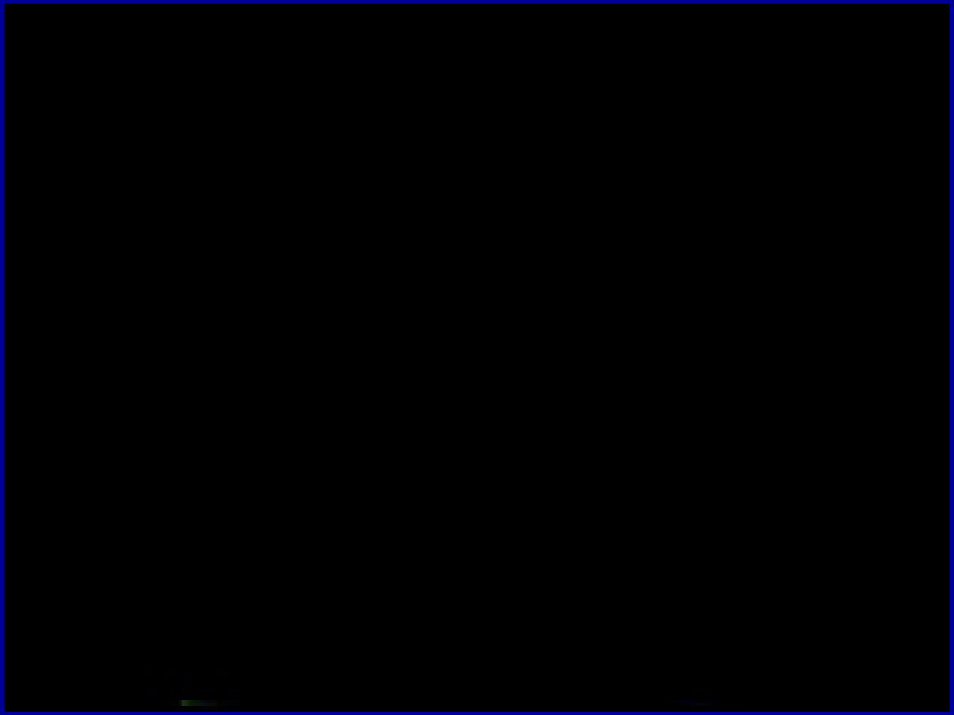
Visualization

Images are used to aid in understanding of data

Height Fields and Contours
Scalar Fields
Volume Rendering
Vector Fields

[chapter 26]

Tumor

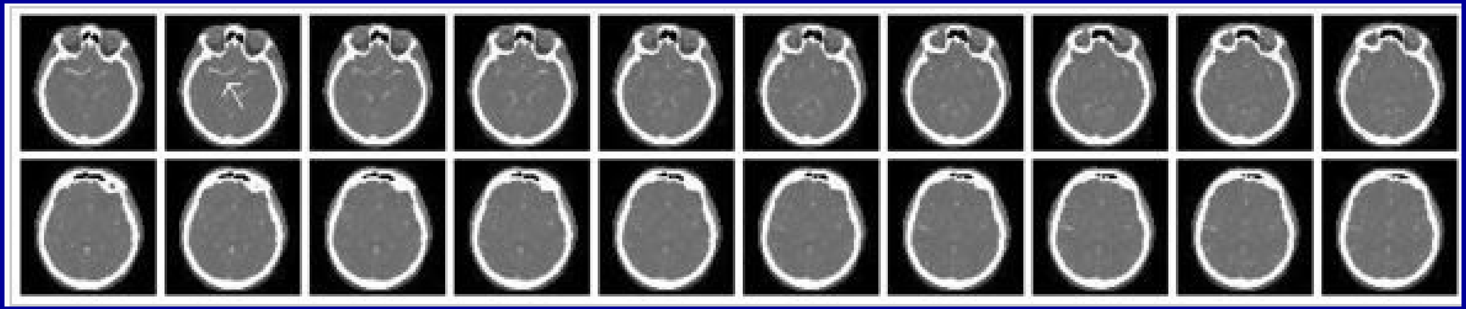


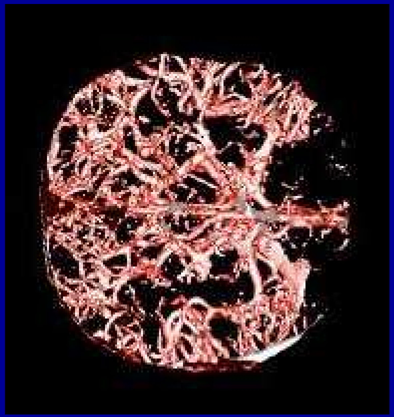
SCI, Utah

Scientific Visualization

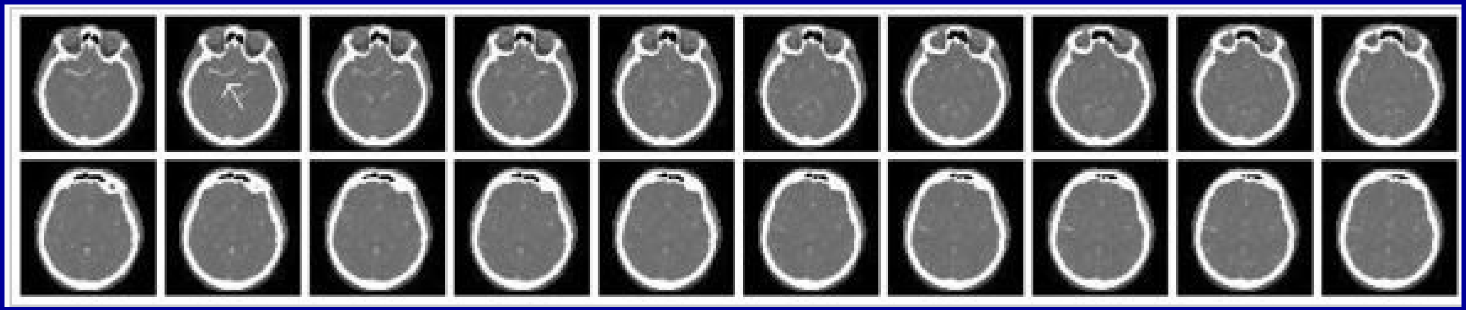
- Visualize large datasets in scientific and medical applications
- Generally do not start with a 3D model

CT Scan - whiter means higher radiodensity



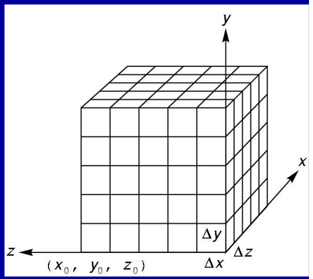


Scientific Visualization



- Must deal with very large data sets
  - CT or MRI, e.g. $512 \times 512 \times 200 \approxeq$ 50MB points
  - Visible Human $512 \times 512 \times 1734 \approxeq$ 433 MB points
- Visualize both real-world and simulation data
  - Visualization of Earthquake Simulation Data
  - Visualizations of simulated room fires
  - Fluid simulation

Types of Data

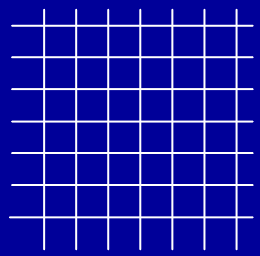


- Scalar fields (2D or 3D volume of scalars)
  - E.g., x-ray densities (MRI, CT scan)
- Vector fields (3D volume of vectors)
  - E.g., velocities in a wind tunnel
- Tensor fields (3D volume of tensors [matrices])
  - E.g., stresses in a mechanical part

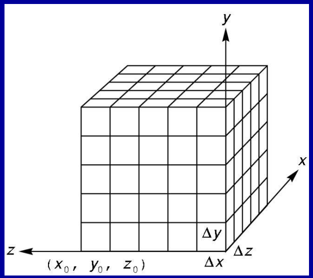
All could be static or through time

Outline

- 2D Scalar Field (Height Fields) $z = f(x,y)$



- 3D Scalar Fields $v = f(x,y,z)$



- Volume Rendering

- Vector Fields

Blood flow in
human carotid artery

2D Scalar Field

- z = f(x,y)

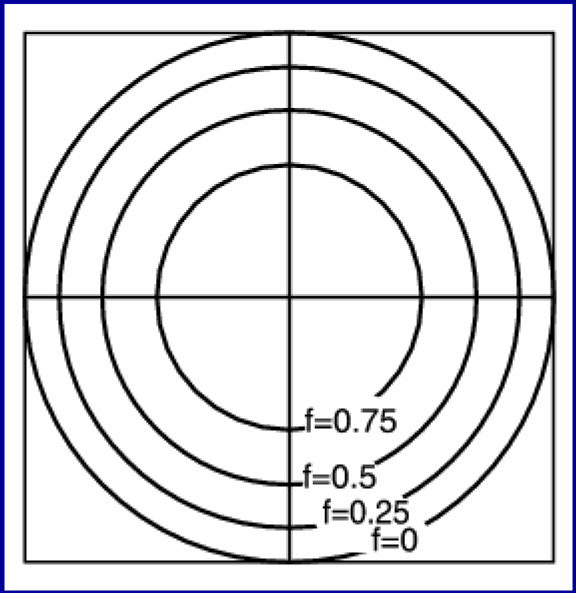
$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 & \end{cases}$$

How do you visualize this function?

2D Scalar Field

- z = f(x,y)

$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 \end{cases}$$



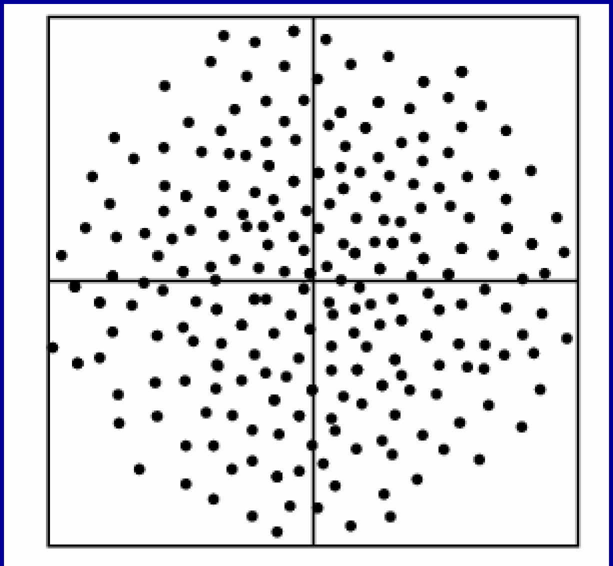
Contours

Topographical maps to indicate elevation

2D Scalar Field

- z = f(x,y)

$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 \end{cases}$$

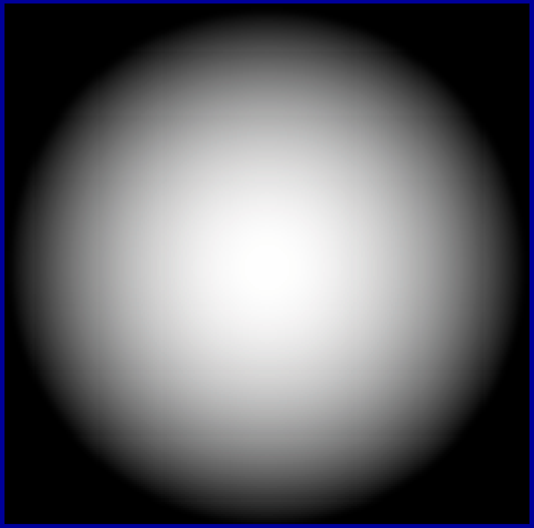


Density plot
Density is proportional to the value of the function

2D Scalar Field

- z = f(x,y)

$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 \end{cases}$$



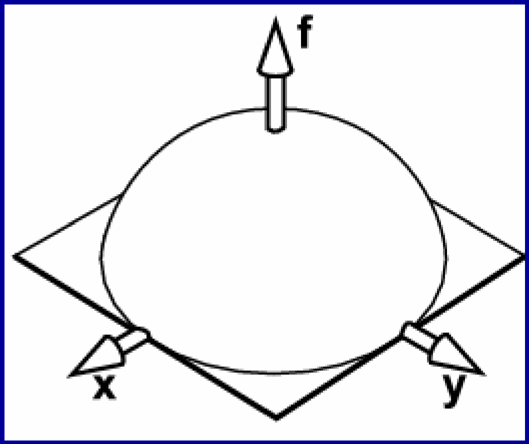
Gray scale density plot

z = 0 => (0,0,0)
z = 0.25 => (0,0,1)
z = 0.5 => (1,0,0)
z = 0.75 => (1,1,0)
z = 1.0 => (1,1,1)

2D Scalar Field

- z = f(x,y)

$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 & \end{cases}$$

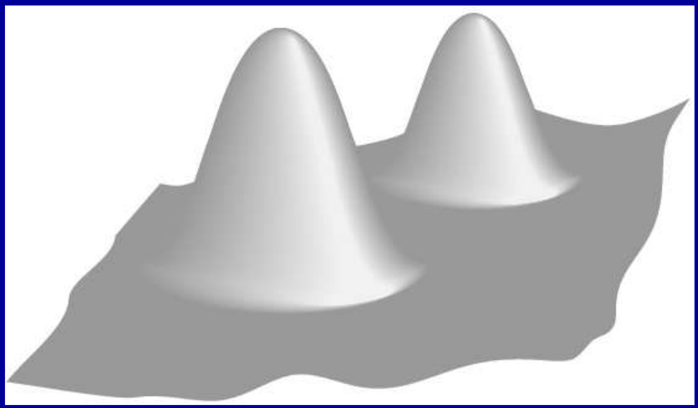


Height plot
Shows shape of the function

Height Field

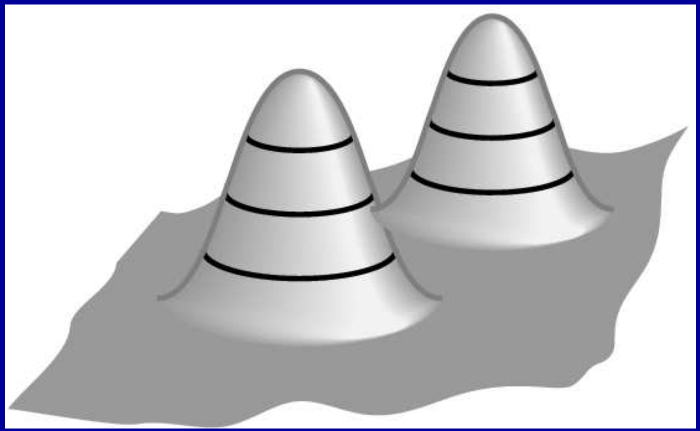
- Visualizing an explicit function

$z = f(x,y)$



- Adding contour curves

$f(x,y) = c$



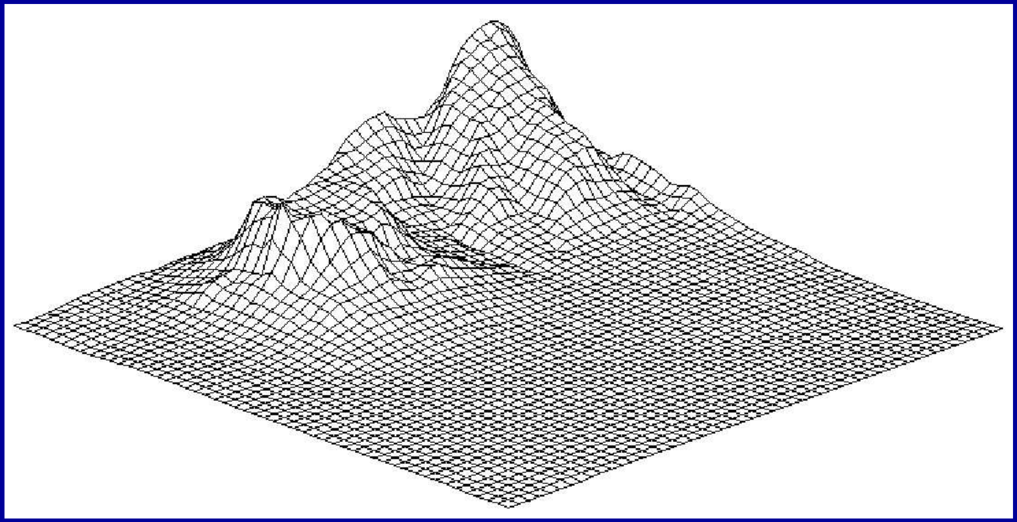
Meshes

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

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

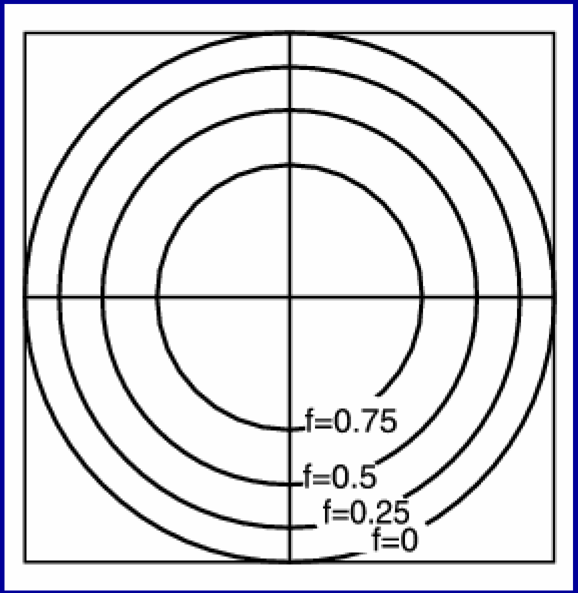
$$z_{i\,j} = f(x_i, y_j)$$

- Generate quadrilateral or triangular mesh
- [Asst 1]



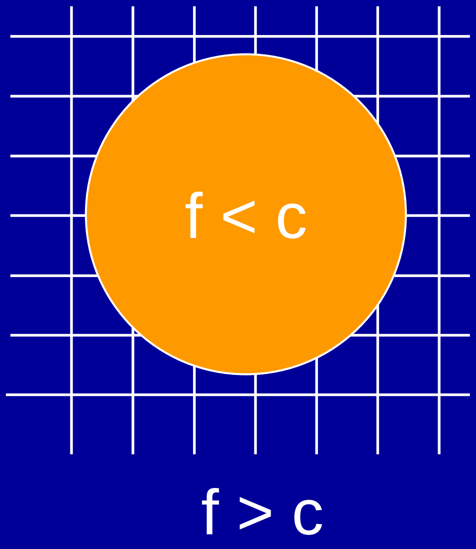
Contour Curves

$$f(x,y) = \begin{cases} 1 - x^2 - y^2, & \text{if } x^2 + y^2 < 1 \\ 0 \end{cases}$$



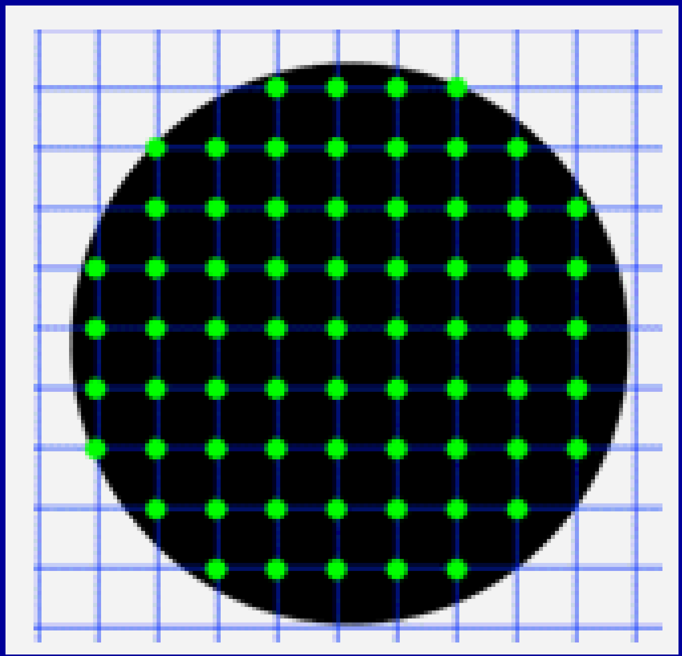
- Contour curve at f(x,y) = c
- How can we draw the curve?
- Sample at regular intervals for x,y

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

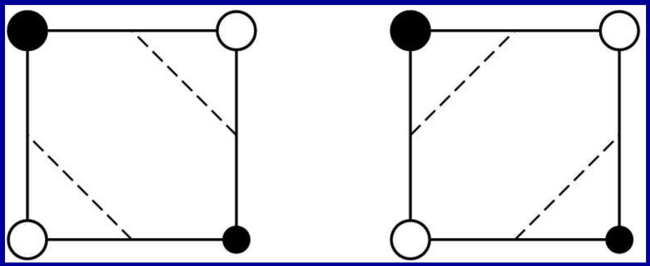


Marching Squares

- Sample function f at every grid point x_i, y_j
- For every point $f_{i\,j} = f(x_i, y_j)$ either $f_{i\,j} \leq c$ or $f_{i\,j} > c$

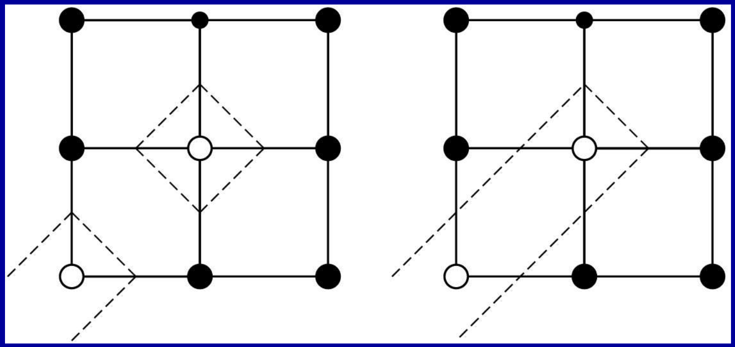


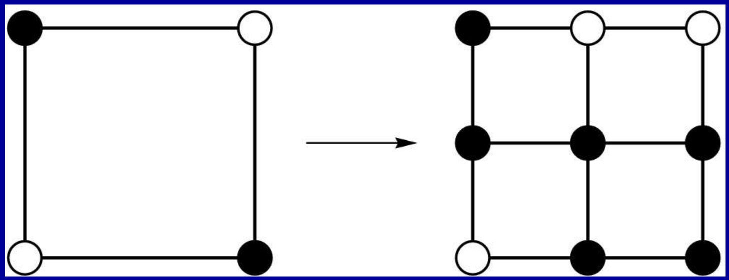
Ambiguities of Labelings



Ambiguous labels

Different resulting contours

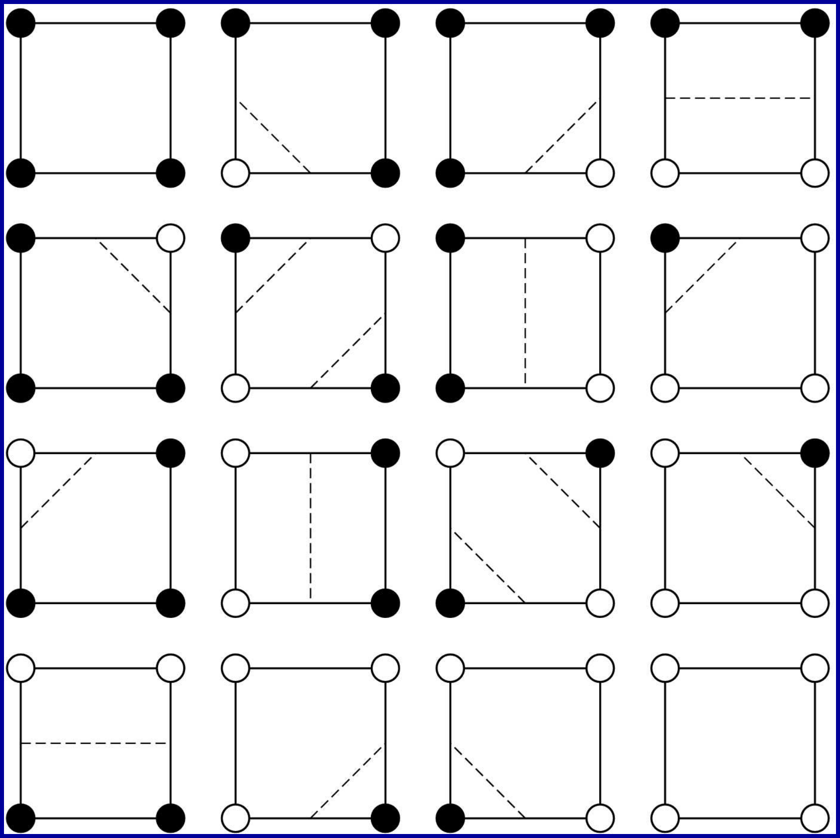




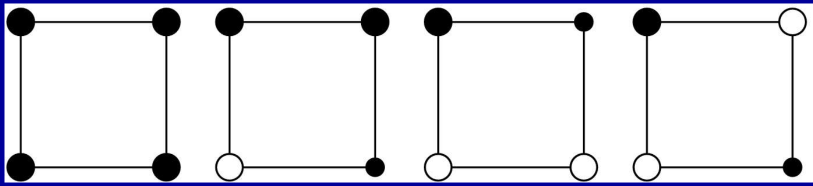
Resolution by subdivision (where possible)

Cases for Vertex Labels

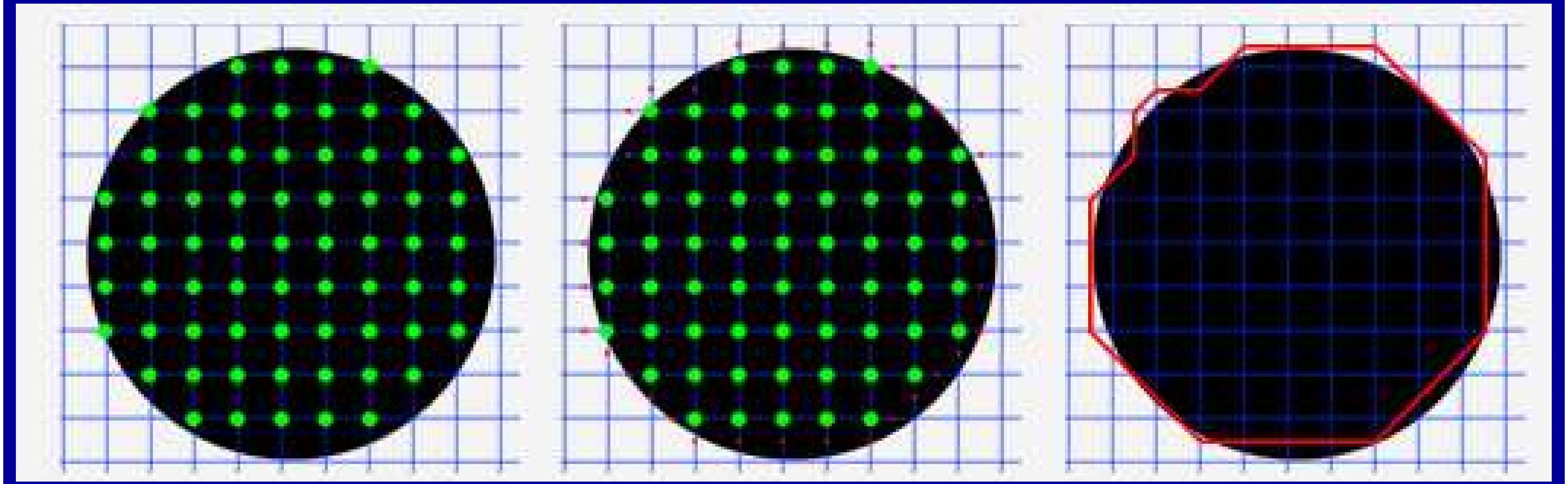
16 cases for vertex labels



4 unique mod. symmetries



Marching Squares Examples



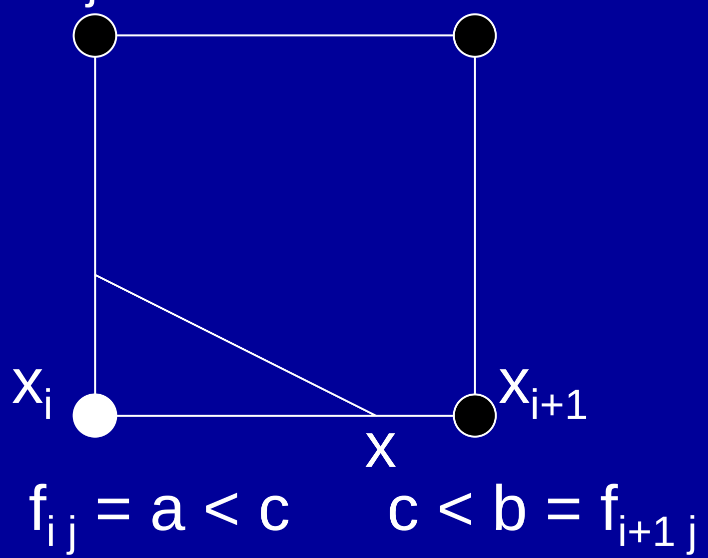
Can you do better?

Interpolating Intersections

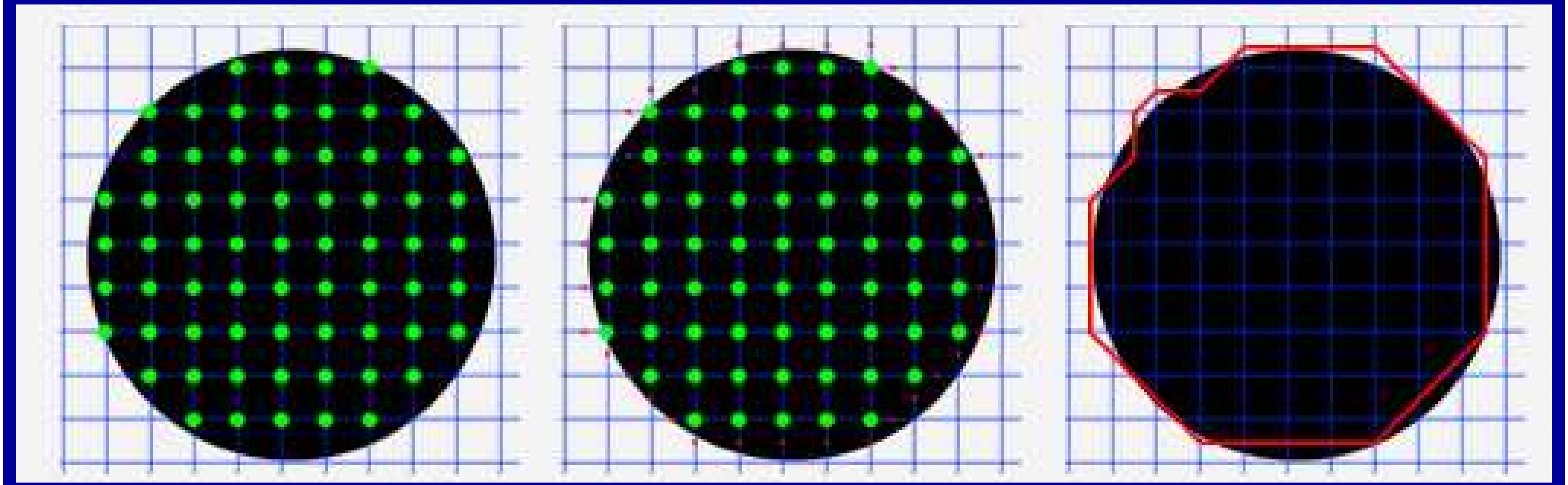
- Approximate intersection
  - Midpoint between x_i, x_{i+1} and y_j, y_{j+1}
  - Better: interpolate
- If $f_{i\,j} = a$ is closer to c than $b = f_{i+1\,j}$ then intersection is closer to (x_i, y_j):

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

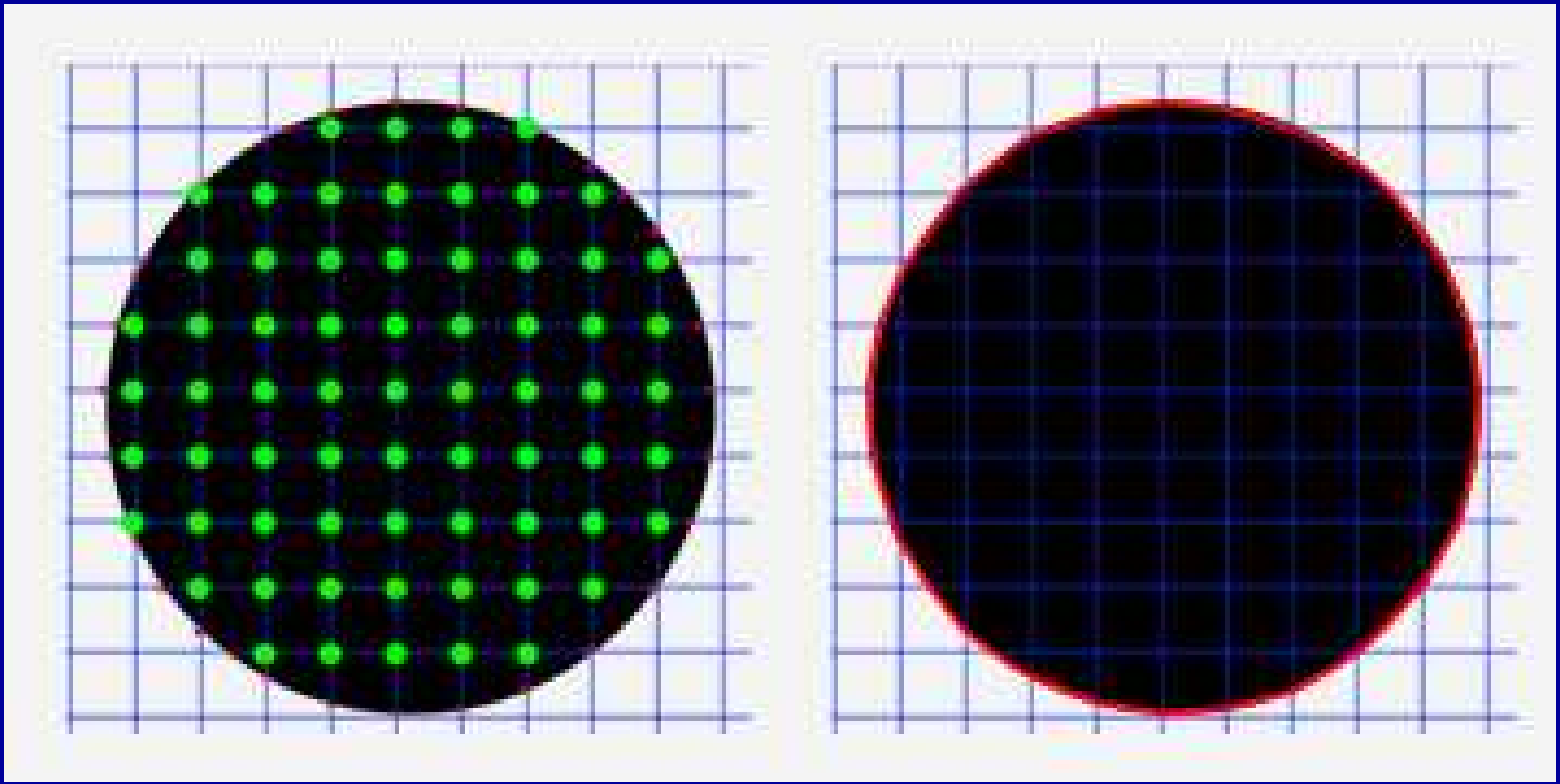
- Analogous calculation for y direction



Marching Squares Examples



Marching Squares Examples

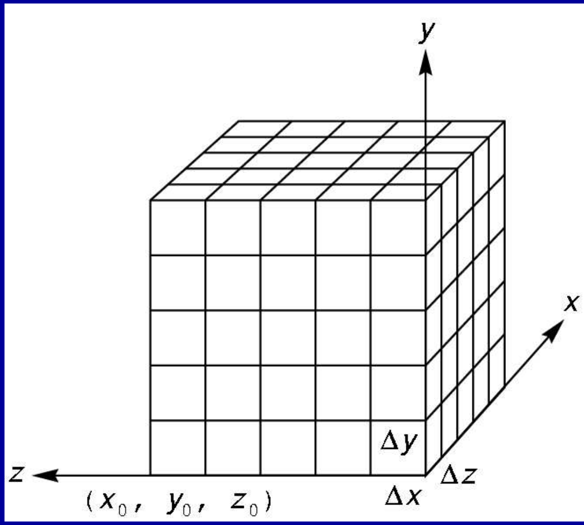


Adaptive Subdivision

Outline

- 2D Scalar Fields
- 3D Scalar Fields
- Volume Rendering
- Vector Fields

3D Scalar Fields



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

$$
\begin{aligned}
x_i &= x_0 + i\Delta x \\
y_j &= y_0 + j\Delta y \\
z_k &= z_0 + k\Delta z
\end{aligned}
$$

- Represent as voxels

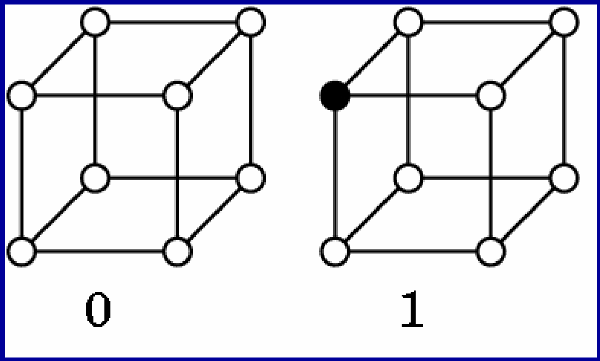
- Two rendering methods
  - –Isosurface rendering
  - –Direct volume rendering (use all values [next])

Isosurfaces

- Generalize contour curves to 3D
- Isosurface given by $f(x,y,z) = c$
  - $f(x, y, z) < c$ inside
  - $f(x, y, z) = c$ surface
  - $f(x, y, z) > c$ outside

Marching Cubes

- Display technique for isosurfaces
- 3D version of marching squares
- How many possible cases?

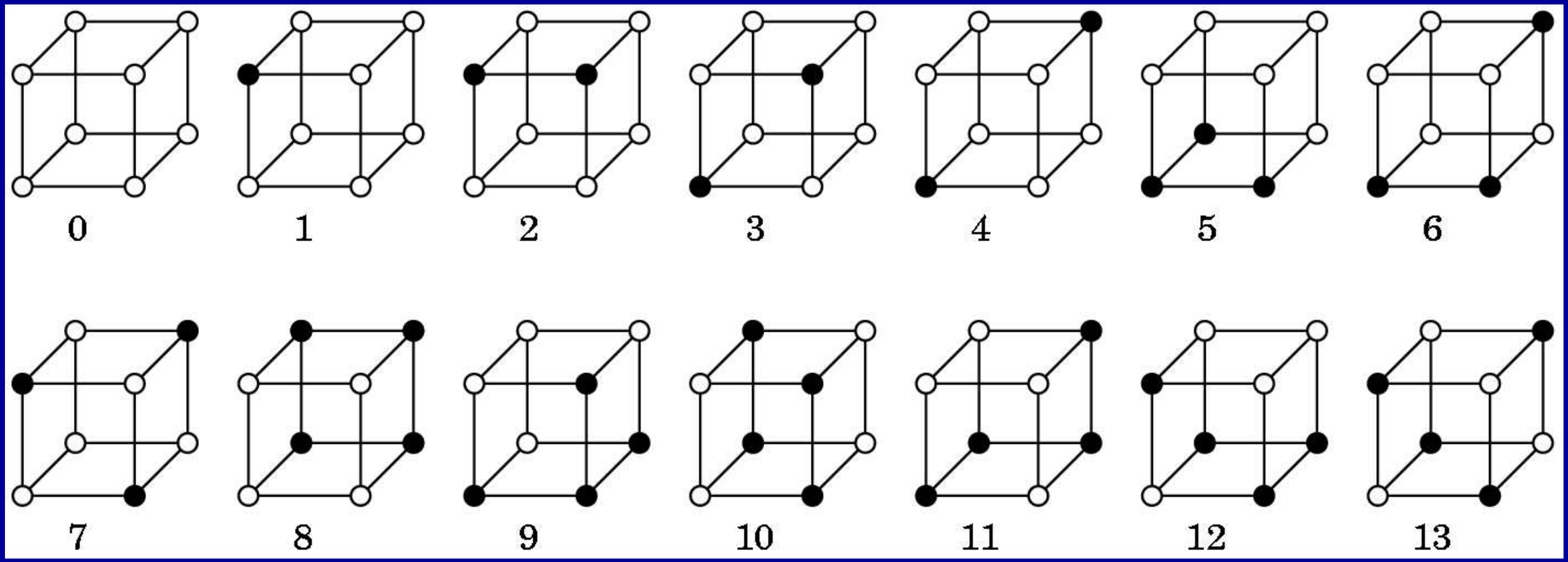


...

$2^8 = 256$

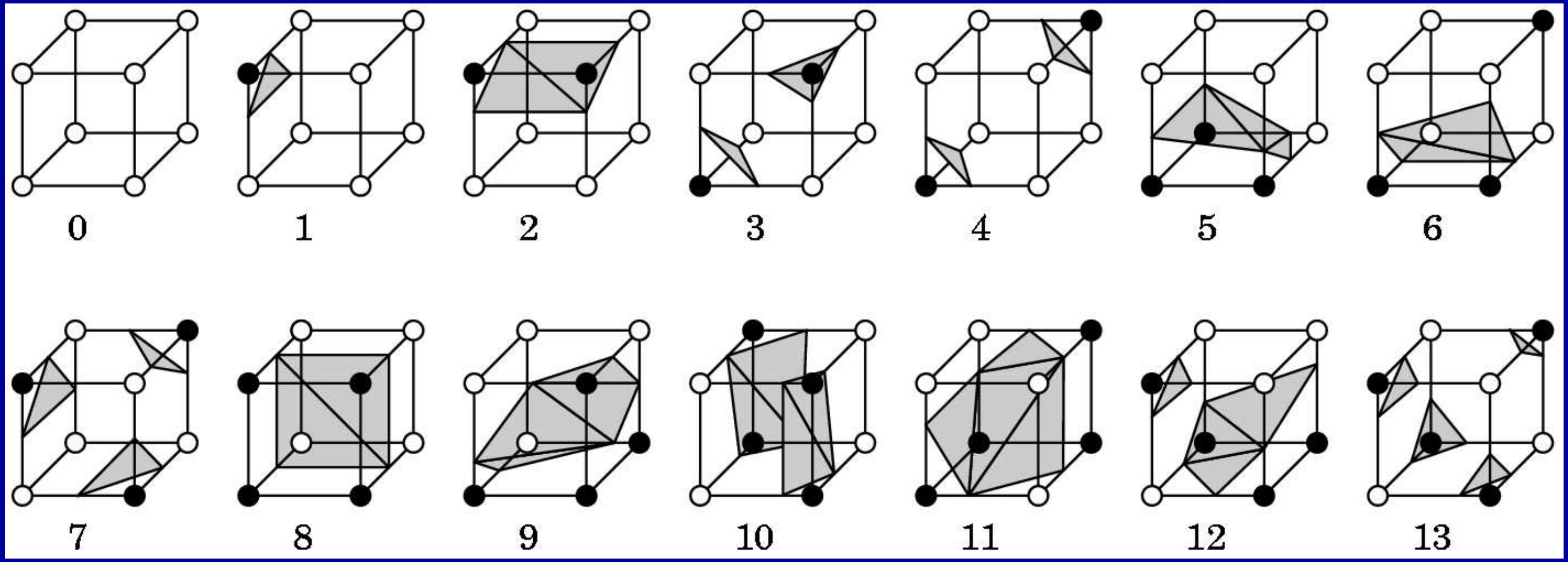
Marching Cubes

- 14 cube labelings (after elimination symmetries)

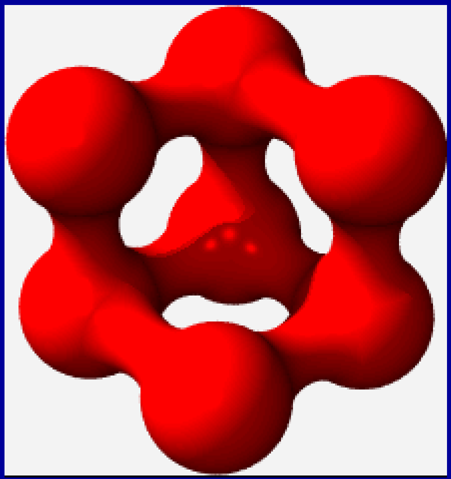


Marching Cube Tessellations

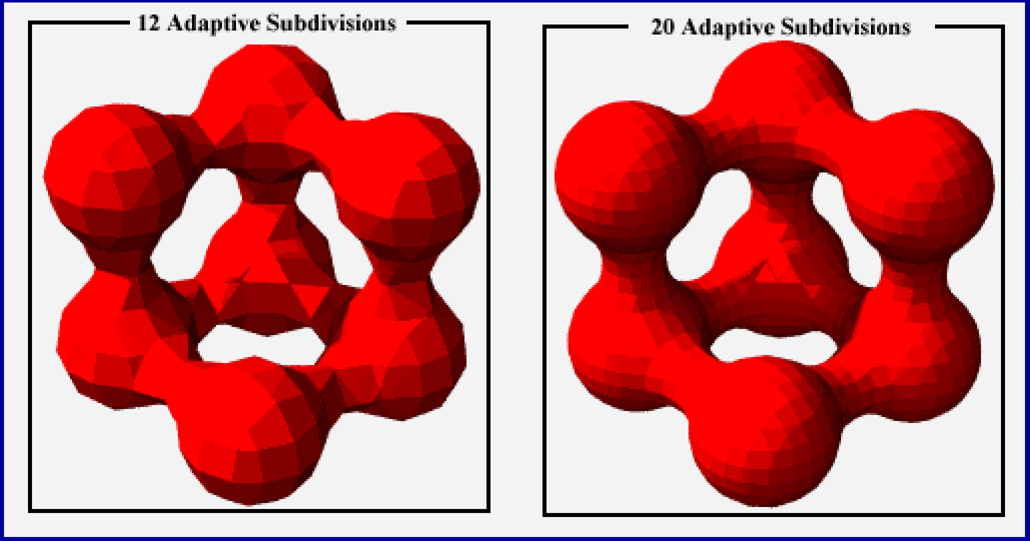
- Generalize marching squares, just more cases
- Interpolate as in 2D
- Ambiguities similar to 2D



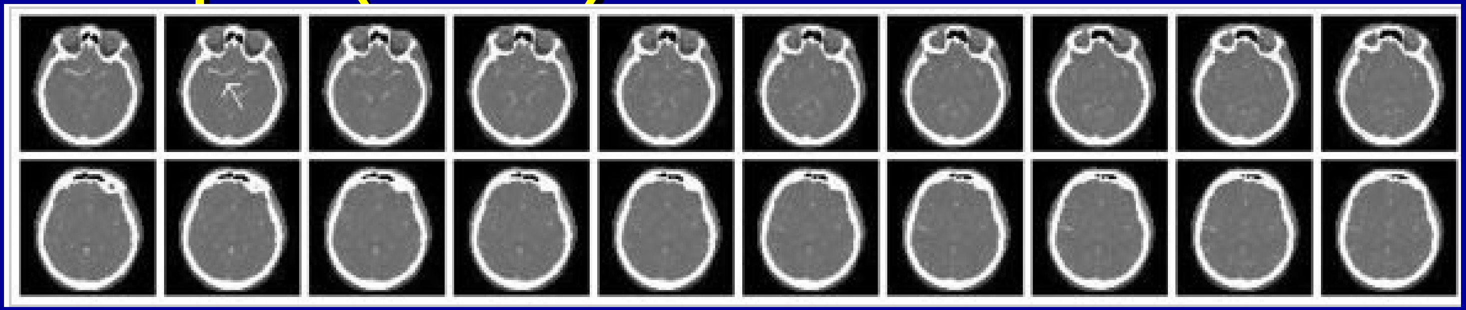
Marching Squares Examples

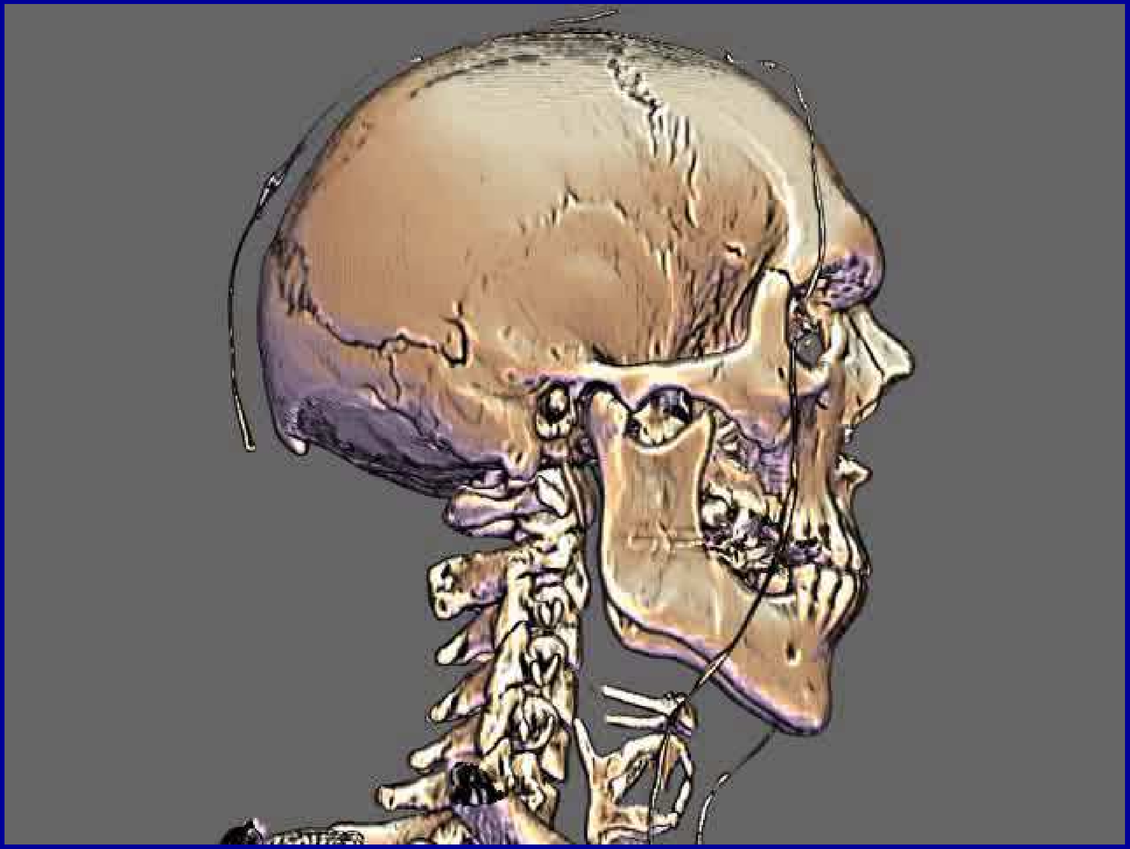


Marching Squares Examples



Example (Utah)



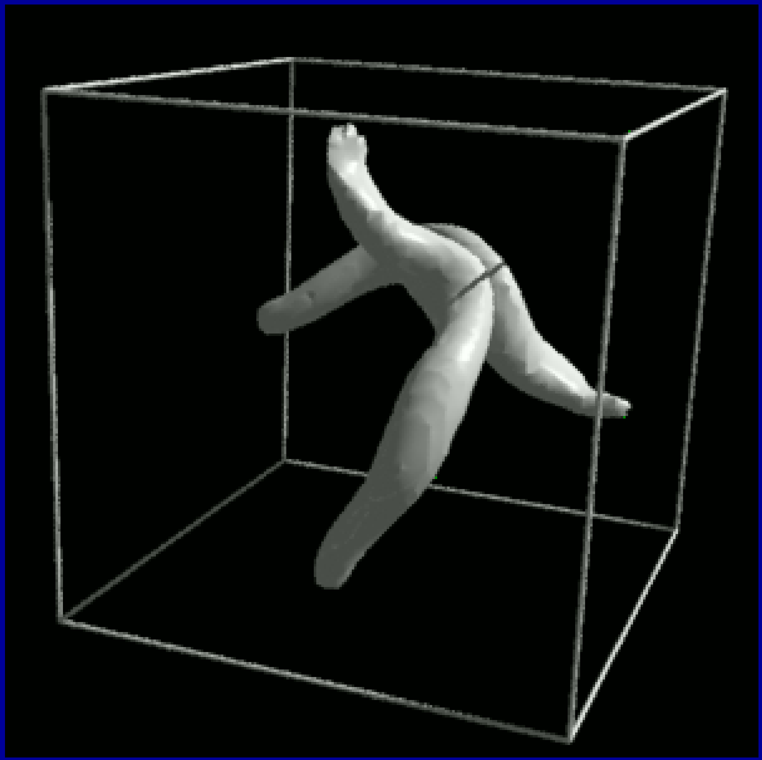


Outline

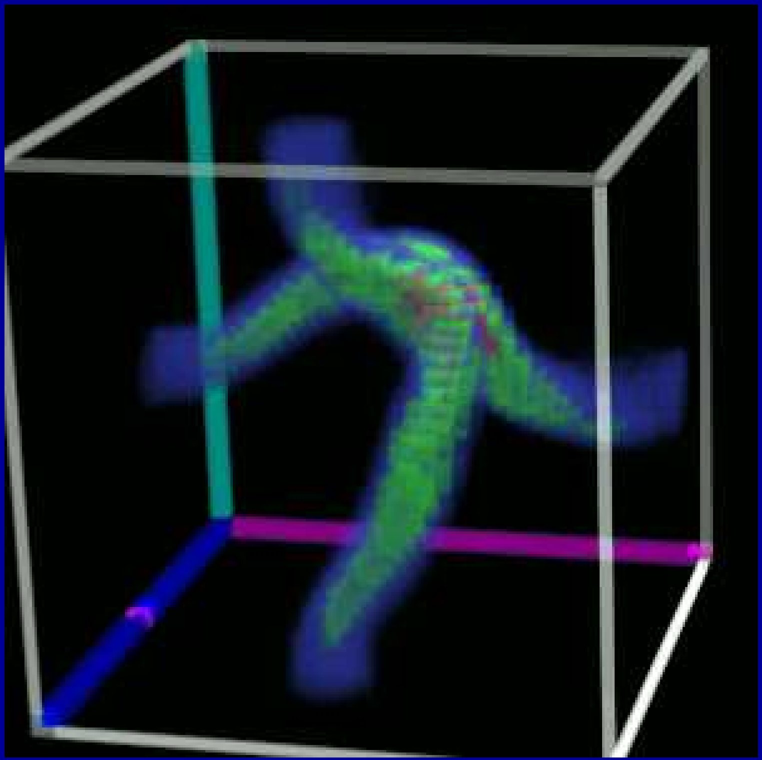
- 2D Scalar Fields
- 3D Scalar Fields
- Volume Rendering
- Vector Fields

Volume Rendering

- Some data is more naturally modeled as a volume, not a surface
- Use all voxels and transparency (a-values)



Ray-traced isosurface
$f(x,y,z)=c$



Same data, rendered
as a volume

Why Bother with Volume Rendering?

- Not all voxels contribute to final image
- Could miss most important data by selecting wrong isovalue

- All voxels contribute to the image
  - more informative
  - less misleading (the isosurface of noisy data is unpredictable)

- Simpler and more efficient than converting a very complex data volume (like the visible human) to polygons and then rendering them

Surface vs. Volume Rendering

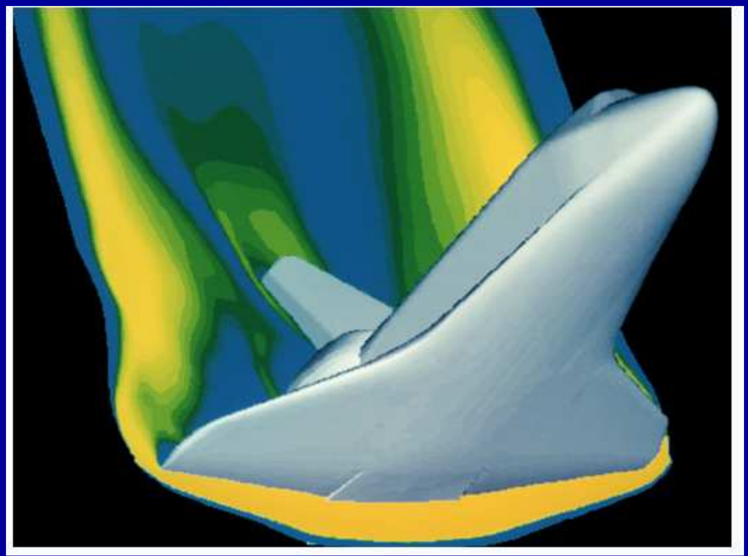
- 3D model of surfaces
- Convert to triangles
- Draw primitives
- Lose or disguise data
- Good for opaque objects

- 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 Dynamics (CFD) – aerodynamic simulations
  - Meteorology – atmospheric pressure, temperature, wind speed, wind direction, humidity, precipitation
  - Astrophysics – simulate galaxies

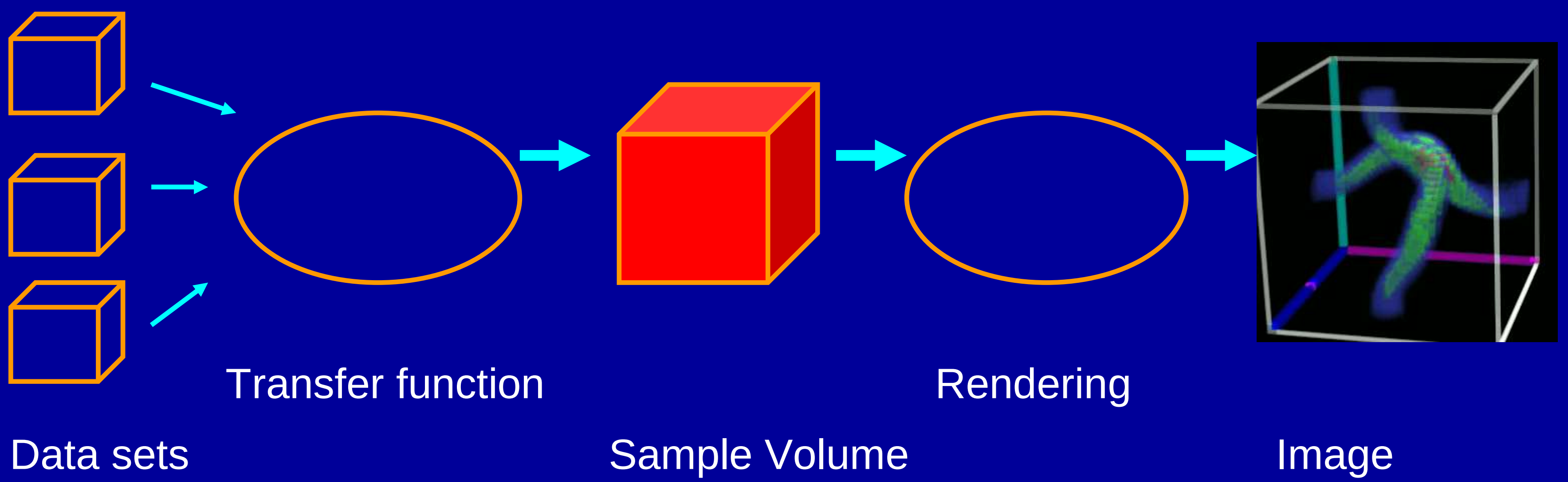
Simulate gravitational contraction of complex N-body systems



A computer simulation of high velocity air flow around the Space Shuttle.

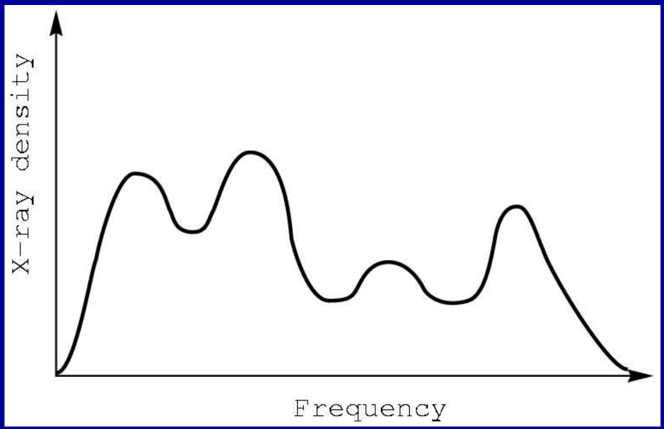
Volume Rendering Pipeline

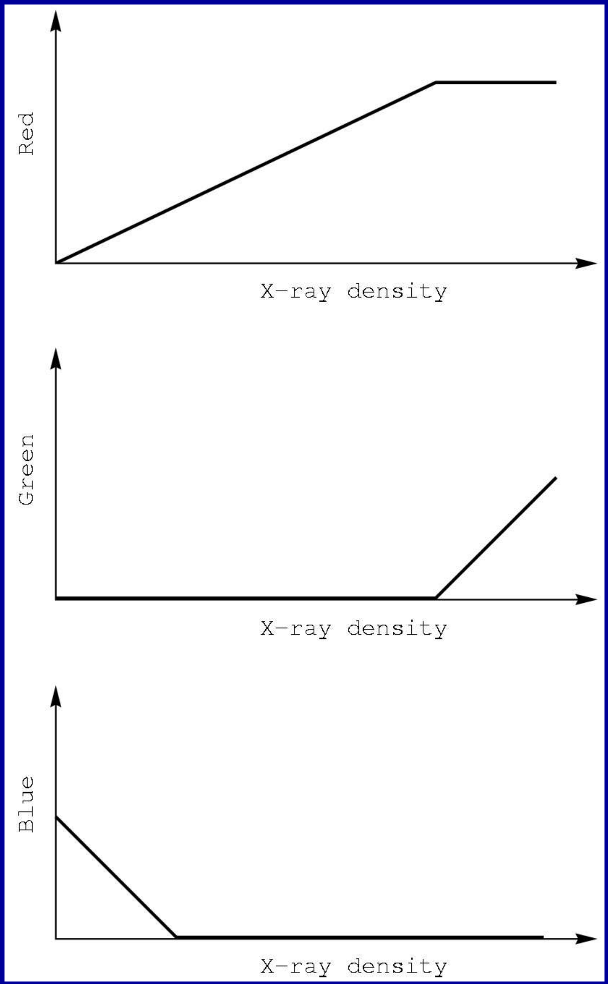
- Data volumes come in all types: tissue density (CT), wind speed, pressure, temperature, value of implicit function.
- Data volumes are used as input to a transfer function, which produces a sample volume of colors and opacities as output.
  – Typical might be a 256x256x64 CT scan
- That volume is rendered to produce a final image.



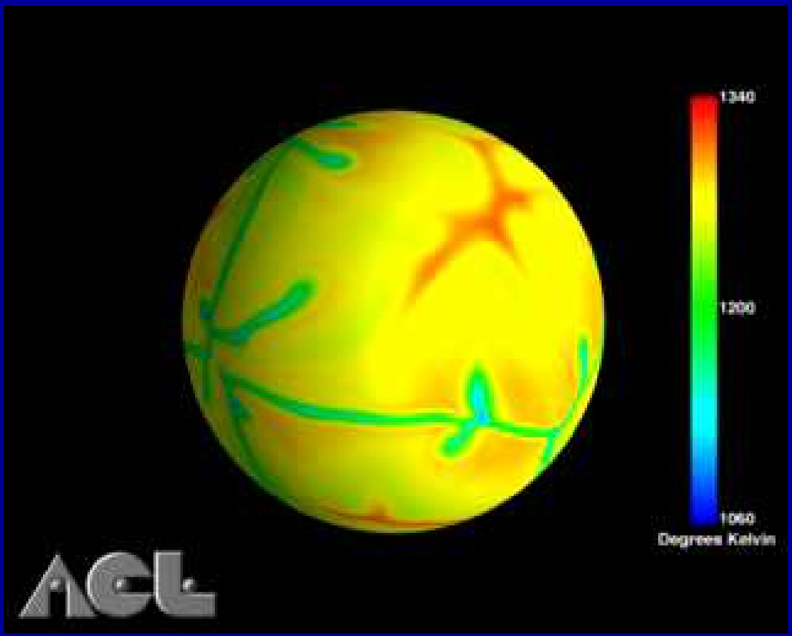
Transfer Functions

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



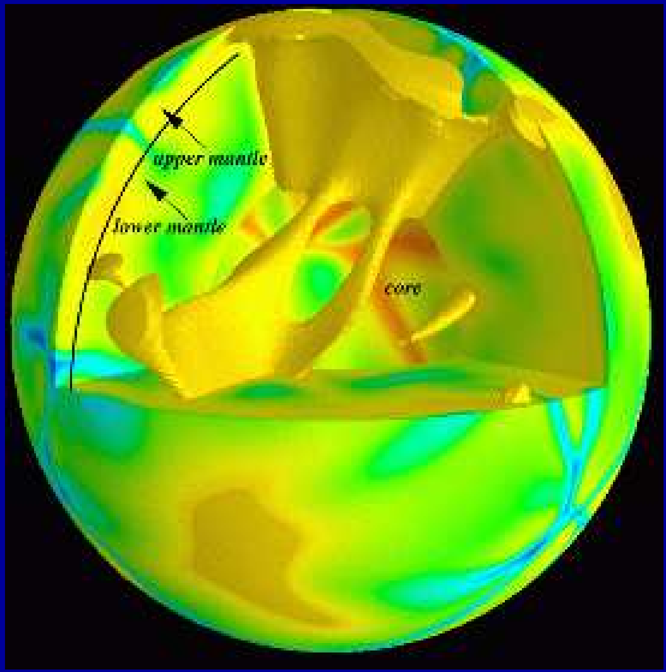


Transfer Function Example

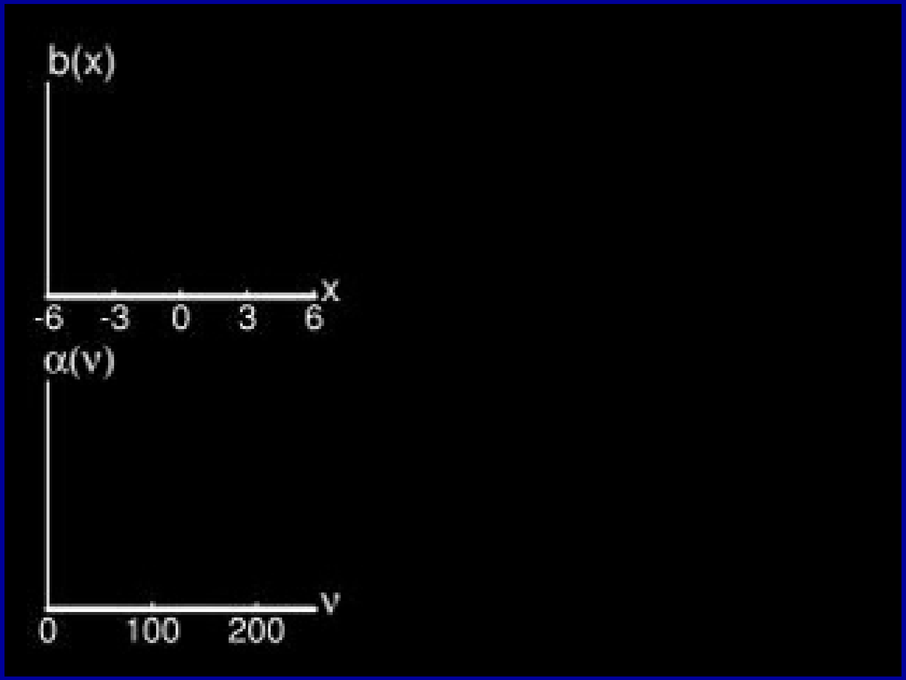


Scientific Computing and Imaging (SCI)
University of Utah

Mantle Convection



Transfer Function Example

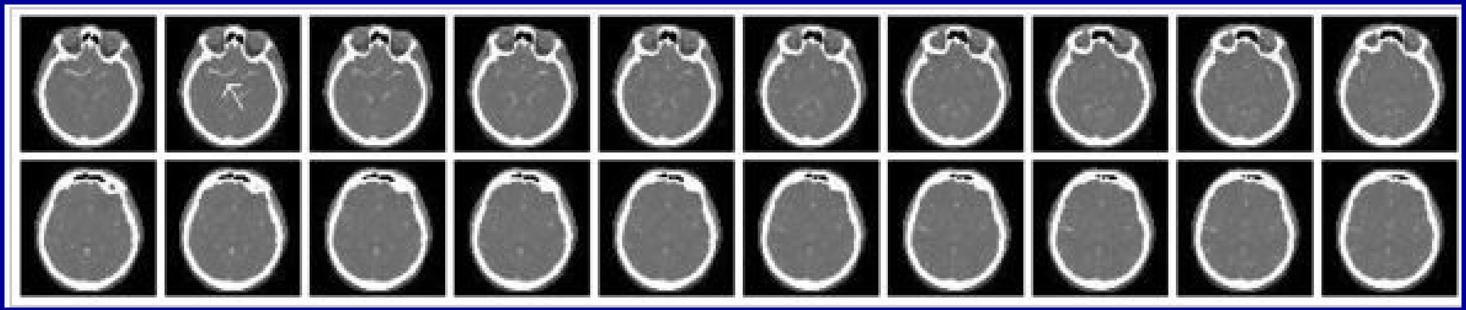


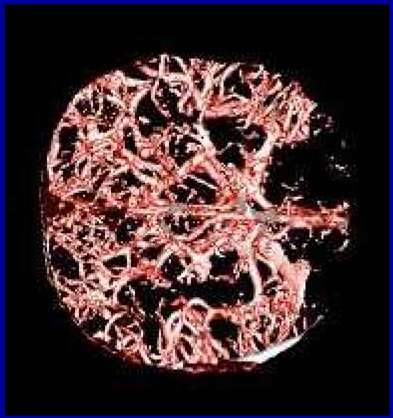
G. Kindlmann

Volume Rendering Pipeline

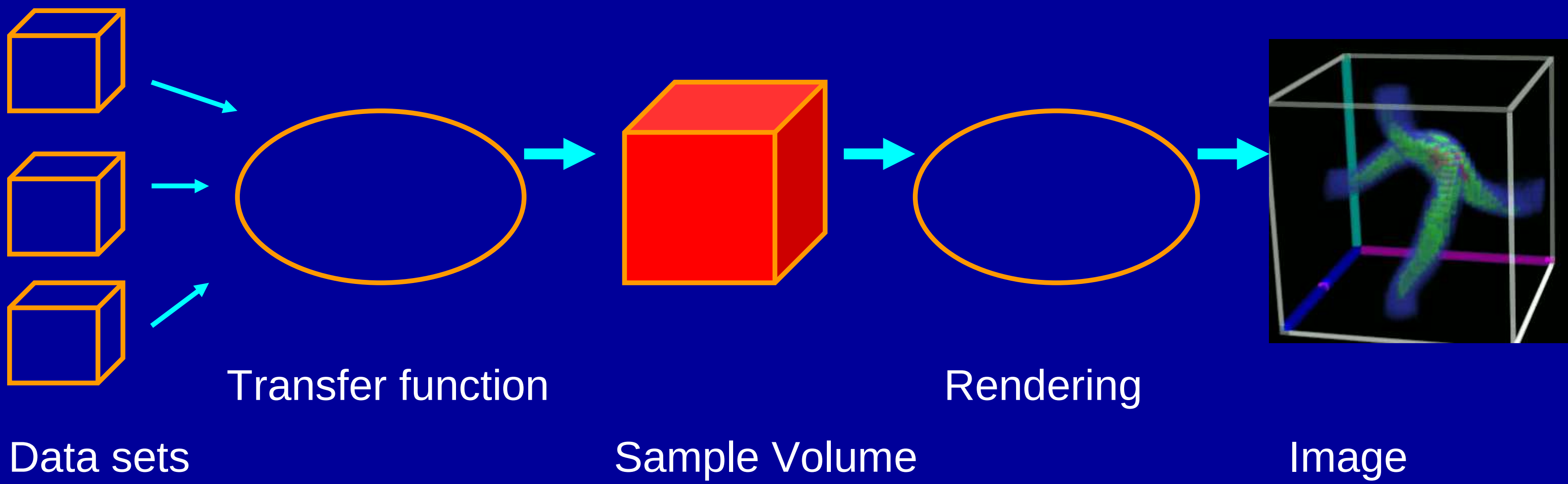
- Use opacity for emphasis

CT Scan - whiter means higher radiodensity





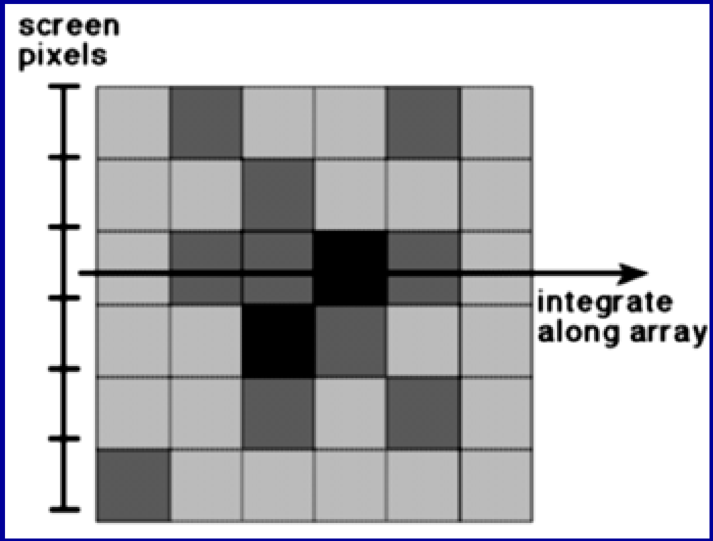
Volume Rendering



- Three volume rendering techniques
  - Volume ray casting
  - Splatting
  - 3D texture mapping

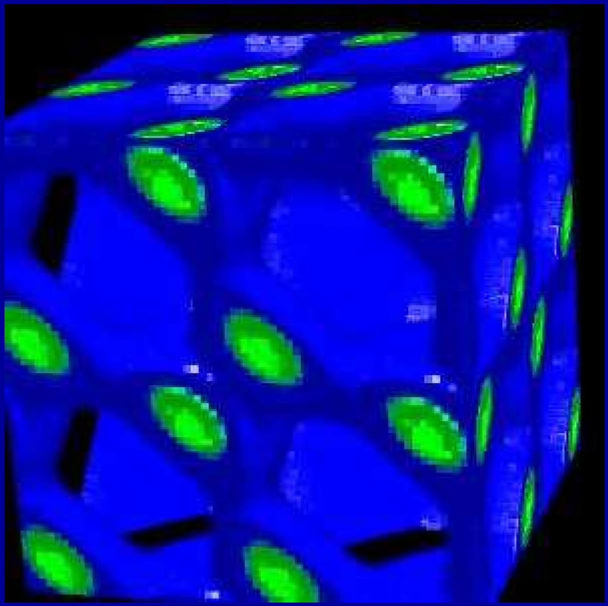
Volume Ray Casting

- Ray Casting
  - Integrate color and opacity along the ray
  - Simplest scheme just takes equal steps along ray, sampling opacity and color
  - Grids make it easy to find the next cell



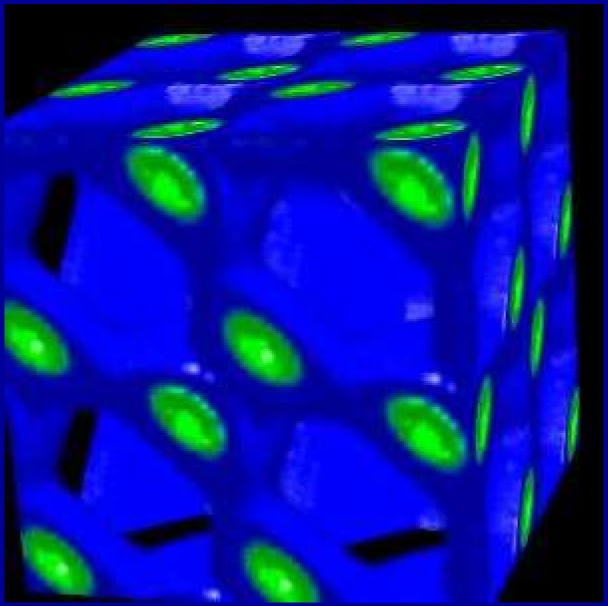
Trilinear Interpolation

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

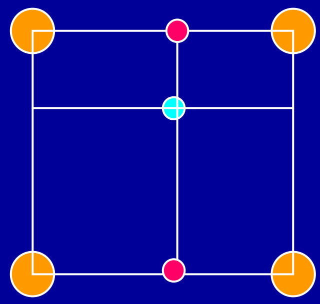


Nearest neighbor

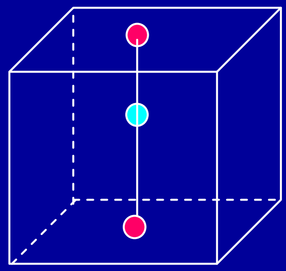
Trilinear interpolation



Trilinear Interpolation



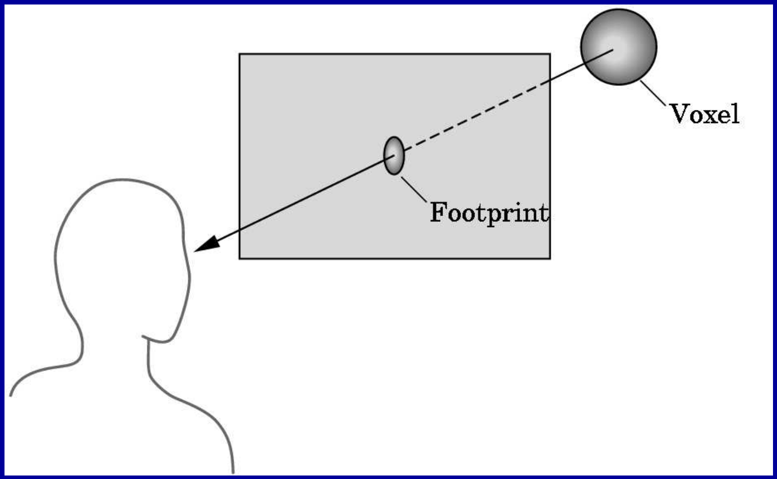
Bilinear interpolation

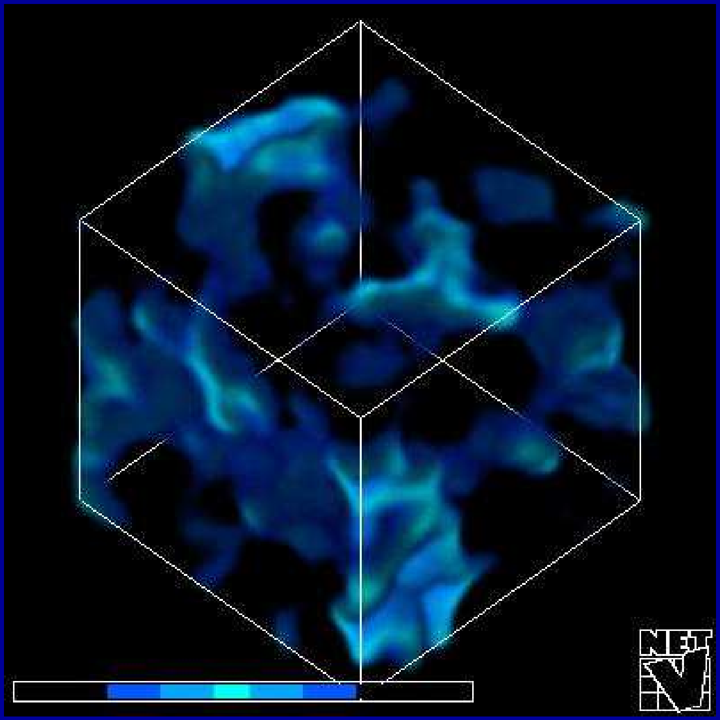


Trilinear interpolation

Splatting

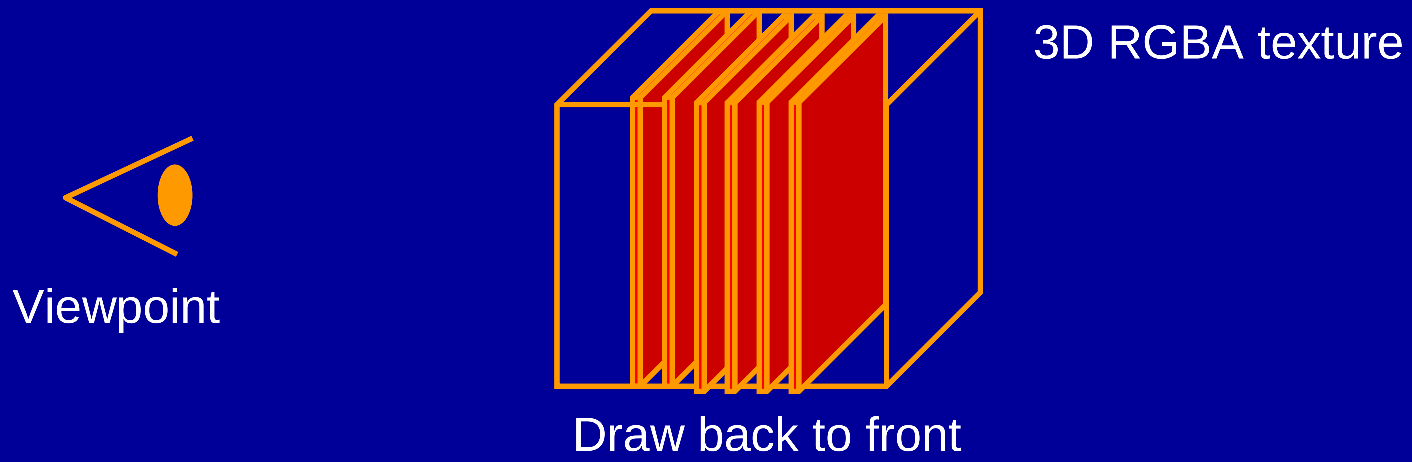
- Alternative to ray tracing
- Assign shape to each voxel (e.g., sphere or Gaussian)
- Project onto image plane (splat)
- Draw voxels back-to-front
- Composite (a-blend)





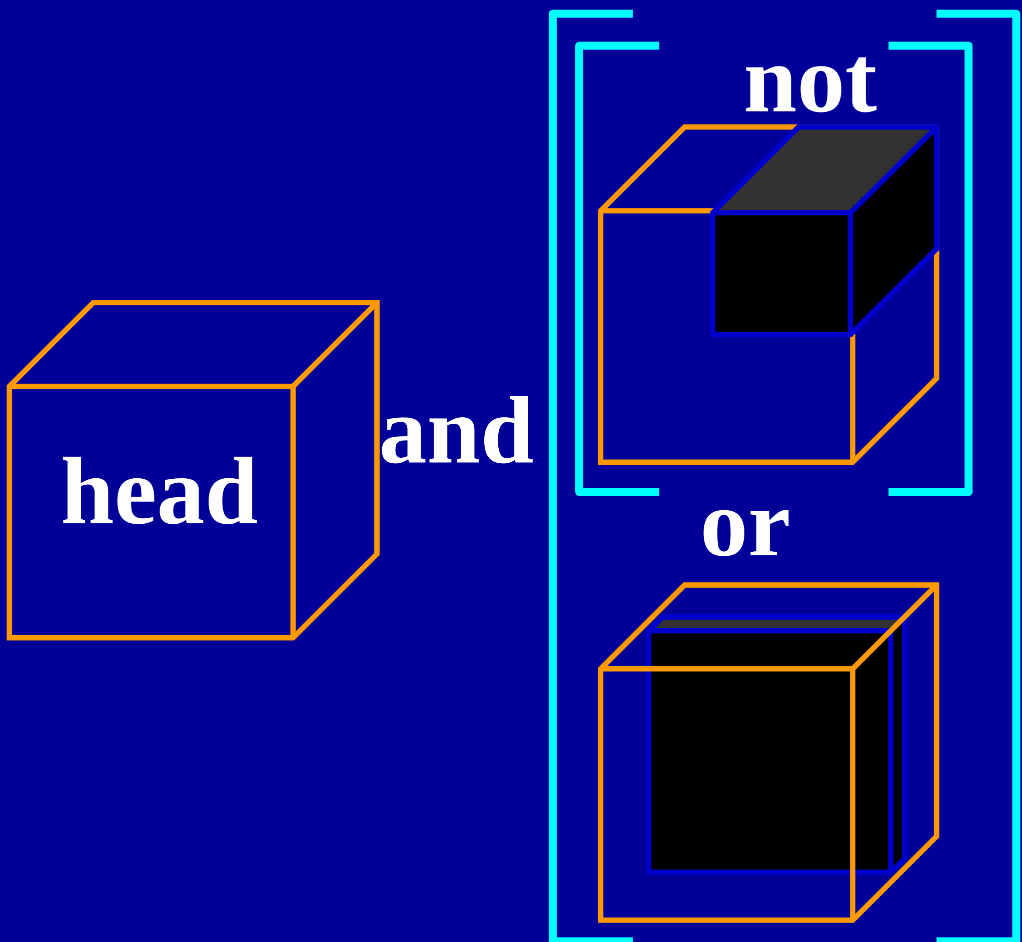
3D Textures

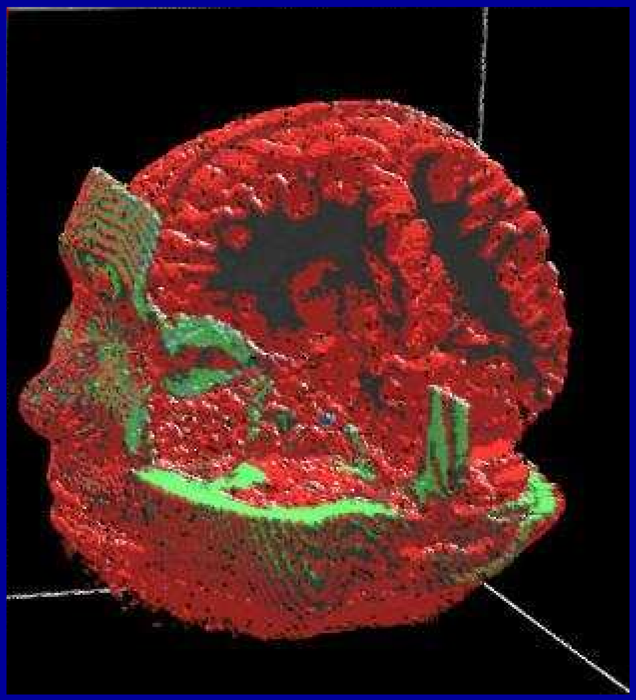
- Alternative to ray tracing, splatting
- Build a 3D texture (including opacity)
- Draw a stack of polygons, back-to-front
- Efficient if supported in graphics hardware
- Few polygons, much texture memory



Other Techniques

- Use CSG for cut-away





Acceleration of Volume Rendering

- Basic problem: Huge data sets
- Octrees
- Use error measures to stop iteration
- Exploit parallelism

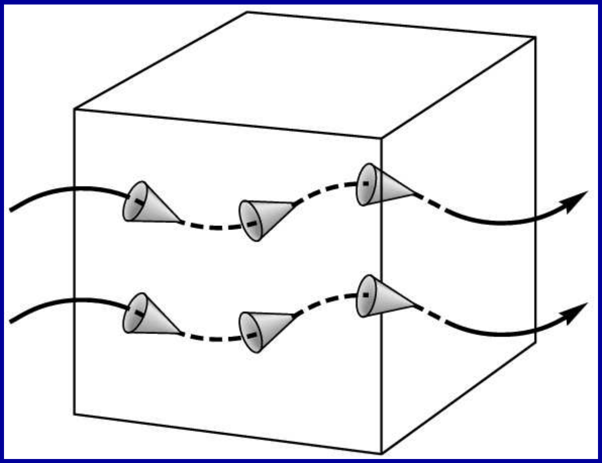
Outline

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

Vector Fields

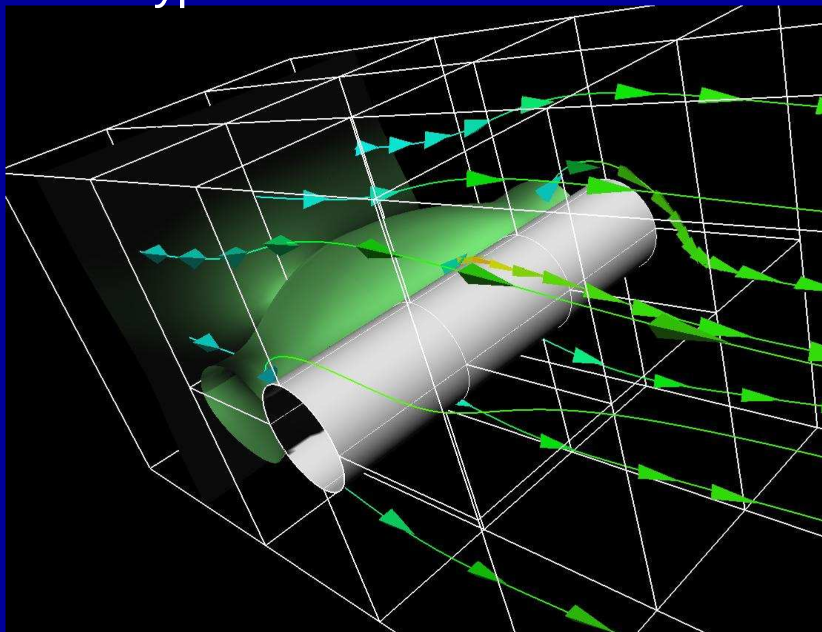
- Visualize vector at each (x,y,z) point
  - Example: velocity field
- Hedgehogs
  - Use 3D directed line segments (sample field)
  - Orientation and magnitude determined by vector
- Glyph
  - Use other geometric primitives
  - Cones

Blood flow in
human carotid artery

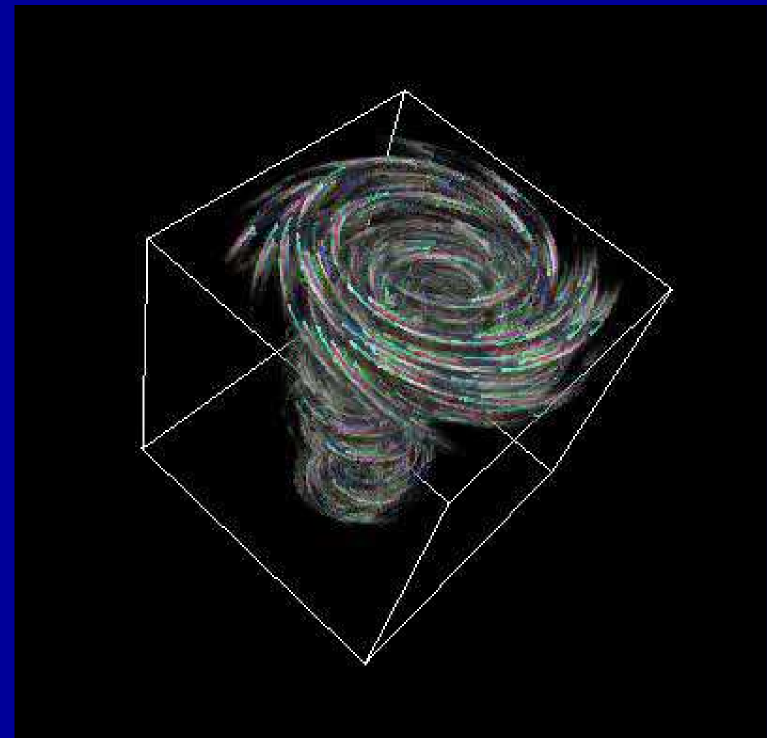


Vector Fields (Utah)

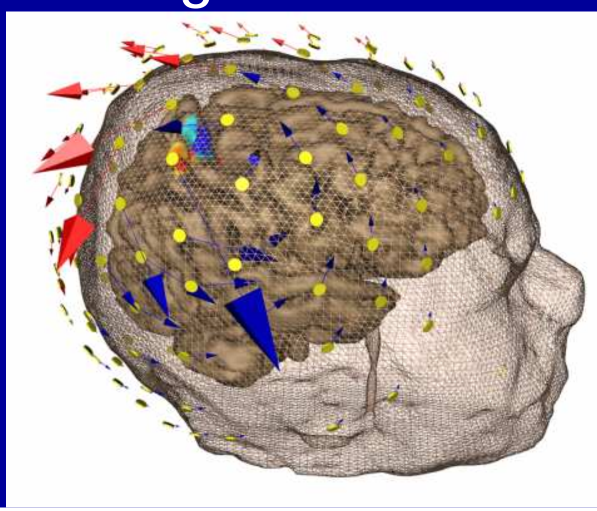
Glyphs for air flow



Tornado

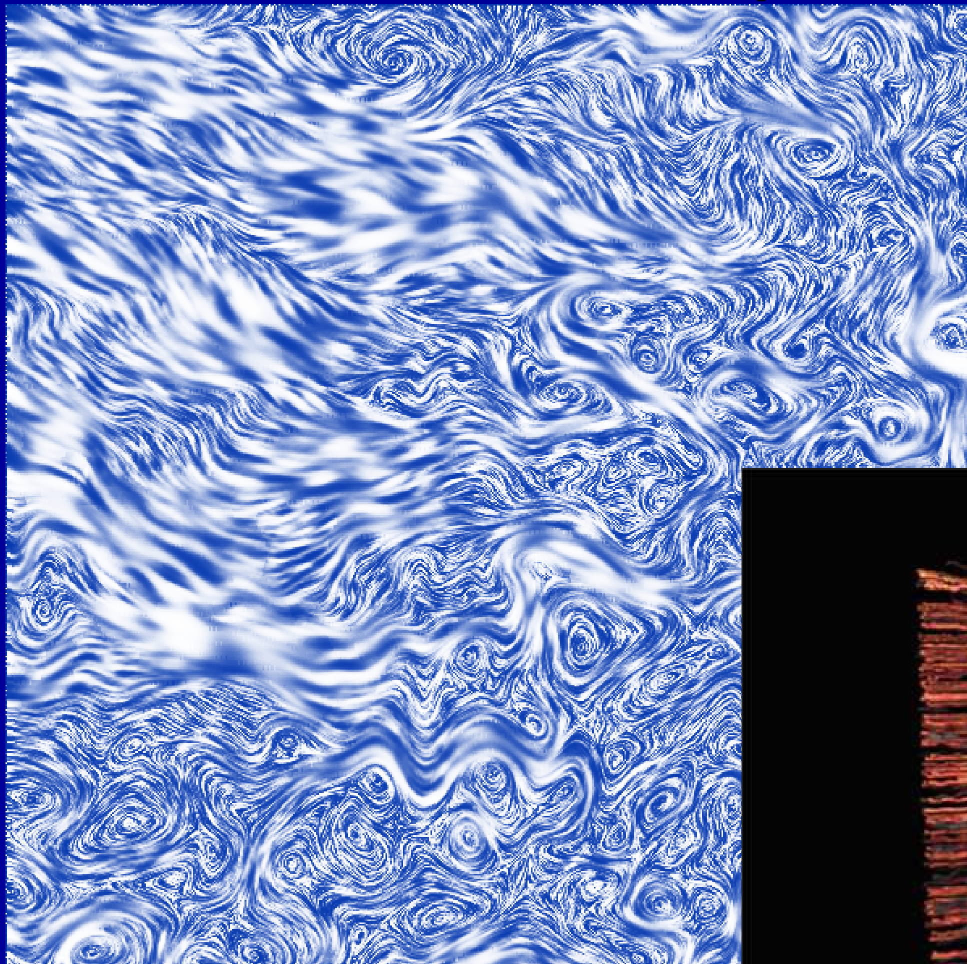


Magnetic field

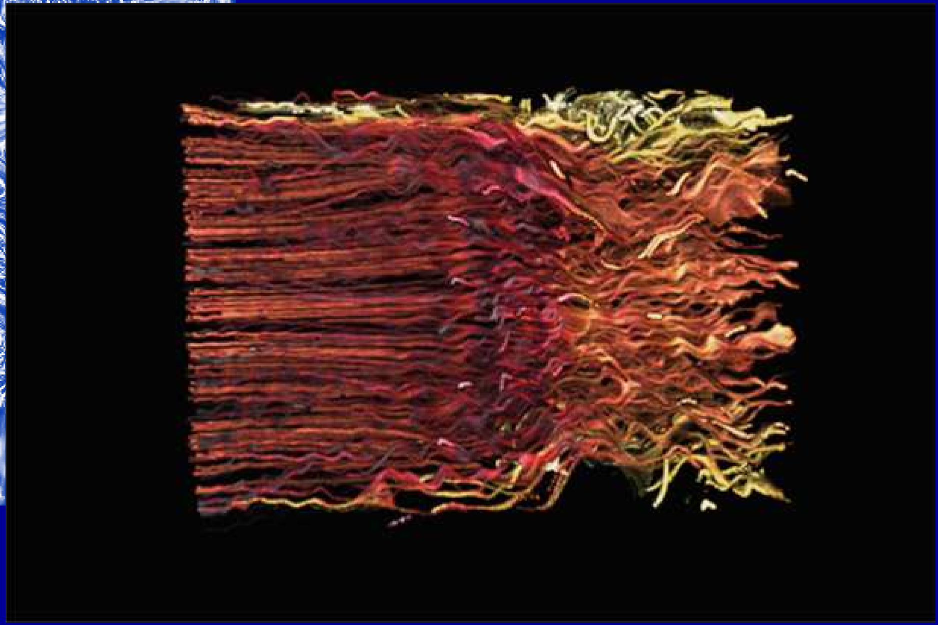


Plasma disruption

More Flow Examples

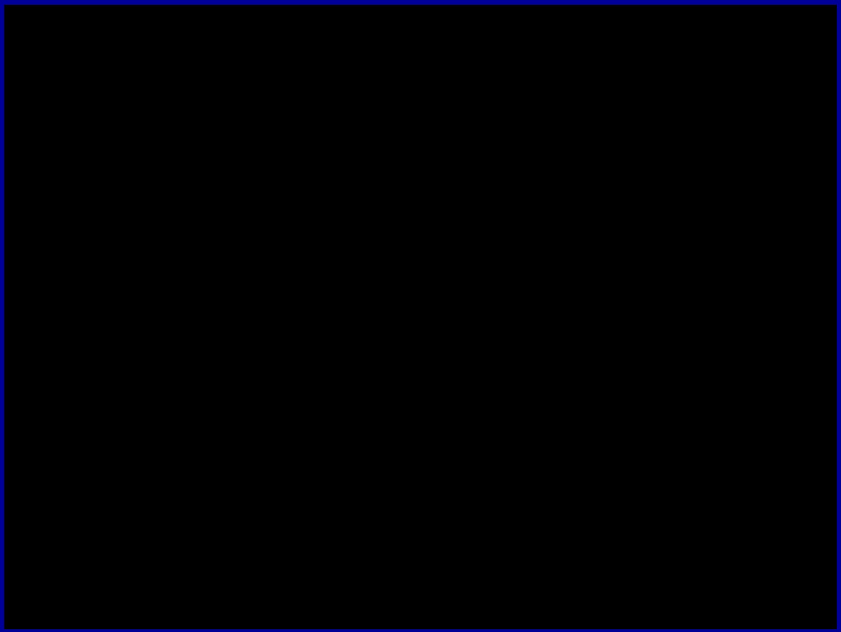


Banks and Interrante

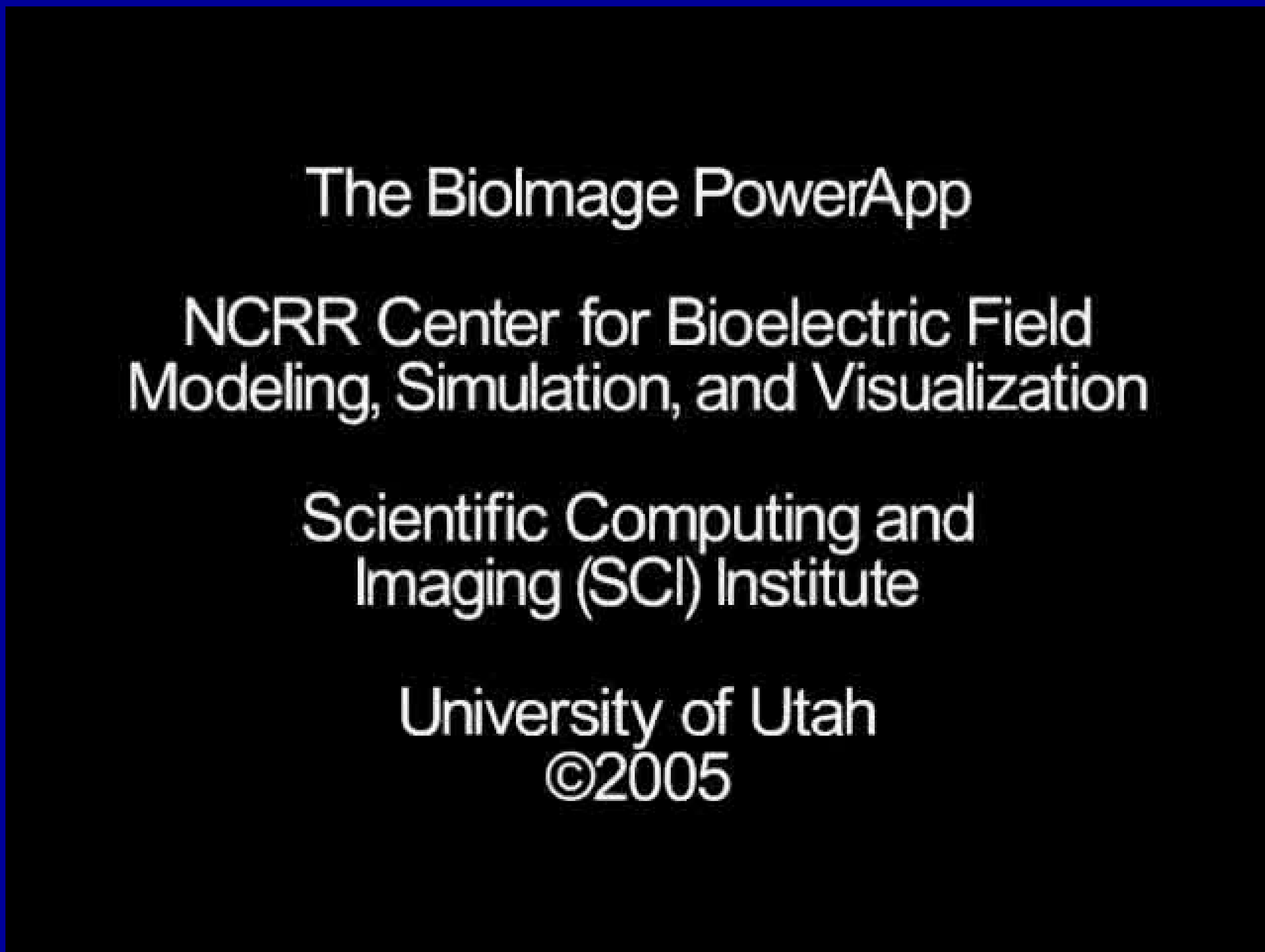


Interaction: Data Probe

SCI, Utah



Example of visualization application



University of Utah

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

Summary

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

Announcements

- Course Evaluation is now open
- Until Monday, May 7th
- Please complete the evaluation

- We read it and listen to what you say