

Visualization

Height Fields and Contours
Scalar Fields
Volume Rendering
Vector Fields
[Ch. 2.11]

April 27, 2011
Jernej Barbic

University of Southern California

<http://www.bcf.usc.edu/~jbarbic/cs480-s11/>

1

Scientific Visualization

- Generally do not start with a 3D triangle model
- Must deal with very large data sets
 - MRI, e.g. 512 x 512 x 200 = 50MB points
 - Visible Human 512 x 512 x 1734 = 433 MB points
- Visualize both real-world and simulation data
- User interaction
- Automatic search for relevant data

2

Types of Data

- Scalar fields (3D volume of scalars)
 - E.g., x-ray densities (MRI, CT scan)
- Vector fields (3D volume of vectors)
 - E.g., velocities in a wind tunnel
- Tensor fields (3D volume of tensors [matrices])
 - E.g., stresses in a mechanical part
- Static or dynamic through time

3

Height Field

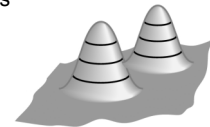
- Visualizing an explicit function

$$z = f(x,y)$$



- Adding contour curves

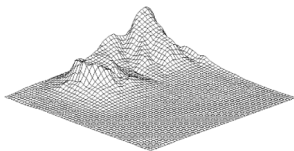
$$g(x,y) = c$$



4

Meshes

- Function is sampled (given) at $x_i, y_j, 0 \leq i, j \leq n$
- Assume equally spaced
$$x_i = x_0 + i\Delta x$$
$$y_j = y_0 + j\Delta y$$
$$z_{ij} = f(x_i, y_j)$$
- Generate quadrilateral or triangular mesh
- [Assignment 1]



5

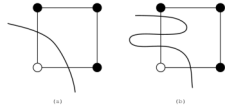
Contour Curves

- Recall: implicit curve $f(x,y) = 0$
- $f(x,y) < 0$ inside, $f(x,y) > 0$ outside
- Here: contour curve at $f(x,y) = c$
- Implicit function f sampled at regular intervals for x,y
$$x_i = x_0 + i\Delta x$$
$$y_j = y_0 + j\Delta y$$
- How can we draw the curve?

6

Marching Squares

- Sample function f at every grid point x_i, y_j
- For every point $f_{ij} = f(x_i, y_j)$ either $f_{ij} \leq c$ or $f_{ij} > c$
- Distinguish those cases for each corner x
 - White: $f_{ij} \leq c$
 - Black: $f_{ij} > c$
- Now consider cases for curve
- Assume "smooth"
- Ignore $f_{ij} = 0$



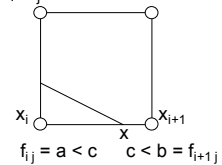
7

Interpolating Intersections

- Approximate intersection
 - Midpoint between x_i, x_{i+1} and y_j, y_{j+1}
 - Better: interpolate
- If $f_{ij} = a$ is closer to c than $b = f_{i+1,j}$ then intersection is closer to (x_i, y_j) :

$$\frac{x - x_i}{x_{i+1} - x} = \frac{c - a}{b - c}$$

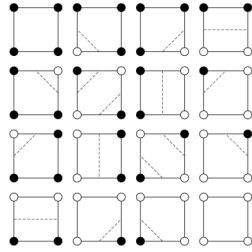
- Analogous calculation for y direction



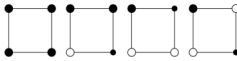
8

Cases for Vertex Labels

16 cases for vertex labels

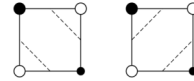


4 unique cases modulo symmetries



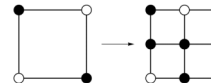
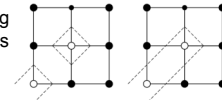
9

Ambiguities of Labelings



Ambiguous labels

Different resulting contours



Resolution by subdivision (if such higher resolution available / possible)

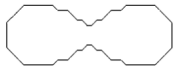
10

Marching Squares Examples

- Ovals of Cassini, 50 x 50 grid

$$f(x,y) = (x^2 + y^2 + a^2)^2 - 4a^2x^2 - b^4$$

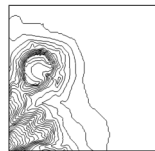
$$a = 0.49, b = 0.5$$



Midpoint



Interpolation



Contour plot of Honolulu data

11

Outline

- Height Fields and Contours
- Scalar Fields
- Volume Rendering
- Vector Fields

12

Scalar Fields

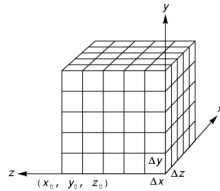
- Volumetric data sets
- Example: tissue density
- Assume again regularly sampled

$$x_i = x_0 + i\Delta x$$

$$y_j = y_0 + j\Delta y$$

$$z_k = z_0 + k\Delta z$$

- Represent as voxels



13

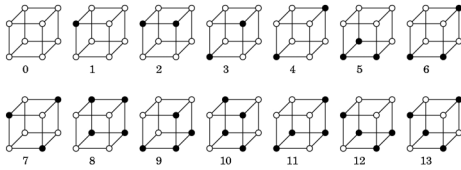
Isosurfaces

- $f(x,y,z)$ represents volumetric data set
- Two rendering methods
 - Isosurface rendering
 - Direct volume rendering (use all values [next])
- Isosurface given by $f(x,y,z) = c$
- Recall implicit surface $g(x, y, z)$:
 - $g(x, y, z) < 0$ inside
 - $g(x, y, z) = 0$ surface
 - $g(x, y, z) > 0$ outside
- Generalize right-hand side from 0 to c

14

Marching Cubes

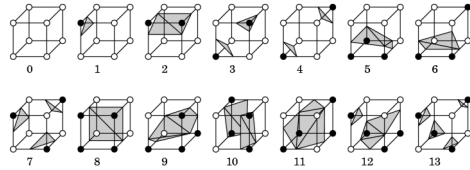
- Display technique for isosurfaces
- 3D version of marching squares
- 14 cube labelings (after elimination of symmetries)



15

Marching Cube Tessellations

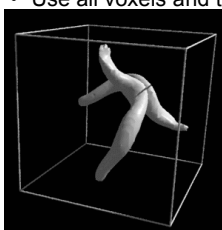
- Generalize marching squares, just more cases
- Interpolate as in 2D
- Ambiguities similar to 2D



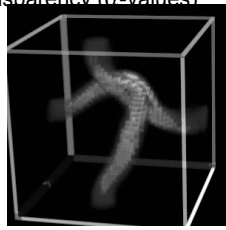
16

Volume Rendering

- Sometimes isosurfaces are unnatural or do not give sufficient information
- Use all voxels and transparency (α -values)



Ray-traced isosurface



Volume rendering

17

Surface vs. Volume Rendering

- | | |
|---------------------------|-----------------------------|
| • 3D model of surfaces | • Scalar field in 3D |
| • Convert to triangles | • Convert it to RGBA values |
| • Draw primitives | • Render volume "directly" |
| • Lose or disguise data | • See data as given |
| • Good for opaque objects | • Good for complex objects |

18

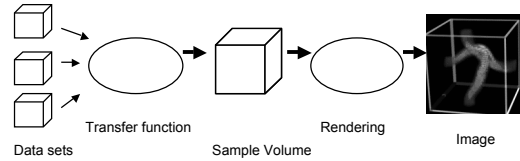
Sample Applications

- Medical
 - Computed Tomography (CT)
 - Magnetic Resonance Imaging (MRI)
 - Ultrasound
- Engineering and Science
 - Computational Fluid Dynamic (CFD)
 - Aerodynamic simulations
 - Meteorology
 - Astrophysics

19

Volume Rendering Pipeline

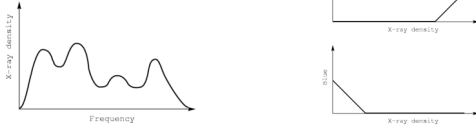
- Transfer function: from data set to colors and opacities
 - Example: 256 x 256 x 64 x 2 = 4 MB
 - Example: use colormap (8 bit color, 8 bit opacity)



20

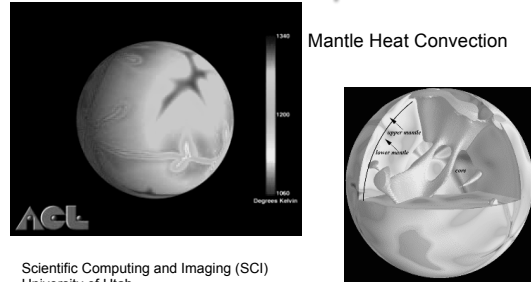
Transfer Functions

- Transform scalar data values to RGBA values
- Apply to every voxel in volume
- Highly application dependent
- Start from data histogram
- Opacity for emphasis



21

Transfer Function Example



Scientific Computing and Imaging (SCI)
University of Utah

22

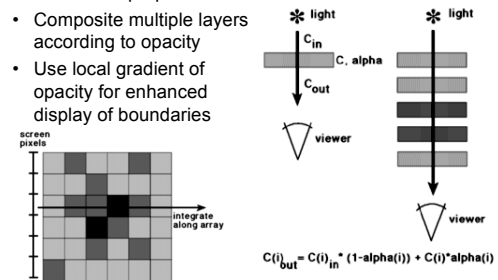
Volume Ray Casting

- Three volume rendering techniques
 - Volume ray casting
 - Splatting
 - 3D texture mapping
- Ray Casting
 - Integrate color through volume
 - Consider lighting (surfaces?)
 - Use regular x,y,z data grid when possible
 - Finite elements when necessary (e.g., ultrasound)
 - 3D-rasterize geometrical primitives

23

Accumulating Opacity

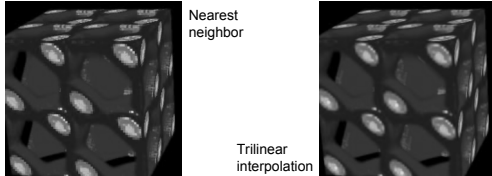
- $\alpha = 1.0$ is opaque
- Composite multiple layers according to opacity
- Use local gradient of opacity for enhanced display of boundaries



24

Trilinear Interpolation

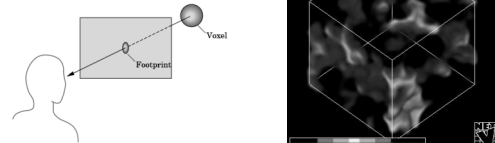
- Interpolate to compute RGBA away from grid
- Nearest neighbor yields blocky images
- Use trilinear interpolation
- 3D generalization of bilinear interpolation



25

Splatting

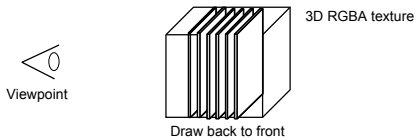
- Alternative to ray tracing
- Assign shape to each voxel (e.g., Gaussian)
- Project onto image plane (splat)
- Draw voxels back-to-front
- Composite (α -blend)



26

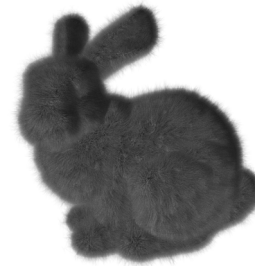
3D Textures

- Alternative to ray tracing, splatting
- Build a 3D texture (including opacity)
- Draw a stack of polygons, back-to-front
- Efficient if supported in graphics hardware
- Few polygons, much texture memory



27

Example: 3D Textures



Emil Praun'01

28

Example: 3D Textures

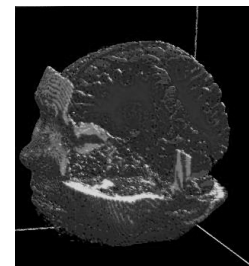
Emil Praun'01



29

Other Techniques

- Use CSG for cut-aways



30

Acceleration of Volume Rendering

- Basic problem: Huge data sets
- Must program for locality (cache)
- Divide into multiple blocks if necessary
 - Example: marching cubes
- Use error measures to stop iteration
- Exploit parallelism

31

Outline

- Height Fields and Contours
- Scalar Fields
- Volume Rendering
- Vector Fields

32

Vector Fields

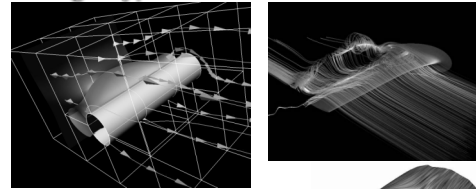
- Visualize vector at each (x,y,z) point
 - Example: velocity field
 - Example: hair
- Hedgehogs
 - Use 3D directed line segments (sample field)
 - Orientation and magnitude determined by vector
- Animation
 - Use for still image
 - Particle systems

Blood flow in
human carotid artery



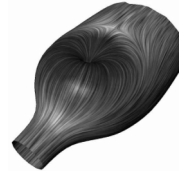
33

Using Glyphs and Streaklines



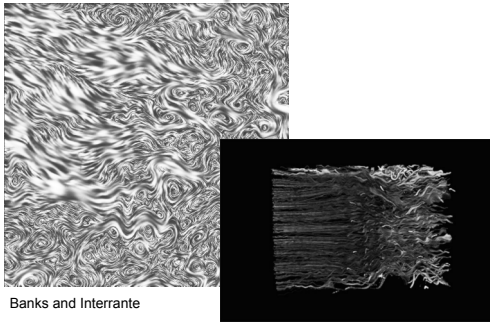
Glyphs for air flow
University of Utah

Glyph = marker (for example,
an arrow) used for data visualization



34

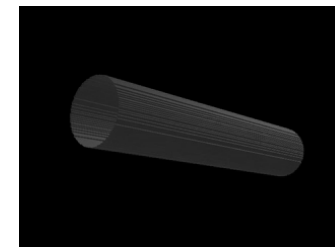
More Flow Examples



Banks and Interrante

35

Example: Jet Shockwave



P. Sutton
University of Utah

<http://www.sci.utah.edu/>

36

Summary

- Height Fields and Contours
- Scalar Fields
 - Isosurfaces
 - Marching cubes
- Volume Rendering
 - Volume ray tracing
 - Splatting
 - 3D Textures
- Vector Fields
 - Hedgehogs
 - Animated and interactive visualization

37