Approximating Shortest Paths on a Convex Polytope in Three Dimensions

Pankaj Agarwal, Sariel Har-Peled, Micha Sharir, Kasturi Varadarajan

(Hershberg and Suri, Practical Methods for Approximating Shortest Paths on a Convex Polytope in R3)
(Chen and Han, Shortest Paths on a Polyhedron)

Why is it hard?

- Paths pass through edges
- Intersection points have high algebraic complexity
- Best exact answer is quadratic

Key Ideas

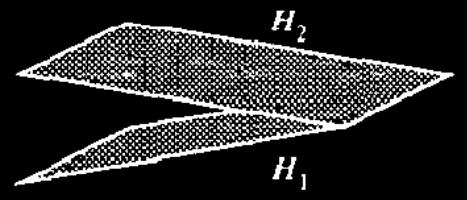
- Exact solution of approximate problem
  - Used twice
- Unfolding of paths

Algorithm Overview

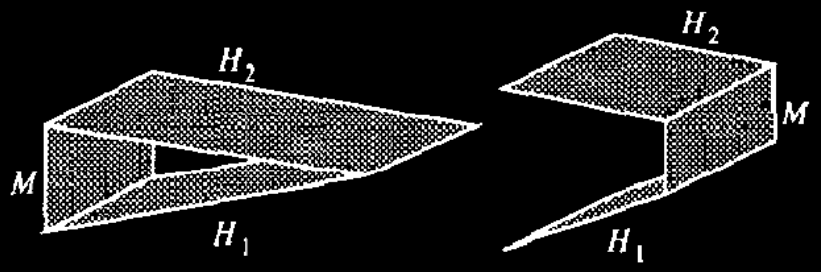
- Coarsely approximate shape
  - Compute factor of 2 bound
- Approximate the right amount
  - Using previous estimate
  - Compute epsilon shortest path
- Project path onto shape
  - This shortens the path

Coarse approximation

- Factor of 2
- Approximate with sets of wedges containing shape
- Find shortest path on each wedge

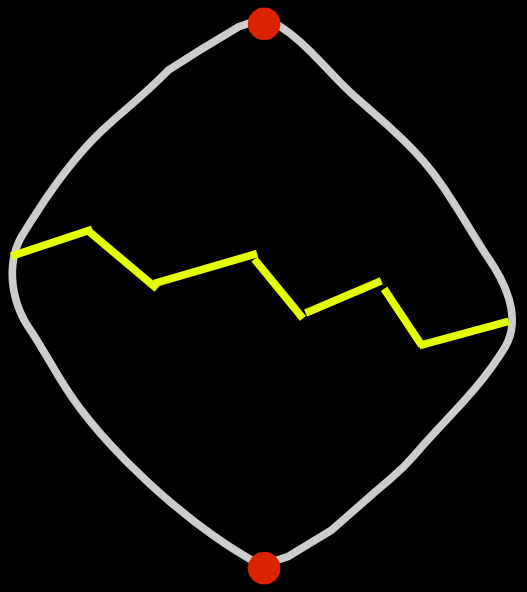


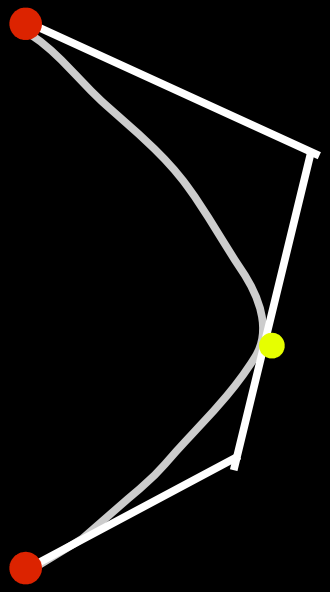
(a)



Coarse Approximation

- Define wedges using horizon edges
- Take shortest path over wedges

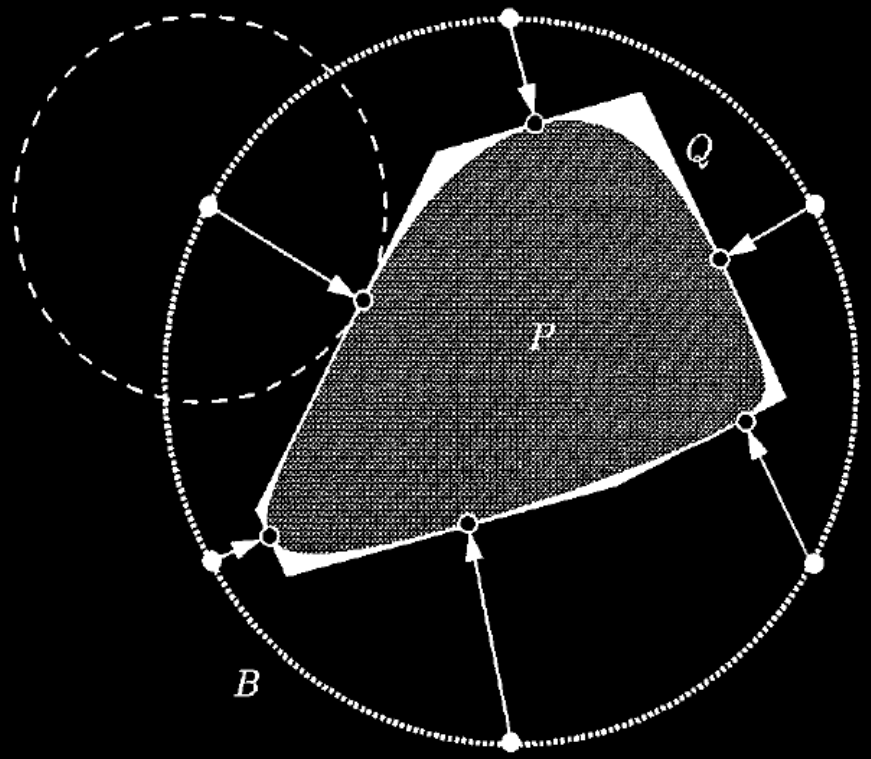




Hershberger and Suri

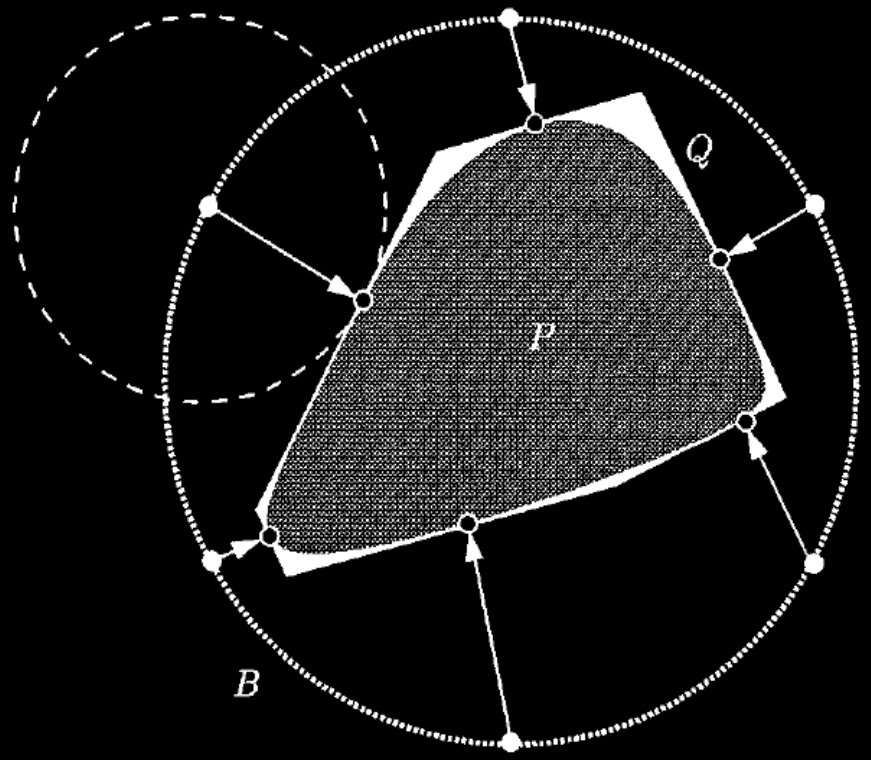
Approximating the Shape

- Enclose shape in sphere
- Sample sphere
- Find closest points/support planes on shape



Approximating the Shape

- Can compute in O(n+(1/u)log n) time
- A polytope with O(1/u) vertices
- Within polytope expanded by u

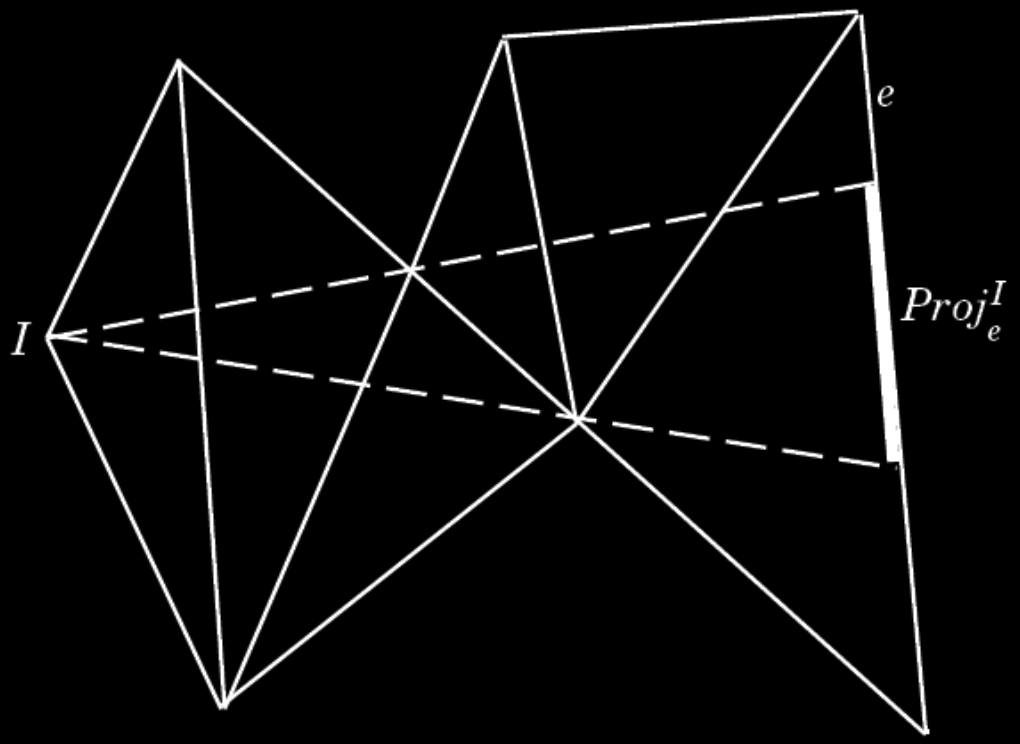


Bounding Path Length

- Prove that there is a short path near the surface
  - Can offset a path from the surface
  - Simplify path in the Gaussian map
  - Still outside
- Path on surface is shorter

Compute Exact Shortest Path

- Unfold polyhedron

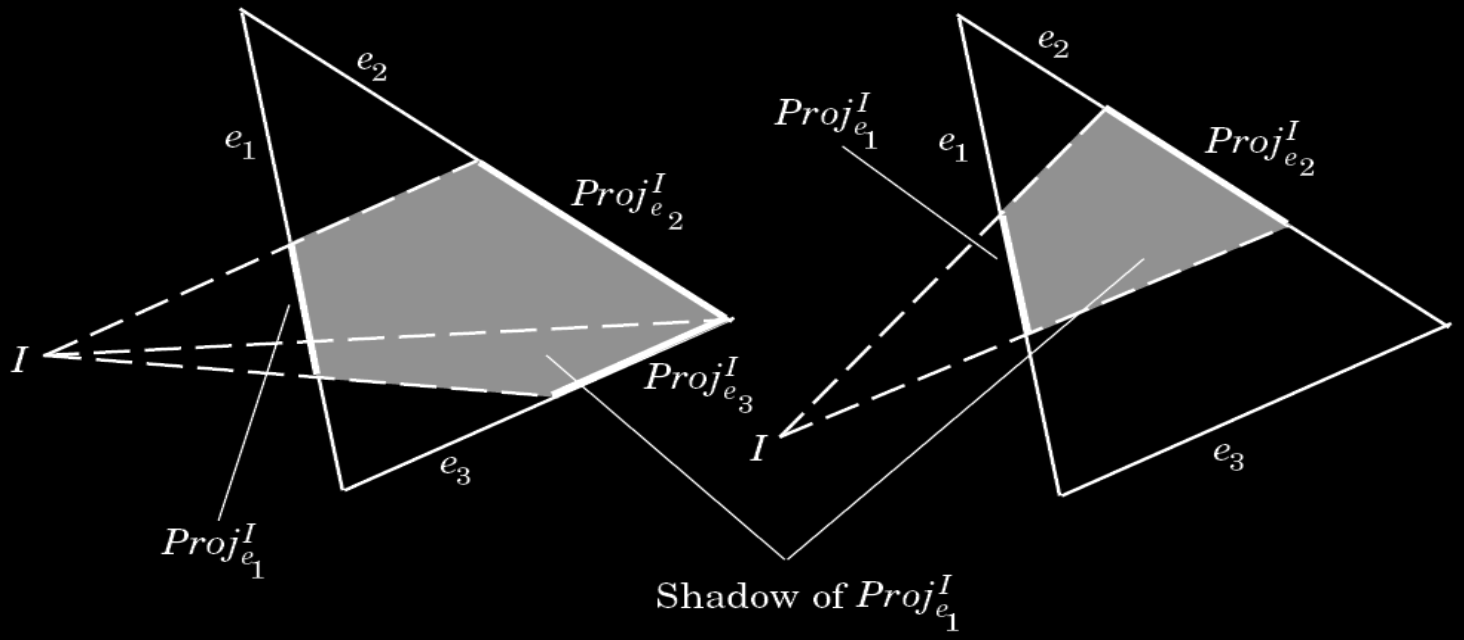


Exact Algorithm

- N iterations of
  - Take each leaf
  - If it splits, see if it is shorter than other instances of split
- Only ever n leaves

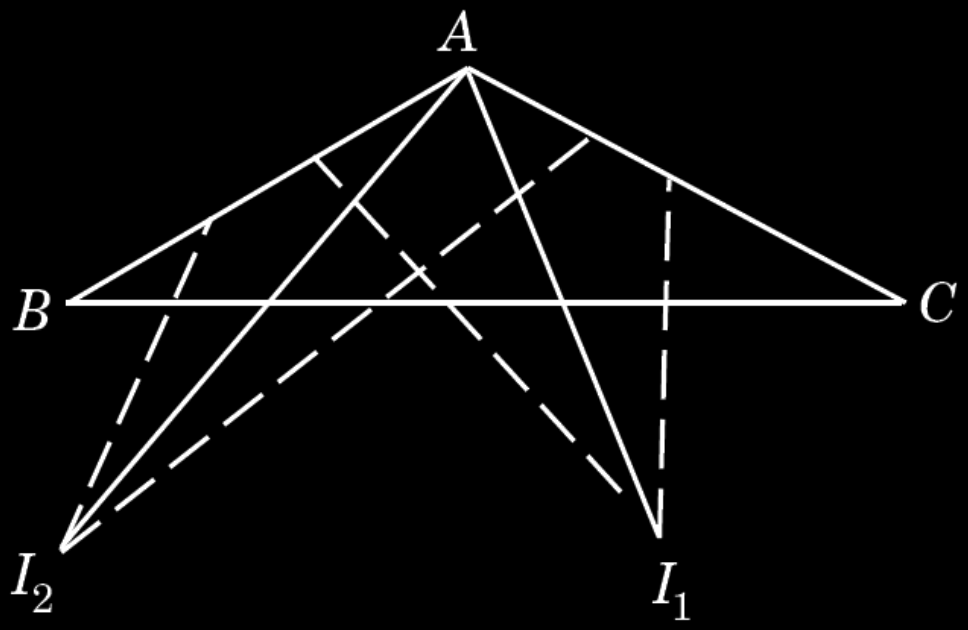
Bounding Size

- Tree depth n
- Naive unfolding is exponential
  - Each node can have 2 children



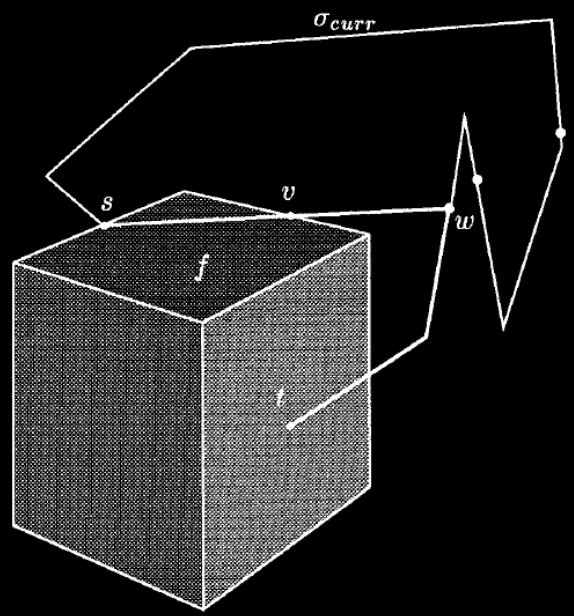
Trimming

- “One angle one split”



Snapping the Path

- Iteratively project the path
- Can only shorten path
- Use Gaussian diagram to rapidly find intersections
  - The path being snapped is has few edges



Tying it Together

- Want path between s and t
- Compute d, 2-bound on the path length
  - Time O(n)
- Set $r = \varepsilon^{3/2} d/440$
- Compute an approximation with r vertices
  - Time O(n)
- Slice approximation to put s,t on surface
- Compute exact shortest path on surface
  - Time $O(1/\varepsilon^3)$

Shortest Path to All Vertices

- Cannot compute all 2 approximations

- Compute 2.4 approximation one to all
  - O(n log n) time
- Otherwise more or less identical (just done n times)
- Result is $O(n/\varepsilon^3 + n/\varepsilon^{1.5} \log n)$