• volume rendering
• color basics (start)
Volume rendering

- Volume data:
  3D grid of voxels

- Goal: Look inside the data

- Source: scanners (MRI, CT, CAT), 3D-seismic, simulations (report topic)

- Each voxel has a value (e.g., density)

- Show density only via color - no geometry!
• Alternative: isosurfaces

• Problems:
  • Need to find good iso-value ("segmentation")
  • even with transparent isosurfaces, it’s difficult to stack them

• Advantages:
  • fast (hardware rendering)
  • ability to using triangle shading
Direct Volume Rendering

- Every voxel values to be mapped to an opacity and a color (RGBα).
- Voxels are composited (back-to-front) into in image (viewpoint dependent, may include light)
transfer function:

- converts voxel value ("density", x-axis) to an RGBA value (y-axis)
- Here: 4 different functions
- Simple ramp, a piecewise linear function or an arbitrary table.
- allows “fuzzy” boundaries
(a) Direct volume rendered  
(b) Isosurface rendered
• time ?
Color and color mapping

• Color specification in DX?

• Color names (X11 system):
  "red", "black", "limegreen“, “navy blue”, “medium goldenrod”

• On monitor: RGB model

• think: combining colored lights

• (additive model) mixture of primary colors:
  Red, Green, Blue
• no light: black, “full light”: white
• full red + full green = yellow
• Each channel (R,G,B) ranges from 0.0 to 1.0 (DX):
• $[1.0 \ 0.0 \ 1.0] = \text{yellow}$
• typically: each channel (band) uses 8 bits (ints)
• also: 16 bits, 32 bits
• floating point numbers
• 256 different “shades” of each primary color
• 256 x 256 x 256 colors = 16.78 M
• Extra channel for transparency (reverse: opacity)
• RGBα: Opacity (alpha-value) 0-100%
  0% = fully transparent, DX: 0.0 to 1.0
Wrap-up

- Midterm report topics?
- Lab: finish ch. 4 + exercises (HW)