Overview

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

Implicit Solid Modeling

• Idea:
  – Represents solid as the set of points where an implicit global function takes on certain values
    • Usually
      • $F(x,y,z) < 0$, points inside of object
      • $F(x,y,z) = 0$, points on object’s surface
      • $F(x,y,z) > 0$, points outside of object
  – 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
  $$\left(\frac{x}{a}\right)^{2s_1} + \left(\frac{y}{b}\right)^{2s_2} + \left(\frac{z}{c}\right)^{2s_2} = 1$$

• Torus
  $$\left(\sqrt{x^2 + y^2} - \frac{d}{2}\right)^{2s_2} + \left(\frac{z}{e}\right)^{2s_3} = 1$$

• General form
  $$a_1\cdot x^2 + b_1\cdot y^2 + c_1\cdot z^2 + 2f_1\cdot yz + 2g_1\cdot xz + 2h_1\cdot xy + 2p_1\cdot x + 2q_1\cdot y + 2r_1\cdot z + d_1 = 0$$

Superellipsoid Surfaces

• Generalization of ellipsoid
• Control parameters $s_1$ and $s_2$
  $$\left(\frac{x}{a}\right)^{2s_1} + \left(\frac{y}{b}\right)^{2s_2} + \left(\frac{z}{c}\right)^{2s_2} = 1$$
  • If $s_1 = s_2 = 1$ then regular ellipsoid
  • Has an implicit and a parametric form!

The general superellipsoid has a parametric representation in terms of surface parameters $-\infty < u, v < \infty$:

$$x(u,v) = Ac(v,s_1)c(u,s_2)$$
$$y(u,v) = Bc(v,s_1)c(u,s_2)$$
$$z(u,v) = Cs(v,s_1)$$

where the auxiliary functions are:

$$c(\omega, m) = \text{sgn}(\omega) \cos(m\omega)$$
$$s(\omega, m) = \text{sgn}(\omega) \sin(m\omega)$$

and the sign function $\text{sgn}(x)$ is:

$$\text{sgn}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ +1, & x > 0 \end{cases}$$
Superellipsoid Surfaces

- Normals defined by
  \[ n_x(u,v) = \frac{1}{A} c(v,2-s_1) c(u,2-s_2) \]
  \[ n_y(u,v) = \frac{1}{B} c(v,2-s_1) s(u,2-s_2) \]
  \[ n_z(u,v) = \frac{1}{C} s(v,2-s_1) \]
- A, B and C are scale factors of the X, Y & Z coordinates
- \( s_1 \) is the shape parameter for longitude lines
- \( s_2 \) is the shape parameter for latitude lines

CSG with Superquadrics

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
  \[ f(x,y,z) = \sum \beta_i e^{-\rho r^2} - T = 0 \]
- Where \( r^2 = x^2 + y^2 + z^2 \) 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 polyhedron (no holes):
  #Vertices - #Edges + #Faces = 2
• If formula is true the surface is closed
Euler’s Formula (Generalized)

\[ \text{#Vertices} - \text{#Edges} + \text{#Faces} - \text{#Holes_in_faces} = 2 (\text{#Components} - \text{Genus}) \]

- 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 an 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></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSFV</td>
<td>Make a shell, a face and a vertex</td>
<td></td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSG</td>
<td>Make a shell and a hole</td>
<td></td>
<td></td>
<td>+1</td>
<td>+1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEKL</td>
<td>Make an edge and kill a loop</td>
<td></td>
<td></td>
<td></td>
<td>+1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

\( \text{Loop } L \rightarrow H, \quad \text{Shell } S \rightarrow C \)

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
- Voxels
  - 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
Quadtree Data Structure

Octree

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

Boolean Operations on Octrees

S U T
S ∩ T

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

Foley/VanDam, 1990/1994

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

Statistical Representations

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

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

Application: Surface Texture
Programming Assignment 3

- Process command line arguments
- Read in control points from file
- Double loop through u & v parameters
- For each (u, v) pair compute 3D point on Bezier patch
- Once you've computed the 3D points, define the triangles that connect them
- If shading, compute exact normals at each mesh vertex
- Output all data as Open Inventor

End