Chapter 6. 3D modeling
  6.1 Introduction
  6.2 Planar surface modeling
  6.3 Solid modeling
6.1 Introduction

An scene can contain different type of objects (clouds, trees, stones, buildings, furniture etc.). For all of them, a wide variety of representation models are available

- Polygonal surfaces and quadrics
- Spline surfaces
- Volumetric models
- Solid modeling (boundary B-Rep, spatial division Octrees)
- Procedural models (fractals, particle systems,…)
- Physic based modeling
6.2 Planar surface modeling

• Wireframe modeling
  – Elements: points, lines, arcs and circles, conic and curves
  – Advantages: easy to build, low memory requirements and storage
  – Disadvantages:
    • ambiguous representation (hidden-lines removal algorithms)
    • lack of visual coherence (line-inclusion algorithms)
6.2 Planar surface modeling

• Polygonal modeling
  – *Polygon mesh*: vertex, edges and polygon collection
    where each edge is shared by two polygons as maximum
    • vertex: point with coordinates \(x, y, z\)
    • edge: line segment that joins two vertices
    • polygon: close sequence of edges
  – There are different type of representation that can be used at the same time in a same application
    • Explicit
    • Pointers to list of vertices
    • Pointers to list of edges
  – Criteria to evaluate different representations:
    • time
    • space
    • topological information
6.2 Planar surface modeling

• Explicit representation
  – Each polygon is represented by a list of vertex coordinates
    \[ P = ((x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)) \]
  – Vertices are stored in order (clockwise or counter-clockwise)
  – Shared vertices are duplicated
  – There is no explicit representation for shared vertices and edges

– Advantages:
  • Efficient representation for individual polygons

– Problems:
  • High storage cost
  • In order to move a vertex, it is necessary to traverse all the polygons
  • If the edges are drawn, the shared ones are drawn twice

Convex polygon

Concave polygon
6.2 Planar surface modeling

- Pointers to list of vertices
  - Each vertex is stored once in a list

\[ V = ((x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)) \]

- A polygon is defined as a list of indexes (or pointers) to the list of vertices

\[ P_1 = (1, 2, 4) \]
\[ P_2 = (4, 2, 3) \]

- Advantages
  - Each vertex is stored just once
  - Coordinates of vertices can be easily changed

- Problems
  - Difficult to find polygons that share an edge
  - Shared edges are still drawn twice
6.2 Planar surface modeling

- Pointers to list of edges
  - Again a list of vertices
  - A polygon is represented as a list of indexes to a list of edges
  - Each edge points to two vertices and to the polygons it belongs to
- Advantages
  - Each vertex is stored just once
  - The shared edges are drawn just once
- Problem
  - Difficult to determine which edges share a vertex (in all the representation seen, in fact)

\[
E_1 = (V_1, V_2, P_1, \lambda) \\
E_2 = (V_2, V_3, P_2, \lambda) \\
E_3 = (V_3, V_4, P_2, \lambda) \\
E_4 = (V_4, V_2, P_1, P_2) \\
E_5 = (V_4, V_1, P_1, \lambda) \\
P_1 = (E_1, E_4, E_5) \\
P_2 = (E_2, E_3, E_4) \\
V = (V_1, V_2, V_3, V_4) = ((x_1, y_1, z_1), \ldots, (x_4, y_4, z_4))
\]
6.2 Planar surface modeling

- Polygonal meshes
  - Triangle strip
    - For n vertices, produces (n-2) connected triangles
  - Triangle fan
    - For n vertices, produces (n-2) connected triangles
  - Mesh of quadrilaterals
    - Generates a mesh of (n-1) times (m-1) quadrilaterals for n times m vertices
6.3 Solid modeling

- Extrusion
  - Useful representation in order to build object with rotational or translational symmetries and others
  - Objects are defined with a 2D primitive and a path in 3D space

![Diagram of Linear and Circular Extrusion]

- **Linear extrusion**
  - Path vertical

- **Circular extrusion**
  - Path along the circumference of the primitive
6.3 Solid modeling

- Constructive Solid Geometry (CSG)
  - A new solid is obtained by applying union, intersection and difference operations over two initial solids
  - There is an initial set of primitives (blocks, cones, cylinders, spheres, revolution surfaces, …)
  - Objects designed with this method are represented with a binary tree
6.3 Solid modeling

• Spatial numbering (Octrees)
  – Hierarchical tree structure, in which each node is a region of the 3D space
  – 3D space regions are divided into octants (cubes), 8 elements are stored in each node of the tree.
  – Node can be of 3 types: white (outside the object), black (inside the object) or grays (neither inside nor outside)
  – Individual elements of the 3D space are called *voxels*