

Kinetic Algorithms

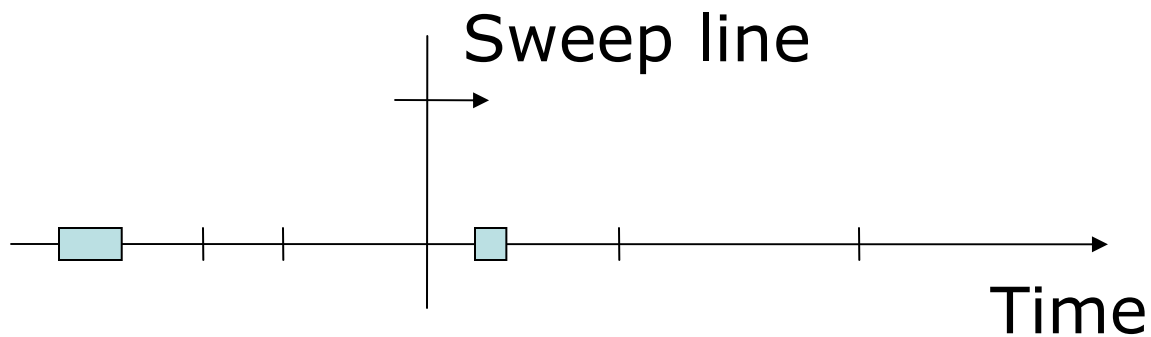
Data Structures for Mobile
Data

Introduction

- Key concepts are that geometric constructs (convex hull, delauney triangulation, etc.) are defined based on "simple" geometric properties. The simple properties can be verified with certificates. Idea of kinetic algorithms is to process and update these certificates as points move and the certificates fail.
- (hopefully, not so many failures)
- (hopefully, easy to fix)
- If detecting/fixing failures is as expensive as re-processing, don't use kinetic algorithms.

Sweep line in time?

- Resemble to a sweep-line/plane algorithm
- Use of a global event queue as interface between KDS and object in motion.

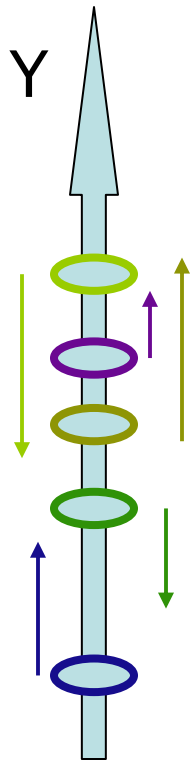


Object motions

- Assume:
 - The motion of the objects is known in advance
 - If constant, can draw out objects in x, y, t and answer questions about 3d shape (that expresses all positions through time).
 - If not constant, need to use a "sweep line" that accepts changes in the future.

Sample 1

- Consider the following 1D situation
 - Given a set of n points moving continuously along the y -axis, we are interested to know which is the topmost point.
 - Constant speed
 - Arbitrary initial configuration



Solution 1

(batch processing)

- Draw the lines in the ty -plane
- Compute the upper envelope of the set of lines. $O(n \log(n))$

- What does "a point changing how it moves" look like?

Solution 2

Schedule-Deschedule

- Maintain, on-line, the sorted order along the y-axis
- Schedule event that is the first time when two consecutive points cross.
 - Destroy and create 2 adjacencies
 - Schedule and deschedule up to 2 new events
- Not efficient: process up to n^2 events.

Solution 3

Kinetic Tournament

- Consider a divide-and-conquer algorithm, like a tournament from bottom to up for computing a global leader. $O(n)$ comparisons
- As long as each of the comparison remain valid the identity of the maximum remains valid.
- These comparisons are "certificates".

Kinetic tournament?

- How many certificates do we keep?
- What are the events?
- What event processing do we need?

Complexity

- Assume pts maintain speed.
- How many events are there?
 - different kinds of events...
 - new "max"
 - and internal certificate failure.

Keys to Kinetic Algorithms

- "Simple Certificates"
- Not too many
- Local (1 certificate failure should lead to few new tests)

- Kinetic Delaunay Triangulation?

- What events?

- How many
 - (n^2n^3) .

- Kinetic collision detection?
 - 2D collision detection
 - Ignore one object inside another