University of British Columbia
CPSC 314 Computer Graphics
Jan-Apr 2007

Tamara Munzner

Visualization

Week 11, Fri Mar 30

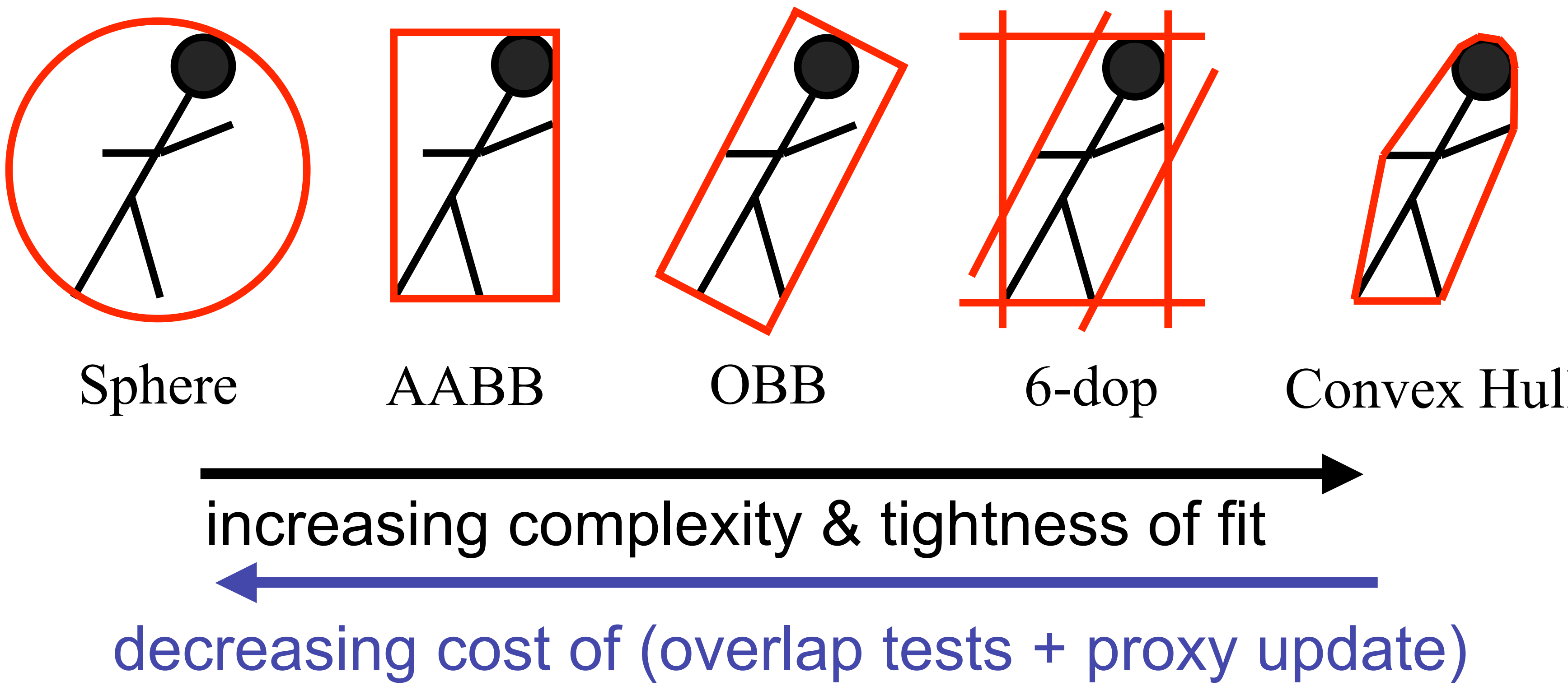
http://www.ugrad.cs.ubc.ca/~cs314/Vjan2007

News

- extra TA office hours in lab for hw/project Q&A
  - next week: Thu 4-6, Fri 10-2
  - last week of classes:
    - Mon 2-5, Tue 4-6, Wed 2-4, Thu 4-6, Fri 9-6
- final review Q&A session
  - Mon Apr 16 10-12
- reminder: no lecture/labs Fri 4/6, Mon 4/9

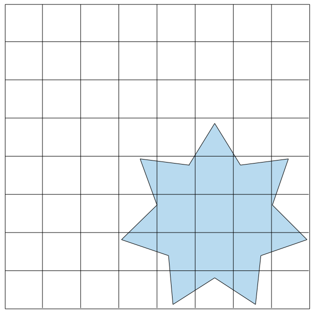
Review: Collision Proxy Tradeoffs

- collision proxy (bounding volume) is piece of geometry used to represent complex object for purposes of finding collision
- proxies exploit facts about human perception
  - we are bad at determining collision correctness
  - especially many things happening quickly

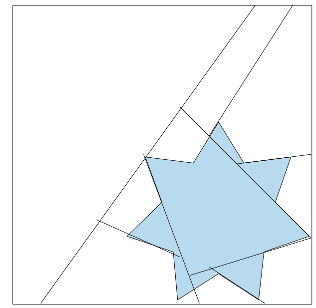


Review: Spatial Data Structures

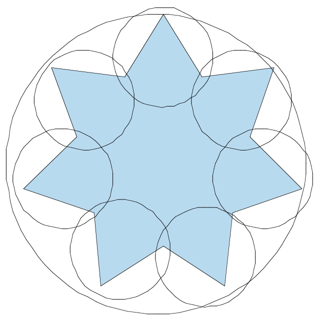
uniform grids



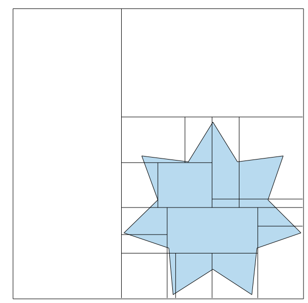
BSP trees



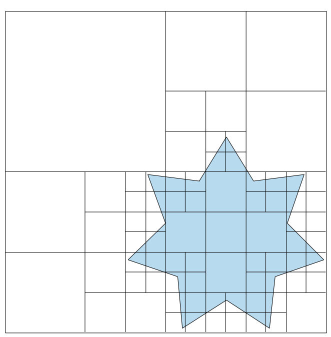
bounding volume hierarchies



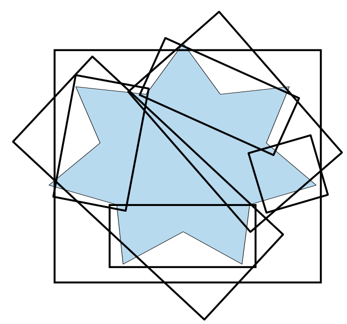
kd-trees



octrees



OBB trees

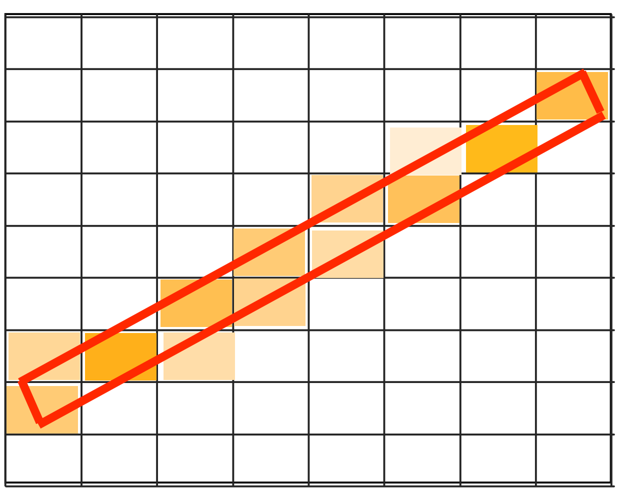


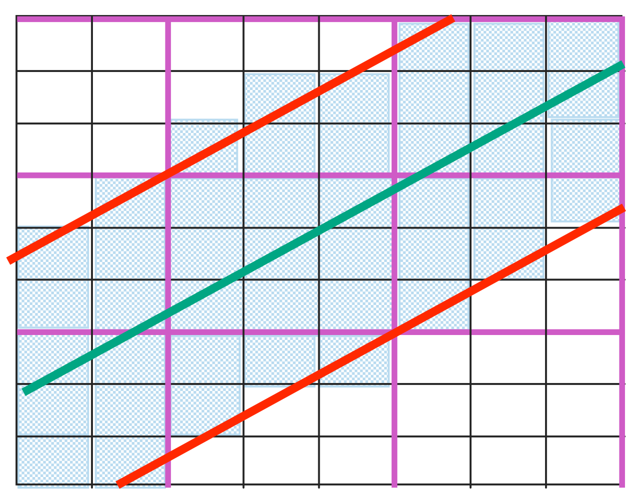
Review: Aliasing

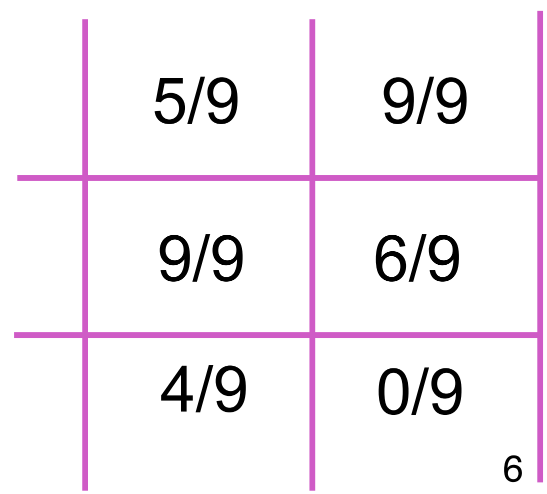
- incorrect appearance of high frequencies as low frequencies
- to avoid: antialiasing
  - supersample
    - sample at higher frequency
  - low pass filtering
    - remove high frequency function parts
    - aka prefiltering, band-limiting

Review: Supersample and Average

- supersample: create image at higher resolution
  - e.g. 768x768 instead of 256x256
  - shade pixels wrt area covered by thick line/rectangle
- average across many pixels
  - e.g. 3x3 small pixel block to find value for 1 big pixel
  - rough approximation divides each pixel into a finer grid of pixels

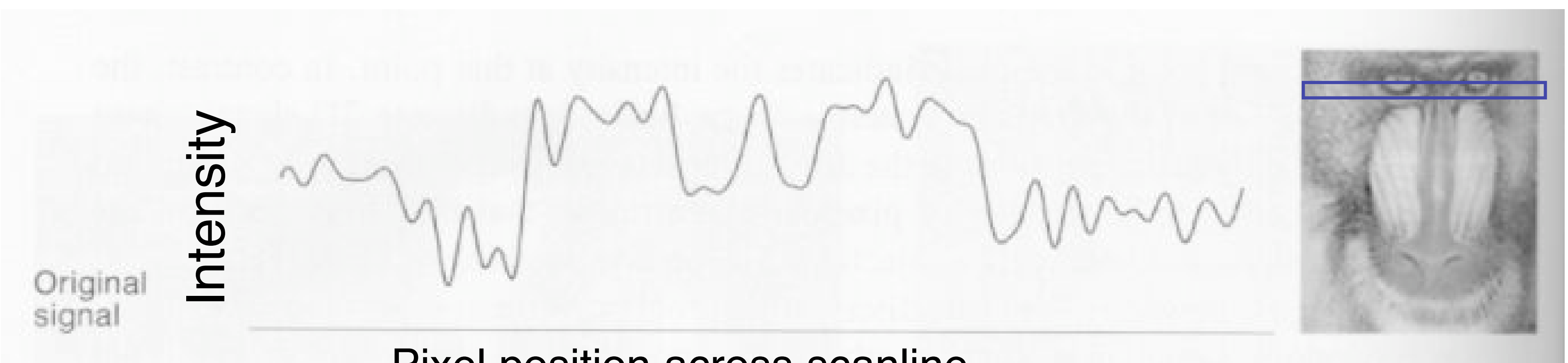






Review: Image As Signal

- 1D slice of raster image
  - discrete sampling of 1D spatial signal
- theorem
  - any signal can be represented as an (infinite) sum of sine waves at different frequencies



Examples from Foley, van Dam, Feiner, and Hughes

Review: Sampling Theorem and Nyquist Rate

- Shannon Sampling Theorem
  - continuous signal can be completely recovered from its samples iff sampling rate greater than twice maximum frequency present in signal
- sample past Nyquist Rate to avoid aliasing
  - twice the highest frequency component in the image's spectrum

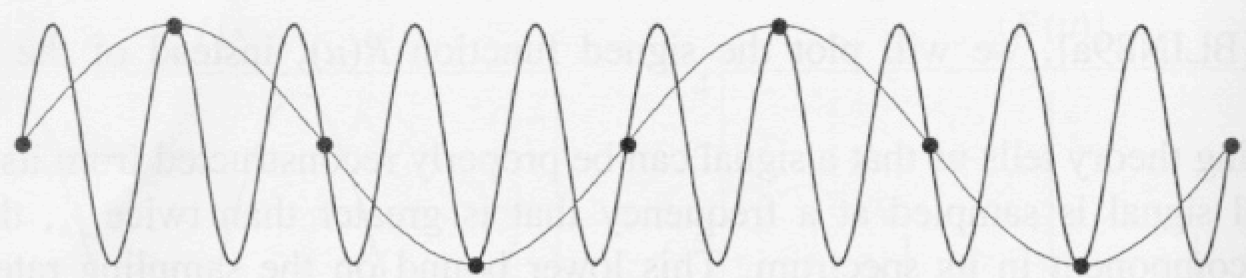
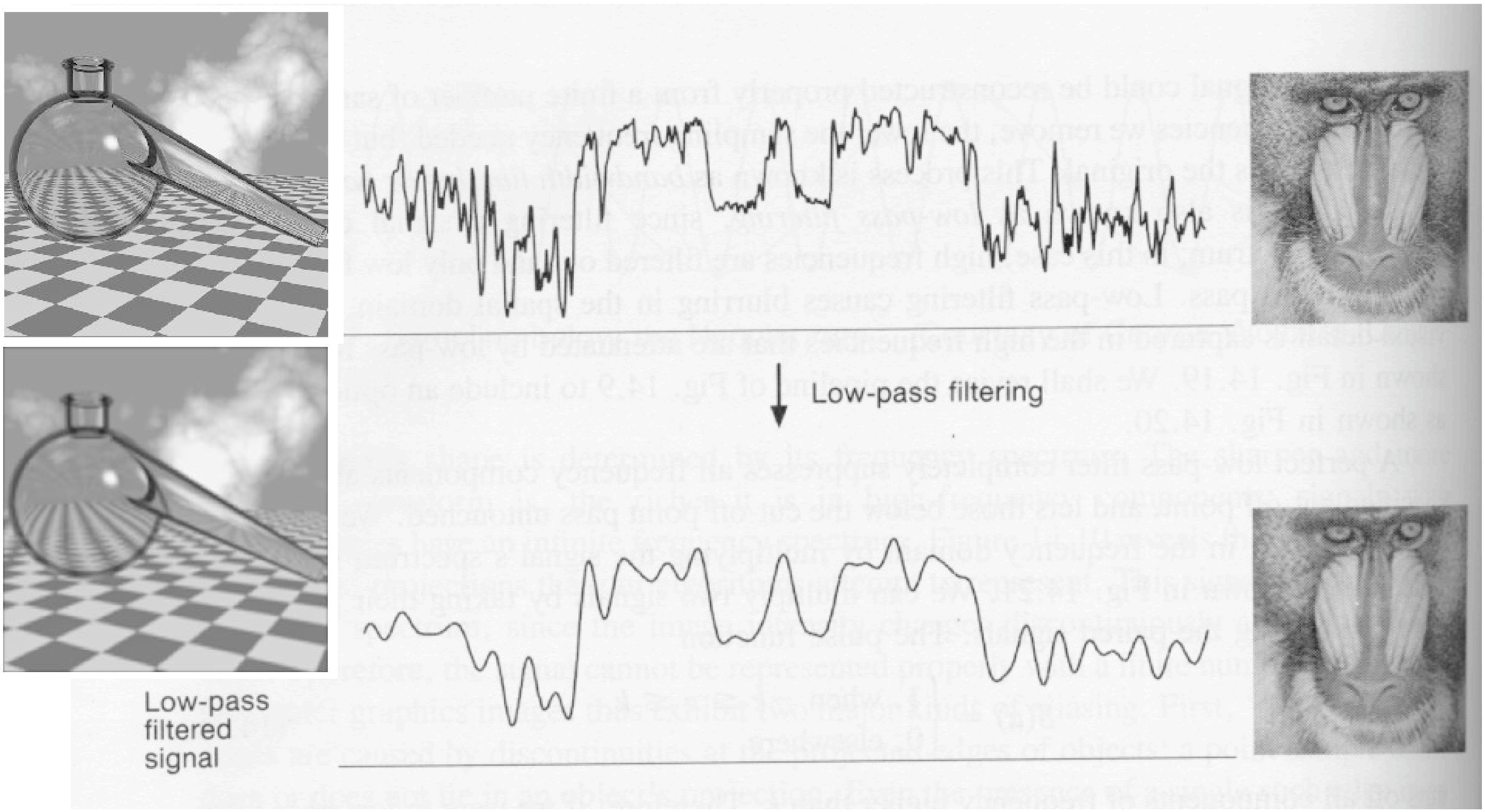


Fig. 14.17 Sampling below the Nyquist rate. (Courtesy of George Wolberg, Columbia University.)

Review: Low-Pass Filtering



Scientific Visualization

Reading

- FCG Chapter 23

Surface Graphics

- objects explicitly defined by surface or boundary representation
  - mesh of polygons

200 polys **1000 polys** **15000 polys**

Surface Graphics

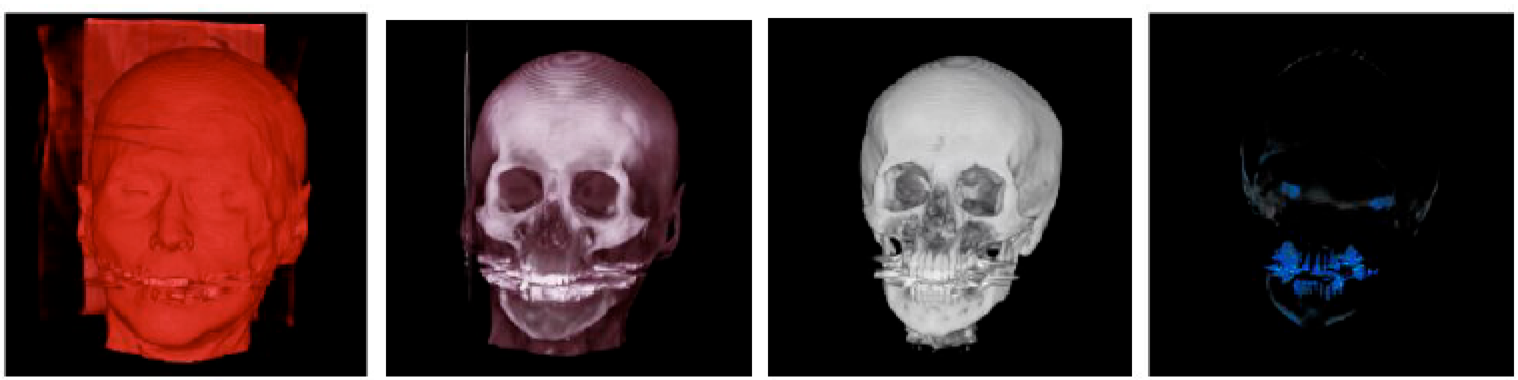
- pros
  - fast rendering algorithms available
  - hardware acceleration cheap
  - OpenGL API for programming
  - use texture mapping for added realism
- cons
  - discards interior of object, maintaining only the shell
  - operations such cutting, slicing & dissection not possible
  - no artificial viewing modes such as semi-transparencies, X-ray
  - surface-less phenomena such as clouds, fog & gas are hard to model and represent

Volume Graphics

- for some data, difficult to create polygonal mesh
- voxels: discrete representation of 3D object
  - volume rendering: create 2D image from 3D object
- translate raw densities into colors and transparencies
  - different aspects of the dataset can be emphasized via changes in transfer functions

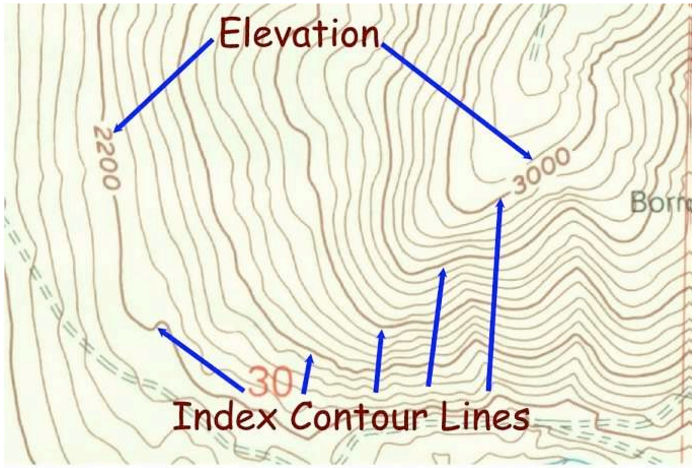
Volume Graphics

- pros
  - formidable technique for data exploration
- cons
  - rendering algorithm has high complexity!
  - special purpose hardware costly (~$3K-$10K)

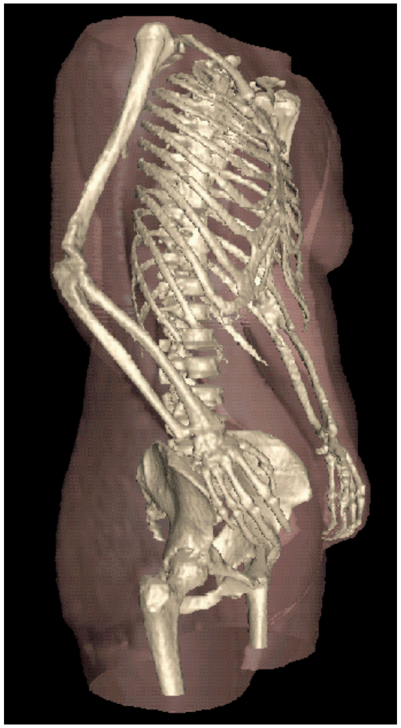


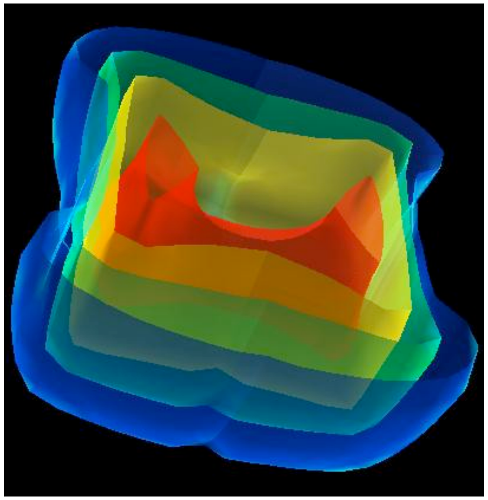
volumetric human head (CT scan)

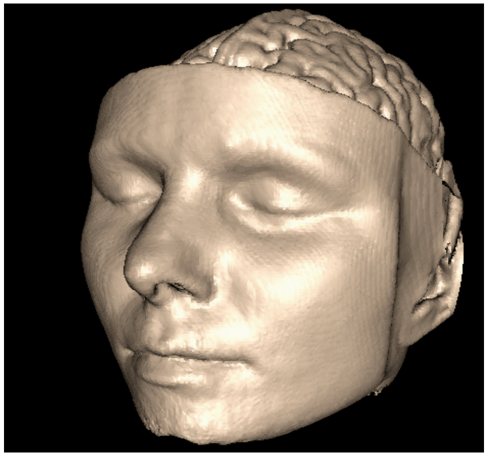
Isosurfaces



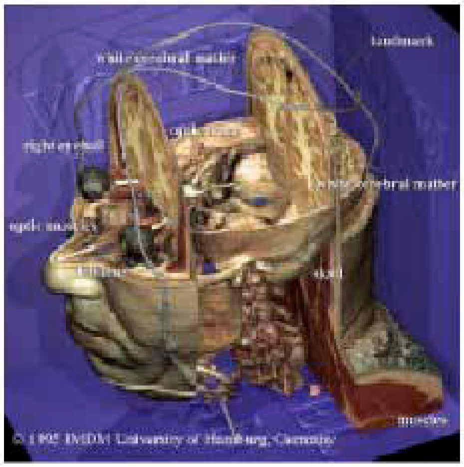
- 2D scalar fields: isolines
  - contour plots, level sets
  - topographic maps
- 3D scalar fields: isosurfaces



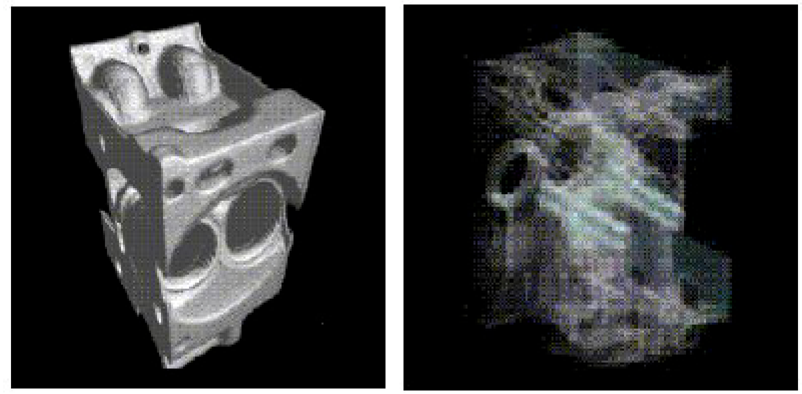




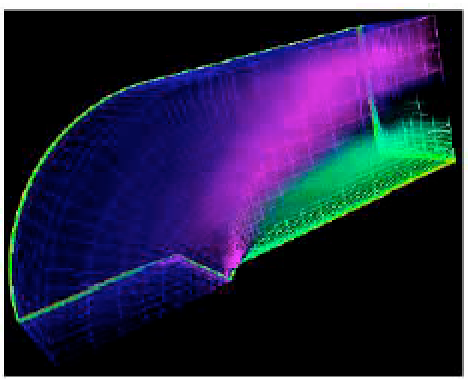
Volume Graphics: Examples



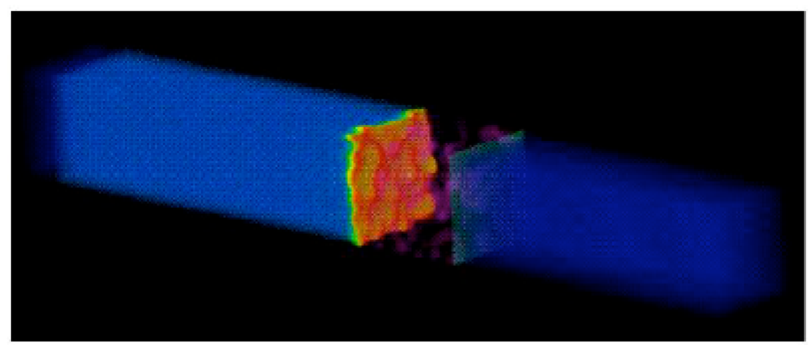
anatomical atlas from visible human (CT & MRI) datasets



industrial CT - structural failure, security applications



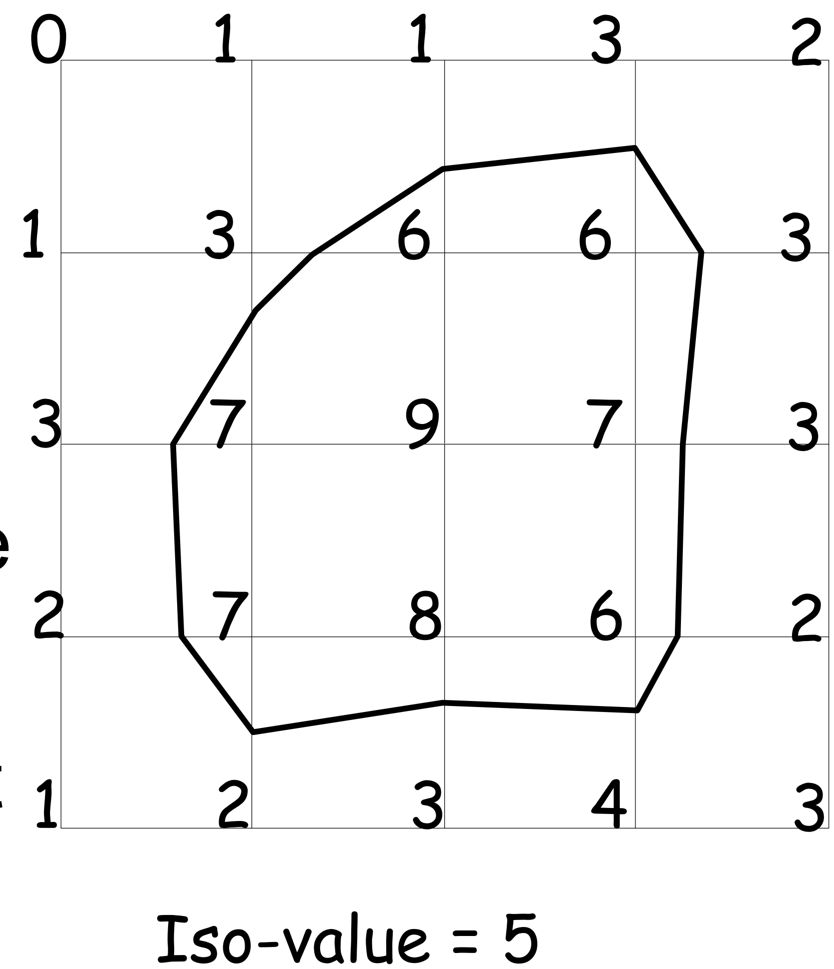
flow around airplane wing



shockwave visualization: simulation with Navier-Stokes PDEs

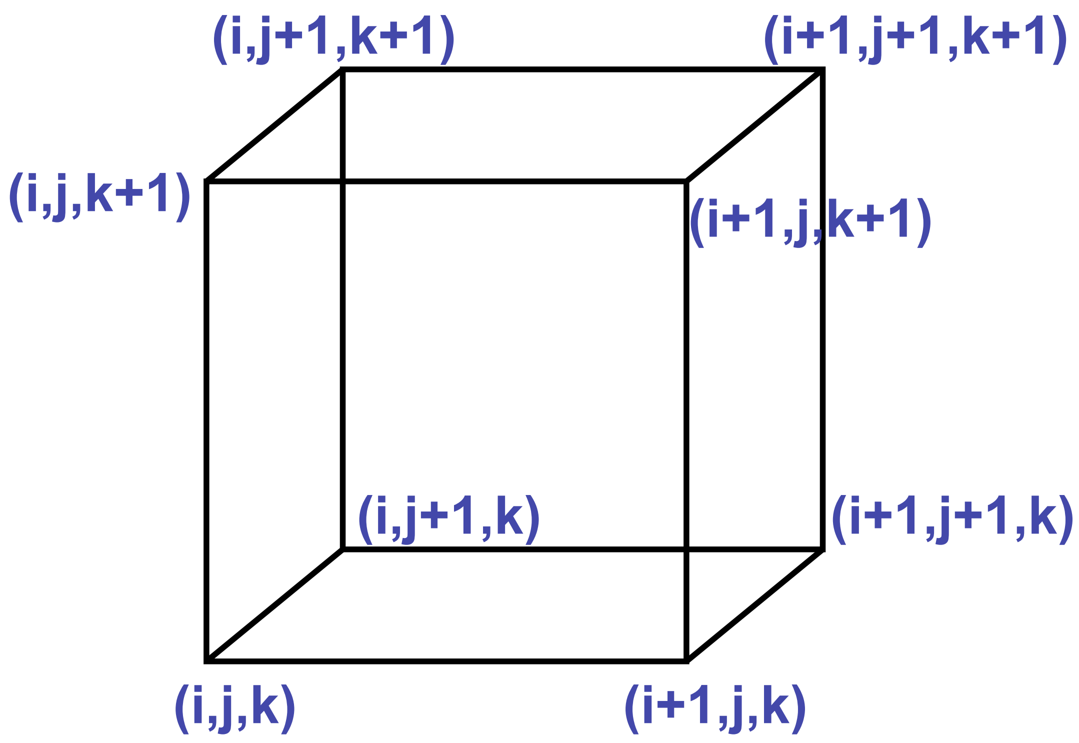
Isosurface Extraction

- array of discrete point samples at grid points
  - 3D array: voxels
- find contours
  - closed, continuous
  - determined by iso-value
- several methods
  - marching cubes is most common



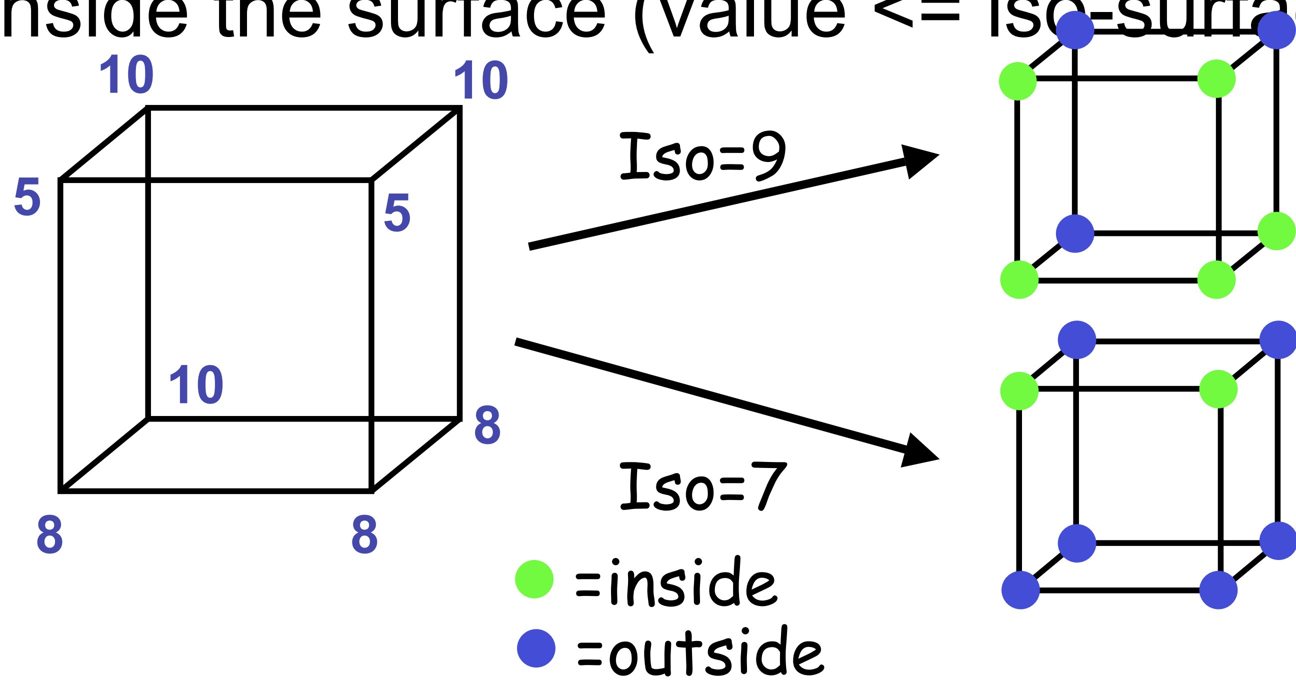
MC 1: Create a Cube

- consider a cube defined by eight data values



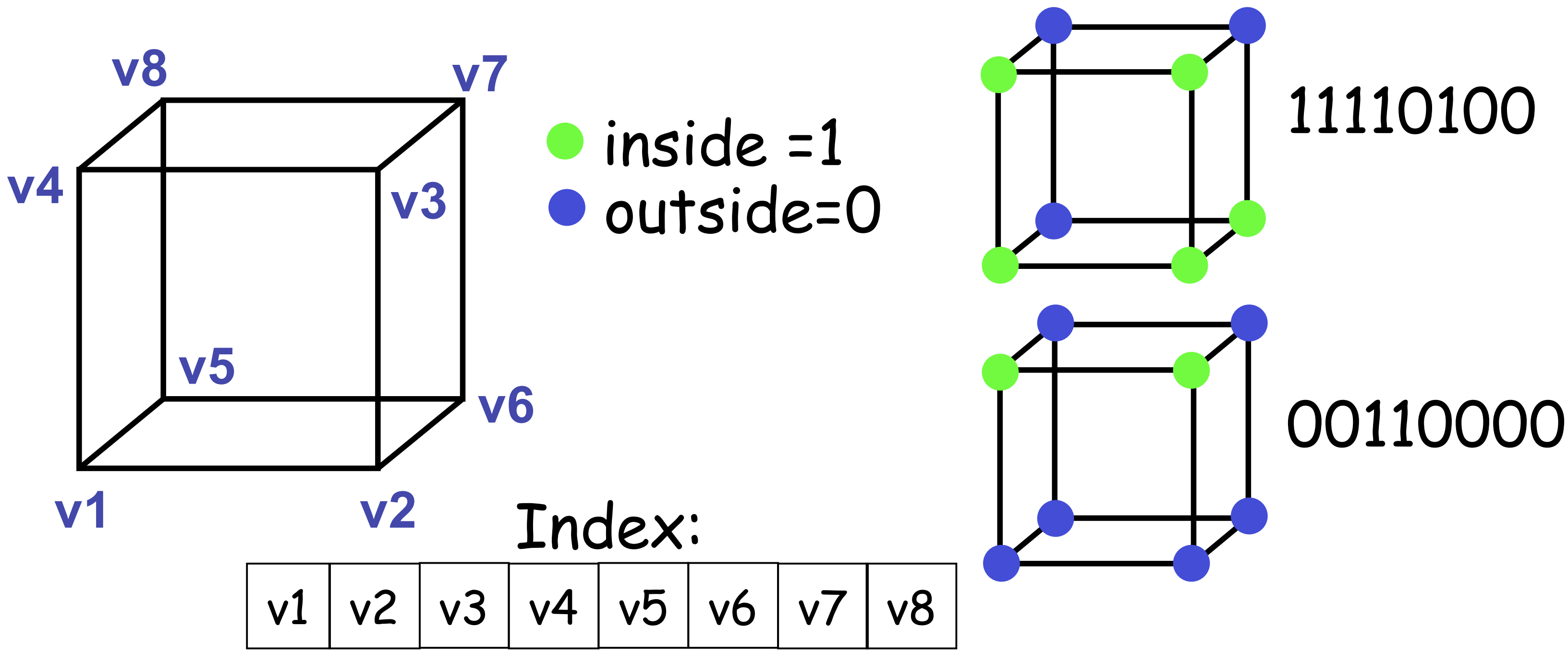
MC 2: Classify Each Voxel

- classify each voxel according to whether lies
  - outside the surface (value > iso-surface value)
  - inside the surface (value <= iso-surface value)



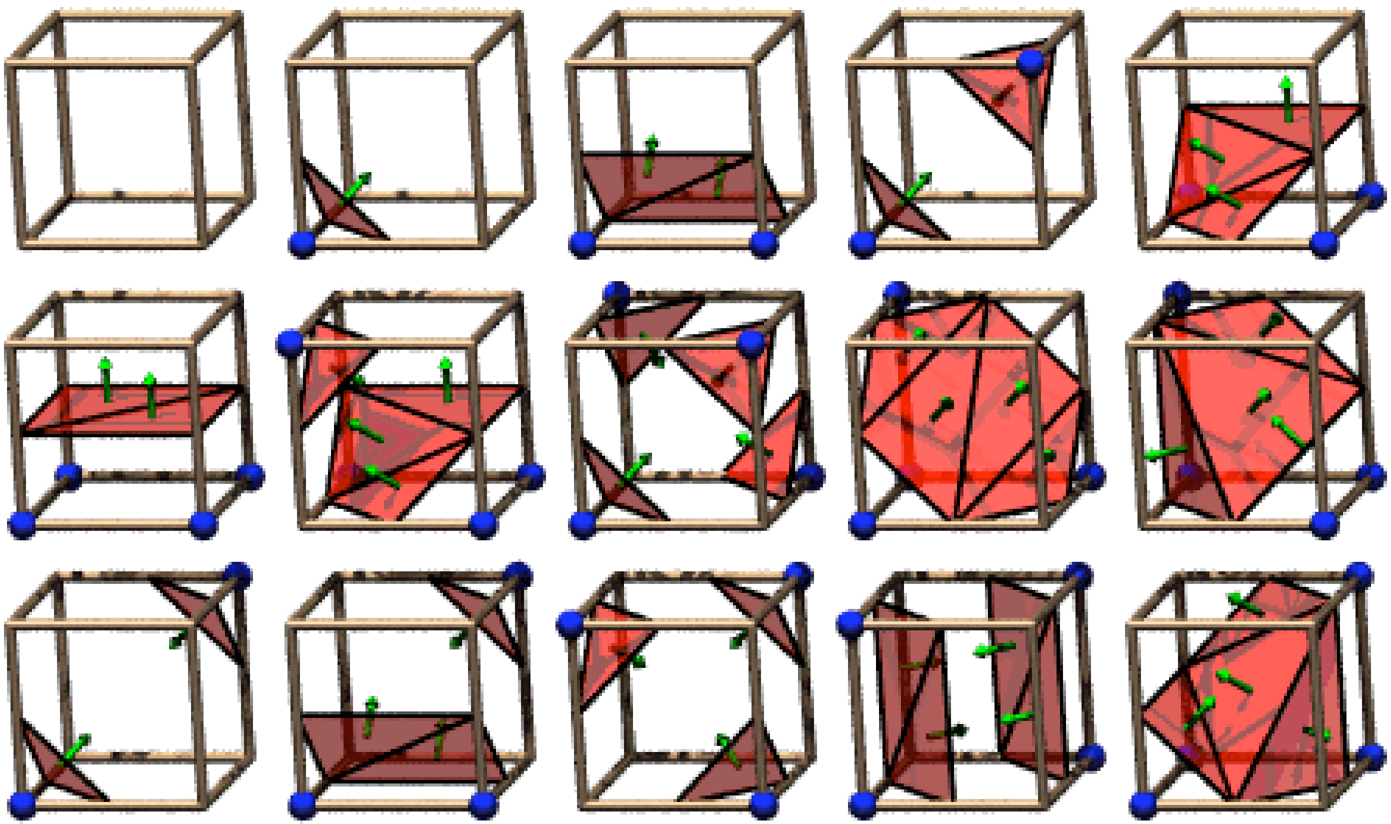
MC 3: Build An Index

- binary labeling of each voxel to create index



MC 4: Lookup Edge List

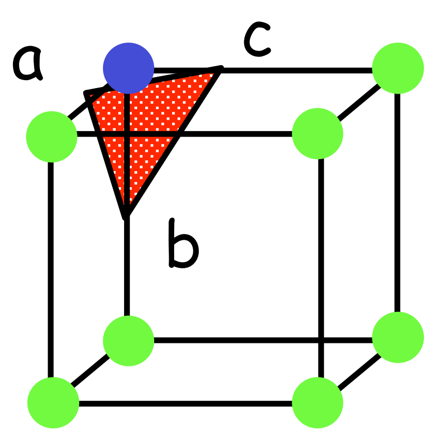
- use index to access array storing list of edges
  - all 256 cases can be derived from 15 base cases



The 15 Cube Combinations

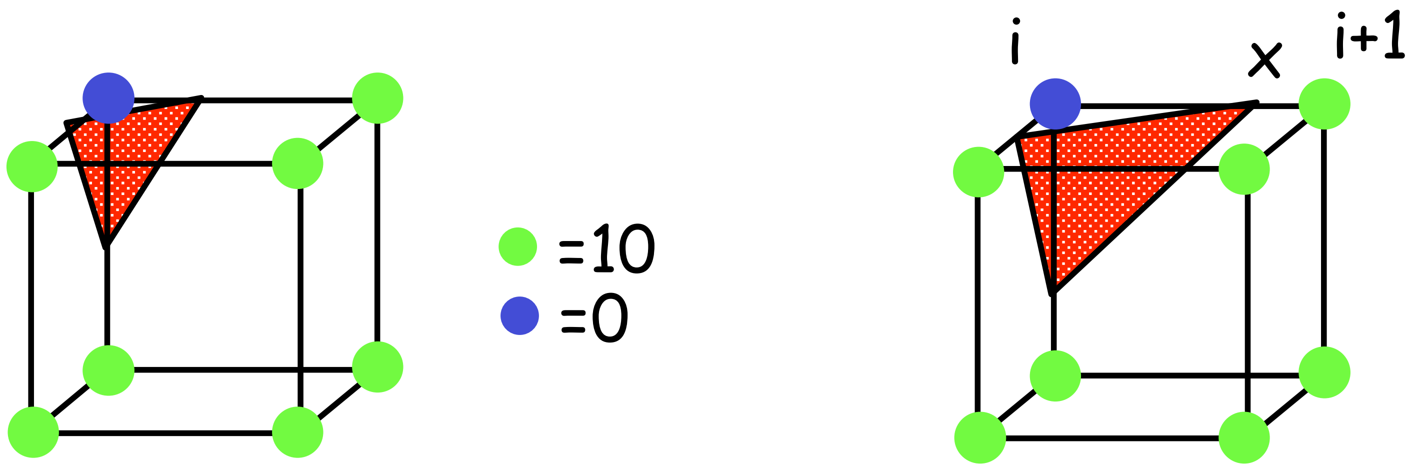
MC 4: Example

- index = 00000001
- triangle 1 = a, b, c



MC 5: Interpolate Triangle Vertex

- for each triangle edge
  - find vertex location along edge using linear interpolation of voxel values



T=5

$$x = i + \left(\frac{T - v[i]}{v[i+1] - v[i]} \right)$$

T=8

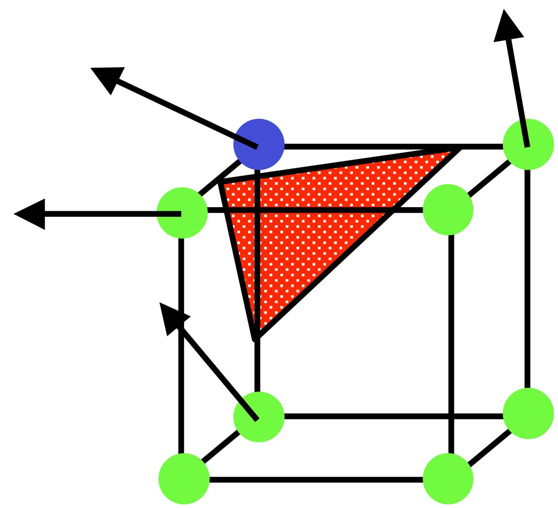
MC 6: Compute Normals

- calculate the normal at each cube vertex
  - use linear interpolation to compute the polygon vertex normal

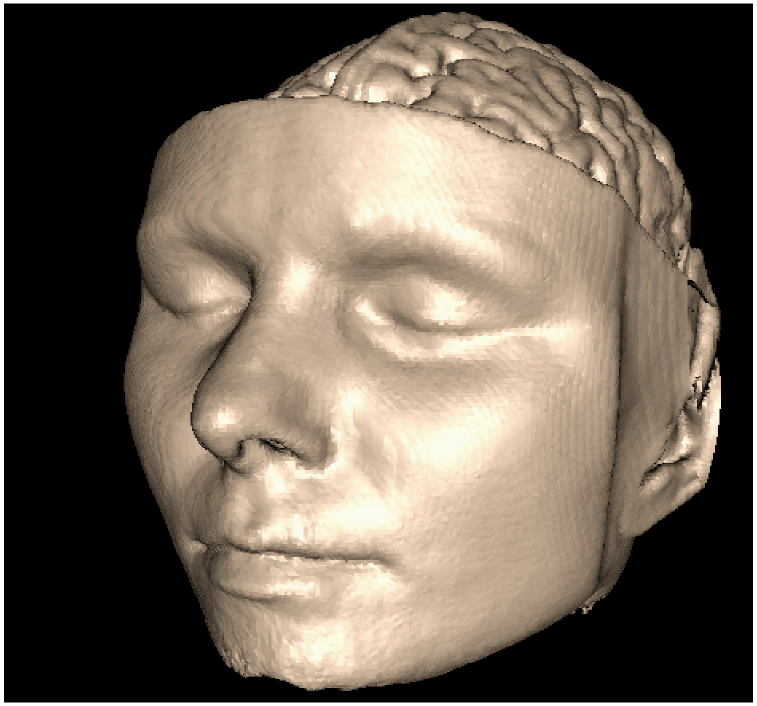
$$G_x = v_{i+1,j,k} - v_{i-1,j,k}$$

$$G_y = v_{i,j+1,k} - v_{i,j-1,k}$$

$$G_z = v_{i,j,k+1} - v_{i,j,k-1}$$

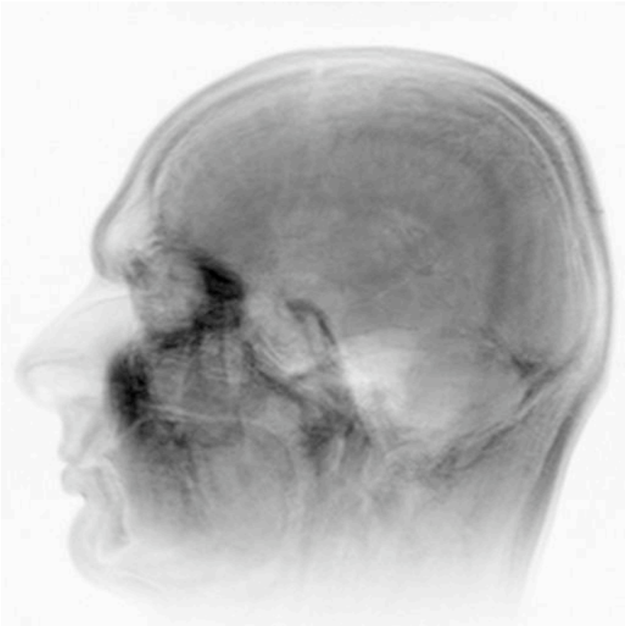


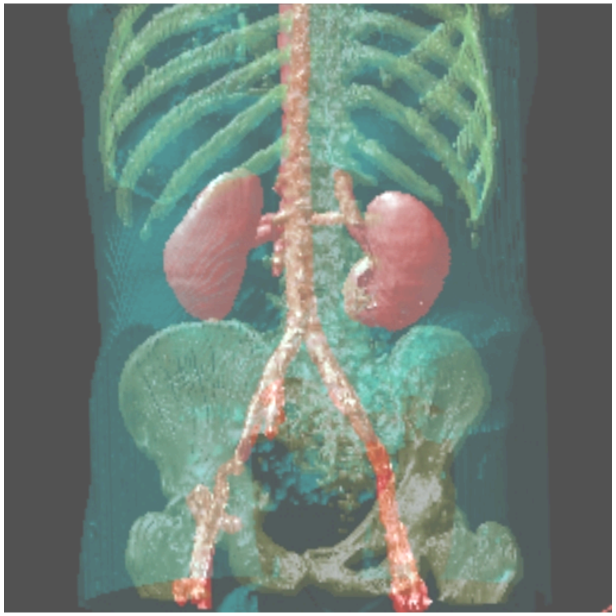
MC 7: Render!



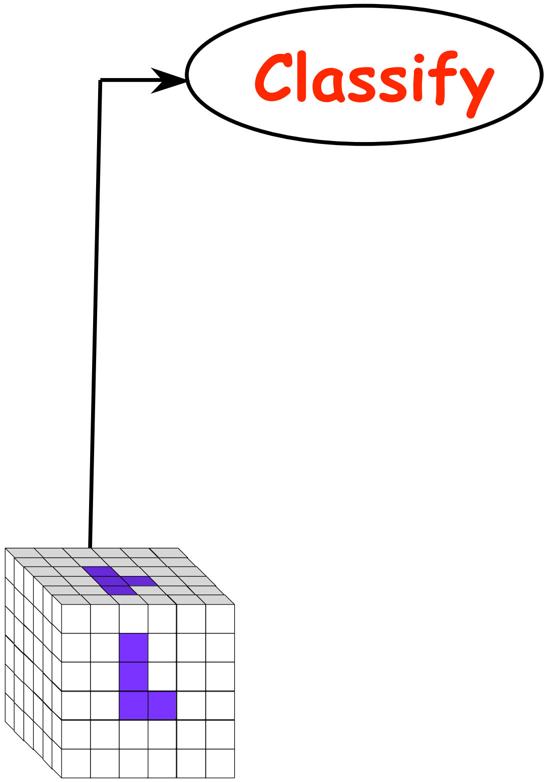
Direct Volume Rendering

- do not compute surface



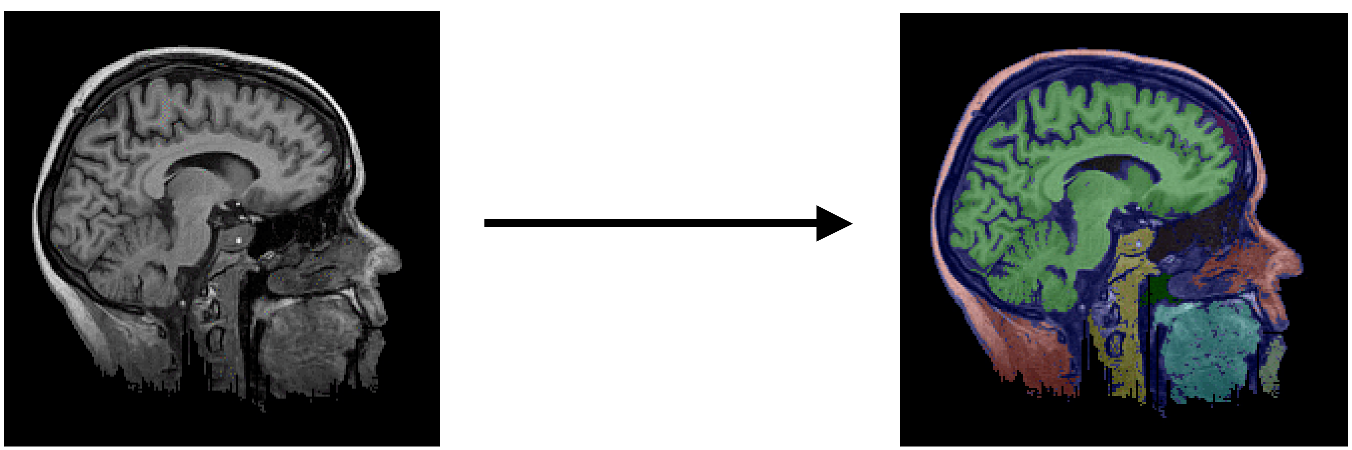


Rendering Pipeline



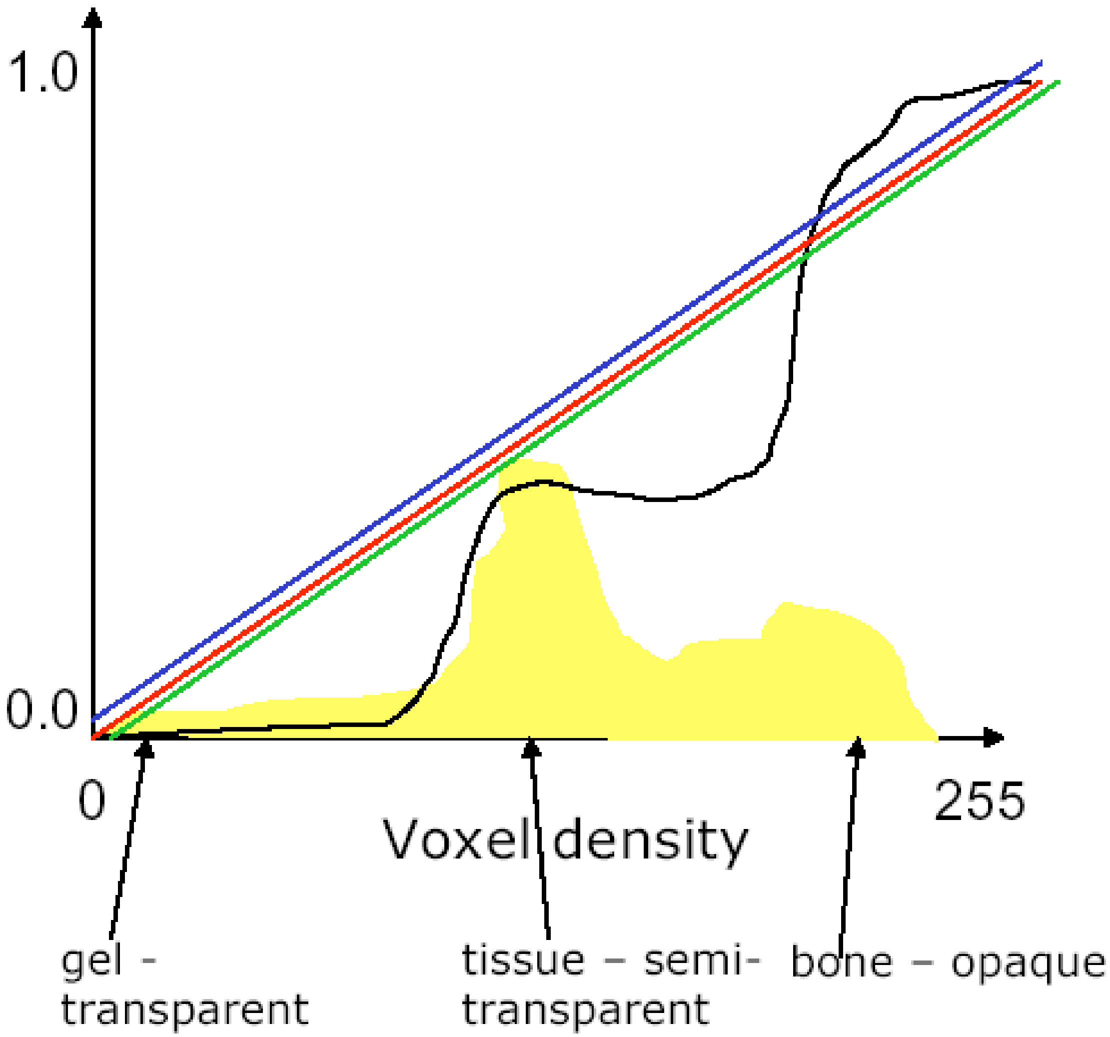
Classification

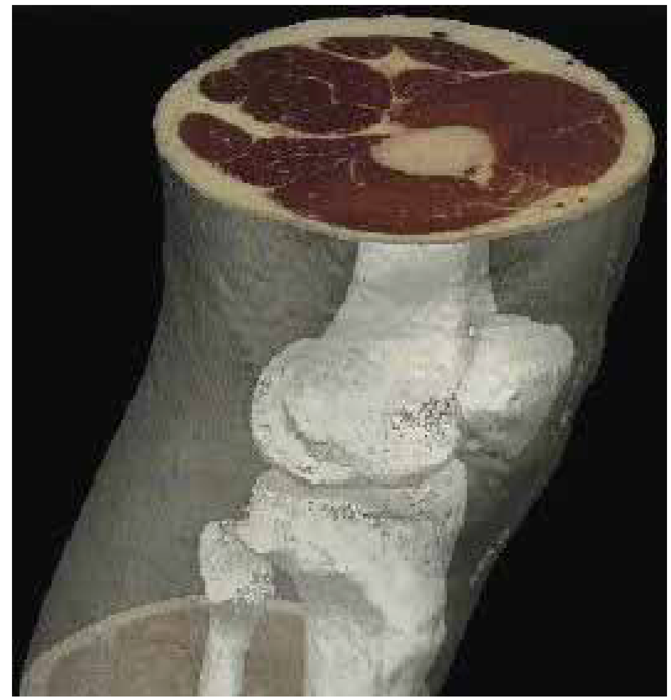
- data set has application-specific values
  - temperature, velocity, proton density, etc.
- assign these to color/opacity values to make sense of data
- achieved through transfer functions



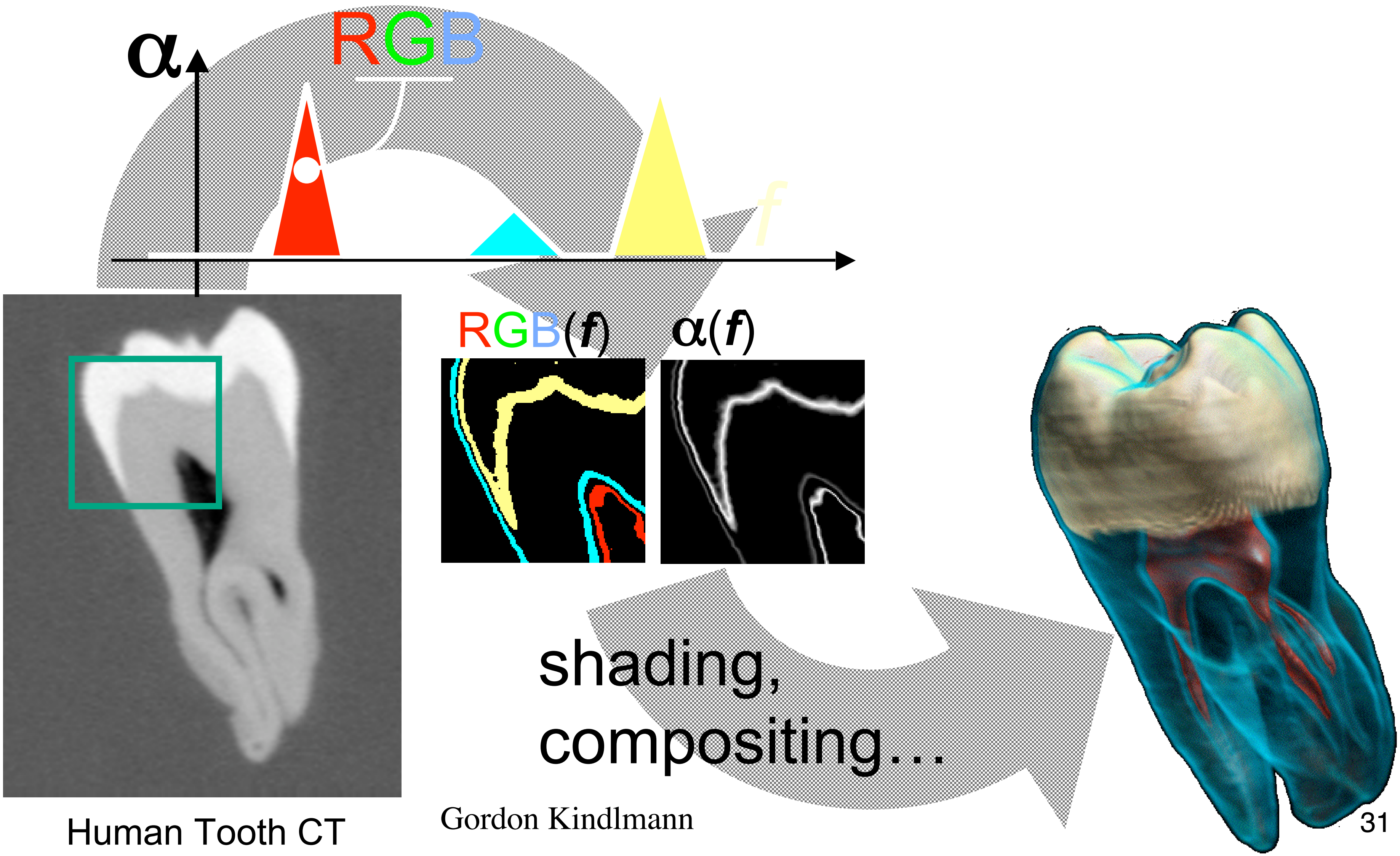
Transfer Functions

- map data value to color and opacity





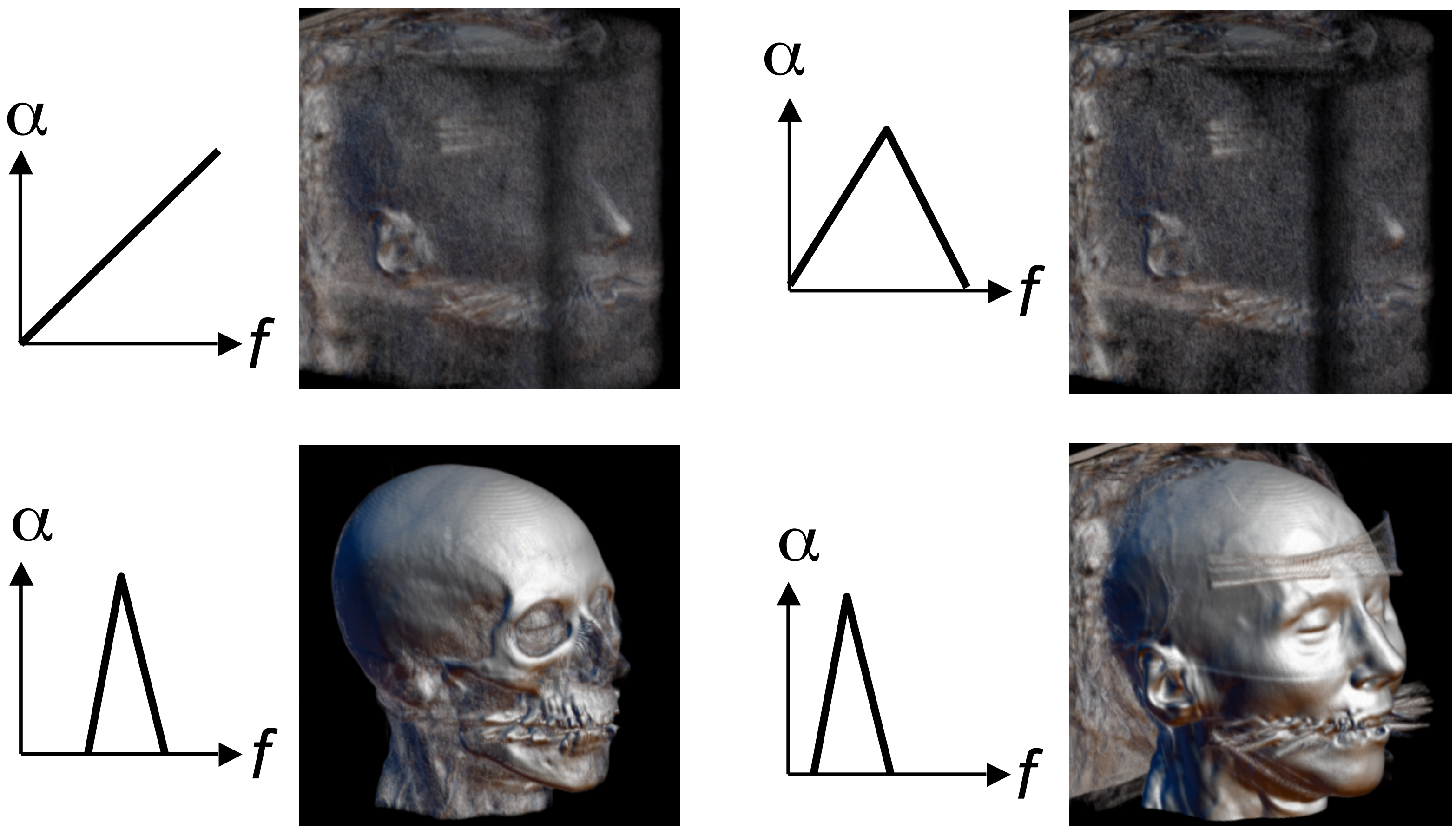
Transfer Functions



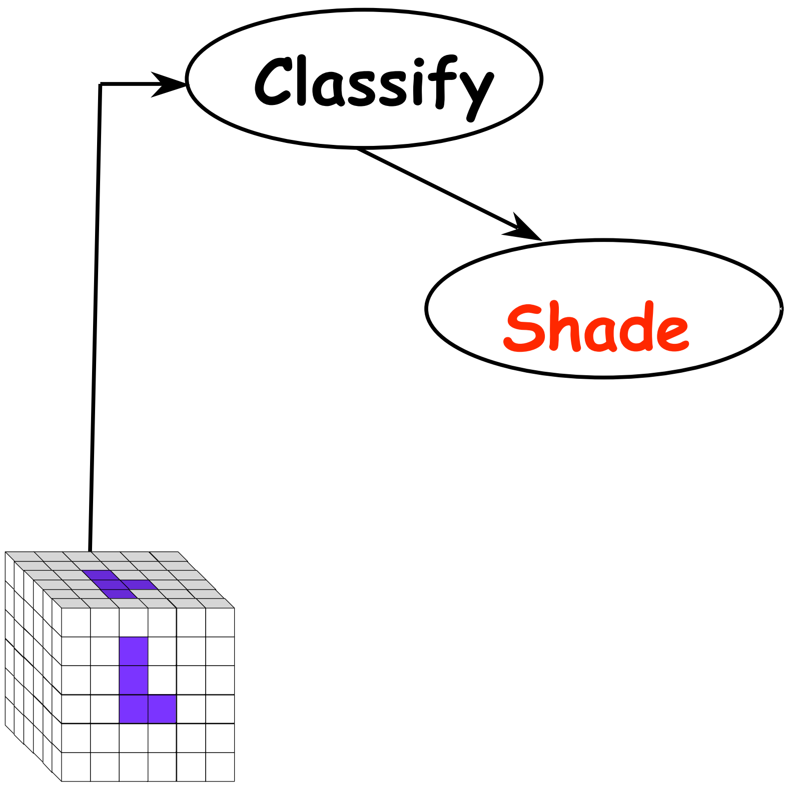
Human Tooth CT

Setting Transfer Functions

- can be difficult, unintuitive, and slow

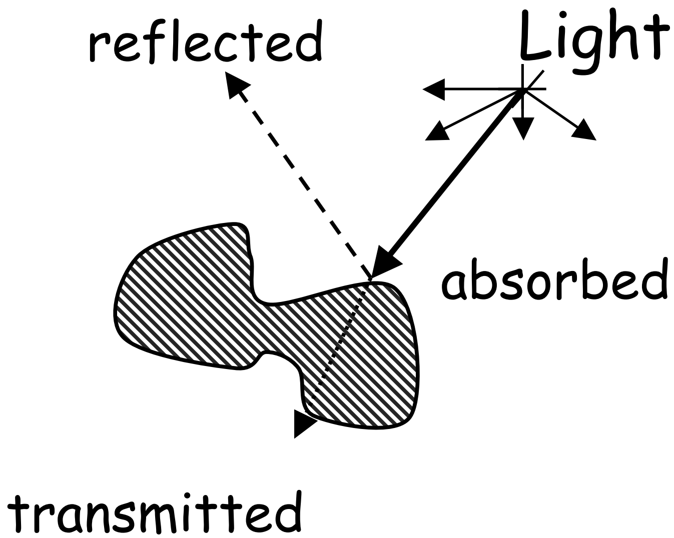


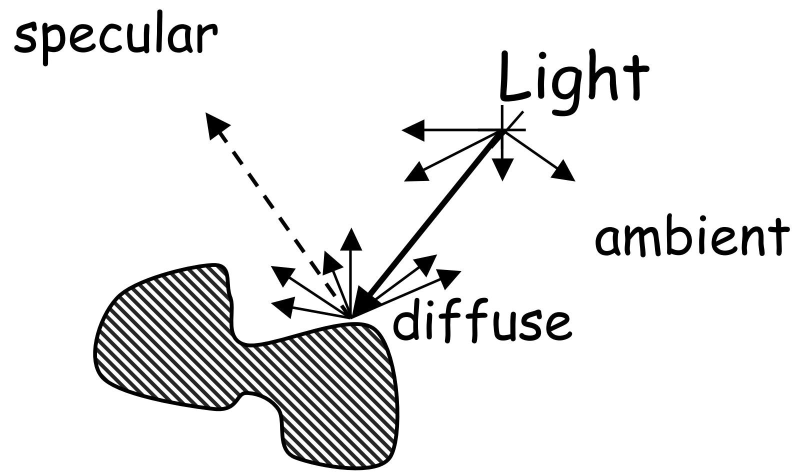
Rendering Pipeline



Light Effects

- usually only consider reflected part



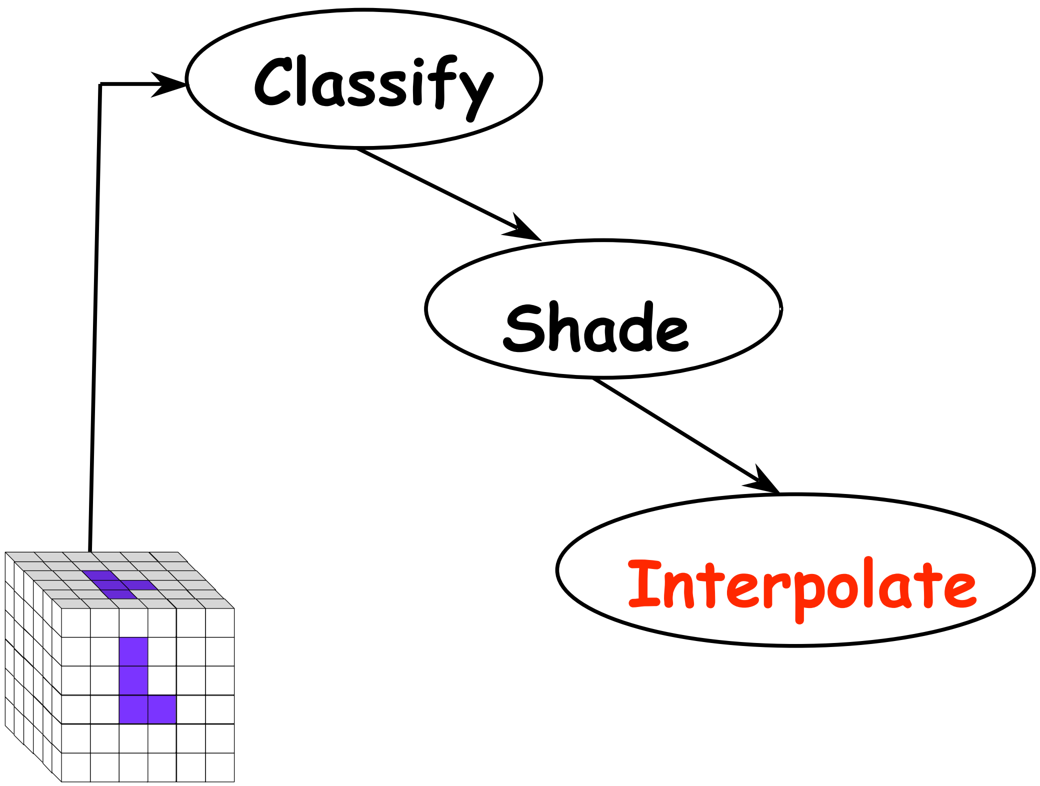


Light=refl.+absorbed+trans.

Light=ambient+diffuse+specular

$$I = k_a I_a + k_d I_d + k_s I_s$$

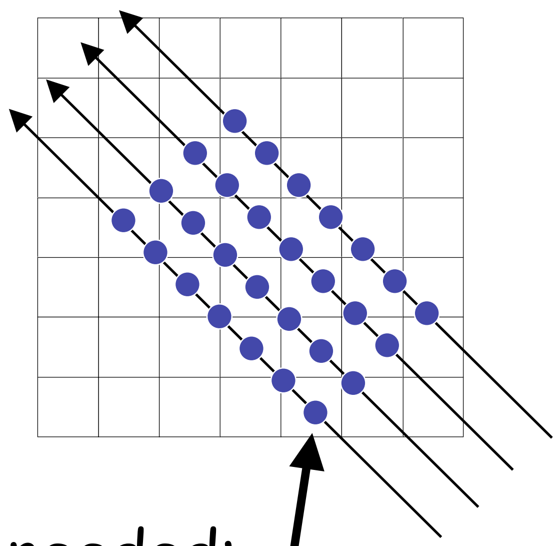
Rendering Pipeline



Interpolation

2D

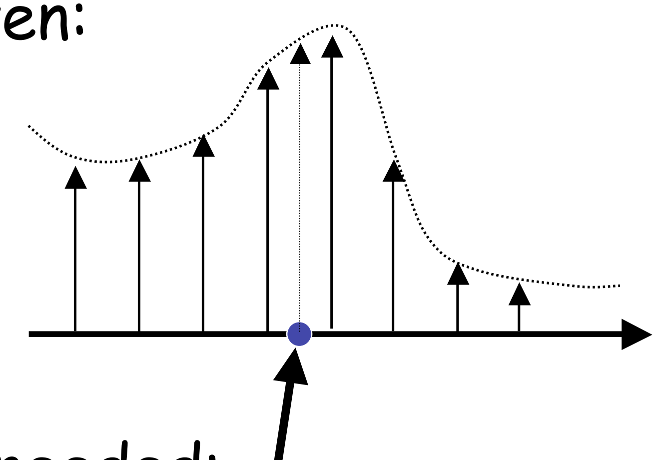
- given:



- needed:

1D

- given:

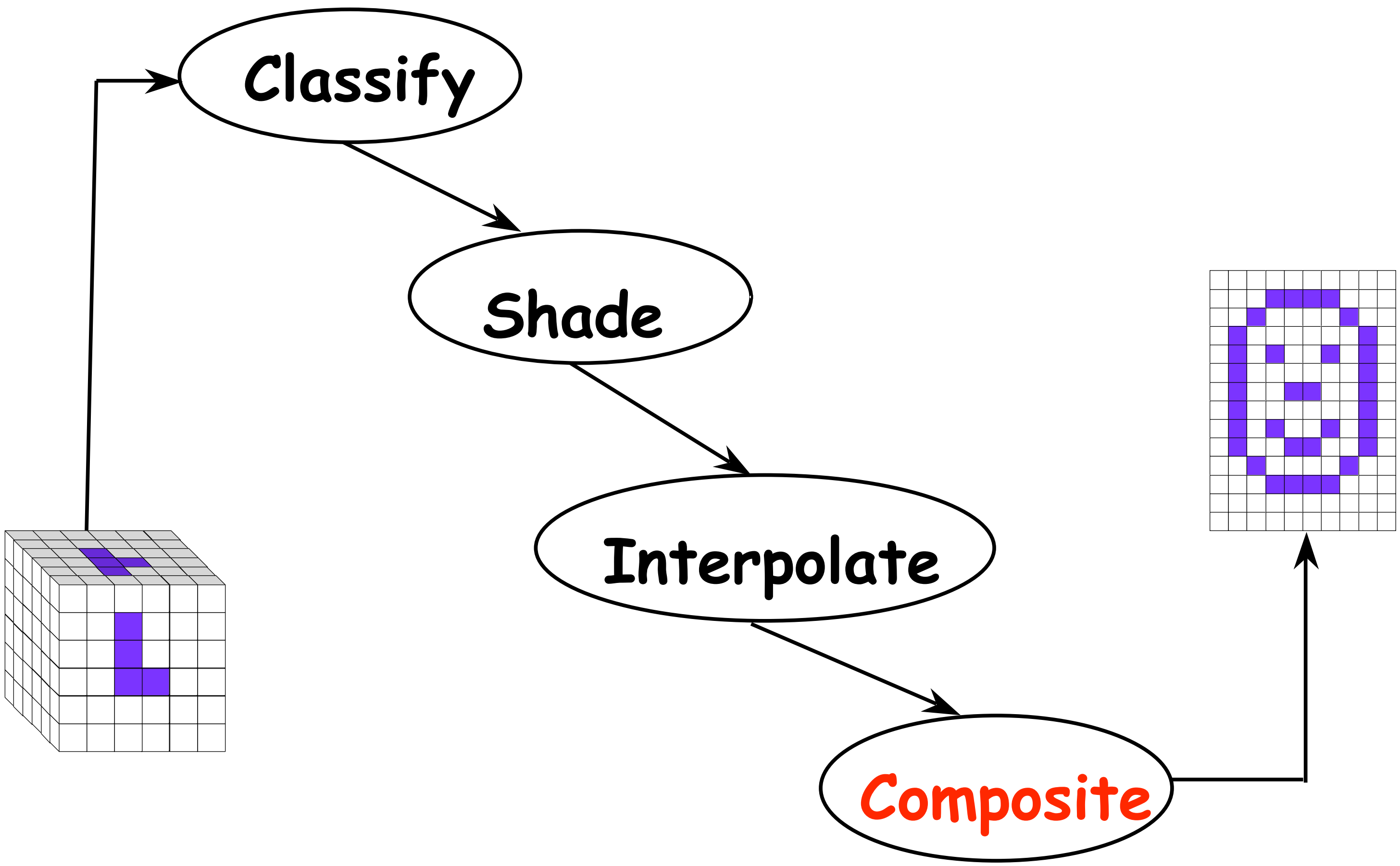


- needed:

nearest neighbor

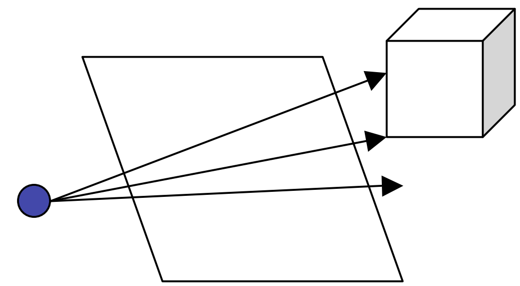
linear

Rendering Pipeline

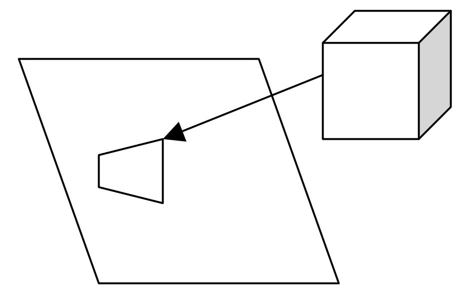


Volume Rendering Algorithms

- ray casting
  - image order, forward viewing

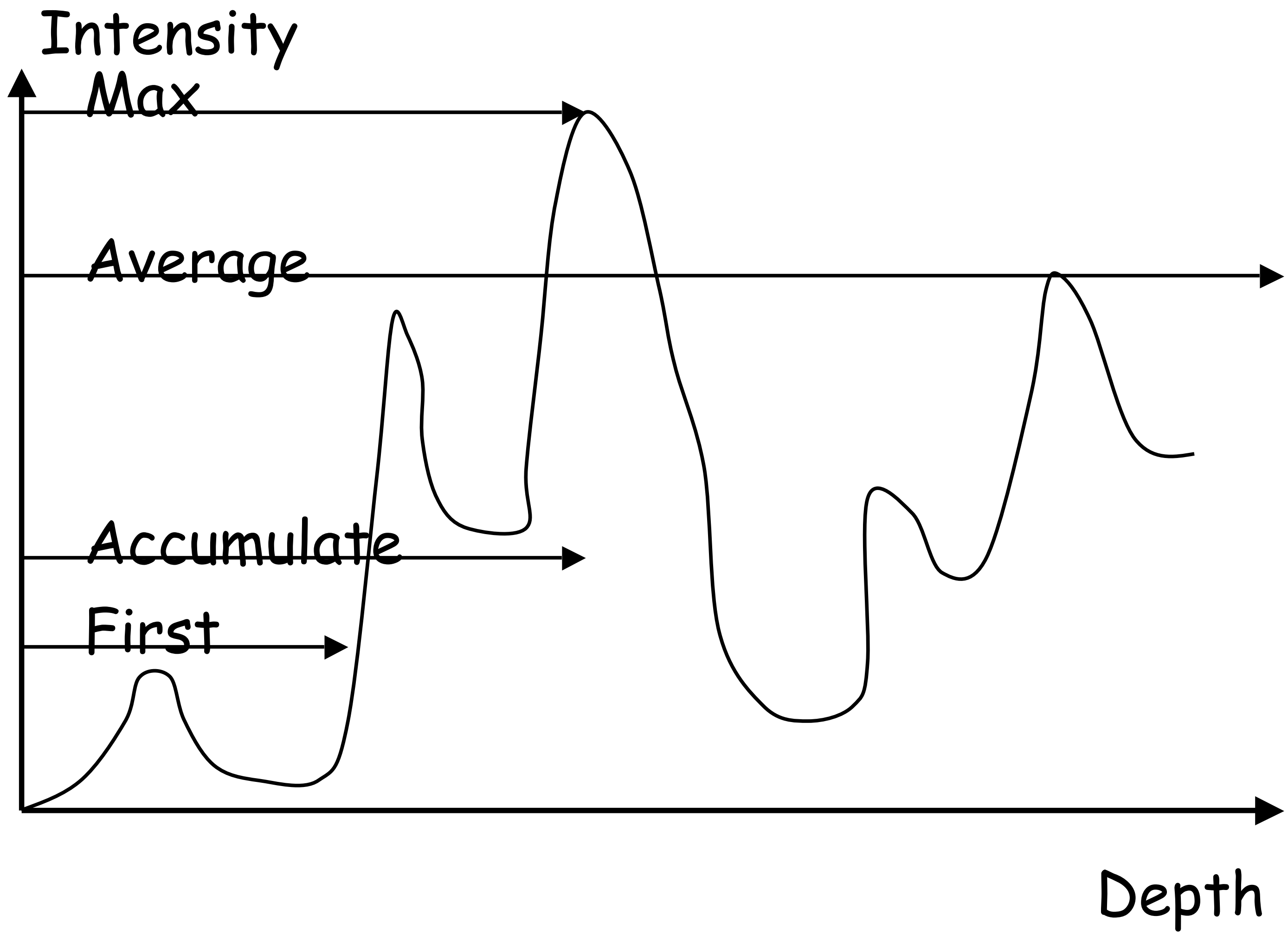


- splatting
  - object order, backward viewing



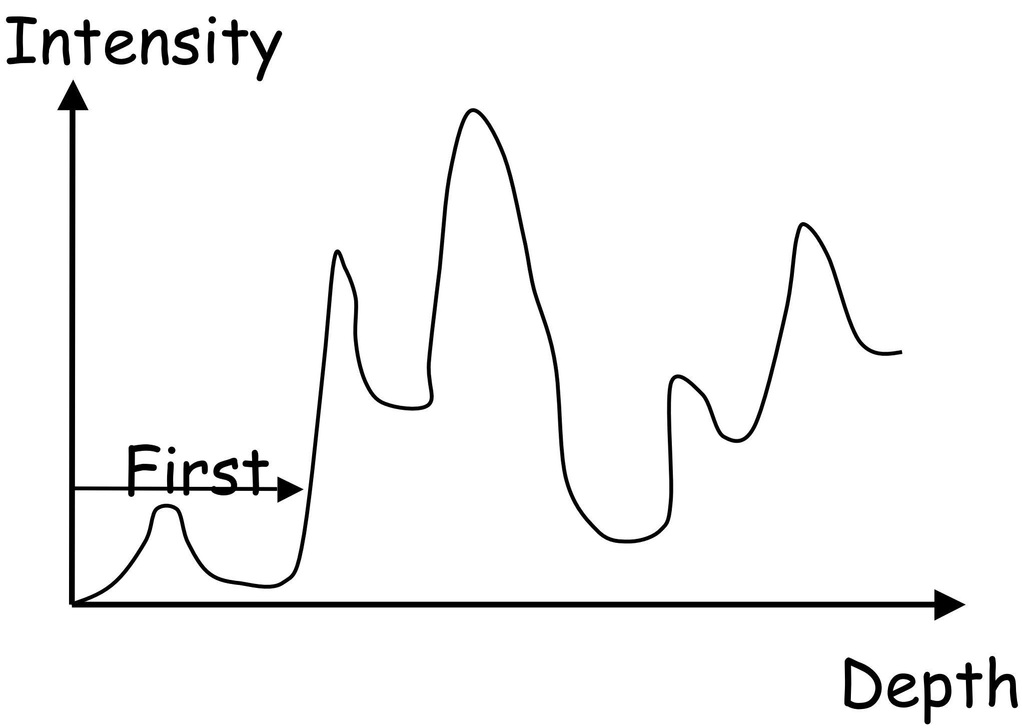
- texture mapping
  - object order
  - back-to-front compositing

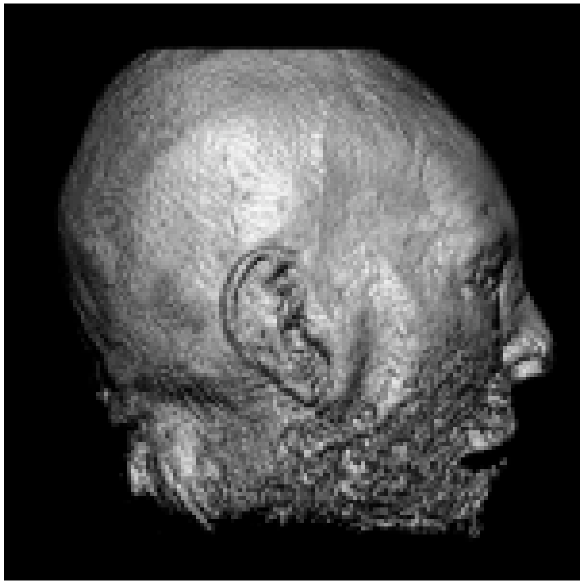
Ray Traversal Schemes



Ray Traversal - First

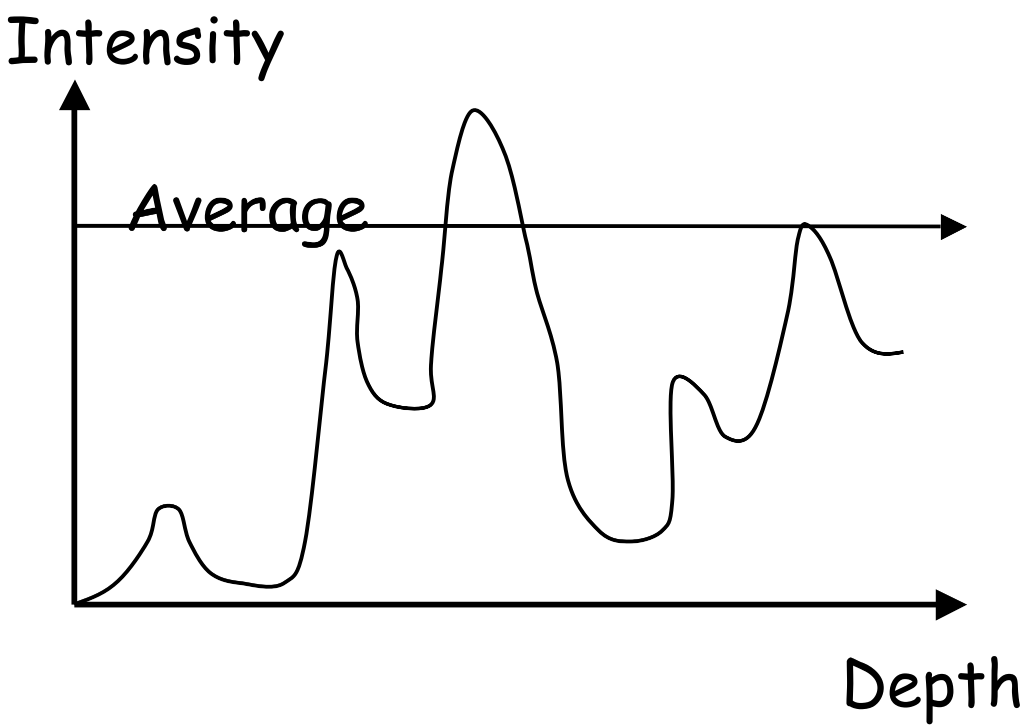
- first: extracts iso-surfaces (again!)

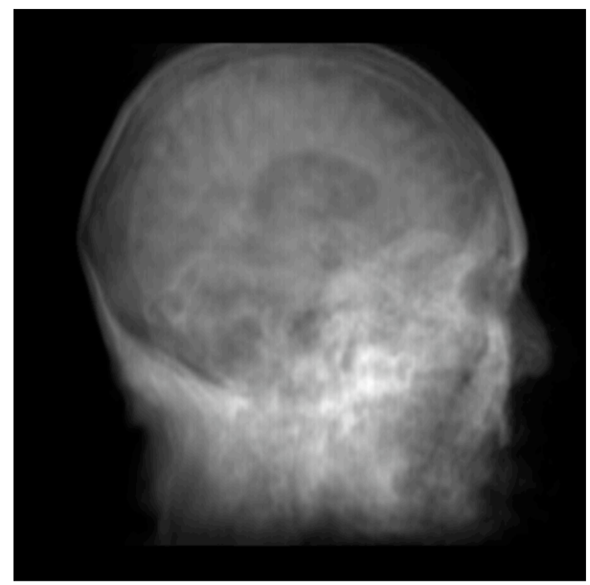




Ray Traversal - Average

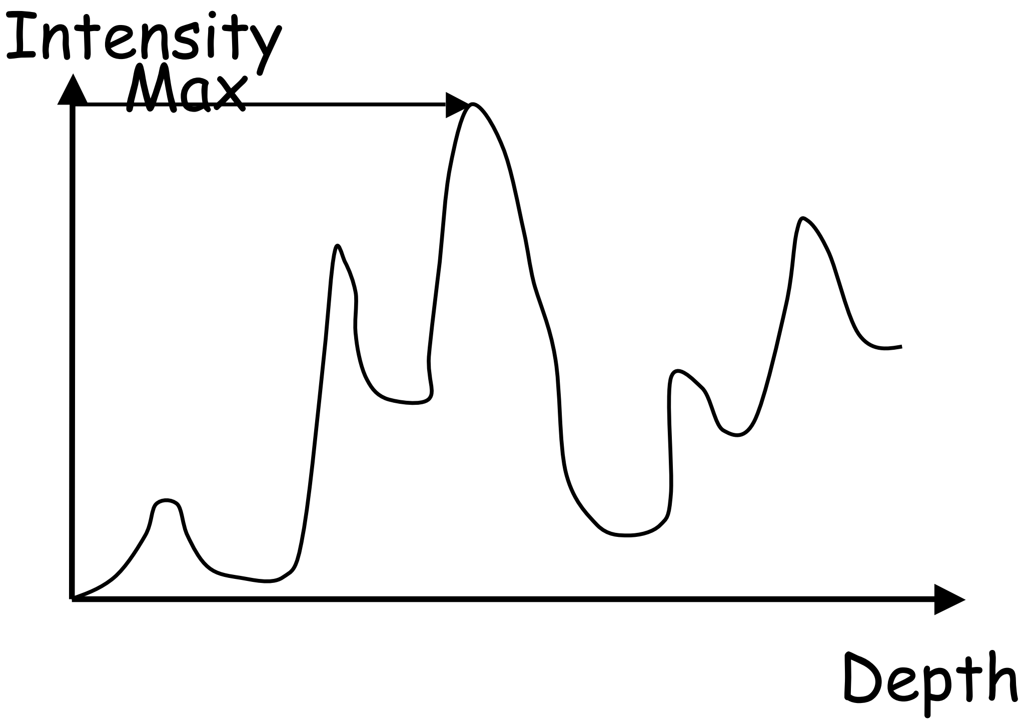
- average: looks like X-ray

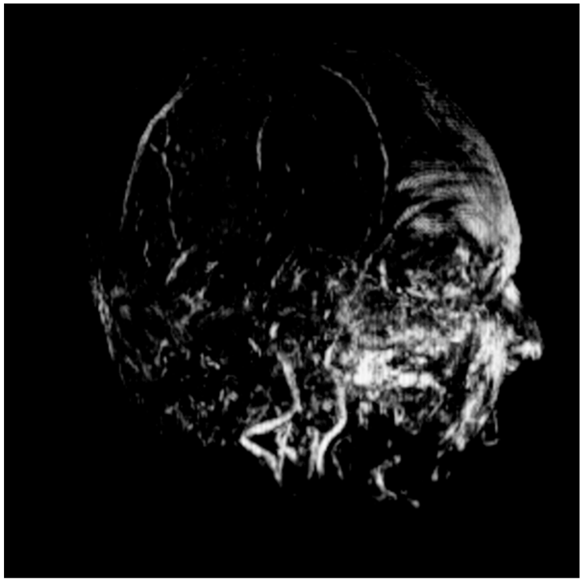




Ray Traversal - MIP

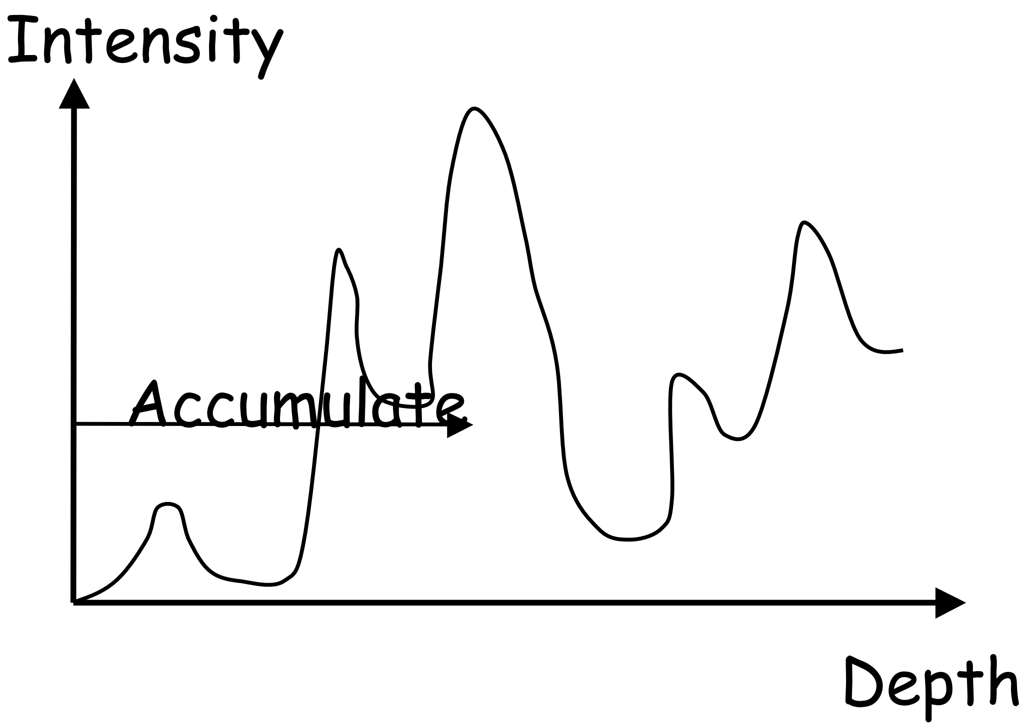
- max: Maximum Intensity Projection
  - used for Magnetic Resonance Angiogram

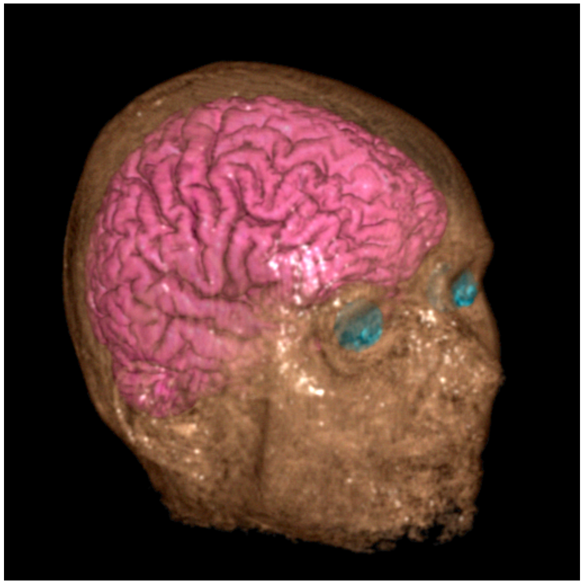




Ray Traversal - Accumulate

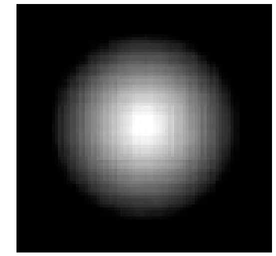
- accumulate: make transparent layers visible



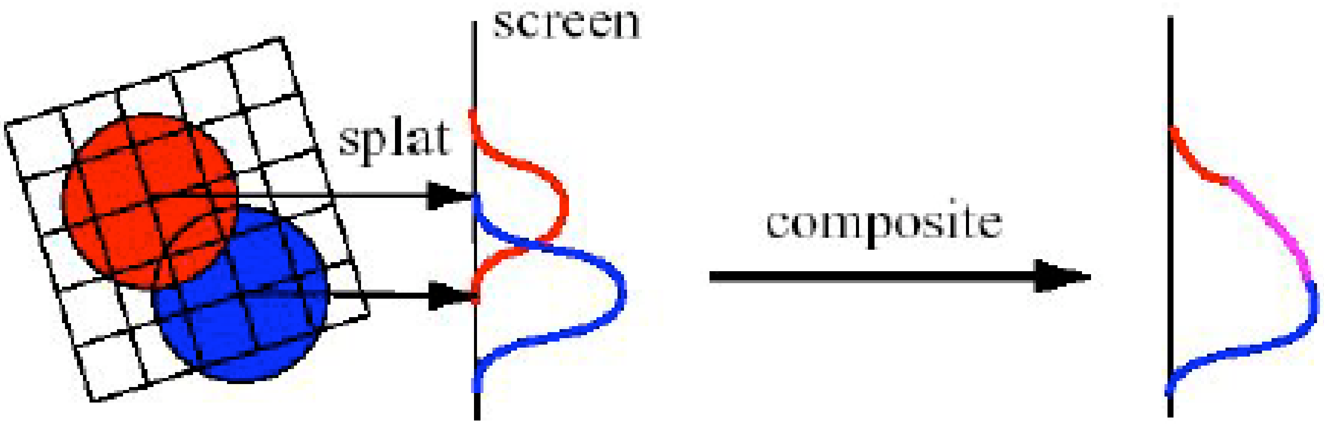


Splatting

- each voxel represented as fuzzy ball
  - 3D gaussian function
  - RGBa value depends on transfer function

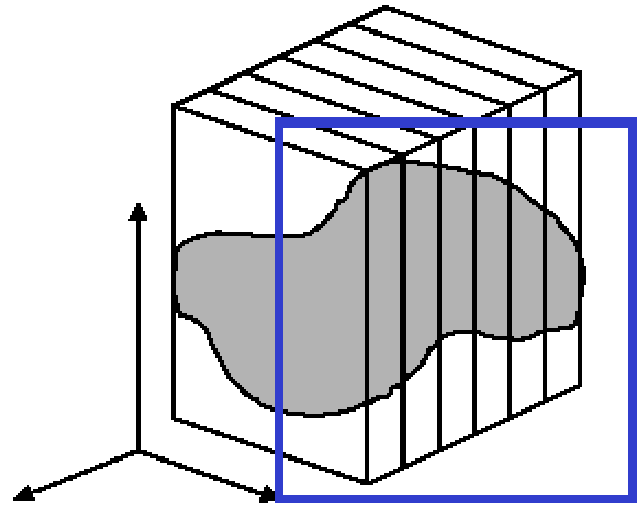


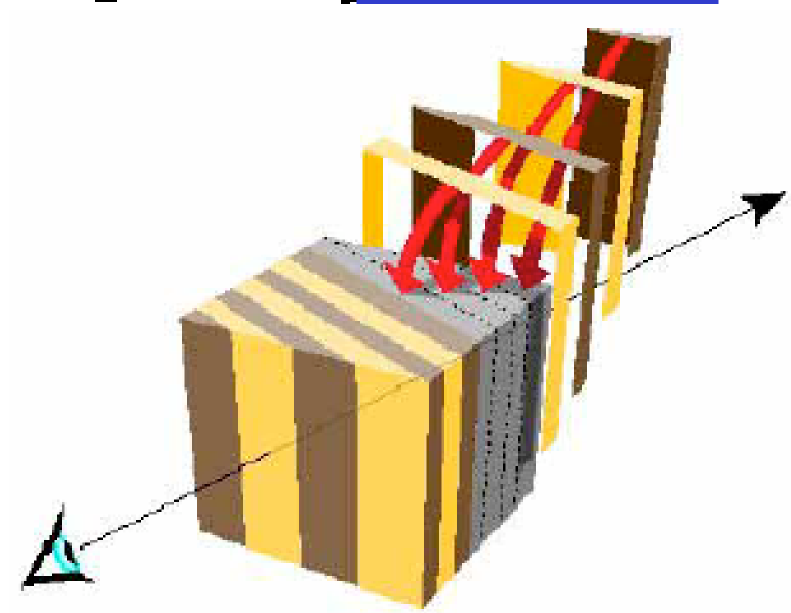
- fuzzy balls projected on screen, leaving footprint called splat
  - composite front to back, in object order



Texture Mapping

- 2D: axis aligned 2D textures
  - back to front compositing
  - commodity hardware support
  - must calculate texture coordinates, warp to image plane
- 3D: image aligned 3D texture
  - simple to generate texture coordinates





InfoVis Example: TreeJuxtaposer

- side by side comparison of evolutionary trees
  - stretch and squish navigation
  - guaranteed visibility
  - progressive rendering
- demo - downloadable from http://olduvai.sf.net/tj