11. Visibility Graphs

Lecture on Thursday 6^{th} November, 2008 by Michael Hoffmann <hoffmann@inf.ethz.ch>

11.1 Sorting all Angular Sequences.

Theorem 11.1 *Consider a set P of n points in the plane. For a point $q \in P$ let $c_P(q)$ denote the circular sequence of points from $S \setminus \{q\}$ ordered counterclockwise around q (in order as they would be encountered by a ray sweeping around q). All $c_P(q)$, $q \in P$, collectively can be obtained in $O(n^2)$ time.*

Proof. Consider the projective dual P^* of P. An angular sweep around a point $q \in P$ in the primal plane corresponds to a traversal of the line q^* from left to right in the dual plane. (A collection of lines through a single point q corresponds to a collection of points on a single line q^* and slope corresponds to x-coordinate.) Clearly, the sequence of intersection points along all lines in P^* can be obtained by constructing the arrangement in $O(n^2)$ time. In the primal plane, any such sequence corresponds to an order of the remaining points according to the slope of the connecting line; to construct the circular sequence of points as they are encountered around q, we have to split the sequence obtained from the dual into those points that are to the left of q and those that are to the right of q; concatenating both yields the desired sequence. □

11.2 Segment Endpoint Visibility Graphs

A fundamental problem in motion planning is to find a short(est) path between two given positions in some domain, subject to certain constraints. As an example, suppose we are given two points $p, q \in \mathbb{R}^2$ and a set $S \subset \mathbb{R}^2$ of obstacles. What is the shortest path between p and q that avoids S?

Observation 11.2 *The shortest path between two points that does not cross a set of polygonal obstacles (if it exists) is a polygonal path whose interior vertices are obstacle vertices.*

One of the simplest type of obstacle conceivable is a line segment. In general the plane may be disconnected with respect to the obstacles, for instance, if they form a closed curve. However, if we restrict the obstacles to pairwise disjoint line segments then there is always a free path between any two given points. Apart from start and goal position, by the above observation we may restrict our attention concerning shortest paths to straight line edges connecting obstacle vertices, in this case, segment endpoints.

Definition 11.3 *Consider a set S of n disjoint line segments in $\mathbb{R}^2$. The* **segment endpoint visibility graph** *$\mathcal{V}(S)$ is a plane straight line graph defined on the segments endpoints. Two segment endpoints p and q are connected in $\mathcal{V}(S)$ if and only if*

- *the line segment* $\overline{pq}$ *is in* S *or*
- $\overline{pq} \cap s \subseteq \{p, q\}$ *for every segment* $s \in S$.

If all segments are on the convex hull, the visibility graph is complete. If they form parallel chords of a convex polygon, the visibility graph consists of copies of K_4, glued together along opposite edges and the total number of edges is linear only. These graphs are Hamiltonian. :-)

Constructing $\mathcal{V}(S)$ for a given set S of disjoint segments in a brute force way takes $O(n^3)$ time. (Take all pairs of endpoints and check all other segments for obstruction.)

Theorem 11.4 (Welzl 1985) *The segment endpoint visibility graph of* n *disjoint line segments can be constructed in worst case optimal* $O(n^2)$ *time.*

Proof. We have seen above how all sorted angular sequences can be obtained from the dual line arrangement in $O(n^2)$ time. Topologically sweep the arrangement from left to right (corresponds to changing the slope of the primal rays from $-\infty$ to $+\infty$) while maintaining for each segment endpoint p the segment $s(p)$ it currently "sees" (if any). Initialize by brute force in $O(n^2)$ time (direction vertically downwards). Each intersection of two lines corresponds to two segment endpoints "seeing" each other along the primal line whose dual is the point of intersection. In order to process an intersection, we only need that all preceding (located to the left) intersections of the two lines involved have already been processed. This order corresponds to a topological sort of the arrangement graph where all edges are directed from left to right. A topological sort can be obtained, for instance, via (reversed) post order DFS in linear time.

When processing an intersection, there are four cases. Let p and q be the two points involved such that p is to the left of q.

1. The two points belong to the same input segment $\rightarrow$ output the edge pq, no change otherwise.
2. q is obscured from p by $s(p) \rightarrow$ no change.
3. q is endpoint of $s(p) \rightarrow$ output pq and update $s(p)$ to $s(q)$.
4. Otherwise q is endpoint of a segment t that now obscures $s(p) \rightarrow$ output pq and update $s(p)$ to t.

Thus any intersection can be processed in constant time and the overall runtime of this algorithm is quadratic. $\square$