Overview

- 3D solid model representations
  - Implicit models
  - Super/quadrics
  - Blobbies
  - Swept objects
  - Boundary representations
  - Spatial enumerations
  - Distance fields
  - Quadtrees/octrees
  - Stochastic models

Implicit Solid Modeling

- Computer Algebra meets CAD
- Idea:
  - Represents solid as the set of points where an implicit global function takes on certain value
    - \( F(x,y,z) < \text{val} \)
  - Primitive solids are combined using CSG
  - Composition operations are implemented by functionals which provide an implicit function for the resulting solid

Quadratic Surfaces

- Sphere
  \[ x^2 + y^2 + z^2 = r^2 \]
- Ellipsoid
- Torus
- General form
  \[ ax^2 + by^2 + cz^2 + 2fyz + 2gxz + 2hxy + 2p + x + 2q + y + 2r + z + d = 0 \]

Superellipsoid Surfaces

- Generalization of ellipsoid
- Control parameters \( s_1 \) and \( s_2 \)
  \[ \frac{x}{s_1}^{s_1} + \frac{y}{s_2}^{s_2} + \frac{z}{1}^{1} = 1 \]
- If \( s_1 = s_2 = 1 \) then regular ellipsoid
- Has an implicit and parametric form!

CSG with Superquadrics
CSG with Superellipsoids

Blobby Objects
- Do not maintain shape, topology
  - Water drops
  - Molecules
  - Force fields
- But can maintain other properties, like volume

Gaussian Bumps
- Model object as a sum of Gaussian bumps/blobs
  \[ f(x, y, z) = \sum b_k e^{-a_k r_k^2} \quad T = 0 \]
- Where \( r_k^2 = x_k^2 + y_k^2 + z_k^2 \) and \( T \) is a threshold.

Metaballs (Blinn Blobbies)

Ray-traced Metaballs

Implicit Modeling System U. of Calgary
- Combine “primitives”
  - Points, lines, planes, polygons, cylinders, ellipsoids
- Calculate field around primitives
- View iso-surface of implicit function
Implicit Modeling System
U. of Calgary
Can apply blends and warps

Sweep Representations
• An alternative way to represent a 3D object
• Idea
  – Given a primitive (e.g., polygon, sphere)
  – And a sweep (e.g., vector, curve…)
  – Define solid as space swept out by primitive

Sweep Representations
• Issues:
  – How to generate resulting surface?
  – What about self-intersections?
  – How to define intersection?

Approximate Representations
• Idea: discretize the world!
• Surface Models
  – Mesh, facet and polygon representations
• Volume Models
  – spatial enumeration
  – voxelization

Examples
• From exact to facets….

Boundary Representation
Solid Modeling
• The de facto standard for CAD since ~1987
  – BReps integrated into CAGD surfaces + analytic surfaces + boolean modeling
• Models are defined by their boundaries
• Topological and geometric integrity constraints are enforced for the boundaries
  – Faces meet at shared edges, vertices are shared, etc.
Let’s Start Simple: 
Polyhedral Solid Modeling

- Definition
  - Solid bounded by polygons whose edges are each a member of an even number of polygons
  - A 2-manifold: edges members of 2 polygons

Properties of 2-Manifolds

- For any point on the boundary, its neighborhood is a topological 2D disk
- If not a 2-manifold, neighborhood not a disk

Euler’s Formula

- For simple polyhedra (no holes):
  \#Vertices - \#Edges + \#Faces = 2

Euler’s Formula (Generalized)

\#Vertices - \#Edges + \#Faces - \#Holes_in_faces = 2 (\#Components - Genus)
- Genus is the # holes through the object
- Euler Operators have been the basis of several modeling systems (Mantyla et al.)

Euler Operators

<table>
<thead>
<tr>
<th>Operator Name</th>
<th>Meaning</th>
<th>(V)</th>
<th>(E)</th>
<th>(F)</th>
<th>(L)</th>
<th>(S)</th>
<th>(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEV</td>
<td>Make a edge and a vertex</td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFE</td>
<td>Make a face and an edge</td>
<td></td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSFY</td>
<td>Make a shell, a face and a vertex</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSG</td>
<td>Make a shell and a hole</td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEKL</td>
<td>Make an edge and kill a loop</td>
<td>+1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Steps to Creating a Polyhedral Solid Modeler

- Representation
  - Points, Lines/Edges, Polygons
- Modeling
  - Generalization of 3D clipping to non-convex polyhedra, enables implementation of booleans
State of the Art: BRep Solid Modeling

- ... but much more than polyhedra
- Two main (commercial) alternatives
  - All NURBS, all the time
    - Pro/E, SDRC, ...
  - Analytic surfaces + parametric surfaces + NURBS + ... all stitched together at edges
    - Parasolid, ACIS, ...

Issues in Boundary Representation Solid Modeling

- Very complex data structures
  - NURBS-based winged-edges, etc
- Complex algorithms
  - Manipulation, booleans, collision detection
- Robustness
- Integrity
- Translation
- Features
- Constraints and Parametrics

Other Issues: Non-Manifold Solids

- There are cases where you may need to model entities that are not entirely 3D

Cell Decomposition

- Set of primitive cells
- Parameterized
- Often curved
- Compose complex objects by gluing cells together
- Used in finite-element analysis

Spatial Occupancy Enumeration

- Brute force
  - A grid
- Pixels
  - Picture elements
- Volumes
  - Volume elements
- Quadtrees
  - 2D adaptive representation
- Octrees
  - 3D adaptive representation
  - Extension of quadtrees

Brute Force Spatial Occupancy Enumeration

- Impose a 2D/3D grid
  - Like graph paper or sugar cubes
- Identify occupied cells
- Problems
  - High fidelity requires many cells
- “Modified”
  - Partial occupancy
Distance Volume
- Store signed distance to surface at each voxel

Offset Surfaces from Distance Volumes

Quadtree
- Hierarchically represent spatial occupancy
- Tree with four regions
  - NE, NW, SE, SW
  - "dark" if occupied

Octree
- 8 octants 3D space
  - Left, Right, Up, Down, Front, Back

Boolean Operations on Octrees
Adaptive Distance Fields

- Quadtrees/Octrees that store distances

Applications for Spatial Occupancy Enumeration

- Many different applications
  - GIS
  - Medical
  - Engineering Simulation
  - Volume Rendering
  - Video Gaming
  - Approximating real-world data

Issues with Spatial Occupancy Enumeration

- Approximate
  - Kind of like faceting a surface, discretizing 3D space
  - Operationally, the combinatorics (as opposed to the numerics) can be challenging
  - Not as good for applications wanting exact computation (e.g. tool path programming)

Binary Space Partition Trees (BSP Trees)

- Recursively divide space into subspaces
- Arbitrary orientation and position of planes
- Homogeneous regions are leaves called in/out cells

Statistical Representations

- Store density (material vs. void)
- Statistical description of geometry
- Goal – describe the porosity without storing the geometry information

Stochastic Geometry

- Need some way of converting a solid into some representative statistical form
- From each material voxel, calculate the distance to the nearest voxel that is not material
- Repeat for void voxels
- Store distributions:
  - one for empty space
  - one for material
  - density value

Distance vs. Probability

Generated by Termite Agents Simulation.
Application: Biological Models

- Bone tissue
- MRI data
- Other biological data
- Solid modeling

MRI scan of left shoulder

Bone matrix from scanned data

Application: Surface Texture

Programming Assignment 5

- Extend XPM to 60 different RGB colors
- Read 3 models and assign each a color
- Implement Z-buffer rendering
- Implement front & back cutting planes
  - Only render parts of models between planes
- Implement linear depth-cueing
  - Color = base_color*[z-far]/(near-far)
- Re-use and extend 2D polygon filling

End