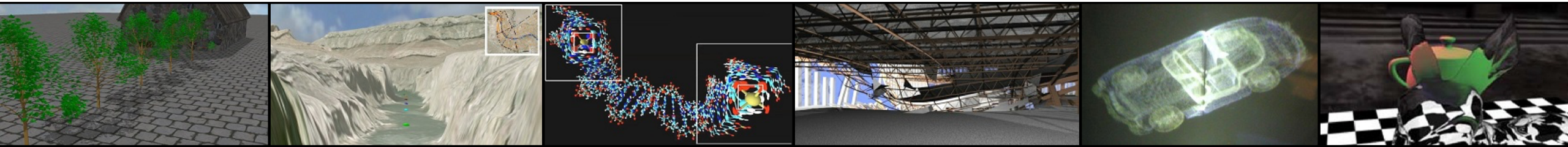
COT 4521: Introduction to Computational Geometry

Voronoi Diagram - Fortune's Algorithm

Paul Rosen
Assistant Professor
University of South Florida



Fortune's algorithm

- **Fortune's algorithm** is a sweep line algorithm for generating a Voronoi diagram from a set of points in a plane using $O(n \log n)$ time and $O(n)$ space

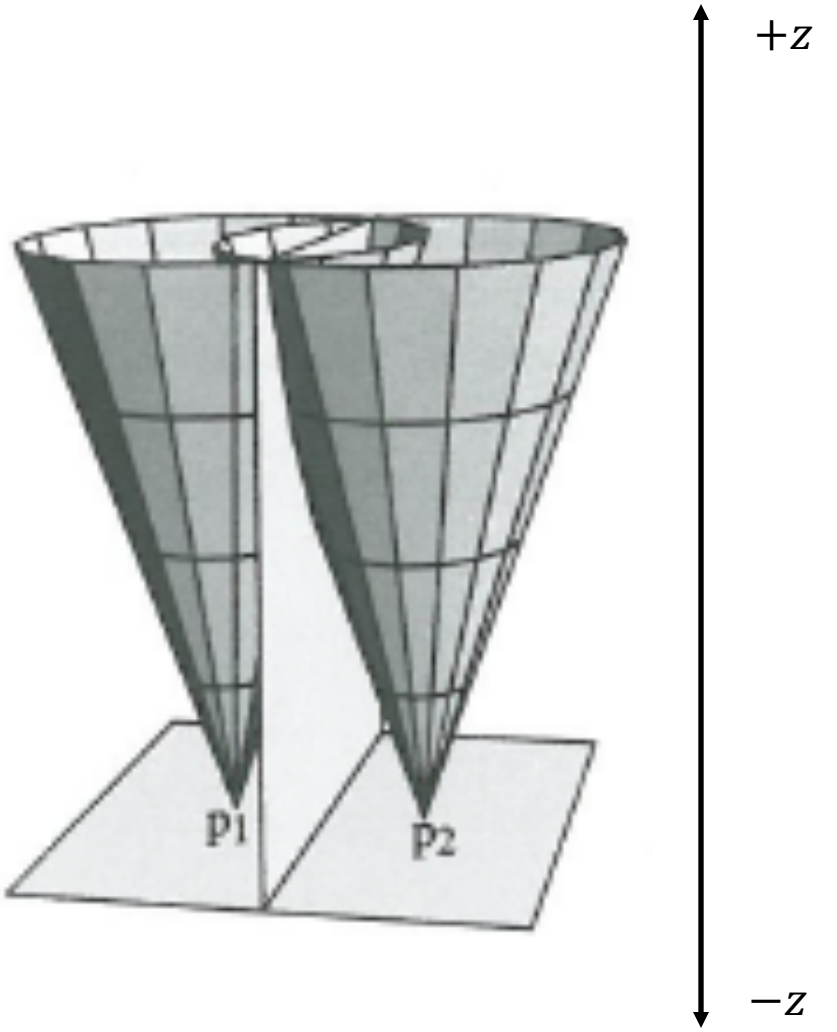
- The strategy in a plane sweep algorithm is to sweep a horizontal line (sweep line) (L) from top to bottom
  - Information maintained about the part of the Voronoi diagram of the sites above L that cannot be changed by sites below L

Fortune's Algorithm

- The algorithm maintains both a *sweep line* and a *beach line*, which both move through the plane as the algorithm progresses.
  - Horizontal sweep line maintains order of construction
  - Beach line maintains portion of diagram, which cannot change due to sites below sweep line

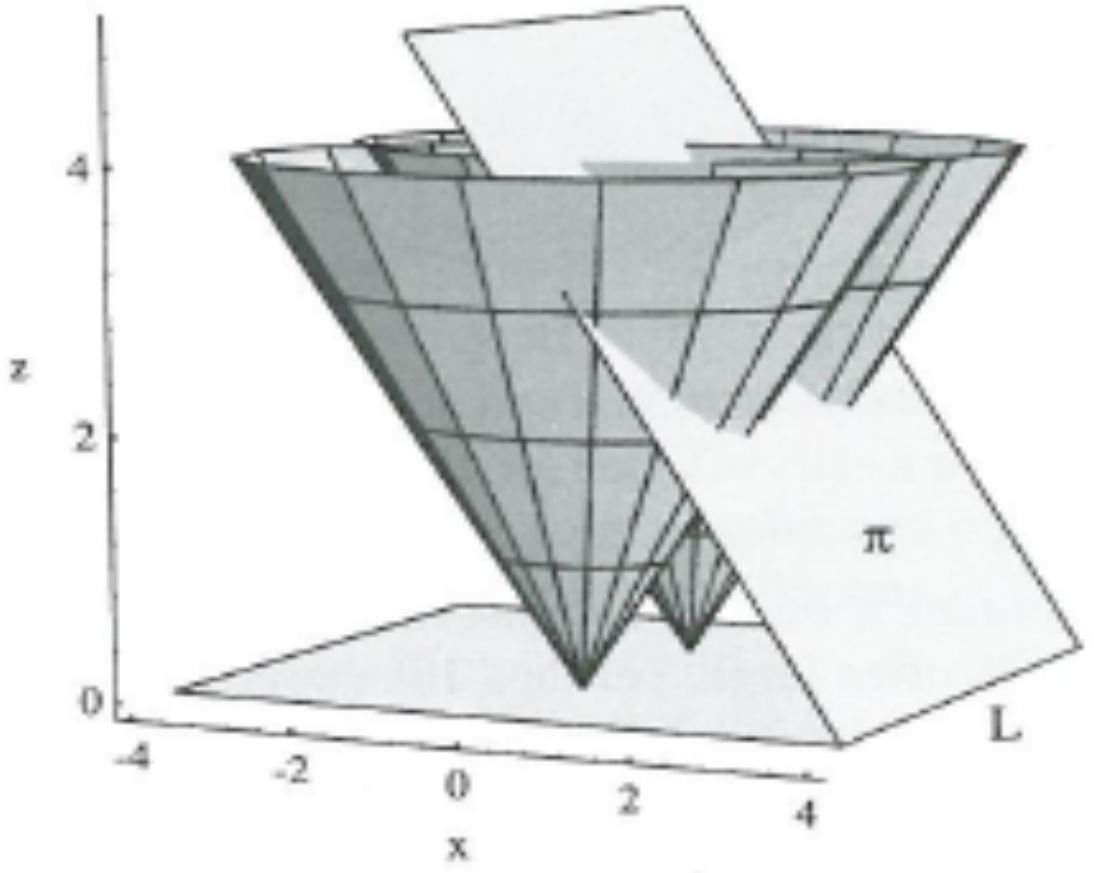
FORTUNE'S ALGORITHM

- CONSIDER TWO NEARBY CONES
- IF CONES OVER ALL SITES ARE OPAQUE, AND THEY ARE VIEWED FROM $z \to -\infty$, WHAT IS SEEN IS PRECISELY THE VORONOI DIAGRAM



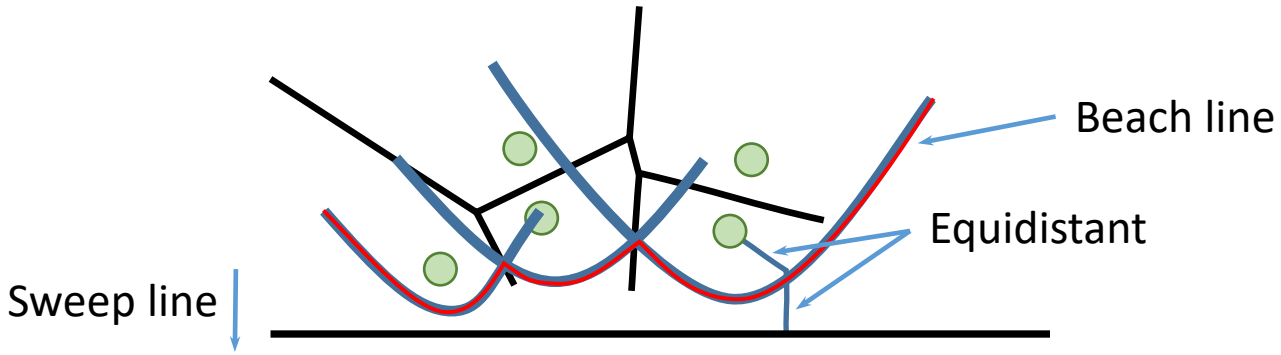
FORTUNE'S ALGORITHM

- FORTUNE'S IDEA: SWEEP THE CONES WITH A PLANE π, SLANTED AT 45° TO THE XY-PLANE



Beach line

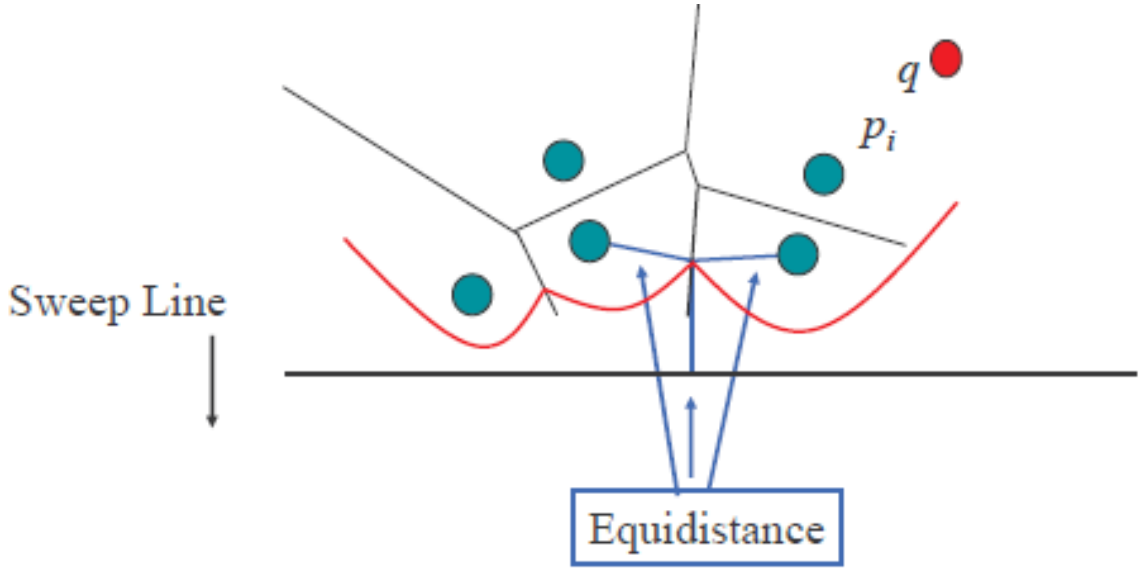
- Which points are closer to a site above the sweep line than to the sweep line itself?



- The set of parabolic arcs form a beach-line that bounds the locus of all such points
- The beach line is monotone

EDGES

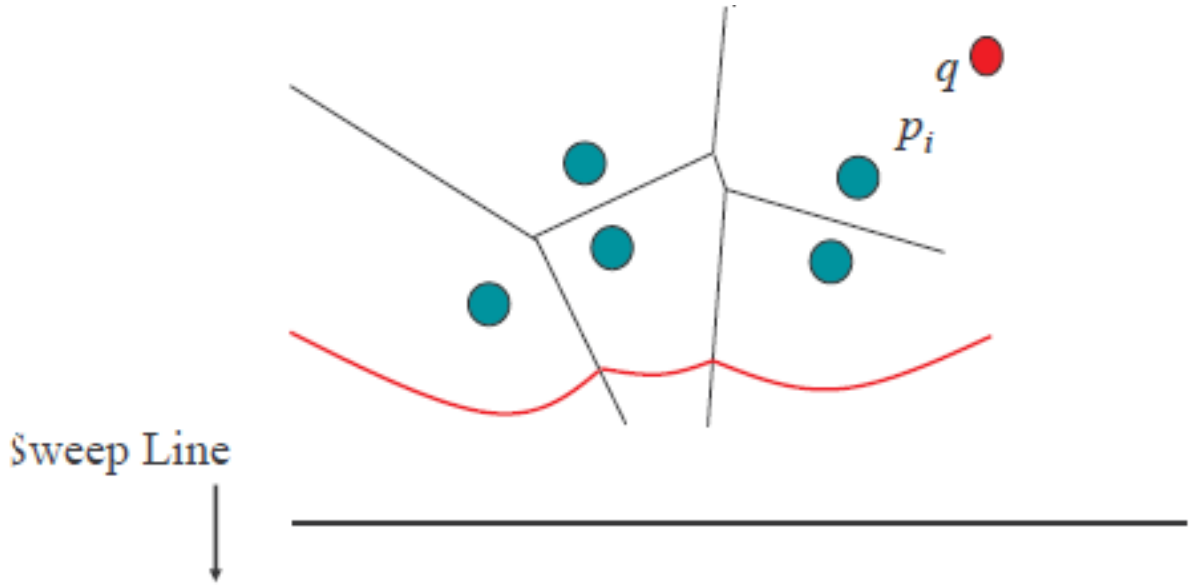
- BREAKPOINTS TRACE OUT VORONOI EDGES.



- THE ONLY WAY IN WHICH A NEW ARC CAN APPEAR ON THE BEACH LINE IS THROUGH A SITE EVENT

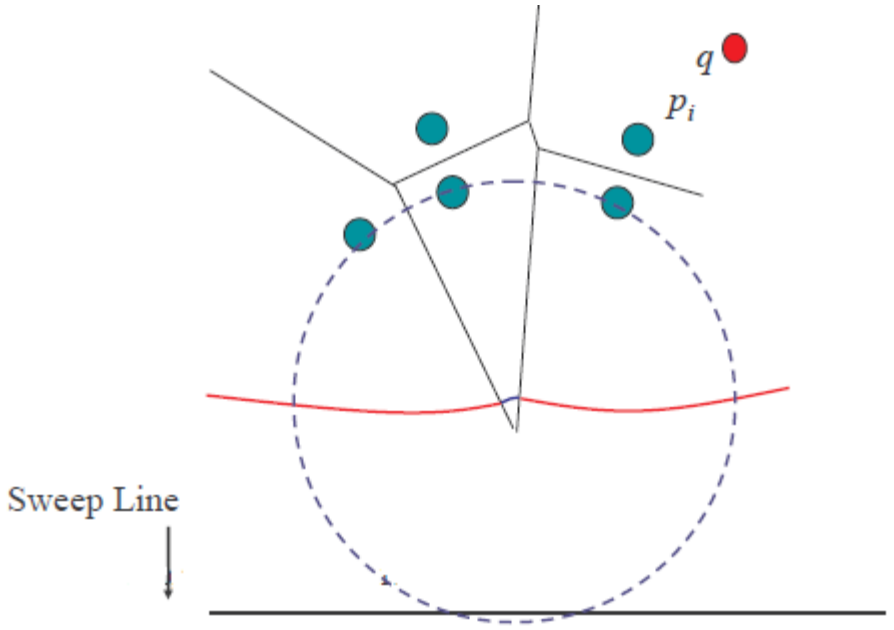
EDGES

- ARCS FLATTEN OUT AS SWEEP LINE MOVES DOWN.



EDGES

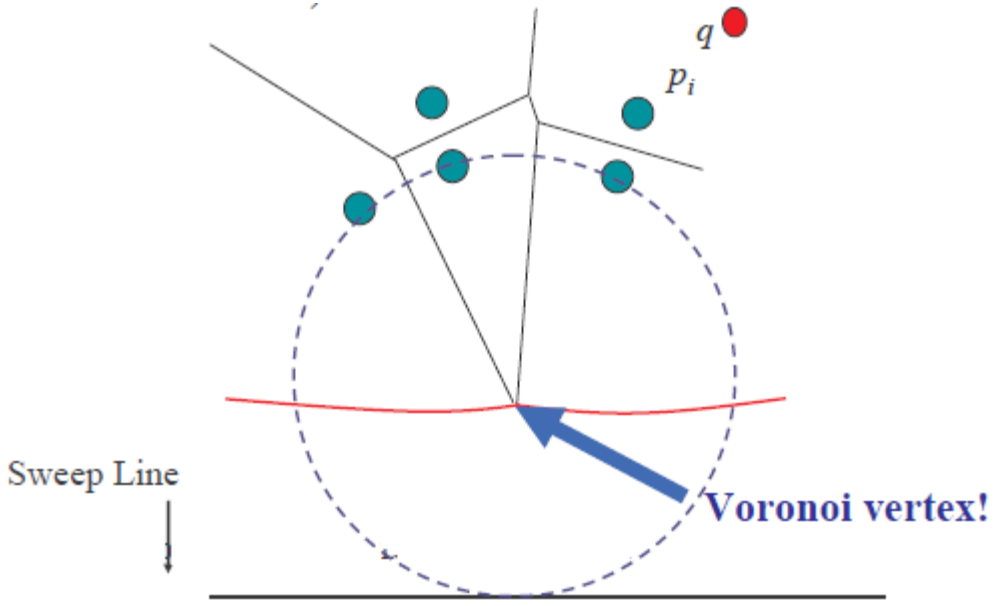
- EVENTUALLY, THE MIDDLE ARC DISAPPEARS.



- THE ONLY WAY IN WHICH AN EXISTING ARC CAN DISAPPEAR FROM THE BEACH LINE IS THROUGH A CIRCLE EVENT

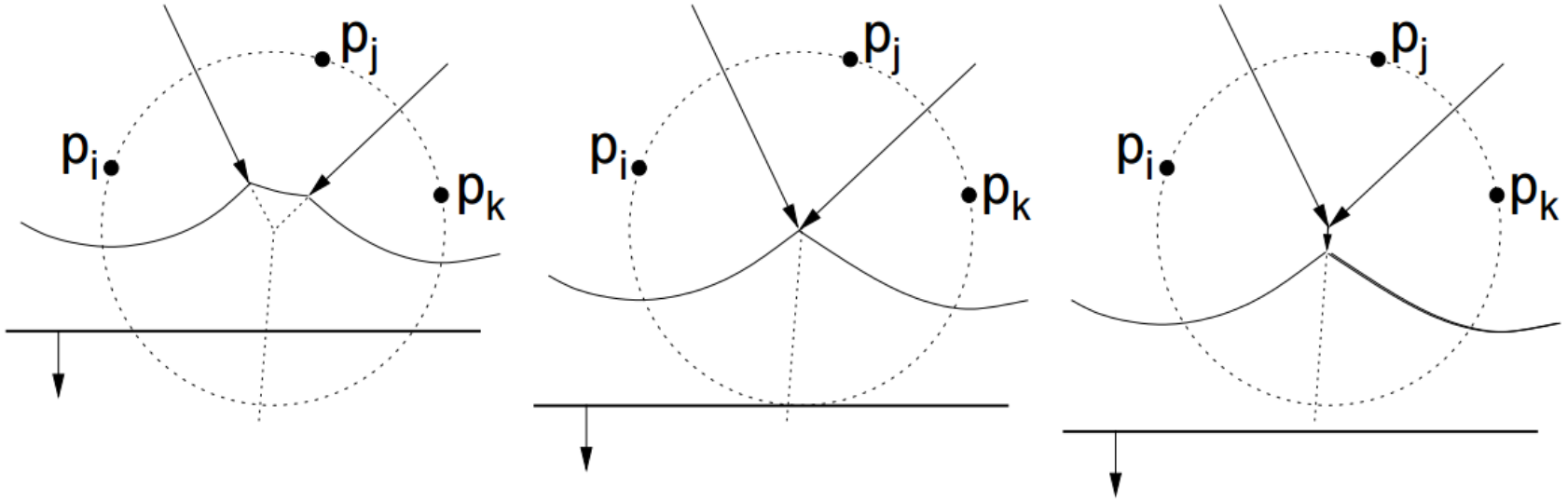
Circle Event

- We have detected a circle that is empty (contains no sites) and touches 3 or more sites.



Circle Event

- We have detected a circle that is empty (contains no sites) and touches 3 or more sites when the sweep line passes the lowest point in the circle.



Beach Line Properties

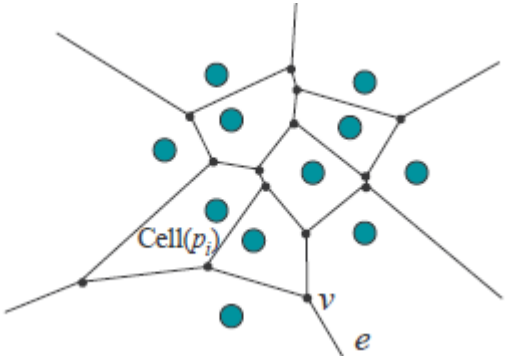
- Voronoi edges are traced by the break points as the sweep line moves down.
  - Emergence of a new break point(s) (from formation of a new arc or a fusion of two existing break points) identifies a new edge
  - Voronoi vertices are identified when two break points meet (fuse).
  - Decimation of an old arc identifies new vertex

Data Structures

- Current state of the Voronoi diagram
  - Doubly-linked list
- The beach line is represented by a balanced binary search tree
- The event queue is implemented as priority queue
  - Priority event queue sorted on decreasing y-coordinate

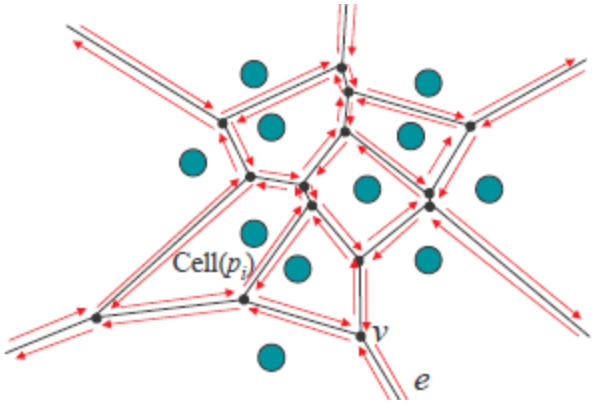
Doubly Linked List (*D*)

- Goal: a simple data structure that allows an algorithm to traverse a Voronoi diagram's segments, cells and vertices



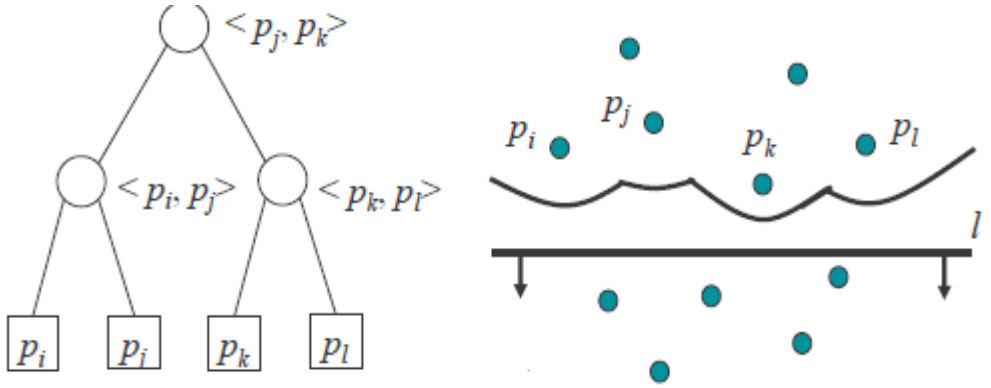
Doubly Linked List (*D*)

- Divide segments into unidirectional half-edges
- A chain of counter-clockwise half-edges forms a cell
- Define a half-edge's "twin" to be its opposite half-edge of the same segment



Balanced Binary Tree

- A balanced binary search tree is used to maintain the status of the beach line.
- Internal nodes represent break points between two arcs
  - Also contains a pointer to the *D* record of the edge being traced
- Leaf nodes represent arcs, each arc is in turn represented by the site that generated it
  - Also contains a pointer to a potential circle event

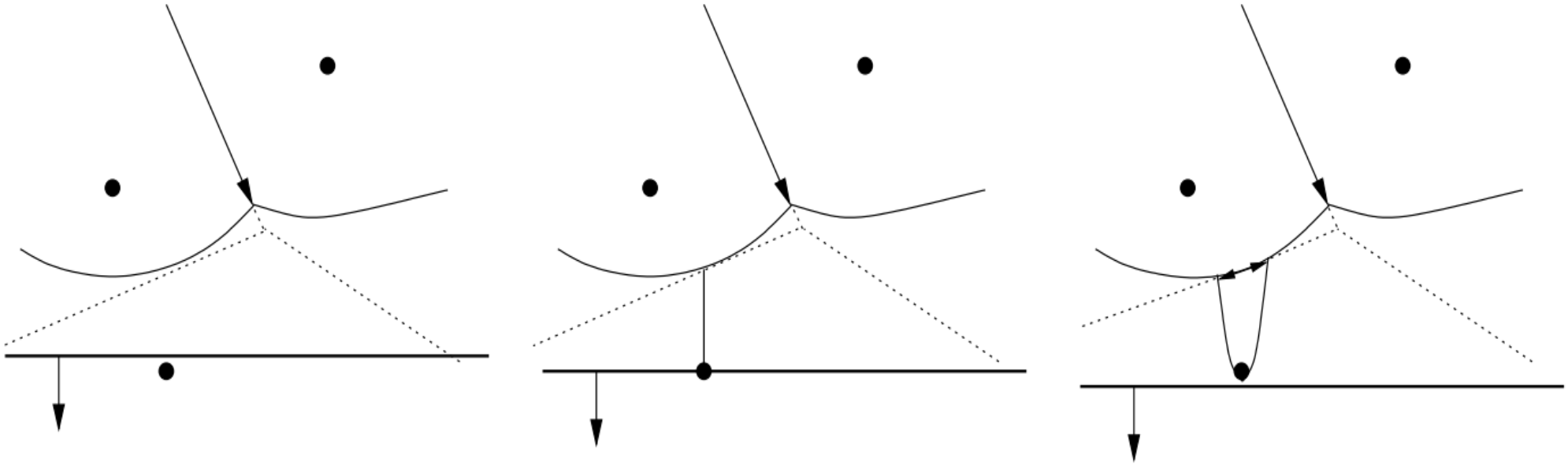


Event Queue (*Q*)

- An event is an interesting point encountered by the sweep line that makes discrete stops at events as it sweeps from top to bottom
- Consists of:
  - Site Events – when the sweep line encounters a new site point
  - Circle Events – when the sweep line encounters the *bottom* of an empty circle touching 3 or more sites
- Events are prioritized based on y-coordinate

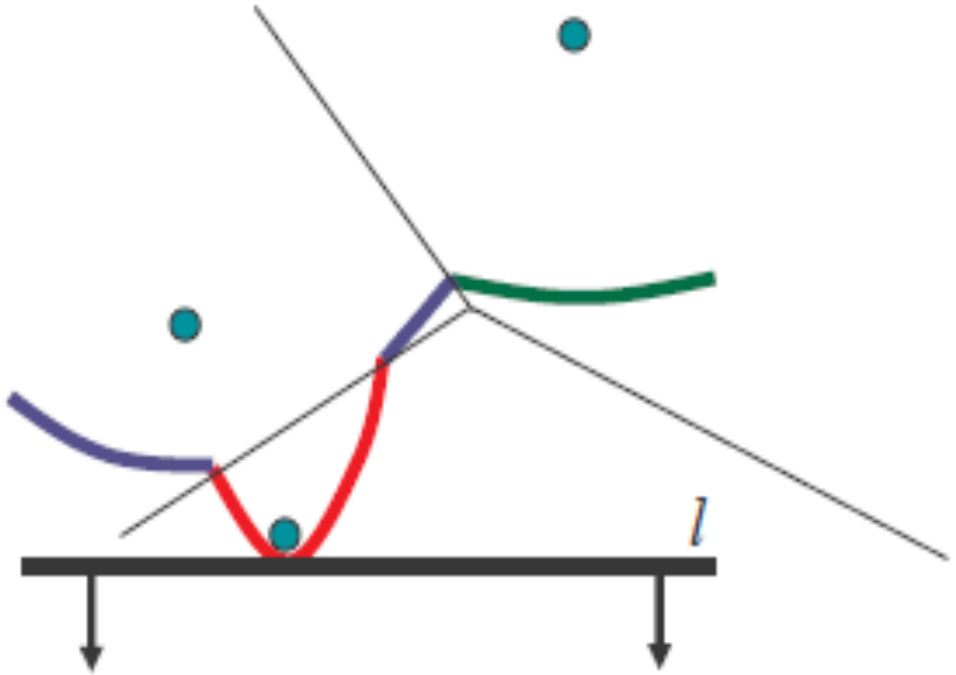
Site Event

- A new arc appears when a new site appears.



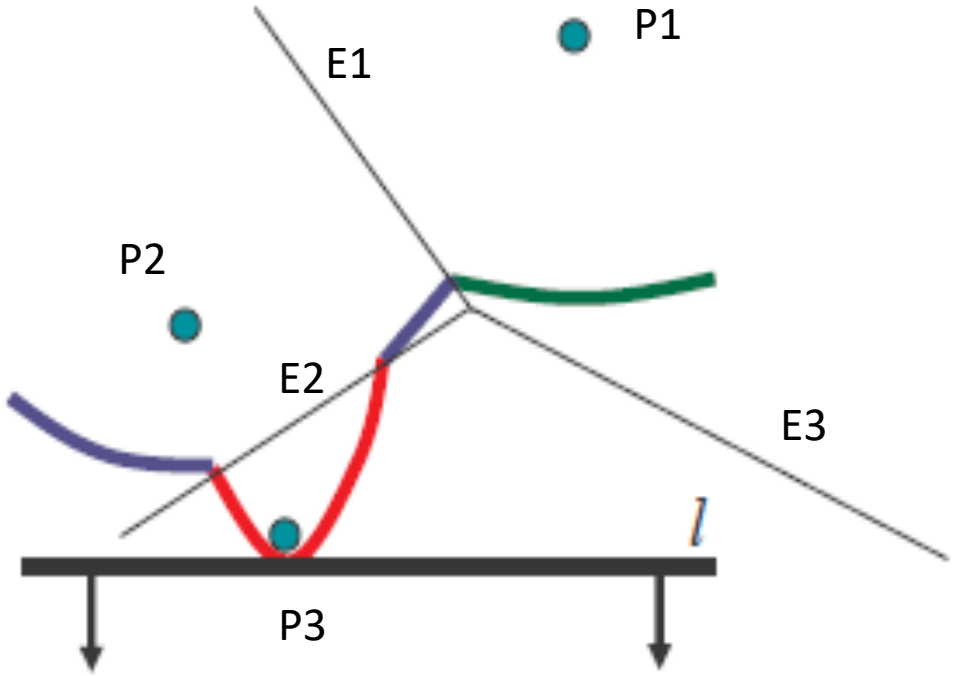
Site Event

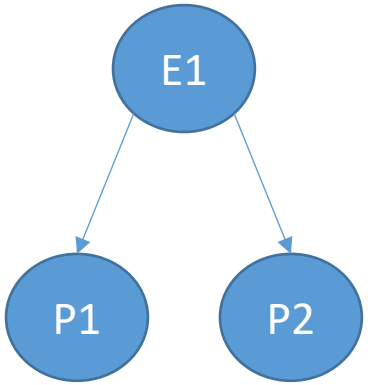
- Original arc above the new site is broken into two arcs
  - Number of arcs on beach line is O(n)

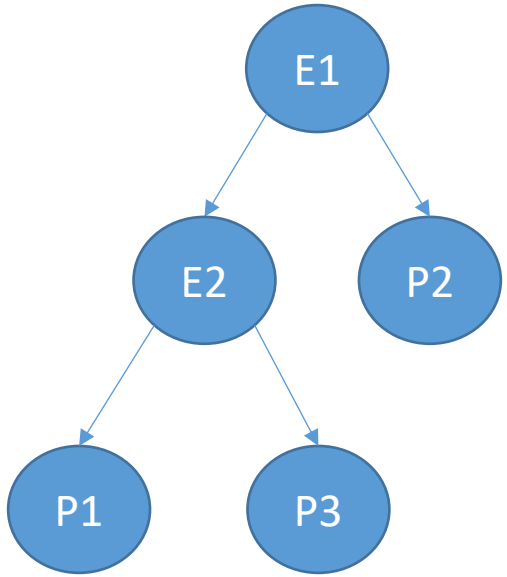


Site Event

- Effects on Binary Tree

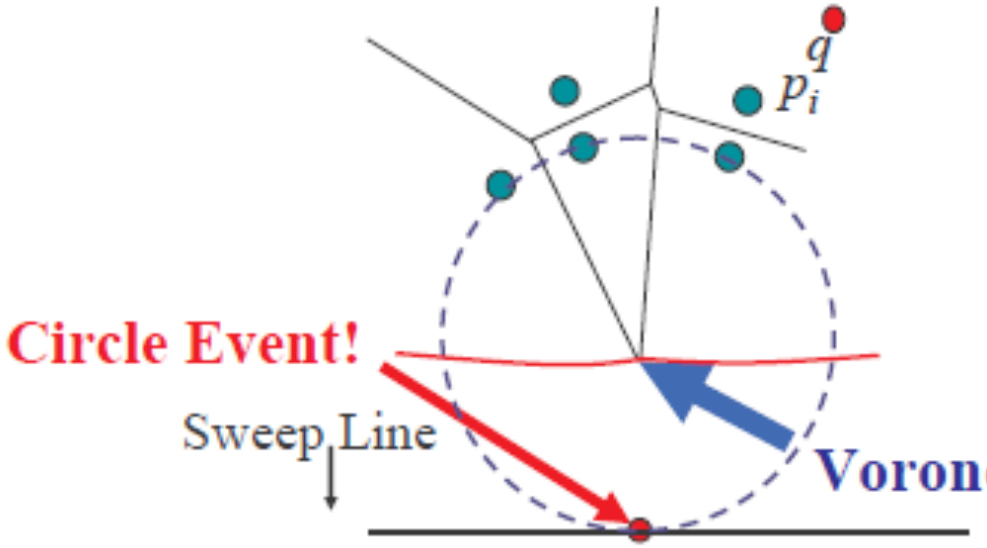






CIRCLE EVENT

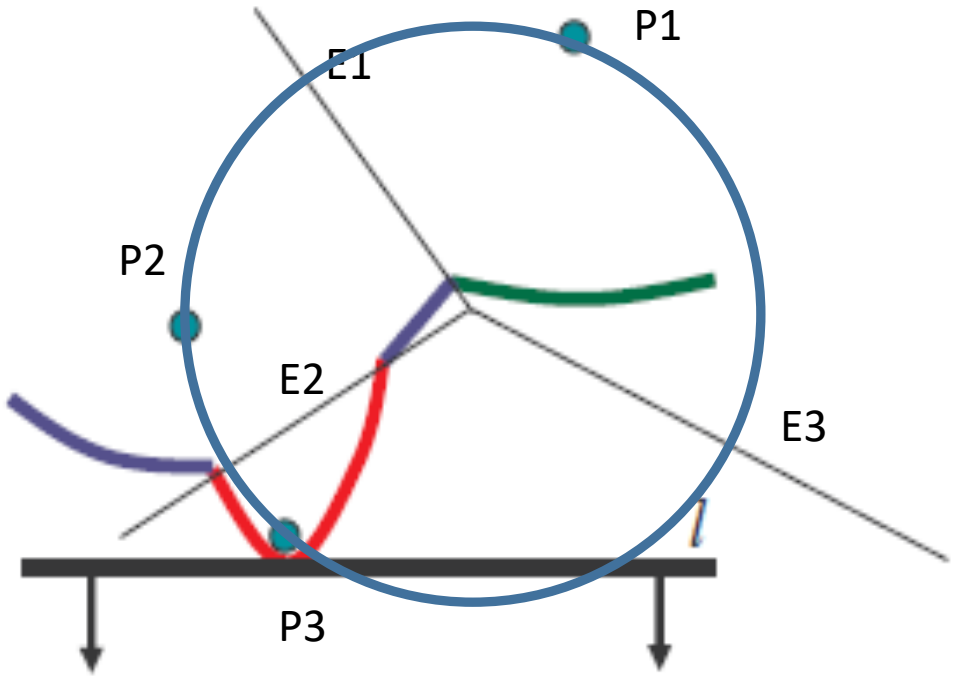
- AN ARC DISAPPEARS WHENEVER AN EMPTY CIRCLE TOUCHES THREE OR MORE SITES AND IS TANGENT TO THE CIRCLE.

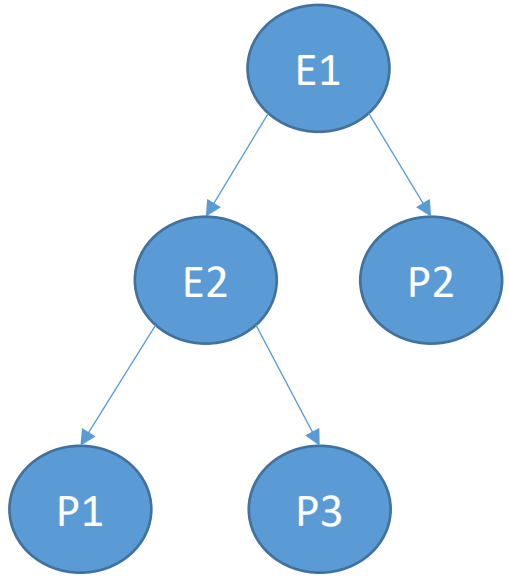


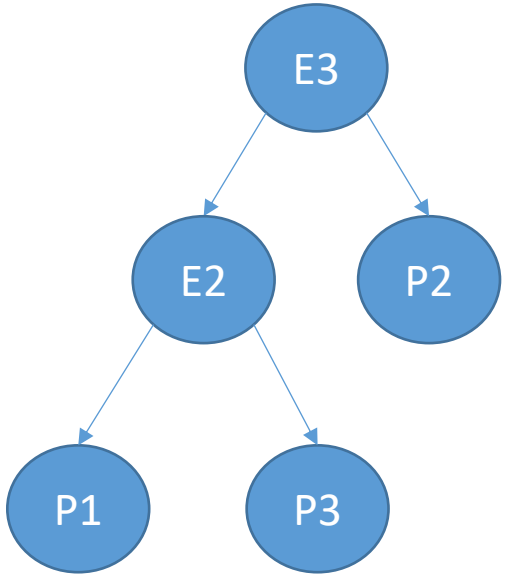
- SWEEP LINE HELPS DETERMINE THAT THE CIRCLE IS INDEED EMPTY.

Circle Event

- Effects on Binary Tree
  - Other circle events will remove nodes from the tree



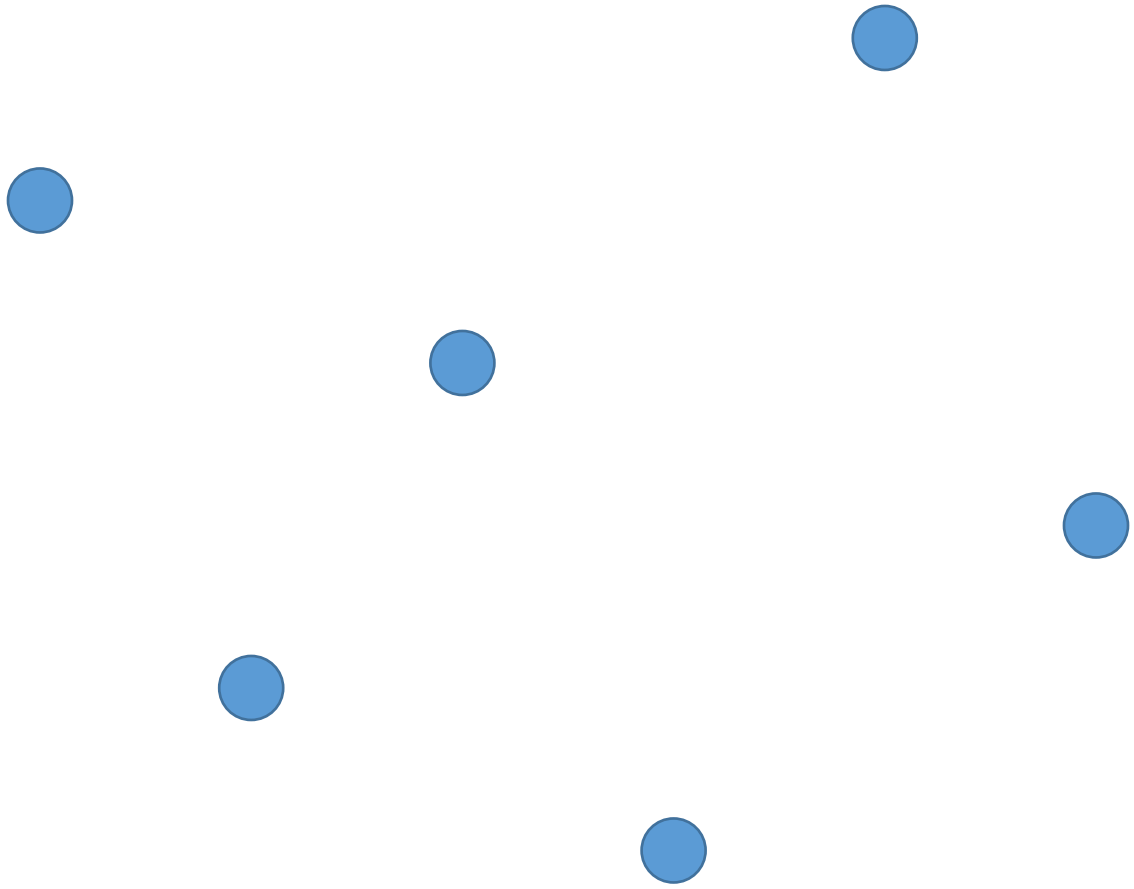




Event Queue Summary

- Site Events are
  - given as input
  - represented by the (x,y)-coordinate of the site point
- Circle Events are
  - represented by the (x,y)-coordinate of the lowest point of an empty circle touching three or more sites
  - computed on the fly (intersection of the two bisectors in between the three sites)
  - "anticipated": these newly generated events may represented by the (x,y)-coordinate of the lowest point of an empty circle touching three or more sites; they can be false and need to be removed later
- Event Queue prioritizes events based on their y-coordinates

EXAMPLE



Summarizing Data Structures

- Current state of the Voronoi diagram
  - Doubly linked list of half-edges, vertices, cell records
- Current state of the beach line
  - Keep track of break points
- Inner nodes of binary search tree; represented by a tuple
  - Keep track of arcs currently on beach line
- Leaf nodes of binary search tree; represented by a site that generated the arc
- Current state of the sweep line
  - Priority event queue sorted on decreasing y-coordinate

Degenerate Cases

- Events in Q share the same y-coordinate
  - Can additionally sort them using x-coordinate
- Circle event involving more than 3 sites
  - Current algorithm produces multiple degree 3 Voronoi vertices joined by zero-length edges
  - Can be fixed in post processing

Degenerate Cases

- Site points are collinear (break points neither converge or diverge)
  - Bounding box takes care of this
- One of the sites coincides with the lowest point of the circle event
  - Do nothing

Handling Site Events

1. Locate the leaf representing the existing arc that is above the new site • Delete the potential circle event in the event queue	O(N log N)
2. Break the arc by replacing the leaf node with a sub tree representing the new arc and break points	O(1)
3. Add a new edge record in the link list	O(1)
4. Check for potential circle event(s), add them to queue if they exist • Store in the corresponding leaf of T a pointer to the new circle event in the queue	O(1)

Handling Circle Events

1. Delete from T the leaf node of the disappearing arc and its associated circle events in the event queue	O(N log N)
2. Add vertex record in doubly link list	O(1)
3. Create new edge record in doubly link list	O(1)
4. Check the new triplets formed by the former neighboring arcs form potential circle events	O(1)

Total Running Time

- Each new site can generate at most two new arcs beach line can have at most $2N - 1$ arcs
- Each "false circle event" can be changed to a real event O(N) events
- Site/Circle Event Handler O(log N)

O(N log N) total running time