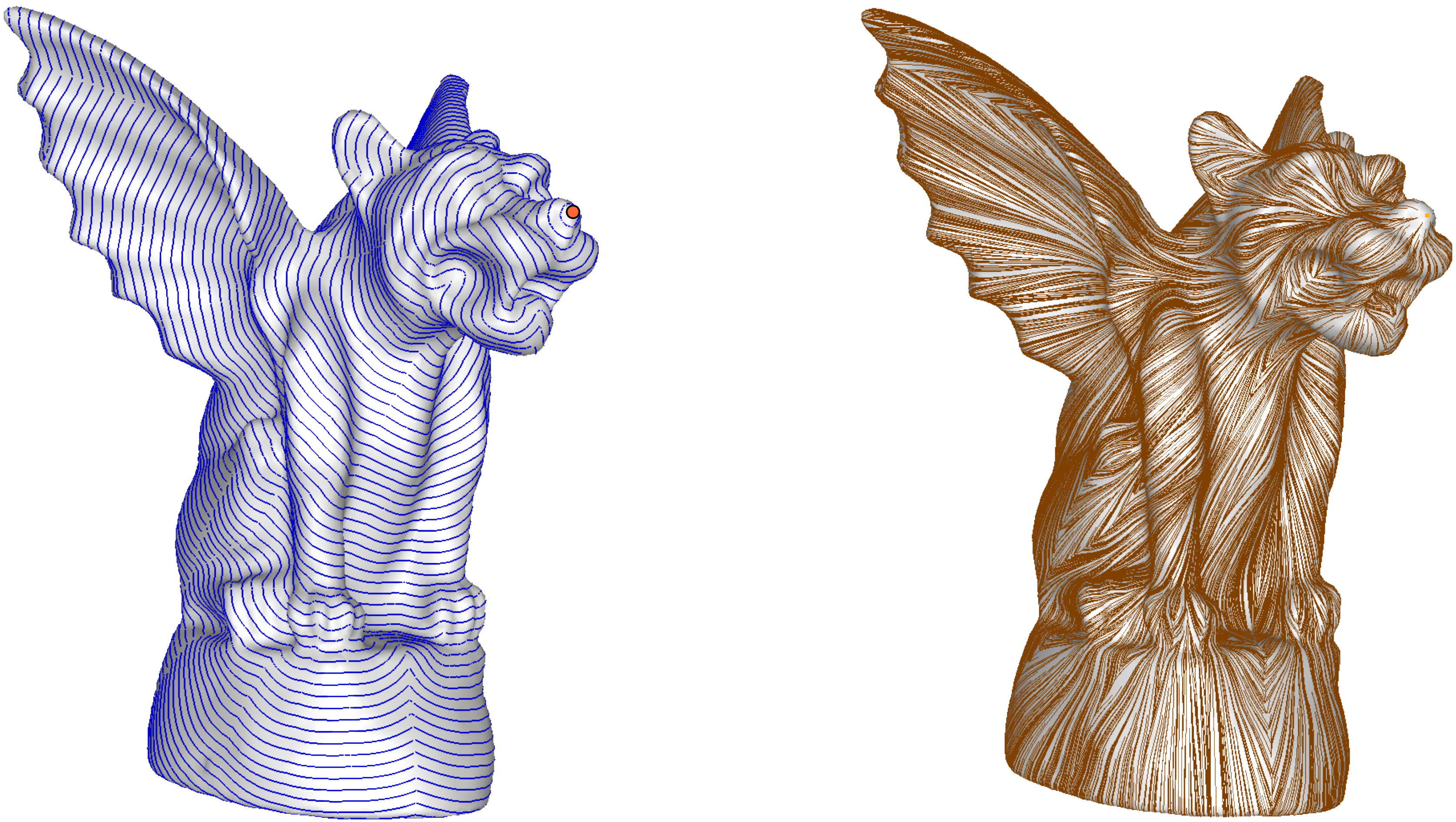
Surazhsky et al.



Exact and Approximate Shortest Paths

Siddhartha Chaudhuri http://www.cse.iitb.ac.in/~cs749

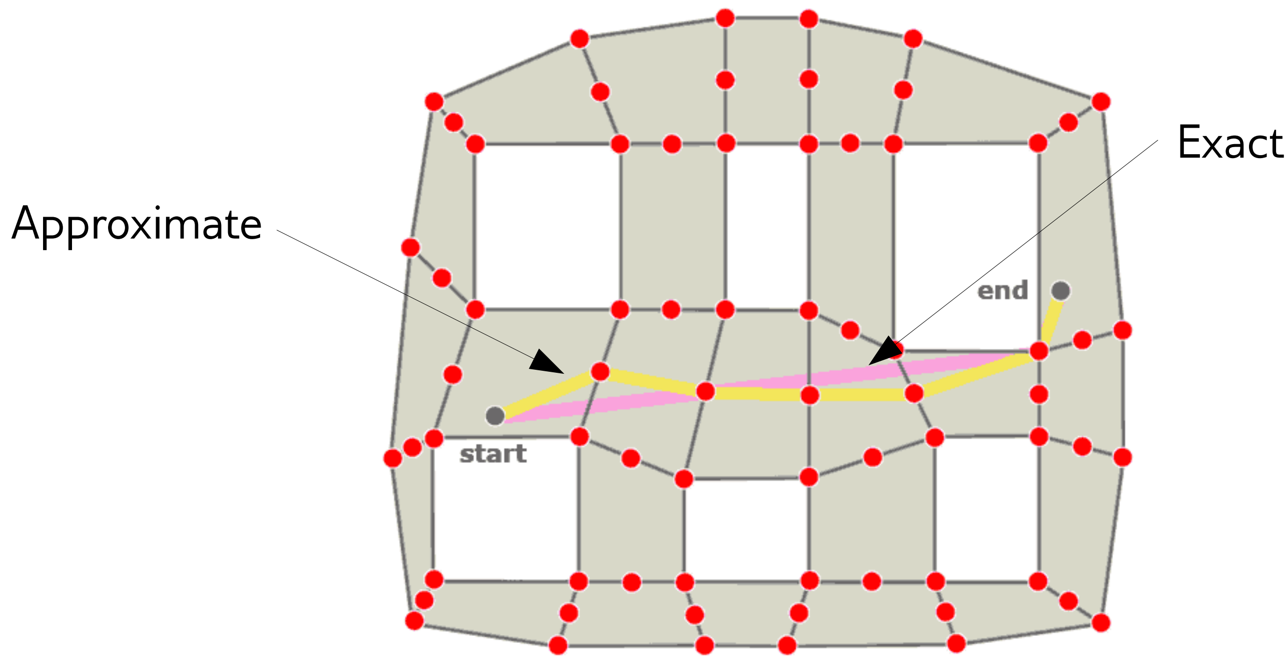
Overview

- Last class:
  - Distances on surfaces: applications and complications
  - Definition of geodesics

    $\neq$ shortest paths, but often conflated!
- Today:
  - An algorithm for finding exact shortest paths on a mesh
  - Can be modified to quickly give approximately shortest paths

Challenge

- The shortest path traverses **interiors** of triangles
  - Where does the path cross edges?
  - Continuous optimization problem, much harder than shortest paths on discrete graphs



A historical perspective

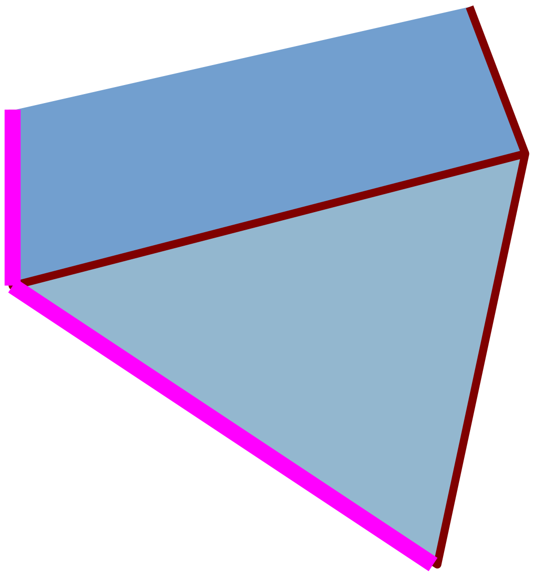
- Exact:
  - Single source, all destinations
    - Mitchell/Mount/Papadimitriou [1987]: $O(n^2 \log n)$
    - Chen/Han [1996]: $O(n^2)$
  - Single source, single destination
    - Kapoor [1999]: $O(n \log^2 n)$
- Approximate:
  - Insert extra edges: Lanthier [1997]
  - Iterative optimization: Kanai/Suzuki [2001], Martinez et al. [2004]
  - Fast-marching: Kimmel/Sethian [1998, $O(n \log n)$]
  - Fast-sweeping: Zhao [2005, $O(n)$]
  - Window merging: Surazhsky et al. [2005, $O(n \log n)$]
  - Heat flow: Crane et al. [2013]

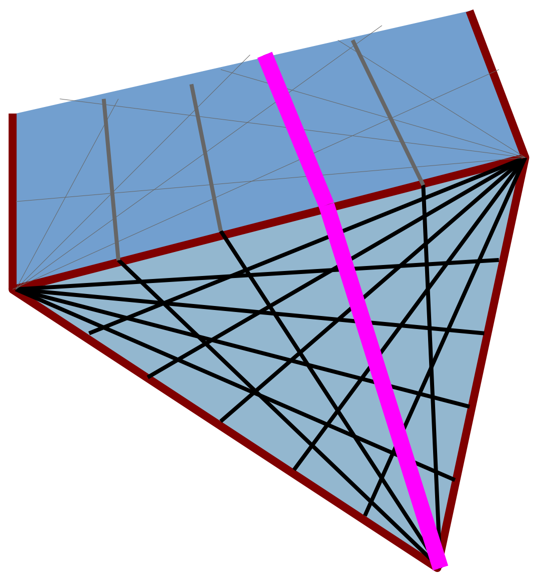
n = number of triangles in the mesh

Note that asymptotic behavior ≠ real-world performance

Edge Insertion

- **Problem:** Edges crudely model surface connectivity
- **Solution:** Add more edges!
  - Finer-grained discrete problem
  - Produces better approximations with more edges, at the cost of runtime complexity
    - But typically converges at around 6 points per edge

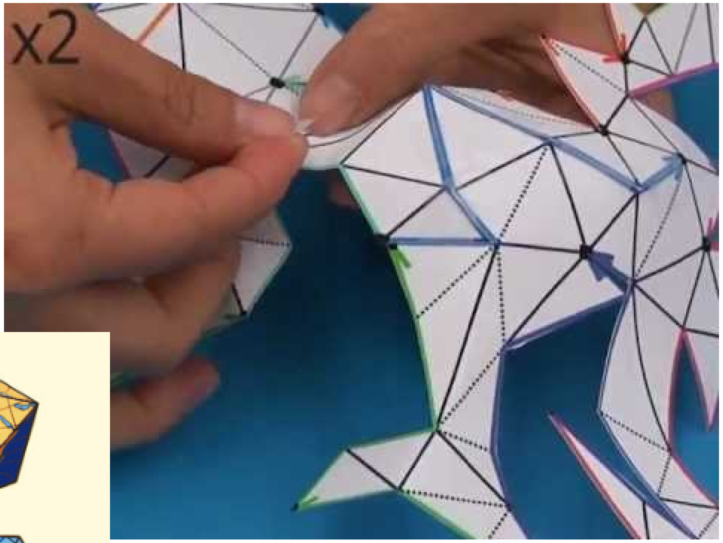


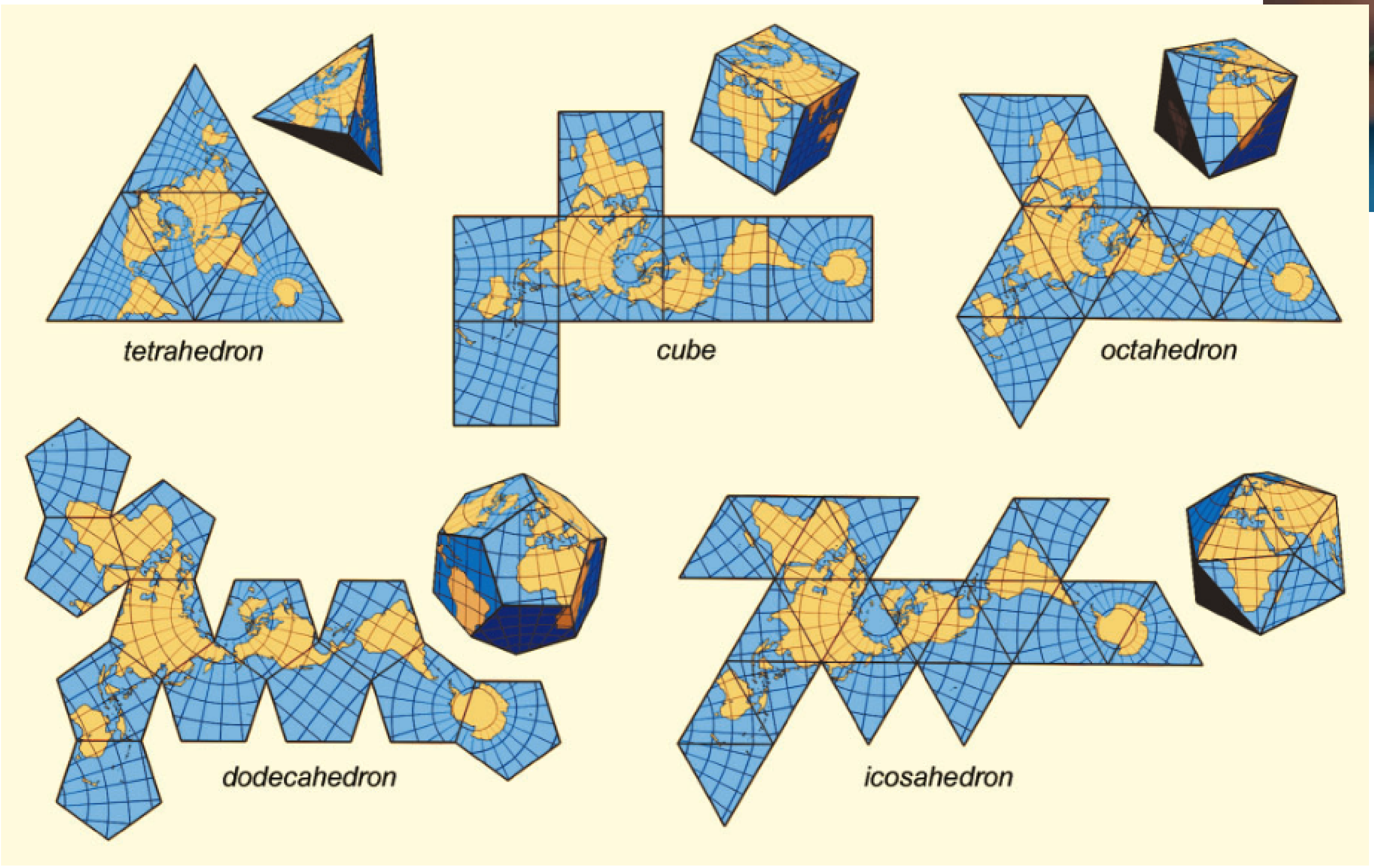


Lanthier 1997

A crucial observation 1/2

- A mesh can be unfolded

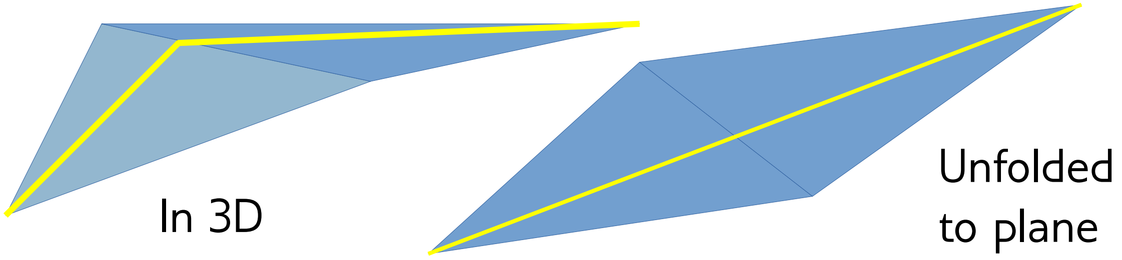


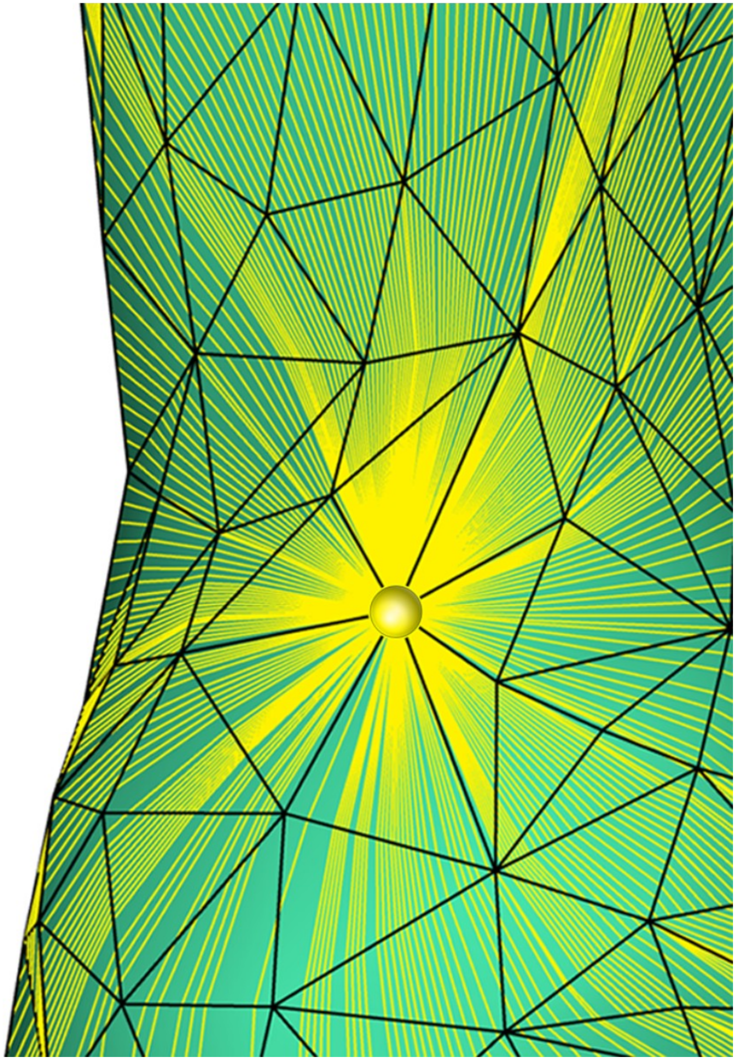


van Wijk 2008, Takahashi 2011

A crucial observation 2/2

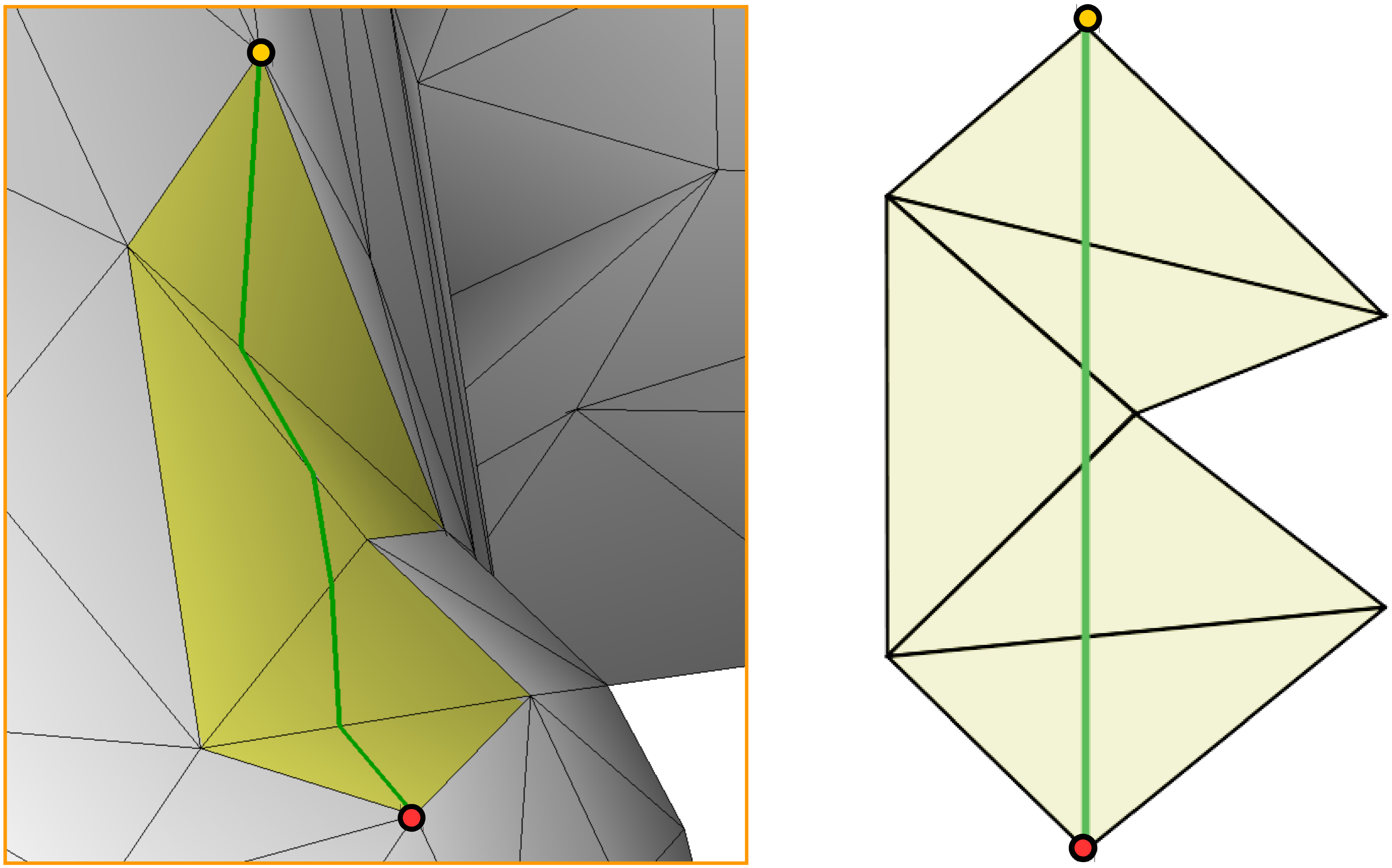
- Shortest paths can be visualized as rays emanating from point in all directions
  - Interior to triangle: shortest path must be straight line
  - Crossing edge: shortest path corresponds to straight line when two triangles are unfolded into common plane.





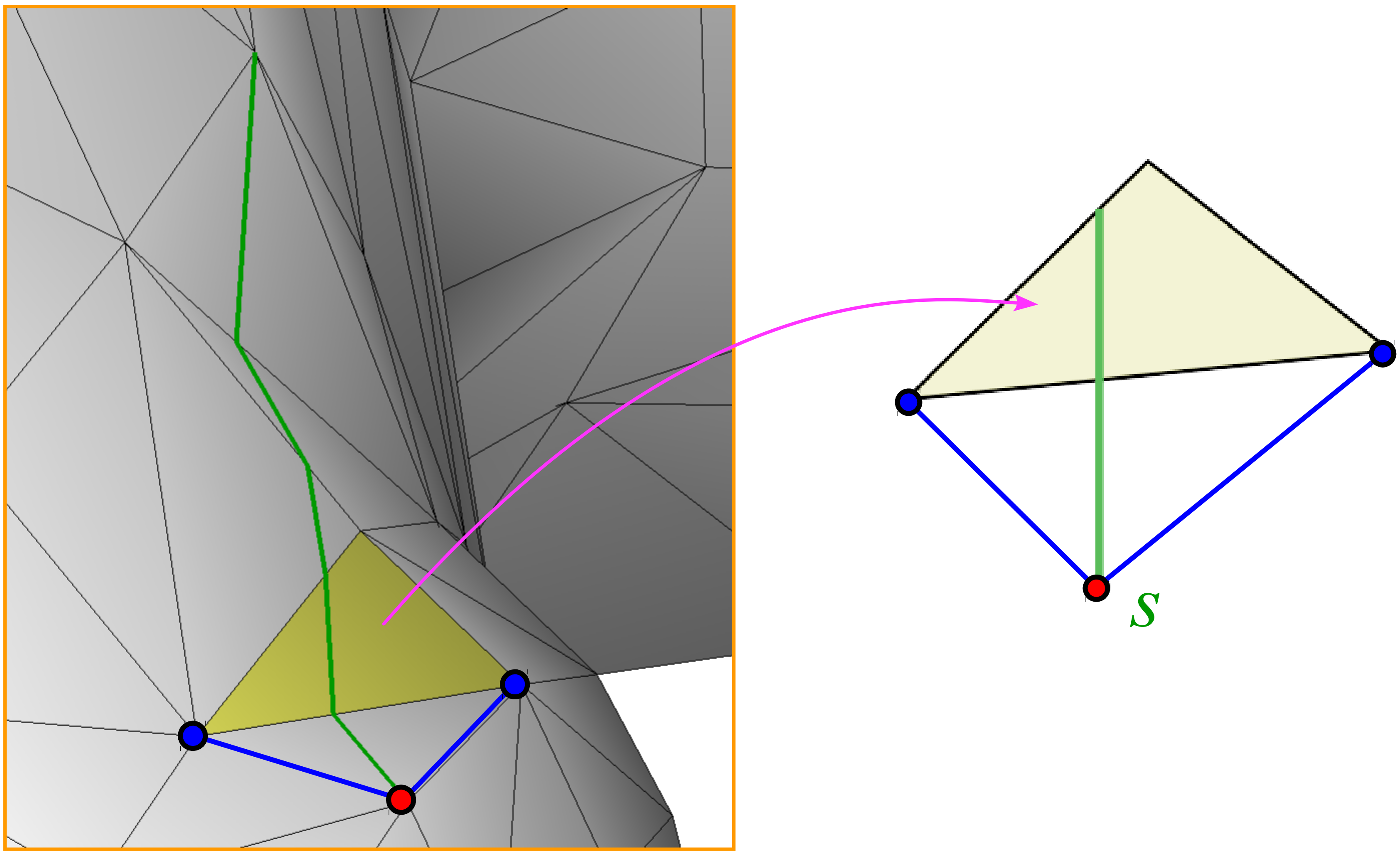
Surazhsky et al. 2005

Unfolding the shortest path



Surazhsky et al. 2005

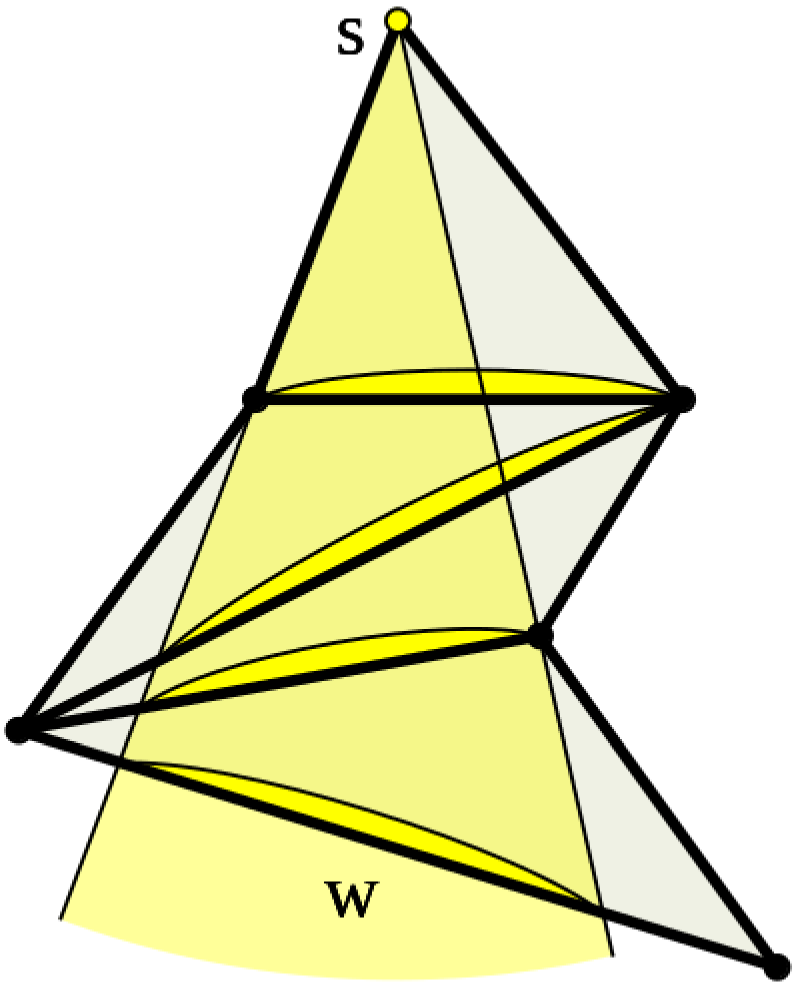
Source vertex unfolded to triangle plane



Surazhsky et al. 2005

Windows

- A **window** is a segment of an edge over which all shortest paths to the source traverse the **same sequence of faces**
- Within a window, distance computations can be performed ***atomically*** (no need to worry about routing in the mesh)

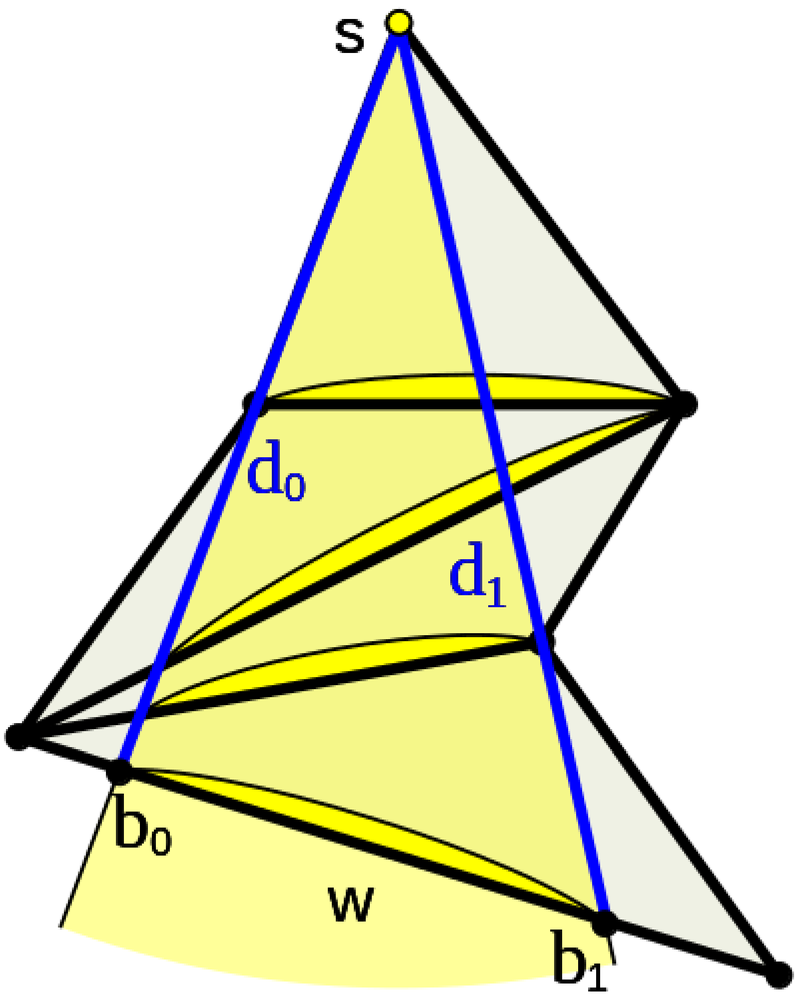


Unfolded layout

Surazhsky et al. 2005

Window specified by 5-tuple

- b_0, b_1: local x-coordinates of endpoints on edge
- d_0, d_1: distances from endpoints to source s
- τ : direction to source (side of edge where s lies)



Unfolded layout

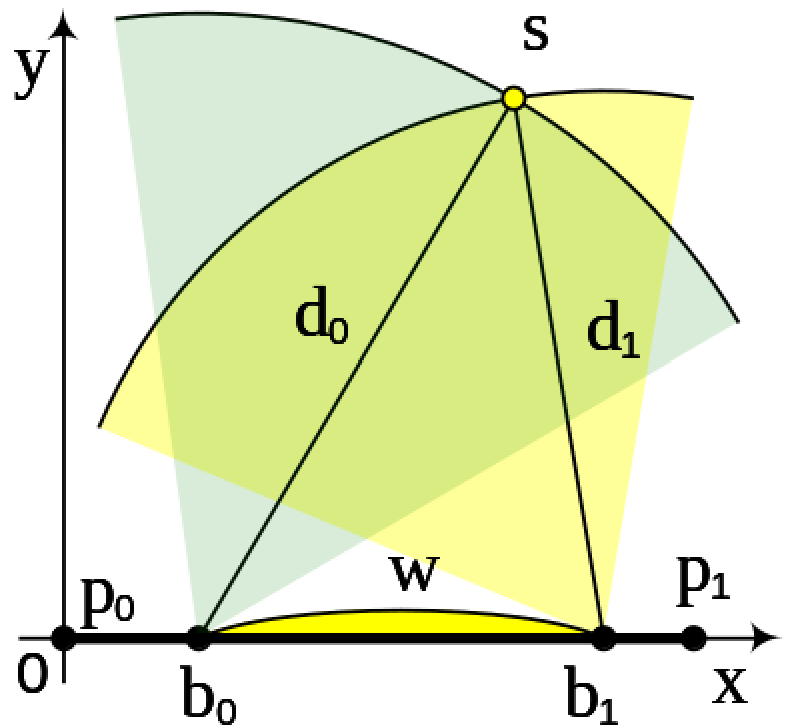
Surazhsky et al. 2005

Source reconstruction

- Given two distances d_0, d_1, recover the source $s = (x, y)$
- Computation is simple via local coordinate system

$$(x - b_0)^2 + y^2 = d_0^2$$
$$(x - b_1)^2 + y^2 = d_1^2$$

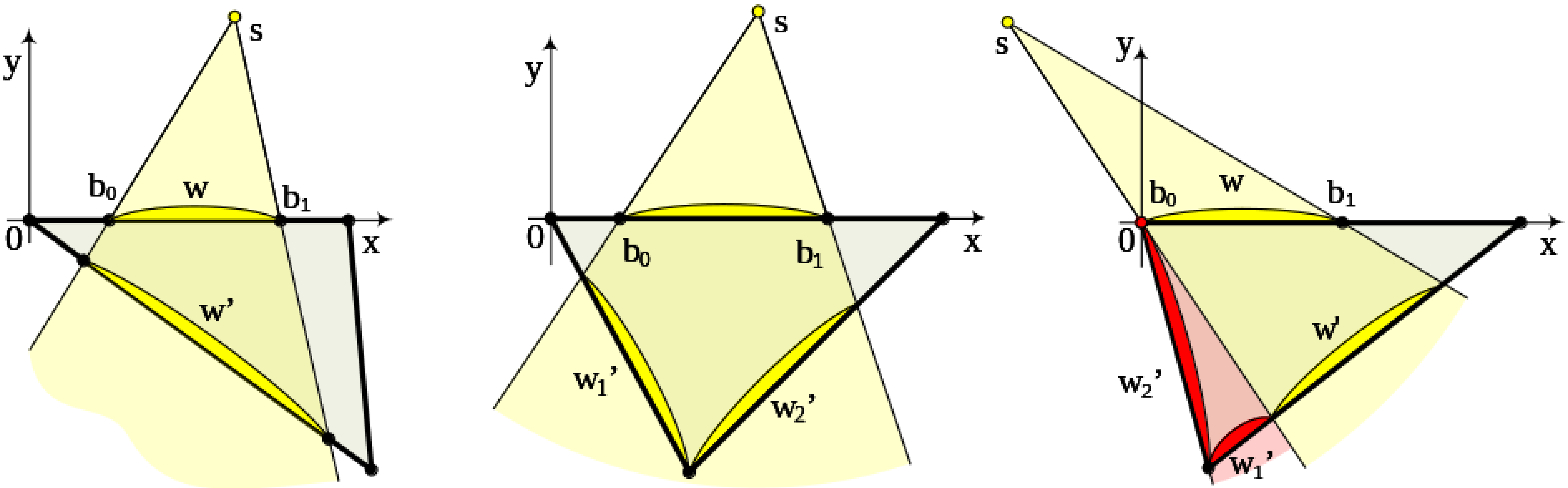
$$x = \frac{d_0^2 - d_1^2 - b_0^2 + b_1^2}{2(b_1 - b_0)}$$
$$y = +\sqrt{d_0^2 - (x - b_0)^2}$$



Surazhsky et al. 2005

Basic Idea: Window Propagation

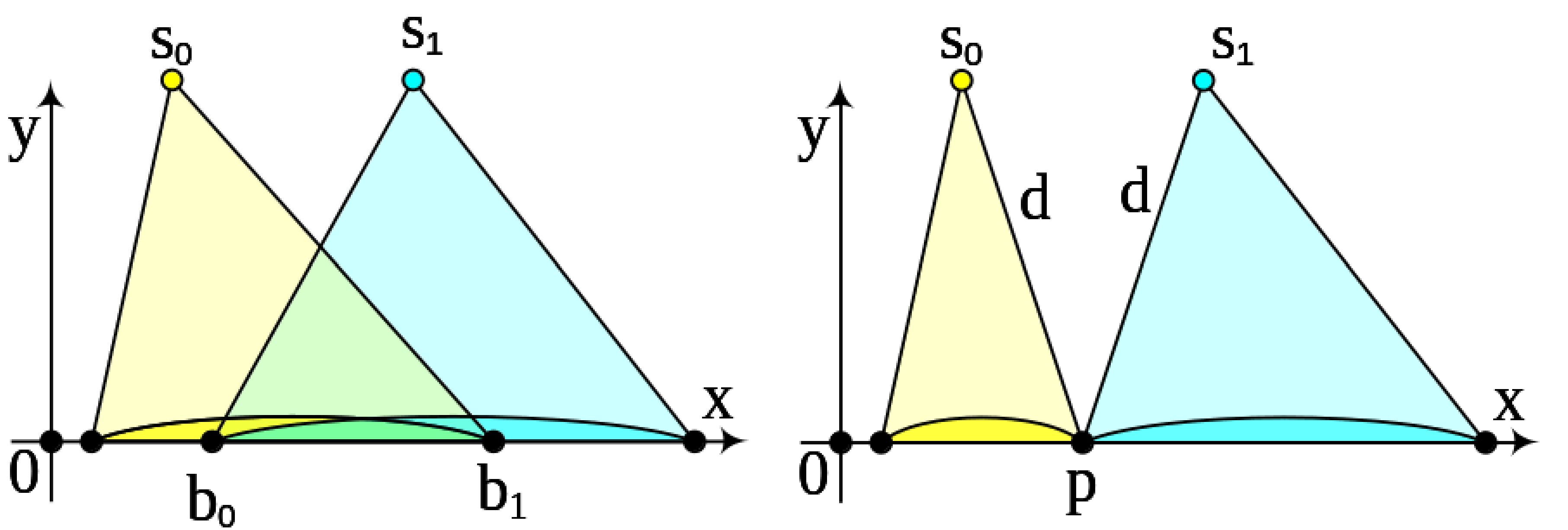
- Step from triangle to adjacent triangle
- Windows on an edge create new windows on other edges of new triangle
  - The cone “sees” new edges as we enter the new triangle
- Can create one, two or three new windows



Surazhsky et al. 2005

Overlapping Windows

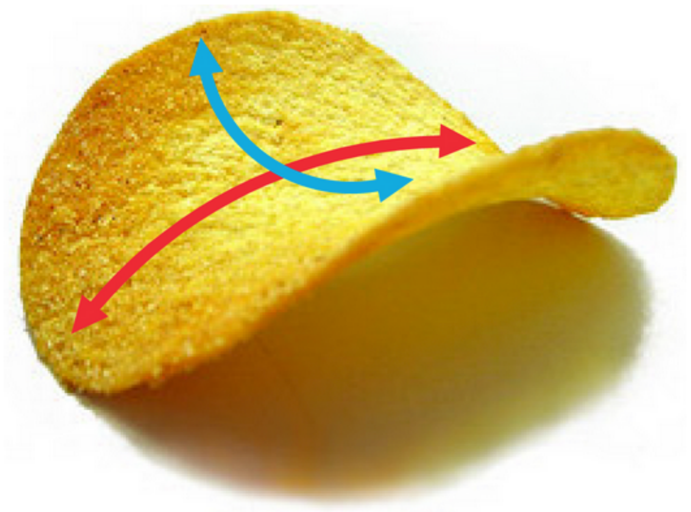
- Find equidistant point on edge
- Cut off overlapping parts that define larger distances
- Distance function is continuous on edge



Surazhsky et al. 2005

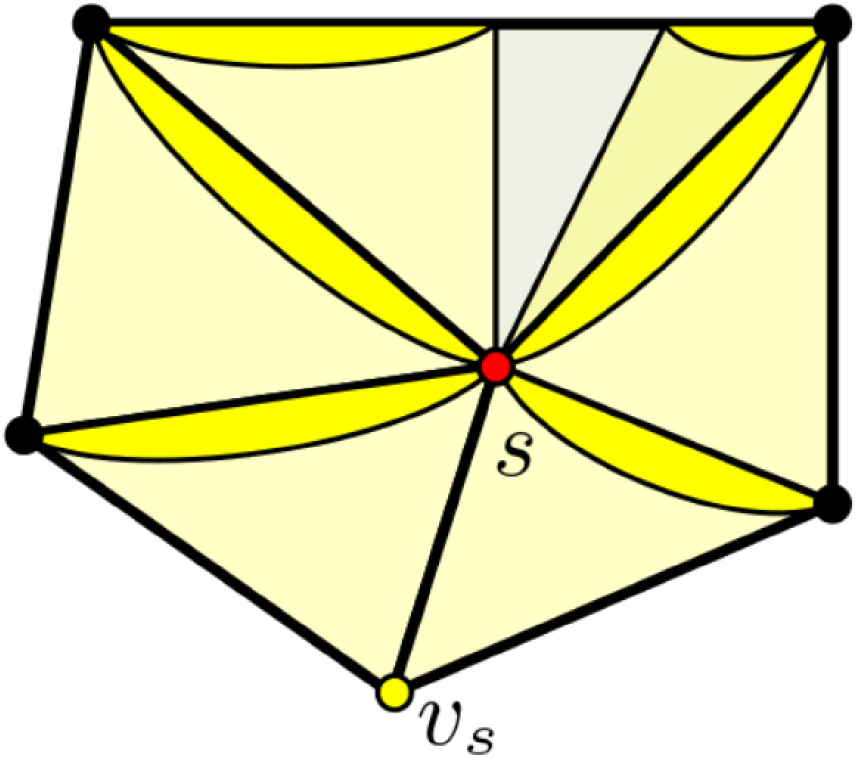
Hyperbolic (saddle) vertices

- If the sum of face angles at a vertex is $> 2\pi$, it is called a hyperbolic/saddle vertex
- It cannot be unfolded onto a plane without foldovers (overlapping faces)
- The shortest path can pass through boundary, hyperbolic ($> 2\pi$) and parabolic ($= 2\pi$) vertices
  - Hyperbolic vertices need special handling

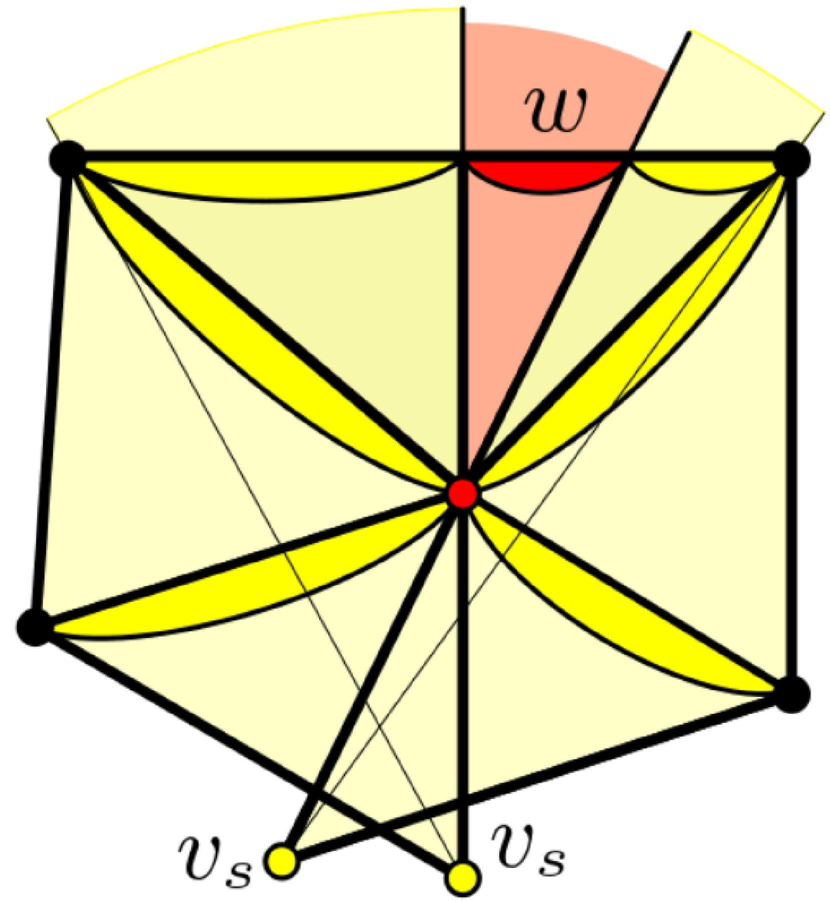


Stephanie K. Fleming

Hyperbolic (saddle) vertices



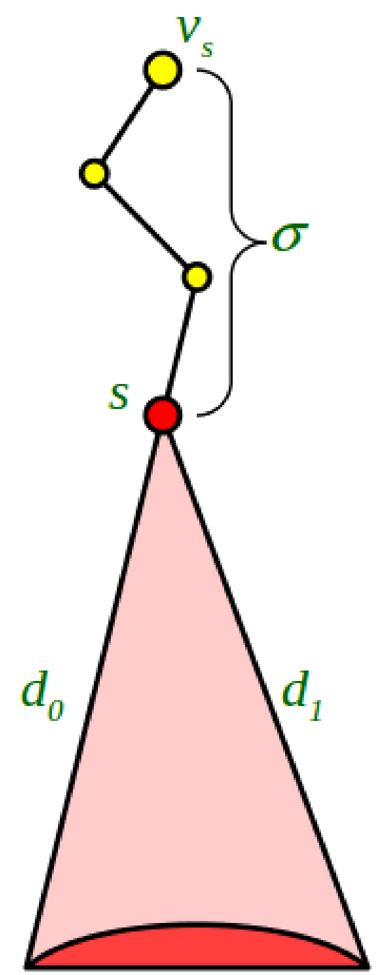
Unflattened region near saddle vertex s. Part of the upper triangle is not visible by rays from source vertex v_s



Unfolding to plane of upper triangle reveals two different images of v_s, neither of which is visible from red region. All shortest paths to w pass through saddle vertex.

Hyperbolic (saddle) vertices

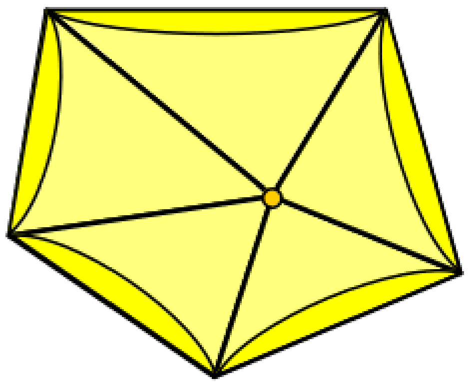
- **Solution:** Treat saddle (or boundary) vertex as **pseudo-source**
  - All shortest paths route through pseudo-source
  - ... so we can originate our visibility cones from the pseudo-source, and add its distance from the actual source
- Window as 6-tuple: $\{ b_0, b_1, d_0, d_1, \sigma, \tau \}$
  - σ : distance from **pseudo-source** to source

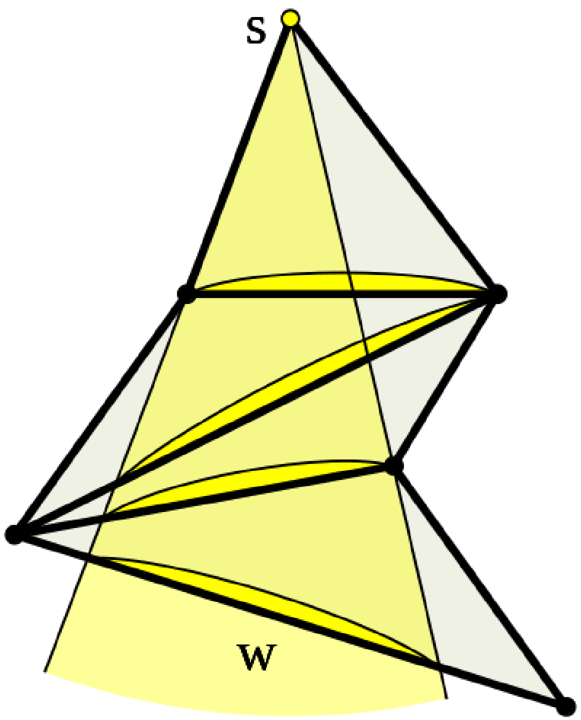


Surazhsky et al. 2005

The algorithm

- Initialize queue Q with a window for each edge adjacent to source s, sorted by distance to source
- Until Q is empty
  - **select** (and remove) a window from Q
  - **propagate** selected window
  - **update** Q with new windows
- The algorithm fully covers each edge with non-overlapping windows





Surazhsky et al. 2005

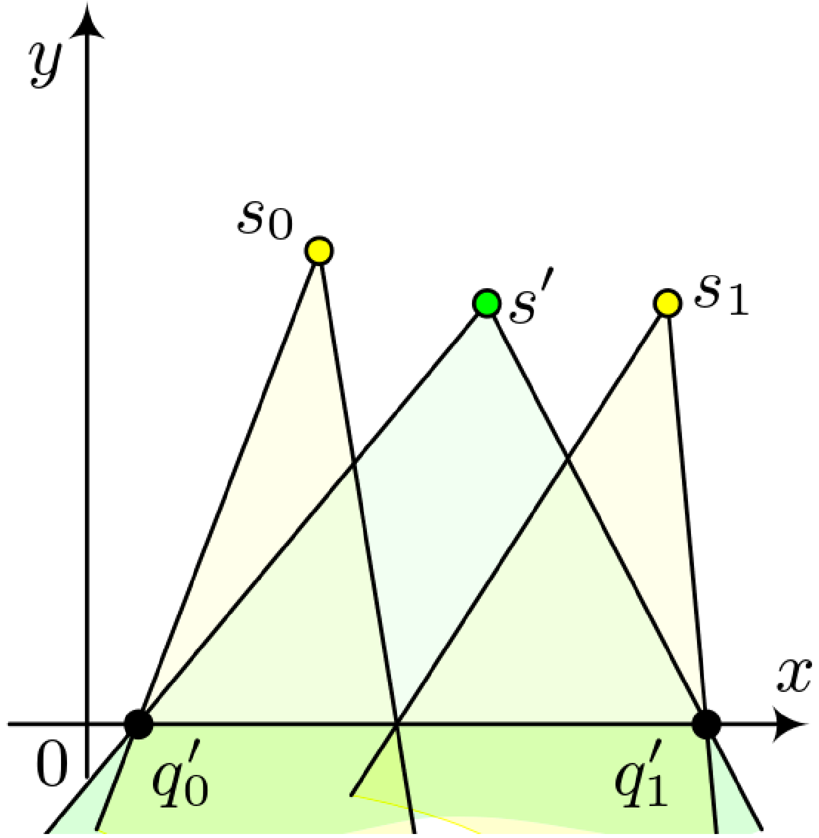
Approximating Algorithm

- Basic idea: merge windows
  - Two original windows must have **direction** values τ in agreement
  - Original windows must define **similar distances** on their union
  - Distance function along edge must be **continuous**
  - **Visibility** region of new window must cover the union of visibility regions of original windows

Surazhsky et al. 2005

Merging Windows

- Find a new source $s = (x, y)$ and σ, for which the distances at the endpoints b_0 and b_1 are preserved:



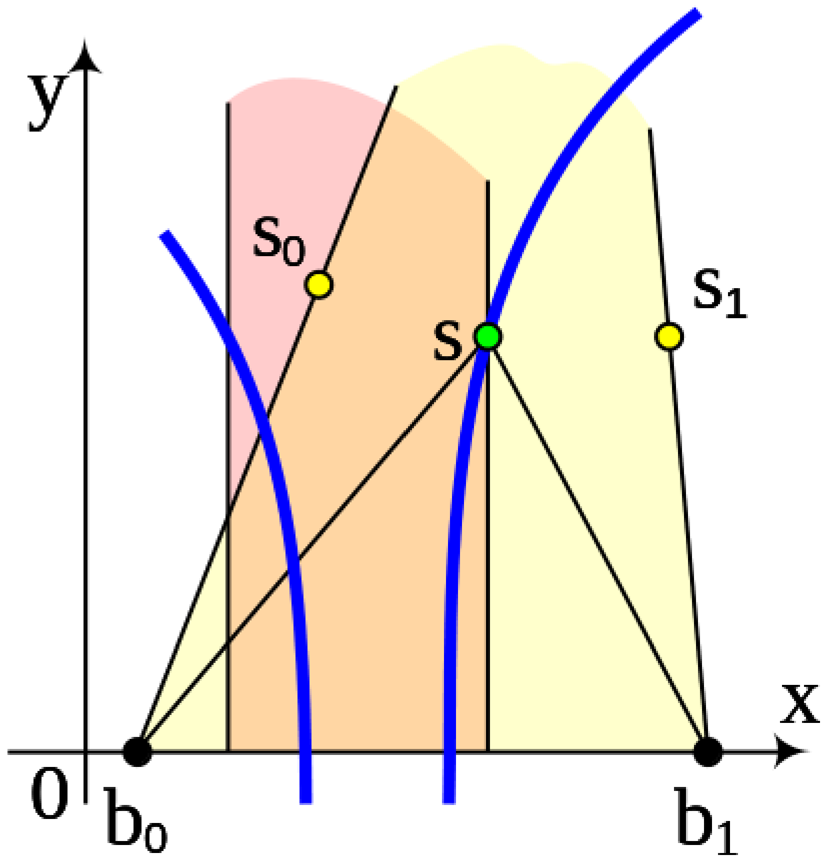
Surazhsky et al. 2005

Merging Windows

- Find a new source $s = (x, y)$ and σ, for which the distances at the endpoints b_0 and b_1 are preserved:

$$(x - b_0)^2 + y^2 = (d_0 - \sigma)^2$$
$$(x - b_1)^2 + y^2 = (d_1 - \sigma)^2$$

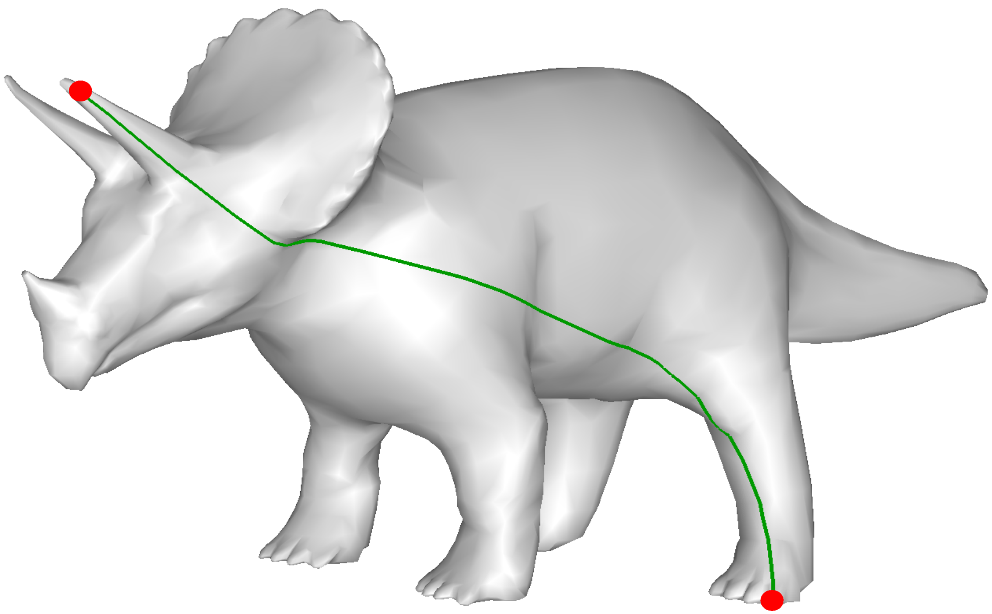
- $s = (x, y)$ lies on a conic curve (quadratic algebraic curve)
- $s = (x, y)$ must lie in the yellow area – visibility must not be reduced) with $y > 0$
- $\sigma > 0$ corresponds to the pink area



Surazhsky et al. 2005

Shortest path between two vertices

- Sequence of pruned searches to locate exact shortest path
- Exact algorithm is invoked only on a thin region surrounding geodesic
- Upper bound is the length of the approximate path obtained by Djikstra search on edge graph, refined by output of approximation algorithm
- Lower bound initially represented by Euclidean distance, then replaced with output of approximation algorithm



Surazhsky et al. 2005