Approximating the Smallest k-Enclosing Geodesic Disc in a Simple Polygon

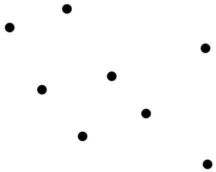
Prosenjit Bose[1] Anthony D'Angelo[1] Stephane Durocher[2]

[1]Carleton University

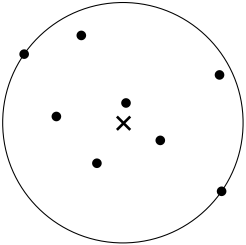
[2]University of Manitoba

18th Algorithms and Data Structures Symposium

Problem



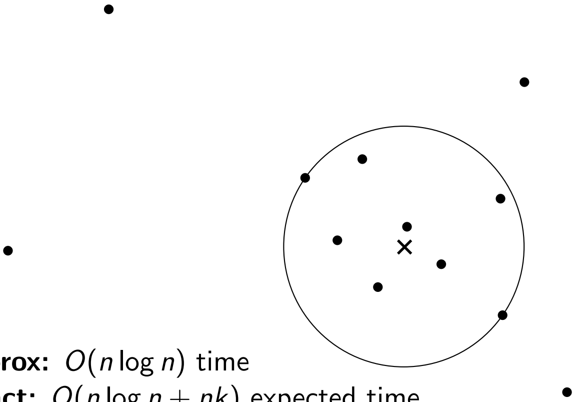
Problem



exact: $O(n)$ time

(Megiddo, '83), (Welzl, '91)

Problem

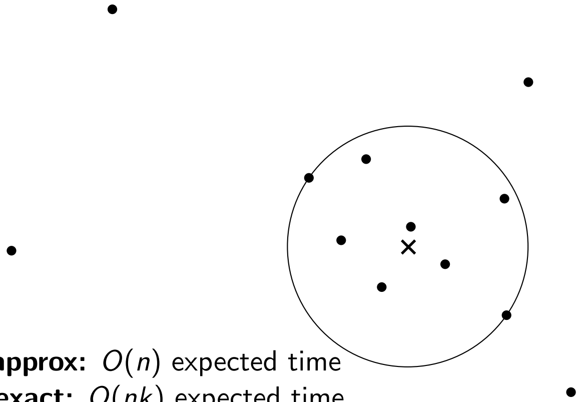


2-**approx:** $O(n \log n)$ time

exact: $O(n \log n + nk)$ expected time

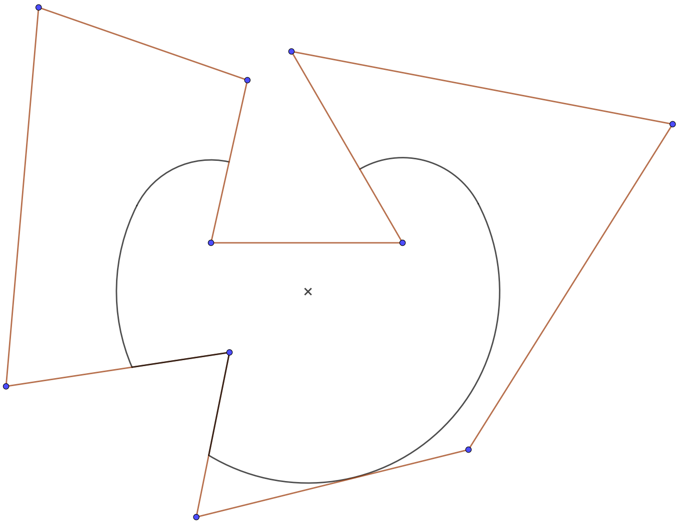
(Matoušek, '95)

Problem

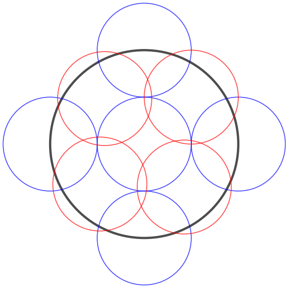


2-**approx:** $O(n)$ expected time
exact: $O(nk)$ expected time
(Har-Peled & Mazumdar, '05)

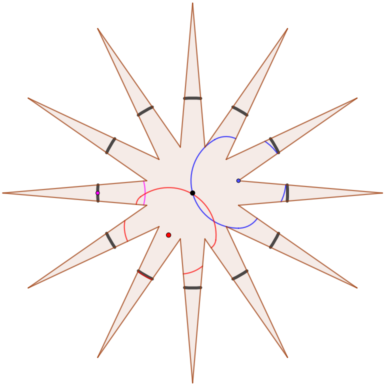
Problem



Problem



Euclidean



Geodesic

Problem

Smallest k-Enclosing Geodesic Disc

Given:

- a simple polygon P defined by a set of m vertices in $\mathbb{R}^2$, r of which are reflex vertices
- a set S of n points of $\mathbb{R}^2$ contained in P

Find a geodesic disc of minimum radius centred and contained in P that contains k points of S.

Problem

KEG disc k-**enclosing geodesic disc**
geodesic disc centred and contained in P that contains k points of S

Problem

KEG disc k-**enclosing geodesic disc**
geodesic disc centred and contained in P that contains k points of S

SKEG disc **smallest** k-**enclosing geodesic disc**
KEG disc with smallest radius

Problem

KEG disc — *k*-**enclosing geodesic disc**
geodesic disc centred and contained in P that contains k points of S

SKEG disc — **smallest** *k*-**enclosing geodesic disc**
KEG disc with smallest radius

2-SKEG disc — a KEG disc whose radius is at most twice that of a SKEG disc

Problem

Theorem

We compute a 2-SKEG disc:

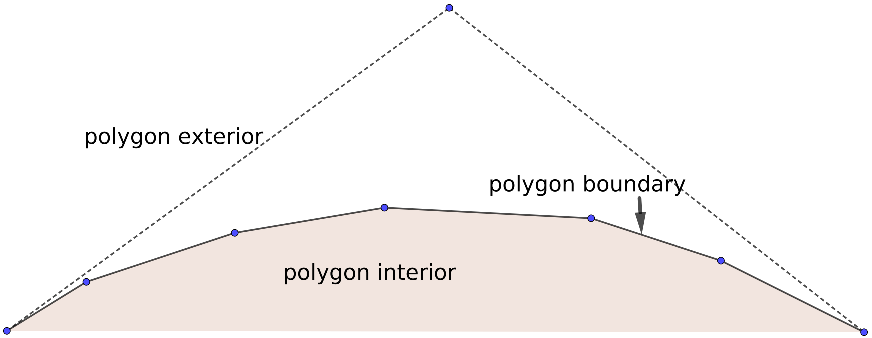
- *if* $k \in \omega(n/\log n)$, *in* $O((n^2/k)\log n \log r + m)$ *time and* $O(n+m)$ *space, but we compute a 2-SKEG disc with high probability (i.e., at least* $1 - 1/n$*)*
- *if* $k \in O(n/\log n)$, *in* $O(n \log^2 n \log r + m)$ *expected time,* $O(n+m)$ *expected space*

N.B.: When $k \in \Theta(n)$, the runtime is $O(n \log n \log r + m)$

Tools

Polygon Simplification

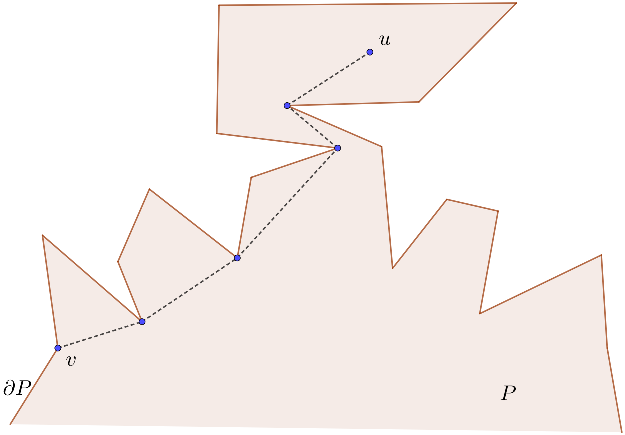
$O(m)$ time **(Aichholzer et al., '14)**
Size $O(r)$, preserves visibility, shortest paths



Tools

Shortest-Path Query Data Structure

$O(r)$ build time **(Guibas, Hershberger, '89)**
$O(\log r)$ query time



Random Sampling

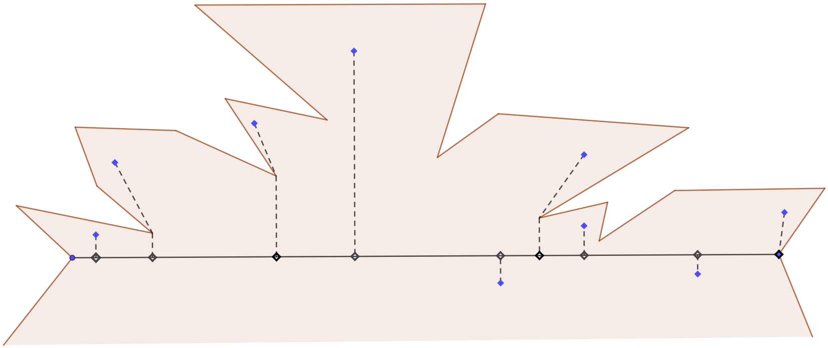
Compute a random sample of $(n/k)\ln n$ points of S and compute the $(k-1)^{\text{st}}$ closest point of S to each.

Theorem

RS-Algo computes a 2-SKEG disc with probability at least $1 - 1/n$ in $O((n^2/k)\log n \log r + m)$ time and $O(n + m)$ space.

Divide-and-Conquer

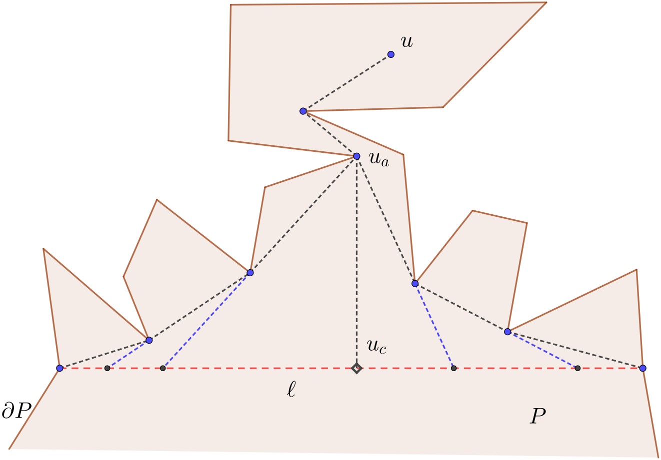
Projection



Divide-and-Conquer

Computing Projections

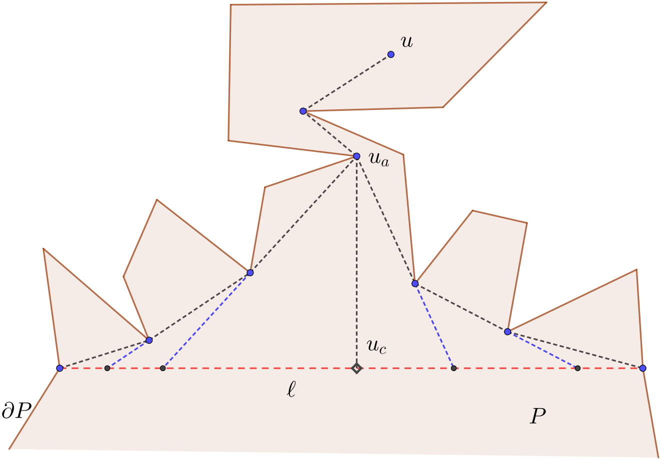
(Pollack, Sharir, Rote, '89)



Divide-and-Conquer

Computing Projections

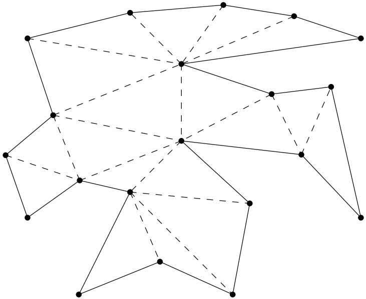
(Pollack, Sharir, Rote, '89) $\implies \angle u_c$ closest to $\pi/2$



Divide-and-Conquer

Balanced Hierarchical Polygon Decomposition

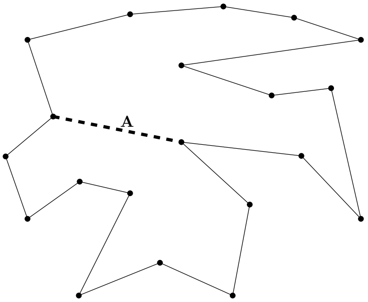
(Chazelle, '82), (Guibas et al., '87) $O(r)$ time / space



Divide-and-Conquer

Balanced Hierarchical Polygon Decomposition

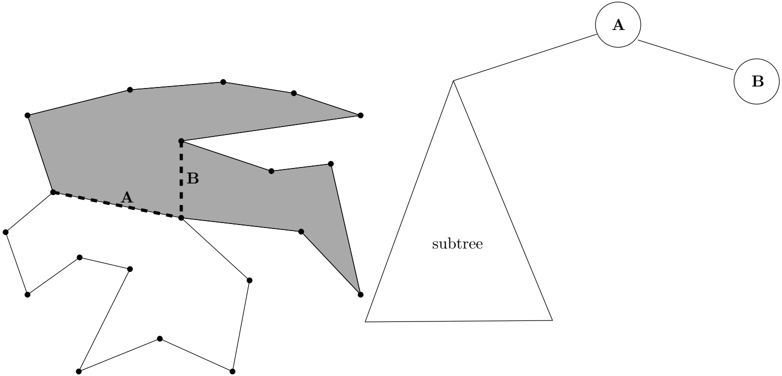
(Chazelle, ’82**), (Guibas et al.,** ’87**)** $O(r)$ time / space



Divide-and-Conquer

Balanced Hierarchical Polygon Decomposition

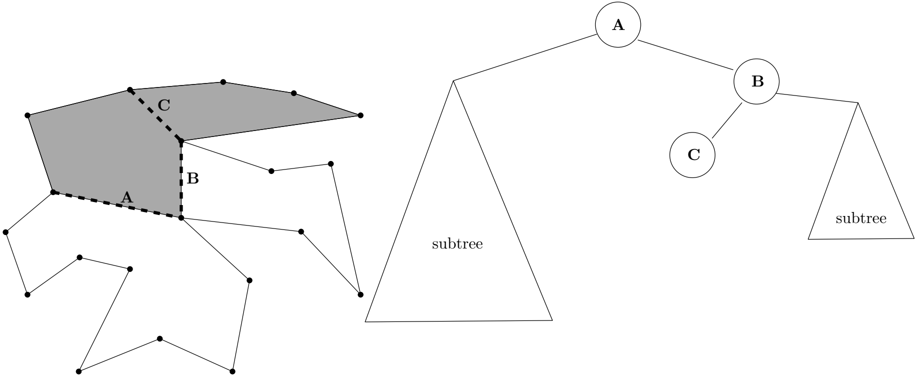
(Chazelle, '82**), (Guibas et al.,** '87**)** $O(r)$ time / space



Divide-and-Conquer

Balanced Hierarchical Polygon Decomposition

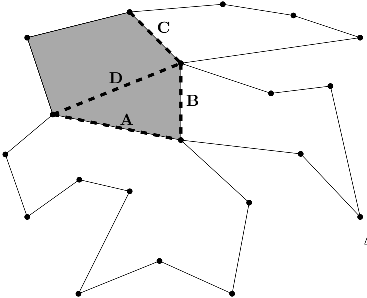
(Chazelle, ’82**), (Guibas et al.,** ’87**)** $O(r)$ time / space

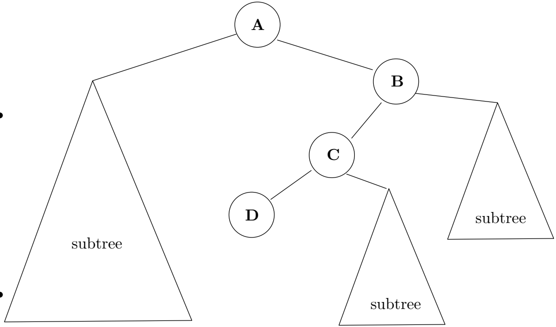


Divide-and-Conquer

Balanced Hierarchical Polygon Decomposition

(Chazelle, '82), (Guibas et al., '87) $O(r)$ time / space

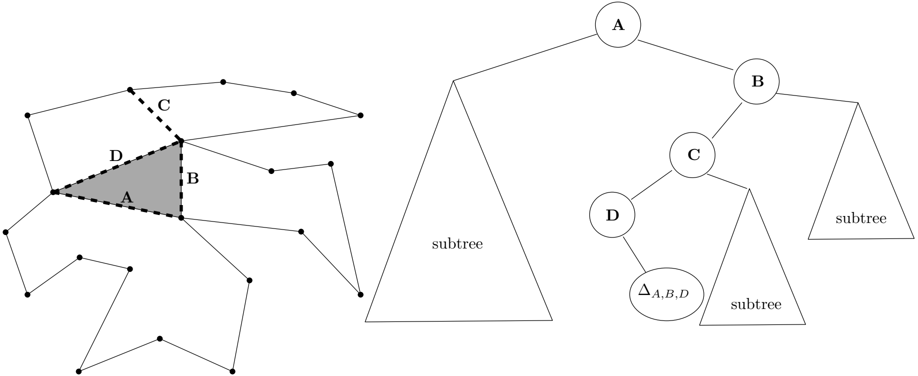




Divide-and-Conquer

Balanced Hierarchical Polygon Decomposition

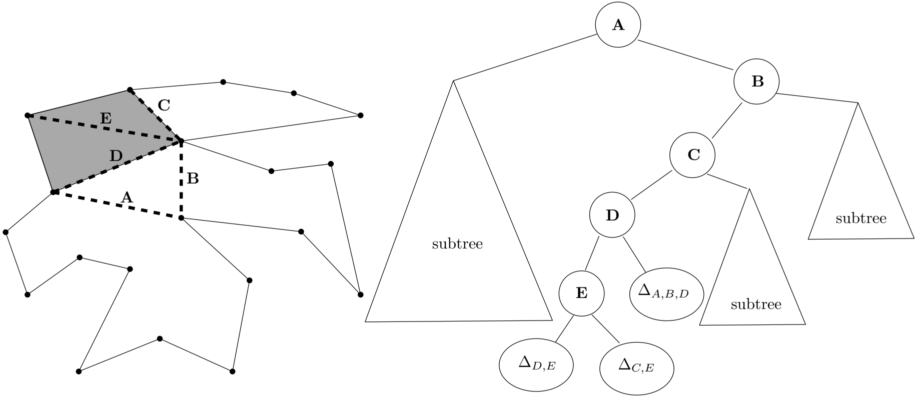
(Chazelle, ’82**), (Guibas et al.,** ’87**)** $O(r)$ time / space



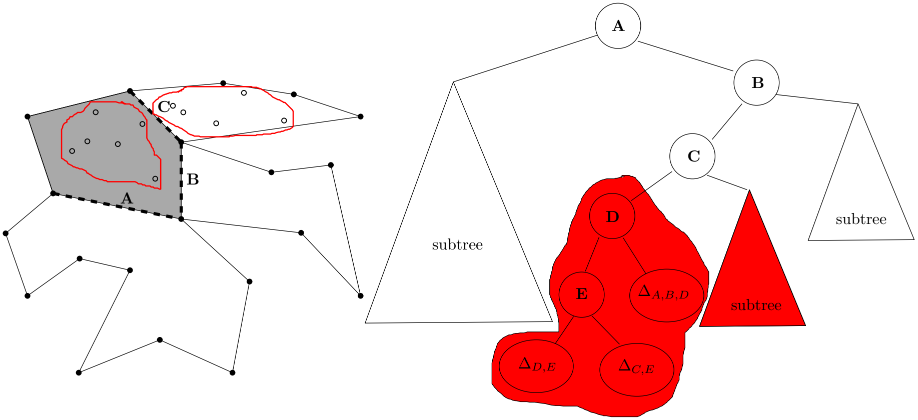
Divide-and-Conquer

Balanced Hierarchical Polygon Decomposition

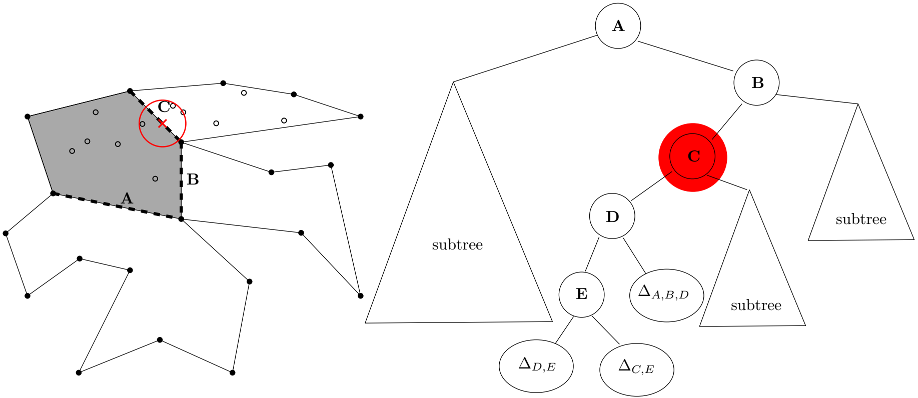
(Chazelle, ’82), (Guibas et al., ’87) $O(r)$ time / space



Divide-and-Conquer

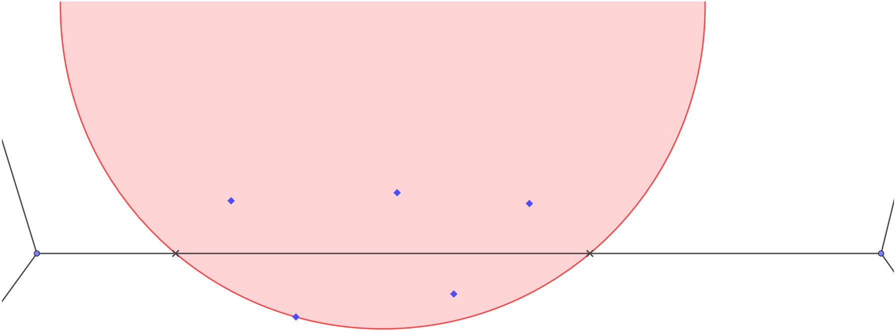


Divide-and-Conquer

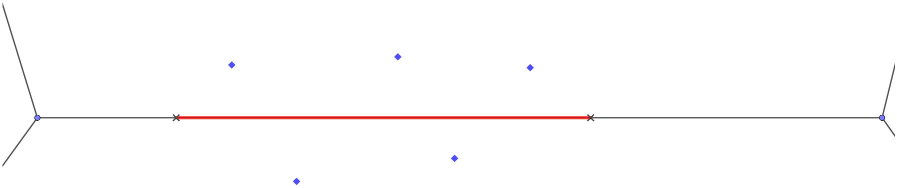


Divide-and-Conquer

Merge



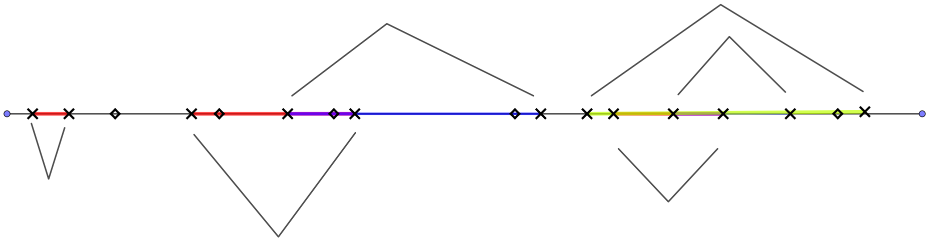
Divide-and-Conquer
Merge



Divide-and-Conquer

Merge

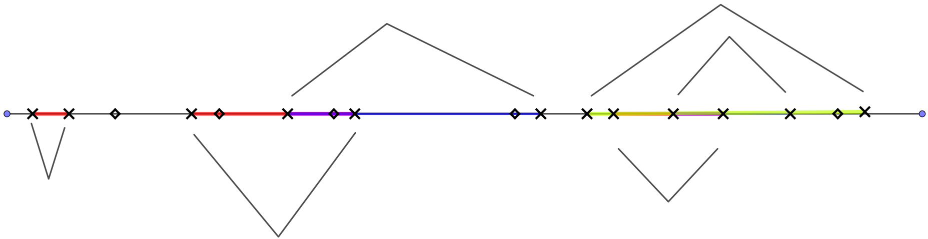
- pick a candidate projection at random



Divide-and-Conquer

Merge

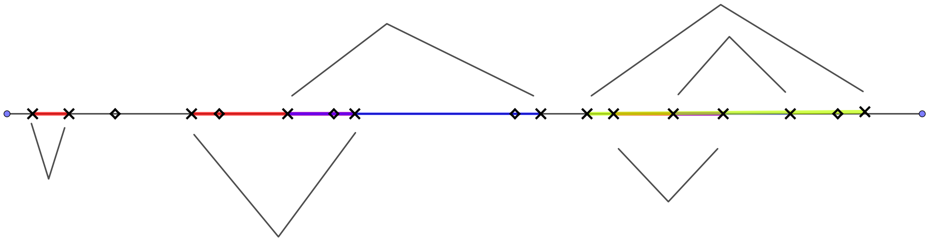
- pick a candidate projection at random
- $\rho =$ distance to k^{th} closest neighbour of S



Divide-and-Conquer

Merge

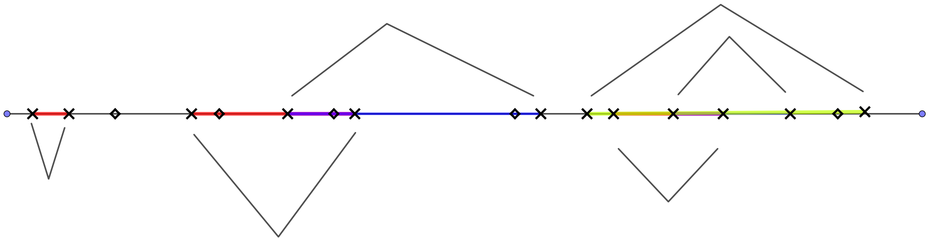
- pick a candidate projection at random
- $\rho =$ distance to k^{th} closest neighbour of S
- for each point of S, compute the interval along the diagonal within distance ρ



Divide-and-Conquer

Merge

- pick a candidate projection at random
- $\rho =$ distance to k^{th} closest neighbour of S
- for each point of S, compute the interval along the diagonal within distance ρ
- overlay, count depth of candidate projections, update candidates



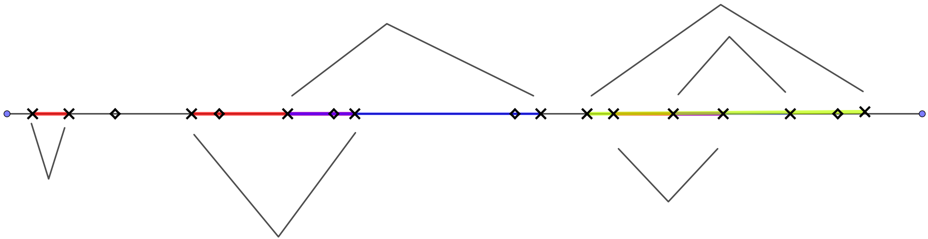
Divide-and-Conquer

Merge

- pick a candidate projection at random
- $\rho =$ distance to k^{th} closest neighbour of S
- for each point of S, compute the interval along the diagonal within distance ρ
- overlay, count depth of candidate projections, update candidates

(Devroye, '88) "Theory of Records"

$O(\log n)$ iterations with high probability



Theorem

DI-Algo computes a 2-SKEG disc in $O(n \log^2 n \log r + m)$ *expected time and* $O(n + m)$ *expected space.*

Comparing to Higher Order Geodesic VDs

Ignoring Polylogs

Can be solved exactly with higher-order geodesic VDs in worst-case time $O(k^2 n + k^2 r + \min(kr, r(n-k)) + m)$.

Comparing to Higher Order Geodesic VDs

Ignoring Polylogs

2-**SKEG:** $O(n + m)$
OKGVD:

Comparing to Higher Order Geodesic VDs

Ignoring Polylogs

2-SKEG: $O(n+m)$

OKGVD:

- for $k \in \Theta(1)$: $O(n+m)$
- for $k \in \Omega(n)$, $k < n-1$: $O(\text{more than } n^3)$ time

The End

Related Results

- Coverings/packing simple polygon with geodesic discs [11, 13]
- Geodesic centre, simple polygon [1, 3, 5, 10, 12]
  Geodesic 2-centre, simple polygon [9, 13]
- Geodesic centre, n points in simple m-gon: $O(m + n\log(mn))$ [2, 7, 12]
  Geodesic 2-centre, n points in simple m-gon: $O(n(m+n)\log^3(m+n))$ [8]
- Simple m-gon, n points, all geodesic discs of radius ρ that contain at least k points [4]: for output size $Y \in O(nm)$
  (ignoring polylogs)
  $O(m + (Ym)^{2/3} + Y + n^2)$
- Geodesic k-Nearest Neighbour Queries (static) [6]:
  built in $O(n * \text{polylog})$ expected time
  queries in $O(k * \text{polylog})$ expected time

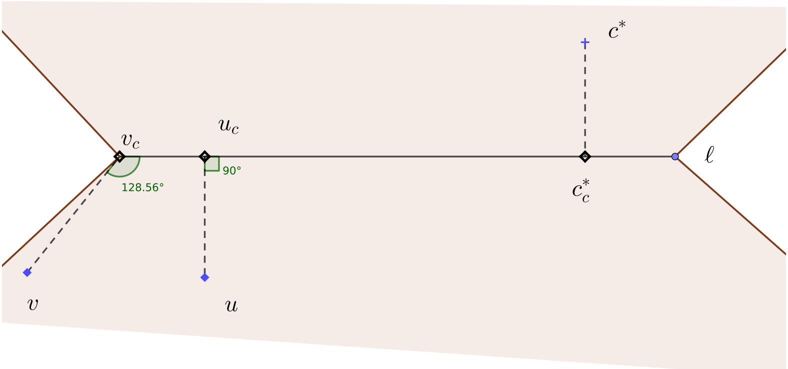
Divide-and-Conquer

Merge Disc Contains Some Projection

(Pollack, Sharir, Rote, '89) [10]

$\angle u u_c c^* \geq \pi/2$

$\implies \mathrm{d}(u, c^*) > \mathrm{d}(u_c, c^*), \mathrm{d}(u_c, u)$



References I

[1] H. Ahn, L. Barba, P. Bose, J. D. Carufel, M. Korman, and E. Oh. A linear-time algorithm for the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 56(4):836–859, 2016.

[2] B. Aronov, S. Fortune, and G. T. Wilfong. The furthest-site geodesic voronoi diagram. *Discrete & Computational Geometry*, 9:217–255, 1993.

[3] T. Asano and G. Toussaint. Computing the geodesic center of a simple polygon. In *Discrete Algorithms and Complexity*, pages 65–79. Elsevier, 1987.

[4] M. G. Borgelt, M. J. van Kreveld, and J. Luo. Geodesic disks and clustering in a simple polygon. *Int. J. Comput. Geometry Appl.*, 21(6):595–608, 2011.

References II

[5] P. Bose and G. T. Toussaint. Computing the constrained euclidean geodesic and link center of a simple polygon with application. In *Computer Graphics International*, pages 102–110. IEEE Computer Society, 1996.

[6] S. de Berg and F. Staals. Dynamic data structures for k-nearest neighbor queries. *Computational Geometry*, 111:101976, 2023.

[7] L. J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and System Sciences*, 39(2):126–152, 1989.

[8] E. Oh, S. W. Bae, and H. Ahn. Computing a geodesic two-center of points in a simple polygon. *Comput. Geom.*, 82:45–59, 2019.

[9] E. Oh, J. D. Carufel, and H. Ahn. The geodesic 2-center problem in a simple polygon. *Comput. Geom.*, 74:21–37, 2018.

[10] R. Pollack, M. Sharir, and G. Rote. Computing the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 4:611–626, 1989.

References III

[11] G. Rabanca and I. Vigan. Covering the boundary of a simple polygon with geodesic unit disks. *CoRR*, abs/1407.0614, 2014.

[12] G. Toussaint. Computing geodesic properties inside a simple polygon. *Revue D'Intelligence Artificielle*, 3(2):9–42, 1989.

[13] I. Vigan. Packing and covering a polygon with geodesic disks. *CoRR*, abs/1311.6033, 2013.