

IFT 6113

GEODESICS

tiny.cc/ift6113



Mikhail Bessmeltsev

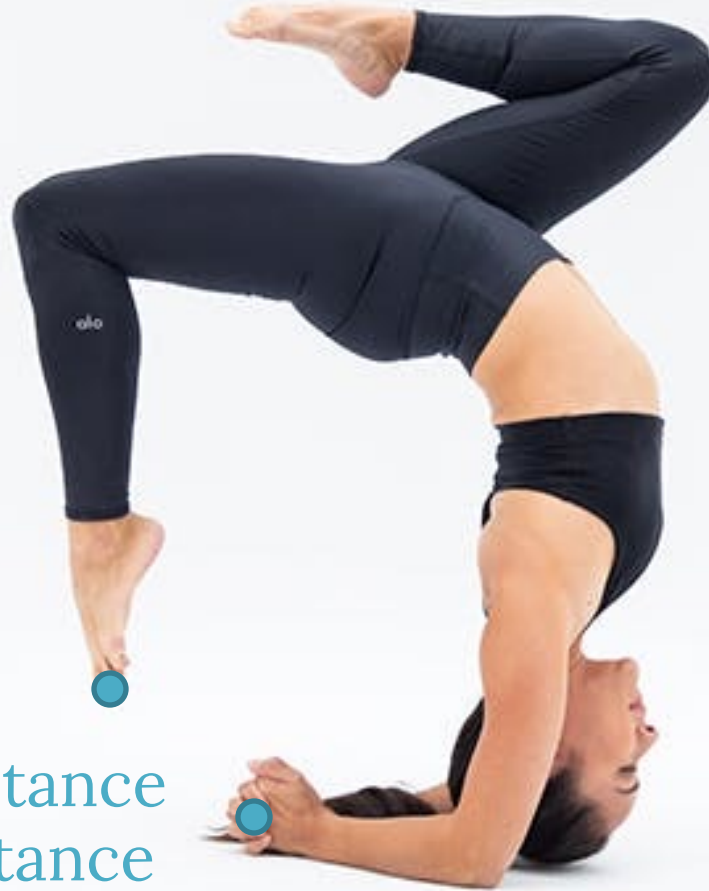
Geodesic Distance



Extrinsically close
Intrinsicly far

Geodesic Distance

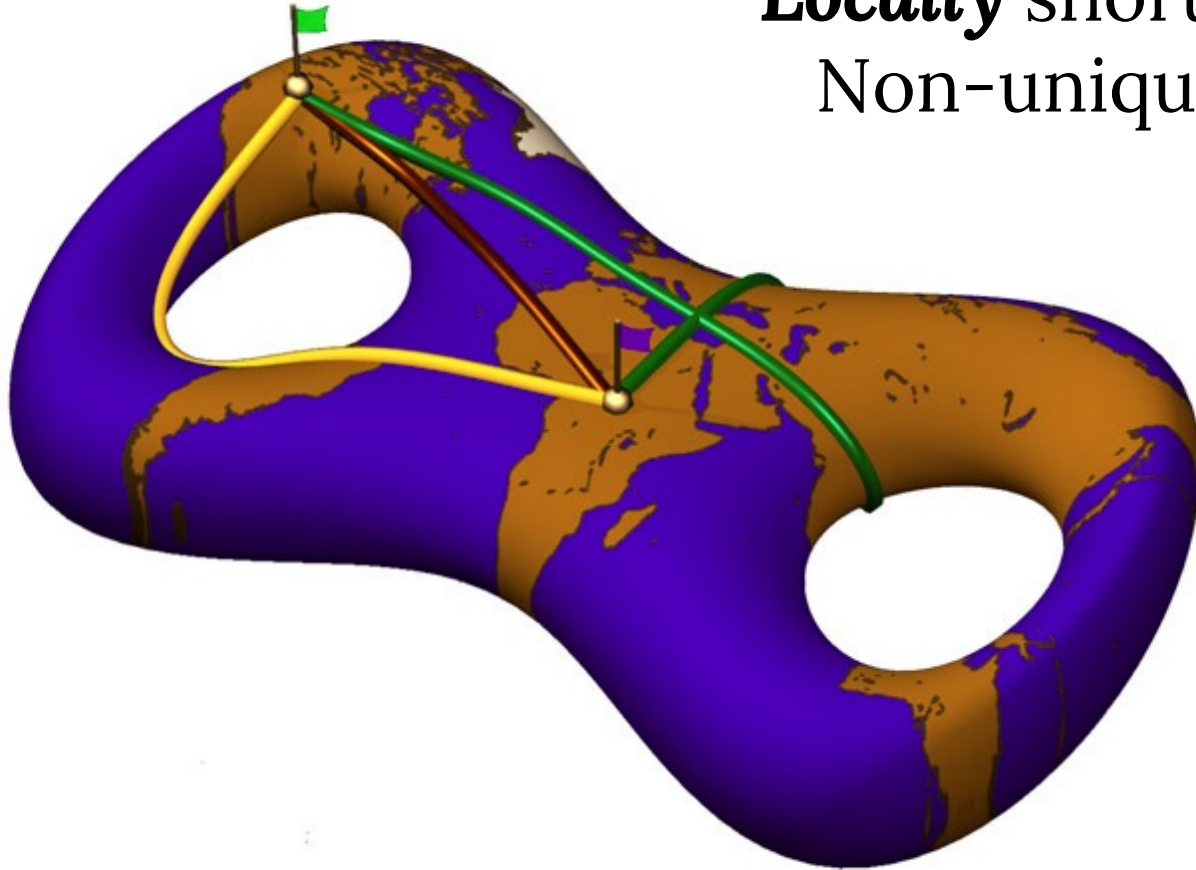
Length of a shortest path on a surface



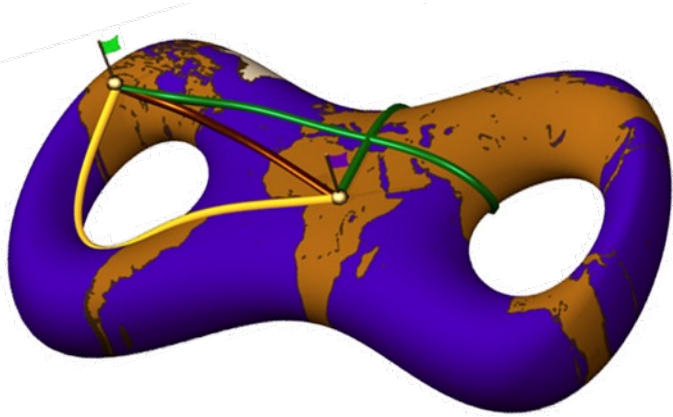
Small Euclidean distance
Large geodesic distance

Geodesics

Locally shortest
Non-unique!



Possible questions



Locally shortest



Single source

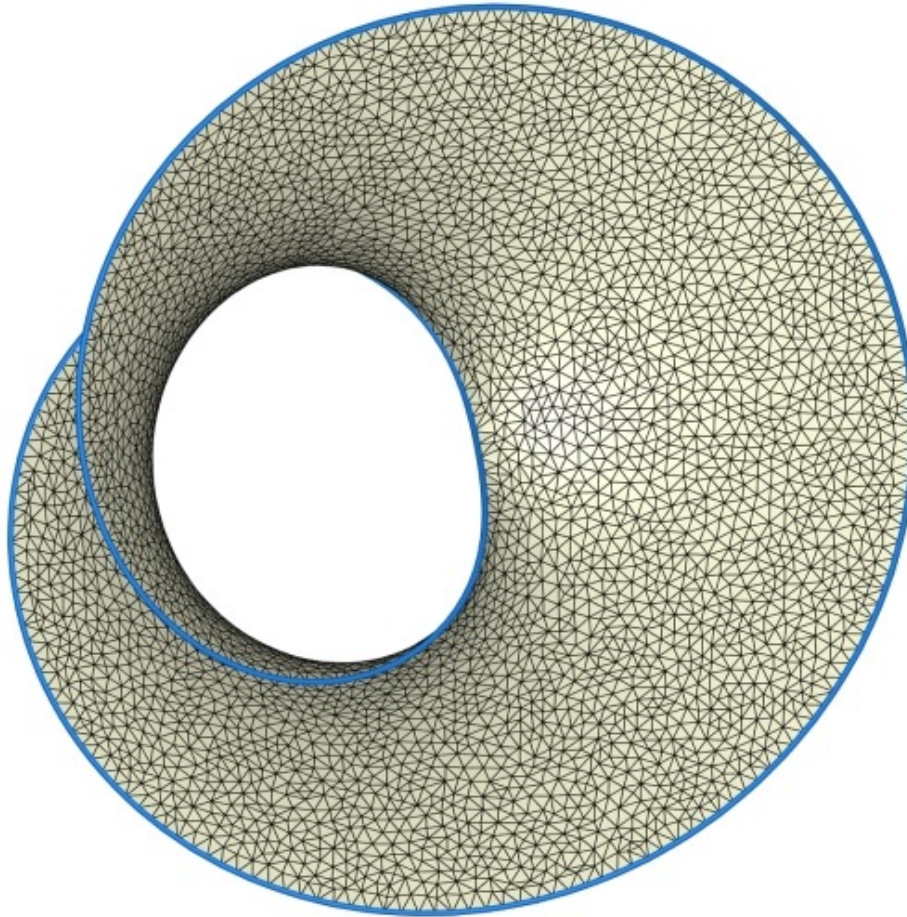


Multi-source

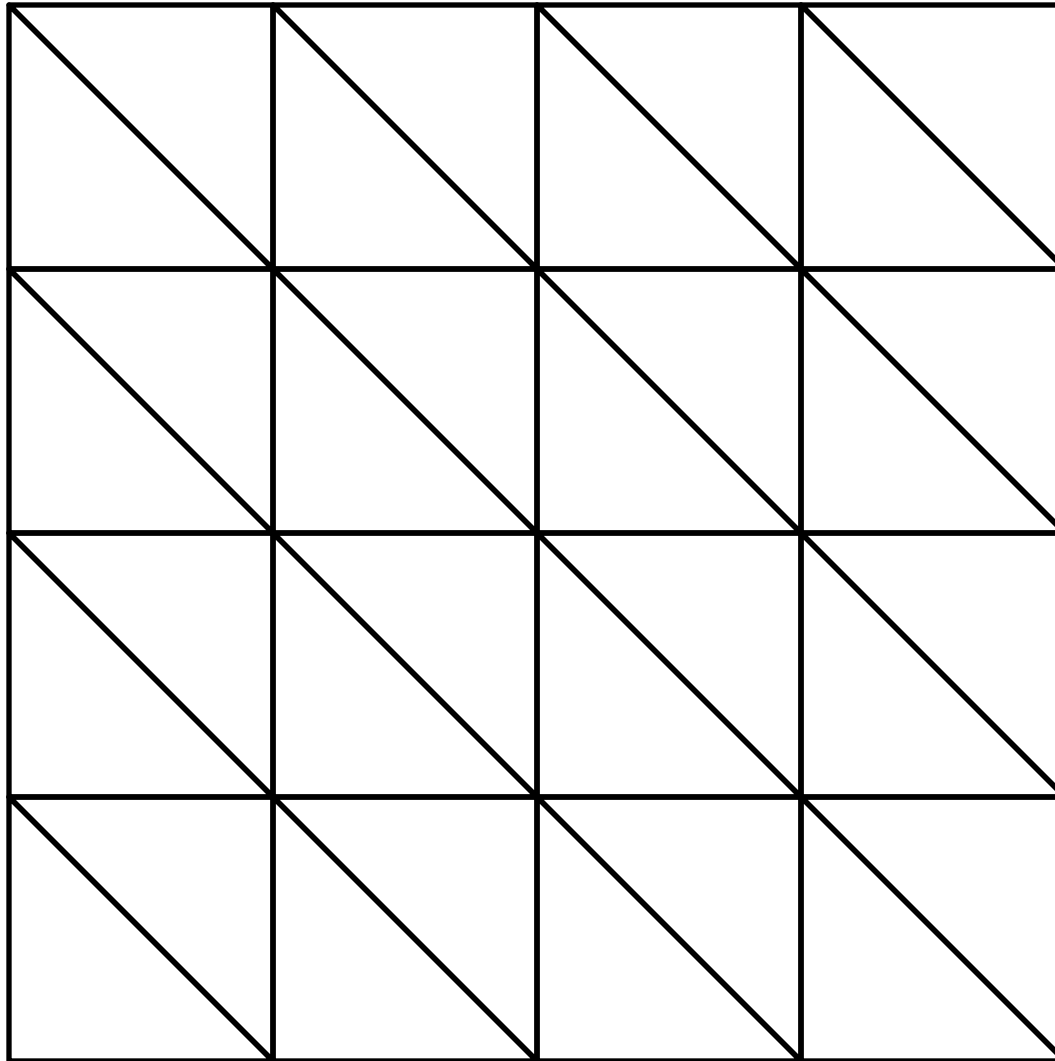


All-pairs

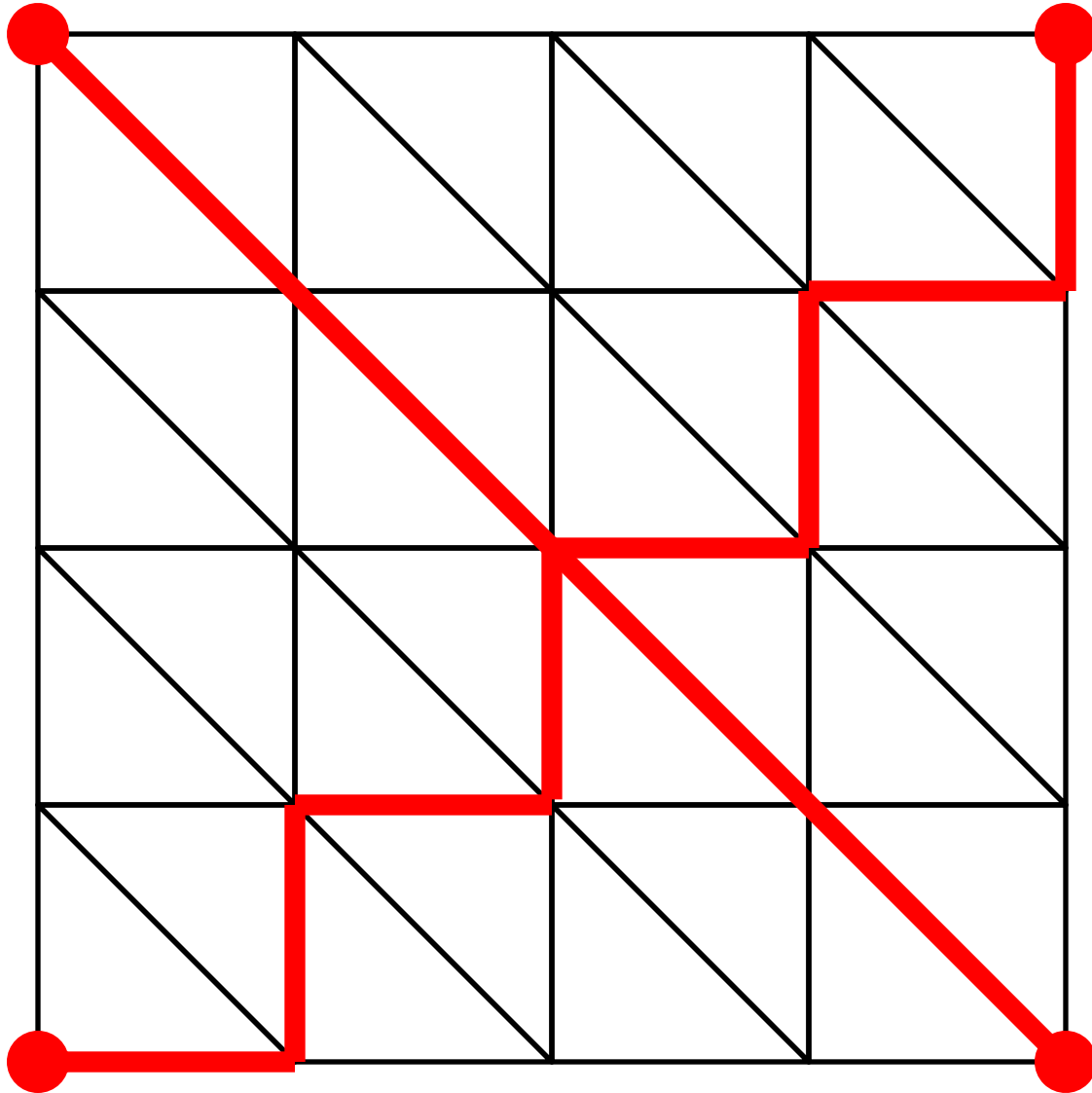
Mesh ~ Graph
find shortest path?



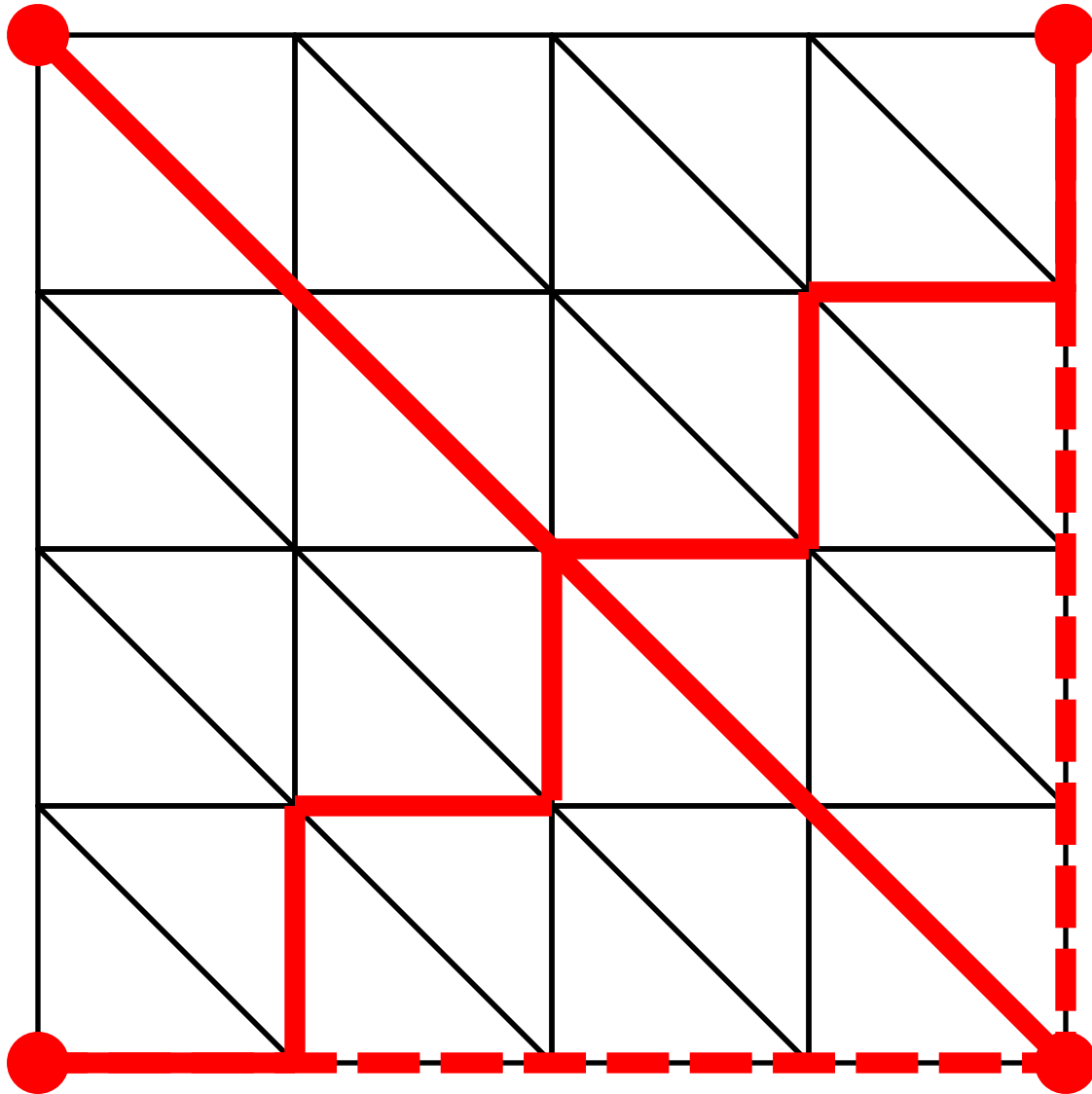
Mesh ~ Graph
find shortest path?



Mesh ~ Graph
find shortest path?



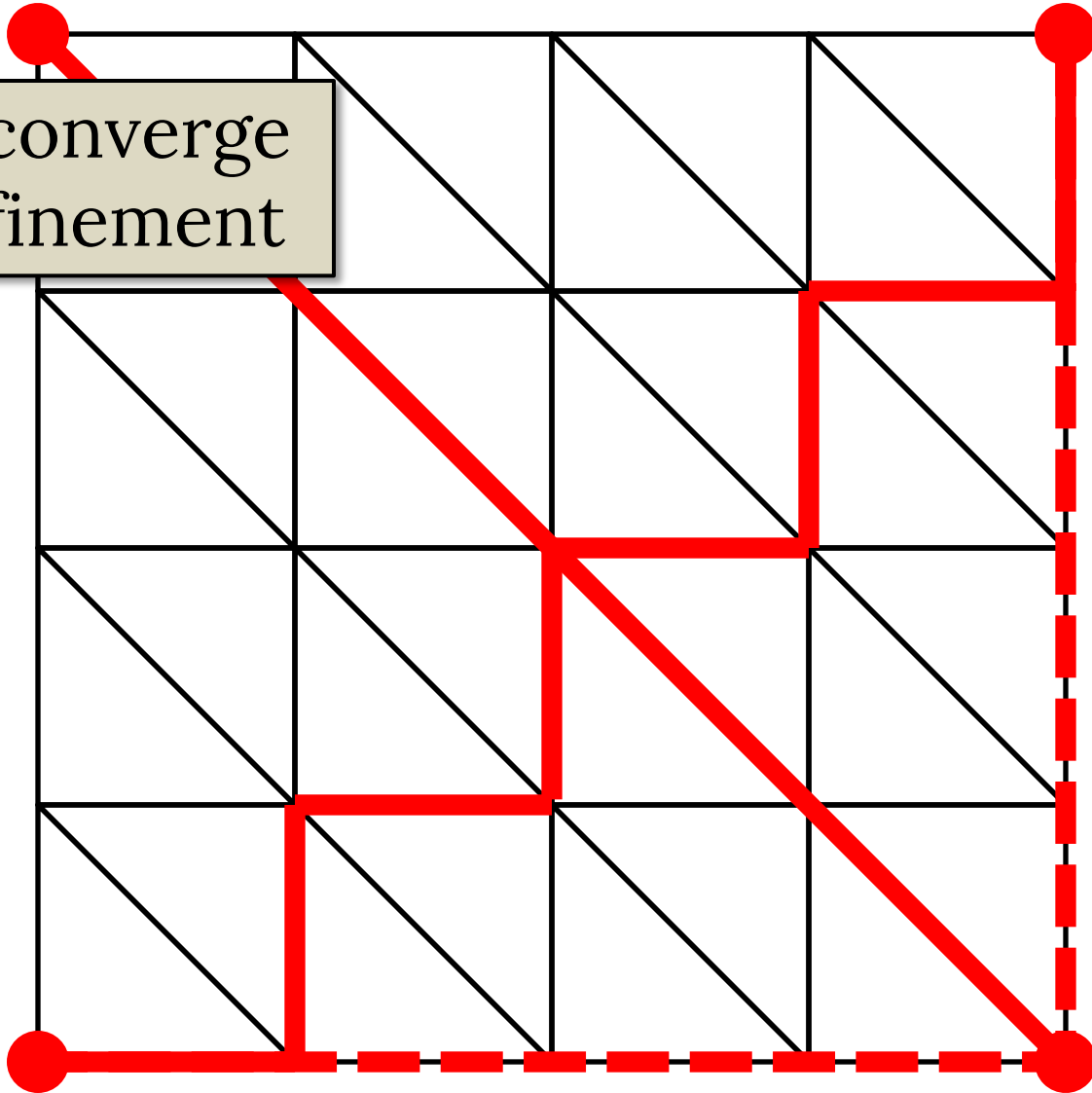
Mesh ~ Graph
find shortest path?



Mesh ~ Graph

find shortest path?

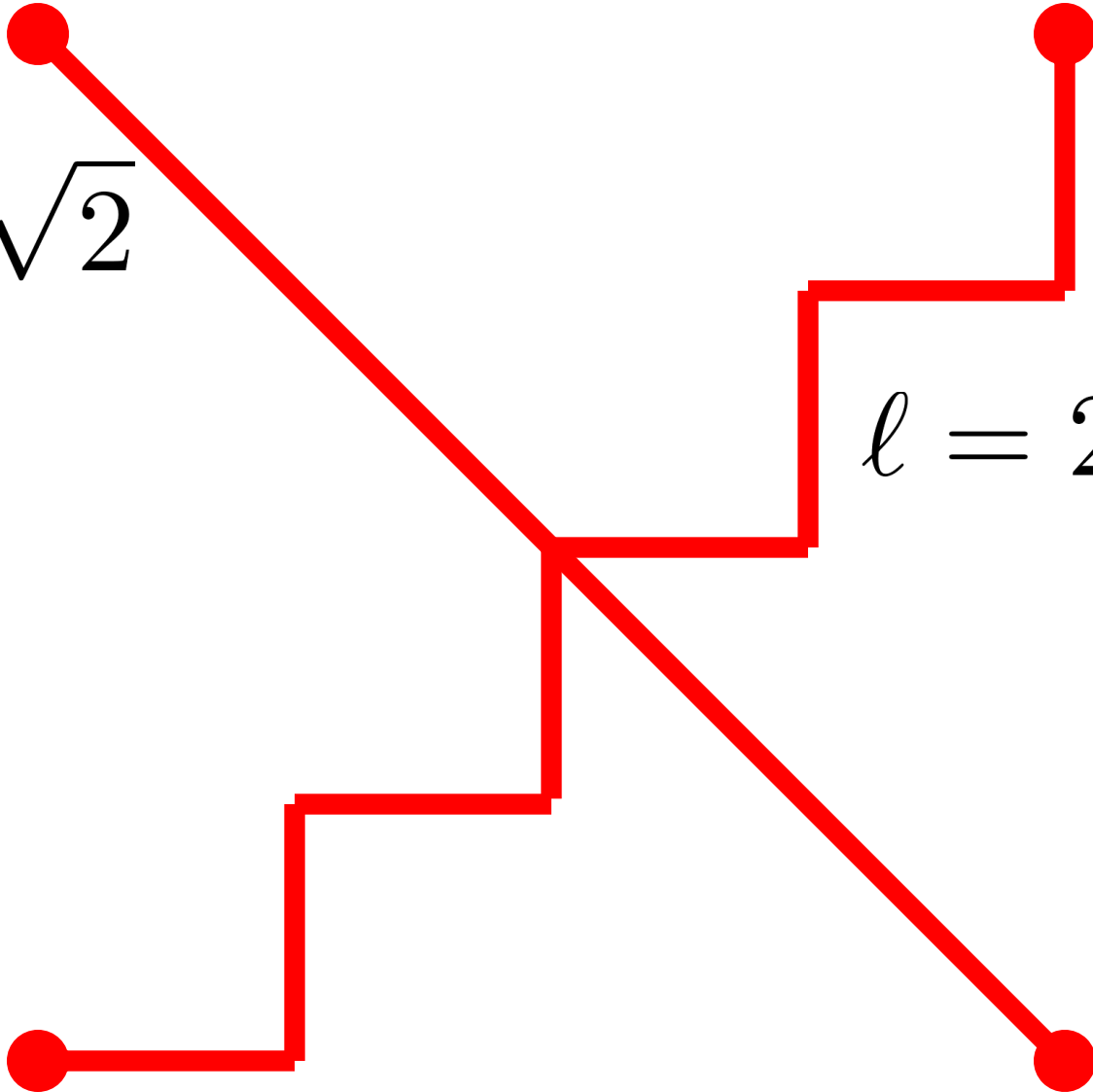
May not converge
under refinement



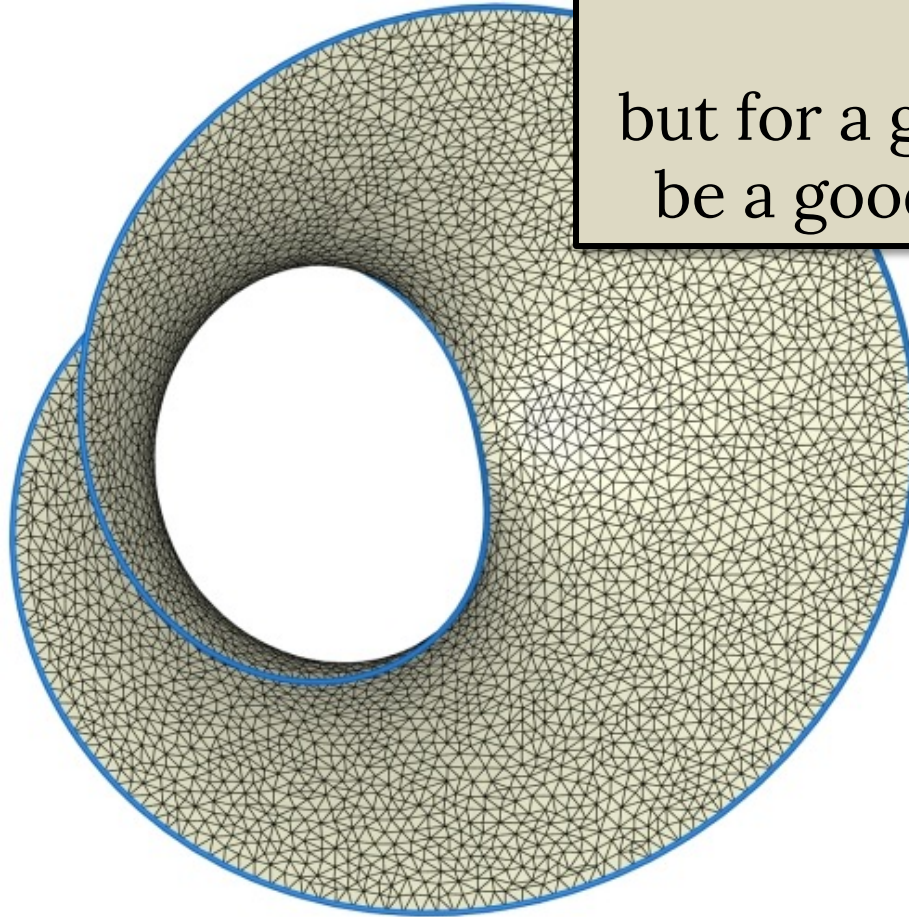
Mesh ~ Graph
find shortest path?

$$l = \sqrt{2}$$

$$l = 2$$



Mesh ~ Graph find shortest path?

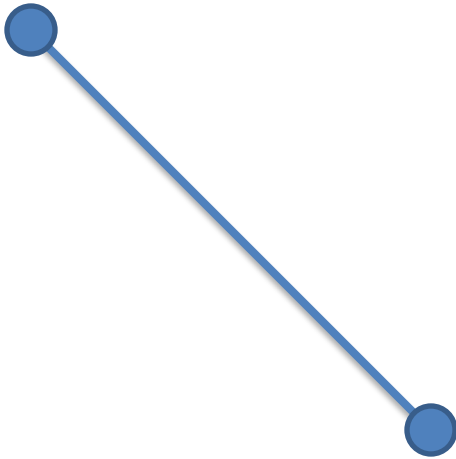


No,
but for a good mesh, it may
be a good approximation

How to discretize geodesic distance?

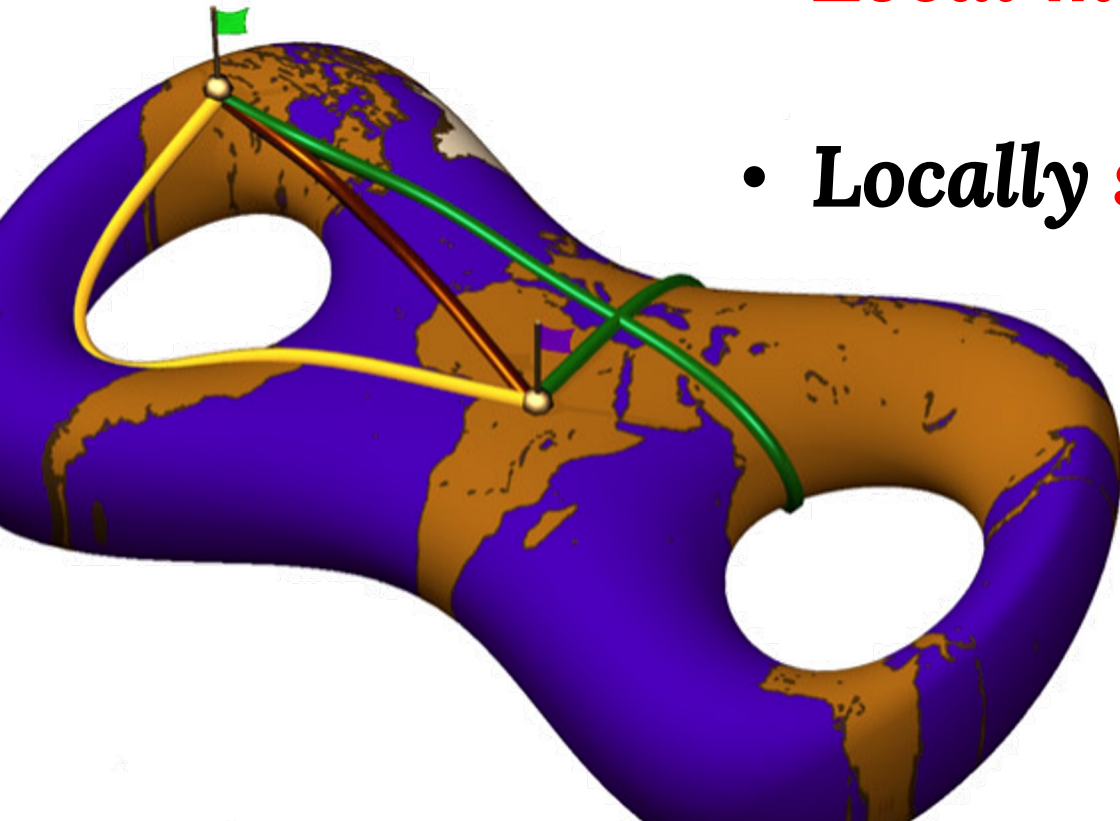
Euclidean Space

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



Surfaces: choose one

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



Not the same!

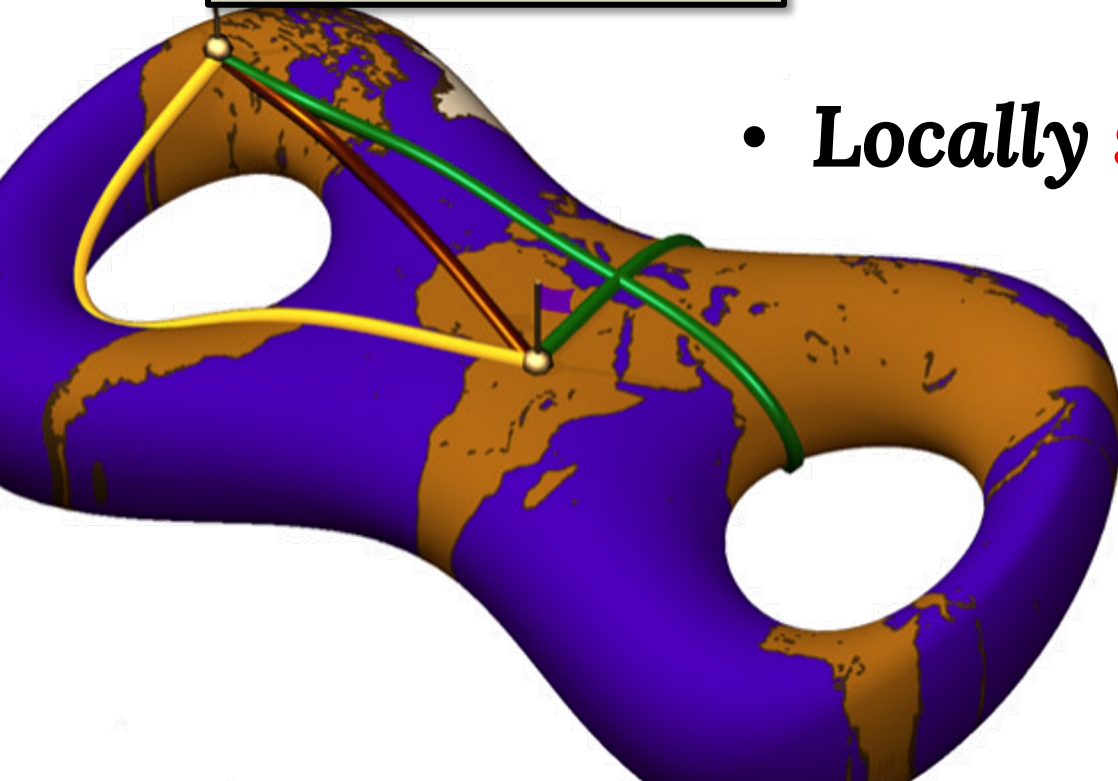
Surfaces: choose one

- **Globally shortest path**

Let's find

Local minimizer of length

- **Locally straight path**



Not the same!

Recall:

Arc Length

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

Wouldn't it be nice?

$$\int_a^b \|\gamma'(t)\|^2 dt$$

Energy of a Curve

Not the length, but turns out we can optimize it instead!

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

Note: we do not assume arclength parameterization

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

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

= when parameterized by arc length.

First Variation of Arc Length

Lemma. Let $\gamma_t: [a, b] \rightarrow S$ be a family of curves with fixed endpoints in surface S ; assume γ is parameterized by arc length at $t=0$. Then,

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

Corollary. $\gamma: [a, b] \rightarrow S$ is a geodesic iff

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

Intuition

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

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



Intuition

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0$$

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



Intuition

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0$$

***“parallel transport along the curve
preserves the tangent vector to the curve”***

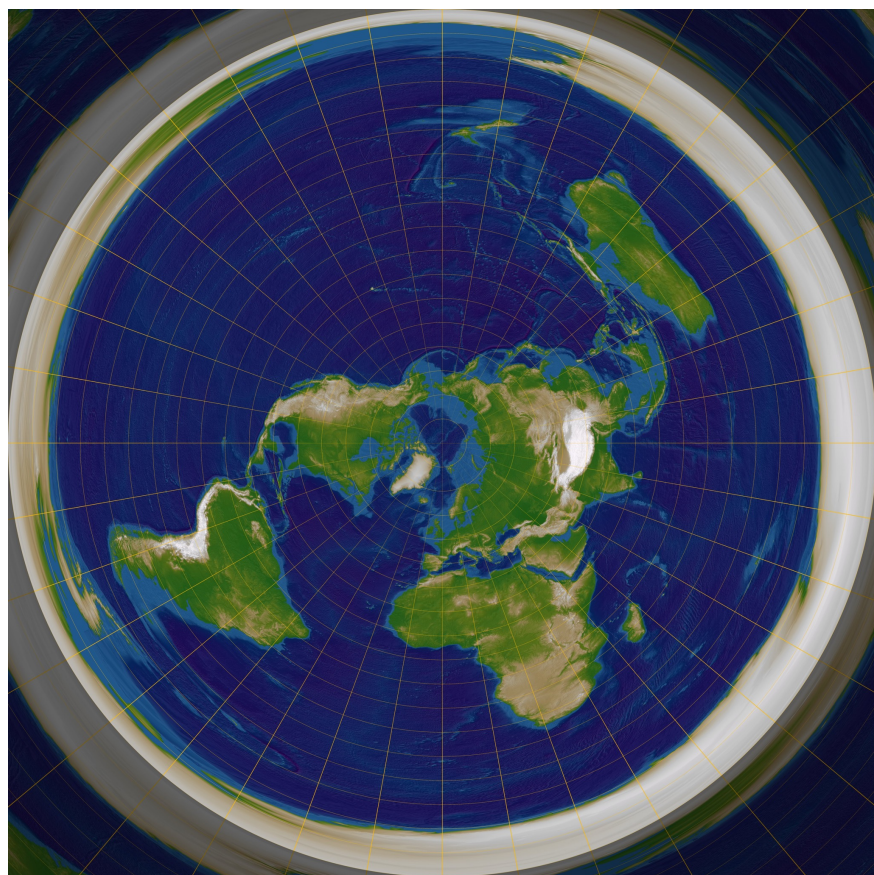


Two Local Perspectives

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

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

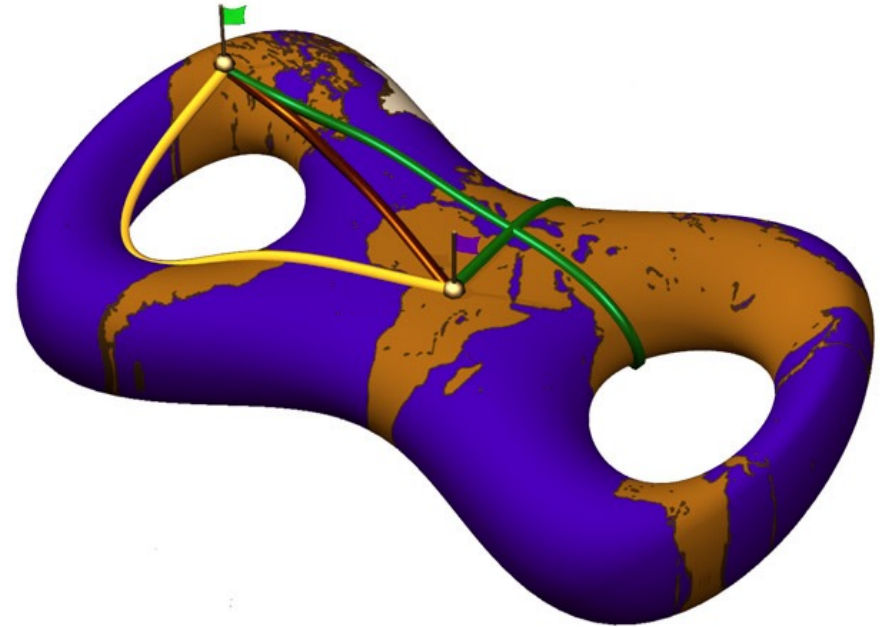
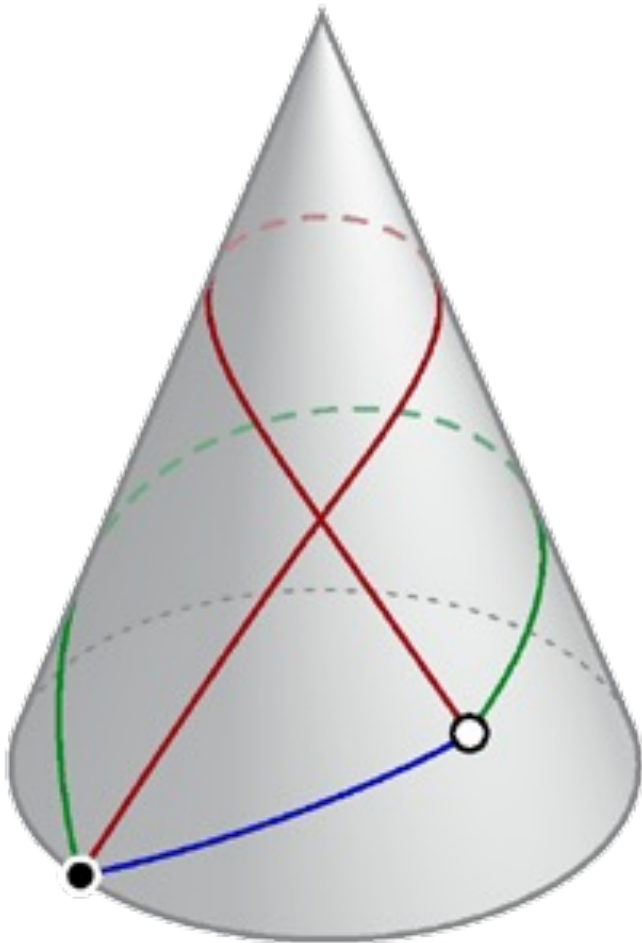
Exponential Map



$$\exp_p(v) := \gamma_v(1)$$

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

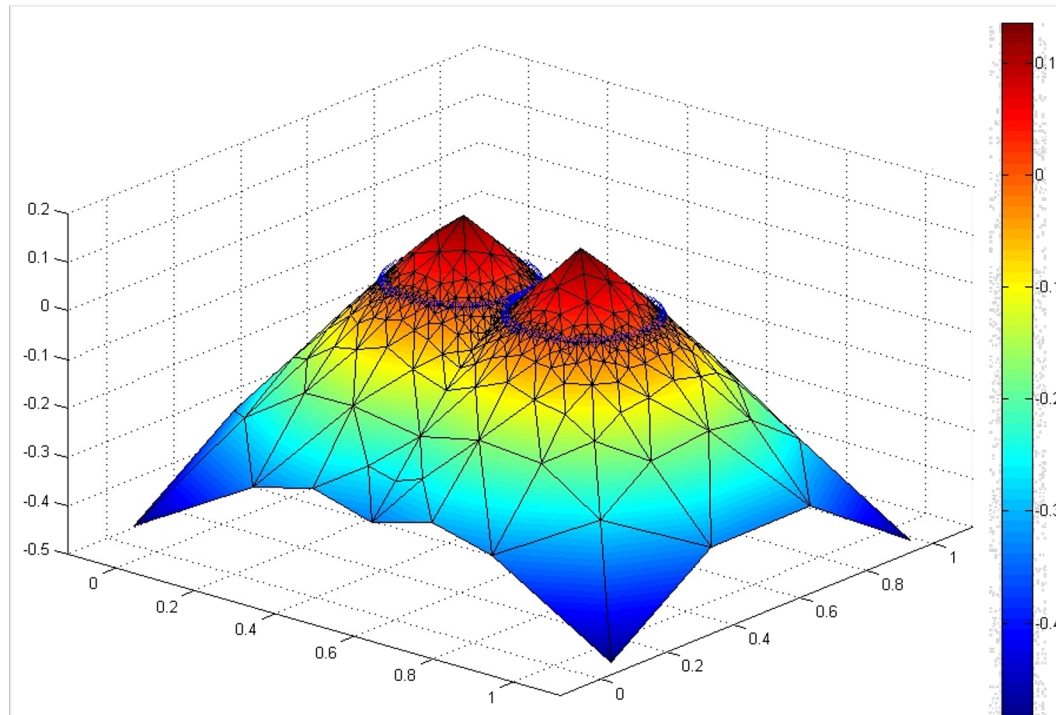
Instability of Geodesics



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

Eikonal Equation

$$\|\nabla u\|_2 = 1$$



```
\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:

$$d_0 = 0$$

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

$$S = \{\}$$

Dijkstra's Algorithm

v_0 = Source vertex

d_i = Current distance to vertex i

S = Vertices with known optimal distance

Iteration k :

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

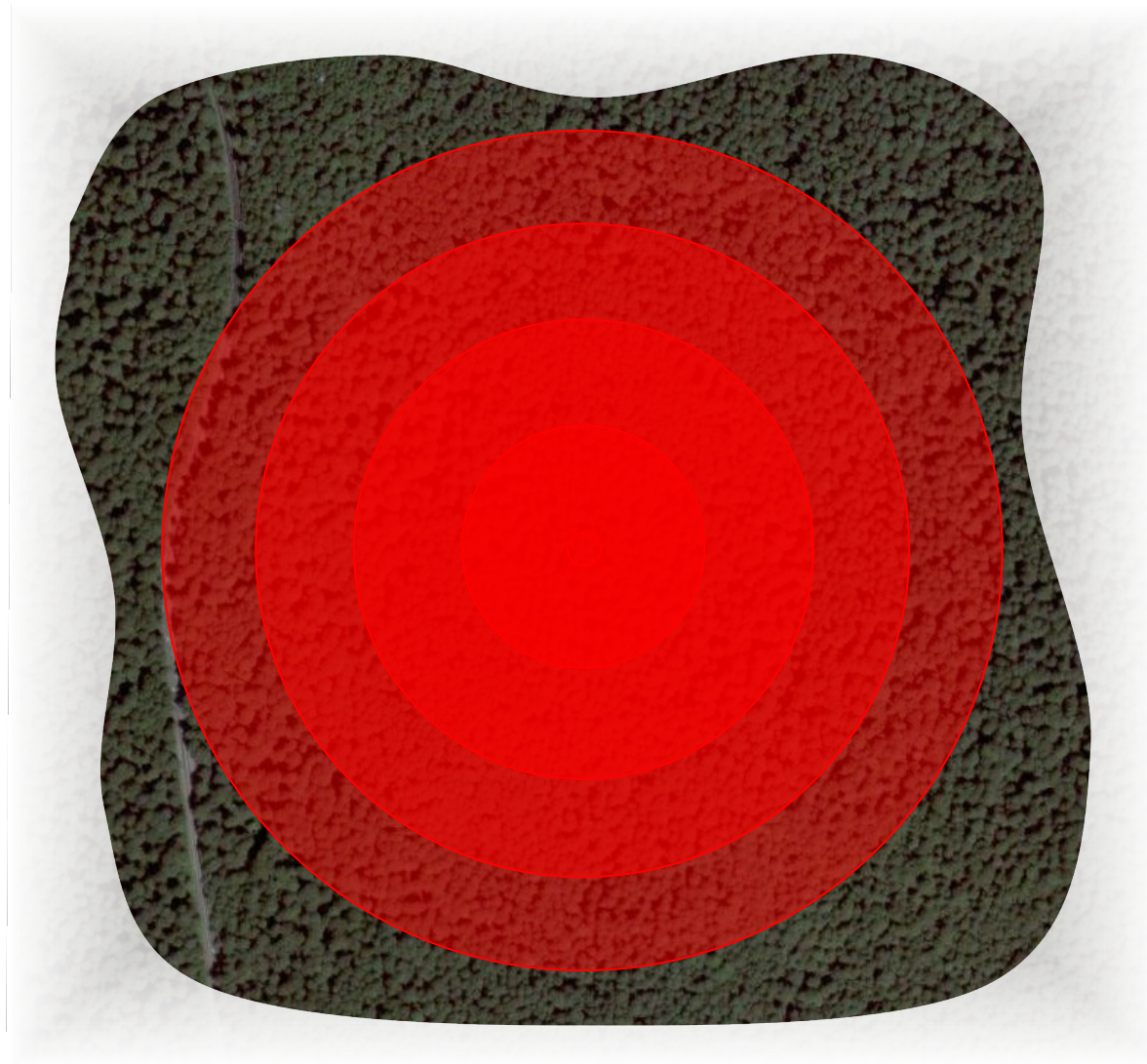
$$S \leftarrow v_k$$

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

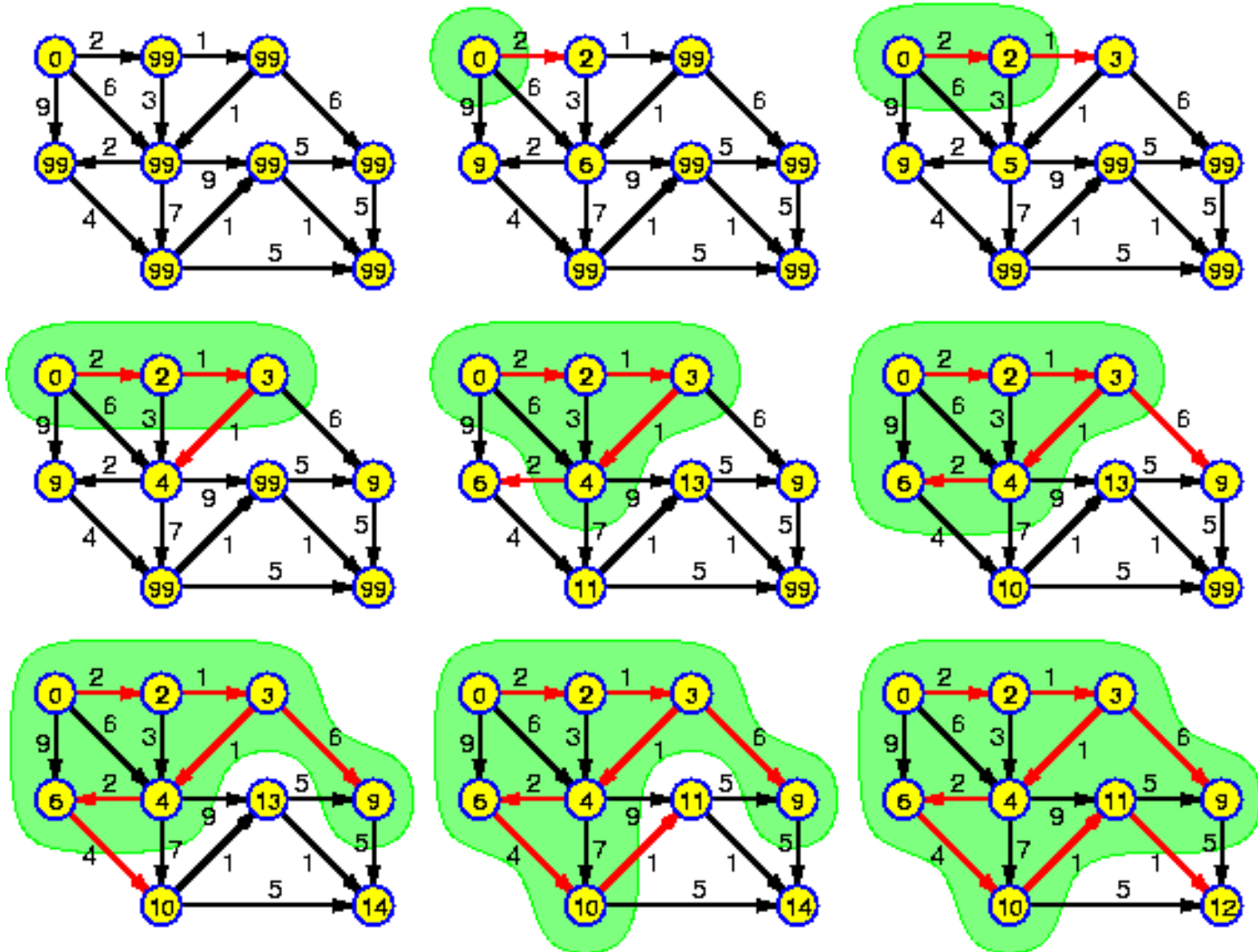
Inductive
proof:

During each iteration, S remains optimal.

Advancing Fronts



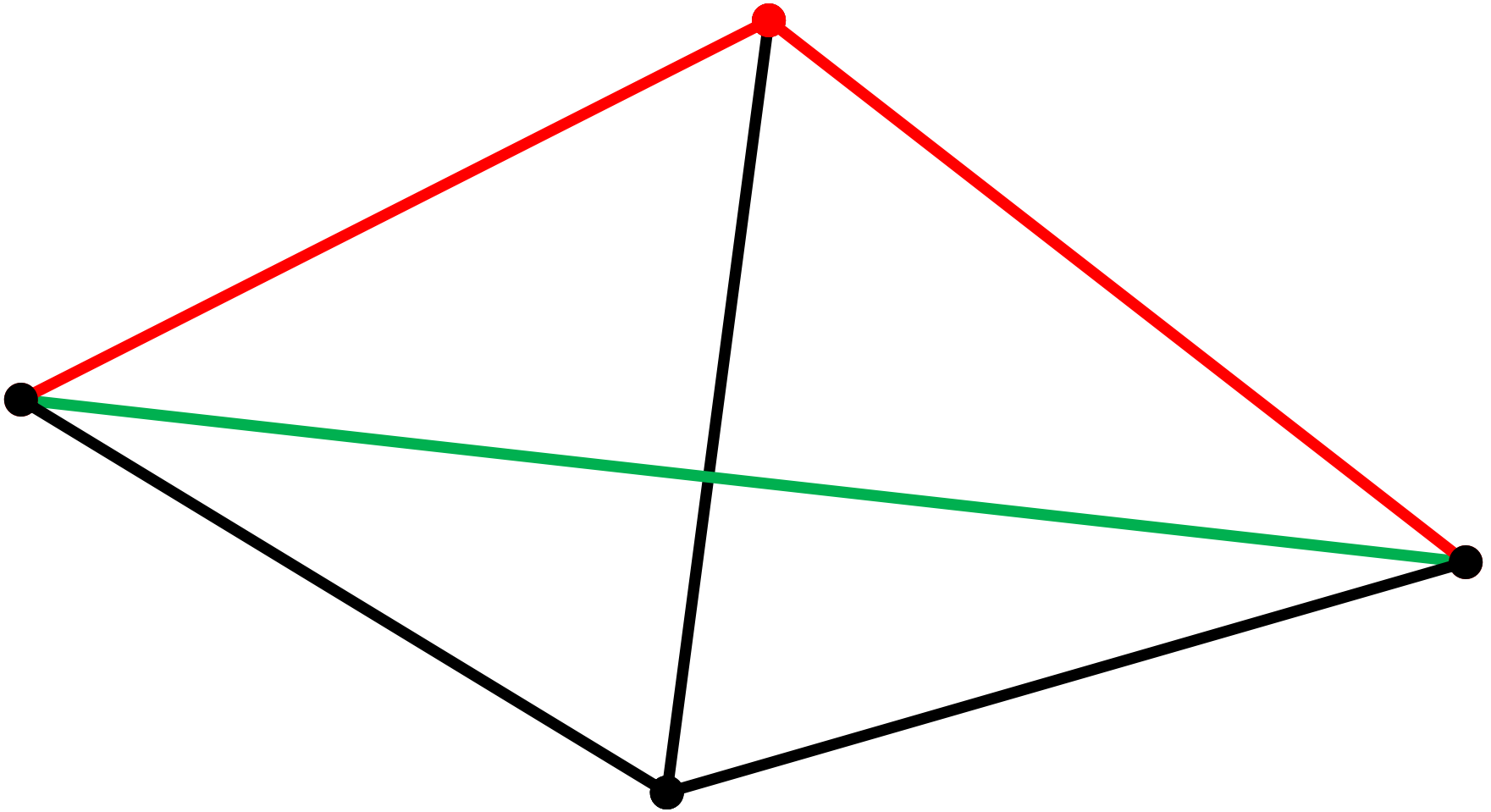
Example



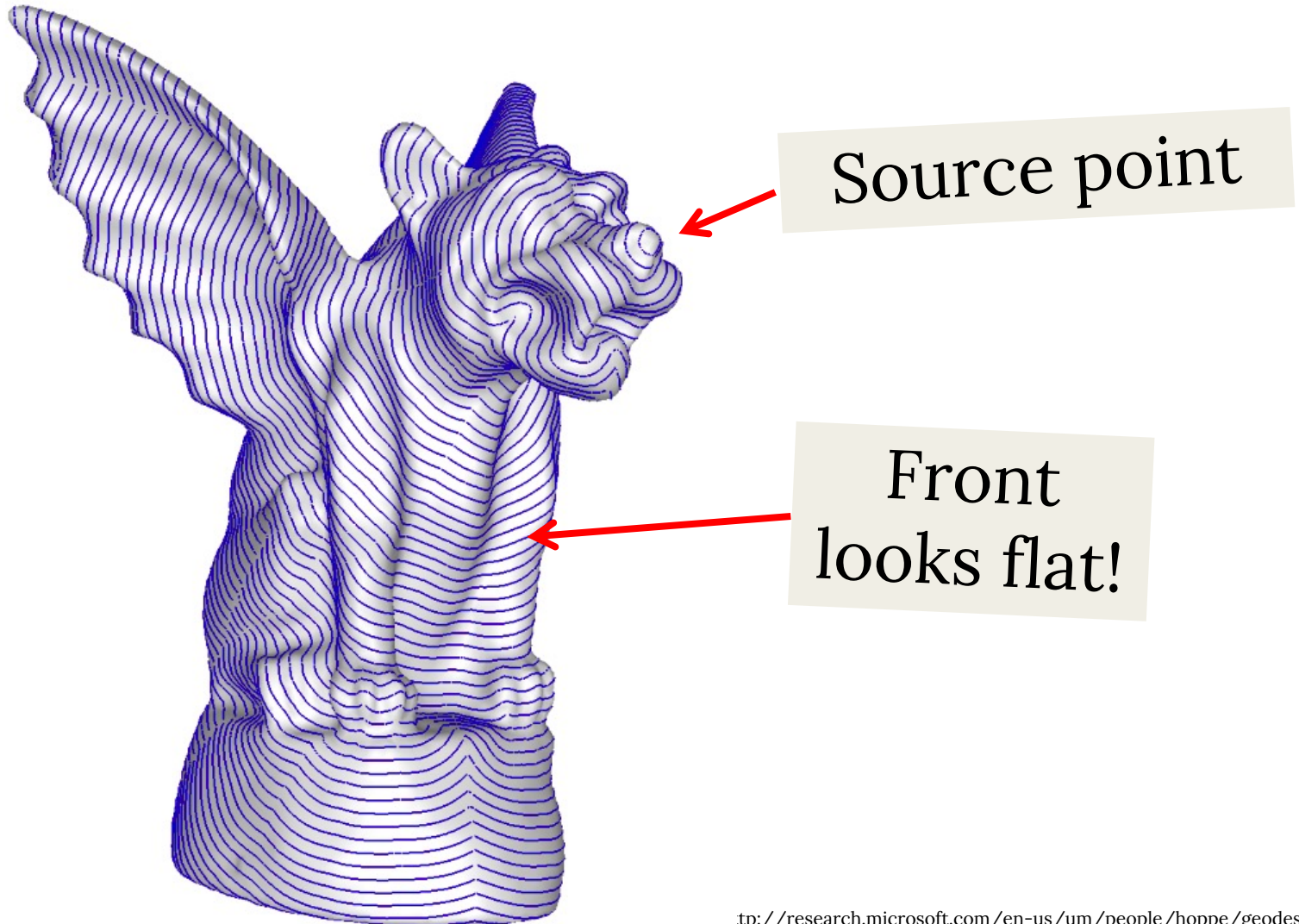
Fast Marching

*Approximately solving Eikonal
equation with a (modified)
Dijkstra algorithm*

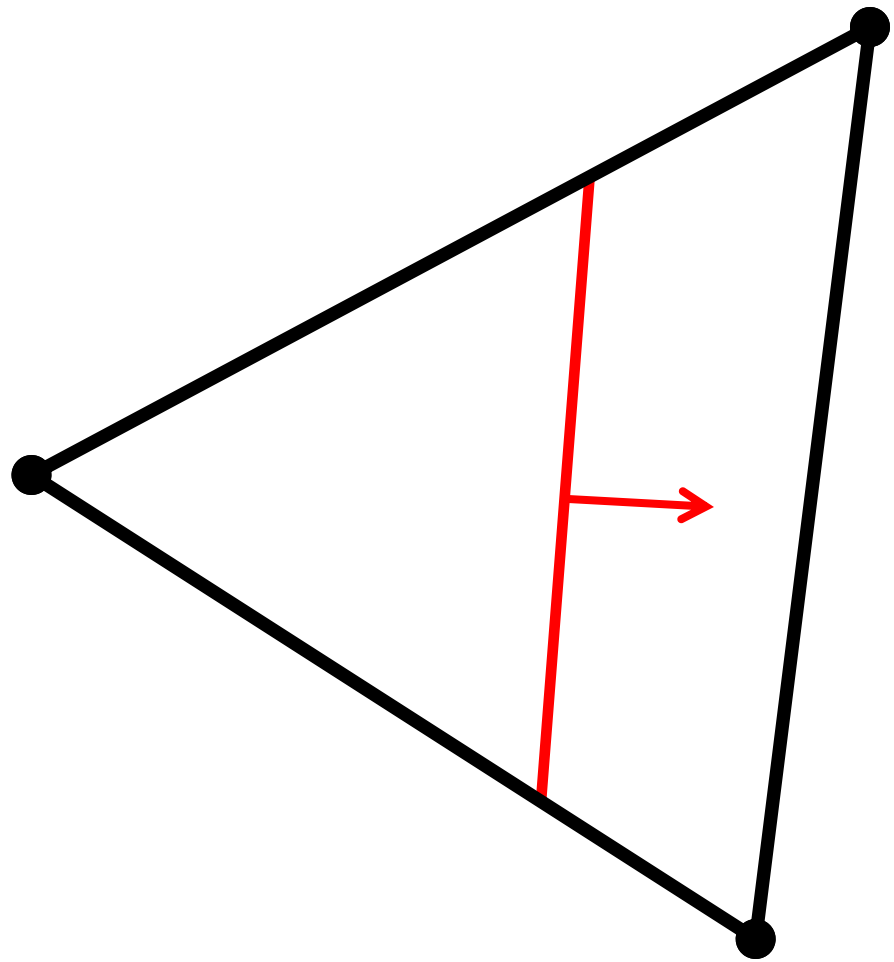
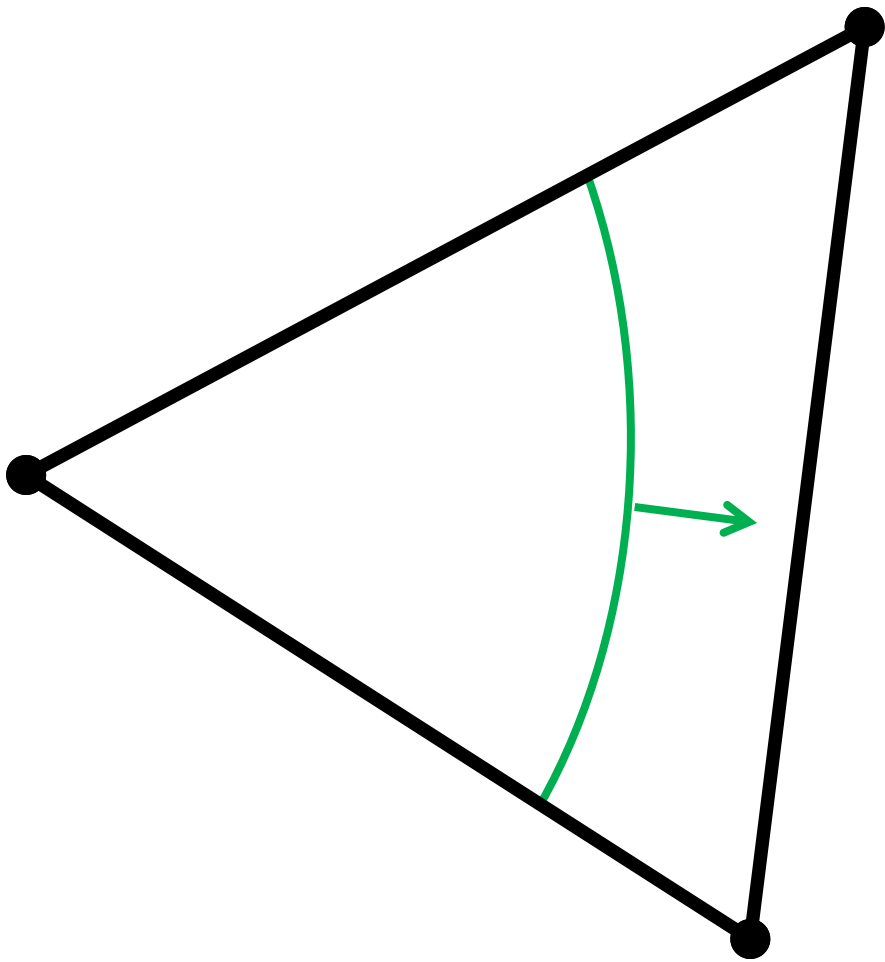
Problem



Planar Front Approximation



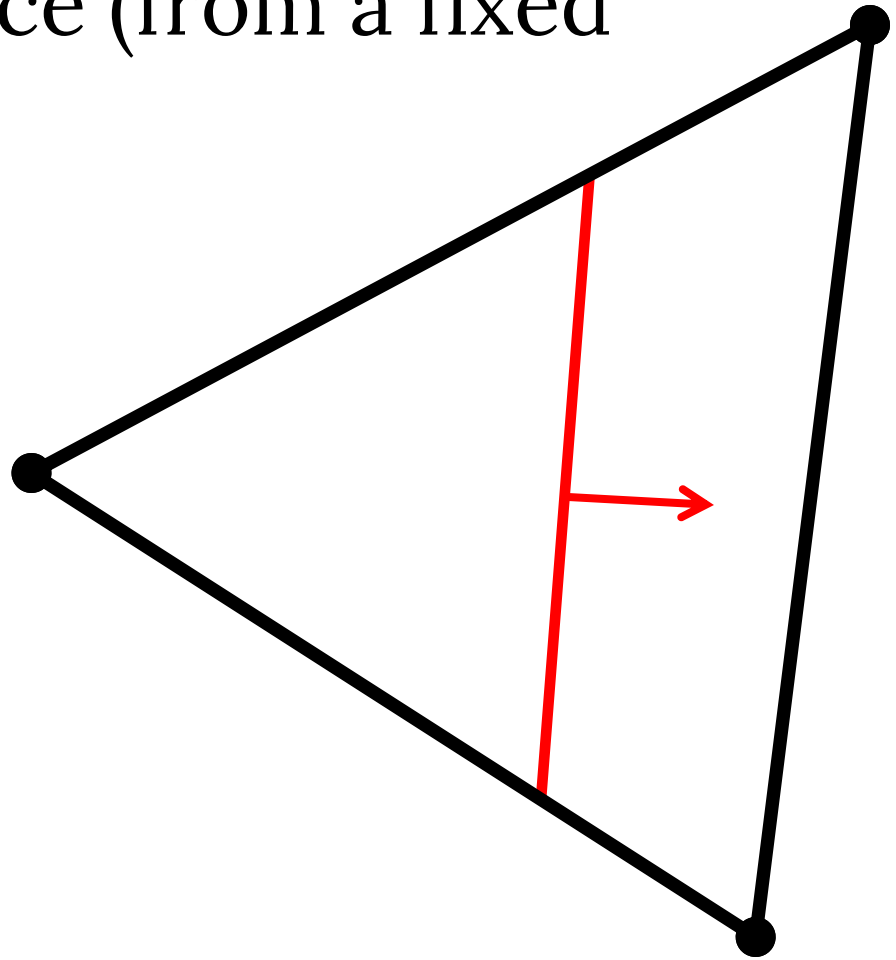
At Local Scale



Notation

$d(x)$ - geodesic distance (from a fixed vertex x_0)

- also time



Fast Marching: algorithm

- Initialization:

$$d(x_0) = 0$$

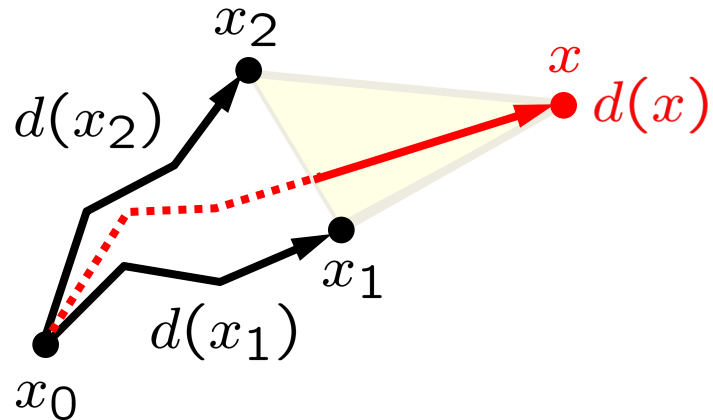
Mark x_0 as **black** //computed

Red = $N(x_0)$ //front

All other vertices are **green** //not computed

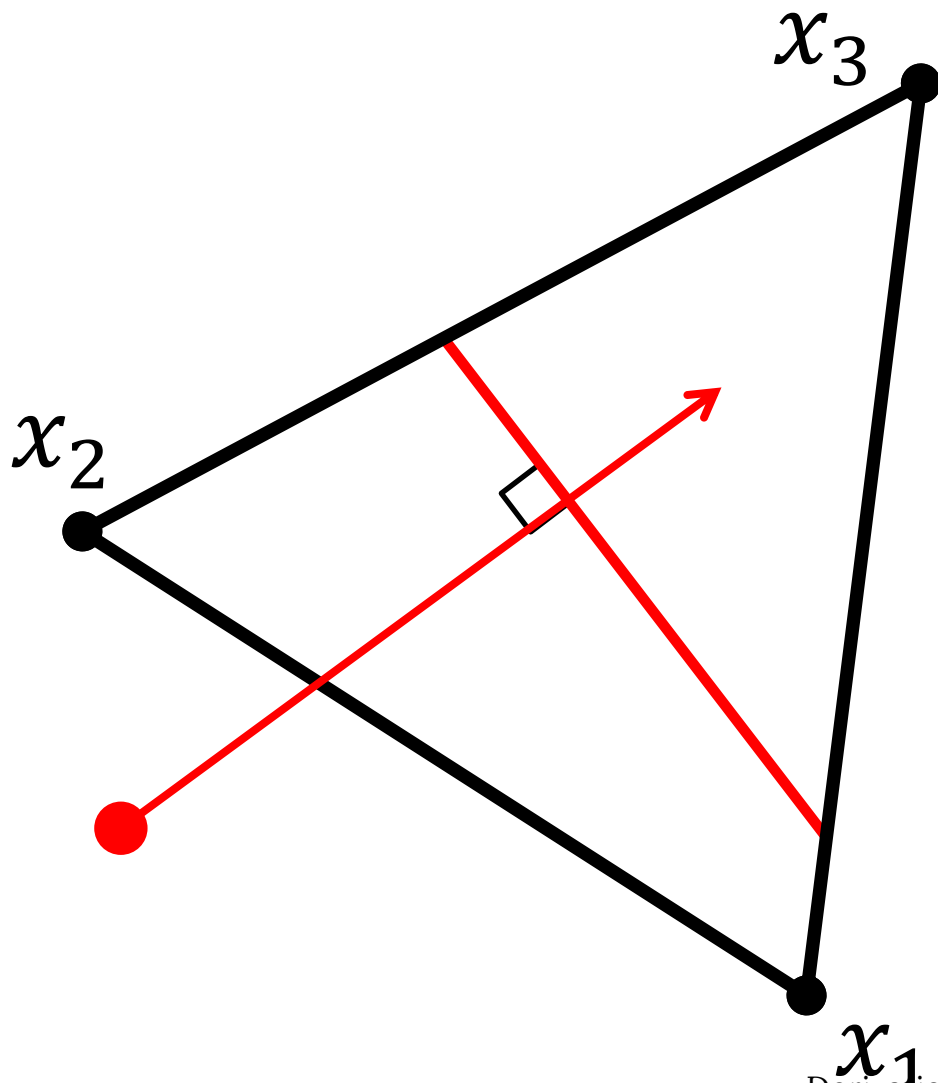
- While not everything is **black**
 - Find a **red** vertex x_1 with min d
 - Update neighbor triangles of x_1
 - Update only non-black vertices
 - Mark those as **red**
 - Mark x_1 as **black**

Fast Marching: Update Step



Vertex x updated from triangle $\ni x$

Distance computed from the other triangles vertices



Given:

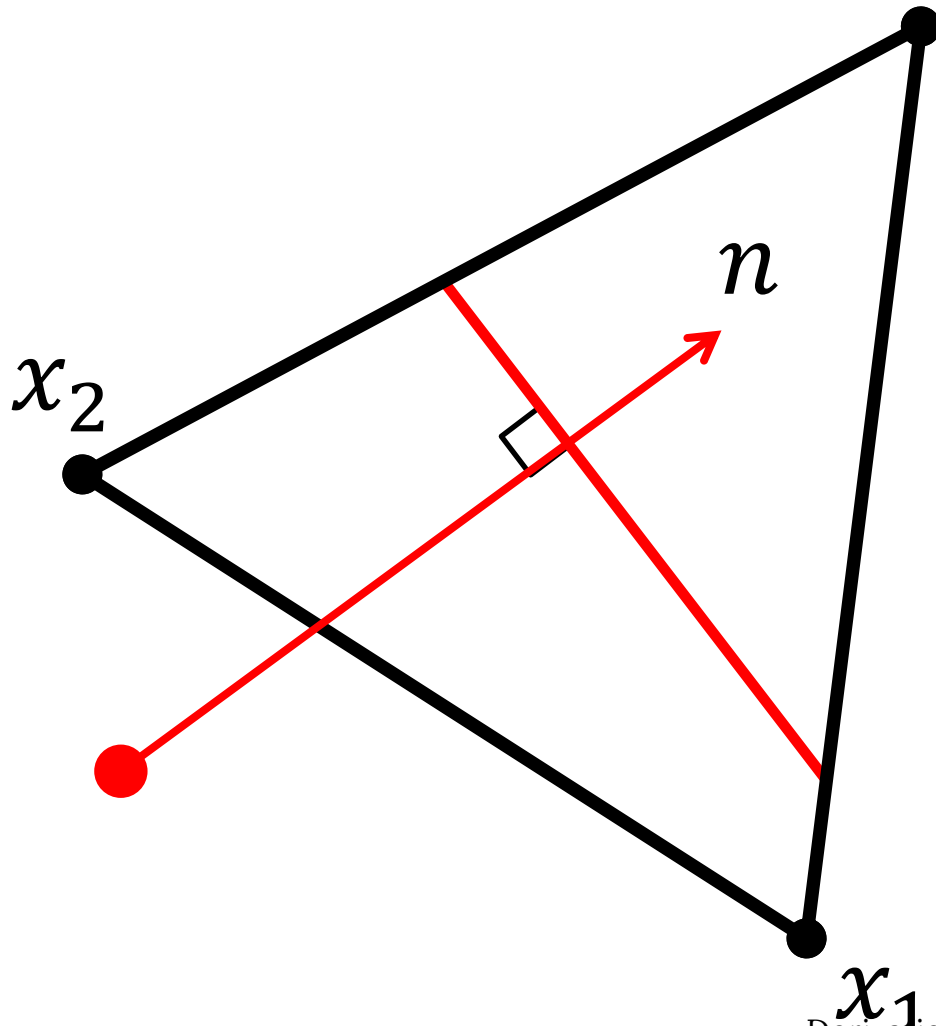
x_1, x_2, x_3 : in-plane
coordinates (assume
 $x_3 = 0$)

d_1, d_2 : geodesic
distance/time for x_1, x_2

Find:

$d_3?$

Planar Calculations



Given:

$$d_1 = n^\top x_1 + p$$

$$d_2 = n^\top x_2 + p$$

$$d = V^\top n + p \mathbf{1}_{2 \times 1}$$

Find:

$$d_3 = n^\top x_3 + p = p$$

Planar Calculations

$$d = V^{\top} n + p \mathbf{1}_{2 \times 1}$$

↓

$$n = V^{-\top} (d - p \mathbf{1}_{2 \times 1})$$

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

$$= p^2 \mathbf{1}_{2 \times 1}^{\top} Q \mathbf{1}_{2 \times 1} - 2p \mathbf{1}_{2 \times 1}^{\top} Q d + d^{\top} Q d$$

$$Q := (V^{\top} V)^{-1}$$

Planar Calculations

$$1 = p^2 \cdot \mathbf{1}_{2 \times 1}^\top Q \mathbf{1}_{2 \times 1} - 2p \cdot \mathbf{1}_{2 \times 1}^\top Q d + d^\top Q d$$

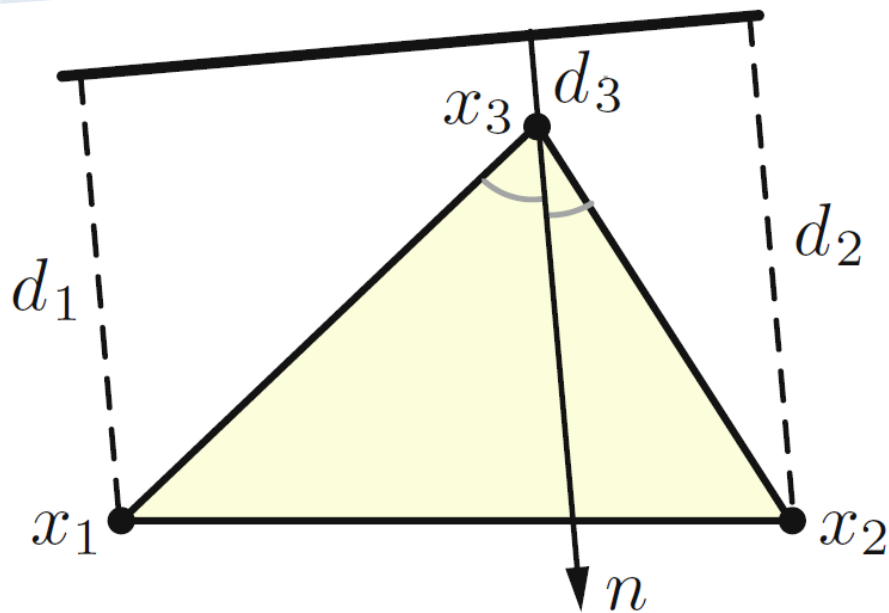
Quadratic equation for p

Find:

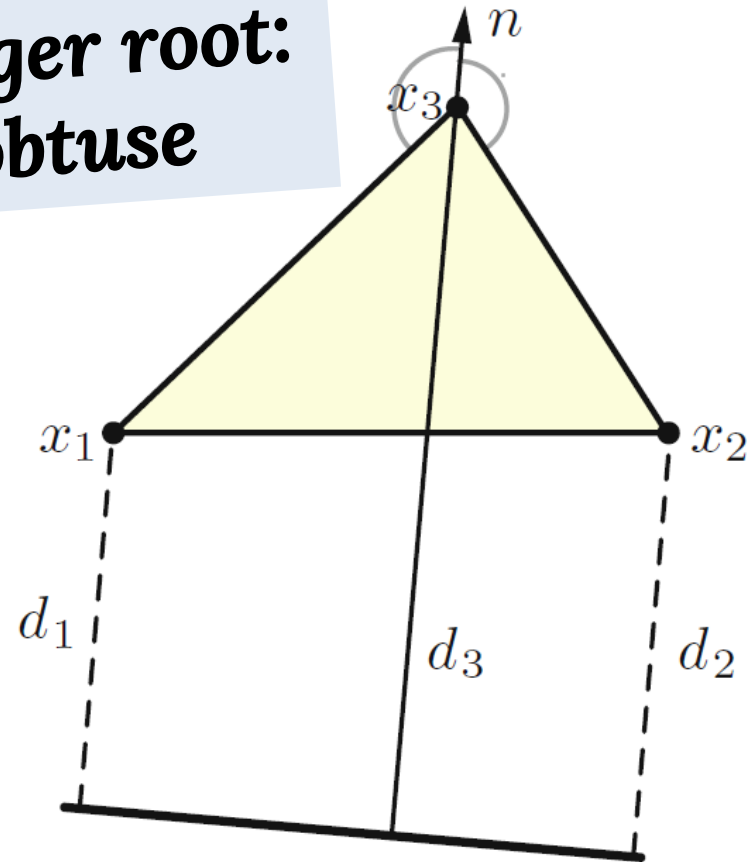
$$d_3 = n^\top \cancel{x_3^0} + p = p$$

Two Roots

**Smaller root:
acute**



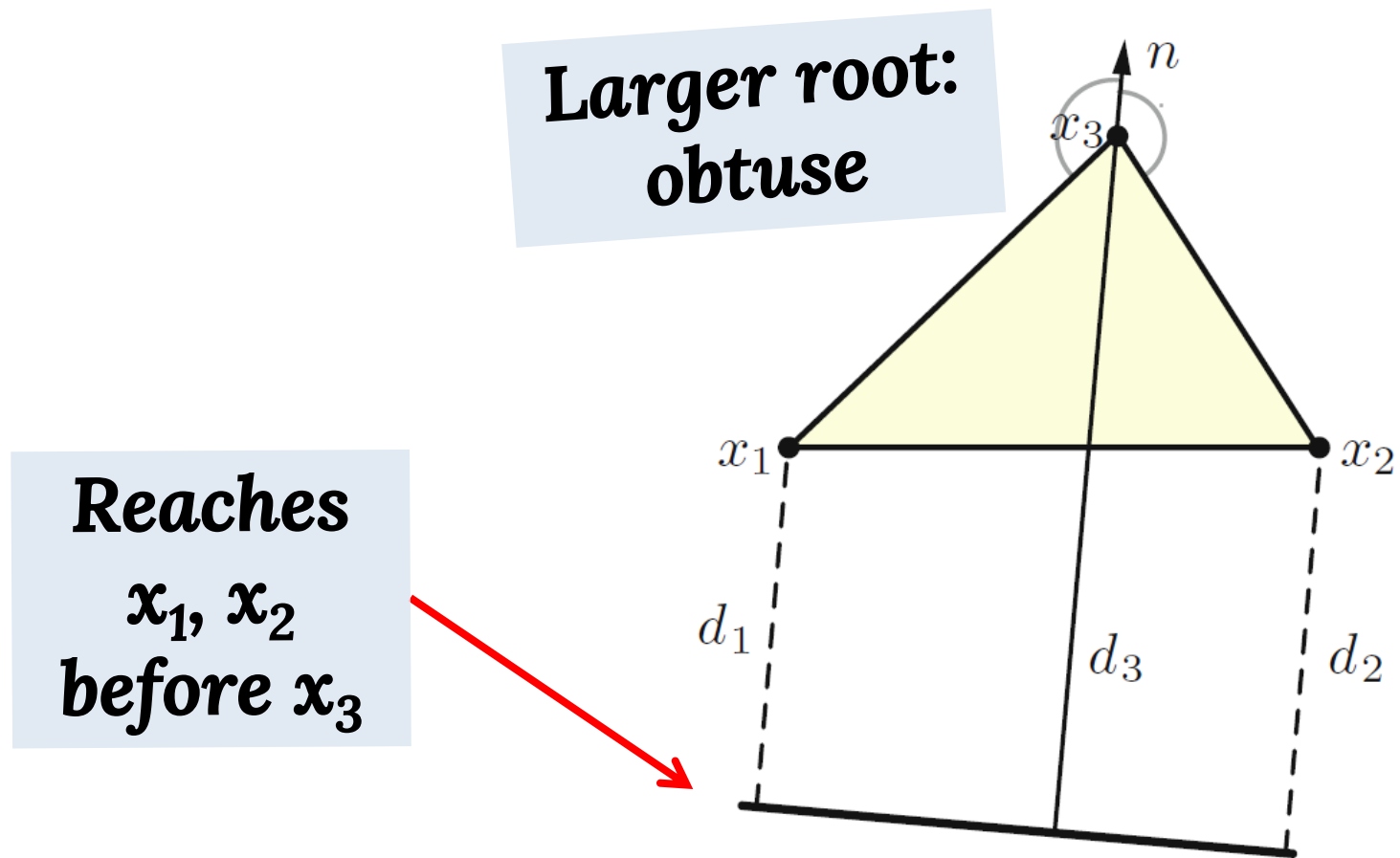
**Larger root:
obtuse**



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

Two orientations for the normal

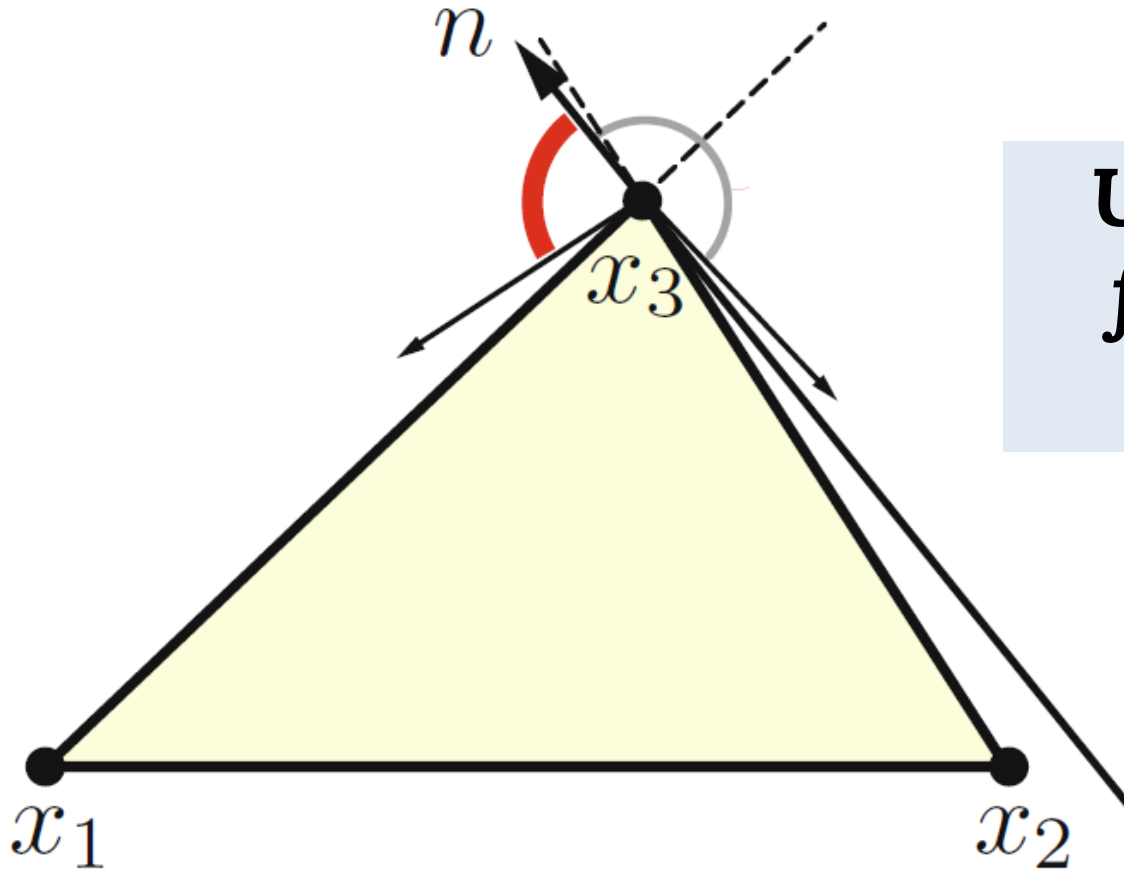
Larger Root: Consistent



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

Two orientations for the normal

Additional Issue

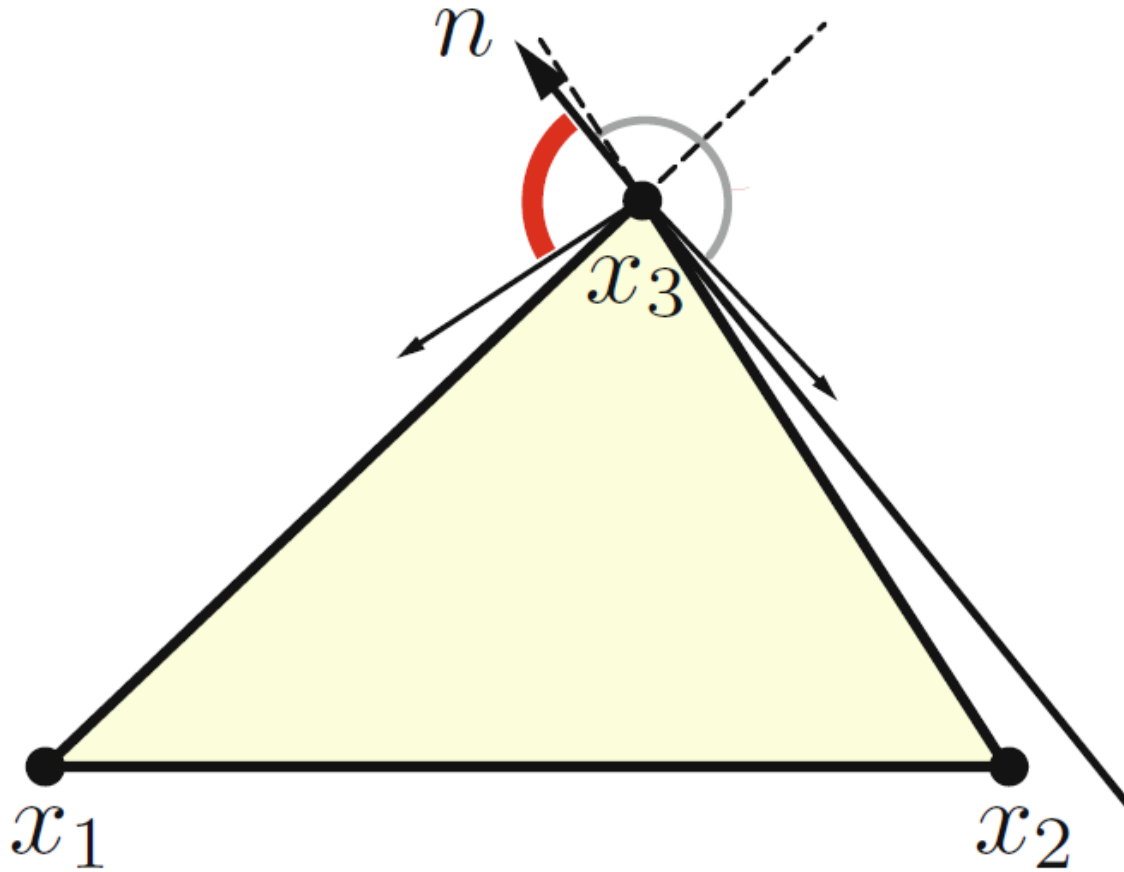


Update should be from a different triangle!

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

Front from outside the triangle

Condition for Front Direction



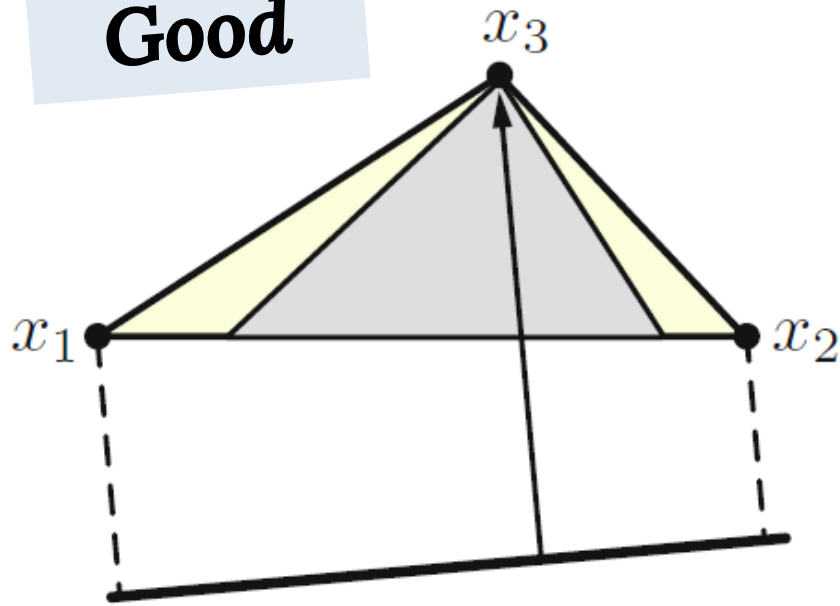
$$QV^T n < 0$$

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

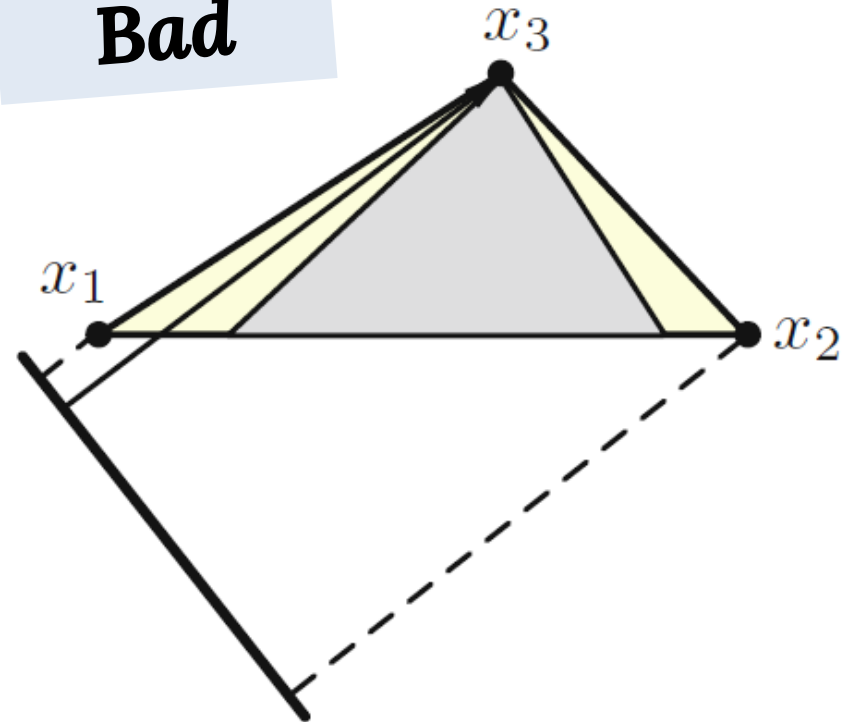
Front from outside the triangle

Obtuse Triangles

Good



Bad



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

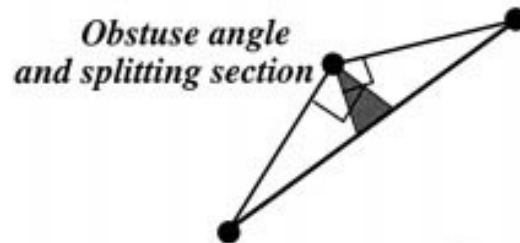
Must reach x_3 after x_1 and x_2

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]



Summary: Update Step

input : non-obtuse triangle with the vertices x_1, x_2, x_3 , and the corresponding arrival times d_1, d_2, d_3

output : updated d_3

1 Solve the quadratic equation

$$p = \frac{1_{2 \times 1}^T Q d + \sqrt{(1_{2 \times 1}^T Q d)^2 - 1_{2 \times 1}^T Q 1_{2 \times 1} \cdot (d^T Q d - 1)}}{1_{2 \times 1}^T Q 1_{2 \times 1}}.$$

where $V = (x_1 - x_3, x_2 - x_3)$, and $d = (d_1, d_2)^T$.

2 Compute the front propagation direction $n = V^{-T}(d - p \cdot 1_{2 \times 1})$

3 **if** $(V^T V)^{-1} V^T n < 0$ **then**

4 $d_3 \leftarrow \min\{d_3, p\}$

5 **else**

6 $d_3 \leftarrow \min\{d_3, d_1 + \|x_1\|, d_2 + \|x_2\|\}$

7 **end**

Fast Marching vs. Dijkstra

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

Eikonal Equation

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

Greek: "Image"

$$\begin{aligned} 1 &= n^\top n \\ &= (d - p\mathbf{1}_{2 \times 1})^\top X^{-1} X^{-\top} (d - p\mathbf{1}_{2 \times 1}) \\ &= p^2 \cdot \mathbf{1}_{2 \times 1}^\top Q \mathbf{1}_{2 \times 1} - 2p \cdot \mathbf{1}_{2 \times 1}^\top Q d + d^\top Q d \\ Q &:= (X^\top X)^{-1} \end{aligned}$$

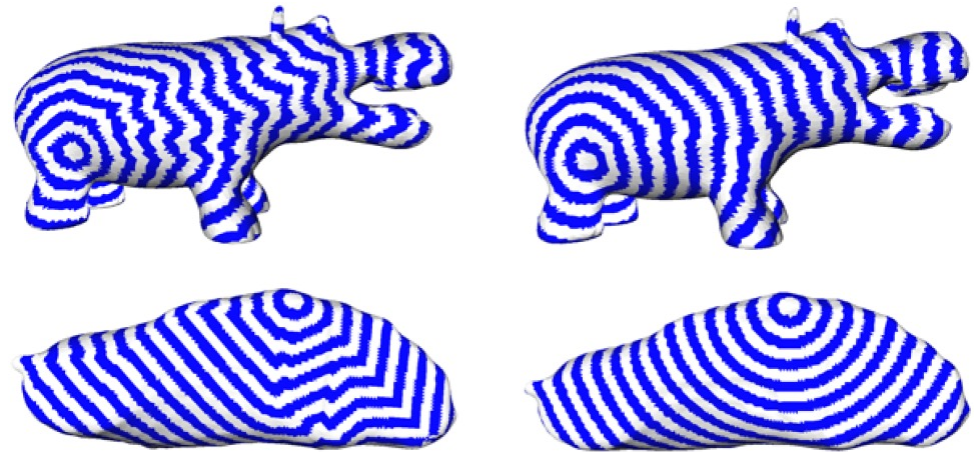
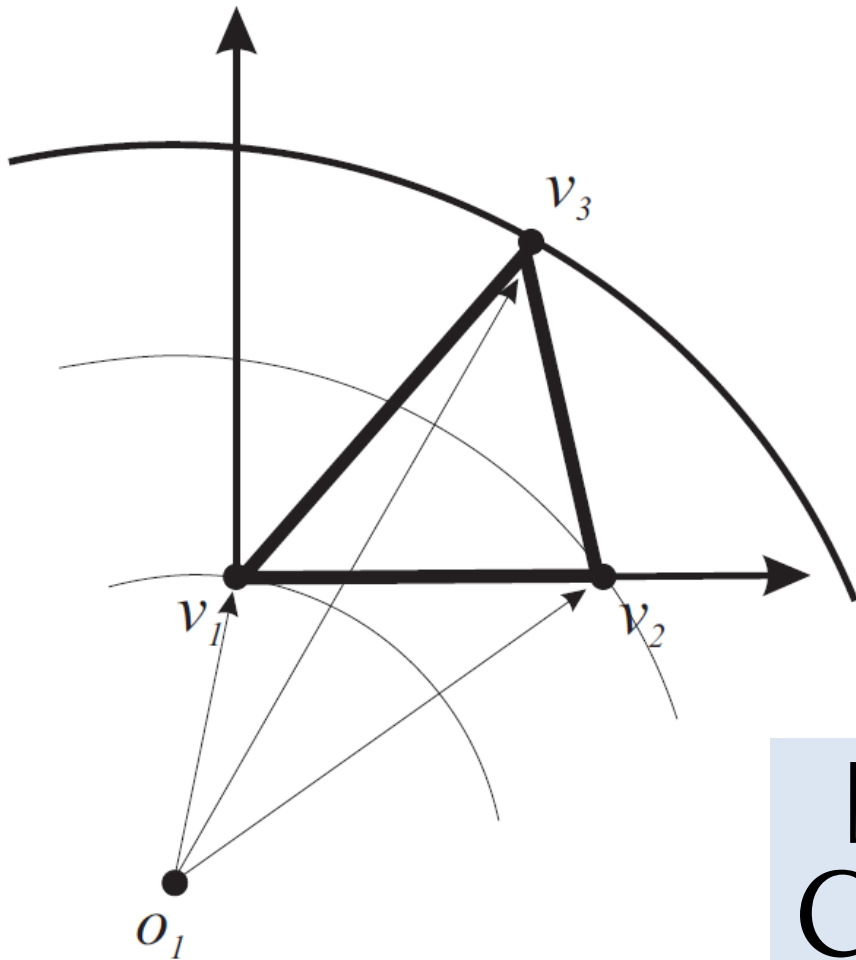
Solutions are geodesic distance

 **WARNING**



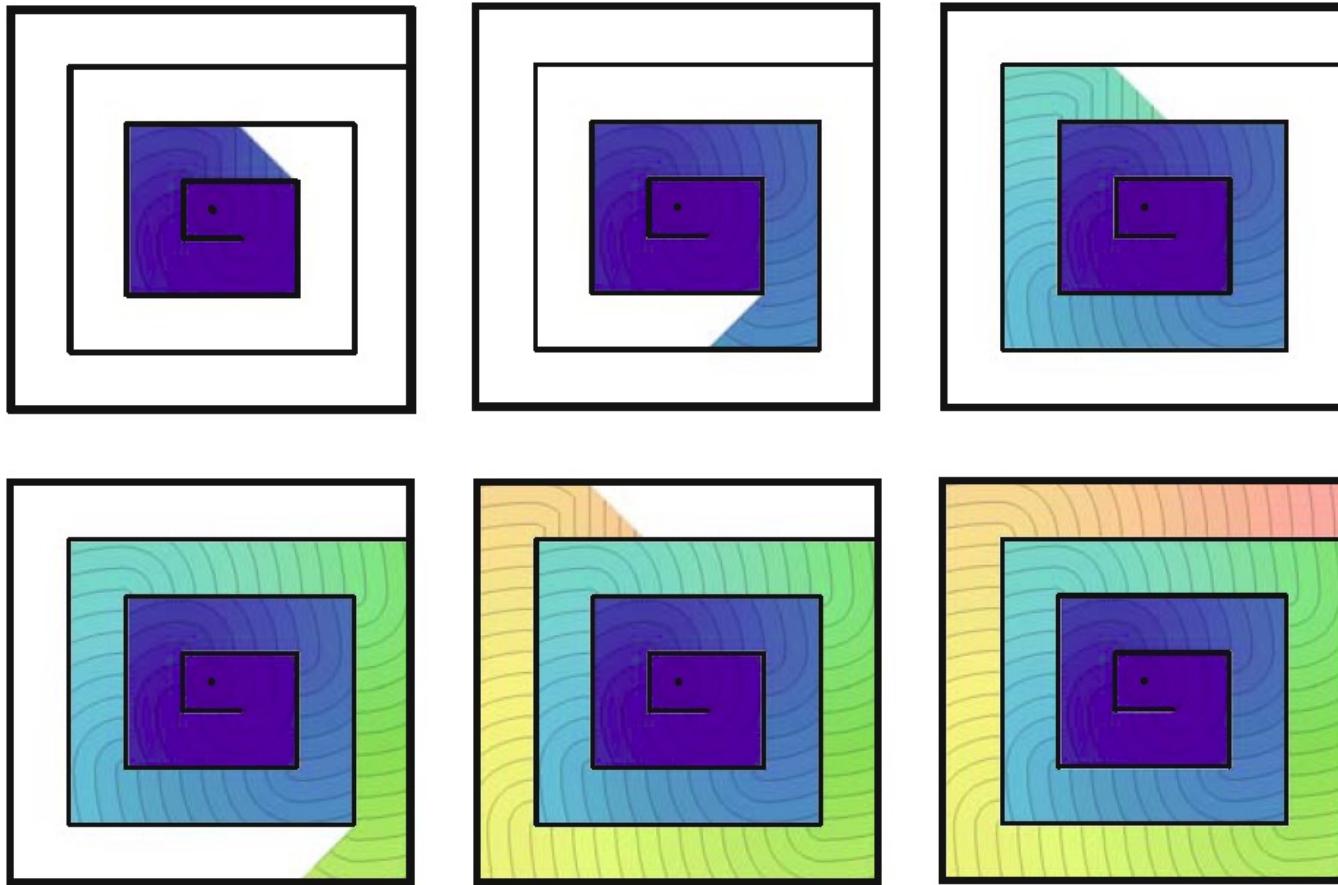
**STILL AN
APPROXIMATION**

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 Fast Marching

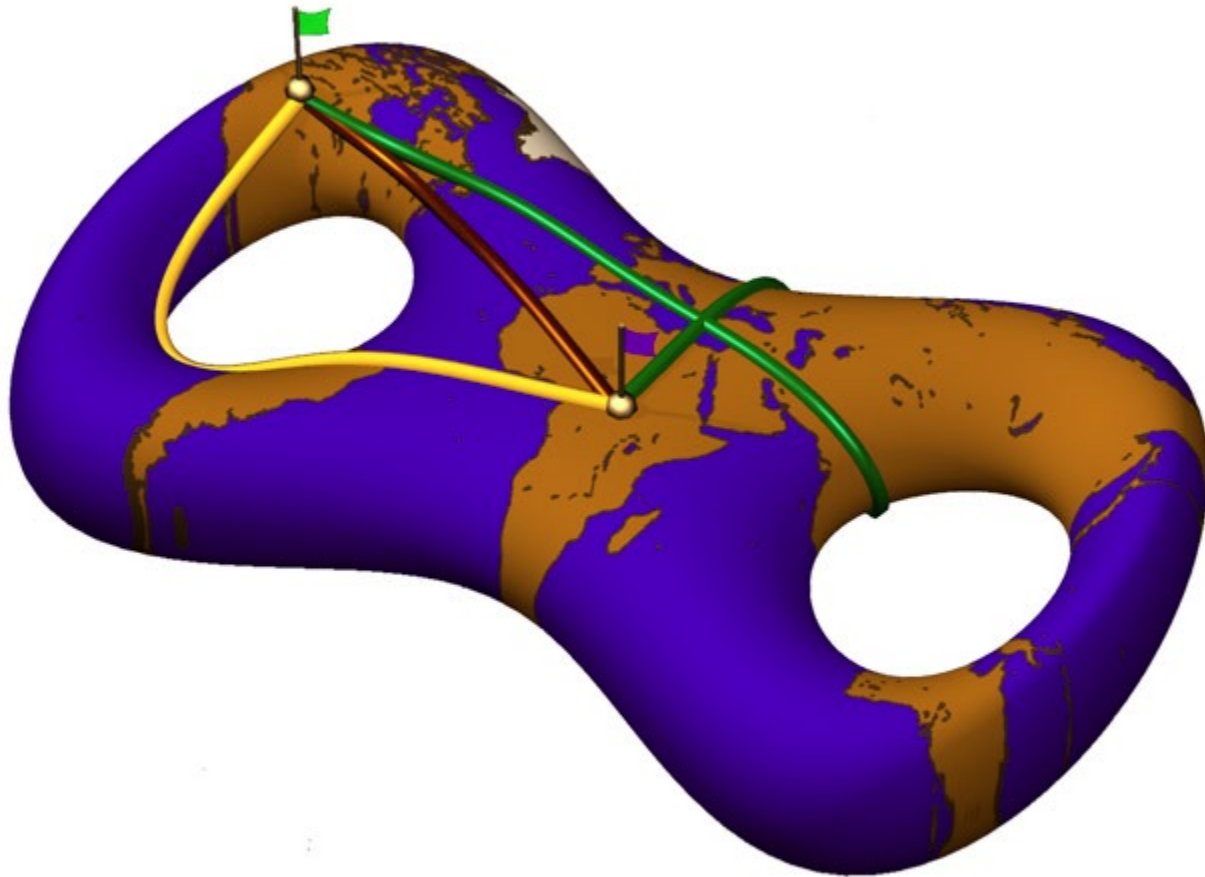
Algorithm 1 The Heat Method

- 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 $\Delta \phi = \nabla \cdot X$.
-



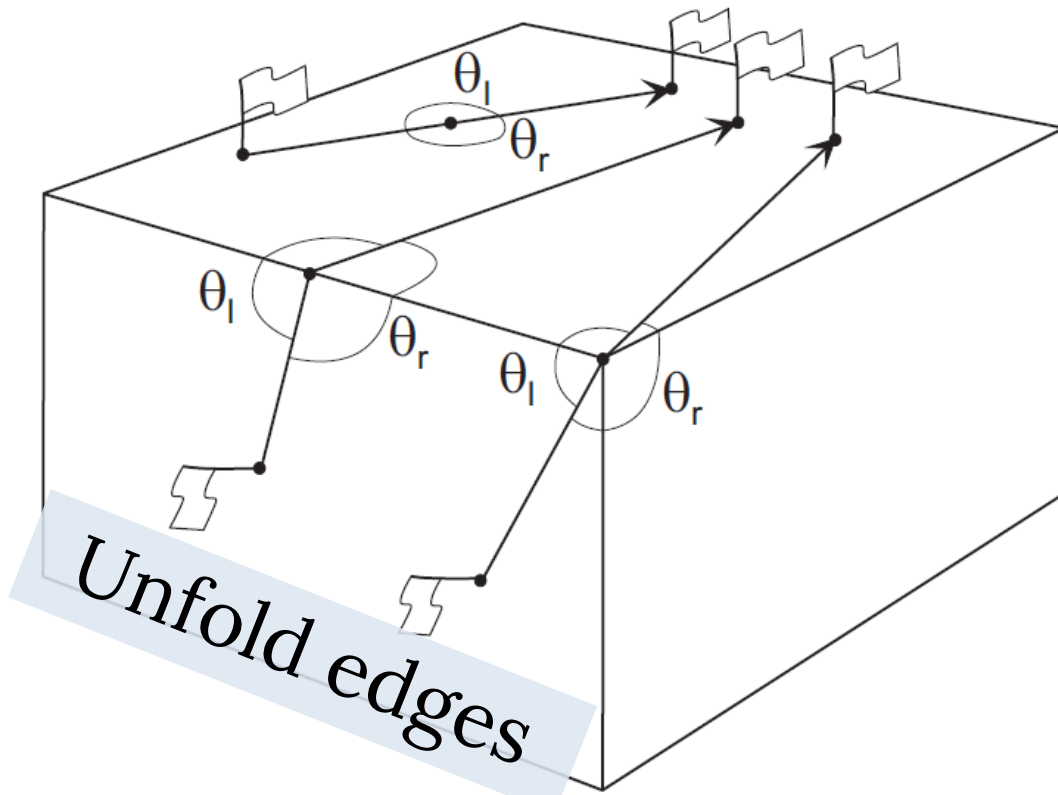
Crane, Weischedel, and Wardetzky. "Geodesics in Heat." TOG 2013.

Tracing Geodesic Curves



Trace gradient of distance function

Initial Value Problem



Equal left and right angles

Unfold edges

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

Trace a single geodesic exactly

Exact Geodesics

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

© 1987 Society for Industrial and Applied Mathematics
005

THE DISCRETE GEODESIC PROBLEM*

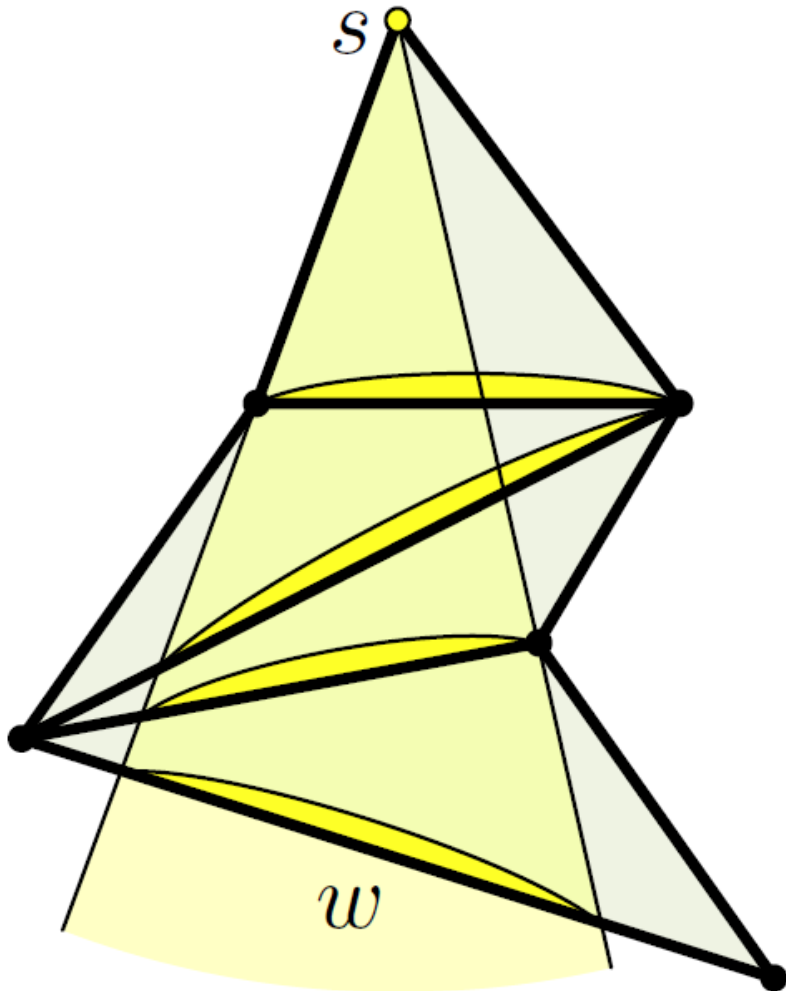
JOSEPH S. B. MITCHELL[†], DAVID M. MOUNT[‡] AND CHRISTOS H. PAPADIMITRIOU[§]

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.

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 1996, Sander et al. 2003]) and the result generally has less distortion if the cuts are geodesic. Geodesic paths are also used in mesh segmentation, as done in [Katz and Tal 2003; Funtowicz et al.

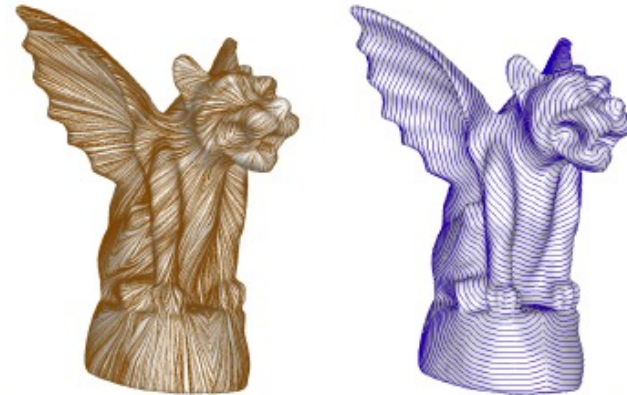


Figure 1: Geodesic paths from a source vertex, and isolines of the geodesic distance function.

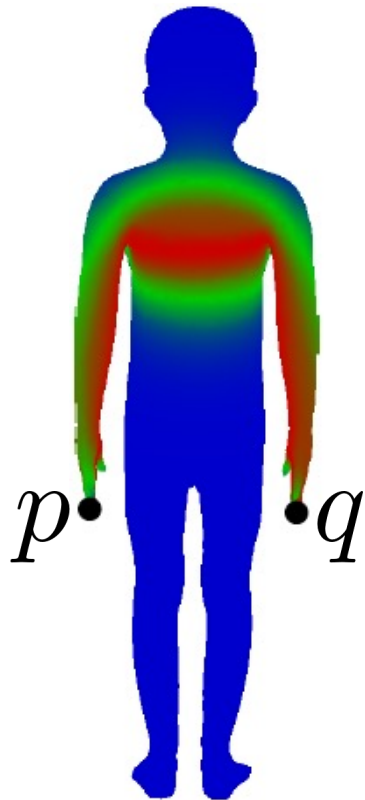
tance function over the edges, the implementation is actually practical even though, to our knowledge, it has never been done previously. We demonstrate that the algorithm’s worst case running time of $O(n^2 \log n)$ is pessimistic, and that in practice, the algorithm runs in sub-quadratic time. For 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 $O(n \log n)$ time even for small error thresholds.

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

Fuzzy Geodesics

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



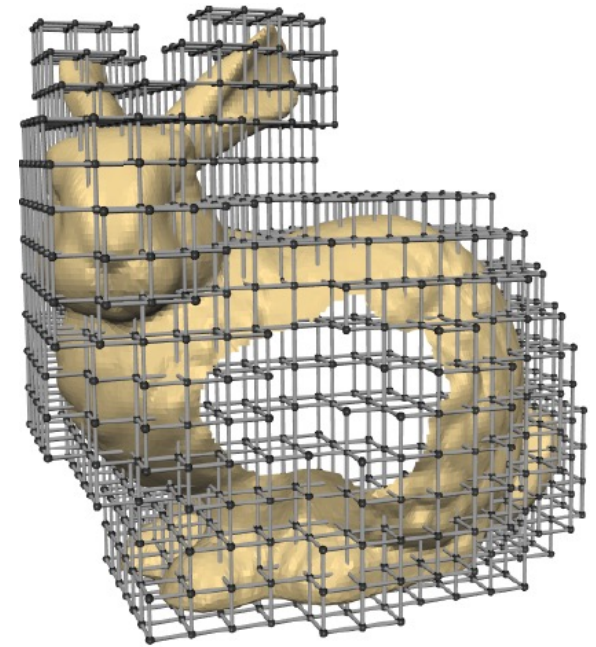
Function on surface
expressing difference
in triangle inequality

“Intersection” by
pointwise multiplication

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

Stable version of geodesics

Stable Measurement



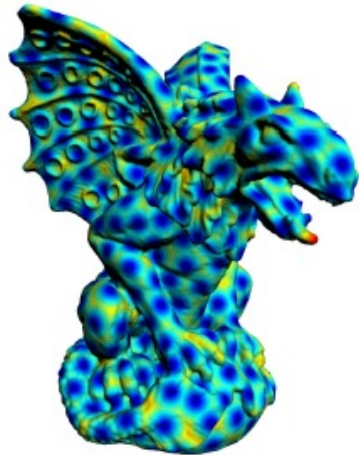
Morphological operators to fill holes rather than remeshing

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

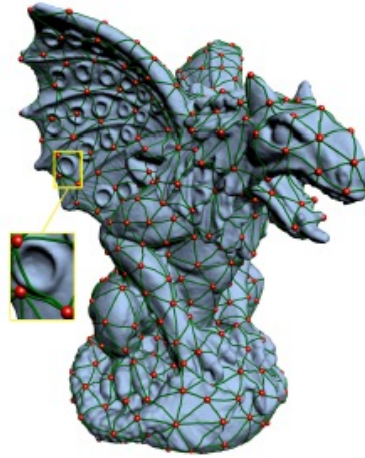
All-Pairs Distances



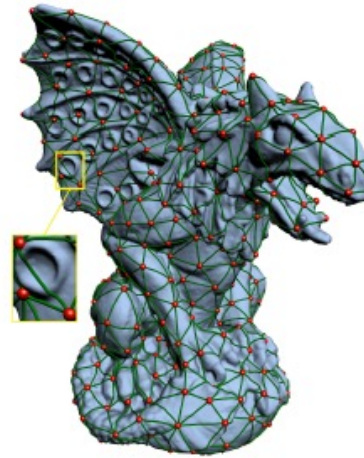
Sample
points



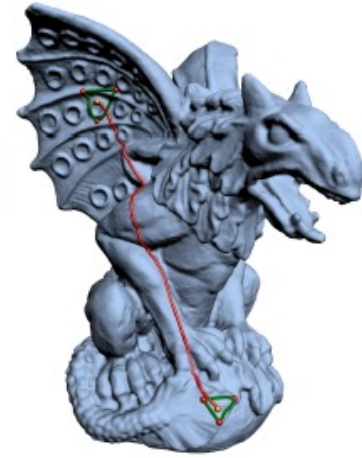
Geodesic
field



Triangulate
(Delaunay)



Fix edges



Query
(planar
embedding)

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

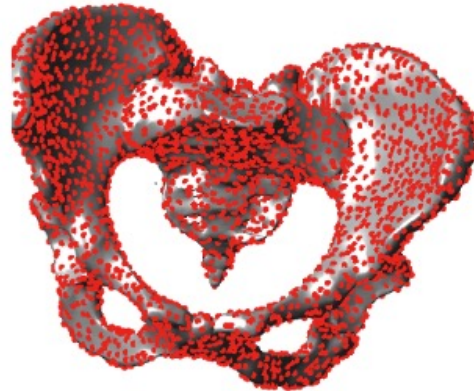
Geodesic Voronoi & Delaunay



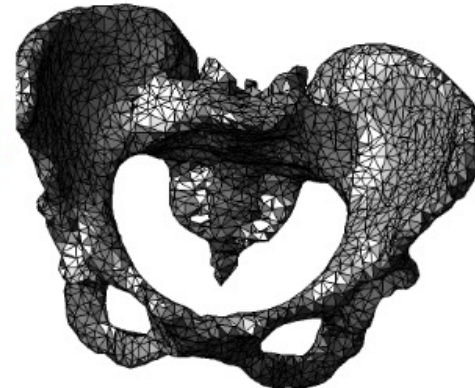
$N = 1000$ samples



Triangulation



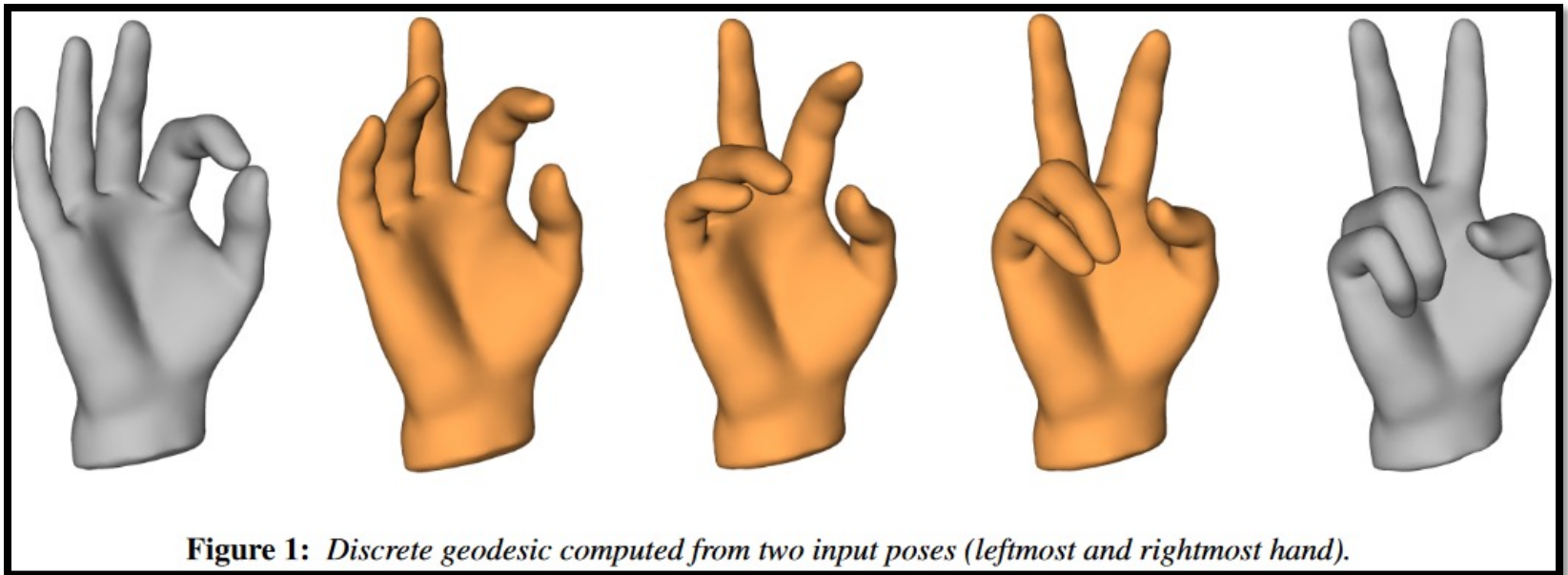
$N = 10000$ samples



Triangulation

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

High-Dimensional Problems



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 Alamgir¹

Ulrike von Luxburg^{1,2}

MORTEZA@TUEBINGEN.MPG.DE

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 show 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 & Orlicsky (2005) extend these results to ε -graphs from a general density p by introducing a weight function that depends on the distance between points.

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.

bitrary density p

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 not positive definite if the Riemannian manifold is not flat.

Heat kernel is PD!

we show that for spaces with conditionally negative definite distance, the geodesic Laplacian kernel can be generalized while retaining positive definiteness. In contrast, that geodesic Laplacian kernels can be generalized to curved spaces, including spheres and hyperbolic spaces. Our theoretical results are verified empirically.

Kernel	Extends to general	
	Metric spaces	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 $\lambda > 0$?

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.

and show the following results, summarized in Table 1.