

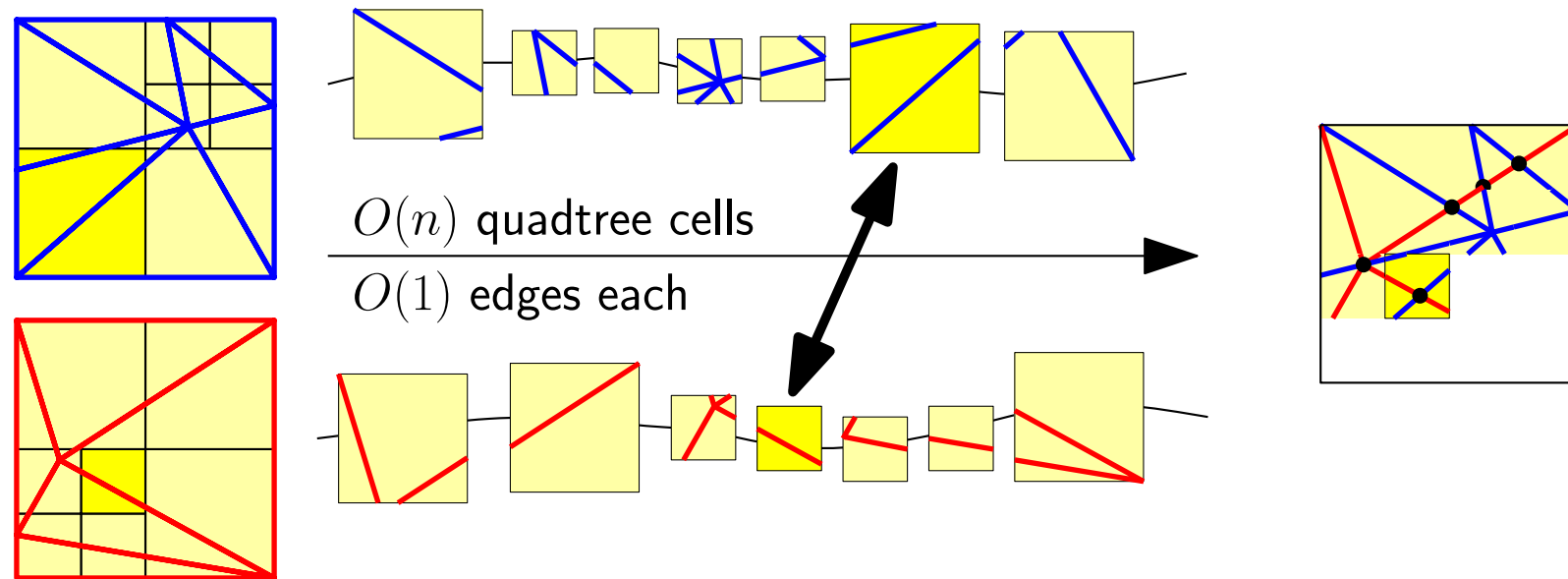
I/O-Efficient Map Overlay and Point Location in Low-Density Subdivisions

Mark de Berg

Herman Haverkort

Shripad Thite

Laura Toma



n = input size; M = main memory size; B = disk block size; $scan(n) < sort(n) \ll n$

For low-density triangulations / sets of line segments*, there is a data structure that supports:

- map overlay in $O(scan(n))$ I/O's;
- range queries in $O(\frac{1}{\epsilon}(\log_B n) + scan(k_\epsilon))$ I/O's;
- point location in $O(\log_B n)$ I/O's;
- (triangulations only) updates in $O(\log_B n)$ I/O's;

The data structures are built with $O(sort(n))$ I/Os.

*) for any circle C , number of intersecting segments bigger than $diam(C)$ is at most a constant