Meshes and Computational Geometry
Master 2 Informatique ID3D – Université Lyon 1- Ecole Centrale de Lyon

TP4 – Split triangles into 3, Flip edges, Elementary geometric predicates

Your triangular mesh data structure can be used to store triangulations of the surface of an object (3D points) as well as triangulations of points in 2D (z=0). The aim of this session is to add several elementary operations to your triangular mesh data-structure.

General operations

- **Triangle split**

Complete your `mesh` and/or `face` classes with an operation that splits a triangle `face` into 3 by insertion of a new vertex that is located at the position provided in parameter of the member function. Ensure that the integrity of the data structure is well preserved.

- **Edge split**

This operation is useful to insert a vertex within an existing edge. Complete your `mesh` and/or `face` classes with an operation that splits an edge into 2 by insertion of a new vertex that is located at the position provided in parameter of the member function. The incident `faces` are also divided into two `faces`. Ensure that the integrity of the data structure is well preserved.

- **Edge flip**

Complete your `mesh` classes with an operation that flips the edge shared by two neighboring triangular faces. Once again, you should ensure that the integrity of the data structure is well preserved.

Note that there are several ways to specify an edge in the face-based data structure (which means you can provide several overloads of the flip and split operations).

Operations for 2D or terrains triangulations

- **Orientation test**

Given three 2D points (z=0), write the geometric predicate returning a positive value if these 3 points are oriented counter-clockwise, negative if they are oriented clockwise or 0 if they are aligned for the arithmetic used to encode the coordinates.

- **“In triangle” test**

Given a 2D triangle, write the geometric predicate returning a positive value if an input point is located inside a triangle, negative if it is located outside or 0 if it is located on the boundary.

- **Naive insertion of a point in a 2D triangulation**

Complete your `mesh` class to provide a member function that inserts a point in a 2D triangulation, with no care of triangles quality. For insertion outside the convex hull, infinite edge flip operations can be used, or a bounding box can be used.

Improve the quality of a triangulation, Delaunay triangulation

- Provide a predicate function to check whether an edge is locally Delaunay.
- Provide a procedure that turns a 2D naïve triangulation of a set of 2D points into a Delaunay triangulation. You should implement Lawson's algorithm that we have studied in the class.
- Given a Delaunay triangulation, provide a way to update it after the insertion of a point using Lawson's algorithm only on a small subset of the edges.