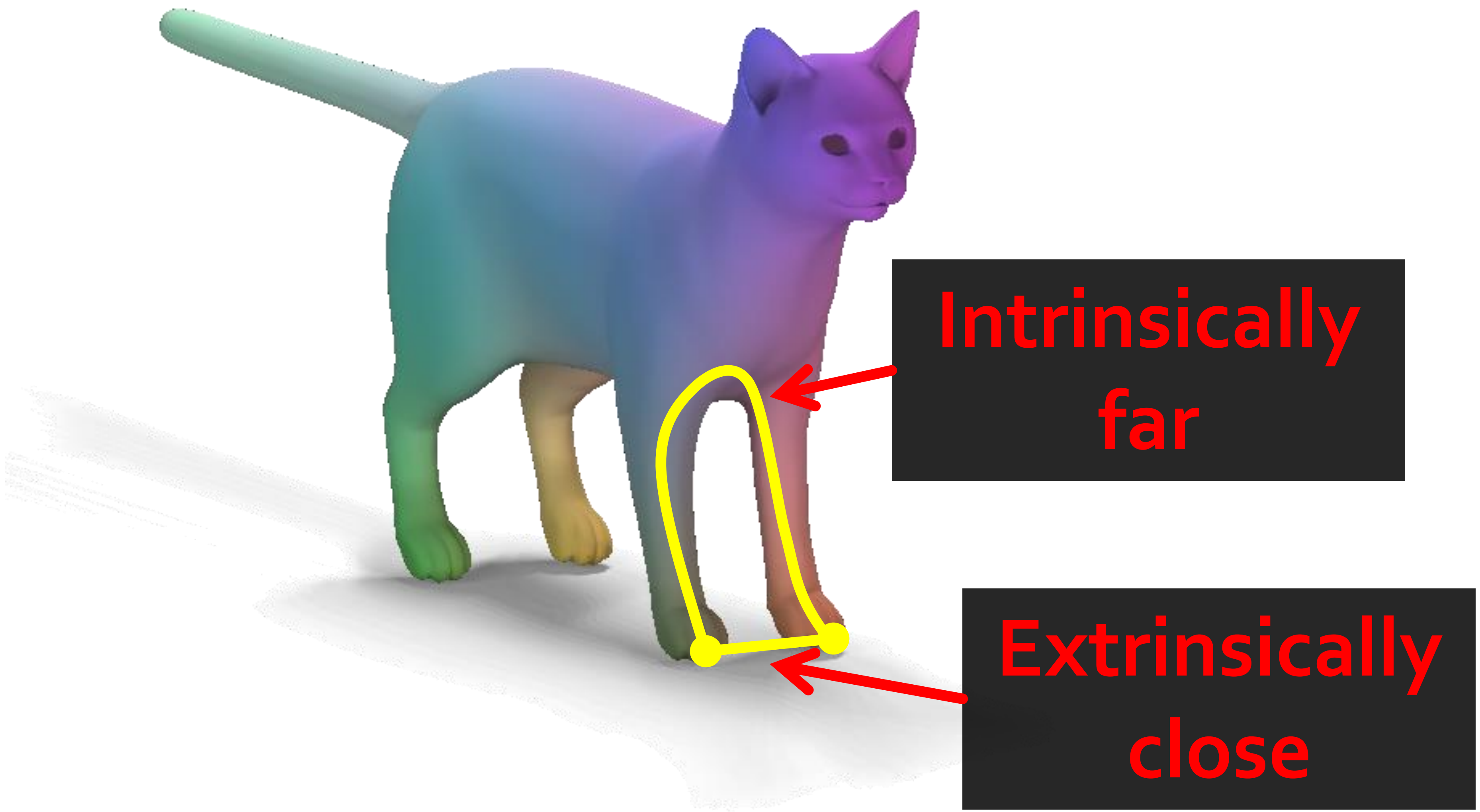
Geodesic Distances: Intro & Theory

Justin Solomon

6.8410: Shape Analysis
Spring 2023

MIT EECS

Geodesic Distances

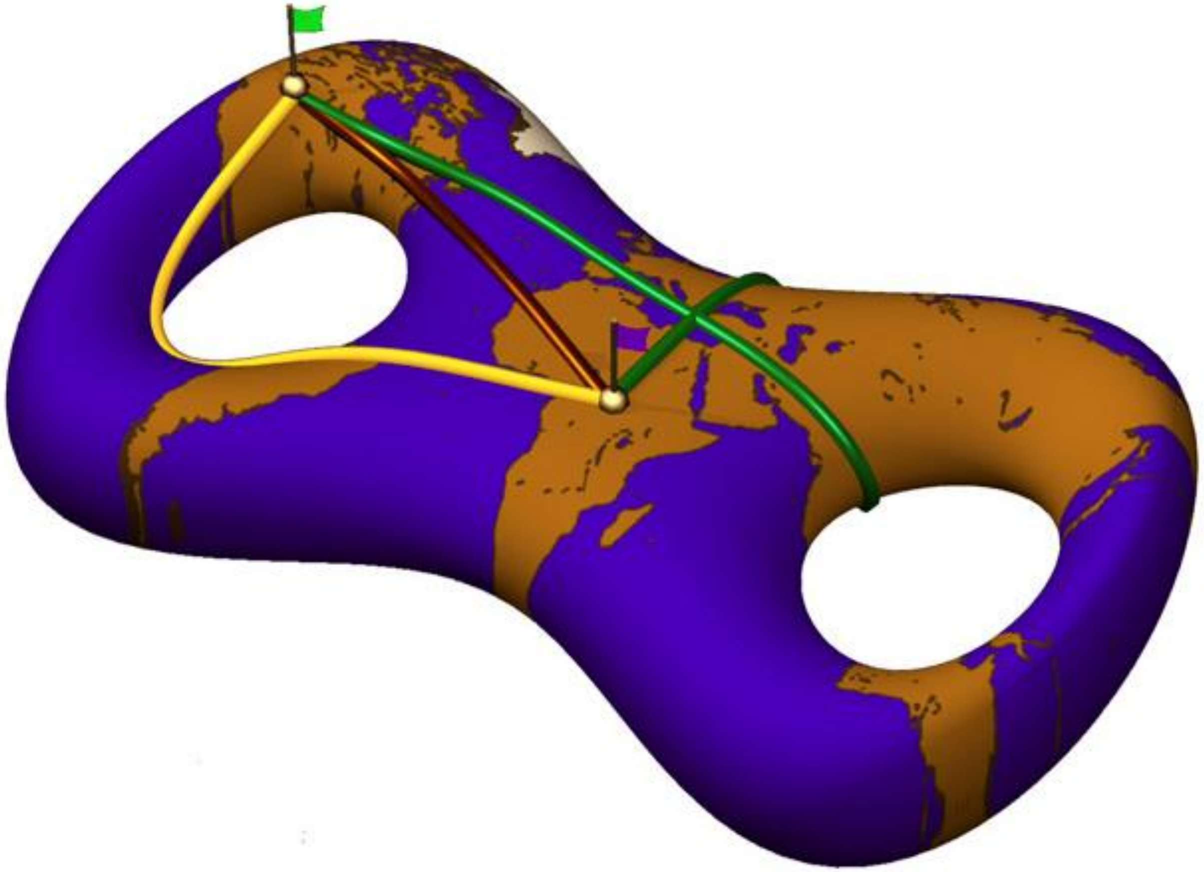


Geodesic distance

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

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

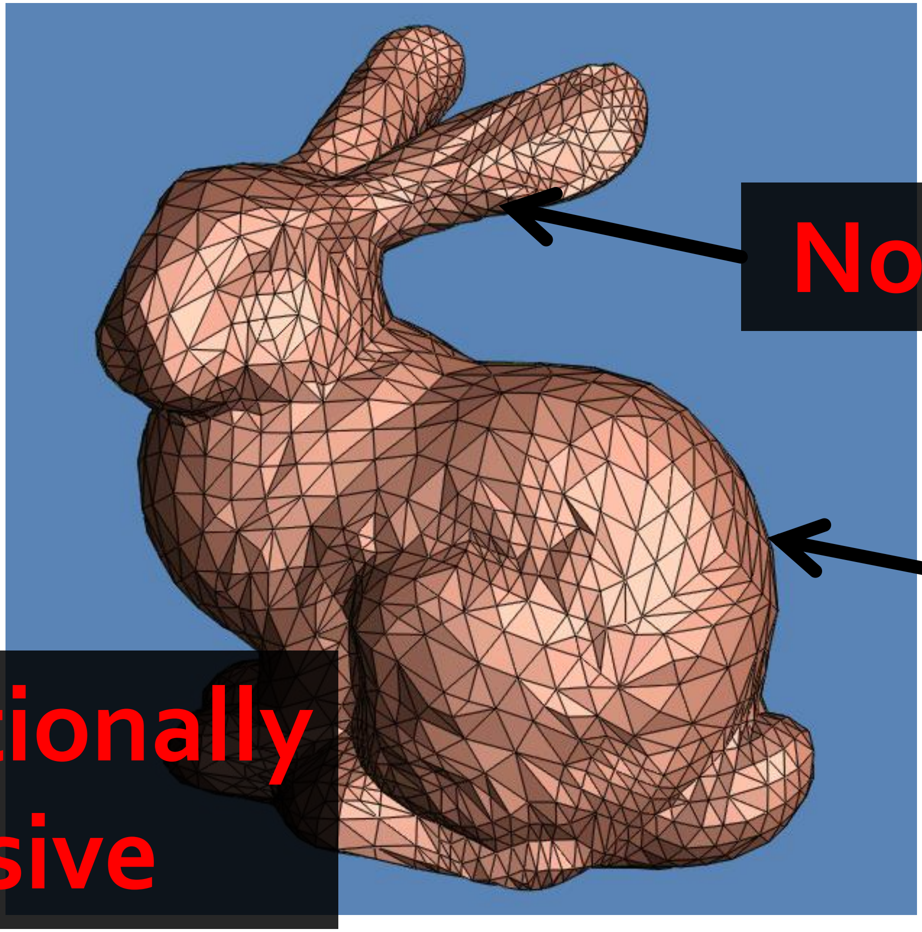
Complicated Problem



Straightest Geodesics on Polyhedral Surfaces (Polthier and Schmies)

Local minima

Reality Check



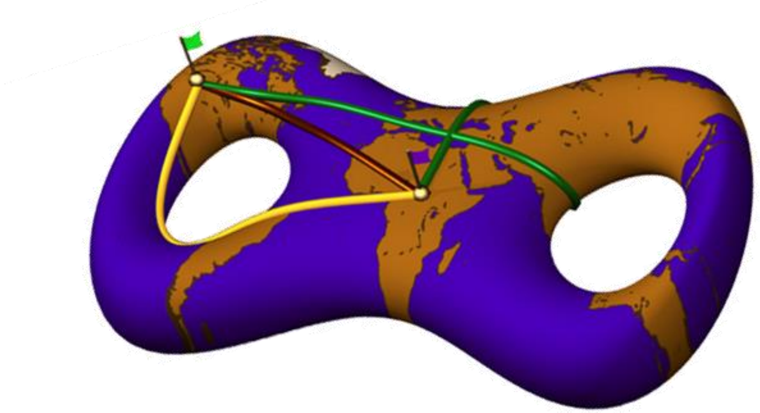
Not OK

OK

Computationally expensive

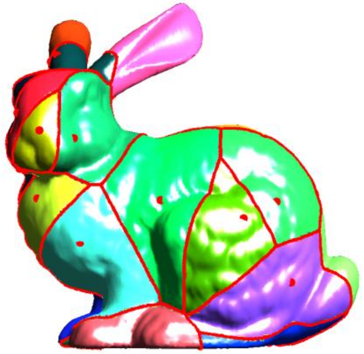
Extrinsic may suffice for near vs. far

Related Queries

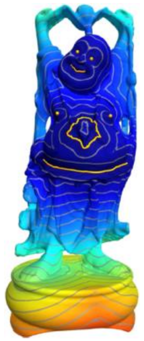


Locally short

Single source



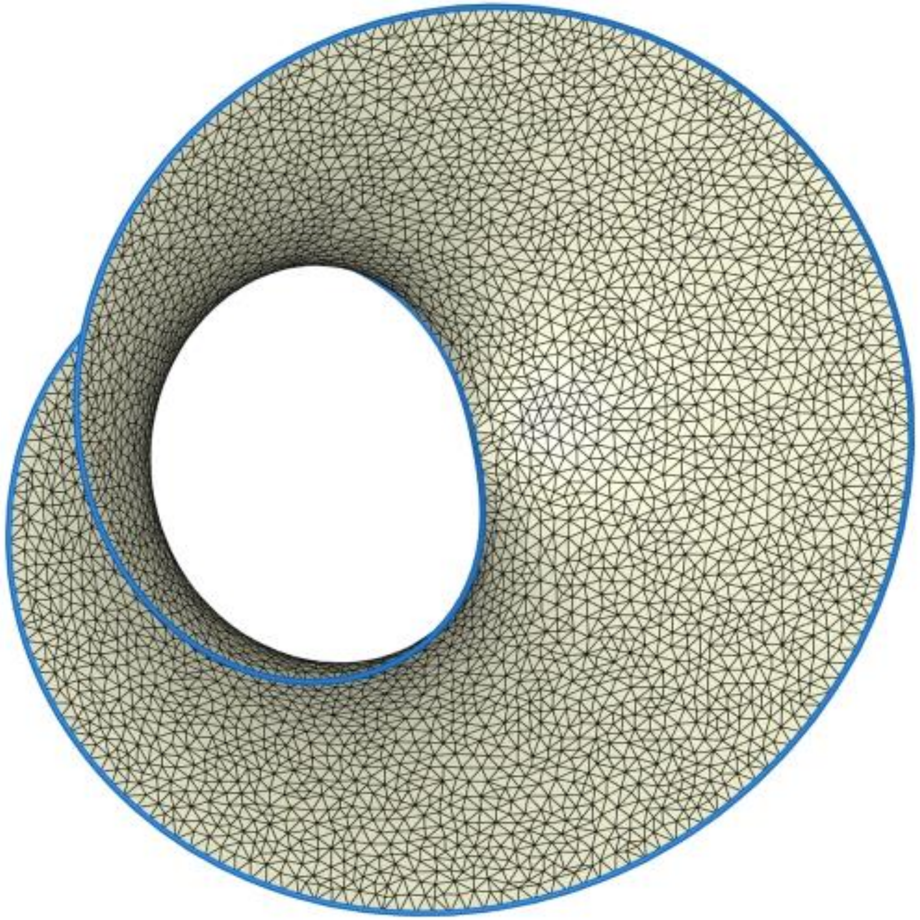
Multi-source



All-pairs

https://www.ceremade.dauphine.fr/~peyre/teaching/manifold/tp3.html http://www.sciencedirect.com/science/article/pii/S0010448511002260

Computer Scientists' Approach

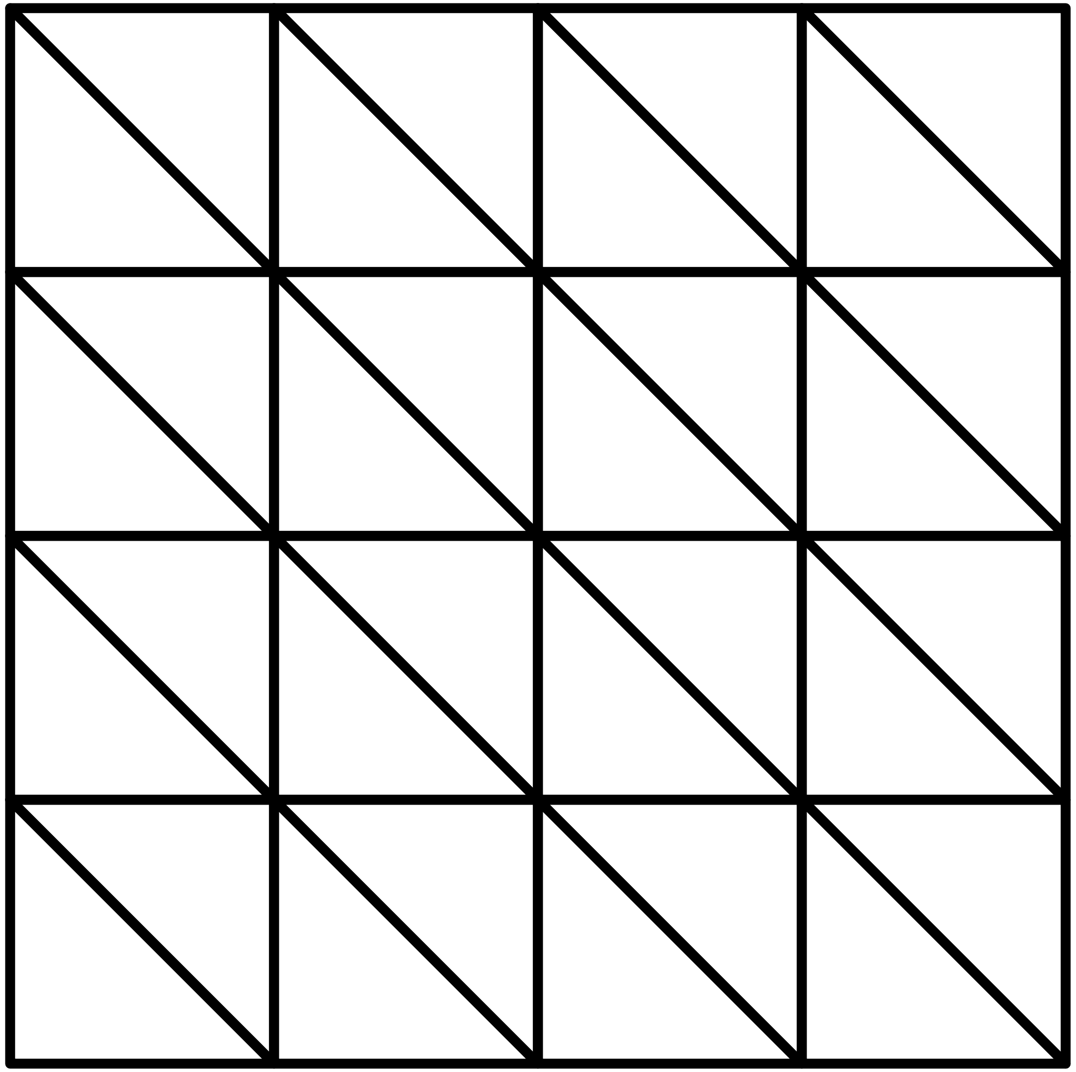


Approximate geodesics as paths along edges

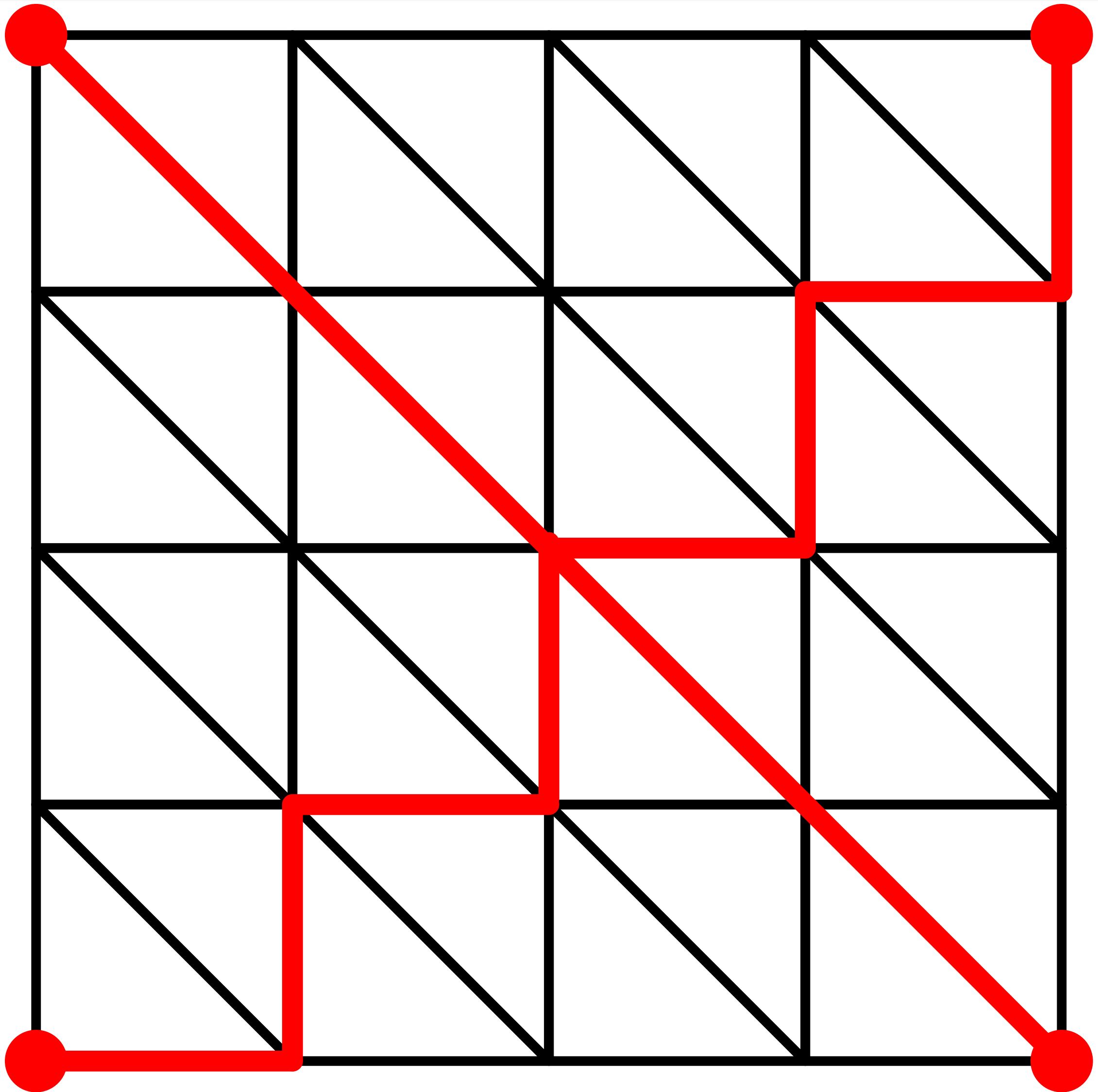
http://www.cse.ohio-state.edu/~tamaldey/isotopic.html

Meshes are graphs

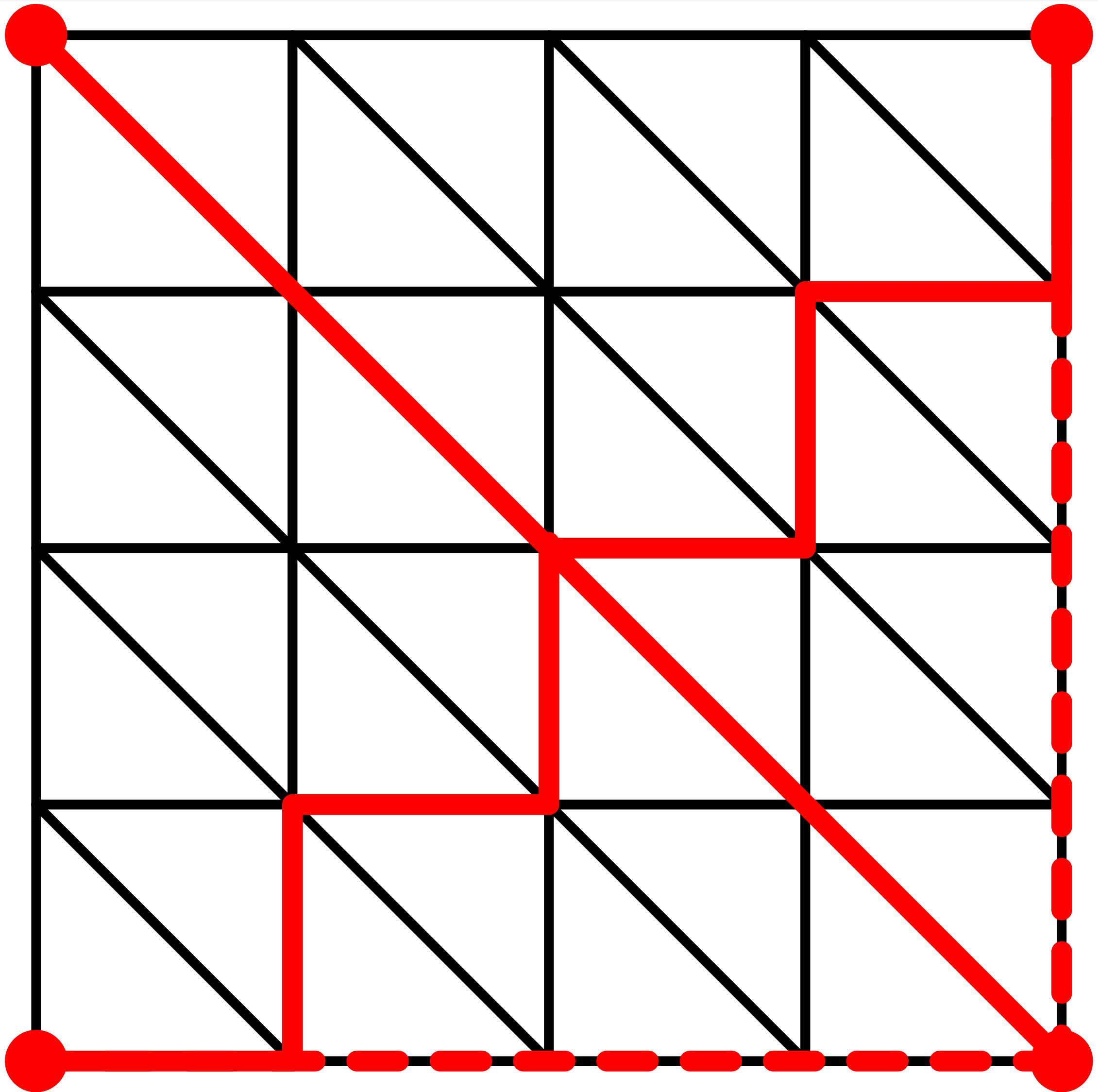
Pernicious Test Case



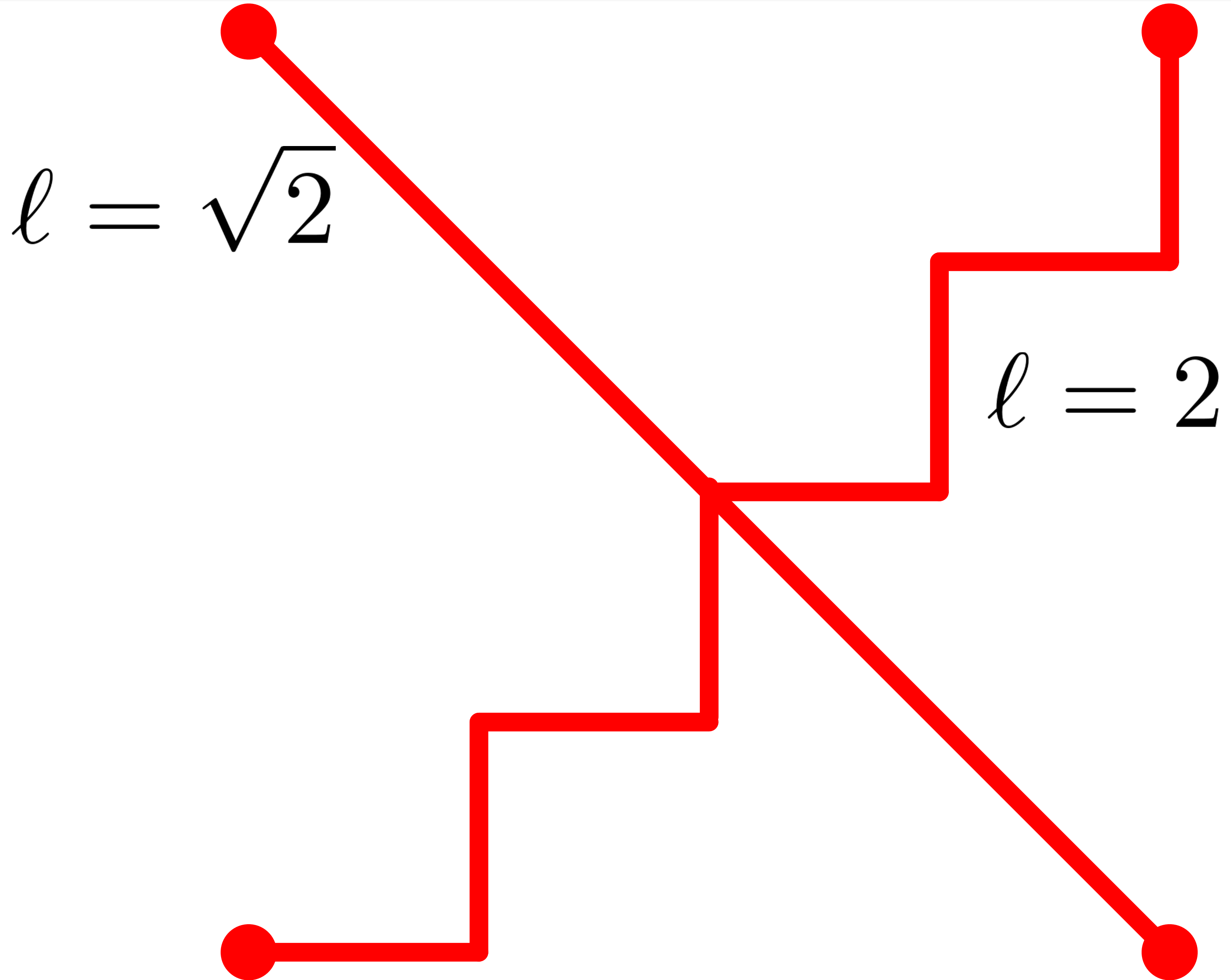
Pernicious Test Case



Pernicious Test Case



Distances



What Happened

- Asymmetric
- Anisotropic
- May not improve under refinement

Conclusion 1

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

Often an acceptable approximation.

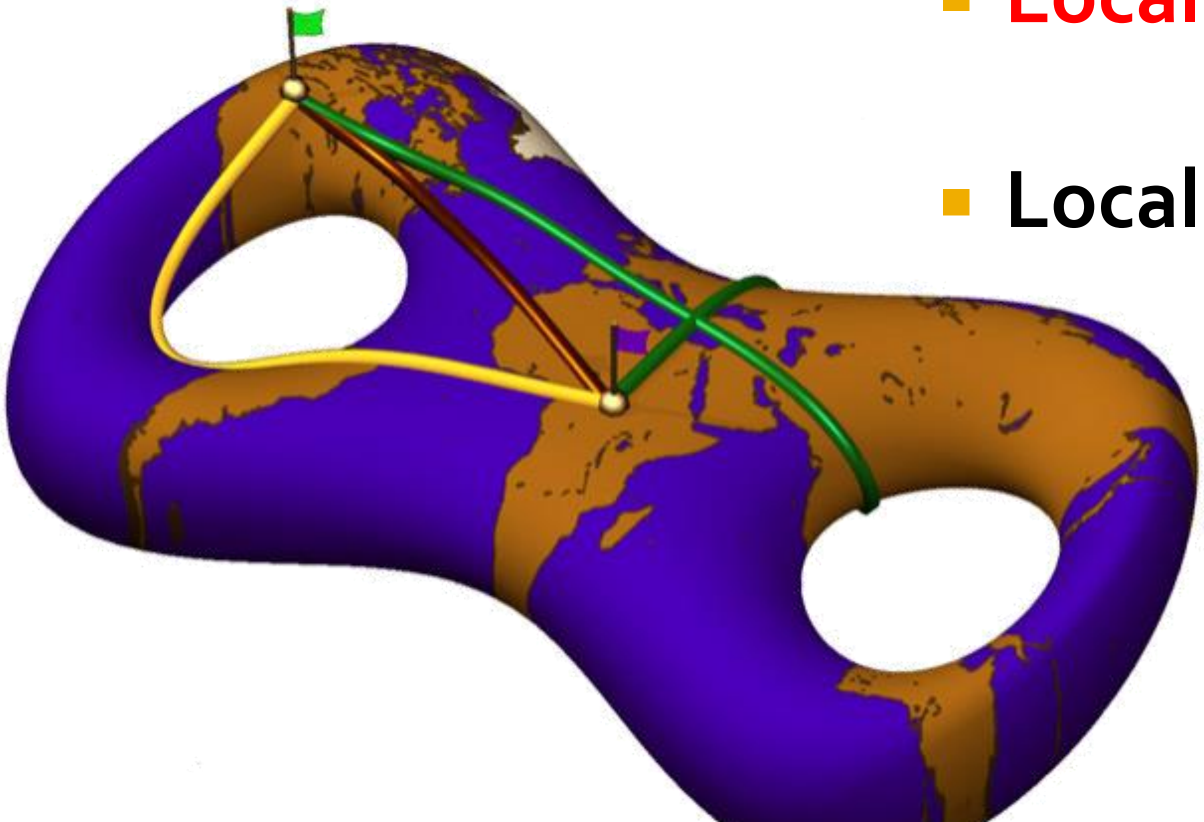
Conclusion 2

Geodesic distances
need special discretization.

So, we need to understand the theory!

Three Possible Definitions

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



Not the same!

Recall:

Arc Length

$$\int_a^b \|\gamma'(t)\|_2 \, dt$$

Geodesic Distance: Global Definition

Definition (Geodesic distance). *The* geodesic distance *between two points* $\mathbf{p}, \mathbf{q} \in \mathcal{M}$ *on a submanifold* $\mathcal{M}$ *is given by*

$$d_{\mathcal{M}}(\mathbf{p}, \mathbf{q}) := \begin{cases} \inf_{\gamma:[0,1]\to\mathcal{M}} & L[\gamma] \\ \quad \textit{subject to} & \gamma(0) = \mathbf{p} \\ & \gamma(1) = \mathbf{q} \\ & \gamma \in C^1([0,1]). \end{cases}$$

Here, the curve γ *connects* $\mathbf{p}$ *to* $\mathbf{q}$*, and we are minimizing arc length as defined in* (3.2)*. A curve* γ *realizing this infimum is known as a* global (minimizing) geodesic *curve.*

Energy of a Curve

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

Easier to work with:

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

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

Equality exactly when parameterized with constant speed.

First Variation of Arc Length

Proposition *Let* $\gamma_t : [a, b] \to \mathcal{M}$ *be a family of curves with fixed endpoints* $\mathbf{p}, \mathbf{q} \in \mathcal{M}$ *on submanifold* $\mathcal{M}$*, and for convenience assume* γ *is parameterized by arc length at* $t = 0$*. Then,*

$$\frac{d}{dt}E[\gamma_t] = -\int_a^b \left(\frac{d\gamma_t(s)}{dt} \cdot \mathrm{proj}_{T_{\gamma_t(s)}\mathcal{M}}[\gamma_t''(s)] \right) ds.$$

Here, we do not assume s *is an arc length parameter when* $t \neq 0$*.*

First Variation of Arc Length

Proposition *If a curve $\gamma : [a,b] \to \mathcal{M}$ is a geodesic, then*

$$\mathrm{proj}_{T_{\gamma(s)}\mathcal{M}}[\gamma''(s)] \equiv 0$$

for $s \in (a,b)$.

Intuition

$$\text{proj}_{T_{\gamma(s)}\mathcal{M}}\left[\gamma''(s)\right] \equiv 0$$

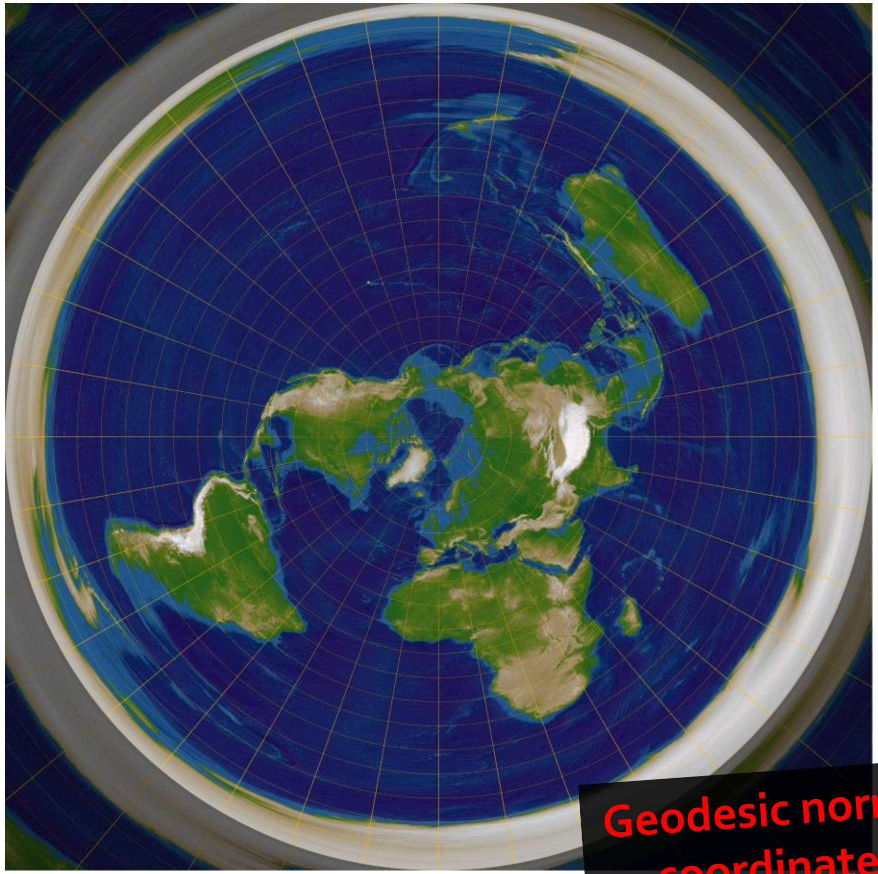
- The only acceleration is out of the surface
  - No steering wheel!

Two Local Perspectives

$$\mathrm{proj}_{T_{\gamma(s)}\mathcal{M}}\left[\gamma''(s)\right] \equiv 0$$

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

Exponential Map

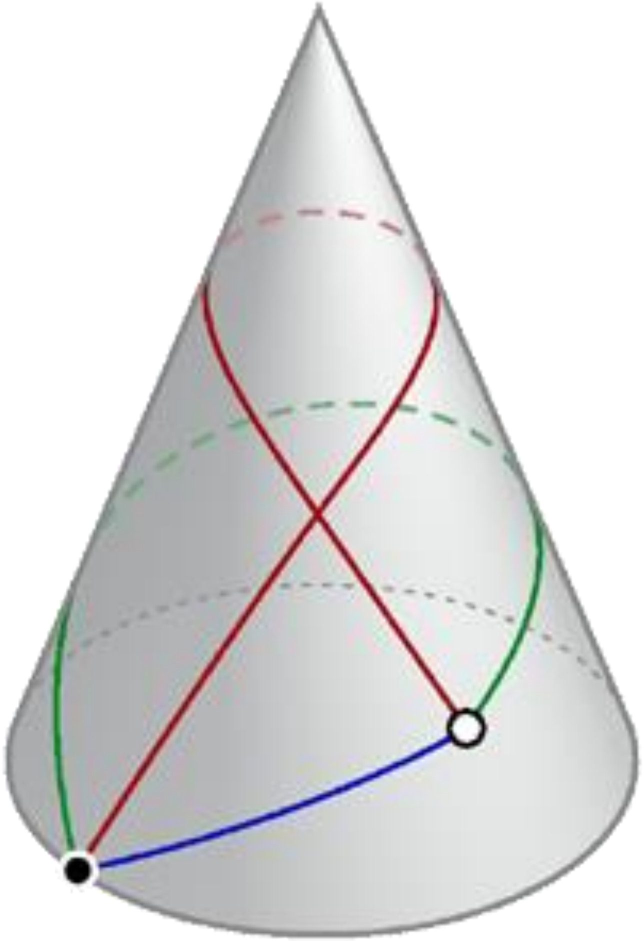


https://en.wikipedia.org/wiki/Exponential_map_(Riemannian_geometry)

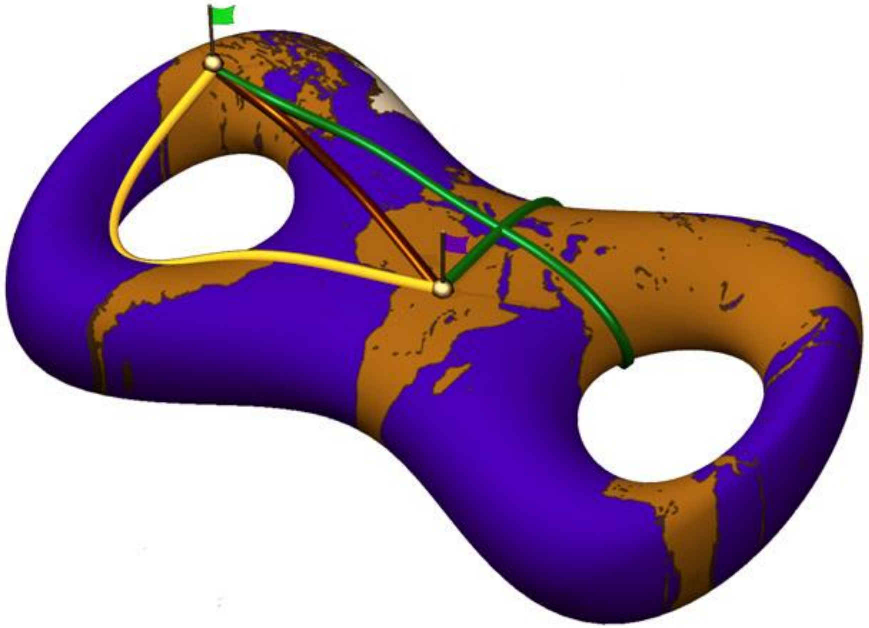
$$\exp_{\mathbf{p}}(\mathbf{v}) := \gamma_{\mathbf{v}}(1)$$

$\gamma_v(1)$ where γ_v is (unique) geodesic from *p* with velocity *v*.

Instability of Geodesics

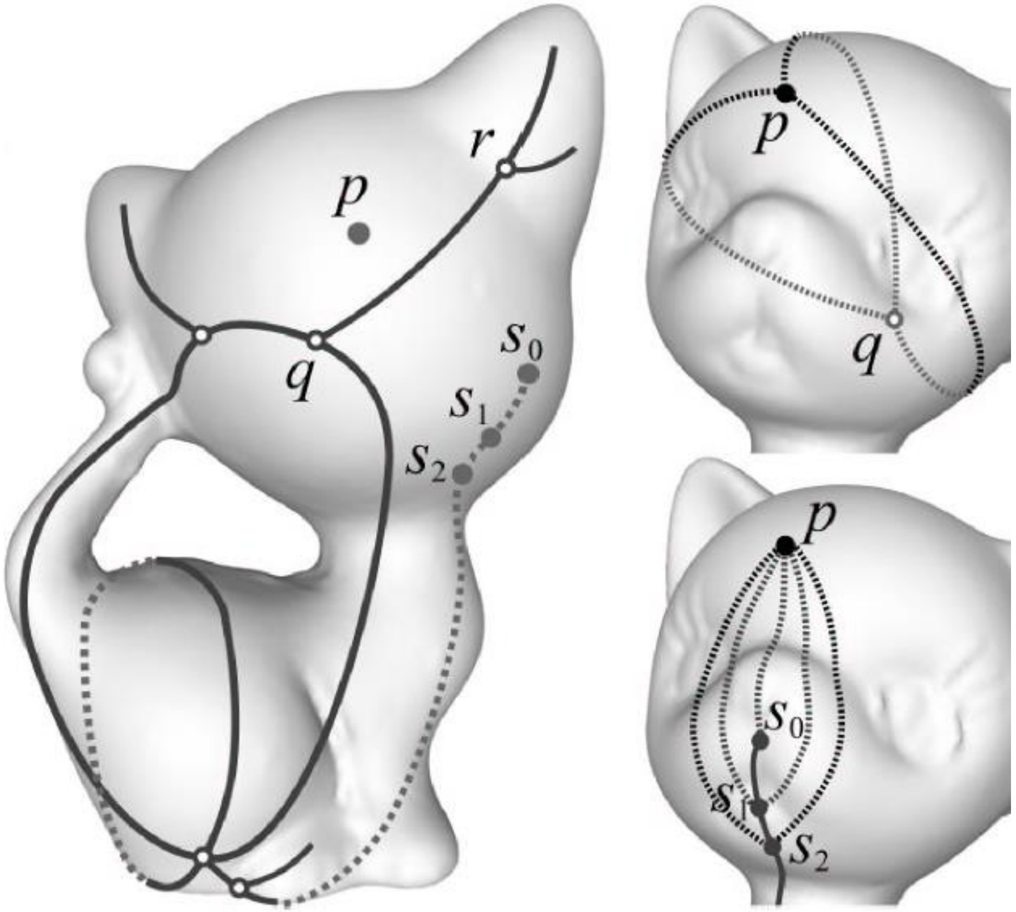


http://parametricwood2011.files.wordpress.com/2011/01/cone-with-three-geodesics.png



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

Cut Locus



"Cut Logus and Topology from Surface Point Data" (Dey & Li 2009)

Cut point:

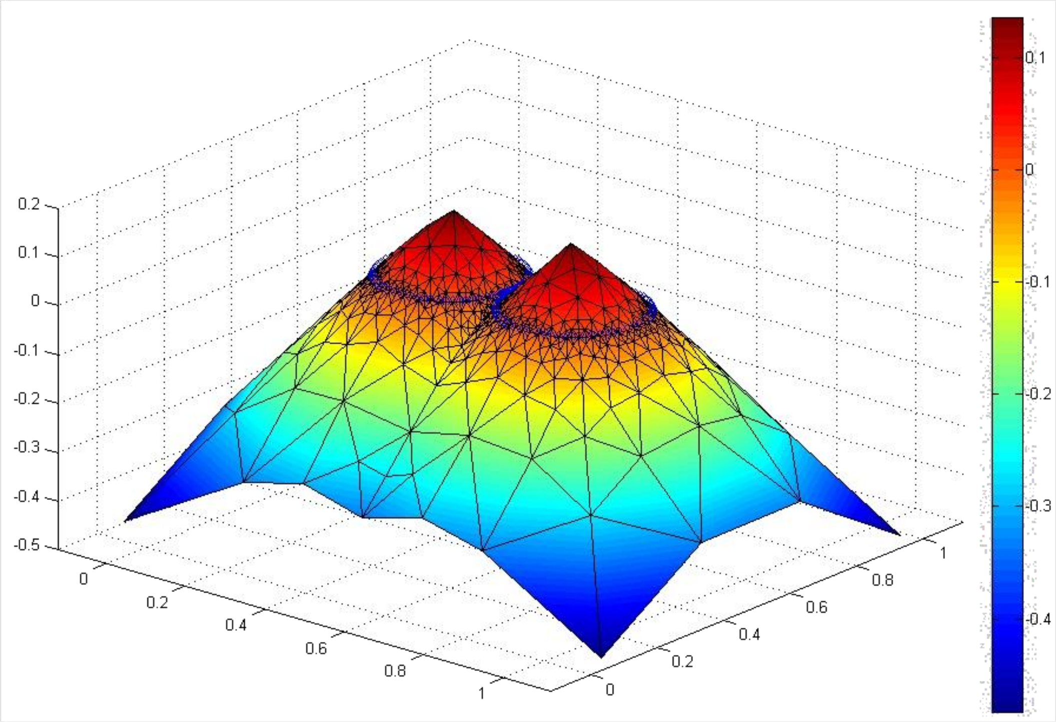
Point where geodesic ceases to be minimizing

Set of cut points from a source *p*

Eikonal Equation

$$\|\nabla u(\mathbf{p})\|_2 = 1 \ \forall \mathbf{p} \in \mathcal{M}$$

eikonal = “image” (Greek)



https://www.mathworks.com/matlabcentral/fileexchange/24827-hamilton-jacobi-solver-on-unstructured-triangular-grids/content/HJB_Solver_Package/@SolveEikonal/SolveEikonal.m

Geodesic Distances: Intro & Theory

Justin Solomon

6.8410: Shape Analysis

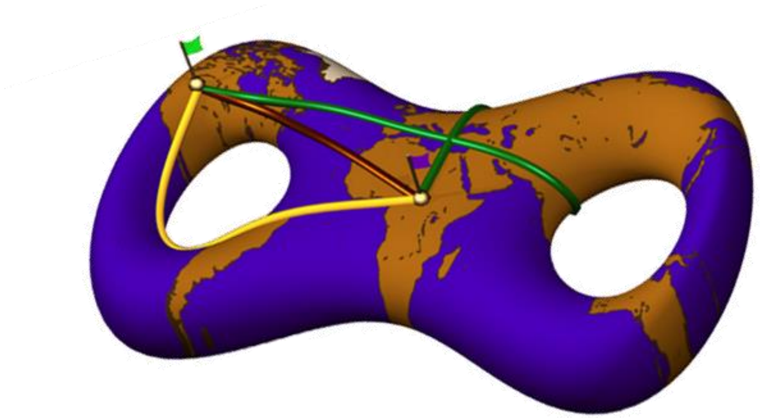
Spring 2023

Geodesic Distances: Algorithms

Justin Solomon

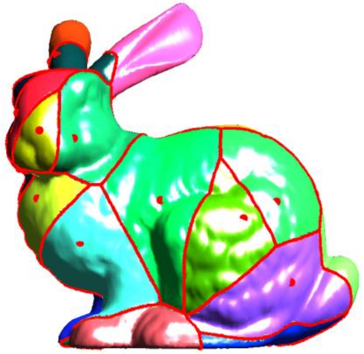
6.8410: Shape Analysis
Spring 2023

Reminder: Geodesic Distance Queries

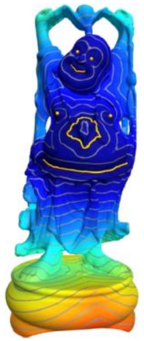


Locally short

Single source



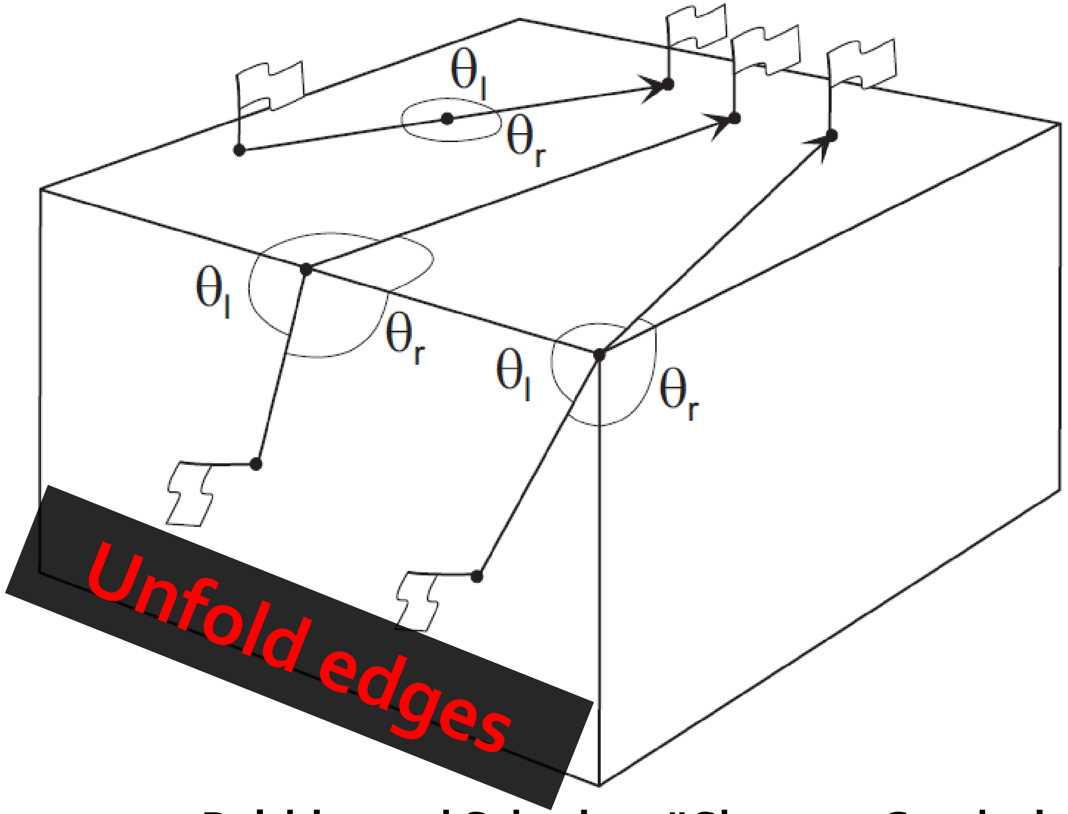
Multi-source



All-pairs

https://www.ceremade.dauphine.fr/~peyre/teaching/manifold/tp3.html http://www.sciencedirect.com/science/article/pii/S0010448511002260

Initial Value Problem: Straightest Geodesics

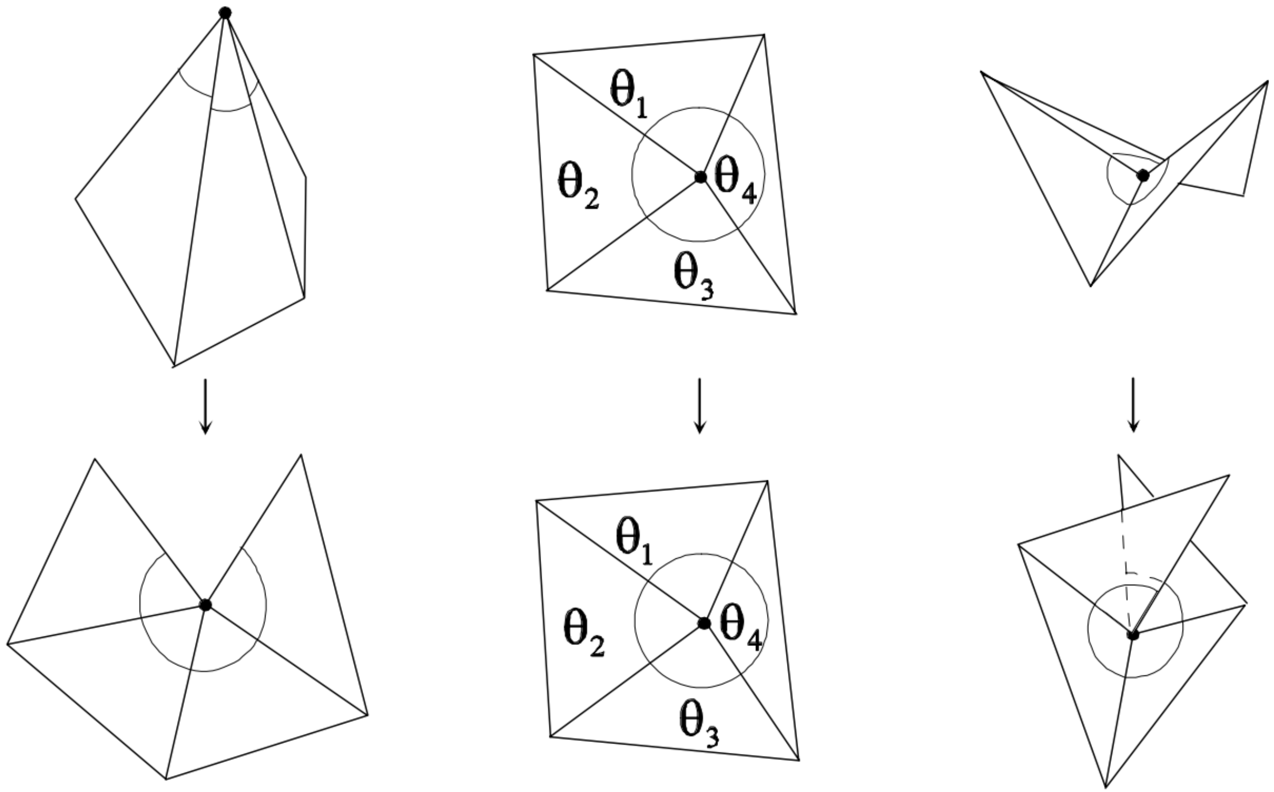


Equal left and right angles

Polthier and Schmies. "Shortest Geodesics on Polyhedral Surfaces." SIGGRAPH course notes 2006.

Trace a single geodesic exactly

Intuition: Unfolding



Spherical Vertex
$2\pi - \Sigma\theta_i > 0$

Euclidean Vertex
$2\pi - \Sigma\theta_i = 0$

Hyperbolic Vertex
$2\pi - \Sigma\theta_i < 0$

Are They Shortest Paths?

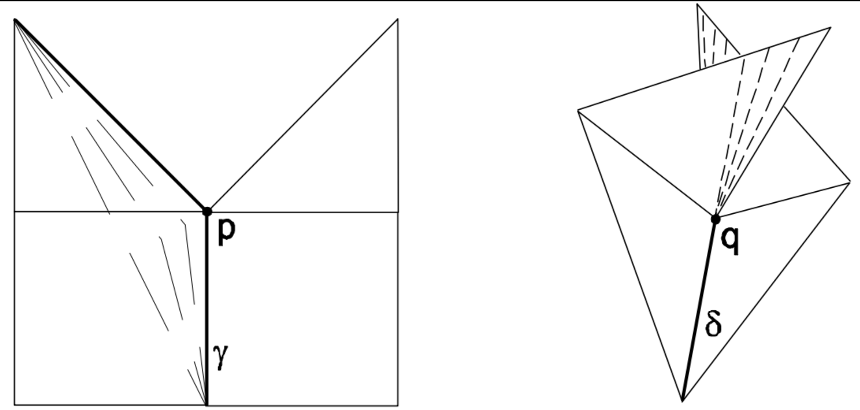


Figure 5: Locally shortest geodesics cannot be extended through a spherical vertex p and there exist multiple continuations at a hyperbolic vertex q.

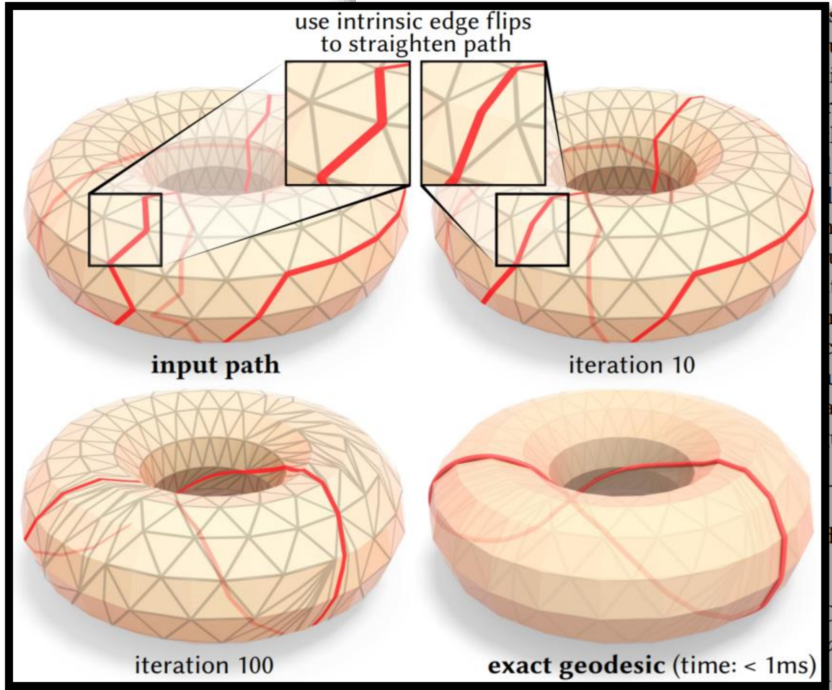
K>0 (spherical): Straightest geodesic is never shortest
K<0 (hyperbolic): Multiple shortest but one straightest

New Algorithm for Geodesic Paths

You Can Find Geodesic Paths in Triangle Meshes by Just Flipping Edges

NICHOLAS SHARP and KEENAN CRANE, Carnegie Mellon University

This paper introduces a new approach to computing geodesics on polyhedral surfaces—the basic idea is to iteratively perform *edge flips*, in the same spirit as the classic Delaunay flip algorithm. This process also produces a
sics, which is immediately
umerical simulation. More
iven sequence of edges into
rossings (formally: it finds a
hm is guaranteed to termi-
runtimes are on the order
ions of triangles. The same
ple paths, including closed
urves. We explore how the
ts and segmentation bound-
ng the notion of *constrained*
es, and providing accurate
uations (PDEs). Evaluation
tes that the method is both
ulations.



→ **Shape modeling**.

dge flip, triangulation

Can Find Geodesic Paths in
ans. Graph. 39, 6, Article 249
45/3414685.3417839

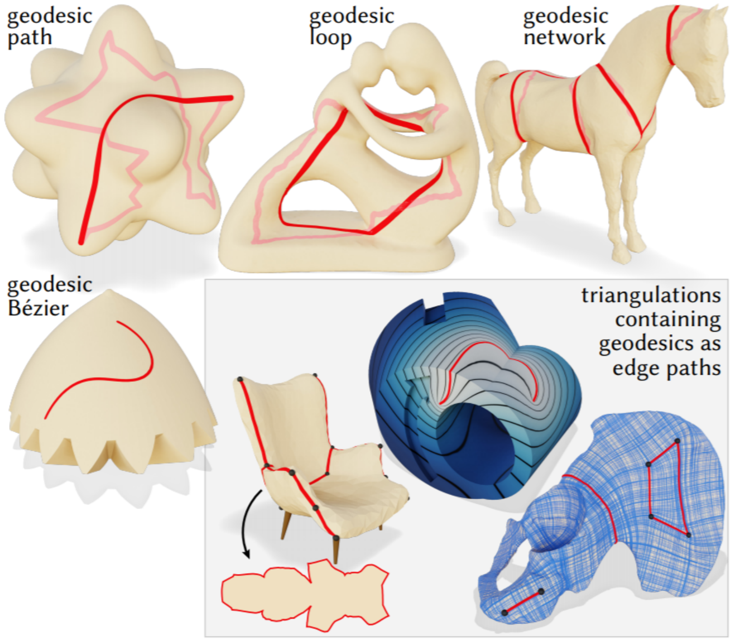


Fig. 1. We introduce an edge-flip based algorithm for computing geodesic paths, loops, and networks on triangle meshes. The algorithm also yields a triangulation containing these curves as edges, which can be used directly for subsequent geometry processing (*e.g.*, for cutting, or for solving PDEs).

1 INTRODUCTION

A *geodesic* is the natural generalization of a straight line to a curved surface: it is a trajectory of zero acceleration, or equivalently, a

but rather to find locally shortest curves within the given isotopy class, *i.e.*, to "pull the given curves tight."

Importantly, geodesics are *intrinsic*: they do not depend at all

SIGGRAPH Asia 2020

Globally Shortest Path?

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

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

$$d(v) = \text{Current distance to vertex } v$$

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

Initialization:

$$d(v_0) = 0$$

$$d(v) = \infty \ \forall v \in V \backslash \{v_0\}$$

$$S = \{\}$$

Dijkstra's Algorithm

$$v_0 = \text{Source vertex}$$
$$d(v) = \text{Current distance to vertex } v$$
$$S = \text{Vertices with known optimal distance}$$

Iteration *k*:

$$v = \arg\min_{v \in V \setminus S} d(v)$$

$$S \leftarrow S \cup \{v\}$$

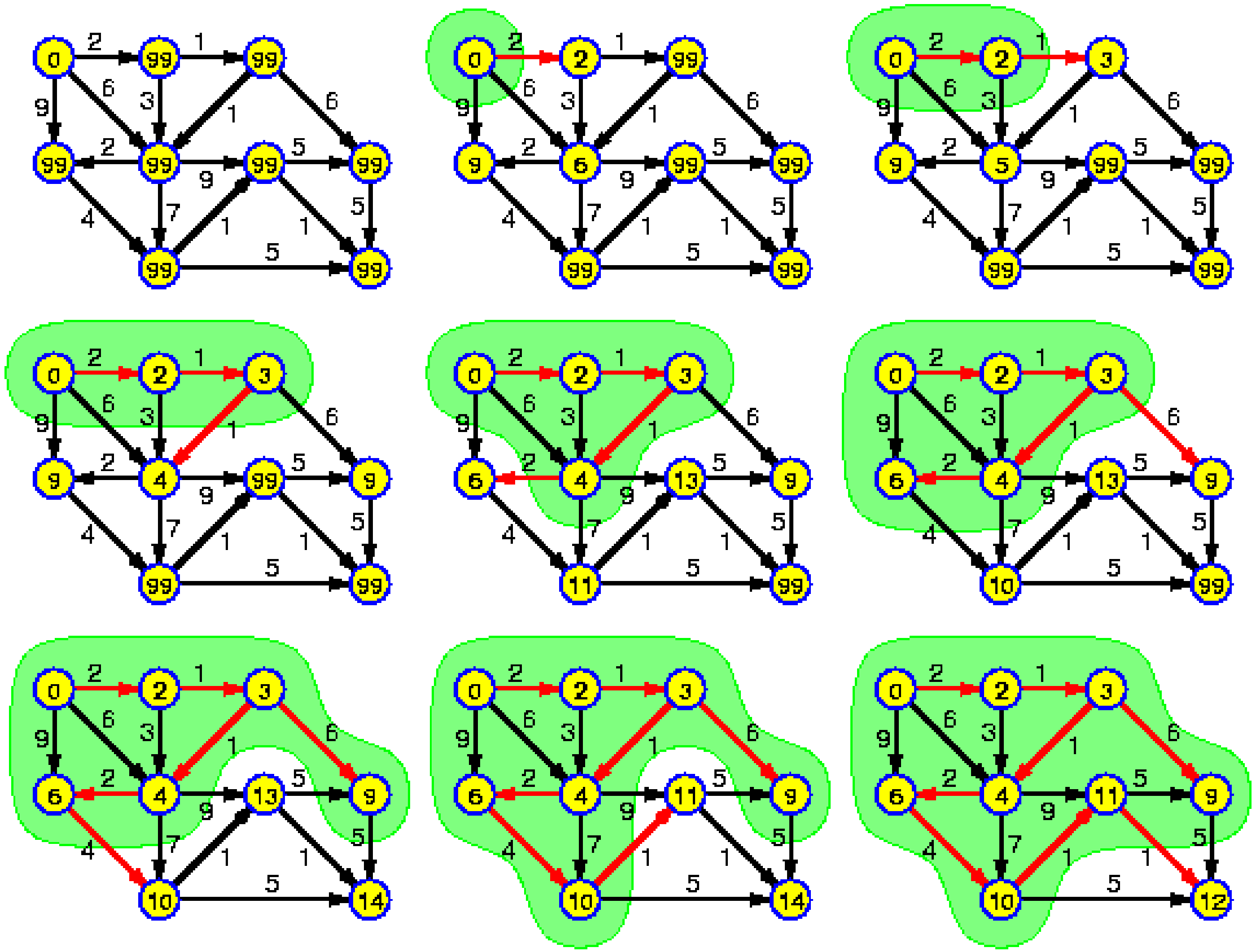
$$d(u) \leftarrow \min\{d(u), d(v) + w(e)\}\ \forall e = (u,v) \in E$$

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

$$O(|E| + |V| \log |V|)$$

Advancing Fronts

Example

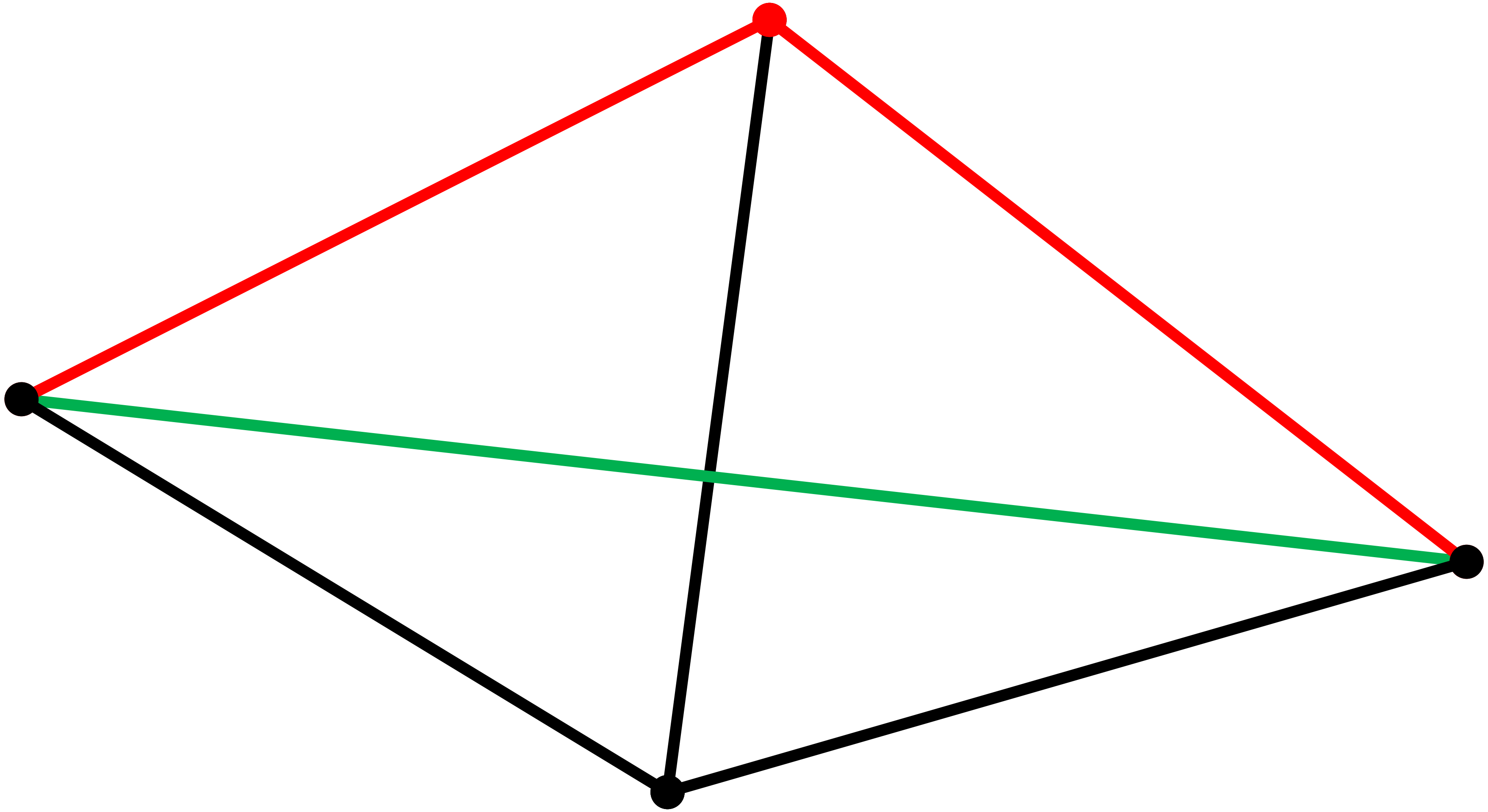


http://www.iekucukcay.com/wp-content/uploads/2011/09/dijkstra.gif

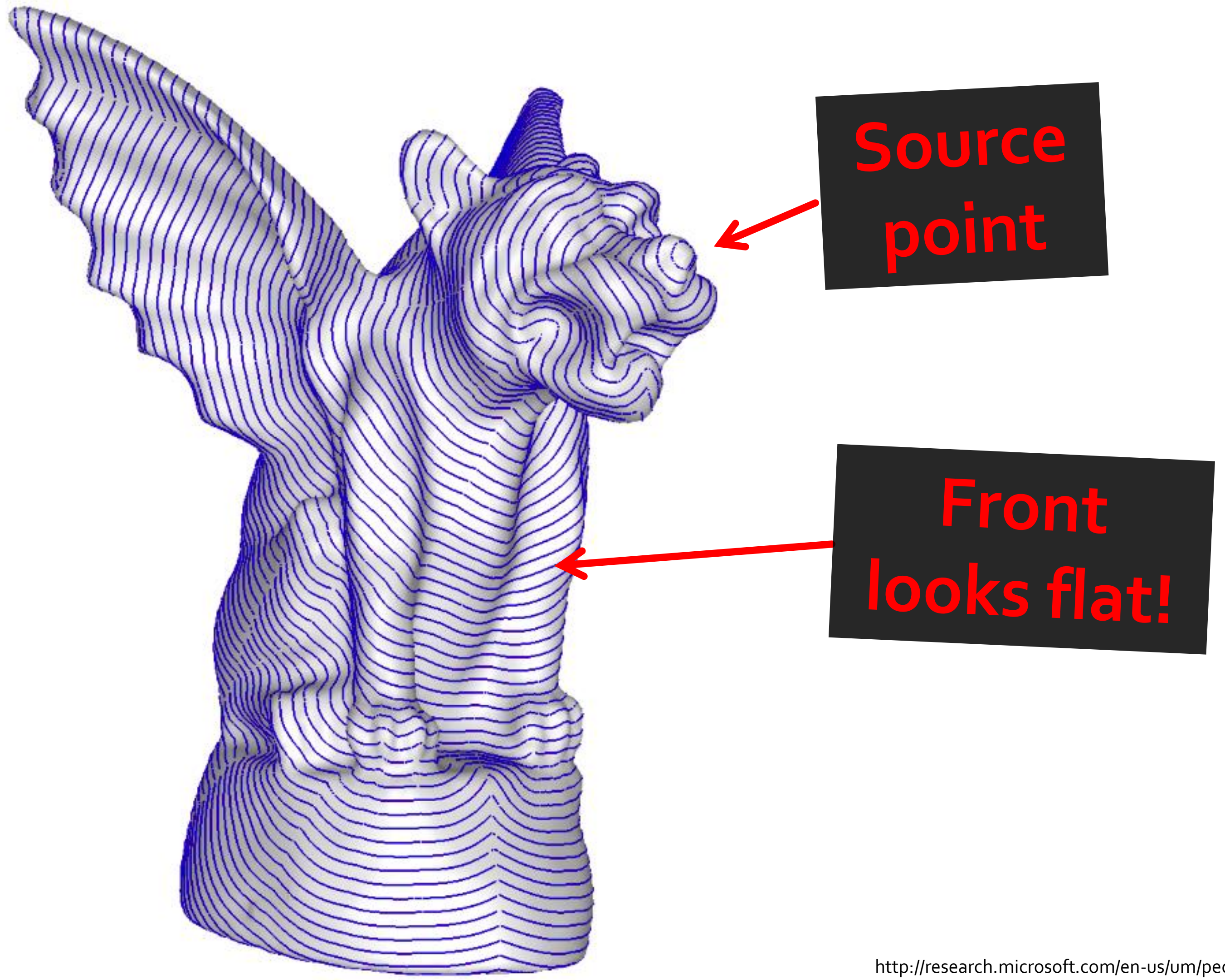
Fast Marching

Dijkstra's algorithm, modified to approximate geodesic distances.

Problem

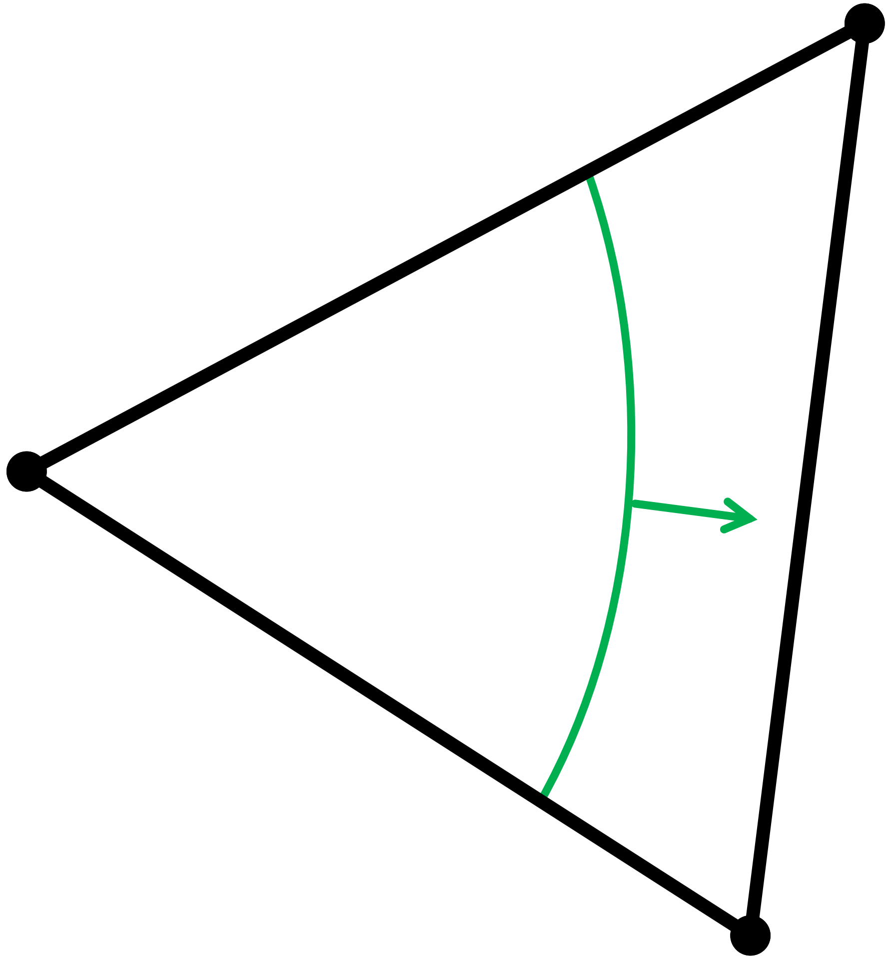


Planar Front Approximation



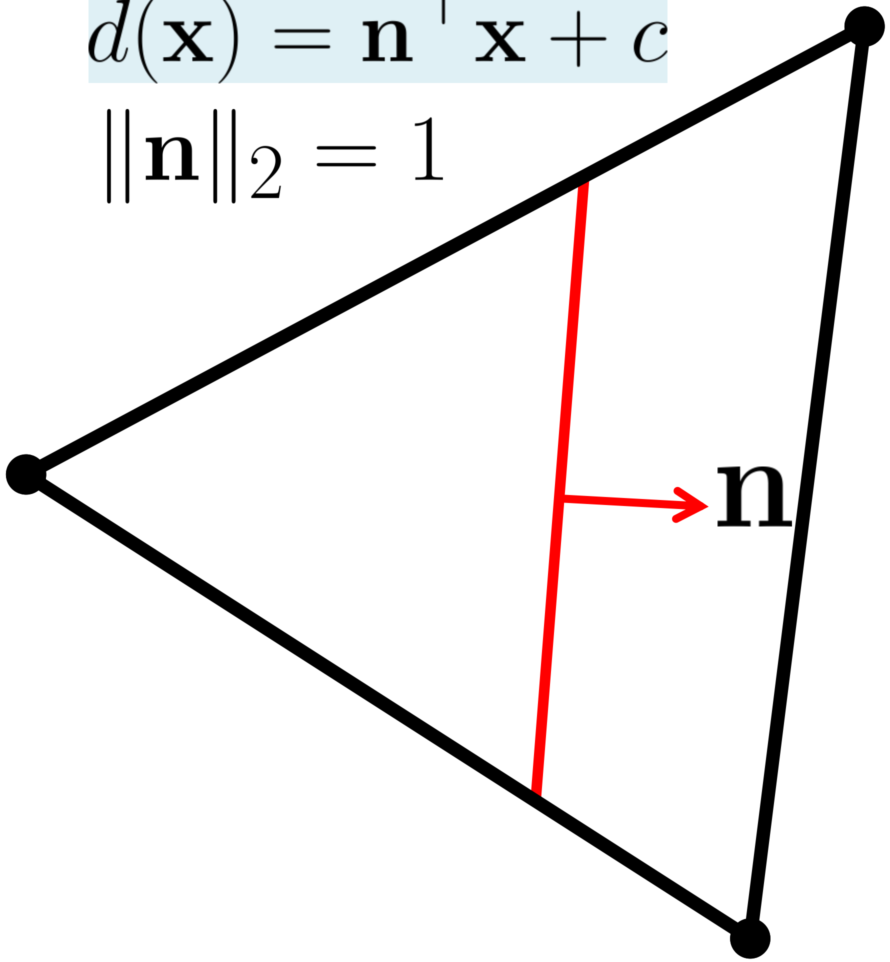
http://research.microsoft.com/en-us/um/people/hoppe/geodesics.pdf

At Local Scale

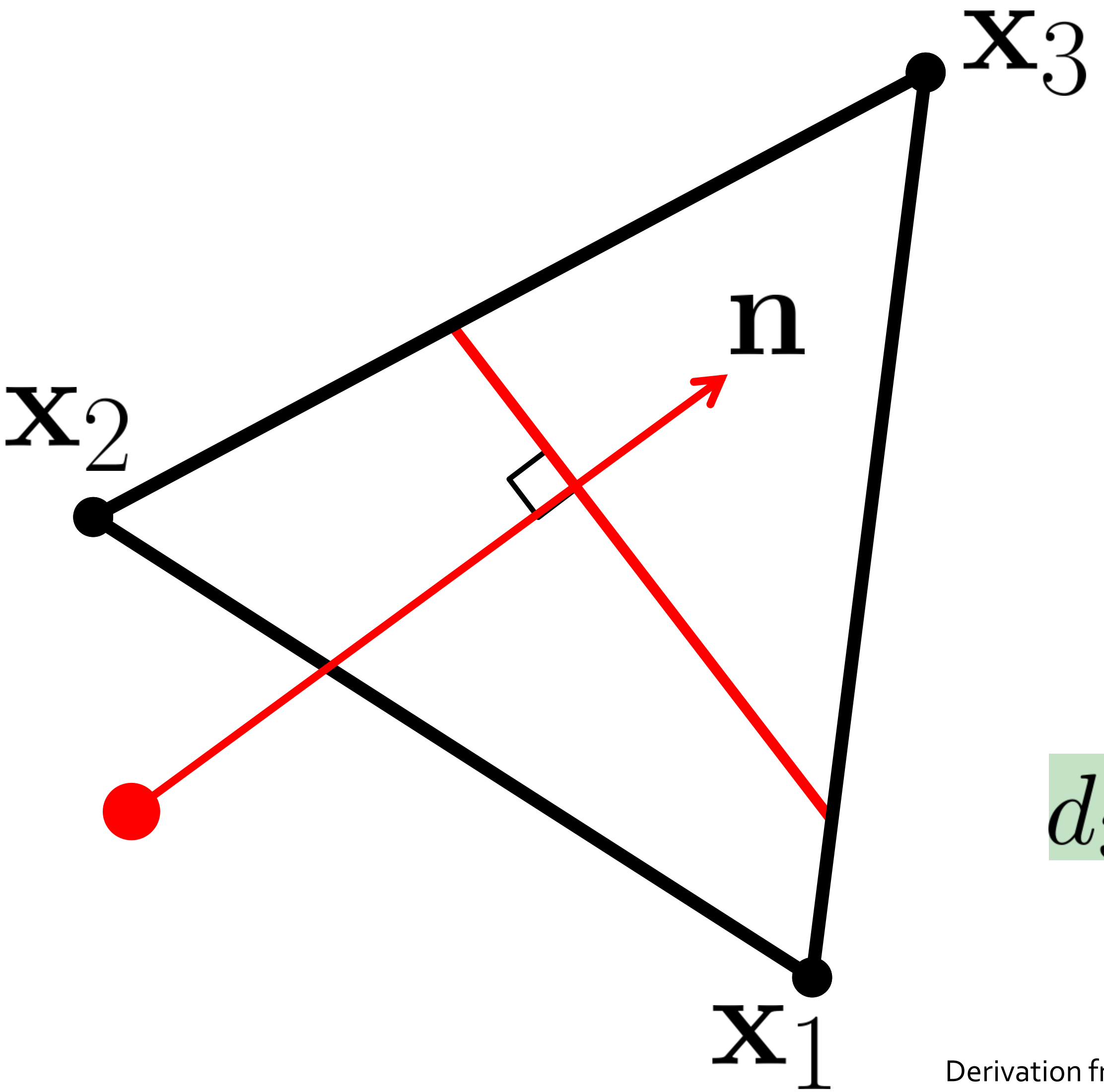


$$d(\mathbf{x}) = \mathbf{n}^\top \mathbf{x} + c$$

$$\|\mathbf{n}\|_2 = 1$$



Planar Calculations



Given:

$$d_1 = \mathbf{n}^\top \mathbf{x}_1 + c$$

$$d_2 = \mathbf{n}^\top \mathbf{x}_2 + c$$

$$\mathbf{d} = X^\top \mathbf{n} + c\mathbf{1}$$

Find:

$$d_3 = \mathbf{n}^\top \mathbf{x}_3^{\,0} + c = c$$

Derivation from Bronstein et al., *Numerical Geometry of Nonrigid Shapes*

Planar Calculations

$$\mathbf{d} = X^\top \mathbf{n} + c\mathbf{1}$$

$$\downarrow$$

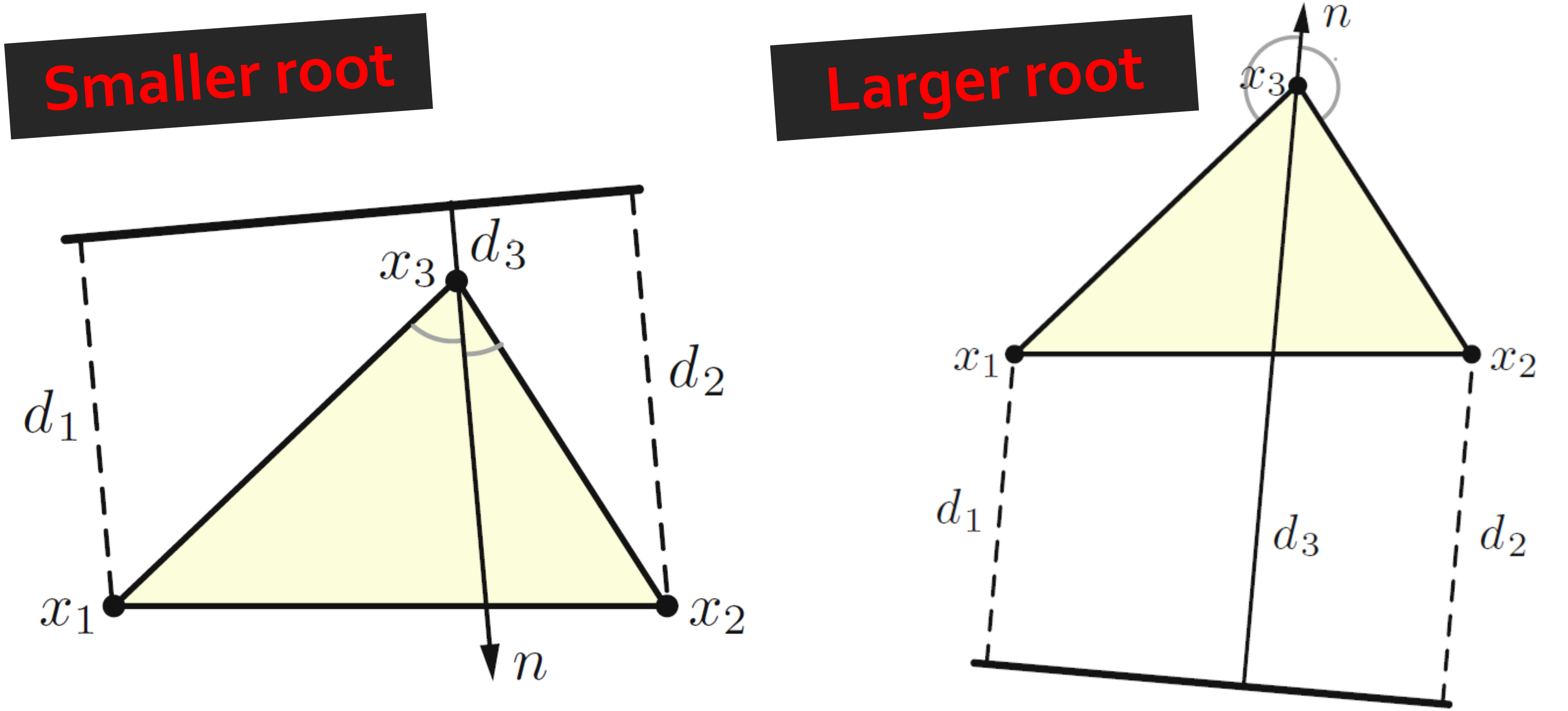
$$\mathbf{n} = X^{-\top}(\mathbf{d} - c\mathbf{1})$$

$$1 = \mathbf{n}^\top \mathbf{n}$$

$$= [\mathbf{1}^\top (X^\top X)^{-1}\mathbf{1}]c^2 + [-2\mathbf{1}^\top (X^\top X)^{-1}\mathbf{d}]c + [\mathbf{d}^\top (X^\top X)^{-1}\mathbf{d}]$$

Quadratic equation for $c = d_3$!

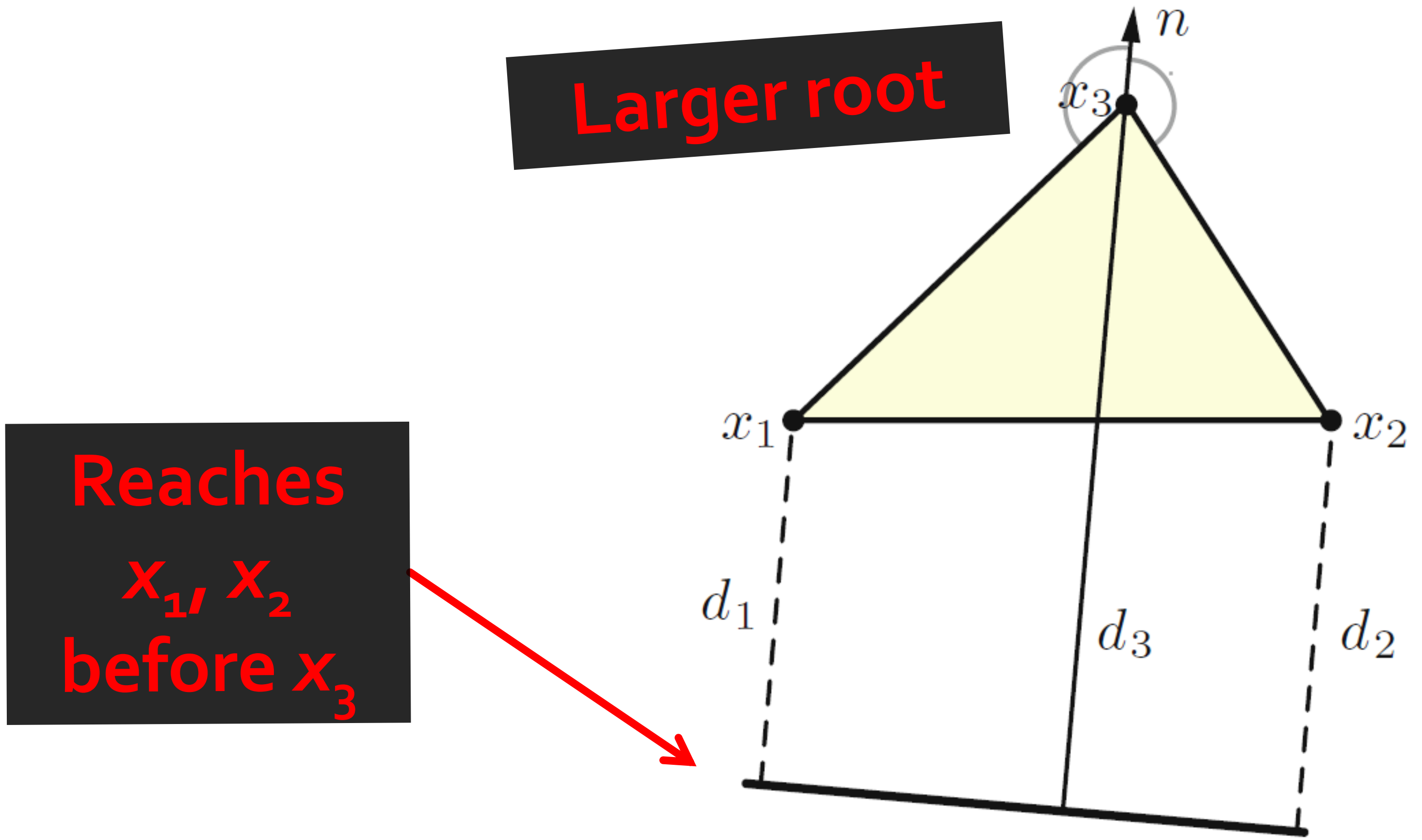
Two Roots



Bronstein et al., *Numerical Geometry of Nonrigid Shapes*

Two orientations for the normal

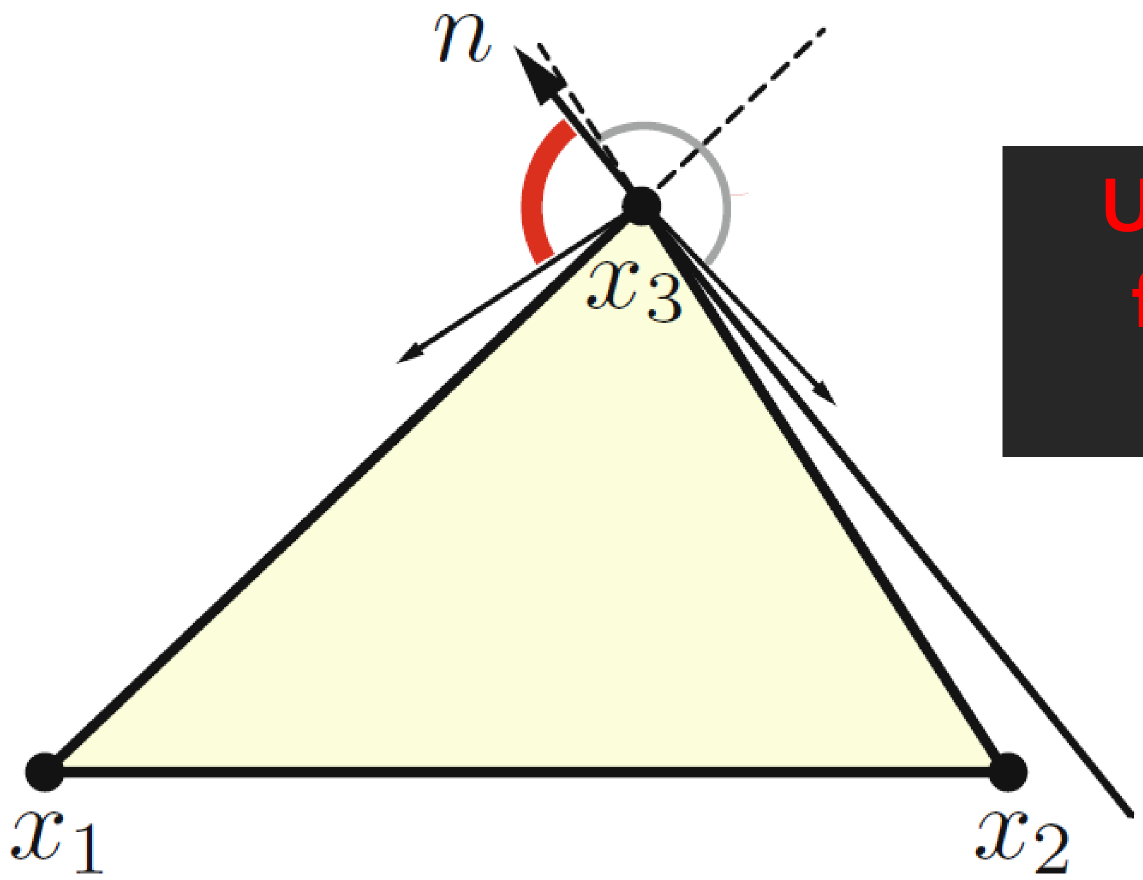
Larger Root: Consistent



Bronstein et al., *Numerical Geometry of Nonrigid Shapes*

Two orientations for the normal

Additional Issue

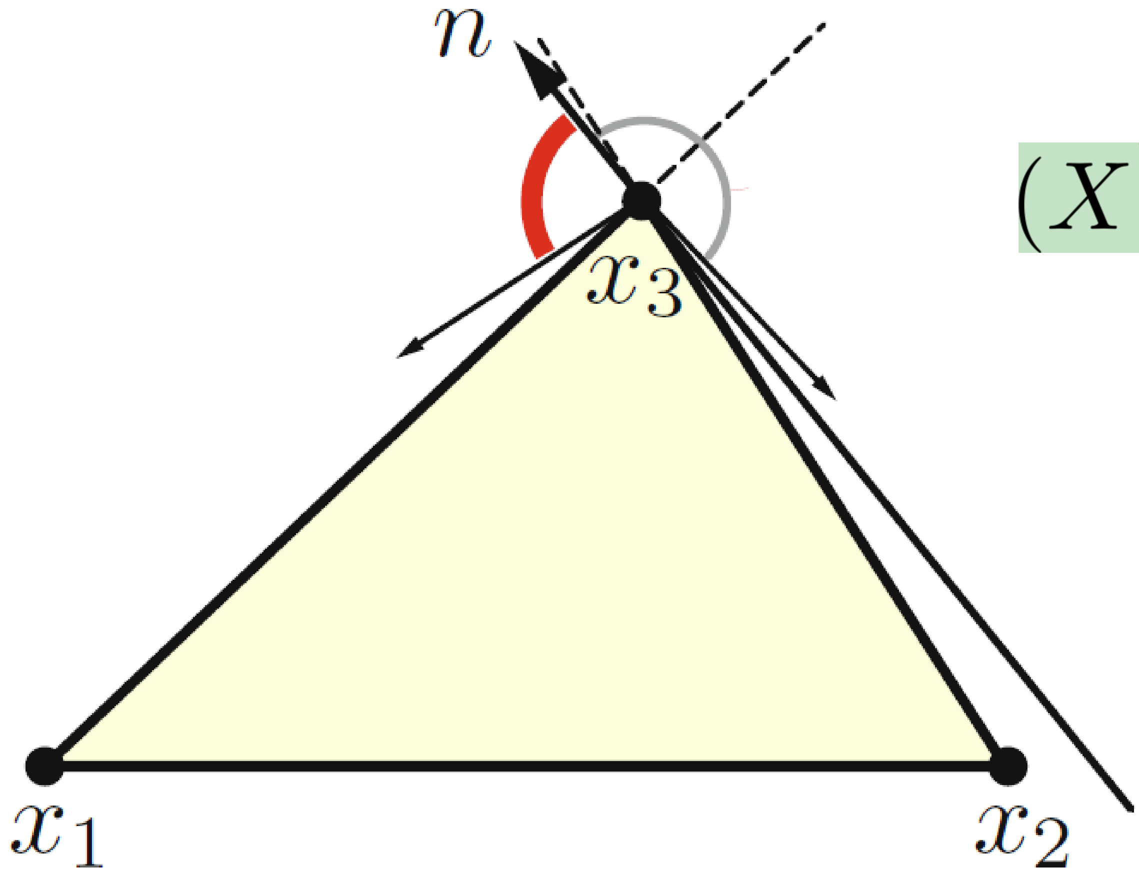


Update should be from a different triangle!

Bronstein et al., *Numerical Geometry of Nonrigid Shapes*

Front from outside the triangle

Condition for Front Direction



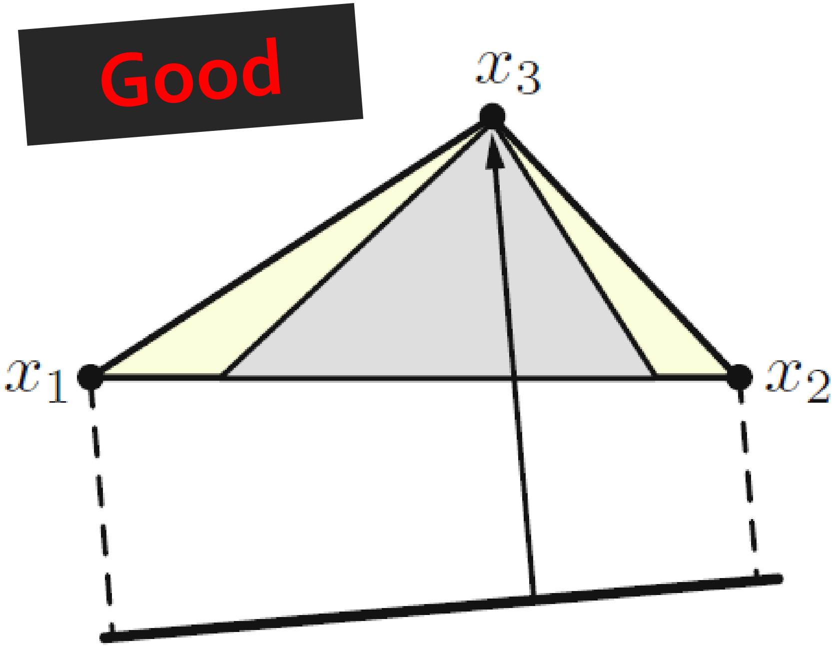
$$(X^\top X)^{-1} X^\top \mathbf{n} < 0$$

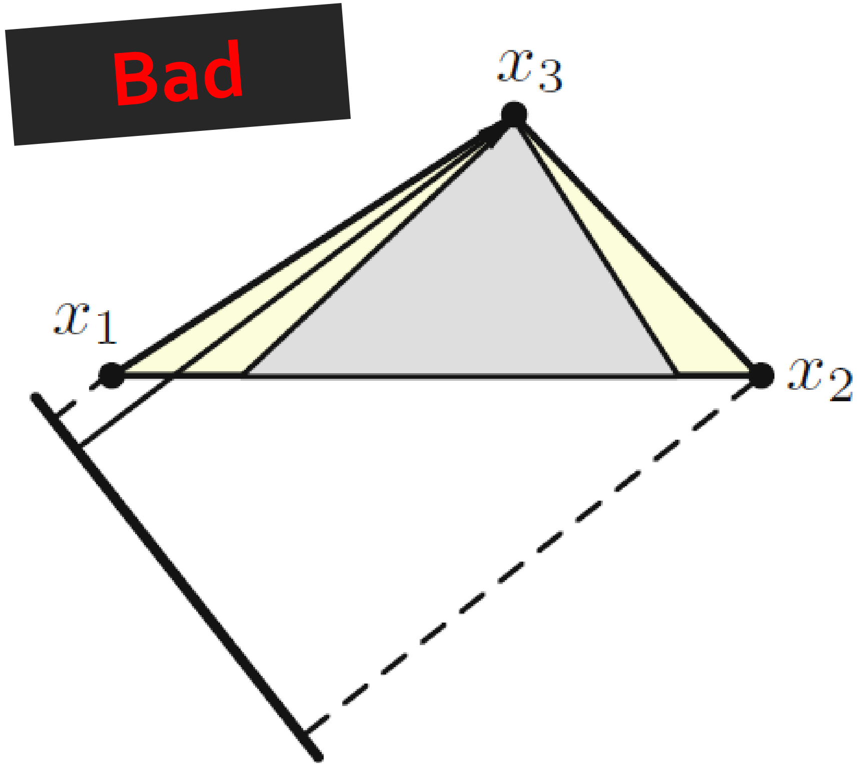
Exercise!

Bronstein et al., *Numerical Geometry of Nonrigid Shapes*

Front from outside the triangle

Obtuse Triangles





Bronstein et al., *Numerical Geometry of Nonrigid Shapes*

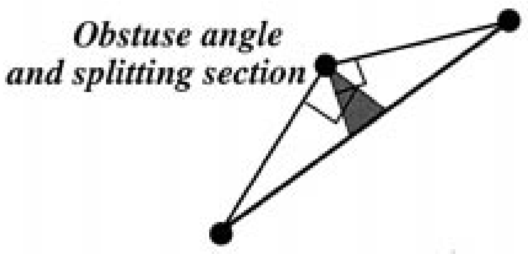
Must reach x_3 after x_1 and x_2

Fixing the Issues

- Alternative edge-based update:

$$d_3 \leftarrow \min\{d_3, d_1 + \|x_1\|, d_2 + \|x_2\|\}$$

- Add connections as needed
  [Kimmel and Sethian 1998]



Fast Marching vs. Dijkstra

- Modified update step
- Update all triangles adjacent to a given vertex

Eikonal Equation

$$\|\nabla d\|_2 = 1$$

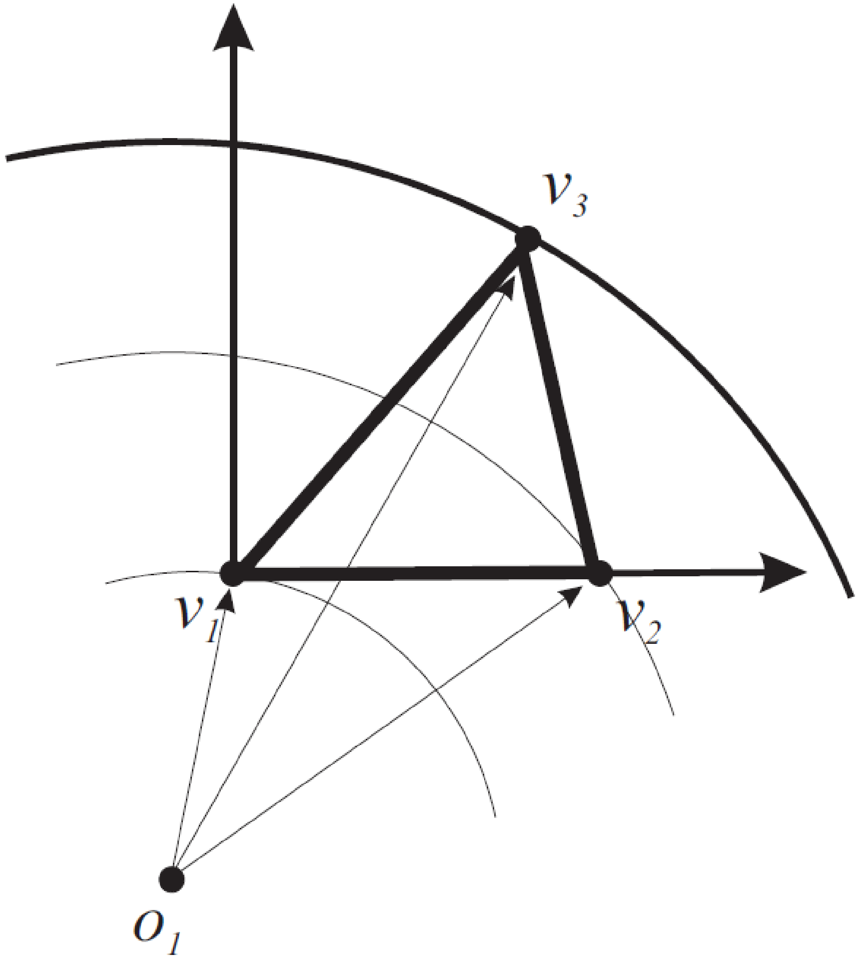
↓

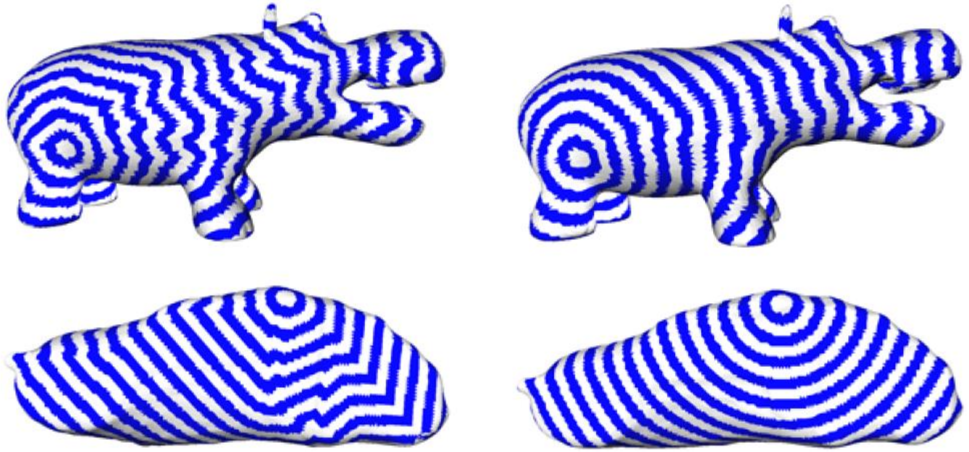
$$\|\mathbf{n}\|_2 = 1$$

Solutions are geodesic distance

A much better one!

Modifying Fast Marching

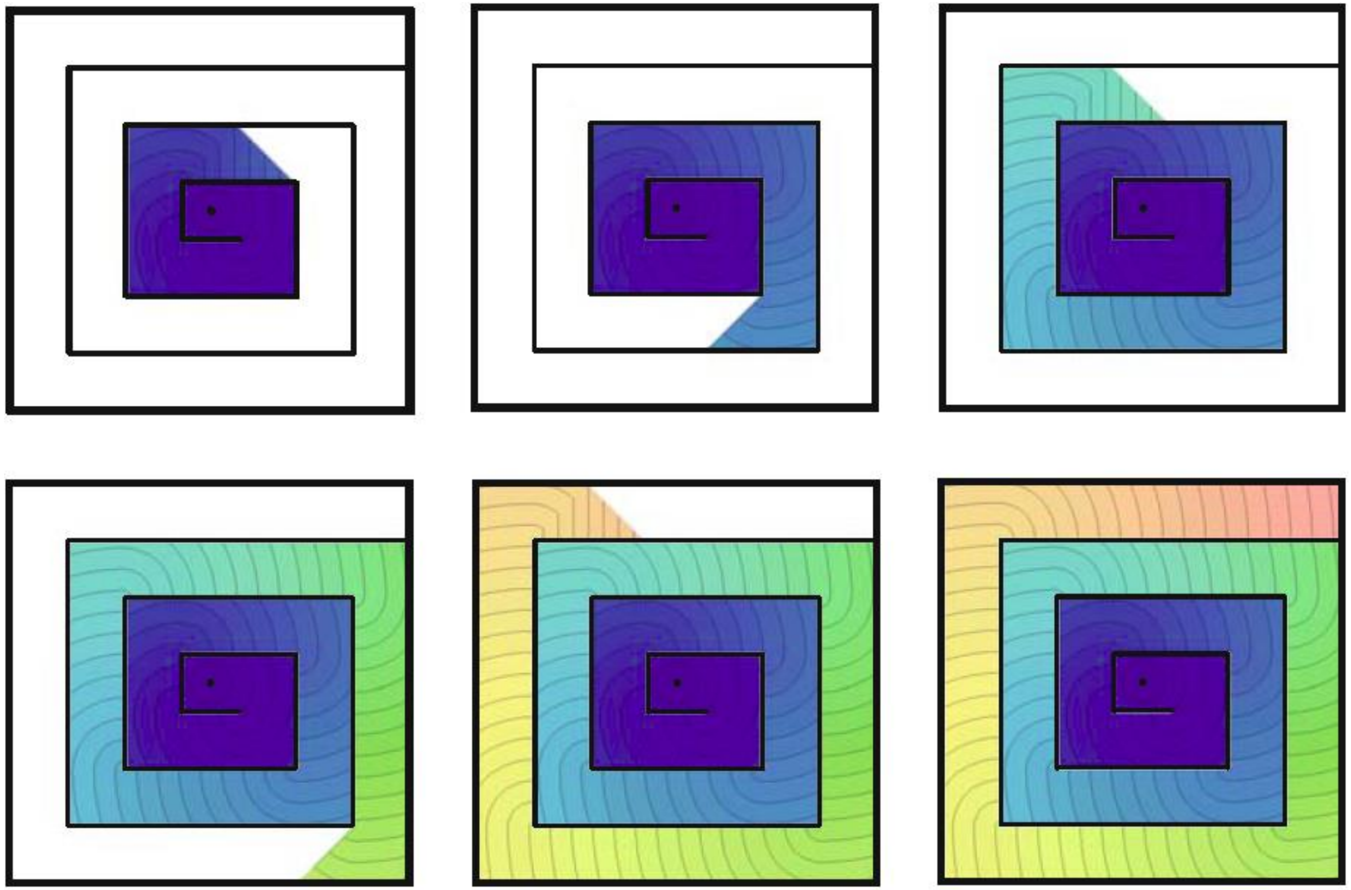




[Novotni and Klein 2002]:

Circular wavefront

Modifying Fast Marching



Raster scan and/or parallelize

Bronstein, *Numerical Geometry of Nonrigid Shapes*

Grids and parameterized surfaces

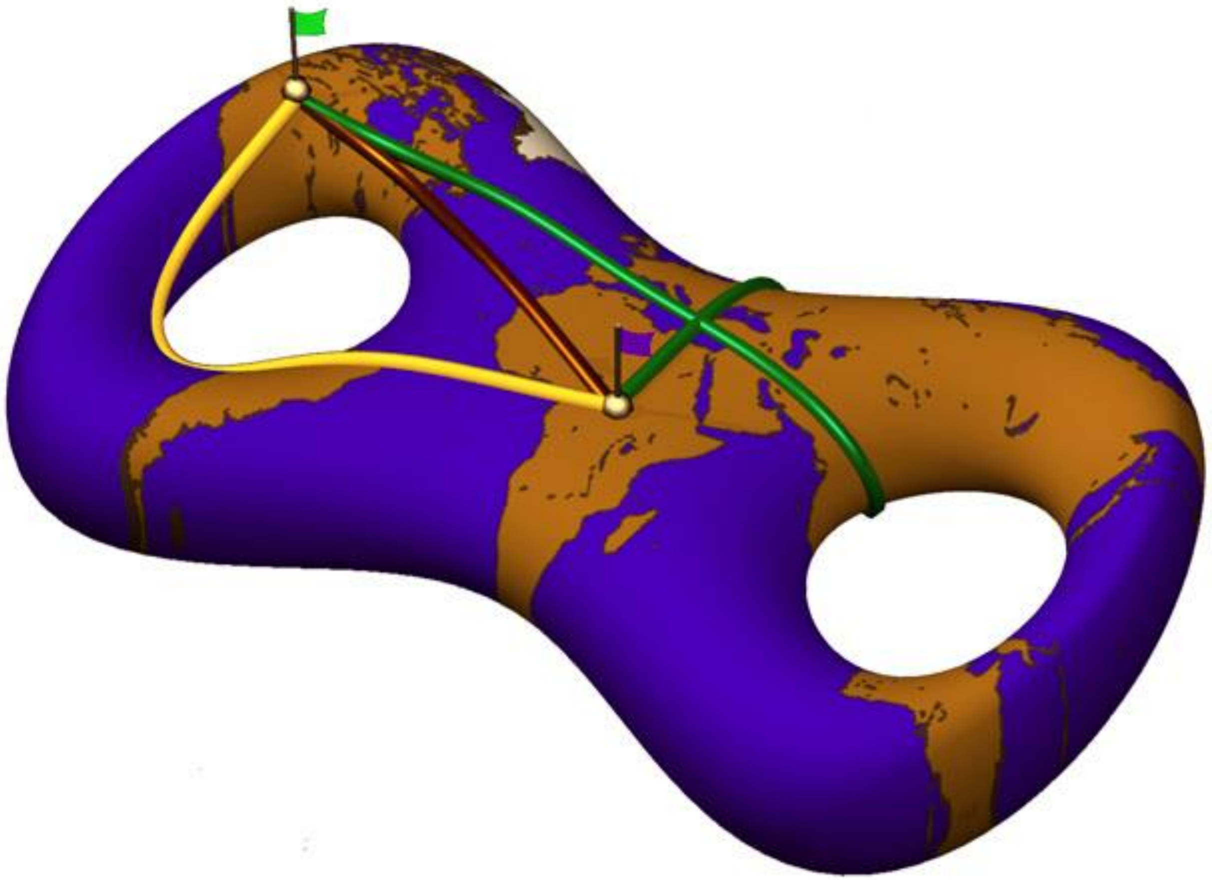
Alternative to Eikonal Equation

Algorithm 1 The Heat Method

I. Integrate the heat flow $\dot{u} = \Delta u$ for time t.
II. Evaluate the vector field $X = -\nabla u / |\nabla u|$.
III. Solve the Poisson equation $\Delta \phi = \nabla \cdot X$.

Crane, Weischedel, and Wardetzky. "Geodesics in Heat." TOG 2013.

Tracing Geodesic Curves



Trace gradient of distance function

Exact Geodesics

SIAM J. COMPUT.
Vol. 16, No. 4, August 1987

005

THE DISCRETE GEODESIC PROBLEM*

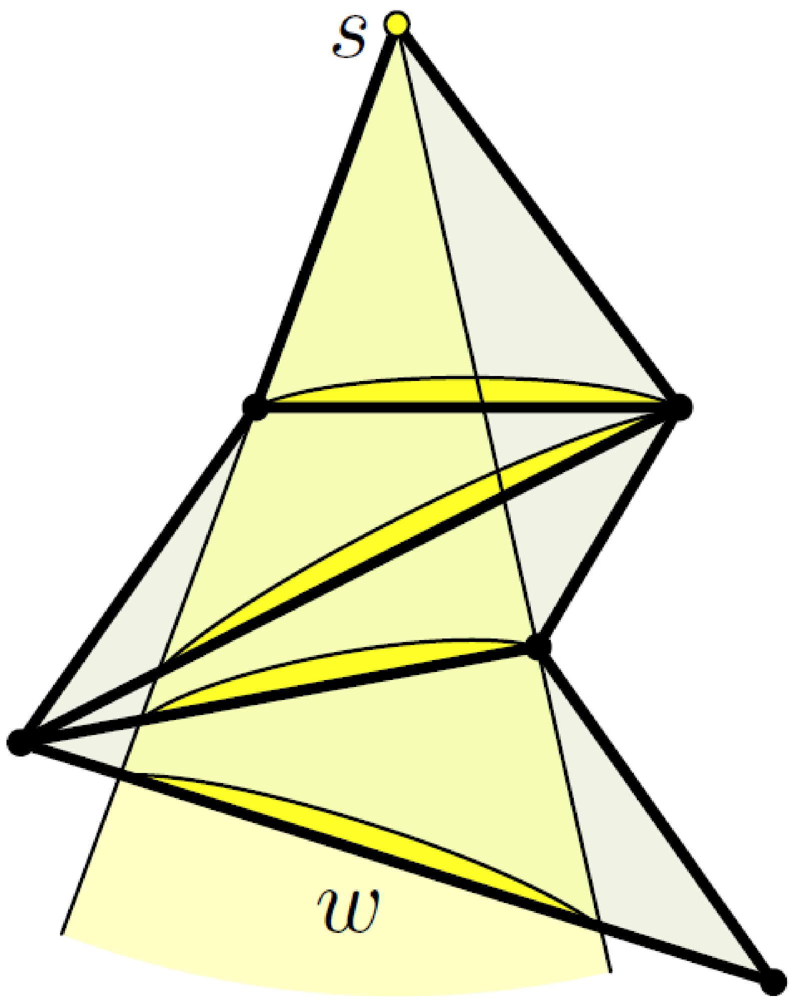
JOSEPH S. B. MITCHELL†, DAVID M. MOUNT‡ AND CHRISTOS H. PAPADIMITRIOU§

Abstract. We present an algorithm for determining the shortest path between a source and a destination on an arbitrary (possibly nonconvex) polyhedral surface. The path is constrained to lie on the surface, and distances are measured according to the Euclidean metric. Our algorithm runs in time $O(n^2 \log n)$ and requires $O(n^2)$ space, where n is the number of edges of the surface. After we run our algorithm, the distance from the source to any other destination may be determined using standard techniques in time $O(\log n)$ by locating the destination in the subdivision created by the algorithm. The actual shortest path from the source to a destination can be reported in time $O(k+\log n)$, where k is the number of faces crossed by the path. The algorithm generalizes to the case of multiple source points to build the Voronoi diagram on the surface, where n is now the maximum of the number of vertices and the number of sources.

Key words. shortest paths, computational geometry, geodesics, Dijkstra's algorithm

AMS(MOS) subject classification. 68E99

MMP Algorithm: Big Idea



Dijkstra-style front with *windows* explaining source.

Image from: Surazhsky et al. "Fast Exact and Approximate Geodesics on Meshes." SIGGRAPH 2005.

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 [illegible]
computer graphics applications. [illegible]
mesh often involves cutting the [illegible]
(e.g. [Krishnamurthy and Levoy 1996; Sander et al. 2003]), and

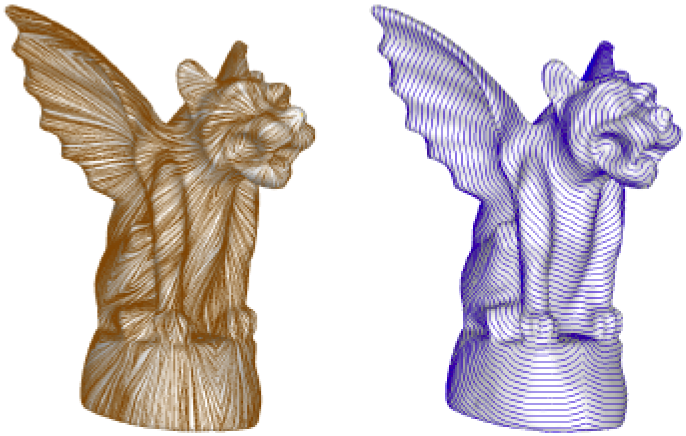


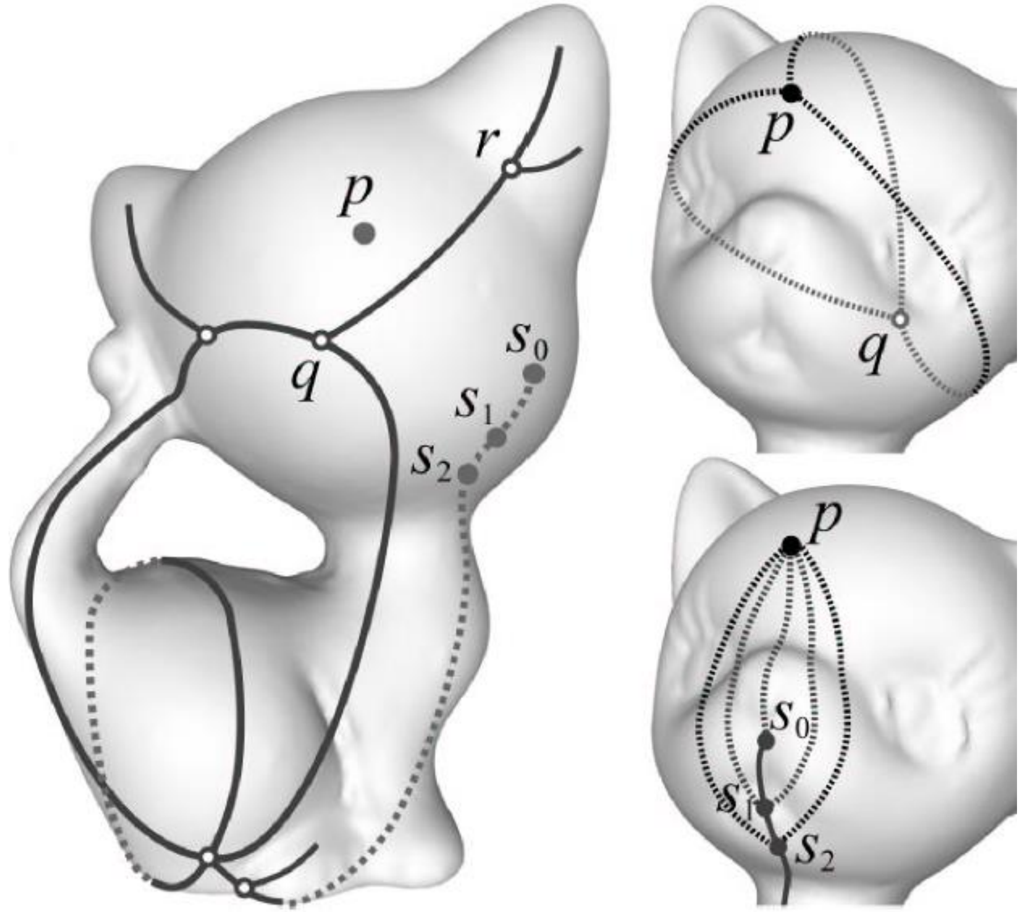
Figure 1: Geodesic paths from a source vertex, and isolines of the geodesic distance function.

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.

[illegible]
proximations with *bounded* error. In practice, the algorithm runs in $O(n \log n)$ time even for small error thresholds.

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

Recall: Cut Locus



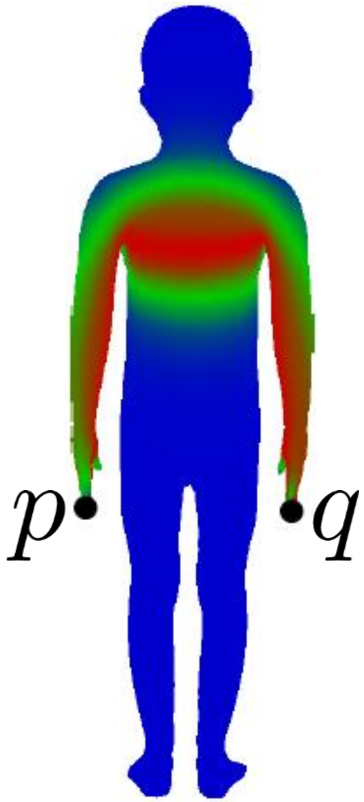
http://www.cse.ohio-state.edu/~tamaldey/paper/geodesic/cutloc.pdf

Cut point:
Point where geodesic ceases to be minimizing

Set of cut points from a source *p*

Fuzzy Geodesics

$$G^{\sigma}_{p,q}(x) := \exp(-|d(p,x) + d(x,q) - d(p,q)|/\sigma)$$



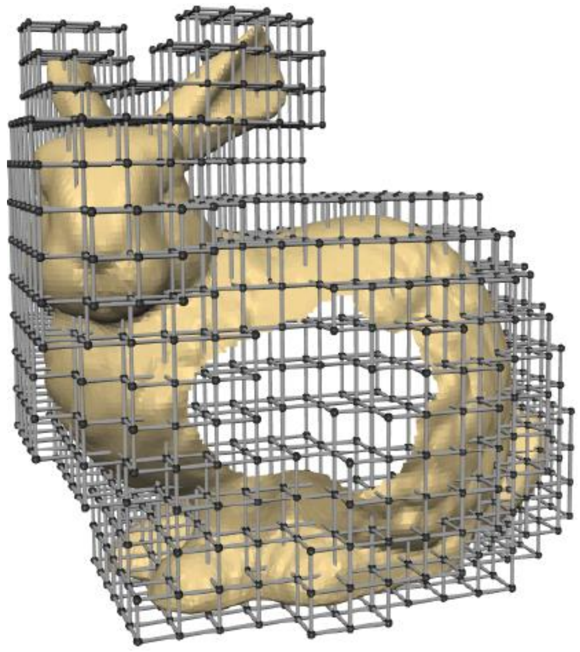
Function on surface expressing difference in triangle inequality

"Intersection" by pointwise multiplication

Sun, Chen, Funkhouser. "Fuzzy geodesics and consistent sparse correspondences for deformable shapes." CGF2010.

Stable version of geodesic distance

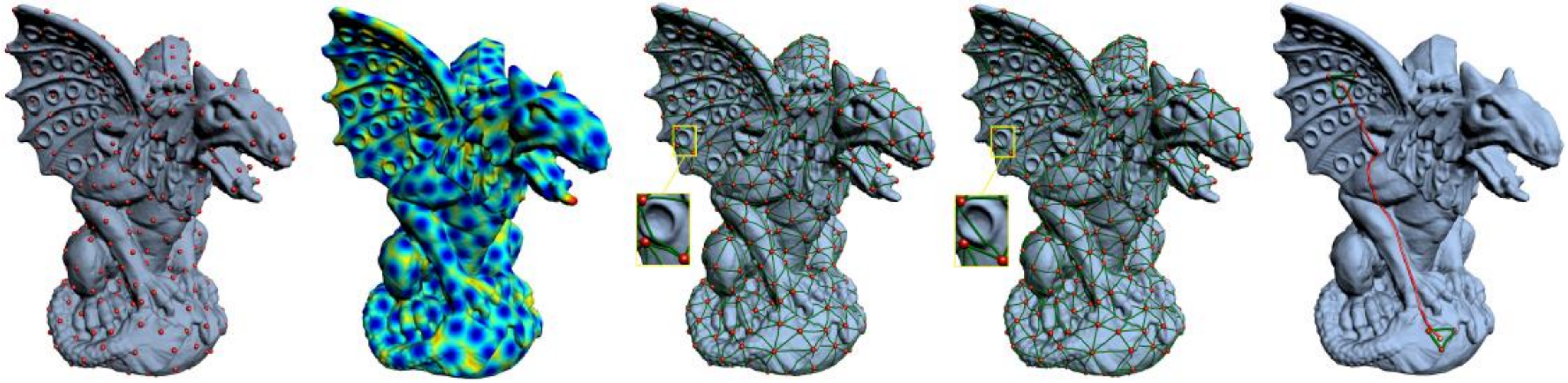
Stable Measurement



Morphological operators to fill holes rather than remeshing

Campen and Kobbelt. "Walking On Broken Mesh: Defect-Tolerant Geodesic Distances and Parameterizations." Eurographics 2011.

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

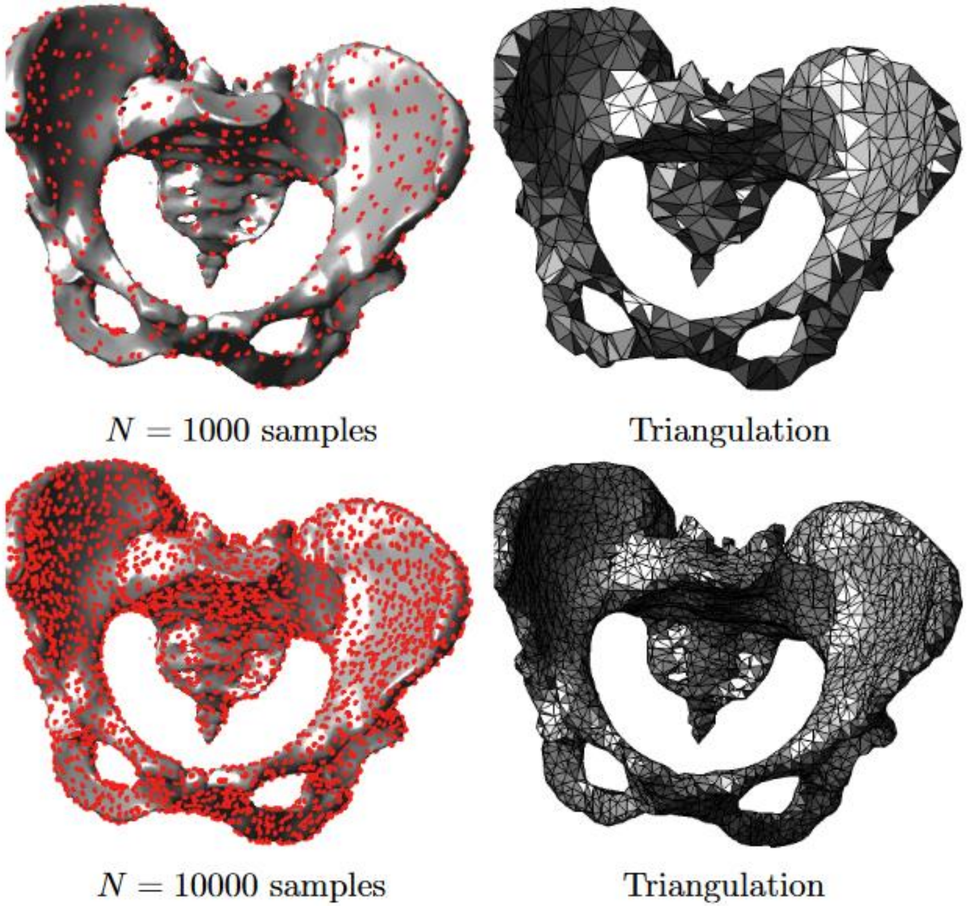


Fig. 4.12 *Geodesic remeshing with an increasing number of points.*

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

High-Dimensional Problems

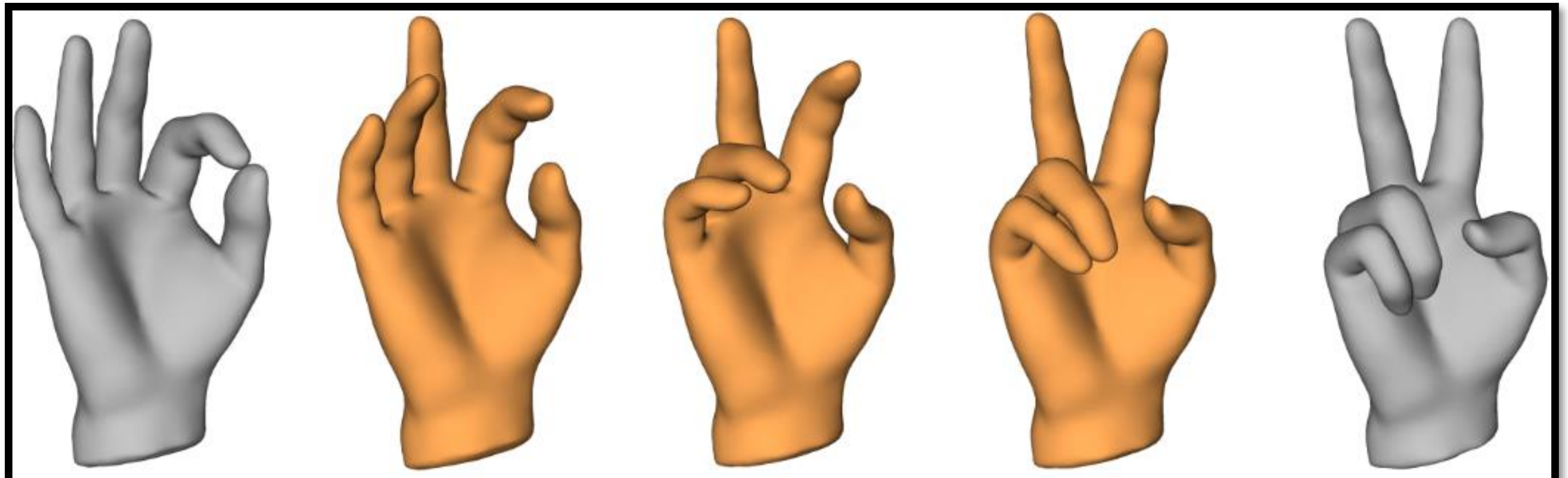


Figure 1: *Discrete geodesic computed from two input poses (leftmost and rightmost hand).*

Heeren et al. *Time-discrete geodesics in the space of shells*. SGP 2012.

In ML: Be Careful!

Shortest path distance in random k-nearest neighbor graphs

Morteza Alamgir[1] MORTEZA@TUEBINGEN.MPG.DE
Ulrike von Luxburg[1,2] ULRIKE.LUXBURG@TUEBINGEN.MPG.DE
[1] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2] Department of Computer Science, University of Hamburg, Germany

Abstract

Consider a weighted or unweighted k-nearest neighbor graph that has been built on n data points drawn randomly according to some density p on $\mathbb{R}^d$. We study the convergence of the shortest path distance in such graphs as the sample size tends to infinity. We prove that for unweighted kNN graphs, this distance converges to an unpleasant distance function on the underlying space whose properties are detrimental to machine learning. We also study the behavior of the shortest path distance in weighted kNN graphs.

The first question has already been studied in some special cases. Tenenbaum et al. (2000) discuss the case of ε- and kNN graphs when p is *uniform* and D is the geodesic distance. Sajama & Orlitsky (2005) extend these results to
introducing edg
timate of the u
Hwang & Hero
graphs whose v
and whose edge

There is little
Tenenbaum et a
special case with $h(x) = x$ and uniform p. Hwang &

We prove that for unweighted kNN graphs, this distance converges to an unpleasant distance function on the underlying space whose properties are detrimental to machine learning.

Intriguing Theoretical Progress

APPROXIMATING GEODESICS VIA RANDOM POINTS

ERIK DAVIS AND SUNDER SETHURAMAN

ABSTRACT. Given a 'cost' functional F on paths γ in a domain $D \subset \mathbb{R}^d$, in the form $F(\gamma) = \int_0^1 f(\gamma(t), \dot{\gamma}(t))dt$, it is of interest to approximate its minimum cost and geodesic paths. Let $X_1, \ldots, X_n$ be points drawn independently from D according to a distribution with a density. Form a random geometric graph on the points where X_i and X_j are connected when $0 < |X_i - X_j| < \epsilon$, and the length scale $\epsilon = \epsilon_n$ vanishes at a suitable rate.

For a general class of functionals F, associated to Finsler and other distances on D, using a probabilistic form of Gamma convergence, we show that the minimum costs and geodesic paths, with respect to types of approximating discrete 'cost' functionals, built from the random geometric graph, converge almost surely in various senses to those corresponding to the continuum cost F, as the number of sample points diverges. In particular, the geodesic path convergence shown appears to be among the first results of its kind.

1. INTRODUCTION

Understanding the 'shortest' or geodesic paths between points in a medium is an intrinsic concern in diverse applied problems, from 'optimal routing' in networks and disordered materials to 'identifying manifold structure in large data sets', as well as in studies of probabilistic $\mathbb{Z}^d$-percolation models, since the seminal paper of [5] (cf. recent survey [4]). See also [17], [18], [19], [20], [21], [22] which consider percolation in $\mathbb{R}^d$ continuum settings.

There are sometimes abstract formulas for the geodesics, from the calculus of variations, or other differential equation approaches. For instance, with respect to a patch of a Riemannian manifold (M, g), with $M \subset \mathbb{R}^d$ and tensor field $g(\cdot)$, it is known that the distance function $U(\cdot) = d(x, \cdot)$, for fixed x, is a viscosity solution of the Eikonal equation $\|\nabla U(y)\|_{g(y)^{-1}} = 1$ for $y \neq x$, with boundary condition $U(x) = 0$. Here, $\|v\|_A = \sqrt{\langle v, Av\rangle}$, where $\langle \cdot, \cdot \rangle$ is the standard innerproduct on $\mathbb{R}^d$. Then, a geodesic γ connecting x and z may be recovered from U by solving a 'descent' equation, $\dot{\gamma}(t) = -\eta(t)g^{-1}(\gamma(t))\nabla U(\gamma(t))$, where $\eta(t)$ is a scalar function

Roughly:
Statistical convergence of approximate geodesics on geometric graphs.

Γ convergence?!

In ML: Be Careful!

Geodesic Exponential Kernels: When Curvature and Linearity Conflict

Aasa Feragen
DIKU, University of Copenhagen
Denmark
aasa@diku.dk

François Lauze
DIKU, University of Copenhagen
Denmark
francois@diku.dk

Søren Hauberg
DTU Compute
Denmark
sohau@dtu.dk

Abstract

We consider kernel methods on general geodesic metric spaces and provide both negative and positive results. First we show that the common Gaussian kernel can only be generalized to a positive definite kernel on a geodesic metric space if the space is flat. As a result, for data on a Riemannian manifold, the geodesic Gaussian kernel is only positive definite if the Riemannian manifold is Euclidean. This implies that any attempt to design geodesic Gaussian kernels on curved Riemannian manifolds is futile. However, we show that for spaces with conditionally negative definite distances the geodesic Laplacian kernel can be generalized while retaining positive definiteness. This implies that geodesic Laplacian kernels can be generalized to some curved spaces, including spheres and hyperbolic spaces. Our theoretical results are verified empirically.

Preview:

Heat kernel is PD!

Kernel	Extends to general Metric spaces	Extends to general Riemannian manifolds
Gaussian ($q = 2$)	No (only if flat)	No (only if Euclidean)
Laplacian ($q = 1$)	Yes, iff metric is CND	Yes, iff metric is CND
Geodesic exp. ($q > 2$)	Not known	No

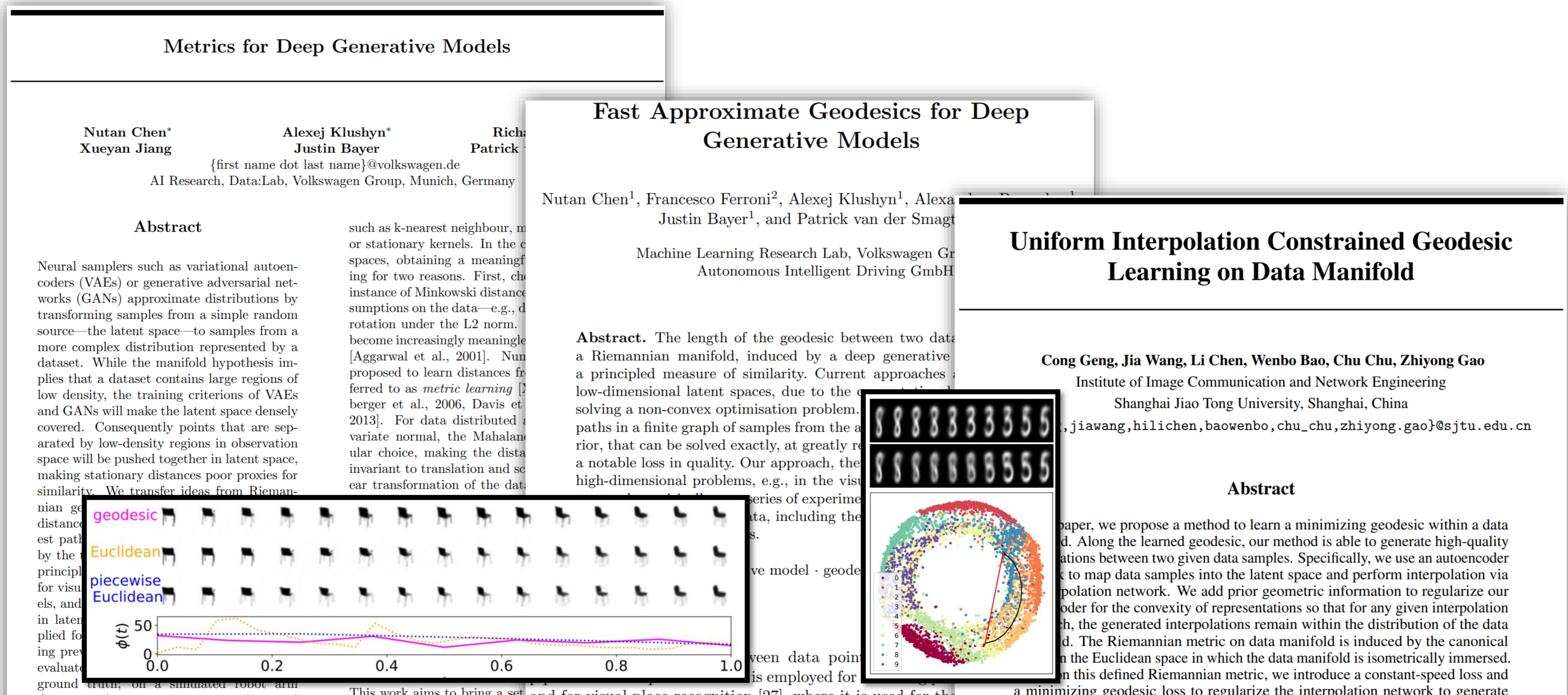
Table 1. Overview of results: For a geodesic metric, when is the geodesic exponential kernel (1) positive definite for all $\lambda > 0$?

While this idea has an appealing similarity to familiar Eu-
clid…
the…

Theorem 2. *Let M be a complete, smooth Riemannian manifold with its associated geodesic distance metric d. Assume, moreover, that $k(x, y) = \exp(-\lambda d^2(x, y))$ is a PD geodesic Gaussian kernel for all $\lambda > 0$. Then the Riemannian manifold M is isometric to a Euclidean space.*

- The *geodesic Gaussian kernel* is positive definite (PD) for all $\lambda > 0$ only if the underlying metric space is

Renewed Interest in Practical Aspects



Geodesic Distances: Algorithms

Justin Solomon

6.8410: Shape Analysis

Spring 2023