A Parallel DCEL Proposal

Andres Calderon

University of California, Riverside

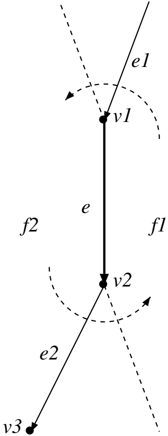
June 12, 2019

Outline

What is a DCEL?

- Doubly connected edge list (DCEL) is a data structure suited to represent a connected planar graph embedded in the plane (Muller-Preparata, 2017).
- A planar embedding of $G = (V, E)$ does not allow crossing edges.
- DCEL captures key topological information about **vertices**, **edges** and **faces**.

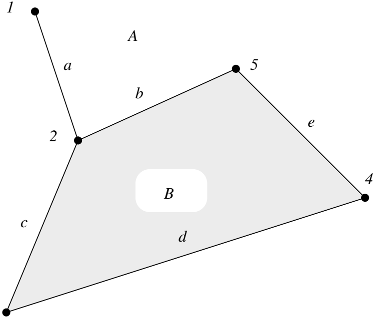
DCEL examples



E	*V1*	*V2*	*F1*	*F2*	*P1*	*P2*
e	*v1*	*v2*	*f1*	*f2*	*e1*	*e2*
e1	*v0*	*v1*	*f1*	*f2*		*e*
e2	*v2*	*v3*	*f1*	*f2*	*e*	

(Freiseisen, 1998)

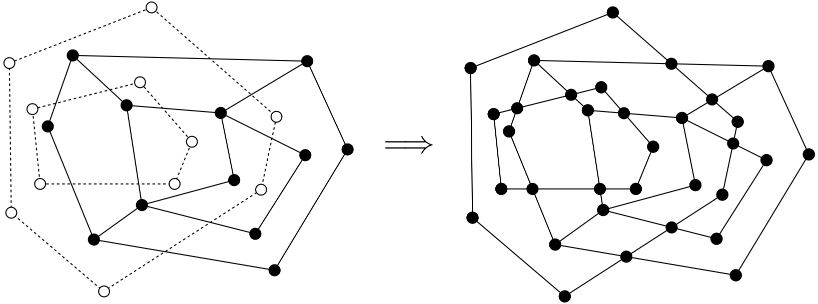
DCEL examples



E	V1	V2	F1	F2	P1	P2
a	1	2	A	A	a	c
b	2	5	A	B	a	e
c	2	3	B	A	b	d
d	3	4	B	A	c	e
e	4	5	B	A	d	b

(Freiseisen, 1998)

Computing the overlay



(de Berg et al, 2008)

The algorithm

1. Find all intersections over the edges of the polygons.
2. Construct a planar graph by inserting all edges and all intersections (DCEL).
3. Traverse this planar graph in order to perform an intersection, union or difference.

Advantages

- DCEL captures topological information.
- DCEL allows multiple spatial operations.
- DCEL can be constructed from this in $\mathcal{O}(n\log(n))$ time using $\mathcal{O}(n)$ additional memory.
- DCEL allows boolean operations in $\mathcal{O}(n)$ time using $\mathcal{O}(n)$ additional space.

Outline

Literature review

- A full reference list of relevant papers is available at: `http://www.cs.ucr.edu/~acald013/public/RIDIR/R04/DCELReport.html`.
- First 5 entries contains solid introductory material about DCELs, specially de Berg et al. (2008) and Freiseisen (1998).
- Next entries discuss alternative methods for map overlay, particularly exploring sequential techniques and parallel spatial joins.
- *There is no mention to a distributed DCEL implementation.*

Literature review

- Sequential techniques (more relevant):
  - G. Barequet, "DCEL: A Polyhedral Database and Programming Environment". IJCGA, vol. 08, pp. 619-636, Oct. 1998.
  - T. Asano and G. Rote, "Constant-Working-Space Algorithms for Geometric Problems". JoCG, vol. 2, pp. 46-68, 2009.
  - W. Freiseisen, "Colored DCEL for Boolean Operations in 2D". Tech report. 1998.

Literature review

- Parallel spatial joins (more relevant):
  - S. Puri and S. K. Prasad, “Efficient Parallel and Distributed Algorithms for GIS Polygonal Overlay Processing”. Washington, DC, USA, 2013, pp. 2238-2241.
  - I. Sabek and M. F. Mokbel, “On Spatial Joins in MapReduce”. 2017, pp. 1-10.
  - C. Zhou, Z. Chen, and M. Li, “A parallel method to accelerate spatial operations involving polygon intersections”. IJGIS, vol. 32, no. 12, pp. 2402-2426, Dec. 2018.
  - W. R. Franklin, S. V. G. de Magalhes, and M. V. A. Andrade, “Data Structures for Parallel Spatial Algorithms on Large Datasets”. 2018, pp. 1619.

Current (sequential) implementations

- CGAL: Computational Geometry Algorithms Library in C++ (binding for Java and Python are available). `https://www.cgal.org/`.
- dyn4j: A 100% Java 2D collision detection and physics engine. `http://www.dyn4j.org/`.
- anglyan/dcel: Python implementation of a doubly connected edge list. `https://github.com/anglyan/dcel`.
- vpranckaitis/muller-preparata-algorithm: DCEL algorithm implementation and visualisation with Scala. `https://github.com/vpranckaitis/muller-preparata-algorithm`.

Outline

The proposal

A parallel DCEL implementation to support big spatial data overlay operations.

What is the problem?

- Currently, most map overlay methods are sequential based.
- For layers with thousand of polygons the execution time is not feasible.
- Most of techniques are oriented to a specific spatial operation (intersection, union, difference, ...).
- Parallel techniques divide the data into partitions and duplicate features if needed in order to solve the problem locally.
- Data structures collecting topological properties are not explored.

Why is it important?

- The rise of big (spatial) data makes necesary to count with fast and efficient techniques for spatial analysis.
- In particular, the RIDIR project has to deal with spatial operations between layers collecting thousand of counties nation-wide. The versatility and efficiency of the spatial methods is cardinal for their studies.
- It should be interesting to count with intermediate data structures that allow multiple map overlay queries.
- DCEL allows linear time to compute spatial operations.

What are the limitations of related work?

- Topological data structures are common in computational geometry. However, most implementations are sequential and they do not scale appropiately on large spatial datasets.
- Parallel spatial joins add complexity due to spatial partitioning necessarily introduces replication.

Why is it challenging?

- Initial stages of the DCEL construction expect it fits in main memory.
- Subsequent operations should be able to query the DCEL in a transparent way.

What are our novel contributions?

- At the best of our knowledge, there is not a distributed DCEL implementation.
- It is necesary to design procedures for partitioning and merging during the DCEL construction.
- The spatial methods that run over the DCEL are based on boolean operations. They should be adjusted accordingly to a distributed DCEL.

What is the validation method?

- We can compare the new implementation to sequential versions of the algorithm.
- To test scalability we can perform benchmarks to the current versions of `area_tables` and `area_interpolate`.