Working with Implicit Surfaces and Point Clouds

Guillermo Sapiro
Electrical and Computer Engineering
University of Minnesota
guille@ece.umn.edu

Supported by NSF, ONR, PECASE, CAREER, NIH
Overview

• Motivation
• Geodesics and generalized geodesics
• Comparing point clouds
• Meshless geometric subdivision
• The future and concluding remarks
Motivation

- **Implicit surfaces**
  - Facilitate fundamental computations
  - Natural representation for many algorithms (e.g., medical imaging)
  - Part of the computation very often (distance functions)

- **Point clouds**
  - Natural representation for 3D scanners
  - Natural representation for manifold learning

- **Dimensionality independent**
  - Pure geometry (no artificial meshes, etc)
Geodesics and Generalized Geodesics

Joint with Facundo Memoli
Motivation: A Few Examples
Motivation: A Few Examples (cont.)
Motivation: What is a Geodesic?

\[ d^g_S(p, x) = \inf_C \int_p^x g(C)ds \]
Motivation: What is a Geodesic?

\[ d^g_S(p, x) = \inf_C \int_{p}^{x} g(C) \, ds \]
Background: Distance and Geodesic Computation via Dijkstra

- **Complexity:** $O(n \log n)$

- **Advantage:** Works in any dimension and with any geometry (graphs)

- **Problems:**
  - Not consistent
  - Unorganized points?
  - Noise?
  - Implicit surfaces?
Background: Distance and Geodesic Computation via Dijkstra

- **Complexity:** $O(n \log n)$
- **Advantage:** Works in any dimension and with any geometry (graphs)
- **Problems:**
  - Not consistent
  - Unorganized points?
  - Noise?
  - Implicit surfaces?
Background: Distance and Geodesic Computation via Dijkstra

- **Complexity:** $O(n \log n)$
- **Advantage:** Works in any dimension and with any geometry (graphs)
- **Problems:**
  - Not consistent
  - Unorganized points?
  - Noise?
  - Implicit surfaces?
Background: Distance and Geodesic Computation via Dijkstra

- **Complexity:** $O(n \log n)$
- **Advantage:** Works in any dimension and with any geometry (graphs)
- **Problems:**
  - Not consistent
  - Unorganized points?
  - Noise?
  - Implicit surfaces?
Background: Distance and Geodesic Computation via Dijkstra

- **Complexity:** $O(n \log n)$
- **Advantage:** Works in any dimension and with any geometry (graphs)
- **Problems:**
  - Not consistent
  - Unorganized points?
  - Noise?
  - Implicit surfaces?
Background: Distance Functions as Hamilton-Jacobi Equations

- $g =$ weight on the hyper-surface
- The $g$-weighted distance function between two points $p$ and $x$ on the hyper-surface $S$ is:

$$
\nabla_S d^g_S(p, x) = g
$$
\[ \| \nabla_s d^g_s(p, x) \| = g \]
Background: Computing Distance Functions as Hamilton-Jacobi Equations

- Solved in $O(n \log n)$ by Tsitsiklis, by Sethian, and by Helmsen, only for Euclidean spaces and Cartesian grids.

$$\nabla d^g(p, x) = g$$

- Solved only for acute 3D triangulations by Kimmel and Sethian.
A real time example
The Problem

How to compute intrinsic distances and geodesics for

- General dimensions
- Implicit surfaces
- Unorganized noisy points (hyper-surfaces just given by examples)
Our Approach

- We have to solve

\[ \nabla_S d_S^g(p, x) = g \]
Basic Idea
Basic Idea
Basic Idea

**Theorem** (Memoli-Sapiro):

\[
\left| d^g - d^g_S \right| \rightarrow 0
\]
Basic idea

\[ |d^g - d^g_s| \rightarrow \begin{cases} h^{1/2} & \text{general} \\ h & \text{local analytic} \\ h^\gamma, \gamma > 1 & \text{"smart" metric} \end{cases} \]
Why is this good?

\[
\nabla_s d^g_s(p, x) = g
\]

\[
\nabla d^g(p, x) = g
\]
Implicit Form Representation

\[ S = \text{level} \text{− set of } \Psi : \mathbb{R}^n \to \mathbb{R} = \{ x : \Psi(x) = 0 \} \]

Figure from G. Turk
Data extension

\[ I : M \rightarrow \mathbb{R} \]

- Embed M:
  \[ M = \{ x : \Psi(x) = 0 \} \]

- Extend I outside M:
  \[ \frac{\partial I}{\partial t} + \text{sign}(\psi) (\nabla I \cdot \nabla \psi) = 0 \]
Examples
Examples
Examples
Examples
Unorganized points
Unorganized points (cont.)
Unorganized points
Randomly sampled manifolds (with noise)

**Theorem** (Memoli - S. 2002):

\[
\max_{p,q \in S} \left( d_S(p,q) - d_{\mathcal{P}_n(h)}^h(p,q) \right) \leq C_S \sqrt{h}
\]

\[
P\left( \max_{p,q \in \mathcal{S}} \left( d_S(p,q) - d_{\Omega_n^h}(p,q) \right) > \varepsilon \right) \xrightarrow{n \uparrow \infty} 0
\]

\[
\lim_{h,n} P\left( d_{\mathcal{H}}(S, \Omega_n^h) > \varepsilon \right) = 0
\]
Examples (VRML)
Examples
Intrinsic Voronoi of Point Clouds
Intermezzo: de Silva, Tenenbaum, et al...
**Main Problem:**

- Doesn’t address noisy examples/measurements:
  Much less robust to noise!
Error increases with the number of samples!
Intermezzo: de Silva, Tenenbaum, et al...

**Problems:**

- Doesn’t address noisy examples/measurements: Much less robust to noise!
- Only convex surfaces
- Uses Dijkstra (back to non consistency)
- Doesn’t work for implicit surface representations
Is this a geodesic?
Generalized geodesics: Harmonic maps

Find a smooth map from two manifolds \((M,g)\) and \((N,h)\) such that

\[
\min_{C:M \to N} \int_{\Omega} \left\| \nabla_M C \right\|^p \, d\text{vol}_M
\]

\[
\left( \frac{\partial C}{\partial t} = \right)
\Delta_M C + A_N(C) \left< \nabla_M C, \nabla_M C \right> \geq 0
\]
Examples

- **M is an Euclidean space and N the real line**

\[ \Delta C = 0 \]

- **M = [0,1], geodesics!**

\[
\frac{\partial^2 C}{\partial t^2} + A_N(C) < \nabla_M C, \nabla_M C > = 0
\]
Color Image Enhancement
(with B. Tang and V. Caselles)
Implicit surfaces

- Domain and target are implicitly represented: Simple Cartesian numerics

\[
\frac{\partial C}{\partial t} = \text{div}(P_{\nabla \psi} \nabla C) + \left( \sum_k H_\Phi \left\langle \frac{\partial C}{\partial x_k}, \frac{\partial C}{\partial x_k} \right\rangle \right) \| \nabla \Phi \| 
\]
Example: Chroma denoising on a surface (with Bertalmio, Cheng, Osher)
Example: Direction denoising
(with Bertalmio, Cheng, Osher)
Application (with G. Gorla and V. Interrante)
Texture mapping denoising
Texture mapping denoising
Examples
(with Betalmio, Cheng, Osher)
Vector field visualization (e.g., principal directions) 
(with Bertalmio, Cheng, Osher)
Concluding remarks

• A general computational framework for distance functions, geodesics, and generalized geodesics

• Implicit hyper-surfaces and un-organized points
Comparing Point Clouds

Joint with Facundo Memoli
What is and Motivation

• **Comparing point clouds**
  - Dimension independent
  - Geometric
  - Bending (isometric) invariant
  - Supported by theory and computational framework
The Gromov-Hausdorff Distance

- **Hausdorff distance**

\[
 d_Z^H(X, Y) \overset{\triangle}{=} \max\left(\sup_{x \in X} d(x, Y), \sup_{y \in Y} d(y, X)\right)
\]

- **Gromov-Hausdorff distance**

\[
 d_{GH}(X, Y) \overset{\triangle}{=} \inf_{Z, f, g} d_Z^H(X, Y)
\]

\[
 f : X \to Z, \quad g : Y \to Z \text{ isometric embeddings}
\]
Key question

- How to estimate the Gromov-Hausdorff distance from noisy samples of the metric space
First step: Working with point clouds

Let $X$ and $Y$ be compact metric spaces, $X_m$ an $r$-covering of $X$ and $Y_{m'}$ an $r'$-covering of $Y$. Then

$$|d_{\mathcal{G}_H}(X, Y) - d_{\mathcal{G}_H}(X_m, Y_{m'})| \leq r + r'$$

• **Consequence:** Working with point clouds “is possible”
How we compute the distance?

\[ d_{\mathcal{I}}(X, Y) \triangleq \min_{\pi \in \mathcal{P}_n} \max_{1 \leq i, j \leq n} \frac{1}{2} |d_X(x_i, x_j) - d_Y(y_{\pi_i}, y_{\pi_j}) | \]

\[ d_{\mathcal{G}_\mathcal{H}}(X, Y) \leq d_{\mathcal{I}}(X, Y) \]

\[ d_{\mathcal{G}_\mathcal{H}}(X, Y) \leq R_X + R_Y + d_{\mathcal{I}}(X, Y) \]

**Consequence:** If we see a small pairwise distance, the objects are isometric.
The need for a probabilistic framework

Let \((X, d_X)\) and \((Y, d_Y)\) be any pair of given compact metric spaces and let \(\eta = d_{\mathcal{H}}(X, Y)\). Also, let \(N_{X, n}^{(r, s)} = \{x_1, \ldots, x_n\}\) be given. Then, given \(\alpha > 0\) there exist points \(\{y_1^\alpha, \ldots, y_n^\alpha\} \subset Y\) such that

1. \(d_{\mathcal{I}}(N_{X, n}^{(r, s)}, \{y_1^\alpha, \ldots, y_n^\alpha\}) \leq (\eta + \alpha)\)

2. \(B_Y(\{y_1^\alpha, \ldots, y_n^\alpha\}, r + 2(\eta + \alpha)) = Y\)

3. \(d_Y(y_i^\alpha, y_j^\alpha) \geq s - 2(\eta + \alpha)\) for \(i \neq j\).
The need for a probabilistic framework (cont.)

- The problem is well posed
- No reason for the y’s to be given:

\[
d_I(N_{X,n}^{(r,s)}, N_{Y,n}^{(\tilde{r},\tilde{s})}) \leq d_I(N_{X,n}^{(r,s)}, N_{Y,n}^{(r,s)}) + d_I(N_{Y,n}^{(\tilde{r},\tilde{s})}, N_{Y,n}^{(r,s)})
\]

\[
= 0 + \text{small}(r, \tilde{r})
\]
- We need probabilistic bounds!
The probabilistic framework

- **Bottleneck distance** between two samples of the same space:

\[
d^Z_B(Z, Z') \triangleq \min_{\pi \in \mathcal{P}_n} \max_k d_Z(z_k, z'_{\pi_k}) \geq d_I(Z, Z')
\]

- Using concepts from intrinsic Voronoi diagrams and coupon collector theorem we have:
Let \((Z, d_Z)\) be a smooth compact submanifold of \(IR^d\). Given a covering \(N_{Z,n}^{(r,s)}\) of \(Z\) and a number \(p \in (0,1)\), there exists a positive integer \(m = m_n(p)\) such that if \(Z_m = \{z_k\}_{k=1}^m\) is a sequence of i.i.d. points sampled uniformly from \(Z\), with probability \(p\) one can find a set of \(n\) different indices \(\{i_1, \ldots, i_n\} \subset \{1, \ldots, m\}\) with

\[
d_B^{Z}(N_{Z,n}^{(r,s)}, \{z_{i_1}, \ldots, z_{i_n}\}) \leq r
\]
Let $X$ and $Y$ compact submanifolds of $\mathbb{R}^d$. Let $N_{X,n}^{(r,s)}$ be a covering of $X$ with separation $s$ such that for some positive constant $c$, $s - 2d_{\mathcal{H}}(X,Y) > c$. Then, given any number $p \in (0,1)$, there exists a positive integer $m = m_n(p)$ such that if $Y_m = \{y_k\}_{k=1}^m$ is a sequence of i.i.d. points sampled uniformly from $Y$, we can find, with probability at least $p$, a set of $n$ different indices $\{i_1, \ldots, i_n\} \subset \{1, \ldots, m\}$ such that

$$d_{\mathcal{I}}(N_{X,n}^{(r,s)}, \{y_{i_1}, \ldots, y_{i_n}\}) \leq 3d_{\mathcal{H}}(X,Y) + r$$
Computational considerations

- Bounds on the number of sample points needed
- Covers of Y found using farthest point sampling.
- Geodesic distances for points on X and Y
- Select matching points of X and Y following our theory
Examples
Meshless Geometric Subdivision

Joint with
Carsten Moenning
Facundo Memoli
Nira Dyn
N. Dodgson
What is and Motivation

- **Mesh based subdivision**
  - Refinement (add points and edges)
  - Averaging

- **Mesh not really geometric**
What is and Motivation (cont.)

- Point clouds are natural for 3D scanners
- Point clouds are the “true” geometry
- Point clouds are dimensionality independent
- All operations are geometric
Main Steps

- Intrinsic point cloud simplification
- Intrinsic proximity information
- Geodesic centroid computation
- Intrinsic subdivision scheme
Intrinsic point cloud simplification

- Follows Meonning & Dodgson
- Based on progressive farthest point sampling
- Computed based on intrinsic Voronoi diagram (uses distance on point clouds)
- Guaranteed bounds on distance between samples
Intrinsic proximity information

- “Replaces” (non-geometric) connectivity in mesh techniques
- Given by neighbors from the intrinsic Voronoi
- Easily updated when the point cloud is refined (using geodesics on point clouds)
Geodesic centroid computation
Geodesic centroid computation (cont.)

\[
\text{centroid} := \min_g \frac{1}{2} \sum_{k=1}^{n} w_k d_M^2(g, p_k)
\]

\[
V(g) := \sum_{k=1}^{n} w_k \nabla_M \frac{1}{2} d_M^2(g, p_k) = 0
\]

\[
g_0 := \prod_M \left( \sum_{k=1}^{n} w_k p_k \right) \rightarrow -V(g) \text{ centroid}
\]
Intrinsic subdivision scheme

- **Geometric averaging rule**: Replace the point by the geodesic centroid of its intrinsic neighborhood

- **Refinement rule**: For each neighbor, insert the geodesic centroid of the joint neighborhood
Example

Quantitative study in the paper
Example
Conclusions

- Work with implicit surfaces and point clouds!!!
Thanks


Sulcii extraction on meshes
(with A. Bartesaghi)

Follows Kimmel-Sethian