Mesh Laplacian and Manifold Harmonics

Julie Digne

LIRIS - CNRS

14/10/2019
About the second part of this course

- Part V (14/10, 21/10, 04/11): Mesh Laplacian and Manifold harmonics
- Part VI (18/11, 25/11): Optimal Transport (with Nicolas)
- Part VII (02/12): Markov Random Fields
- Part VIII (09/12, 16/12): Machine Learning for Graphics and Vision
Part V: Mesh Laplacian and Manifold Harmonics

- Building a Mesh-Laplacian and applying it for deformation problems
- How to perform Fourier-like analysis for functions defined on mesh surfaces?
Part VII: Markov Random Fields and Texture Synthesis

- Markov Random Fields for modeling Textures, denoising, segmentation
Part VIII: Sparse Modeling and Patch-based image Synthesis

- Modeling an image via a *sparse representation*
- Application to super-resolution, terrain synthesis...
Course validation

- **Paper reading** A list of papers to choose from is available on the course webpage. These papers go much further than the course. You’re expected to be able to explain the method and answer questions on the paper.

- **Project** A project should be chosen from the set of proposed projects. These projects should take you around $x$ hours work: these projects can be found on the course webpage. The code as well as a report on the experiments run are expected.

Either way...

You should choose a project in the first part of the course and a paper to read in the second part of the course or conversely.
Today: Laplacian and Manifold Harmonics Basis

- A shape defined by a mesh.
- Functions defined on a manifold surface $\mathcal{M}$ with values in $\mathbb{R}$.
- Define a basis for representing such functions.
Outline

1. Building a Laplacian Operator on meshes
2. Surface Parameterization
3. Applications of the Laplacian on meshes
4. Manifold Harmonics Basis
5. Applications of the MHT
6. Functional Maps
7. A better basis?
Laplacian operator

For one-dimensional functions

Let $f$ be a function defined on $\mathbb{R}$, then the laplacian $\Delta f = \text{div grad } f = \frac{\partial^2 f}{\partial t^2}$ (eq. to $\Delta f = \nabla \cdot \nabla f$)
Laplacian operator

For one-dimensional functions

Let $f$ be a function defined on $\mathbb{R}$, then the laplacian $\Delta f = \text{div} \text{ grad } f = \frac{\partial^2 f}{\partial t^2}$ (eq. to $\Delta f = \nabla \cdot \nabla f$)

For one-dimensional discrete functions

Let $f$ be a sampled function: $f : [1, N] \rightarrow \mathbb{R}$, then the laplacian

$$\Delta f = \frac{1}{2}(f(j+1) - 2f(j) + f(j-1))$$
A basis for sampled functions

\[ f : i \rightarrow f_i \text{ for } i = 1 \cdots N \]

- Simplest basis:
A basis for sampled functions

\[ f : i \rightarrow f_i \text{ for } i = 1 \cdots N \]

- Simplest basis: a set of \( N \) Dirac functions regularly sampled \( b_i(j) = \delta_{ij} \), for \( i, j \in 1 \cdots N \)

Building a Laplacian Operator on meshes
A basis for sampled functions

\[ f : i \rightarrow f_i \text{ for } i = 1 \cdots N \]

- Simplest basis: a set of \( N \) Dirac functions regularly sampled \( b_i(j) = \delta_{ij} \), for \( i, j \in 1 \cdots N \)
- Build the matrix \( L \) such that \( \Delta f = Lf \).
A basis for sampled functions

\[ f : i \rightarrow f_i \text{ for } i = 1 \cdots N \]

- Simplest basis: a set of \( N \) Dirac functions regularly sampled \( b_i(j) = \delta_{ij} \), for \( i, j \in 1 \cdots N \)
- Build the matrix \( L \) such that \( \Delta f = Lf \).
- How are the eigenfunctions of the Laplacian operator?
A basis for sampled functions

\[ f : i \rightarrow f_i \text{ for } i = 1 \cdots N \]

- Simplest basis: a set of \( N \) Dirac functions regularly sampled \( b_i(j) = \delta_{ij} \), for \( i, j \in 1 \cdots N \)
- Build the matrix \( L \) such that \( \Delta f = Lf \).
- How are the eigenfunctions of the Laplacian operator? Orthogonal since the Laplacian is self-adjoint
A basis for sampled functions

\[ f : i \rightarrow f_i \text{ for } i = 1 \cdots N \]

- Simplest basis: a set of \( N \) Dirac functions regularly sampled \( b_i(j) = \delta_{ij} \), for \( i, j \in 1 \cdots N \)
- Build the matrix \( L \) such that \( \Delta f =Lf \).
  
  How are the eigenfunctions of the Laplacian operator? Orthogonal since the Laplacian is self-adjoint

  How are the eigenvectors of \( L \)?
A basis for sampled functions

\[ f : i \rightarrow f_i \text{ for } i = 1 \cdots N \]

- Simplest basis: a set of \( N \) Dirac functions regularly sampled \( b_i(j) = \delta_{ij} \), for \( i, j \in 1 \cdots N \)
- Build the matrix \( L \) such that \( \Delta f = Lf \).
- How are the eigenfunctions of the Laplacian operator? **Orthogonal** since the Laplacian is self-adjoint
- How are the eigenvectors of \( L \)? **Orthogonal** if the Laplacian matrix is a real symmetric matrix.
An example on a 1D signal

1D-Laplacian - Finite Elements

\[ \nabla u(i) = u(i + 1) - u(i) \]

\[ \Delta u(i) = \frac{u(i + 1) + u(i - 1) - 2u(i)}{2} \]
Basis functions for 1D signals
Laplacian operator

For two-dimensional functions

Let $f$ be a function defined on $\mathbb{R}$, then the laplacian $\Delta f = \text{div grad } f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
Laplacian operator

For two-dimensional functions

Let \( f \) be a function defined on \( \mathbb{R} \), then the laplacian \( \Delta f = \text{div} \, \text{grad} \, f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \)

For two-dimensional discrete functions

Let \( f \) be a sampled function: \( f : [1, N]^2 \rightarrow \mathbb{R} \), then the laplacian

\[
\Delta f = \frac{1}{2} (f(i, j + 1) - 2f(i, j) + f(i, j - 1)) + \frac{1}{2} (f(i + 1, j) - 2f(i, j) + f(i - 1, j))
\]
Basis functions for a 2D grid domain
Functions defined on a manifold surface

- What is a manifold surface?
Functions defined on a manifold surface

- What is a manifold surface?
- How to represent numerically a manifold surface?
A Riemann surface $S$ is a separated (Hausdorff) topological space endowed with an atlas: For every point $x \in S$ there is a neighborhood $V(x)$ containing $x$ homeomorphic to the unit disk of the complex plane. These homeomorphisms are called charts. The transition maps between two overlapping charts are required to be holomorphic.
A Riemann surface $S$ is a separated (Hausdorff) topological space endowed with an atlas: For every point $x \in S$ there is a neighborhood $V(x)$ containing $x$ homeomorphic to the unit disk of the complex plane. These homeomorphisms are called charts. The transition maps between two overlapping charts are required to be holomorphic.

- At each point of the surface one can find an intrinsic parameterization $T(u, v)$. 
Riemann Surface

A Riemann surface $S$ is a separated (Hausdorff) topological space endowed with an atlas: For every point $x \in S$ there is a neighborhood $V(x)$ containing $x$ homeomorphic to the unit disk of the complex plane. These homeomorphisms are called charts. The transition maps between two overlapping charts are required to be holomorphic.

- At each point of the surface one can find an intrinsic parameterization $T(u, v)$.
- We restrict this small introduction to surfaces of dimension 2 embedded in $\mathbb{R}^3$. 
Let $S$ be a smooth surface embedded in $\mathbb{R}^3$, parameterized over a bounded domain $\Omega \subset \mathbb{R}^2$ with parameterization:

$$
\mathbf{x}(u, v) = \begin{pmatrix}
x(u, v) \\
y(u, v) \\
z(u, v)
\end{pmatrix}
$$

- Define $\mathbf{x}_u(u_0, v_0) = \frac{\partial \mathbf{x}}{\partial u}(u_0, v_0)$
- $\mathbf{x}_v(u_0, v_0) = \frac{\partial \mathbf{x}}{\partial v}(u_0, v_0)$ is tangent to the curve on the surface defined by $s \rightarrow \mathbf{x}(u_0, v_0 + s)$.
- $\mathbf{x}_u(u_0, v_0)$ and $\mathbf{x}_v(u_0, v_0)$ are two vectors tangent to the surface $S$.
- If the parameterization is regular, ($||\mathbf{x}_u \times \mathbf{x}_v|| \neq 0$), these vectors span the tangent plane to the surface at $\mathbf{x}(u_0, v_0)$.
Normal computation

- If the parameterization is regular, the normal to the surface is computed as:
  \[ n = \frac{x_u \times x_v}{\|x_u \times x_v\|} \]

- **Directional derivatives** Given a direction \( w \) in the tangent plane, the directional derivative of \( S \) in direction \( w \) is the tangent to the curve \( C_w(t) = x(u_0, v_0 + tw) \)
First Fundamental Form

Definition (First Fundamental Form)

The **First Fundamental Form** is defined as $I = J \cdot J^T$ (2 × 2 matrix). or equivalently:

$$I = \begin{pmatrix} x_u^T x_u & x_u^T x_v \\ x_v^T x_u & x_v^T x_v \end{pmatrix}$$

where $J$ is the Jacobian matrix of $S$: $J = (x_u \quad x_v)$ (3 × 2 matrix).
Why is the first fundamental useful?

- \( \mathbf{a} \in \Omega, \tilde{a} \) corresponding tangent vector: \( \| \mathbf{a} \|^2 = \tilde{a}^T J^T J \tilde{a} = \tilde{a}^T I \tilde{a} \)
- Length of a curve \( C(t) = \mathbf{x}(u(t), v(t)) \):
  \[
  l_{[a,b]} = \int_{[a,b]} (u_t v_t) I (u_t v_t)^T
  \]
- Surface Area \( A = \int \int_A \sqrt{\det I} \, du dv \)
**Definition (Second Fundamental Form)**

The **Second fundamental form** characterizes the way a surface bends:

\[
II = \begin{pmatrix}
    x_{uu}^T \cdot n & x_{uv}^T \cdot n \\
    x_{uv}^T \cdot n & x_{vv}^T \cdot n
\end{pmatrix}
\]

It is a quadratic form on the tangent plane to the surface.
As a starter: curvature of a curve
Normal Curvature

**Definition (Normal Curvature)**

For each tangent vector $\mathbf{t}$ at a point $p$ of the surface, the normal curvature is defined as:

$$\kappa_n(t) = \frac{\mathbf{t}^T \cdot I / \cdot \mathbf{t}}{\mathbf{t}^T \cdot I / \cdot \mathbf{t}}.$$  

- The normal curvature varies with $\mathbf{t}$.  

*Image from Crane et al. 2013*
Principal curvatures and directions

Definition (Principal curvature)
Let \( \kappa_1 \) be the minimum of \( \kappa_n(t) \) (normal curvature at \( p \)) and \( \kappa_2 \) be the maximum of \( \kappa_n(t) \). \( \kappa_1 \) and \( \kappa_2 \) are called the principal curvatures of the surface at \( p \).

- If \( \kappa_1 \neq \kappa_2 \), the two associated tangent vectors \( t_1 \) and \( t_2 \) are called principal directions and they are orthogonal.
- \( \kappa_1, \kappa_2, t_1, t_2 \) are the eigenvalues and eigenvectors of the Shape Operator:

\[
S = I^{-1} \cdot II
\]
If \( \kappa_1 = \kappa_2 \), the point is called an umbilic or umbilical point and the surface is locally spherical.

\[ \kappa_n(t) = \kappa_1 \cos^2 \phi + \kappa_2 \sin^2 \phi \quad \text{(Euler)} \]

\( \phi \): angle between \( t_1 \) and \( t \)

\( (t_1, t_2, n) \) is called the local intrinsic coordinate system.
The **Curvature Tensor** is a symmetric $3 \times 3$ matrix $C$ whose eigenvalues are $(\kappa_1, \kappa_2, 0)$ and corresponding eigenvectors $(t_1, t_2, n)$. More precisely:

$$C = PDP^{-1}$$

where $P$ is the matrix whose columns are $t_1, t_2, n$ and $D$ is a diagonal matrix with diagonal values $\kappa_1, \kappa_2, 0$. 

**Definition (Curvature Tensor)**
Curvature Tensor Estimation

**Definition (Normal cycles)**

For each edge of the meshed surface, $\kappa_2 = 0$ and $\kappa_1 = \beta(e)$ is the dihedral angle between the normals of the two facets adjacent to edge $e$. Let: $\bar{e} = e/\|e\|

\[
C(v) = \frac{1}{A(v)} \sum_{e \in \mathcal{N}(v)} \beta(e) \|e \cap A(v)\| \bar{e} \cdot \bar{e}^T.
\]

Morvan, Cohen-Steiner 2003
• **Mean curvature** average of the normal curvature: $H = \frac{\kappa_1 + \kappa_2}{2}$

• **Gaussian curvature** product of the principal curvature $K = \kappa_1 \cdot \kappa_2$

• **Elliptical point** point with positive Gaussian curvature

• **Hyperbolic point** point with negative Gaussian curvature

• **Parabolic point** point with zero Gaussian curvature
Example: a cylinder

- $\kappa_1 = \frac{1}{r}$; $\kappa_2 = 0$
- $H = \frac{1}{2r}$
- $K = 0$
Example: a torus
Laplace Beltrami Operator

Definition (Laplace operator)

The Laplace operator is the divergence of the gradient operator:

\[ \Delta f = \text{div } \nabla f = \text{div} \left( \begin{pmatrix} \frac{\partial f}{\partial u} \\ \frac{\partial f}{\partial v} \end{pmatrix} \right) = \frac{\partial^2 f}{\partial u^2} + \frac{\partial^2 f}{\partial v^2} \]
Laplace Beltrami Operator

Definition (Laplace operator)
The Laplace operator is the divergence of the gradient operator:

\[ \Delta f = \text{div } \nabla f = \text{div } \left( \frac{\partial f}{\partial u} \right) = \frac{\partial^2 f}{\partial u^2} + \frac{\partial^2 f}{\partial v^2} \]

Definition (Laplace-Beltrami operator)
The Laplace-Beltrami operator extends the notion of Laplacian to a function defined on a surface.

\[ \Delta_S f = \text{div}_S \nabla_S f \]

Where \( \text{div}_S \) and \( \nabla_S \) are defined on the manifold.
Representing manifold surfaces

**Mesh Surface**

Polygonal meshes are a piecewise linear approximation of the shape. It is a set of polygons linked together by edges.
Representing manifold surfaces

**Mesh Surface**

Polygonal meshes are a piecewise linear approximation of the shape. It is a set of polygons linked together by edges.

- Triangular or quadrilateral meshes are used.
Each point on the surface can be expressed in terms of barycentric coordinates of the three vertices of the facet it belongs to.
Triangular Meshes

Each point on the surface can be expressed in terms of barycentric coordinates of the three vertices of the facet it belongs to.

Euler Formula

Link between the number of triangles $F$, edges $E$ and vertices $V$ of a closed non-intersecting triangular mesh [Coxeter89] with genus $g$ (number of handles in the surface).

$$V - E + F = 2(1 - g)$$
"Manifoldness"

- **2-manifold surface**: at each point, the surface is locally homeomorphic to a disk (or half disk if the vertex lies on the boundary).
“Manifoldness”

- **2-manifold surface**: at each point, the surface is locally homeomorphic to a disk (or half disk if the vertex lies on the boundary).
- **In practice**, each edge is adjacent to at most two triangles. Two surface sheets cannot meet at a vertex.
“Manifoldness”

- **2-manifold surface**: at each point, the surface is locally homeomorphic to a disk (or half disk if the vertex lies on the boundary).
- In practice, each edge is adjacent to at most two triangles. Two surface sheets cannot meet at a vertex.
- **Closed surface**: Each edge has exactly two adjacent edges.
Statistics about meshes

- Number of triangles \( \approx 2 \times \) number of vertices
- Number of edges \( \approx 3 \times \) number of vertices
- Average number of incident edges to a vertex: 6

**Notation**

\( N(i) \): 1-ring neighborhood of vertex \( i \): set of vertices adjacent to vertices \( i \)
Outline

1. Building a Laplacian Operator on meshes
2. Surface Parameterization
3. Applications of the Laplacian on meshes
4. Manifold Harmonics Basis
5. Applications of the MHT
6. Functional Maps
7. A better basis?
Why parameterize a surface?
A surface parameterization is a function mapping a subset of the 2D domain to the surface. This mapping is a one to one correspondence.

Only for surfaces defined as meshes \((v_1, \cdots, v_n)\) vertices of the mesh.

\((u, v)\) in the parameter space corresponds to a surface point:
\[
x(u, v) = \alpha_i v_i + \alpha_j v_j + \alpha_k v_k,
\]
with \((v_i, v_j, v_k)\) the triangle containing \(x(u, v)\).
Difficulty

The mapping of the surface onto the 2D plane involves some deformation (e.g. well known Mercator projection of the 3D globe). Therefore the parameterization methods will strive to reduce the deformation with respect to particular quantities: areas, edge lengths or angle for example.
Barycentric Mapping

Valid Parameterization [Tutte 60]

Given a triangulated surface homeomorphic to a disk, if the \((u, v)\) coordinates at the boundary vertices lie on a convex polygon, and if the coordinates of the internal vertices are a convex combination of their neighbors, then the \((u, v)\) coordinates form a valid parameterization (without self-intersections)
Barycentric Mapping

Valid Parameterization [Tutte 60]

Assume the surface is homeomorphic to a disc. Interior vertices have indices $1, \ldots, n_{int}$ and boundary vertices $n_{int} + 1, \ldots, n$, the coordinates $(u_i, v_i)$ corresponding to vertex $v_i$ are given by:

$$\forall i \in 1, \ldots, n_{int} \quad a_{ii} \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \sum_{j \neq i} a_{ij} \begin{pmatrix} u_j \\ v_j \end{pmatrix}$$

where $a_{ij}$ are such that: $\forall i \in 1, \ldots, n$

$$\begin{cases} a_{ij} > 0 & \text{if } v_i, v_j \text{ are linked by an edge} \\ a_{ii} = -\sum_{j \neq i} a_{ij} \\ a_{ij} = 0 & \text{otherwise.} \end{cases}$$
Turn this definition into a construction algorithm

- Start by fixing the boundary vertices images on a planar convex polygon
Turn this definition into a construction algorithm

- Start by fixing the boundary vertices images on a planar convex polygon
- Solve for the inner vertices coordinates by noticing that:

\[
\forall i \in 1, \cdots, n_{\text{int}}, \quad \sum_{j=1}^{n_{\text{int}}} a_{ij} u_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} u_j = \bar{u}_i
\]

\[
\forall i \in 1, \cdots, n_{\text{int}}, \quad \sum_{j=1}^{n_{\text{int}}} a_{ij} v_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} v_j = \bar{v}_j
\]
Turn this definition into a construction algorithm

- Start by fixing the boundary vertices images on a planar convex polygon
- Solve for the inner vertices coordinates by noticing that:

\[
\forall i \in 1, \cdots, n_{\text{int}}, \sum_{j=1}^{n_{\text{int}}} a_{ij} u_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} u_j = \bar{u}_i
\]

\[
\forall i \in 1, \cdots, n_{\text{int}}, \sum_{j=1}^{n_{\text{int}}} a_{ij} v_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} v_j = \bar{v}_j
\]

- Solve \( Au = \bar{u} \) and \( Av = \bar{v} \): for small meshes, a Gauss-Seidel solver can be used, otherwise sparse matrix methods can be efficient.
Turn this definition into a construction algorithm

- Start by fixing the boundary vertices images on a planar convex polygon
- Solve for the inner vertices coordinates by noticing that:

  \[ \forall i \in 1, \cdots, n_{\text{int}}, \sum_{j=1}^{n_{\text{int}}} a_{ij} u_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} u_j = \bar{u}_i \]

  \[ \forall i \in 1, \cdots, n_{\text{int}}, \sum_{j=1}^{n_{\text{int}}} a_{ij} v_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} v_j = \bar{v}_j \]

- Solve \( Au = \bar{u} \) and \( Av = \bar{v} \): for small meshes, a Gauss-Seidel solver can be used, otherwise sparse matrix methods can be efficient.

- Gauss Seidel: Move each point \((u_i, v_i)\) to the barycenter of its neighborhood

  \[ u^{t+1}_i \leftarrow \frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} u^t_j \]

  \[ v^{t+1}_j \leftarrow \frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} v^t_j \]
Tutte-Floater Parameterization Result
What are the $a_{ij}$?

- **Naive choice**
  \[
  \begin{aligned}
  a_{i,i} &= -|N(i)| \\
  a_{ij} &= 1 \text{ if } v_i \text{ and } v_j \text{ are linked by an edge} \\
  a_{ij} &= 0 \text{ otherwise}.
  \end{aligned}
  \]
What are the $a_{ij}$?

**Naive choice**

$$
\begin{cases}
  a_{i,i} & = -|\mathcal{N}(i)| \\
  a_{ij} & = 1 \text{ if } v_i \text{ and } v_j \text{ are linked by an edge} \\
  a_{ij} & = 0 \text{ otherwise.}
\end{cases}
$$

- We'll see that we can do better
• How do we fix the boundary on a convex polygon?
• The shape of the boundary is sometimes far from convex!

Image from Karni et al. 2005
Through a conformal map, the curves iso-u and iso-v on the surface must be orthogonal (ie their tangent vectors are orthogonal)

The two gradient vectors are thus orthogonal at all points of the surface.

$$\mathbf{x}_v = \mathbf{n} \times \mathbf{x}_u$$
Least Squares Conformal Maps [Lévy et al. 2002]

- If $\chi : (u, v) \rightarrow (x, y)$ is a conformal mapping:
  \[ N(u, v) \times \frac{\partial \chi}{\partial u} = \frac{\partial \chi}{\partial v} \]

- In the local frame of the triangle, the conformality condition becomes:
  \[ \frac{\partial \chi}{\partial u} - i \frac{\partial \chi}{\partial v} = 0 \]

- Equivalently, the inverse map $U(x, y) = u + iv$ of $\chi$ satisfies:
  \[ \frac{\partial U}{\partial x} + i \frac{\partial U}{\partial y} = 0. \]

- By Riemann’s theorem there should exist such a mapping.
Constraining the parameterization

**Constraints**

- The edges of the triangulation should be mapped to straight lines.
- The mapping should vary linearly in each triangle.
New expression for the parameterization

Finding $U$

Minimize for a triangle:

$$C(T) = \int_T \left| \frac{\partial U}{\partial x} + i \frac{\partial U}{\partial y} \right| dA = A_T \left| \frac{\partial U}{\partial x} + i \frac{\partial U}{\partial y} \right|$$

The minimization over the whole triangulation becomes:

$$C(T) = \sum_{T \in \mathcal{T}} C(T)$$

which can be rewritten as

$$C(x) = \|Ax - b\|^2.$$
Results

- Iso-lines are stable with respect to mesh refinement.
- Combined with a segmentation of the mesh into charts.

From [Lévy et al. 2002]
A practical method: Angle Based Flattening (ABF) [Sheffer, de Sturler, 2001]

- A conformal map preserves angles.
- Penalize parameterizations that do not preserve angle

**Angle-Based Flattening Energy**

\[ E_{ABF}(\alpha) = \sum_{T \in \mathcal{T}} \sum_{k=1}^{3} \left( \frac{\alpha^T_k - \beta^T_k}{\beta^T_k} \right)^2 \]

\(\alpha^T_k\) is the angle of triangle \(T\) incident to vertex \(k\) in the parameterization domain, and \(\beta^T_k\) is the same angle measured on the mesh.
Ensuring that we have a valid parameterization

- Valid triangles: \( \forall T, \alpha_1^T + \alpha_2^T + \alpha_3^T = \pi \)
- Planar star of a vertex: \( \forall v \in V_{\text{int}}, \sum_{T \text{ incident to } v} \alpha_i^T = 2\pi \)
- Coherence of edge length
  \( \forall v \in V_{\text{int}}, \prod_{T \text{ incident to } v} \sin \alpha_1^T = \prod_{T \text{ incident to } v} \sin \alpha_2^T \)

Edge length incoherence in the star of a vertex. [Image from Sheffer et al. 2001]
How do we minimize this energy?

- Compute a stationary point of the Lagrangian of the constrained quadratic minimization problem using Newton’s method. [Sheffer 2001]
- Linearized approximation [Zayer et al. 2007]
Comparison between LSCM and ABF

Left: LSCM, right: ABF
Comparison between LSCM and ABF

Left: LSCM, right: ABF
When processing a function on a manifold...

- In general, parameterizing a manifold on a plane cannot be done without distortion.
- Much better to analyze the function directly on the manifold mesh.

Need for differential analysis tools

- Gradient of the function on the surface, Laplacian of the function on the surface.
Building a Laplacian on a mesh

- Mesh triangles are not regular in general
Building a Laplacian on a mesh

- Mesh triangles are not regular in general
  - Triangle edges DO NOT have constant length
Building a Laplacian on a mesh

- Mesh triangles are not regular in general
  - Triangle edges DO NOT have constant length
  - Triangle angles ARE NOT constant
Building a Laplacian on a mesh

- Mesh triangles are not regular in general
  - Triangle edges DO NOT have constant length
  - Triangle angles ARE NOT constant
- Yet we need to account for the function variations on the surface
Building a Laplacian on a mesh

- Mesh triangles are not regular in general
  - Triangle edges DO NOT have constant length
  - Triangle angles ARE NOT constant
- Yet we need to account for the function variations on the surface

Mesh Laplacian

There exist many different Laplacians. We follow the terminology of [Zhang et al. 2007] and [Vallet and Levy 2008]
A combinatorial Laplacian depends solely on the connectivity of the mesh.
Combinatorial Laplacian

A combinatorial Laplacian depends solely on the connectivity of the mesh.

- The Laplacian is computed independently of its geometrical embedding.
Combinatorial Laplacians

**Definition**

Given a triangular manifold mesh with \( N \) vertices \((v_i)_{i=1\ldots N}\), let \( E \) be the set of edges. The uniform Laplacian, *umbrella operator* is defined as a matrix \( L \) such that:

\[
L_{i,j} = \begin{cases} 
1, & \text{if } (v_i, v_j) \in E \\
0 & \text{otherwise}
\end{cases}
\]
Combinatorial Laplacians

Definition

Given a triangular manifold mesh with $N$ vertices $(v_i)_{i=1\ldots N}$, let $E$ be the set of edges. The uniform Laplacian, *umbrella operator* is defined as a matrix $L$ such that:

$$L_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases}$$

- Directly derived from the graph Laplacian.
Combinatorial Laplacians

- **Tutte Laplacian**

\[
L_{ij} = \begin{cases} 
\frac{1}{d_i} & \text{if } (i, j) \in E \\
0 & \text{otherwise} 
\end{cases}
\]
Combinatorial Laplacians

- **Tutte Laplacian**

\[
L_{ij} = \begin{cases} 
\frac{1}{d_i} & \text{if } (i, j) \in E \\
0 & \text{otherwise}
\end{cases}
\]

- **Normalized Graph Laplacian**

\[
L_{ij} = \begin{cases} 
\frac{1}{\sqrt{d_i d_j}} & \text{if } (i, j) \in E \\
0 & \text{otherwise}
\end{cases}
\]
Combinatorial Laplacians

- **Tutte Laplacian**
  \[
  L_{ij} = \begin{cases} 
  \frac{1}{d_i} & \text{if } (i, j) \in E \\
  0 & \text{otherwise}
  \end{cases}
  \]

- **Normalized Graph Laplacian**
  \[
  L_{ij} = \begin{cases} 
  \frac{1}{\sqrt{d_i d_j}} & \text{if } (i, j) \in E \\
  0 & \text{otherwise}
  \end{cases}
  \]

- **Other Discretizations:** Mean Value Coordinates, Wachspress coordinates....
Combinatorial Laplacian

- $\phi$ a real-valued function defined on a surface $S$ (surface mesh $(V, E, F)$)
- Let $f$ be the restriction of $\phi$ to the vertices $V$.
- $Q$ oriented incidence matrix corresponding to $(V, E, F)$, $Q \in \mathbb{R}^{|E| \times |V|}$
- Notation: $e_{ij}$ index of the edge from $i$ to $j$.

Gradient of $f$ along edges

$$Q_{e_{ij}} = \begin{cases} 
1 & \text{if there is an edge from } i \text{ to } j \\
-1 & \text{if there is an edge from } j \text{ to } i \\
0 & \text{otherwise}
\end{cases}$$

$$(Q^T f)_{e_{ij}} = \frac{f_i - f_j}{\rho(i, j)}$$

where $\rho(i, j)$ is the length of edge $(i, j)$ for example

Then $(Q^T f)_{e_{ij}}$ approximates the gradient of $f$ along edge $i, j$
Combinatorial Laplacian

**Relation with the Laplace Beltrami**

Left-multiplication by $Q^T$ approximates the gradient operator. Then $(Q \cdot Q^T)$ approximates the Laplacian operator on $S$. 
Combinatorial Laplacian

Relation with the Laplace Beltrami

Left-multiplication by $Q^T$ approximates the gradient operator. Then $(Q \cdot Q^T)$ approximates the Laplacian operator on $S$. As the triangulation of the mesh becomes sufficiently dense, the restriction of $\Delta \phi$ to $V$ is expected to be close to $(QQ^T)f$
Processing with Combinatorial Laplacians


Compression using Normalized Graph Laplacian
Geometric Laplacian

Pinkall Polthier 93

\[(Pf)_i = \sum_{j \in N_i} \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij})(f_i - f_j)\]
Geometric Laplacian

- Let \((\phi_j)_{j=1}^N\) be a set of continuous piecewise linear functions on the surface:

\[
\phi_j(v_i) = \delta_{ij}
\]
Geometric Laplacian

- Let \((\phi_j)_{j=1}^N\) be a set of continuous piecewise linear functions on the surface:

\[
\phi_j(v_i) = \delta_{ij}
\]

- Let \(u\) be a function defined on the surface \(u = \sum_j x_j \phi_j\).
Geometric Laplacian

- Let \((\phi_j)_{j=1}^N\) be a set of continuous piecewise linear functions on the surface:
  \[ \phi_j(v_i) = \delta_{ij} \]

- Let \(u\) be a function defined on the surface \(u = \sum_j x_j \phi_j\). We need to compute, for all \(j\), \(\langle \Delta u, \phi_j \rangle\).
Geometric Laplacian

- Let \((\phi_j)_{j=1}^N\) be a set of continuous piecewise linear functions on the surface:

\[ \phi_j(v_i) = \delta_{ij} \]

- Let \(u\) be a function defined on the surface \(u = \sum_j x_j \phi_j\). We need to compute, for all \(j\), \(\langle \Delta u, \phi_j \rangle\).

- Green’s formula:

\[
\langle \Delta u, \phi_j \rangle = \sum_{t \text{ triangle}} \langle \nabla u, \nabla \phi_j \rangle_t + \sum_{t \text{ triangle}} \langle \mathbf{N} u, \phi_j \rangle_t
\]
Geometric Laplacian

Let \((\phi_j)_{j=1}^N\) be a set of continuous piecewise linear functions on the surface:

\[
\phi_j(v_i) = \delta_{ij}
\]

Let \(u\) be a function defined on the surface \(u = \sum_j x_j \phi_j\). We need to compute, for all \(j\), \(\langle \Delta u, \phi_j \rangle\).

Green’s formula:

\[
\langle \Delta u, \phi_j \rangle = \sum_{t \text{ triangle}} \langle \nabla u, \nabla \phi_j \rangle_t + \sum_{t \text{ triangle}} \langle N u, \phi_j \rangle_t
\]

On a closed surface: \(\langle \Delta u, \phi_j \rangle = \sum_{t \text{ triangle}} \langle \nabla u, \nabla \phi_j \rangle_t\)
Geometric Laplacian

- Let $\phi_j^{N\ j=1}$ be a set of continuous piecewise linear functions on the surface:
  \[ \phi_j(v_i) = \delta_{ij} \]

- Let $u$ be a function defined on the surface $u = \sum_j x_j \phi_j$. We need to compute, for all $j$, $\langle \Delta u, \phi_j \rangle$.

- Green’s formula:
  \[
  \langle \Delta u, \phi_j \rangle = \sum_{t \text{ triangle}} \langle \nabla u, \nabla \phi_j \rangle_t + \sum_{t \text{ triangle}} \langle Nu, \phi_j \rangle_t
  \]

- On a closed surface: $\langle \Delta u, \phi_j \rangle = \sum_{t \text{ triangle}} \langle \nabla u, \nabla \phi_j \rangle_t$

And finally...

Since $u = \sum_i x_i \phi_i$, $\langle \Delta u, \phi_j \rangle = \sum_{t \text{ triangle}} \sum_i x_i \langle \nabla \phi_i, \nabla \phi_i \rangle_t$, we need to compute:
\[
\langle \nabla \phi_i, \nabla \phi_j \rangle_t
\]
for all $i, j$ and triangle $t$
Building the Laplacian

What is the Gradient of $\phi_i$ in a triangle that contains vertex $v_i$ as a function of the area and the opposite edge.
What is the Gradient of $\phi_i$ in a triangle that contains vertex $v_i$ as a function of the area and the opposite edge.

Compute $\langle \nabla \phi_i, \nabla \phi_i \rangle_t$ with respect to $\alpha_i$ and $\beta_i$.
Building the Laplacian

- What is the Gradient of $\phi_i$ in a triangle that contains vertex $v_i$ as a function of the area and the opposite edge.
- Compute $\langle \nabla \phi_i, \nabla \phi_i \rangle_t$ with respect to $\alpha_i$ and $\beta_i$
- Compute $\langle \nabla \phi_i, \nabla \phi_j \rangle_t$ with respect to $\theta$ the angle opposite to vertices $i$ and $j$.

Alternative Construction

There also exists a beautiful construction of the geometric Laplacian based on Discrete Exterior Calculus and Hodge star.
Building the Laplacian

What is the Gradient of $\phi_i$ in a triangle that contains vertex $v_i$ as a function of the area and the opposite edge.

Compute $\langle \nabla \phi_i, \nabla \phi_i \rangle_t$ with respect to $\alpha_i$ and $\beta_i$.

Compute $\langle \nabla \phi_i, \nabla \phi_j \rangle_t$ with respect to $\theta$ the angle opposite to vertices $i$ and $j$.

Deduce the cotangent formula.
What is the Gradient of $\phi_i$ in a triangle that contains vertex $v_i$ as a function of the area and the opposite edge.

Compute $\langle \nabla \phi_i, \nabla \phi_i \rangle_t$ with respect to $\alpha_i$ and $\beta_i$

Compute $\langle \nabla \phi_i, \nabla \phi_j \rangle_t$ with respect to $\theta$ the angle opposite to vertices $i$ and $j$.

Deduce the cotangent formula

**Alternative Construction**

There also exists a beautiful construction of the geometric Laplacian based on *Discrete Exterior Calculus* and Hodge star.
Geometric Laplacian

Meyer et al. 2002

Make it pointwise by dividing by the area of the cell joining adjacent triangles barycenters.

\[(Pf)_i = \frac{1}{A_i} \sum_{j \in N_i} \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij})(f_i - f_j)\]
Meyer et al. 2002

Make it pointwise by dividing by the area of the cell joining adjacent triangles barycenters.

\[(Pf)_i = \frac{1}{A_i} \sum_{j \in N_i} \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij})(f_i - f_j)\]

- \(P\) is not symmetric!
**Geometric Laplacian**

**Meyer et al. 2002**

Make it pointwise by dividing by the area of the cell joining adjacent triangles barycenters.

\[(Pf)_i = \frac{1}{A_i} \sum_{j \in N_i} \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij})(f_i - f_j)\]

- \(P\) is not symmetric!
- Unless we adapt the scalar product \(<u, v>_B = u^T B v\).
Geometric vs Combinatorial Laplacian

Initial triangulation, smoothing with the uniform laplacian, smoothing using the cotan laplacian.
There exists no perfect Laplace-Beltrami

**Required property for a continuous Laplace Beltrami operator**

- **(NULL)** $\Delta u = 0$ if $u$ is constant

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a continuous Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

### Required property for a continuous Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$
- (LOC) $p \neq q \Rightarrow \Delta u(p)$ indep. of $u(q)$

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a continuous Laplace Beltrami operator:

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$
- (LOC) $p \neq q \Rightarrow \Delta u(p)$ indep. of $u(q)$
- (LIN) $\Delta u = 0$ if $S$ is part of an Euclidean plane and $u = ax + by + c$

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a continuous Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$
- (LOC) $p \neq q \Rightarrow \Delta u(p)$ indep. of $u(q)$
- (LIN) $\Delta u = 0$ if $S$ is part of an Euclidean plane and $u = ax + by + c$
- (MAX) Max-principle: harmonic functions have no local maxima/minima at interior points

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a continuous Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$
- (LOC) $p \neq q \Rightarrow \Delta u(p)$ indep. of $u(q)$
- (LIN) $\Delta u = 0$ if $S$ is part of an Euclidean plane and $u = ax + by + c$
- (MAX) Max-principle: harmonic functions have no local maxima/minima at interior points
- (PSD) $E_D(u) = \langle \Delta u, u \rangle_{L^2} \geq 0$ if $u$ smooth enough & vanishes on the boundaries

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a discrete Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a discrete Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a discrete Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$
- (LOC) If $p$ is not a neighbor of $q$ then $\Delta u(p)$ indep. of $u(q)$

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a discrete Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$
- (LOC) If $p$ is not a neighbor of $q$ then $\Delta u(p)$ indep. of $u(q)$
- (LIN) $\Delta u_p = 0$ if $S$ is locally a plane and $u$ is a linear function on the plane.

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Required property for a discrete Laplace Beltrami operator

- (NULL) $\Delta u = 0$ if $u$ is constant
- (SYM) $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$
- (LOC) If $p$ is not a neighbor of $q$ then $\Delta u(p)$ indep. of $u(q)$
- (LIN) $\Delta u_p = 0$ if $S$ is locally a plane and $u$ is a linear function on the plane.
- (POS) $w_{ij} \geq 0$ and there exists at least one vertex such that $w_{ij} > 0$ (discrete Maximum Principle).

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami operator.

**Required property for a discrete Laplace Beltrami operator**

- **(NULL)** $\Delta u = 0$ if $u$ is constant
- **(SYM)** $\langle \Delta u, v \rangle = \langle u, \Delta v \rangle$
- **(LOC)** If $p$ is not a neighbor of $q$ then $\Delta u(p)$ is independent of $u(q)$
- **(LIN)** $\Delta u_p = 0$ if $S$ is locally a plane and $u$ is a linear function on the plane.
- **(POS)** $w_{ij} \geq 0$ and there exists at least one vertex such that $w_{ij} > 0$ (discrete Maximum Principle).
- **(PSD)** $L$ is positive semi-definite with respect to the standard inner product.

[Wardetsky et al. 2007]
There exists no perfect Laplace-Beltrami

Theorem (Wardetsky et al. 2007)

NOT all meshes admit Laplacians satisfying (SYM)+(LOC)+(LIN)+(POS) simultaneously.
What conditions do these Laplacian discretization breaks?

- Umbrella operator $w_{ij} = 1$ if $(i, j)$ is an edge (0 otherwise)
What conditions do these Laplacian discretization breaks?

- Umbrella operator $w_{ij} = 1$ if $(i, j)$ is an edge (0 otherwise) $\rightarrow$ (LIN)
What conditions do these Laplacian discretization breaks?

- Umbrella operator $w_{ij} = 1$ if $(i, j)$ is an edge (0 otherwise) $\rightarrow$ (LIN)
- Tutte Laplacian $w_{ij} = \frac{1}{d_i}$ if $(i, j)$ is an edge (0 otherwise)
What conditions do these Laplacian discretization breaks?

- Umbrella operator $w_{ij} = 1$ if $(i, j)$ is an edge (0 otherwise) $\rightarrow$ (LIN)
- Tutte Laplacian $w_{ij} = \frac{1}{d_i}$ if $(i, j)$ is an edge (0 otherwise) $\rightarrow$ (LIN)
What conditions do these Laplacian discretization breaks?

- **Umbrella operator** $w_{ij} = 1$ if $(i, j)$ is an edge (0 otherwise) $\rightarrow$ (LIN)
- **Tutte Laplacian** $w_{ij} = \frac{1}{d_i}$ if $(i, j)$ is an edge (0 otherwise) $\rightarrow$ (LIN)
- **Cotangent Laplacian**
What conditions do these Laplacian discretization breaks?

- Umbrella operator $w_{ij} = 1$ if $(i, j)$ is an edge (0 otherwise) $\rightarrow$ (LIN)
- Tutte Laplacian $w_{ij} = \frac{1}{d_i}$ if $(i, j)$ is an edge (0 otherwise) $\rightarrow$ (LIN)
- Cotangent Laplacian $\rightarrow$ (POS)
Laplacian discretizations

<table>
<thead>
<tr>
<th>Mean Value</th>
<th>Loc</th>
<th>Lin</th>
<th>Pos</th>
<th>PSD</th>
<th>Con</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic Del</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>?</td>
</tr>
<tr>
<td>Combinatorial Cotan</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>
Proof from Wardetsky et al. 2007

- $w_{ij}$ are interpreted as pulling or pushing stress in a mechanical system.
Proof from Wardetsky et al. 2007

- $w_{ij}$ are interpreted as pulling or pushing stress in a mechanical system.
- The (LIN) $0 = \sum_j w_{ij}(x_i - x_j)$ assumption is equivalent to Euler-Lagrange equilibrium when boundary vertices are held fixed.
Proof from Wardetsky et al. 2007

- \( w_{ij} \) are interpreted as pulling or pushing stress in a mechanical system.
- The (LIN) \( 0 = \sum_j w_{ij} (x_i - x_j) \) assumption equivalent to Euler-Lagrange equilibrium when boundary vertices are held fixed.

**Theorem (Maxwell-Cremona)**

The system is in equilibrium iff there exists an orthogonal reciprocal dual framework.
Laplacian Comparisons

Combinatorial Laplacian, unweighted cotan, weighted cotan, two versions of the symmetrized weighted cotan
Some references

- Laplace-Beltrami eigenfunctions: Towards an algorithm that understands geometry, B. Lévy 2006.
- Botsch et al. 2010 Polygonal Mesh Processing
- Hormann; Lévy, Sheffer, 2007, Mesh Parameterization: Theory and Practice
Outline

1 Building a Laplacian Operator on meshes
2 Surface Parameterization
3 Applications of the Laplacian on meshes
4 Manifold Harmonics Basis
5 Applications of the MHT
6 Functional Maps
7 A better basis?
Laplacian Smoothing

- $x : \Omega \subset \mathbb{R}^2 \rightarrow S$, $\frac{\partial x}{\partial t} = \Delta x$
- Iterative smoothing: $x^{t+1} = x^t + h\Delta x^t$

Laplacian flow iterations: Filtering of the Bunny after several iterations. Image courtesy M. Botsch.
Variational formulation of the mesh fairing problem

- Model the noisy behavior of a surface as an energy: noisy surfaces are “expensive” while *fair* surfaces “are cheap”.
- What is a fair surface?
- Membrane energy: (minimize surface area)

\[
E_M(x) = \int \int_\Omega \left( \left\| \frac{\partial x}{\partial u} \right\|^2 + \left\| \frac{\partial x}{\partial v} \right\|^2 \right) dudv
\]

for a parameterized surface \( x : \Omega \rightarrow \mathbb{R}^3 \)

- Minimize \( E_M \) s.t. the vertices on the boundary \( \partial \Omega \) are fixed.
Calculus of Variations

- Similar to energy minimization on images.
- Compute \( \frac{\partial E_M(x + \varepsilon \nu)}{\partial \varepsilon} \bigg|_{\varepsilon=0} \)

\[
\frac{\partial E_M(x + \varepsilon \nu)}{\partial \varepsilon} \bigg|_{\varepsilon=0} = 2 \int_{\Omega} x_u \cdot \nu_u + x_v \cdot \nu_v \, dudv
\]

- Using the divergence theorem:

\[
\Delta x(u, v) = 0 \quad \forall (u, v) \in \Omega
\]
Membrane energy minimization

The minimization of the membrane energy on a discrete surface amounts to:

\[ \Delta x = 0 \iff Lx = 0 \text{ where } L \text{ is the Laplacian matrix} \]

- Sparse linear system solve to get the optimal positions.

Image from [Botsch et al. 2004]
Thin Plate Energy

Instead of minimizing the area, minimize the curvatures:

$$E_{TP}(\mathbf{x}) = \int_{\Omega} \kappa_1^2 + \kappa_2^2 \, dudv$$

This energy becomes:

$$E_{TP}(\mathbf{x}) = \int_{\Omega} \|x_{uu}\|^2 + \|x_{vv}\|^2 + 2\|x_{uv}\|^2 \, dudv$$

transforms into:

$$L^2 \mathbf{x} = 0$$
Spectral Compression [Karni-Gotsman 2000]

- Usual distortion measure: displacement of the vertices $\|v'_i - v_i\|_2^2$

$GL(v_i) = v_i - \sum_j (\ell - 1)_{ij} v_j \sum_j (\ell - 1)_{ij}$

where $\ell_{ij} = \|v_i - v_j\|_2^2$
Spectral Compression [Karni-Gotsman 2000]

- Usual distortion measure: displacement of the vertices $\|v'_i - v_i\|_2^2$
- Add a spectral measure: $\|v'_i - v_i\|_2^2 + \|GL(v'_i) - GL(v_i)\|_2^2$

$$GL(v_i) = v_i - \frac{\sum_j \ell_{ij}^{-1} v_j}{\sum_j \ell_{ij}^{-1}} \text{ where } \ell_{ij} = \|v_i - v_j\|_2$$
Spectral Compression [Karni-Gotsman 2000]

- Usual distortion measure: displacement of the vertices $\|v'_i - v_i\|^2$
- Add a spectral measure: $\|v'_i - v_i\|^2 + \|GL(v'_i) - GL(v_i)\|^2$

$$GL(v_i) = v_i - \frac{\sum_j \ell^{-1}_{ij} v_j}{\sum_j \ell^{-1}_{ij}} \text{ where } \ell_{ij} = \|v_i - v_j\|_2$$
Surface deformation formulation

- Model surface deformation $d$ with prescribed geometric constraints.
- Express the problem as a minimization of \textit{physical} energies
- Penalize stretching and bending of the surface.

Image from [Botsch et al. 2005]
Small Reminder

- **First Fundamental Form**

\[ I = \begin{pmatrix} x_u^T x_u & x_u^T x_v \\ x_u^T x_v & x_v^T x_v \end{pmatrix} \]

- **Second Fundamental Form**

\[ II = \begin{pmatrix} x_{uu}^T \cdot n & x_{uv}^T \cdot n \\ x_{uv}^T \cdot n & x_{vv}^T \cdot n \end{pmatrix} \]
Control bending and stretching

- Penalize a change of $l$ as a change of the lengths: stretching penalization
- Penalize a change in $\|\|$ as a bending change: bending penalization
Energy formulation

Energy Formulation

$S$ a surface deformed into $S'$. Both are parameterized over $\Omega$.

$$E(S') = \int_{\Omega} k_s \| I'(u, v) - I(u, v) \| dudv + \int_{\Omega} k_b \| II'(u, v) - II(u, v) \| dudv$$

$k_s$ and $k_b$ control the penalization on stretching and blending.

Very expensive!
Simplication

Energy Formulation

d: displacement field

\[ E(d) = \int_{\Omega} k_s (\| d_u(u,v) \|^2 + \| d_v(u,v) \|^2) dudv \]

\[ + \int_{\Omega} k_b (\| d_{uu}(u,v) \|^2 + \| d_{uv}(u,v) \|^2 + \| d_{vv}(u,v) \|^2) dudv \]

\( k_s \) and \( k_b \) control the penalization on stretching and blending.
\textit{Euler-Lagrange} equation for the minimizer:

\[-k_S \Delta d + k_b \Delta^2 d = 0\]

$\Delta^2$ is the bi-Laplacian: Laplacian of Laplacians

Bi-Laplacian:

\[
\Delta^2 f_i = \frac{1}{2A_i} \sum_{v_j \in N_1(v_i)} (\cotan \alpha_{ij} + \cotan \beta_{ij})(\Delta f_j - \Delta f_i)
\]
Discretization with constraints

\[
\begin{cases}
-k_S \Delta d_i + k_b \Delta^2 d_i &= 0 \text{ if } p_i \text{ is a free vertex} \\
\bar{d}_i &= d_i \text{ if } p_i \text{ is a handle} \\
\bar{d}_i &= 0 \text{ if } p_i \text{ is fixed}
\end{cases}
\]

This rewrites:

\[
\begin{pmatrix}
-k_s L + k_b L^2
\end{pmatrix}
\begin{pmatrix}
d_1 \\
d_2 \\
\vdots \\
d_n
\end{pmatrix}
= 
\begin{pmatrix}
b_1 \\
b_2 \\
\vdots \\
b_n
\end{pmatrix}
\]

\(b_1, \cdots, b_n\) corresponds to the prescribed motions for the fixed vertices and moving handles.
Handle moves $\Rightarrow$ right-hand side changes

- Solve the linear system at each frame!
- Efficient linear solver: Sparse Cholesky, matrix factorization reusable though the right hand side changes.
- Trick: pre-compute basis functions for the deformation: at each frame evaluate the basis functions [Kobbelt et al. 2004]
Advantages/Drawbacks

- Intuitive deformation since based on physical properties
- Computationally expensive
- Linearization artefacts: fine-scale surface details are not well handled.
Initial: $k_s = 1/k_b = 0$; $k_s = 0/k_b = 1$; $k_s = 1/k_b = 10$. Image from [Botsch and Sorkine 2008]
Deformation with thin shell. Image from [Botsch and Sorkine 2008]
How can we deform a mesh while preserving the details and geometric consistency?
Laplacian Editing

How can we deform a mesh while preserving the details and geometric consistency?

**Principle**

Describe the mesh using a set of differentials, fix it for some vertices and solve for the rest of the vertices.

[Alexa 2003], [Sorkine et al. 2005]
Laplacian Editing

- Vertex $v_i$ is not represented by its absolute coordinate but by the difference to its neighbors:
Vertex $v_i$ is not represented by its absolute coordinate but by the difference to its neighbors: $\delta_i = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j$.
Vertex $v_i$ is not represented by its absolute coordinate but by the difference to its neighbors: $\delta_i = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j$. Exactly the combinatorial Laplacian.
Laplacian Editing

- Vertex $v_i$ is not represented by its absolute coordinate but by the difference to its neighbors: $\delta_i = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j$ Exactly the combinatorial Laplacian
- Fix the position of some vertices and Solve for the rest by fitting the Laplacian Coordinates to the given coordinates.
Laplacian Editing

- Vertex \( v_i \) is not represented by its absolute coordinate but by the difference to its neighbors: \( \delta_i = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j \). Exactly the combinatorial Laplacian.
- Fix the position of some vertices and Solve for the rest by fitting the Laplacian Coordinates to the given coordinates.

**Exact fitting**

\( v_i, \ i = 1 \cdots m \) vertices with unknown position, \( v_i = u_i \) vertices with known position \( i = m + 1 \cdots n \). Solve for:

\[
\begin{align*}
Lv'_i &= \delta_i \text{ for } i = 1 \cdots m \\
v'_i &= u_i \text{ for } i = m + 1 \cdots n
\end{align*}
\]
Small introduction to Cholesky Factorization

**Cholesky Factorization**

$L$ a symmetric positive definite matrix. There exists a lower-triangular matrix $M$ such that:

$$L = M \cdot M^T$$
Small introduction to Cholesky Factorization

Cholesky Factorization

$L$ a symmetric positive definite matrix. There exists a lower-triangular matrix $M$ such that:

$$L = M \cdot M^T$$

- $l_{ij} = (MM^T)_{ij} = \sum_{k=1}^{\min(i,j)} m_{ij} m_{ik}$
Small introduction to Cholesky Factorization

**Cholesky Factorization**

$L$ a symmetric positive definite matrix. There exists a lower-triangular matrix $M$ such that:

$$L = M \cdot M^T$$

- $l_{ij} = (MM^T)_{ij} = \sum_{k=1}^{\min(i,j)} m_{ij} m_{ik}$
- $\forall i \leq j \quad l_{ij} = \sum_{k=1}^{i} m_{ik} m_{jk}$
Small introduction to Cholesky Factorization

Cholesky Factorization

$L$ a symmetric positive definite matrix. There exists a lower-triangular matrix $M$ such that:

$$L = M \cdot M^T$$

- $l_{ij} = (MM^T)_{ij} = \sum_{k=1}^{\min(i,j)} m_{ij} m_{ik}$
- $\forall i \leq j \ l_{ij} = \sum_{k=1}^{i} m_{ik} m_{jk}$

Algorithm

1. $i = 1$: $m_{11} = \sqrt{l_{11}}$
2. $l_{1j} = m_{11} m_{j1}$ so that $m_{j1} = \frac{l_{j1}}{m_{11}}$
3. For $i \geq 2$:

   $$l_{ii} = \sum_{k=1}^{i} m_{ik}^2 \Rightarrow m_{ii}^2 = l_{ii} - \sum_{k=1}^{i-1} m_{ik}^2$$

   $$l_{ij} = \sum_{k=1}^{i} m_{ik} m_{jk} \Rightarrow m_{ji} = \frac{l_{ij} - \sum_{k=1}^{i-1} m_{ik} m_{jk}}{m_{ii}}$$
Small introduction to Cholesky Factorization

Better

\[ L = MDM^T \] where \( M \) is lower triangular with 1 on the diagonal and \( D \) is a diagonal matrix.

\[
D_{jj} = L_{jj} - \sum_{k=1}^{j-1} M_{jk}^2 D_{kk}
\]

\[
M_{ij} = \frac{1}{D_{jj}} (L_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} D_{kk}) \text{ for } i > j
\]
Small introduction to Cholesky Factorization

Better

\[ L = M D M^T \] where \( M \) is lower triangular with 1 on the diagonal and \( D \) is a diagonal matrix.

\[
D_{jj} = L_{jj} - \sum_{k=1}^{j-1} M_{jk}^2 D_{kk}
\]

\[
M_{ij} = \frac{1}{D_{jj}} (L_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} D_{kk}) \text{ for } i > j
\]

- avoids square roots
L = MDM^T where M is lower triangular with 1 on the diagonal and D is a diagonal matrix.

\[ D_{jj} = L_{jj} - \sum_{k=1}^{j-1} M_{jk}^2 D_{kk} \]

\[ M_{ij} = \frac{1}{D_{jj}} (L_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} D_{kk}) \text{ for } i > j \]

- avoids square roots
- Both decompositions can be easily linked.
Sparse Cholesky decomposition

**Principle**

Reorder the sparse matrix by permuting columns and lines first and then solve.

- Compute a permutation matrix $P$
Sparse Cholesky decomposition

**Principle**
Reorder the sparse matrix by permuting columns and lines first and then solve.

- Compute a permutation matrix $P$
- Solve $P^TLP = MM^T$ exploiting $P^TLP$ structure
Sparse Cholesky Solver

- Once the Cholesky decomposition is obtained, the solve is trivial:
  \[
  \begin{cases}
  My = b \\
  M^T x = y
  \end{cases}
  \]

- The *bandwidth* of matrix $L$ is preserved through the decomposition $\beta(M) \leq \beta(L)$

- Complexity reduction from $O(n^3)$ to $O(n^2 \beta)$ for the factorization.
- Complexity reduction from $O(n^2)$ to $O(n \beta)$ for the final solve.
Least Squares Laplacian Editing

- Solving *exactly* the editing problem gives unstable solutions $\Rightarrow$ Look for a least squares solution instead
Least Squares Laplacian Editing

- Solving \textit{exactly} the editing problem gives unstable solutions $\Rightarrow$ Look for a least squares solution instead

Least Squares Fitting

$$E(V') = \sum_{i=1}^{m} \| \delta_i - L v'_i \|^2 + \sum_{i=m+1}^{n} \| v'_i - u_i \|$$

yields a sparse linear system.
Least Squares Laplacian Editing

- Solving *exactly* the editing problem gives unstable solutions ⇒ Look for a least squares solution instead

### Least Squares Fitting

\[
E(V') = \sum_{i=1}^{m} \| \delta_i - L v'_i \|^2 + \sum_{i=m+1}^{n} \| v'_i - u_i \|
\]

yields a sparse linear system.

- This formulation is not robust to rotation or scale. The details structure can only be translated.
Rotation and Scale Invariant Laplacian Coordinates

- Compute a per-vertex transformation.
Rotation and Scale Invariant Laplacian Coordinates

Compute a per-vertex transformation. Render the LC almost invariant to rotation and translation
Rotation and Scale Invariant Laplacian Coordinates

- Compute a per-vertex transformation. Render the LC almost invariant to rotation and translation

**Laplacian Editing Energy**

\[
E(V') = \sum_{i=1}^{m} \| T_i(V')\delta_i - L v'_i \|^2 + \sum_{i=m+1}^{n} \| v'_i - u_i \|^2
\]

and solve for \(V'\) and \(T_i\).
Rotation and Scale Invariant Laplacian Coordinates

Assume $T_i$ is linear and solve for $T_i$ with respect to the neighborhood.

If $T_i$ unconstrained, membrane energy hence loss of details.

Applications of the Laplacian on meshes
Assume $T_i$ is linear and solve for $T_i$ with respect to the neighborhood.

**Laplacian Editing Energy**

$$\min_{T_i} \| T_i v_i - v_i' \|^2 + \sum_{j \in N(i)} \| T_i v_j - v_j' \|^2$$

If $T_i$ unconstrained, membrane energy hence loss of details.

Applications of the Laplacian on meshes
Assume $T_i$ is linear and solve for $T_i$ with respect to the neighborhood.

Laplacian Editing Energy

$$\min_{T_i} \| T_i v_i - v_i' \|^2 + \sum_{j \in N(i)} \| T_i v_j - v_j' \|^2$$

If $T_i$ unconstrained, membrane energy hence loss of details.
Assume \( T_i \) is linear and solve for \( T_i \) with respect to the neighborhood.

Laplacian Editing Energy

\[
\min_{T_i} \| T_i v_i - v_i' \|^2 + \sum_{j \in N(i)} \| T_i v_j - v_j' \|^2
\]

If \( T_i \) unconstrained, membrane energy hence loss of details.

Constraints on \( T_i \): translation, rotation and isotropic scaling
Solving for $T_i$

- Solving for the translation: homogeneous coordinates
Solving for $T_i$

- Solving for the translation: homogeneous coordinates
- Rotation Matrix: Can be written as $\exp(H)$ with $H$ a skew-symmetric matrix.
Solving for $T_i$

- Solving for the translation: homogeneous coordinates
- Rotation Matrix: Can be written as $\exp(H)$ with $H$ a skew-symmetric matrix.
- Rotation + Isotropic Scale Matrix: $s \exp(H)$
Solving for $T_i$

- Solving for the translation: homogeneous coordinates
- Rotation Matrix: Can be written as $\exp(H)$ with $H$ a skew-symmetric matrix.
- Rotation + Isotropic Scale Matrix: $s \exp(H)$

Small Reminder

For each skew-symmetric matrix $H$, there exists a vector $h$ such that $Hx = h \times x$ (cross product)
Exercise
Linearize $s \exp(H)$
Exercise

Linearize $s \exp(H)$

$$\exp H = I + H + \frac{1}{2} H^2 + \frac{1}{3!} H^2 + \cdots$$
Exercise

Linearize $s \exp(H)$

- $\exp H = I + H + \frac{1}{2} H^2 + \frac{1}{3!} H^2 + \cdots$
- $H^2 = hh^T - h^T h$
Linearization

Exercise

Linearize $s \exp(H)$

- $\exp H = I + H + \frac{1}{2} H^2 + \frac{1}{3!} H^2 + \cdots$
- $H^2 = hh^T - h^T h I$
- Deduce $H^{2n}$ and $H^{2n-1}$ for $n \geq 1$

\[
\begin{align*}
H^{2n} &= (-h^T h)^{n-1} hh^T + (-h^T h)^n I \\
H^{2n-1} &= (-h^T h)^{n-1} H
\end{align*}
\]
Linearization

Exercise

Linearize $s \exp(H)$

- $\exp H = I + H + \frac{1}{2} H^2 + \frac{1}{3!} H^2 + \cdots$
- $H^2 = hh^T - h^T h I$
- Deduce $H^{2n}$ and $H^{2n-1}$ for $n \geq 1$

$$
\begin{align*}
H^{2n} &= (-h^T h)^n I hh^T + (-h^T h)^{n-1} I \\
H^{2n-1} &= (-h^T h)^{n-1} H
\end{align*}
$$

- $s \exp H = s(\alpha I + \beta H + \gamma hh^T)$
Linearization

Transformation

Ignoring the quadratic part yields

\[ T_i = \begin{pmatrix} s & -h_3 & h_2 & t_x \\ h_3 & s & -h_1 & t_y \\ -h_2 & h_1 & s & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \]

(valid for small angles)
Ignoring the quadratic part yields

\[
T_i = \begin{pmatrix}
  s & -h_3 & h_2 & t_x \\
  h_3 & s & -h_1 & t_y \\
  -h_2 & h_1 & s & t_z \\
  0 & 0 & 0 & 1
\end{pmatrix}
\]

(valid for small angles)

Solve for \( T_i \) by minimizing \( \| A_i \begin{pmatrix} s_i & h_i & t_i \end{pmatrix}^T - b_i \|^2 \) where:

\[ A_i = \begin{pmatrix}
  v_{k_i} & 0 & v_{k_i} & -v_{k_i} & 1 & 0 & 0 \\
  v_{k_i} & -v_{k_i} & 0 & v_{k_i} & 0 & 1 & 0 \\
  v_{k_i} & v_{k_i} & -v_{k_i} & 0 & 0 & 0 & 1 \\
  \vdots
\end{pmatrix}, \ k \in \{i\} \cup \mathcal{N}_i \\
\]

\[ b_i = \begin{pmatrix}
  v_{k_i} \\
  v_{k_i} \\
  v_{k_i} \\
  \vdots
\end{pmatrix}, \ k \in \{i\} \cup \mathcal{N}_i. \]

Can you write the solution?
Results: deformation

Image from Sorkine et al.
Results: detail transfer

(a) (b) (c) (d)

Image from Sorkine et al.
Results: Shape Editing in 2D

Image from Sorkine et al.
Surface Parameterization

Recall the parameterization algorithm

- Start by fixing the boundary vertices images on a planar convex polygon
Surface Parameterization

Recall the parameterization algorithm

- Start by fixing the boundary vertices images on a planar convex polygon
- Solve for the inner vertices coordinates by noticing that:

\[ \forall i \in 1, \cdots, n_{\text{int}}, \sum_{j=1}^{n_{\text{int}}} a_{ij} u_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} u_j = \bar{u}_i \]

\[ \forall i \in 1, \cdots, n_{\text{int}}, \sum_{j=1}^{n_{\text{int}}} a_{ij} v_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} v_j = \bar{v}_j \]
Surface Parameterization

Recall the parameterization algorithm

- Start by fixing the boundary vertices images on a planar convex polygon
- Solve for the inner vertices coordinates by noticing that:

\[ \forall i \in 1, \cdots, n_{int}, \sum_{j=1}^{n_{int}} a_{ij} u_j = - \sum_{j=n_{int}+1}^{n} a_{ij} u_j = \bar{u}_i \]

\[ \forall i \in 1, \cdots, n_{int}, \sum_{j=1}^{n_{int}} a_{ij} v_j = - \sum_{j=n_{int}+1}^{n} a_{ij} v_j = \bar{v}_j \]

- Solve \( Au = \bar{u} \) and \( Av = \bar{v} \): for small meshes, a Gauss-Seidel solver can be used, otherwise sparse matrix methods can be efficient.
Surface Parameterization

Recall the parameterization algorithm

- Start by fixing the boundary vertices images on a planar convex polygon
- Solve for the inner vertices coordinates by noticing that:

\[ \forall i \in 1, \cdots, n_{\text{int}}, \sum_{j=1}^{n} a_{ij} u_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} u_j = \bar{u}_i \]

\[ \forall i \in 1, \cdots, n_{\text{int}}, \sum_{j=1}^{n} a_{ij} v_j = - \sum_{j=n_{\text{int}}+1}^{n} a_{ij} v_j = \bar{v}_j \]

- Solve \( Au = \bar{u} \) and \( Av = \bar{v} \): for small meshes, a Gauss-Seidel solver can be used, otherwise sparse matrix methods can be efficient.
- Gauss Seidel: Move each point \((u_i, v_i)\) to the barycenter of its neighborhood

\[ u_{i}^{t+1} \leftarrow \frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} u_{j}^{t} \]

\[ v_{j}^{t+1} \leftarrow \frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} v_{j}^{t} \]
What are the $a_{ij}$?

- **Naive choice**

$$
\begin{align*}
  a_{i,i} &= -|N(i)| \\
  a_{ij} &= 1 \text{ if } v_i \text{ and } v_j \text{ are linked by an edge} \\
  a_{ij} &= 0 \text{ otherwise.}
\end{align*}
$$
What are the $a_{ij}$?

**Naive choice**

\[
\begin{aligned}
    a_{i,i} &= -|\mathcal{N}(i)| \\
    a_{ij} &= 1 \text{ if } v_i \text{ and } v_j \text{ are linked by an edge} \\
    a_{ij} &= 0 \text{ otherwise.}
\end{aligned}
\]

**Better: Laplace Beltrami Operator**

\[
\begin{aligned}
    a_{ij} &= \frac{1}{2A_{v_i}} (\cotan\alpha_{ij} + \cotan\beta_{ij}) \text{ if } v_i \text{ and } v_j \text{ are linked by an edge} \\
    a_{ii} &= -\sum_{i \neq j} a_{ij} \\
    a_{ij} &= 0 \text{ otherwise}
\end{aligned}
\]
Comparison

original mesh  uniform weights  cotan weights (shape preserving)  mean value
Outline

1 Building a Laplacian Operator on meshes
2 Surface Parameterization
3 Applications of the Laplacian on meshes
4 Manifold Harmonics Basis
5 Applications of the MHT
6 Functional Maps
7 A better basis?
Fourier 1D

- 1D Fourier Transform of $f : \mathbb{R} \rightarrow \mathbb{C}$

$$F(w) = \int_{-\infty}^{\infty} f(x) \exp(-2\pi iwx) dx$$

- Inverse Fourier transform

$$f(x) = \int_{-\infty}^{\infty} F(w) \exp(2\pi iwx) dw$$

- Fourier transform of $f +$ reconstruction of $f$ by thresholding on the frequencies:

$$\tilde{f}(x) = \int_{-w_{\text{max}}}^{w_{\text{max}}} F(w) \exp(2\pi iwx) dw = \int_{-w_{\text{max}}}^{w_{\text{max}}} \langle f, e_w \rangle e_w dw$$
The Laplacian manifold harmonics basis is the set of eigenfunctions of the Laplacian.

Definition

The Laplacian manifold harmonics basis is the set of eigenfunctions of the Laplacian.
Manifold Harmonics Basis

Definition

The Laplacian manifold harmonics basis is the set of eigenfunctions of the Laplacian.

- On a mesh it means taking the eigenvectors of matrix $L$. 
Manifold Harmonics Basis

**Definition**

The Laplacian manifold harmonics basis is the set of eigenfunctions of the Laplacian.

- On a mesh it means taking the eigenvectors of matrix $L$.
- In general it would mean finding the Eigendecomposition of a $N \times N$ matrix. This is intractable for large meshes.
A quick reminder of Eigendecomposition

- If $A$ is a symmetric real valued matrix, it can be expressed as $A = U^T DU$ where $D$ is a diagonal matrix with eigenvalues $(\lambda_1, \cdots, \lambda_N)$ on the diagonal, $\lambda_i$ are nonnegative.
- $A$ is self-adjoint: $< v, Au > = < A^T v, u >$
- Theorem (Ky-Fan) $\sum_{i=1}^{N} \lambda_i = \min_{U \in SO(N)} tr(U^T AU)$
A small remark on Fiedler’s vector

Fiedler’s vector

The Fiedler’s vector is the eigenvector associated to the smallest nonzero eigenvalue.

- Theorem: the eigenvectors $(\Phi_i)_{i=1}^K$ minimize:

$$\forall i, \min \Phi_i \Phi_i^T A \Phi_i$$

where $\|\Phi\|_2 = 1, \langle \Phi_i, \Phi_k \rangle = 0$, for $k<i$.
A small remark on Fiedler’s vector

Fiedler’s vector

The Fiedler’s vector is the eigenvector associated to the smallest nonzero eigenvalue.

Theorem: the eigenvectors \((\Phi_i)_{i=1}^K\) minimize:

\[
\forall i, \min_{\|\Phi\|_2=1, \langle \Phi_i, \Phi_k \rangle = 0, \text{ for } k<i} \Phi_i^T A \Phi_i
\]

What can you deduce for the Fiedler’s vector?
A small remark on Fiedler’s vector

Fiedler’s vector

The Fiedler’s vector is the eigenvector associated to the smallest nonzero eigenvalue.

- Theorem: the eigenvectors $(\Phi_i)_{i=1}^K$ minimize:

$$\forall i, \min_{\|\Phi\|_2=1, <\Phi_i,\Phi_k>=0, \text{for } k<i} \Phi_i^T A \Phi_i$$

- What can you deduce for the Fiedler’s vector?
  - Solution of $\min_{u_1=0, \|u\|_2=1} \sum_i w_{ij} (u_i - u_j)^2$
A small remark on Fiedler’s vector

**Fiedler's vector**

The Fiedler’s vector is the eigenvector associated to the smallest nonzero eigenvalue.

- **Theorem**: the eigenvectors \((\Phi_i)_{i=1}^K\) minimize:

  \[
  \forall i, \min_{\|\Phi\|_2=1, \langle \Phi_i, \Phi_k \rangle = 0, \text{ for } k<i} \Phi_i^T A \Phi_i
  \]

- **What can you deduce for the Fiedler’s vector?**
  - Solution of \(\min_{u^T 1 = 0, \|u\|_2 = 1} \sum_i w_{ij} (u_i - u_j)^2\)
  - Solution to the minimum linear arrangement problem:
A small remark on Fiedler’s vector

**Fiedler’s vector**

The Fiedler’s vector is the eigenvector associated to the smallest nonzero eigenvalue.

- **Theorem:** the eigenvectors $(\Phi_i)_{i=1}^K$ minimize:

$$\forall i, \min_{\|\Phi\|_2=1, \langle \Phi_i, \Phi_k \rangle = 0, \text{for } k<i} \Phi_i^T A \Phi_i$$

- What can you deduce for the Fiedler’s vector?
  - Solution of $\min_{u^T 1=0, \|u\|_2=1} \sum_i w_{ij} (u_i - u_j)^2$
  - Solution to the minimum linear arrangement problem: An ordering of the vertices that best preserves the distance between vertices
Manifold Harmonics Transform and Inverse Transform

Manifold Harmonics Transform (MHT)

A function defined on the vertices of a mesh \( f = \sum_i x_i f_i \). Then:

\[
\tilde{f}_i = \langle f, \phi_i \rangle = \sum_{j=1}^{n} x_i \langle f_i, \phi_j \rangle \quad \text{(MHT)}
\]

\[
f = \sum_{i=1}^{n} \tilde{f}_i \phi_i \quad \text{(Inverse MHT)}.
\]
The spatial length of eigenfunction $\phi_i$ is linked to the corresponding eigenvalue $\lambda_i$ by:

$$\omega_i = \sqrt{\lambda_i}$$

Let $F$ be a transfer function: $F : \Omega \rightarrow \mathbb{R}$ associating each frequency to a real value.

### Fourier filtering

Filtering $f$ by $F$:

$$f_{\text{filtered}} = \sum_{i=1}^{N} F(w_i) \tilde{f}_i \phi_i$$
Low-frequency and high-frequency filtering

- High frequency term: \( f_{hf} = f - \sum_{k=1}^{m} \tilde{f}_k \phi_k \)
Low-frequency and high-frequency filtering

- High frequency term: \( f^{hf} = f - \sum_{k=1}^{m} \tilde{f}_k \phi_k \)
- Filtering the high frequency term:

\[
f^{filtered} = \sum_{k=1}^{m} F(w_k) \tilde{f}_k \phi_k + F^{hf} f^{hf}
\]
Low-frequency and high-frequency filtering

- High frequency term: \( f^{hf} = f - \sum_{k=1}^{m} \tilde{f}_k \phi_k \)
- Filtering the high frequency term:

\[
f^{filtered} = \sum_{k=1}^{m} F(w_k) \tilde{f}_k \phi_k + F^{hf} f^{hf}
\]

- \( F^{hf} \) is the average coefficient for all high frequencies.
Computation times

- Mesh with more than a few thousand vertices, extract all eigenvectors is computationally intractable
Computation times

- Mesh with more than a few thousand vertices, extract all eigenvectors is computationally intractable
- Divide the problem: [Karni 2000] partition the mesh into several submeshes
Mesh with more than a few thousand vertices, extract all eigenvectors is computationally intractable.

Divide the problem: [Karni 2000] partition the mesh into several submeshes.

Sparse solvers are not efficient for low eigenvalues → precisely what we are interested in.
Spectrum shift: $\Delta_S = \Delta - \lambda_S I$

[Vallet and Lévy 2008]
Efficient Computation of the Manifold Harmonics Basis

- Spectrum shift: $\Delta_S = \Delta - \lambda_S I$
- Spectrum Swap: compute the eigenvalues of $\Delta_S^{-1}$

[Vallet and Lévy 2008]
Efficient Computation of the Manifold Harmonics Basis

- Spectrum shift: $\Delta_S = \Delta - \lambda_S I$
- Spectrum Swap: compute the eigenvalues of $\Delta_S^{-1}$
- Link between the eigenvalues of $\Delta_S^{-1}$ and those of $\Delta$

[Vallet and Lévy 2008]
Efficient Computation of the Manifold Harmonics Basis

- Spectrum shift: $\Delta_S = \Delta - \lambda_S I$
- Spectrum Swap: compute the eigenvalues of $\Delta_S^{-1}$
- Link between the eigenvalues of $\Delta_S^{-1}$ and those of $\Delta$
- High end of $\Delta_S^{-1}$ spectrum corresponds to a band of eigenvalues around $\lambda_S$

[Vallet and Lévy 2008]
Algorithm for Band-by-band computation of the MHB
[Vallet, Lévy 2008]

1. \( \lambda_S = 0, \lambda_{last} = 0 \)
2. Compute \( \Delta_S = \Delta - \lambda_S I \)
3. Compute the 50 first eigenvectors \((\phi_k, \mu_k)\) of \( \Delta_S^{-1} \)
4. For \( i = 1 \cdots 50 \)
   1. \( \lambda_k \leftarrow \lambda_S + 1/\mu_k \)
   2. if \( \lambda_k > \lambda_{last} \) save \((\phi_k, \lambda_k)\)
5. \( \lambda_S \leftarrow \max(\lambda_k) + 0.4(\max \lambda_k - \min \lambda_k) \)
6. \( \lambda_{last} \leftarrow \max \lambda_k \)
When the memory is limited...

**Memory-limited Filtering**
Perform the filtering while computing the basis.
Dependency on the support

For images: whatever the content of the image, the basis functions are the same.
Dependency on the support

For images: whatever the content of the image, the basis functions are the same.

For surfaces: whatever the function defined on the surface, the basis functions are the same.
Dependency on the support

For images: whatever the content of the image, the basis functions are the same.
For surfaces: whatever the function defined on the surface, the basis functions are the same.
However, if the surface changes (geometrically or combinatorially), the basis functions change.
Decomposition on the MHB

Spectral decomposition of the shape (image from Vallet, Lévy 2008)
Outline

1. Building a Laplacian Operator on meshes
2. Surface Parameterization
3. Applications of the Laplacian on meshes
4. Manifold Harmonics Basis
5. Applications of the MHT
6. Functional Maps
7. A better basis?
Mesh smoothing

- Replace $f$ by the $(x, y, z)$ coordinate vector
- Build the MHB, decompose the coordinates on this basis
- Reconstruct the coordinates using the first $k$ basis functions

More complex filtering can be achieved by convolution or using different transfer functions.

Reconstruction with truncated basis (image from Vallet, Lévy 2008)
Figure 8: Filtering Stanford’s bunny. Results similar to geofilter are obtained, with the addition of interactivity, and without any shrinking effect.
Is it practical for surface fairing?

- Implies extracting eigenvectors and eigenvalues of a $n \times n$ matrix ($n$: number of vertices)
- Cheaper filtering needed!
- Laplacian flow (heat diffusion)
Laplace Beltrami Eigenfunctions: Shape DNA

Values of the MHB on isometry-deformed shapes.
Non-Rigid Registration of shapes

**Intrinsic Isometric Deformation**

A shape deformation is said to be an intrinsic isometric deformation if the geodesic distances are preserved under the deformation.

*Image from [Crane et al. 2013]*
Pose oblivious shape signatures

**Goal**

Find a characteristic of a shape that does not change with intrinsic isometric deformations.
Pose oblivious shape signatures

**Goal**

Find a characteristic of a shape that does not change with intrinsic isometric deformations.

For example:
Pose oblivious shape signatures

**Goal**
Find a characteristic of a shape that does not change with intrinsic isometric deformations.

For example:
- Local diameter combined with centricity $\rightarrow$ 2D histogram signature [Gal 2007]
Pose oblivious shape signatures

Goal

Find a characteristic of a shape that does not change with intrinsic isometric deformations.

For example:

- Local diameter combined with centricity $\rightarrow$ 2D histogram signature [Gal 2007]
- Laplace Beltrami eigenfunctions. [Rustamov 2007]
Local Diameter

The Shape-Diameter Function (SDF): scalar function defined on the mesh surface. It measures the diameter of the object’s volume in the neighborhood of each point on the surface.

- Cone aligned with the inward oriented normal
- Rays cast inside the cone.
- SDF: weighted average of all rays lengths which fall within one standard deviation from the median of all lengths.
- Weights = inverse of the angle between the ray to the center of the cone.

Image from [Shapira et al. 2008]
Shape Diameter Function

SDF partitioning and skeleton obtained. Image from [Shapira et al. 2008]
Global Point Signature

- A shape signature at each point $p$ is computed as:

$$GPS(p) = \left( \frac{1}{\sqrt{\lambda_1}} \Phi_1(p), \frac{1}{\sqrt{\lambda_2}} \Phi_2(p), \ldots, \frac{1}{\sqrt{\lambda_N}} \Phi_N(p) \right)$$

- $GPS$ provides an embedding of the surface in the GPS domain.

Global Point Signature, Image [from Rustamov 2007], 15 eigenvalues are used, yielding a GPS space of dimension 15 represented by multidimensional scaling.
Heat Kernel Signature

- Proposed simultaneously by [Gebal et al. 2009], [Sun et al. 2009]
Heat Kernel Signature

- Proposed simultaneously by [Gebal et al. 2009], [Sun et al. 2009]
- Goal: Describe points on the shape in a stable, efficient, multiscale way.
Given a compact Riemannian manifold $S$ without boundary,

- Some initial heat distribution $f : S \to \mathbb{R}^+$ at time 0.
- $u(x, t) : S \times \mathbb{R}^+ \to \mathbb{R}^+$ amount of heat at a point $x \in S$ at time $t$

### Heat kernel

$u(x, t)$ satisfies the heat equation:

$$\frac{\partial u}{\partial t} = -\Delta_S u \text{ and } u(x, 0) = f(x).$$

The heat kernel on $S$ is the unique function $k^S_t : \mathbb{R}^+ \times S \times S \to \mathbb{R}^+$ such that for all $f \in L^2$, $x \in S$ and $t \in \mathbb{R}^+$,

$$u(x, t) = \int_S k_t(x, y)f(y)dy.$$
Given a compact Riemannian manifold, the heat kernel has the following eigen decomposition:

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

where \( \lambda_i \) and \( \phi_i \) are the eigenvalues and eigenfunctions of the Laplace operator.

Given a point \( x \) on a manifold \( S \), the function \( k_t(x, \cdot) \) defined on the surface for each \( t \) is a good descriptor. ⇒ too expensive in practice!
Theorem

- Given a compact Riemannian manifold, the heat kernel has the following eigen decomposition:

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

where \( \lambda_i \) and \( \phi_i \) are the eigenvalues and eigenfunctions of the Laplace operator.

- Given a point \( x \) on a manifold \( S \), the function \( k_t(x, \cdot) \) defined on the surface for each \( t \) is a good descriptor. \( \Rightarrow \) too expensive in practice!
Definition

Given a point $x$ on a manifold $S$ its Heat Kernel Signature $HKS(x)$ is defined as:

$$HKS : \mathbb{R}^+ \rightarrow \mathbb{R}$$

$$HKS(x, t) = k_t^S(x, x)$$
Definition

- Given a point \( x \) on a manifold \( S \) its Heat Kernel Signature \( HKS(x) \) is defined as:

\[
HKS : \mathbb{R}^+ \rightarrow \mathbb{R}
\]

\[
HKS(x, t) = k^S_t(x, x)
\]

Informative Theorem

Despite restricting the signature to \( \mathbb{R}^+ \times S \) and dropping the entire spatial domain, under mild assumptions, \((k_t(x, x))_{t>0}\) keeps all of the information of \((k_t(x, \cdot))_{t>0}\).
Heat Kernel Signatures on 4 points of the dragon. Image from [Sun et al. 2009]

**Scaled HKS**

The HKS is scaled:

$$HKS(x) = \frac{k_t(x, x)}{\int_S k_t(y, y)dy}$$
Isometry Characterization

Isometry

If the eigenvalues of the Laplace-Beltrami operators on $S$ and $S'$ are not repeated and $T$ is a homeomorphism from $S$ to $S'$, then $T$ is isometric if and only if $k^S_t(x, x) = k^{S'}_t(T(x), T(x))$ for any $x \in S$ and $t > 0$. 
Properties of the HKS

- Multiscale information about the neighborhood of a point
Properties of the HKS

- Multiscale information about the neighborhood of a point
- Comparison between two points of a shape between scales $t_1$ and $t_2$

$$d_{[t_1, t_2]}(x, y) = \left( \int_{[t_1, t_2]} \left( \frac{|k_t(x, x) - k_t(y, y)|}{\int_S k_t(x, x')dx'} \right)^2 d \log t \right)^{\frac{1}{2}}$$
\( d_{[t_1, t_2]}(x, y) \) for \( t_1, t_2 \) small (left) and small \( t_1 \), large \( t_2 \) (right). Image from [Sun et al. 2009]
Practical algorithm

- Compute the HKS at each point for values of $t$ sampled logarithmically over the time domain.
Practical algorithm

- Compute the HKS at each point for values of $t$ sampled logarithmically over the time domain.
- Integral estimated as the $L^2$ distance between the vectors at each point.
Practical algorithm

- Compute the HKS at each point for values of $t$ sampled logarithmically over the time domain.
- Integral estimated as the $L^2$ distance between the vectors at each point.
- Threshold on the distance to find matches
Practical algorithm

- Compute the HKS at each point for values of $t$ sampled logarithmically over the time domain.
- Integral estimated as the $L^2$ distance between the vectors at each point.
- Threshold on the distance to find matches

**HKS and scales**

At small scales, points with similar small neighborhoods are reported. At large scale, only those with similar global structure are found.
Image from [Sun et al. 2009]
Matching through Heat Kernel Signatures [Ovsjanikov et al. 2009]

Problem Statement

Match two shapes that undergo some deformation that is not necessarily rigid but is an intrinsic isometry.

Intrinsic isometry deformation between two shapes. Image from [Ovsjanikov et al. 2009]

Morphing applications; interpolation for moving shapes.
[Ovsjanikov et al. 2010]

Under mild genericity conditions the knowledge of a single correspondence can be used to recover an isometry defined on entire shapes.
Under mild genericity conditions the knowledge of a single correspondence can be used to recover an isometry defined on entire shapes.

- The image of every point is characterized by the preservation of the heat kernel to the given correspondence points.
Isometry

**Theorem**

Let $S$ and $S'$ be two compact, connected Riemannian manifolds without boundary. A map $T$ from $S$ to $S'$ is an isometry if and only if

$$k^S_t(x, y) = k^{S'}_t(T(x), T(y))$$

for all $x, y \in S$, $t > 0$. 

Applications of the MHT
Isometry

**Theorem**

Let $S$ and $S'$ be two compact, connected Riemannian manifolds without boundary. A map $T$ from $S$ to $S'$ is an isometry if and only if $k^S_t(x, y) = k^{S'}_t(T(x), T(y))$ for all $x, y \in S$, $t > 0$.

- For small values of $t$, the heat kernel $k^S_t(x, \cdot)$ can be well approximated by the heat kernel of a small geodesic neighborhood of the point $x$. 
Isometry

**Theorem**

Let $S$ and $S'$ be two compact, connected Riemannian manifolds without boundary. A map $T$ from $S$ to $S'$ is an isometry if and only if

$$k_t^S(x, y) = k_t^{S'}(T(x), T(y)) \text{ for all } x, y \in S, t > 0.$$ 

- For small values of $t$, the heat kernel $k_t^S(x, \cdot)$ can be well approximated by the heat kernel of a small geodesic neighborhood of the point $x$.
- $t$ can be interpreted as the geometric scale of the kernel.
Heat Kernel Map

Definition (Heat Kernel Map)

Given a manifold $S$ and a source point $p \in S$, the heat kernel map is defined as:

$$\Phi^S_p : S \to F; \Phi^S_p(x) = k^S_t(p, x)$$

Where $F$ is the space of functions from $\mathbb{R}^+$ to $\mathbb{R}^+$.

Under mild assumptions, the Heat Kernel Map is injective:

$$\Phi^S_p(x) = \Phi^S_p(y) \Rightarrow x = y$$
Heat Kernel Map

**Definition (Heat Kernel Map)**

Given a manifold $S$ and a source point $p \in S$, the heat kernel map is defined as:

$$\Phi^S_p : S \rightarrow F; \Phi^S_p(x) = k^S_t(p, x)$$

Where $F$ is the space of functions from $\mathbb{R}^+$ to $\mathbb{R}^+$. Under mild assumptions, the Heat Kernel Map is injective:

$$\Phi^S_p(x) = \Phi^S_p(y) \Rightarrow x = y$$

**Mild Assumptions:** $S$ is a generic manifold (no repeated eigenvalue), $p$ is a generic point (for all $\Phi_i$ eigenfunction of $\Delta$, $\Phi_i(p) \neq 0$. 

Applications of the MHT
From one point correspondence to a correspondence on the whole shape

- Given a generic point $p$ of $S$, there is at most one isometry $f$ from $S$ to $S'$ that sends $p$ to a given point $q \in S'$.
- If such an isometry exists, the image $f(x)$ of a point $x$ is characterized by the preservation of the heat kernel to $p$: $f(x)$ is the only point such that $k_t^S(p, x) = k_t^S(q, f(x))$ for all $t > 0$.

**Consequence**

Given a single correspondence, the correspondence for every other point of a shape can be recovered.
Algorithm

1. **Feature Detection:** Detect features on $S$ and $S'$
2. **Correspondence Search:** Match features from $S$ to features from $S'$
3. **Match Propagation:** Propagate the correspondences on the surface
Given two shapes $S$ and $S'$ detect a sparse set of feature points $P \subset S$ and $Q \subset S'$

**HKS feature points**

The HKS feature points are local maxima of the heat kernel signature for a large time $t$: $p$ is a feature point iff:

$$
\forall x \in \mathcal{N}_2(p) k_t^S(p, p) > k_t^S(x, x)
$$
Correspondence Search

For each couple \((p_i, q_j) \in P \times Q\), compare \(\Phi_{p_i}^S\) and \(\Phi_{q_j}^{S'}\) for various \(t\).
Correspondence Search

- For each couple \((p_i, q_j) \in P \times Q\), compare \(\Phi^S_{p_i}\) and \(\Phi^{S'}_{q_j}\) for various \(t\).
- In practice: For each \(x \in S\) compute \((k_n^S(p_i, x))_{k=1\ldots t_l}\) and similarly for each \(y \in S'\) \((k_n^{S'}(q_j, y))_{n=1\ldots t_l}\).
Correspondence Search

- For each couple \((p_i, q_j) \in P \times Q\), compare \(\Phi^S_{p_i}\) and \(\Phi^{S'}_{q_j}\) for various \(t\).
- In practice: For each \(x \in S\) compute \((k^n_S(p_i, x))_{k=1\ldots t_i}\) and similarly for each \(y \in S'\) \((k^n_{S'}(q_j, y))_{n=1\ldots t_i}\).
- The match quality is:

\[
E(p_i, q_j) = \sum_{x \in S} \min_{y \in S'} \| \Phi^S_{p_i}(x) - \Phi^{S'}_{q_j}(y) \|^2
\]
Correspondence Search

- For each couple \((p_i, q_j) \in P \times Q\), compare \(\Phi^S_{p_i}\) and \(\Phi^S'_{q_j}\) for various \(t\).
- In practice: For each \(x \in S\) compute \((k^S_n(p_i, x))_{k=1\ldots t_l}\) and similarly for each \(y \in S'\) \((k^{S'}_n(q_j, y))_{n=1\ldots t_l}\).
- The match quality is:
  \[
  E(p_i, q_j) = \sum_{x \in S} \min_{y \in S'} \| \Phi^S_{p_i}(x) - \Phi^{S'}_{q_j}(y) \|^2
  \]

- Proposed distance: \(\infty\) norm multiplied by the scale parameter \(n\).
Match Propagation

Retrieving the correspondences

Under the genericity assumption, for a given \( x \in S \) there exists a unique \( y \in S' \) such that:

\[
\| \Phi^S_{p_i}(x) - \Phi^{S'}_{q_j}(y) \| = 0
\]

In practice: \( f(x) = \text{argmin}_{y \in S} \| \Phi^S_{p_i}(x) - \Phi^{S'}_{q_j}(y) \| \)
Match Propagation

Retrieving the correspondences

Under the genericity assumption, for a given $x \in S$ there exists a unique $y \in S'$ such that:

$$\|\Phi^S_{p_i}(x) - \Phi^{S'}_{q_j}(y)\| = 0$$

In practice: $f(x) = \arg\min_{y \in S} \|\Phi^S_{p_i}(x) - \Phi^{S'}_{q_j}(y)\|$ 

- nearest neighbor search in the space of heat kernel maps
Matching between deformed shapes via the one point isometric. [Ovsjanikov et al. 2010]
Outline

1. Building a Laplacian Operator on meshes
2. Surface Parameterization
3. Applications of the Laplacian on meshes
4. Manifold Harmonics Basis
5. Applications of the MHT
6. Functional Maps
7. A better basis?
Shape functions

Principle

Representation for maps between pairs of shapes as linear transformations between the corresponding function space.

Image from Ovsjanikov et al. 2012
Context

- \( T : S \to S' \) be a bijective mapping between manifolds \( S \) and \( S' \).
- *Functional representation* of the mapping \( T \).

\[
T_F : \mathcal{F}(S, \mathbb{R}) \to \mathcal{F}(S', \mathbb{R})
\]
\[
f \to g = f \circ T^{-1}
\]

**Properties of \( T_F \)**

- Given \( T \), \( T_F \) is a linear operator.
- Given \( T_F \), the bijective mapping \( T \) can be recovered.
Manifold Harmonics Basis returns!

\[ f = \sum_i \alpha_i \Phi_i^S \]

\[ T_F(f) = \sum_i \alpha_i T_F(\Phi_i^S) \]

\( T_F(\Phi_i^S) \) is a function defined on \( S' \):

\[ T_F(\Phi_i^S) = \sum_j c_{ij} \Phi_j^{S'} \]

\( (C = (c_{ij}) \text{ possibly infinite matrix.}) \)

If \( T \) is an isometry, \( c_{ij} \neq 0 \) iff \( \Phi_i^S \) and \( \Phi_j^{S'} \) correspond to the same eigenvalue.

**Truncated MHB**

Only the \( n \) first eigenfunctions are used (Typically \( n = 100 \)).
Expressing constraints on mappings

- Function preservation expresses a linear constraint on $C$: Two matching functions $f$ and $g$ (vectors $a$ and $b$), $Ca = b$
- Commutativity with linear operators (e.g. symmetry): Functional operators $S_F$ and $R_F$: $CS_F = R_F C$
- If the basis functions are orthonormal or $T$ is volume preserving, then $C$ associated to $T$ is orthonormal. If $T$ is an isometry, $C$ commutes with the Laplace-Beltrami operator.
Algorithm:

1. Compute a set of descriptors for $S$ and $S'$ → create function preservation constraints
2. If landmark correspondences exist, create the corresponding function preservation constraints
3. Include relevant commutativity constraints (symmetry, Laplace-Beltrami)
4. Linear system for $C$, least-squares solve
5. Refine $C$
6. Deduce point correspondences if needed.
Refinement of matrix $C$

Refinement needed because: $C$ has to be orthonormal, and $C\Phi_i^S$ must coincide with some $\Phi_{j}^{S'}$.

**Refinement process**

Iterate:

1. For each column $C_0\Phi_i^S$, find the closest $\Phi_{j(i)}^{S'}$

2. Find the optimal orthonormal $C$ minimizing $\sum_i \| C\Phi_i^S - \phi_{j(i)}^{S'} \|$

3. Set $C_0 = C$
Point to point correspondence recovery

- Naive method: a highly peaked function around $x$ on $S$ solve for:
  \[ y = \arg\max (T_F(f)) \]

- Better: $\delta_x$ has decomposition $\alpha_i(x) = \Phi_i^S(x)$ For every point in $C\Phi^S$, find the nearest point in $\Phi^{S'}$
  
- can be done via a kd tree
Application: Shape Matching

(a) target  
(b) Cat10  
(c) Cat1  
(d) Cat2  
(e) Cat6

Image from [Ovsjanikov et al. 2012]
Application: Segmentation Transfer

user-input; indicator function of a segment; transferred segmentation; image from [Ovsjanikov et al. 2012]
Outline

1 Building a Laplacian Operator on meshes
2 Surface Parameterization
3 Applications of the Laplacian on meshes
4 Manifold Harmonics Basis
5 Applications of the MHT
6 Functional Maps
7 A better basis?
Better basis?

- Disadvantages: not localized $\Rightarrow$ not resilient to holes
- Function Localization by adding a sparsity constraint on the eigenfunctions [Neumann et al. 2014].
Reformulation of the Manifold Harmonics Basis problem

Solve for a set of \( N \) functions \( \Phi_k \) such that

\[
\min_{\Phi_k, k=1\ldots N} \sum_{k=1}^{N} \langle \Phi_k, \Delta \Phi_k \rangle
\]

s.t. \( \langle \Phi_k, \Phi_j \rangle = \delta_{jk} \)
Reformulation of the Manifold Harmonics Basis problem

Solve for a set of $N$ functions $\Phi_k$ such that

$$\min_{\Phi_k, k=1 \ldots N} \sum_{k=1}^{N} \langle \Phi_k, \Delta \Phi_k \rangle$$

s.t. $\langle \Phi_k, \Phi_j \rangle = \delta_{jk}$

- Idea: add a penalty on the width of the support of the functions
Compressed Manifold Modes

Solve for a set of $N$ functions $\Phi_k$ such that

$$\min_{\Phi_k, k=1 \ldots N} \sum_{k=1}^{N} \langle \Phi_k, \Delta \Phi_k \rangle + \lambda \| \Phi_k \|_1$$

s.t. $\langle \Phi_k, \Phi_j \rangle = \delta_{jk}$
Localized Manifold Harmonics Basis

Compressed Manifold Modes

Solve for a set of $N$ functions $\Phi_k$ such that

$$\min_{\Phi_k, k=1 \ldots N} \sum_{k=1}^{N} \langle \Phi_k, \Delta \Phi_k \rangle + \lambda \| \Phi_k \|_1$$

s.t. $\langle \Phi_k, \Phi_j \rangle = \delta_{jk}$

- The $\| \|_1$ term penalizes basis functions with wide support.
A small remark on $\ell^1$

- Pseudo-norms $\ell^p$ with $p < 1$ favor sparsity, \textit{i.e.} few nonzero coefficients.
- Does $\ell^1$ favors sparsity?
A small remark on $\ell^1$

- Pseudo-norms $\ell^p$ with $p < 1$ favor sparsity, *i.e.* few nonzero coefficients.
- Does $\ell^1$ favors sparsity?
- The gradient of the $\ell^1$ regularization is piecewise constant.
A small remark on $\ell^1$

- Pseudo-norms $\ell^p$ with $p < 1$ favor sparsity, *i.e.* few nonzero coefficients.
- Does $\ell^1$ favors sparsity?
- The gradient of the $\ell^1$ regularization is piecewise constant
- The gradient of the $\ell^2$ regularization is linearly decreasing near 0
Solution for the Compressed Manifold Modes

Mesh CMM

Solve for $\Phi$ such that:

$$\min_{\Phi} \text{Tr}(\Phi^T L\Phi) + \lambda \|\Phi\|_1$$

s.t. $\Phi^T D\Phi = I$ with $D$ the area matrix.

- This problem is solved by ADMM
ADMM

Alternating Directions Method of Multipliers can solve problems in the form

\[ f(x) + g(z) \text{ s.t. } Ax + Bz = c \]
Optimization

**ADMM**
Alternating Directions Method of Multipliers can solve problems in the form

\[ f(x) + g(z) \text{ s.t. } Ax + Bz = c \]

- To reformulate, introduce

\[ l(\Phi) = \begin{cases} 
0 & \text{if } \Phi^T D \Phi = I \\
\infty & \text{otherwise}
\end{cases} \]
Formulation

$$\min_{\Phi, S, E} I(\Phi) + \text{Tr}(EL^T E) + \mu \|S\|_1$$

s.t.  
$$E = \Phi$$  
$$S = \Phi$$
Optimization

Formulation

\[
\min_{\Phi, S, E} I(\Phi) + \text{Tr}(EL^TE) + \mu \|S\|_1 \\
\text{s.t. } E = \Phi \quad S = \Phi
\]

ADMM Formulation

\[
\min_{\Phi, M} f(\Phi) + g(M) \\
\text{s.t. } A\Phi + BM = 0
\]

where \( M = \begin{pmatrix} E \\ S \end{pmatrix} \), \( g(M) = \begin{pmatrix} \text{Tr}(EL^TE) \\ \mu \|S\|_1 \end{pmatrix} \), \( A = \begin{pmatrix} I \\ I \end{pmatrix} \) and \( B = \begin{pmatrix} -I & 0 \\ 0 & -I \end{pmatrix} \).
Optimization

Formulation

\[
\min_{\Phi, S, E} \; I(\Phi) + \text{Tr}(EL^TE) + \mu \|S\|_1 \\
\text{s.t.} \; E = \Phi \\quad S = \Phi
\]

ADMM Formulation

\[
\min_{\Phi, M} \; f(\Phi) + g(M) \\
\text{s.t.} \; A\Phi + BM = 0
\]

where \( M = \begin{pmatrix} E \\ S \end{pmatrix} \), \( g(M) = \begin{pmatrix} \text{Tr}(EL^TE) \\ \mu \|S\|_1 \end{pmatrix} \), \( A = \begin{pmatrix} I \\ I \end{pmatrix} \) and \( B = \begin{pmatrix} -I & 0 \\ 0 & -I \end{pmatrix} \).

The final algorithm iterates over the minimization on \( \Phi, E \) and \( S \).
Optimization

- ADMM is guaranteed to converge for convex $f$ and $g$. 
Optimization

- ADMM is guaranteed to converge for convex $f$ and $g$
- $l$ is nonconvex.
ADMM is guaranteed to converge for convex $f$ and $g$

- $I$ is nonconvex.

Instead of a global minimum, a local minimum is found.
A better basis?
Influence of the sparsity weight $\mu$

(a) $\mu = 0$  
(b) $\mu = \frac{1}{100} \cdot N$  
(c) $\mu = 1 \cdot N$  
(d) $\mu = 10 \cdot N$
Reconstruction with a very small number of frequencies

(a) Input Mesh       (b) MH reconstr.       (c) CMM reconstr.

Reconstruction with a very small number of frequencies
CMM in shapes with missing data

(a) 5 MH’s computed separately on the hand (left) and the partial hand (right)

(b) 5 of the proposed CMMs computed separately on the hand (left) and the partial hand (right)
CMM on isometry-deformed shapes
Functional Maps using CMM
Conclusion

- Manifold Harmonics Basis is at the heart of Mesh processing
- There is no perfect Laplace-Beltrami operator
- Laplace-Beltrami operator for point clouds, another way to represent a shape.