Biomedical Informatics 260

Visualization of Medical Images

Lecture 2

David Paik, PhD

Spring 2017

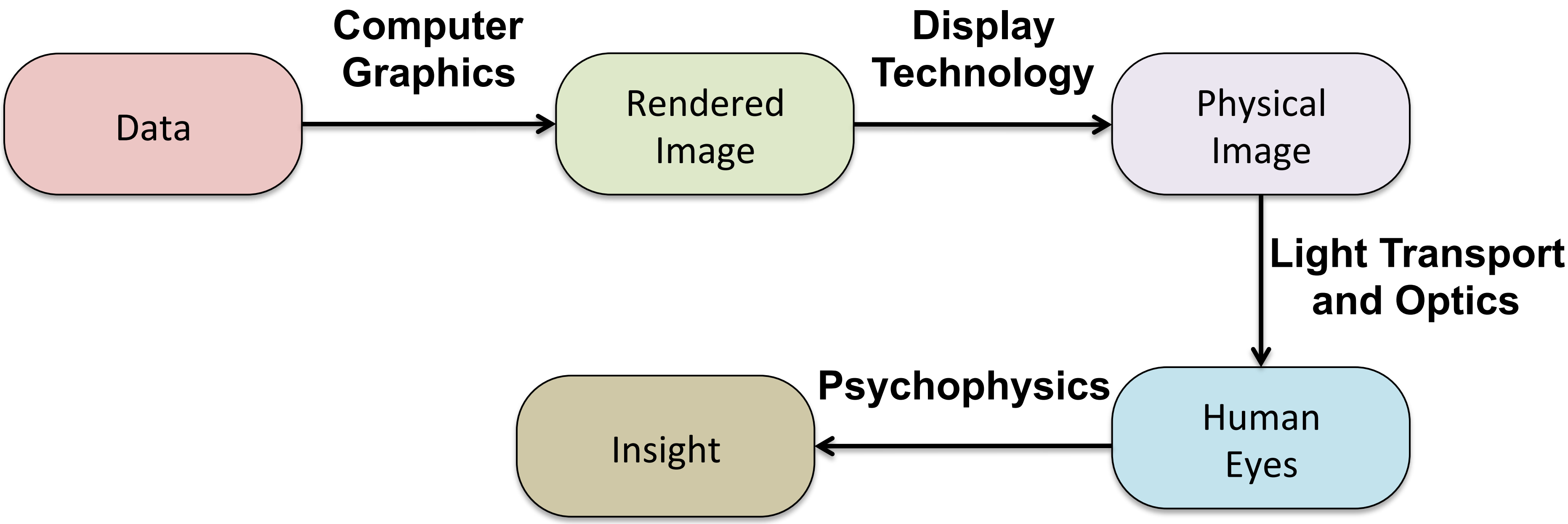
Last Lecture: Imaging Modalities

1. How images are acquired
2. Meaning of pixel values

Today: Visualization

- How do we go from an array of pixel values to a displayed images to human insight about image content?
- Processing images always starts with visualization
- Methods covered today:
  - 2D Computer Graphics
  - 3D Computer Graphics
  - Image Fusion
  - Perception

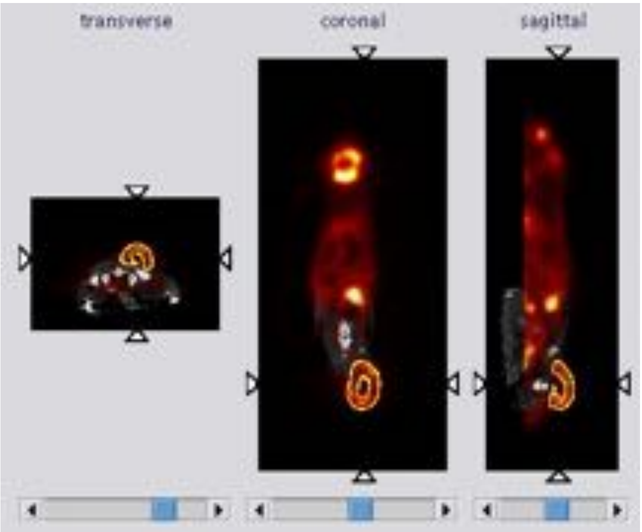
Visualization: From Data to Insight



2D Computer Graphics

Image Viewing Environments

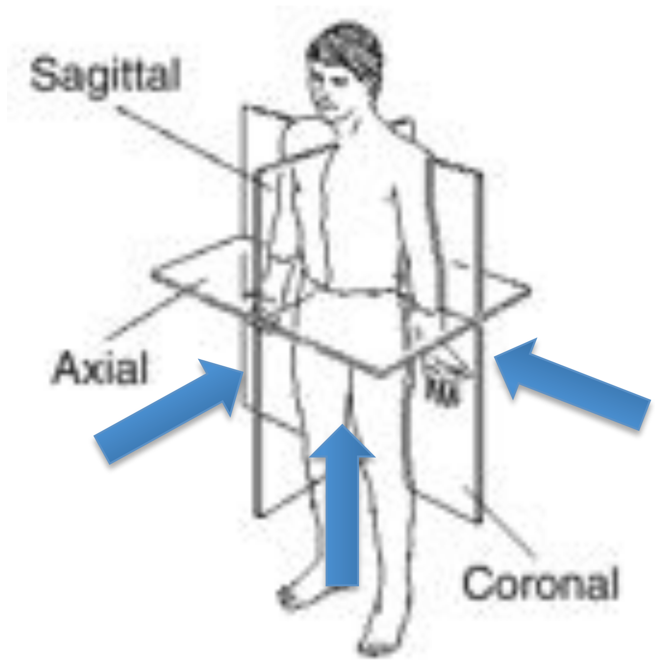
Typical PACS Workstation

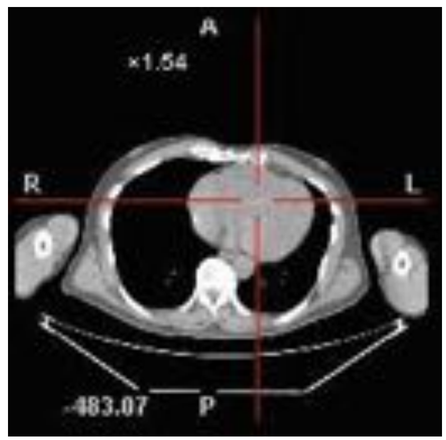


Typical Research Software

PACS = Picture Archiving and Communication System

Multiplanar Reconstruction (MPR)





Axial
(transverse)

Coronal

Sagittal

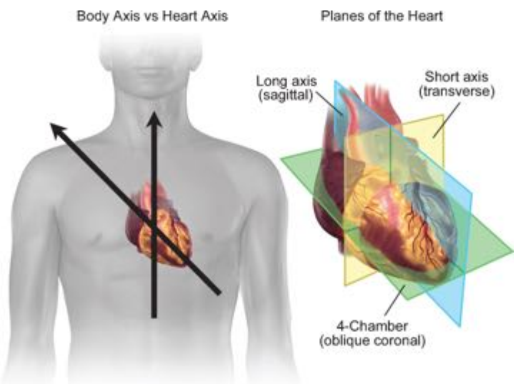
Radiologist convention:

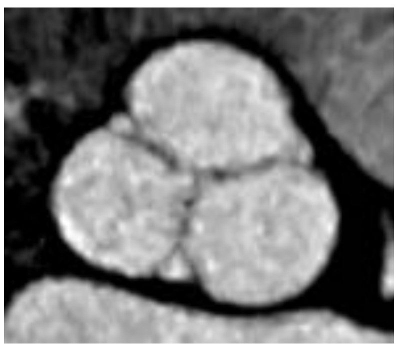
View from the bottom

View from the front
Note R-L flipping!

View from left

Multiplanar Reconstruction (MPR)

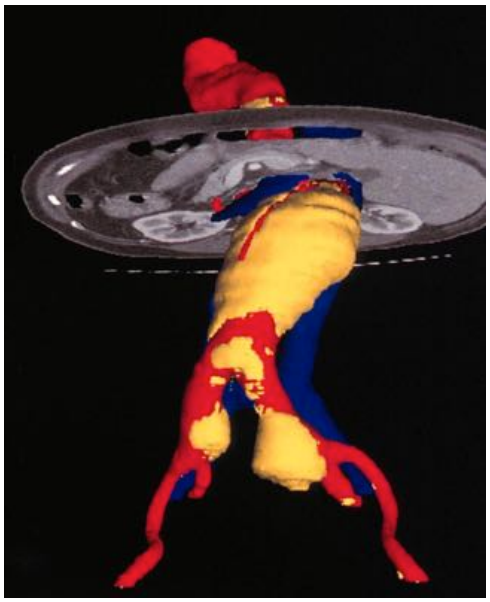




Oblique

(often linked to the axis of an organ)

Multiple Planes

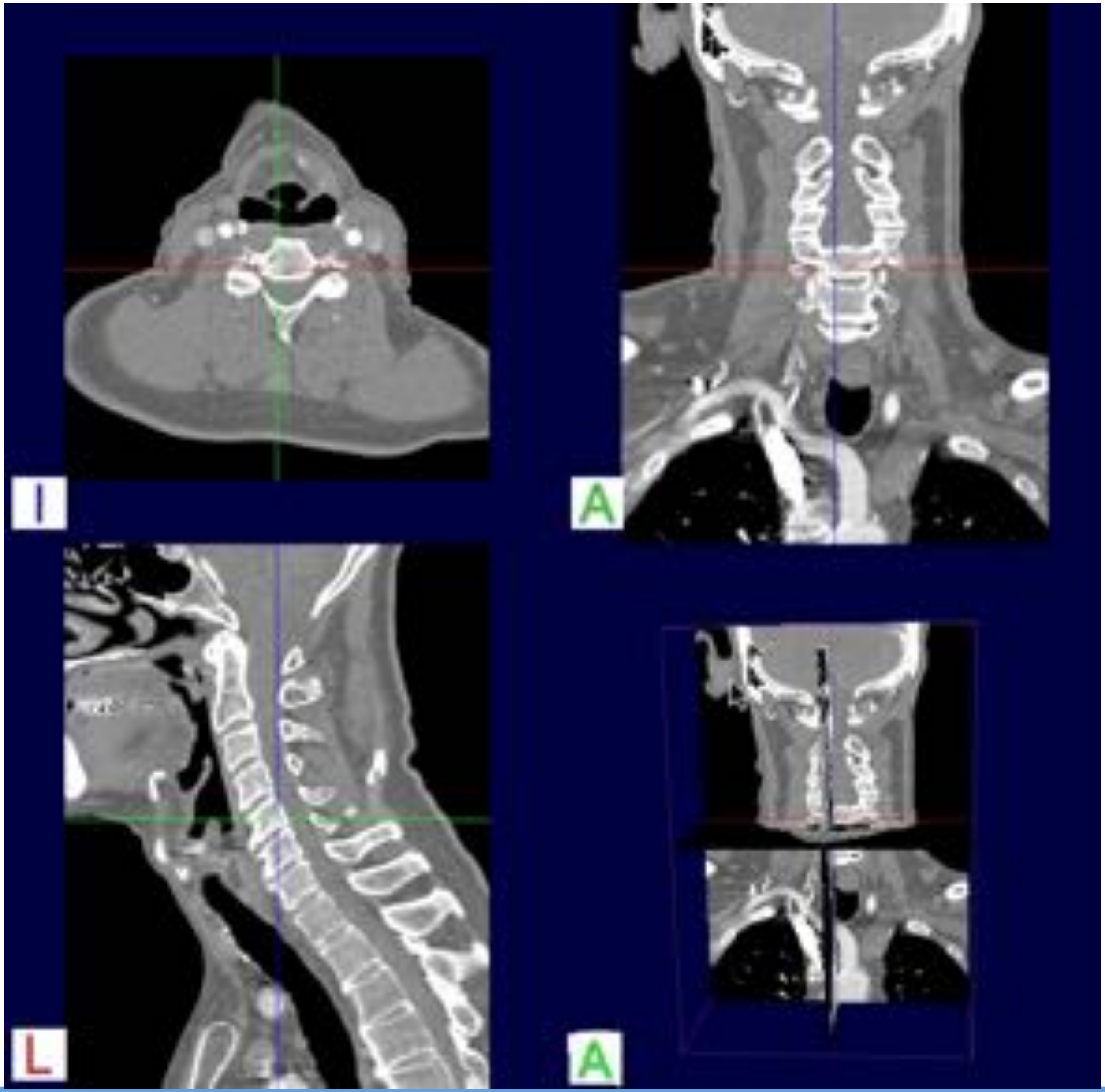


2D+3D

MPR

Axial

Coronal



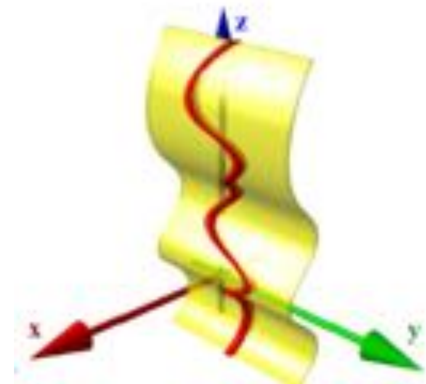
Sagittal

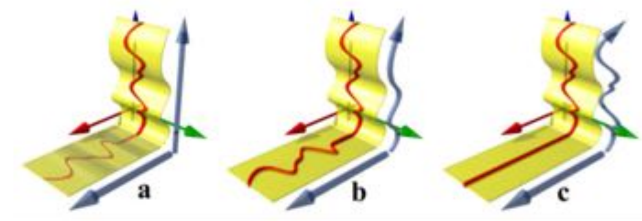
3D

Oblique Plane

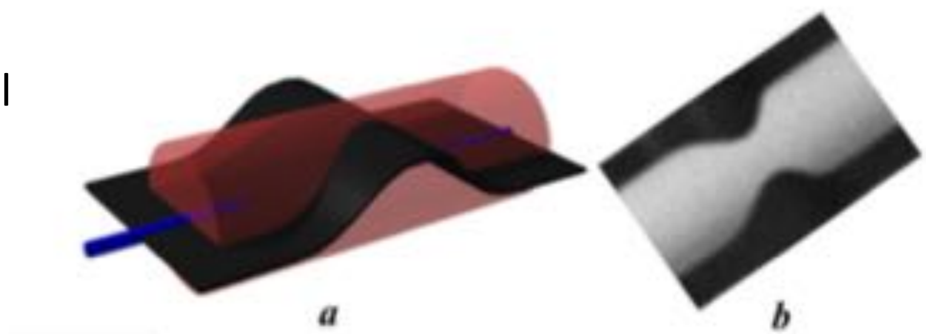
Curved Planar Reformations (CPR)

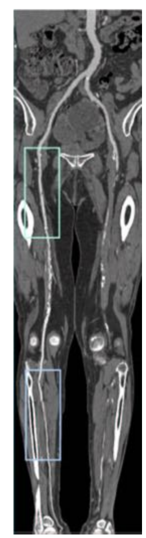
- Centerline is determined
  - Manually or automatically
- Sampling along parallel lines
- Various methods for assembling sampled lines into final image
- Pros and Cons
  - Single image to show a long region
  - Artificial stenosis artifact possible





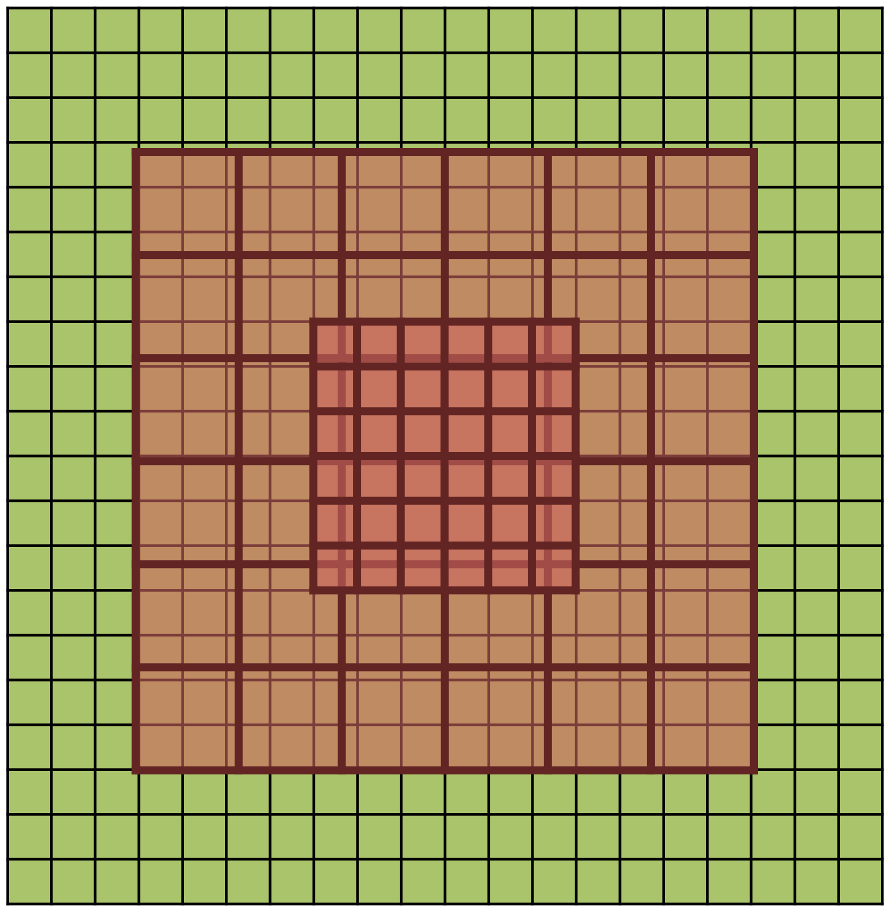
Potential Pitfall of CPR:



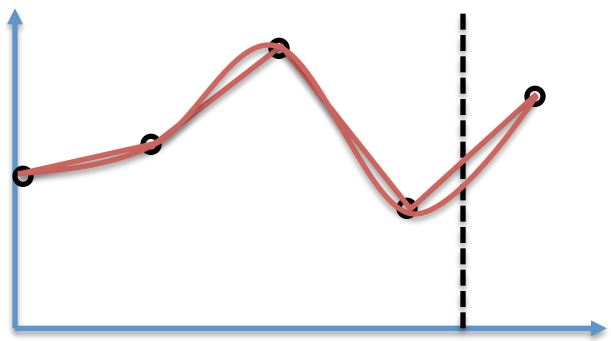


Kanitsar et al., IEEE Viz 2003

Image Pixels Don't Always Align With Screen Pixels

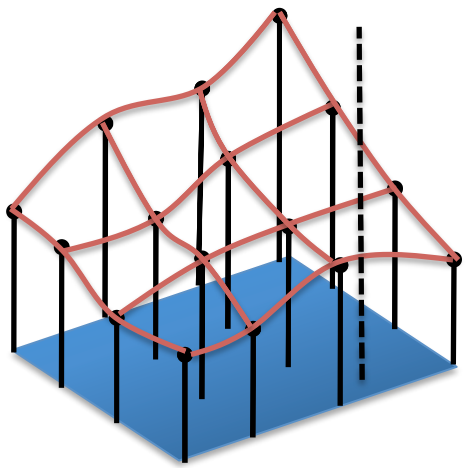


Interpolation



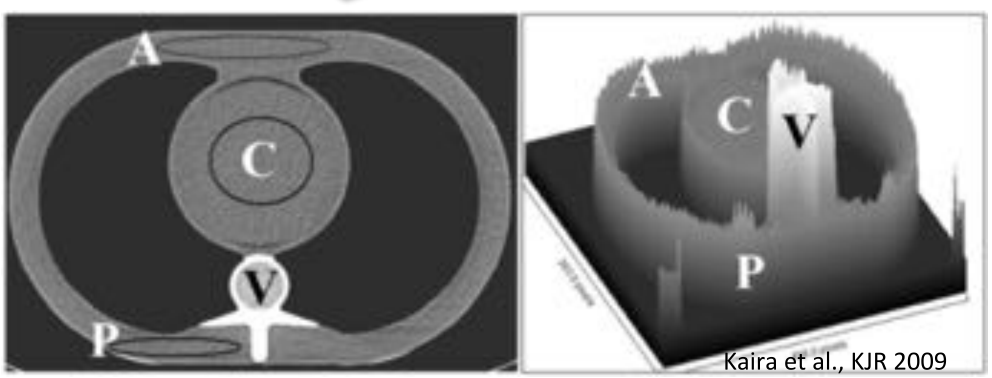
Linear Interpolation (“connect the dots”)

Higher Order Interpolation (smooth)



2D images are just two dimensional surface plots where height is image intensity

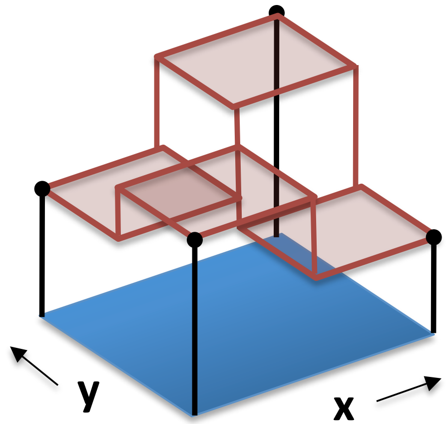
This naturally extends to 3D



Kaira et al., KJR 2009

Thinking of image intensity as a height will be a recurring theme in this course

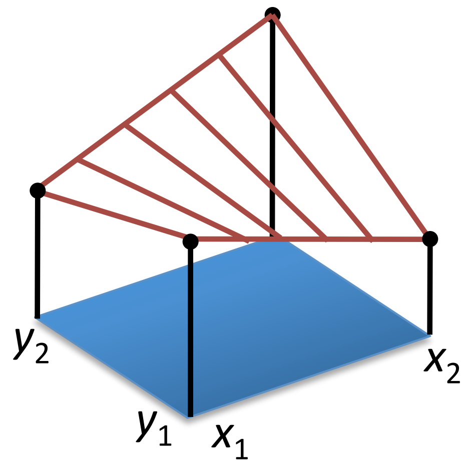
Image Interpolation



Nearest Neighbor
Interpolation

$$f(\mathbf{x},\mathbf{y}) = f\left([\mathbf{x}],[\mathbf{y}]\right)$$

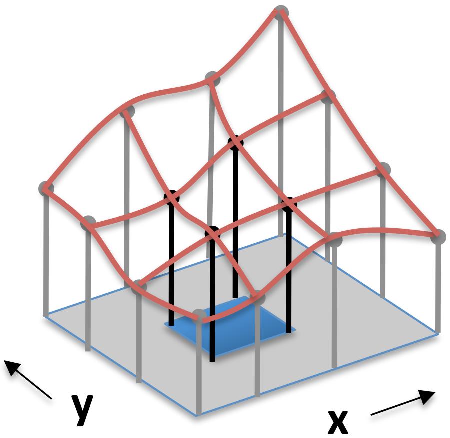
[ ] is rounding function



Bilinear
Interpolation

$$f(\mathbf{x},\mathbf{y}) = \frac{x_2-\mathbf{x}}{x_2-x_1}\cdot\frac{y_2-\mathbf{y}}{y_2-y_1}f(x_1,y_1) + \frac{\mathbf{x}-x_1}{x_2-x_1}\cdot\frac{y_2-\mathbf{y}}{y_2-y_1}f(x_2,y_1) +$$
$$\frac{x_2-\mathbf{x}}{x_2-x_1}\cdot\frac{\mathbf{y}-y_1}{y_2-y_1}f(x_1,y_2) + \frac{\mathbf{x}-x_1}{x_2-x_1}\cdot\frac{\mathbf{y}-y_1}{y_2-y_1}f(x_2,y_2)$$

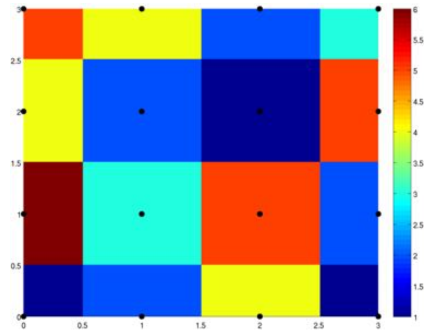
Image Interpolation



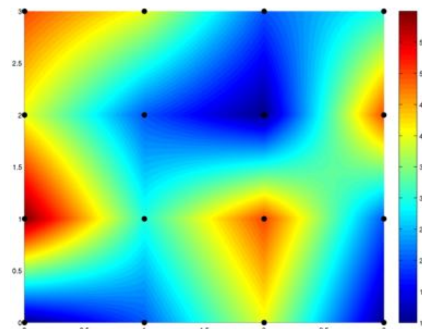
Bicubic Interpolation

$$f(\mathbf{x},\mathbf{y}) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}\mathbf{x}^i\mathbf{y}^j$$

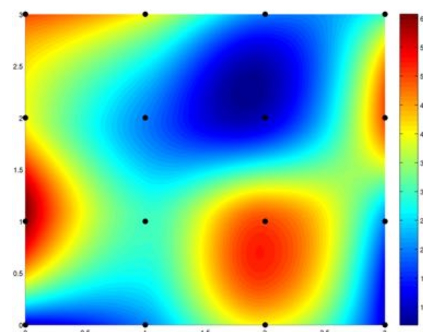
Nearest Neighbor



Bilinear

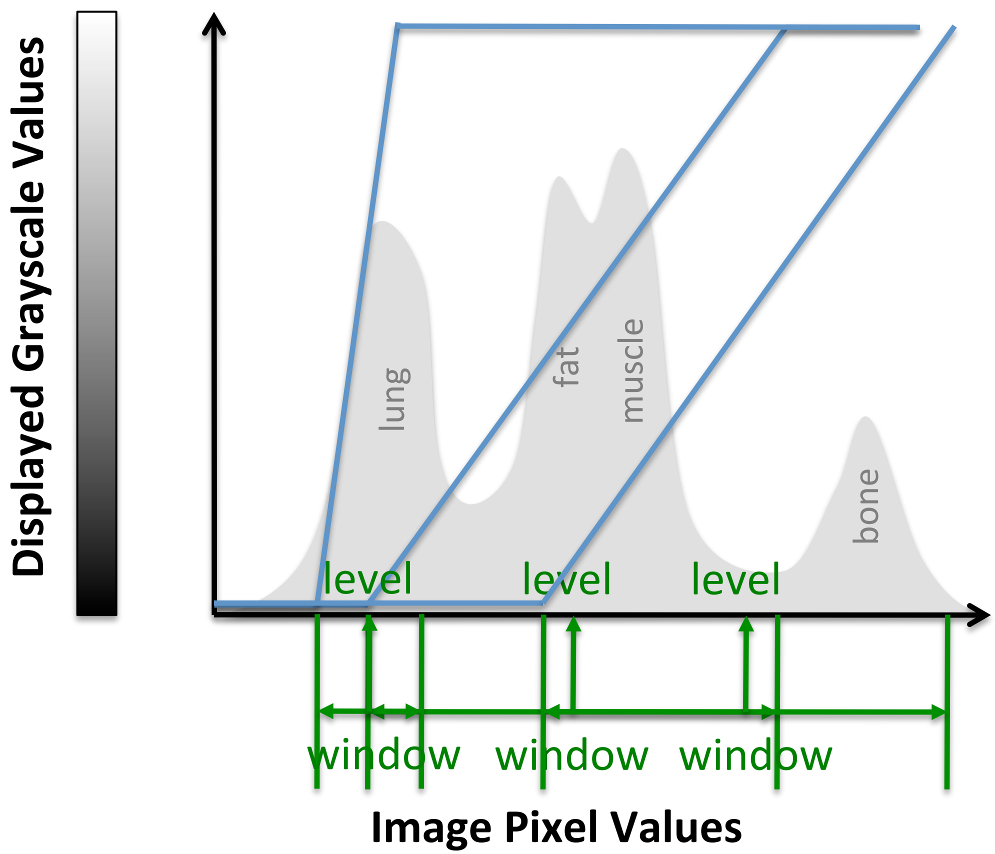


Bicubic



Intensity Scale Mapping

"Window Leveling"



Original

High Level

Low Level, Narrow Window

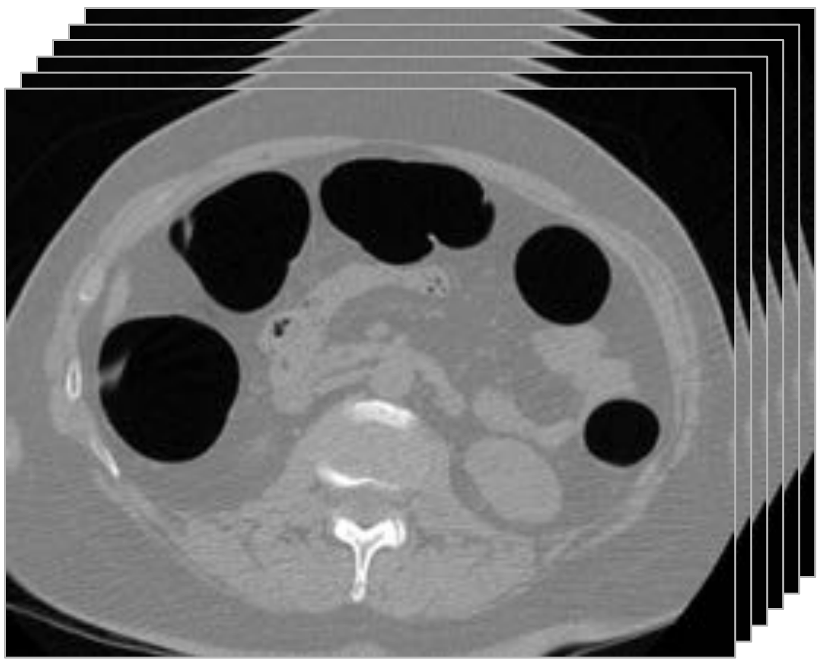
Most imaging modalities: 16 bits (65,536 values)
Most displays (and human eye): 8 bits (256 values)

(color mapping is complicated, more on this later)

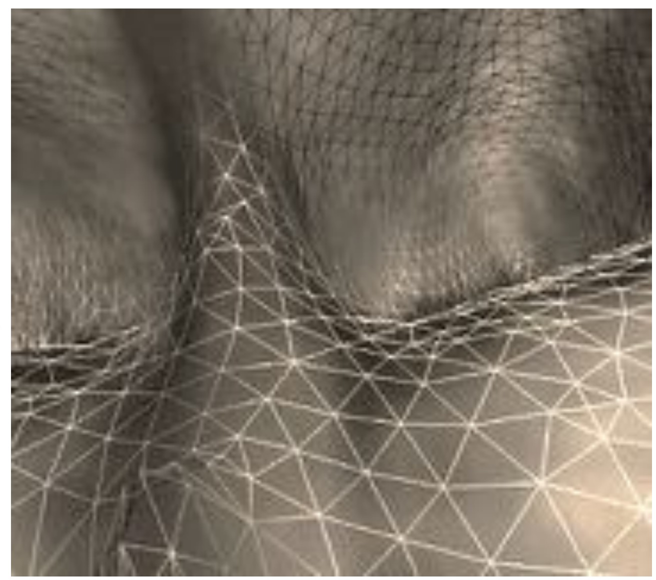
3D Computer Graphics

Marching Cubes Algorithm

- The goal is to take a 3D array of scalar values, find an iso-intensity surface, and then make a triangulated mesh surface of it



3D Image Dataset

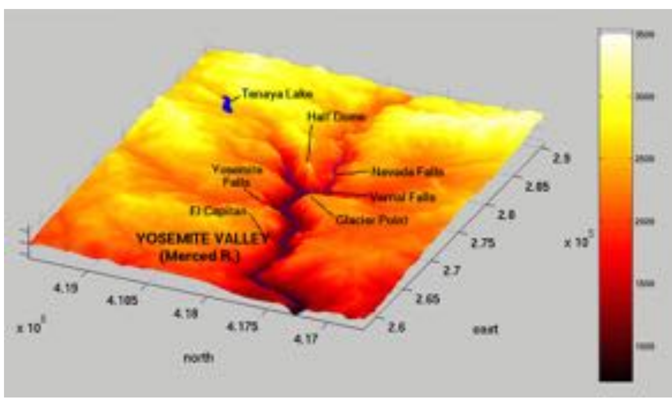


Radetzky et al., AIM 2002

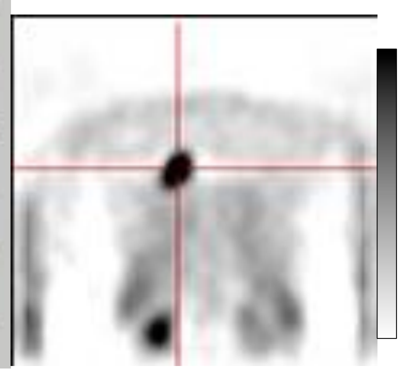
Triangular Mesh

Marching Cubes Algorithm

- But first, let's look at the simpler case of "Marching Squares" for 2D images

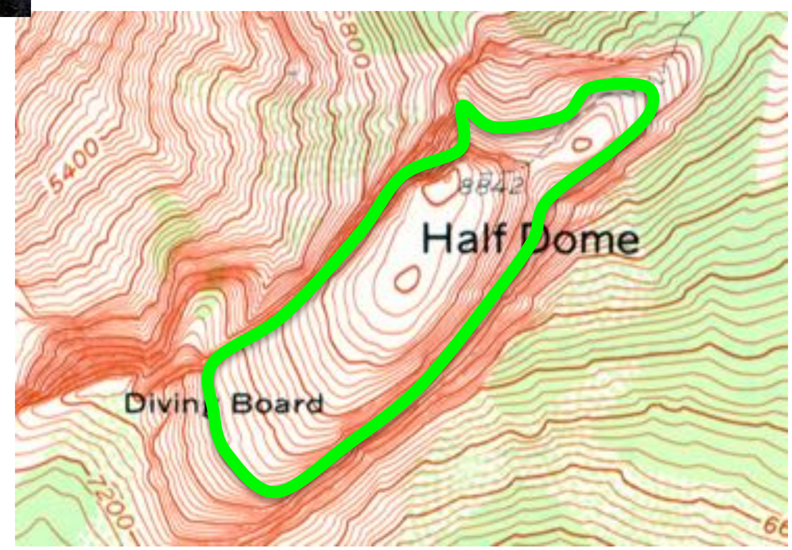


Yosemite Valley



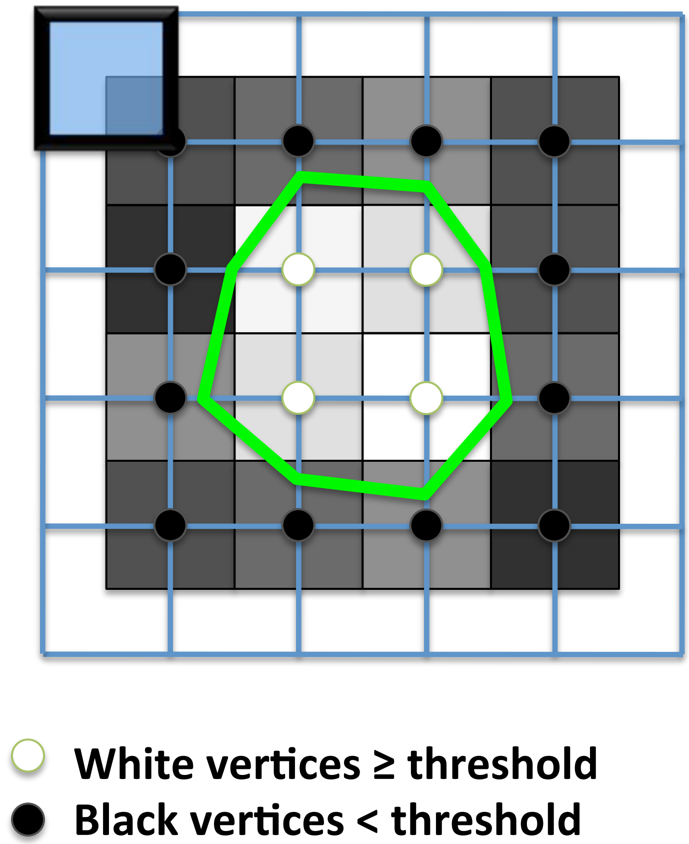
PET Scan

(assume altitude data is sampled on a grid)

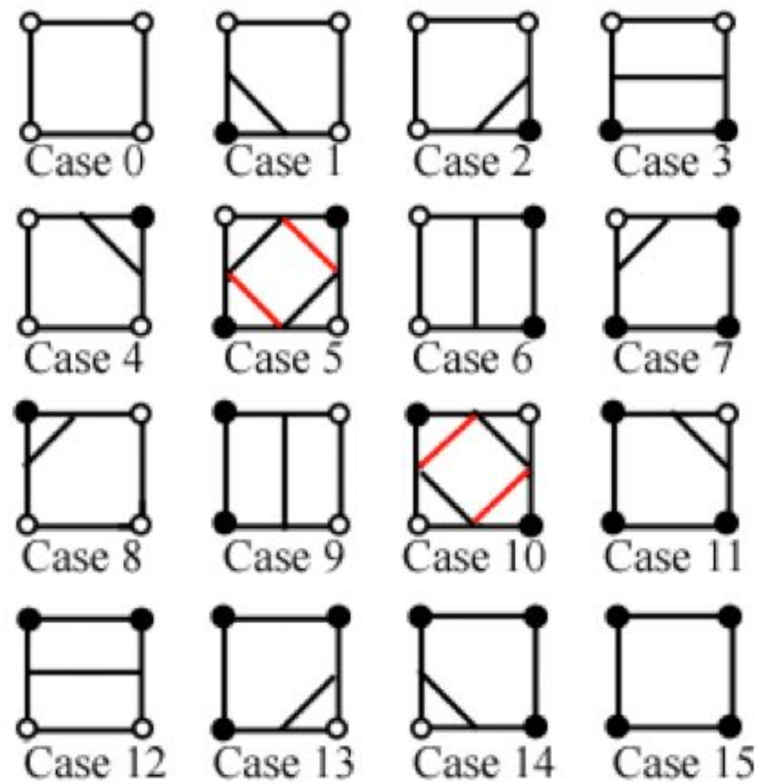


8000 ft iso-contour

Marching Squares (in 2D)



Examine squares connecting 4 pixel centers

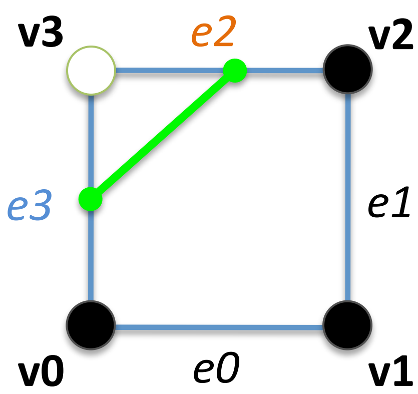


All 16 possibilities

Placement of line segment vertices on the edges done by linear interpolation
(note that Case 5 and Case 10 are ambiguous)

D Lindgrand, U Nice

Marching Squares Algorithm Details



Case 7

square_index is a 4-bit number showing which vertices are black (which of the 16 cases)

square_index =

v3	v2	v1	v0

$= 0111_2$

__ + 4 + 2 + 1 = **7**

edge_table is a pre-defined lookup table for all 16 cases and returns a 4-bit number indicating which of the 4 cube edges are intersected by the contour

edge_table[7] = 12 = 1100_2 =

e3	*e2*	*e1*	*e0*

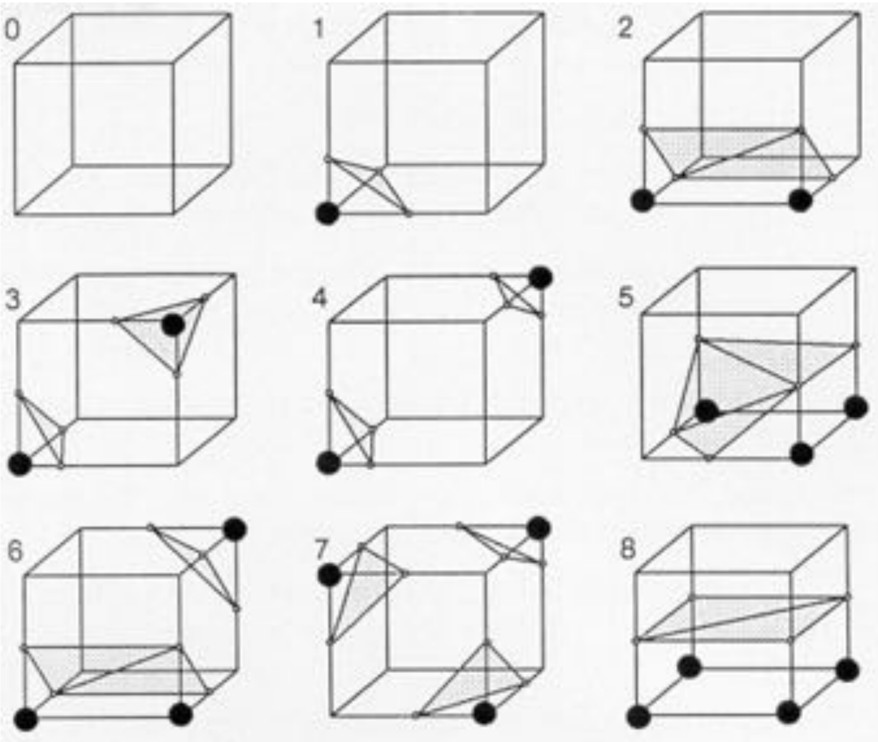
thus *e3* and *e2* are intersected by line segments

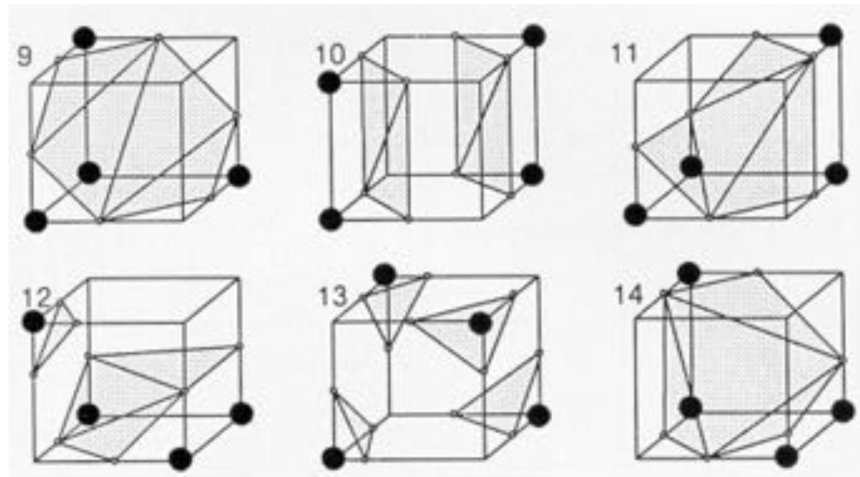
line_table is a pre-defined lookup table of all 16 cases and returns a list of pairs of intersected edges that make line segments

line_table[7] = { **3**, **2**, -1, -1, -1 }

e3-to-e2 is a line segment (2 line segments max; -1 indicates end of list)

Marching Cubes (in 3D)





Lorensen and Cline, Comp Graph 1987

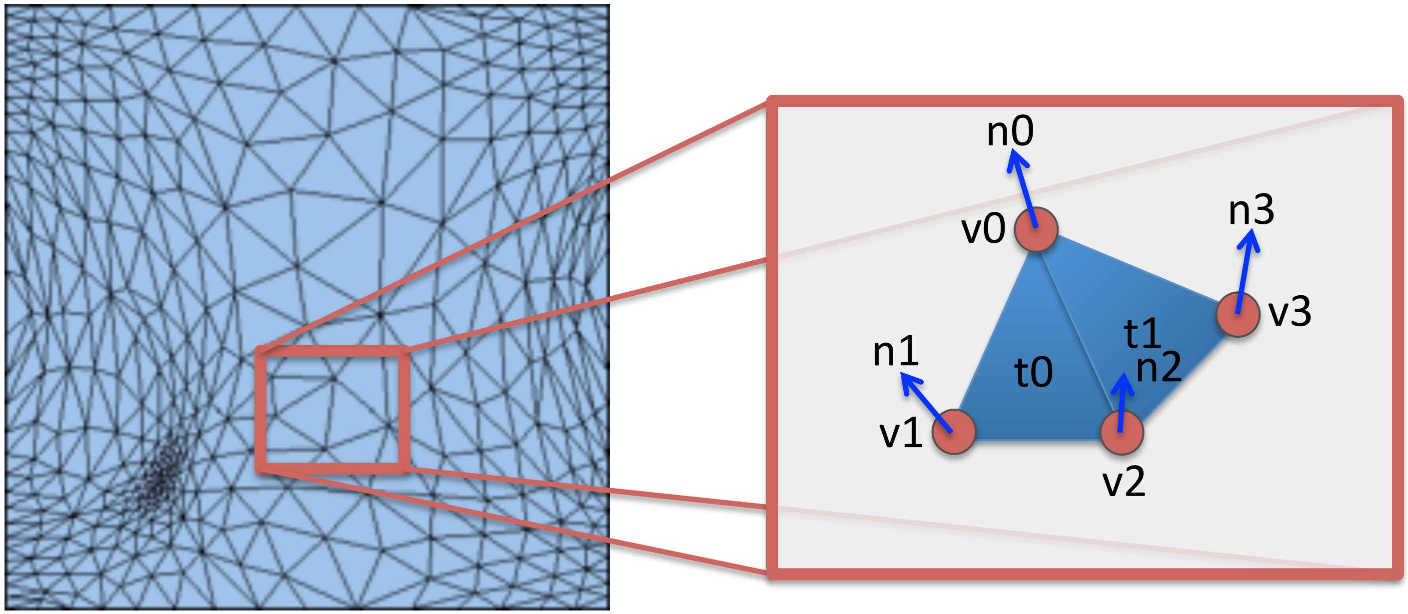
256 cases total
15 unique cases shown here

Cases 3,4,6,7,10,12,13 are ambiguous

cube_index, edge_table and triangle_table are directly analogous to marching squares except:

- there are 256 cube cases (instead of 16 square cases)
- there are 12 cube edges (instead of 4 square edges)
- triangles are triplets of intersected edges (instead of line segments as pairs)
- there is a maximum of 5 possible triangles per cube (instead of max 2 line segments per square)

Mesh Data Structure



Vertex List

v0 = (91.3, 32.4, 14.8)
v1 = (90.1, 31.3, 14.3)
v2 = (91.9, 31.2, 14.9)
v3 = (93.2, 31.8, 14.7)
...

(Must be careful not to redundantly add vertices)

Triangle List

t0 = (v0, v1, v2)
t1 = (v0, v2, v3)
...

(order of vertices determines inside vs. outside direction)

Normal List

n0 = (0.11, -0.08, 0.91)
n1 = (0.13, -0.03, 0.90)
n2 = (-0.03, 0.05, 0.95)
n3 = (0.01, -0.02, 0.99)
...

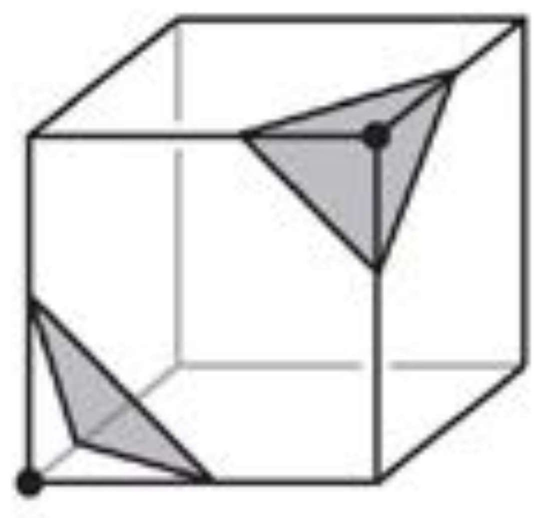
(Marching Cubes doesn't tell you how to calculate normals at each vertex; needed for smooth surface shading)

Questions:

What would be an alternative way to triangulate this case?

How might you choose one vs. the other?

Why might you choose one vs. the other?



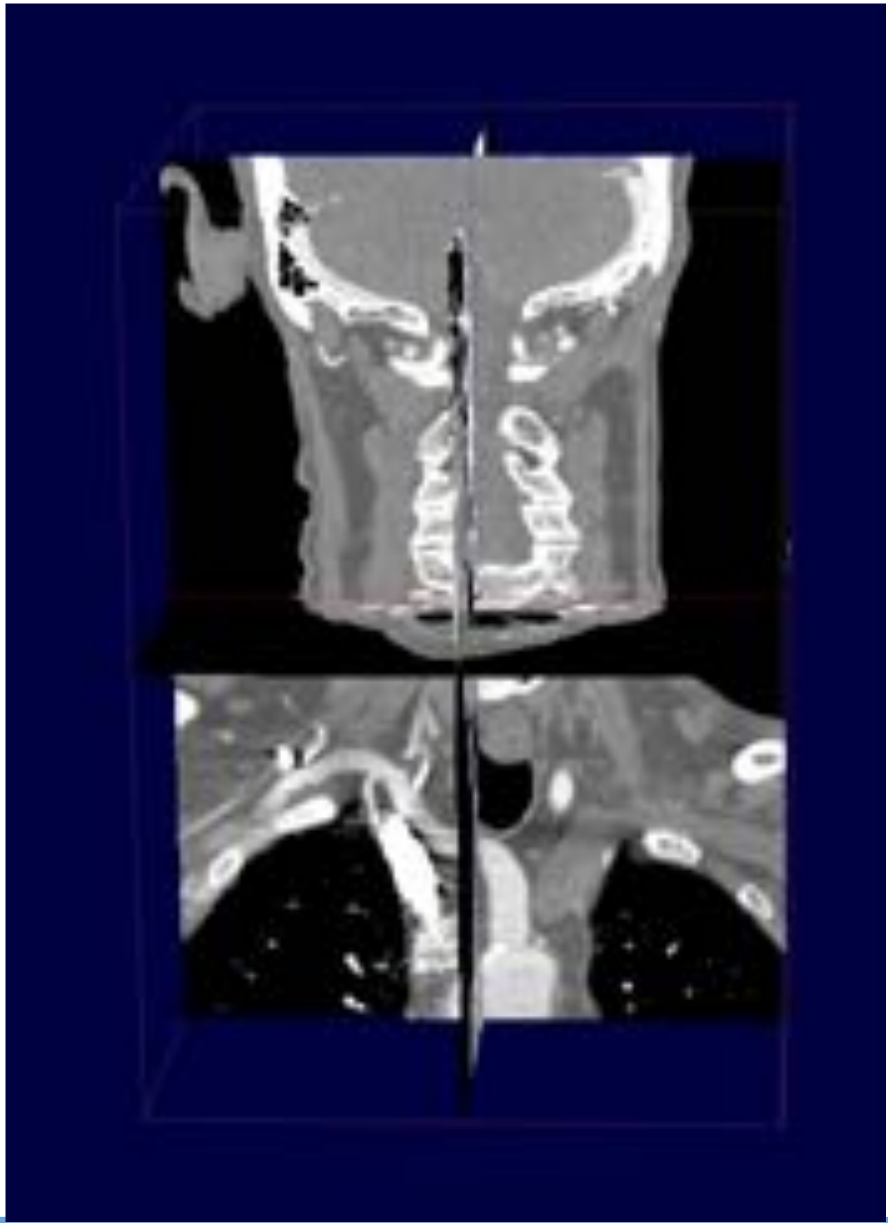
Shaded Surface Display

- Triangle mesh made from images
  - Marching cubes is the classic method but isn't the only method
  - Meshes can be decimated, smoothed, adaptively refined
- Surface mesh can be rendered into an image using standard graphics routines
- Pros and Cons
  - Very fast
  - Surface geometry visualized well
  - Only as good as meshing routine
  - Inner structures obscured

Example rendering of human lungs from CT

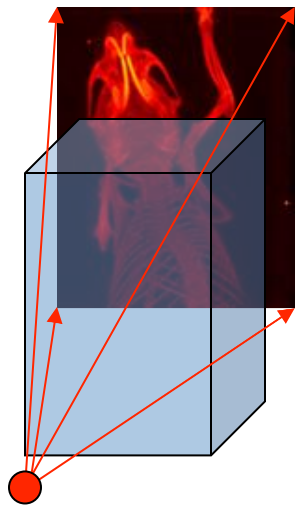
Typical game character: 2k-20k triangles
Typical medical image: 500k-10M triangles

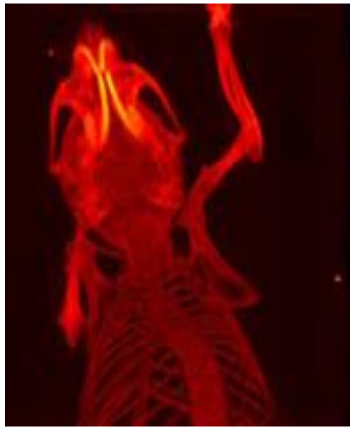
Shaded Surface Display



Maximum/Minimum/Mean Intensity Projection

- Rays are mathematically cast through the 3D image and the maximum/minimum/mean (interpolated) intensity encountered is put into that 2D image pixel
- Rays may be divergent for perspective or parallel for an orthographic view
- Viewpoint may be rotated around dataset
- Pros and Cons
  - Bright objects well visualized
  - May have overlap (e.g., spine & aorta)
  - Simple, fast, pseudo-3D
  - Rendered 2D image is semi-quantitative



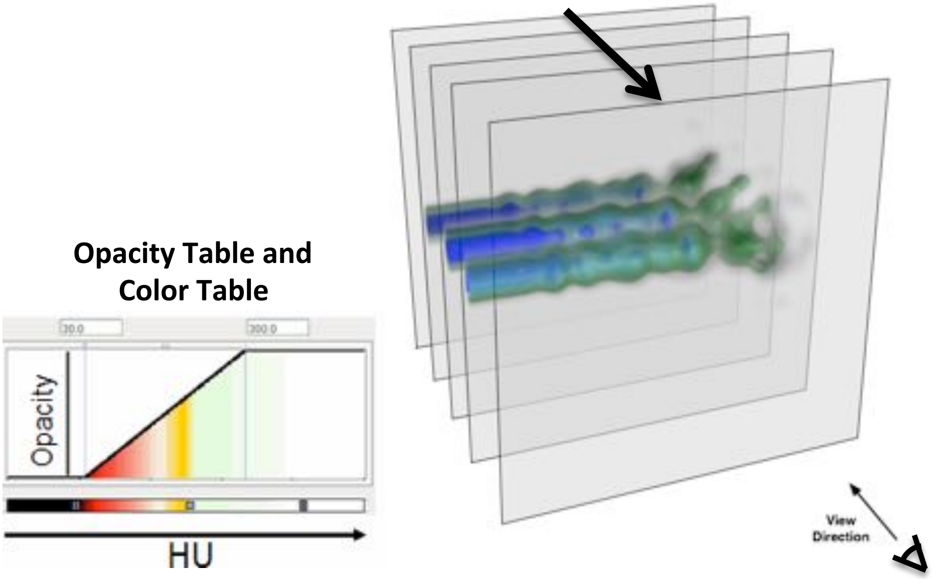


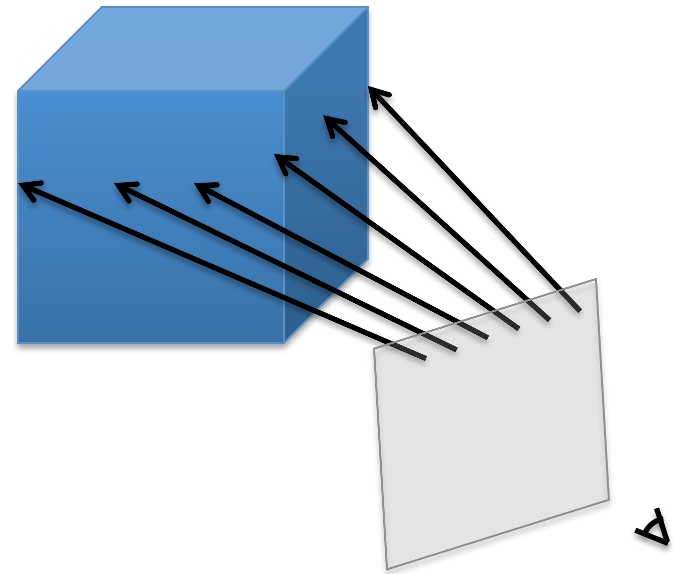
Maximum/Minimum/Mean Intensity Projection

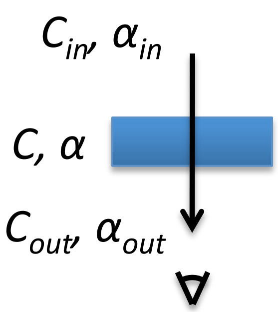
Mean Intensity Projection

Maximum Intensity Projection

Direct Volume Rendering

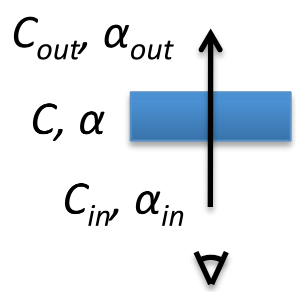






$$C_{out} = (1-\alpha)\cdot C_{in} + C$$
$$\alpha_{out} = (1-\alpha)\cdot \alpha_{in} + \alpha$$

Object Order Volume Rendering (back-to-front)

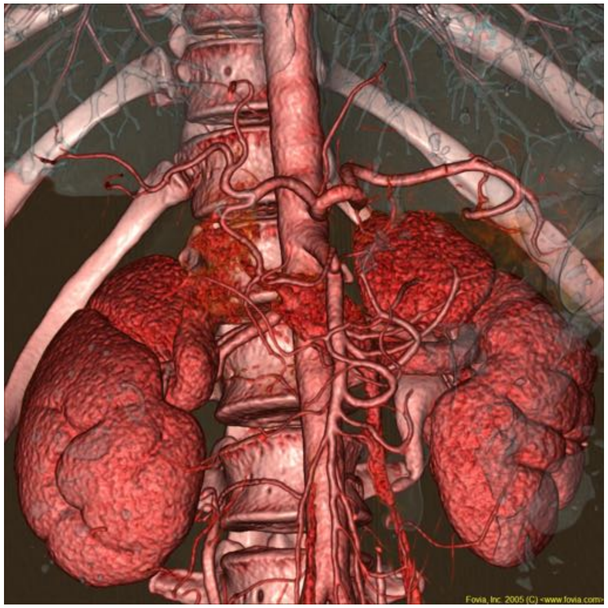


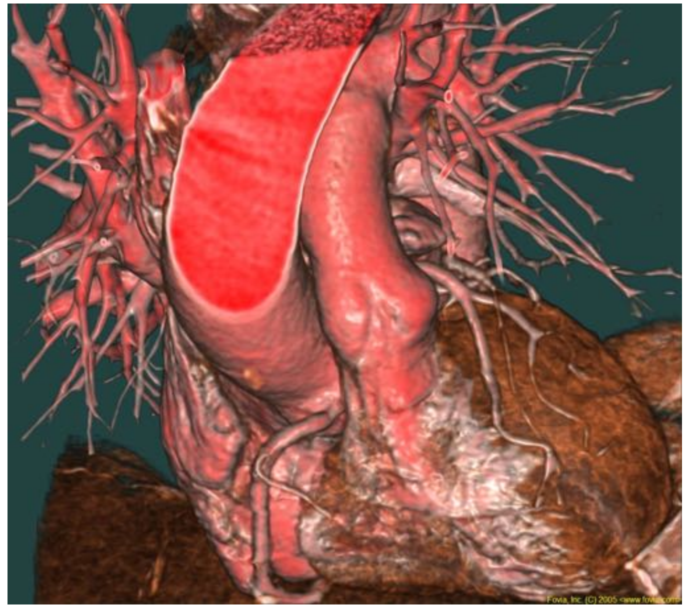
$$C_{out} = C_{in} + (1-\alpha_{in})\cdot C$$
$$\alpha_{out} = \alpha_{in} + (1-\alpha_{in})\cdot \alpha$$

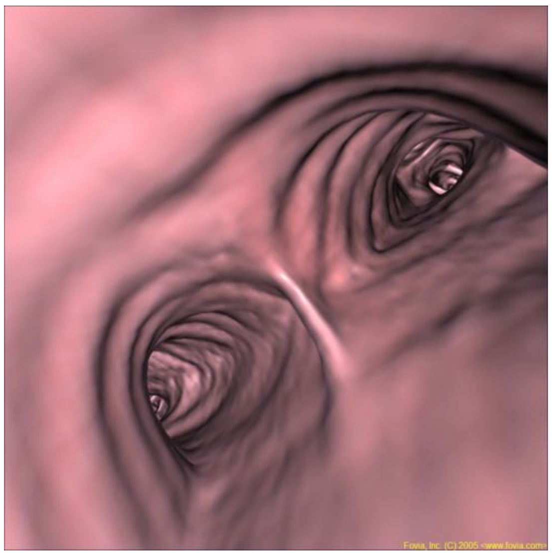
Image Order Volume Rendering (front-to-back)

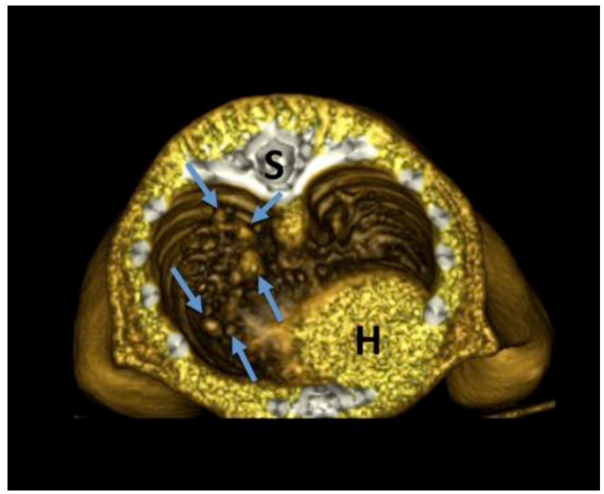
http://pmod.com
http://www.equalizergraphics.com/

Volume Rendering









Fovia.com
Osirix

Volume Rendering

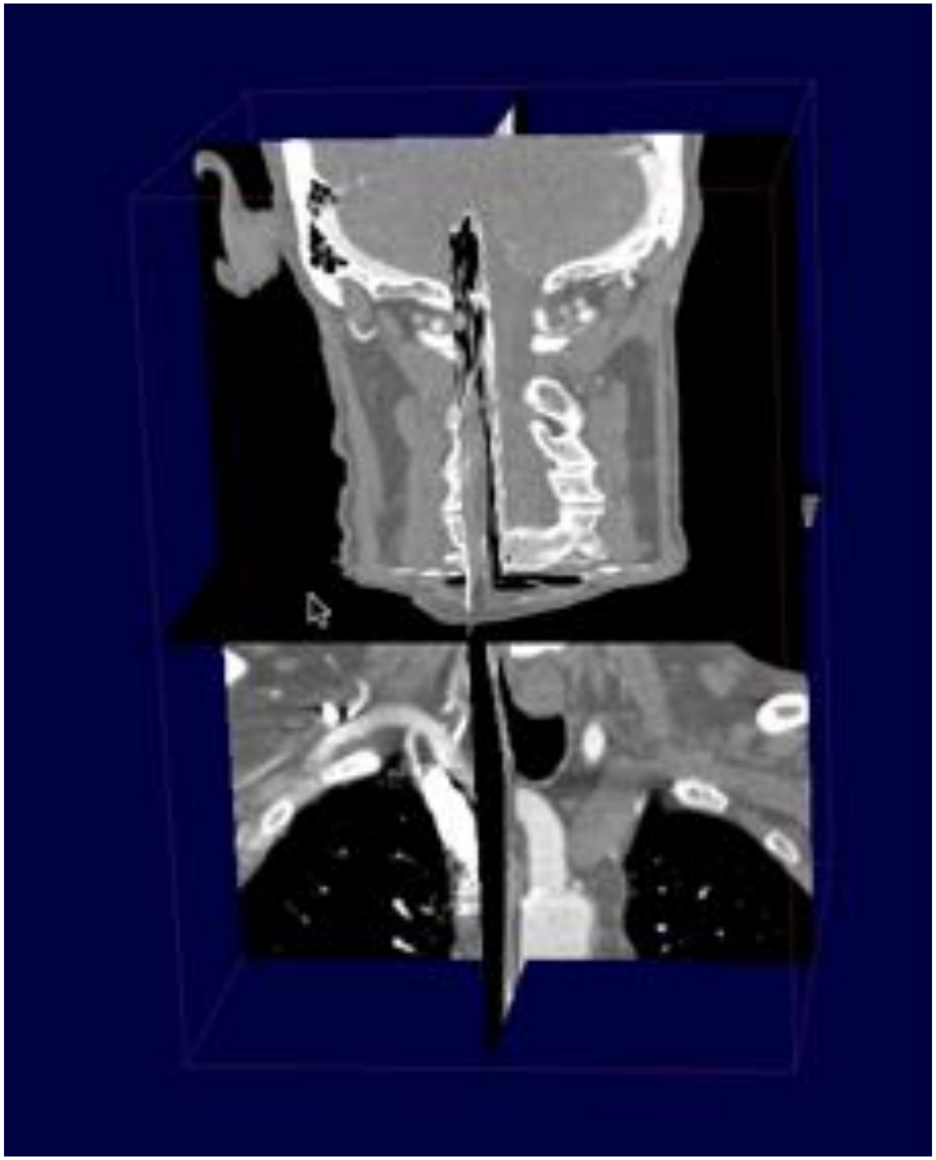
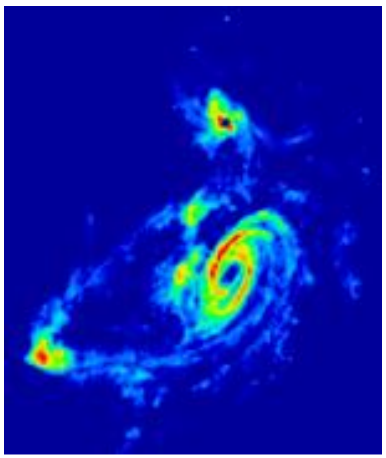
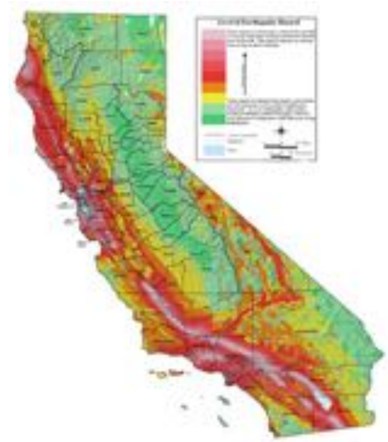


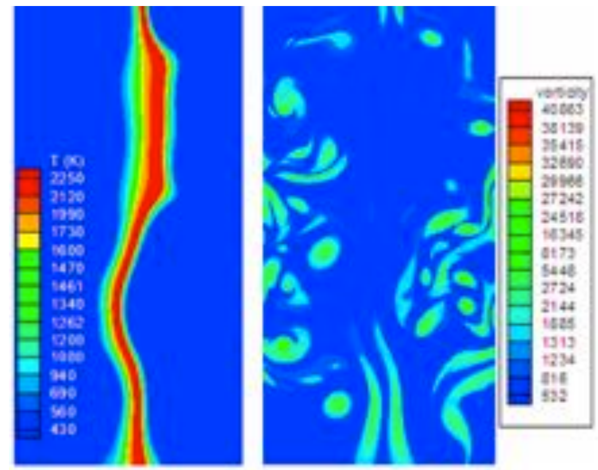
Image Fusion

Image Fusion

- Pseudocoloring used in many fields
  - Assigns 3-component color to 1-component scalar data by using a color lookup table
  - Astronomy, geography, fluid simulations, etc.

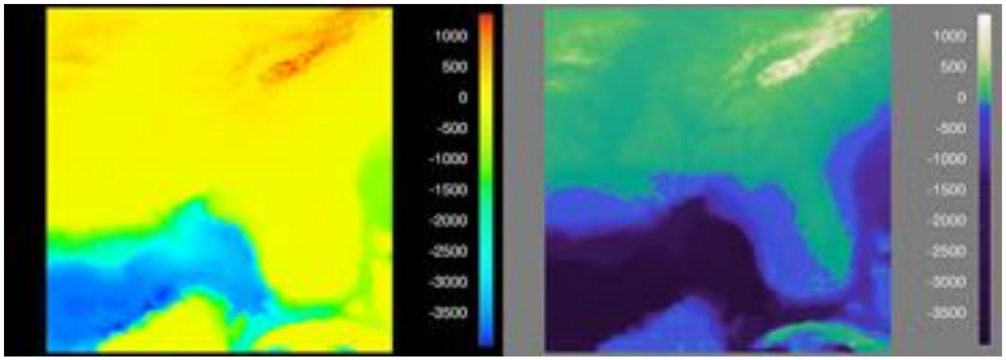




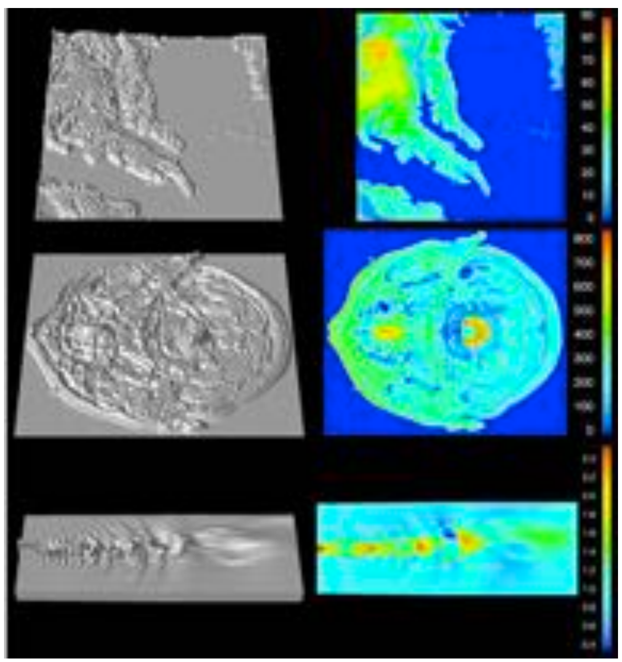


- What kind of information is to be revealed?
  - Metric: quantity at each point
  - Form: shape and structure
- Combine anatomic (e.g., CT) and functional (e.g., PET)

Pseudocoloring Artifacts

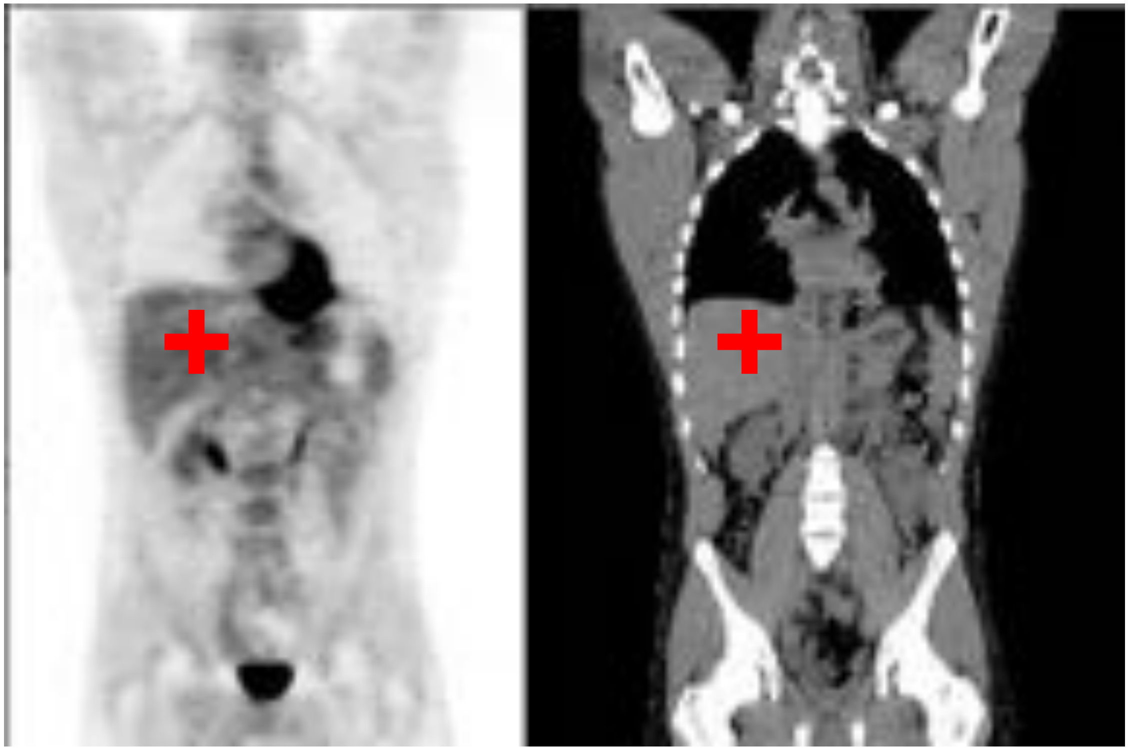


False Negative Artifacts



False Positive Artifacts

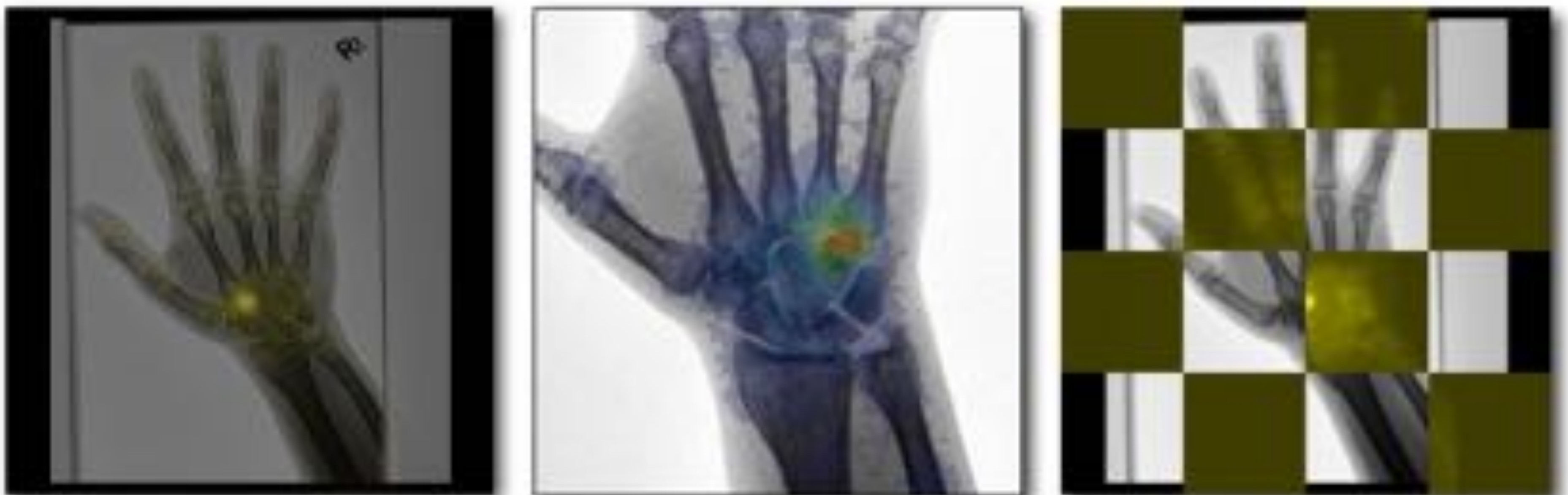
Parallel Display



Linked cursor to match up locations

Alternate Pixel Display

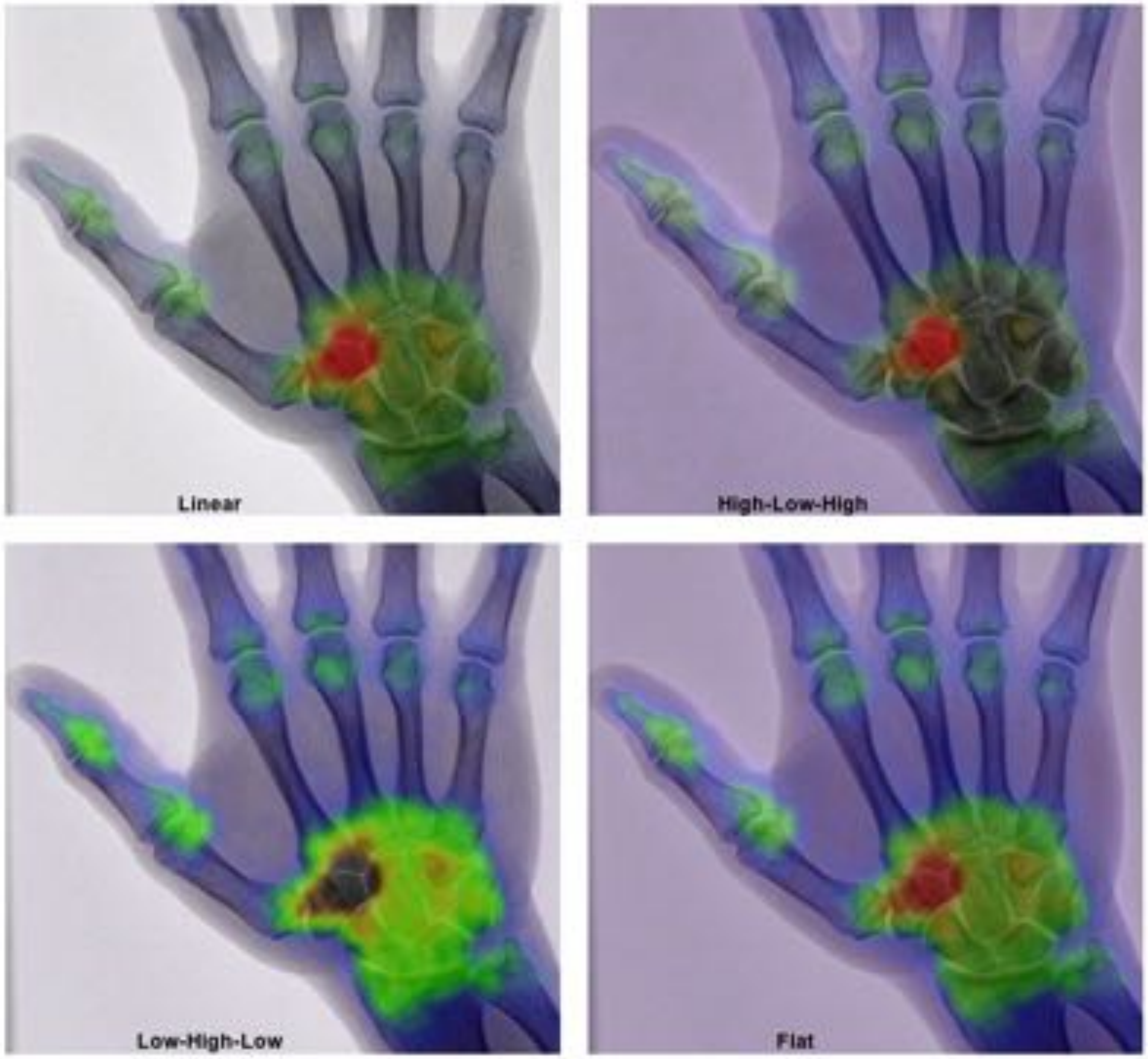
"Screen Door Transparency"



Generated by Kieran Maher, Nov 2006

Varying the size of the checkerboard to get different effects

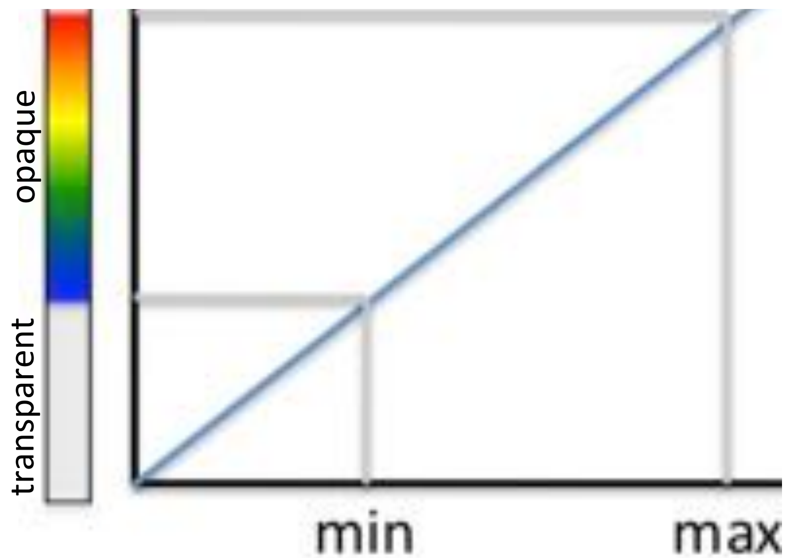
Alpha Blending

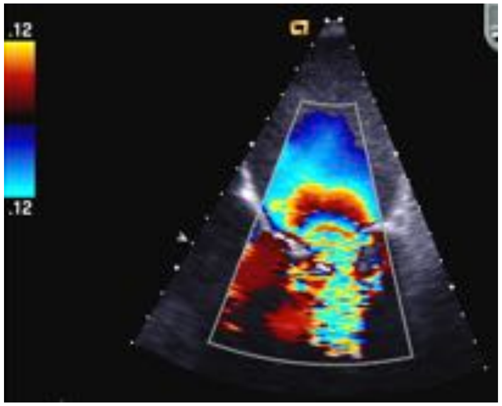


$$I = \alpha I_1 + (1-\alpha) I_2$$

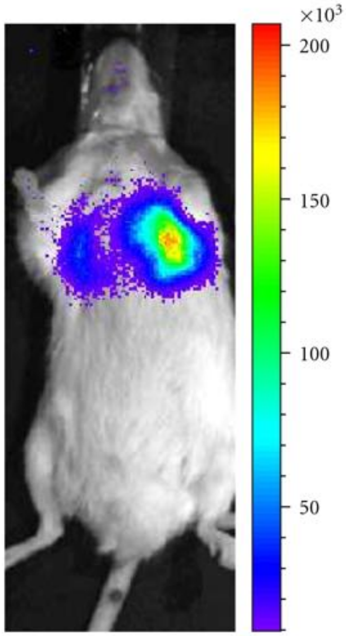
Generated by Kieran Maher, Nov 2006

Alpha Blending





Doppler Ultrasound



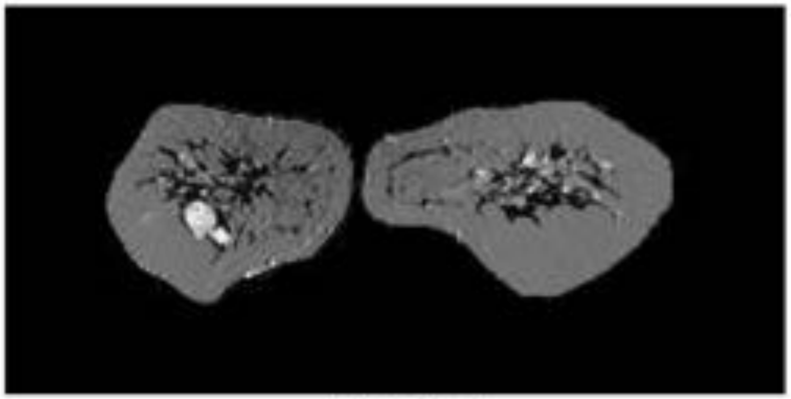
BLI

Chen et al, Circulation 2011

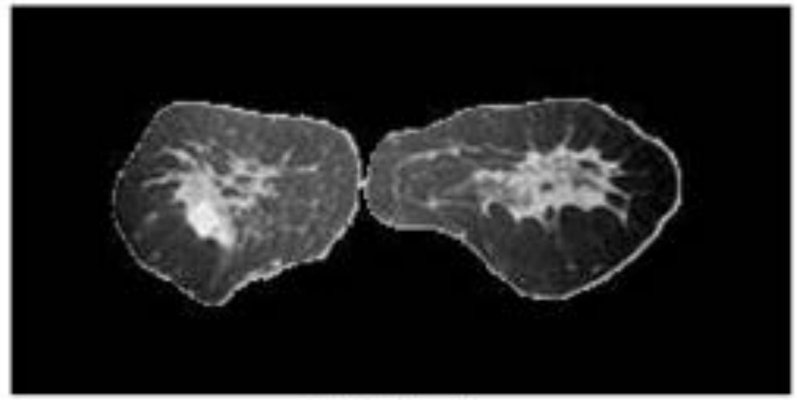
α (opacity or 1-transparency) can be a function of pixel intensity

Typically, functional information shown in color, overlaid on anatomy in grayscale

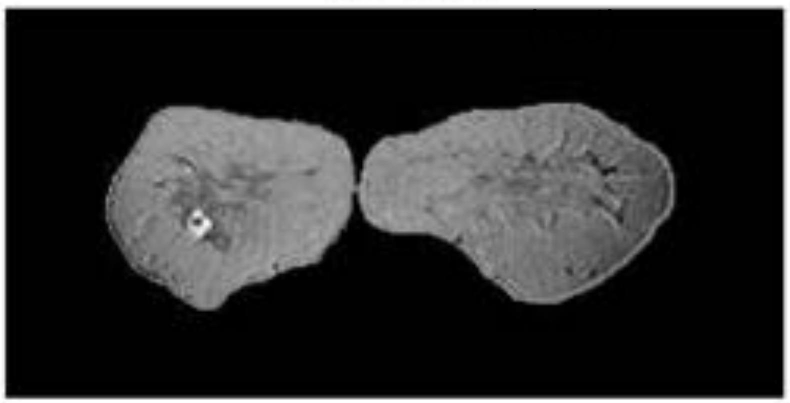
RGB Fusion



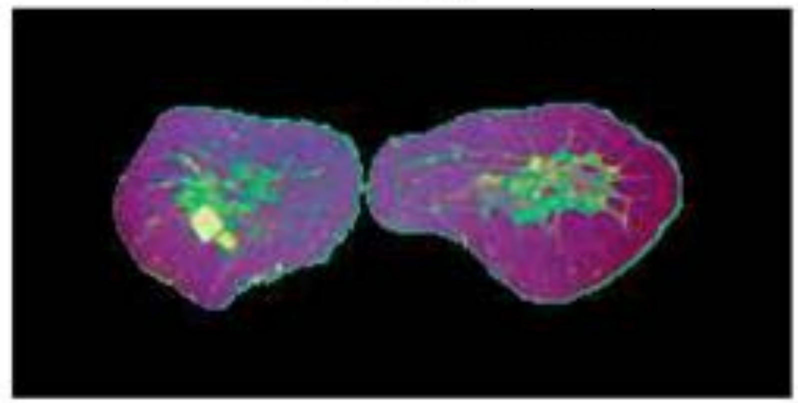
KPCA I_1 (red)



KPCA I_2 (green)



KPCA I_3 (blue)

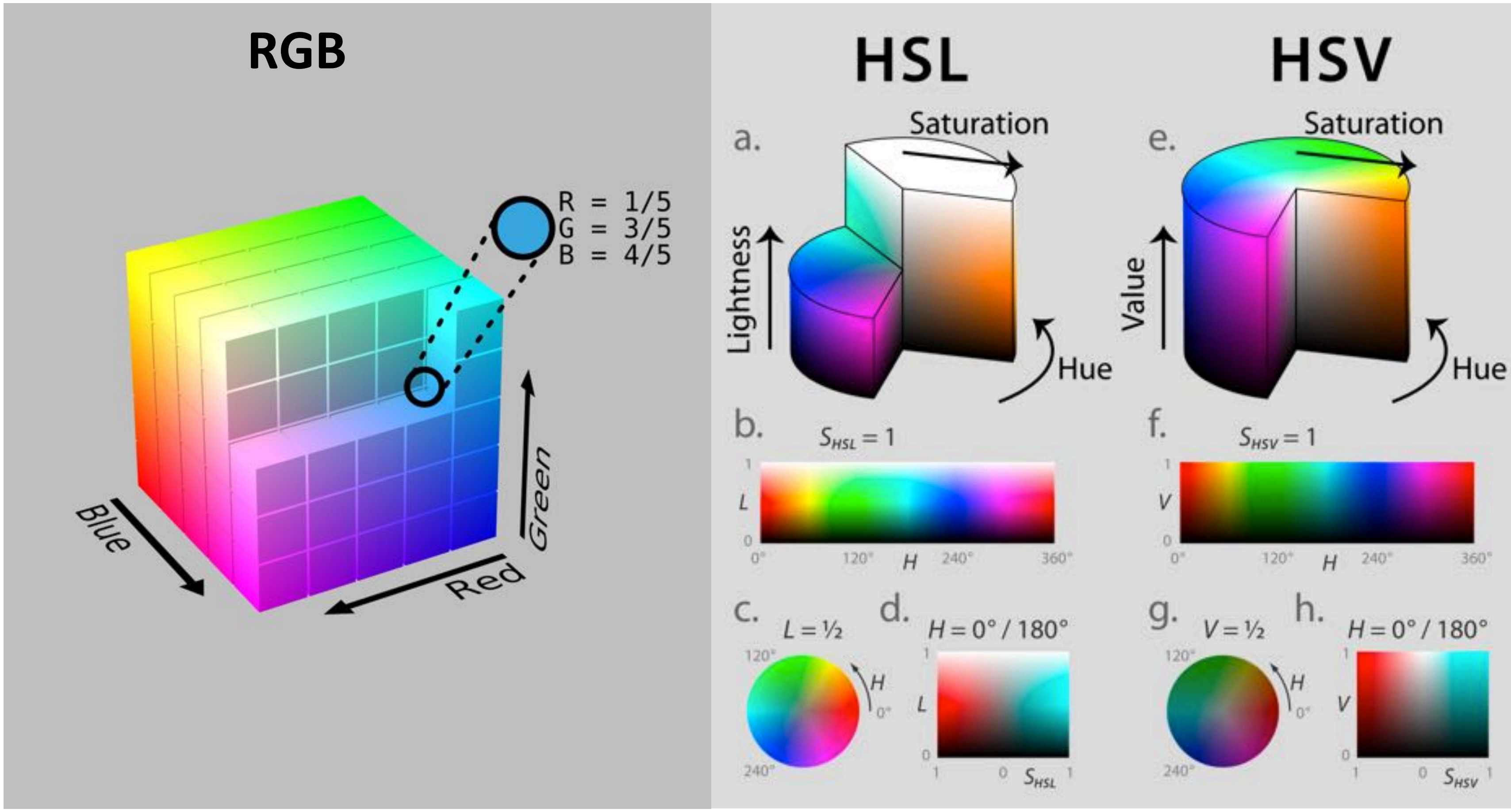


KPCA I_{RGB}

Twellmann et al., Biomed Eng Onl 2004

Assigning a color channel (red, green, blue) to each of three images but perception of three channels is intertwined

Color Spaces



wikipedia.org

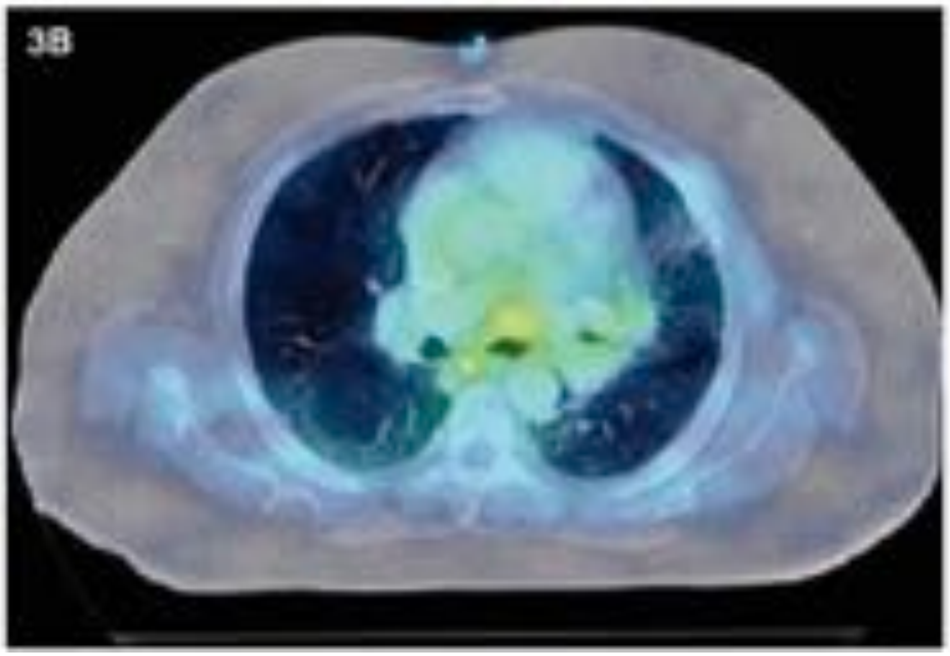
Red
Green
Blue

Hue
Saturation
Lightness

Hue
Saturation
Value

Lightness/Hue Encoding

- CT rendered in lightness channel
- PET rendered in hue channel

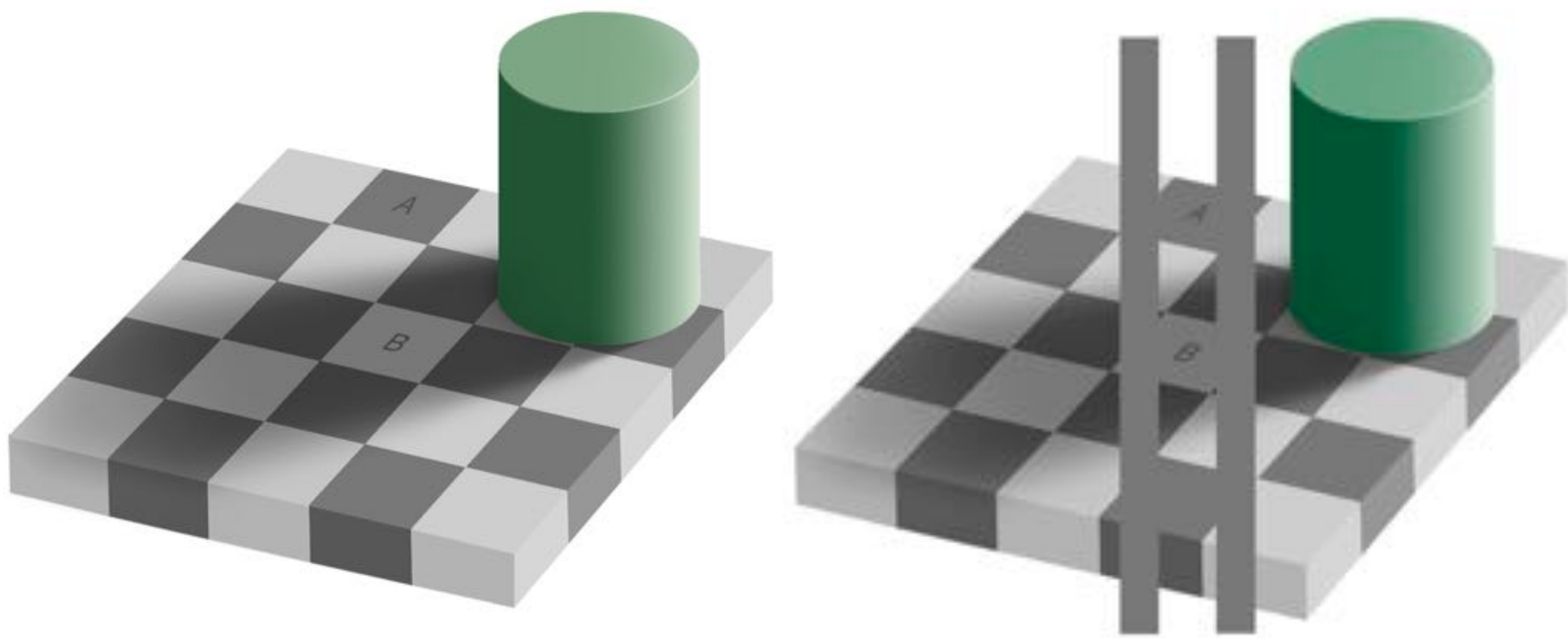


Thomas et al., Mol Im Bio 2003

Assumption: Lightness and hue can be perceived more or less independently (at least better than RGB)

Visual Perception

Perception of Luminosity



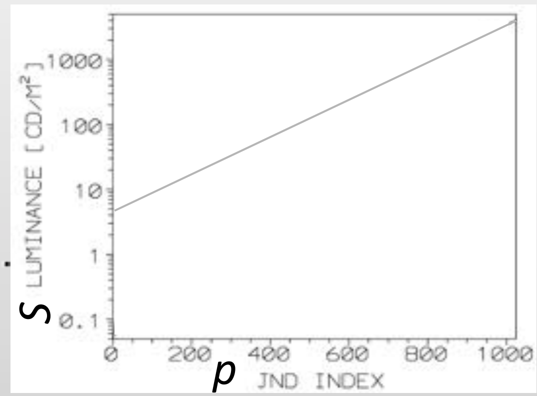
Edward Adelson
wikipedia.org

Perception of Luminosity

Weber Fechner Law

$$dp = k\frac{dS}{S},$$

$$p = k \ln \frac{S}{S_0}.$$

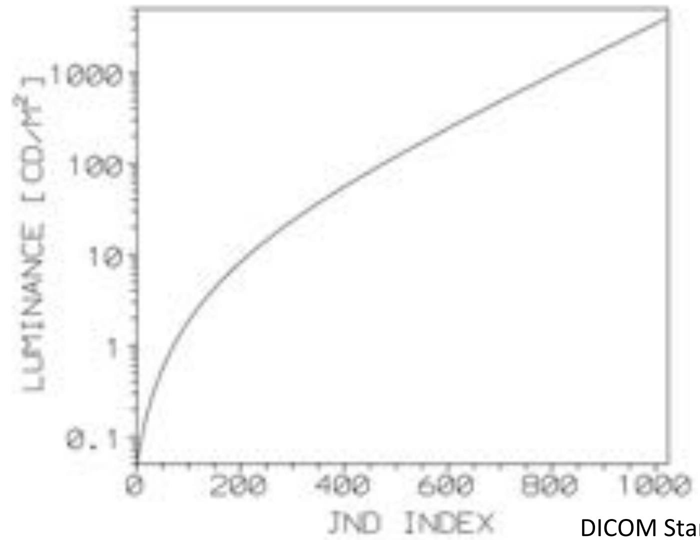


Minimally perceptible (p) difference proportional to a percentage of overall stimulus (S)

First order approximation but it doesn't hold at low stimulus

DICOM GSDF

(Digital Imaging and Communications in Medicine Grayscale Standard Display Function)



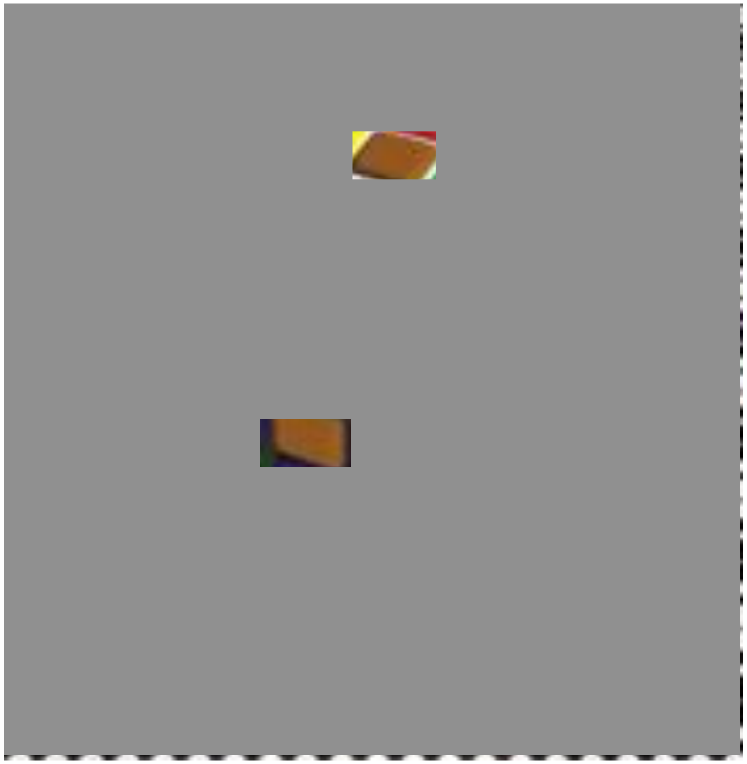
DICOM Standard 3.14

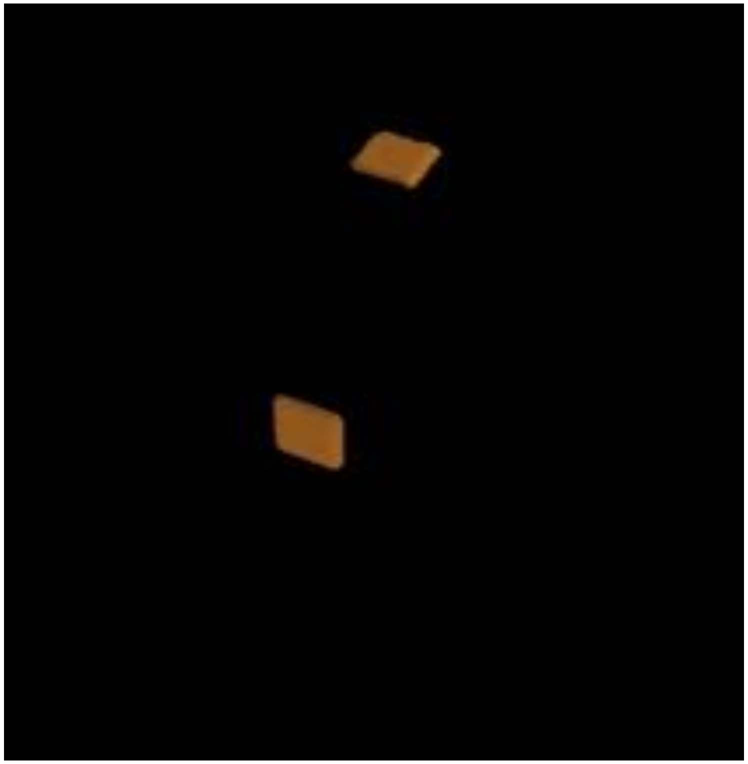
$$\log_{10} L(j) = \frac{a + c\cdot Ln(j) + e\cdot (Ln(j))^2 + g\cdot (Ln(j))^3 + m\cdot (Ln(j))^4}{1 + b\cdot Ln(j) + d\cdot (Ln(j))^2 + f\cdot (Ln(j))^3 + h\cdot (Ln(j))^4 + k\cdot (Ln(j))^5}$$

JND = Just Noticeable Difference

DICOM GSDF is used to calibrate clinical image displays to transform pixel values to a perceptually uniform gamut of grayscale values

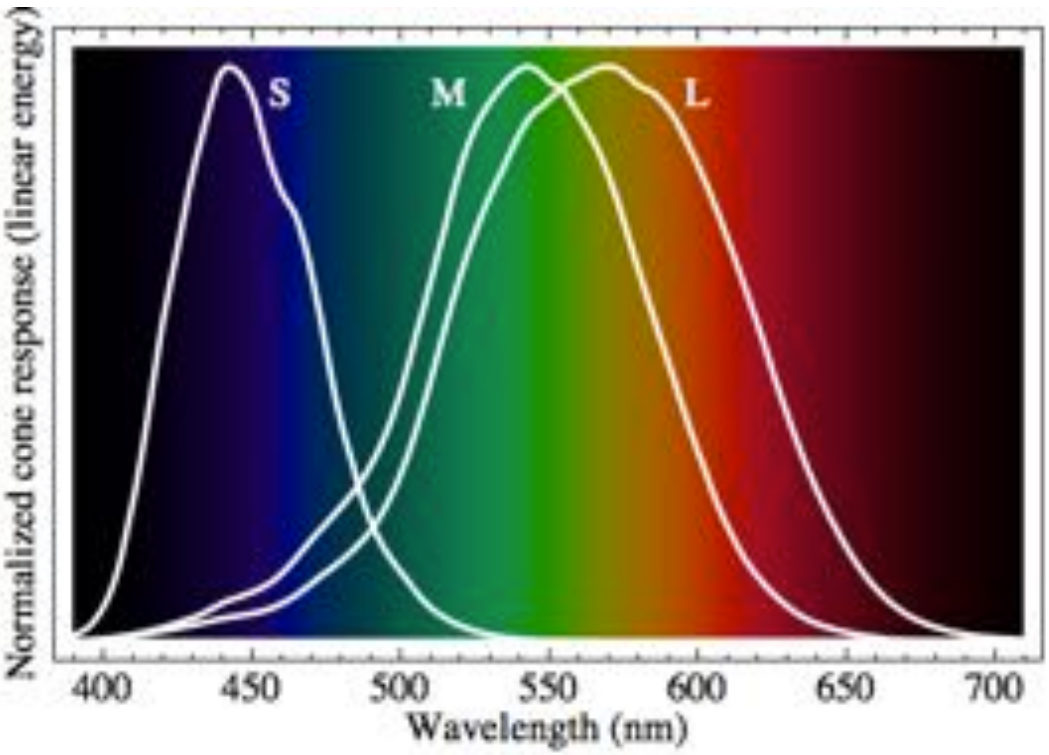
Perception of Color



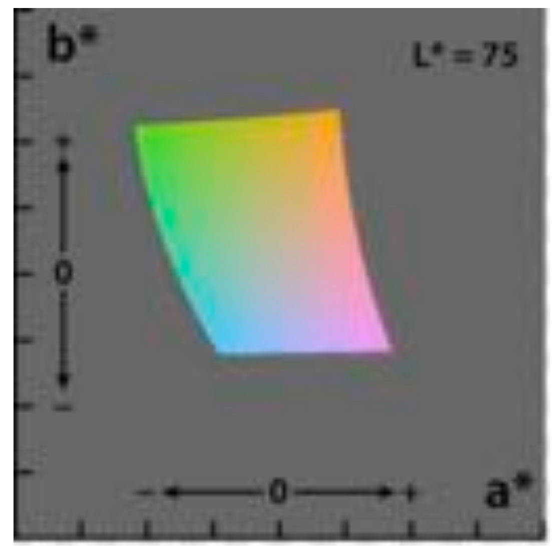


http://www.moillusions.com/2008/02/color-tile-illusion-new-aspect.html

Perception of Color



Human trichromat vision



CIE L*a*b* color space
approximates perceptual uniformity

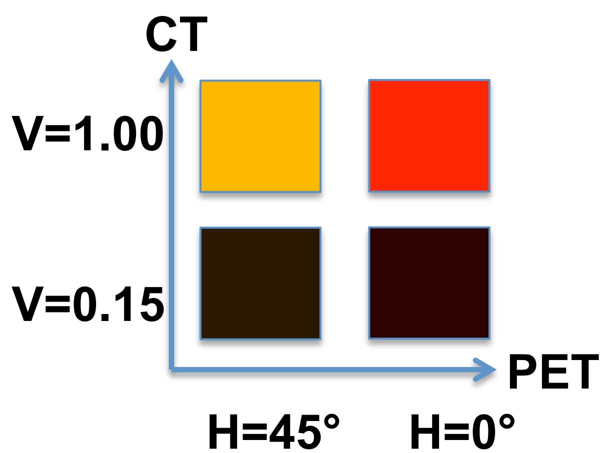
(Note: spectral properties of visible light is not inherently limited three degrees of freedom, this is just a limitation of the human visual system)

wikipedia.org

Bezold-Brücke Effect

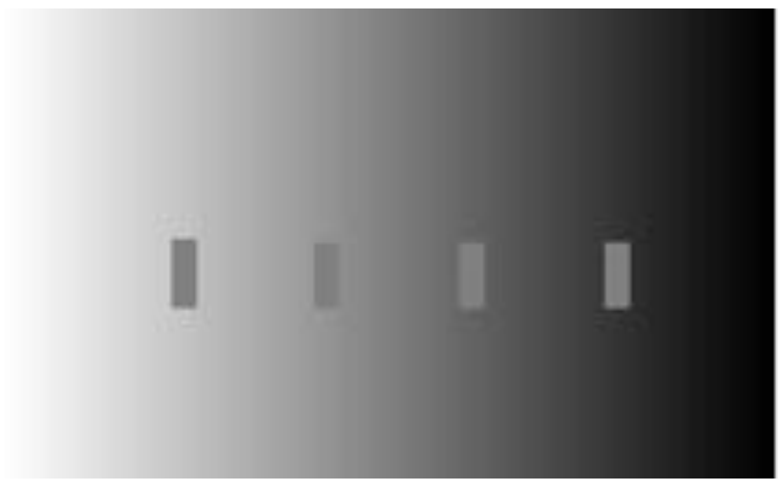
ca. 1874

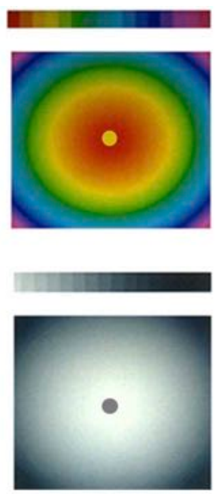
- Perception of lightness and hue are not independent
  - Still better than RGB
- As lightness changes (at constant hue), the perception of hue changes
  - Very difficult to determine the hue of a nearly black pixel
  - e.g., perceived PET value depends on the underlying CT value

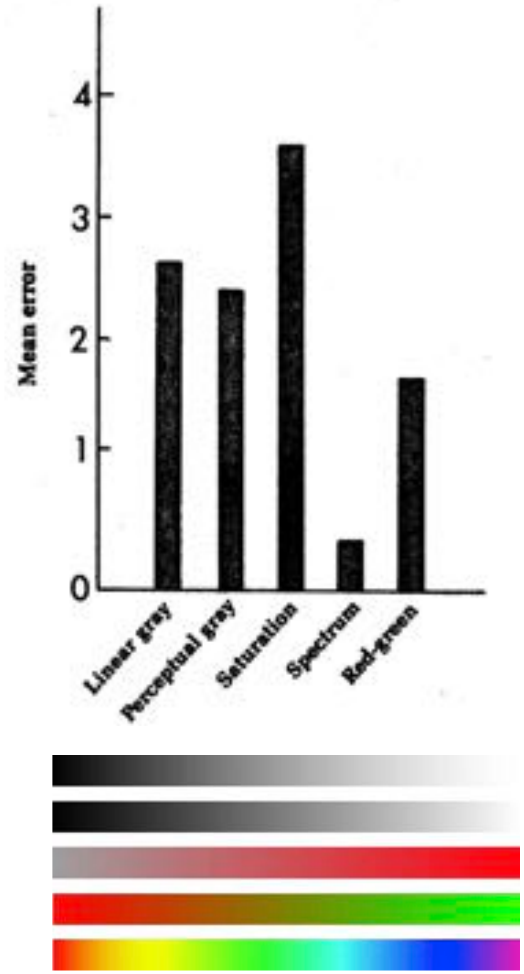


Metric Information

- Simultaneous Contrast



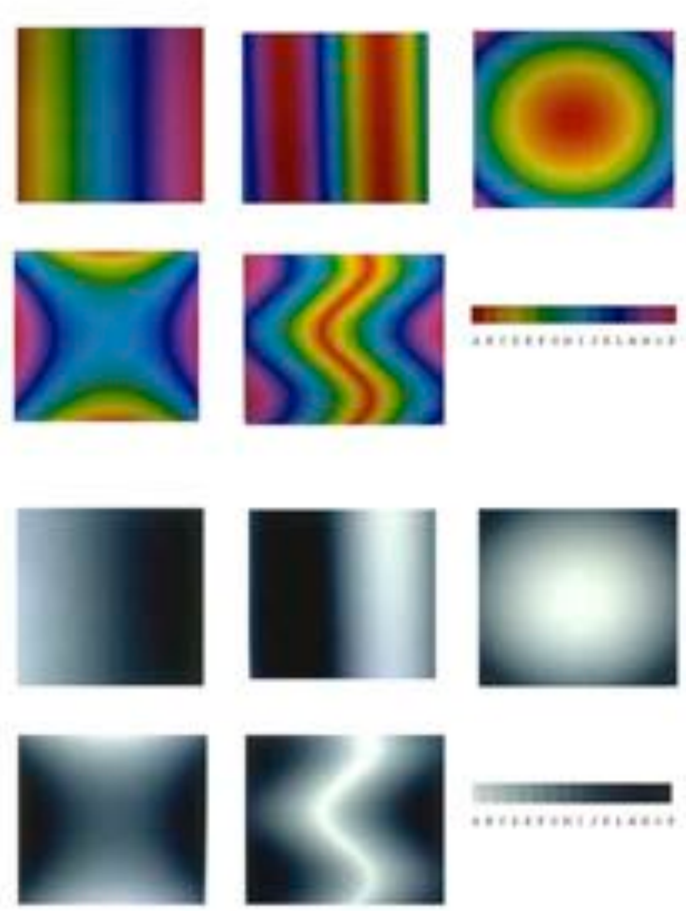




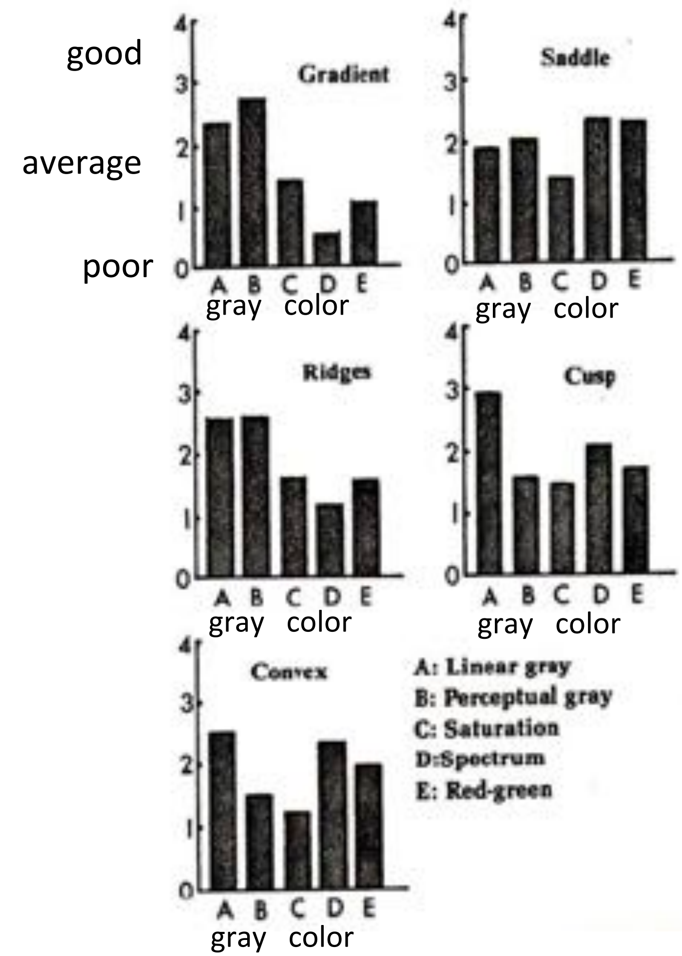
Metric quantities best shown by hue (e.g., spectrum)

Ware et al., IEEE CG&A 1988

Form Information



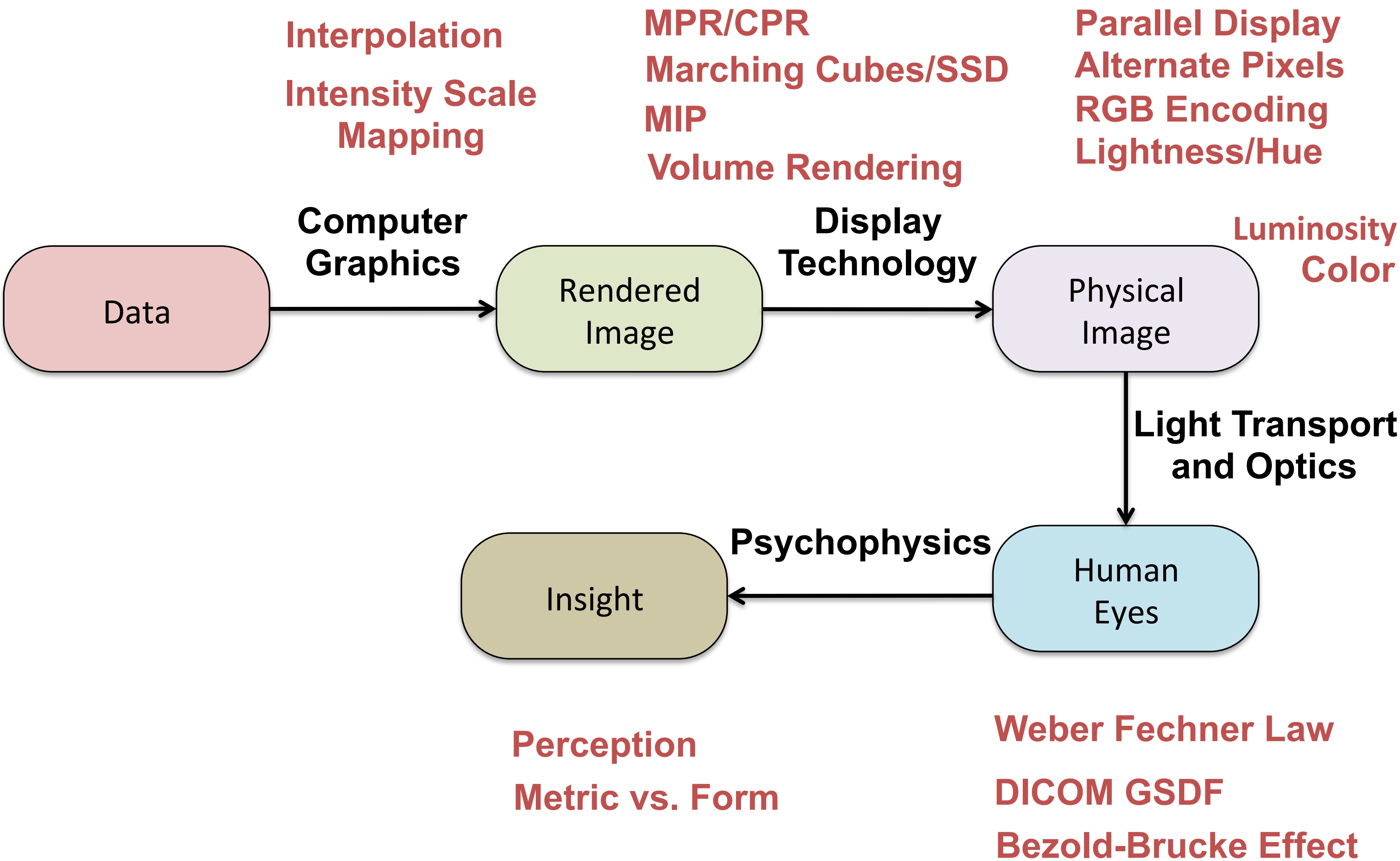
“How effective was the color sequence?”



Form best shown by luminance (e.g., grayscale)

Ware et al., IEEE CG&A 1988

Visualization: From Data to Insight



What does it mean for you?

- You now understand the pipeline from an array of pixel values to human insight
- Visualization covers a wider topics than just computer graphics
- Human perception is a critical consideration

Next Lecture:

Image Segmentation