Geodesic Data Processing

Gabriel Peyré
CEREMADE, Université Paris-Dauphine

www.numerical-tours.com
Local vs. Global Processing

Local Processing

Differential Computations

Surface filtering

Fourier on Meshes

Global Processing

Geodesic Computations

Front Propagation on Meshes

Surface Remeshing
Overview

- **Metrics and Riemannian Surfaces.**

- Geodesic Computation - Iterative Scheme

- Geodesic Computation - Fast Marching

- Shape Recognition with Geodesic Statistics

- Geodesic Meshing
Parametric Surfaces

Parameterized surface: \( u \in \mathbb{R}^2 \mapsto \varphi(u) \in M. \)
Parametric Surfaces

Parameterized surface: \( u \in \mathbb{R}^2 \mapsto \varphi(u) \in \mathcal{M} \).

Curve in parameter domain: \( t \in [0, 1] \mapsto \gamma(t) \in \mathcal{D} \).
Parametric Surfaces

Parameterized surface: $u \in \mathbb{R}^2 \mapsto \varphi(u) \in \mathcal{M}$.

Curve in parameter domain: $t \in [0, 1] \mapsto \gamma(t) \in \mathcal{D}$.

Geometric realization: $\tilde{\gamma}(t) \overset{\text{def.}}{=} \varphi(\gamma(t)) \in \mathcal{M}$. 
Parametric Surfaces

Parameterized surface: $u \in \mathbb{R}^2 \mapsto \varphi(u) \in \mathcal{M}$.

Curve in parameter domain: $t \in [0, 1] \mapsto \gamma(t) \in \mathcal{D}$.

Geometric realization: $\bar{\gamma}(t) \overset{\text{def.}}{=} \varphi(\gamma(t)) \in \mathcal{M}$.

For an embedded manifold $\mathcal{M} \subset \mathbb{R}^n$:

First fundamental form: $I_\varphi = \left( \left\langle \frac{\partial \varphi}{\partial u_i}, \frac{\partial \varphi}{\partial u_j} \right\rangle \right)_{i,j=1,2}$.

Length of a curve

$L(\gamma) \overset{\text{def.}}{=} \int_0^1 \| \bar{\gamma}'(t) \| dt = \int_0^1 \sqrt{\gamma'(t) I_{\gamma(t)} \gamma'(t)} dt$. 
Isometric and Conformal

Surface not homeomorphic to a disk:

$\mathcal{M}$ is locally isometric to the plane: $I_\varphi = \text{Id}$.

*Exemple*: $\mathcal{M} = \text{cylinder}$. 
Isometric and Conformal

Surface not homeomorphic to a disk:

\[ M \text{ is locally isometric to the plane: } I_\varphi = \text{Id.} \]

\textit{Exemple: } \( M = \text{cylinder.} \)

\( \varphi \) is conformal: \( I_\varphi(u) = \lambda(u)\text{Id.} \)

\textit{Exemple: } stereographic mapping plane\( \leftrightarrow \)sphere.
Riemannian Manifold

*Riemannian manifold:* \( \mathcal{M} \subset \mathbb{R}^n \) (locally)

*Riemannian metric:* \( H(x) \in \mathbb{R}^{n \times n} \), symmetric, positive definite.

Length of a curve \( \gamma(t) \in \mathcal{M} \):
\[
L(\gamma) \overset{\text{def.}}{=} \int_0^1 \sqrt{\gamma'(t)^T H(\gamma(t)) \gamma'(t)} \, dt.
\]
Riemannian Manifold

Riemannian manifold: $\mathcal{M} \subset \mathbb{R}^n$ (locally)

Riemannian metric: $H(x) \in \mathbb{R}^{n \times n}$, symmetric, positive definite.

Length of a curve $\gamma(t) \in \mathcal{M}$: $L(\gamma) \overset{\text{def.}}{=} \int_0^1 \sqrt{\gamma'(t)^T H(\gamma(t)) \gamma'(t)} \, dt$.

Euclidean space: $\mathcal{M} = \mathbb{R}^n$, $H(x) = \text{Id}_n$. 

$W(x)$
Riemannian Manifold

Riemannian manifold: $\mathcal{M} \subset \mathbb{R}^n$ (locally)

Riemannian metric: $H(x) \in \mathbb{R}^{n \times n}$, symmetric, positive definite.

Length of a curve $\gamma(t) \in \mathcal{M}$: $L(\gamma) \overset{\text{def.}}{=} \int_0^1 \sqrt{\gamma'(t)^T H(\gamma(t)) \gamma'(t)} \, dt$.

Euclidean space: $\mathcal{M} = \mathbb{R}^n$, $H(x) = \text{Id}_n$.

2-D shape: $\mathcal{M} \subset \mathbb{R}^2$, $H(x) = \text{Id}_2$. 
Riemannian Manifold

Riemannian manifold: $\mathcal{M} \subset \mathbb{R}^n$ (locally)

Riemannian metric: $H(x) \in \mathbb{R}^{n \times n}$, symmetric, positive definite.

Length of a curve $\gamma(t) \in \mathcal{M}$: $L(\gamma) \overset{\text{def.}}{=} \int_0^1 \sqrt{(\gamma'(t))^T H(\gamma(t)) \gamma'(t)} \, dt$.

Euclidean space: $\mathcal{M} = \mathbb{R}^n$, $H(x) = \text{Id}_n$.

2-D shape: $\mathcal{M} \subset \mathbb{R}^2$, $H(x) = \text{Id}_2$.

Isotropic metric: $H(x) = W(x)^2 \text{Id}_n$. 

$W(x)$
Riemannian Manifold

*Riemannian manifold*: $\mathcal{M} \subset \mathbb{R}^n$ (locally)

*Riemannian metric*: $H(x) \in \mathbb{R}^{n \times n}$, symmetric, positive definite.

Length of a curve $\gamma(t) \in \mathcal{M}$: $L(\gamma) \overset{\text{def.}}{=} \int_0^1 \sqrt{\gamma'(t)^T H(\gamma(t)) \gamma'(t)} \, dt$.

*Euclidean space*: $\mathcal{M} = \mathbb{R}^n$, $H(x) = \text{Id}_n$.

*2-D shape*: $\mathcal{M} \subset \mathbb{R}^2$, $H(x) = \text{Id}_2$.

*Isotropic metric*: $H(x) = W(x)^2 \text{Id}_n$.

*Image processing*: image $I$, $W(x)^2 = (\varepsilon + \|\nabla I(x)\|)^{-1}$. 

\[ W(x) \]
**Riemannian Manifold**

*Riemannian manifold:* $\mathcal{M} \subset \mathbb{R}^n$ (locally)

*Riemannian metric:* $H(x) \in \mathbb{R}^{n \times n}$, symmetric, positive definite.

Length of a curve $\gamma(t) \in \mathcal{M}$:

\[ L(\gamma) \overset{\text{def.}}{=} \int_0^1 \sqrt{\gamma'(t)^T H(\gamma(t)) \gamma'(t)} \, dt. \]

**Euclidean space:** $\mathcal{M} = \mathbb{R}^n$, $H(x) = \text{Id}_n$.

**2-D shape:** $\mathcal{M} \subset \mathbb{R}^2$, $H(x) = \text{Id}_2$.

**Isotropic metric:** $H(x) = W(x)^2 \text{Id}_n$.

**Image processing:** image $I$, $W(x)^2 = (\varepsilon + \|\nabla I(x)\|)^{-1}$.

**Parametric surface:** $H(x) = I_x$ (1\textsuperscript{st} fundamental form).
Riemannian Manifold

Riemannian manifold: $\mathcal{M} \subset \mathbb{R}^n$ (locally)

Riemannian metric: $H(x) \in \mathbb{R}^{n \times n}$, symmetric, positive definite.

Length of a curve $\gamma(t) \in \mathcal{M}$: $L(\gamma) \overset{\text{def.}}{=} \int_0^1 \sqrt{(\gamma(t))^T H(\gamma(t)) \gamma(t)} \, dt$.

Euclidean space: $\mathcal{M} = \mathbb{R}^n$, $H(x) = \text{Id}_n$.

2-D shape: $\mathcal{M} \subset \mathbb{R}^2$, $H(x) = \text{Id}_2$.

Isotropic metric: $H(x) = W(x)^2 \text{Id}_n$.

Image processing: image $I$, $W(x)^2 = (\varepsilon + \|\nabla I(x)\|)^{-1}$.

Parametric surface: $H(x) = I_x$ ($1^{\text{st}}$ fundamental form).

DTI imaging: $\mathcal{M} = [0, 1]^3$, $H(x)$ = diffusion tensor.
Geodesic Distances

Geodesic distance metric over $\mathcal{M} \subset \mathbb{R}^n$

$$d_\mathcal{M}(x, y) = \min_{\gamma(0)=x, \gamma(1)=y} L(\gamma)$$

Geodesic curve: $\gamma(t)$ such that $L(\gamma) = d_\mathcal{M}(x, y)$.

Distance map to a starting point $x_0 \in \mathcal{M}$: $U_{x_0}(x) \overset{\text{def.}}{=} d_\mathcal{M}(x_0, x)$. 
Geodesic Distances

Geodesic distance metric over $\mathcal{M} \subset \mathbb{R}^n$

$$d_\mathcal{M}(x, y) = \min_{\gamma(0)=x, \gamma(1)=y} L(\gamma)$$

Geodesic curve: $\gamma(t)$ such that $L(\gamma) = d_\mathcal{M}(x, y)$.

Distance map to a starting point $x_0 \in \mathcal{M}$: $U_{x_0}(x) \overset{\text{def.}}{=} d_\mathcal{M}(x_0, x)$. 

Euclidean

metric

godesics

$U_{x_0}$
Geodesic Distances

Geodesic distance metric over $\mathcal{M} \subset \mathbb{R}^n$

$$d_\mathcal{M}(x, y) = \min_{\gamma(0)=x, \gamma(1)=y} L(\gamma)$$

Geodesic curve: $\gamma(t)$ such that $L(\gamma) = d_\mathcal{M}(x, y)$.

Distance map to a starting point $x_0 \in \mathcal{M}$: $U_{x_0}(x) \overset{\text{def.}}{=} d_\mathcal{M}(x_0, x)$. 

Euclidean

Shape
Geodesic Distances

Geodesic distance metric over $\mathcal{M} \subset \mathbb{R}^n$

$$d_{\mathcal{M}}(x, y) = \min_{\gamma(0)=x, \gamma(1)=y} L(\gamma)$$

Geodesic curve: $\gamma(t)$ such that $L(\gamma) = d_{\mathcal{M}}(x, y)$.

Distance map to a starting point $x_0 \in \mathcal{M}$: $U_{x_0}(x) \overset{\text{def.}}{=} d_{\mathcal{M}}(x_0, x)$. 
Geodesic Distances

Geodesic distance metric over $\mathcal{M} \subset \mathbb{R}^n$

$$d_{\mathcal{M}}(x, y) = \min_{\gamma(0)=x, \gamma(1)=y} L(\gamma)$$

Geodesic curve: $\gamma(t)$ such that $L(\gamma) = d_{\mathcal{M}}(x, y)$.

Distance map to a starting point $x_0 \in \mathcal{M}$: $U_{x_0}(x) \overset{\text{def.}}{=} d_{\mathcal{M}}(x_0, x)$. 

![Geodesic Distances Diagram](image-url)
Geodesic Distances

Geodesic distance metric over \( \mathcal{M} \subset \mathbb{R}^n \)

\[
d_{\mathcal{M}}(x, y) = \min_{\gamma(0)=x, \gamma(1)=y} L(\gamma)
\]

Geodesic curve: \( \gamma(t) \) such that \( L(\gamma) = d_{\mathcal{M}}(x, y) \).

Distance map to a starting point \( x_0 \in \mathcal{M} \): \( U_{x_0}(x) \overset{\text{def.}}{=} d_{\mathcal{M}}(x_0, x) \).
Anisotropy and Geodesics

Tensor eigen-decomposition:

\[ H(x) = \lambda_1(x)e_1(x)e_1(x)^T + \lambda_2(x)e_2(x)e_2(x)^T \]

with \( 0 < \lambda_1 \leq \lambda_2 \),

\[ \{ \eta \mid \eta^*H(x)\eta \leq 1 \} \]

\[ \lambda_2(x)^{-\frac{1}{2}} \]

\[ e_2(x) \]

\[ \lambda_1(x)^{-\frac{1}{2}} \]

\[ e_1(x) \]
Anisotropy and Geodesics

Tensor eigen-decomposition:

\[ H(x) = \lambda_1(x)e_1(x)e_1(x)^T + \lambda_2(x)e_2(x)e_2(x)^T \]

with \( 0 < \lambda_1 \leq \lambda_2 \),

\[ \{ \eta \mid \eta^* H(x) \eta \leq 1 \} \]

Geodesics tend to follow \( e_1(x) \).
Anisotropy and Geodesics

Tensor eigen-decomposition:

$$H(x) = \lambda_1(x)e_1(x)e_1(x)^T + \lambda_2(x)e_2(x)e_2(x)^T \quad \text{with} \quad 0 < \lambda_1 \leq \lambda_2,$$

$$\{\eta \mid \eta^*H(x)\eta \leq 1\}$$

Geodesics tend to follow $e_1(x)$.

Local anisotropy of the metric: $\alpha(x) = \frac{\lambda_1(x) - \lambda_2(x)}{\lambda_1(x) + \lambda_2(x)} \in [0, 1]$
Isotropic Metric Design

Image-based potential: \( H(x) = W(x)^2 \text{Id}_2, \, W(x) = (\varepsilon + |f(x) - c|)^\alpha \)
Isotropic Metric Design

**Image-based potential:** \( H(x) = W(x)^2 \text{Id}_2 \), \( W(x) = (\varepsilon + |f(x) - c|)^\alpha \)

*Image* \( f \)  *Metric* \( W(x) \)  *Distance* \( U_{x_0}(x) \)  *Geodesic curve* \( \gamma(t) \)

**Gradient-based potential:** \( W(x) = (\varepsilon + \|\nabla_x f\|)^{-\alpha} \)

*Image* \( f \)  *Metric* \( W(x) \)  *\( U_{\{x_0,x_1\}} \)  *Geodesics*
Isotropic Metric Design: Vessels

Remove background: \( \tilde{f} = G_\sigma \ast f - f \), \( \sigma \approx \)vessel width.

\[ W = (\varepsilon + \max(\tilde{f}, 0))^{-\alpha} \]
Isotropic Metric Design: Vessels

Remove background: $\tilde{f} = G_\sigma \ast f - f$, $\sigma \approx$ vessel width.

3D Volumetric datasets:
Overview

- Metrics and Riemannian Surfaces.

- **Geodesic Computation - Iterative Scheme**

- Geodesic Computation - Fast Marching

- Shape Recognition with Geodesic Statistics

- Geodesic Meshing
Eikonal Equation and Viscosity Solution

Distance map: \( U(x) = d(x_0, x) \)

Theorem: \( U \) is the unique viscosity solution of
\[
\| \nabla U(x) \|_{H(x)^{-1}} = 1 \quad \text{with} \quad U(x_0) = 0
\]
where \( \| v \|_A = \sqrt{v^* A v} \)
Eikonal Equation and Viscosity Solution

Distance map: \( U(x) = d(x_0, x) \)

\[
\text{Theorem: } U \text{ is the unique viscosity solution of }
\begin{align*}
\| \nabla U(x) \|_{H(x)^{-1}} &= 1 \\
\text{with } U(x_0) &= 0
\end{align*}
\]

where \( \| v \|_A = \sqrt{v^* Av} \)

Geodesic curve \( \gamma \) between \( x_1 \) and \( x_0 \) solves

\[
\gamma'(t) = -\eta_t H(\gamma(t))^{-1} \nabla U_{x_0}(\gamma(t)) \quad \text{with} \quad \begin{align*}
\gamma(0) &= x_1 \\
\eta_t &> 0
\end{align*}
\]
Eikonal Equation and Viscosity Solution

Distance map: \( U(x) = d(x_0, x) \)

**Theorem:** \( U \) is the unique viscosity solution of
\[
\| \nabla U(x) \|_{H(x)^{-1}} = 1 \quad \text{with} \quad U(x_0) = 0
\]

where \( \| v \|_A = \sqrt{v^* A v} \)

Geodesic curve \( \gamma \) between \( x_1 \) and \( x_0 \) solves
\[
\gamma'(t) = -\eta_t H(\gamma(t))^{-1} \nabla U_{x_0}(\gamma(t)) \quad \text{with} \quad \begin{cases} \gamma(0) = x_1 \\ \eta_t > 0 \end{cases}
\]

**Example:** isotropic metric \( H(x) = W(x)^2 \text{Id}_n \),
\[
\| \nabla U(x) \| = W(x) \quad \text{and} \quad \gamma'(t) = -\eta_t \nabla U(\gamma(t))
\]
**Simplified Proof**

\[ U(x) = \min_{\gamma: x_0 \to x} L(\gamma) = \int_0^1 \sqrt{\langle H(\gamma(t))\gamma'(t), \gamma'(t) \rangle} \, dt \]

\[ V \text{ solving } \left\{ \begin{array}{l} \| \nabla V(x) \|^2_{H^{-1}} = \langle H^{-1}(x)\nabla V(x), \nabla V(x) \rangle = 1, \\ V(x_0) = 0. \end{array} \right. \]
Simplified Proof

\[ U(x) = \min_{\gamma : x_0 \to x} L(\gamma) = \int_0^1 \sqrt{\langle H(\gamma(t))\gamma'(t), \gamma'(t) \rangle} dt \]

V solving \( \begin{cases} \| \nabla V(x) \|^2_{H^{-1}} = \langle H^{-1}(x) \nabla V(x), \nabla V(x) \rangle = 1, \\ V(x_0) = 0. \end{cases} \)

\[ U \geq V \]

Let \( \gamma : x_0 \to x \) be any smooth curve.

If \( V \) is smooth on \( \gamma \):

\[ \langle \gamma', \nabla V \rangle = \langle H^{1/2}\gamma', H^{-1/2}\nabla V \rangle \leq \| H^{1/2}\gamma' \| \| H^{-1/2}\nabla V \| = 1 \]

C.S.
Simplified Proof

\[ U(x) = \min_{\gamma:x_0 \to x} L(\gamma) = \int_0^1 \sqrt{\langle H(\gamma(t))\gamma'(t), \gamma'(t) \rangle} \, dt \]

Let \( \gamma: x_0 \to x \) be any smooth curve.

If \( V \) is smooth on \( \gamma \):

\[ \langle \gamma', \nabla V \rangle = \langle H^{1/2}\gamma', H^{-1/2}\nabla V \rangle \leq \| H^{1/2}\gamma' \| \| H^{-1/2}\nabla V \| = 1 \]

\[ L(\gamma) = \int_0^1 \| H^{1/2}\gamma' \| \geq \int_0^1 \langle \gamma', \nabla V \rangle = V(\gamma(1)) - V(\gamma(0)) = V(x) \]

\[ \implies U(x) = \min_{\gamma} L(\gamma) \geq V(x) \]
Let $x$ be arbitrary.
Define: $\eta'(t) = -H^{-1}(\eta(t))\nabla V(\eta(t))$
$\eta(0) = x$
Simplified Proof (cont.)

Let \( x \) be arbitrary.

Define: \[ \eta'(t) = -H^{-1}(\eta(t))\nabla V(\eta(t)) \]
\[ \eta(0) = x \]

If \( V \) is smooth on \( \gamma([0, t_{\text{max}}]) \), then

\[ \frac{dV(\eta(t))}{dt} = \langle \eta'(t), \nabla V(\eta(t)) \rangle = -1 \]

\[ \implies \gamma(t_{\text{max}}) = x_0 \]
Let $x$ be arbitrary.

Define: \[ \eta'(t) = -H^{-1}(\eta(t))\nabla V(\eta(t)) \]
\[ \eta(0) = x \]

If $V$ is smooth on $\gamma([0, t_{\text{max}}])$, then

\[ \frac{dV(\eta(t))}{dt} = \langle \eta'(t), \nabla V(\eta(t)) \rangle = -1 \]

\[ \Rightarrow \quad \gamma(t_{\text{max}}) = x_0 \]

One has:

\[ \langle H\eta', \eta' \rangle = \langle H^{-1}\nabla V, \nabla V \rangle = 1 \]

\[ U(x) \leq L(\eta) = \int_0^{t_{\text{max}}} \sqrt{\langle H\eta', \eta' \rangle} = \int_0^{t_{\text{max}}} \langle H\eta', \eta' \rangle = 1 \]

\[ = -\int_0^{t_{\text{max}}} \langle \gamma', \nabla V \rangle = -V(\gamma(t_{\text{max}})) + V(\gamma(0)) = V(x) \]

\[ \equiv 0 \]
Discretization

Control (derivative-free) formulation:

\[ U(x) = d(x_0, x) \]

is the unique solution of

\[ U(x) = \Gamma(U)(x) = \min_{y \in B(x)} U(y) + d(x, y) \]
Discretization

Control (derivative-free) formulation:

\[ U(x) = d(x_0, x) \] is the unique solution of

\[ U(x) = \Gamma(U)(x) = \min_{y \in B(x)} U(y) + d(x, y) \]

Manifold discretization: triangular mesh.

\[ U \] discretization: linear finite elements.

\[ H \] discretization: constant on each triangle.
Discretization

Control (derivative-free) formulation:

\[ U(x) = d(x_0, x) \] is the unique solution of

\[ U(x) = \Gamma(U)(x) = \min_{y \in B(x)} U(y) + d(x, y) \]

Manifold discretization: triangular mesh.

\( U \) discretization: linear finite elements.

\( H \) discretization: constant on each triangle.

\[ U_i = \Gamma(U)_i = \min_{f=(i,j,k)} V_{i,j,k} \]

\[ V_{i,j,k} = \min_{0 \leq t \leq 1} tU_j + (1 - t)U_k \]

\[ + \|tx_j + (1 - t)x_k - x_i\|_{H_{ijk}} \]

→ explicit solution (solving quadratic equation).

→ on regular grid: equivalent to upwind FD.
Update Step on a triangulation

$$\Gamma(U)_i = \min_{f=(i,j,k)} V_{i,j,k}$$

Discrete Eikonal equation:

$$V_{i,j,k} = \min_{0 \leq t \leq 1} tu_j + (1 - t)u_k + \|tx_j + (1 - t)x_k - x_i\| H_{i,j,k}$$
Update Step on a triangulation

\[ \Gamma(U)_i = \min_{f=(i,j,k)} V_{i,j,k} \]

Discrete Eikonal equation:

\[ V_{i,j,k} = \min_{0 \leq t \leq 1} tU_j + (1 - t)U_k + \|tx_j + (1 - t)x_k - x_i\|_{H_{ij,k}} \]

Distance function in \((i, j, k)\):

\[ U(x) = \langle x - x_i, g \rangle + d \]

Unknowns: gradient \(= V_{i,j,k} \)
Update Step on a triangulation

\[ \Gamma(U)_i = \min_{f = (i, j, k)} V_{i, j, k} \]

Discrete Eikonal equation:

\[ V_{i, j, k} = \min_{0 \leq t \leq 1} tU_j + (1 - t)U_k + \|tx_j + (1 - t)x_k - x_i\|_{H_{i,j,k}} \]

Distance function in \((i, j, k)\):

\[ U(x) = \langle x - x_i, g \rangle + d \]

Unkowns: gradient \[\Rightarrow V_{i, j, k}\]

Notations:

\[ H_{i,j,k} = w^2 \text{Id}_3 \quad \text{(for simplifity)} \]

\[ X = (x_j - x_i, x_k - x_i) \in \mathbb{R}^{d \times 2} \]

\[ u = (U_j, U_k) \in \mathbb{R}^2 \]

\[ \mathbb{I} = (1, 1) \in \mathbb{R}^2 \]

\[ S = (X^*X)^{-1} \in \mathbb{R}^{2 \times 2} \]
Find $g = X\alpha$, $\alpha \in \mathbb{R}^2$ and $d = V_{i,j,k}$.

$$X^* g + d\mathbb{I} = u \implies \alpha = S(u - d\mathbb{I})$$
Update Step on a triangulation (cont.)

Find $g = X\alpha, \alpha \in \mathbb{R}^2$ and $d = V_{i,j,k}$.

$$X^*g + dI = u \implies \alpha = S(u - dI)$$

Discrete Eikonal equation:

$$\iff \|\nabla U(x_i)\| = \|g\| = w$$
Update Step on a triangulation (cont.)

Find $g = X\alpha, \alpha \in \mathbb{R}^2$ and $d = V_{i,j,k}$.

$$X^*g + d\| = u \implies \alpha = S(u - d\|)$$

**Discrete Eikonal equation:**

$$\iff \|\nabla U(x_i)\| = \|g\| = w$$

**Quadratic equation:**

$$\|XS(u - d\|)\|^2 = w^2$$

$$\implies d^2 - 2bd + c = 0$$

\[
\begin{align*}
    a &= \langle S\|, \| \rangle \\
    b &= \langle S\|, u \rangle \\
    c &= \langle Su, u \rangle - w^2
\end{align*}
\]
Update Step on a triangulation (cont.)

Find \( g = X\alpha, \alpha \in \mathbb{R}^2 \) and \( d = V_{i,j,k} \).

\[
X^* g + d\mathbb{I} = u \implies \alpha = S(u - d\mathbb{I})
\]

**Discrete Eikonal equation:**

\[
\Leftrightarrow \| \nabla U(x_i) \| = \| g \| = w
\]

**Quadratic equation:**

\[
\| XS(u - d\mathbb{I}) \|^2 = w^2
\]

\[
\implies d^2 - 2bd + c = 0
\]

Admissible solution:

\[
d = \frac{b + \sqrt{\delta}}{a}, \quad \delta = b^2 - ac
\]

\[
\Gamma(u_i) = \begin{cases} 
d & \text{if } \alpha \leq 0 \\
\min(d_j, d_k) & \text{otherwise.}
\end{cases}
\]

\[
d_j = U_j + W_i \| x_i - x_j \|
\]
Numerical Schemes

Fixed point equation: \( U = \Gamma(U) \)

\( \Gamma \) is monotone: \( U \preceq V \implies \Gamma(U) \preceq \Gamma(V) \)

Iterative schemes: \( U^{(0)} = 0, U^{(\ell+1)} = \Gamma(U^{(\ell)}) \)

\( \implies U^{(\ell+1)} \succeq U^{(\ell)} \leq C < +\infty \)

\( U^{(\ell)} \to U \) solving \( \Gamma(U) = U \)
Numerical Schemes

Fixed point equation: \( U = \Gamma(U) \)

\( \Gamma \) is monotone: \( U \leq V \implies \Gamma(U) \leq \Gamma(V) \)

Iterative schemes: \( U^{(0)} = 0, U^{(\ell+1)} = \Gamma(U^{(\ell)}) \)

\( \implies U^{(\ell+1)} \geq U^{(\ell)} \leq C < +\infty \)

\( U^{(\ell)} \to U \) solving \( \Gamma(U) = U \)

Minimal path extraction:

\[ \gamma^{(\ell+1)} = \gamma^{(\ell)} - \eta \ell H(\gamma^{(\ell)})^{-1} \nabla U(\gamma^{(\ell)}) \]
Numerical Examples on Meshes
Discretization Errors

For a mesh with $N$ points: $U^{[N]} \in \mathbb{R}^N$ solution of $\Gamma(U^{[N]}) = U^{[N]}$

Continuous geodesic distance $U(x)$.

Linear interpolation: $\tilde{U}^{[N]}(x) = \sum_i U_i^{[N]} \varphi_i(x)$

Uniform convergence: $\|\tilde{U}^{[N]} - U\|_{\infty} \xrightarrow{N \to +\infty} 0$
Discretization Errors

For a mesh with $N$ points: $U^N \in \mathbb{R}^N$ solution of $\Gamma(U^N) = U^N$

Continuous geodesic distance $U(x)$.

Linear interpolation: $\tilde{U}^N(x) = \sum_i U_i^N \varphi_i(x)$

Uniform convergence: $\|\tilde{U}^N - U\|_\infty \xrightarrow{N \to +\infty} 0$

Numerical evaluation:

$$\frac{1}{N} \sum_i |U_i^N - U(x_i)|^2$$
Overview

• Metrics and Riemannian Surfaces.

• Geodesic Computation - Iterative Scheme

• **Geodesic Computation - Fast Marching**

• Shape Recognition with Geodesic Statistics

• Geodesic Meshing
Causal Updates

Causality condition: \( \forall j \sim i, \Gamma(U)_i \geq U_j \)

→ The value of \( U_i \) depends on \( \{U_j\}_j \) with \( U_j \leq U_i \).

→ Compute \( \Gamma(U)_i \) using an optimal ordering.

→ Front propagation, \( O(N \log(N)) \) operations.
Causal Updates

Causality condition: \( \forall j \sim i, \Gamma(U)_i \geq U_j \)

\( \rightarrow \) The value of \( U_i \) depends on \( \{U_j\}_j \) with \( U_j \leq U_i \).

\( \rightarrow \) Compute \( \Gamma(U)_i \) using an optimal ordering.

\( \rightarrow \) Front propagation, \( O(N \log(N)) \) operations.

Isotropic \( H(x) = W(x)^2 \text{Id}, \) square grid.

\[ u = \Gamma(U)_i \text{ is the solution of} \]
\[ \max(u - U_{i-1,j}, u - U_{i+1,j}, 0)^2 + \]
\[ \max(u - U_{i,j-1}, u - U_{i,j+1}, 0)^2 = h^2 W_{i,j}^2 \]

(upwind derivatives)
Causal Updates

Causality condition:  \( \forall j \sim i, \Gamma(U)_i \geq U_j \)

→ The value of \( U_i \) depends on \( \{U_j\}_j \) with \( U_j \leq U_i \).
→ Compute \( \Gamma(U)_i \) using an optimal ordering.
→ Front propagation, \( O(N \log(N)) \) operations.

Isotropic \( H(x) = W(x)^2 \text{Id} \), square grid.

\[ u = \Gamma(U)_i \text{ is the solution of } \]
\[ \max(u - U_{i-1,j}, u - U_{i+1,j}, 0)^2 + \]
\[ \max(u - U_{i,j-1}, u - U_{i,j+1}, 0)^2 = h^2 W_{i,j}^2 \]

(upwind derivatives)

Surface (first fundamental form) 
triangulation with no obtuse angles.
Front Propagation

Front $\partial F_t$, $F_t = \{ i \mid U_i \leq t \}$

State $S_i \in \{Computed, Front, Far\}$

Algorithm: Far $\rightarrow$ Front $\rightarrow$ Computed.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select <em>front</em> point 🟠 with minimum $U_i$</td>
</tr>
<tr>
<td>2</td>
<td>Move from <em>Front</em> ○ to <em>Computed</em> ⬤.</td>
</tr>
<tr>
<td>3</td>
<td>Update $U_j = \Gamma(U)_j$ for neighbors ⬤ and ○</td>
</tr>
</tbody>
</table>
Fast Marching on an Image
Fast Marching on Shapes and Surfaces
Volumetric Datasets
Propagation in 3D
Overview

- Metrics and Riemannian Surfaces.
- Geodesic Computation - Iterative Scheme
- Geodesic Computation - Fast Marching
- **Shape Recognition with Geodesic Statistics**
- Geodesic Meshing
Bending Invariant Recognition

Shape articulations:

[Zoopraxiscope, 1876]
Bending Invariant Recognition

Shape articulations:

[Zoopraxiscope, 1876]

Surface bendings:

[Elad, Kimmel, 2003]. [Bronstein et al., 2005].
2D Shapes

2D shape: connected, closed compact set $S \subset \mathbb{R}^2$. Piecewise-smooth boundary $\partial S$.

Geodesic distance in $S$ for uniform metric:

$$d_S(x, y) \overset{\text{def.}}{=} \min_{\gamma \in \mathcal{P}(x, y)} L(\gamma) \quad \text{where} \quad L(\gamma) \overset{\text{def.}}{=} \int_0^1 |\gamma'(t)| \, dt,$$
Distribution of Geodesic Distances

Distribution of distances

to a point $x$: $\{d_M(x, y)\}_{y \in \mathcal{M}}$
Distribution of Geodesic Distances

Distribution of distances to a point \( x \): \( \{d_\mathcal{M}(x, y)\}_{y \in \mathcal{M}} \)

Extract a statistical measure

\[
a_0(x) = \min_y d_\mathcal{M}(x, y).
\]

\[
a_1(x) = \text{median}_y d_\mathcal{M}(x, y).
\]

\[
a_2(x) = \max_y d_\mathcal{M}(x, y).
\]
Distribution of Geodesic Distances

Distribution of distances to a point $x$: \( \{d_M(x, y)\}_{y \in M} \)

Extract a statistical measure

\[ a_0(x) = \min_y d_M(x, y). \]
\[ a_1(x) = \text{median}_y d_M(x, y). \]
\[ a_2(x) = \max_y d_M(x, y). \]
Benging Invariant 2D Database

[Image - 369x461 to 807x665]

[Ling & Jacobs, PAMI 2007]

Our method
(min,med,max)

max only
[Ion et al. 2008]

→ State of the art retrieval rates on this database.
Perspective: Textured Shapes

Take into account a texture $f(x)$ on the shape.

Compute a saliency field $W(x)$, e.g. edge detector.

Compute weighted curve lengths: 
\[ L(\gamma) \overset{\text{def.}}{=} \int_{0}^{1} W(\gamma(t)) \|\gamma'(t)\| dt. \]
Overview

- Metrics and Riemannian Surfaces.
- Geodesic Computation - Iterative Scheme
- Geodesic Computation - Fast Marching
- Shape Recognition with Geodesic Statistics
- Geodesic Meshing
Triangulation \((\mathcal{V}, \mathcal{F})\): Vertices \(\mathcal{V} = \{v_i\}_{i=1}^M\). Faces \(\mathcal{F} \subset \{1, \ldots, M\}^3\).

Image approximation: \(f_M = \sum_{m=1}^{M} \lambda_m \varphi_m\)

\[
\lambda = \arg\min_{\mu} \| f - \sum_{m} \mu_m \varphi_m \|
\]

\(\varphi_m(v_i) = \delta_i^m\) is affine on each face of \(\mathcal{F}\).
Meshing Images, Shapes and Surfaces

Triangulation \((\mathcal{V}, \mathcal{F})\):
- Vertices \(\mathcal{V} = \{v_i\}_{i=1}^M\).
- Faces \(\mathcal{F} \subset \{1, \ldots, M\}^3\).

Image approximation:

\[
\lambda = \arg \min_{\mu} \|f - \sum_{m=1}^M \mu_m \varphi_m\|
\]

\[
\varphi_m(v_i) = \delta_i^m \text{ is affine on each face of } \mathcal{F}.
\]

There exists \((\mathcal{V}, \mathcal{F})\) such that \(\|f - f_M\| \leq C_f M^{-2}\)

Optimal \((\mathcal{V}, \mathcal{F})\): NP-hard.
Meshing Images, Shapes and Surfaces

Triangulation \((\mathcal{V}, \mathcal{F})\): Vertices \(\mathcal{V} = \{v_i\}_{i=1}^{M}\). Faces \(\mathcal{F} \subset \{1, \ldots, M\}^3\).

Image approximation: \(f_M = \sum_{m=1}^{M} \lambda_m \varphi_m\)

\[
\lambda = \arg\min_{\mu} \|f - \sum_{m} \mu_m \varphi_m\|
\]

\(\varphi_m(v_i) = \delta_i^m\) is affine on each face of \(\mathcal{F}\).

There exists \((\mathcal{V}, \mathcal{F})\) such that \(\|f - f_M\| \leq C_f M^{-2}\).

Optimal \((\mathcal{V}, \mathcal{F})\): NP-hard.

Domain meshing:

Conforming to complicated boundary.
Capturing PDE solutions:
Boundary layers, chocs ...
Riemannian Sizing Field

Sampling \( \{ x_i \}_{i \in I} \) of a manifold.

Distance conforming:
\[ \forall x_i \leftrightarrow x_j, \ d(x_i, x_j) \approx \varepsilon \]

Triangulation conforming:
\[ \Delta = ( x_i \leftrightarrow x_j \leftrightarrow x_k ) \subset \{ x \ \| x - x\Delta \|_{T(x\Delta)} \leq \eta \} \]

Building triangulation
\[ \iff \]
Ellipsoid packing
\[ \iff \]
Global integration of local sizing field
Geodesic Sampling

Sampling \( \{x_i\}_{i \in I} \) of a manifold.
Sampling \( \{x_i\}_{i \in I} \) of a manifold.

Farthest point algorithm: [Peyré, Cohen, 2006]

\[
x_{k+1} = \operatorname{argmax}_x \min_{0 \leq i \leq k} d(x_i, x)
\]
Sampling \( \{x_i\}_{i \in I} \) of a manifold.

Farthest point algorithm: [Peyré, Cohen, 2006]

\[
x_{k+1} = \arg\max_x \min_{0 \leq i \leq k} d(x_i, x)
\]

Geodesic Voronoi:

\[
C_i = \{x \mid \forall j \neq i, d(x_i, x) \leq d(x_j, x)\}
\]
Geodesic Sampling

Sampling \( \{x_i\}_{i \in I} \) of a manifold.

Farthest point algorithm:  [Peyré, Cohen, 2006]

\[
x_{k+1} = \arg\max x \min_{0 \leq i \leq k} d(x_i, x)
\]

Geodesic Voronoi:

\[
C_i = \{x \mid \forall j \neq i, d(x_i, x) \leq d(x_j, x)\}
\]

Geodesic Delaunay connectivity:

\[
(x_i \leftrightarrow x_j) \iff (C_i \cap C_j \neq \emptyset)
\]

→ geodesic Delaunay refinement.

→ distance conforming. → triangulation conforming if the metric is “gradded”.
Adaptive Meshing
Adaptive Meshing

Texture | Metric | Uniform | Adaptive

# samples
Linear approximation $f_M$ with $M$ linear elements. Minimize approximation error $\| f - f_M \|_{L^p}$. 

Isotropic
Linear approximation $f_M$ with $M$ linear elements.

Minimize approximation error $\|f - f_M\|_{L^p}$.

$L^\infty$ optimal metrics for smooth functions:

Images: $T(x) = |H(x)|$ (Hessian)

Surfaces: $T(x) = |C(x)|$ (curvature tensor)
Approximation Driven Meshing

Linear approximation $f_M$ with $M$ linear elements.

Minimize approximation error $\|f - f_M\|_{L^p}$.

$L^\infty$ optimal metrics for smooth functions:

Images: $T(x) = |H(x)|$ (Hessian)

Surfaces: $T(x) = |C(x)|$ (curvature tensor)

For edges and textures: → use structure tensor.

[Peyrê et al, 2008]
Approximation Driven Meshing

Linear approximation $f_M$ with $M$ linear elements.
Minimize approximation error $\|f - f_M\|_{L^p}$.

$L^\infty$ optimal metrics for smooth functions:
- Images: $T(x) = |H(x)|$ (Hessian)
- Surfaces: $T(x) = |C(x)|$ (curvature tensor)

For edges and textures: → use structure tensor. 
[Peysré et al, 2008]

→ extension to handle boundary approximation. 
[Peysré et al, 2008]
Conclusion

Riemannian tensors encode geometric features.
   → Size, orientation, anisotropy.

*Computing geodesic distance:*
   iterative vs. propagation.
Conclusion

Riemannian tensors encode geometric features.
→ Size, orientation, anisotropy.

Computing geodesic distance:
iterative vs. propagation.

Using geodesic curves: image segmentation.

Using geodesic distance: image and surface meshing