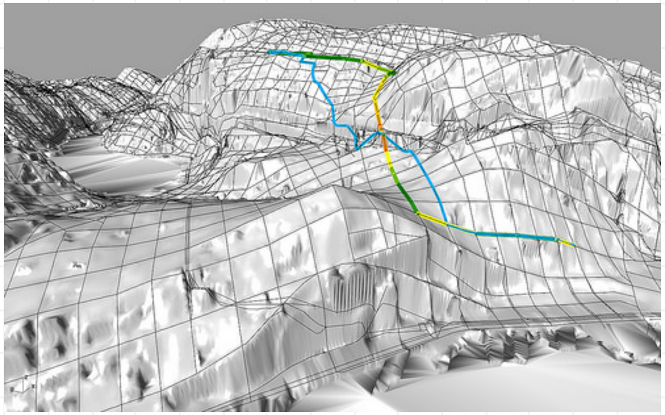
shortest path on a polyhedral surface

generalizes polygonal domain

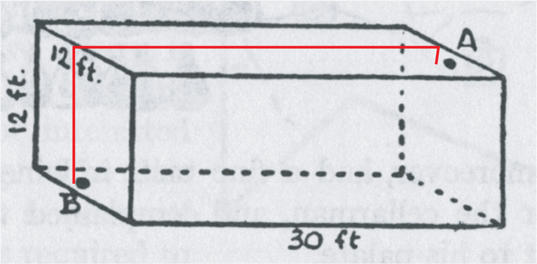


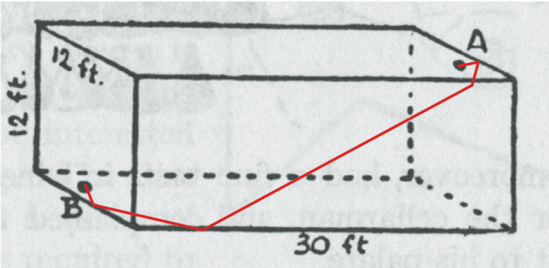
shortest paths follow straight lines in *unfolding*

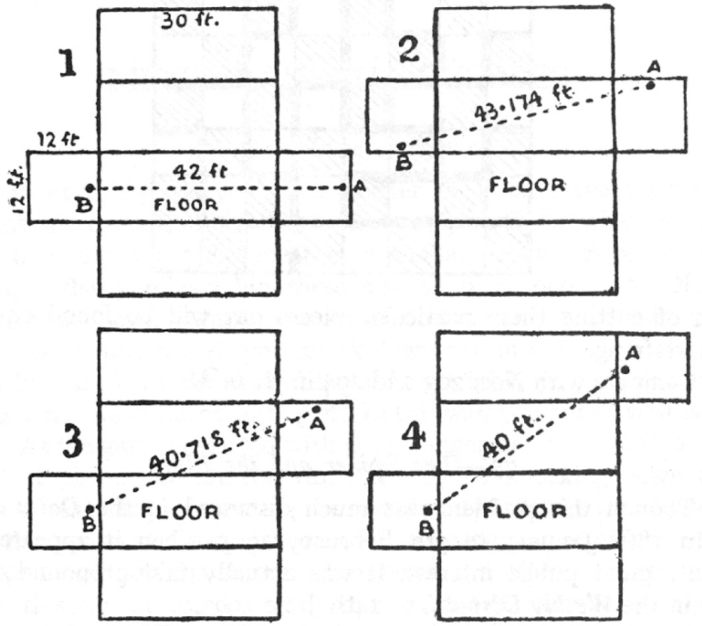
the spider and the fly problem

Dudeney, The Canterbury Puzzles, 1958

?







Chen and Han. 1996.

[**Shortest paths** on a **polyhedron**, **Part** I: Computing **shortest paths**](http://scholar.google.ca/scholar?q=Shortest+paths+on+a+polyhedron%2C+Part+I%3A&btnG=&hl=en&as_sdt=0%2C5)
J Chen, Y Han - International Journal of Computational Geometry & …, 1996 - World Scientific
We present an algorithm for determining the **shortest path** between any two points along the surface of a **polyhedron** which need not be convex. This algorithm also computes for any source point on the surface of a **polyhedron** the inward layout and the subdivision of the **...**
Cited by 115Related articlesAll 4 versionsCiteSave

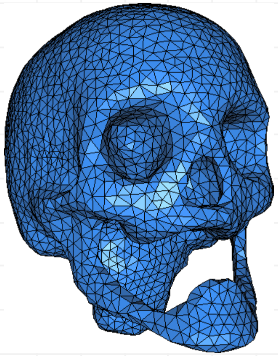
From: http://scholar.google.ca/scholar?q=Shortest+paths+on+a+polyhedron%2C+Part+I%3A&btnG=&hl=en&as_sdt=0%2C5

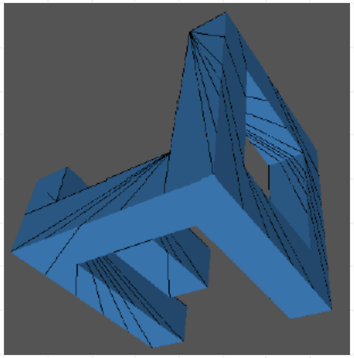
Shortest paths on a polyhedron surface in O(n^2) time, O(n) space.
Solves single-source version, query target point t. Builds shortest path map.
O(log n) query O(log n + output)

Input: polyhedral surface — triangles in 3-space, joined edge-to-edge (every edge in 2 triangles).
n = # triangles.

Before: continuous Dijkstra O(n^2 log n). Mitchell, Mount, Papadimitriou

LEDA

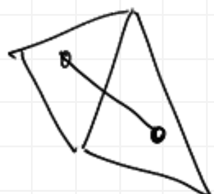

Rineau and Yvinec


Kaneva O'Rourke

Chen and Han. Shortest paths on polyhedron surface.

Start with convex polyhedron (then no shortest path goes through a vertex because a taut string won't).
Fixed t — vertex.

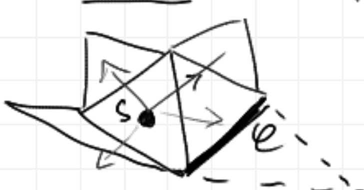
Claim 1. If you unfold two adjacent Δ's to the plane, any shortest path crosses in a straight line



Claim 2. A shortest path does not enter a face (triangle) twice.

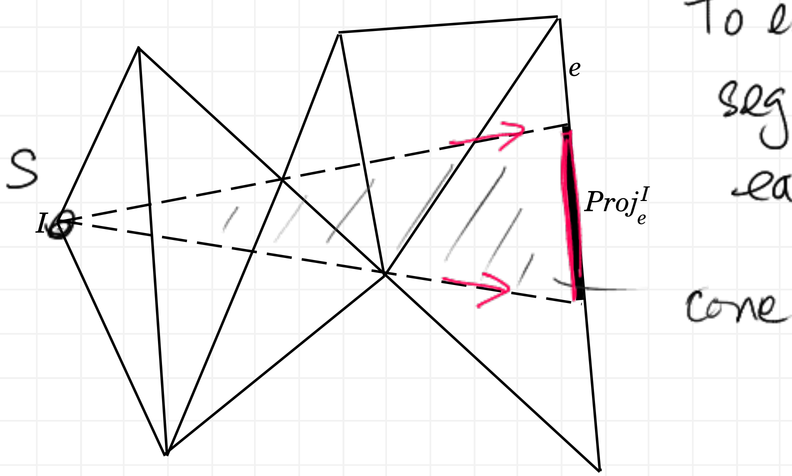
Claim 3. Two shortest paths do not intersect (except at source/target).

<u>Idea</u> Unfold from source point s.

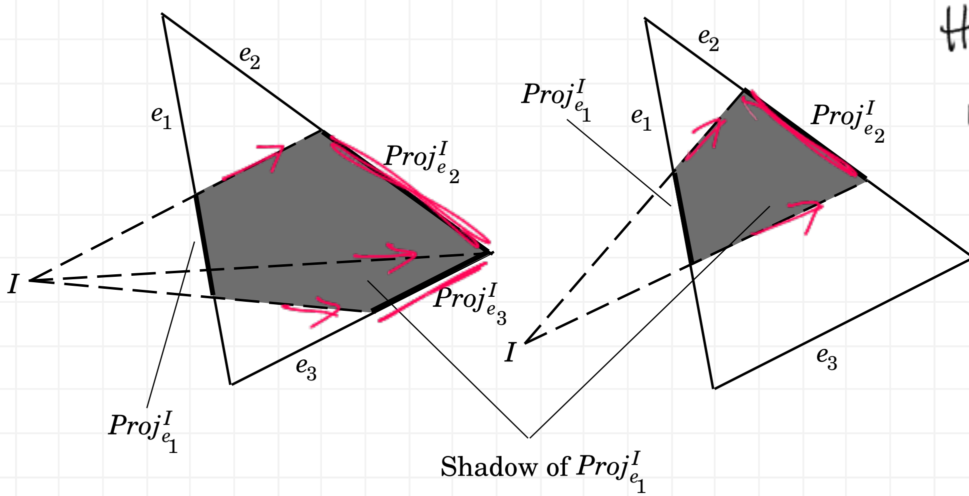


at edge e there is a unique "next" triangle.

Chen and Han. Shortest paths on polyhedron surface.



To expand past edge e, keep segments on e reached by each "cone" from s.

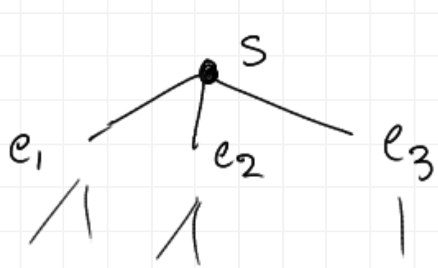


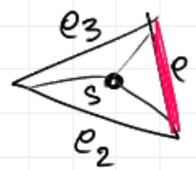
How a cone expands into next triangle

Keep track of segments and rays to endpoints

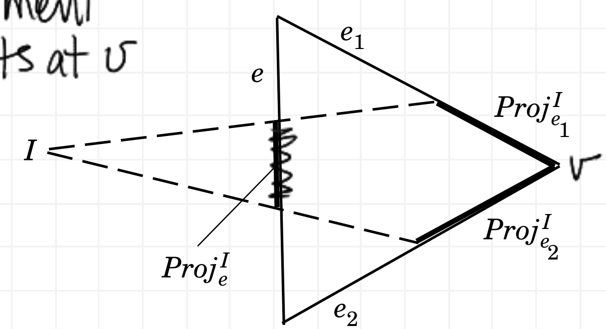
Chen and Han. Shortest paths on polyhedron surface.

Build segment tree.





segment splits at v



Two children, $(e_1, I, Proj^I_{e_1})$ and $(e_2, I, Proj^I_{e_2})$

Nodes are segments.

Node has 1 or 2 children.

Lemma After depth n the tree captures all shortest paths.

Pf. A shortest path goes through at most n faces (since we don't repeat faces)

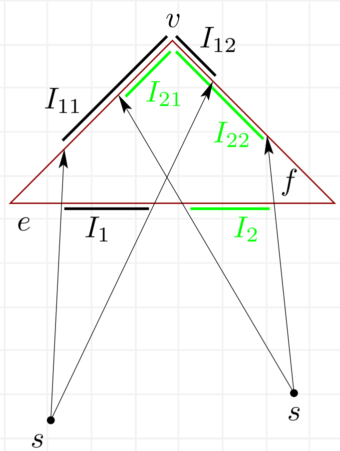
Compare all path $s \to t$ in tree for shortest.

ok but this is exponentially big

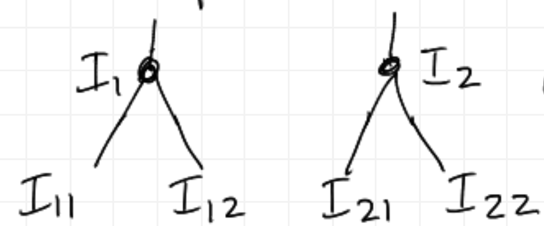
Chen and Han. Shortest paths on polyhedron surface.

Pruning the Tree

Lemma ("one-vertex one-cut")



Suppose 2 segments I_1, I_2 both split at vertex v



in tree

Then we can trim one of the 4 children.

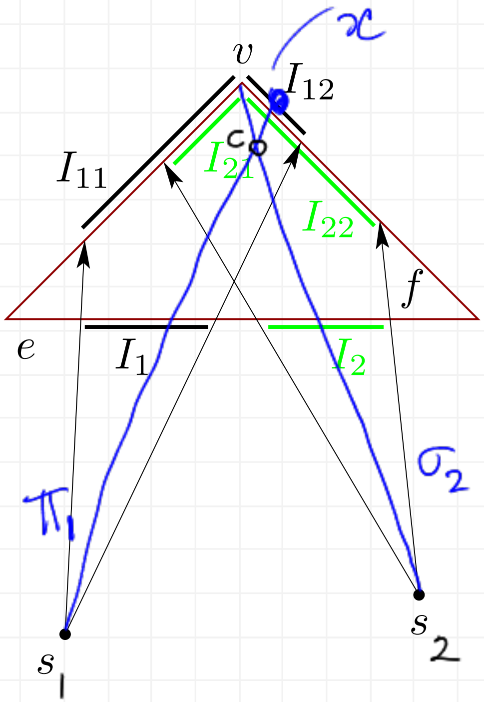
Pf. σ_1 = shortest path $s \to v$ in cone through I_1

σ_2 - - - $s \to v$ - - I_2

Suppose $|\sigma_2| \leq |\sigma_1|$

Claim I_{12} is never shortest.

Chen and Han. Shortest paths on polyhedron surface.



$|\sigma_2| < |\sigma_1|$

will show shorter path

Claim 1 I_{12} is useless

Pf Consider path π_1 $s_1 \to x$ in I_{12}

Crosses σ_2 at c

notation $\sigma(a,b)$ – subpath $a \to b$

Claim 2 $|\pi_1(s_1, c)| \geq |\sigma_2(s_2, c)|$

Thus $|\pi_1| \geq \underbrace{|\sigma_2(s_2,c)| + |\pi_1(c,x)|}$

so this is better than π_1 and I_{12} is useless.

Pf. of claim 2

$$|\pi_1(s_1, c)| + \cancel{|\sigma_2(c, v)|} \geq |\sigma_1| \geq |\sigma_2|$$

$$= |\sigma_2(s_2, c)| + \cancel{|\sigma_2(c, v)|}$$

Chen and Han. Shortest paths on polyhedron surface.

Size of segment tree:

$O(n)$ leaves because each $\underbrace{\text{vertex } v \text{ in triangle } T}_{O(n)}$ contributes one "branch"

height $O(n)$ (no path goes through $> n$ triangles)

Thus $O(n^2)$ size of tree. Time $O(n^2)$ to construct level by level (each new level has $O(n)$ children)

Further reduce space

compress chain of single vertices (= cone through successive Δ's)

binary tree w/ $O(n)$ leaves has $O(n)$ nodes

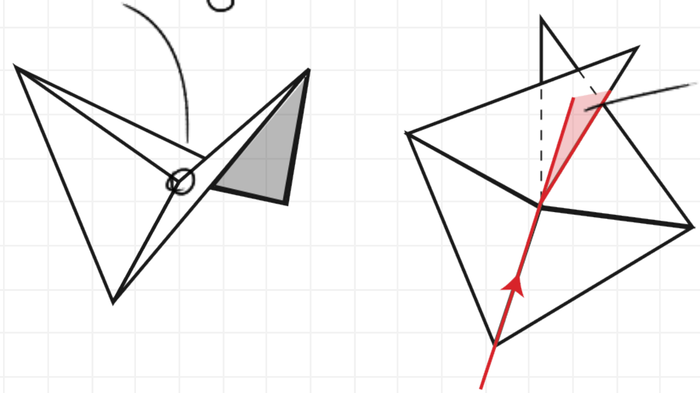
Time $O(n^2)$

Chen and Han. Shortest paths on polyhedron surface.

Dealing with non-convex vertices. — actually negative curvature vertices.

Shortest paths may go through these vertices (think saddle-points vs. mountain tops).

vertex of negative curvature

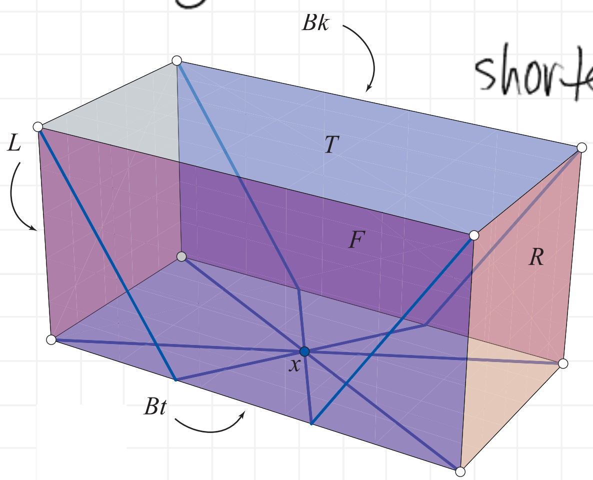


range of locally shortest paths

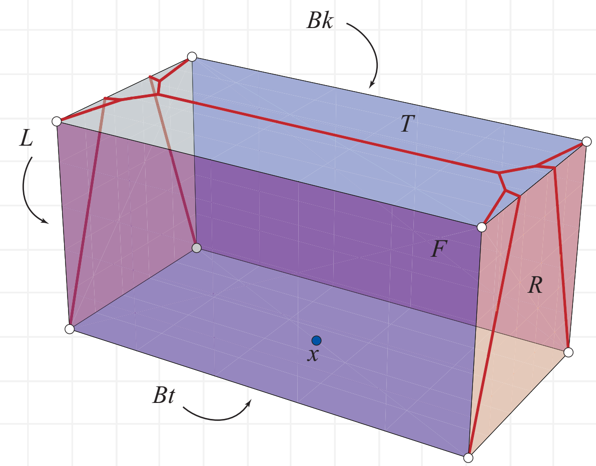
vertex is treated as a "pseudo-source"

Chen and Han. Shortest paths on polyhedron surface.

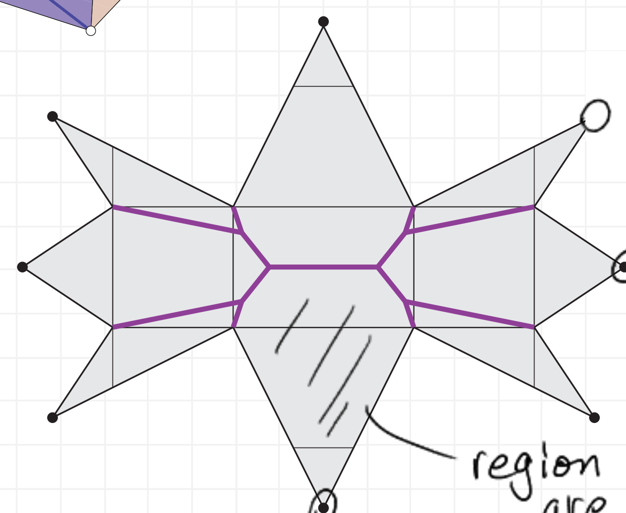
Building shortest path map for querying t



shortest paths to vertices



Can be done.
Uses Voronoi diagram on unfolding.



— copies of x

region where shortest paths are combinatorially the same.

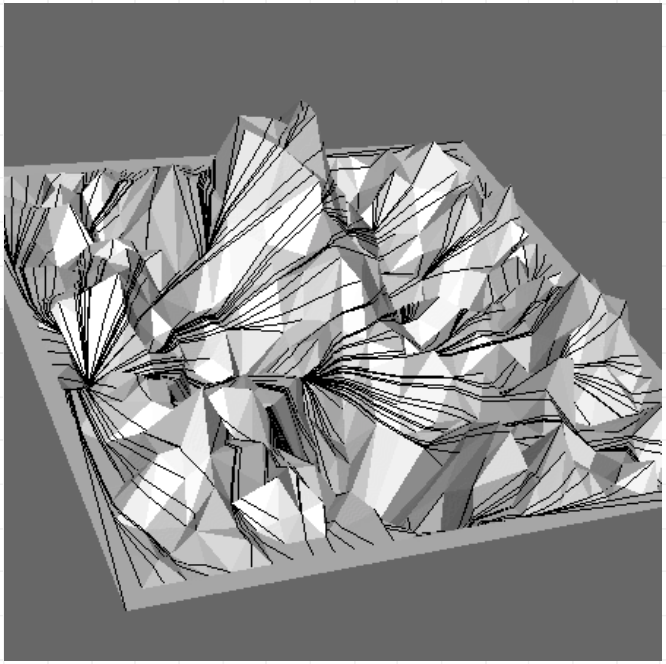
Chen and Han. Shortest paths on polyhedron surface.

Implementation

[PS] An implementation of Chen & Han's shortest paths algorithm
B Kaneva, J O'Rourke - 2000 - cs.smith.edu
Abstract In 1990 **Chen** and **Han** proposed a quadratic algorithm for nding the shortest paths from one source point to all vertices on a polyhedral surface. We report on a C++ **implementation** of their algorithm, to our knowledge the rst publicly available **...**
Cited by 58Related articlesAll 5 versionsCiteSaveMore

From: http://scholar.google.ca/scholar?hl=en&q=an+implementation+of+chen+and+han%27s+&btnG=&as_sdt=1%2C5&as_sdtp=

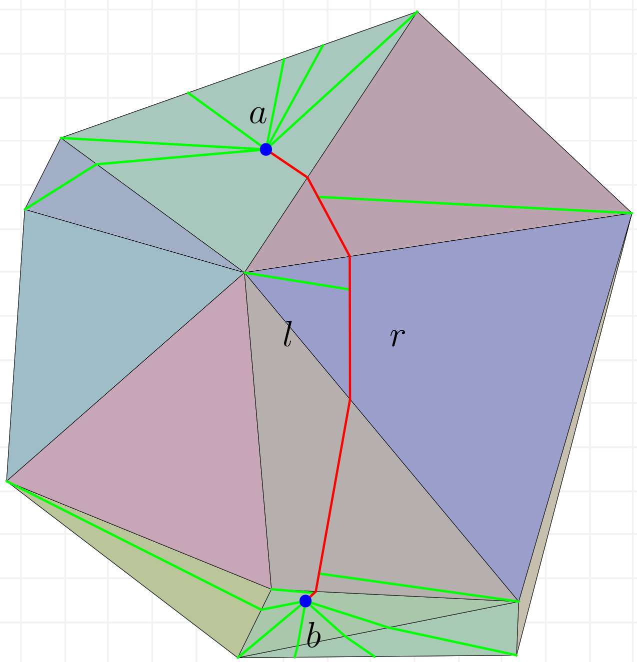
main numerical issue:
which of two paths to a vertex v is shorter.



3 extensions of Chen and Han

1. with Stephen Kiazyk

source s is a segment, not just a point



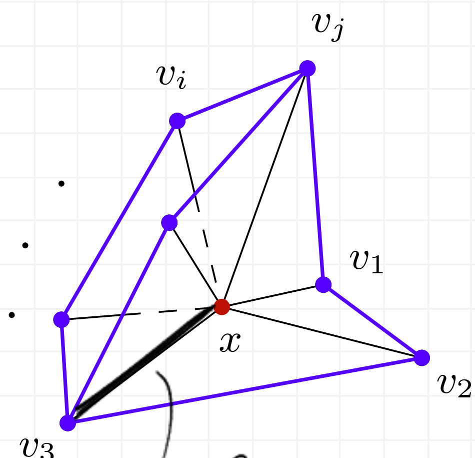
3 extensions of Chen and Han

2. with Daniela Maftuleac and Megan Owen

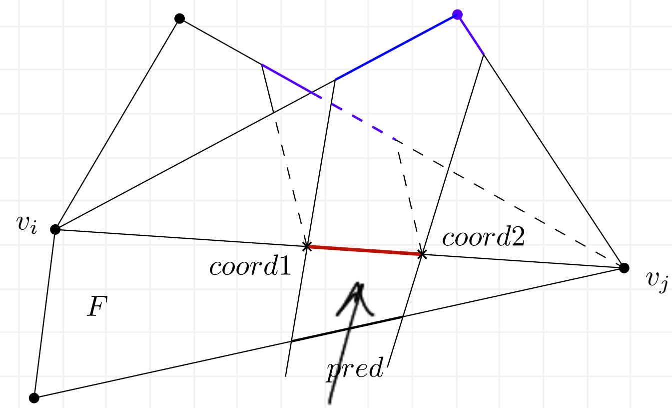
$O(n^2)$ time $O(n)$ space

shortest paths in 2D CAT(0) complexes

a curvature constraint s.t. locally shortest paths are unique.



can have > 2 triangles attached to an edge.



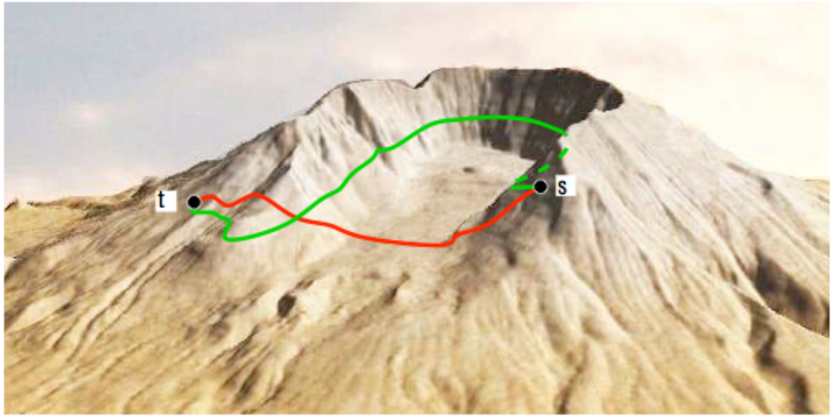
how to expand a segment into 2 "next" triangles.

3 extensions of Chen and Han

3. with Mustaq Ahmed

shortest descending paths
on polyhedral terrain

P vs NP-complete?



Chen and Han needs

- method to extend a shortest path into "next" face — can do – Snell's law
- method to find shortest path to specified target through specified face sequence *
- one-vertex one-cut to prune the segment tree — true

this is the hard part.