Solid Modeling

Michael Kazhdan

(600.457)

HB 10.15 – 10.17, 10.22
FvDFH 12.1, 12.2, 12.6, 12.7

Marching Cubes, Lorensen et al. 1987
Solid Modeling

So far, we have focused on representing models with (triangular) meshes that approximate the surface/boundary of the model.

Advantages:

• Easy to visualize in graphics hardware

Limitations:

• Some models cannot be represented by a boundary
• It can be difficult to intersect two models
Solid Modeling

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

Visible Human
(National Library of Medicine)
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

Addy Ngan and Zaijin Guan
Overview

- Implicit Surfaces
- Voxels
- Quadtrees and Octrees
Implicit Surfaces

Given a real-valued function in 3D, \( F(x, y, z) \), the implicit surface defined by \( F \) is the collection of points for which \( F(x, y, z) = 0 \).

• Example: quadric
  - \( F(x, y, z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k \)
Implicit Surfaces

Given a real-valued function in 3D, \( F(x, y, z) \), the implicit surface defined by \( F \) is the collection of points for which \( F(x, y, z) = 0 \).

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

\[
\left( \frac{x}{r_x} \right)^2 + \left( \frac{y}{r_y} \right)^2 + \left( \frac{z}{r_z} \right)^2 - 1 = 0
\]

Ellipsoids

Image courtesy of http://www.geom.uiuc.edu/
Implicit Surfaces

Given a real-valued function in 3D, \( F(x, y, z) \), the implicit surface defined by \( F \) is the collection of points for which \( F(x, y, z) = 0 \).

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

\[
\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 - \left(\frac{z}{r_z}\right)^2 \pm 1 = 0
\]

Hyperboloids

Image courtesy of http://www.geom.uiuc.edu/
Implicit Surfaces

Given a real-valued function in 3D, $F(x, y, z)$, the implicit surface defined by $F$ is the collection of points for which $F(x, y, z) = 0$.

• Example: quadric
  
  $F(x, y, z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k$

\[
\left(\frac{x}{r_x}\right)^2 \pm \left(\frac{y}{r_y}\right)^2 + 2z = 0
\]

Paraboloids

Image courtesy of http://www.geom.uiuc.edu/
Implicit Surfaces

Blobby Models

Express the implicit surface as a sum of Gaussians:

\[ F(x, y, z) = \sum_i F_i(x, y, z) \]

\[ F_i(x, y, z) = \alpha_i e^{-\left( (x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 \right)/2\sigma_i^2} \]

- \((x_i, y_i, z_i)\) is the center of the Gaussian
- \(\alpha_i\) controls the contribution of the Gaussian
  - Smaller values mean that the Gaussian contributes less to the overall implicit function
- \(\sigma_i\) controls the width of the Gaussian
  - Larger values mean that the Gaussian falls off more sharply and the effect is more local
Implicit Surfaces

Blobby Models

The more functions we use, the more accurate the reconstruction.

But this could also makes the function more difficult to sample.

Muraki, 1991
Implicit Surfaces

\[ F(x, y, z) = \sum_i F_i(x, y, z) \]

If the functions \( F_i \) are compactly supported, evaluation can be done in sub-linear time.

Chen et al., SIGGRAPH 04
Implicit Surfaces

• Advantages:
  ◦ Easy to test if a point is on the surface
  ◦ Easy to test if a point is inside the surface
  ◦ Easy to intersect two surfaces

• Disadvantages:
  ◦ Hard to describe complex shapes
  ◦ Hard to evaluate complex functions
  ◦ Hard to enumerate points on surface
Overview

• Implicit Surfaces
• Voxels
• Quadtrees and Octrees
Voxels

• Partition space into uniform grid
  ◦ Grid cells are called a voxels (like pixels)

• Each voxel has a value associated to it.
Voxels

• Partition space into uniform grid
  ◦ Grid cells are called a voxels (like pixels)

• Each voxel has a value associated to it.
  ◦ Binary Voxel Grids:
    » Value is 0 if the voxel is outside the model
    » Value is 1 if the voxel is inside
Binary Voxel Boolean Operations

• Compare objects voxel by voxel
  ⊕ Trivial
Binary Voxel Visualization

- Draw the faces between on and off voxels.
Voxels

- Partition space into uniform grid
  - Grid cells are called *voxels* (like pixels)

- Each voxel has a value associated to it.
  - Binary Voxel Grids:
  - Continuous Voxel Grids:
    » Each voxel stores a continuous value (e.g. density, temperature, color, etc.)
Continuous Voxel Visualization

- Slicing
- Ray-Casting
- Iso-Surface Extraction
Voxel Display

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

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

• Ray casting
  ○ Integrate density along rays through pixels

Engine Block
Stanford University
Voxel Display

• Iso-Surface Extraction
  ◦ Treat the voxel grid as a regular sampling of a function $F(x, y, z)$, and extract the iso-surface with $F(x, y, z) = \delta$. 

Iso-Value = $\delta_1$

Iso-Value = $\delta_2$
Marching Cubes Algorithm

- Iso-Surfaces analog with 2D grid
  - Assume each grid location has scalar value
  - If one of the vertices of an edge has value larger than $\delta$ and the other has value less than $\delta$, find the point on the edge whose linear interpolation is equal to $\delta$.
  - Connect the new edge points with line segments.

Note that the number of edges on which we insert vertices must be even.
Marching Cubes Algorithm

- Iso-Surfaces analog with 2D grid
  - Break up into the $2^4 = 16$ different possible cases
  - Assign a rule for surface reconstruction for each of the different cases.

Note that certain configurations are ambiguous.
Marching Cubes Algorithm

• Iso-Surfaces analog with 2D grid
  ◦ Break up into the $2^4 = 16$ different possible cases
  ◦ Assign a rule for surface reconstruction for each of the different cases.
  ◦ Combine the line segments generated from the different grid cells.

As long as the position of the iso-vertices is defined by values at the end-points, adjacent cells will define consistent (connected) segments.
Assigning iso-vertex position (linear):

If we have a function with \( f(0) = a \) and \( f(1) = b \), we can fit a linear interpolant:

\[
f(x) = a + (b - a)x
\]

Then the for the function to have value \( \delta \):

\[
f(x) = \delta
\]

\[
dx = \frac{\delta - a}{b - a}
\]
Marching Cubes Algorithm

Assigning iso-vertex position (cubic):

If we also know $f(-1)$ and $f(2)$ we can fit Cardinal B-spline to the four values and find the roots of the polynomial in the range $[0,1]$.

**Note**: Since we are using an interpolating spline, we are guaranteed to find an odd number of roots in the interval.
Marching Cubes Algorithm

• Iso-Surface with 3D grid
  ◦ Break up into the $2^8 = 256$ different possible cases
  ◦ Assign a rule for surface reconstruction for each of the different cases.
  ◦ Combine the surface segments generated from the different grid cells.
Voxels

Continuous voxel grids are essentially 3D images. Many of the operations that we applied to 2D images can also be applied to voxel grids:

- Sampling
- Contrast
- Edge detection
- Smoothing
Voxels

• Advantages
  ○ Simple
  ○ 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

- Implicit Surfaces
- Voxels
- Quadtrees & Octrees
Quadtrees (2D) & Octrees (3D)

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

FvDFH Figure 12.21

Uniform Voxels  Quadtree
Quadtrees (2D) and Octrees (3D)

- Octree = same idea but with “octants” or cubes
- Expected case: number of nodes $\approx$ to perimeter or surface area

FvDFH Figure 12.24
Quadtree Boolean Operations

A

B

A ∩ B

A ∪ B

FvDFH Figure 12.24
Octree Display

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

Even if corner values are set consistently, how do we define the values of the function along faces and edges shared by cells at different resolutions?
Octree Display

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

For isosurfacing, the two sides of the face might not even have the same number of zero-crossings.
Octree Display

• Extend voxel methods
  ○ Slicing
  ○ Ray casting
  ○ Isosurface extraction
Octree Display

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