Robot Motion Planning:
Plane Sweep Algorithms

Jin Seob Kim, Ph.D.
Senior Lecturer, ME Dept., LCSR, JHU

1 Plane Sweep Algorithm

In this section, the general concept of the plane sweep algorithm is introduced. The pseudocodes below are from [1].

Note that, for our purpose, the set of all the vertices of C-obstacles (including the bounding box) are the only data of input. Sorting the points in this set in the increasing order by their x-coordinate is necessary. Then each visit to a point in this set is referred to as an *event*.

Algorithm 1 FindIntersections(S)

Input: A set S of line segments in the plane
Output: The set of intersection points among the segments in S, with for each intersection point the segments that contain it.

```
1: Initialize an empty event queue Q. Next, insert the segment endpoints into Q; when a left
   endpoint is inserted, the corresponding segment should be stored with it.
2: Initialize an empty status structure T.
3: while Q ≠ ∅ do:
4:     Determine the next event point p in Q and delete it.
5:     HandleEventPoint(p)
6: end while
```

Algorithm 2 $\mathsf{HandleEventPoint}(p)$

Let $\mathcal{L}(p)$ be the set of segments whose left endpoint is p; these segments are stored with the event point p.
Find all segments stored in T that contain p; they are adjacent in T. Let $\mathcal{R}(p)$ denote the subset of segments found whose right endpoint is p, and let $\mathcal{C}(p)$ denote the subset of segments found that contain p in their interior.
if $\mathcal{L}(p) \cup \mathcal{R}(p) \cup \mathcal{C}(p)$ contains more than one segment **then**:
    Report p as an intersection, together with $\mathcal{L}(p)$, $\mathcal{R}(p)$, and $\mathcal{C}(p)$.
end if
Delete the segments in $\mathcal{R}(p) \cup \mathcal{C}(p)$ from T.
Insert the segments in $\mathcal{L}(p) \cup \mathcal{C}(p)$ into T. The order of the segments in T should correspond to the order in which they are intersected by a sweep line just right to p.
if $\mathcal{L}(p) \cup \mathcal{C}(p) = \emptyset$ **then**:
    Let s_l and s_r be the left and right neighbors of p in T.
    $\mathsf{FindNewEvent}(s_l, s_r, p)$
else
    Let s' be the leftmost segment of $\mathcal{L}(p) \cup \mathcal{C}(p)$ in T.
    Let s_l be the left neighbor os s' in T.
    $\mathsf{FindNewEvent}(s_l, s', p)$
    Let s'' be the rightmost segment of $\mathcal{L}(p) \cup \mathcal{C}(p)$ in T.
    Let s_r be the right neighbor of S'' in T.
    $\mathsf{FindNewEvent}(s'', s_r, p)$
end if

Algorithm 3 $\mathsf{FindNewEvent}(s_l, s_r, p)$

if s_l and s_r intersect left to the sweep line, or on it and to the right of the current event point p, and the intersection is not yet present as an event in Q **then**:
    Insert the intersection point as an event into Q.
end if

2 Rotational Plane Sweep Algorithm

This algorithm, in algorithm 4, can be adapted in constructing the visibility graph. The pseudocode is adopted from [2].

Algorithm 4 Rotational Plane Sweep Algorithm

Input: A set of vertices $\{v_i\}$ (whose edges do not intersect) and a vertex v.
Output: A subset of vertices from $\{v_i\}$ that are within line of sight of v.

For each vertex v_i, calculate α_i, the angle from the horizontal axis to the line segment $\overline{vv_i}$.
Create the vertex list E, containing the α_i's sorted in increasing order.
Create the active list S, containing the sorted list of edges that intersect the horizontal half-line emanating from v.
for all α_i **do**:
  if v_i is visible to v **then**:
    Add the edge (v, v_i) to the visibility graph.
  end if
  if v_i is the beginning of an edge, E, not in S **then**:
    Insert the E into S.
  end if
  if v_i is the end of an edge in S **then**:
    Delete the edge from S.
  end if
end for

References

[1] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, 2nd edition, 2000.

[2] Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki, and Sebastian Thrun. *Princinples of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.