Lecture 3
Half-edge Data Structure Review, Geometry of Curves and Surfaces (I)
Overview

- Quick review of triangle mesh representation and halfedge data structure
- Some exercise
- Basic Geometry of Curves and Surfaces
Halfedge Data Structure Representation

- Why do we need it?
  - Compared with using the vertex table + face table directly?
- Incidence information of the connectivity is stored as a big network indexed by the halfedges:
  - Each face has how many halfedges? How many halfedge pointers does a face keep?
  - Each edge has how many halfedges? How many halfedge pointers does an edge keep?
  - Each vertex has how many halfedges? How many halfedge pointers does a vertex keep?
  - Each halfedge has: next(), prev(), target(), source(), face(), edge(), twin()...
Using Halfedge Data Structure (1)

- Examples:
  1. How to check whether a face is on the boundary?
  2. How to check whether an edge is on the boundary?
  3. How to check whether a vertex is on the boundary?
    1) How to find a “most-clockwise-in-halfedge” of a boundary vertex? (How to rotate a halfedge \texttt{he0} clockwise (clw) about its target?)
    2) How to find a “most-clockwise-out-halfedge” of a boundary vertex?
    3) How to find a “most-counterclockwise-in-halfedge” of a boundary vertex?
    4) How to find a “most-counterclockwise-out-halfedge” of a boundary vertex?
Using Halfedge Data Structure (2)

Solution for 3. 1):

```c
Halfedge * MostCLWHalfedgeAboutTarget(Halfedge * he0) {
    Halfedge * he1 = he0; // the result is stored in he1
    Halfedge * nhe = he1->next()->twin();
    Do while (nhe & nhe!=he0) {
        he1 = nhe;
        nhe = nhe->next()->twin();
    }
    Return he1;
}
```
Example 4. How to find the one-ring neighboring vertices of a vertex $v$?

**Example codes to print one-ring vertex of a given vertex “cv”**
(Method 1: Try to traverse using the half-edge data structure)

```cpp
Vertex * cv;
...
Halfedge * he0 = cv->he();
Halfedge * he = he0;
Do {
    Vertex * v = he->source();
    std::cout << v->id() << std::endl;
    he = he->he_next();
    he = he->he_twin();
} while (he!=he0);
```
Using Halfedge Data Structure (4)

Example 4. How to find the one-ring neighboring vertices of a vertex \( v \)?
(the previous method has been implemented in the iterator classes.)

**Example codes to print one-ring vertex of a given vertex “cv”**
(Method 2: Using the “iterator” class, when you have the mesh library)

```cpp
Vertex * cv;
...
For (VertexVertexIterator vvit(cv); !vvit.end(); ++vvit){
    Vertex * v = * vvit;
    std::cout << v->id() << std::endl;
}
```
Using Halfedge Data Structure (5)

Example 5. How to Travel along the boundary loop?

Example codes to traverse the boundary
(given a boundary halfedge he0, print all following halfedges in the boundary loop):
Halfedge * he0; // suppose it is a boundary halfedge
Halfedge * he; //the current boundary halfedge to print
Do {
    std::cout << he << std::endl;
    he=MostCLWHalfedgeAboutTarget(he);
    he=he->next();
} while (he!=he0)
Using Halfedge Data Structure (6)

More Complex Example: How to do subdivision:

One example:

Another example:

**to split a face (type 2)**

Face * f0;  // the face we want to split
...
- Create a new vertex nv ← the mass center of f0
- Create three new edges, six new halfedges
- Update half-edges, forming three cycles
- Create three new faces, link edges, halfedges accordingly
- Delete the original face