VORONOI DIAGRAMS AND DELAUNAY TRIANGULATIONS

The documentation is available here: http://www.cgal.org/ then follow the links Package Overview/ Triangulations and Delaunay Triangulations / 2D Triangulations / Delaunay triangulations

The application is events based, and can insert interactively new points in a Delaunay triangulation.

1) browse the code of the application, to understand the events mechanism.

Remark: the class template DT (in “dt.h”) derives from a Delaunay triangulation class, and has been already enriched with two render methods, and a method that generates random points.

WARMING

2) Complete the code that displays the edges of the Voronoi diagram.

3) Add an option to display in red the edges of the convex hull (without using convex_hull()). Add a keyboard shortcut “C”.

4) Enlighten the face hovered by the mouse, using the method locale, and the “mouseMove” event, by drawing an OpenGL primitive of type GL_TRIANGLE.

5) (optional) Given a line defined by two points, display the set of faces intersected by the line, using the Line_face_circulator.

6) Enlighten the vertex nearest to the mouse, using the nearest_vertex method, and the “mouseMove” event.

SIMPLIFIED LLOYD ALGORITHM

The Lloyd algorithm interactively moves the vertices of a Delaunay triangulation to the barycenter of their Voronoi cell.

7) Compute the barycenters of the Voronoi cells

a few advices:

First set up a method:

```
Point centroid_voronoi_cell(Vertex_handle v)
```

that circulates of the neighboring faces of v (using Face_circulator) and call dual(Face_handle) if f is finite. Si f is infinite, then return v->point(), because the barycenter of the Voronoi cell is not defined.

8) Add an option to display the Voronoi cells barycenters

9) Implement the lloyd() method.