Biomedical Informatics 260

Visualization of Medical Images

Lecture 2

David Paik, PhD

Spring 2019

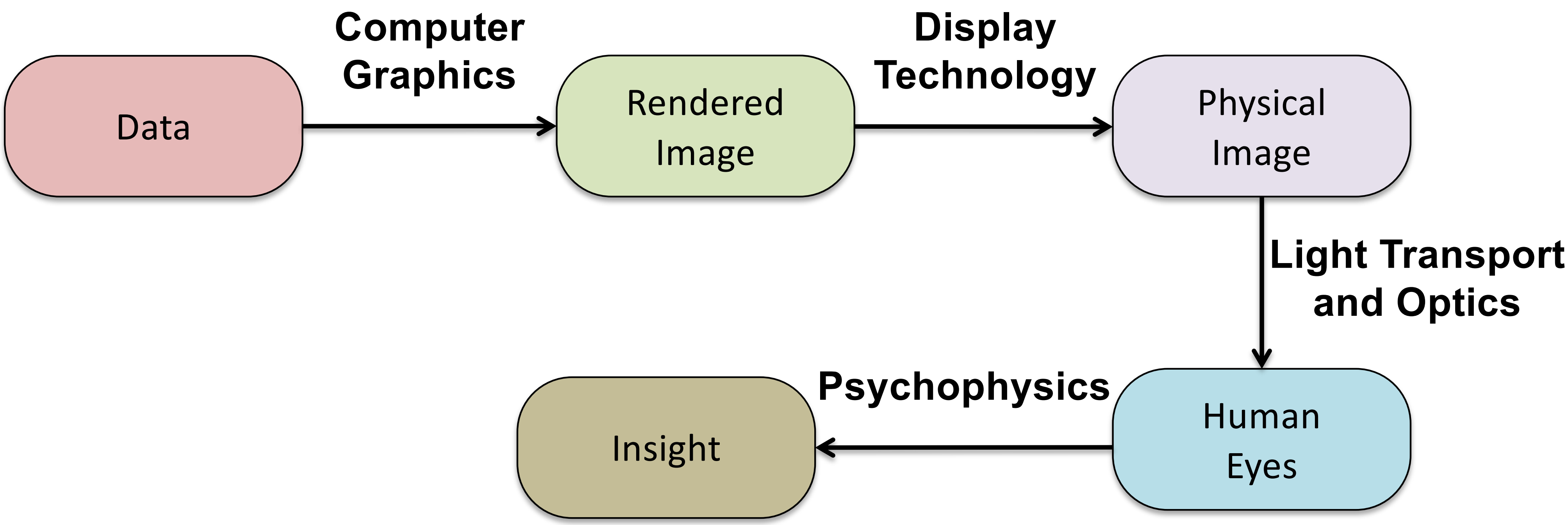
Disclaimer

- Dr. Paik is Chief Scientific Officer for and has a financial interest in Sirona Medical Inc.
  - No topics discussed in this course are intended as an endorsement of *any* commercial product or service

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
  - Even at large scale
- Topics covered today:
  - Medical Image Data
  - 2D Computer Graphics
  - 3D Computer Graphics
  - Image Fusion
  - Psychophysics

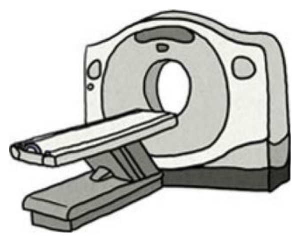
Visualization: From Data to Insight



Medical Image Data

DICOM Model of the Real World

As larger scale analyses are becoming more common, leveraging existing data models is increasingly important!



Single-frame

or

Multi-frame

Patient

Study 1
Study 2
Study 3
…

Series 1
Series 2
Series 3
…

Image 1
Image 2
Image 3
…

Unique patient matching is an unsolved problem

Prior studies may be important for longitudinal analysis

Can include:
Localizer/scout
Scanned doc

Usually a 2D array
Can be multi-frame 3D array

Patient ID
Patient Name
Birth Date

Study UID
Accession #

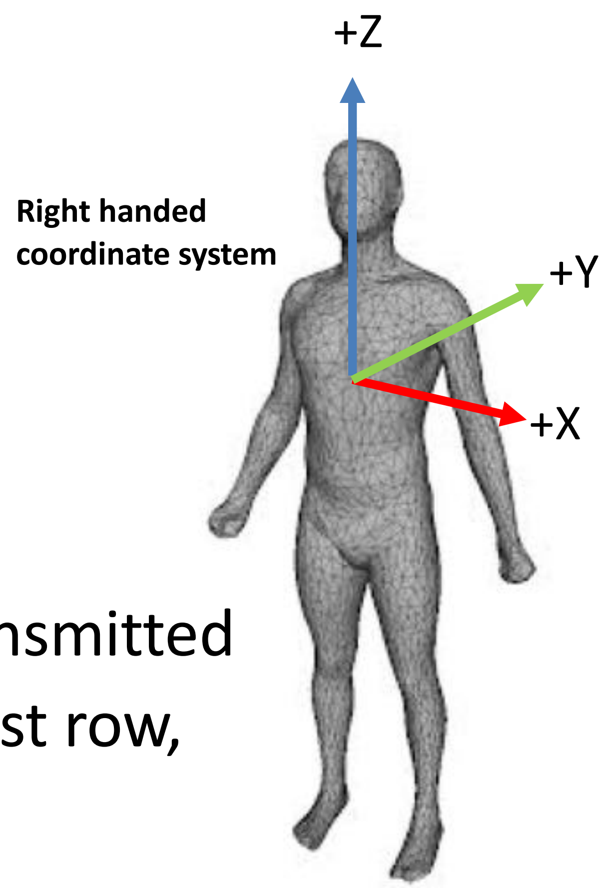
Series UID
Series #

Instance UID
Instance #

Useful tidbit: UIDs are worldwide unique identifiers

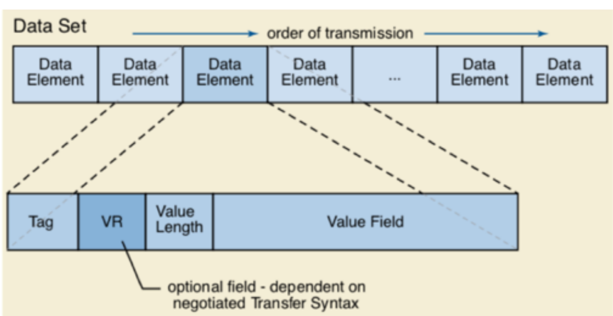
DICOM Coordinates

- "LPS" anatomic coordinate system
  - +X is patient left
  - +Y is patient posterior
  - +Z is patient superior
- Each image has
  - Image Position – (x,y,z) of first voxel transmitted
  - Image Orientation – (x,y,z) vectors of first row, first column spatial direction
    - *Not always axis aligned!*
- If Frame of Reference UIDs are shared:
  - Images are in the same coordinate system
  - Can be visualized/navigated together



DICOM Information Model

- **IOD** ∈ { CR Image, CT Image, Enhanced CT Image, … }
  - **IE** ∈ { Patient, Study, Series, Equipment, Frame of Reference, Image, … }
    - **Module** ∈ { Image Plane, Image Pixel, CT Image, … }
      - **Data Element** ∈ { (0008,0008) Image Type, (0018,0060) KVP, … }



- **Tag:** (Group#,Element#)
- **VR:** data type (can be implicit or explicit)
- **Value Length:** byte length (can be undefined)
- **Value Field:** the data (ASCII or binary depending on VR)

Note that one VR type is Sequence (SQ) that allows for nested data elements

DICOM Files

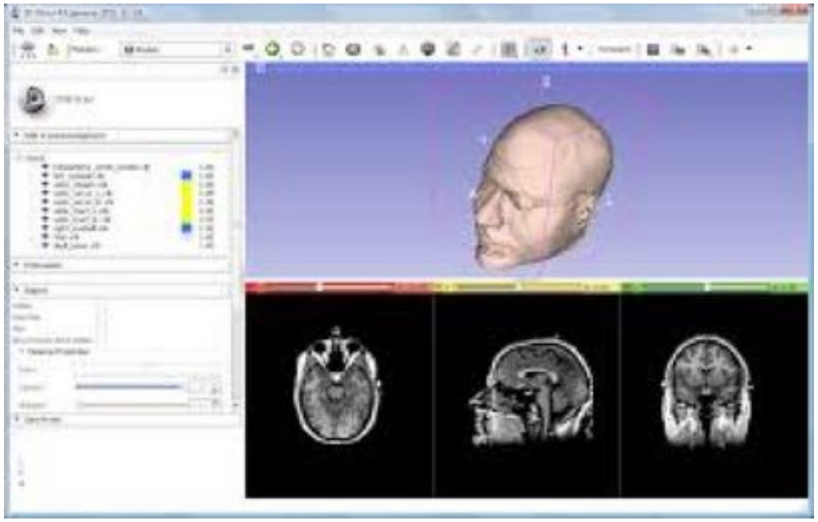
```
Preamble: 128 Bytes, usually all 0

Prefix: "DICM"

Meta Information:
(0002,0002) UI [1.2.840.10008.5.1.4.1.1.4]                        # 26,1 Media Storage SOP Class UID
(0002,0003) UI [1.3.6.1.4.1.14519.5.2.1.4792.2001.3694…4]         # 64,1 Media Storage SOP Instance UID
(0002,0010) UI [1.2.840.10008.1.2.1]                              # 20,1 Transfer Syntax UID
(0002,0012) UI [1.2.40.0.13.1.1]                                  # 16,1 Implementation Class UID
(0002,0013) SH [dcm4che-1.4.27]                                   # 14,1 Implementation Version Name

Dataset:
(0008,0005) CS [ISO_IR 100]                                       # 10,1-n Specific Character Set
(0008,0008) CS [ORIGINAL\PRIMARY\M_SE\M\SE]                       # 26,2-n Image Type
(0008,0014) UI [1.3.6.1.4.1.14519.5.2.1.4792.2001.2086…4]         # 64,1 Instance Creator UID
(0008,0016) UI [1.2.840.10008.5.1.4.1.1.4]                        # 26,1 SOP Class UID
(0008,0018) UI [1.3.6.1.4.1.14519.5.2.1.4792.2001.3694…4]         # 64,1 SOP Instance UID
(0008,0020) DA [20080812]                                         # 8,1 Study Date
(0008,0021) DA [20080812]                                         # 8,1 Series Date
(0008,0022) DA [20080812]                                         # 8,1 Acquisition Date
(0008,0030) TM [081003]                                           # 6,1 Study Time
(0008,0031) TM [082114.43000]                                     # 12,1 Series Time
(0008,0050) SH [5282018218189626]                                 # 16,1 Accession Number
(0008,0060) CS [MR]                                               # 2,1 Modality
(0008,0070) LO [XXXXXXX Medical Systems ]                         # 24,1 Manufacturer
(0008,1032) SQ (Sequence with undefined length)                   # u/l,1 Procedure Code Sequence
  (fffe,e000) na (Item with undefined length)
    (0008,0100) SH [MBBWW ]                                       # 6,1 Code Value
    (0008,0102) SH [BROKER]                                       # 6,1 Coding Scheme Designator
    (0008,0104) LO [MRI Breast Bilateral w and w/o Contrast]      # 46,1 Code Meaning
    (0008,010b) CS [N ]                                           # 2,1 Context Group Extension Flag
  (fffe,e00d)
(fffe,e0dd)
...
```

DICOM is a data/file format *and* a transmission protocol

2D Computer Graphics

Image Viewing Environments

Basic Functionality: Scroll through stacks of 2D images



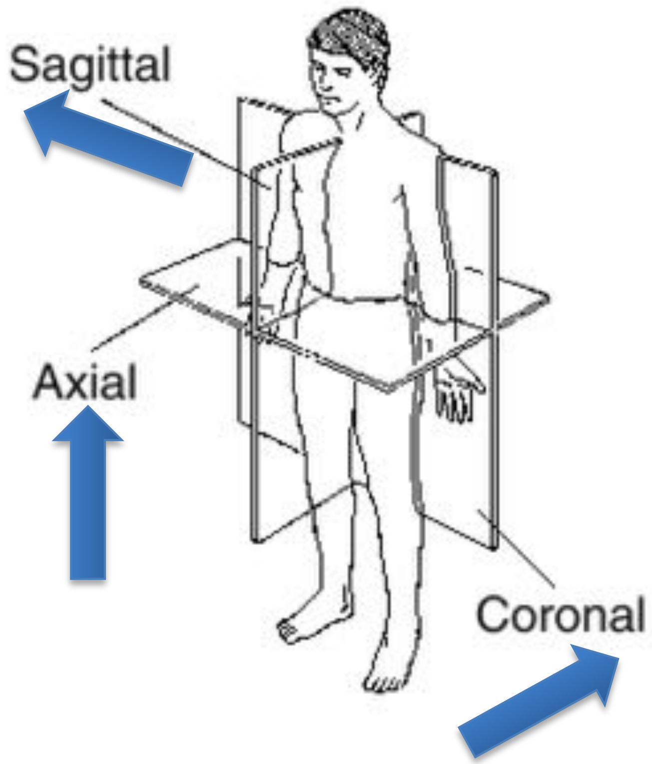
Typical Research Software

Typical Clinical Workstation

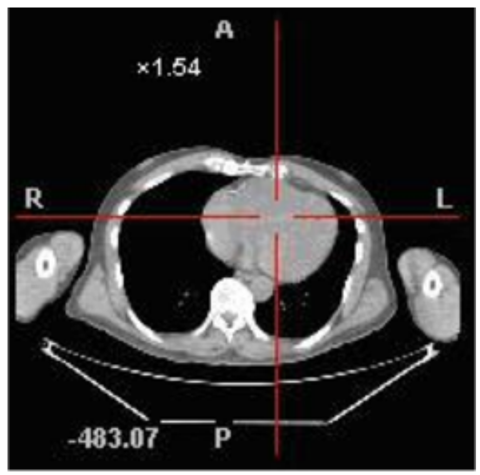
RIS = Radiology Information System (patient info, worklists)
PACS = Picture Archiving and Communication System (image DB, viewing)

Multiplanar Reconstruction (MPR)

Appropriate if data is near isotropic



L = Left **R** = Right
P = Posterior **A** = Anterior
S = Superior **I** = Inferior



Axial
(transverse)

Coronal

Sagittal

Radiologist convention:

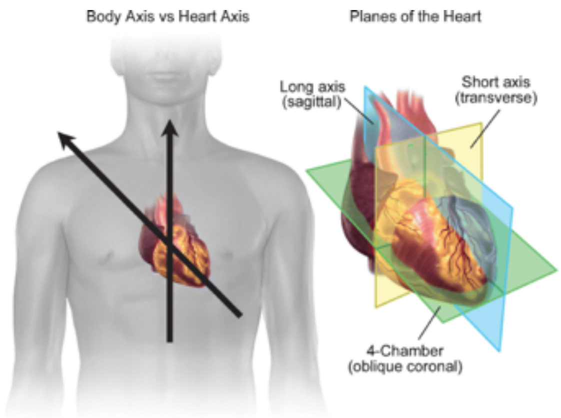
View from the bottom
Note R-L flipping!

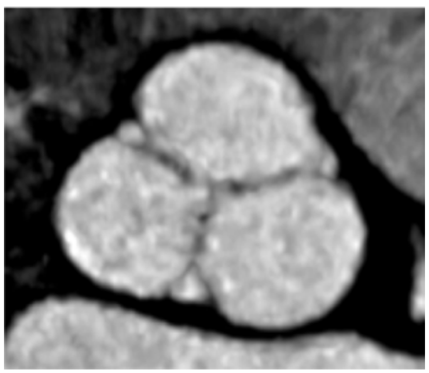
View from the front
Note R-L flipping!

View from left

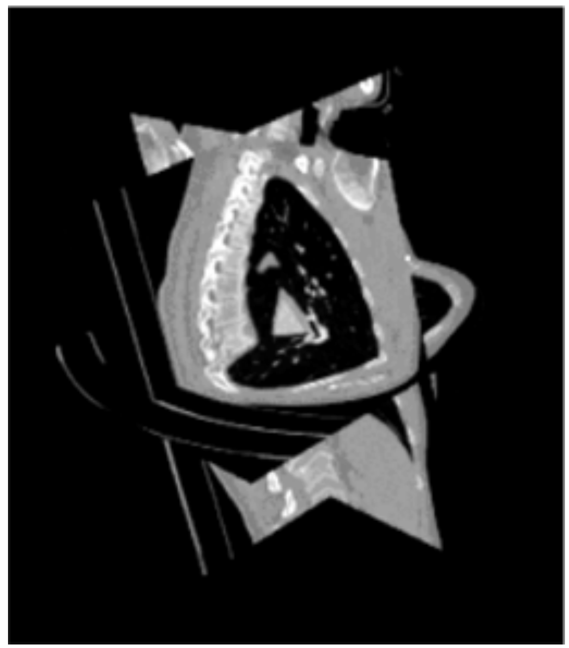
Multiplanar Reconstruction (MPR)

Appropriate if data is near isotropic

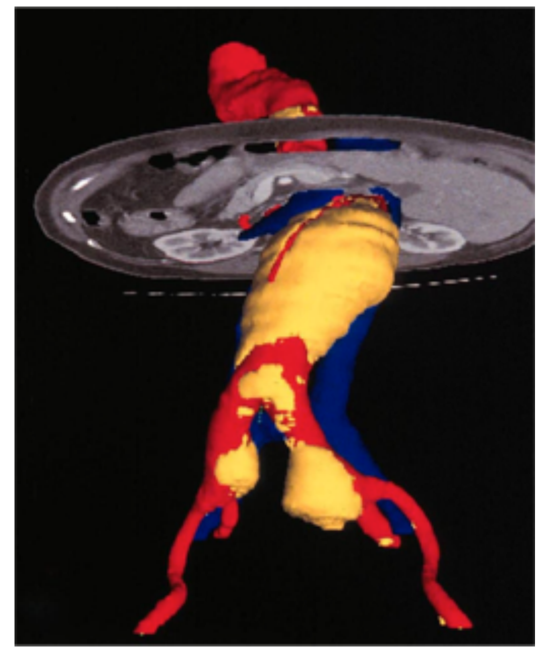




Oblique



Multiple Planes



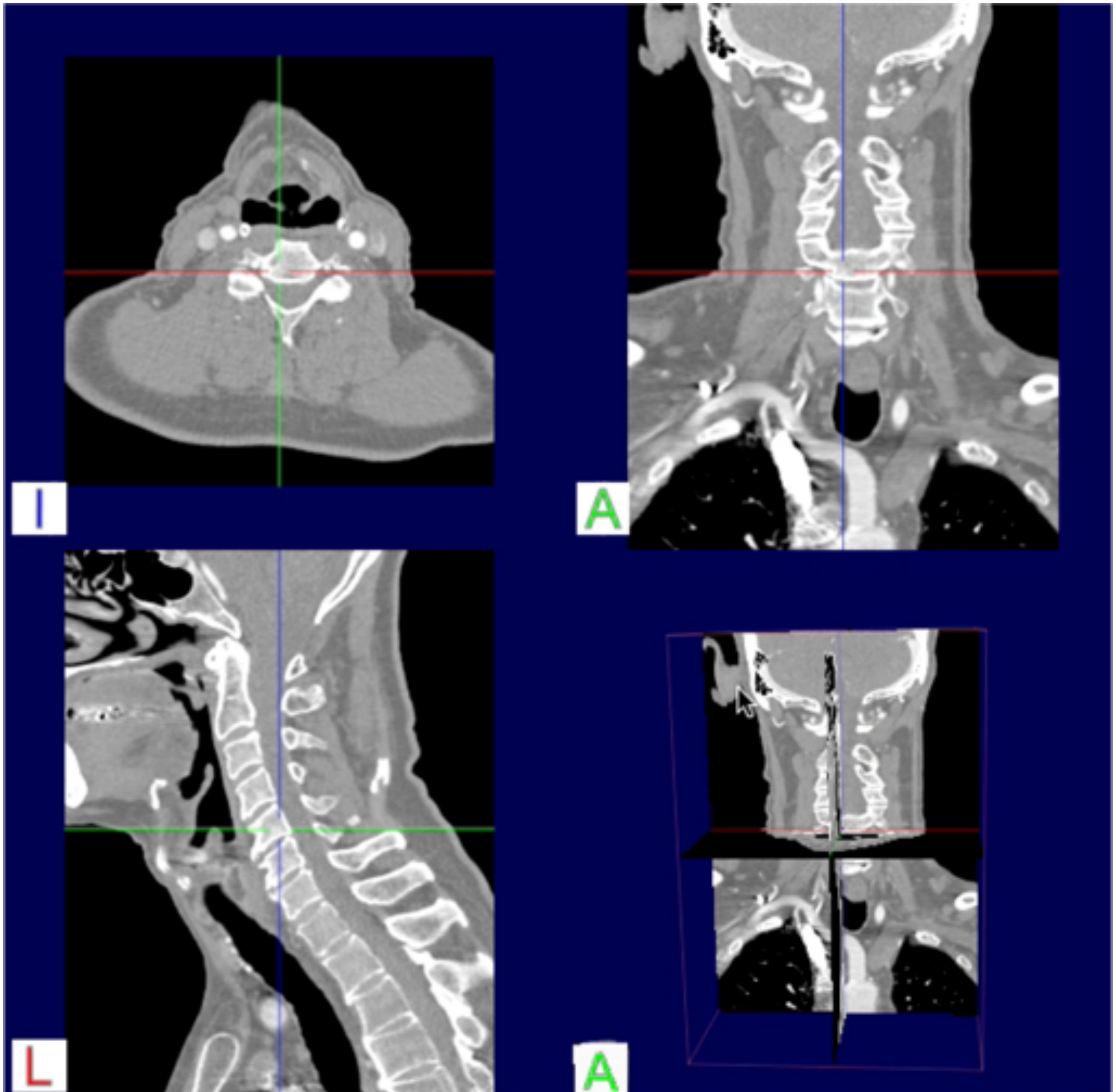
2D+3D

(often linked to the axis of an organ)

Multiplanar Reconstruction (MPR)

Axial

Coronal



Sagittal

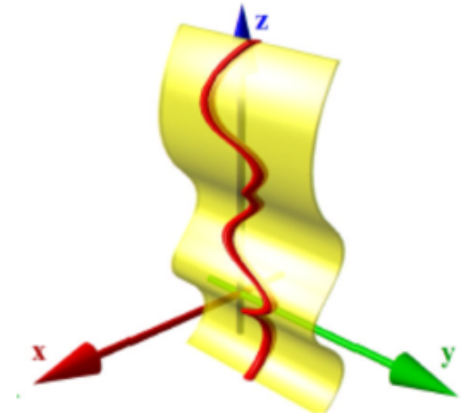
3D

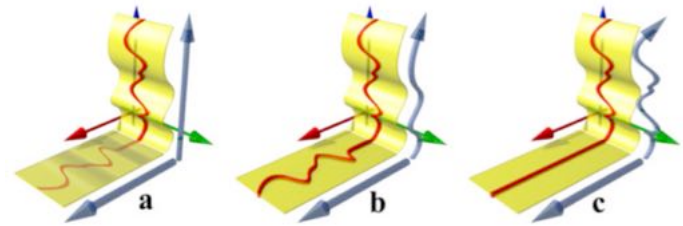
Oblique Plane

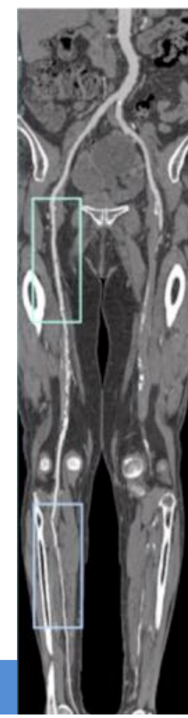
<u>This example:</u>
Swinging between coronal and sagittal views

Curved Planar Reformations (CPR)

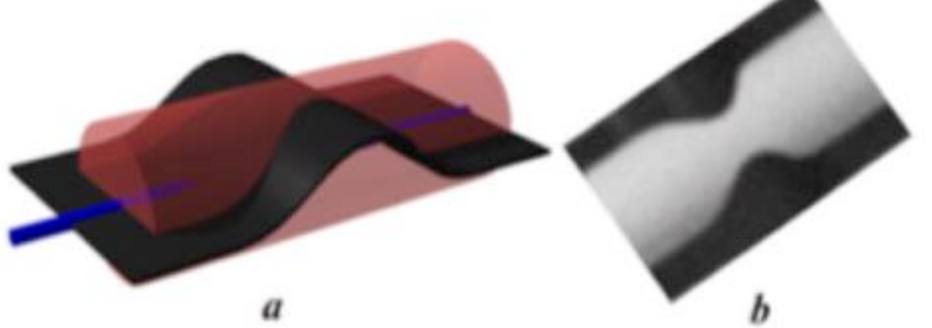
- For slicing through long curved anatomy
- Centerline is defined
  - Manually or automatically
- Sampling along parallel lines
- Various methods for assembling sampled lines into final image
- Pros and Cons
  - Single image to show a long region
  - Artificial stenosis artifact possible







Potential Pitfall of CPR:



Kanitsar et al., IEEE Viz 2003

Curved Planar Reformations (CPR)

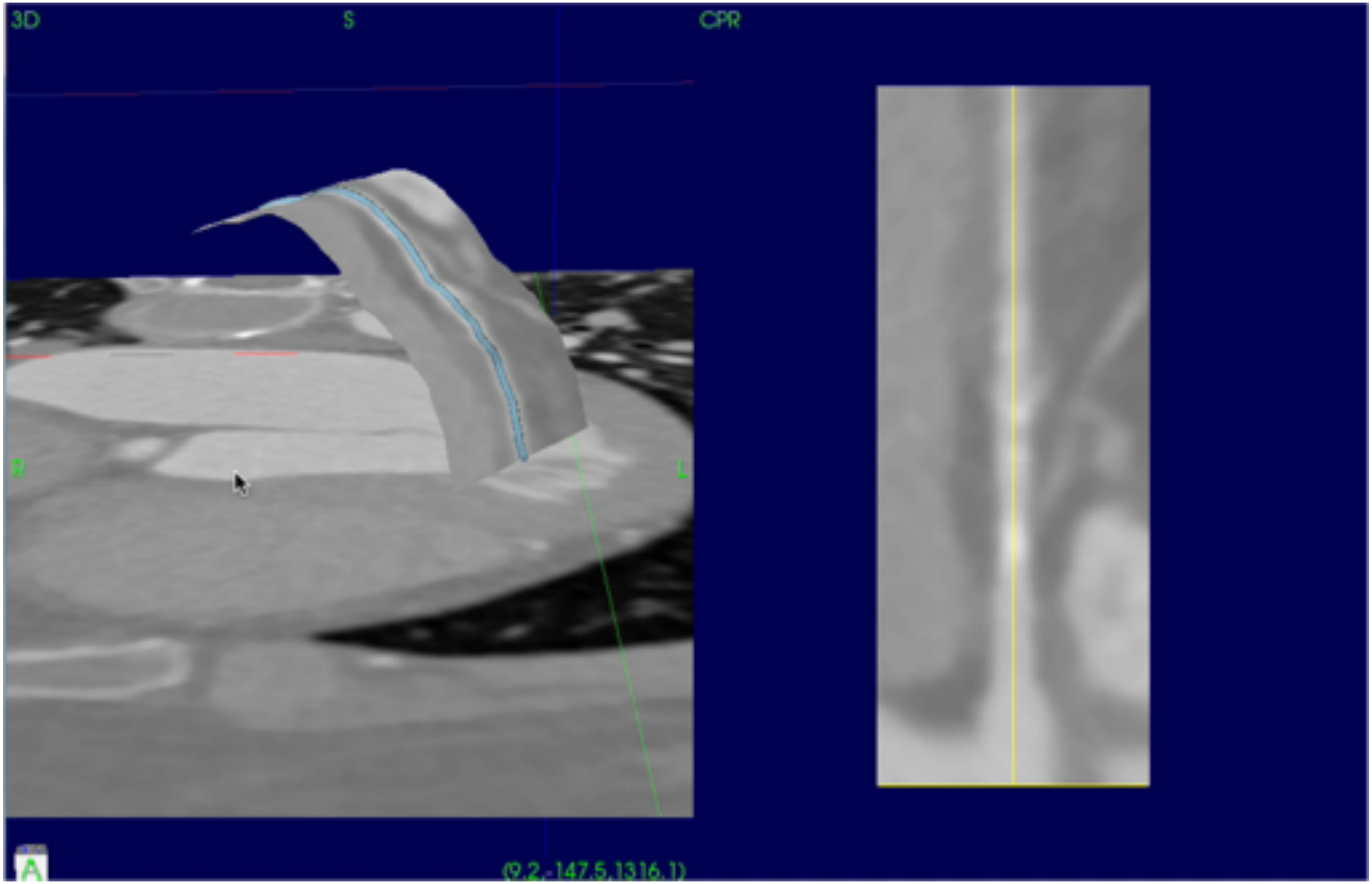
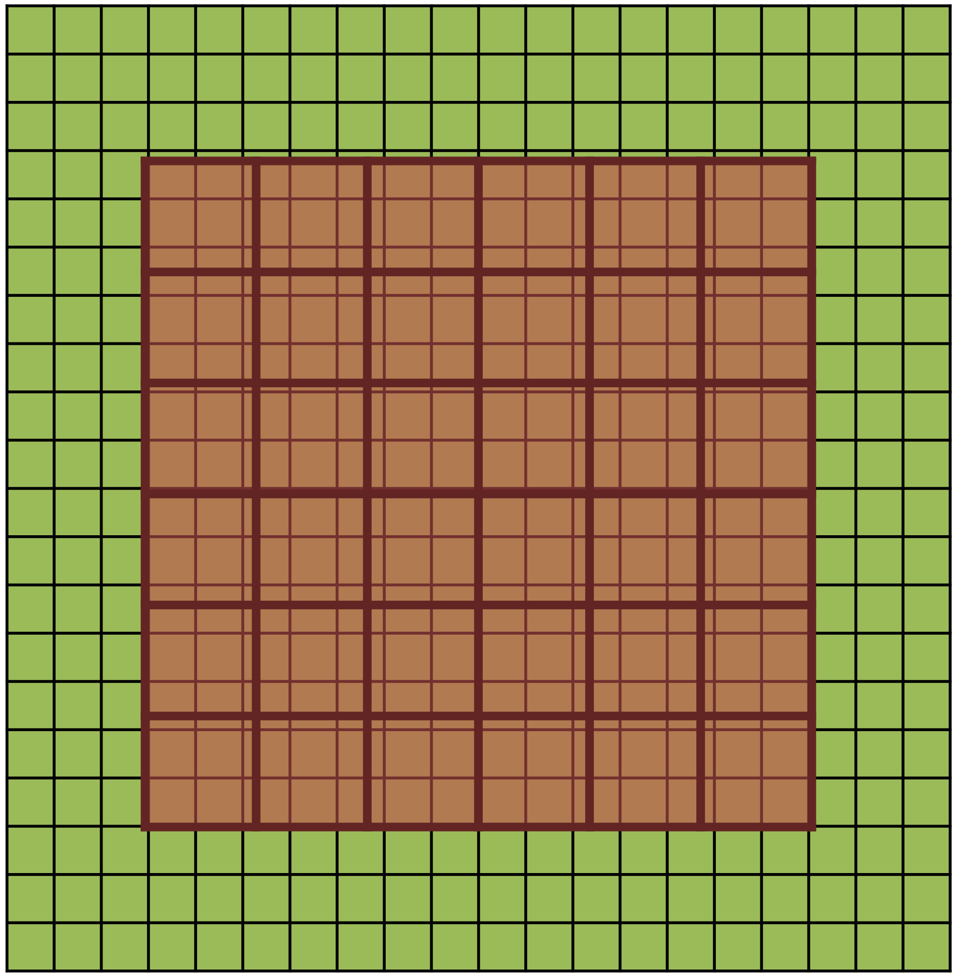


Image Interpolation

Up-sampling, down-sampling, rotation, registration, non-rigid transformations, etc. when you need to re-grid pixels



Down-sampling example:

Original Image

New Image

NOTE: down-sampling can cause aliasing artifacts if not carefully done! More on this in the convolution lecture

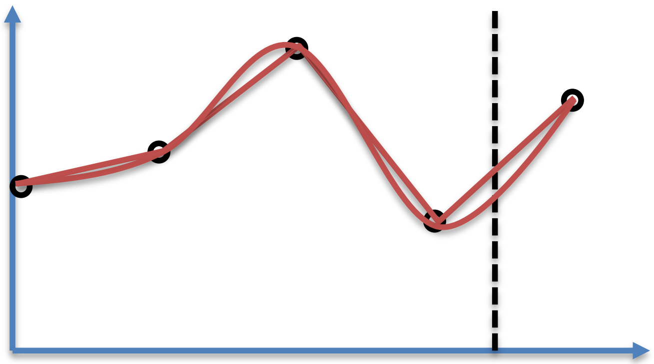
Image Interpolation

For some deep learning methods, a very constrained up-sampling problem is posed where up-sampling rates are integral multiples and there is no rotation. More on this in the deep learning lecture

222		105		211		94
199		188		141		51
201		150		138		222
247		215		186		103

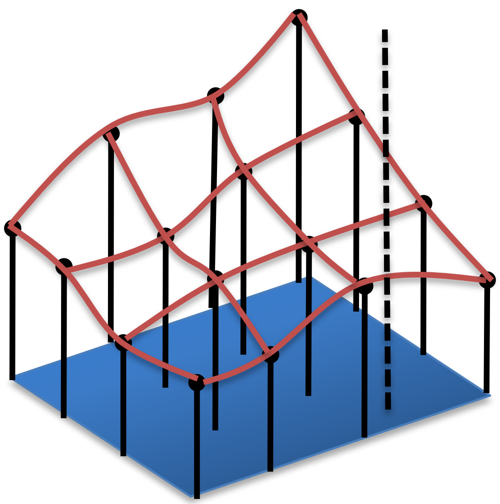
But in this lecture, we will consider the more general case where the sampling rate may not be an integral multiple and there may be rotation/deformation of the grid

Interpolation



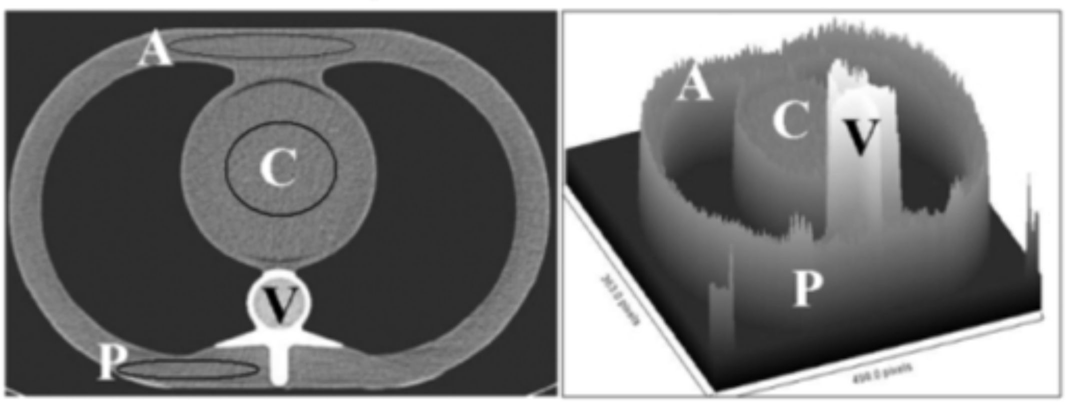
Linear Interpolation ("connect the dots")

Higher Order Interpolation (smooth)



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

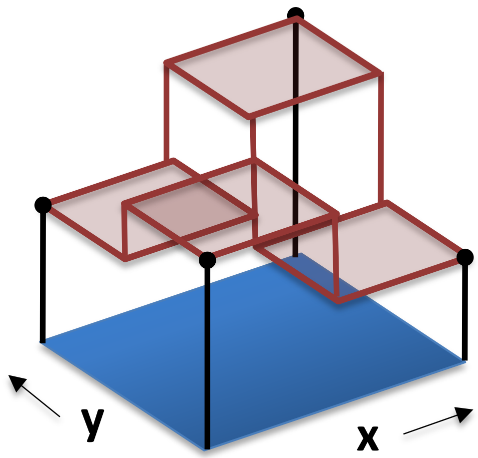
This naturally extends to 3D



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

Kaira et al., KJR 2009

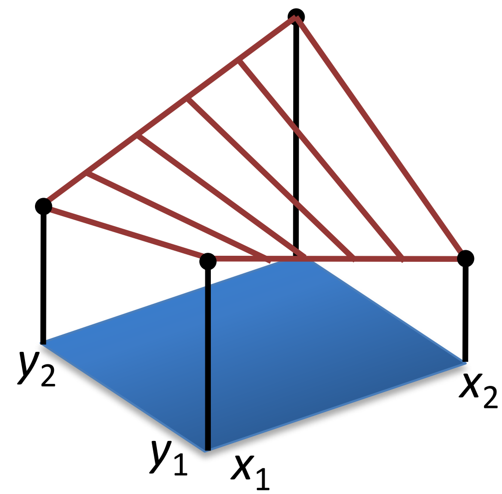
Image Interpolation



Nearest Neighbor
Interpolation

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

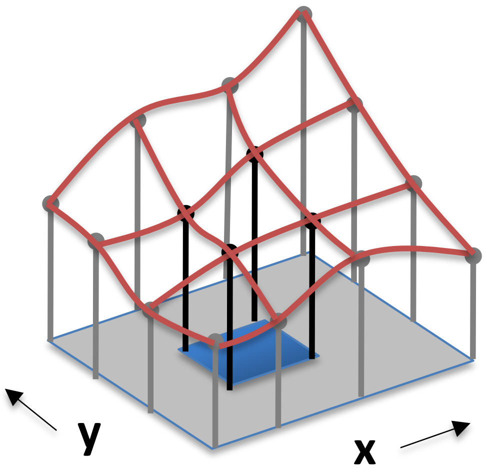
[ ] is rounding function



Linear
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

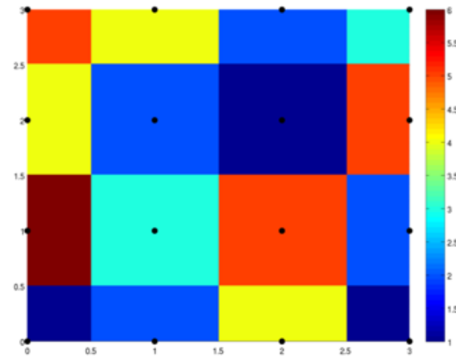


Cubic
Interpolation

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

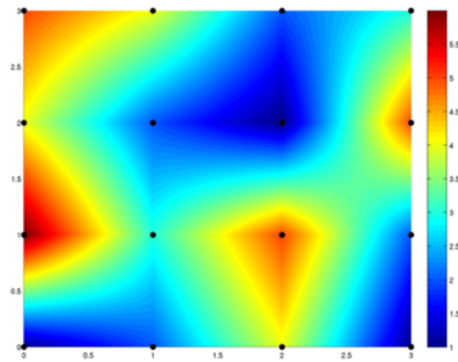
Nearest Neighbor

(useful on categorical label maps)



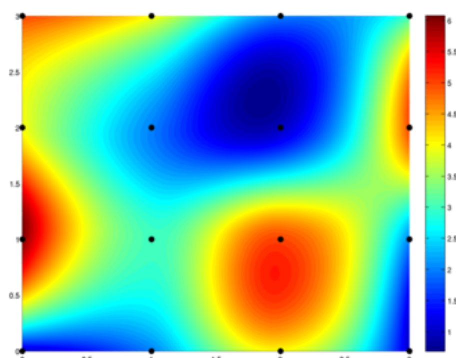
Linear

(fast, most common method)

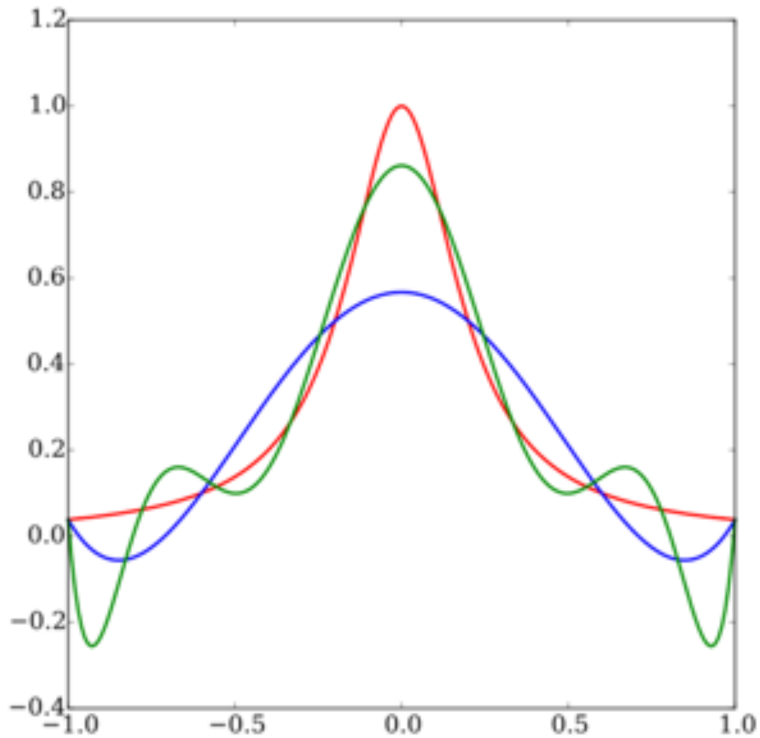


Cubic

(not often justified)



Runge Phenomenon
Higher Order Polynomial Interpolation



wikipedia

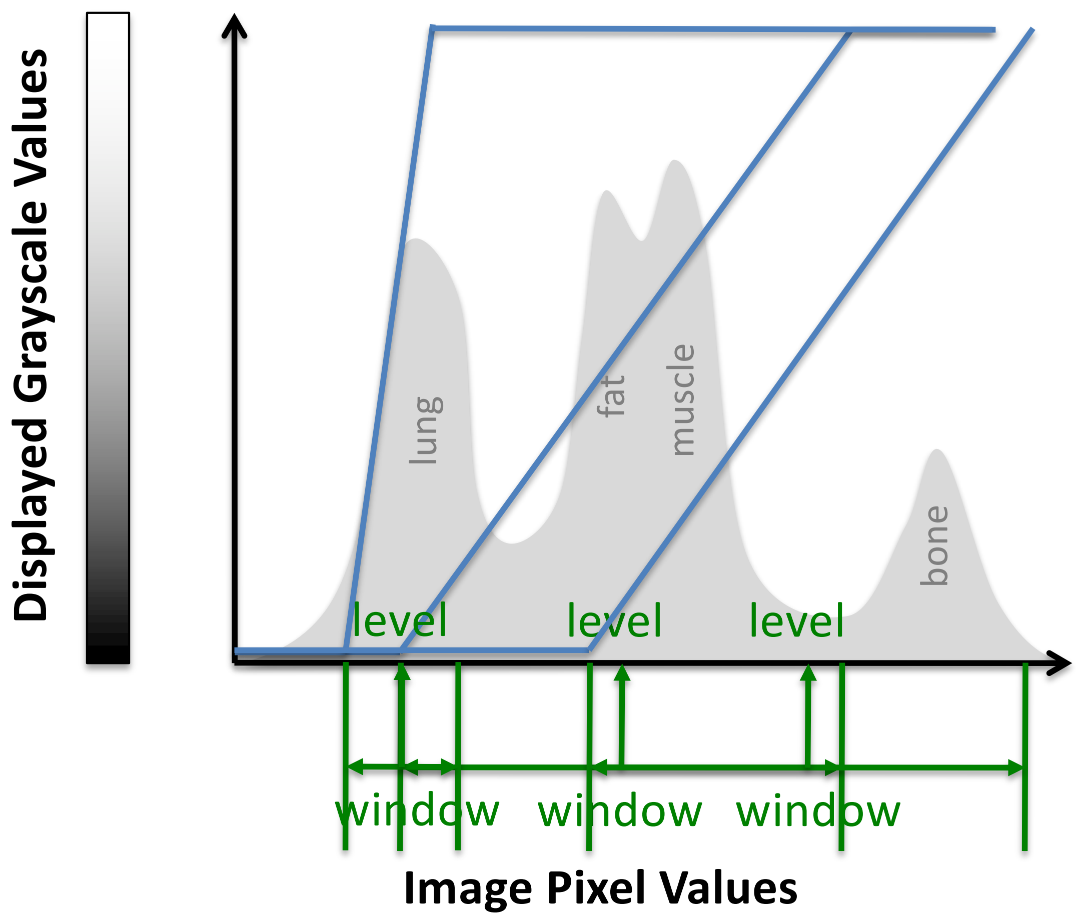
Original function: $f(x) = \dfrac{1}{1 + 25x^2}$

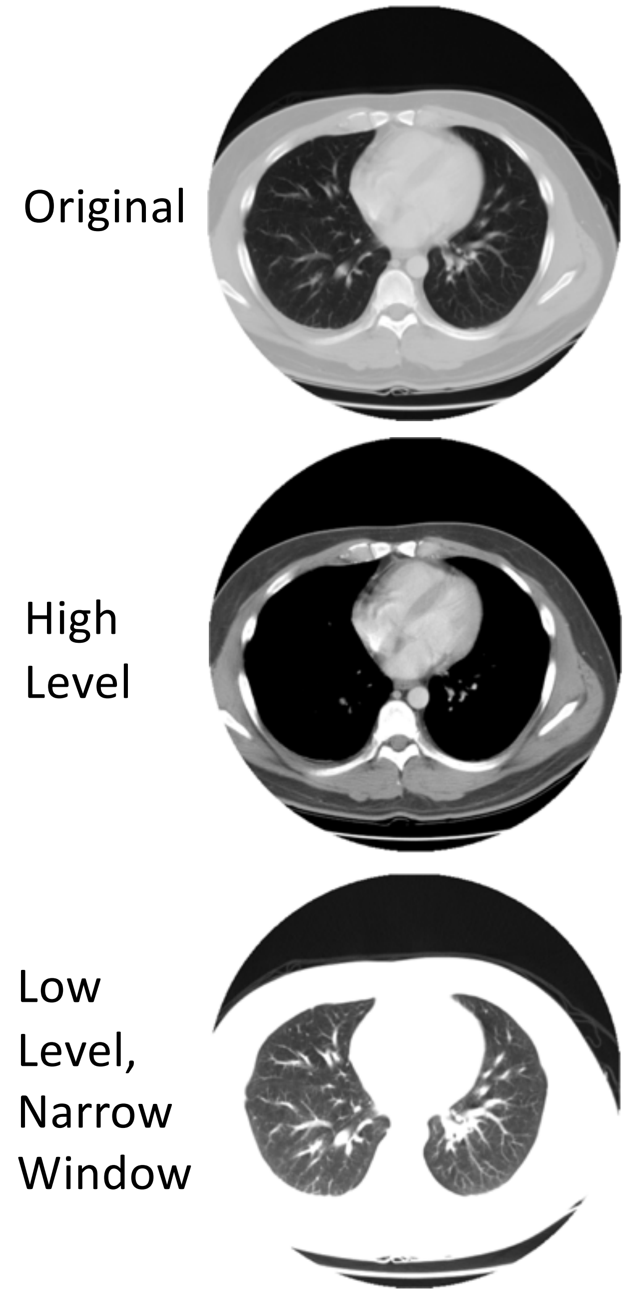
5^{th} order interpolating polynomial
9^{th} order interpolating polynomial

Interpolating between evenly spaced samples

Intensity Scale Mapping

"Window Leveling"

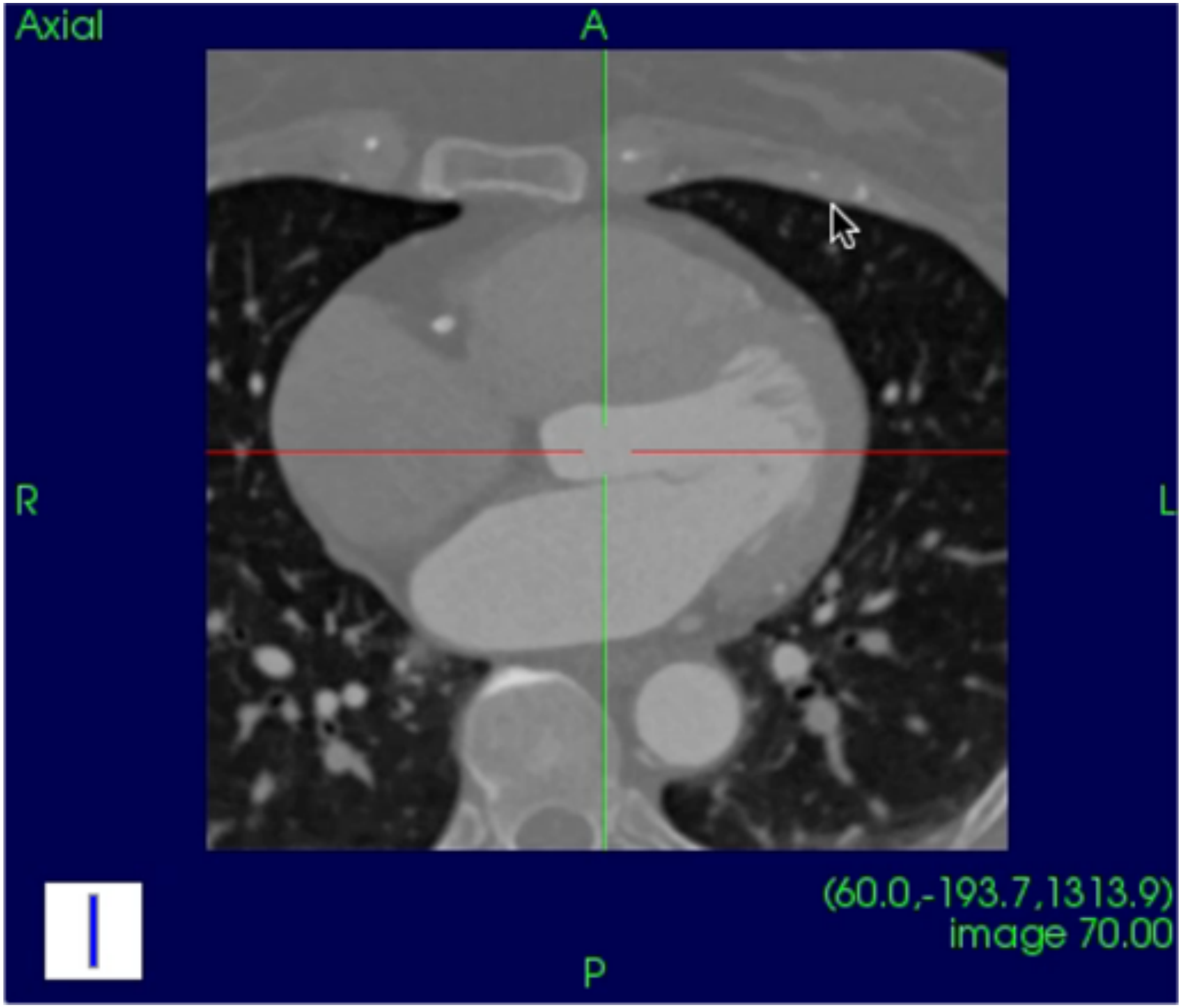




Most imaging modalities: 16 bits (65,536 values)
Most displays (and human eye): 8 bits (256 values)

(color mapping is complicated, more on this later)

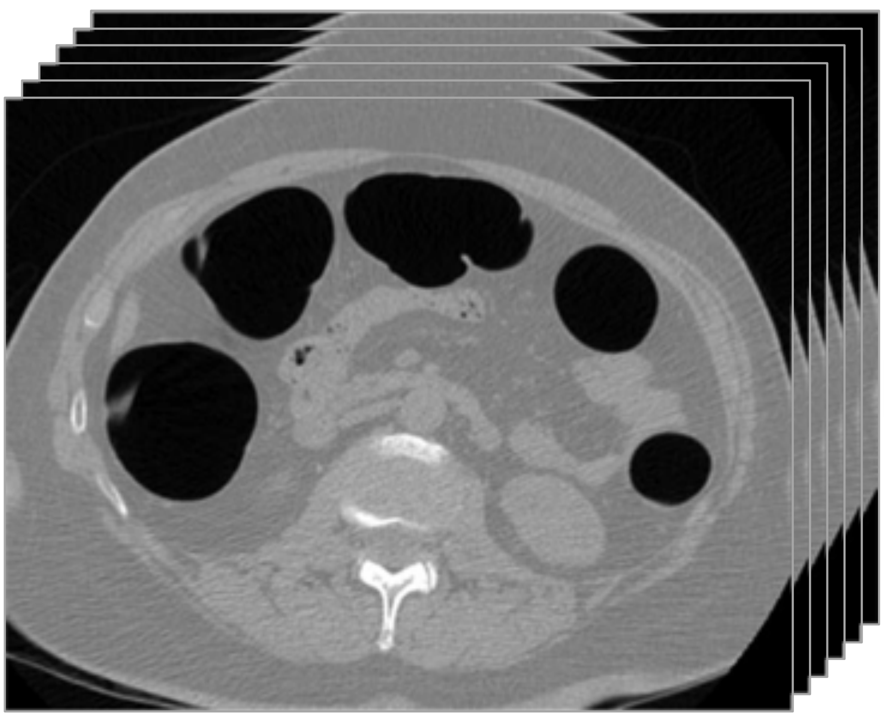
Window Leveling



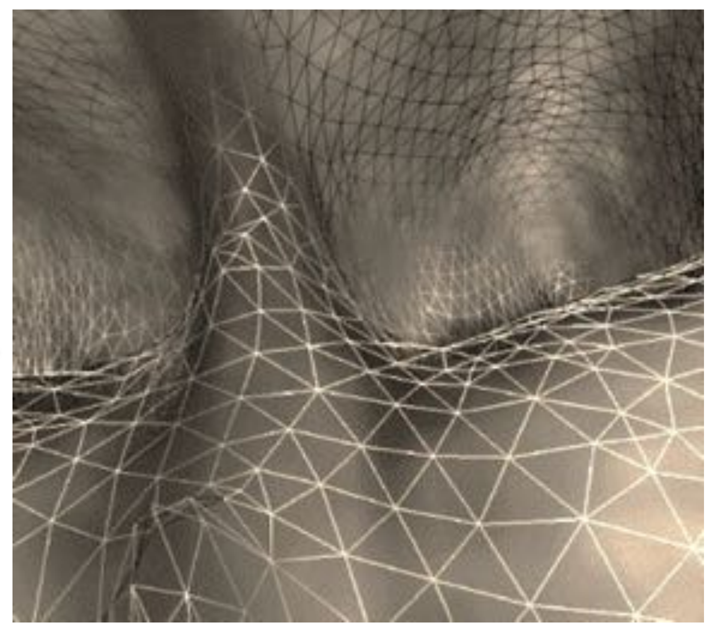
3D Computer Graphics

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

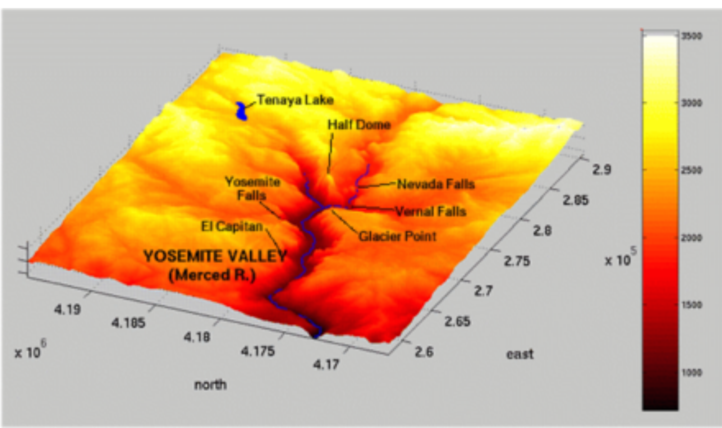


Radetzky et al., AIM 2002

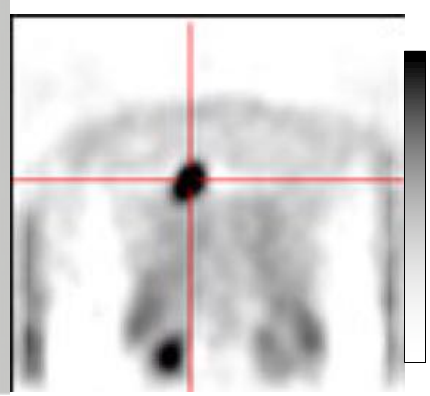
Triangular Mesh

Marching Cubes Algorithm

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



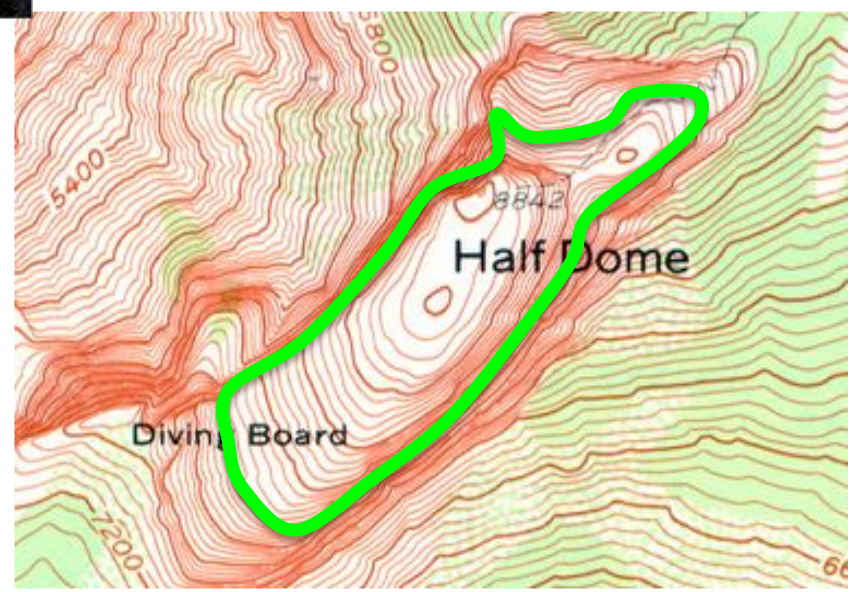
Yosemite Valley



PET Scan

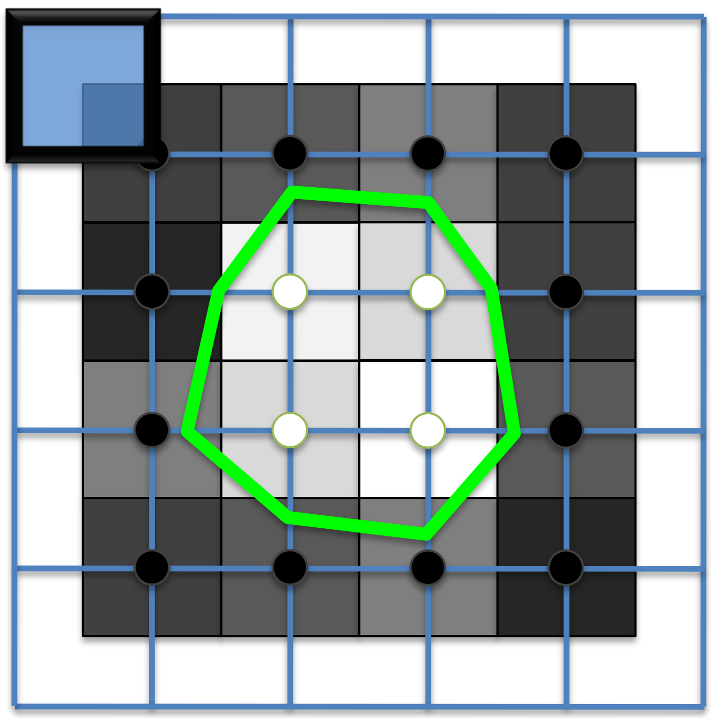
(note black/white flipped)

(assume altitude data is sampled on a grid)

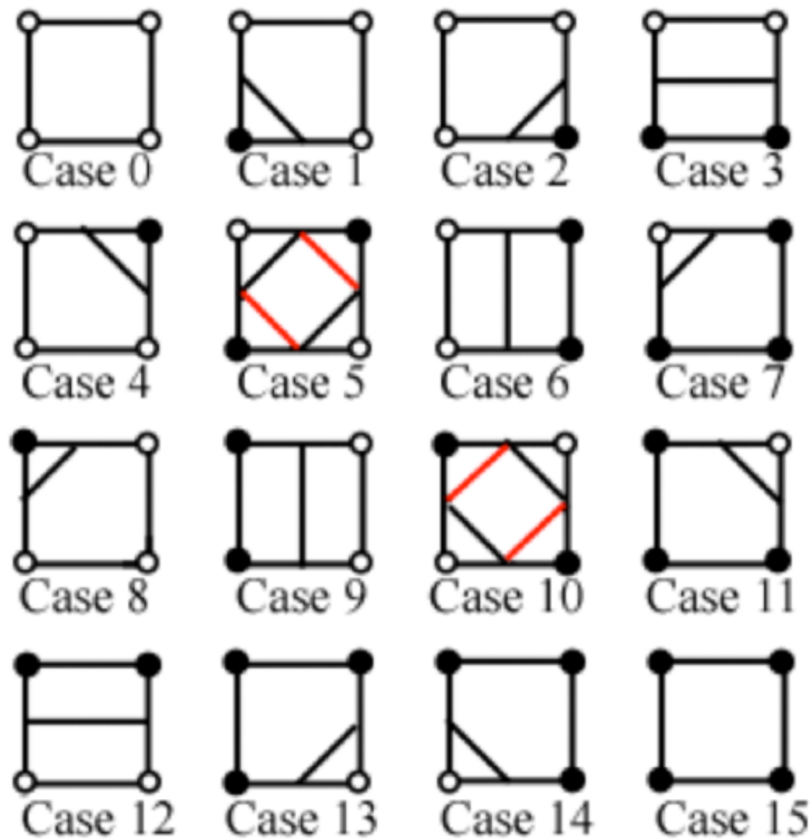


8000 ft iso-contour

Marching Squares (in 2D)



Examine squares connecting 4 pixel centers

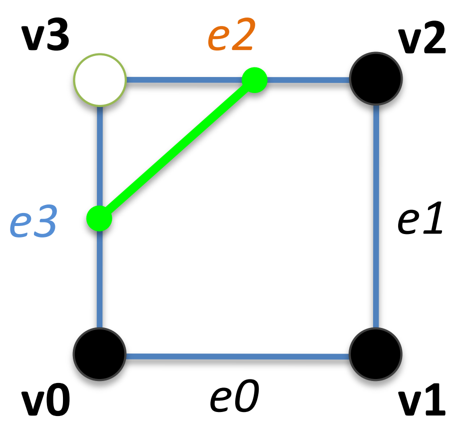


All 16 possibilities

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* |

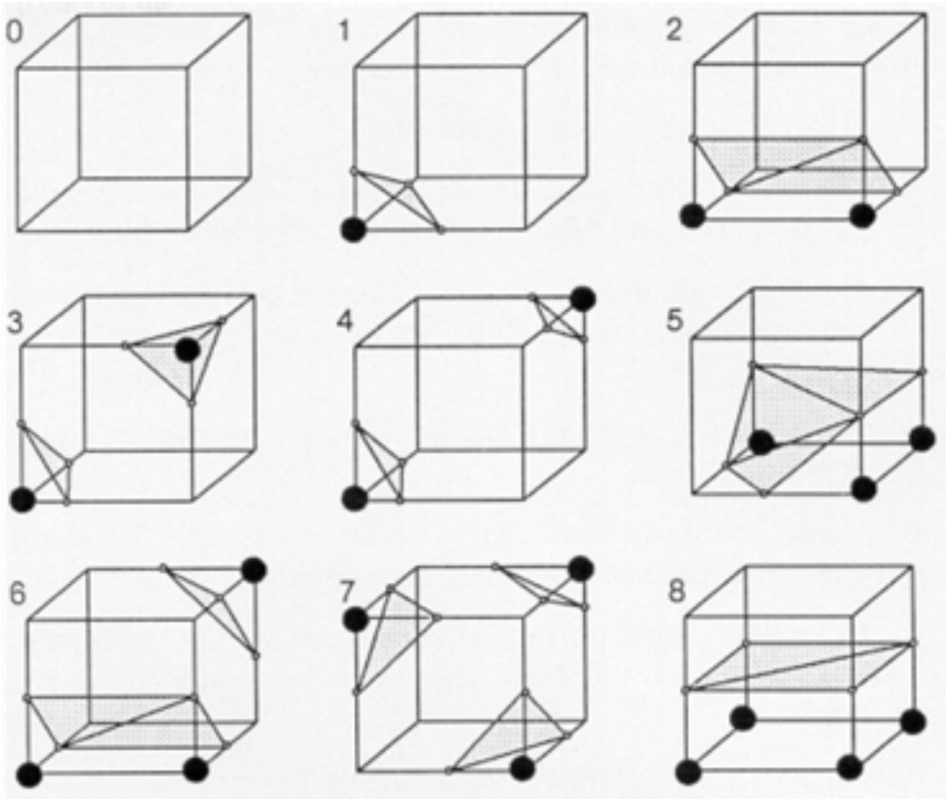
thus *e3* and *e2* are intersected by line segments

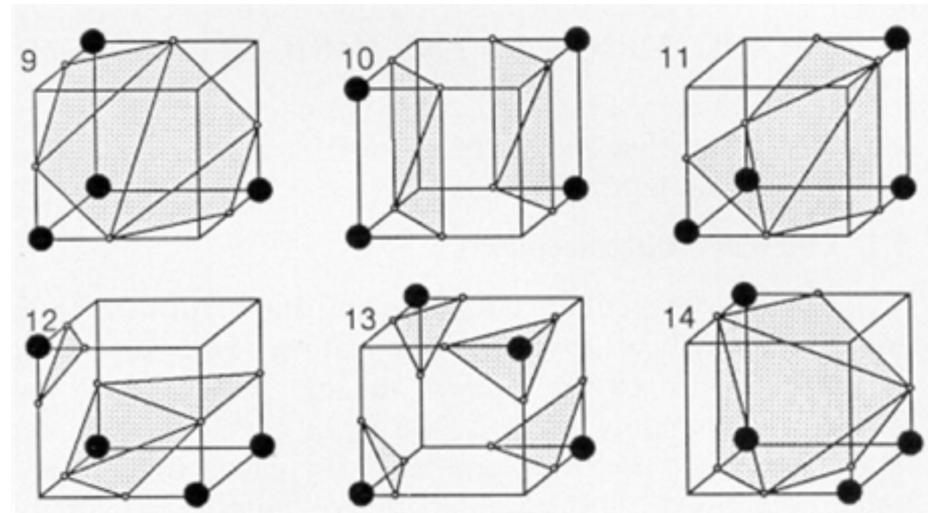
`line_table` is a pre-defined lookup table of all 16 cases and returns a list of pairs of intersected edges that make line segments

`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

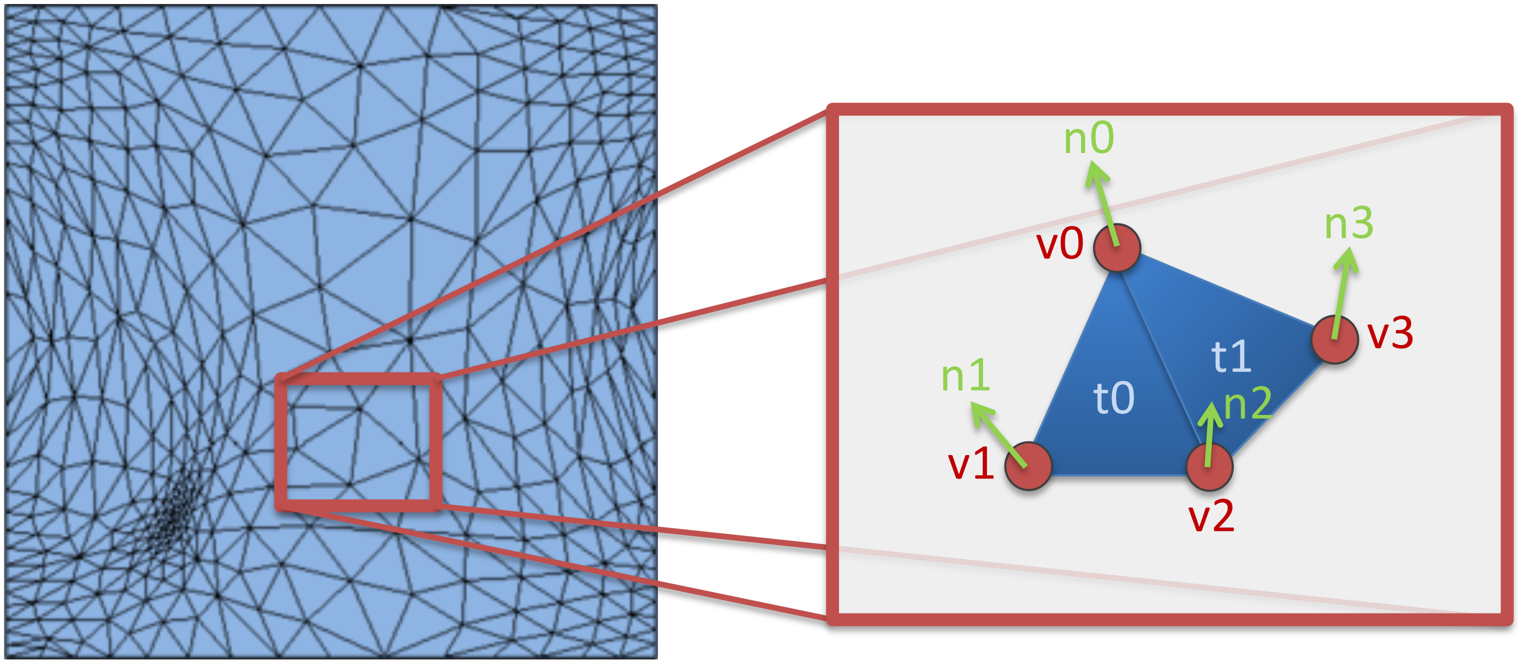
256 cases total
15 rotationally unique cases shown here

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

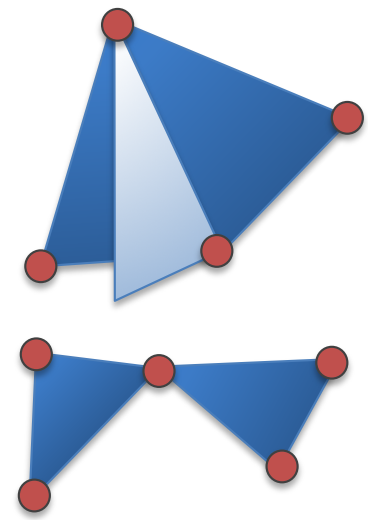
`cube_index`, `edge_table` and `triangle_table` are directly analogous to marching squares except:

- 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)

Basic Mesh Data Structure



Non-manifold Mesh:



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)
...

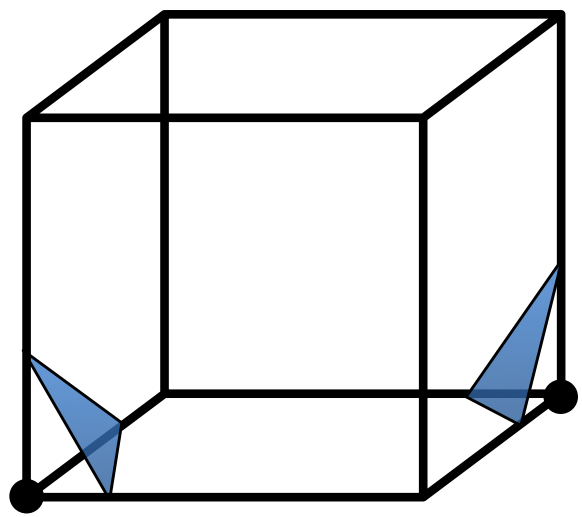
(Marching Cubes doesn't tell you how to calculate normals at each vertex; needed for smooth surface shading)

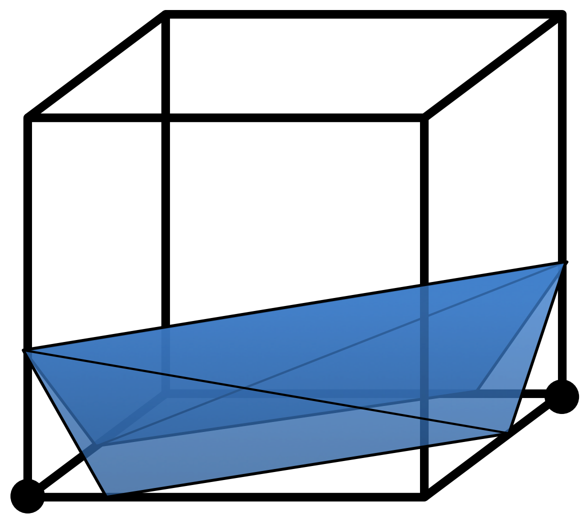
Questions:

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?





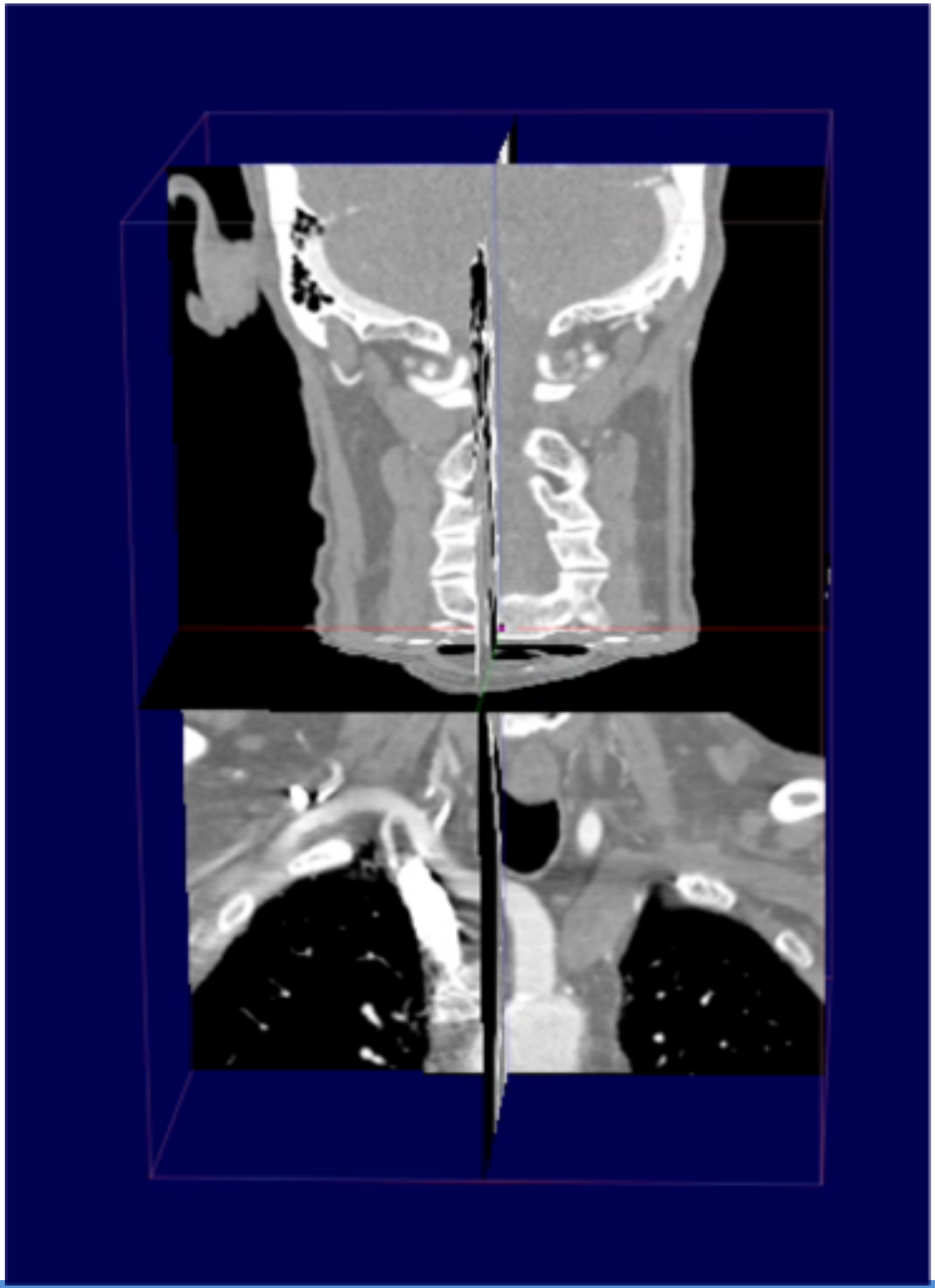
Shaded Surface Display

- 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
  - Good for visualizing computed models
  - Inner structures obscured

Typical game character: 23k triangles (e.g., Fortnite)
Typical medical model: 500k-10M triangles

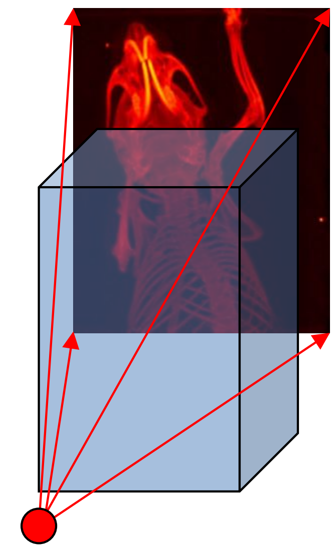
Example rendering of human lungs from CT

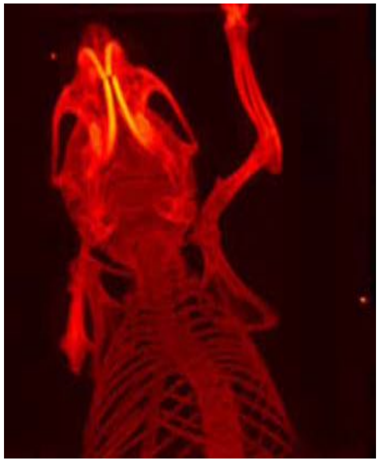
Shaded Surface Display



Mean/Max Intensity Projection

- Rays are mathematically cast through the 3D image and the mean/max (interpolated) intensity encountered is put into that 2D output 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





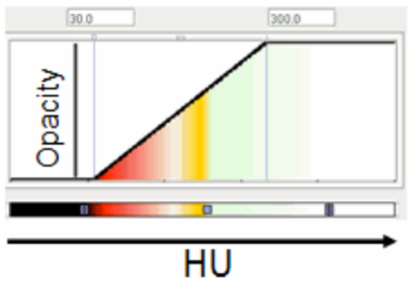
Mean/Max Intensity Projection

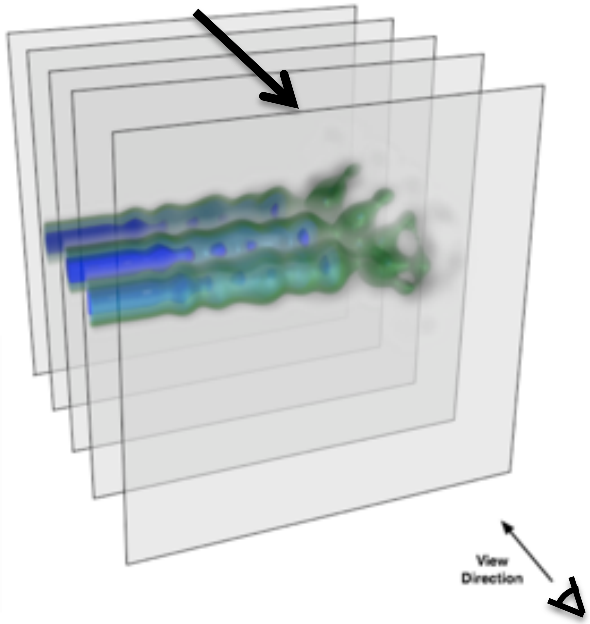
Mean Intensity Projection

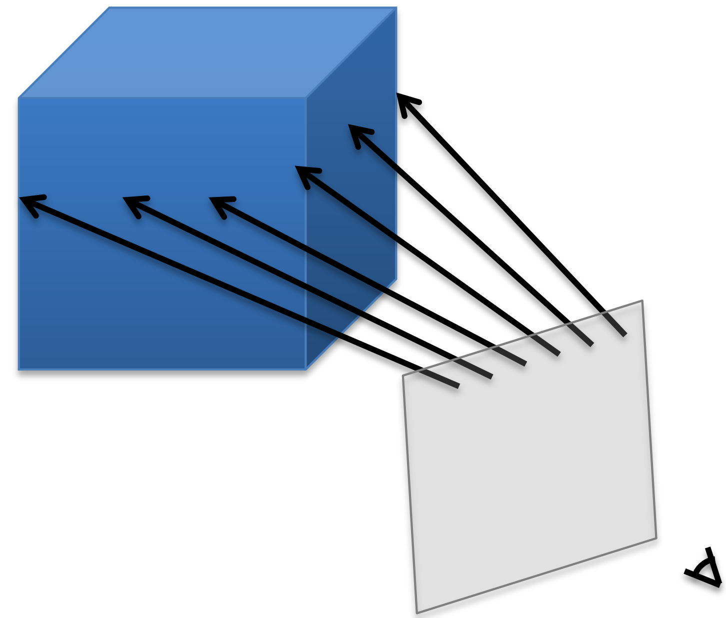
Maximum Intensity Projection

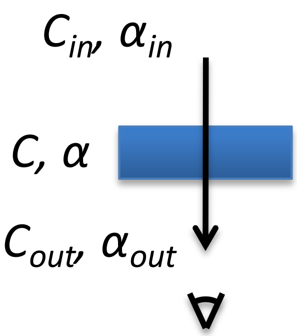
Direct Volume Rendering

Opacity Table and Color Table





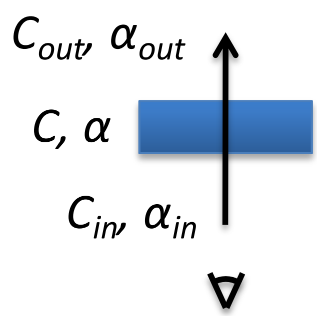




$$C_{out} = (1-\alpha)\cdot C_{in} + C$$

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

Object Order Volume Rendering (back-to-front)



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

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

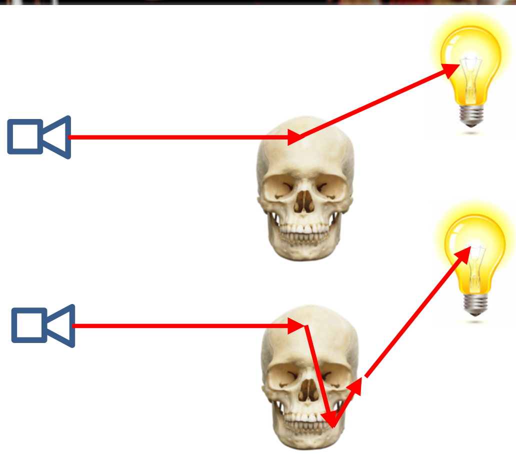
Image Order Volume Rendering (front-to-back)

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

Volume Rendering

Norman Gellada, Cedars Sinai

Ray Tracing
(aka Cinematic Rendering)



Ray Casting

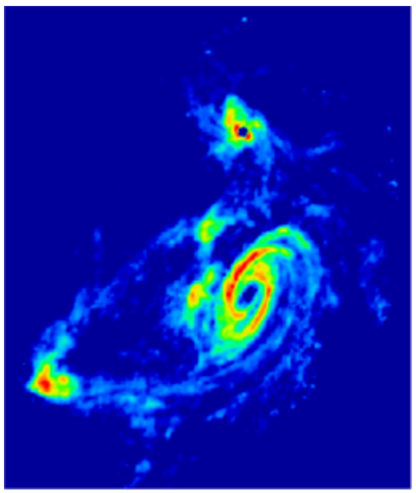
Ray Tracing

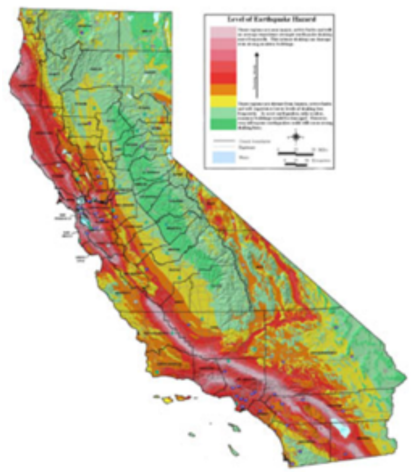
Norman Gellada, Cedars Sinai

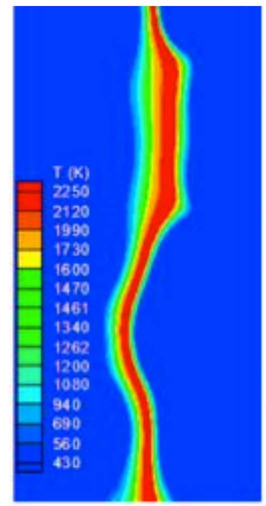
Image Fusion

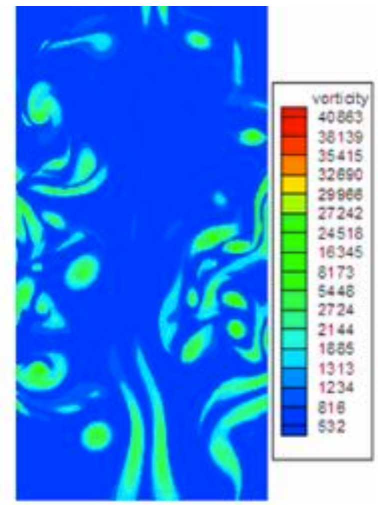
Image Fusion

- Image fusion is the combination of information from 2 or more images
- Pseudo-coloring used in many scientific and engineering fields
  - Assigns 3-component color to 1-component scalar data by using a color lookup table
  - Astronomy, geography, fluid simulations, etc.



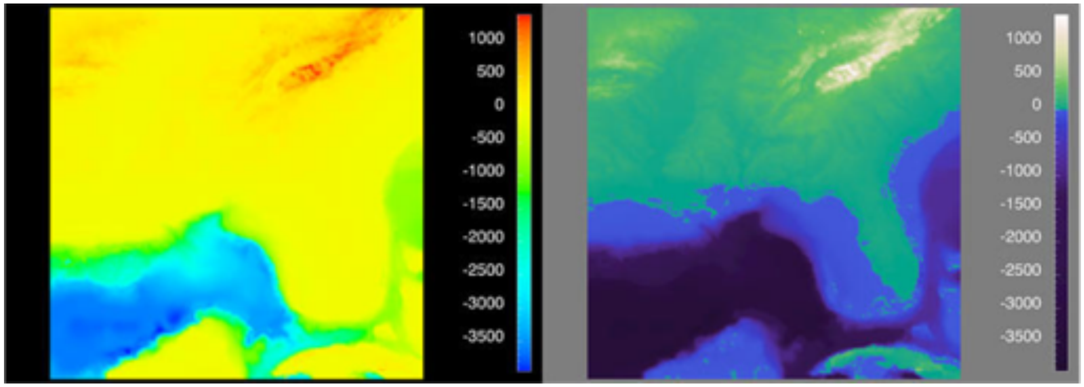




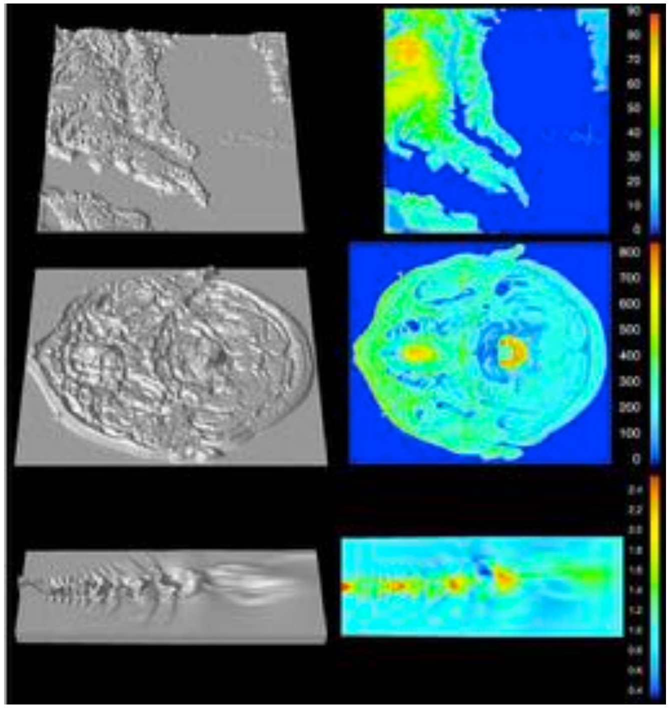


- What kind of information is to be revealed?
  - Metric: quantity at each point
  - Form: shape and structure
- Combine anatomic (e.g., CT) and functional (e.g., PET)

Pseudocoloring Artifacts

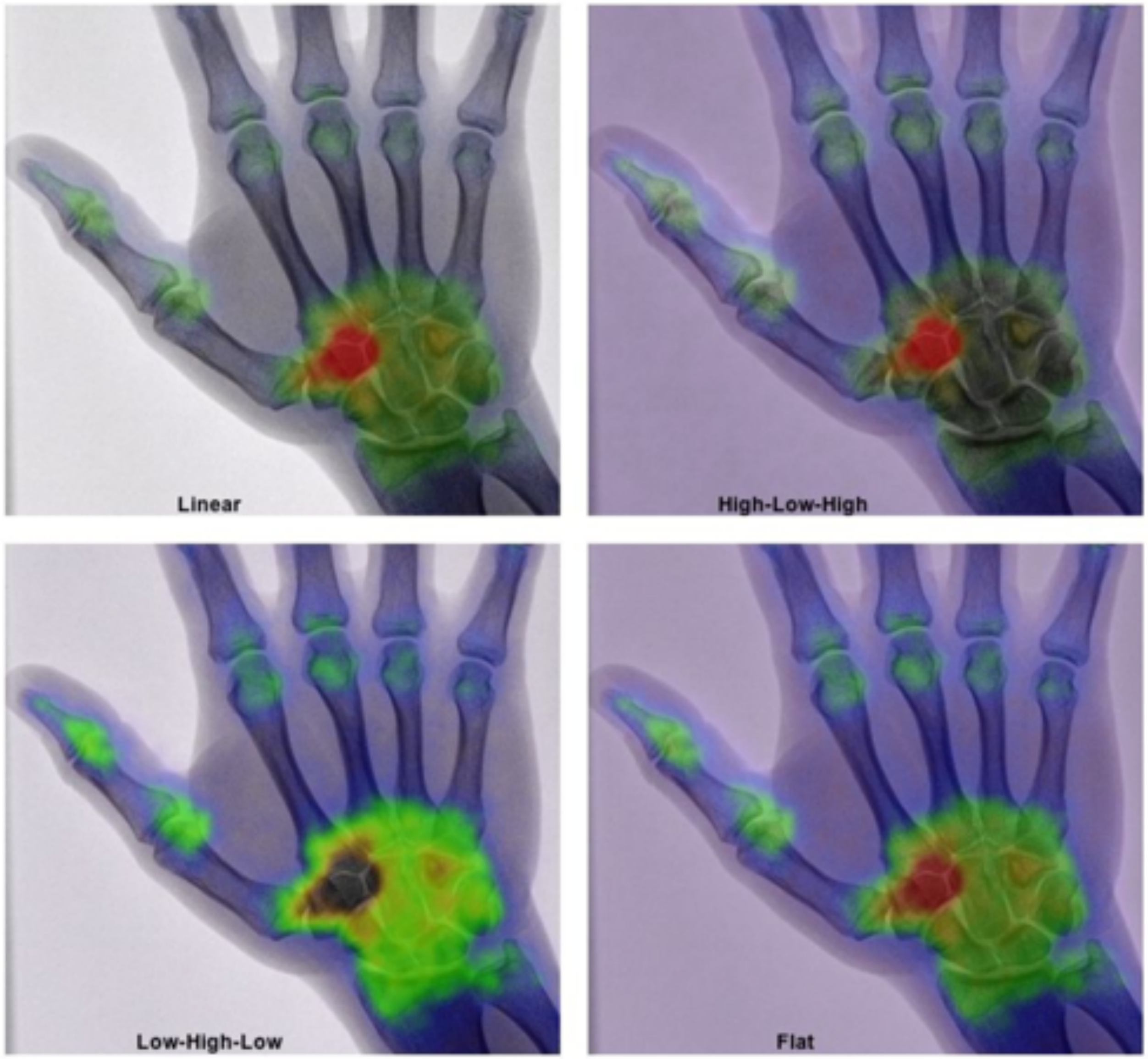


False Negative Artifacts



False Positive Artifacts

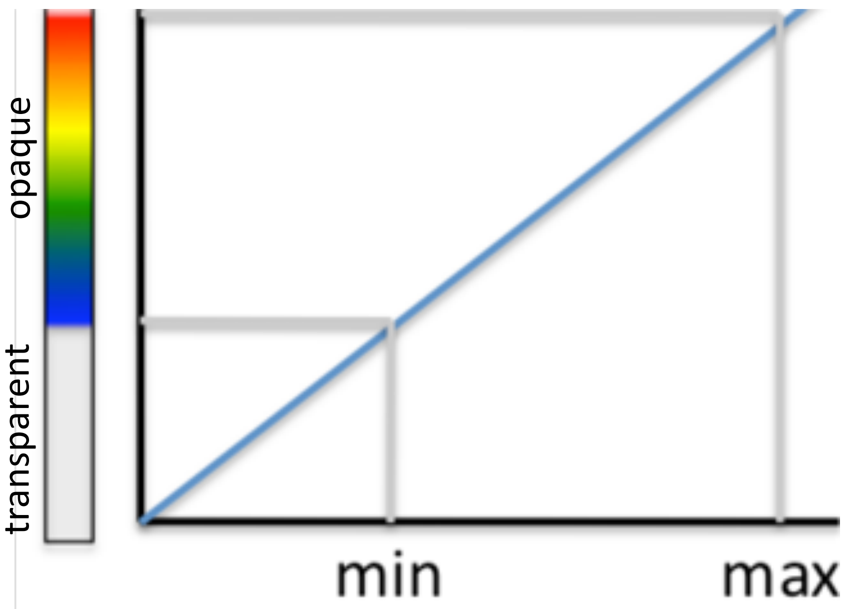
Alpha Blending

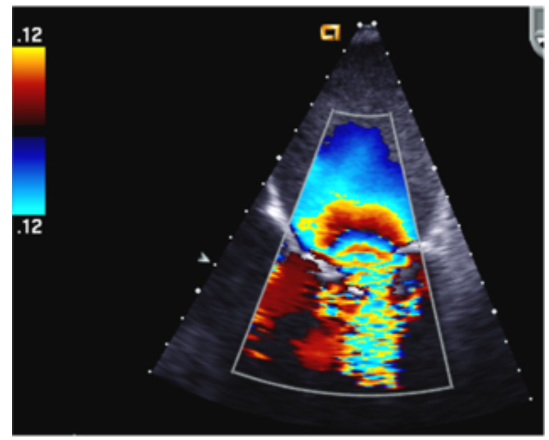


$$I = \alpha I_1 + (1 - \alpha) I_2$$

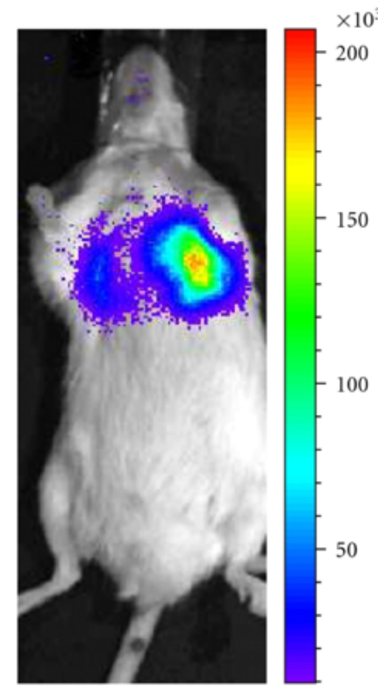
Generated by Kieran Maher, Nov 2006

Alpha Blending





Doppler Ultrasound



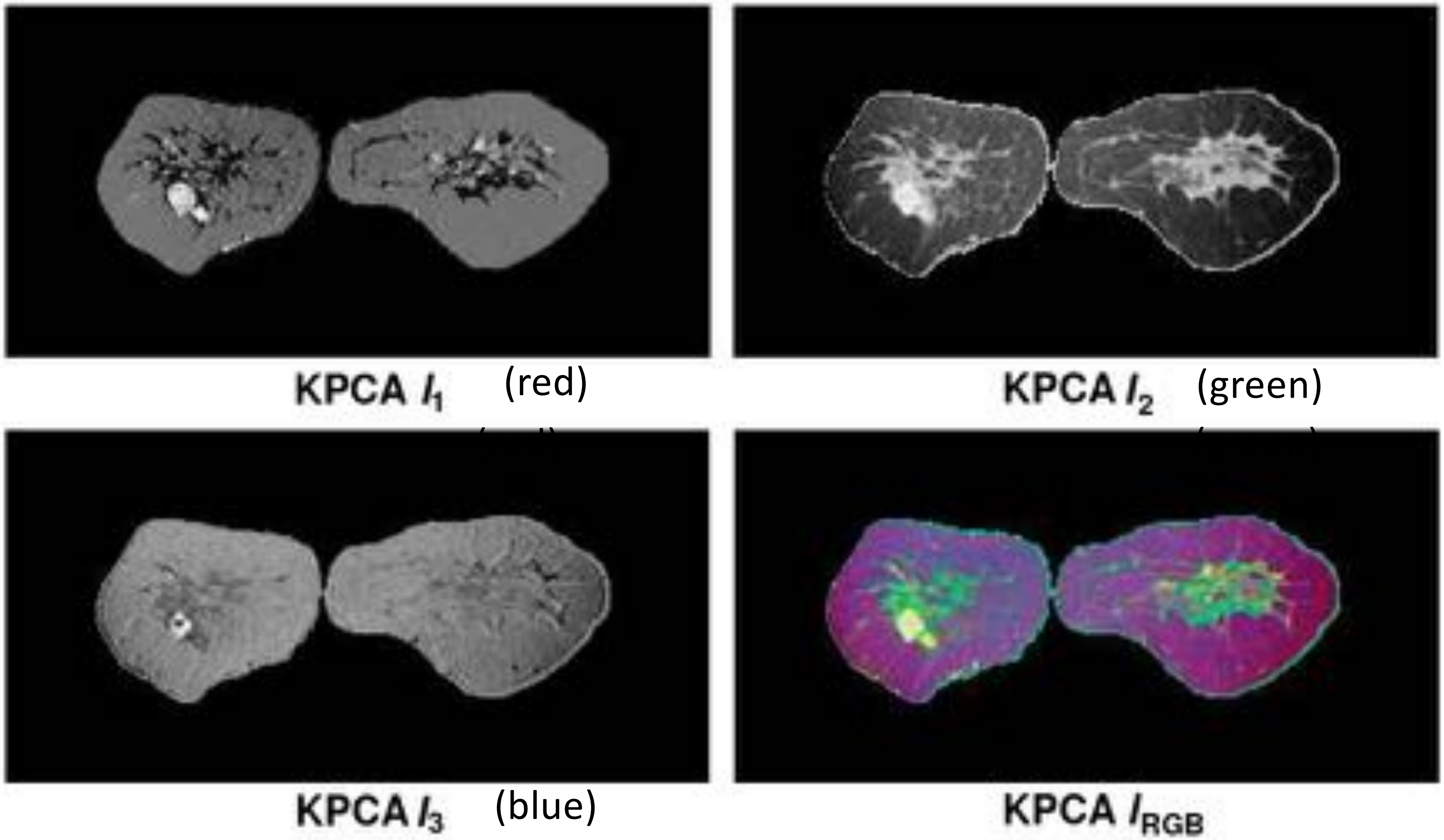
BLI

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. Low biological activity made fully transparent so you can see background anatomy for context.

RGB Fusion



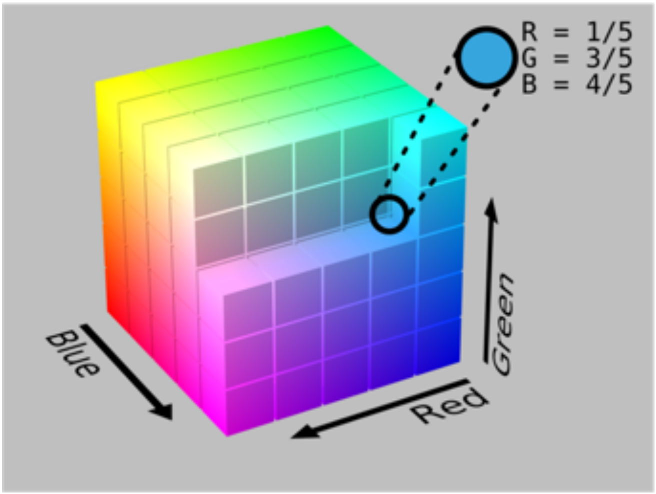
Twellmann et al., Biomed Eng Onl 2004

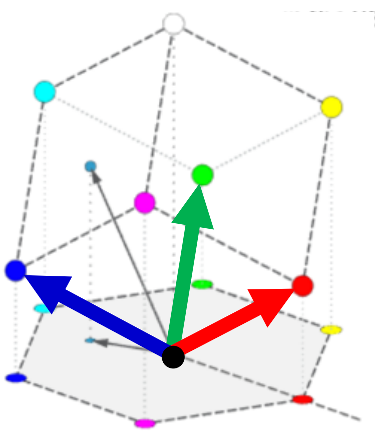
Assigning a color channel (red, green, blue) to each of three images but perception of three channels is intertwined

Color Spaces

RGB

Red Green Blue





HSL

Hue Saturation Lightness

HSV

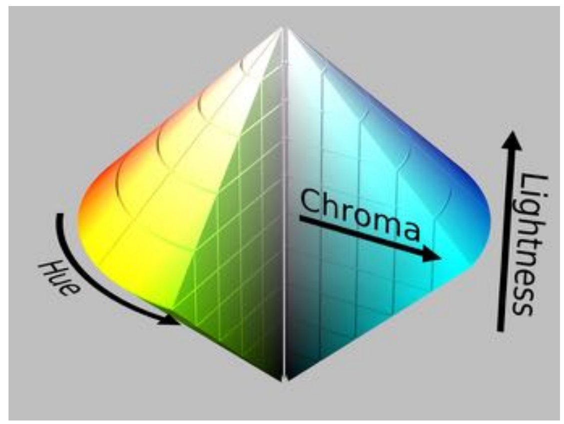
Hue Saturation Value

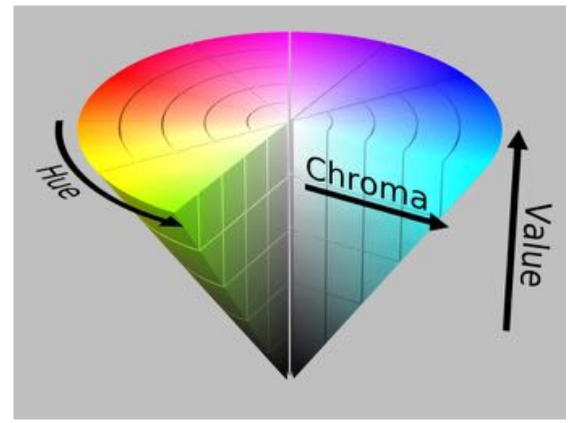
$M = max_{R,G,B} \quad m = min_{R,G,B} \quad C = M - m$

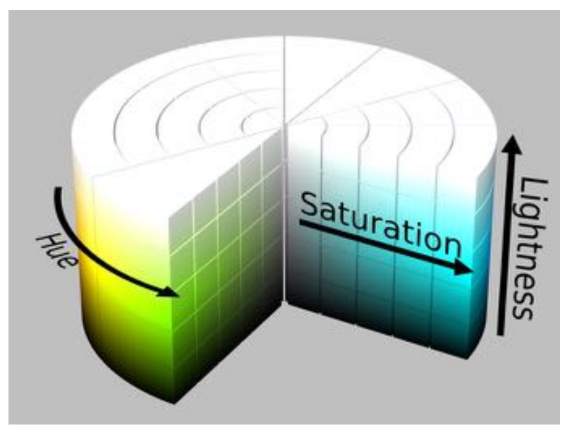
$$H = \begin{cases} 60° \cdot \frac{G-B}{C} & if\ M = R \\ 60° \cdot \frac{B-R}{C} + 120° & if\ M = G \\ 60° \cdot \frac{R-G}{C} + 240° & if\ M = B \end{cases}$$

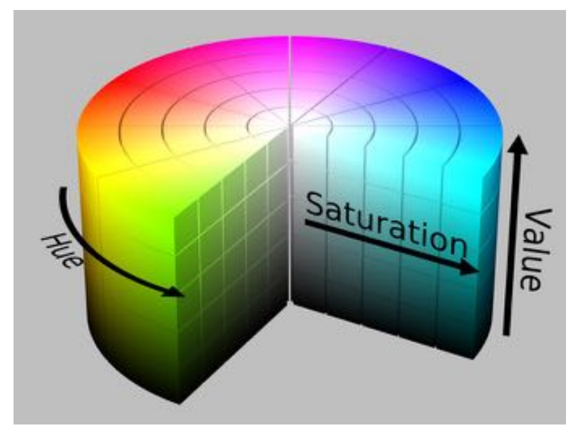
$$S_{HSL} = \frac{C}{1 - |M + m - 1|} \qquad L = \frac{M+m}{2}$$

$$S_{HSV} = \frac{C}{M} \qquad \mathrm{V = M}$$





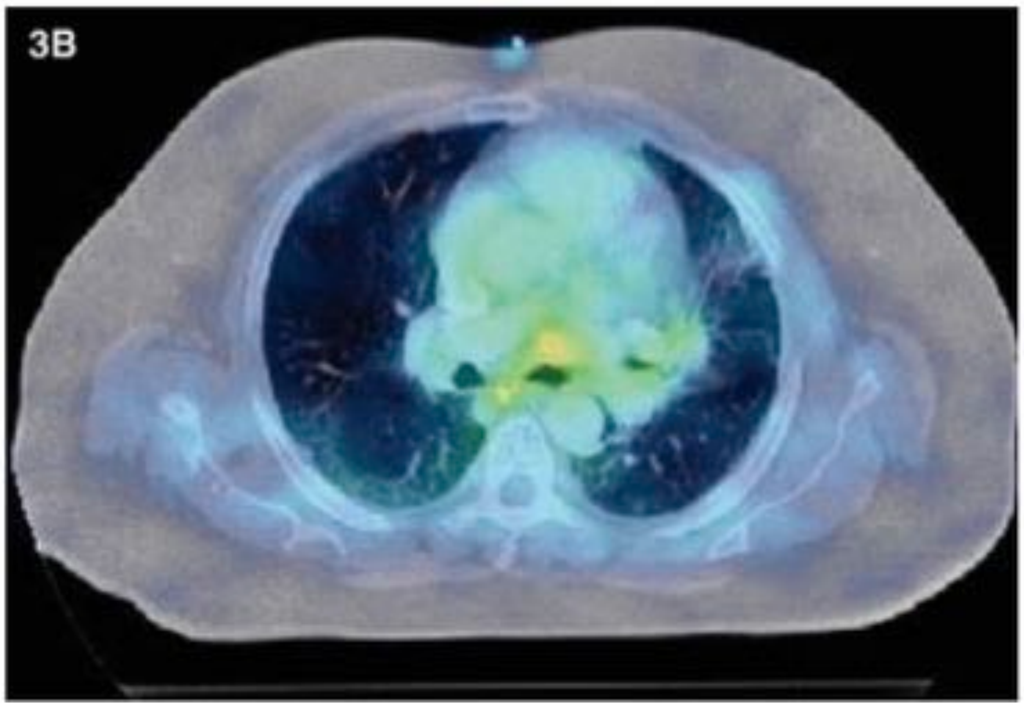




wikipedia.org

Lightness/Hue Encoding

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

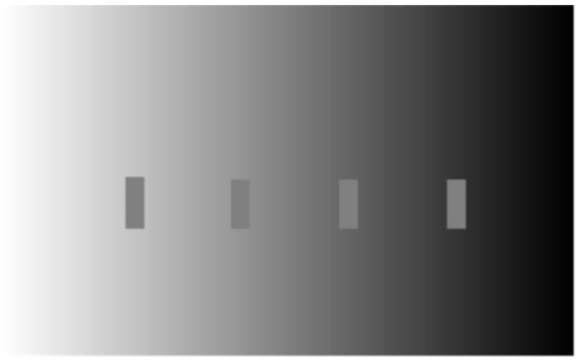


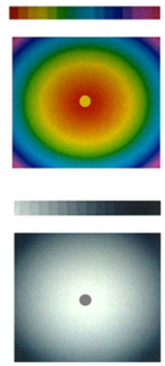
Thomas et al., Mol Im Bio 2003

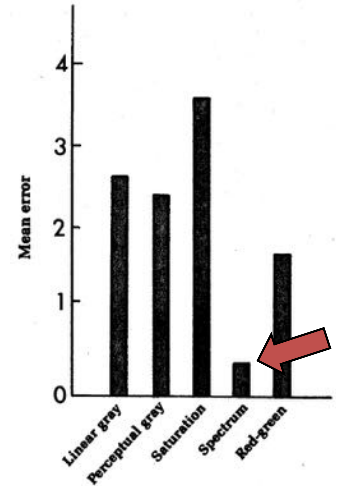
Assumption: Lightness and hue can be perceived more or less independently (at least better than RGB)

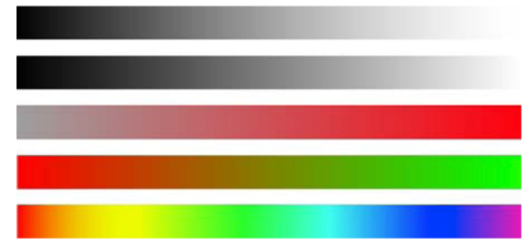
Visual Perception

Metric vs. Form Information

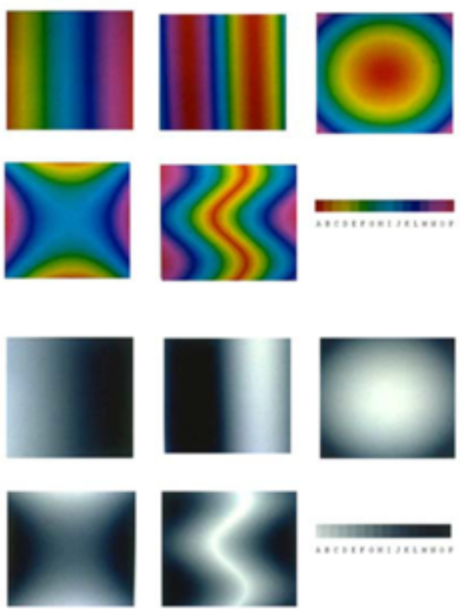




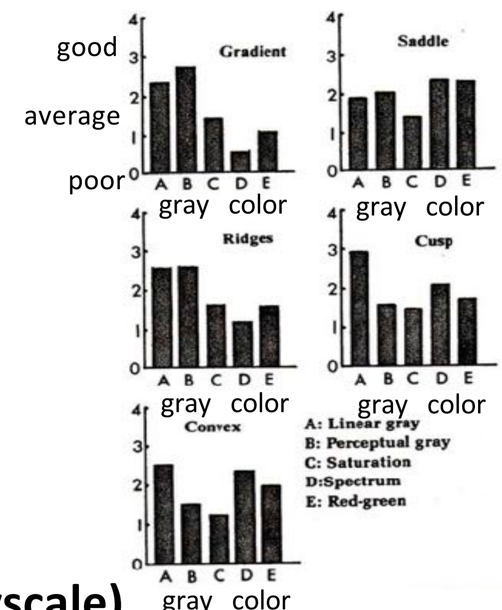




Metric quantities best shown by hue (e.g., spectrum)



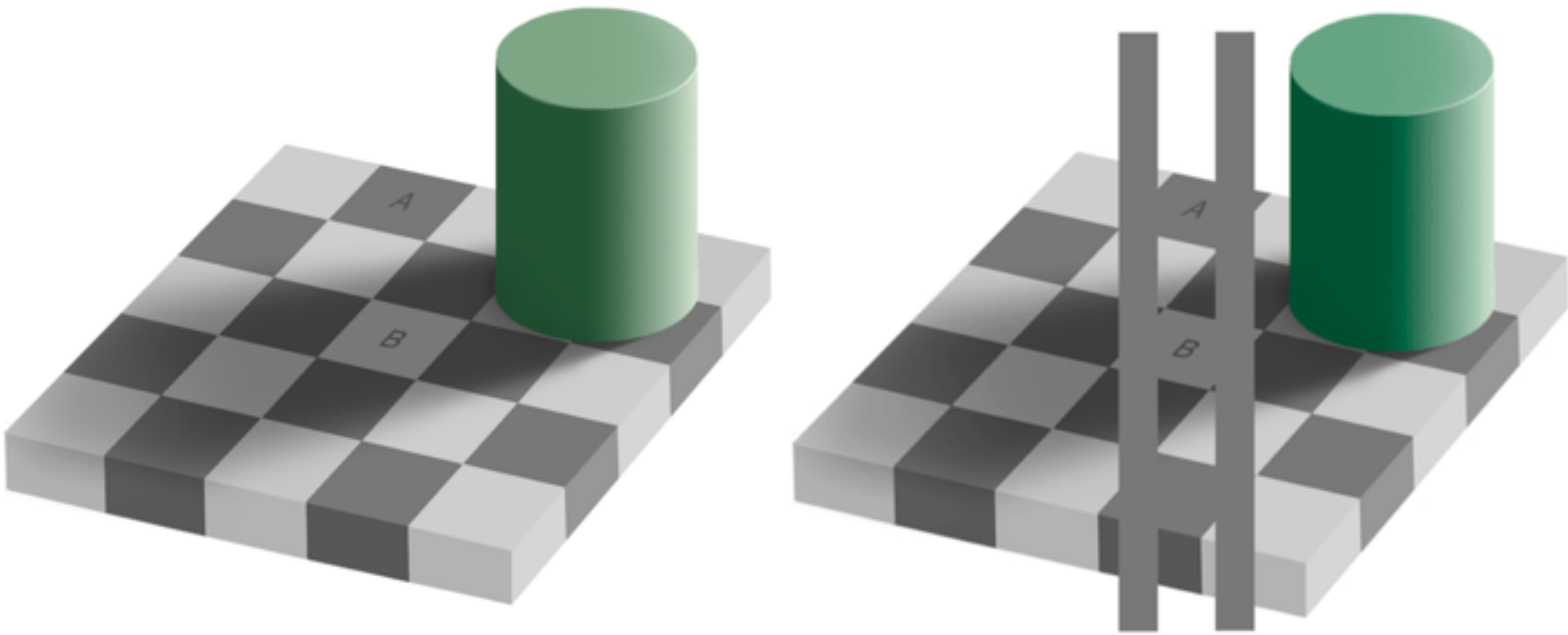
“How effective was the color sequence?”



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

Ware et al., IEEE CG&A 1988

Perception of Luminosity



Context affects perception

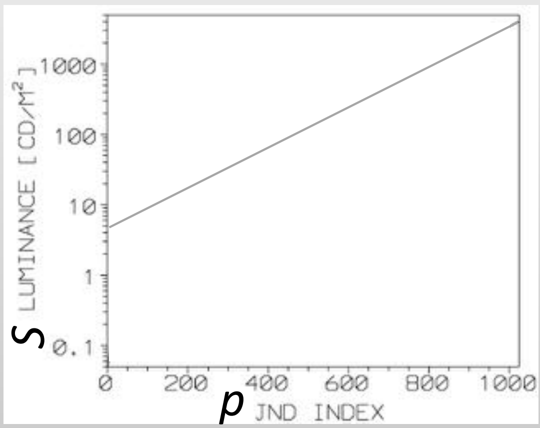
Edward Adelson
wikipedia.org

Perception of Luminosity

Weber-Fechner Law

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

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

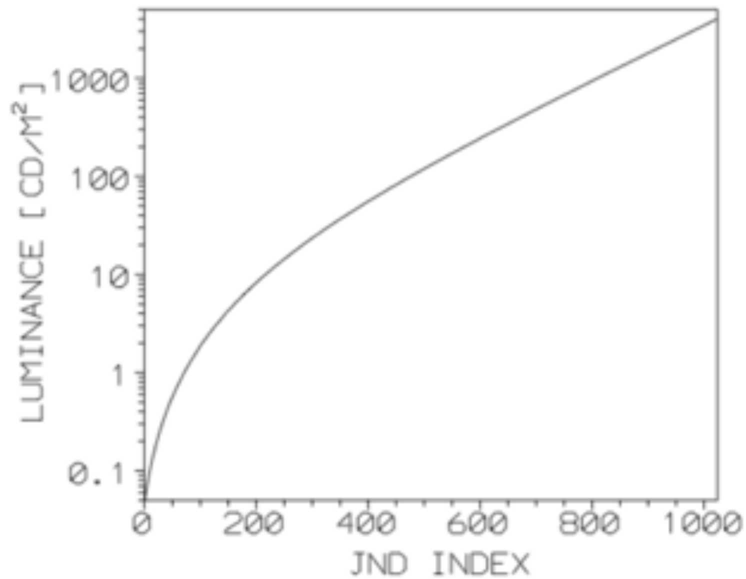


Minimally perceptible dp) difference proportional to a percentage of overall stimulus (dS/S)

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

DICOM GSDF

(Grayscale Standard Display Function)

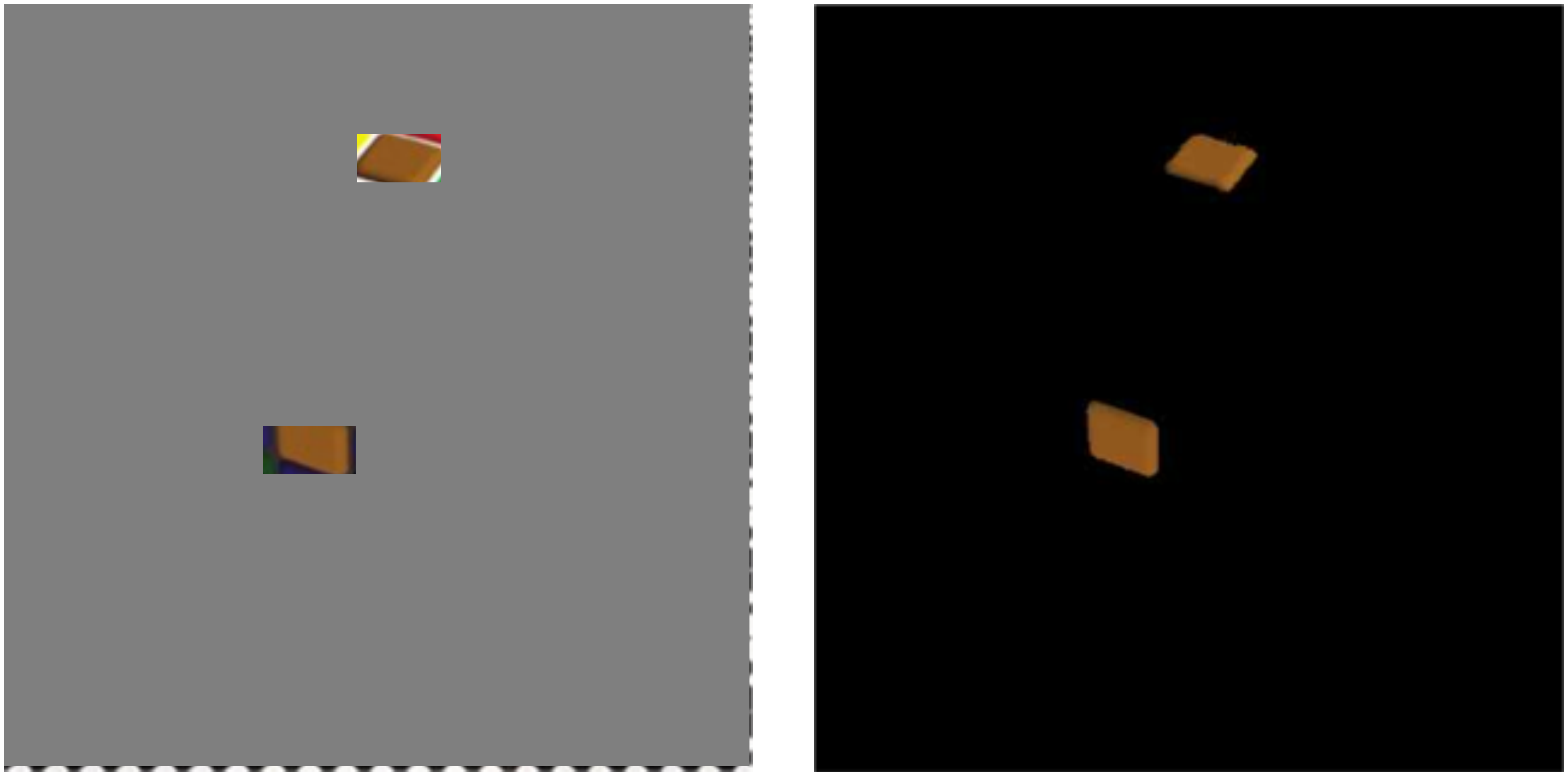


$$\log_{10} L(j) = \frac{a + c\cdot Ln(j) + e\cdot (Ln(j))^2 + g\cdot (Ln(j))^3 + m\cdot (Ln(j))^4}{1 + b\cdot Ln(j) + d\cdot (Ln(j))^2 + f\cdot (Ln(j))^3 + h\cdot (Ln(j))^4 + k\cdot (Ln(j))^5}$$

$JND = j =$ Just Noticeable Difference

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

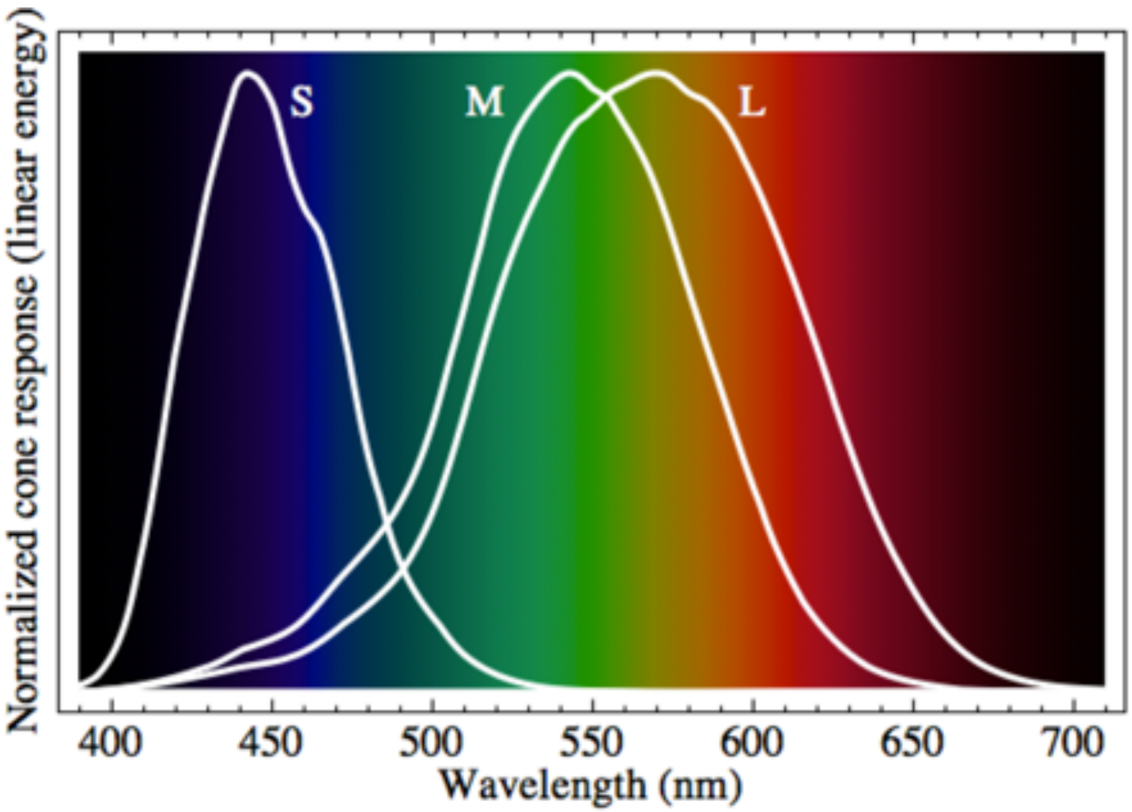
Perception of Color



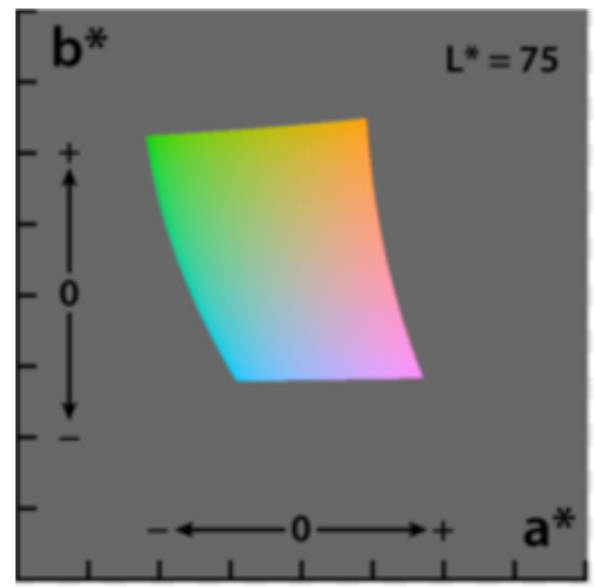
Context affects perception

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

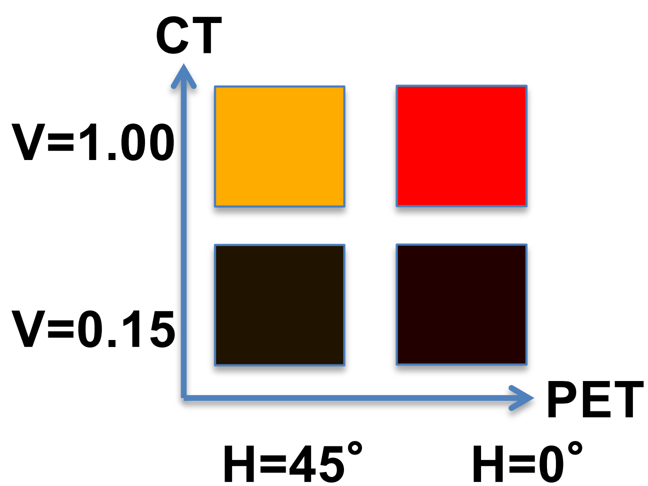
(Note: spectral properties of visible light is not inherently limited three degrees of freedom, this is just a limitation of the human visual system)

wikipedia.org

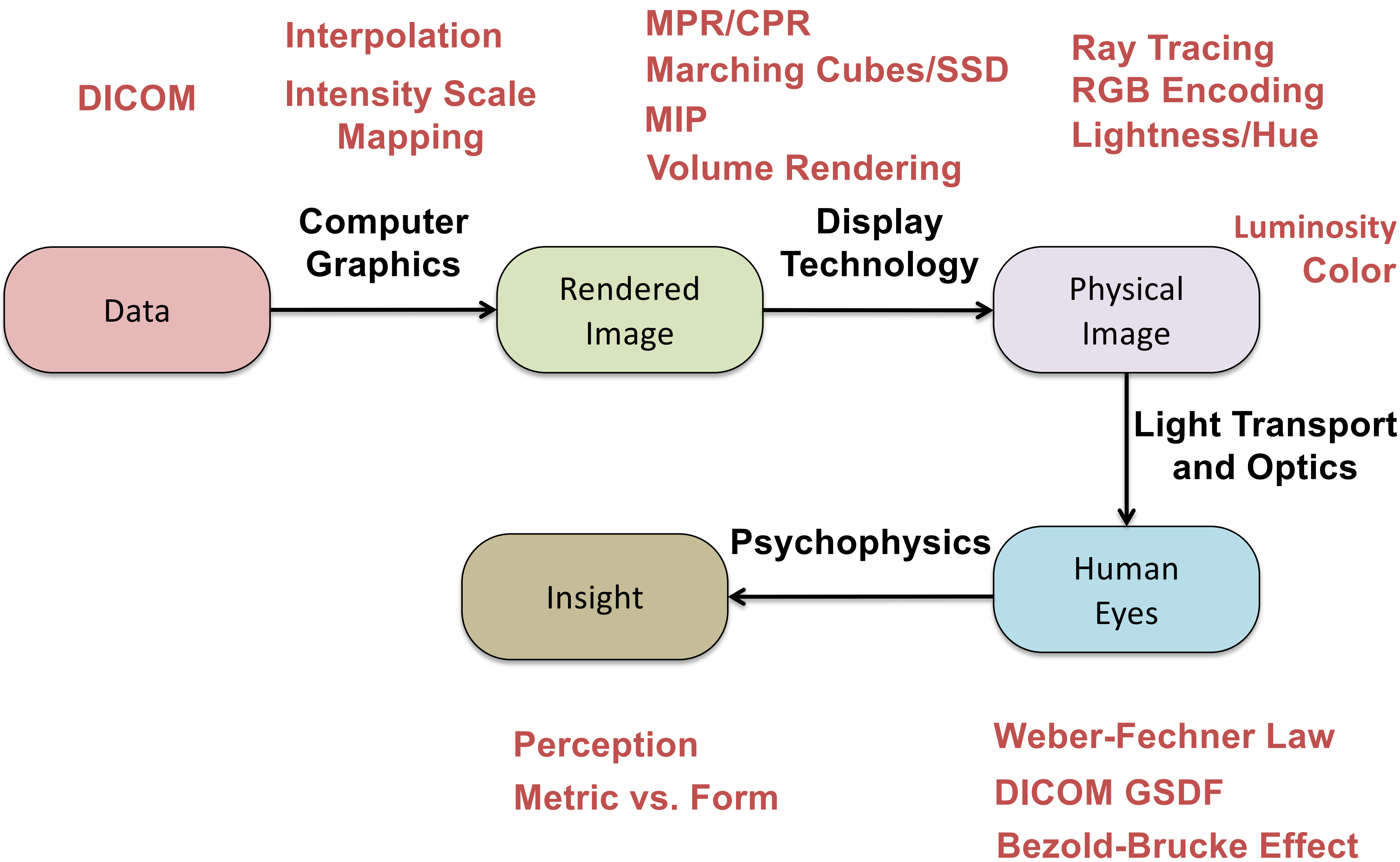
Bezold-Brücke Effect

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



Visualization: From Data to Insight



What does it mean for you?

- Understanding the pipeline from an array of pixel values to human insight
- Visualization covers a wider topics than just computer graphics
- Human perception is an important factor

Next Lecture:

Image Segmentation