# 15-462 Computer Graphics I Lecture 20

### Visualization

Height Fields and Contours

Scalar Fields

**Volume Rendering** 

**Vector Fields** 

[Angel Ch. 12]

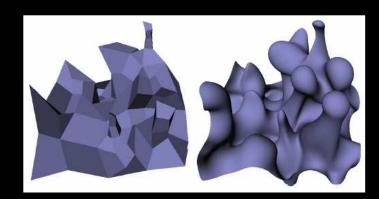
November 20, 2003

Doug James
Carnegie Mellon University

http://www.cs.cmu.edu/~djames/15-462/Fall03

### Scientific Visualization

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



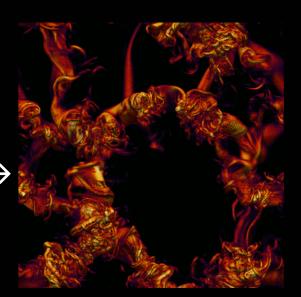
### Types of Data

- 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 [Angel 12.7]
- Static or through time

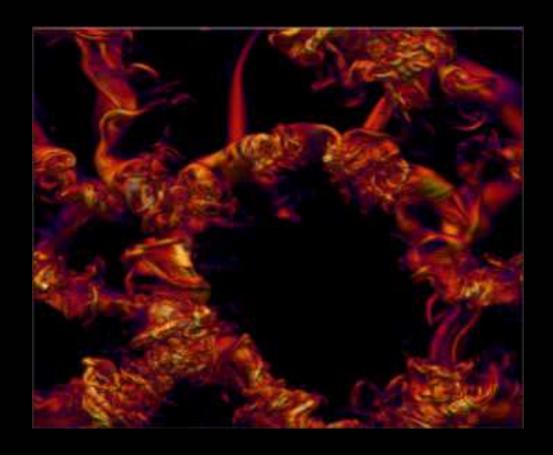
# Example: Turbulent convection

- Penetrative turbulent convection in a compressible ideal gas (PTCC)
- http://www.vets.ucar.edu/vg/PTCC/index.shtml

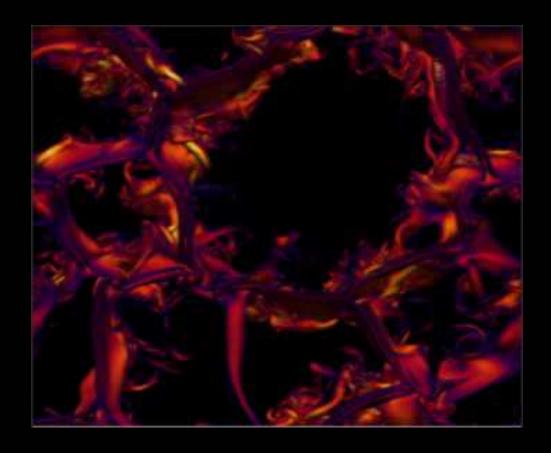
plot of enstrophy (vorticity squared) →



# Example: Turbulent convection



# Example: Turbulent convection

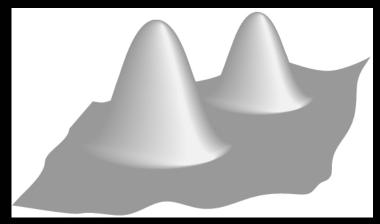


How do we visualize volume data like this?

# Height Field

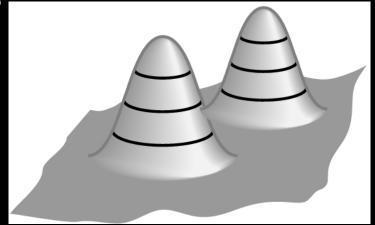
Visualizing an explicit function

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



Adding contour curves

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



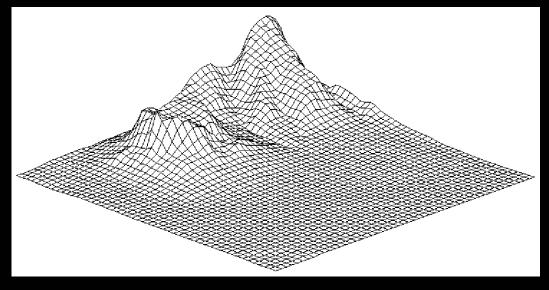
### Meshes

- Function is sampled (given) at  $x_i$ ,  $y_i$ ,  $0 \le i$ ,  $j \le n$
- Assume equally spaced

$$\begin{array}{rcl}
x_i & = & x_0 + i\Delta x \\
y_j & = & y_0 + j\Delta y
\end{array}$$

$$z_{ij} = f(x_i, y_j)$$

- Generate quadrilateral or triangular mesh
- [Asst 1]



#### **Contour Curves**

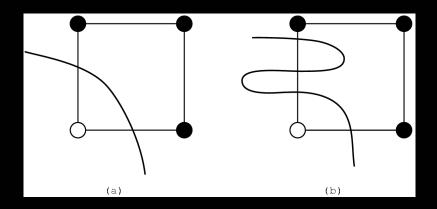
- Recall: implicit curve f(x,y) = 0
- f(x,y) < 0 inside, f(x,y) > 0 outside
- Here: contour curve at f(x,y) = c
- Sample at regular intervals for x,y

$$\begin{array}{rcl} x_i & = & x_0 + i\Delta x \\ y_j & = & y_0 + j\Delta y \end{array}$$

How can we draw the curve?

# Marching Squares

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

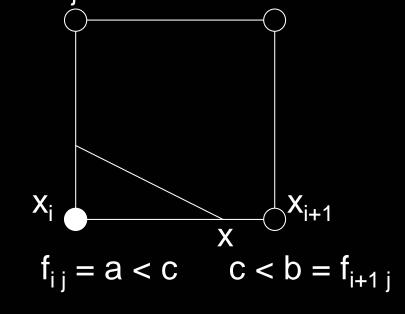


# Interpolating Intersections

- Approximate intersection
  - Midpoint between x<sub>i</sub>, x<sub>i+1</sub> and y<sub>i</sub>, y<sub>i+1</sub>
  - Better: interpolate
- If f<sub>ij</sub> = a is closer to c than b = f<sub>i+1 j</sub> then intersection is closer to (x<sub>i</sub>, y<sub>i</sub>):

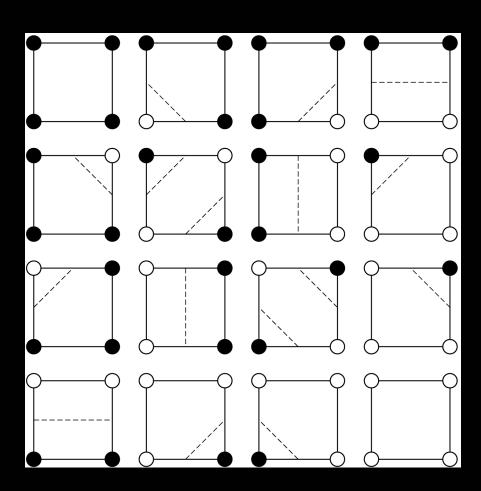
$$\frac{x - x_i}{x_{i+1} - x} = \frac{c - a}{b - c}$$

 Analogous calculation for y direction

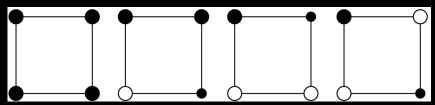


### Cases for Vertex Labels

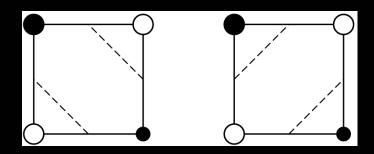
16 cases for vertex labels



4 unique mod. symmetries

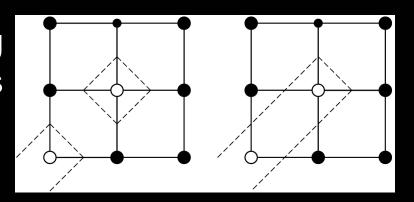


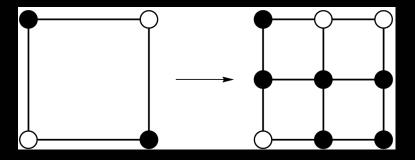
# **Ambiguities of Labelings**



Ambiguous labels

Different resulting contours



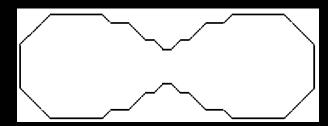


Resolution by subdivision (where possible)

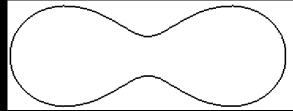
# Marching Squares Examples

Ovals of Cassini, 50 × 50 grid

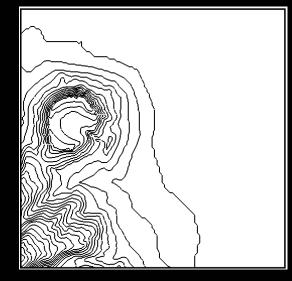
$$f(x,y) = (x^2 + y^2 + a^2)^2 - 4a^2x^2 - b^4$$
  
  $a = 0.49, b = 0.5$ 



Midpoint



Interpolation



Contour plot of Honolulu data

### **Outline**

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

### Scalar Fields

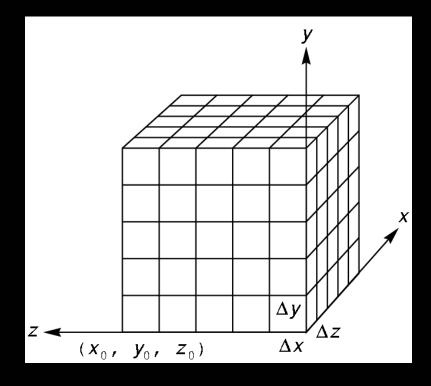
- Volumetric data sets
- Example: tissue density
- Assume again regularly sampled

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

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

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

Represent as voxels

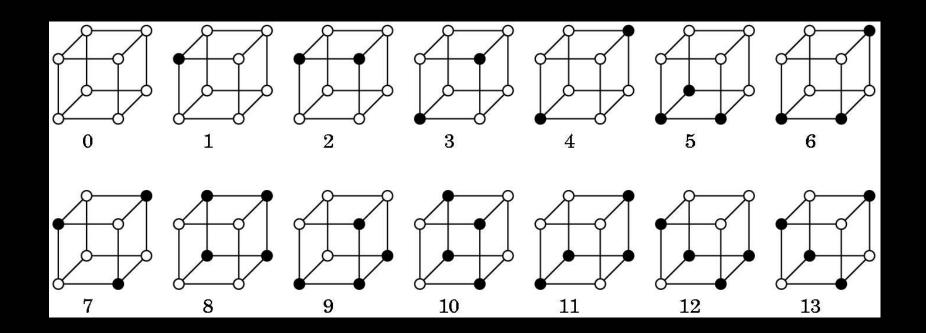


### Isosurfaces

- 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

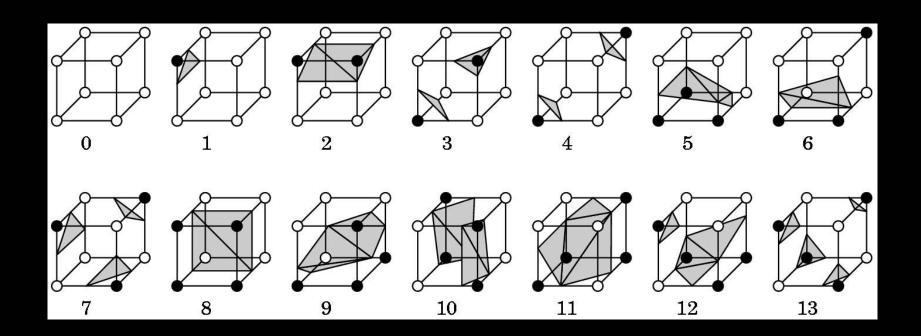
# Marching Cubes

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



## Marching Cube Tessellations

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

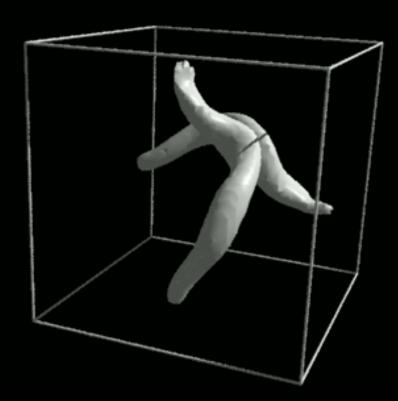


# Marching Cubes Demo

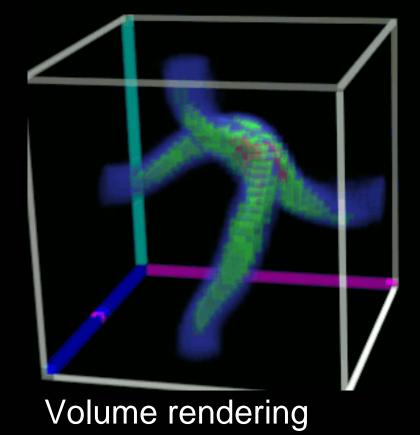


# Volume Rendering

- Sometimes isosurfaces are unnatural
- Use all voxels and transparency (α-values)



Ray-traced isosurface



### Surface vs. Volume Rendering

- 3D model of surfaces
- Convert to triangles
- Draw primitives
- Lose or disguise data
- Good for opaque objects

- Scalar field in 3D
- Convert to RGBA values
- Render volume "directly"
- See data as given
- Good for complex objects

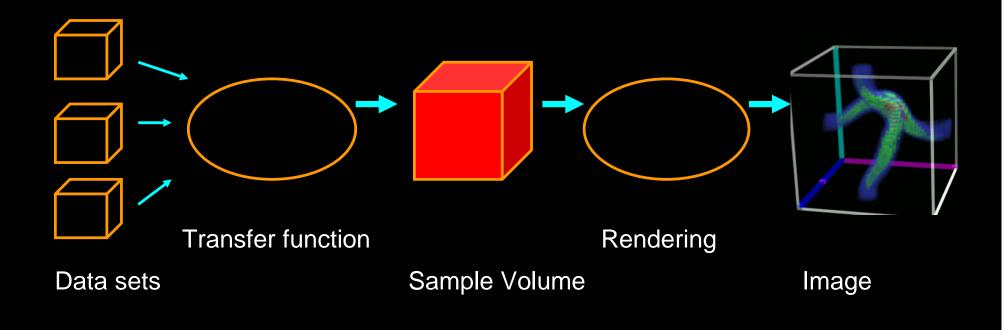
# Sample Applications

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

# Volume Rendering Pipeline

04/15/2003

- Transfer function: from data set to colors and opacities
  - Example:  $256 \times 256 \times 64 \times 2 = 4 \text{ MB}$
  - Example: use colormap (8 bit color, 8 bit opacity)

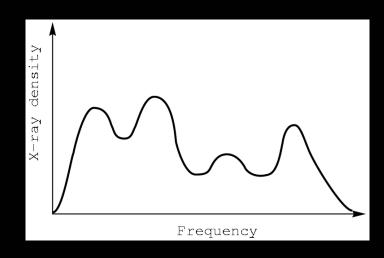


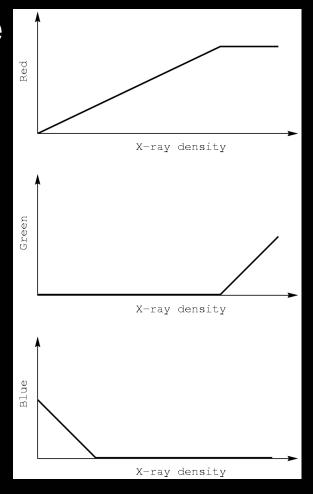
15-462 Graphics I

24

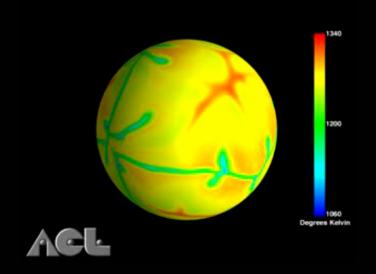
#### **Transfer Functions**

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

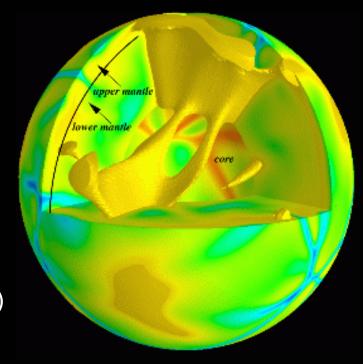




# Transfer Function Example



Mantle Convection



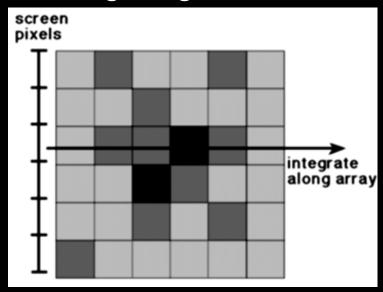
Scientific Computing and Imaging (SCI)
University of Utah

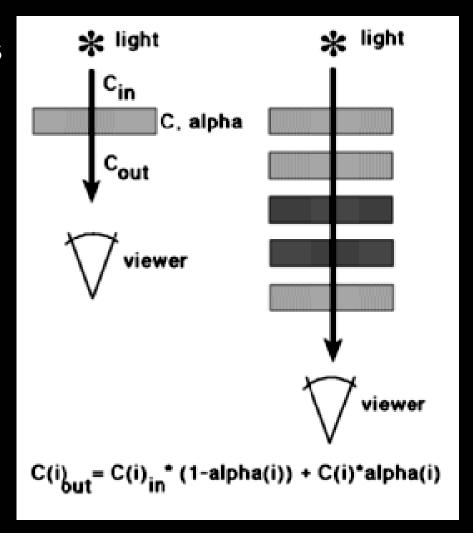
# Volume Ray Casting

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

# **Accumulating Opacity**

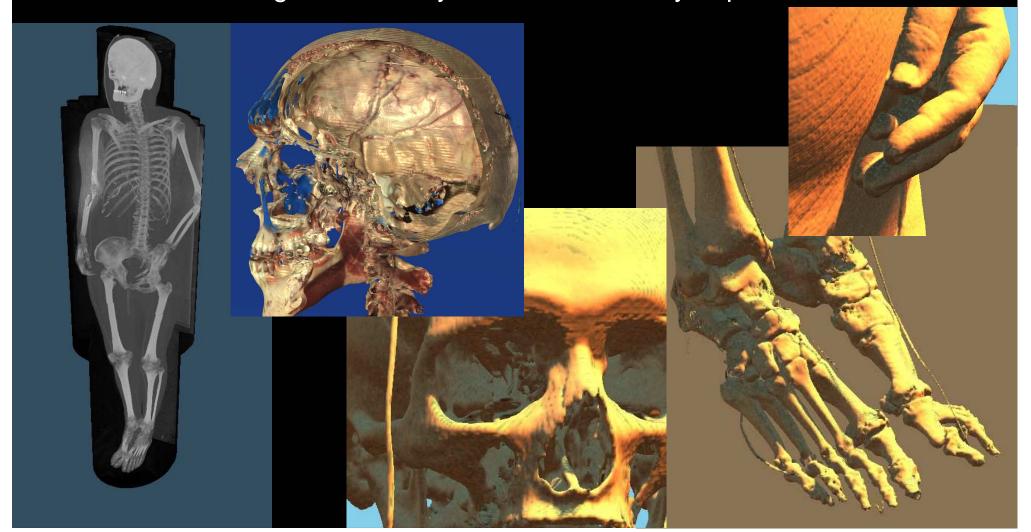
- $\alpha$  = 1.0 is opaque
- Composity multiple layers according to opacity
- Use local gradient of opacity to detect surfaces for lighting





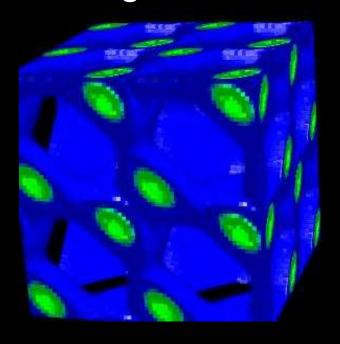
### Real-Time Ray Tracer (Utah)

- Implemented on SGI Origin 3000 ccNUMA architecture up to 256 processors (now working on a distrbuted version)
- Approximately linear speedup
- Load balancing and memory coherence are key to performance



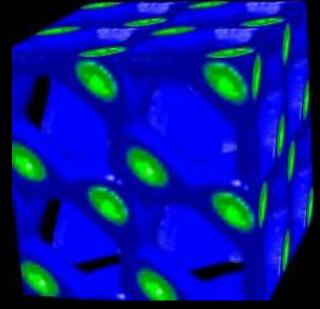
### Trilinear Interpolation

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



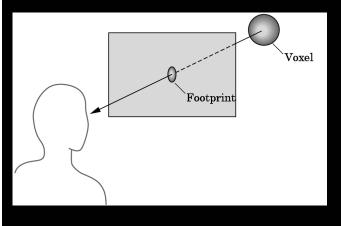
Nearest neighbor

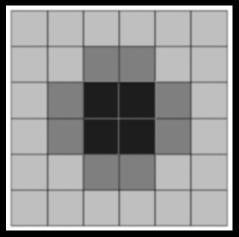
Trilinear interpolation



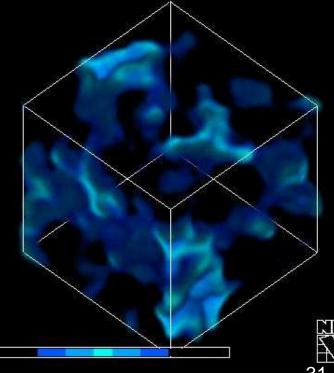
# **Splatting**

- Wonderfully simple
- Working back-to-front (or front-to-back), draw a "splat" for each chunk of data
- Easy to implement, but not as accurate as ray casting
- Works reasonably for non-uniform data



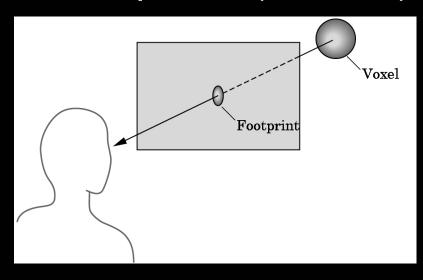


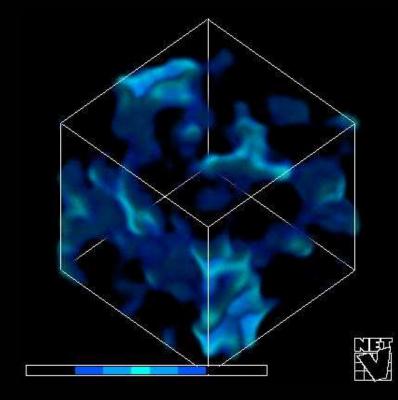
closeup of a splat



# **Splatting**

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

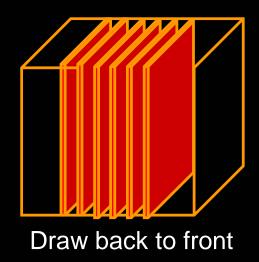




### 3D Textures

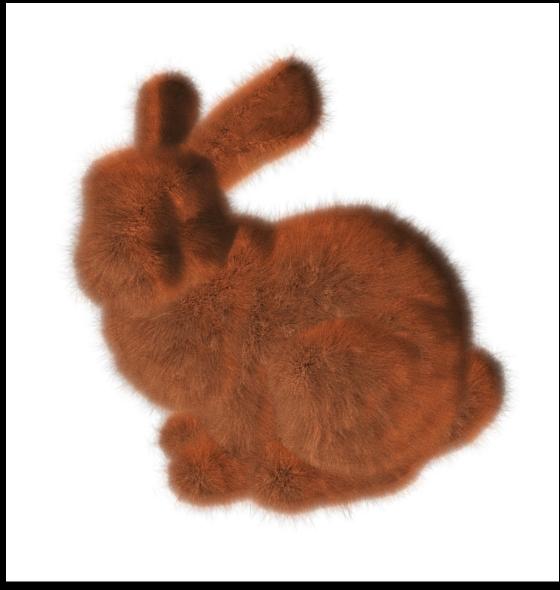
- Alternative to ray tracing, splatting
- Build a 3D texture (including opacity)
- Draw a stack of polygons, back-to-front
- Efficient if supported in graphics hardware
- Few polygons, much texture memory





3D RGBA texture

# Example: 3D Textures



Emil Praun'01

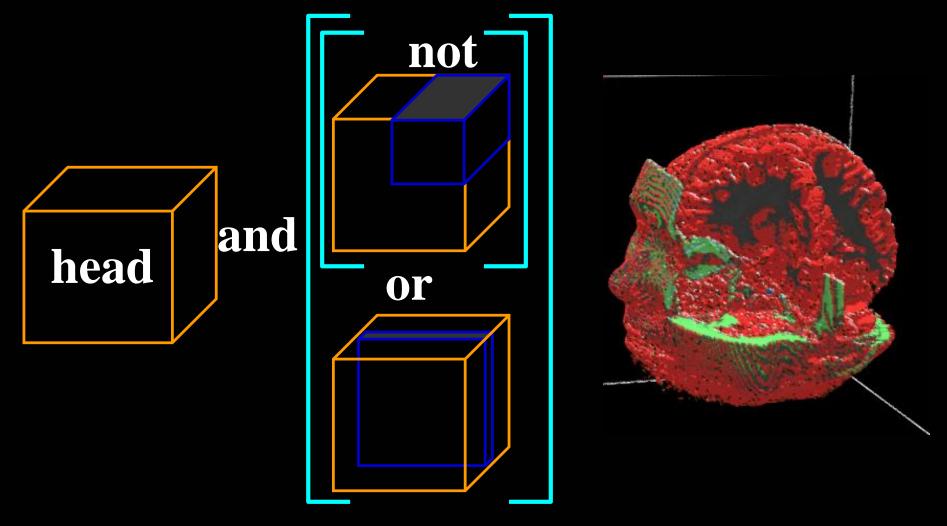
# Example: 3D Textures

Emil Praun'01

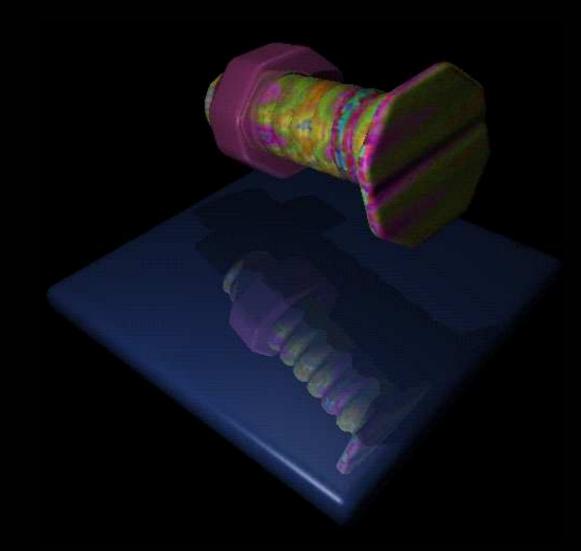


### Other Techniques—CSG is Easy

- The transfer function can be used to mask a volume or merge volumes
- You are still confined to the grid, of course



# Another CSG example



(VolVis again)

### Teaser: Explosion Rendering



Smoke Simulation For Large Scale Phenomena, Nguyen et al., SIGGRAPH 2003.

#### Acceleration of Volume Rendering

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

#### **Outline**

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

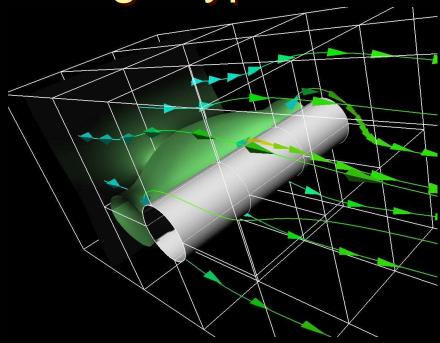
#### **Vector Fields**

- 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

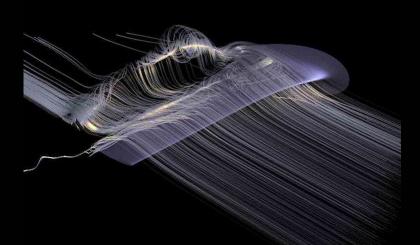
Blood flow in human carotid artery



# Using Glyphs and Streaks

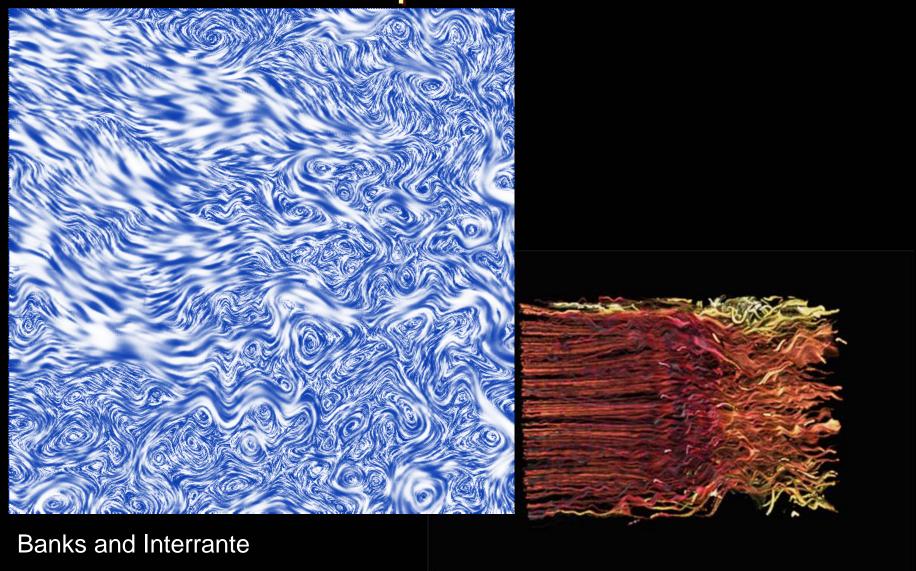


Glyphs for air flow



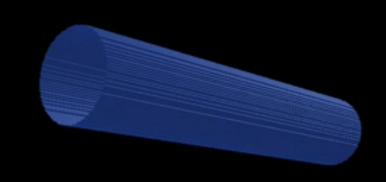
University of Utah

# More Flow Examples



## Example: Jet Shockwave

P. Sutton
University of Utah



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



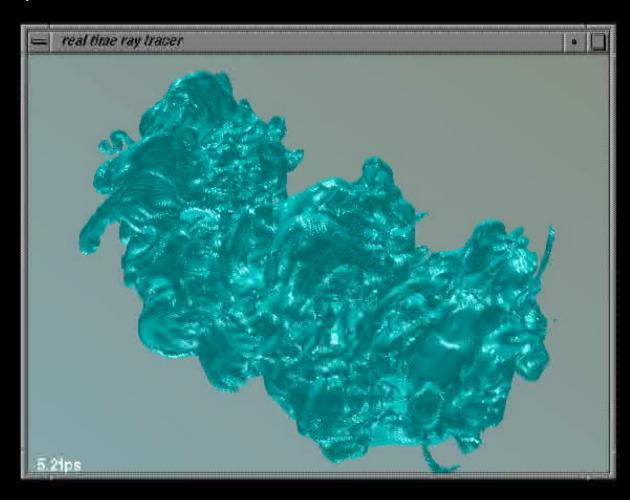
#### Interaction: Data Probe

SCI, Utah



#### Jet shockwave

- Jet Shockwave: 256x256x256 (DOE AVTC)
  - Kelvin-Helmholtz instability (supersonic jet)
  - Time dependent interactive visualization



SCI, Utah

#### 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

# Top Visualization Research Issues

- Error and Uncertainty Visual Representation
- Perceptual Issues
- Quantify Effectiveness
- Time Dependent Visualization
- Pipeline Complexity
- Think About the Science
- Interesting Feature Detection
- Leverage both Hardware and Software Methods
- Multi-field Visualization

#### From Chris Johnson, Utah