**Working with Meshes**

CS 176 Winter 2011

---

**Surfaces/Meshes**

We’ll stick to triangles

CS 176 Winter 2011

---

**Discrete Surfaces**

Setup

- topology & geometry
- simplicial complex: “triangle mesh”
- 2-manifold \( K = \{V, E, T\} \)
  \( V = \{v_i\} \quad E = \{e_{ij}\} \quad T = \{t_{ijk}\} \)  
- Euler characteristic
  \( F - E + V = 2(1 - g) = \chi \)

CS 176 Winter 2011

---

**What’s a Mesh?**

Formally

- abstract simplicial complex \( K \)
  - singletons, pairs, triples, ... of integers
    \( V = \{1, 2, 3, \ldots\} \quad E = \{\{i, j\}, \{k, l\}, \ldots\} \)
    \( F = \{\{i, j, k\}, \{j, i, l\}, \ldots\} \)
  - containment property
    \( \rho \in K \wedge \sigma \subseteq \rho \Rightarrow \sigma \in K \)
  - partial order \( \preceq \), face, coface, \( \emptyset \)

CS 176 Winter 2011

---

**Simplicial Complex**

Topological realization

- identify \( V \) with unit vectors in \( \mathbb{R}^N \)
  \( |K| = \bigcup_{\sigma \in K} |\sigma| \)  
- subset topology of ambient space
- closure, star, and link
  \( ClL = \{p|\rho \subseteq \sigma, \sigma \in L\} \)
  \( StL = \{p|\sigma \subseteq \rho, \sigma \in L\} \)  
  \( L - \emptyset \)

CS 176 Winter 2011

---

**Topological Structure**

2-manifold (with boundary)

- every point has an open, (half-) disklike subset surrounding it

\[ |K| \text{ 2-manifold iff } |St v| \approx \mathbb{R}^2 \]

\[ |St \sigma| = \bigcup_{\rho \in St \sigma} \text{ int}|\rho| \]

CS 176 Winter 2011
**Topological Invariants**

Euler characteristic
- for surfaces: $F-E+V=\chi=2(g-1)$
- not required to be simplicial
- more generally for simplicial complexes
  \[ \chi(K) = \sum_{\emptyset \neq \rho \in K} (-1)^{\dim \rho} \]

**Simplicial Complex**

Geometric realization
- the concrete embedding $\pi_v(K)$
  \[ \pi_v : \mathbb{R}^n \rightarrow \mathbb{R}^3 \]
- vertex images specify everything
- piecewise linear approximation
- presumably approximation of underlying smooth surface

---

**Mesh Structure**

Input
- typically
  - list of vertices (how long?)
  - list of triangles (until EOF)
- need to build mesh structure
  - infer topology
  - check topology
  - oriented (orientable?)

**Building the Mesh**

What do we need?
- array of pointers to vertices
- choices for basic topology primitive
  - (half-)edges
  - different variants
  - triangles
  - we'll use triangles

---

**Types of Operations**

What do we need to support?
- iterate over all vertices (easy)
- iterate over all triangles (easy)
- for a triangle visit
  - incident vertices (easy)
  - incident triangles (easy)
Types of Operations

What about edges?
- visit all edges
- not explicitly represented...
- do we need edges? Yes!
- discover triangle adjacencies
- map pairs of integers to triangles
  \[ e_{ij} \mapsto \{t_{ijk}, t_{jil}\} \]

Operations to Support

For later (think about it now...)
- edge collapse
- legality?
- edge flip

Data Structures

Triangles
- consistent ordering of vertex and triangle incidences

```c
Triangle {
  Vertex *v[3];
  Triangle *t[3];
}
```
- triangles across from vertices

What Data Where?

Attributes
- normal, color, texture coordinates
- later: forces, velocities, mass
- why not just lay everything out in arrays?
- changes in structure!
- very hard to debug...

Examples

Vertex normals
- gradient of volume
  \[ n_i = 1/2 \sum_{t_{ijk}} (p_j - p_i) \times (p_k - p_i) \]

```
\forall v_i : n_i = 0
\forall t_{ijk} : a_{ijk} = (p_j - p_i) \times (p_k - p_i)
\forall t_{ijk} :
  a_{i+} = a_{ijk}
  a_{j+} = a_{ijk}
  a_{k+} = a_{ijk}
\forall v_i : N_i = n_i/|n_i|
```

Example

Gaussian curvature

\[ \forall v_i : K_i = 2\pi - \sum_{t_{ijk}} \alpha_{ijk} \]
\[ \forall v_i \in V \setminus \partial V : K_i = 2\pi \]
\[ \forall v_i \in \partial V : K_i = \pi \]
\[ \forall t_{ijk} :
  K_{ij} = \text{atan2}(|a_{ijk}|, (p_j - p_i) \cdot (p_k - p_i))
  K_{ji} = \text{atan2}(|a_{ijk}|, (p_k - p_i) \cdot (p_j - p_i))
  K_{kj} = \text{atan2}(|a_{ijk}|, (p_i - p_k) \cdot (p_j - p_k))
\]
**Principles**

As you write code...
- assumptions are ok, but you must assert them explicitly
- orientability
- 2-manifold property
- avoid storing the same information multiple times
- nasty to keep current under changes

---

**Other Tricks**

As you write code
- use two sided lighting
- abstract the iterators!
  - what about boundary vertices?
- keep iterators sorted
  - interior then boundary vertices
  - interior then boundary triangles