Good Approximations for the Relative Neighbourhood Graph

Diogo Vieira Andrade (PUC-Rio)

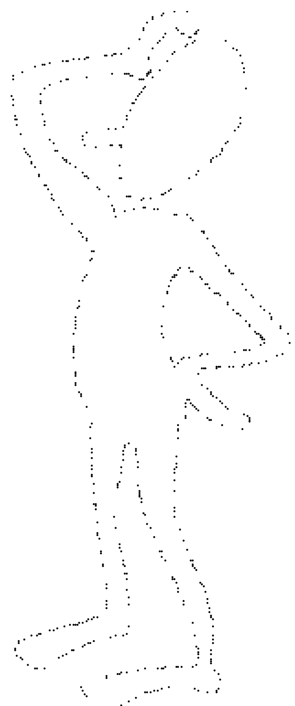
Luiz Henrique de Figueiredo (IMPA)

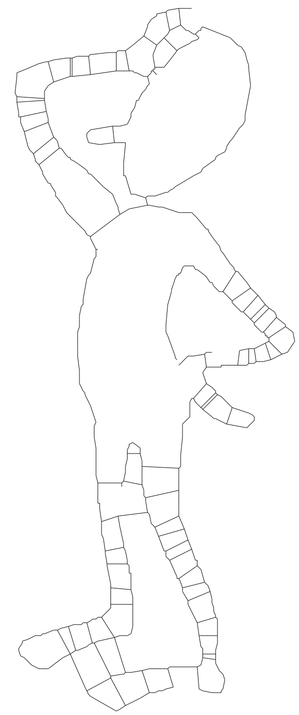
Outline

- Computational morphology
- The relative neighbourhood graph
- Computing the relative neighbourhood graph
- The Urquhart graph
- Results
- Conclusion
- Open problems

Computational morphology

Computational morphology = computational extraction of perceptually meaningful structure from dot patterns.



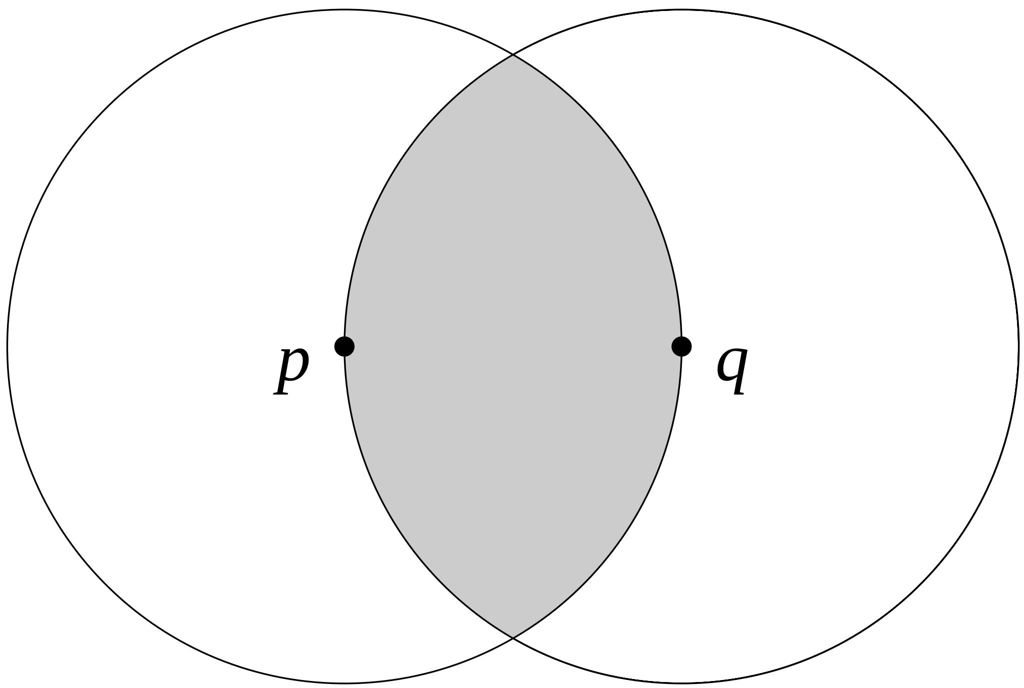


Toussaint (1980) introduced RNG as tool for computational morphology.

The relative neighbourhood graph

$S =$ set of points in the plane.

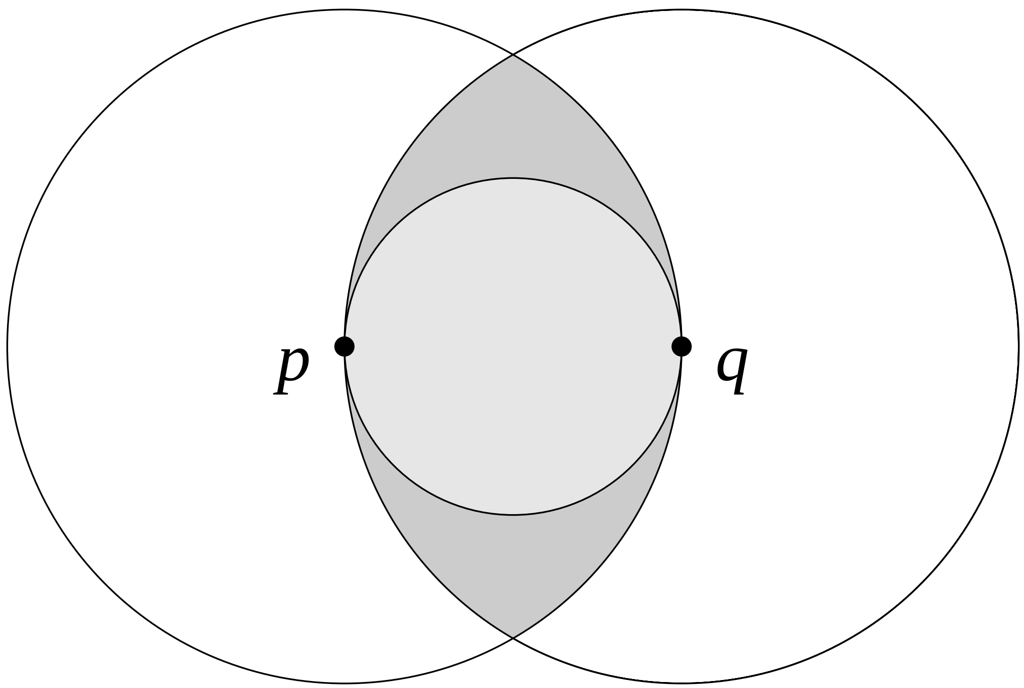
The edges in RNG(S) are defined by $p, q \in S$ with empty *lune*.



The relative neighbourhood graph

$S =$ set of points in the plane.

The edges in RNG(S) are defined by $p, q \in S$ with empty *lune*.



RNG(S) $\subseteq$ GG(S) $\subseteq$ DT(S)

Computing the relative neighbourhood graph

- Brute-force algorithm from definition takes time $O(n^3)$.
- Restriction to DT(S) gives extraction in time $O(n^2)$.

- Supowit (1983) extracts in time $O(n \log n)$.
- Jaromczyk & Kowaluk (1987) extract in time $O(n\,\alpha(n, n))$.
- Jaromczyk, Kowaluk & Yao (1991?) extract in time $O(n)$.

- Lingas (1994) extracts in time $O(n)$
  - ◇ simple algorithm, never implemented.

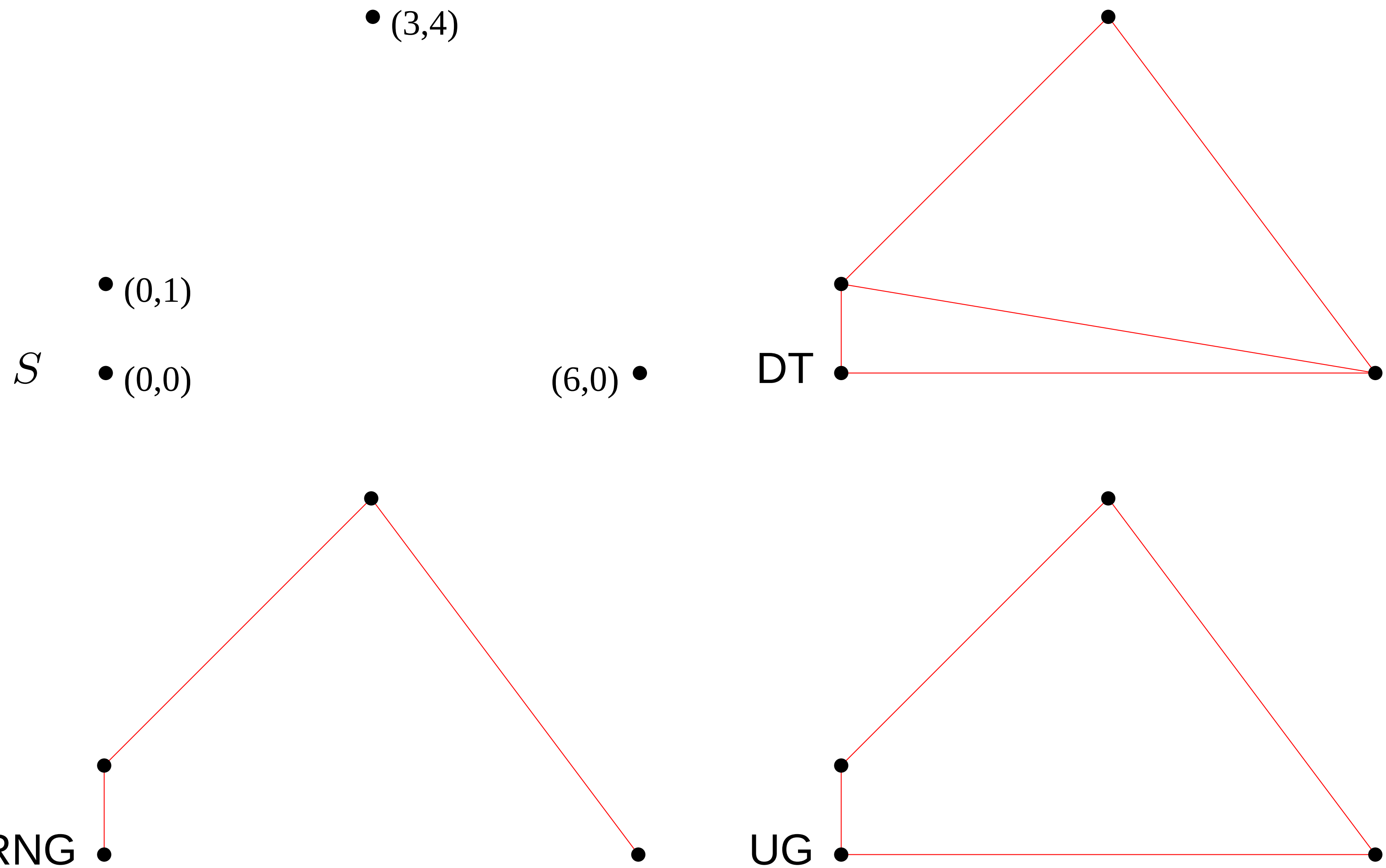
The Urquhart graph

- Idea by Urquhart (1980): test only Delaunay neighbours!
  - ◇ remove longest edge from each Delaunay triangle
  - ◇ common mistake!
  - ◇ new graph: Urquhart graph RNG$(S) \subseteq$ UG$(S) \subseteq$ GG(S)

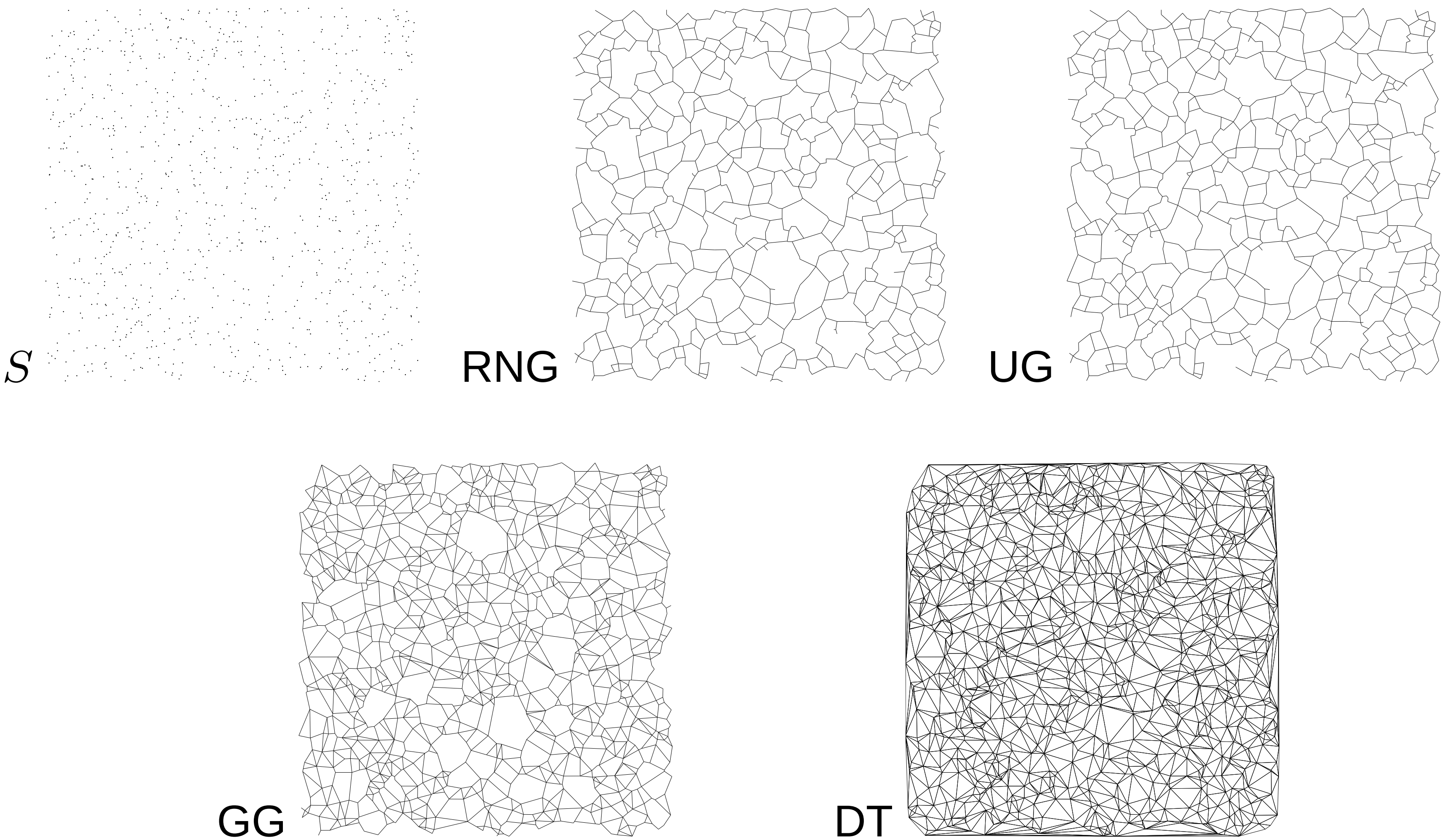
- Toussaint (1980) proposed UG(S) as *approximation* to RNG(S)

- Our theme: how good is this approximation?
  - ◇ How close is UG(S) to RNG(S)?
    - · compare number of edges.
  - ◇ Is UG(S) good for computational morphology?
    - · see pictures!

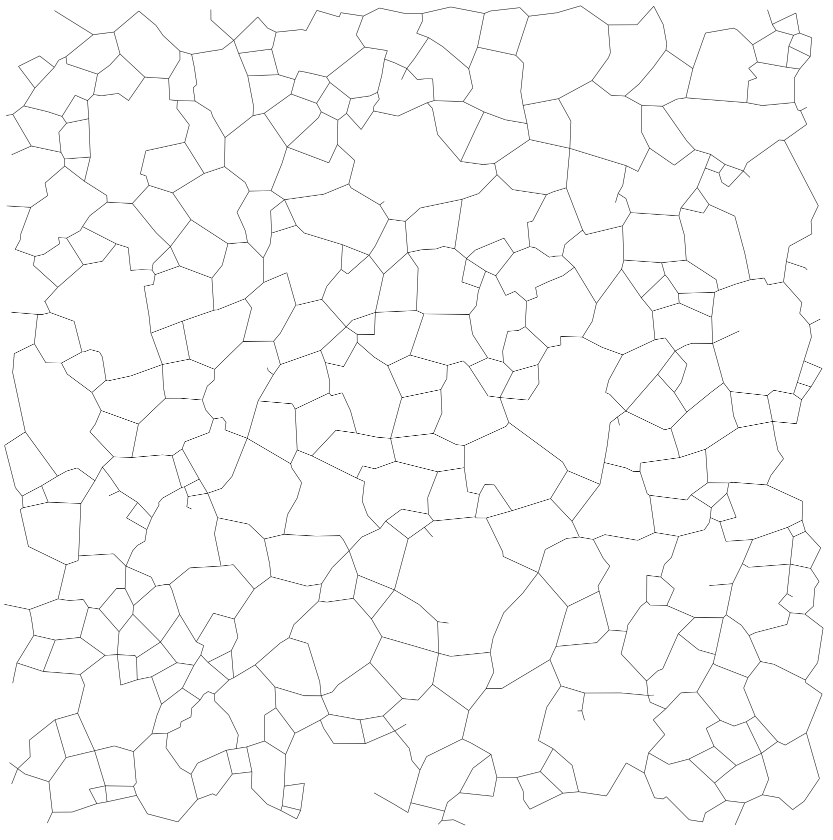
UG $\neq$ RNG



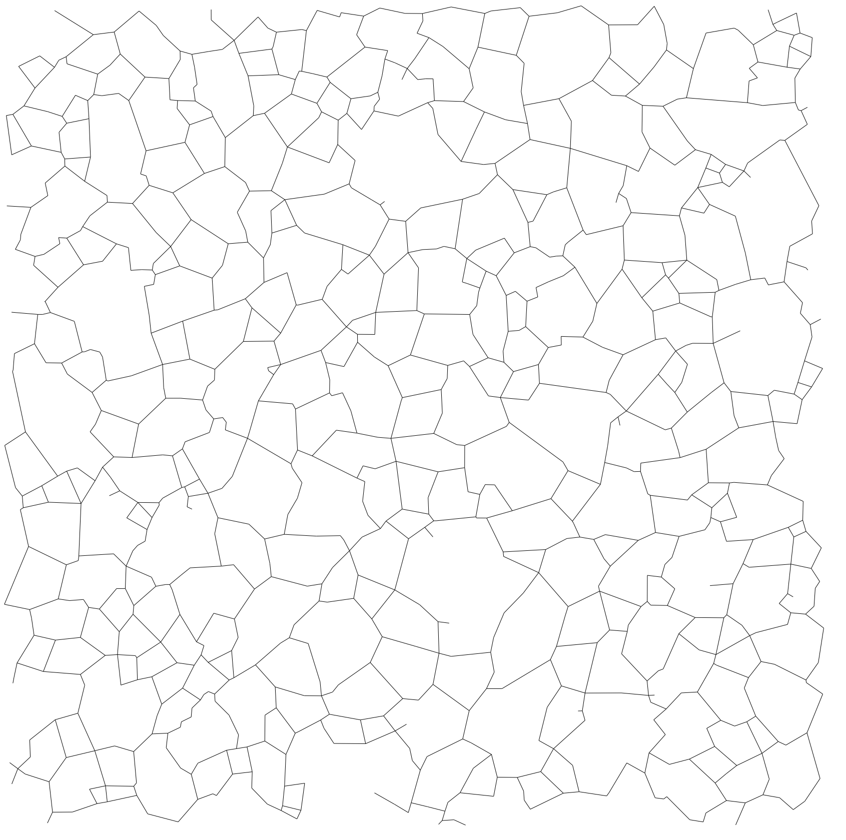
Results: random points in a square



Results: random points in a square

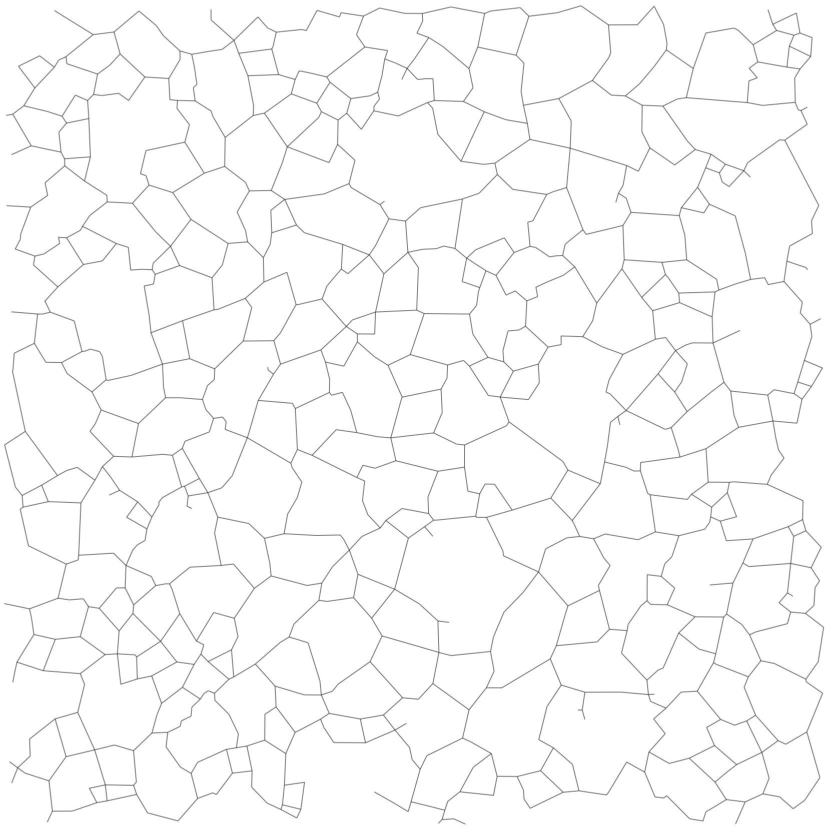


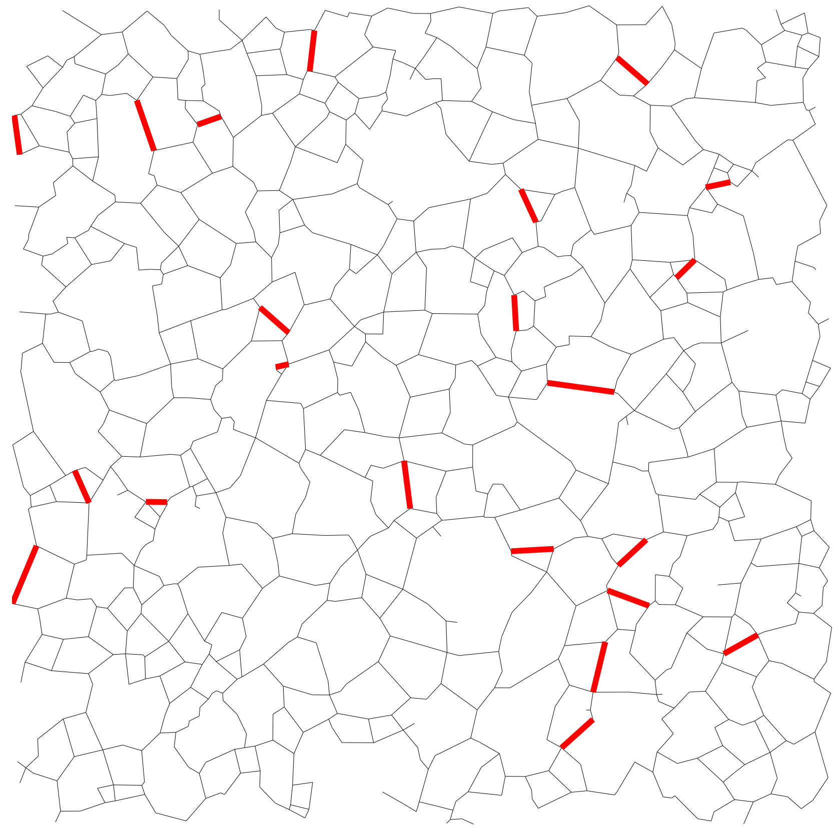
RNG 1241 edges



UG 1263 edges

Results: random points in a square

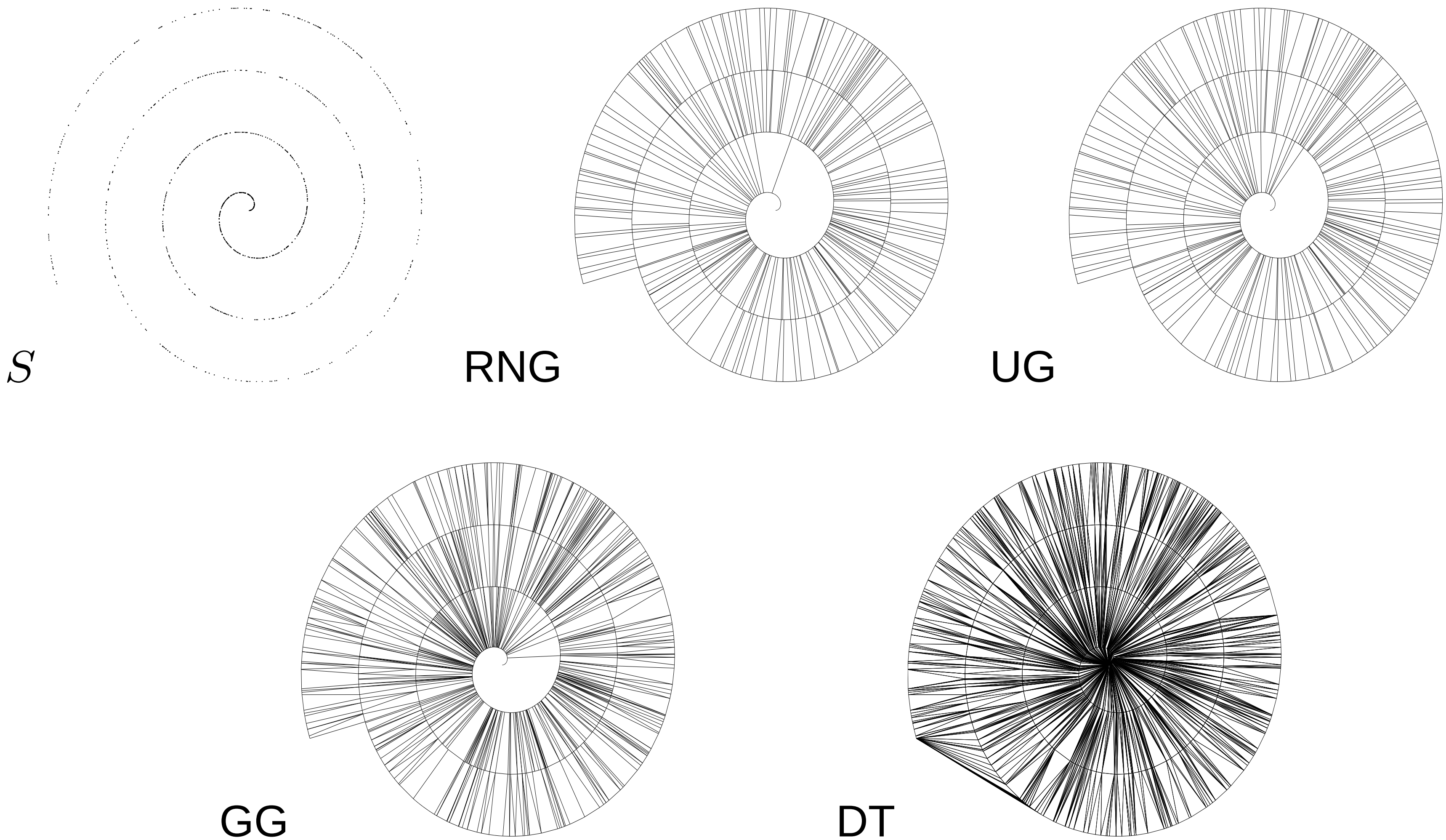




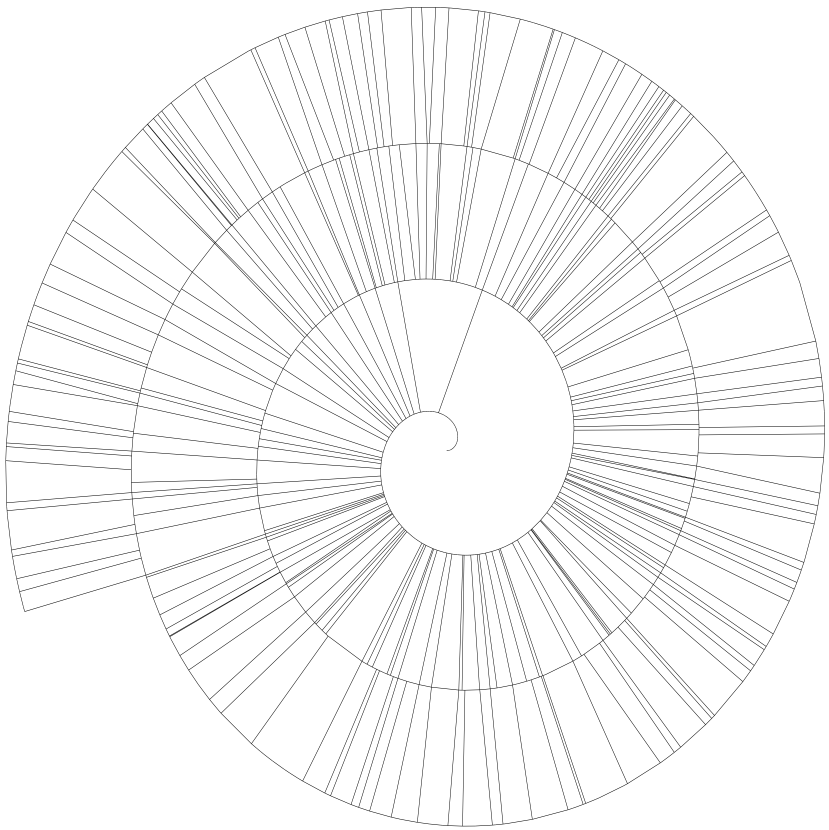
RNG 1241 edges

UG $1263 = 1241 + 22$ edges

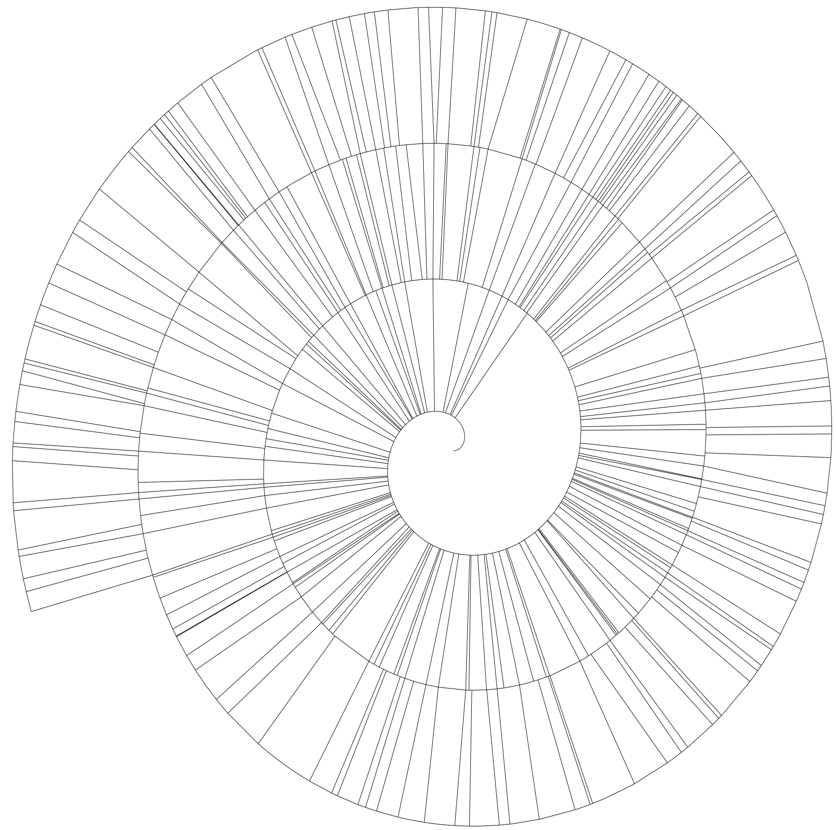
Results: random points on a spiral



Results: random points on a spiral

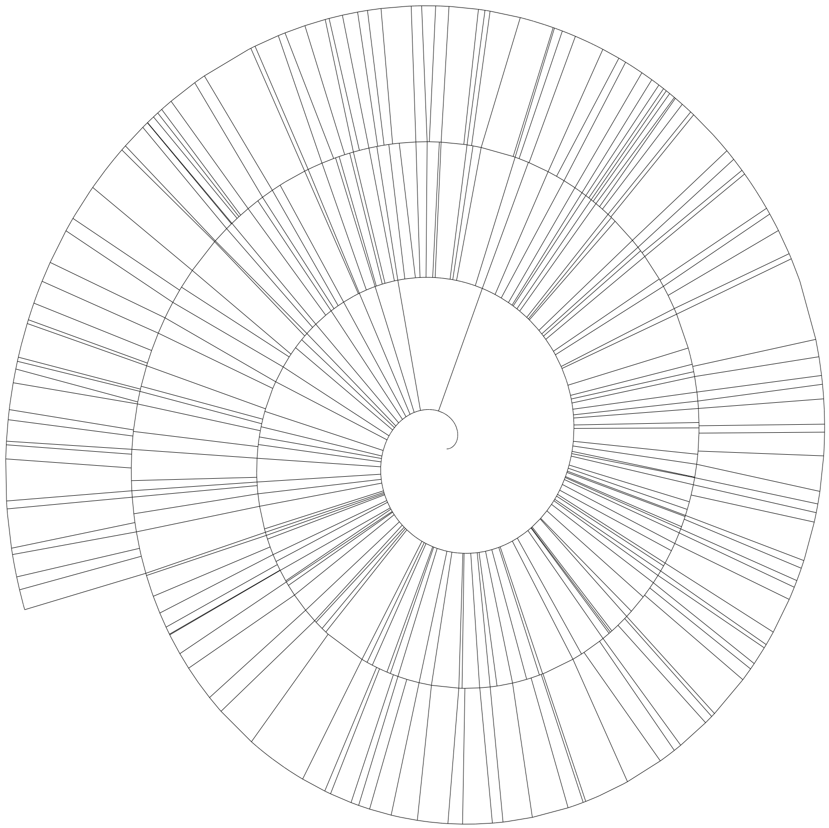


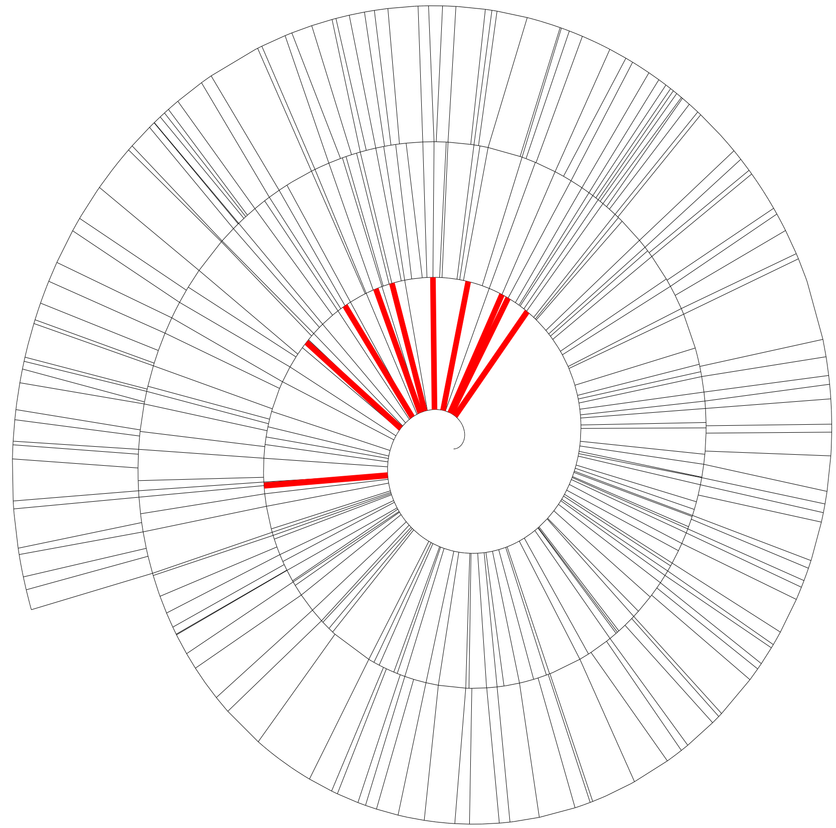
RNG 1291 edges



UG 1301 edges

Results: random points on a spiral

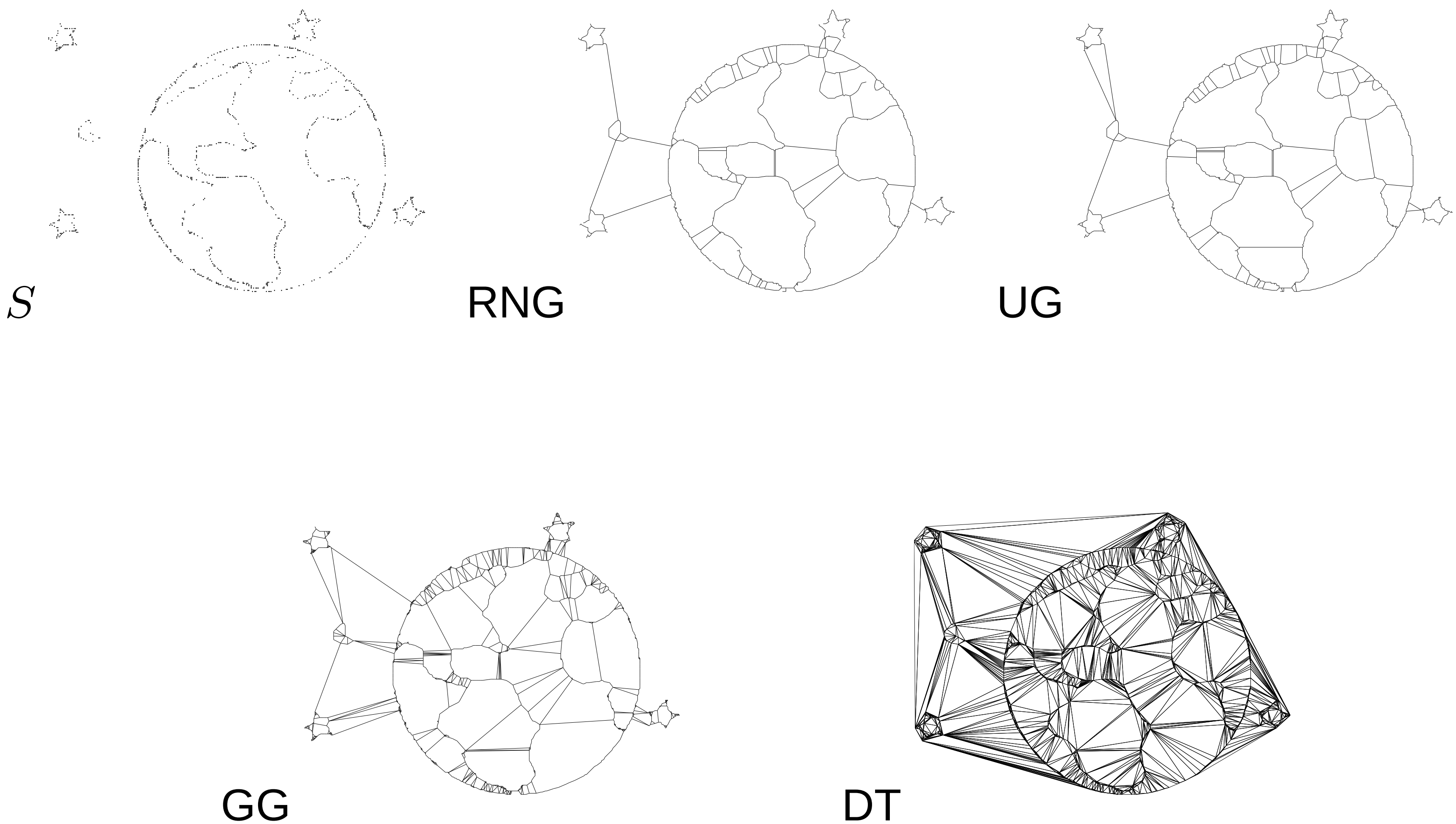




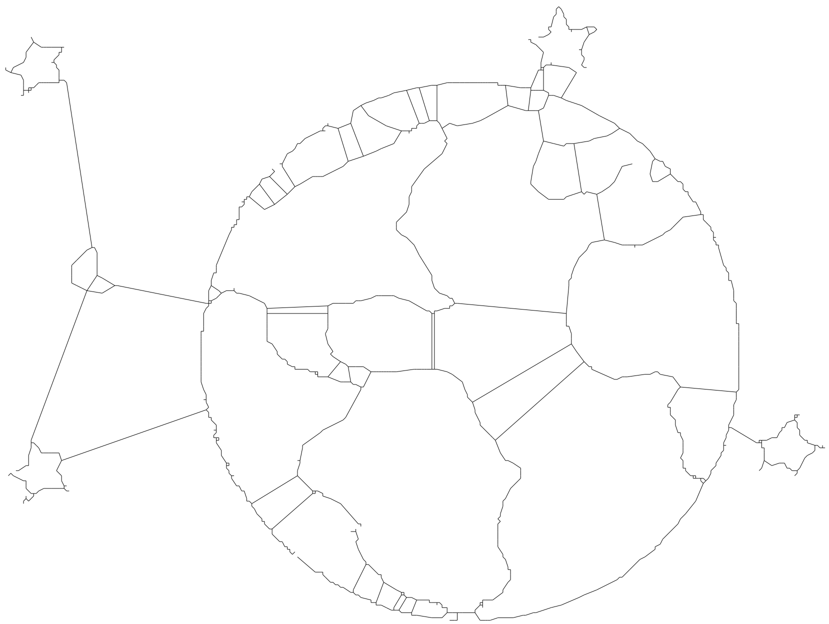
RNG 1291 edges

UG $1301 = 1291 + 10$ edges

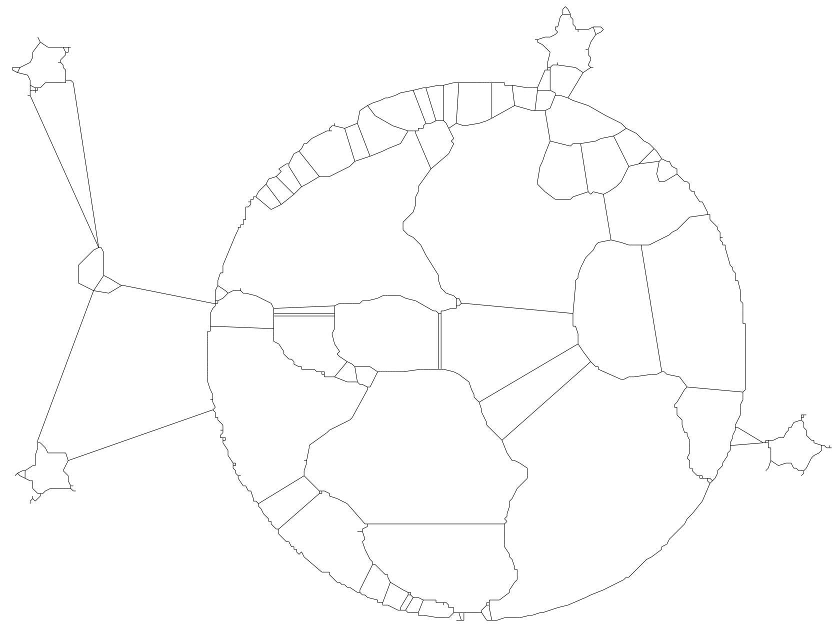
Results: random point on line art: *earth*



Results: random point on line art: *earth*

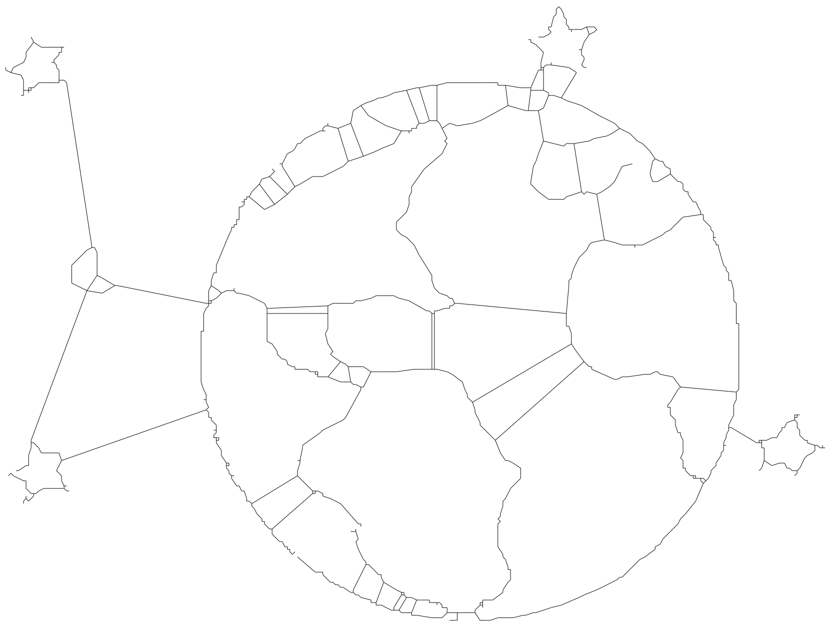


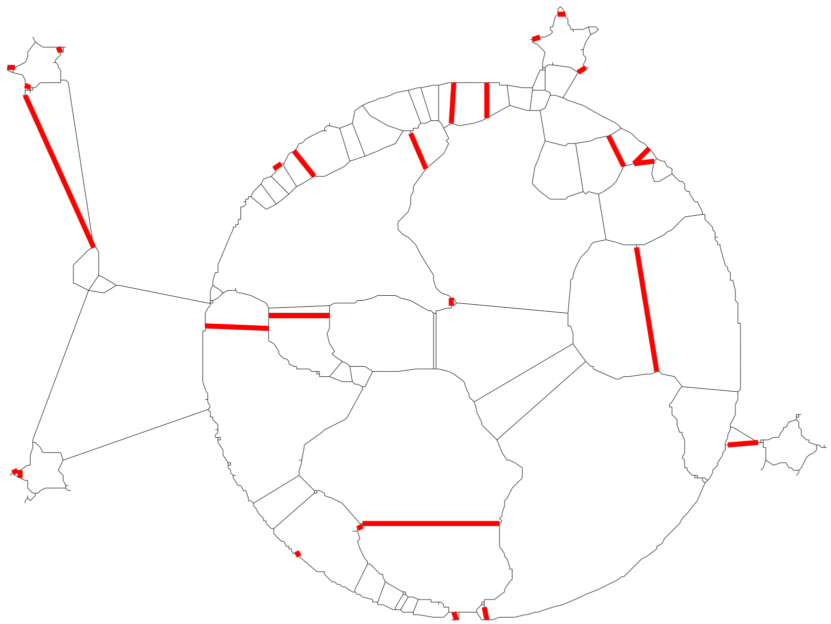
RNG 1089 edges



UG 1116 edges

Results: random point on line art: *earth*

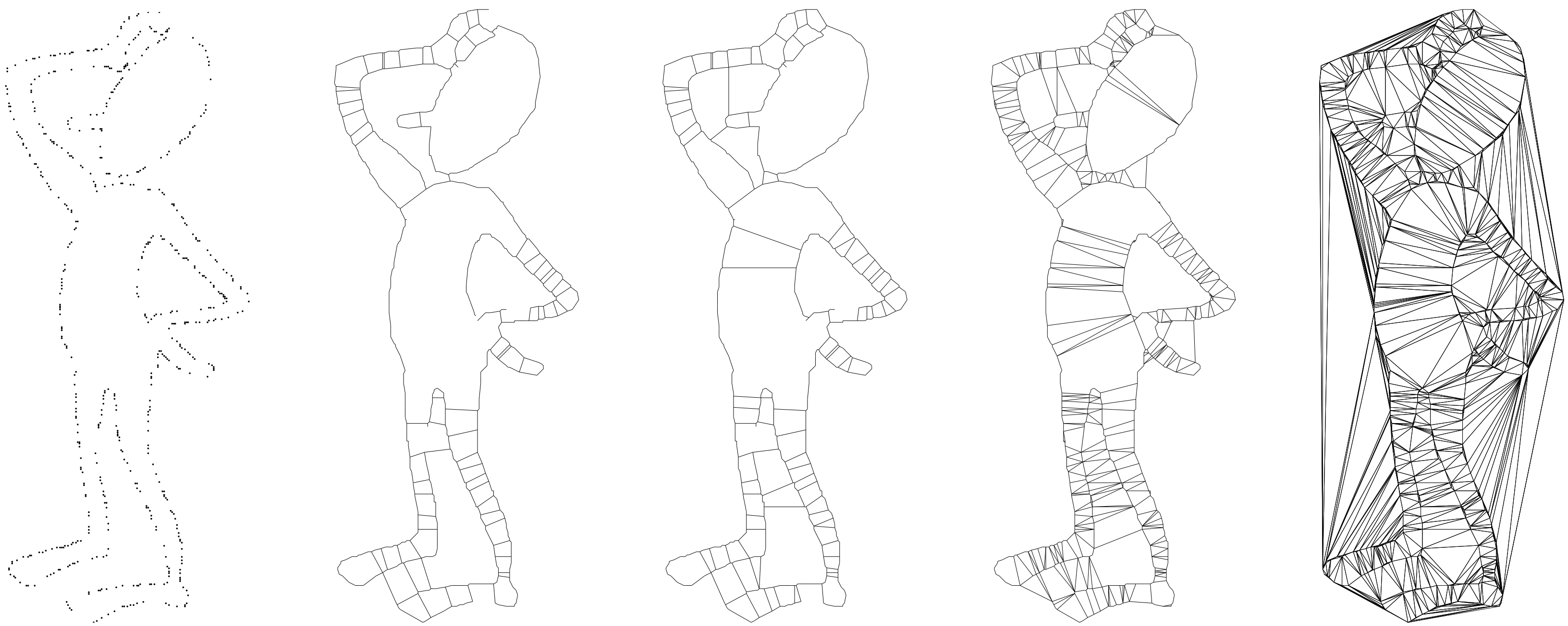




RNG 1089 edges

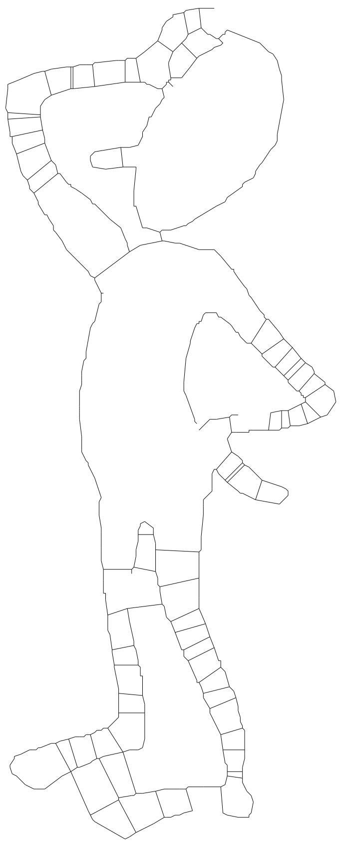
UG $1116 = 1089 + 27$ edges

Results: random point on line art: *man*

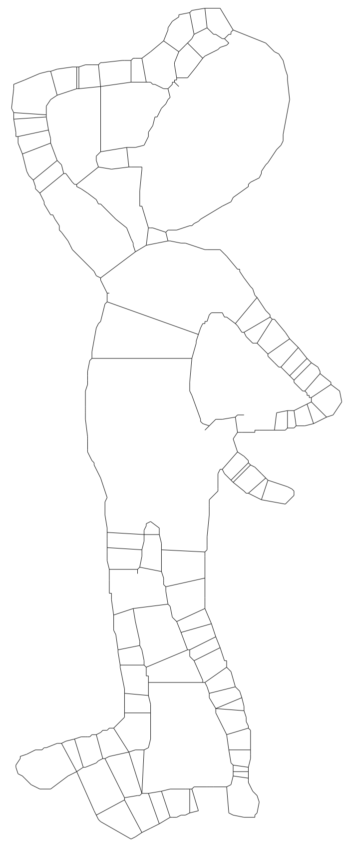


S RNG UG GG DT

Results: random point on line art: *man*

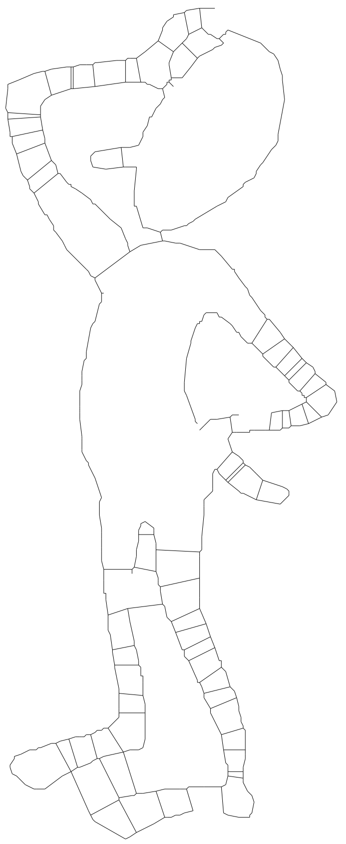


RNG 663 edges

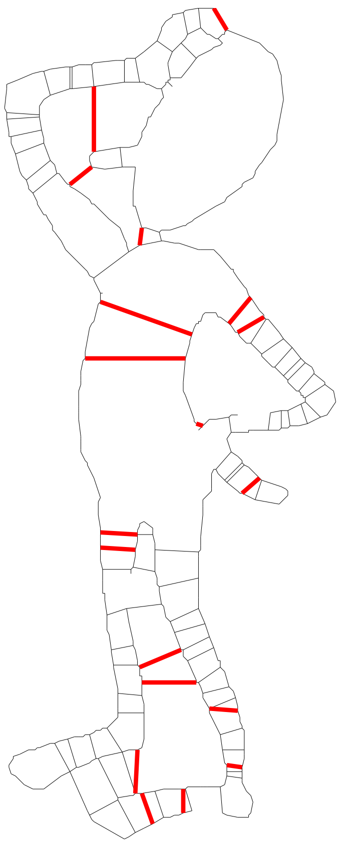


UG 682 edges

Results: random point on line art: *man*



RNG 663 edges



UG $682 = 663 + 19$ edges

Conclusion

- UG(S) good approximation to RNG(S):
  - ◇ only about 2% additional edges for random samples

- Easy to extract UG(S) from DT(S) in linear time.

- Good, free, robust, optimal implementations of DT(S) at *netlib*:
  - ◇ *Triangle*, by Jonathan Richard Shewchuk
  - ◇ *sweep2*, by Steve Fortune

Open problems

- Compare implementations
  - ◇ Supowit (1983)
  - ◇ Lingas (1994)

- Probabilistic results à la Devroye (1988):
  - ◇ $E_{\text{GG}}(N) \sim 2N$
  - ◇ $E_{\text{RNG}}(N) \sim (1.27 + o(1))N$
  - ◇ $E_{\text{UG}}(N) \sim \text{???}\ N$

Thanks

- Godfried Toussaint
- Therese Biedl
- CNPq (Brazilian agency)
- You all for your attention!