Accepted paper with abstract, SoCG'09

Mashhood Ishaque, Bettina Speckmann and Csaba Toth

Shooting Permanent Rays Among Disjoint Polygons in the Plane

We present a data structure for ray shooting and insertion in the free space among disjoint polygonal obstacles with a total of n vertices in the plane. The portion of each query ray between the starting point and the first obstacle hit is inserted permanently as a new obstacle. Our data structure uses $O(n \log n)$ space and preprocessing time, and it supports m successive ray shooting and insertion queries in $O(n \log^2 n + m \log m)$ total time. We present two applications for our data structure:

(1) Our data structure supports efficient implementation of *auto-partitions* in the plane, that is, binary space partitions where each partition is done along a supporting line of an input segment. If n input line segments are fragmented into m pieces by an auto-partition, then it can now be implemented in $O(n \log^2 n + \mathrm{m} \log m)$ time. This improves the expected runtime of Patersen and Yao's classical randomized auto-partition algorithm for n disjoint line segments in the plane to $O(n \log^2 n)$.

(2) If we are given disjoint polygonal obstacles with a total of n vertices in the plane, a permutation of the reflex vertices, and a half-line at each reflex vertex that partitions the reflex angle into two convex angles, then the convex partitioning algorithm draws a ray emanating from each reflex vertex in the prescribed order in the given direction until it hits another obstacle, a previous ray, or infinity. The previously best implementation (with a semi-dynamic ray shooting data structure) requires $O(n^{3/2 - \varepsilon/2})$ time using $O(n^{1 + \varepsilon})$ space. Our data structure improves the runtime to $O(n \log^2 n)$.