Voronoi with Obstacles, Delaunay Constrained Triangulation and Delaunay 3D

Marc Comino Trinidad

December 13, 2014

Table of contents

Table of Contents

Outline

- Let $P = \{p_i\}, i \in 1..n, p_i \in \mathbb{R}^2, n \in [2, \infty)$ be a set of generator points.
- Let $O = \{O_i\}, i \in 1.n_o, n_o \in [1..\infty)$ be a set of closed regions we will call obstacles.

Assumptions

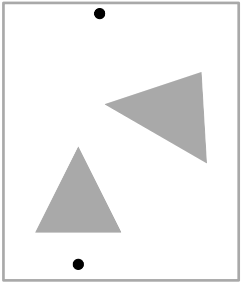
- Obstacles are non-transparent, non-traversable, do not overlap each-other, are connected and *do not have holes*.
- Points of P are not allowed to be located inside the obstacles.
- Furthermore, the obstacles are polygons.

Definitions

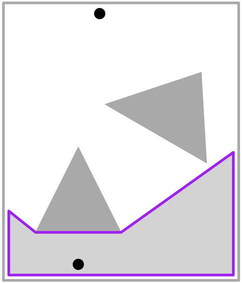
- **Shortest-path (geodesic) distance:** length of the shortest continuous path connecting a point p and a point $p_i \in P$ that does not intersect any obstacle. We denote it as $d_{sp}(p, p_i)$.
- **Visibility polygon:**

$$VIS(p_i) = \{p | \overline{p_i p} \cap O_J = \emptyset, j \in 1..n_0, p \in \mathbb{R}^2\}$$

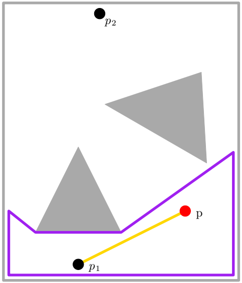
Visibility Polygon



Visibility Polygon



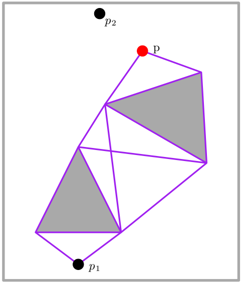
Visibility Polygon



Definitions

- **Geometric Graph:** A geometric graph is a graph that can be represent with a set **geometric points** and a set **geometric lines** connecting these points. We denote it as $G(Q, L)$.
- Let L_{vis} be the set $\{\overline{q_i q_j} | q_j \in Vis(q_i), i < j, j \in 1..n\}$.
- A **Visibility Graph** on $G(Q, L)$ is defined as $G(Q, L_{vis})$.

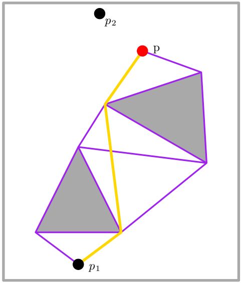
Visibility Graph



Finding Shortest Path using Visibility Graph

1. Let $Q = \{p, p_i, v(O)\}$, $L = e(O)$.
2. Build $G(Q, L_{vis})$ from $G(Q, L)$.
3. Find shortest path between p and p_i in $G(Q, L)$ using Dijkstra or similar.

Finding Shortest Path using Visibility Graph



Definitions

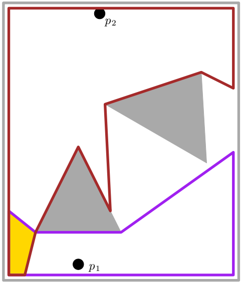
- **Shortest-path Voronoi Region:**

$$V(p_i) = \{p | d_{sp}(p, p_i) \leq d_{sp}(p, p_j), j \in 1..n\}$$

- **Shortest-path (geodesic) Voronoi Diagram:**

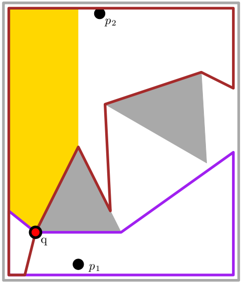
$$\mathcal{V}_{sp}(P, O, d_{sp}, \mathbb{R}^2) = \{V(p_i)\}, i \in 1..n$$

Voronoi Regions Bisector



$$b(p_1, p_2) = \{x | \ \|x - p_1\| = \|x - p_2\|\}$$

Voronoi Regions Bisector



$$b(p_1, p_2) = \{x \mid \|x - q\| + \|q - p_1\| = \|x - p_2\|\}$$

Table of Contents

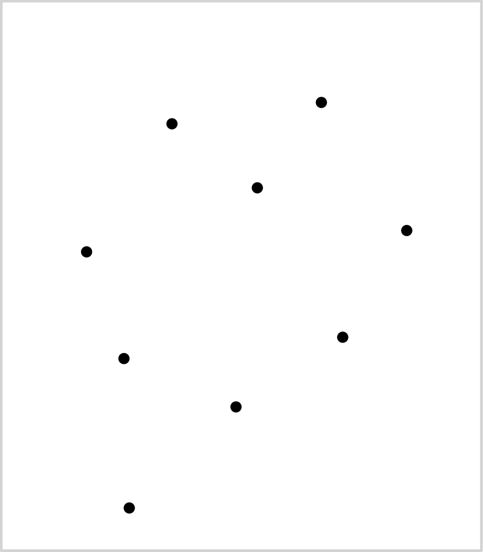
Outline

- Let $P = \{p_i\}, i \in 1..n, p_i \in \mathbb{R}^2, n \in [2, \infty)$ be a set of generator points.
- Let $O = \{O_i\}, i \in 1..n_o, n_o \in [1..\infty)$ be a set of line segments we will call obstacles.

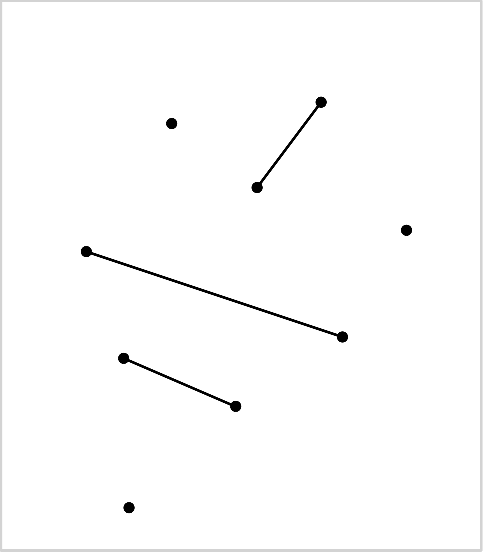
Assumptions

- Obstacles do not intersect each-other.
- The end points of the line segments O are all generator points in P.

An Example



An Example



Definitions

- **Visibility Shortest-path distance:**

$$d_{vsp}(p, p_i) \begin{cases} \|x - x_i\| & if\ p_i \in VIS(p), \\ \infty & otherwise \end{cases}$$

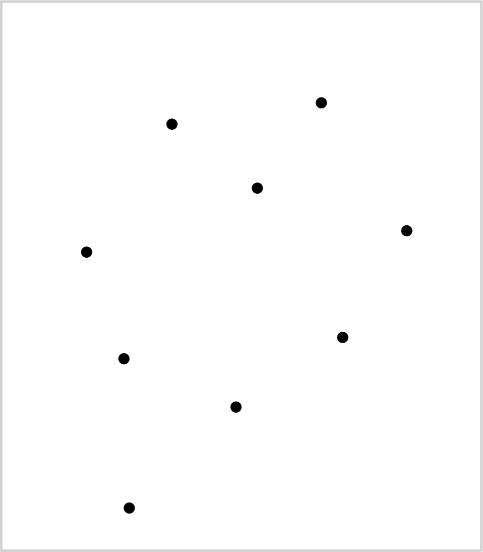
- **Visibility Shortest-path Voronoi Region:**

$$V(p_i) = \{p | d_{vsp}(p, p_i) \leq d_{vsp}(p, p_j), j \in 1..n\}$$

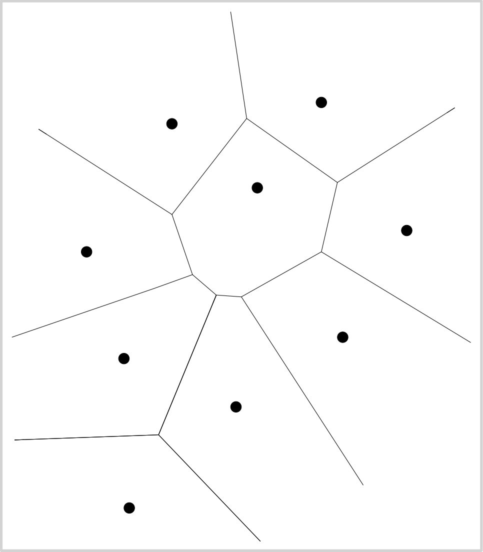
- **Visibility Shortest-path Voronoi Diagram:**

$$\mathcal{V}_{vsp}(P, O, d_{vsp}, \mathbb{R}^2) = \{V(p_i)\}, i \in 1..n$$

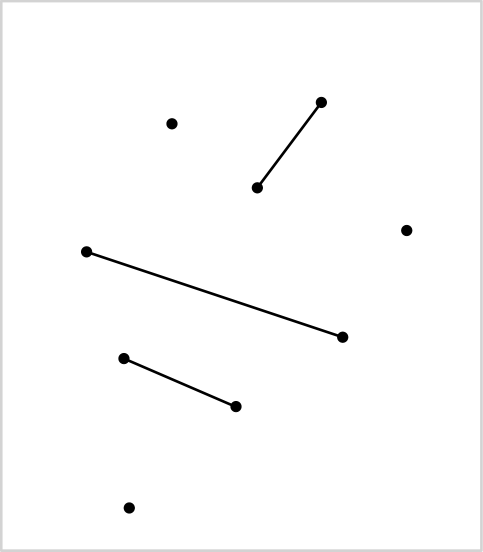
An Example



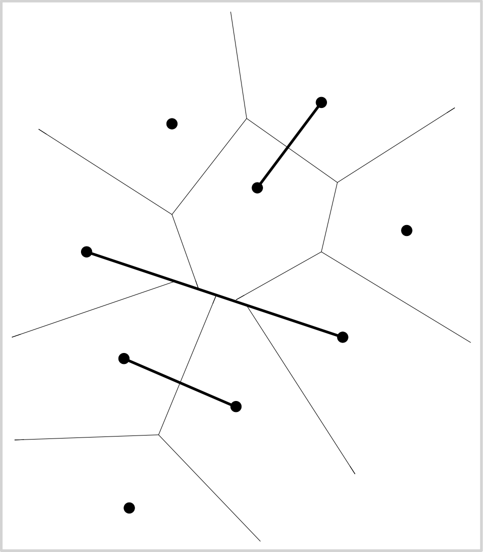
An Example



An Example



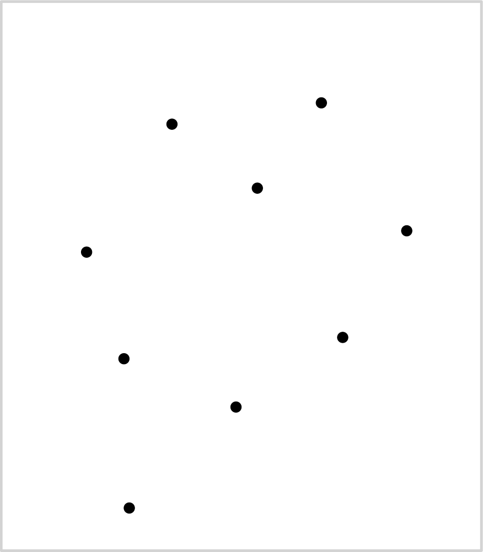
An Example



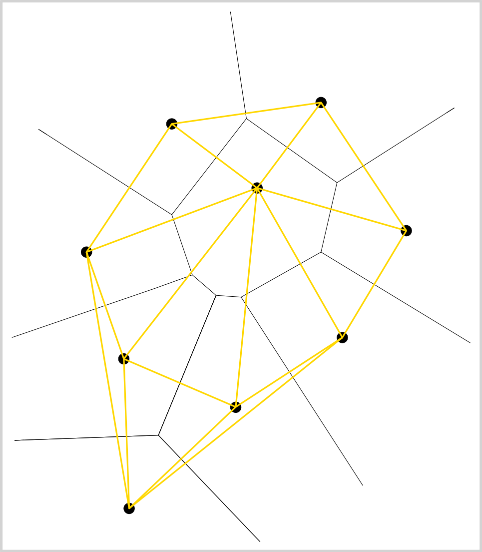
Properties

- A Visibility Shortest-Path Voronoi region may be non-convex.
- A point on an edge on a $\mathcal{V}_{vsp}$ may not be equally distant from two or more points.
- The dual of the $\mathcal{V}_{vsp}$ is NOT a constrained Delaunay triangulation.
- The dual of the $\mathcal{V}_{vsp}$ is a subgraph of the corresponding constrained Delaunay triangulation.

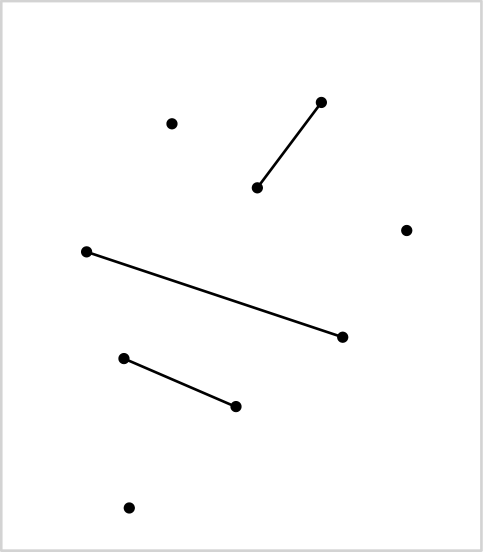
An Example



An Example



An Example



An Example

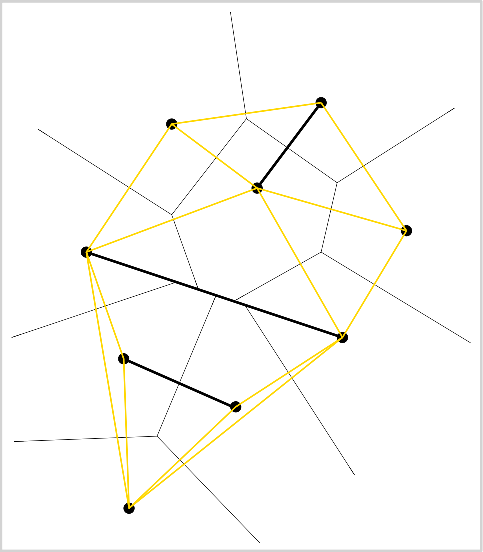
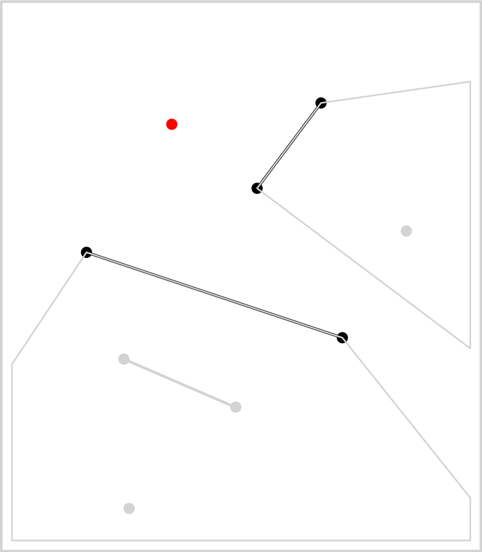


Table of Contents

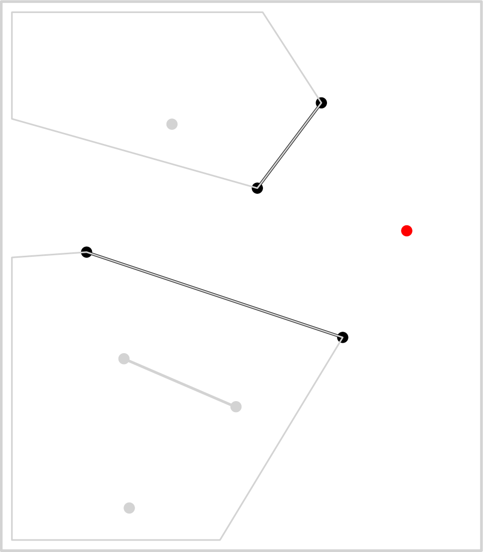
Definitions

- **Constrained Delaunay Triangulation:** Let $G(P_o, L_o)$ be a planar straight-line graph representing obstacles and Q a set of points. The constrained Delaunay triangulation is a triangulation spanning $P = P_o \cup Q$ satisfying the condition that the cirumcircle of each triangle does not contain in its interior any other vertex which is visible from the vertices of the triangle.

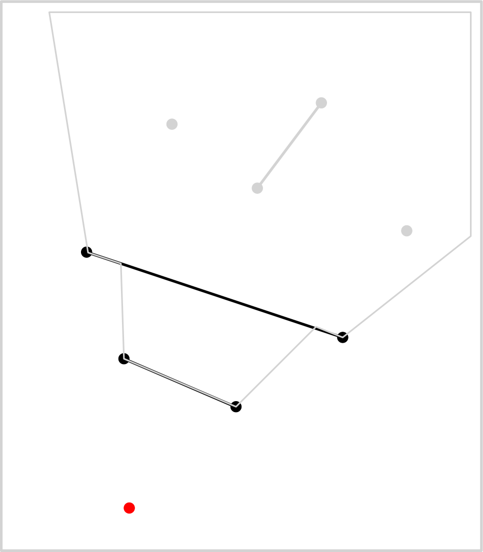
Visibility



Visibility



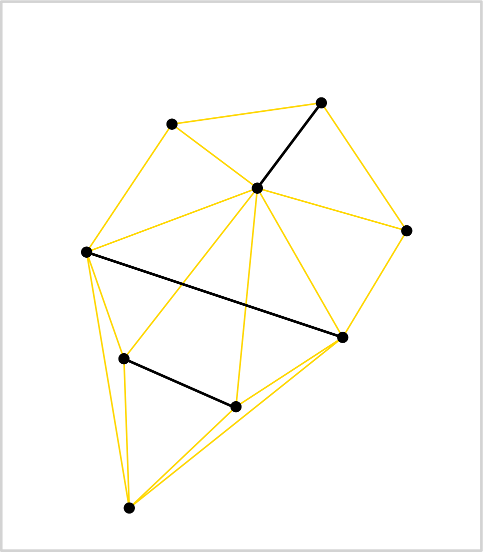
Visibility



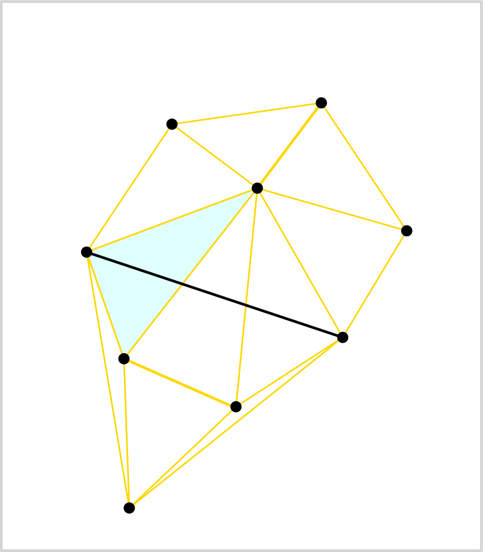
Algorithms

1. Processing points and constraints together:
   1.1 Divide and Conquer by P. Chew[Chew(1989)].
   1.2 Incremental by Floriani *et al.*[De Floriani and Puppo(1992)].
2. Processing points and constraints separately:
   2.1 Removing triangles pierced by a constraint and re-triangulating.
   2.2 Flipping edges until integrating the constraint.

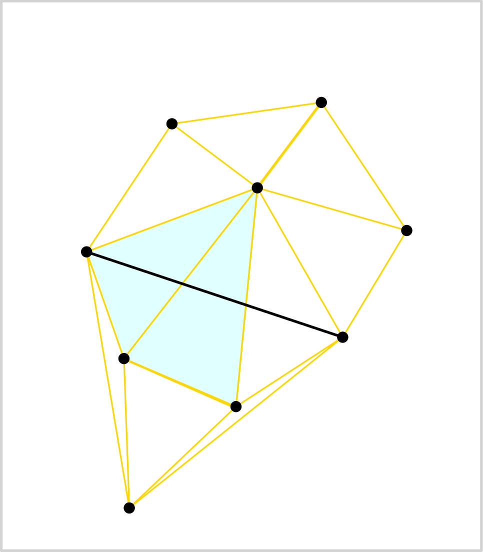
Removing triangles



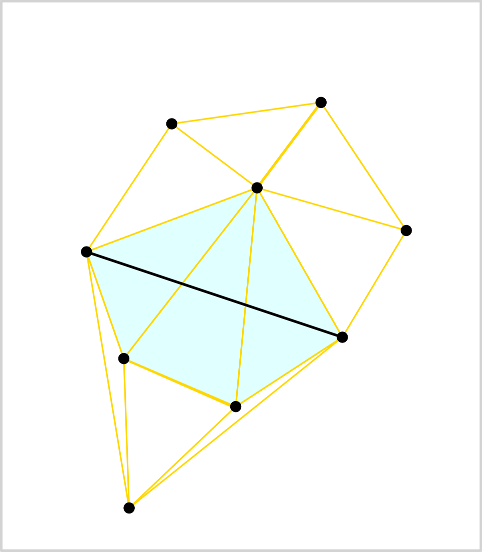
Removing triangles



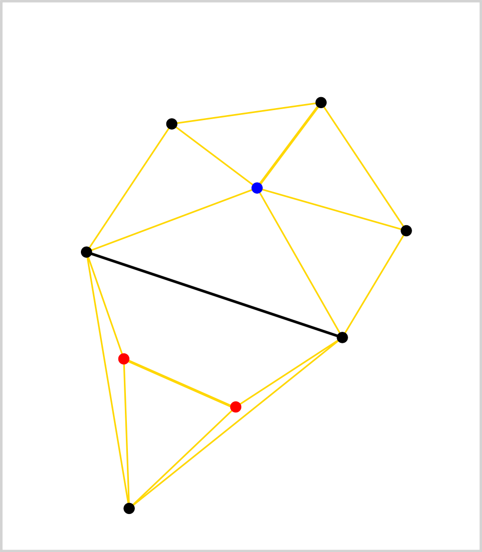
Removing triangles



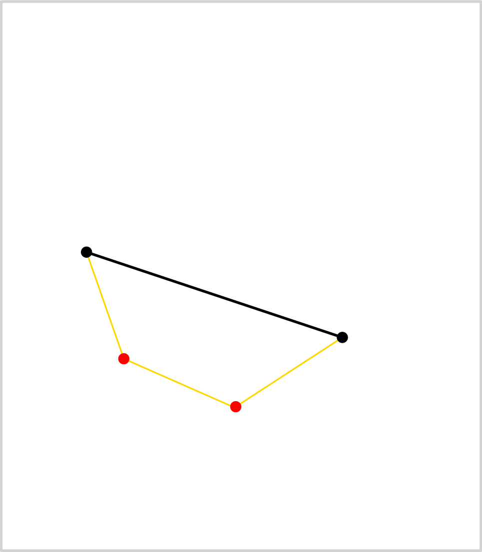
Removing triangles



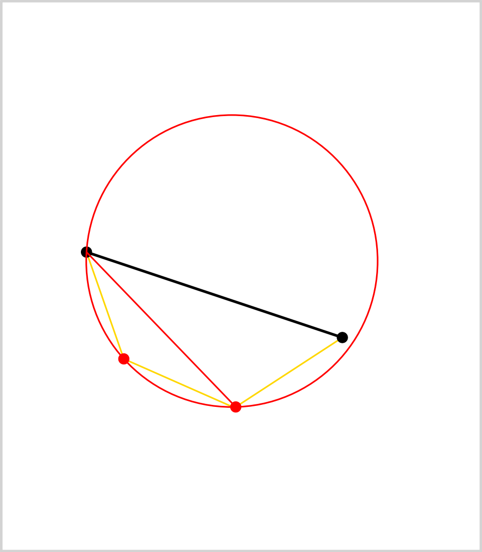
Removing triangles



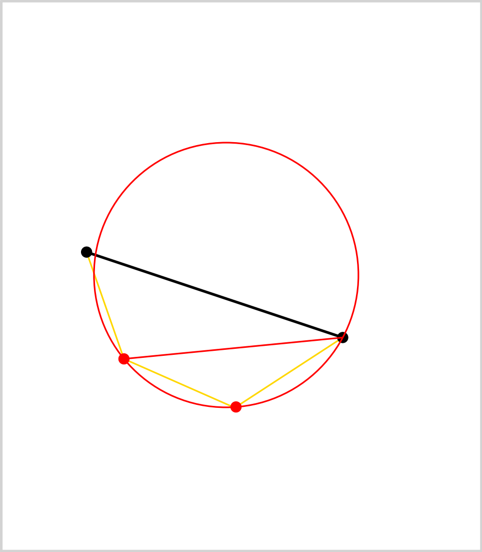
Removing triangles



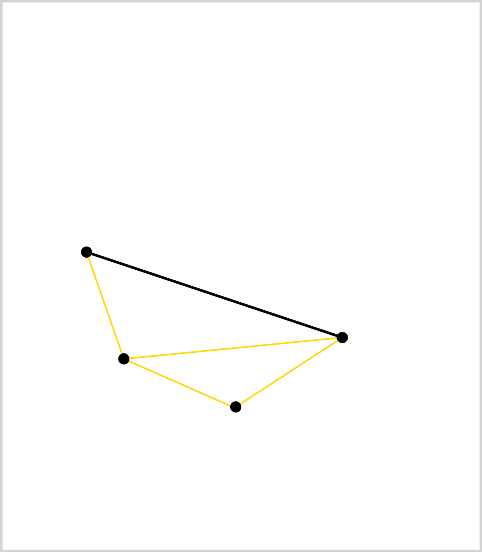
Removing triangles



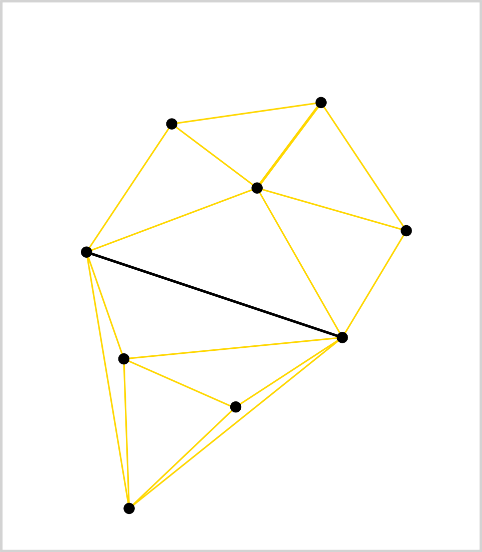
Removing triangles



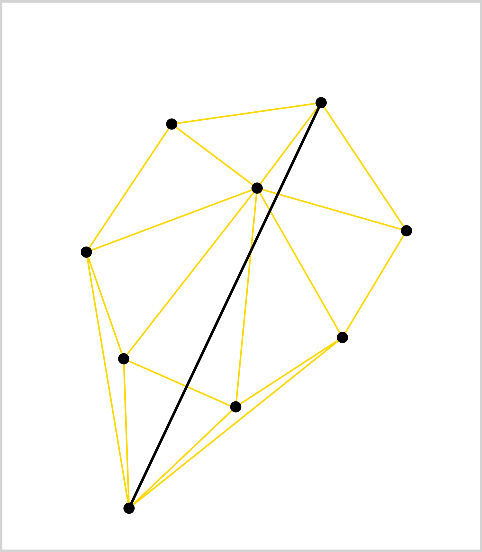
Removing triangles



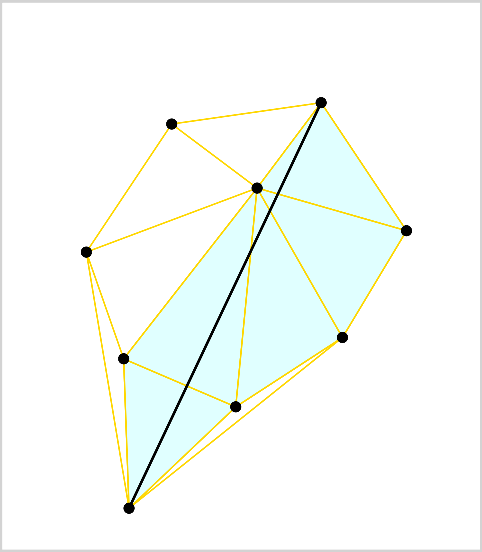
Removing Triangles



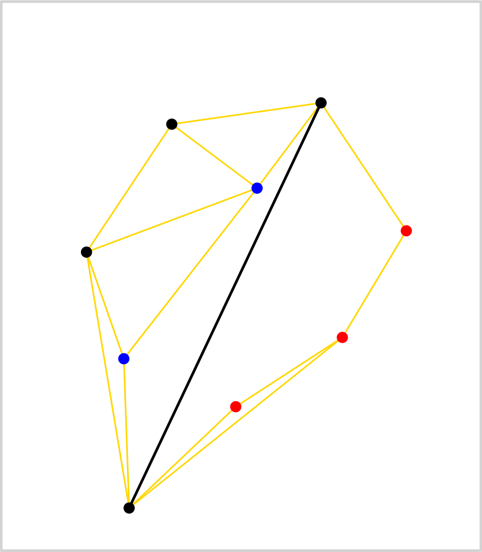
Removing Triangles



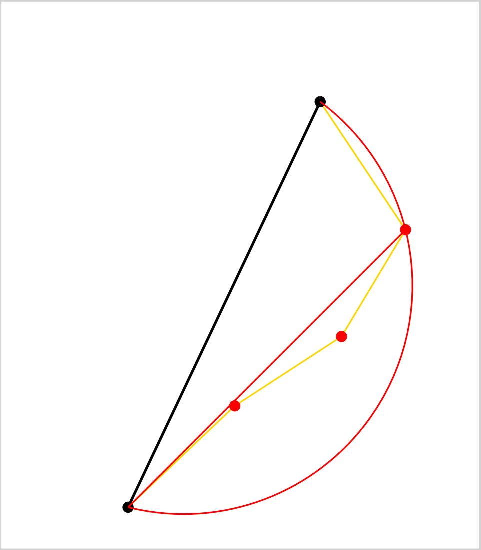
Removing Triangles



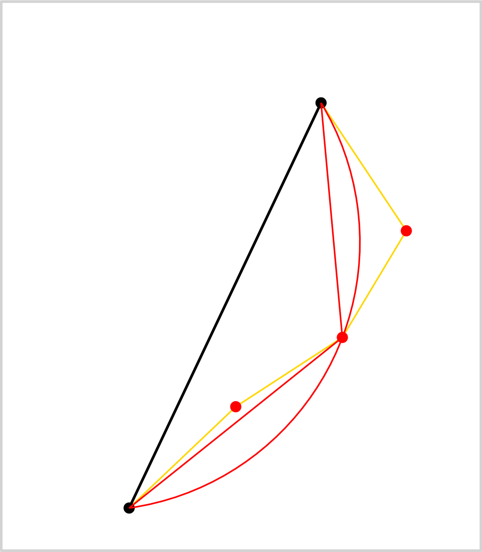
Removing Triangles



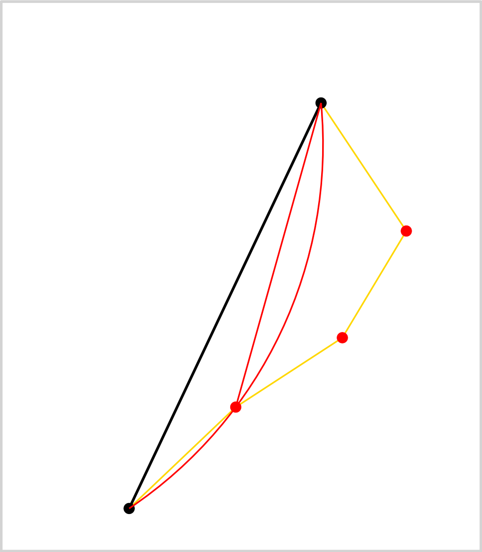
Removing Triangles



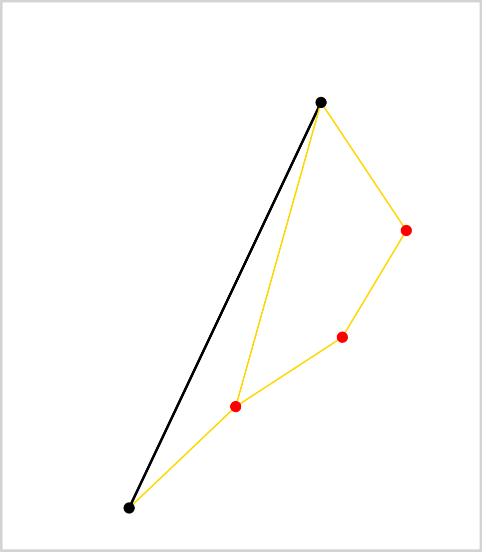
Removing Triangles



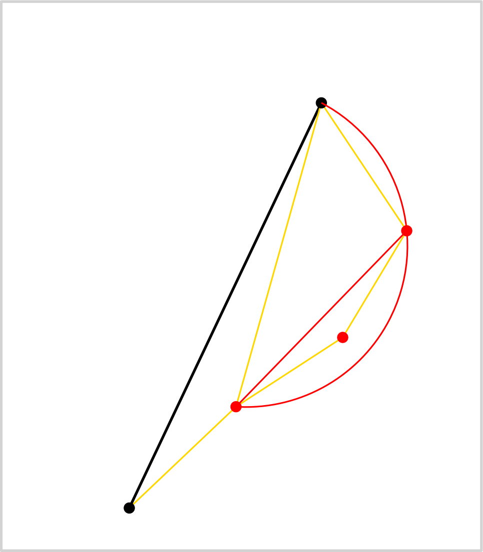
Removing Triangles



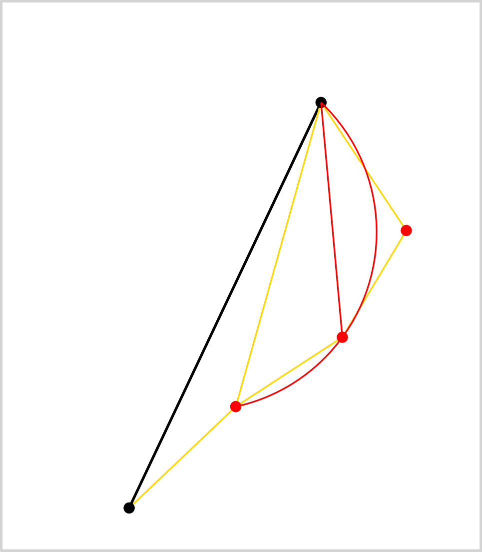
Removing Triangles



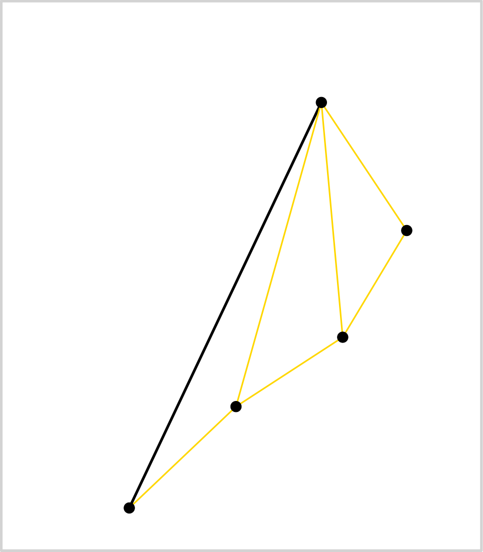
Removing Triangles



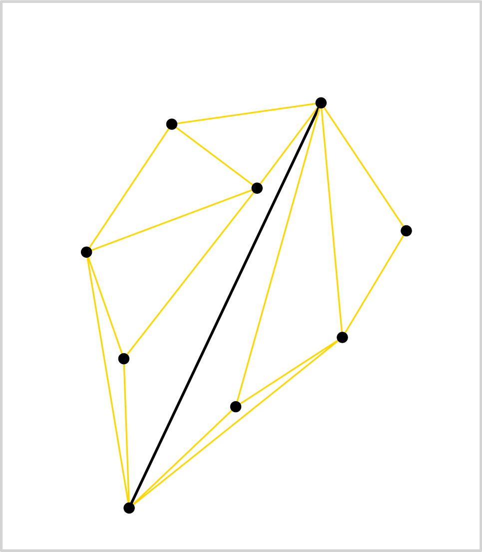
Removing Triangles



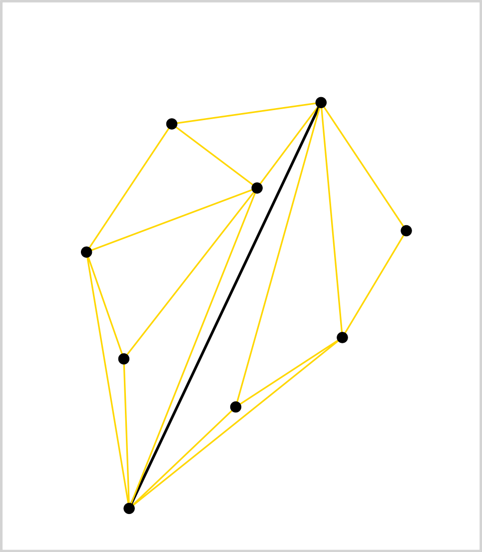
Removing Triangles



Removing Triangles



Removing Triangles



Removing Triangles Proof

1. The constrained edge must be an edge in the new CDT.
2. The removed edges can not belong to the new CDT because we need it to be a planar graph.
3. The remaining edges belonged to the previous CDT and therefore fulfil the empty circumcircle condition.
4. The new CDT can not contain edges crossing from the upper to the lower polygon since the new constraining edge prevent their visibility.
5. In consequence, the new CDT can be obtained by making a Delaunay triangulation of the upper and lower polygons.
6. By construction, (5) is guaranteed.

Table of Contents

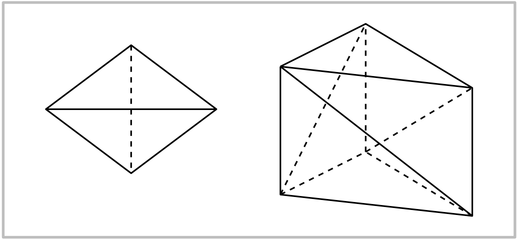
Definitions

- Let $\mathcal{V}(P)$ be a Voronoi diagram generated by $P = \{pi\} \in \mathbb{R}^m, m + 1 \leq n \leq \infty$, where the generators points are non-collinear.
- Let $Q = \{q_i\}, i \in 1..n_v$ be the set of Voronoi vertices in $\mathcal{V}(P)$.
- Let $F(q_i)$ be the set of p_j that generate the faces sharing q_i.
- Let CH_i be the convex hull of the vertices in $F(q_i)$.
- We call m-dimensional Delaunay Tessellation to the set $\mathcal{D}(P) = \{CH_i\}, i \in 1..n_v$.
- We call Delaunay Tetrahedralization to the 3-dimensional Delaunay Tesselation.

Properties

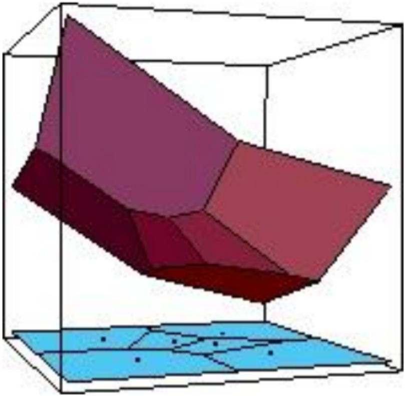
- The number of tetrahedra spanning n generator points is of the order n^2.
- Whilst Delaunay triangulation tends to avoid elongated and thin triangles as much as possible, this is not always true in 3D.
- There might not be a tetrahedral tessellation that satisfies a given constrain.
- There is no know algorithm that judges in polynomial time whether a given constrain admits a tetrahedralization. (This was back in 2000, Vera do you know anything more recent?)

Properties



Algorithm

- Unified approach to constructing d-dimensional Voronoi Diagrams: Solve the convex hull problem in $d+1$.



Algorithm

1. For every $p_i = \{x_1, x_2, ..., x_d\} \in P$:

$$p_i = \{x_1, x_2, ..., x_d\} \mapsto p_i^* = \{x_1, x_2, ..., x_d, x_1^2 + x_2^2 + ... + x_d^2\}$$

2. Build $CH(\{p_i^*\})$ using any algorithm in the literature.
3. Project all lower faces of $CH(\{p*_i\})$ in the direction parallel to the $(d+1)$st coordinates axis, onto the original d-dimensional space.

Algorithm

- Gift wraping $O(n^{\lfloor d/2 \rfloor + 1})$.
- Incremental $O(n^{\lfloor (d+1)/2 \rfloor} + n \log n)$.
- Shelling $O(n^{\lfloor d/2 \rfloor} \log n)$.
- Divide and Conquer $O(n^{\lfloor d/2 \rfloor} + n \log n)$.
- 3D Voronoi diagram can be built in $O(n^2)$.

Table of Contents

CGAL

"Never implement a geometric algorithm before checking if CGAL has it!
It is a reliable and efficient sofware!"

V. Sacristán (2014)

CGAL 4.5 - 2D Triangulation
http://doc.cgal.org/latest/Triangulation_2/index.html#Section_2D_Triangulations_Constrained
CGAL 4.5 - 3D Triangulations
http://doc.cgal.org/latest/Triangulation_3/index.html

Links and References

L Paul Chew.
Constrained delaunay triangulations.
Algorithmica, 4(1-4):97–108, 1989.

L. De Floriani and E. Puppo.
An on-line algorithm for constrained delaunay triangulation.
CVGIP: Graph. Models Image Process., 54(4):290–300, July 1992.
ISSN 1049-9652.
doi: 10.1016/1049-9652(92)90076-A.
URL http://dx.doi.org/10.1016/1049-9652(92)90076-A.

Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu.
Spatial tessellations: concepts and applications of Voronoi diagrams, volume 501.
John Wiley & Sons, 2009.