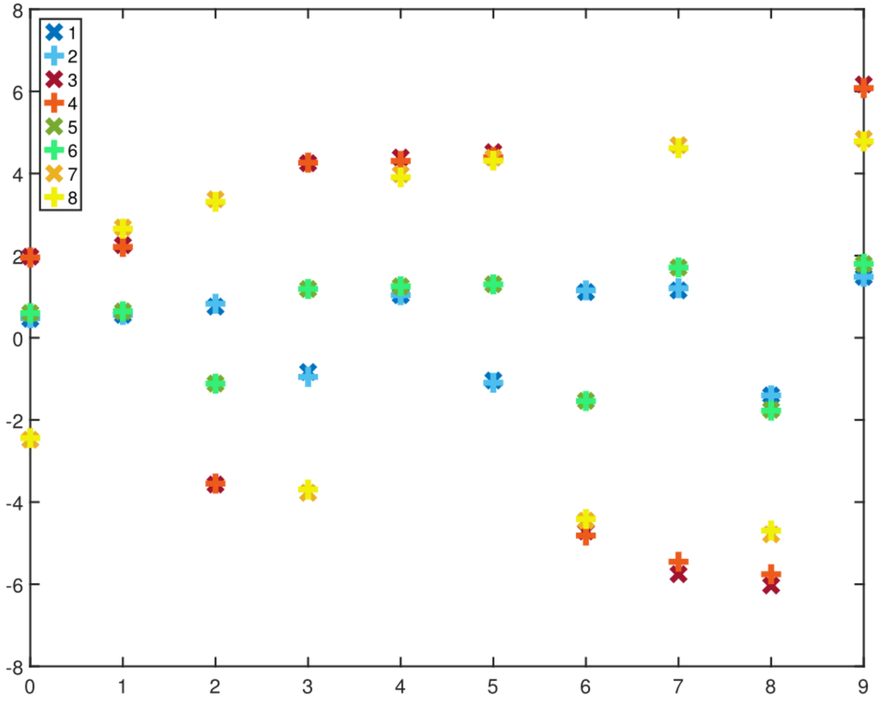




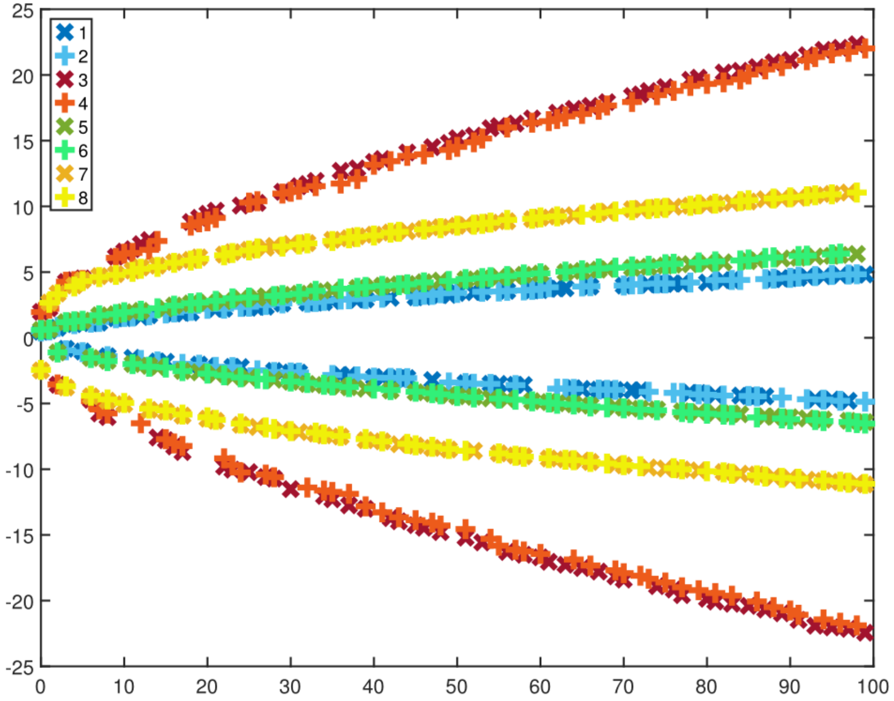
Comparison with Dirac Operator [Liu et al. 2017]



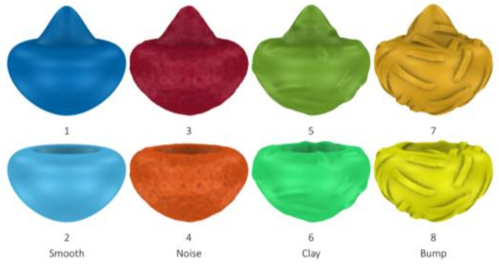
Comparison with Dirac Operator [Liu et al. 2017]



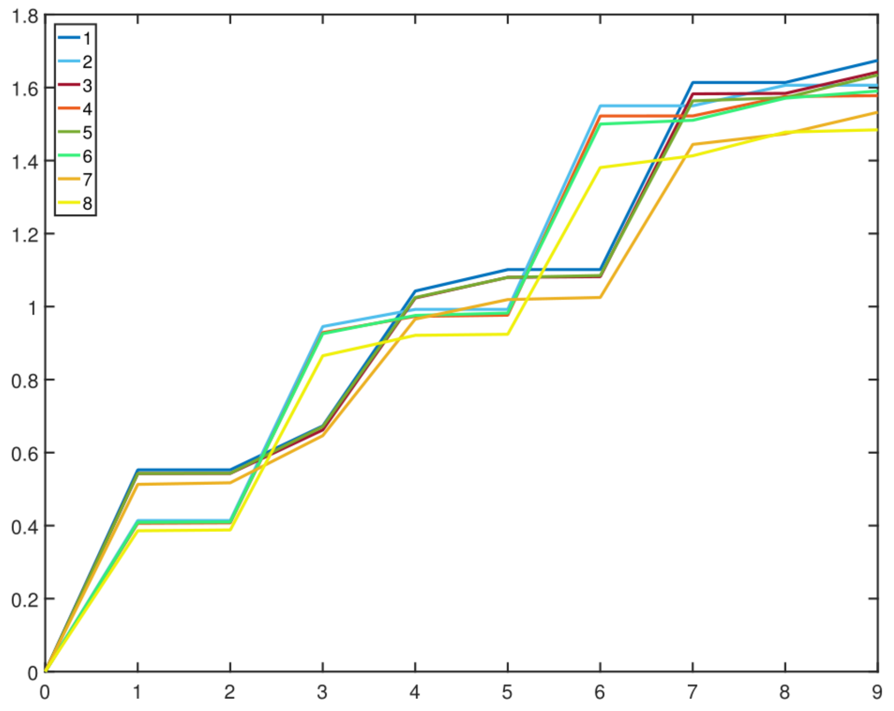
Top 10 Dirac Eigenvalues.



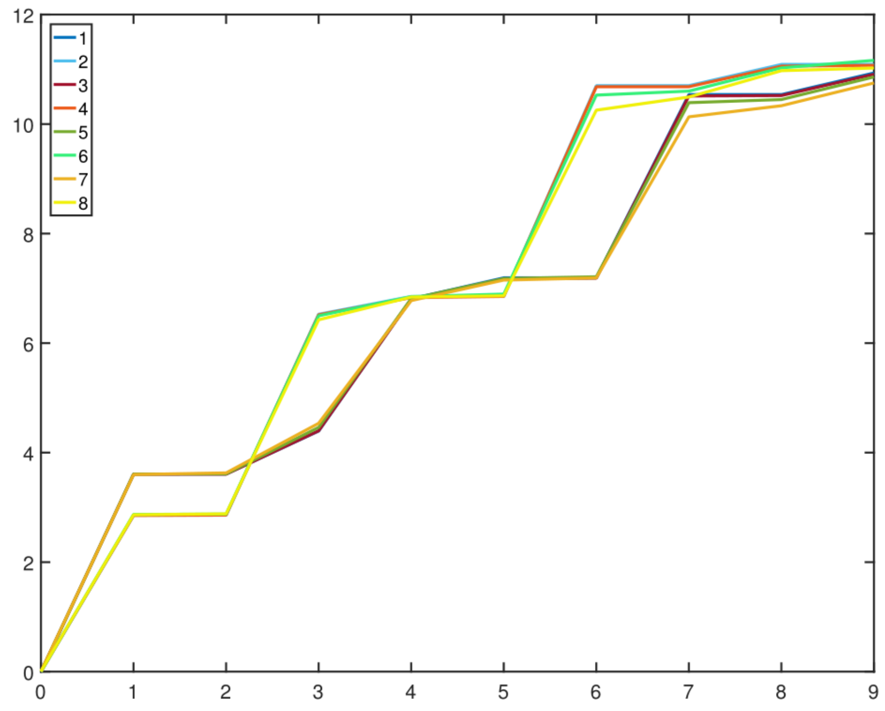
Top 100 Dirac Eigenvalues.



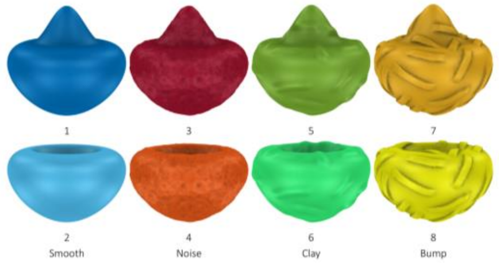
Comparison with Dirac Operator [Liu et al. 2017]



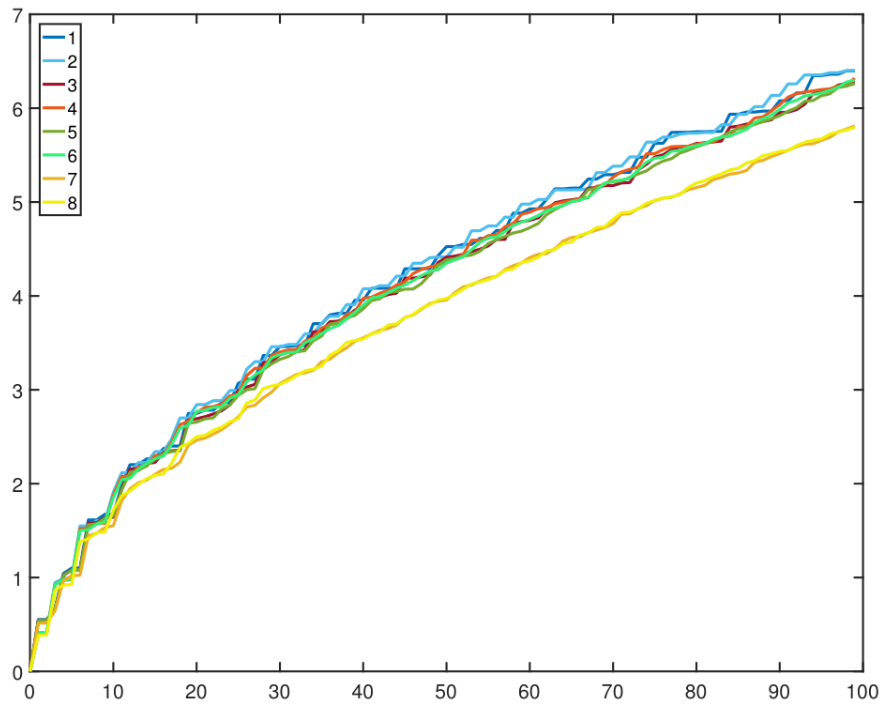
Steklov Eigenvalues.



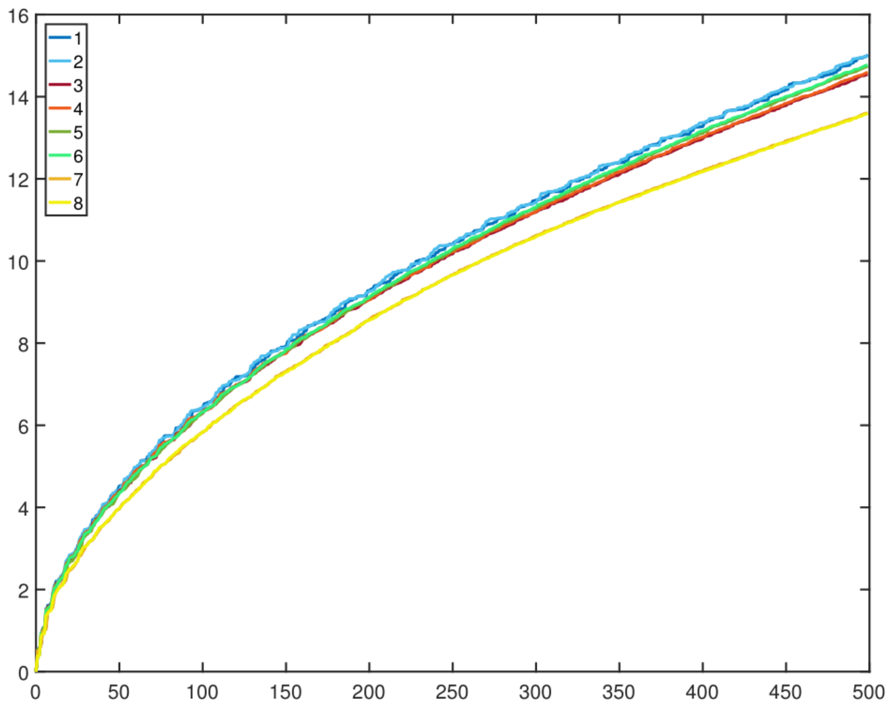
Scaled Steklov Eigenvalues.



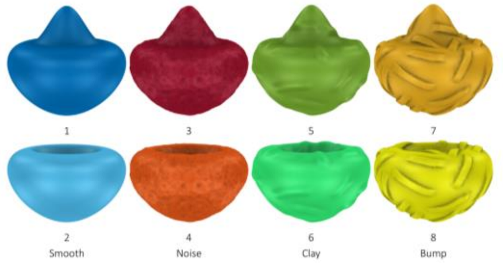
Comparison with Dirac Operator [Liu et al. 2017]



Top 100 Steklov Eigenvalues.



Top 500 Steklov Eigenvalues.



Conclusion

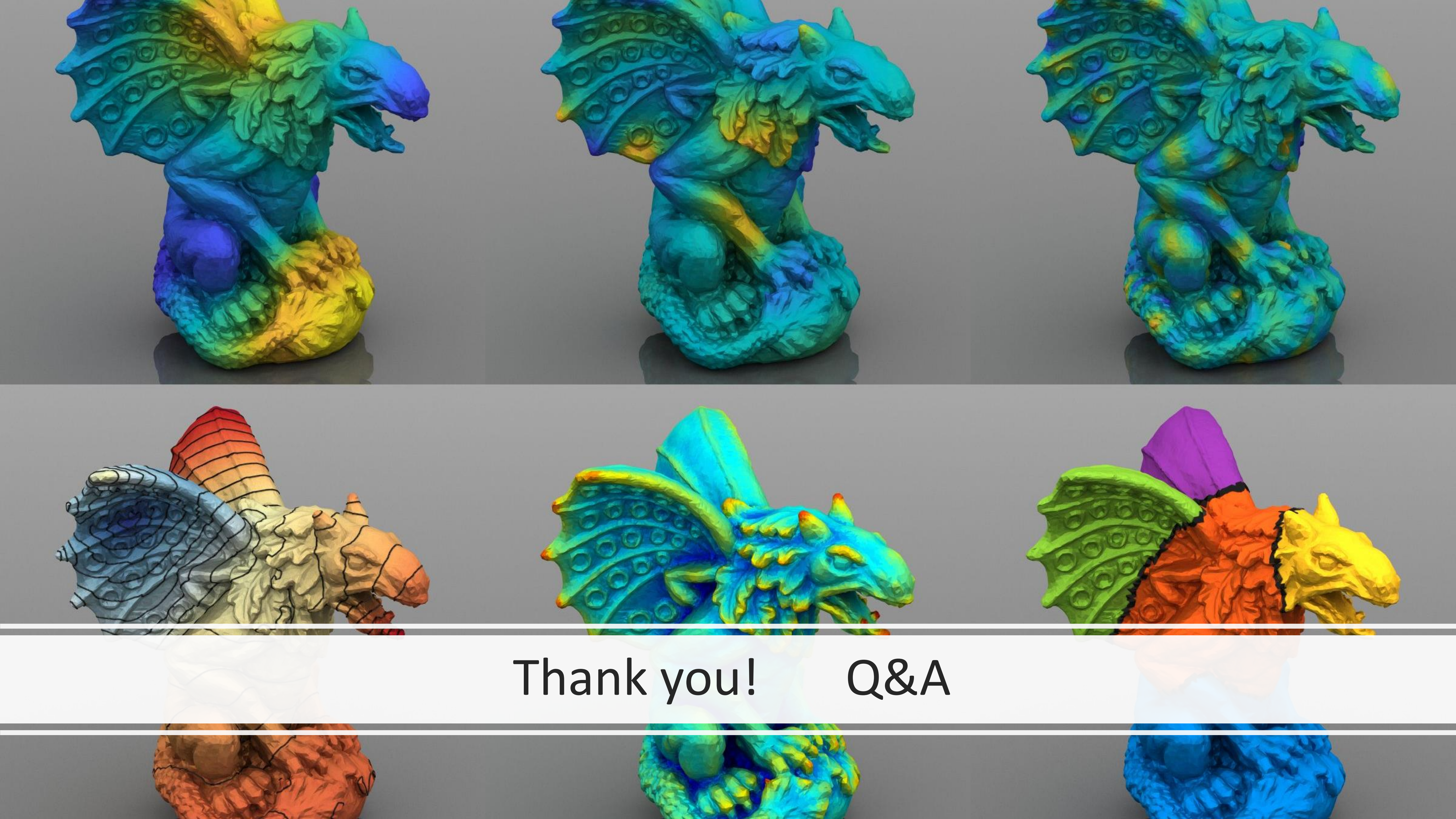
- Surface-only approach using the boundary element method.
- An operator approach for extrinsic geometry for many applications.

Code available

Future Work

- Integral geometry operators.
- Justify the use of DtN operator in various applications:
  - Shape deformation, physical simulation, skinning animation, interpolation weights, volumetric parameterization, meshing, vector and frame field design, statistical learning on manifolds, and geometric deep learning.
- A mathematical theory for open surfaces and point clouds.
- Open question: “Can you hear the shape of a drum (from Steklov eigenvalues)?”

Code available



Thank you! Q&A

Backup Slides

Single Layer Potential

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

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

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

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

Physically, $\mathcal{K}$ maps an input electric dipole density distribution ϕ 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 $\mathcal{V}$:

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

where the integral is understood in the sense of Cauchy principal value. Physically, $\mathcal{T}$ maps an input electric charge density distribution ϕ 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 $\mathcal{K}$:

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

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

DtN as the Composition of Boundary Operators

The DtN operator $\mathcal{S}$ can be written as the composition of operators:

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

where $\mathcal{V}, \mathcal{K}, \mathcal{T}, \mathcal{H}$ are boundary integral operators.

- Boundary integral operators that are straightforward to discretize.
- Can be symbolically defined for open surfaces.

Reformulation

The eigenvalue problem $\mathbf{S}\mathbf{u} = \lambda \mathbf{M}\mathbf{u}$

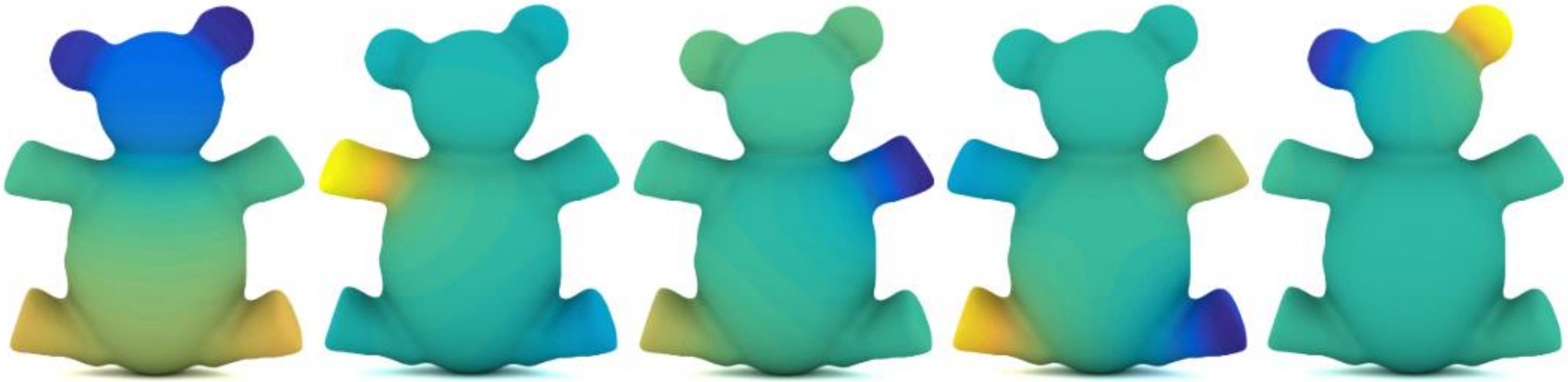
$$\mathbf{S} := \mathbf{H} + (0.5\mathbf{M} + \mathbf{T})\mathbf{V}^{-1}(0.5\mathbf{M} + \mathbf{K}). \tag{5}$$

Reformulated as

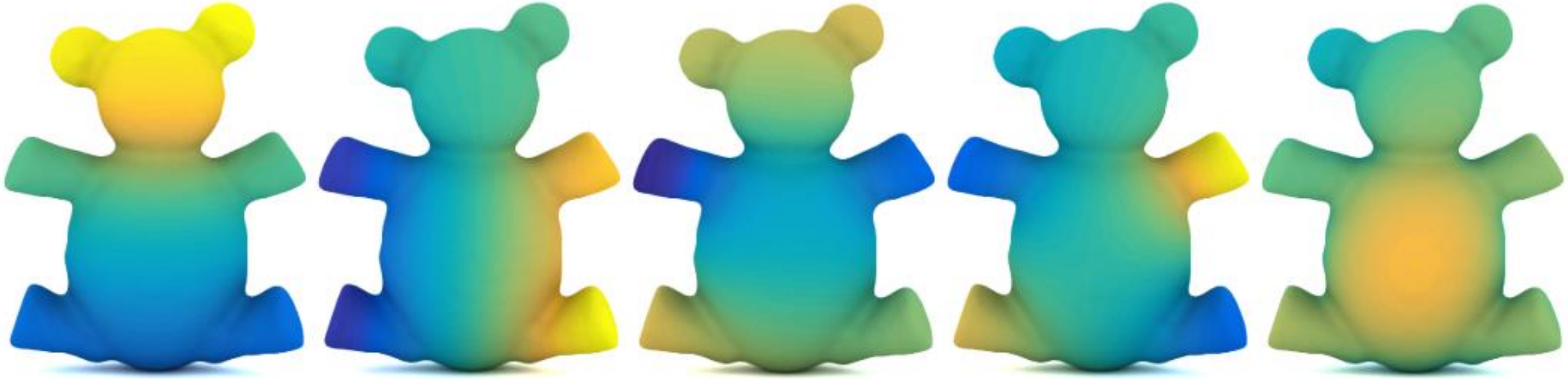
$$\begin{bmatrix} \mathbf{V} & -\mathbf{Q} \\ \mathbf{Q}^\top & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mathbf{u} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{0} & \\ & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mathbf{u} \end{bmatrix}$$

where $\mathbf{Q} := 0.5\mathbf{M} + \mathbf{K}$

Shape Segmentation

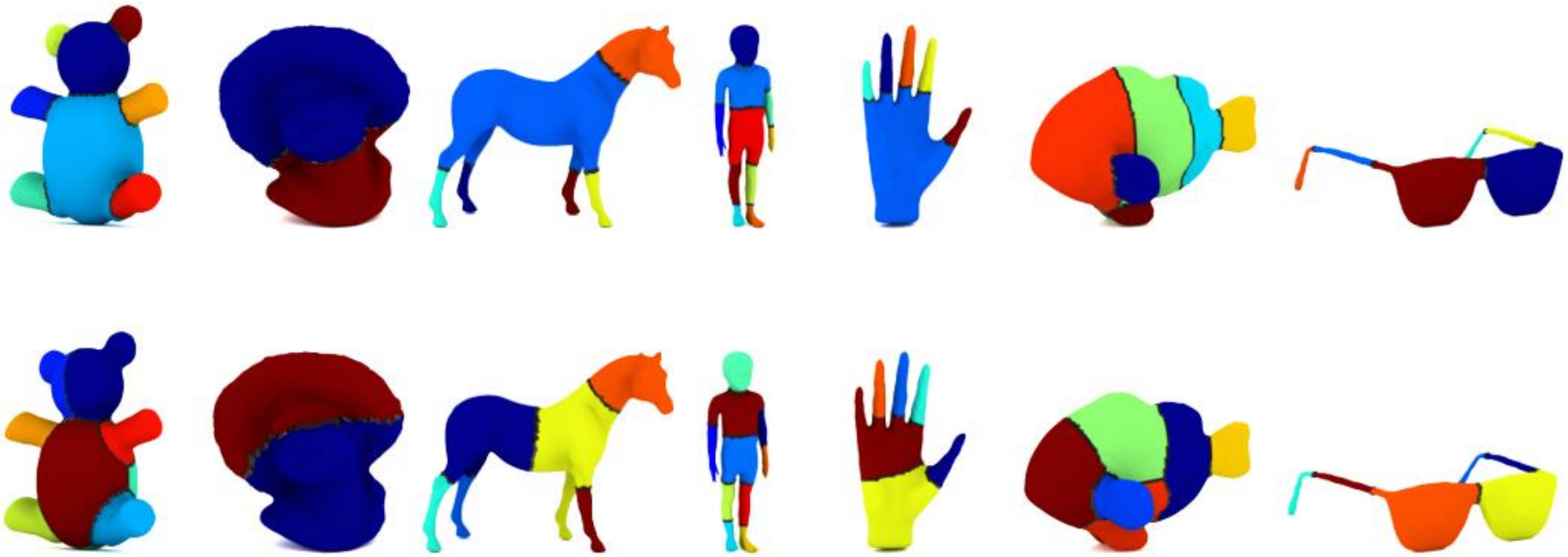


Steklov



Laplacian

Shape Segmentation

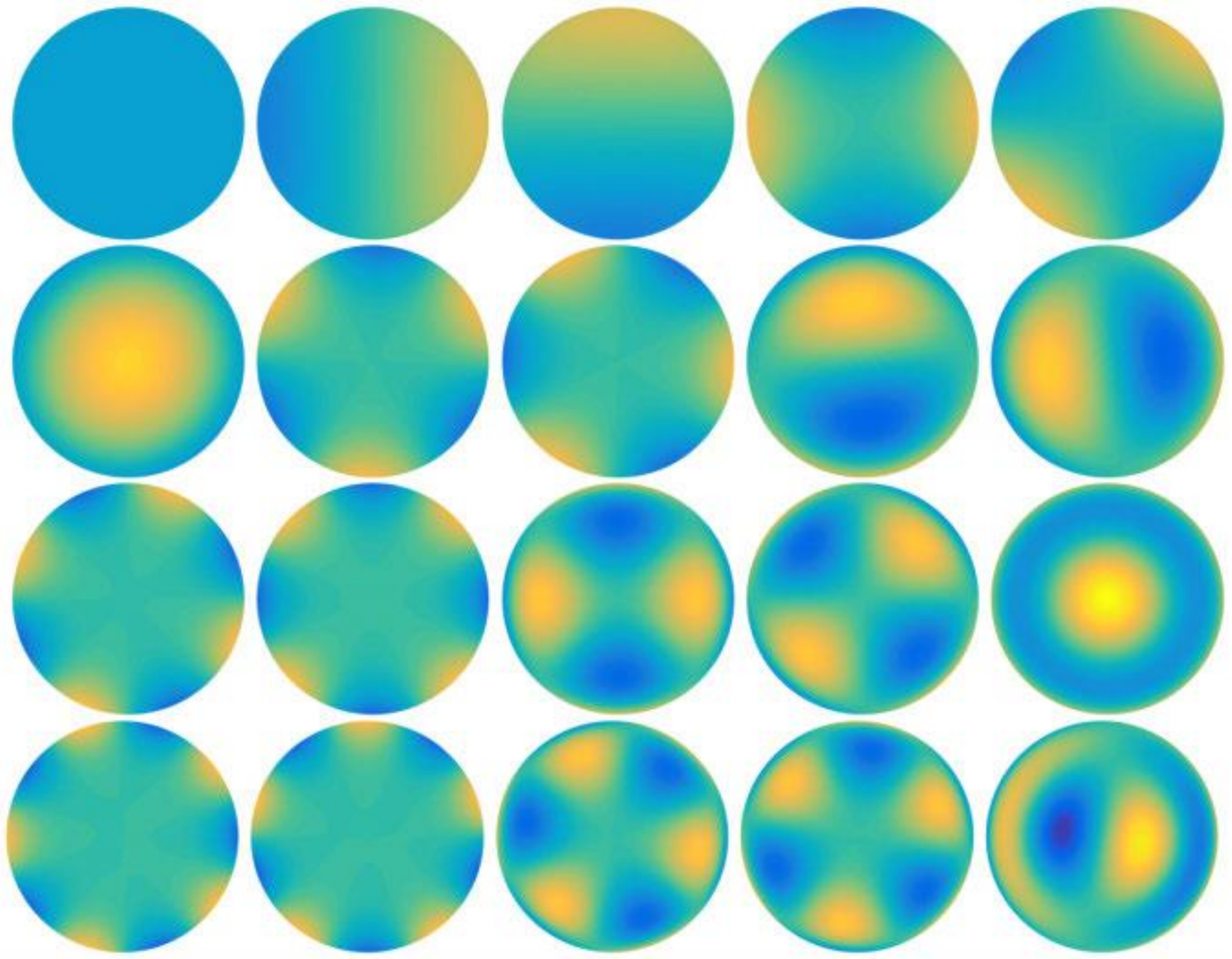


Steklov

Lapla-
cian

Generalization to Open Surfaces

- Example: Hemisphere.



Thanks you!

Questions?