A Scalable DCEL implementation

Andres Calderon

University of California, Riverside

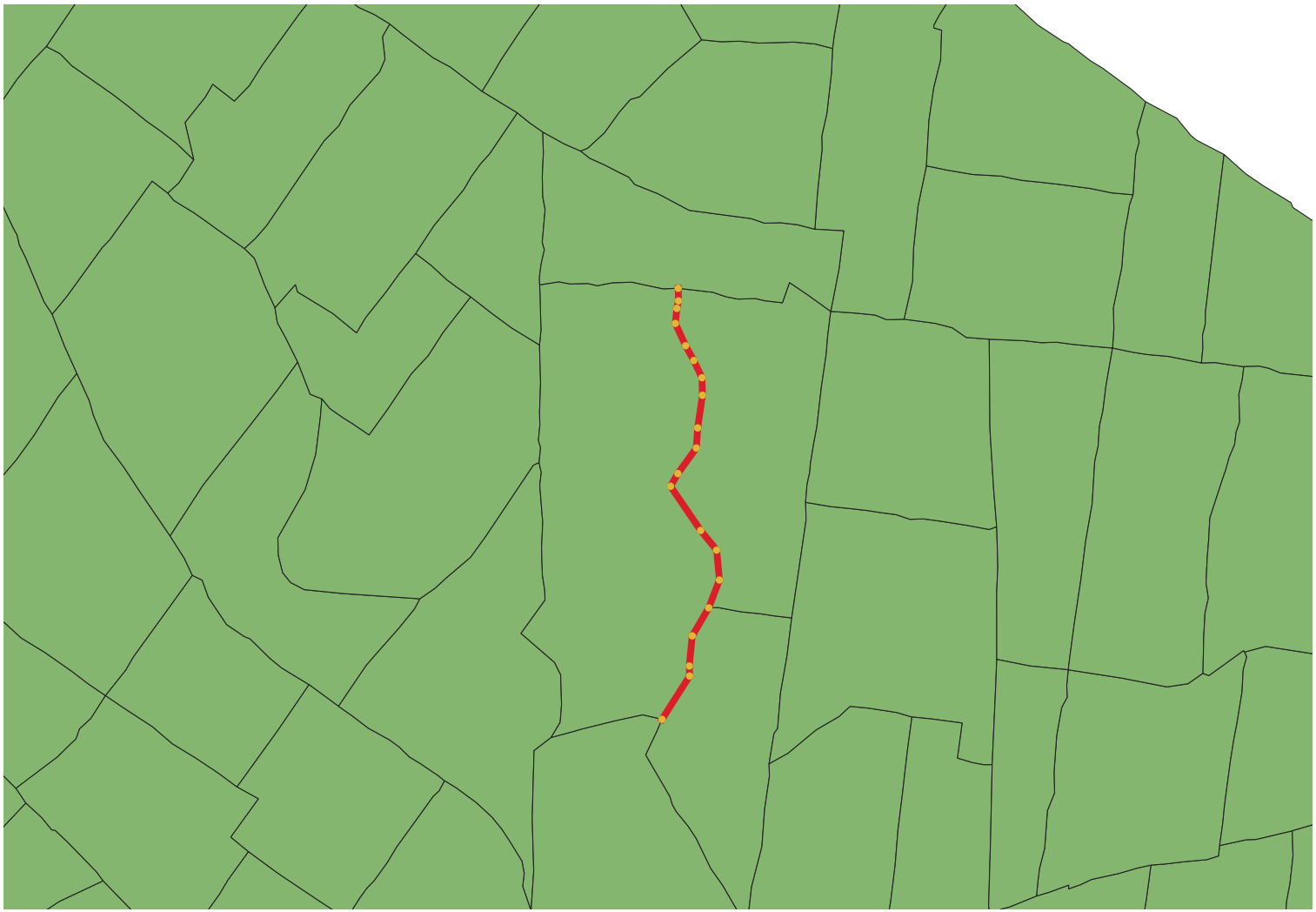
April 24, 2020

Improvements on CGAL implementation...

- From *CGAL Arrangements and Their Applications: A Step-by-Step Guide...*
  - Inserting segments which are disjoint from all of those present in the existing arrangement is quite straightforward...
  - However, "Inserting a curve[1] that intersects with the curves already in the arrangement is much more complicated and requires the application of nontrivial geometric algorithms. (Section 2.2.2 Modifying the Arrangement, pg 26)"

[1] A continuous curve c in $\mathbb{R}^2$ is called x-monotone, if every vertical line intersects it at a single point at most.

Improvements on CGAL implementation...



Improvements on CGAL implementation...

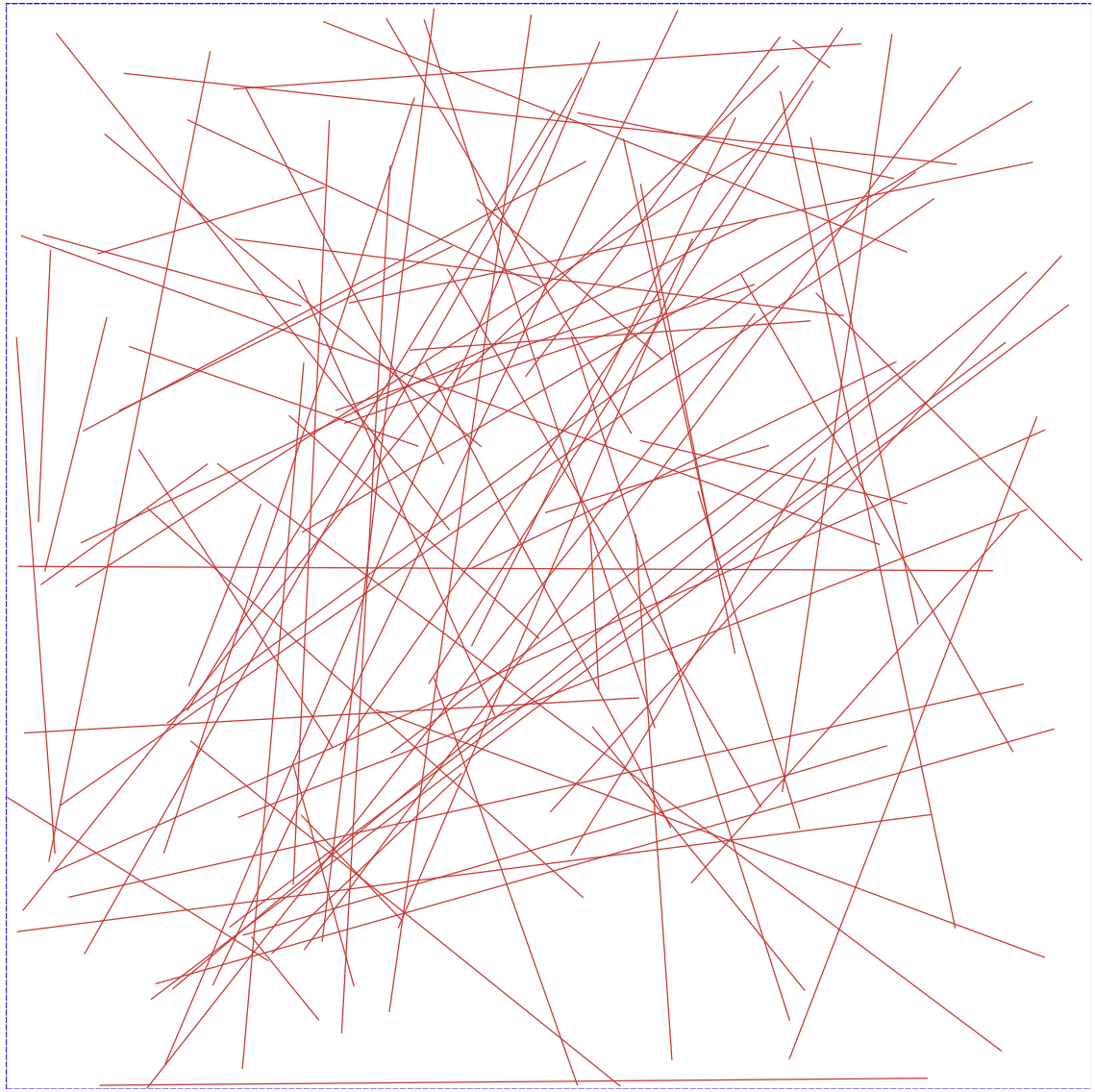
- From *CGAL Arrangements and Their Applications: A Step-by-Step Guide...*
  - CGAL provides two strategies for insertion of intersecting edges: Incremental (I am using this one) and Aggregate.
  - But, "(...) in practice it is recommended that **the aggregate construction process be used** even for dense arrangements.(Section 3.4.2 Aggregate Insertion Functions, pg 55)"
  - Aggregate insertion functions do not issue any point-location queries. The book states: "there is a trade-off between construction time and query time in each of the point-location strategies, which affects the running times of the incremental insertion process."

Improvements on CGAL implementation...

- Indeed, from [1] ...
  - Incremental insertion is based on the zone framework. It can take $\mathcal{O}(n^2)$.
  - Aggregate insertion is based on the plane-sweep framework. It takes $\mathcal{O}((k+n)(\log n))$ where k is the number of intersections between the segments.

[1] `https://tinyurl.com/y8m5dqwj`

Working on random edge generator...



What's next...

- Currently working on the aggregate strategy to read layer of polygons...
- Benchmark the new implementation with datasets from the random edge generator...