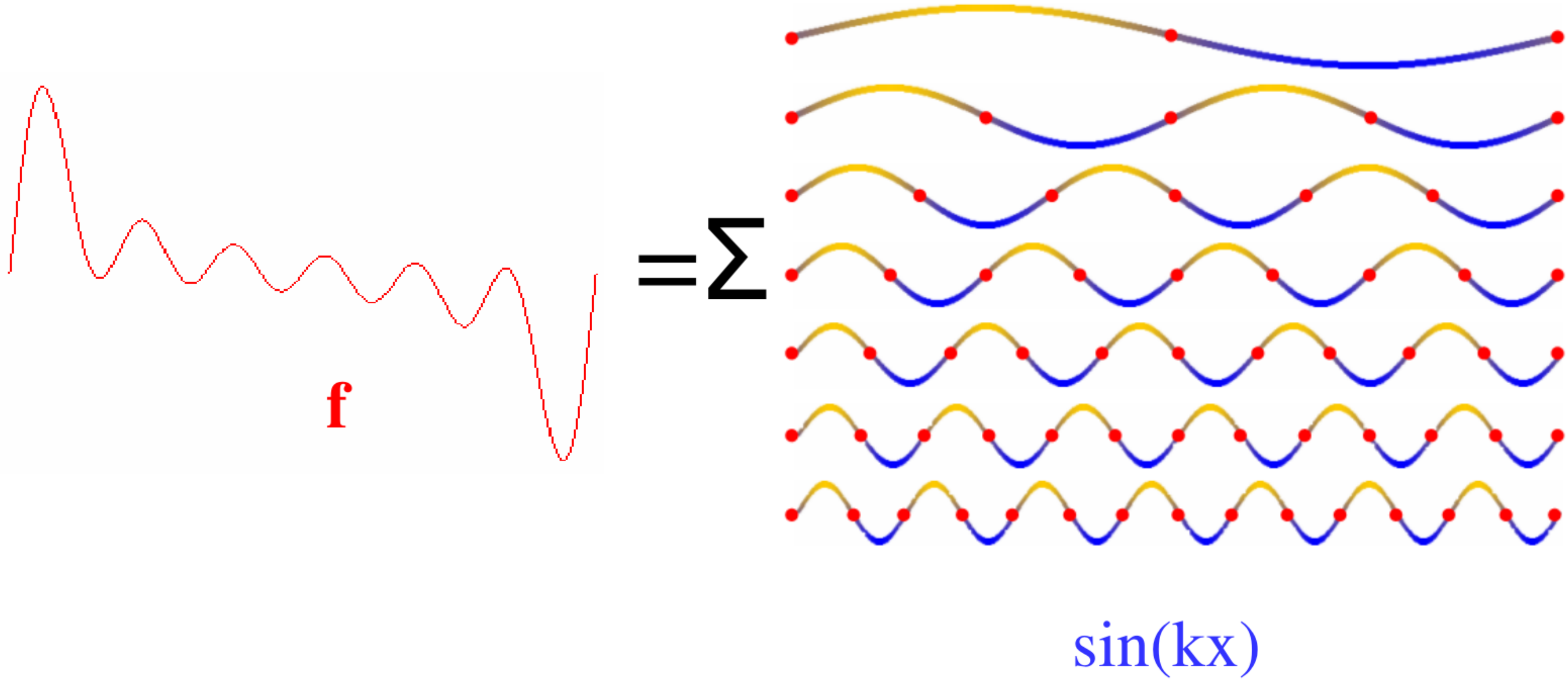
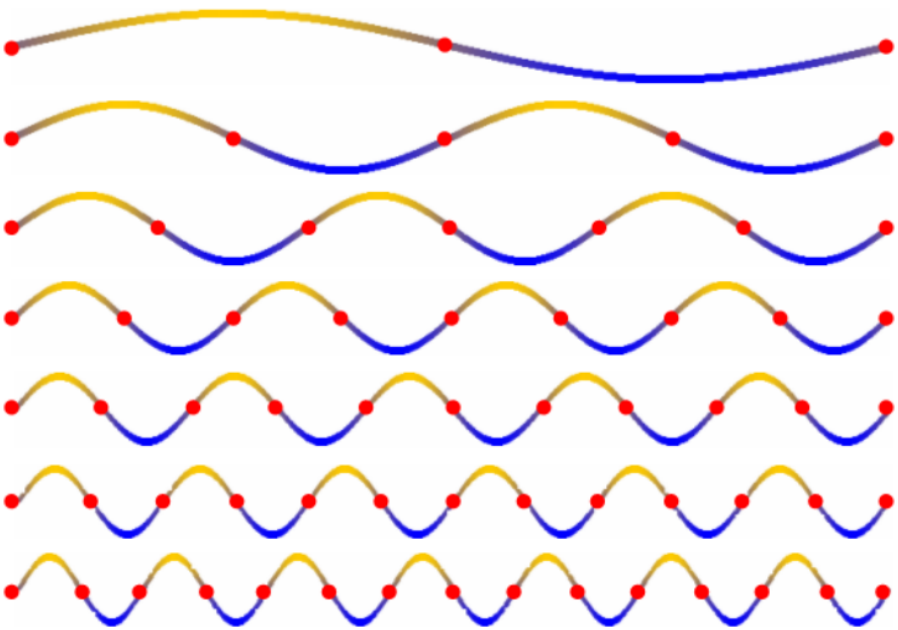
Spectral geometry on triangle meshes

Harmonics and spectral filtering



Harmonics and spectral filtering

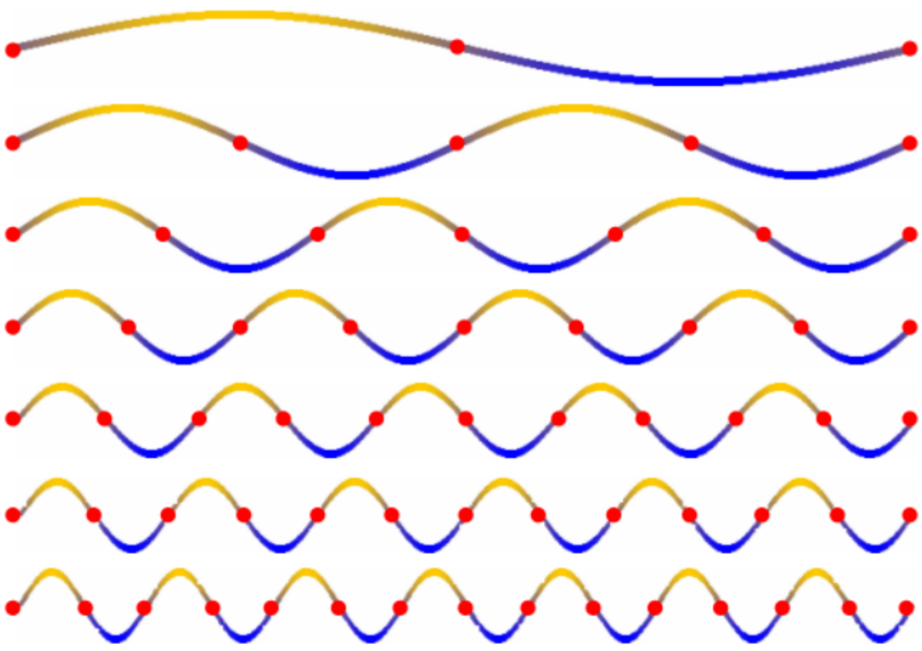


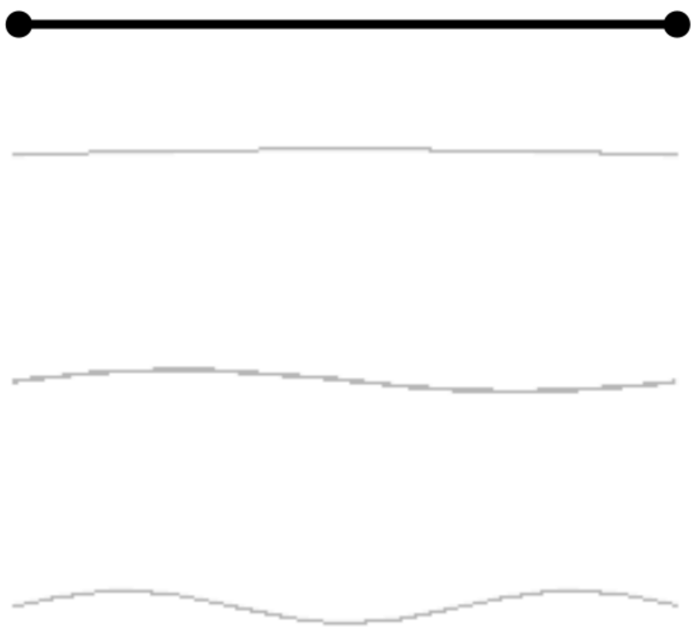
sin(kx)

on

?

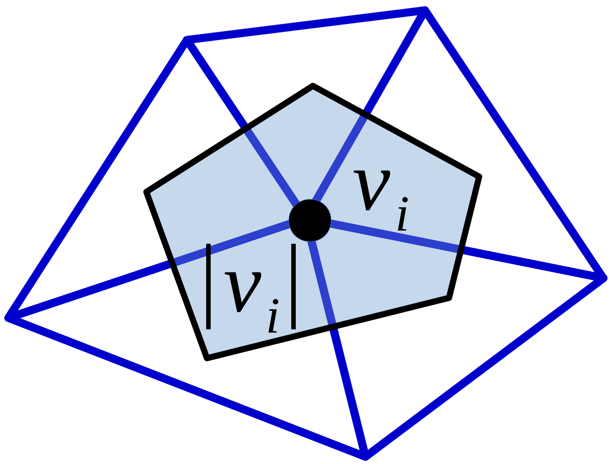
Harmonics and spectral filtering

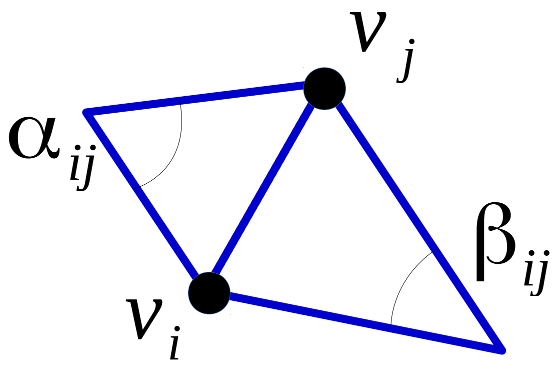




Strings harmonics = eigenvectors of unidimensional Laplacien

Reminder : Laplacian of scalar functions





$$\nabla^2 f(v_i) = \frac{1}{|v_i|} \sum_{e_{ij}} \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2} \left(f(v_j) - f(v_i) \right)$$

Note : You may see the version without 1/|v_i| here and there. Once again, the version without is the **integrated** operator (integrated over the area around vertex v_i), and the version with is the point-wise operator.

Spectral decomposition

Takes scalars defined on vertices, computes the Laplacian at each vertex

$$L \cdot \begin{bmatrix} f(0) \\ f(1) \\ . \\ . \\ f(|V|-1) \end{bmatrix} = \begin{bmatrix} \nabla^2 f(0) \\ \nabla^2 f(1) \\ . \\ . \\ \nabla^2 f(|V|-1) \end{bmatrix}$$

$$\in \mathbb{R}^{|V|\times 1} \qquad \in \mathbb{R}^{|V|\times 1}$$

$$\begin{cases} L(i,j) = \dfrac{1}{|v_i|} \dfrac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2} \\ L(i,i) = -\sum_{e_{ij}} L(i,j) \end{cases}$$

Spectral decomposition

Takes scalars defined on vertices, computes the Laplacian at each vertex

$$L \cdot \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(|V|-1) \end{bmatrix} = \begin{bmatrix} \nabla^2 f(0) \\ \nabla^2 f(1) \\ \vdots \\ \nabla^2 f(|V|-1) \end{bmatrix}$$

$$\in \mathbb{R}^{|V|\times 1} \qquad \in \mathbb{R}^{|V|\times 1}$$

$$\begin{cases} L(i,j) = \dfrac{1}{|v_i|} \dfrac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2} \\ L(i,i) = -\sum_{e_{ij}} L(i,j) \end{cases}$$

Eigenvectors of Laplacian shoud be eigenvectors of L

Spectral decomposition

Takes scalars defined on vertices, computes the Laplacian at each vertex

$$L \cdot \begin{bmatrix} f(0) \\ f(1) \\ . \\ . \\ f(|V|-1) \end{bmatrix} = \begin{bmatrix} \nabla^2 f(0) \\ \nabla^2 f(1) \\ . \\ . \\ \nabla^2 f(|V|-1) \end{bmatrix}$$

$$\in \mathbb{R}^{|V|\times 1} \qquad \in \mathbb{R}^{|V|\times 1}$$

$$\begin{cases} L(i,j) = \frac{1}{|v_i|} \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2} \\ L(i,i) = -\sum_{e_{ij}} L(i,j) \end{cases}$$

PROBLEM ! It is not symmetric : L(i,j) not = L(j,i)

→ **Eigenvectors of L are not orthogonal**

Spectral decomposition

$$L = A^{-1} . L_C$$

Point-wise Laplacian

$$\begin{cases} L(i,j) = \frac{1}{|v_i|} \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2} \\ L(i,i) = -\sum_{e_{ij}} L(i,j) \end{cases}$$

« Integrated » Laplacian :

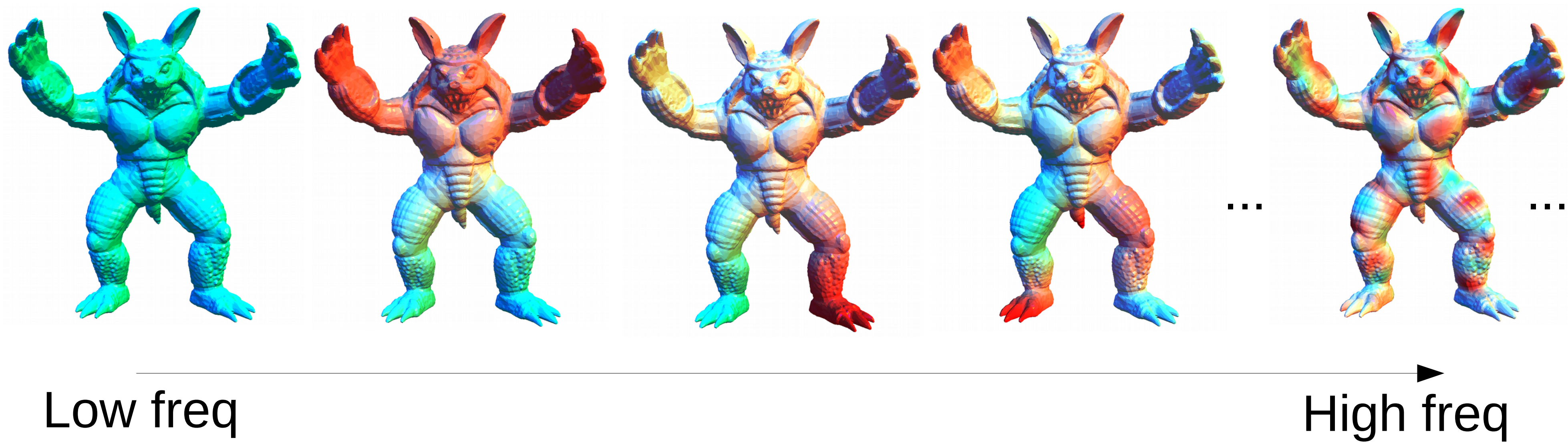
$$\begin{cases} L_C(i,j) = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2} \\ L_C(i,i) = -\sum_{e_{ij}} L_C(i,j) \end{cases}$$

Diagonal mass matrix :

$$A(i,i) = |v_i|$$

Spectral decomposition

« General » eigenvectors of : $-L_C.\psi_i = \lambda_i A.\psi_i$



Pseudo-orthogonality : $\psi_i^T.A.\psi_j = \delta_i^j$ (instead of $\psi_i^T.\psi_j = \delta_i^j$)

C++ : arpack++ (used for the examples made here) , Eigen3 with Spectra

An orthogonal basis

« General » eigenvectors of : $-L_C . \psi_i = \lambda_i A . \psi_i$

Pseudo-orthogonality : $\psi_i^T . A . \psi_j = \delta_i^j$

$$\bar{\psi}_i := \sqrt{A} . \psi_i$$

$$\bar{\psi}_i^T . \bar{\psi}_j = \psi_i^T . \sqrt{A}^T . \sqrt{A} . \psi_j = \psi_i^T . A . \psi_j = \delta_j^i$$

→ $\{\bar{\psi}_i\}_i$ is a good choice for decomposition : It is an orthonormal basis. (choice seen in related works, not the most obvious, see next)

Decomposition

Given a function f on the vertices

$$F_i := \bar{\psi}_i^T . f = \sum_{v_j} \bar{\psi}_i(v_j) f(v_j)$$ is its i^th frequency.

f can be recovered from its frequencies (inverse transform) :

$$f = \sum_i F_i \bar{\psi}_i$$

$$\begin{bmatrix} f \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{\psi}_0 \, \bar{\psi}_1 \ldots \bar{\psi}_{n-1} \end{bmatrix}}_{B^{-1}} . \underbrace{\begin{bmatrix} \bar{\psi}_0^T \\ \bar{\psi}_1^T \\ . \\ . \\ \bar{\psi}_{n-1}^T \end{bmatrix}}_{B} . \begin{bmatrix} f \end{bmatrix}$$

Decomposition (probably more correct)

Given a function f on the vertices

$$F_i := \langle \psi_i | f \rangle = \int_x \psi_i(x) f(x)\, dx = \psi_i^T . A . f$$ is its i^th frequency.

f can be recovered from its frequencies (inverse transform) :

$$f = \sum_i F_i \psi_i$$

$$\begin{bmatrix} f \end{bmatrix} = \underbrace{\begin{bmatrix} \psi_0\, \psi_1 \,.\,.\, \psi_{n-1} \end{bmatrix}}_{\text{Basis}} . \underbrace{\begin{bmatrix} \psi_0^T \\ \psi_1^T \\ . \\ . \\ \psi_{n-1}^T \end{bmatrix} . A . \begin{bmatrix} f \end{bmatrix}}_{\text{Dot product with Basis}}$$

Basis ψ

Dot product with Basis $\psi^T . A . f$

$$(\psi^T . A . \psi = Id)$$

Filtering

Given a function f on the vertices

$F_i := \bar{\psi}_i^{T} . f = \sum_{v_j} \bar{\psi}_i(v_j) f(v_j)$ is its i^th frequency.

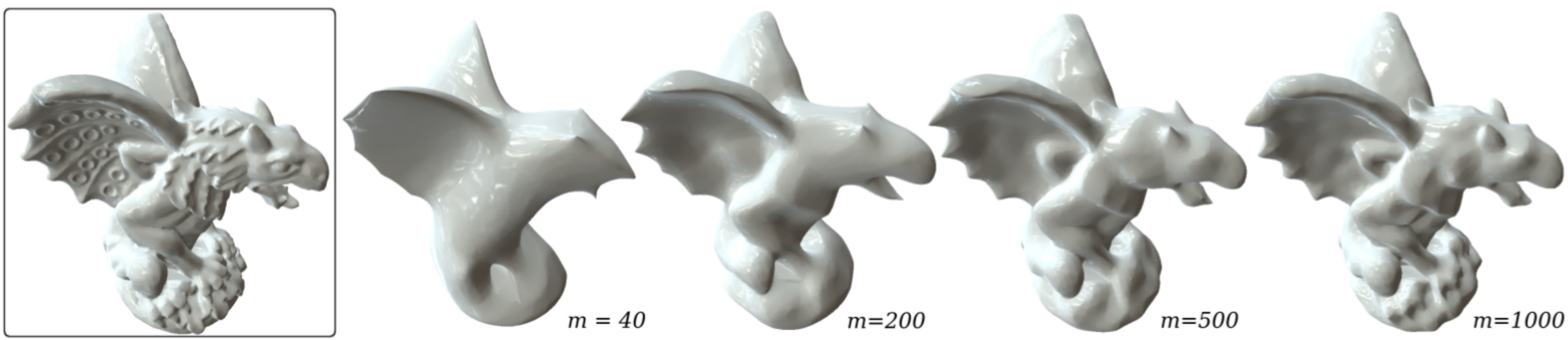
Filter h can be applied on the frequencies: $h \circ f = \sum_i h(F_i) \bar{\psi}_i$

Filtering

Given a function f on the vertices

$$F_i := \bar{\psi}_i^{\,T} . f = \sum_{v_j} \bar{\psi}_i(v_j) f(v_j)$$ is its i^th frequency.

Filter h can be applied on the frequencies: $$h \circ f = \sum_i h(F_i) \bar{\psi}_i$$

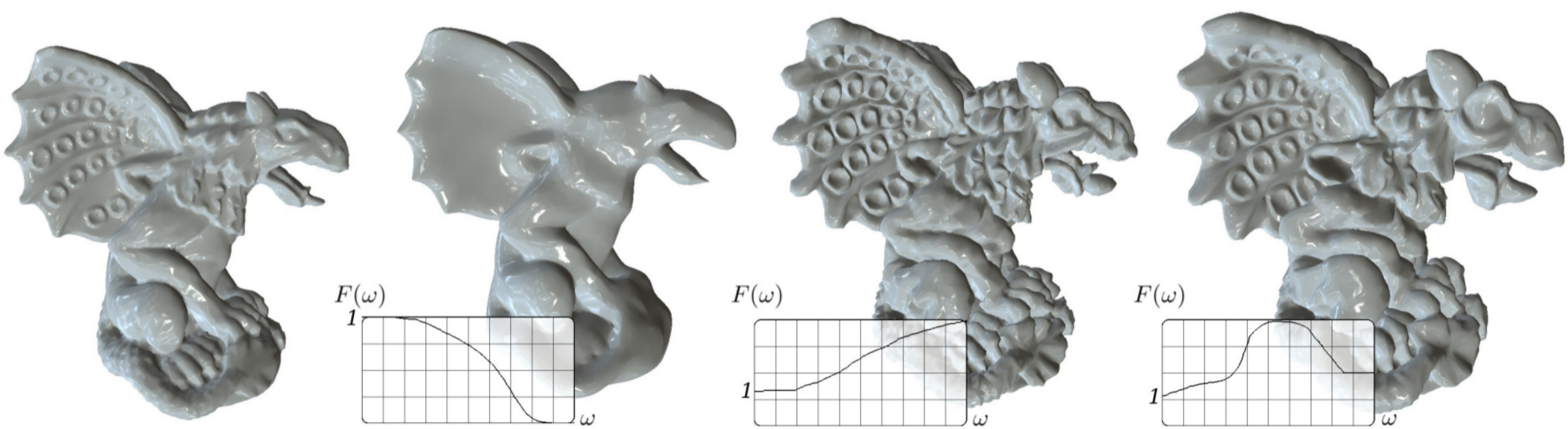


Filtering

Given a function f on the vertices

$F_i := \bar{\psi}_i^T . f = \sum_{v_j} \bar{\psi}_i(v_j) f(v_j)$ is its i^th frequency.

Filter h can be applied on the frequencies: $h \circ f = \sum_i h(F_i) \bar{\psi}_i$



Filtering

Given a function f on the vertices

$F_i := \bar{\psi}_i^{\,T} . f = \sum_{v_j} \bar{\psi}_i(v_j) f(v_j)$ is its i^th frequency.

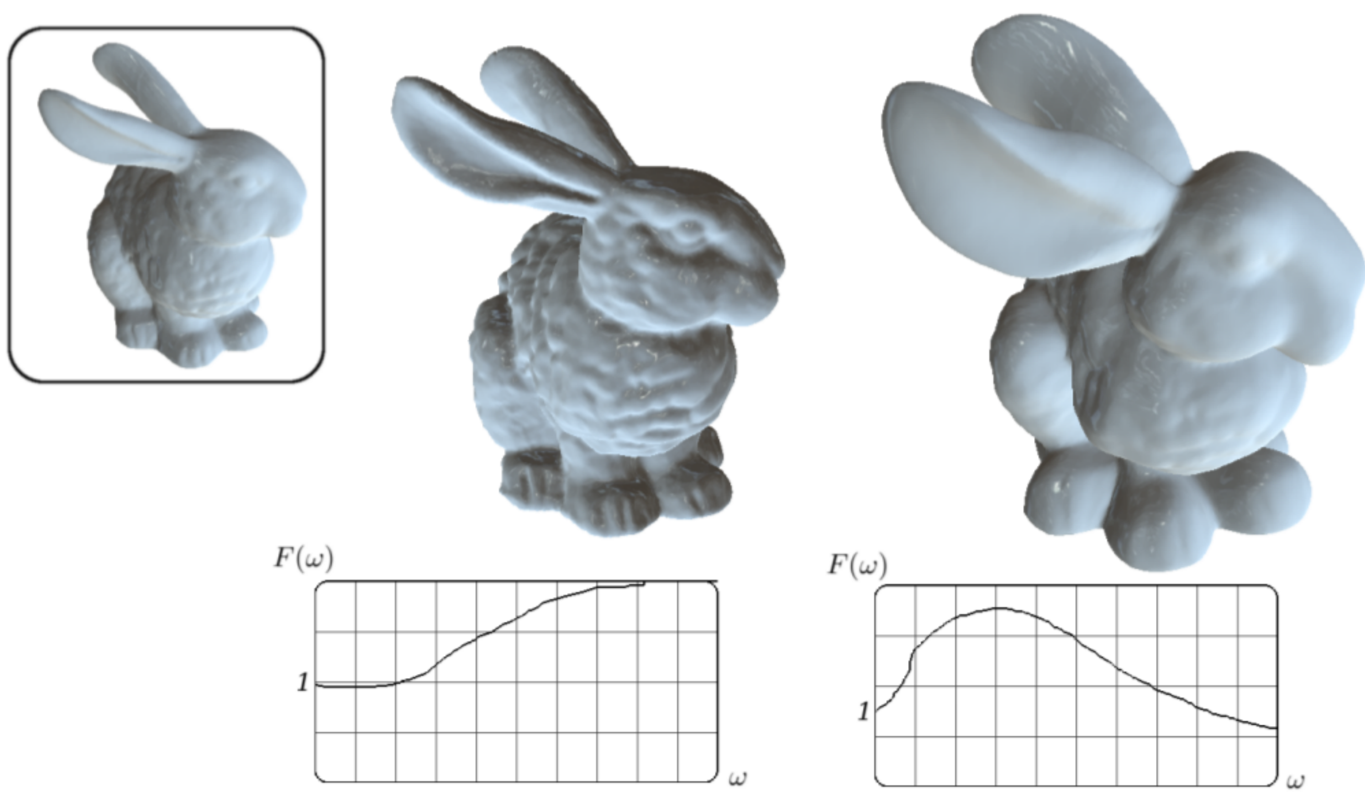
Filter h can be applied on the frequencies: $h \circ f = \sum_i h(F_i) \bar{\psi}_i$

Filtering

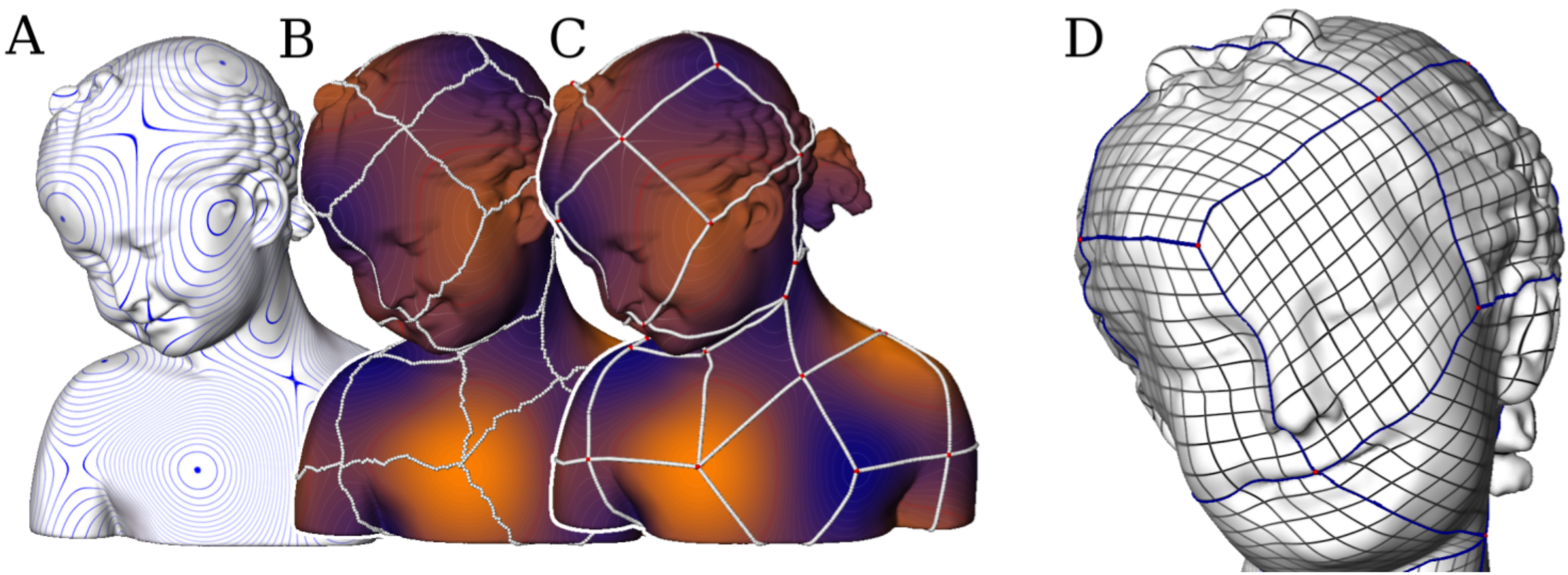
Given a function f on the vertices

$F_i := \bar{\psi}_i^T . f = \sum_{v_j} \bar{\psi}_i(v_j) f(v_j)$ is its i^th frequency.

Filter h can be applied on the frequencies: $h \circ f = \sum_i h(F_i) \bar{\psi}_i$



Quad meshing



Shape retrieval

Heat diffusion

$\{\psi_i\}_i$ is a good basis for heat diffusion :

$$\begin{cases} \partial_t u(x,t) = \nabla_x^2 u(x,t) \\ u(x,0) = u_0(x) \end{cases}$$

$$u(x,t) = \sum_i \alpha_i(t)\psi_i(x)$$ (decompose solution on basis)

$$\partial_t u(x,t) = \nabla_x^2 u(x,t) \longrightarrow \sum_i \dot{\alpha}_i(t)\psi_i = \sum_i -\lambda_i \alpha_i(t)\psi_i$$

$$\dot{\alpha}_i(t) + \lambda_i \alpha_i(t) = 0 \longrightarrow \alpha_i(t) = \alpha_i(0)\exp(-\lambda_i t)$$

and $\alpha_i(0) = \int_x \psi_i(x) u_0(x)\, dx$

$$\longrightarrow u(k,t) = \int_x \sum_i \psi_i(x)\psi_i(k)\exp(-\lambda_i t)\, u_0(x)$$ (in the continuous setting)

→ Closed-form solution

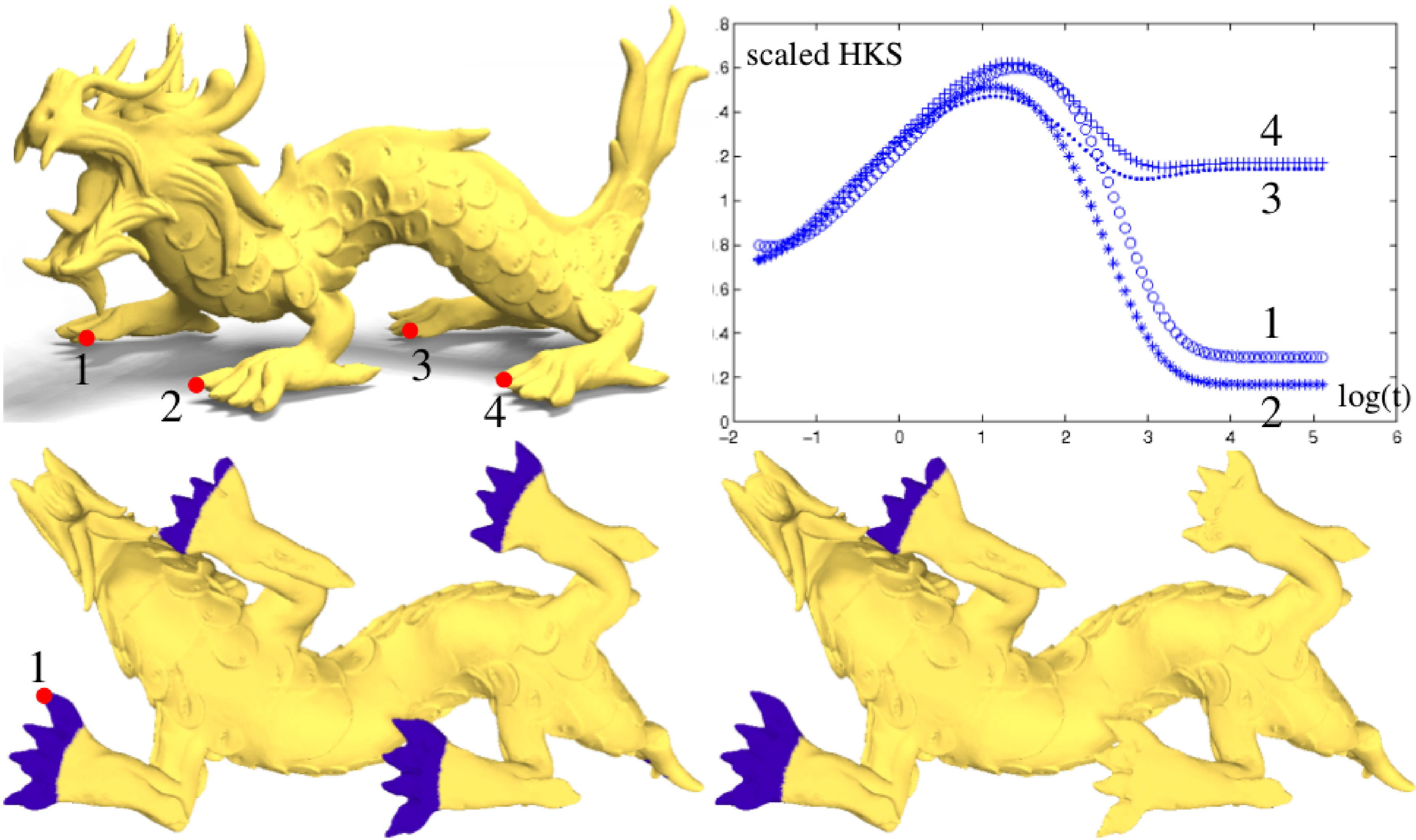
Heat diffusion

$h_t(j,k) = \sum_i \psi_i(j)\,\psi_i(k)\exp(-\lambda_i t)$: heat kernel at (j,k)

$\{h_t(j,j)\}_t$: multi-scale signature of vertex j

t = 1 t = 100 t = 10 000

Heat kernel signature

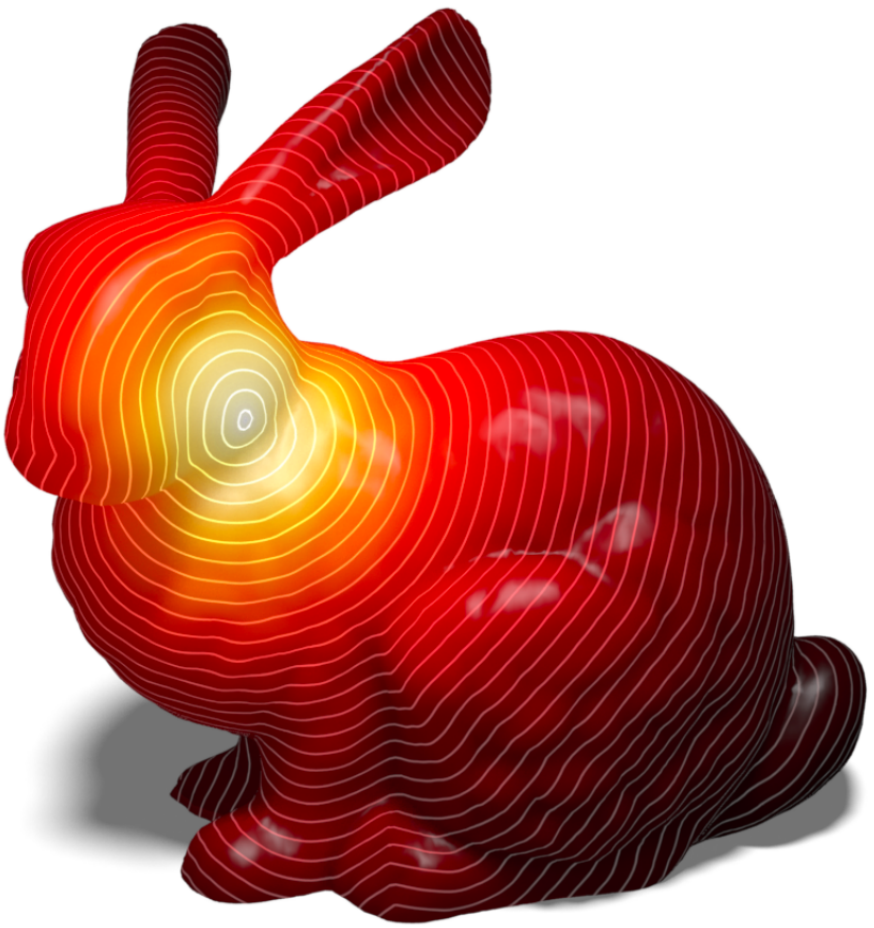


Geodesics in Heat

Link with :

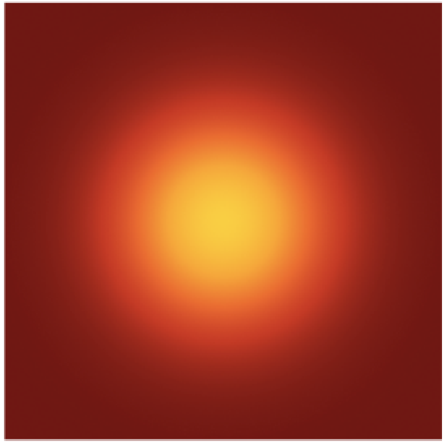
- Spectral properties
- Physics (heat)

Different from front-propagation approaches



[Crane et al. 2008] Geodesics in Heat:A New Approach to Computing Distance Based on Heat Flow

Geodesics in Heat



u

$$\partial_t u = \nabla^2 u$$

$$(u_t - u_0)/t = \nabla^2 u_t$$

$$u_t - t\nabla^2 u_t = u_0$$

$$(id - t\nabla^2) u_t = u_0$$

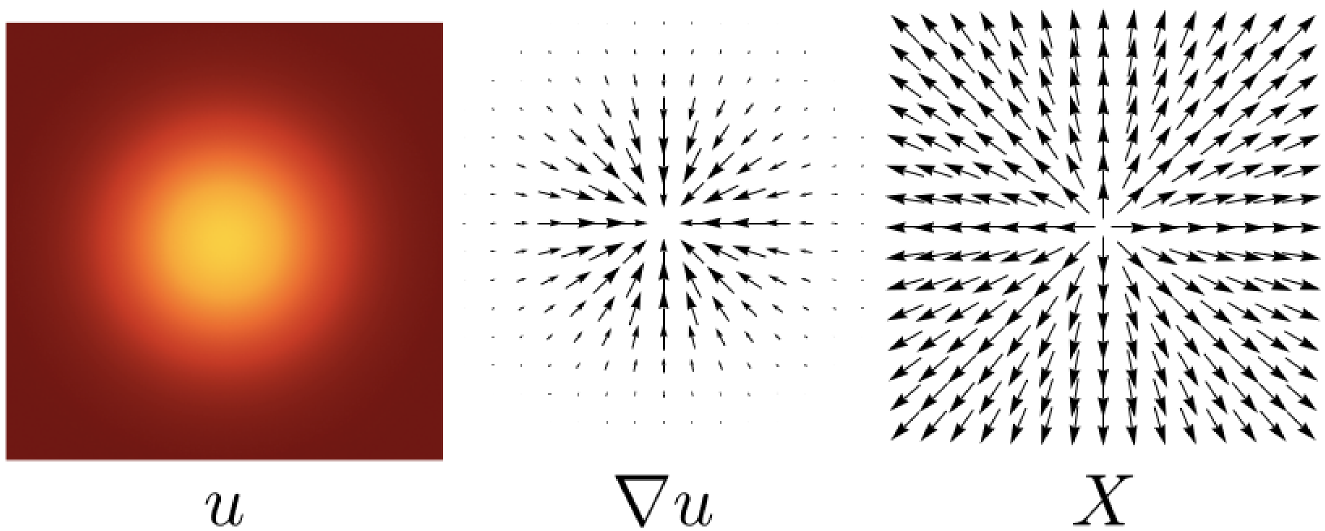
Linear system

Algorithm 1 The Heat Method

I. Integrate the heat flow $\dot{u} = \Delta u$ for some fixed time t.

[Crane et al. 2008] Geodesics in Heat:A New Approach to Computing Distance Based on Heat Flow

Geodesics in Heat



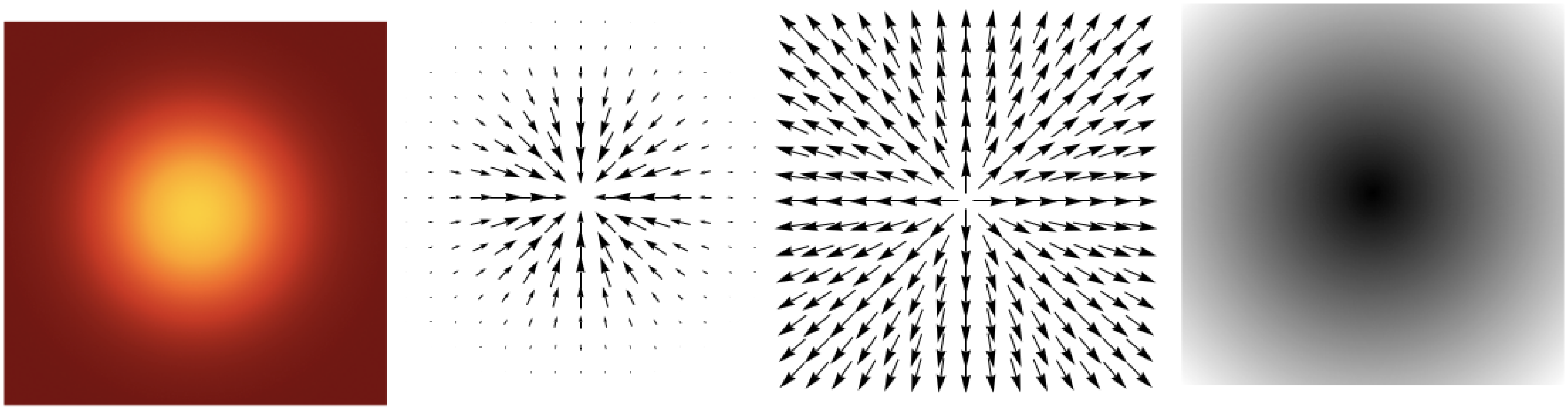
Algorithm 1 The Heat Method

I. Integrate the heat flow $\dot{u} = \Delta u$ for some fixed time t.

II. Evaluate the vector field $X = -\nabla u / |\nabla u|$.

[Crane et al. 2008] Geodesics in Heat:A New Approach to Computing Distance Based on Heat Flow

Geodesics in Heat



u ∇u X ϕ

Algorithm 1 The Heat Method

I. Integrate the heat flow $\dot{u} = \Delta u$ for some fixed time t.

II. Evaluate the vector field $X = -\nabla u / |\nabla u|$.

III. Solve the Poisson equation $\Delta \phi = \nabla \cdot X$.

Linear system

[Crane et al. 2008] Geodesics in Heat:A New Approach to Computing Distance Based on Heat Flow

Geodesics in Heat

Step I $\left(id - t\nabla^2\right)u_t = u_0$

Linear system :
can be prefactored indep of u_0

Step II $\vec{X} = -\nabla u_t / \|\nabla u_t\|$

straightforward

Step III $\nabla^2 \phi = \nabla . \vec{X}$

Linear system :
can be prefactored indep of u_0

[Crane et al. 2008] Geodesics in Heat:A New Approach to Computing Distance Based on Heat Flow

Geodesics in Heat

Laplacian operators have been studied for general polygonal meshes and pointsets :

[Crane et al. 2008] Geodesics in Heat:A New Approach to Computing Distance Based on Heat Flow

Geodesics in Heat : value of t ?

source

t small

t big

(provides smoother approx of geodesics)

[Crane et al. 2008] Geodesics in Heat:A New Approach to Computing Distance Based on Heat Flow

?