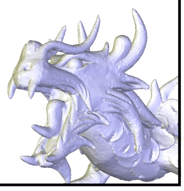
Lyon 1

Mesh and Computational Geometry

Raphaëlle Chaine
Université Claude Bernard Lyon 1

M2 ID3D
Image, Développement
et Technologie 3D
et 3A Centrale



1

Delaunay triangulation in any dimension n

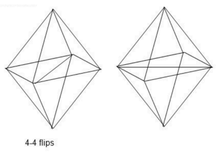
- Paving the convex-hull of the points with n-simplices whose circumscribed sphere is empty
- Warning: Lawson's algorithm is only valid in 2D, because the notion of edge flip is more complicated in larger dimensions

118

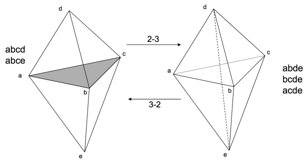
118

Flip in 3D

- 4-4 flip



- 2-3 and 3-2 flips



119

119

Geometric algorithms implementation

- Distinction between exact, combinatorial vs. approximate objects
- Input data (considered as exacts) used to construct exact non combinatorial objects
  - points x,y or x,y,z
- Construction of combinatorial objects from the exact ones
- Approximate objets should be constructed for visualisation purpose only

120

120

Algorithmic using predicates

- The progress of an algorithm should only depend on the sign of predicates evaluated accurately
  - Use of a controlled arithmetic
  - No use of inexact objects in the evaluation of predicates

121

121

Algorithmic using predicates

- Without these precautions, there is a risk of aberrant behaviour of a geometric algorithm
- Example: How to express the simple insertion algorithm in a 2D triangulation according to these criteria?

122

122

Algorithmic using predicates

- Algorithm of simple insertion in a 2D triangulation:
  - Using the three points orientation predicate
    - To perform inclusion tests in a triangle
    - To perform visibility tests on an edge of the convex envelope

123

123

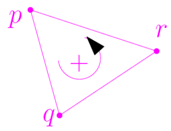
Algorithmic using predicates

- Three 2D points orientation predicate:

$$orientation(p,q,r) = sign(((q-p)\times(r-p)).Oz)$$

$$orientation(p,q,r) = sign((q_x-p_x)(r_y-p_y)-(q_y-p_y)(r_x-p_x))$$

$$= \begin{vmatrix} q_x - p_x & r_x - p_x \\ q_y - p_y & r_y - p_y \end{vmatrix}$$



$$orientation(p,q,r) = sign(\det \begin{bmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{bmatrix})$$

124

124

Algorithmic using predicates

- How to evaluate this orientation predicate?
  - In the case where the input coordinates belong to the regular grid of integers?
  - In the case where the input coordinates are rationals?

125

125

Algorithmic using predicates

- How to evaluate this orientation predicate?
  - In the case of the input coordinates belong tho the regular grid of integers?
  - In the case where the input coordinates are rationals?
  - In both cases the evaluation can be carried out accurately since we benefit from an exact multiplication, addition and subtraction for these types of numbers!

126

126

Algorithmic using predicates

- How to evaluate this orientation predicate?
  - In the case where input coordinates are double?

$$\pm m 2^e \qquad -1023 < e < 1024$$

$$m = 1.m_1 m_2 \dots m_{52} \ (m_i \in \{0,1\})$$

  - The result of the arithmetic operations is rounded to the nearest double

127

127

Algorithmic using predicates

- How to evaluate this orientation predicate?
  - In the case where input coordinates are double?
  - It is only the sign of the predicates that matters
  - The case where the three points are almost aligned can be error-prone

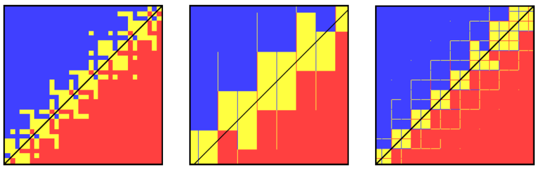
128

128

Algorithmic using predicates

- If the orientation predicate is evaluated using double arithmetic :
  - Result of orientation(p,q,r) with p(p_x+Xu_x, p_y+Yu_y)

$$0 \le X,Y \le 255 \qquad u_x = u_y = 2^{-53}$$



$p: \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$ $q: \begin{pmatrix} 12 \\ 12 \end{pmatrix}$ $r: \begin{pmatrix} 24 \\ 24 \end{pmatrix}$

$\begin{pmatrix} 0.50000000000002531 \\ 0.5000000000000171 \end{pmatrix}$ $\begin{pmatrix} 17.300000000000001 \\ 17.300000000000001 \end{pmatrix}$ $\begin{pmatrix} 24.00000000000005 \\ 24.0000000000000517765 \end{pmatrix}$

$\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$ $\begin{pmatrix} 8.8000000000000007 \\ 8.8000000000000007 \end{pmatrix}$ $\begin{pmatrix} 12.1 \\ 12.1 \end{pmatrix}$

Images by S. Pion

129

Algorithmic using predicates

- If the orientation predicate is evaluated using double arithmetic :
  - There are still values for which the sign is unambiguously certified (need to control the threshold)
  - Otherwise :
    - Consider the coordinates of p, q and r as rational (with a finer precision than that usually considered) and make the exact calculation

130

130

Algorithmic using predicates

- Example : How to express Lawson algorithm using predicates?

131

131

Algorithmic using predicates

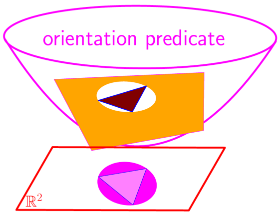
- Lawson Algorithm :
  - **Do not** base the evaluation of a predicate on the construction of an inaccurate temporary object (e. g. the centre of a circumscribed circle)
  - An AB edge should be flipped if the circle circumscribed to one of its 2 incident triangles ABC contains point D located on the other side

132

132

Algorithmic using predicates

- Reminder: the in-circle inclusion test can be expressed as an orientation test in a space of higher dimension (space of spheres)



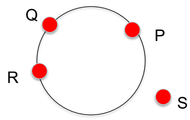
Images par O. Devillers

133

133

Algorithmic using predicates

- Predicate of inclusion of a point s in a circle circumscribed at p,q and r (orientation of the 4 points lifted on the paraboloid centered at p)



- In_cercle(p,q,r,s)
  =-signe(((Φ(q)-Φ(p))x(Φ(r)-Φ(p))).(Φ(s)-Φ(p))

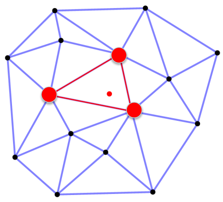
$$= - \,sign \begin{vmatrix} q_x - p_x & r_x - p_x & s_x - p_x \\ q_y - p_y & r_y - p_y & s_y - p_y \\ (q_x-p_x)^2 + (q_y-p_y)^2 & (r_x-p_x)^2 + (r_y-p_y)^2 & (s_x-p_x)^2 + (s_y-p_y)^2 \end{vmatrix}$$

134

134

Back to Delaunay triangulation

- How to modify the incremental algorithm of insertion into a simple triangulation to obtain an incremental algorithm of insertion into a Delaunay triangulation?

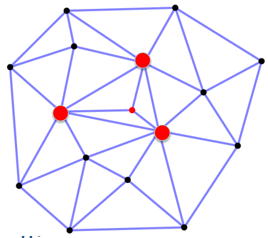


135

135

Incremental insertion into a Delaunay triangulation

- First perform a simple insertion :



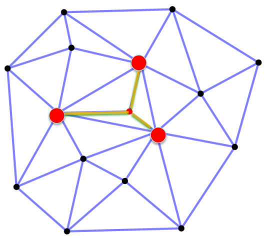
  - Use Lawson flips
  - Is it necessary to test all the edges?

136

136

Delaunay incremental insertion

- Let Δ be the triangle in which P was inserted
  - After the simple insertion, the triangle is star-shaped with respect to P

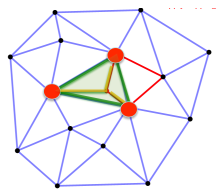


137

137

Incremental Delaunay insertion

- Let Δ be the triangle in which P was inserted
  - After the simple insertion, only the 3 edges of Δ are likely to be candidates for flipping
  - The 3 new edges incident to P could not be flipped
  - The others have not changed their pair of incident triangles

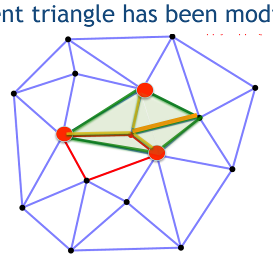


138

138

Incremental Delaunay insertion

- Each flip generates a new edge incident to P and two edges are added to the boundary of the modified area (in green)
- Green edges are likely to be flipped (since one of their incident triangle has been modified)

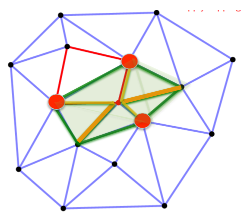


139

139

Incremental Delaunay insertion

- Each flip generates a new edge incident to P and two edges are added to the boundary of the modified area (in green)
- Green edges are likely to be flipped (since one of their incident triangle has been modified)

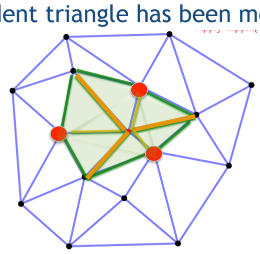


140

140

Incremental Delaunay insertion

- Each flip generates a new edge incident to P and two edges are added to the boundary of the modified area (in green)
- Green edges are likely to be flipped (since one of their incident triangle has been modified)



141

141

Incremental Delaunay Algorithm

- The complexity of each insertion directly corresponds to the final number i of edges incident at the new vertex.
  - Every flipped edge gave birth to an edge incident to P ->there are i-3 flips
  - The flipping test was checked on each edge that was effectively flipped, and also on the green boundary of the resulting modified area (i edges)

142

142

Incremental Delaunay Algorithm

- An insertion outside the convex envelope also starts as the simple insertion into a triangulation
  - Additional flips can be performed on the boundary of the modified area

143

143

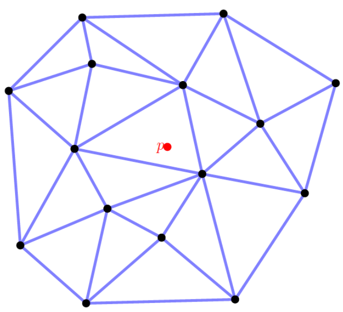
Delaunay incremental insertion (Alternative version)

- We just showed that the modified area of the triangulation is star-shaped around the inserted point P
- Alternative approach for incremental Delaunay :
  - Delete all the triangles whose circumscribed circle contains point P
  - Those triangles are said to be "in conflict" with P
  - Triangulate the conflict zone by star-shaping the conflict area around P

144

144

Incremental Delaunay Algorithm (Alternative Version)

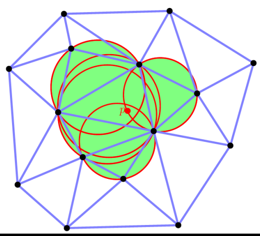


145

145

Delaunay incremental algorithm (Alternative Version)

- Determine the in-conflict triangles
  - Using breadth-first search (or deapth-first search) on the adjacency graph of triangles starting from Δ

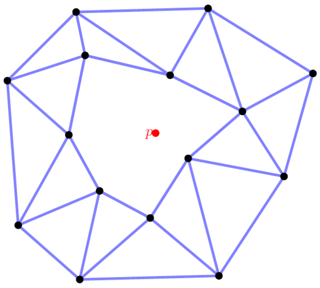


146

146

Delaunay incremental algorithm (Alternative Version)

- Conflict zone determination

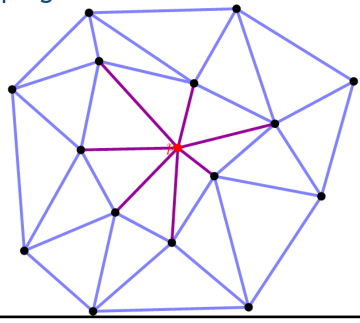


147

147

Delaunay incremental algorithm (Alternative Version)

- Star-shaping of the conflict area



148

148

Delaunay incremental algorithm (Alternative Version)

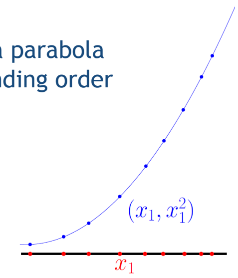
- Star-shaping of the conflict area
- Validity of this algorithm in higher dimension
  - In 2D, the **edges of the boundary** of the conflict zone get connected to the inserted point by constructing new triangles
  - In 3D, the Delaunay triangulation is composed of tetrahedrons. The **triangles of the boundary** of the conflict zone get connected to the inserted point by constructing new tetrahedrons

149

149

Incremental Delaunay Algorithm

- Complexity analysis
- Worst case :
  - Points distributed on a parabola and inserted in descending order of abscissa.

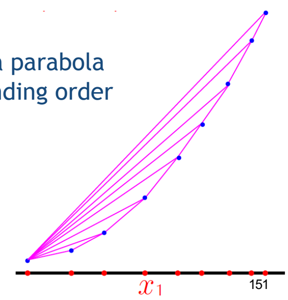


150

150

Incremental Delaunay Algorithm

- Complexity analysis
- Worst case :
  - Points distributed on a parabola and inserted in descending order of abscissa.



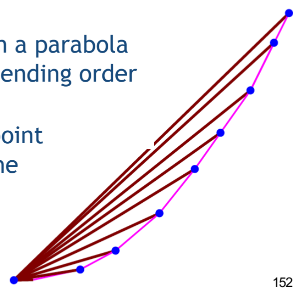
151

151

Incremental Delaunay Algorithm

- Complexity analysis
- Worst case :
  - Points distributed on a parabola and inserted in descending order of abscissa.
  - Each new inserted point conflicts with ALL the triangles

$\Omega(n^2)$



152

152

Incremental Delaunay Algorithm

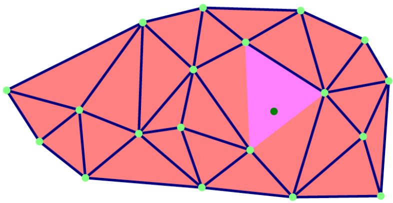
- Complexity analysis
- In average :
  - Complexity dependent on the strategy used to locate the triangle containing the point to be inserted

153

153

Location strategies

- Exhaustive search among all the triangles



$O(n)$

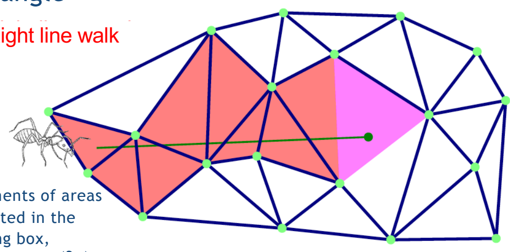
Images by O. Devillers

154

154

Location strategies

- Search from a first randomly selected triangle



3n elements of areas distributed in the bounding box, therefore squrt(3n) elements encountered on average on a straight line

$O(\sqrt{n})$ on average

155

155

Location strategies

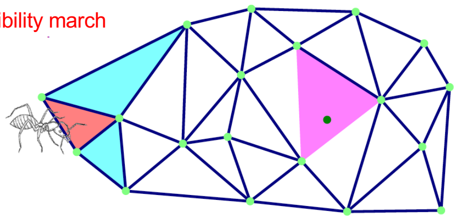
- Search from a first randomly selected triangle
  - Straight walk
  - Requires a predicate of segments intersection

156

156

Location strategies

- Search from a first randomly selected triangle
  - Some minor deviations from the straight line walk

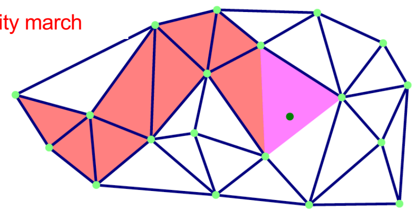


157

157

Location strategies

- Search from a first randomly selected triangle
  - Some minor deviations from the straight line walk



158

158

Location strategies

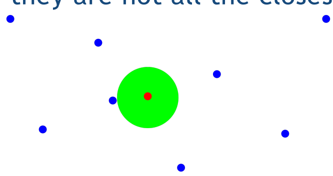
- Search from a first randomly selected triangle
  - Visibility walking
  - Only requires an orientation predicate to find the next triangle to walk in

159

159

Delaunay and proximity in space

- Delaunay triangulation allows to model the notion of proximity between points
- Each point is thus connected to nearby points around it
- Be careful, they are not all the closest!

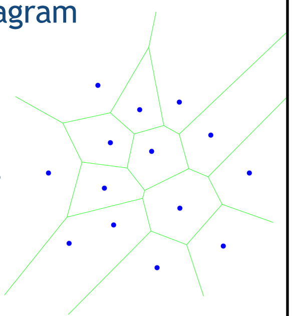


160

160

Voronoi diagram

- Voronoi's cell of a site P_i is the set of points closer to this site than to other sites



$$V_i = \{P \in \mathbb{R}^k \ t.\ que\ PP_i < PP_j \ pour\ tout\ j \neq i\}$$

161

161

Voronoi diagram

- Given a set E of points in $\mathbb{R}^k$, the partitioning of $\mathbb{R}^k$ into cells composed of points having the same nearest neighbour in E is called a Voronoi diagram of E

162

162

Voronoi diagram

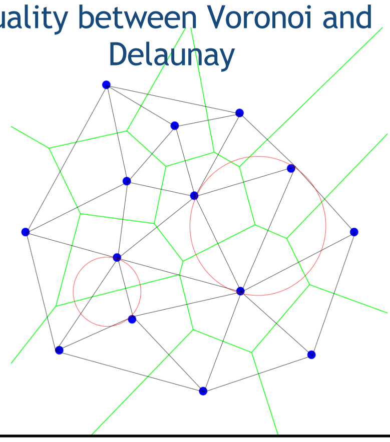
- Possible construction:
  - V_i : intersection of half-spaces h_{ij}^i where h_{ij} is the mediator of segment P_iP_j and h_{ij}^i is the half-space delimited by h_{ij} containing P_i

In practice we will proceed differently!

163

163

Duality between Voronoi and Delaunay



164

164

Duality between Voronoi and Delaunay

- Each Voronoi vertex is located at the center of the circumscribed circle of a Delaunay triangle
- Two Voronoi vertices are connected if they are associated with adjacent triangles

165

165

Coordinates of the centre of the circle circumscribed to a triangle ABC

- Useful for displaying the Voronoi diagram
- 1st possibility:
  - Write the equation of the mediator for each edge
    ex: For the edge AB, set of points M such that $MA^2=MB^2$
  - Solving a system of 2 equations with 2 unknowns (it is enough to take 2 mediators)
  - Numerically unstable

166

166

Coordinates of the centre of the circle circumscribed to a triangle ABC

- 2nd possibility :
  - Let's consider the angles
    $$\hat{A} = \widehat{CAB} \ \hat{B} = \widehat{ABC} \ \hat{C} = \widehat{BCA}$$
  - Then the barycentric coordinates of the centre H of the circumscribed circle with respect to A, B and C are elegantly expressed :

$$H(tan\hat{B}+tan\hat{C}, tan\hat{C}+tan\hat{A}, tan\hat{A}+tan\hat{B})$$

  - Reminder :

$$\tan(\widehat{ABC}) = \frac{\sin(\widehat{ABC})}{\cos(\widehat{ABC})} = sign((\overrightarrow{BC} \times \overrightarrow{BA}) \cdot \vec{k}) \frac{\left\|\overrightarrow{BC} \times \overrightarrow{BA}\right\|}{\overrightarrow{BC} \cdot \overrightarrow{BA}}$$

167

167

Coordinates of the centre of the circle circumscribed to a triangle ABC

$$H = Barycenter((A, \tan\hat{B} + \tan\hat{C}), (B, \tan\hat{C} + \tan\hat{A}), (C, \tan\hat{A} + \tan\hat{B}))$$

- Reminder :

$$\tan(\widehat{ABC}) = \frac{\sin(\widehat{ABC})}{\cos(\widehat{ABC})} = sign((\overrightarrow{BC} \times \overrightarrow{BA}) \cdot \vec{k}) \frac{\left\|\overrightarrow{BC} \times \overrightarrow{BA}\right\|}{\overrightarrow{BC} \cdot \overrightarrow{BA}}$$

$$Barycenter((A, \alpha a), (B, \alpha b), (C, \alpha c))$$
$$= Barycenter((A, a), (B, b), (C, c))$$

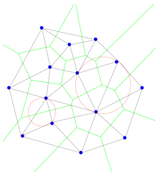
- Ensure to have no more denominators in the expression of your barycentric coordinates (normalization performed afterwards)

168

168

Duality between Voronoi and Delaunay

- Each Delaunay vertex is dual to one Voronoi cell
- Each Delaunay edge is dual to a Voronoi edge
- Each Voronoi vertex is dual to a Delaunay triangle



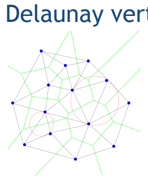
- What about Delaunay, Voronoi and their duality in 3D?

169

169

Duality between Voronoi and Delaunay

- Which data structure for Voronoi?
  - Walking around a Voronoi face is performed by walking through the faces/edges incident at a Delaunay vertex.
  - To move from one Voronoi cell to an adjacent cell is like moving from a Delaunay vertex to an adjacent vertex.



170

170