Geodesic Paths and Distances on Meshes: PDE and Computational Geometry Approaches

Matthieu Pierre Boyer & Antoine Groudiev

École Normale Supérieure

December 16, 2025

Introduction

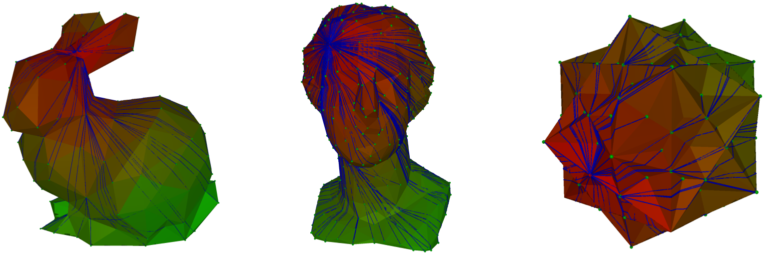


Figure: Geodesic paths from a source point on a bunny, a bust and a icosahedron.

Problem Positioning

Given a mesh of a 2-manifold embedded in $\mathbb{R}^3$ and two points on the mesh, what is the shortest path between the points?

Problem Positioning

Given a mesh of a 2-manifold embedded in $\mathbb{R}^3$ and two points on the mesh, what is the shortest path between the points?

2-**Manifold:** a surface that looks locally like $\mathbb{R}^2$;

Problem Positioning

Given a mesh of a 2-manifold embedded in $\mathbb{R}^3$ and two points on the mesh, what is the shortest path between the points?

2-**Manifold:** a surface that looks locally like $\mathbb{R}^2$;

Mesh: a set $\mathbb{V}$ of vertices, a set $\mathcal{F}$ of faces in $\mathbb{V}^3$.

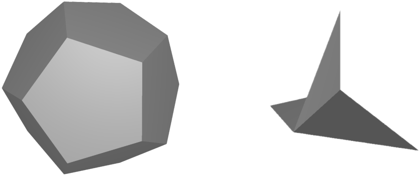


Figure: Examples of a 2-manifold (left) and non-manifold (right) mesh.

Discrete Differential Geometry

The Laplace-Beltrami operator of a function u is given by:

$$(\Delta u)_i = \frac{1}{2A_i} \sum_{e=(i,j)} \left(\cot \alpha_{i,j} + \cot \beta_{i,j}\right) \left(u_i - u_j\right). \tag{1}$$

It quantifies the symmetric deviation of the variation of the value at a point.

Physical Equations

We base ourselves on equations modelling the propagation of a phenomenon:

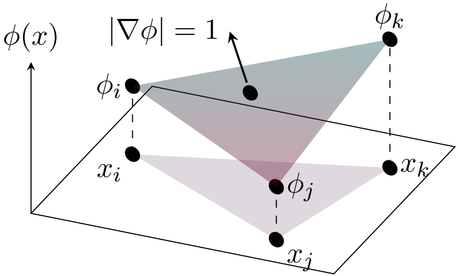
Heat Equation $\delta u_t = \frac{\mathrm{d}}{\mathrm{d}t} u_0$;

Poisson Equation $\Delta u = u_0$ for a fixed distribution u_0;

Spectral Embedding ℓ^2 distance based on eigenspaces of the laplacian Δ.

Fast Marching

We compute a variation of Dijkstra's algorithm based on the Eikonal equation $|\nabla u| = 1$ for wavefront propagation.



Improved Chen-Han (ICH) Algorithm

Idea: Propagate **windows** accross edges of the mesh, encoding shortest paths from a source.

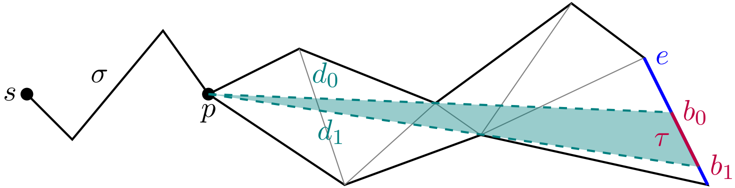


Figure: Illustration of a window $w = (s, p, e, b_0, b_1, d_0, d_1, \sigma)$.

One Angle, One Split Rule

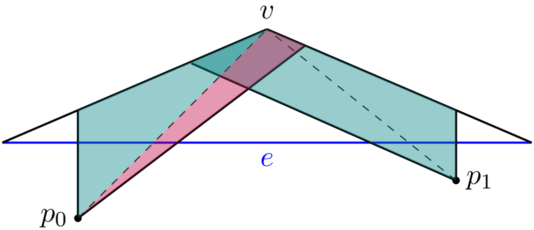
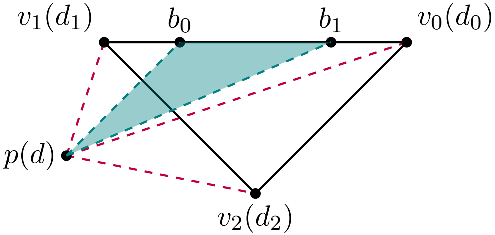


Figure: Illustration of the “one angle, one split” pruning rule: only keep three out of four windows.

Window Filtering



We can discard a window w if:

- $d + \|pb_0\| > d_0 + \|v_0 b_1\|$
- $d + \|pb_1\| > d_1 + \|v_1 b_1\|$
- $d + \|pb_0\| > d_2 + \|v_2 b_0\|$

Compute time: Heat method

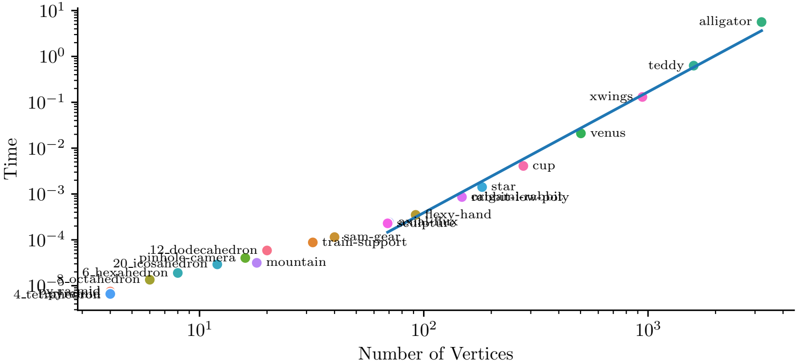


Figure: Benchmark of the *Heat method* as a function of the number of vertices. Slope: 2.63, r^2: 0.99.

Compute time: Fast Marching

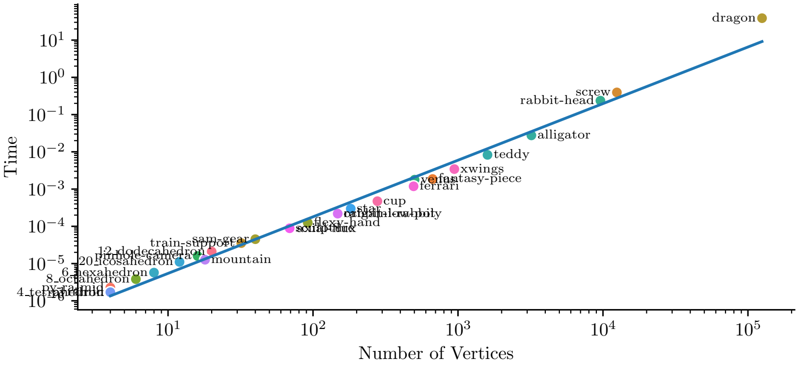


Figure: Benchmark of the *Fast Marching algorithm* as a function of the number of vertices. Slope: 1.51, r^2: 0.98.

Compute time: ICH

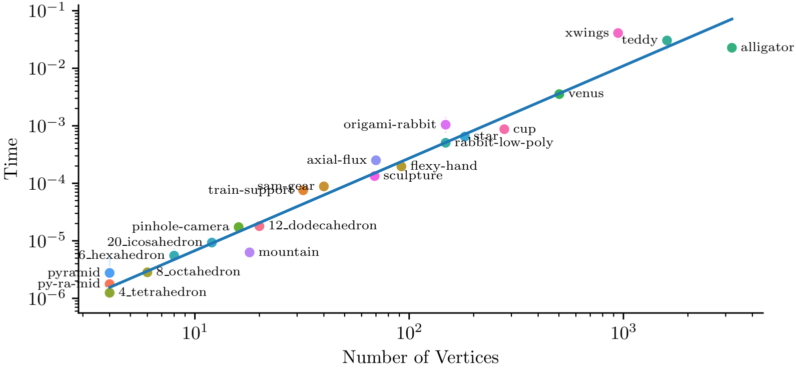


Figure: Benchmark of the *ICH algorithm* as a function of the number of vertices. Slope: 1.60, r^2: 0.96.

Comparison of methods

Method	Theoretical Runtime	Actual Runtime	Precision	Main limits
Heat	$\mathcal{O}(n_{\mathbb{V}}^{\omega})$	$\mathcal{O}(n_{\mathbb{V}}^{\omega})$	Approx.	No paths
Poisson	$\mathcal{O}(n_{\mathbb{V}}^{\omega})$		Approx.	Not geodesics
Spectral	$\mathcal{O}(n_{\mathbb{V}}^{\omega})$		Approx.	Not geodesics
Fast Marching	$\mathcal{O}(n_{\mathbb{V}} \log n_{\mathbb{V}})$	$\mathcal{O}(n_{\mathbb{V}}^{1.5})$	Approx.	Fixed sources
ICH	$\mathcal{O}(n_{\mathbb{V}}^{2} \log n_{\mathbb{V}})$	$\mathcal{O}(n_{\mathbb{V}}^{1.6})$	Exact	Limited to meshes

Bibliography I