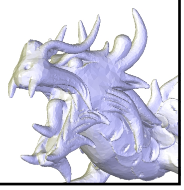
Lyon 1

Mesh and Computational Geometry

Raphaëlle Chaine
Université Claude Bernard Lyon 1

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

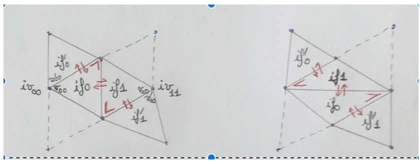
2024-25



1

Operations for triangulating and flipping edges

- Updating the Mesh data structure
  - Division of a triangle into 3 triangles (FaceSplit)
  - Division of an edge into two edges (EdgeSplit)
  - Flip of an edge (Flip)



124

124

Other use of the flip operation

- Insertion of a point P outside the convex-hull of a triangulation
  - New triangles joining P and the boundary edges that are visible from P
  - The infinite faces should be updated accordingly

125

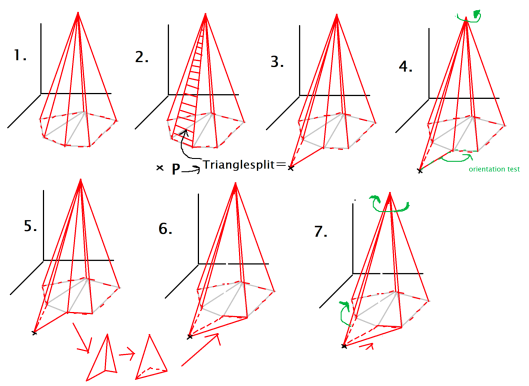
125

Other use of the flip operation

- Insertion of a 2D point P outside the convex-hull of a 2D triangulation
- A possible implementation using the infinite vertex and the flip operation
  - Let InfF be an infinite face incident to a visible edge of the convex-hull
  - Split InfF into 3
  - Iteratively flip the infinite edges bounding the modified area if they are incident to an other infinite face to be destroyed (starting from InfF)

126

126



127

127

Delaunay triangulation in any dimension n

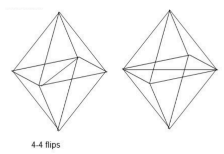
- Paving the convex-hull of the points with n-simplices whose circumscribed sphere is empty
- Warning: Lawson's algorithm is only valid in 2D, because the notion of flip is more complicated in larger dimensions

128

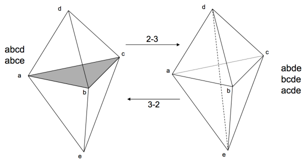
128

Flip in 3D

- 4-4 flip



- 2-3 and 3-2 flips



129

129

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

130

130

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

131

131

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?

132

132

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

133

133

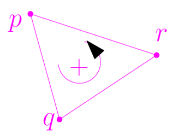
Algorithmic using predicates

- Three 2D points orientation predicate:

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

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

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



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

134

134

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?

135

135

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!

136

136

Algorithmic using predicates

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

$$\pm m 2^{e} \qquad -1023 < e < 1024$$
$$m = 1.m_1 m_2 \ldots m_{52} \ (m_i \in \{0,1\})$$

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

137

137

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
  - Is interval arithmetic enough to disambiguate the sign of a predicate?
  - The case where the three points are almost aligned can be error-prone

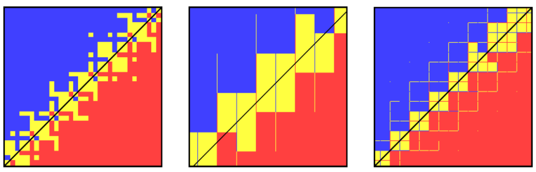
138

138

Algorithmic using predicates

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

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



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

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

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

Images by S. Pion

139

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

140

140

Algorithmic unsing predicates

- Example : How to express Lawson algorithm using predicates?

141

141

Algorithmic using predicates

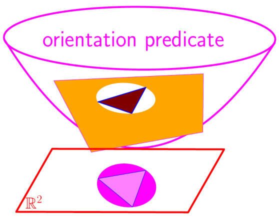
- Lawson Algorithm :
  - An AB edge should be flipped if the circle circumscribed to one of its 2 incident triangles ABC contains point D located on the other side
  - **Do not** use an inaccurate temporary object (e. g. the centre of a circumscribed circle) when evaluating a predicate

142

142

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)



in-sphere predicate

Images par O. Devillers

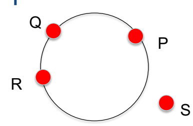
143

143

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)

  Φ = lifting operator



- In_cercle(p,q,r,s)

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

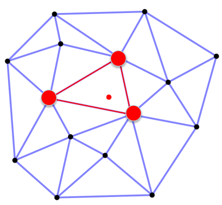
$$= - \mathrm{sign} \begin{vmatrix} q_x - p_x & r_x - p_x & s_x - p_x \\ q_y - p_y & r_y - p_y & s_y - p_y \\ (q_x - p_x)^2 + (q_y - p_y)^2 & (r_x - p_x)^2 + (r_y - p_y)^2 & (s_x - p_x)^2 + (s_y - p_y)^2 \end{vmatrix}$$

144

144

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?

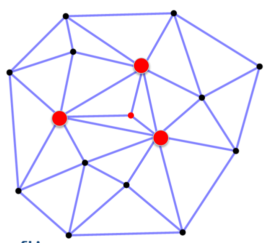


145

145

Incremental insertion into a Delaunay triangulation

- First perform a simple insertion :



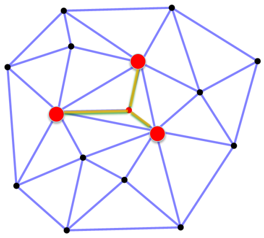
  - Use Lawson flips
  - Is it necessary to test all the edges?

146

146

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

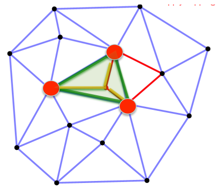


147

147

Incremental Delaunay insertion

- Let Δ be the triangle in which P was inserted
  - After the simple insertion, only the 3 edges of Δ could 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

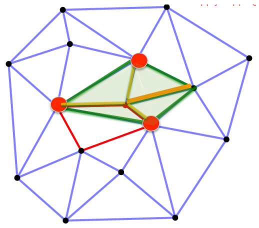


148

148

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)
- New green edges should be checked in turn (one of their incident triangles has been modified)

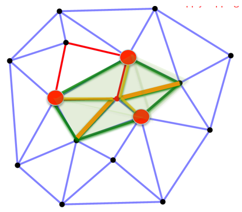


149

149

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)
- New green edges should be checked in turn (one of their incident triangles has been modified)

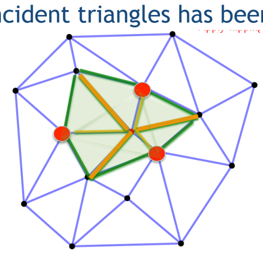


150

150

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)
- New green edges should be checked in turn (one of their incident triangles has been modified)



151

151

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)

152

152

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

153

153

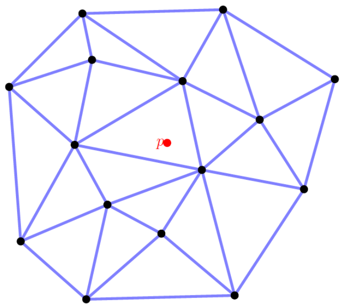
Delaunay incremental insertion (Alternative version)

- 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

154

154

Incremental Delaunay Algorithm (Alternative Version)

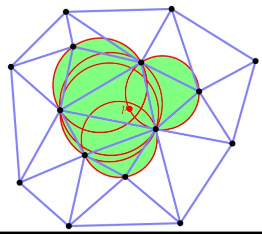


155

155

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

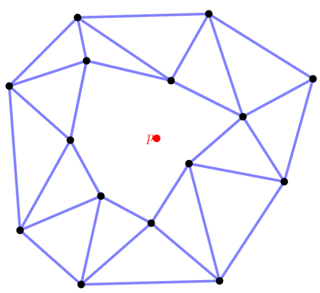


156

156

Delaunay incremental algorithm (Alternative Version)

- Conflict zone determination

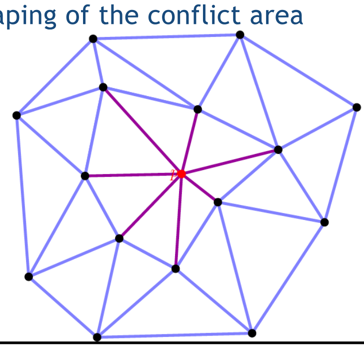


157

157

Delaunay incremental algorithm (Alternative Version)

- Star-shaping of the conflict area



158

158

Delaunay incremental algorithm (Alternative Version)

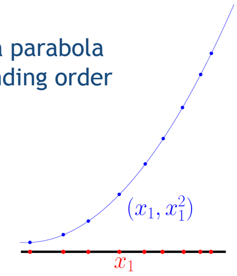
- Star-shaping of the conflict area
- Validity of this algorithm in higher dimension
  - In 2D, the **edges of the boundary** of the conflict zone get connected to the inserted point by constructing new triangles
  - In 3D, the Delaunay triangulation is composed of tetrahedrons. The **triangles of the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

159

159

Incremental Delaunay Algorithm

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

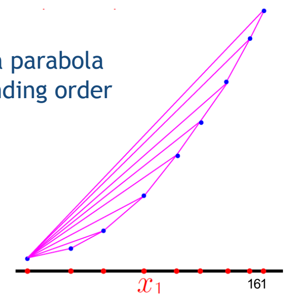


160

160

Incremental Delaunay Algorithm

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



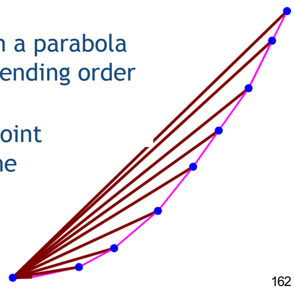
161

161

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

$\Omega(n^2)$



162

162