Implicit Surfaces & Solid Representations

COS 426, Spring 2019
Adam Finkelstein
Princeton University
3D Object Representations

• Raw data
  • Range image
  • Point cloud

• Surfaces
  • Polygonal mesh
  • Subdivision
  • Parametric
    ➢ Implicit

• Solids
  • Voxels
  • BSP tree
  • CSG
  • Sweep

• High-level structures
  • Scene graph
  • Application specific
3D Object Representations

• Desirable properties of an object representation
  • Easy to acquire
  • Accurate
  • Concise
  • Intuitive editing
  • Efficient editing
  • Efficient display
  • Efficient intersections
  • Guaranteed validity
  • Guaranteed smoothness
  • etc.

Large Geometric Model Repository
Georgia Tech
3D Object Representations

• Desirable properties of an object representation
  • Easy to acquire
  • Accurate
  • Concise
  • Intuitive editing
  • Efficient editing
  • Efficient display
  • Efficient intersections
  • Guaranteed validity
  • Guaranteed smoothness
  • etc.

Large Geometric Model Repository
Georgia Tech
Implicit Surfaces

- Represent surface with function over all space
Implicit Surfaces

• Surface defined implicitly by function
Implicit Surfaces

- Surface defined implicitly by function:
  - $f(x, y, z) = 0$ (on surface)
  - $f(x, y, z) < 0$ (inside)
  - $f(x, y, z) > 0$ (outside)
Implicit Surfaces

• Normals defined by partial derivatives
  • Normal - $N(x, y, z) = \text{normalize} \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) = \text{normalize}(\vec{\nabla} f)$

  • Example: circle $x^2 + y^2 - 3^2 = 0$
  • Proof: straightforward with an arbitrary curve $\Gamma(t)$ and the chain rule
  • Max change rate direction of $f$ perpendicular to iso-surface direction
  • Intuition in 2D: skiing downhill on a topo-map
Implicit Surfaces

• Normals defined by partial derivatives
  • Normal \( N(x, y, z) = \text{normalize} \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) = \text{normalize}(\nabla f) \)
  • Tangent \( T = N_P \times N \)
    • on specific plane \( P \), with normal \( N_P \)
    • Otherwise infinite directions

Normals
Tangents
Implicit Surfaces

• Normals defined by partial derivatives
  
  Normal - \( N(x, y, z) = \text{normalize} \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) = \text{normalize}(\mathbf{\nabla} f) \)

  • Tangent – \( T = N_P \times N \)

  • Curvature – change of rate \( N \)
  
    • Computation more involved
  
    • Principal directions – min and max curvature

Normals

Tangents

Curvatures

Bloomenthal
Implicit Surface Properties

(1) Efficient check for whether point is inside

- Evaluate $f(x,y,z)$ to see if point is inside/outside/on
- Example: ellipsoid

\[
f(x, y, z) = \left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 - 1
\]

H&B Figure 10.10
(2) Efficient surface intersections

• Substitute to find intersections

Ray: $P = P_0 + tV$
Sphere: $|P - O|^2 - r^2 = 0$

Substituting for $P$, we get:

$|P_0 + tV - O|^2 - r^2 = 0$

Solve quadratic equation:

$at^2 + bt + c = 0$

where:

$a = 1$
$b = 2 \ V \cdot (P_0 - O)$
$c = |P_0 - O|^2 - r^2 = 0$
Display Signed Distance Field Slices
Hierarchical $hp$-Adaptive Signed Distance Fields

Dan Koschier, Crispin Deul and Jan Bender
Implicit Surface Properties

(3) Efficient boolean operations (CSG – later in this lecture)

• How would you implement:
  Union? Intersection? Difference?

Union

Difference

Bloomenthal
Implicit Surface Properties

(4) Efficient topology changes
  • Surface is not represented explicitly!
(4) Efficient topology changes

- Surface is not represented explicitly!
Example: Modeling

[olivelarouille on Youtube]
Implicit Surface Properties

(5) Computations in the volume
- Allows for continuity and smoothness
- Suitable for tasks such as reconstruction

Poisson Surface Reconstruction [Kazhdan 06]
Example: Surface reconstruction

Online Reconstruction of 3D Objects from Arbitrary Cross-Sections
[Bermano et al. 2011]
Comparison to Parametric Surfaces

• Implicit
  • Efficient intersections & topology changes

• Parametric
  • Efficient “marching” along surface & rendering
Implicit Surface Representations

• How do we define implicit function?
  • $f(x,y,z) = ?$
Implicit Surface Representations

• How do we define implicit function?
  • Algebraics
  • Voxels
  • Basis functions
  • Others
Implicit Surface Representations

• How do we define implicit function?
  ➢ Algebraics
    • Voxels
    • Basis functions
    • Others
Algebraic Surfaces

• Implicit function is polynomial
  • \( f(x,y,z) = ax^d + by^d + cz^d + dx^{d-1}y + dx^{d-1}z + dy^{d-1}x + \ldots \)

\[
f(x, y, z) = \left( \frac{x}{r_x} \right)^2 + \left( \frac{y}{r_y} \right)^2 + \left( \frac{z}{r_z} \right)^2 - 1
\]

H&B Figure 10.10
Algebraic Surfaces

• Most common form: quadrics
  - \( f(x,y,z)=ax^2+by^2+cz^2+2dxy+2eyz+2fxz+2gx+2hy+2jz+k \)

• Examples
  - Sphere
  - Ellipsoid
  - Paraboloid
  - Hyperboloid

http://tutorial.math.lamar.edu/Classes/CalcIII/QuadricSurfaces.aspx
Algebraic Surfaces

• Higher degree algebraics

Cubic  Quartic  Degree six
Algebraic Surfaces

• Equivalent parametric surface
  • Tensor product patch of degree m and n curves yields algebraic function with degree 2mn

Bicubic patch has degree 18!
Algebraic Surfaces

• Intersection
  • Intersection of degree m and n algebraic surfaces yields curve with degree mn

Intersection of bicubic patches has degree 324!
Algebraic Surfaces

- Function extends to infinity
  - Must trim to get desired patch (this is difficult!)
Implicit Surface Representations

• How do we define implicit function?
  • Algebraics
  ➢ Voxels
  • Basis functions
Voxels

• Regular array of 3D samples (like image)
  • Samples are called *voxels* ("volume pixels")
Voxels

• Example isosurfaces

SUNY Stoney Brook

Princeton University
Voxels

• Regular array of 3D samples (like image)
  • Applying reconstruction filter (e.g. trilinear) yields $f(x,y,z)$
  • Isosurface at $f(x,y,z) = 0$ defines surface

2.3  1.7  0.9  0.2
1.2  0.4  0.1  -0.8
0.3  -0.5 -0.7  -1.4
0.2  -0.9 -1.7  -2.5
Voxels

• Iso-surface extraction algorithm
  • e.g., Marching cubes
Voxels

• Iso-surface extraction algorithm
  • e.g., Marching cubes (15 cases)
Example: Marching Cubes

Voxels: 512 x 512 x 184 = 48,234,496
Vertices: 66,718
Triangles: 398,382
Threshold: 943
FPS: 122.179
Voxel Storage

• $O(n^3)$ storage for $n \times n \times n$ grid
  • 1 billion voxels for 1000 x 1000 x 1000
Implicit Surface Representations

• How do we define implicit function?
  • Algebraics
  • Voxels
  ➢ Basis functions
Basis functions

- Implicit function is sum of basis functions
  - Example:

  \[
  f(P) = a_0 e^{-b_0 d(P, P_0)^2} + a_1 e^{-b_1 d(P, P_1)^2} + \cdots - \tau
  \]
Blobby Models

• Implicit function is sum of Gaussians

\[ f(P) = a_0 e^{-b_0 d(P,P_0)^2} + a_1 e^{-b_1 d(P,P_1)^2} + \cdots - \tau \]
Blobby Models

• Sum of two blobs
Blobby Models

• Sum of four blobs
Blobby Model of Head

(a) $N = 1$

(b) $N = 2$
Blobby Model of Head

(c) \( N = 20 \)

(d) \( N = 60 \)
Blobby Model of Face

(a) $N = 1$

(b) $N = 2$
Blobby Model of Face

(c) $N = 10$

(d) $N = 35$
Blobby Model of Face

(e) $N = 70$

(f) $N = 243$
Reconstruction from Point Sets

Input

Implicit
Reconstruction from Point Sets
Reconstruction from Point Sets

- Implicit function is sum of basis functions

\[ \text{dist}(\mathbf{x}) = \sum_i w_i \varphi_i(\mathbf{x}) = \sum_i w_i \varphi(\| \mathbf{x} - \mathbf{c}_i \|) \]
Reconstruction from Point Sets
Implicit Surface Summary

• Advantages:
  • Easy to test if point is on surface
  • Easy to compute intersections/unions/differences
  • Easy to handle topological changes

• Disadvantages:
  • Indirect specification of surface
  • Hard to describe sharp features
  • Hard to enumerate points on surface
    • Slow rendering
## Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>Polygonal Mesh</th>
<th>Implicit Surface</th>
<th>Parametric Surface</th>
<th>Subdivision Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Concise</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Intuitive specification</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Local support</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Affine invariant</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Arbitrary topology</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Guaranteed continuity</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Natural parameterization</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Efficient display</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Efficient intersections</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
3D Object Representations

• Raw data
  • Range image
  • Point cloud

• Surfaces
  • Polygonal mesh
  • Subdivision
  • Parametric
  • Implicit

• Solids
  • Voxels
  • BSP tree
  • CSG
  • Sweep

• High-level structures
  • Scene graph
  • Application specific
Solid Modeling

- Represent solid interiors of objects
Motivation 1

• Some acquisition methods generate solids

Airflow Inside a Thunderstorm
(Bob Wilhelmson,
University of Illinois at Urbana-Champaign)

Visible Human
(National Library of Medicine)
Motivation 2

• Some applications require solids
  • Examples: medicine, CAD/CAM

SUNY Stoney Brook

Intergraph Corporation
Motivation 3

• Some operations are easier with solids
  • Example: union, difference, intersection

Union

Difference

Bloomenthal
3D Object Representations

• Raw data
  • Range image
  • Point cloud

• Surfaces
  • Polygonal mesh
  • Subdivision
  • Parametric
  • Implicit

• Solids
  ➢ Voxels
    • BSP tree
    • CSG
    • Sweep

• High-level structures
  • Scene graph
  • Application specific
Return to Voxels

• Regular array of 3D samples (like image)
Voxels

• Store properties of solid object with each voxel
  • Occupancy
  • Color
  • Density
  • Temperature
  • etc.

Engine Block
Stanford University

Visible Human
(National Library of Medicine)
Voxel Processing

• Signal processing (just like images)
  • Reconstruction
  • Resampling

• Typical operations
  • Blur
  • Edge detect
  • Warp
  • etc.

• Often fully analogous to image processing
Voxel Boolean Operations

• Compare objects voxel by voxel
  • Trivial

\[ U \cap \cap = \]

\[ U \cap \cap = \]

\[ U \cap \cap = \]
Voxel Display

• Isosurface rendering
  • Interpolate samples stored on regular grid
  • Isosurface at $f(x,y,z) = 0$ defines surface
Voxel Display

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

Visible Human
(National Library of Medicine)
Voxel Display

• Ray casting
  • Integrate density along rays: compositing!

Engine Block
Stanford University
Voxel Display

- Extended ray-casting
  - Transfer functions: Map voxel values to opacity and material
  - Normals (for lighting) from density gradient

Bruckner et al. 2007
Voxels

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

• Disadvantages
  • Approximate
  • Not affine invariant
  • Expensive display
  • Large storage requirements
Voxels

- What resolution should be used?

FvDFH Figure 12.21
Quadtrees & Octrees

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

Uniform Voxels  Quadtree (Octree in 3D)

FvDFH Figure 12.21
Quadtree Processing

• Hierarchical versions of voxel methods
  • Finding neighbor cell requires traversal of hierarchy: expected/amortized $O(1)$
Quadtree Boolean Operations

A

B

A \cup B

A \cap B

FvDFH Figure 12.24
3D Object Representations

- **Raw data**
  - Range image
  - Point cloud

- **Surfaces**
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit

- **Solids**
  - Voxels
  - **BSP tree**
  - CSG
  - Sweep

- **High-level structures**
  - Scene graph
  - Application specific
BSP Trees

Object

Binary Spatial Partition

Binary Tree

Naylor
BSP Trees

• Key properties
  • visibility ordering (later)
  • hierarchy of convex regions (useful for collision)
3D Object Representations

- Raw data
  - Range image
  - Point cloud

- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit

- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep

- High-level structures
  - Scene graph
  - Application specific
Constructive Solid Geometry (CSG)

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

FvDFH Figure 12.27
CSG Acquisition

• Interactive modeling programs
  • Intuitive way to design objects
CSG Acquisition

- Interactive modeling programs
  - Intuitive way to design objects

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
3D Object Representations

- **Raw data**
  - Range image
  - Point cloud

- **Surfaces**
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit

- **Solids**
  - Voxels
  - BSP tree
  - CSG
  - Sweep

- **High-level structures**
  - Scene graph
  - Application specific
Sweeps

• Swept volume
  • Sweep one curve along path of another curve

Demetri Terzopoulos
Sweeps

- Surface of revolution
  - Take a curve and rotate it about an axis
Sweeps

• Surface of revolution
  • Take a curve and rotate it about an axis
Modeling a swept curve
<table>
<thead>
<tr>
<th>Feature</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 ops</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>