Our results

$n =$ input size;
$M =$ main memory size;
$B =$ disk block size

$$scan(n) = \frac{n}{B} \quad < \quad sort(n) = \frac{n}{B} \log_{M/B} \frac{n}{B} \quad << \quad n$$

Previously:

- Arge et al.: map overlay in $O(sort(n) + k/B)$ I/O's (complicated, super-linear space)
- Crauser et al.: randomized, linear space

Our results: in $O(sort(n))$ I/O's we can build a data structure that supports:

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

Condition: input must be *fat* triangulation (all angles $>$ positive constant), or a *low-density* set of segments (for any circle C, #intersecting segments $> \mathrm{diam}(C)$ is $O(1)$)