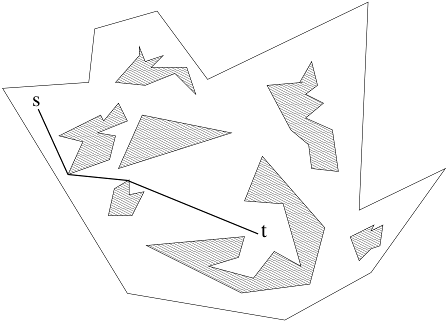
Computing a Euclidean shortest path in the plane with Dijkstra wavefront

R. Inkulu
http://www.iitg.ac.in/rinkulu/

under the guidance of
Sanjiv Kapoor and S. N. Maheshwari

Problem Description



n is the number of vertices in the polygonal domain

m is the number of holes (obstacles)

History

- Known Lower Bound: $\Omega(n + m \lg m)$ time with $O(n)$ space
- Visibility Graph

  [Welzl, 1985]: $O(n^2)$

  [Ghosh, Mount 1991] and [Kapoor, Maheshwari 1988]: $O(n \lg n + E)$

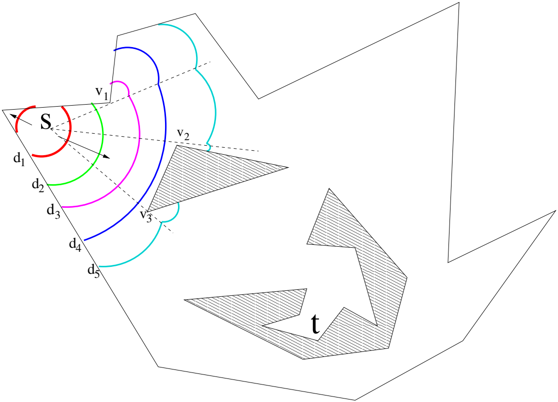
  [Storer, Reif 1994]: $O(T + mn)$
- Wavefront Propagation

  [Mitchell, 1993], [Kapoor 1999]: $O(n^{3/2+\epsilon})$ time and space

  [Hershberger, Suri 1999]: $O(n \lg n)$
- Wavefront Propagation with the restricted Visibility Graph

  This Research: $O(T + m(\lg m)(\lg n))$ time and $O(n)$ space

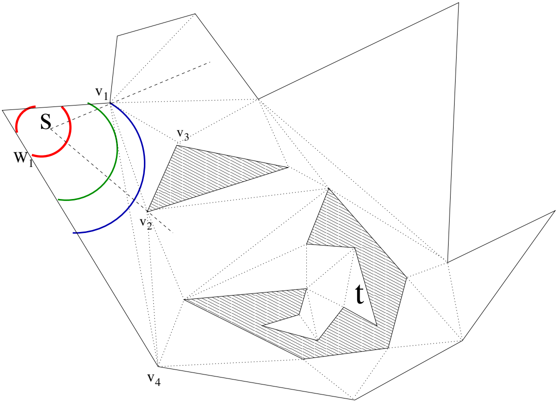
Wavefront Propagation



Wavefront, initiated at *s*, is expanded until it strikes *t*.

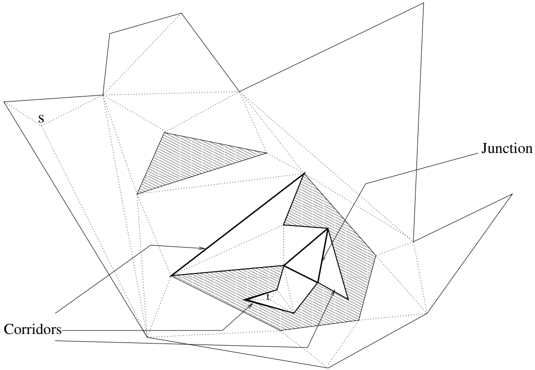
Initially wavefront comprises of single arc; with time, more arcs are added.

Triangulation to guide the wavefront



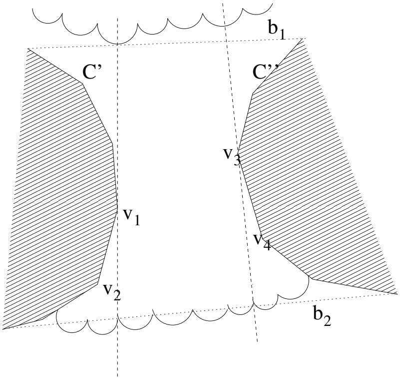
$O(T)$ is the time to triangulate the polygonal region ($O(n + m(\lg m)^{1+\epsilon})$, for a fixed positive constant $\epsilon > 0$)

Hourglasses and Junctions

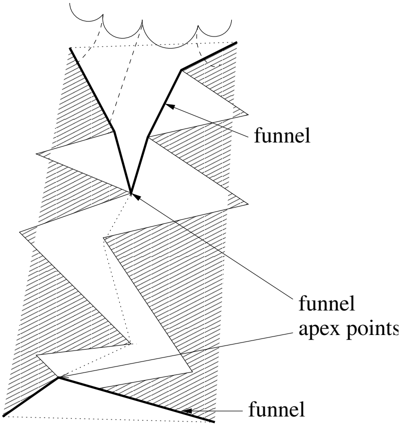


Given a triangulation, finding hourglasses and junctions takes $O(m \lg n)$ time

Open and Closed Hourglasses

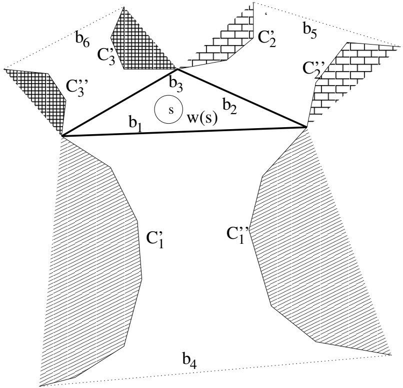


Open hourglass

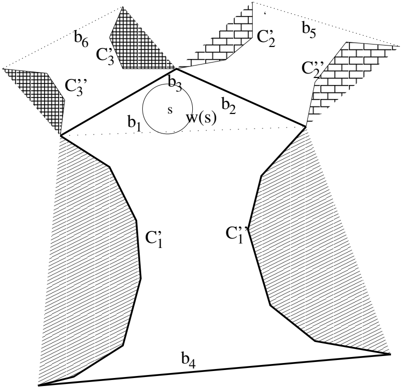


Closed hourglass

Boundary cycle and Event Points

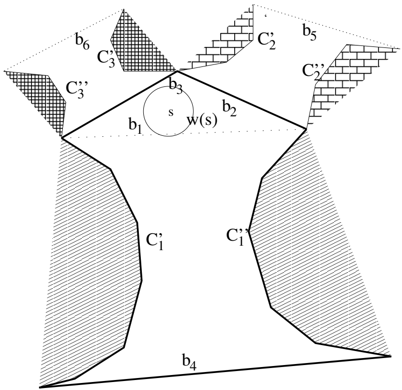


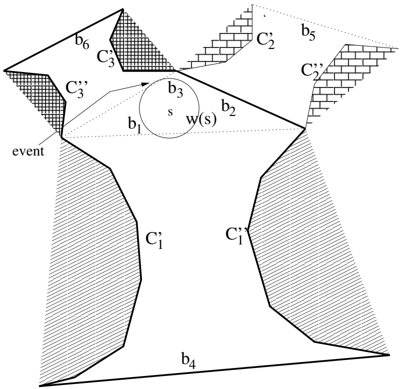
boundary cycle: b_1, b_2, b_3



boundary cycle: $C_1', b_4, C_1'', b_2, b_3$

Boundary cycle and Event Points (cont)

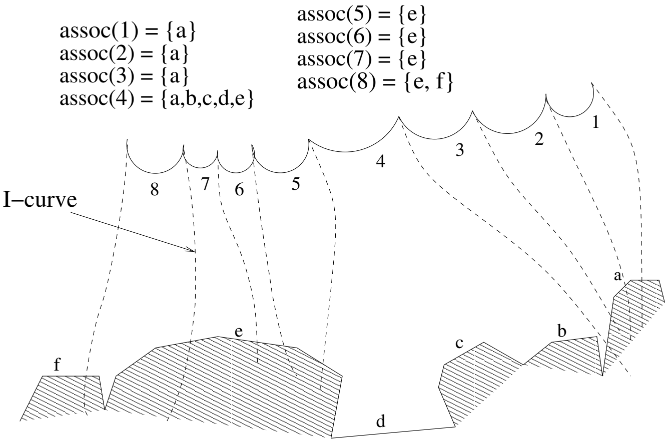




boundary cycle:
$C_1', b_4, C_1'', b_2, b_3$

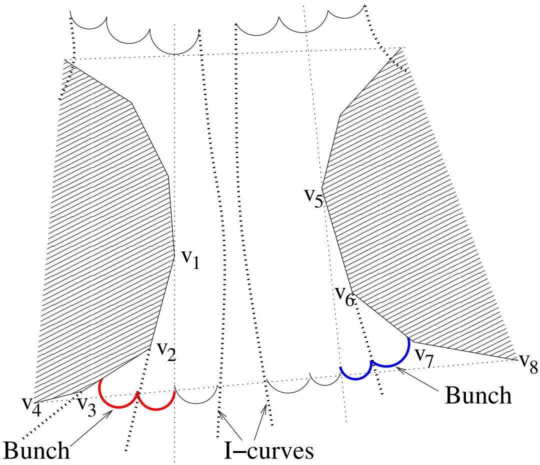
boundary cycle:
$C_1', b_4, C_1'', b_2, C_3', b_6, C_3''$

Associating wavefront segments to sections of boundary

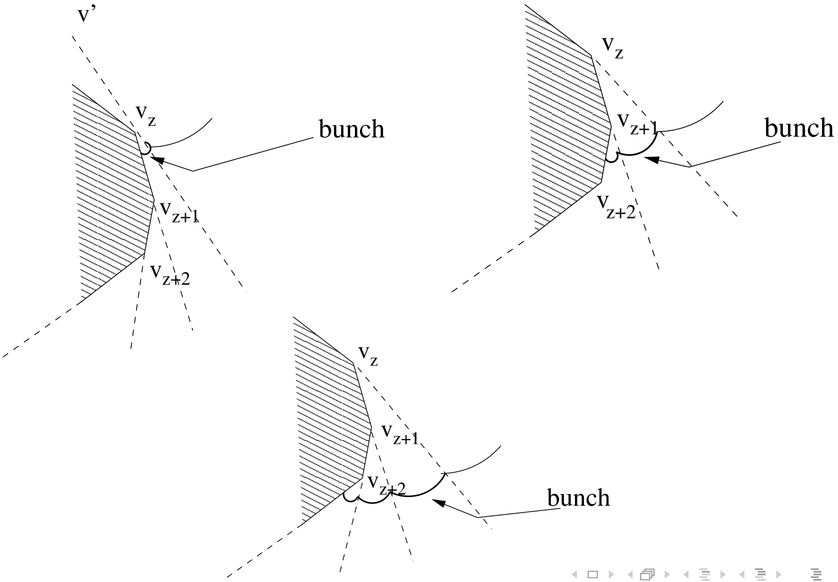


Each I-curve separates two weighted voronoi regions

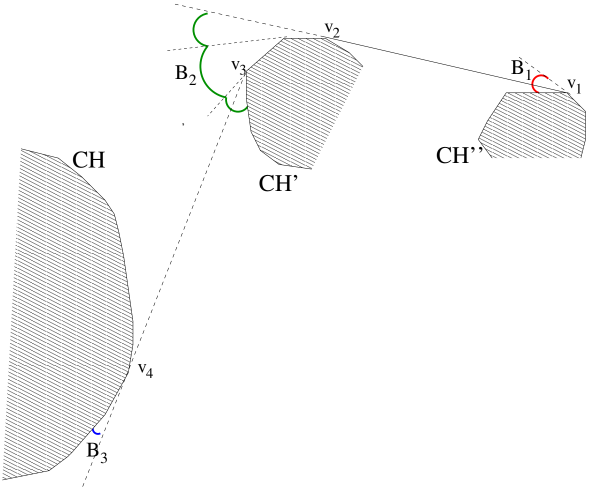
Exploiting convex chain structure: bunches



Bunch progression

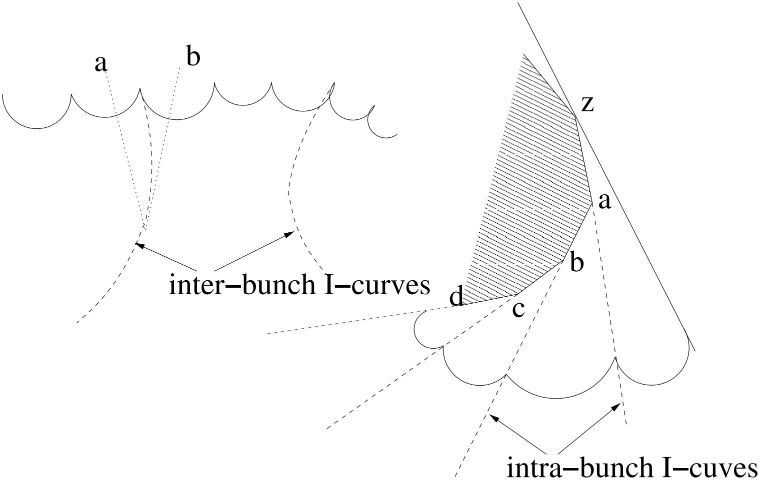


Wavefront: collection of bunches



Bunch B_1 caused bunch B_2, and bunch B_2 caused bunch B_3.

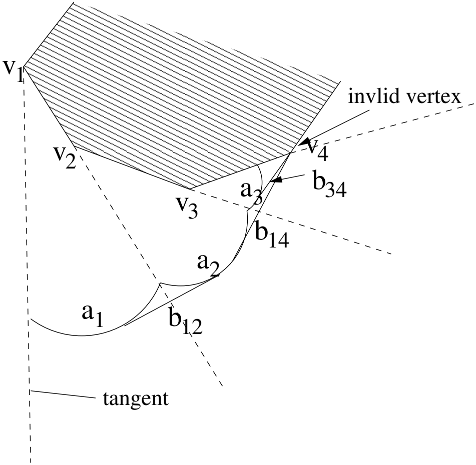
Inter- and Intra-bunch I-curves

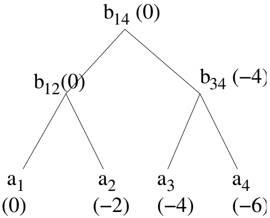


I-curve is a weighted voronoi curve, separating the voronoi regions

Diverging intra-bunch I-curves

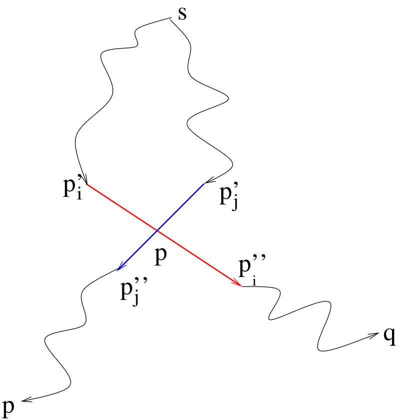
Convex hull approximation of a bunch

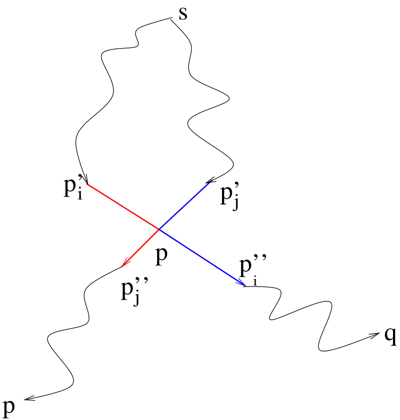




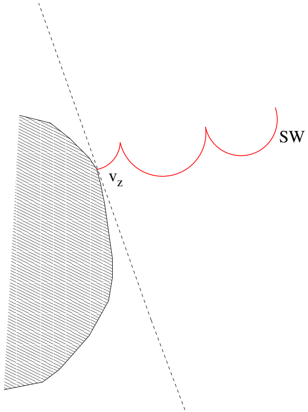
Eager initiation of segments in a bunch

Non-crossing property of shortest paths

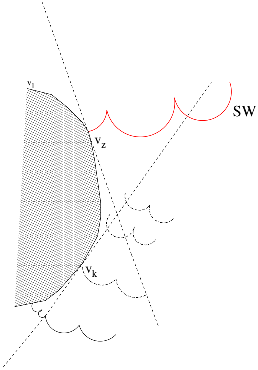




Initiating a new bunch

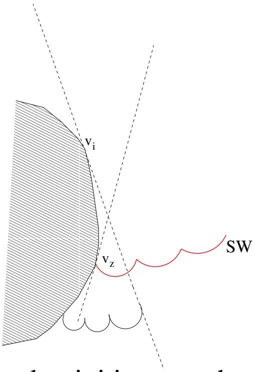


Initiates a new bunch from v_z

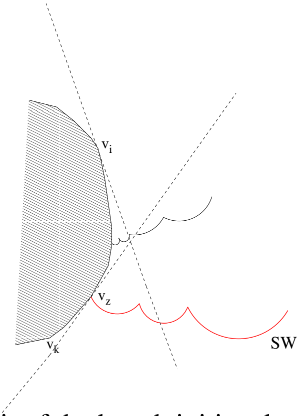


No need to initiate bunch from v_z

Initiating a new bunch (cont)



No need to initiate new bunch from v_z



Split of the bunch initiated at v_i

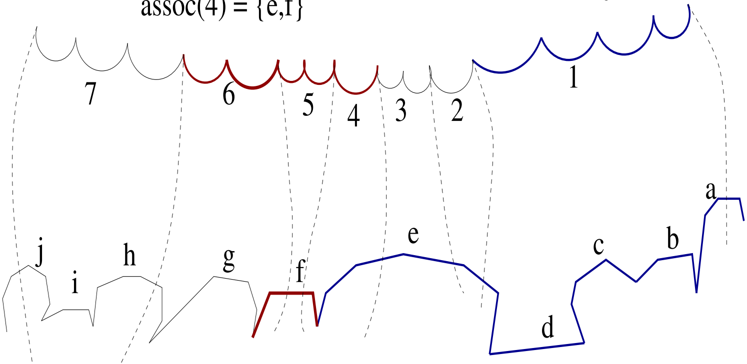
The total number of bunches at any point of execution of the algorithm are $O(m)$

Building and maintaining all BHTs during the entire algorithm takes $O(m \lg n + n)$

Associations & Boundary-sections & Waveform-sections

assoc(1) = {a,b,c,d,e}
assoc(2) = {e}
assoc(3) = {e}
assoc(4) = {e,f}

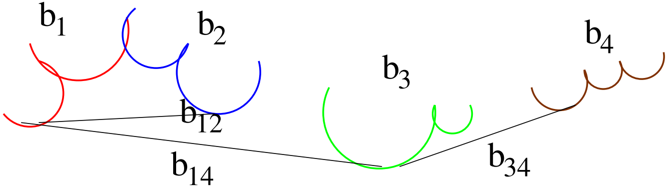
assoc(5) = {f}
assoc(6) = {f,g,h}
assoc(7) = {h,i,j}

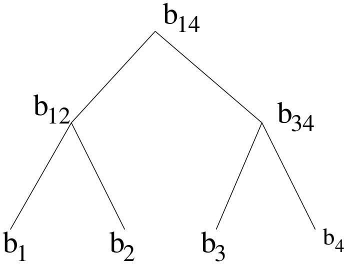


waveform−section(e) = {1,2,3,4}
waveform−section(f) = {4,5,6}

boundary−section(1) = {a,b,c,d,e}
boundary−section(6) = {f,g,h}
boundary−section(7) = {h,i,j}

Convex hull approximation of a section of wavefront

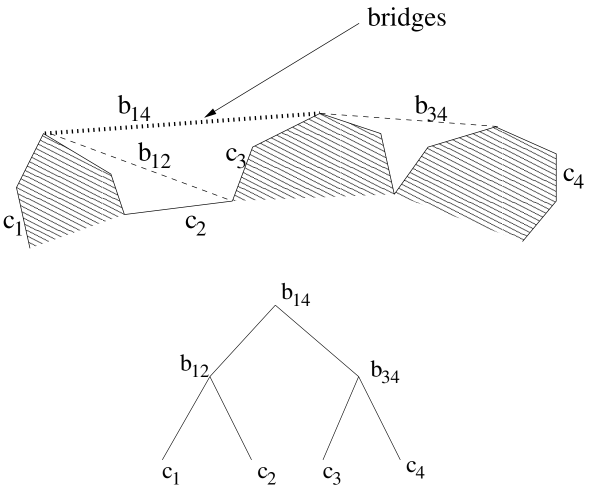




The number of waveform-sections created/updated are $O(m)$

The shortest distance computation between a waveform-section and a bunch takes $O((\lg m)(\lg n))$ amortized time

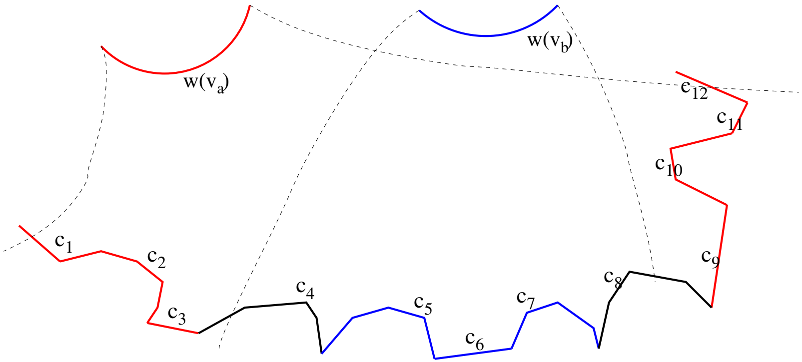
Convex hull approximation of a section of boundary



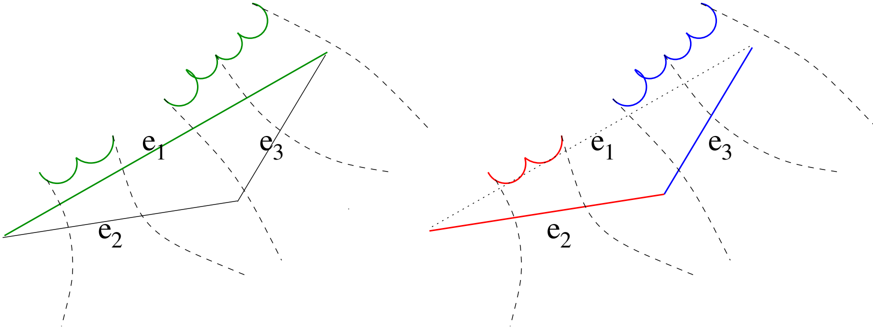
The number of boundary-sections created/updated are $O(m)$

The shortest distance computation between a boundary-section and a bunch takes $O((\lg m)(\lg n))$ amortized time

Contiguity property of associations



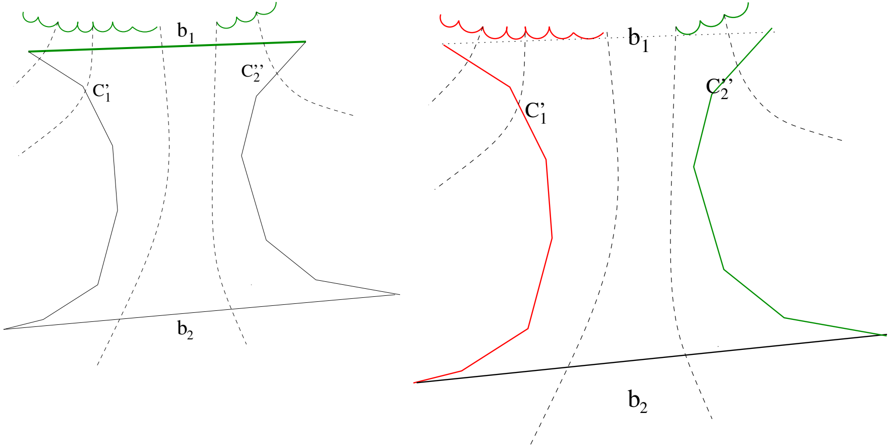
Wavefront split



The total number of splits at all junctions are $O(m)$

The total time complexity of splits at all junctions together $O(m(\lg m)(\lg n))$

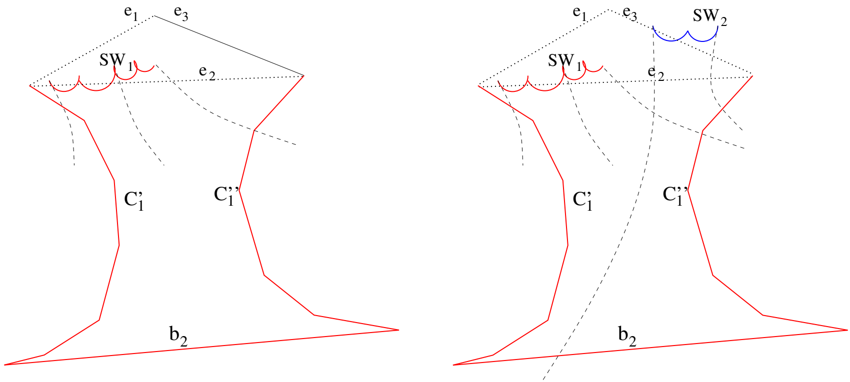
Wavefront split (cont)



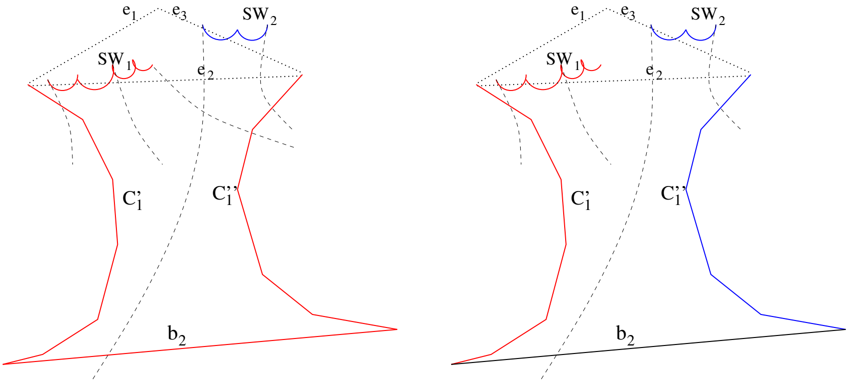
The total number of splits at all hourglasses are $O(m)$

The total time complexity of splits at all hourglasses together $O(m(\lg m)(\lg n))$

Wavefront Merger



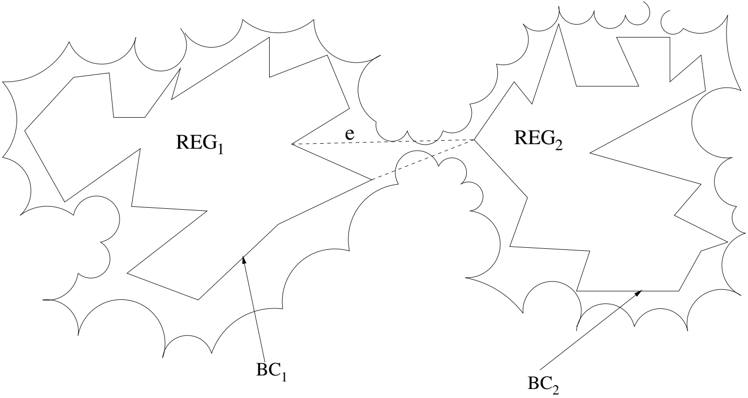
Wavefront Merger (cont)



The total number of merges during the entire algorithm are $O(m)$

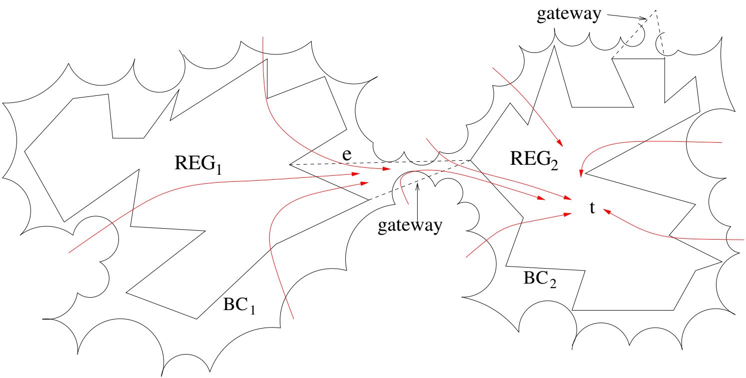
The total processing involved in the merge procedure during the entire algorithm is $O(m(\lg m)(\lg n))$

Boundary splits

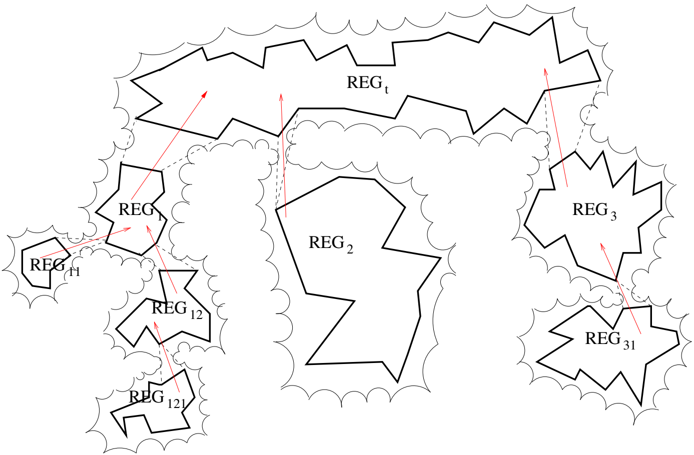


Every wavefront merger causes a boundary split, and vice versa.

Boundary splits (cont)



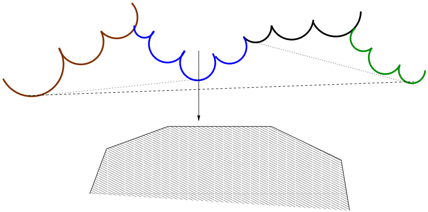
Ordering wavefront mergers at gateways



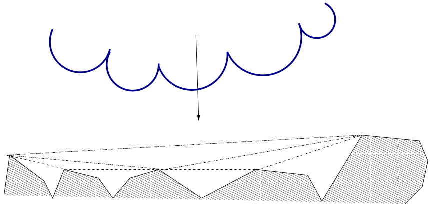
The total number of gateways/boundary splits are $O(m)$

The amortized time involved in boundary splits including orienting the gateways is $O((\lg m)(\lg n))$

Type-I and Type-II Events: Shortest distance computations between sections of wavefront and sections of boundary



Type-I

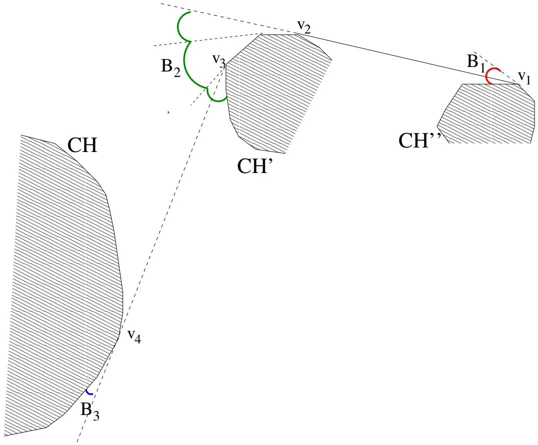


Type-II

The total number of Type-I and Type-II events are $O(m)$.

Time in determining and handling all Type-I and Type-II events together is $O(m(\lg m)(\lg n))$.

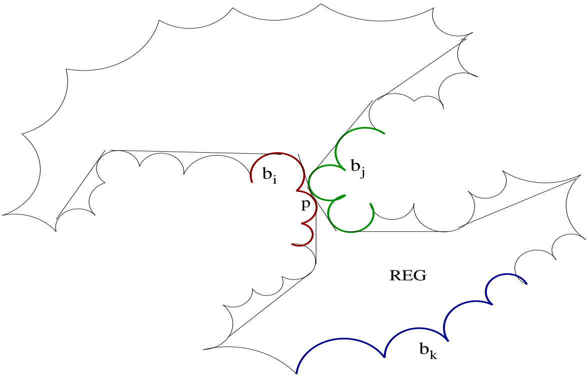
Type-III Event: Initializing bunches



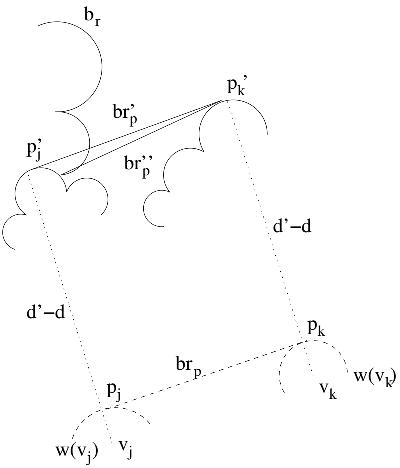
The total number of Type-III events are $O(m)$.

Time in determining and handling all Type-III events together is $O(m(\lg m)(\lg n))$.

Type-IV Event: collision of bunches within a waveform-section tree

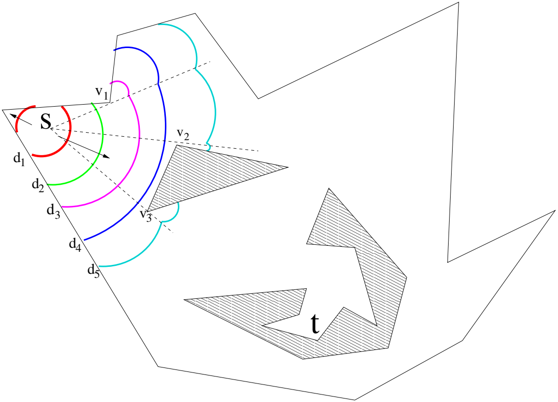


Bridge maintainance in waveform-section trees: Dirty bridges



The total number of Type-IV events are $O(m)$.
Time in determining and handling all Type-IV events together is $O(m(\lg m)(\lg n))$.

Summary



Provides a solution with $O(T + m(\lg m)(\lg n))$ time and $O(n)$ space to Problem 21 of The Open Problems Project (TOPP) of Computational Geometry, which intends for a solution with $O(n + m \lg m)$ time and $O(n)$ space.

Contributions

- Identified wavefront as a sequence of bunches.

Contributions

- Identified wavefront as a sequence of bunches.
- Eager initiation of wavefront segments within a bunch.

Contributions

- Identified wavefront as a sequence of bunches.
- Eager initiation of wavefront segments within a bunch.
- Devised more operations to Overmars structure:

Contributions

- Identified wavefront as a sequence of bunches.
- Eager initiation of wavefront segments within a bunch.
- Devised more operations to Overmars structure:
  - Inserting/Deleting convex chains.

Contributions

- Identified wavefront as a sequence of bunches.
- Eager initiation of wavefront segments within a bunch.
- Devised more operations to Overmars structure:
  - Inserting/Deleting convex chains.
  - Finding shortest distance between a convex chain, and convex chains stored in Overmars structure.

Contributions

- Identified wavefront as a sequence of bunches.
- Eager initiation of wavefront segments within a bunch.
- Devised more operations to Overmars structure:
  - Inserting/Deleting convex chains.
  - Finding shortest distance between a convex chain, and convex chains stored in Overmars structure.
  - Finding first intersection point of a higher order Voronoi curve with the contiguous convex chains stored in Overmars structure.

Contributions

- Identified wavefront as a sequence of bunches.
- Eager initiation of wavefront segments within a bunch.
- Devised more operations to Overmars structure:
  - Inserting/Deleting convex chains.
  - Finding shortest distance between a convex chain, and convex chains stored in Overmars structure.
  - Finding first intersection point of a higher order Voronoi curve with the contiguous convex chains stored in Overmars structure.
- Viewed boundary splits' as a tree, and applied backtracking based wavefront association merger.

Contributions

- Identified wavefront as a sequence of bunches.
- Eager initiation of wavefront segments within a bunch.
- Devised more operations to Overmars structure:
  - Inserting/Deleting convex chains.
  - Finding shortest distance between a convex chain, and convex chains stored in Overmars structure.
  - Finding first intersection point of a higher order Voronoi curve with the contiguous convex chains stored in Overmars structure.
- Viewed boundary splits' as a tree, and applied backtracking based wavefront association merger.
- Characterized bridge movement in waveform-section trees, and the bridge maintainance in terms of dirty bridges.

Contributions

- Identified wavefront as a sequence of bunches.
- Eager initiation of wavefront segments within a bunch.
- Devised more operations to Overmars structure:
  - Inserting/Deleting convex chains.
  - Finding shortest distance between a convex chain, and convex chains stored in Overmars structure.
  - Finding first intersection point of a higher order Voronoi curve with the contiguous convex chains stored in Overmars structure.
- Viewed boundary splits' as a tree, and applied backtracking based wavefront association merger.
- Characterized bridge movement in waveform-section trees, and the bridge maintainance in terms of dirty bridges.
- Improved the time complexity of shortest path computation in polygonal domains.

E. Welzl.
Constructing the visibility graph for n line segments in $O(n^2)$ time.
IPL 1985.

S. K. Ghosh and D.M. Mount.
An output-sensitive algorithm for computing visibility graphs.
SIAM Journal on Computing 1991.

J. A. Storer and J. H. Reif.
Shortest paths in the plane with polygonal obstacles.
Journal of ACM 1994.

S. Kapoor and S. N. Maheshwari.
Efficient algorithms for Euclidean shortest path and visibility problems with polygonal obstacles.
SoCG 1988.

J. S. B. Mitchell.
Shortest paths among obstacles in the plane.
SoCG 1993.

S. Kapoor.
Efficient computation of geodesic shortest paths.
STOC 1999.

J. Hershberger and S. Suri.
An optimal algorithm for Euclidean shortest paths in the plane.
SIAM Journal on Computing 1999.

S. Kapoor, S. N. Maheshwari and J. S. B. Mitchell.
An efficient algorithm for Euclidean shortest path among polygonal obstalces in the plane.
Discrete & Computational Geometry 1997.

S. Kapoor and S. N. Maheshwari.
Efficiently constructing the visibility graph of a simple polygon with obstacles.
SIAM Journal on Computing 2000.