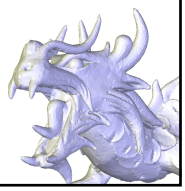
Lyon 1

Mesh and Computational Geometry

Raphaëlle Chaine
Université Claude Bernard Lyon 1

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



1

Data structures

- How to store a triangular mesh so that one can easily navigate through it?

30

30

Data structures

- Geometric information :
  - coordinates of vertex positions
- Topological information :
  - incidence and adjacency relationships between vertices, edges and faces

31

31

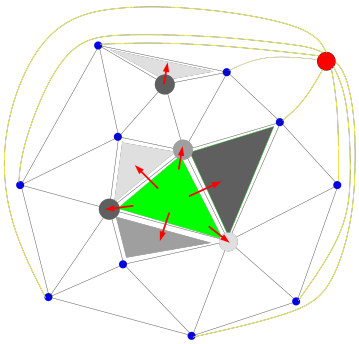
Data structures

- Triangle and vertex based representation (only for triangulated mesh)
  - Triangle :
    - Access to the 3 incident vertices (trigonometric order)
    - Access to the 3 adjacent triangles

    Constraint : vertex i facing adjacent triangle i
  - Vertex :
    - Access to 1 incident triangle
    - Access to the underlying point (geometry)

32

32



33

33

Data structures

- Case of a 2D triangulation
  - Dangling pointers / indexes on the boundary of the convex hull

  OR
  - Addition of a fictitious vertex (called infinite vertex) whose incident triangles are attached to the edges of the convex envelope
- The data structure can also be used for surface triangulations
  - Use dangling pointers/indexes for each hole boundary

    OR
  - Add fictitious vertices for each hole

34

34

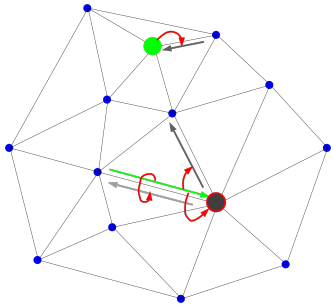
Data structures with edges

- Representation based on ½ edges and vertices
  - ½ edge :
    - Access to the coupled ½ edge
    - Access to the next ½ edge
    - Access to the target vertex
  - Vertex :
    - Access to an ½ edge oriented towards the vertex
    - Access to the underlying point

35

35

Data structures



36

36

Naive and uncompressed format of a mesh in a file

- Files off
- Format description
  - Number of vertices s
  - Number of faces c
  - Vertices coordinates
  - Description of faces (from 0 to c-1)
    - Number of vertices in the face
    - Description of the face by the indexes of its vertices (in counter-clockwise direction in 2D, or by orienting the faces towards the outside in 3D)

37

37

How to load a mesh into the data structure?

- Need to find adjacencies between faces
- Using a *map* while reading faces
  - Associating to each edge encountered (key = pair of vertex indexes) the index of the current face (+ optionally the relative index of the vertex opposite to the edge in the face)
  - When an edge is incident to two faces, the adjacency between these two faces is reported in the structure

38

38

How to load a mesh into a data structure?

- Using a *map* while reading faces
- Complexity in O(s*log(s)) where s number of vertices
- Rque : If you use an hmap instead of a map, you get an almost linear complexity.

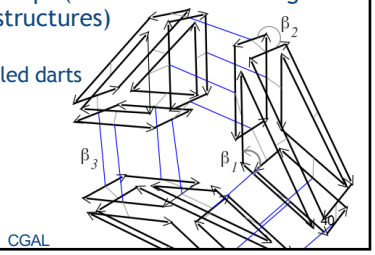
39

39

Data structures

- Case of a 3D triangulation
  - Structure based on tetrahedrons and vertices

  OR
  - Combinatorial maps (extension of half edge or dart-based structures)
    - β_1 : next
    - β_2 et β_3 : coupled darts



40

40

Data structures

- Case of nD triangulations
  - n-simplex and vertex based structures
  - Combinatorial Maps
    - β_1 : next
    - β_2, β_3 … β_n : coupled darts

41

41