3D Object Representations

- Raw data
  - Point cloud
  - Range image
  - Polygon soup

- Surfaces
  - Mesh
  - Subdivision
  - Parametric
  - Implicit

- Solids
  - Voxel
  - BSP tree
  - CSG
  - Sweep

- High-level structures
  - Scene graph
  - Skeleton
  - Application specific

Implicit Surfaces

- Points satisfying: \( F(x,y,z) = 0 \)

**Implicit Surfaces**

- Example: quadric
  \[ f(x,y,z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fyz + 2gx + 2hy + 2jz + k \]

- Common quadric surfaces:
  - Sphere
  - Ellipsoid
  - Torus
  - Paraboloid
  - Hyperboloid

**Implicit Surfaces**

- Advantages:
  - Very concise
  - Guaranteed validity
  - Easy to test if point is on surface
  - Easy to intersect two surfaces

- Disadvantages:
  - Hard to describe complex shapes
  - Hard to enumerate points on surface
  - Hard to draw
### 3D Object Representations

- **Raw data**
  - Point cloud
  - Range image
  - Polygon soup
- **Surfaces**
  - Mesh
  - Subdivision
  - Parametric
  - Implicit
- **Solids**
  - Voxels
  - BSP tree
  - CSG
  - Sweep
- **High-level structures**
  - Scene graph
  - Skeleton
  - Application specific

### Solid Modeling

- Represent solid interiors of objects
  - Surface may not be described explicitly

### Motivation 1

- Some acquisition methods generate solids
  - Example: CAT scan

### Motivation 2

- Some applications require solids
  - Example: CAD/CAM

### Motivation 3

- Some algorithms require solids
  - Example: ray tracing with refraction

### Solid Modeling Representations

- What makes a good solid representation?
  - Accurate
  - Concise
  - Affine invariant
  - Easy acquisition
  - Guaranteed validity
  - Efficient boolean operations
  - Efficient display
Solid Modeling Representations

- Voxels
- Quadtrees & Octrees
- Binary space partitions
- Constructive solid geometry

Voxels

- Partition space into uniform grid
  - Grid cells are called a voxels (like pixels)
- Store properties of solid object with each voxel
  - Occupancy
  - Color
  - Density
  - Temperature
  - etc.

Voxel Storage

- $O(n^3)$ storage for $nxnxn$ grid
  - 1 billion voxels for 1000x1000x1000

Voxel Boolean Operations

- Compare objects voxel by voxel
  - Trivial

Voxel Acquisition

- Scanning devices
  - MRI
  - CAT
- Simulation
  - FEM

SUNY Stoney Brook

Stanford University
**Voxel Display**

- **Isosurface rendering**
  - Render surfaces bounding volumetric regions of constant value (e.g., density)

- **Slicing**
  - Draw 2D image resulting from intersecting voxels with a plane

**Voxel Display**

- **Ray casting**
  - Integrate density along rays through pixels

**Voxels**

- **Advantages**
  - Simple, intuitive, unambiguous
  - Same complexity for all objects
  - Natural acquisition for some applications
  - Trivial boolean operations

- **Disadvantages**
  - Approximate
  - Not affine invariant
  - Large storage requirements
  - Expensive display

**Solid Modeling Representations**

- **Voxels**
- **Quadtrees & Octrees**
- Binary space partitions
- Constructive solid geometry

**Quadtrees & Octrees**

- Refine resolution of voxels hierarchically
  - More concise and efficient for non-uniform objects
Quadtree Boolean Operations

Quadtree Display

• Extend voxel methods
  » Slicing
  » Isosurface extraction
  » Ray casting

Finding neighbor cell requires traversal of hierarchy (O(1))

Solid Modeling Representations

• Voxels
• Quadtrees & Octrees
• Binary space partitions
• Constructive solid geometry

Binary Space Partitions (BSPs)

• Recursive partition of space by planes
  » Mark leaf cells as inside or outside object

BSP Fundamentals

Single geometric operation
Partition a convex region by a hyperplane
Single combinatorial operation
Two child nodes added as leaf nodes

BSP is a Search Structure

Exploit hierarchy of convex regions
Regions decrease in size along any tree path
Regions converge in the limit to the surface
**BSP Acquisition**

- Must construct a “good” binary search structure
  - Efficiency comes from logarithmic tree depth

- “Bad”
- “Good”

**BSP Boolean Operations**

- Divide and conquer
  - Each node V corresponds to a convex region containing all geometry in the subtree rooted at V
  - No intersection with bounding volume of V means no intersection with subtree rooted at V
  - Do detail work only in regions required
  - Boolean operations grow with $O(\log n)$ if “good” tree

**BSP Display**

- Visibility ordering
  - Determine on which side of plane the viewer lies
    - near-subtree -> polygons on split -> far-subtree

**Solid Modeling Representations**

- Voxels
- Quadtrees & Octrees
- Binary space partitions
- Constructive solid geometry

**Constructive Solid Geometry (CSG)**

- Represent solid object as hierarchy of boolean operations
  - Union
  - Intersection
  - Difference

**CSG Acquisition**

- Interactive modeling programs
  - CAD/CAM
CSG Boolean Operations

- Create a new CSG node joining subtrees
  - Union
  - Intersection
  - Difference

CSG Display & Analysis

- Ray casting
  - Union
  - Circle
  - Box

Summary

<table>
<thead>
<tr>
<th></th>
<th>Voxels</th>
<th>Octree</th>
<th>BSP</th>
<th>CSG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td>No</td>
<td>No</td>
<td>Some</td>
<td>Some</td>
</tr>
<tr>
<td>Concise</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Affine invariant</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Easy acquisition</td>
<td>Some</td>
<td>Some</td>
<td>No</td>
<td>Some</td>
</tr>
<tr>
<td>Guaranteed validity</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Efficient boolean operations</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Efficient display</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Taxonomy of 3D Representations

- Discrete
  - Voxels
- Continuous
- Combinatorial
- Functional
  - Topological
    - Set Membership
  - Parametric
    - Algebraic
  - Geometric
    - Mesh
    - Subdivision
    - BSP Tree
    - Cell Complex
    - Bezier
    - B-Spline