Spectral shape analysis for 3D matching

Umberto Castellani, Riccardo Marin, Simone Melzi
University of Verona
July 29, 2020

Laplacian on discrete manifold

Outline

- Discrete data representation for 3D surfaces
- Laplace Beltrami operator on polygonal mesh
- Spectral shape analysis on discrete domain
- Diffusion process on surfaces
- Applications

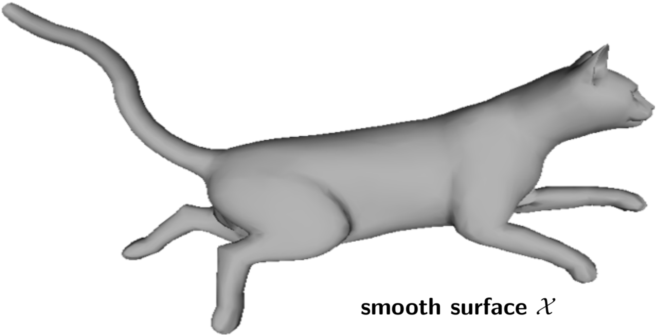
Discrete Laplacian

Laplacian is ubiquitus in many scientific areas. The discrete version of Laplacian depends by the data representation domain:

- **Euclidean Laplacian**: for the euclidean domain like 1D signals, 2D images, or 3D (or higher dimension) volumes.
- **Graph Laplacian**: for data organized in graph structures.
- **Geometric Laplacian** or **Laplace Beltrami**:for 3D shapes that approximate a continuous 2D Riemannian manifold.

Discrete data representation for 3D surfaces

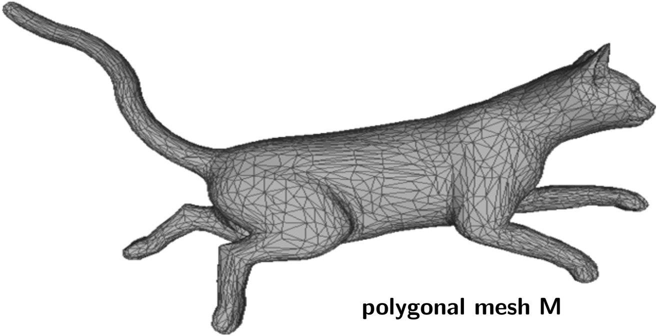
Assumption: meshes are piecewise linear approximations of smooth surfaces



smooth surface $\mathcal{X}$

Discrete data representation for 3D surfaces

Assumption: meshes are piecewise linear approximations of smooth surfaces



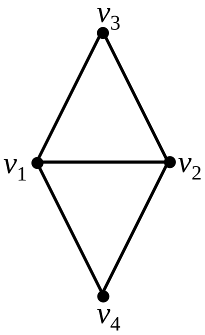
polygonal mesh M

Assumption: meshes are piecewise linear approximations of smooth surfaces

Two components:

\# xyz-coordinates of vertices
v 0 0 0
v 1 0 0
v .5 .866 0
v .5 -.866 0

\# vertex-face adjacency info
f 1 2 3
f 1 4 2



Extension to function on manifolds

Laplace-Beltrami operator

Gradient operator

$$\Delta_{\chi} f = Div_{\chi} \nabla_{\chi} f$$

Scalar function on manifold

Divergence operator

Laplace Beltrami on polygonal mesh

- function on triangular mesh
- gradient on triangular mesh
- divergence on triangular mesh
- cotangent weights scheme
- no free lunch

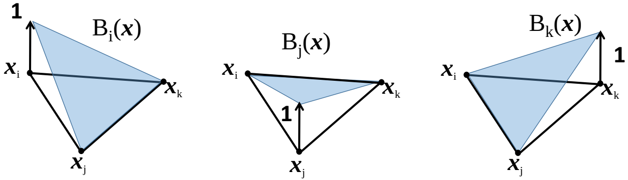
Function on triangular mesh

We assume a piecewise linear function f that is given at each mesh vertex as:

$$f(v_i) = f(\mathbf{x}_i) = f(\mathbf{u}_i) = f_i$$

This function is interpolated linearly within each triangle $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$:

$$f(\mathbf{u}) = f_i B_i(\mathbf{u}) + f_j B_j(\mathbf{u}) + f_k B_k(\mathbf{u})$$

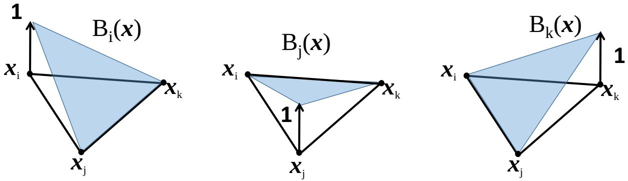


Function on triangular mesh

$\mathbf{u}(u, v)$ is the parameter pair corresponding to the surface point $\mathbf{x}$ in the parameterization induced by the triangle.

B_i, B_j, and B_k are the linear basis functions (i.e., **hat** functions) used for the barycentric interpolation on the triangle satisfying the barycentric condition of partition of unity.

$$B_i(\mathbf{u}) + B_j(\mathbf{u}) + B_k(\mathbf{u}) = 1 \qquad \forall \mathbf{u}$$



Gradient on triangular mesh

The gradient of the function f on the triangular mesh is given as:

$$\nabla f(\mathbf{u}) = f_i \nabla B_i(\mathbf{u}) + f_j \nabla B_j(\mathbf{u}) + f_k \nabla B_k(\mathbf{u})$$

In other words, the gradient is just a linear combination of the basis function gradients weighted by the per-vertex f values.

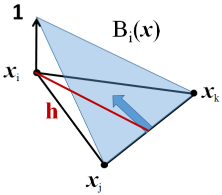
NOTE: the gradients of the basis functions sum to zero by leading to

$$\begin{aligned}
\nabla B_i(\mathbf{u}) &= -(\nabla B_j(\mathbf{u}) + \nabla B_k(\mathbf{u})) \\
&\Downarrow \\
\nabla f(\mathbf{u}) &= -f_i(\nabla B_j(\mathbf{u}) + \nabla B_k(\mathbf{u})) + f_j \nabla B_j(\mathbf{u}) + f_k \nabla B_k(\mathbf{u}) \\
&= -f_i \nabla B_j(\mathbf{u}) - f_i \nabla B_k(\mathbf{u}) + f_j \nabla B_j(\mathbf{u}) + f_k \nabla B_k(\mathbf{u}) \\
&= (f_j - f_i) \nabla B_j(\mathbf{u}) + (f_k - f_i) \nabla B_k(\mathbf{u})
\end{aligned}$$

Gradient on triangular mesh

Let's focus on the gradient of the basis function $\nabla B_i(\mathbf{u})$.

The gradient is constant (i.e., basis functions are linear over each triangle) and the steepest ascent direction of the basis functions is orthogonal to the opposite edge of the corresponding vertex.

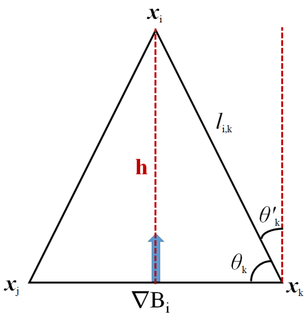


- $\nabla B_i(\mathbf{u})$ has the same direction of $\frac{(\mathbf{x}_k - \mathbf{x}_j)^{\perp}}{||\mathbf{x}_k - \mathbf{x}_j||}$
- the gradient module is $||\nabla B_i(\mathbf{u})|| = \frac{1}{h}$

Why $||\nabla B_i(\mathbf{u})|| = \frac{1}{h}$?

$B_i(\mathbf{x}_i) = 1$ and $B_i(\mathbf{x}_k) = 0$, then[a]

$$\begin{aligned} B_i(\mathbf{x}_i) - B_i(\mathbf{x}_k) &= \nabla B_i \cdot (\mathbf{x}_i - \mathbf{x}_k) \\ 1 &= ||\nabla B_i|| \cdot ||(\mathbf{x}_i - \mathbf{x}_k)|| \cdot cos\theta'_k \\ 1 &= ||\nabla B_i|| \cdot l_{i,k} \cdot sin\theta_k^b \\ 1 &= ||\nabla B_i|| \cdot h \\ &\Downarrow \\ ||\nabla B_i|| &= \frac{1}{h} \end{aligned}$$

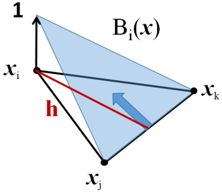


NOTE:
[a] See the Mean value theorem for function with constant gradient,
[b] $\theta'_k = (\frac{\pi}{2} - \theta_k)$, then $cos(\frac{\pi}{2} - \theta_k) = sin\theta_k$ (complementary angles).

Gradient on triangular mesh

Let's focus on the gradient of the basis function $\nabla B_i(\mathbf{u})$.

The gradient is constant (i.e., basis functions are linear over each triangle) and the steepest ascent direction of the basis functions is orthogonal to the opposite edge of the corresponding vertex.



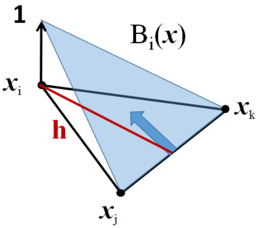
- $\nabla B_i(\mathbf{u})$ has the same direction of $\frac{(\mathbf{x}_k - \mathbf{x}_j)^\perp}{||\mathbf{x}_k - \mathbf{x}_j||}$
- the gradient module is $||\nabla B_i(\mathbf{u})|| = \frac{1}{h}$

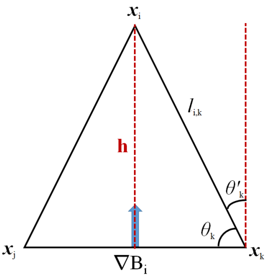
$$\Downarrow$$

$$\nabla B_i(\mathbf{u}) = \frac{1}{h} \frac{(\mathbf{x}_k - \mathbf{x}_j)^\perp}{||\mathbf{x}_k - \mathbf{x}_j||} = \frac{(\mathbf{x}_k - \mathbf{x}_j)^\perp}{2A_T}$$

The gradient of the piecewise linear function f within a triangle T becomes:

$$\begin{aligned}\nabla f(\mathbf{u}) &= (f_j - f_i)\nabla B_j(\mathbf{u}) + (f_k - f_i)\nabla B_k(\mathbf{u}) \\ &= (f_j - f_i)\frac{(\mathbf{x}_i - \mathbf{x}_k)^{\perp}}{2A_T} + (f_k - f_i)\frac{(\mathbf{x}_j - \mathbf{x}_i)^{\perp}}{2A_T}\end{aligned}$$





Extension to function on manifolds

Laplace-Beltrami operator

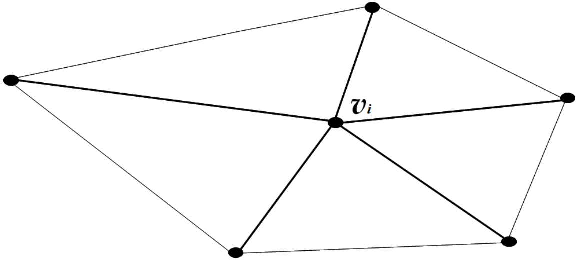
Gradient operator

$$\Delta_{\chi} f = Div_{\chi} \nabla_{\chi} f$$

Scalar function on manifold

Divergence operator

Divergence on triangular mesh

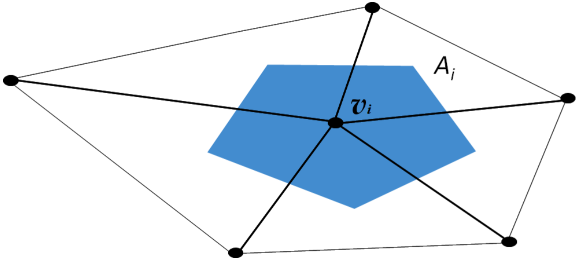


The goal is to evaluate the divergence of the gradient of a piecewise linear function over a local averaging domain $A_i = A(v_i)$.

$$\Downarrow$$

averaging means computing the integral

Divergence on triangular mesh

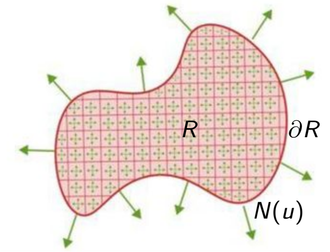


The goal is to evaluate the divergence of the gradient of a piecewise linear function over a local averaging domain $A_i = A(v_i)$.

$$\Downarrow$$

averaging means computing the integral

Divergence on triangular mesh

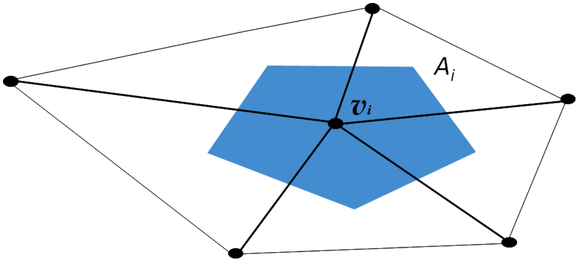


The **divergence theorem** for a vector-valued function $\mathbf{F}$ is given by:

$$\int_R div\mathbf{F}(\mathbf{u})dA = \int_{\partial R} \mathbf{F}(\mathbf{u}) \cdot N(\mathbf{u})ds$$

This theorem relates the integration over the averaging local area R to an integration along the boundary ∂R, where $N(\mathbf{u})$ is the outward pointing unit normal of the boundary.

Laplace Beltrami operator on triangular mesh

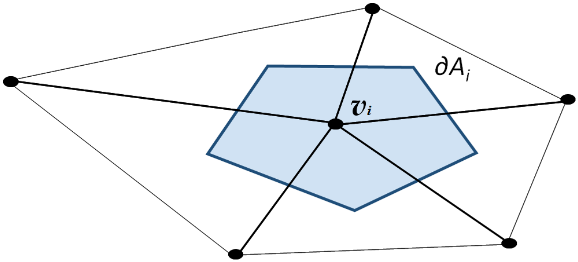


The **divergence theorem** applied for the Laplacian is:

$$\int_{A_i} \Delta f(\mathbf{u})dA = \int_{A_i} div(\nabla f(\mathbf{u}))dA = \int_{\partial A_i} \nabla f(\mathbf{u}) \cdot N(\mathbf{u})ds$$

Approach: approximate differential properties (i.e., Laplacian) at point v_i as spatial average over local mesh neighborhood $\mathcal{N}(v_i)$.
In practice, to get the Laplacian, we need to average over the vertices' Voronoi regions (region for which v_i is the closest vertex).

Laplace Beltrami operator on triangular mesh

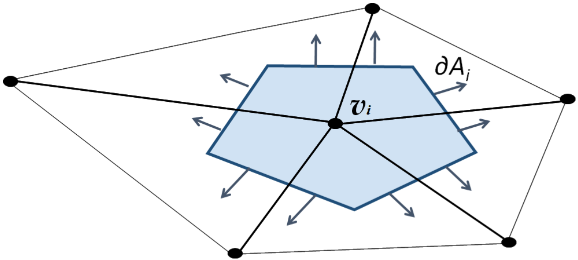


The **divergence theorem** applied for the Laplacian is:

$$\int_{A_i} \Delta f(\mathbf{u}) dA = \int_{A_i} div(\nabla f(\mathbf{u})) dA = \int_{\partial A_i} \nabla f(\mathbf{u}) \cdot N(\mathbf{u}) ds$$

Approach: approximate differential properties (i.e., Laplacian) at point v_i as spatial average over local mesh neighborhood $\mathcal{N}(v_i)$.
In practice, to get the Laplacian, we need to average over the vertices' Voronoi regions (region for which v_i is the closest vertex).

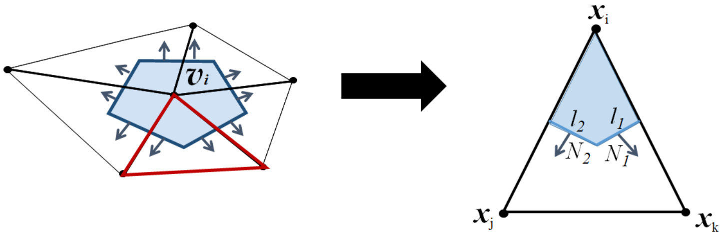
Laplace Beltrami operator on triangular mesh



The **divergence theorem** applied for the Laplacian is:

$$\int_{A_i} \Delta f(\mathbf{u}) dA = \int_{A_i} div(\nabla f(\mathbf{u})) dA = \int_{\partial A_i} \nabla f(\mathbf{u}) \cdot N(\mathbf{u}) ds$$

Approach: approximate differential properties (i.e., Laplacian) at point v_i as spatial average over local mesh neighborhood $\mathcal{N}(v_i)$.
In practice, to get the Laplacian, we need to average over the vertices' Voronoi regions (region for which v_i is the closest vertex).

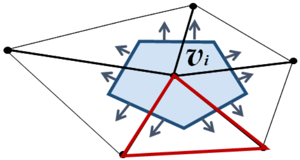


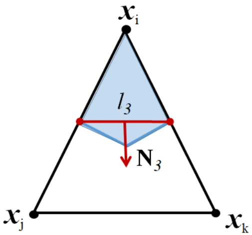
we split the integral into sum over each triangle (i.e., gradient is constant):

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot N(\mathbf{u}) ds = \nabla f(\mathbf{u}) \cdot (\mathbf{N}_1 l_1 + \mathbf{N}_2 l_2)$$

The boundary of the region inside the triangle is defined by the segments l_1 and l_2 with normals $\mathbf{N}_1$ and $\mathbf{N}_2$ respectively.

Laplace Beltrami operator on triangular mesh



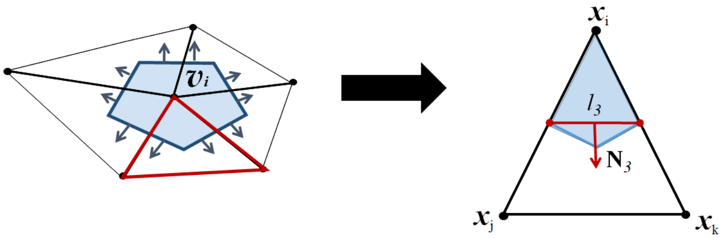


NOTE: according to Voronoi $\mathbf{N}_1 l_1 + \mathbf{N}_2 l_2 + \mathbf{N}_3 l_3 = 0 \Rightarrow \mathbf{N}_1 l_1 + \mathbf{N}_2 l_2 = -\mathbf{N}_3 l_3$, and therefore

$$-\mathbf{N}_3 l_3 = \left(\frac{\mathbf{x}_k + \mathbf{x}_i}{2} - \frac{\mathbf{x}_j + \mathbf{x}_i}{2}\right)^{\perp} = \frac{1}{2}(\mathbf{x}_k - \mathbf{x}_j)^{\perp}$$

i.e., the segment between the midpoints of the two edges (i, j) and (i, k).

Laplace Beltrami operator on triangular mesh



we get:

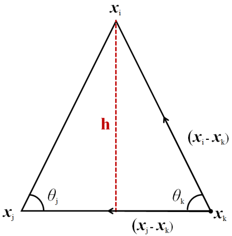
$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot N(\mathbf{u}) ds = \frac{1}{2} \nabla f(\mathbf{u}) \cdot (\mathbf{x}_j - \mathbf{x}_k)^{\perp}$$

plugin-in the gradient we obtain:

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot N(\mathbf{u}) ds = (f_j - f_i) \frac{(\mathbf{x}_i - \mathbf{x}_k)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4 A_T} + $$
$$+ (f_k - f_i) \frac{(\mathbf{x}_j - \mathbf{x}_i)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4 A_T}$$

Laplace Beltrami operator on triangular mesh

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot N(\mathbf{u}) ds = (f_j - f_i) \frac{(\mathbf{x}_i - \mathbf{x}_k)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4A_T} + \\ + (f_k - f_i) \frac{(\mathbf{x}_j - \mathbf{x}_i)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4A_T}$$



NOTE:

- $(\mathbf{x}_i - \mathbf{x}_k)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp} = ||\mathbf{x}_i - \mathbf{x}_k|| \cdot ||\mathbf{x}_j - \mathbf{x}_k|| \cdot cos\theta_k$, and
- $2A_T = h \cdot ||\mathbf{x}_j - \mathbf{x}_k|| = ||\mathbf{x}_i - \mathbf{x}_k|| \cdot sin\theta_k \cdot ||\mathbf{x}_j - \mathbf{x}_k||$

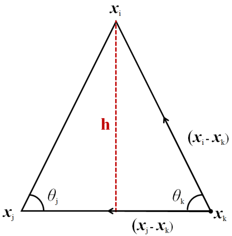
$$\Downarrow$$

$$\frac{(\mathbf{x}_i - \mathbf{x}_k)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{2A_T} = \frac{||\mathbf{x}_i - \mathbf{x}_k|| \cdot ||\mathbf{x}_j - \mathbf{x}_k|| \cdot cos\theta_k}{||\mathbf{x}_i - \mathbf{x}_k|| \cdot ||\mathbf{x}_j - \mathbf{x}_k|| \cdot sin\theta_k} = \frac{cos\theta_k}{sin\theta_k} = cot\theta_k$$

$$\frac{(\mathbf{x}_j - \mathbf{x}_i)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{2A_T} = \frac{||\mathbf{x}_i - \mathbf{x}_j|| \cdot ||\mathbf{x}_j - \mathbf{x}_k|| \cdot cos\theta_j}{||\mathbf{x}_i - \mathbf{x}_j|| \cdot ||\mathbf{x}_j - \mathbf{x}_k|| \cdot sin\theta_j} = \frac{cos\theta_j}{sin\theta_j} = cot\theta_j$$

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot N(\mathbf{u}) ds = (f_j - f_i) \frac{(\mathbf{x}_i - \mathbf{x}_k)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4A_T} + $$

$$+ (f_k - f_i) \frac{(\mathbf{x}_j - \mathbf{x}_i)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp}}{4A_T}$$



NOTE:

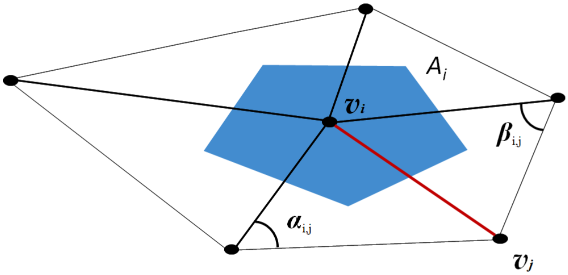
- $(\mathbf{x}_i - \mathbf{x}_k)^{\perp} (\mathbf{x}_j - \mathbf{x}_k)^{\perp} = ||\mathbf{x}_i - \mathbf{x}_k|| \cdot ||\mathbf{x}_j - \mathbf{x}_k|| \cdot cos\theta_k$, and
- $2A_T = h \cdot ||\mathbf{x}_j - \mathbf{x}_k|| = ||\mathbf{x}_i - \mathbf{x}_k|| \cdot sin\theta_k \cdot ||\mathbf{x}_j - \mathbf{x}_k||$

$$\Downarrow$$

$$\int_{\partial A_i \cap T} \nabla f(\mathbf{u}) \cdot N(\mathbf{u}) ds = \frac{1}{2}((f_j - f_i) cot\theta_k + (f_k - f_i) cot\theta_j)$$

where θ_k and θ_j are the angles at vertices v_k and v_j respectively.

Laplace Beltrami operator on triangular mesh

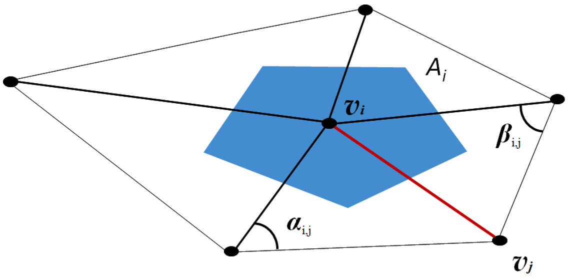


Now we can compute the integral over the entire averaging region A_i:

$$\int_{A_i} \Delta f(\mathbf{u}) dA = \frac{1}{2} \sum_{v_j \in \mathcal{N}_i} (cot\alpha_{i,j} + cot\beta_{i,j})(f_j - f_i)$$

Note: these terms have been re-arranged as a sum over edges.

Laplace Beltrami operator on triangular mesh



Finally, the discrete average of the Laplace-Beltrami operator of a function f at vertex v_i is given by

$$\Delta f(v_i) := \frac{1}{2A_i} \sum_{v_j \in \mathcal{N}_i} (cot\alpha_{i,j} + cot\beta_{i,j})(f_i - f_j)$$

this formulation is the so called **cotangent weights** scheme, i.e., the most widely used discretization of the Laplace-Beltrami operator for triangle meshes.

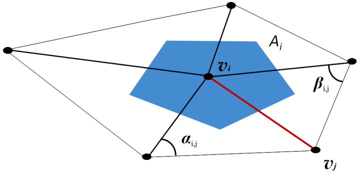
Laplace Beltrami operator on triangular mesh

The discrete LB operator in matrix form is given by:

$$\mathbf{L} = \mathbf{A}^{-1}\mathbf{W}$$

with $\mathbf{A} = diag([A_1, A_2, \cdots, A_n])$, and

$$w_{i,j} = \begin{cases} -\frac{1}{2}(cot\alpha_{i,j} + cot\beta_{i,j}) & i \sim j \\ -\sum_j w_{i,j} & i = j \\ 0 & otherwise \end{cases}$$



NOTE: there are many other different ways to derive this cotanget formulation such as piecewise linear finite elements (FEM), or discrete exterior calculus (DEC). All these different viewpoints yield exact same cotan formula.

Laplace Beltrami operator on triangular mesh

Are the nice properties of the Laplace Beltrami operator preserved on the discrete settings?

- **Simmetry** $\Rightarrow$ for orthogonal basis,
- **Locality** $\Rightarrow$ the effect of the operator on a point depends only by its local neighborhood,
- **Positive Semi Definite (PSD)** $\Rightarrow$ for non negative eigenvalues,
- **Convergence** $\Rightarrow$ solution of discrete PDE with $\mathbf{L}$ converges to the solution of continuous PDE with Δ for $n \rightarrow \infty$.

NOTE:The LB operator $\mathbf{L}$ derived from the cotangent scheme is not symmetric $\Rightarrow$ the eigenbasis are not orthogonal.

No free lunch [Wardetzky, 2007]: there is no discretization of the Laplace-Beltrami operator satisfying simultaneously all the desired properties!

Spectral shape analysis

REMEMBER: we need the eigenfunctions of the Laplace Beltrami operator as Fourier basis for functions defined on the 3D shape.

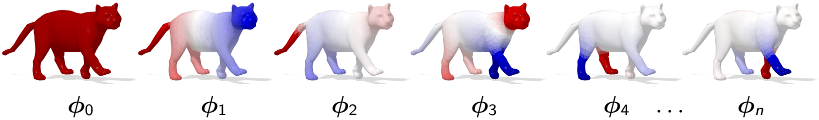
- Spectral decomposition on discrete manifold,
- Inner product for functions on discrete manifold,
- Fourier representation of functions on discrete manifold,

Spectral decomposition on discrete manifold

The eigendecomposition of LB operator is defined as:

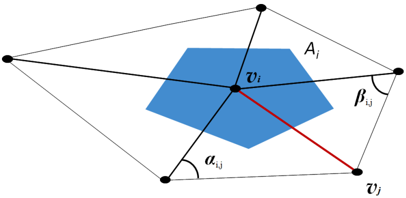
$$\mathbf{L}\phi_{\mathbf{i}} = \lambda_i \phi_{\mathbf{i}} \qquad \forall i = 1, \cdots, n$$

such that $0 = \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$



The eigenfunctions of the LBO are called **manifold harmonics**

Spectral decomposition on discrete manifold



Since $\mathbf{L} = \mathbf{A}^{-1}\mathbf{W}$ we can employ the *generalized* eigendecomposition:

$$\mathbf{W}\phi_\mathbf{i} = \lambda_i \mathbf{A}\phi_\mathbf{i} \qquad \forall i = 1, \cdots, n$$

NOTE:**W** and **A** are symmetric and semi-definite positive

$$\Downarrow$$

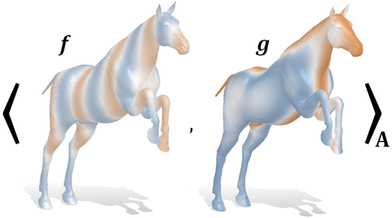
the eigenfunctions are **A**-orthogonal

$$\phi_i^\top \mathbf{A}\phi_i = 1, \qquad and \qquad \phi_i^\top \mathbf{A}\phi_j = 0 \qquad i \neq j$$

IDEA: we can define a new inner product to make the basis orthogonal

$$\langle \mathbf{f}, \mathbf{g} \rangle_{\mathbf{A}} = \sum_{i=1}^{n} A_i f(v_i) g(v_i) = \mathbf{f}^\top \mathbf{A} \mathbf{g}$$

where the functions $\mathbf{f}$ and $\mathbf{g}$ are defined only on the vertices v_i, $\forall i = 1, \cdots, n$



A function $f : \mathcal{M} \rightarrow \mathcal{R}$ can be represented by a Fourier series by:

$$\mathbf{f} = \sum_{i=1}^{n} \langle \mathbf{f}, \phi_i \rangle_{\mathbf{A}} \phi_i$$

In particular, we define $\mathbf{a} = (\langle \mathbf{f}, \phi_1 \rangle_{\mathbf{A}}, \langle \mathbf{f}, \phi_2 \rangle_{\mathbf{A}}, \cdots \langle \mathbf{f}, \phi_n \rangle_{\mathbf{A}})^\top$ the Fourier coefficients (i.e., the **Analysis**). In matrix form we obtain:

$$\mathbf{a} = \mathbf{\Phi}^\top \mathbf{A} \mathbf{f}$$

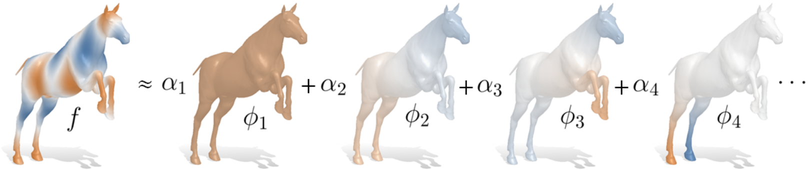
and

$$\mathbf{f} = \mathbf{\Phi} \mathbf{a}$$

is the **Synthesis**, where $\mathbf{\Phi} = (\phi_1, \phi_2, \cdots, \phi_n)$.

Fourier representation of functions on discrete manifold

$$\mathbf{f} = \mathbf{\Phi}\mathbf{a}$$



Fourier representation of functions on discrete manifold

According to the Fourier methodology it is possible to work with a truncated basis $\mathbf{\Phi^k} = (\phi_1, \phi_2, \cdots, \phi_k)$:

$$\mathbf{a} = \mathbf{\Phi}^{k\top}\mathbf{Af}$$

and

$$\begin{aligned}\hat{\mathbf{f}} &= \sum_{i=1}^{k} \alpha_i \phi_i^k \\ &= \mathbf{\Phi}^k \mathbf{a} \\ &= \mathbf{\Phi}^k \mathbf{\Phi}^{k\top}\mathbf{Af}\end{aligned}$$

i.e., $\hat{\mathbf{f}}$ is an approximation of the function $\mathbf{f}$ obtained from the Fourier reconstruction using a limited set of k basis (i.e., low pass representation).

Fourier representation of functions on discrete manifold

$$\hat{\mathbf{f}} = \mathbf{\Phi}^k \mathbf{a}$$

$\boldsymbol{k} = 1$ $\boldsymbol{k} = 50$ $\boldsymbol{k} = 1000$

Diffusion process on surfaces

- Laplace Beltrami and heat diffusion
- heat kernel
- spectral interpretation
- discrete settings

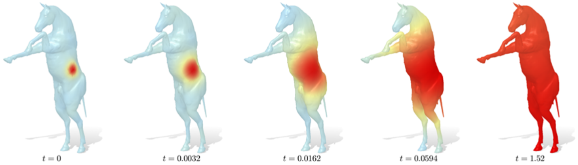
Laplace Beltrami and heat diffusion

The Laplace Beltrami operator is involved on the heat diffusion equation:

$$(\Delta_{\mathcal{X}} + \frac{\partial}{\partial t})u = 0$$

the heat equation governs heat propagation on manifold $\mathcal{X}$

- *the solution* $u(\mathbf{x}, t)$ is the heat distribution at point $\mathbf{x}$ at time t,
- *initial condition* $u_0(\mathbf{x})$ is the heat distribution at time $t = 0$,
- *boundary condition* if manifold has a boundary



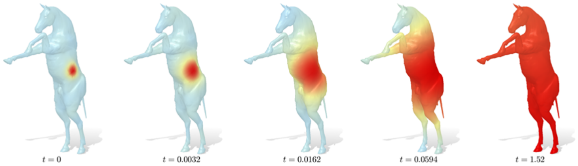
The solution of the heat diffusion equation

$$(\Delta_{\mathcal{X}} + \frac{\partial}{\partial t})u = 0$$

is given by

$$\begin{aligned} u(\mathbf{x}, t) &= \mathbf{H}^t u_0(\mathbf{x}) \\ &= e^{-t\Delta_{\mathcal{X}}} u_0(\mathbf{x}) \end{aligned}$$

where $\mathbf{H}^t$ is the **heat operator**.

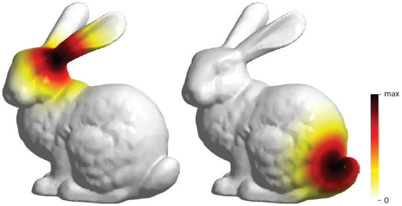


Heat kernel

The solution of the heat diffusion equation can be further exploited as:

$$u(\mathbf{x}, t) = \mathbf{H}^t u_0(\mathbf{x}) = \int_{\mathcal{X}} h_t(\mathbf{x}, \mathbf{y}) u_0(\mathbf{y}) da(\mathbf{y})$$

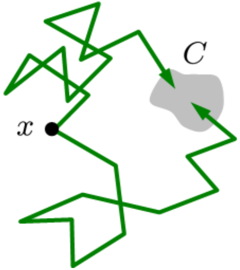
where $h_t(\mathbf{x}, \mathbf{y}) : \mathcal{R}^+ \times \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ is the **heat kernel**, i.e., the solution at point $\mathbf{x}$ at time t with point heat source at $\mathbf{y}$ or the amount of heat transferred from $\mathbf{x}$ to $\mathbf{y}$ in time t.



Probabilistic interpretation

Brownian motion $\mathbf{x}(t)$ starts at point $\mathbf{x}$

- $Pr(\mathbf{x}(t) \in C) = \int_C h_t(\mathbf{x}, \mathbf{y}) da(\mathbf{y})$: Probability that the Brownian motion will be in $C \subset \mathcal{X}$ at time t,
- $h_t(\mathbf{x}, \mathbf{y})$ is the transition probability density from $\mathbf{x}$ to $\mathbf{y}$ by random walk of length t, i.e., average across all paths



Spectral interpretation

- Let $\Delta_{\mathcal{X}}\phi_i = \lambda_i \phi_i$ be the Laplacian eigendecomposition,
- By spectral decomposition theorem we obtain:

$$\begin{aligned} u(\mathbf{x}, t) &= e^{-t\Delta_{\mathcal{X}}} u_0(\mathbf{x}) \\ &= e^{-t\Delta_{\mathcal{X}}} \sum_{i \geq 0} \langle u_0(\mathbf{x}), \phi_i(\mathbf{x}) \rangle_{\mathcal{X}} \phi_i(\mathbf{x}) \\ &= \sum_{i \geq 0} \langle u_0(\mathbf{x}), \phi_i(\mathbf{x}) \rangle_{\mathcal{X}} e^{-\lambda_i t} \phi_i(\mathbf{x}) \\ &= \sum_{i \geq 0} \int_{\mathcal{X}} u_0(\mathbf{y}) \phi_i(\mathbf{y}) da(\mathbf{y}) e^{-\lambda_i t} \phi_i(\mathbf{x}) \\ &= \int_{\mathcal{X}} u_0(\mathbf{y}) \underbrace{\sum_{i \geq 0} e^{-\lambda_i t} \phi_i(\mathbf{x}) \phi_i(\mathbf{y})}_{h_t(\mathbf{x}, \mathbf{y})} da(\mathbf{y}) \end{aligned}$$

- We obtain the spectral version of the heat kernel as:

$$h_t(\mathbf{x}, \mathbf{y}) = \sum_{i \geq 0} e^{-\lambda_i t} \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$$

Discrete settings

In a discrete domain only a small number $k < n$ of eigenfunctions are considered, by leading to

$$h_t(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} e^{-\lambda_i t} \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$$

The heat kernel has very nice properties:

- it is an isometric invariant,
- it is informative,
- it is multi-scale,
- it is stable.

$$\Downarrow$$

This suggest to exploit the heat kernel to capture reliable features of the shape!

Discrete settings

It is interesting to evaluate the **autodiffusion**:

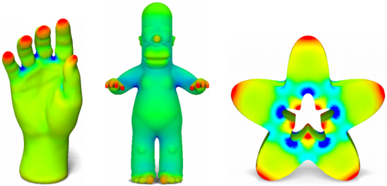
$$h_t(\mathbf{x}, \mathbf{x}) = \sum_{i=1}^{k} e^{-\lambda_i t} \phi_i(\mathbf{x})^2$$

that measures how much heat remains at point $\mathbf{x}$ after time t.

NOTE: for small t, the heat kernel $h_t(\mathbf{x}, \mathbf{x})$ is directly related to the Gaussian curvature at $\mathbf{x}$.

$$\Downarrow$$

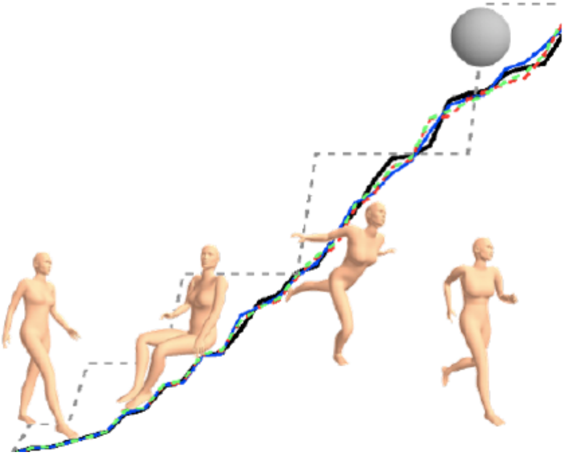
high and low values of $h_t(\mathbf{x}, \mathbf{x})$ correspond to areas with positive and negative Gaussian curvatures respectively.



Applications

- Shape DNA
- Coordinate reconstruction
- Spectral embedding
- Segmentation by Fiedler vector

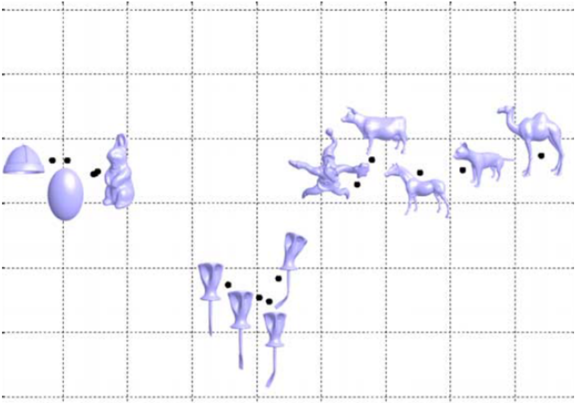
Shape DNA



[Reuter et al., 2006] use Laplacian spectrum $\{\lambda_i\}$ as an isometry-invariant shape descriptor called shape DNA

Shape DNA

- For each shape in the collection, compute its Laplace Beltrami operator,
- Find the k smallest eigenvalues and store them in a vector,
- Compare the shapes by comparing the corresponding vectors.



IDEA: The coordinate $\mathbf{x}_i(x, y, z)$ of the vertices v_i can be considered as a set of three functions:

$$\mathbf{f}_x = \{x_i\} \qquad \mathbf{f}_y = \{y_i\} \qquad \mathbf{f}_z = \{z_i\}$$

We can obtain different representations of the original coordinate functions setting different value of k:

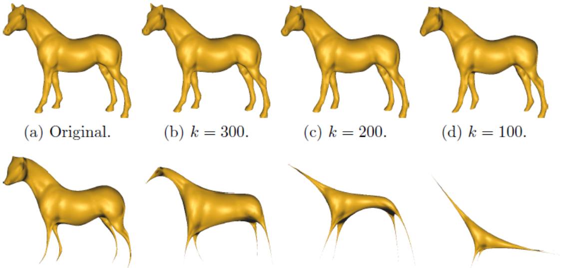
$$\hat{\mathbf{f}}_x = \sum_{i=1}^{k} \langle \mathbf{f}_x, \phi_i \rangle_{\mathbf{A}} \phi_i$$

$$\hat{\mathbf{f}}_y = \sum_{i=1}^{k} \langle \mathbf{f}_y, \phi_i \rangle_{\mathbf{A}} \phi_i$$

$$\hat{\mathbf{f}}_z = \sum_{i=1}^{k} \langle \mathbf{f}_z, \phi_i \rangle_{\mathbf{A}} \phi_i$$

Then the reconstruction of the original shape is recovered by re-arranging the coordinates $\hat{\mathbf{x}}_i(\hat{x}_i, \hat{y}_i, \hat{x}_i)$

Coordinate reconstruction



(a) Original. (b) $k = 300$. (c) $k = 200$. (d) $k = 100$.
(e) $k = 50$. (f) $k = 10$. (g) $k = 5$. (h) $k = 3$.

[Levy et al., 2006] Coordinate reconstruction with k from 3 to 300.

Spectral embedding

The set of the first k LBO eigenfunctions $\phi_0, \phi_i, \cdots, \phi_k$ defines a k-dimensional embedding called **Spectral Embedding**:

$$\Phi : \mathcal{X} \to \mathbb{R}^{k+1}$$

$$\mathbf{p} \to \Phi(\mathbf{p}) = [\phi_0(p), \phi_1(p), \cdots \phi_k(p)], \qquad \forall \mathbf{p} \in \mathcal{X}$$

properties:

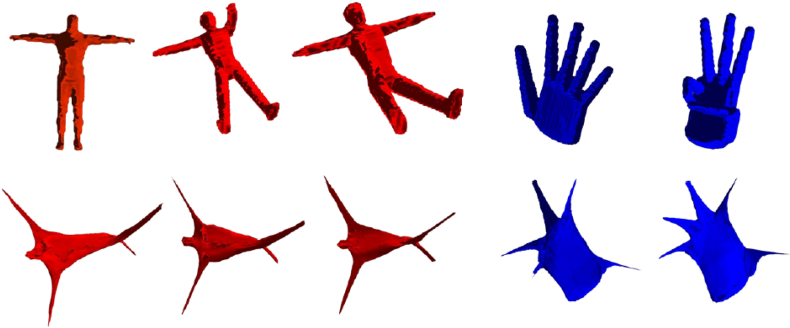
- This embedding is invariant w.r.t. isometric deformations.
- The dimension k is small ≈ 10 or ≈ 100.
- With a small k this embedding is invariant also for near isometric shapes.

limitations:

- The sign of the eigenfunctions is not fixed,
- In the discrete setting the isometry is not easily represented.

Spectral embedding

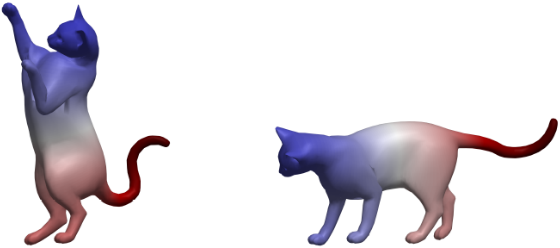
[Zhang et al., 2011] Spectral mesh processing.



[Zhang et al., 2011] Spectral mesh processing.

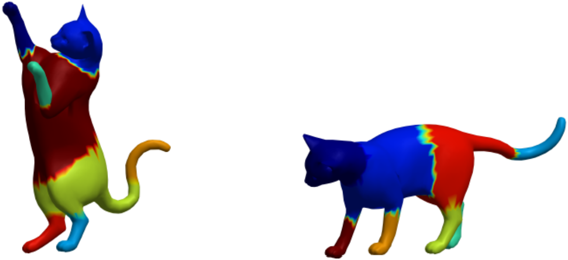
Segmentation by Fiedler vector

REMEMBER: the **Fiedler vector** is defined by the eigenfunction associated with the smallest non-zero eigenvalue of the Laplace-Beltrami operator. The nice properties of the Fiedler vector for graphs remain valid also for 3D shape.



Segmentation by Fiedler vector

From the **Fiedler vector** is possible to define a coherent order of the shape vertices $\Rightarrow$ a shape segmentation can be carried out.



Conclusions

- The geometric Laplacian is very important for the analysis of 3D shapes,
- The discretization of the Laplace Beltrami operator is feasible even if an optimal solution does not exist. The most used method is the cotangent weights scheme.
- The Fourier framework for 3D shapes can be defined from the spectral properties of the Laplace Beltrami operator.

Conclusions

- The Laplace Beltrami operator is employed to compute the heat diffusion on 3D surface from which important properties of the shape can be captured.
- Several applications can be effectively organized for 3D shapes from the Laplace Beltrami operator.
- The theory behind spectral shape analysis is very founded and charming, we have just scratch the surface..., more in the next days...