3D objects representation and data structure

Dr Nicolas Holzschuch
University of Cape Town
e-mail: holzschu@cs.uct.ac.za
Map of the lecture

• Object representations in 3D
  – internal/external

• Data structure
  – vertex list, edge list, winged edge

• Plane equation
  – computing the plane equation
Objects representation

• In application model
• Will be modified by the application
• Ultimately, will be send for display
• Must be adapted for both tasks
Sending to display

• Need the list of faces, with list of vertices for each face
• Redundant information not a problem
  – one vertex may appear several times
• Neighbouring information not needed
• Data structure:
  – list of faces
  – list of vertices for each face
Access by the application

• Modification of the application model
  – moving vertices
  – adding new faces
  – adding new vertices
  – redundant information is excluded

• Needs neighbouring information
  – for coloring
  – for computing average normals (shading)
Internal vs. external

• External data structure:
  – used for display
  – can be very simple

• Internal data structure
  – will have complex manipulations
  – must provide for these manipulations
Vertex list

• List of faces
• For each face:
  – list of pointers to vertices
• Good points:
  – redundancy removed. Space saved.
• Bad points:
  – find which polygons share an edge, or a given vertex?
Edge list

• For each face:
  – list of pointers to edges

• For each edge:
  – the two polygons sharing it
  – the two vertices

• Neighbouring information available
  – faces adjacent to an edge
Edge list: shortcomings

- List of polygons sharing a vertex?
- I move a vertex:
  - I need to find which edges share this vertex
  - must go through the whole list
- I add a face, an edge:
  - must go through the whole list
Winged-edge data structure

Polygon 1

V1

E2

Edge 1

V2

E4

Polygon 2

E3

E5
Winged-edge data structure

• Each vertex also has a pointer to one of its edges
• Each face has a pointer to one of its edges
• Efficient:
  – faces adjacent to one vertex
  – edges adjacent to one vertex
Plane equation

• Each face is a planar polygon
• Plane equation: $ax+by+cz+d = 0$
• Normal to the plane: $(a,b,c)$
• Finding the equation?
Finding a plane equation

- Plane defined by three points: P1, P2, P3
- First, find the normal:
  \[ \mathbf{n} = \mathbf{P1P2} \times \mathbf{P1P3} \]
- If \( \mathbf{n} = 0 \), then it isn’t a plane
- \( \mathbf{n} \) gives \( a, b \) and \( c \)
- Find \( d \) using P1
Using a plane equation

• \( \mathbf{n} \) is fundamental:
  – defines a front and a back
  – \( M \) is in front of the plane: \( \mathbf{P}_1 \mathbf{M} \cdot \mathbf{n} \geq 0 \)
  – \( M \) is behind the plane: \( \mathbf{P}_1 \mathbf{M} \cdot \mathbf{n} \leq 0 \)

• Same classification using the equation:
  \[ ax + by + cz + d \geq 0 \]
3D objects: conclusion

• Data structure essential
  – must be adapted to the task
  – trivial data structure sufficient for display
  – more complex data structure required for application

• Plane equation