Plane Sweep

Gaurav Saxena MCS062260

October 4, 2006

Abstract

Problem Statement: Given n line segements s_i,$i \leq n$,report all the pairwise intersections. Naive approarch would be to take every pair of line and check its intersection. But its complexity would be O(n*n). So we prefer to do it with line line sweep method which has complexity of O((I +2n)logn).

1 Line sweep method:

1. Take a vertical line left to the leftmost point.
2. Move it towards the segments and during this movement, we maintain the list of segments that intersects the vertical line .
3. Also the ordering of intersections is maintained. Ordering in the sense that a particular line segment is above / below a line segment.
4. The moment the vertical line crosses the intersection piont the ordering is changed. The lines which are swapped must be consecutive as those the only one that intersects.
5. The moment any two lines become consecutive they can potentially intersect.

2 lemma

Given two segment S_i and s_j, which interest in a single piont p(and assuming no other line segment passes through this piont) there is a placement of the line sweep prior to this event, such that s_i and s_j are adjacent along the sweep line(and hence will be tested for intersection.)

2.1 Events we track

1. Intersections can flip ordering.
2. Some line segments can begin.
3. Some line segments can end.

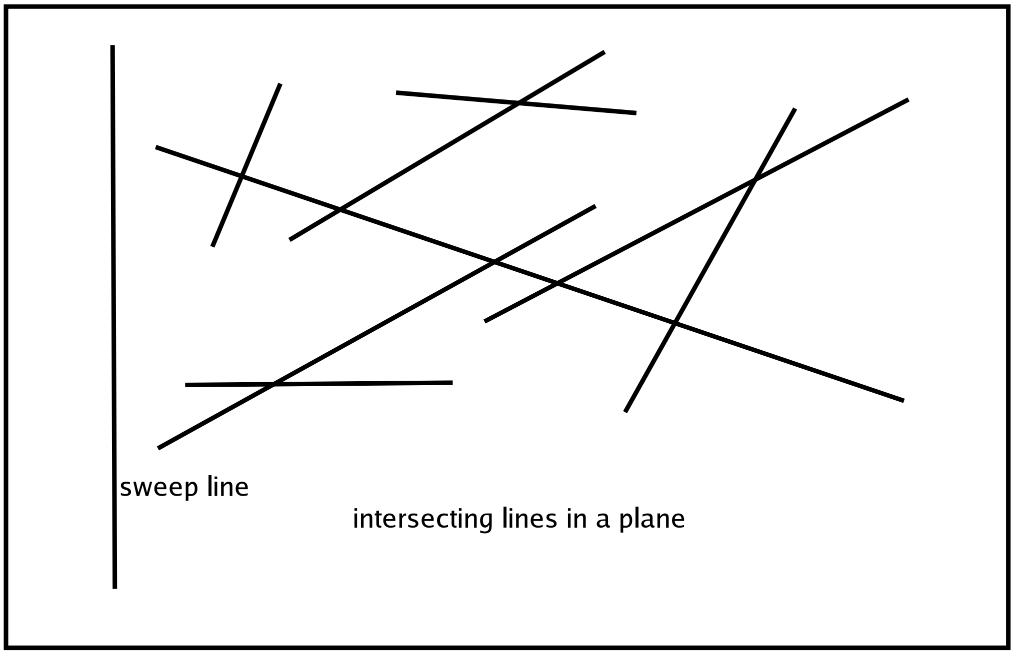

Figure 1:

2.2 Data structures for sweep line

To store the sweep line status , we maintain a balanced binary tree or a skip list whose entries are pionters to the line segment, store in decreasing orderof y coordinate along the current sweep line. Since the sweep line varies, we need 'variable' key.
To do this, the key value in each node of the tree is a pionter to a line segment. To compute the y coordinate of some segment at the location of the current sweep line, we simply take the current x coordinate of the sweep line and plug it into the line equation for this line.
The operations that we need tosupport are those of deleting a line segment , inserting a line segment , swapping the position of two line segment and determining the immediate predecessor and successor of any item. Assuming any balanced binary tree or a skiplist, these operations can be performed in O(logn) time each.

2.3 complexity

By ordering endpionts we can know events 2 and 3 in priori. We maintain an event queue ordered on x coordinate.Before intersection. 2 lines must consecutive and this happens only when an event has occurs. If two lines become consecutive , we check for an intersection and add the intersection if it exits to the event queue. So the over all complexity is O((I + 2n)logn) Where I is the number of intersection pionts, 2n is the ordering of end pionts of n segments. And logn is there due to heap datastructures for events ordering on the sweep line.

Space complexity is O(n +I). But sometimes I is as large as O(n*n).