



# Solid Modeling

Adam Finkelstein  
Princeton University  
COS 426, Spring 2005



# 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



# 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

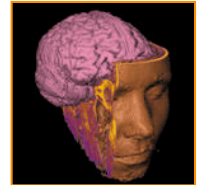


# Solid Modeling

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



Visible Human  
(National Library of Medicine)



SUNY Stony Brook



# Motivation 1

- Some acquisition methods generate solids
  - Example: CAT scan

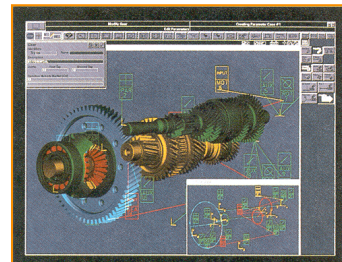


Stanford University



# Motivation 2

- Some applications require solids
  - Example: CAD/CAM

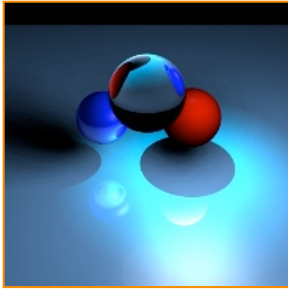


Intergraph Corporation

## Motivation 3



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

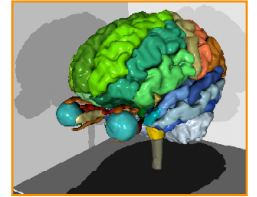


Addy Ngan and Zaijin Guan  
COS 426, 1998  
Princeton University

## Solid Modeling Representations



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



Lorensen

## Solid Modeling Representations



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

## Solid Modeling Representations

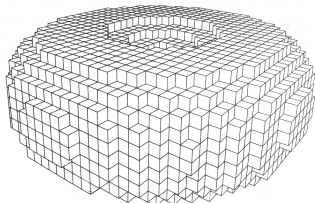


- **Voxels**
- Quadtrees & Octrees
- Binary space partitions
- Constructive solid geometry

## 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.



FvDFH Figure 12.20

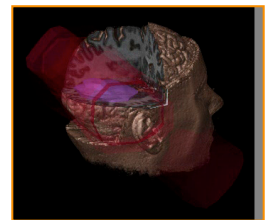
## Voxel Acquisition



- Scanning devices
  - MRI
  - CAT
- Simulation
  - FEM



SUNY Stony Brook

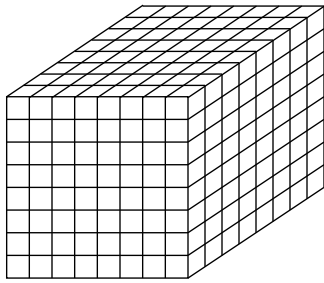


Stanford University

## Voxel Storage



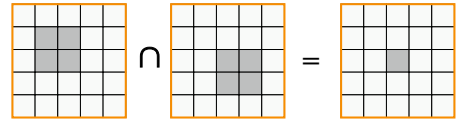
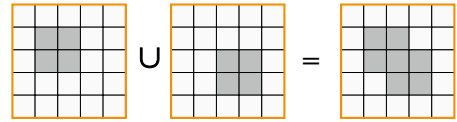
- $O(n^3)$  storage for  $n \times n \times n$  grid
  - 1 billion voxels for  $1000 \times 1000 \times 1000$



## Voxel Boolean Operations



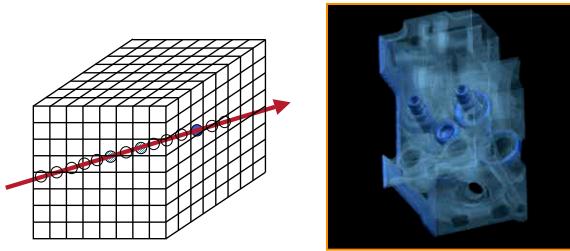
- Compare objects voxel by voxel
  - Trivial



## Voxel Display



- Ray casting
  - Integrate density along rays through pixels

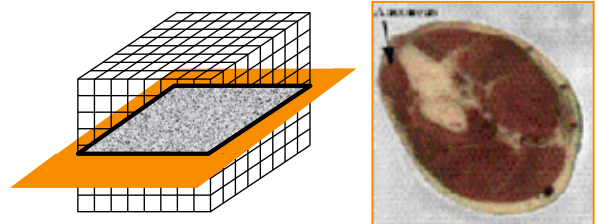


Engine Block  
Stanford University

## Voxel Display



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

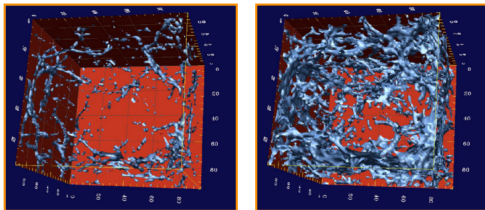


Visible Human  
(National Library of Medicine)

## Voxel Display

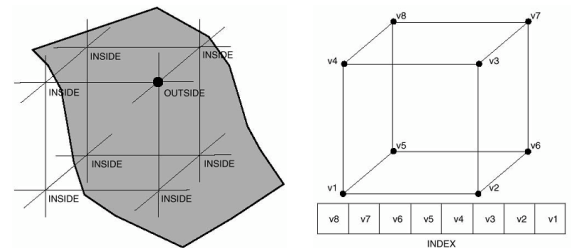


- Isosurface rendering
  - Render surfaces bounding volumetric regions of constant value (e.g., density)

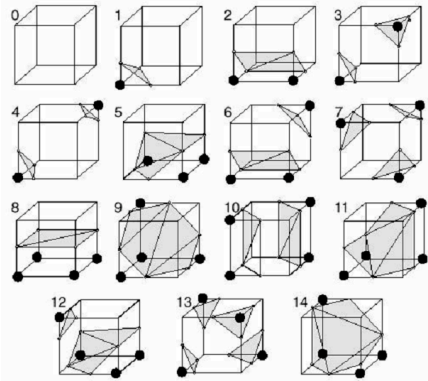


Isosurface Visualization  
Princeton University

## Marching Cubes



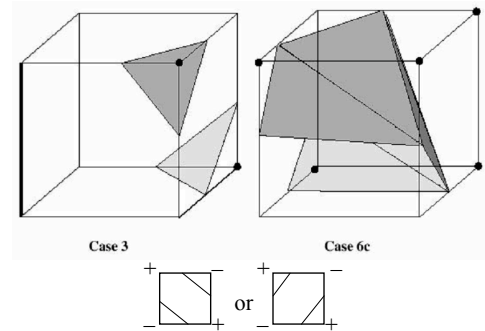
## Marching Cubes (15 Cases)



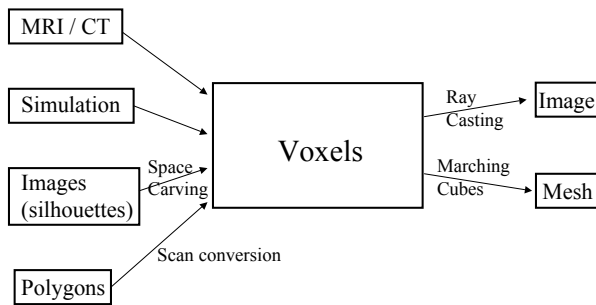
## Marching Cubes



- Sometimes have to match neighbors



## Voxels



## 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

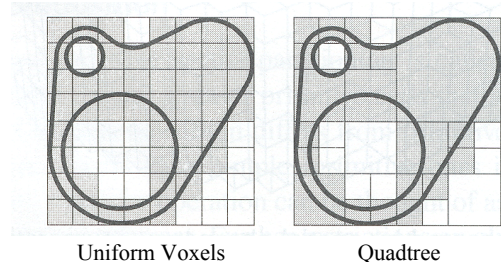


- Voxels
- **Quadtrees & Octrees**
- Binary space partitions
- Constructive solid geometry

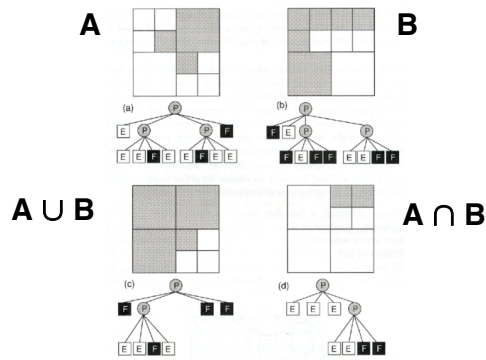
## Quadtrees & Octrees



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



## Quadtree Boolean Operations

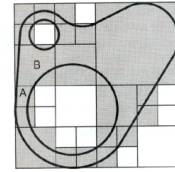


FvDFH Figure 12.24

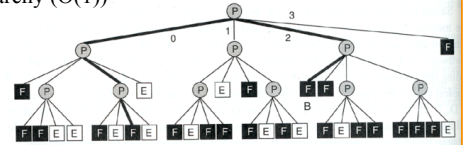
## Quadtree Display



- Extend voxel methods
  - Slicing
  - Isosurface extraction
  - Ray casting



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



FvDFH Figure 12.25

## Solid Modeling Representations

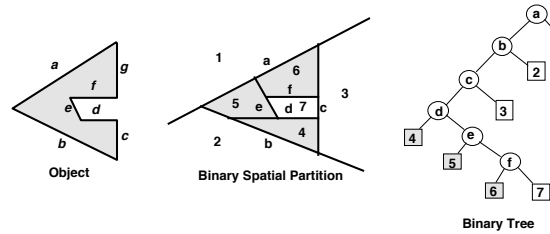


- Voxels
- Quadtrees & Octrees
- Binary space partitions
- Constructive solid geometry

## Binary Space Partitions (BSPs)



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

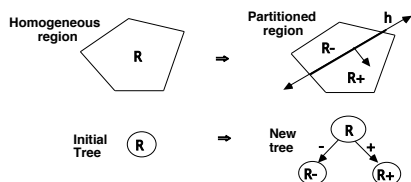


Naylor

## BSP Fundamentals



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

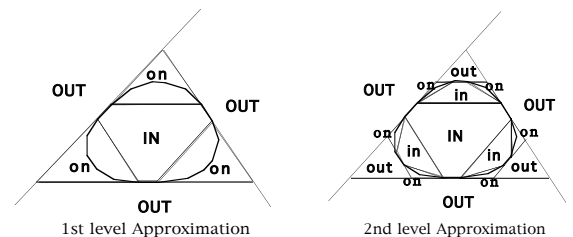


Naylor

## 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

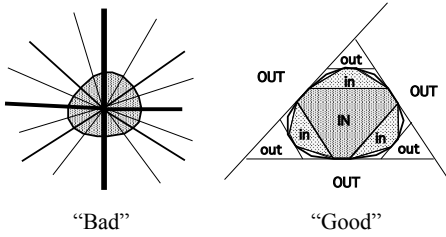


Naylor

## BSP Acquisition



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

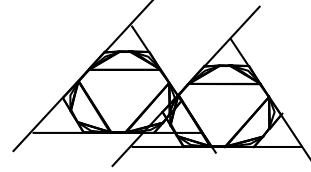


Naylor

## 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

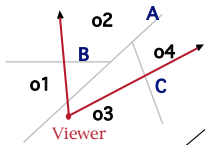


Naylor

## BSP Display



- Hidden surface removal
  - Use painter's algorithm: draw faces back-to-front
    - » far-subtree > faces on split > near-subtree



```
void Draw_BSP_Tree (BSP_tree *tree, point eye)
{
    real result = tree->partition.Classify(eye);
    if (result > 0)
    {
        Draw_BSP_Tree (tree->back, eye);
        tree->polygons.Draw_Polygon_List ();
        Draw_BSP_Tree (tree->front, eye);
    }
    else if (result < 0)
    {
        Draw_BSP_Tree (tree->front, eye);
        tree->polygons.Draw_Polygon_List ();
        Draw_BSP_Tree (tree->back, eye);
    }
    else // result is 0
    {
        // eye point is on the partition plane
        Draw_BSP_Tree (tree->front, eye);
        Draw_BSP_Tree (tree->back, eye);
    }
}
```

Naylor

Viewer

SGI BSP Tree FAQ

## Solid Modeling Representations

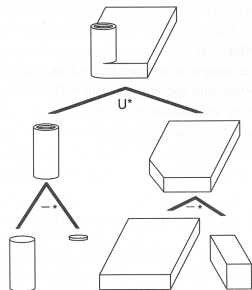


- Voxels
- Quadtrees & Octrees
- Binary space partitions
- **Constructive solid geometry**

## Constructive Solid Geometry (CSG)



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

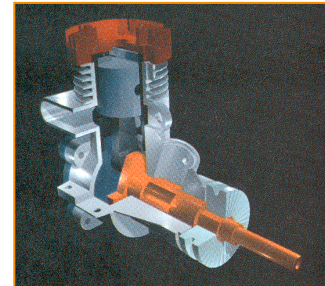


FvDFH Figure 12.27

## CSG Acquisition



- Interactive modeling programs
  - CAD/CAM

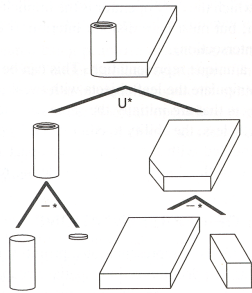


H&B Figure 9.9

## CSG Boolean Operations



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

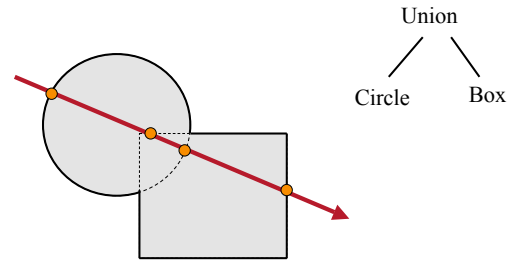


FvDFH Figure 12.27

## CSG Display & Analysis



- Ray casting



## Summary



	Voxels	Octree	BSP	CSG
Accurate	No	No	Some	Some
Concise	No	No	No	Yes
Affine invariant	No	No	Yes	Yes
Easy acquisition	Some	Some	No	Some
Guaranteed validity	Yes	Yes	Yes	No
Efficient boolean operations	Yes	Yes	Yes	Yes
Efficient display	No	No	Yes	No

## Taxonomy of 3D Representations

