Mesh Representation & Modeling
Polygonal Mesh

- Geometric object made of vertices, edges and faces
  - Polyhedron
    - Pyramid
    - Cube
    - Sphere
  - Can also be 2D (although much less interesting)

- Faces are polygons
  - Triangular mesh
  - Quad mesh
  - General n-gons
Polygonal Mesh

- **Vertex** – a point in space
  - Can include more info like color and normal
- **Edge** – a segment between two vertices
- **Face** – a polygon defined by a set of vertices and edges
2-Manifold Polygonal Mesh

- Every edge has at most two faces
  - Exactly two faces if the mesh is closed

- Euler invariance

\[ V - E + F = 2(1 - g) \]

- Where:
  - \( V \) – number of vertices
  - \( E \) – number of edge
  - \( F \) – number of faces
  - \( G \) – mesh genus
Why do we need a data structure?

- To store the mesh
- To make queries
  - Which vertices are connected to vertex $v$?
  - What are the neighbors of face $f$?
- Geometric operation
  - Add vertices
  - Split edges
Polygon Soup

- Face list
  - Lists of coordinates
- Polygons are unrelated

<table>
<thead>
<tr>
<th>face</th>
<th>vertices (ccw)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$(v_1, v_2, v_3)$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>$(v_2, v_4, v_3)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Vertex-Vertex

- Vertex list
  - Vertex coordinates
  - Lists of connected vertices

Edges and faces are implicit
Vertex-Vertex

- Vertex list
  - Vertex coordinates
  - Lists of connected vertices - Adjacency matrix

- Edges and faces are implicit
Face-Vertex

- **Vertex list**
  - Vertex coordinates
  - Lists of incident faces

- **Face list**
  - Lists of incident vertices

---

Vertex list:

- $v_1$: $(x_1, y_1, z_1)$
- $v_2$: $(x_2, y_2, z_2)$
- $v_3$: $(x_3, y_3, z_3)$

Face list:

- Face $f_1$: $(v_1, v_2, v_3)$
- Face $f_2$: $(v_2, v_4, v_3)$
- ...
Winged-Edge


- **Vertex list**
  - Vertex coordinates
  - Lists of incident edges

- **Edge list**
  - List of incident vertices, edges and faces

- **Face list**
  - Vertex coordinates
  - Lists of incident edges

<table>
<thead>
<tr>
<th>vertex</th>
<th>coordinates</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>$(x_1, y_1, z_1)$</td>
<td>$(e_1, e_2)$</td>
</tr>
<tr>
<td>$v_2$</td>
<td>$(x_2, y_2, z_2)$</td>
<td>$(e_2, e_3)$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>$(x_3, y_3, z_3)$</td>
<td>$(e_1, e_5)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>face</th>
<th>vertices (ccw)</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$(v_1, v_2, v_3)$</td>
<td>$(e_1, e_2, e_3)$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>$(v_2, v_4, v_3)$</td>
<td>$(v_3, v_4, v_5)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Winged-Edge

- **Vertex list**
  - Vertex coordinates
  - Lists of incident edges

- **Edge list**
  - List of incident vertices, edges and faces

- **Face list**
  - Vertex coordinates
  - Lists of incident edges

![Diagram showing vertex and edge connections]

<table>
<thead>
<tr>
<th>Edge</th>
<th>Vertices</th>
<th>Edges</th>
<th>Faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>$(v_1, v_3)$</td>
<td>$(e_2, e_3)$</td>
<td>$(f_1)$</td>
</tr>
<tr>
<td>$e_2$</td>
<td>$(v_1, v_2)$</td>
<td>$(e_1, e_3)$</td>
<td>$(f_1)$</td>
</tr>
<tr>
<td>$e_3$</td>
<td>$(v_2, v_3)$</td>
<td>$(e_1, e_2, e_4, e_5)$</td>
<td>$(f_1, f_2)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Normal Estimation

- Normals are not always available
  - Face normals can always be computed
  - Vertex normals are not even defined
    - No correct answer.
- Estimate vertex normals as if shape was smooth
  - Vertex normal is *average* of incident face normals
    - What if some faces are larger than others?
  - Vertex normal is *weighted average* of incident face normals
    - Weights are face areas

Mesh Modeling: Center for Graphics and Geometric Computing, Technion
Smoothing

- Move “pointy” vertices to make the mesh smooth
  - New vertex position is average of neighbors positions
  - Side effect – Mesh shrinkage
    - What will happen after infinite iterations?
Basic Operations

- Each element can be individually transformed
  - Also groups of elements
- Moving vertices
Basic Operations

- Each element can be individually transformed
  - Also groups of elements
- Moving edges
Basic Operations

- Each element can be individually transformed
  - Also groups of elements
- Moving faces
Soft Selection

- Select a vertex, it’s vicinity gets “partially” selected
  - When the vertex is moved, the vicinity makes a “partial” move
Soft Selection

- Select a vertex, its vicinity gets “partially” selected
  - When the vertex is moved the vicinity makes a “partial” move
Refining Meshes

- Edge Split
  - Create a new vertex dividing an edge

- Face Split
  - Create a new edge dividing a face

- And many more
  - Extrusion, Chamfer, Fillet …