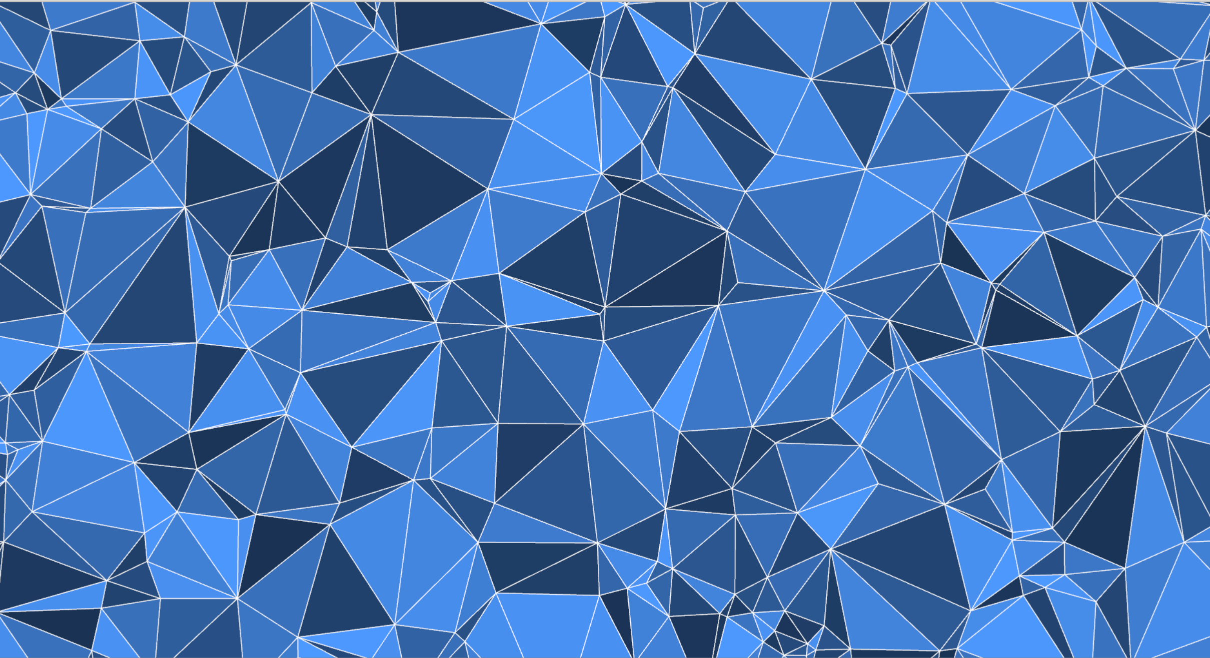
Delaunay Triangulation Mosaic



Algorithmical Geometry: Delaunay Triangulation

Markus Pawellek

December 29, 2021

Outline

Introduction

Introduction: Previous Work and Hands-On Approach

(1) Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator”, 1996

(2) Guibas and Stolfi, “Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams”, 1985

(3) Dwyer, “A Faster Divide-and-Conquer Algorithm for Constructing Delaunay Triangulations”, 1987

(4) Lee and Schachter, “Two Algorithms for Constructing a Delaunay Triangulation”, 1980

Introduction: Overview

Educational Problems:

- Duality to Voronoi Diagrams, Dirichlet
- Incremental, Sweepline, Divide-and-Conquer Algorithms
- Varying Data Structures

Here: Triangular Data Structure and Divide-and-Conquer Algorithm

- Smallest Data Structure
- Fastest Algorithm
- Robust when using tweaks

Background

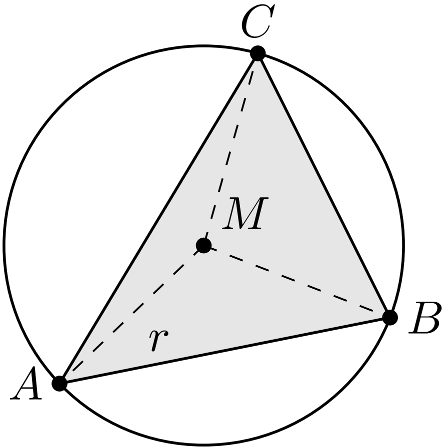
Background: Triangle and Circumcircle

Triangle

$A, B, C \in \mathbb{R}^2$ affinely independent define vertices of a triangle.

Circumcircle

Circle that intersects with all vertices of the triangle.

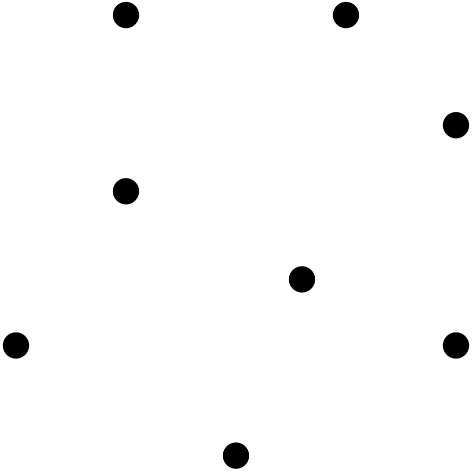


Background: Point Set

Point Set

$\mathcal{V} \subset \mathbb{R}^2$ finite, $\#\mathcal{V} \geq 3$,

affinely span $\mathbb{R}^2$



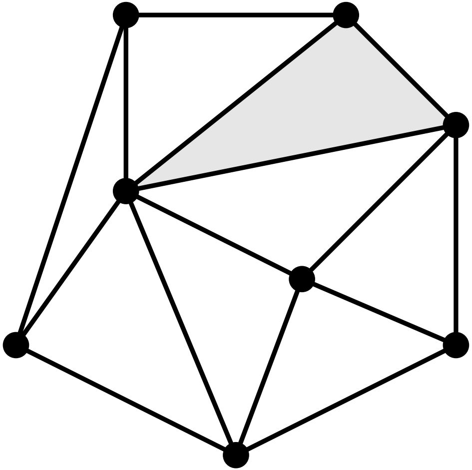
Background: Triangulation

Point Set

$\mathcal{V} \subset \mathbb{R}^2$ finite, $\#\mathcal{V} \geq 3$,
affinely span $\mathbb{R}^2$

Triangulation

Planar straight-line graph over $\mathcal{V}$ such that its edges form a maximal subset of non-crossing edges.



Background: Delaunay Triangulation

Point Set

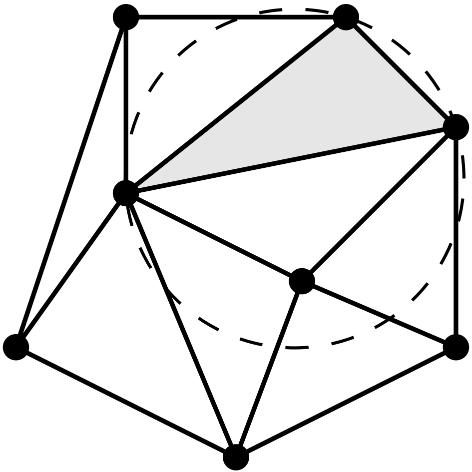
$\mathcal{V} \subset \mathbb{R}^2$ finite, $\#\mathcal{V} \geq 3$,
affinely span $\mathbb{R}^2$

Triangulation

Planar straight-line graph over $\mathcal{V}$ such that its edges form a maximal subset of non-crossing edges.

Delaunay Triangulation

Circumcircle of any triangle contains no other points of $\mathcal{V}$.



Background: Delaunay Triangulation

Point Set

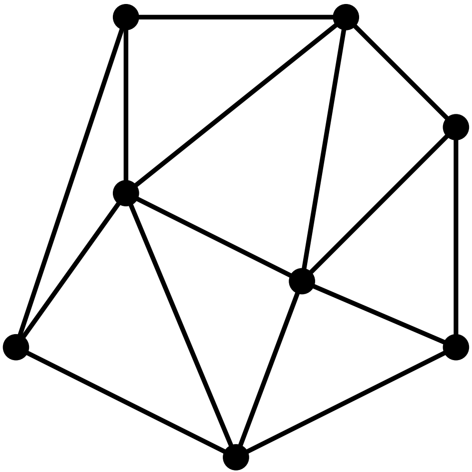
$\mathcal{V} \subset \mathbb{R}^2$ finite, $\#\mathcal{V} \geq 3$,
affinely span $\mathbb{R}^2$

Triangulation

Planar straight-line graph over $\mathcal{V}$ such that its edges form a maximal subset of non-crossing edges.

Delaunay Triangulation

Circumcircle of any triangle contains no other points of $\mathcal{V}$.



Background: Delaunay Triangulation

Point Set

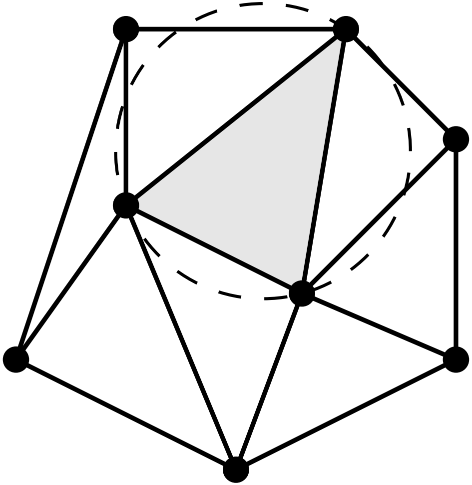
$\mathcal{V} \subset \mathbb{R}^2$ finite, $\#\mathcal{V} \geq 3$, affinely span $\mathbb{R}^2$

Triangulation

Planar straight-line graph over $\mathcal{V}$ such that its edges form a maximal subset of non-crossing edges.

Delaunay Triangulation

Circumcircle of any triangle contains no other points of $\mathcal{V}$.



Background: Properties of Delaunay Triangulation

- Duality to Voronoi Diagram
- always exists
- If no points are cocircular, unique
- optimality: maximization of the minimum angle of all angles
- boundary is convex hull
- Delaunay condition implies triangulation

Background: Existence and Uniqueness of Delaunay Triangulation

Geometric Primitives

Geometric Primitives: Counter-Clockwise

$$\begin{vmatrix} A_x & A_y & 1 \\ B_x & B_y & 1 \\ C_x & C_y & 1 \end{vmatrix} > 0$$

Geometric Primitives: Inside Circumcircle

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix} > 0$$

Data Structures

Algorithm

Implementation

Implementation

- Geometric Primitives need exact computation and therefore arbitrary precision

Applications

Conclusions

Thank you for Your Attention!

References

(1) Jonathan Richard Shewchuk. "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator". In: *Applied Computational Geometry: Towards Geometric Engineering*. Ed. by Ming C. Lin and Dinesh Manocha. Vol. 1148. Lecture Notes in Computer Science. From the First ACM Workshop on Applied Computational Geometry. Springer-Verlag, May 1996, pp. 203–222. URL: https://people.eecs.berkeley.edu/~jrs/papers/triangle.pdf (visited on 11/07/2020).

(2) Leonidas Guibas and Jorge Stolfi. "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams". In: *ACM Transactions on Graphics* 4 (April 1985), pp. 74–123. DOI: 10.1145/282918.282923. URL: http://sccg.sk/~samuelcik/dgs/quad_edge.pdf (visited on 11/07/2020).

(3) Rex A. Dwyer. "A Faster Divide-and-Conquer Algorithm for Constructing Delaunay Triangulations". In: *Algorithmica* 2 (November 1987), pp. 137–151. DOI: 10.1007/BF01840356.

(4) D. T. Lee and B. J. Schachter. "Two Algorithms for

(7) A. Bowyer. "Computing Dirichlet Tessellations". In: *The Computer Journal* 24 (1981), pp. 162–166. DOI: 10.1093/comjnl/24.2.162.

(8) Christoph Burnikel. *Delaunay Graphs by Divide and Conquer*. 1998. URL: https://pure.mpg.de/rest/items/item_1819432_4/component/file_2599484/content (visited on 11/07/2020).

(9) V. Fuetterling, C. Lojewski, and F.-J. Pfreundt. "High-Performance Delaunay Triangulation for Many-Core Computers". In: *High Performance Graphics 2014* (2014), pp. 97–104. DOI: 10.2312/hpg.20141098.

(10) P. Cignoni, C. Montani, and R. Scopigno. "DeWall: A Fast Divide-and-Conquer Delaunay Triangulation Algorithm in E^d". In: *Computer-Aided Design* 30 (1998), pp. 333–341. DOI: 10.1016/S0010-4485(97)00082-1.

(11) Jyrki Katajainen and Markku Koppinen. "Constructing Delaunay Triangulations by Merging Buckets in Quad-Tree Order". In: *Fundamenta Informaticae* 11 (April 1988), pp. 275–288.