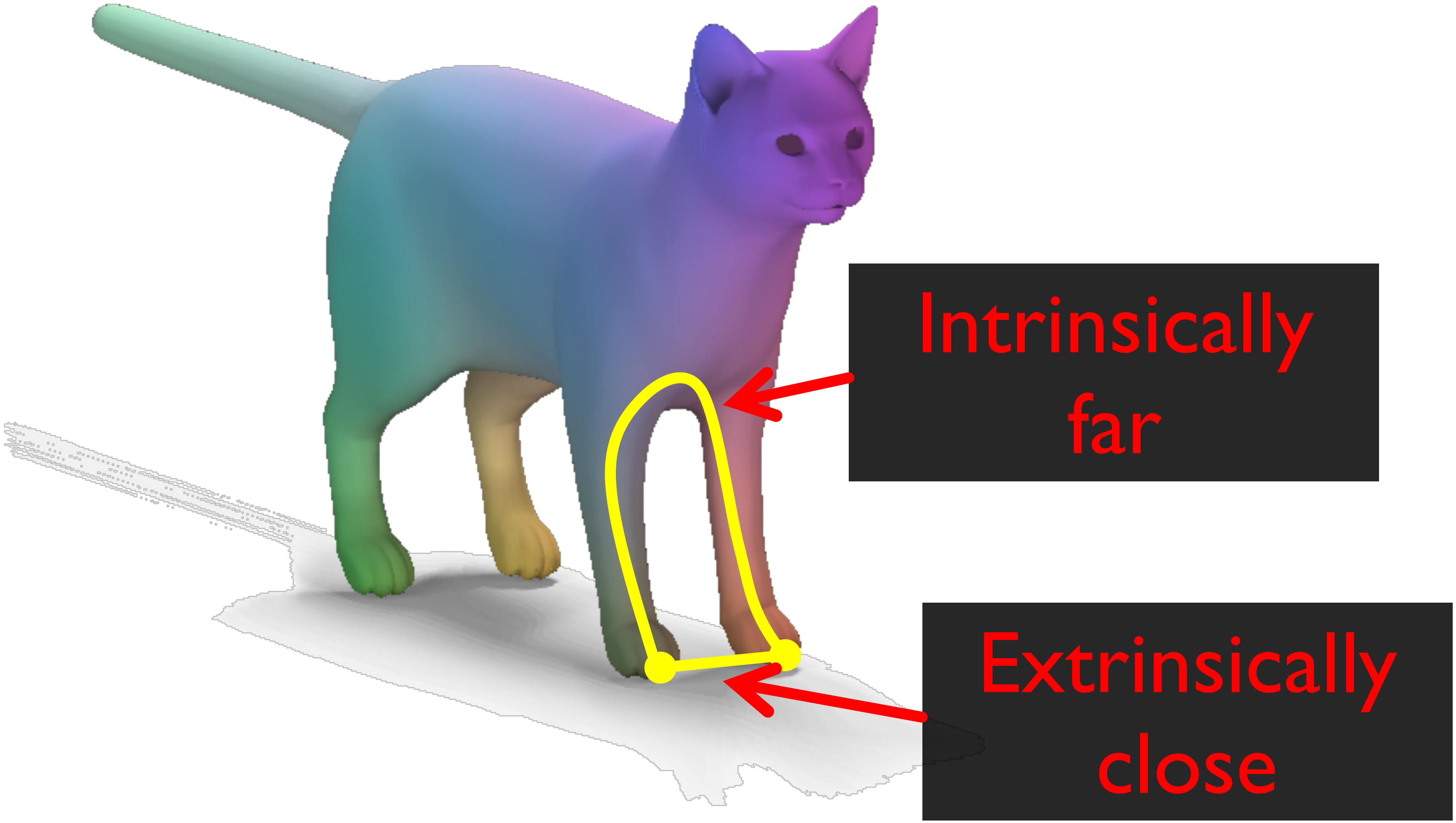
UC San Diego

Geodesics

Instructor: Hao Su

Geodesic Distances

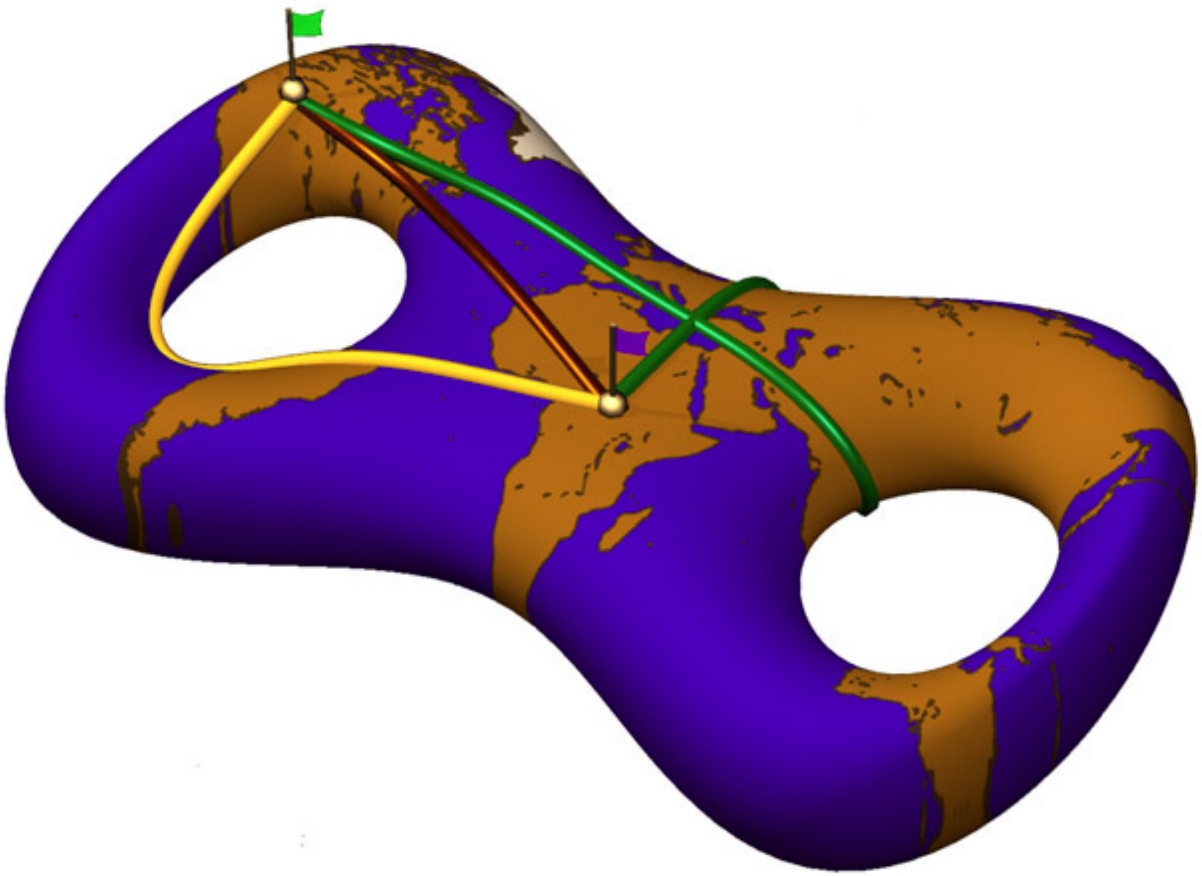


Geodesic distance

[jee-*uh*-**des**-ik **dis**-t*uh*-ns]**:**

Length of the shortest path, constrained not to leave the manifold.

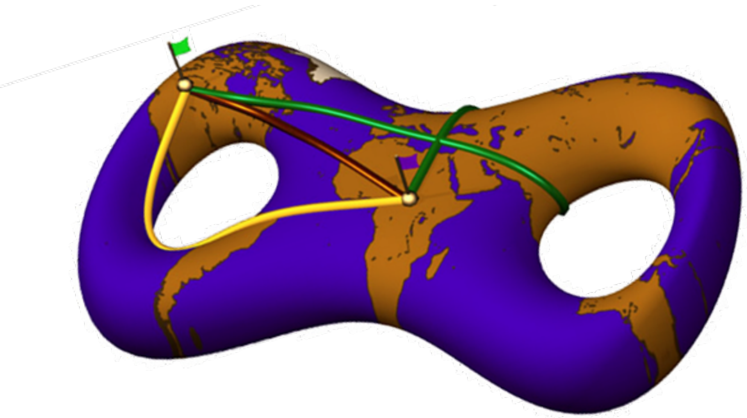
Complicated Problem



Straightest Geodesics on Polyhedral Surfaces (Polthier and Schmies)

Local minima

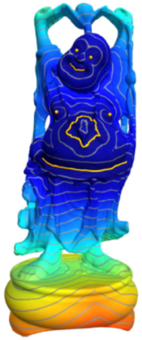
Related Queries



Locally OK

Single source

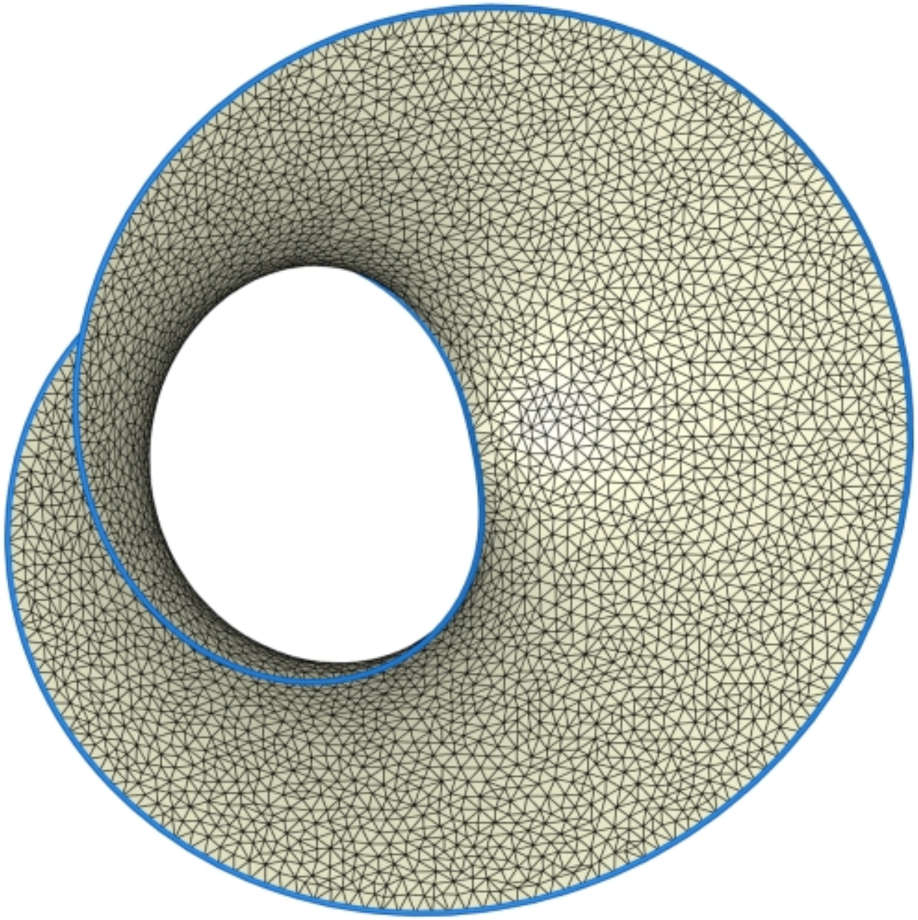
Multi-source



All-pairs

https://www.ceremade.dauphine.fr/~peyre/teaching/manifold/tp3.html http://www.sciencedirect.com/science/article/pii/S001044851100226

Computer Scientists' Approach

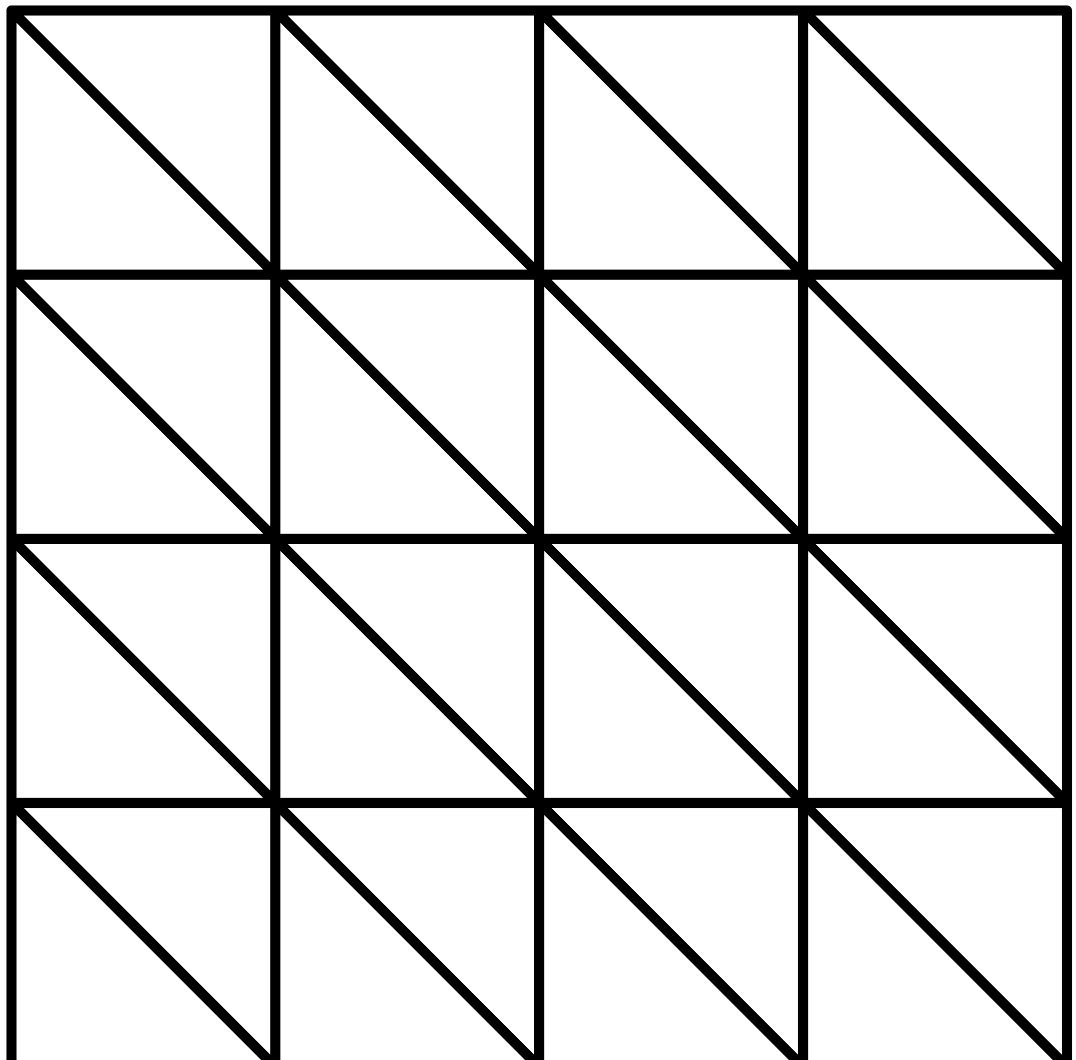


Approximate geodesics as paths along edges

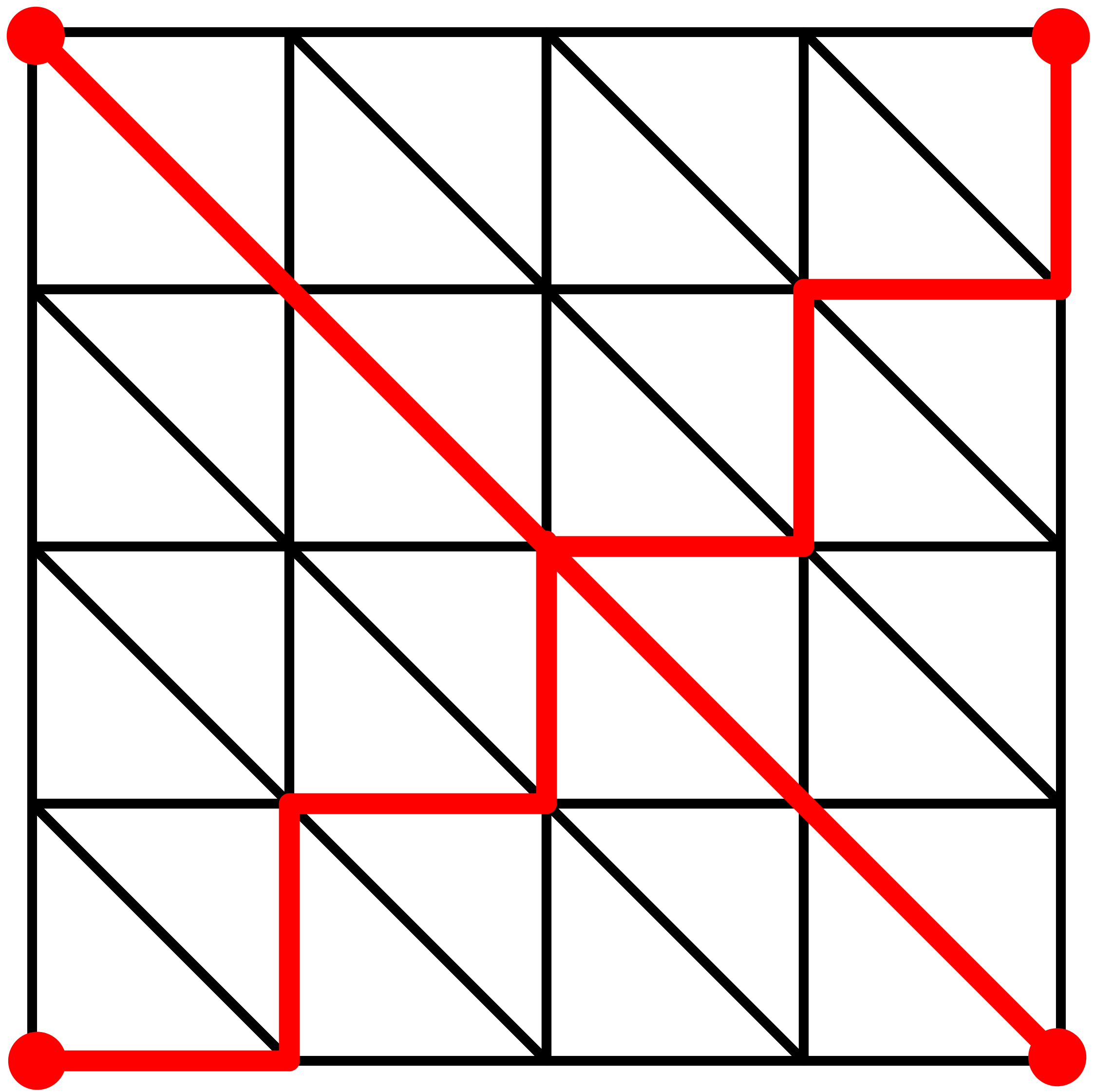
http://www.cse.ohio-state.edu/~tamaldey/isotopic.html

Meshes are graphs

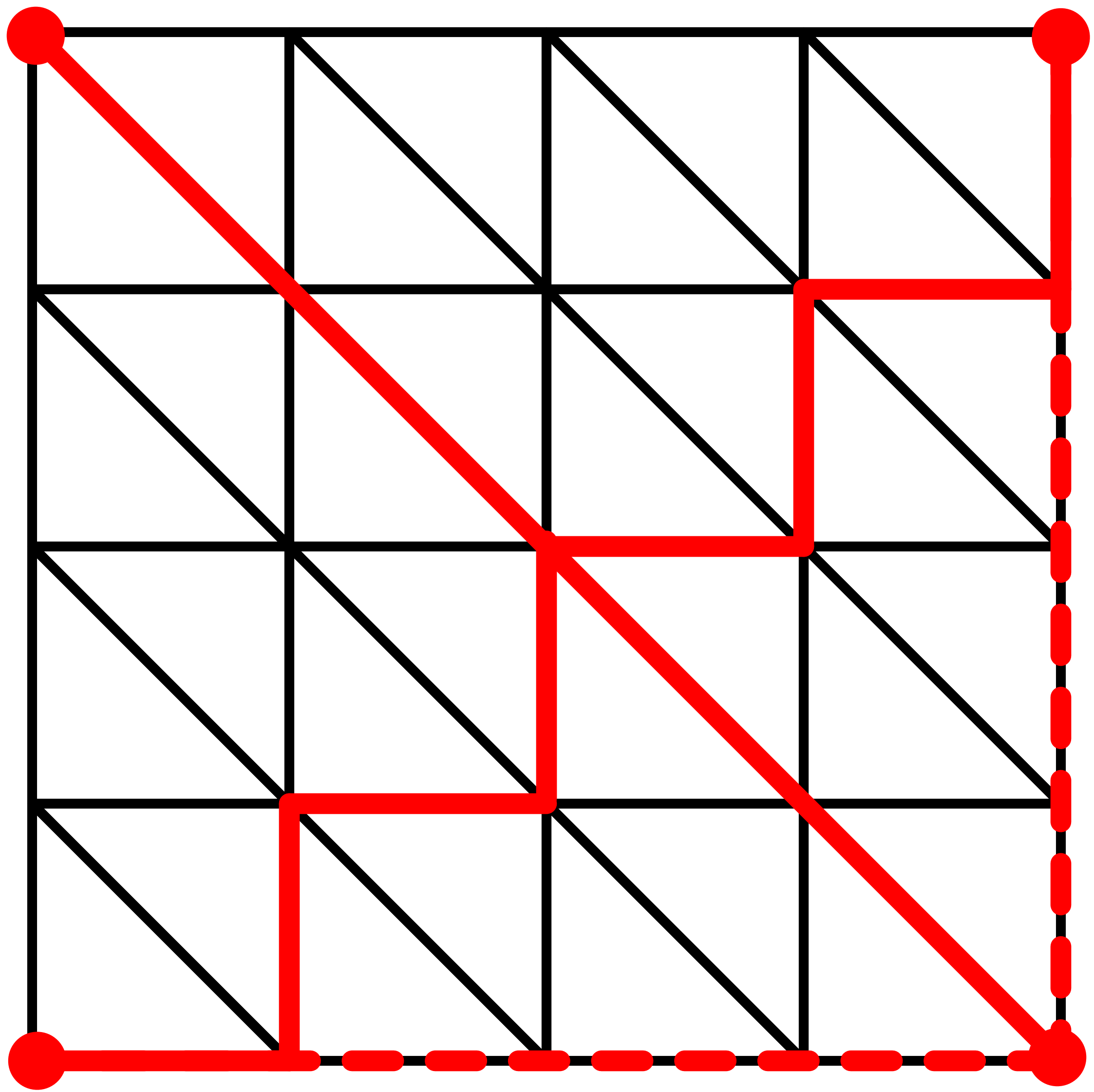
Pernicious Test Case



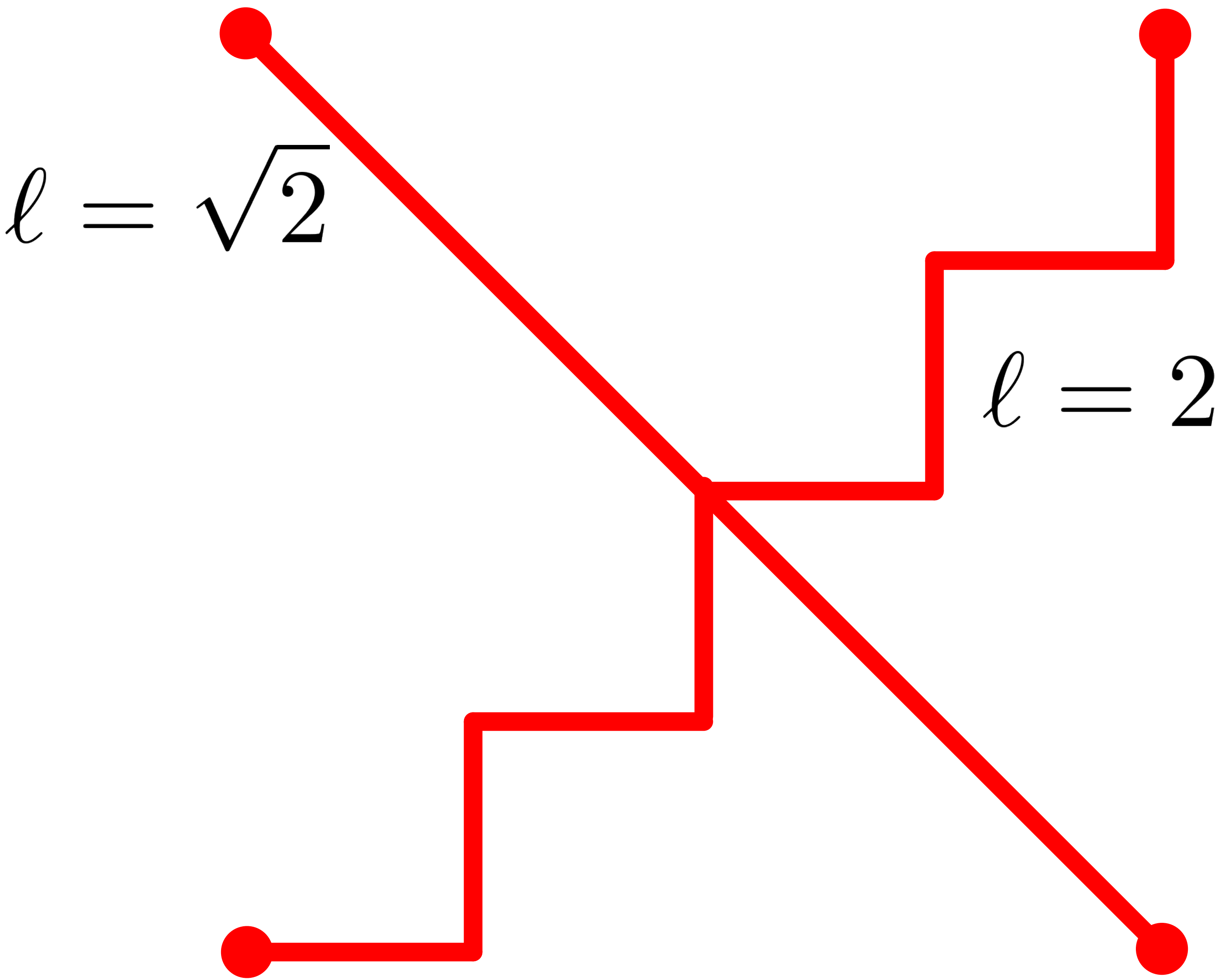
Pernicious Test Case



Pernicious Test Case



Distances



Conclusion 1

Graph shortest-path **does *not* converge** to geodesic distance.

Often an acceptable approximation.

Conclusion 2

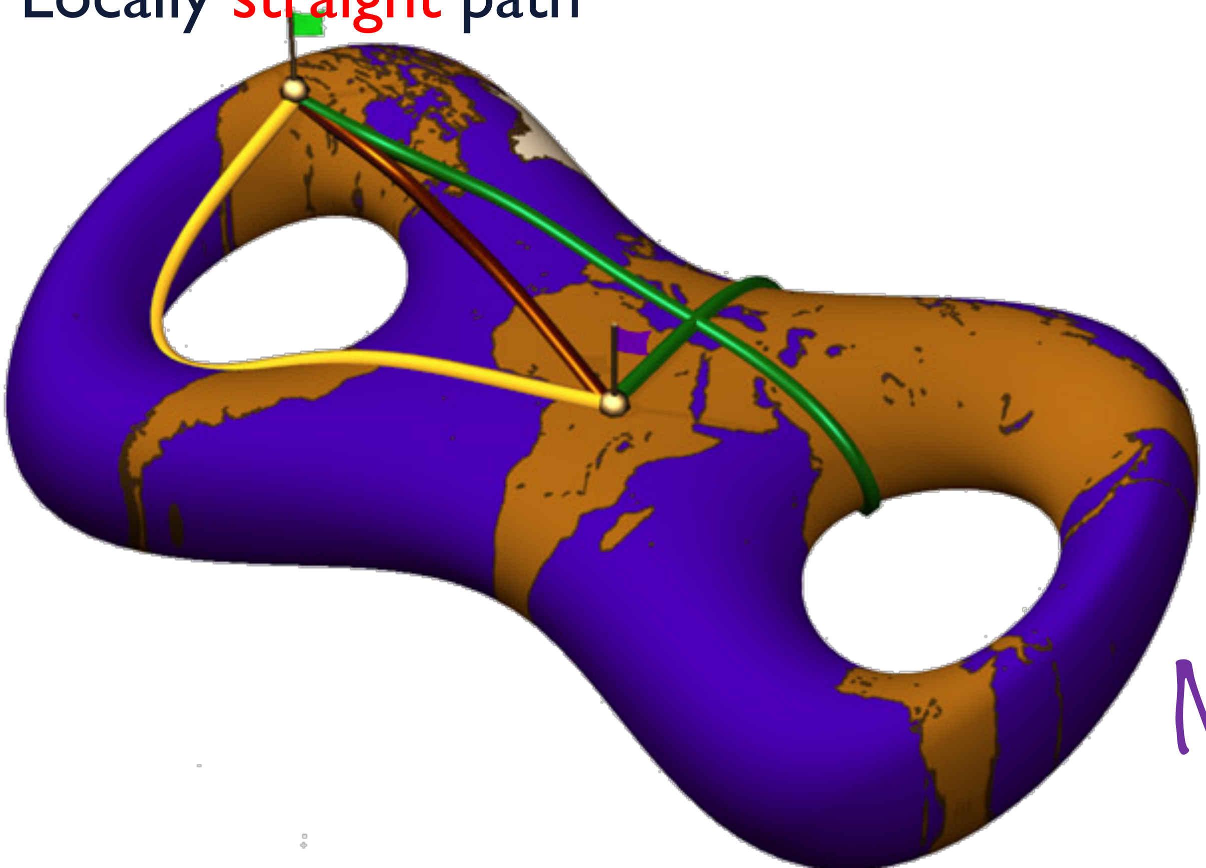
Geodesic distances need special discretization.

So, we need to understand the theory!

`\begin{math}`

Three Possible Definitions

- Globally shortest path
- Local minimizer of length
- Locally straight path



Not the same!

Recall: **Arc Length**

$$\int_a^b \|\gamma'(t)\| \, dt$$

Energy of a Curve

$$L[\gamma] := \int_a^b \|\gamma'(t)\|\, dt$$

Easier to work with:

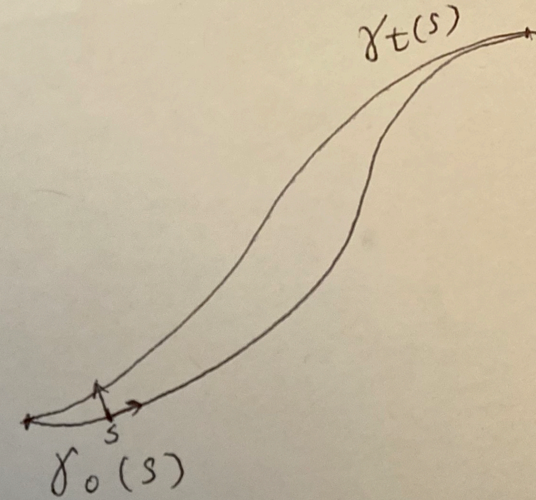
$$E[\gamma] := \frac{1}{2}\int_a^b \|\gamma'(t)\|^2\, dt$$

Lemma: $L^2 \leq 2(b-a)E$

Equality exactly when parameterized by arc length. Proof on board.

$$E[\gamma] = \frac{1}{2}\int_a^b \left(\frac{d\gamma_t(s)}{ds}\right)^2 ds$$

$$\frac{d}{dt}E[\gamma_t] = \int_a^b \frac{\partial \gamma_t(s)}{\partial s} \cdot \frac{\partial^2 \gamma_t(s)}{\partial t \partial s} \, ds$$



note:

$$\frac{\partial}{\partial s}\left\langle \frac{\partial \gamma}{\partial s}, \frac{\partial \gamma}{\partial t} \right\rangle$$

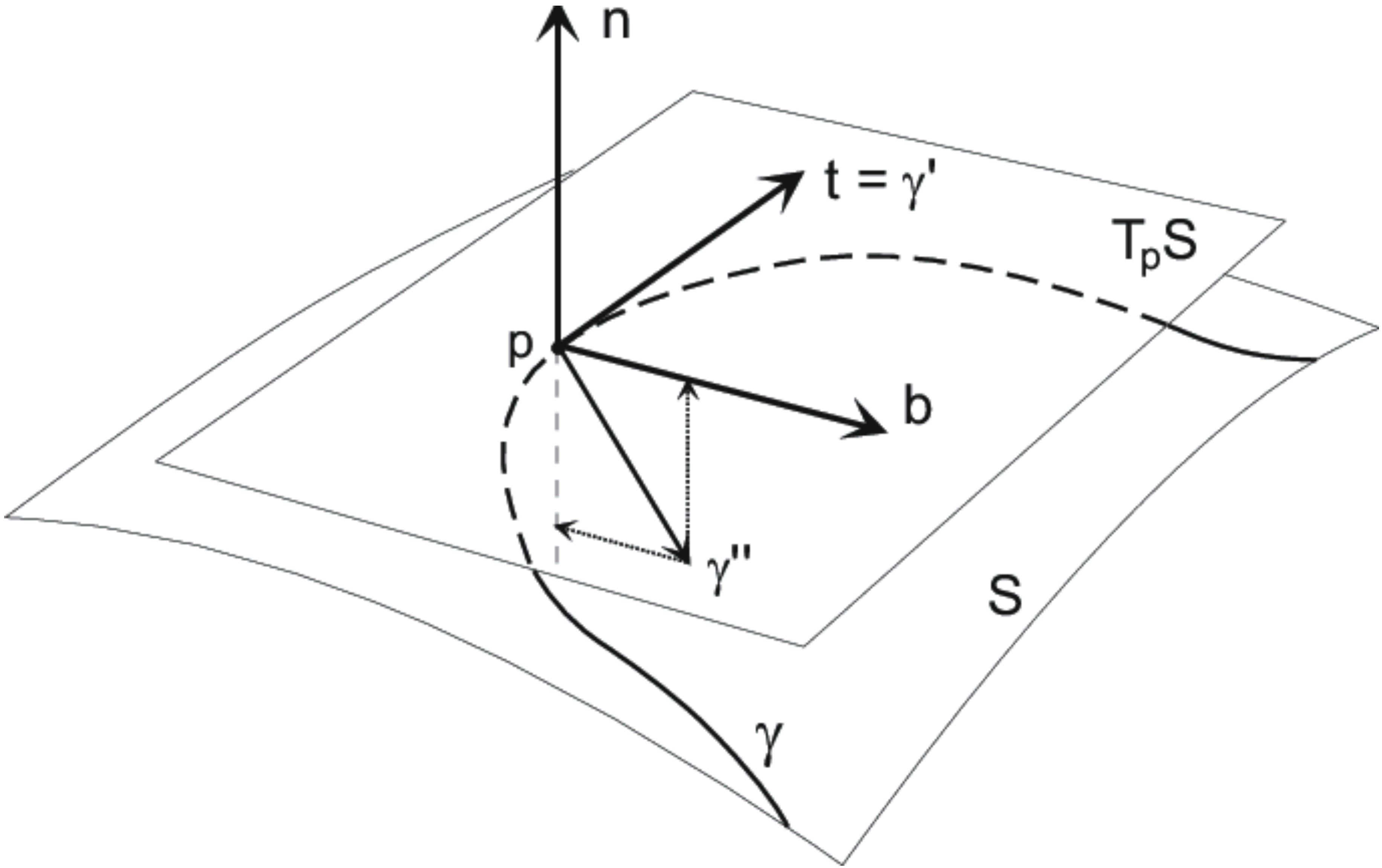
$$= \left\langle \frac{\partial^2 \gamma}{\partial s^2}, \frac{\partial \gamma}{\partial t} \right\rangle + \left\langle \frac{\partial \gamma}{\partial s}, \frac{\partial^2 \gamma}{\partial s \partial t} \right\rangle$$

By parametrization

$$\left\langle \frac{\partial \gamma}{\partial s}, \frac{\partial \gamma}{\partial t} \right\rangle \equiv 0$$

$$\therefore \left\langle \frac{\partial \gamma}{\partial s}, \frac{\partial^2 \gamma}{\partial s \partial t} \right\rangle = -\left\langle \frac{\partial^2 \gamma}{\partial s^2}, \frac{\partial \gamma}{\partial t} \right\rangle$$

curve normal ($\frac{\partial^2 \gamma}{\partial s^2}$), curve tangent ($\frac{\partial \gamma}{\partial t}$).



n
t = γ'
T_pS
p
b
γ''
S
γ

First Variation of Arc Length

Lemma. Let γ be a family of curves with fixed endpoints in surface S; assume γ is parameterized by arc length at t=0. Then,

$$\frac{d}{dt} E[\gamma_t]\bigg|_{t=0} = -\int_a^b \left(\frac{d\gamma_t(s)}{dt} \cdot \mathrm{proj}_{T_{\gamma_t(s)}S} \left[\gamma_t''(s)\right] \right) ds$$

Corollary. γ is a geodesic iff

$$\mathrm{proj}_{T_{\gamma(s)}S} \left[\gamma''(s)\right] = 0$$

Intuition

- The only acceleration is out of the surface
- No steering wheel!

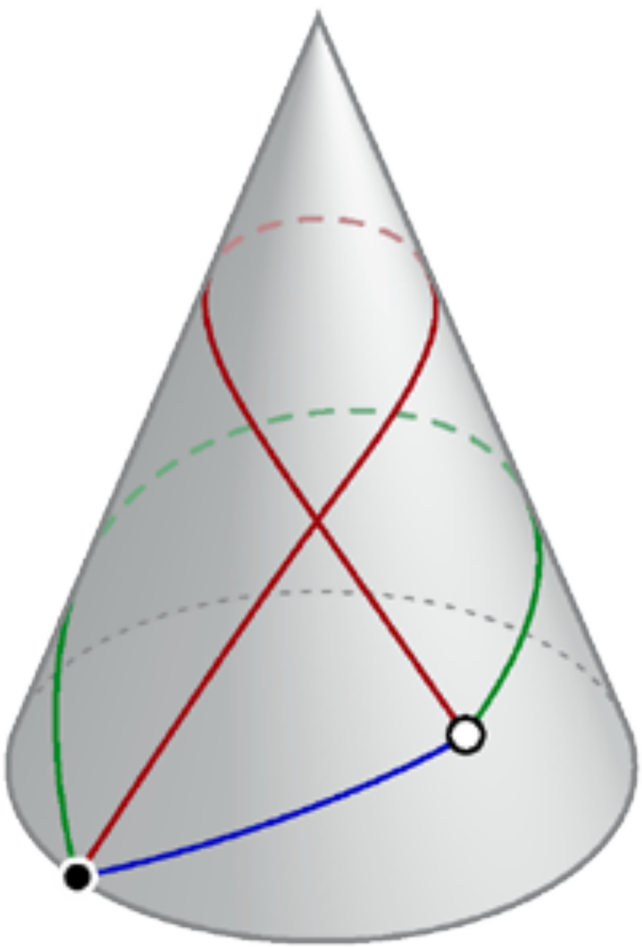
$$\mathrm{proj}_{T_{\gamma(s)}S}\left[\gamma''(s)\right] = 0$$

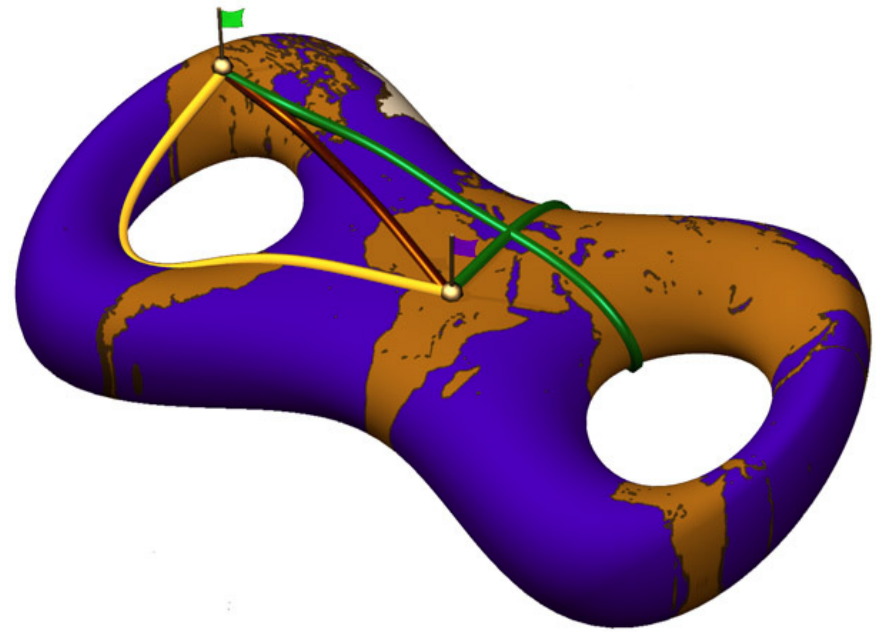
Two Local Perspectives

$$\mathrm{proj}_{T_{\gamma(s)}S}\left[\gamma''(s)\right] = 0$$

- **Boundary value problem**
  - **Given:** $\gamma(0), \gamma(1)$
- **Initial value problem (ODE)**
  - **Given:** $\gamma(0), \gamma'(0)$

Instability of Geodesics





Locally minimizing distance is not enough to be a shortest path!

http://parametricwood2011.files.wordpress.com/2011/01/cone-with-three-geodesics.png

\end{math}

Starting Point for Algorithms

Graph shortest path algorithms are well-understood.

Can we use them (carefully) to compute geodesics?

Useful Principles

"Shortest path had to come from somewhere."

"All pieces of a shortest path are optimal."

Dijkstra's Algorithm

v_0 = Source vertex

d_i = Current distance to vertex i

S = Vertices with known optimal distance

Initialization:

$$d_0 = 0$$

$$d_i = \infty \ \forall i > 0$$

$$S = \{\}$$

Dijkstra's Algorithm

$$v_0 = \text{Source vertex}$$

$$d_i = \text{Current distance to vertex } i$$

$$S = \text{Vertices with known optimal distance}$$

Iteration *k*:

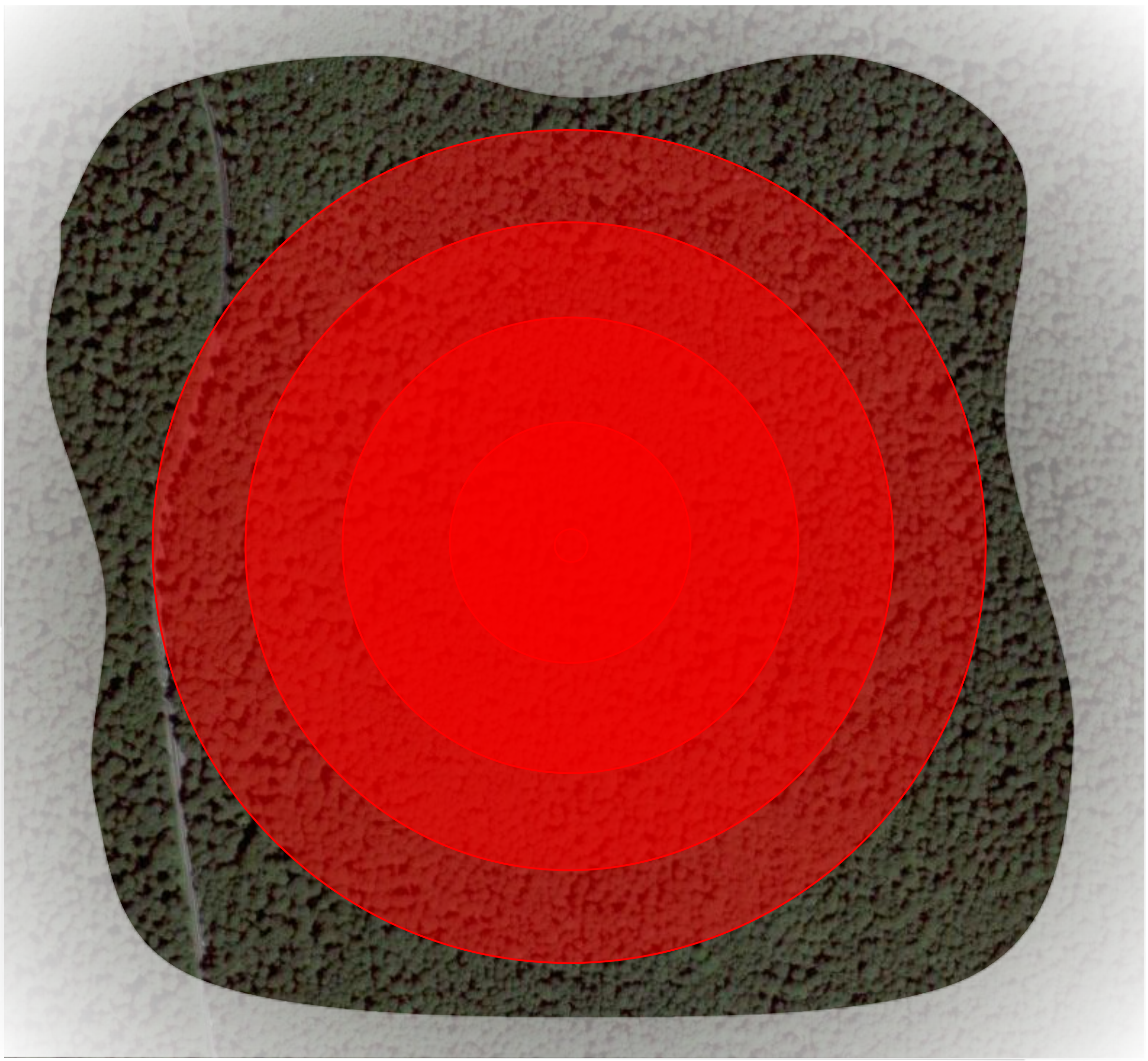
$$k = \arg \min_{v_k \in V \setminus S} d_k$$

$$S \leftarrow v_k$$

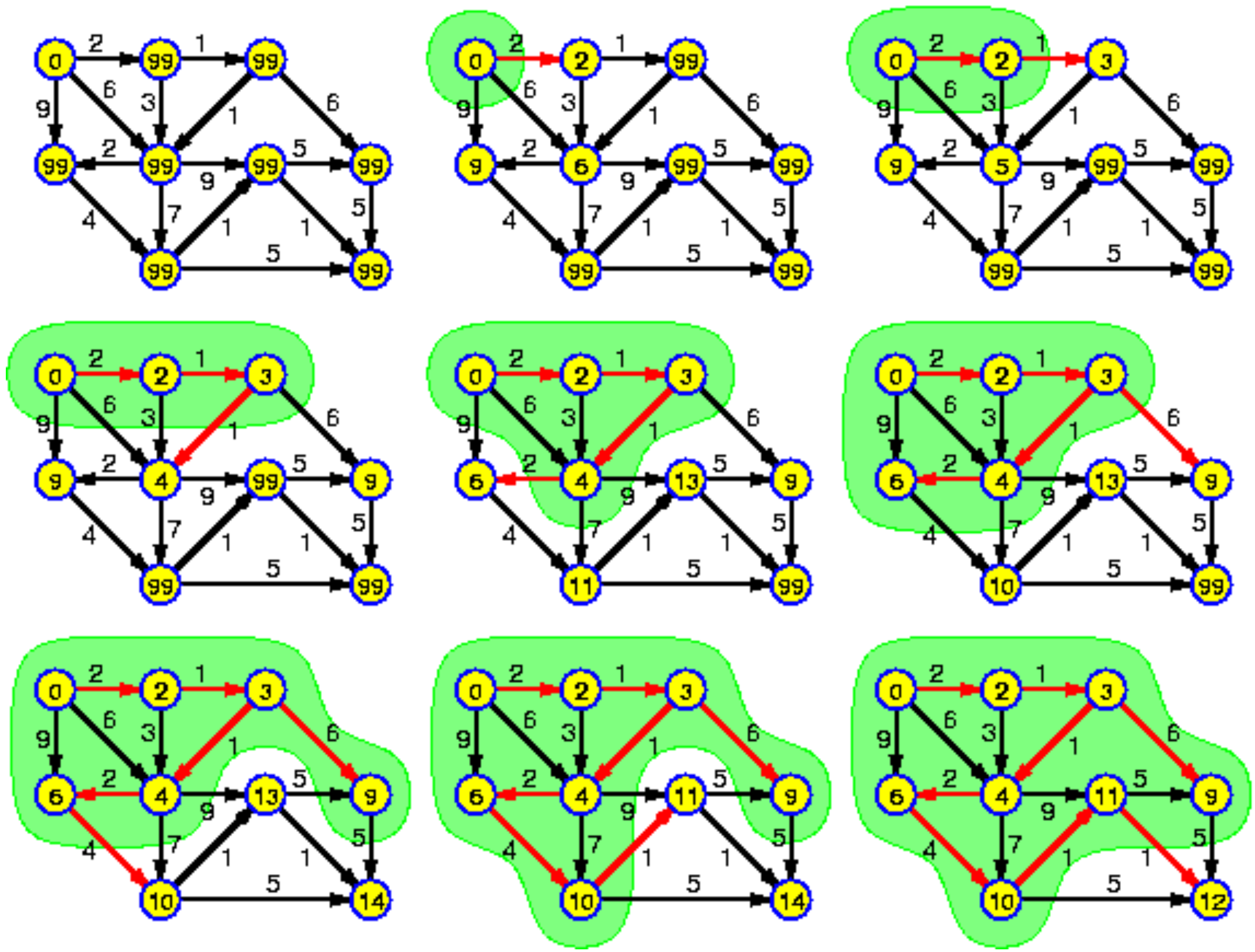
$$d_\ell \leftarrow \min\{d_\ell, d_k + d_{k\ell}\} \; \forall \text{ neighbors } v_\ell \text{ of } v_k$$

Inductive proof: During each iteration, *S* remains optimal.

Advancing Fronts



Example

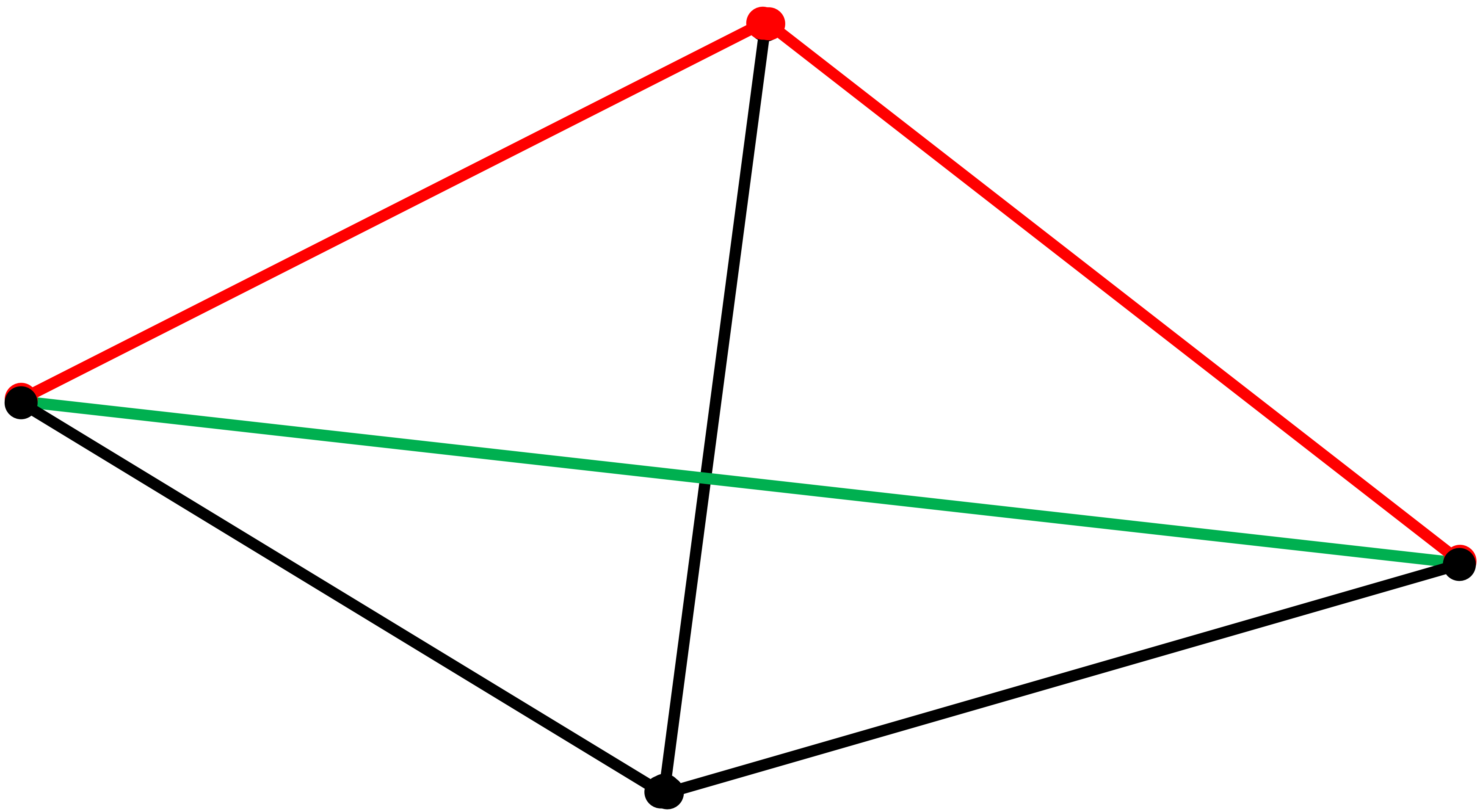


http://www.iekucukcay.com/wp-content/uploads/2011/09/dijkstra.gif

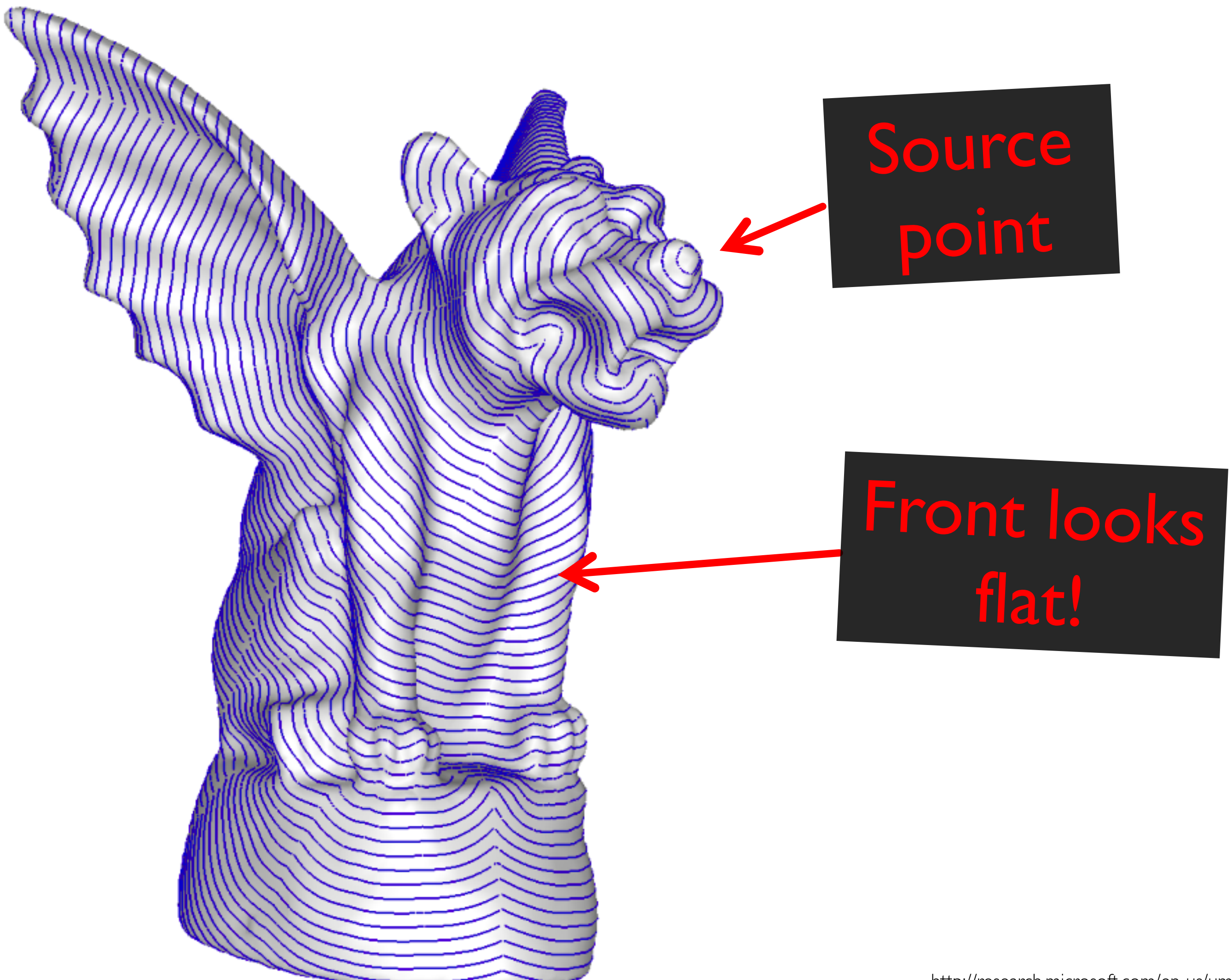
Fast Marching

Dijkstra's algorithm, modified to approximate geodesic distances.

Problem

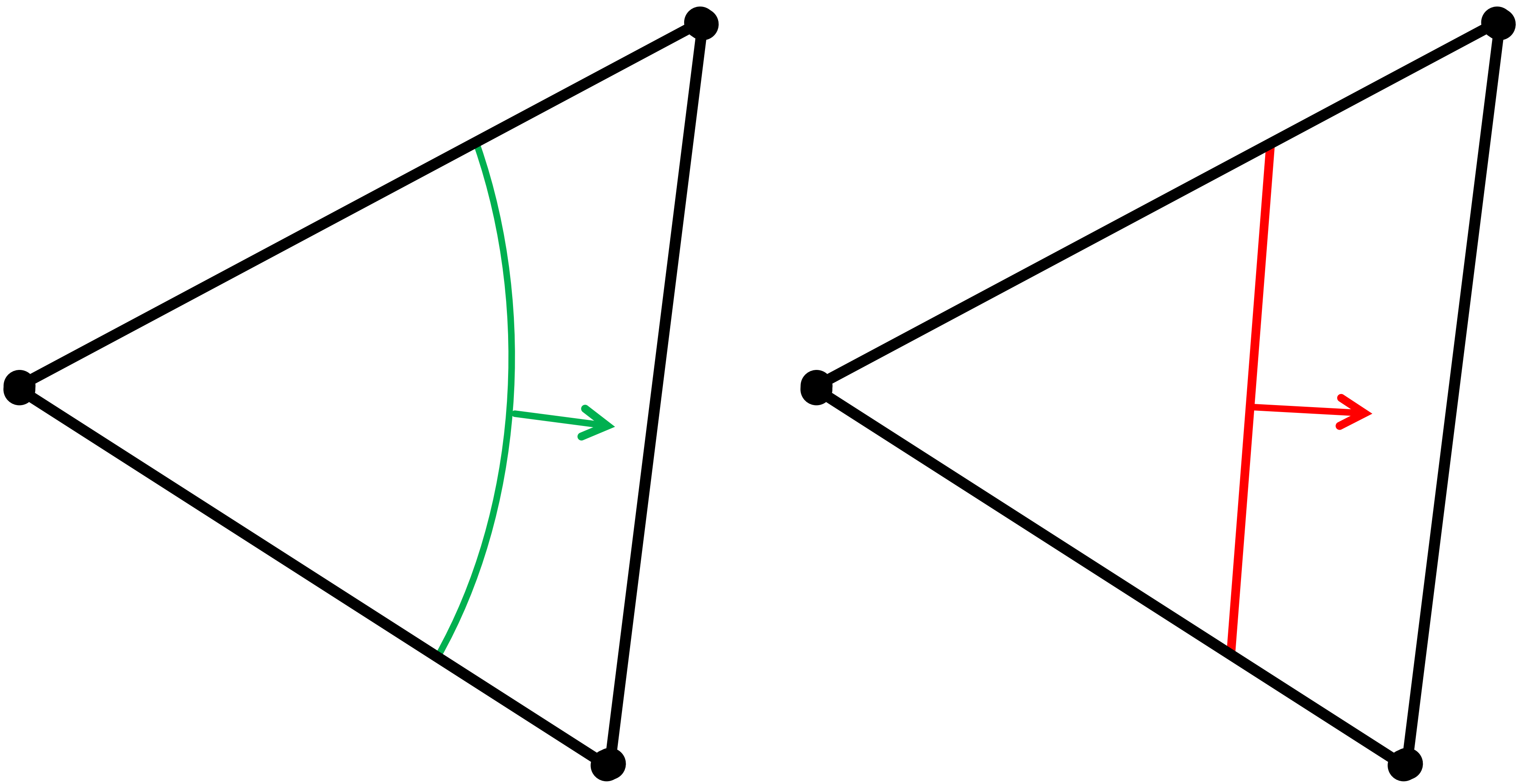


Planar Front Approximation



http://research.microsoft.com/en-us/um/people/hoppe/geodesics.pdf

At Local Scale



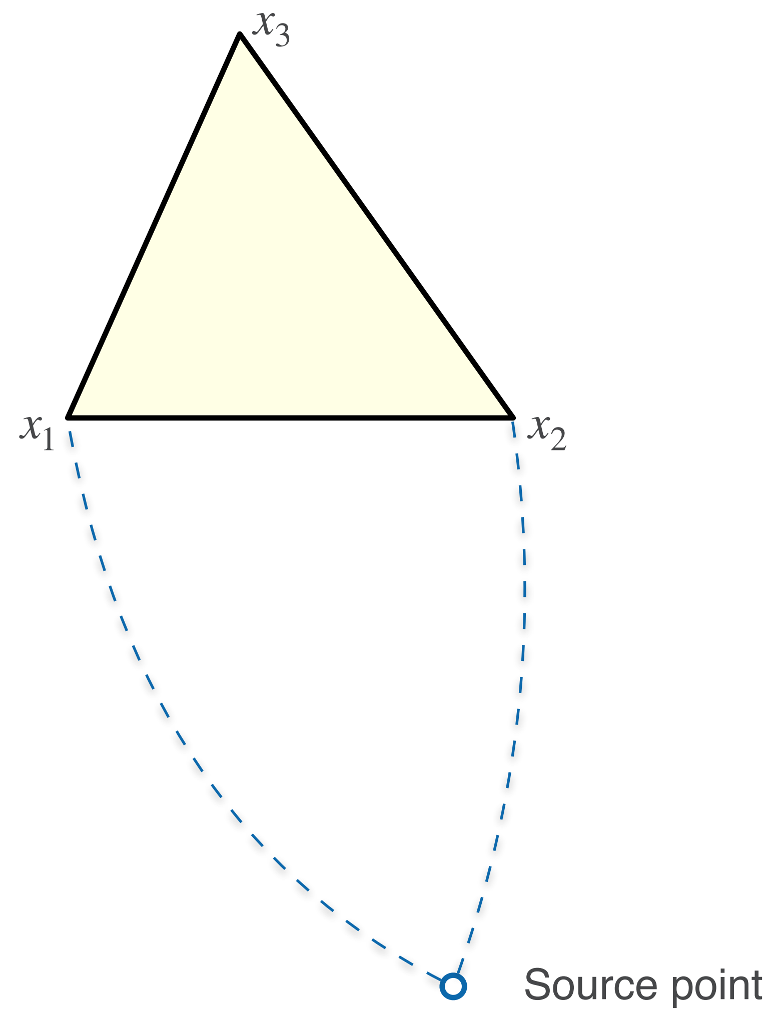
Fast Marching vs. Dijkstra

- Modified update step

- Update all triangles adjacent to a given vertex

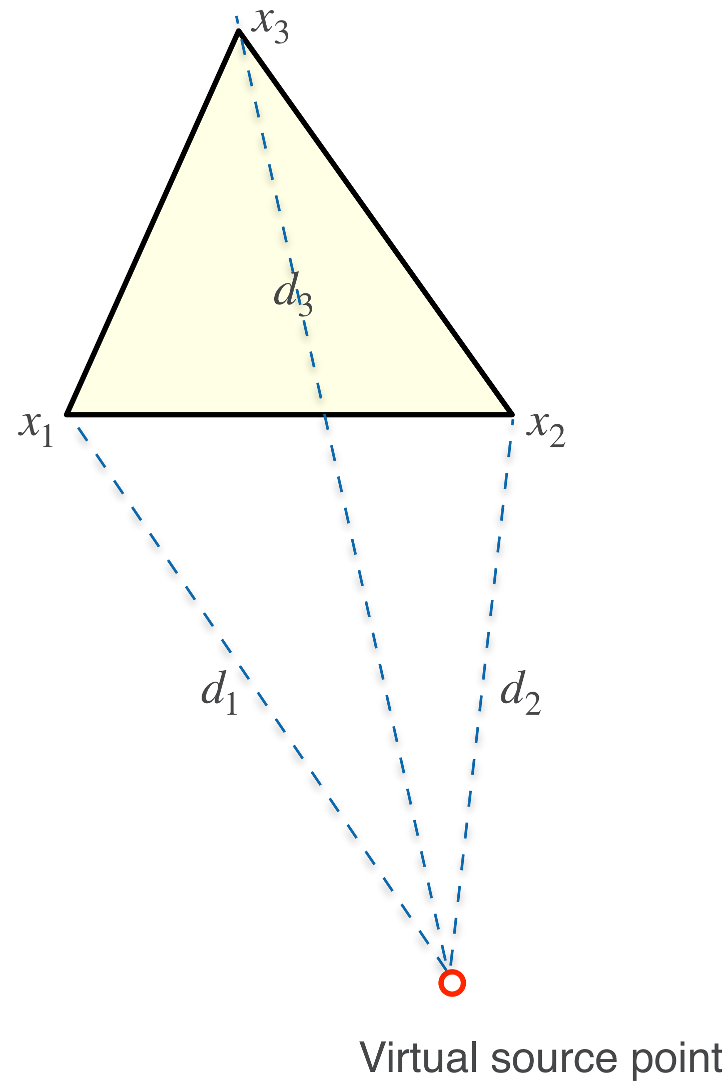
Fast Marching Algorithm

- At x_1 and x_2 stores the shortest paths d_1 and d_2
- Question: shortest path d_3 at x_3

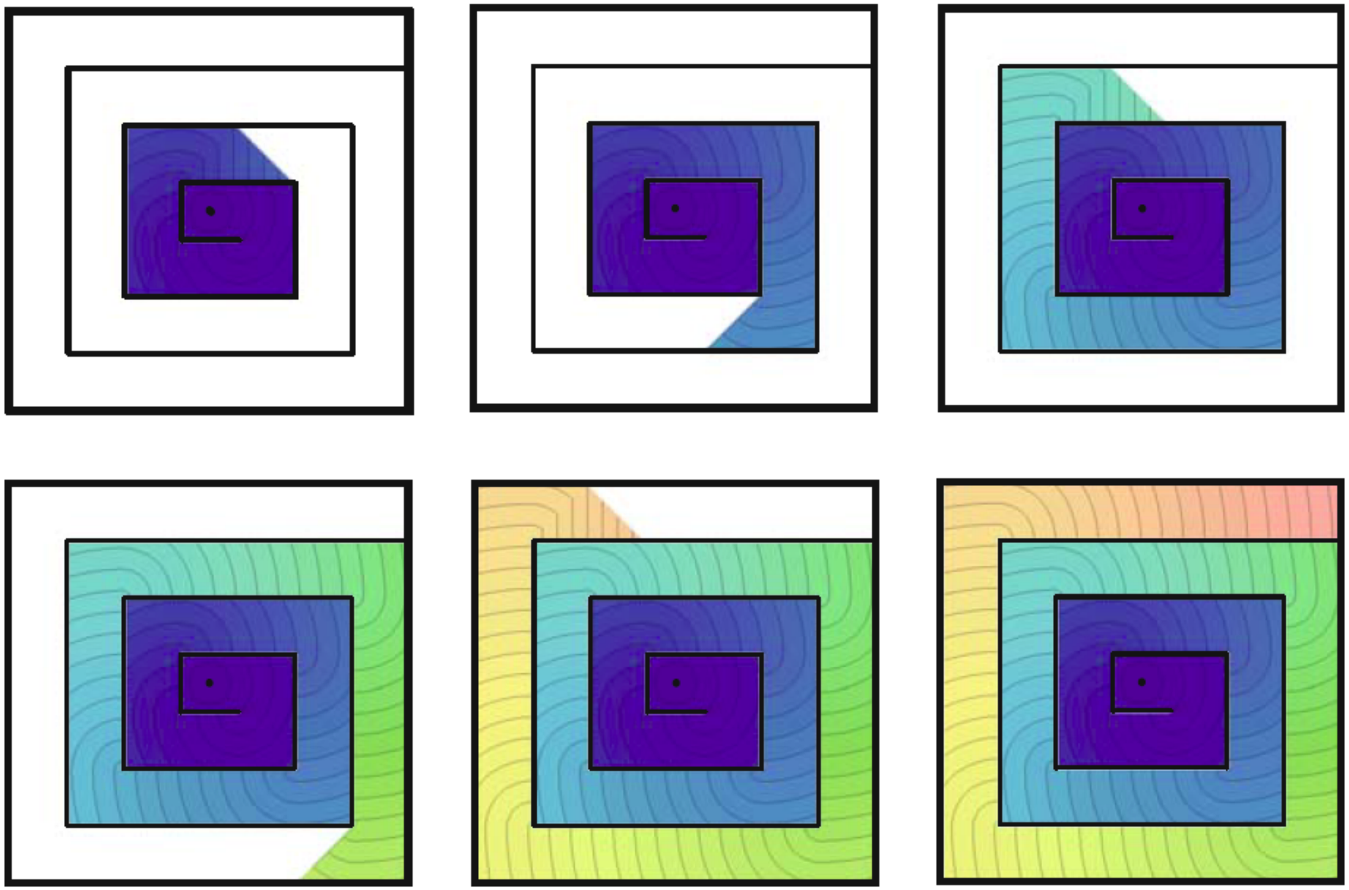


Fast Marching Algorithm

- Solution:
- On the plane containing $\triangle x_1x_2x_3$, build a “virtual” source point
-



Modifying Fast Marching

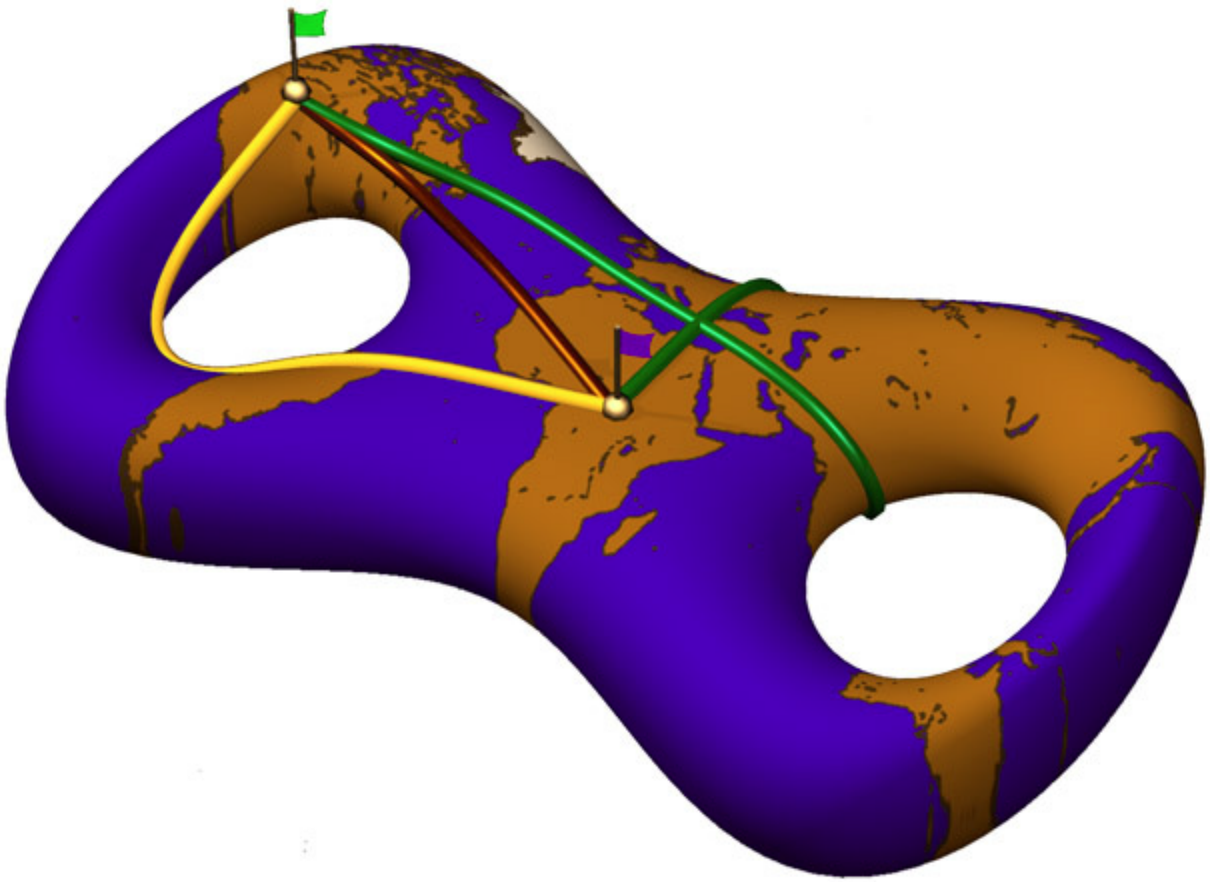


Raster scan and/or parallelize

Bronstein, *Numerical Geometry of Nonrigid Shapes*

Grids and parameterized surfaces

Tracing Geodesic Curves



Trace gradient of distance function

Practical Implementation

Fast Exact and Approximate Geodesics on Meshes

Vitaly Surazhsky, University of Oslo
Tatiana Surazhsky, University of Oslo
Danil Kirsanov, Harvard University
Steven J. Gortler, Harvard University
Hugues Hoppe, Microsoft Research

Abstract

The computation of geodesic paths and distances on triangle meshes is a common operation in many computer graphics applications. We present several practical algorithms for computing such geodesics from a source point to one or all other points efficiently. First, we describe an implementation of the exact "single source, all destination" algorithm presented by Mitchell, Mount, and Papadimitriou (MMP). We show that the algorithm runs much faster in practice than suggested by worst case analysis. Next, we extend the algorithm with a merging operation to obtain computationally efficient and accurate approximations with bounded error. Finally, to compute the shortest path between two given points, we use a lower-bound property of our approximate geodesic algorithm to efficiently prune the frontier of the MMP algorithm, thereby obtaining an exact solution even more quickly.

Keywords: shortest path, geodesic distance.

1 Introduction

In this paper we present practical methods for computing both exact and approximate shortest (i.e. geodesic) paths on a triangle mesh. These geodesic paths typically cut across faces in the mesh and are therefore not found by the traditional graph-based Dijkstra algorithm for shortest paths.

The computation of geodesic paths is a common operation in many computer graphics applications. For example, parameterizing a mesh often involves cutting the mesh into one or more charts (e.g. [Krishnamurthy and Levoy 1 [redacted]
the result generally has less distorti [redacted]
if the cuts are geodesic. Geodesic [redacted]

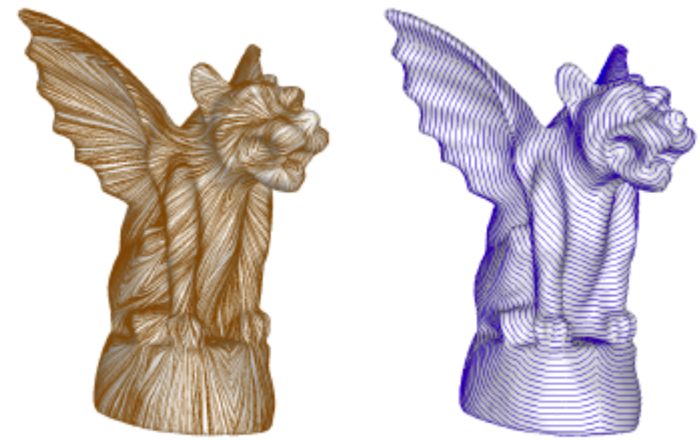


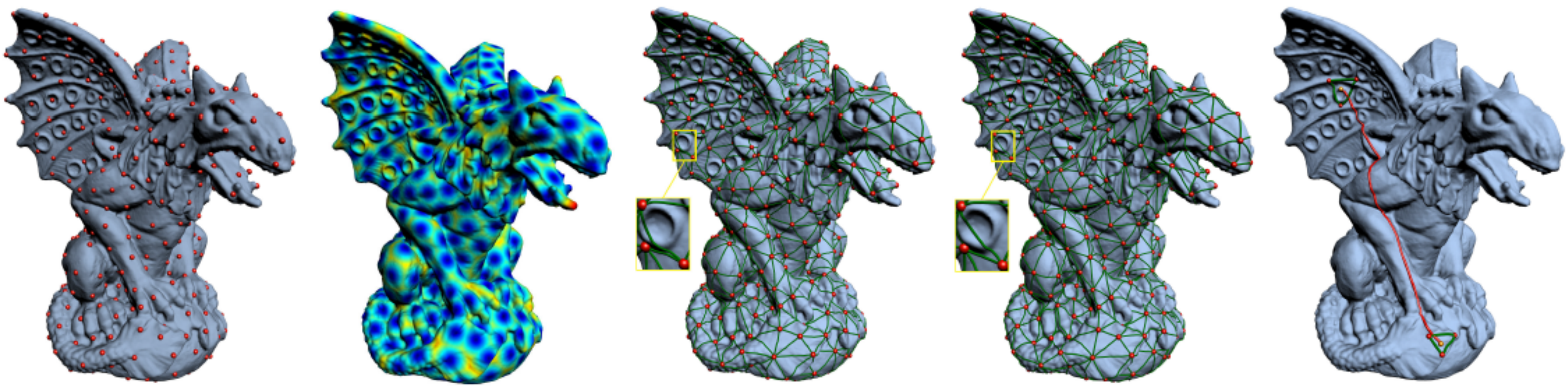
Figure 1: Geodesic paths from a source vertex, and isolines of the geodesic distance function.

tance function over the edges, the implementation is actually practical even though, to our knowledge, it has never been done previously. We demonstrate that the algorithm's worst case running time of $O(n^2 \log n)$ is pessimistic, and that in practice, the algorithm runs in sub-quadratic time. For instance, we can compute the exact geodesic distance from a source point to all vertices of a 400K-triangle mesh in about one minute.

Approximation algorithm We extend the algorithm with a merging operation to obtain computationally efficient and accurate approximations with *bounded* error. In practice, the algorithm runs in [redacted]

http://code.google.com/p/geodesic/

All-Pairs Distances



Sample points | Geodesic field | Triangulate (Delaunay) | Fix edges | Query (planar embedding)

Xin, Ying, and He. "Constant-time all-pairs geodesic distance query on triangle meshes." I3D 2012.

Geodesic Voronoi & Delaunay

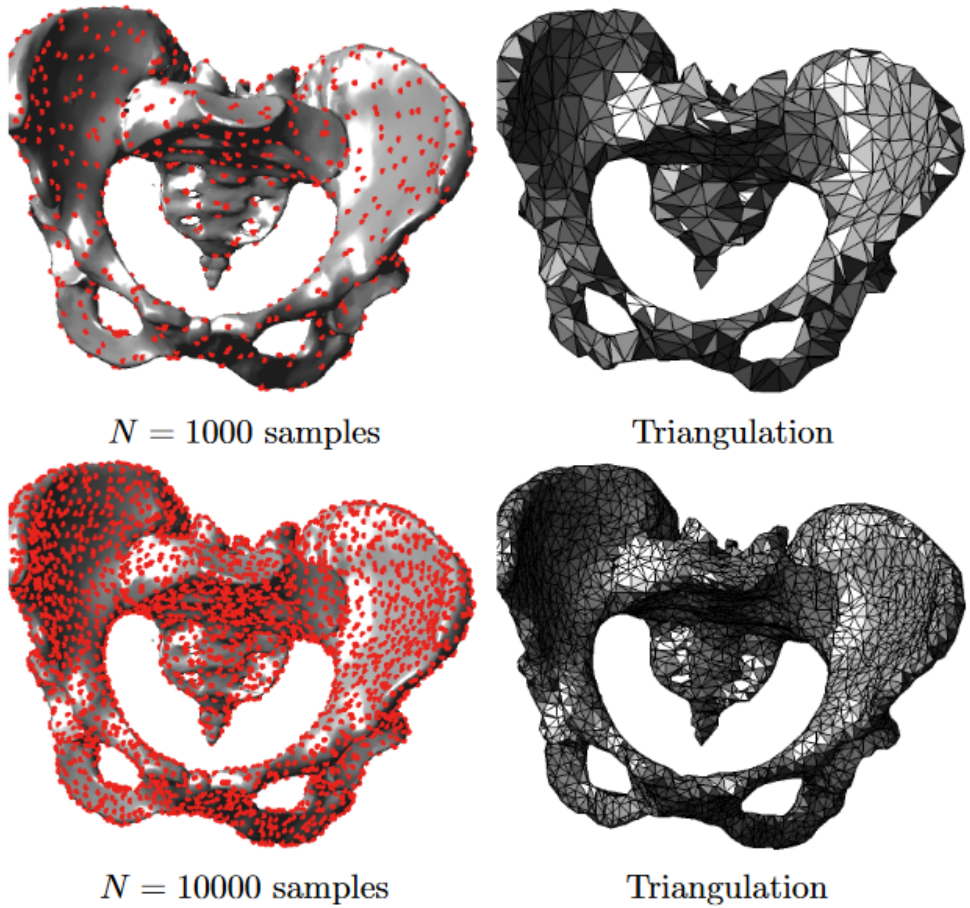


Fig. 4.12 *Geodesic remeshing with an increasing number of points.*

From *Geodesic Methods in Computer Vision and Graphics* (Peyré et al., FnT 2010)