An Introduction to Images
Chapter 01/02 G&W

CS6640/BIOENG6640

Guido Gerig, modified from Ross Whitaker

SCI Institute, School of Computing

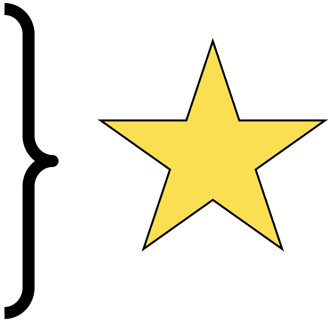
University of Utah

Module 1: Goals

- Understand images as mappings
  - Understand the difference – continuous vs discrete
  - Be able to identify domain and range of an image in a precise way
- Know several examples of images
  - How they are used
  - How they are formed
- Understand domain topology, physical dimensions, and resolution of images
- Understand and be able to use (e.g. reason about and implement)
  - Arithmetic operations, neighborhoods, adjacency, connected components

What Is An *Digital Image*?

- A file you download from the web (e.g. image.jpg)
- What you see on the screen
- An array (regular grid) of data values
- A mapping from one domain to another
  - A discrete sampling (approximation) of a function



Digital Image Acquisition Process

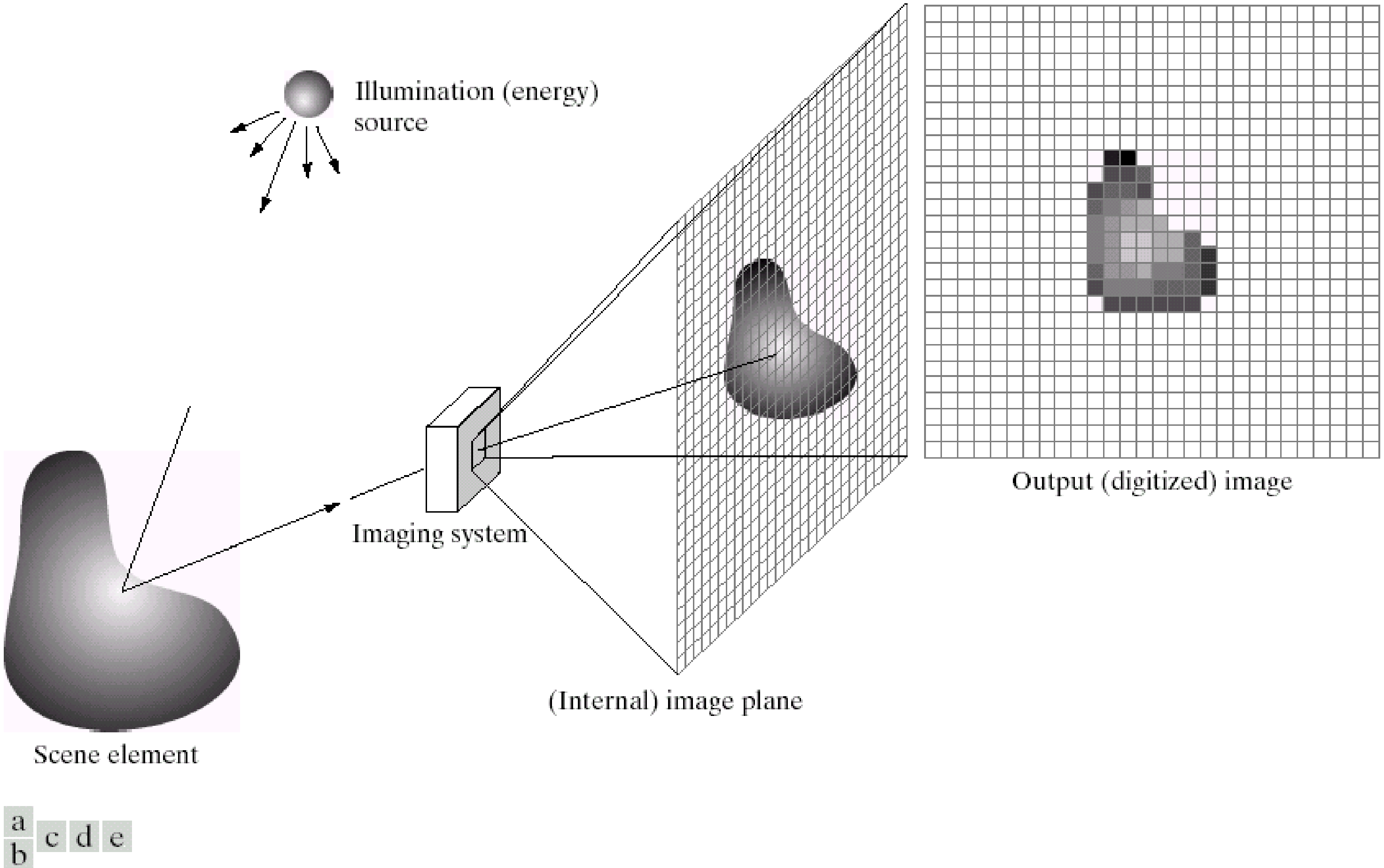
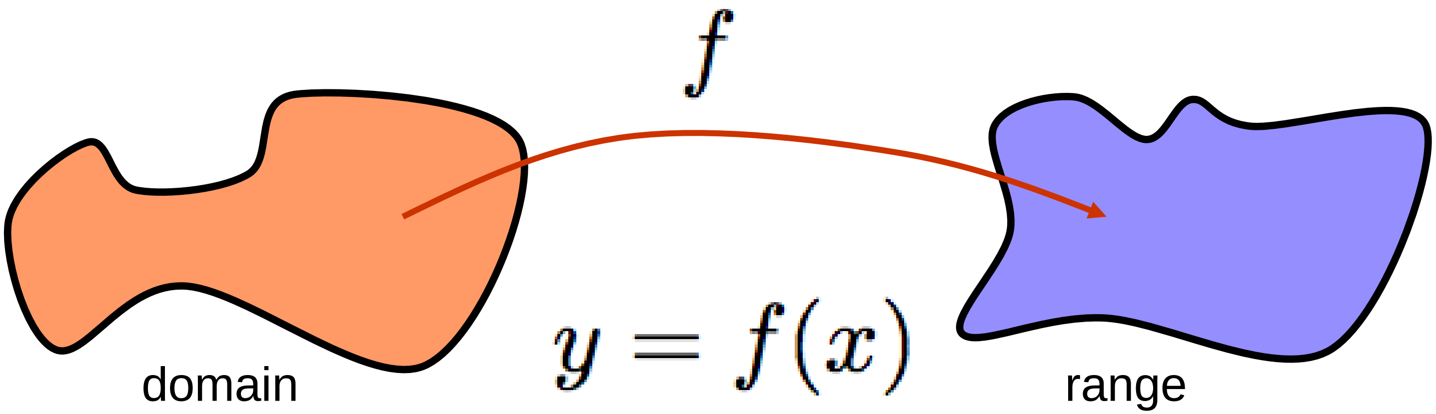


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image As A Mapping (Function)



$$y = f(x)$$

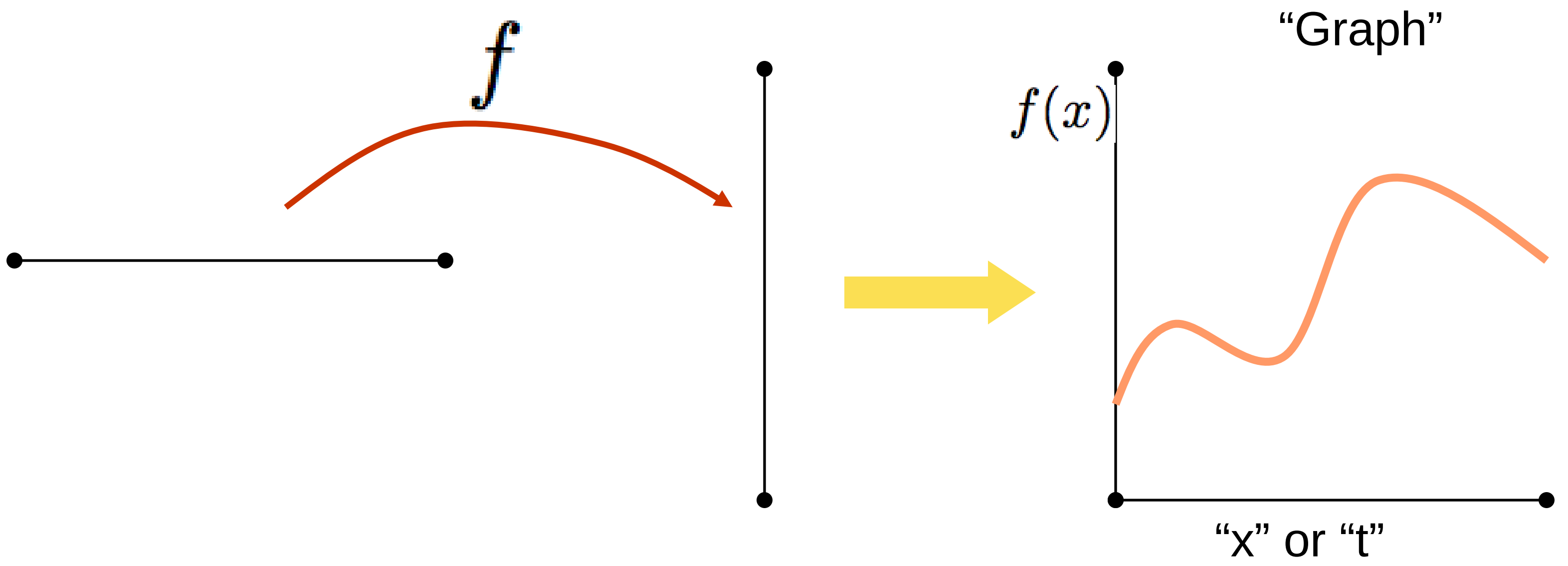
$$f : \mathcal{D} \mapsto \mathcal{R}$$

$$\mathcal{D} \subset \Re^n \text{ and } \mathcal{R} \subset \Re^{\mathrm{m}}$$

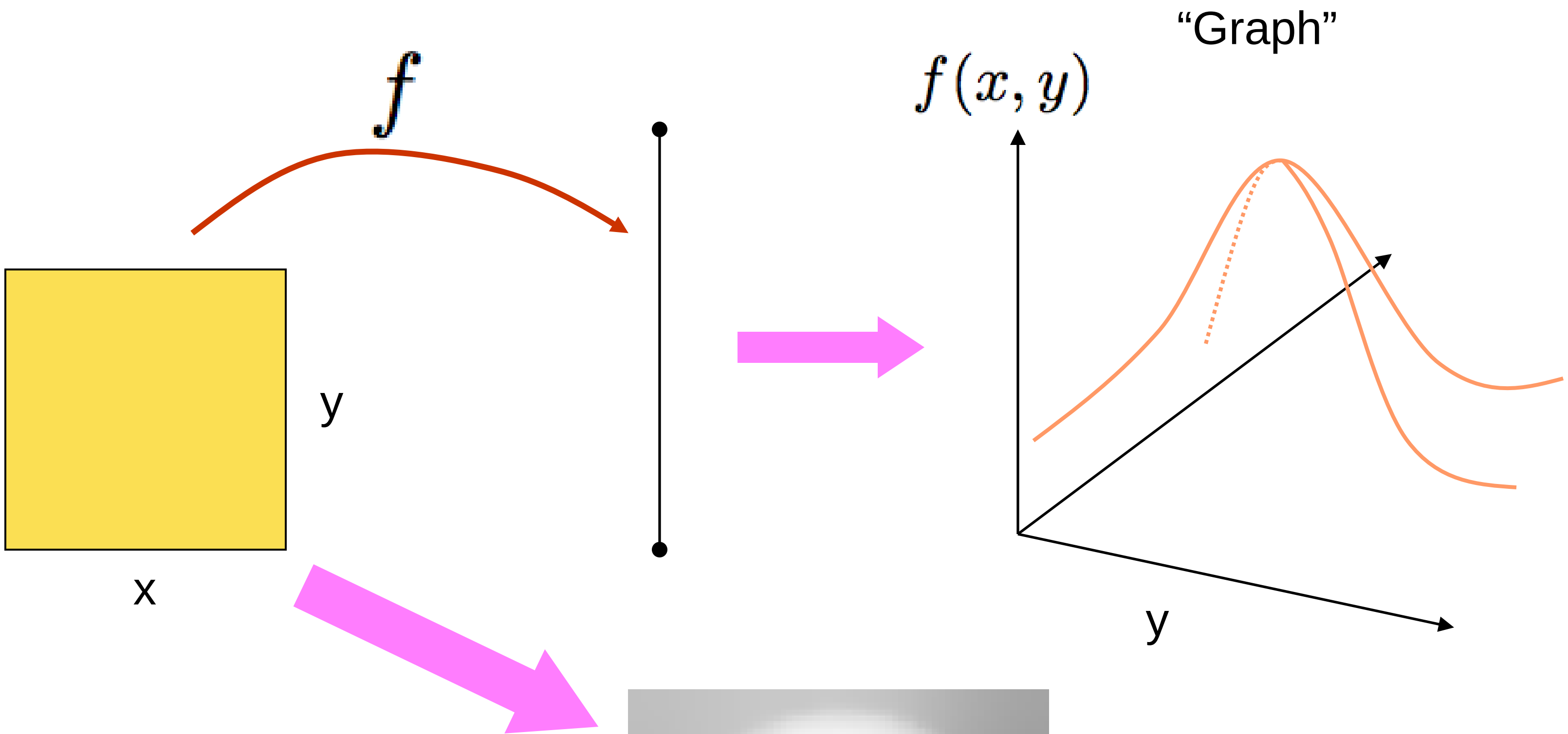
Image As A Mapping: Issues

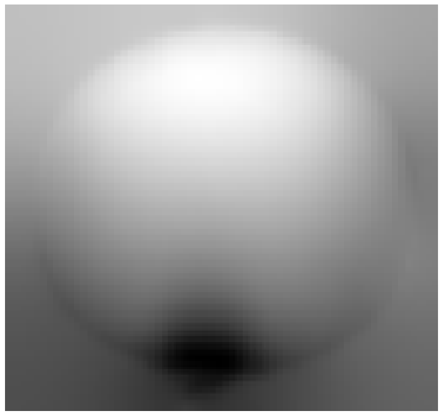
- Dimensionality of domain (n = ?)
- Dimensionality of range (m = ?)
- Typically use shorthand of R^n or R^m
- Discrete or continuous
  - Discrete reasoning/math
  - Continuous math (calculus) –> discrete approximation
  - Issues for both domain and range

Examples of “Images” as Functions



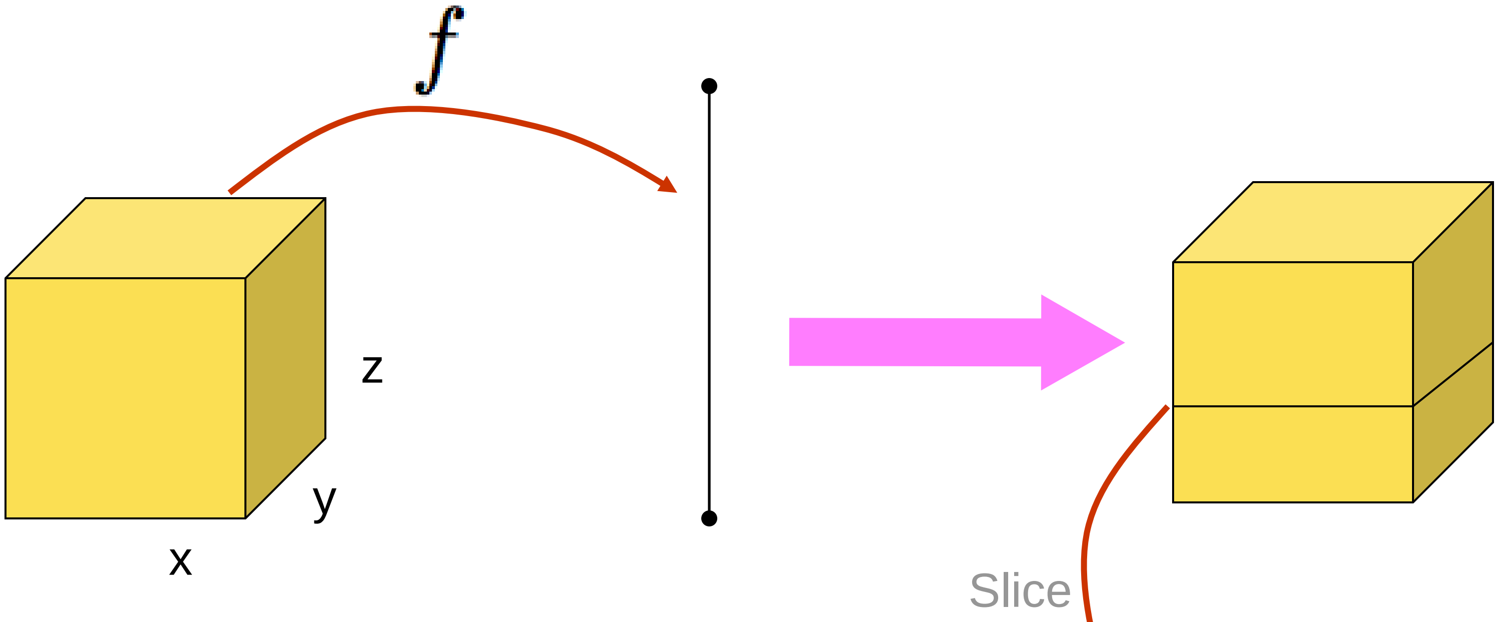
Images As 2D Functions

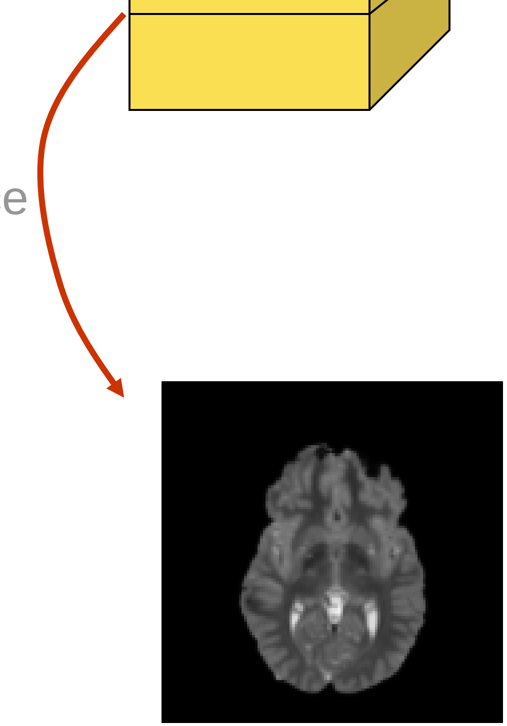




Project brightness on a screen with values at coordinates (x,y) proportional to value of f(x,y)

3D Images – Volumes

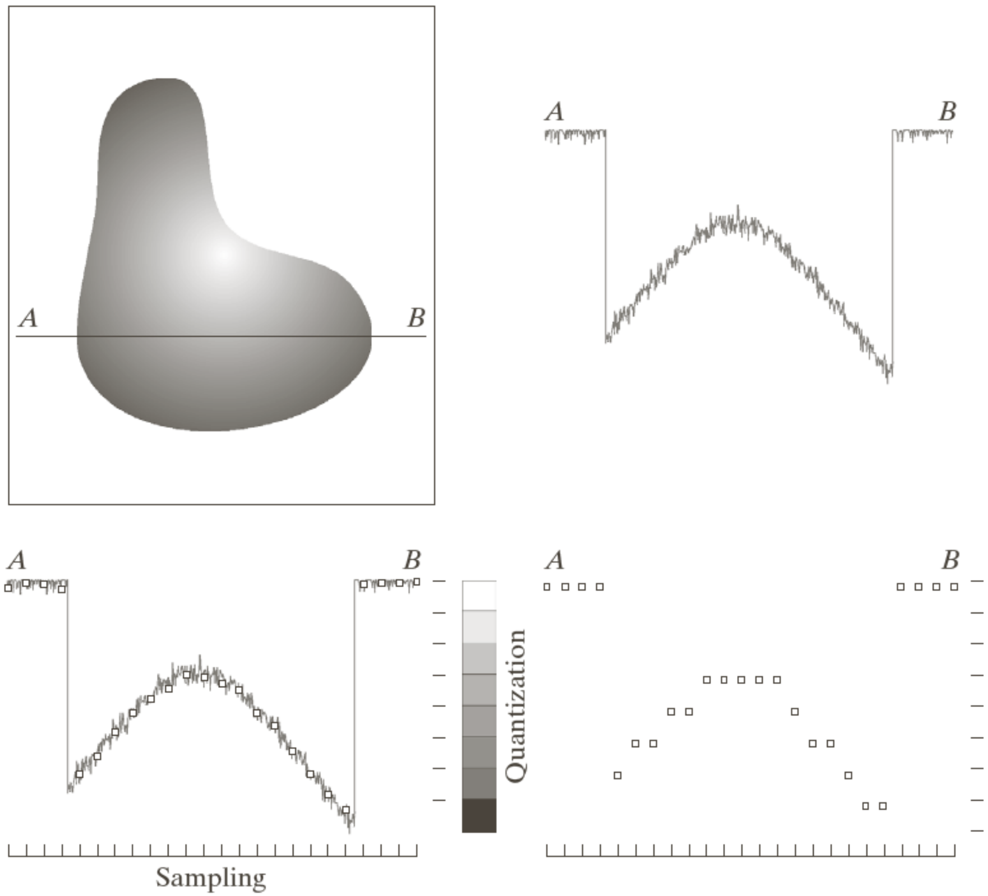




Impossible to graph:

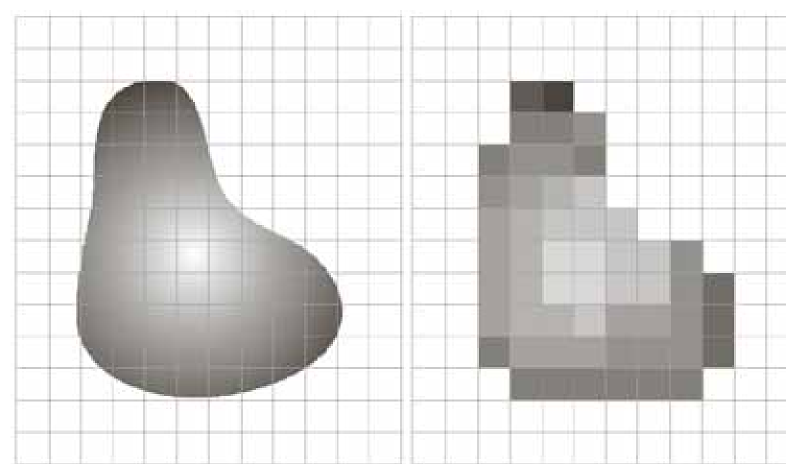
- Slicing
- Volume rendering
- Scientific visualization

Digital Image: Continuous to Discrete



a b
c d

FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.



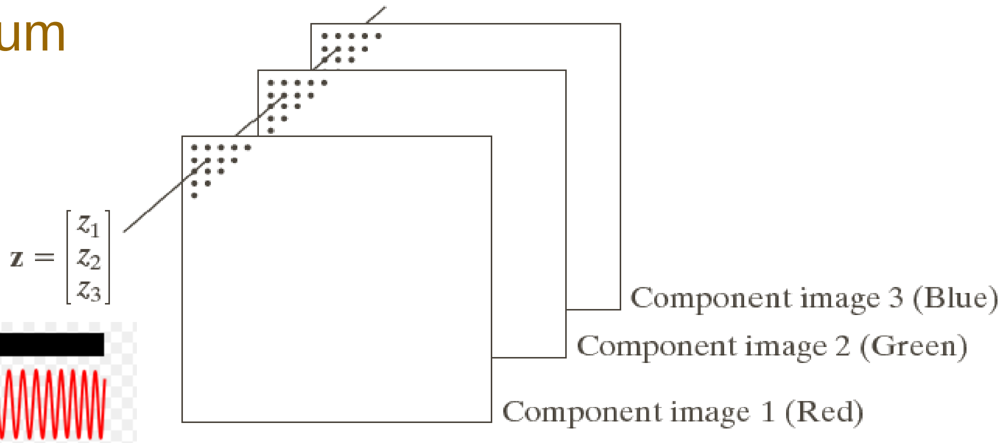
a b

FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Sampling (space) and Quantization (intensity)

Multivalued Images

- Color images: mappings to some subset of R3
  - Color spaces: RBG, HSV, etc.
- Spectral imagery
  - Measure energy at different bands within the electromagnetic spectrum
    - E.g. Satellite images



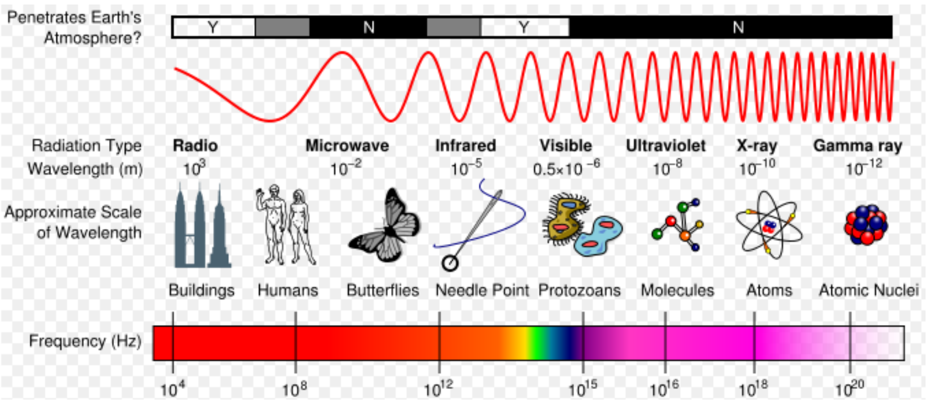
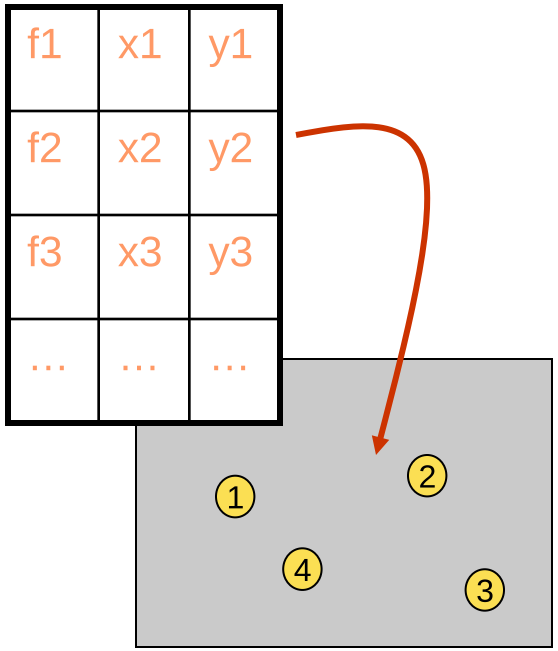


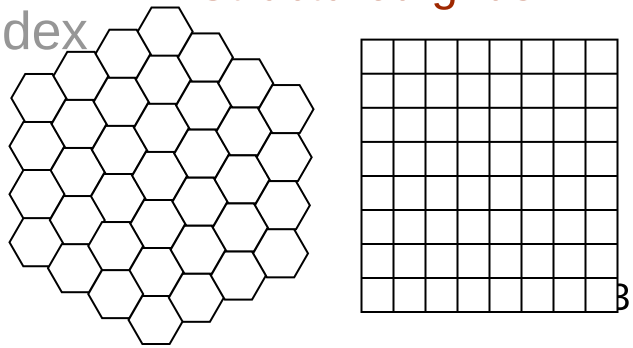
Image As Grid of Values

- Two views
  - Domain is a discrete set of samples
  - Samples are points from an underlying continuous function
- How is the grid organized?
  - Unstructured
    - Points specified by position and value
  - Structured grids
    - Position inferred from structure/index
    - 1D, 2D, 3D, ….
    - Sizes w, w x h, w x h x d

Unstructured grid



Structured grids

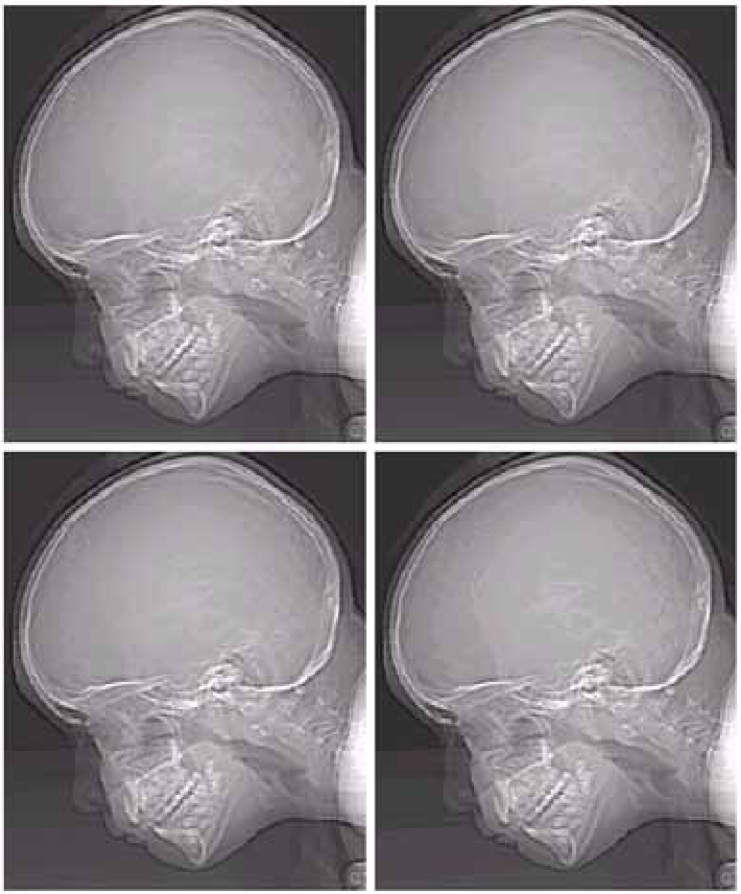


Sampling Effect of spatial resolution

a b
c d

FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.

Quantization: Effect of intensity levels

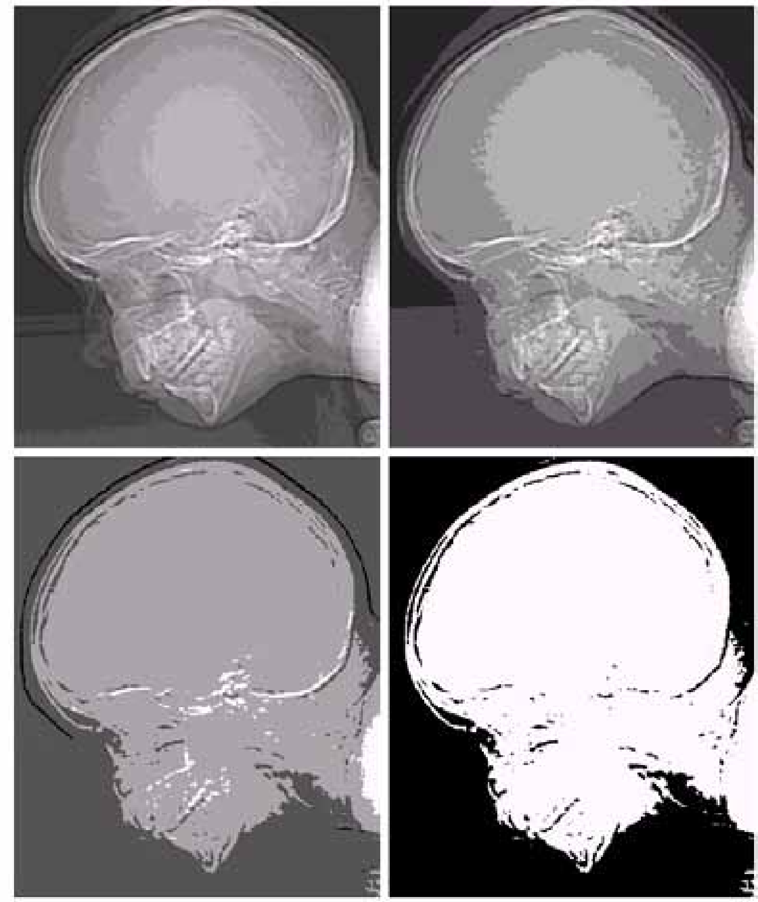


a b
c d

FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

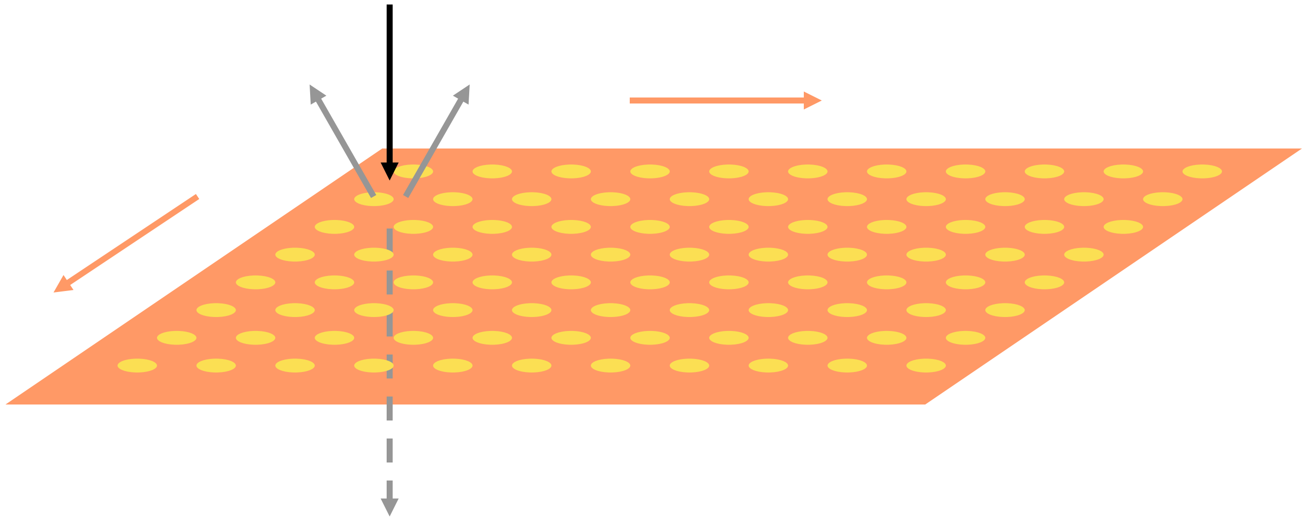
e f
g h

FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

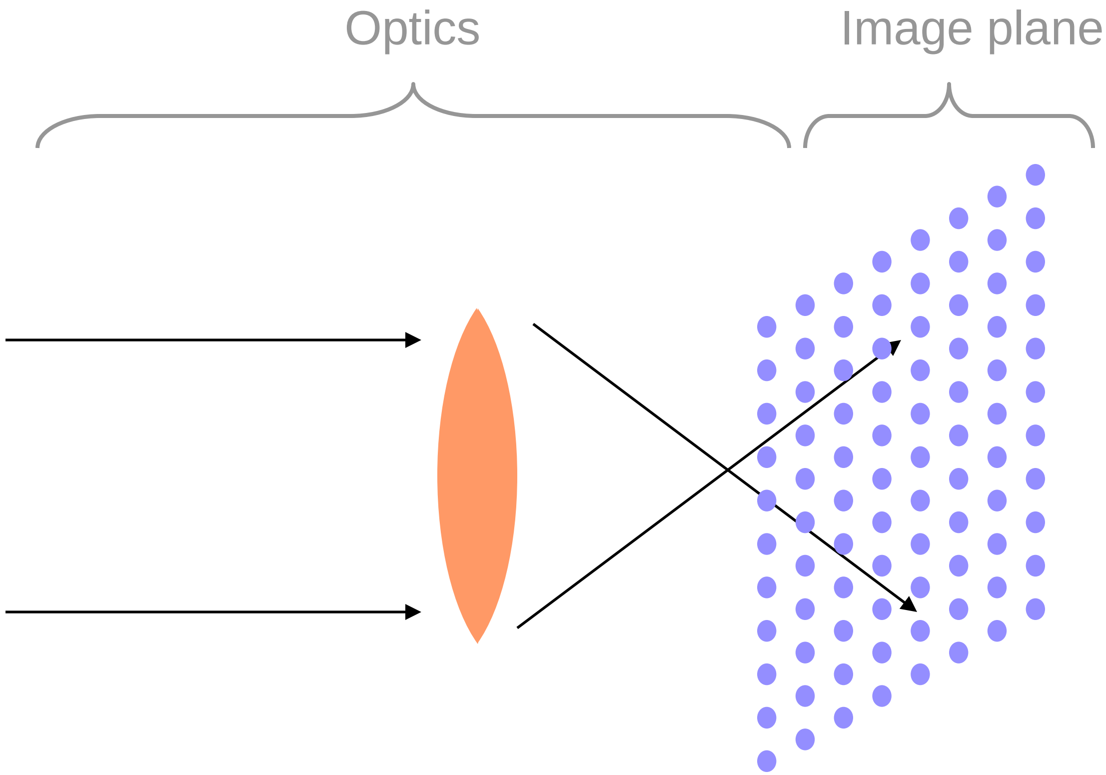


Where Do Digital Images Come From?

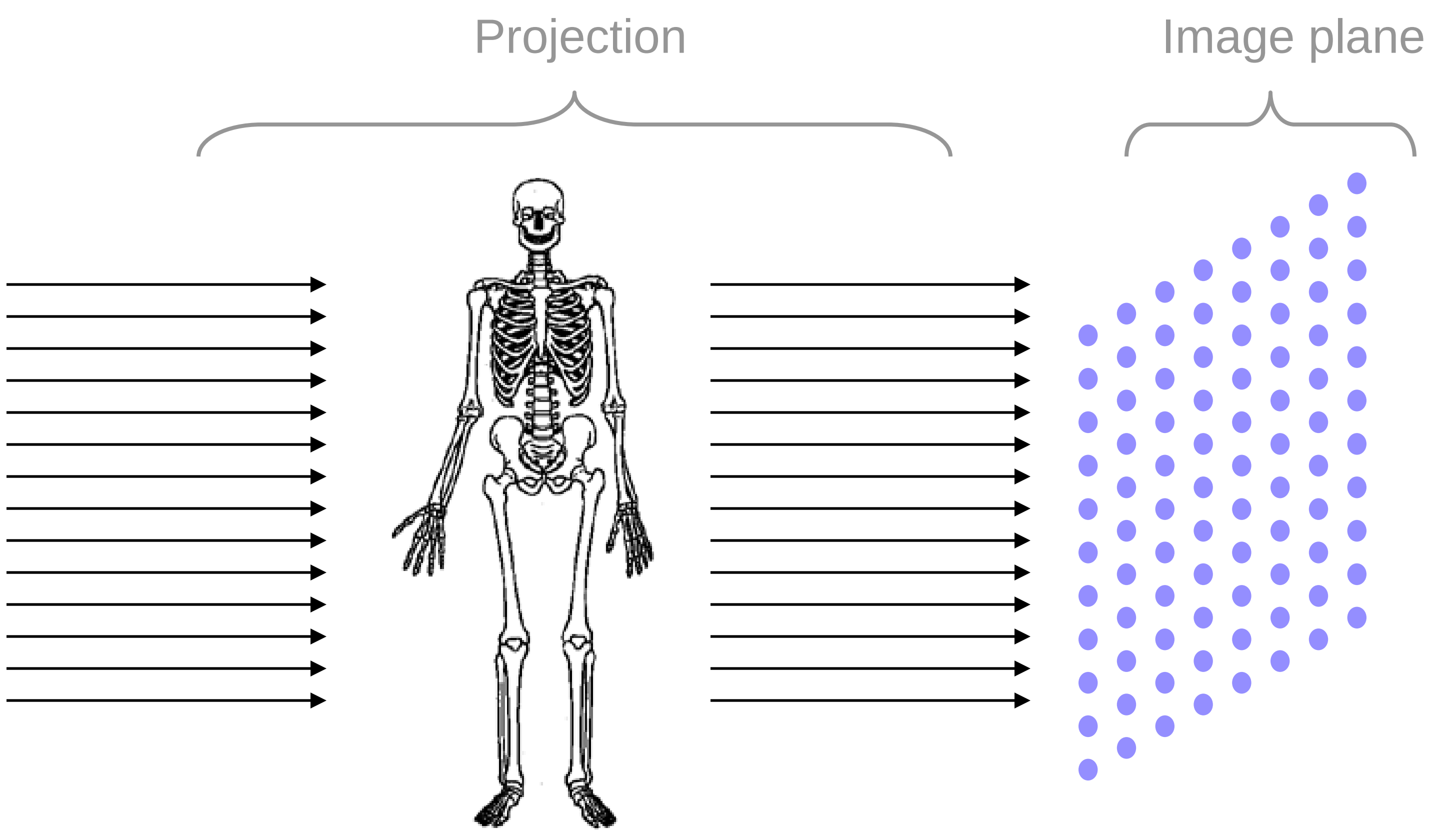
- Digitizing film or paper
  - Rasterize, sample reflectance/transmission on grid



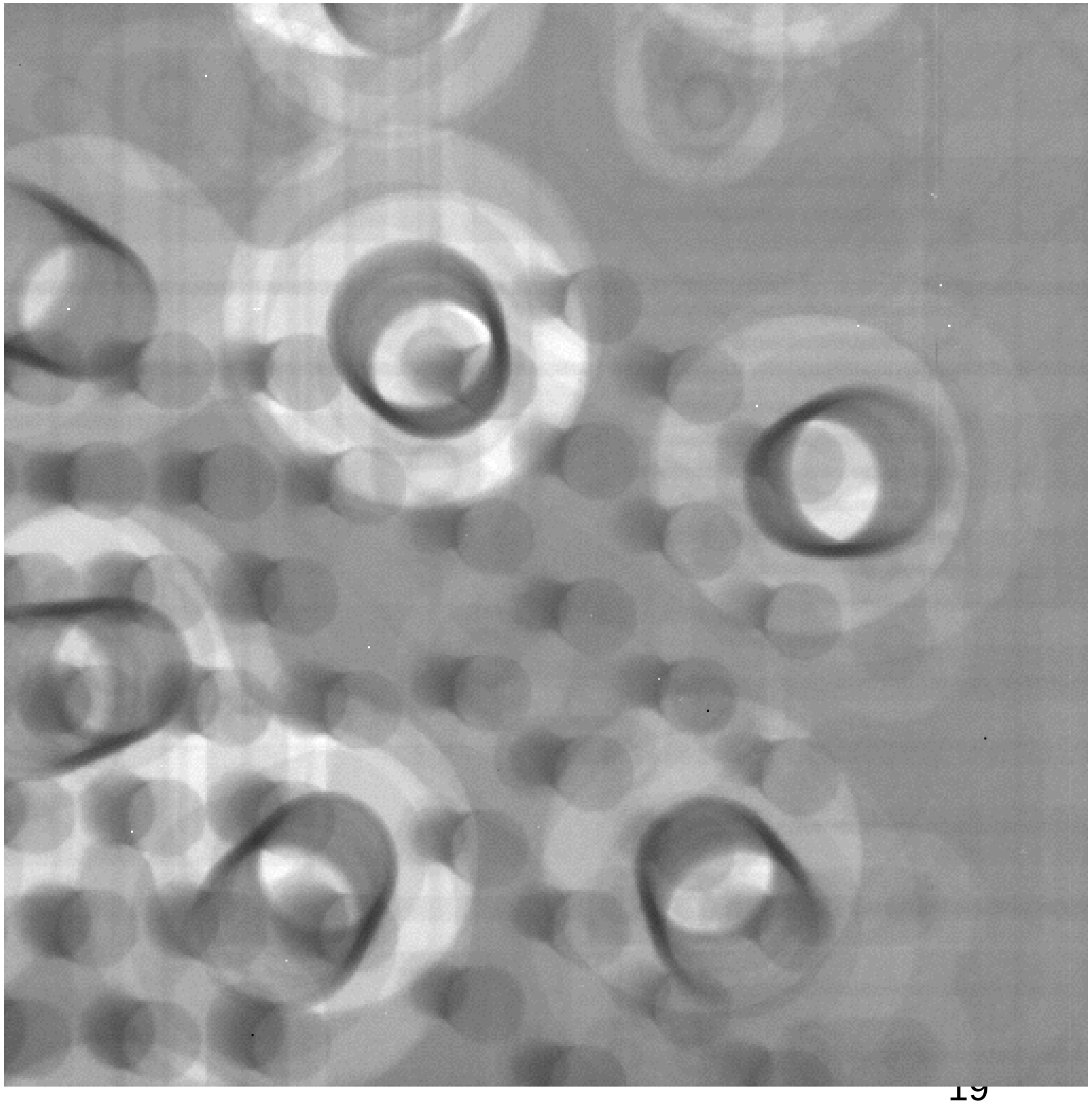
CCD Cameras



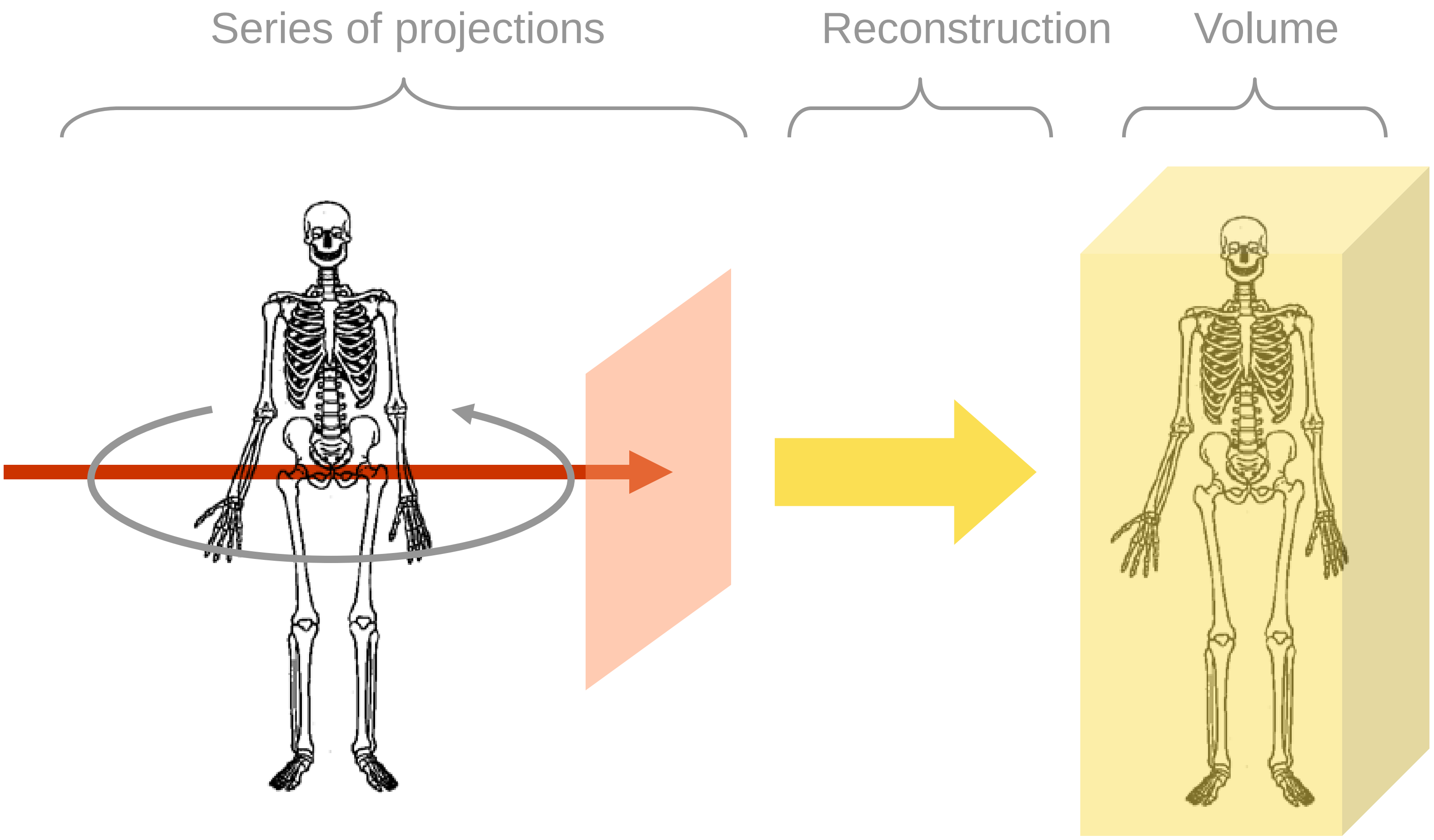
X-Rays



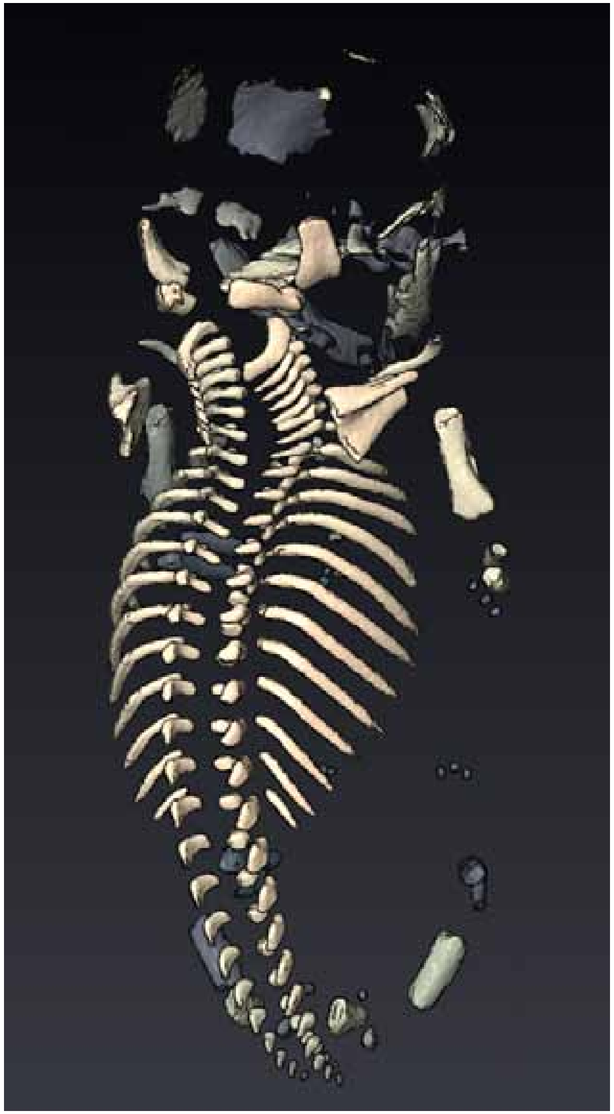
X-Ray Images



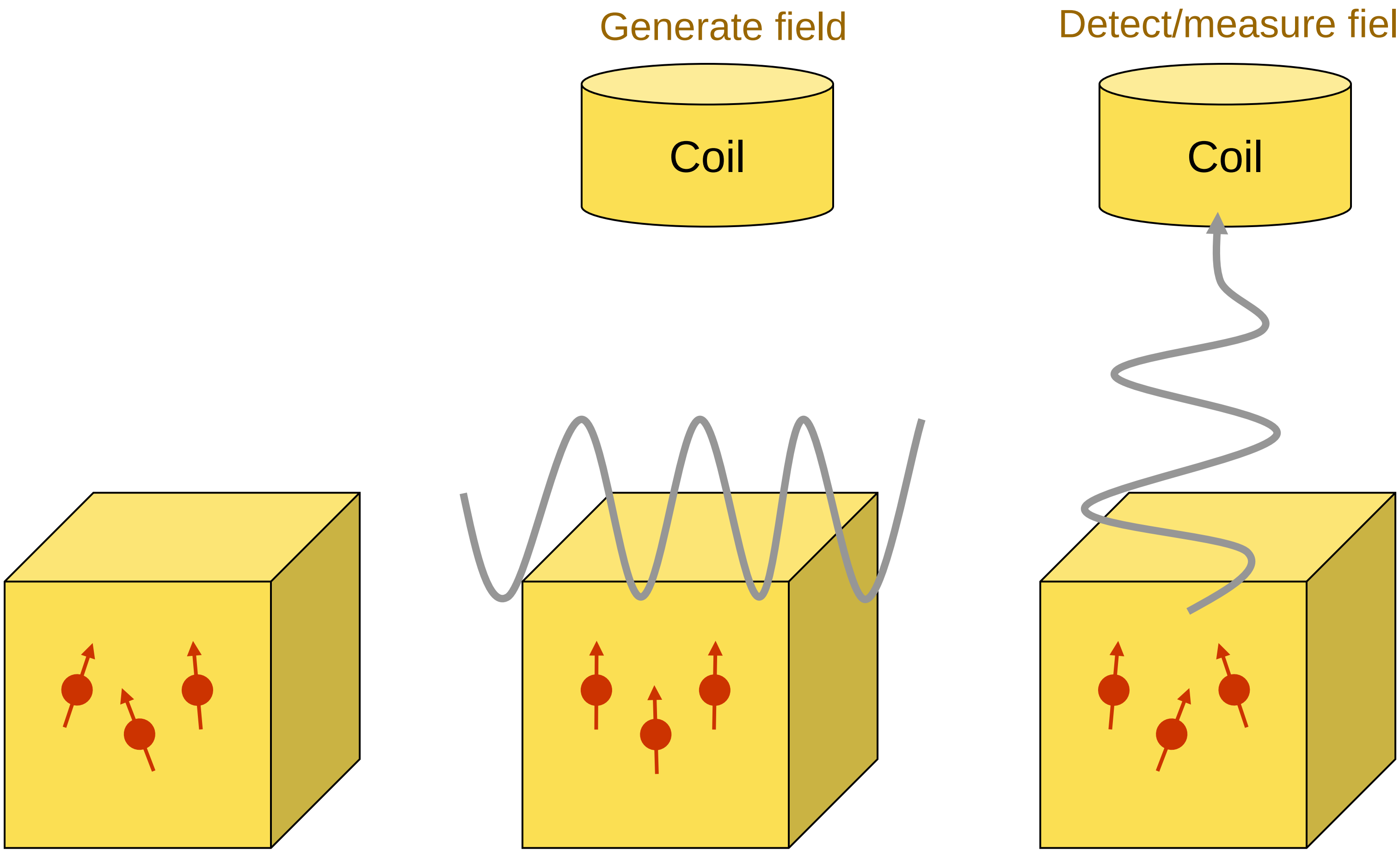
Computed Tomography



CT (CAT)



Magnetic Resonance Imaging

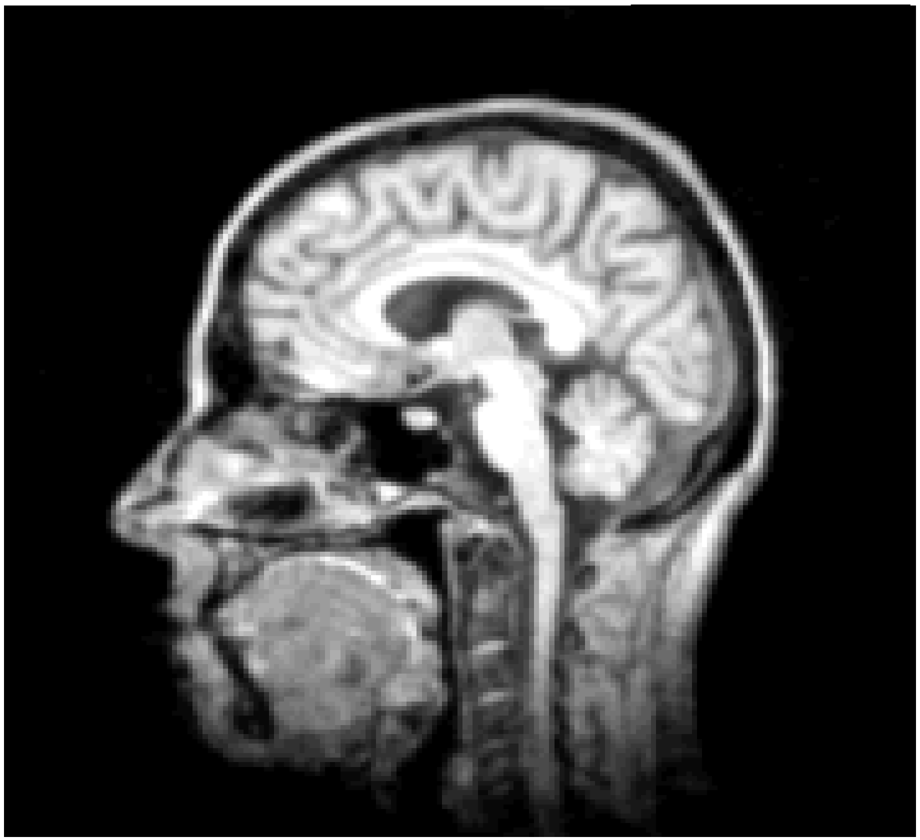


Random orientation (water molecules)

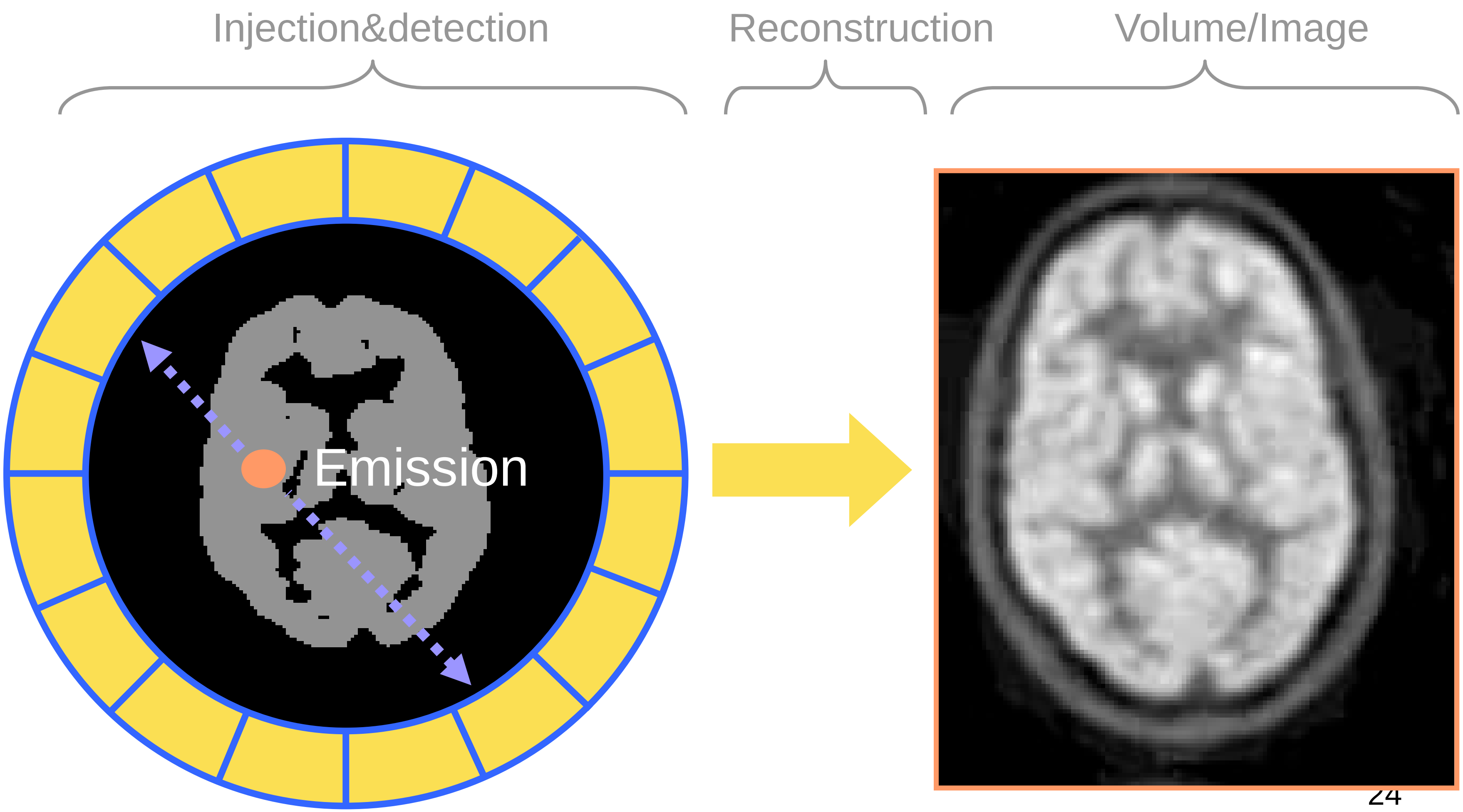
Magnetic field – align spins

Spins become random (generate field)

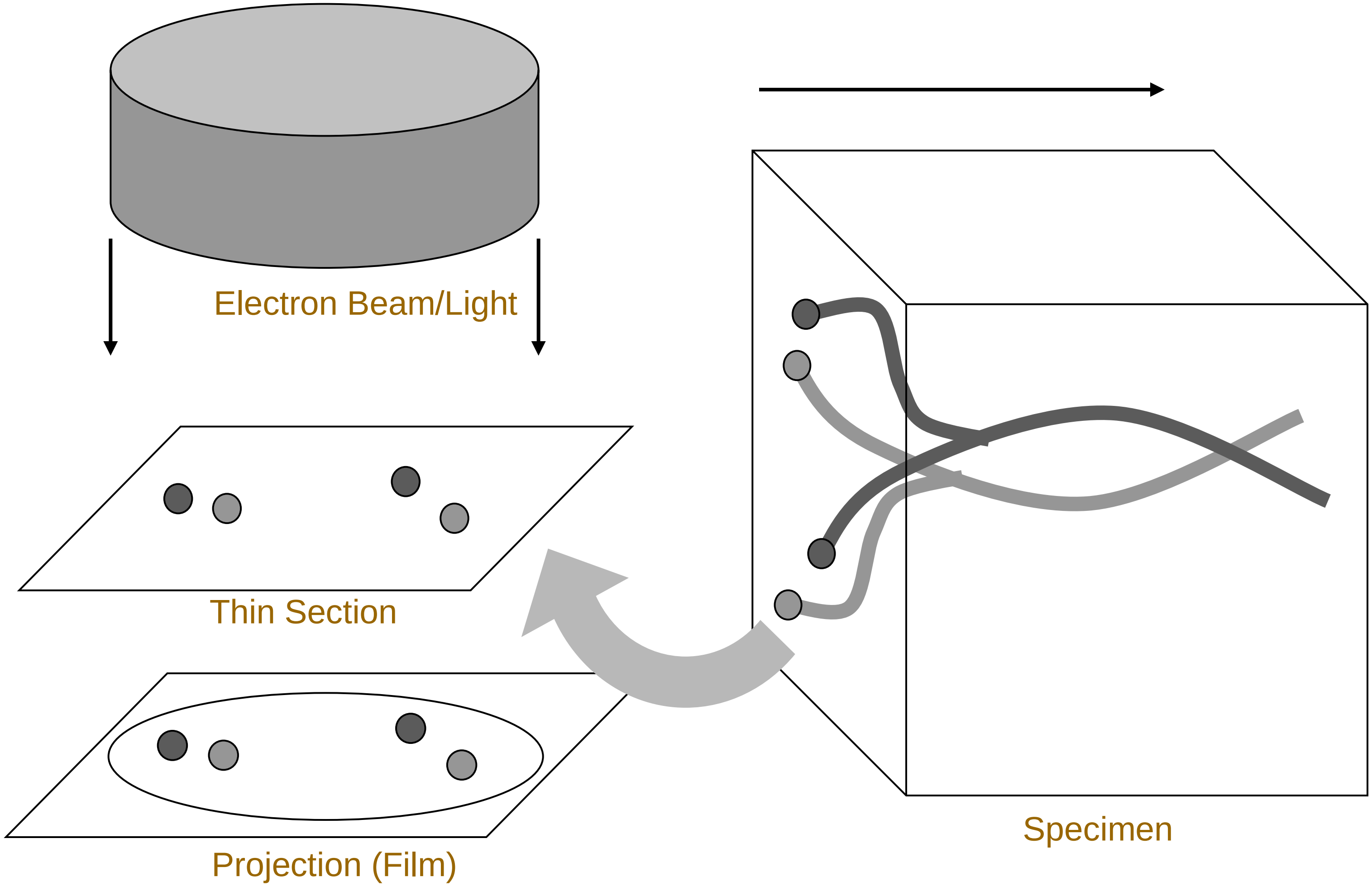
MRI



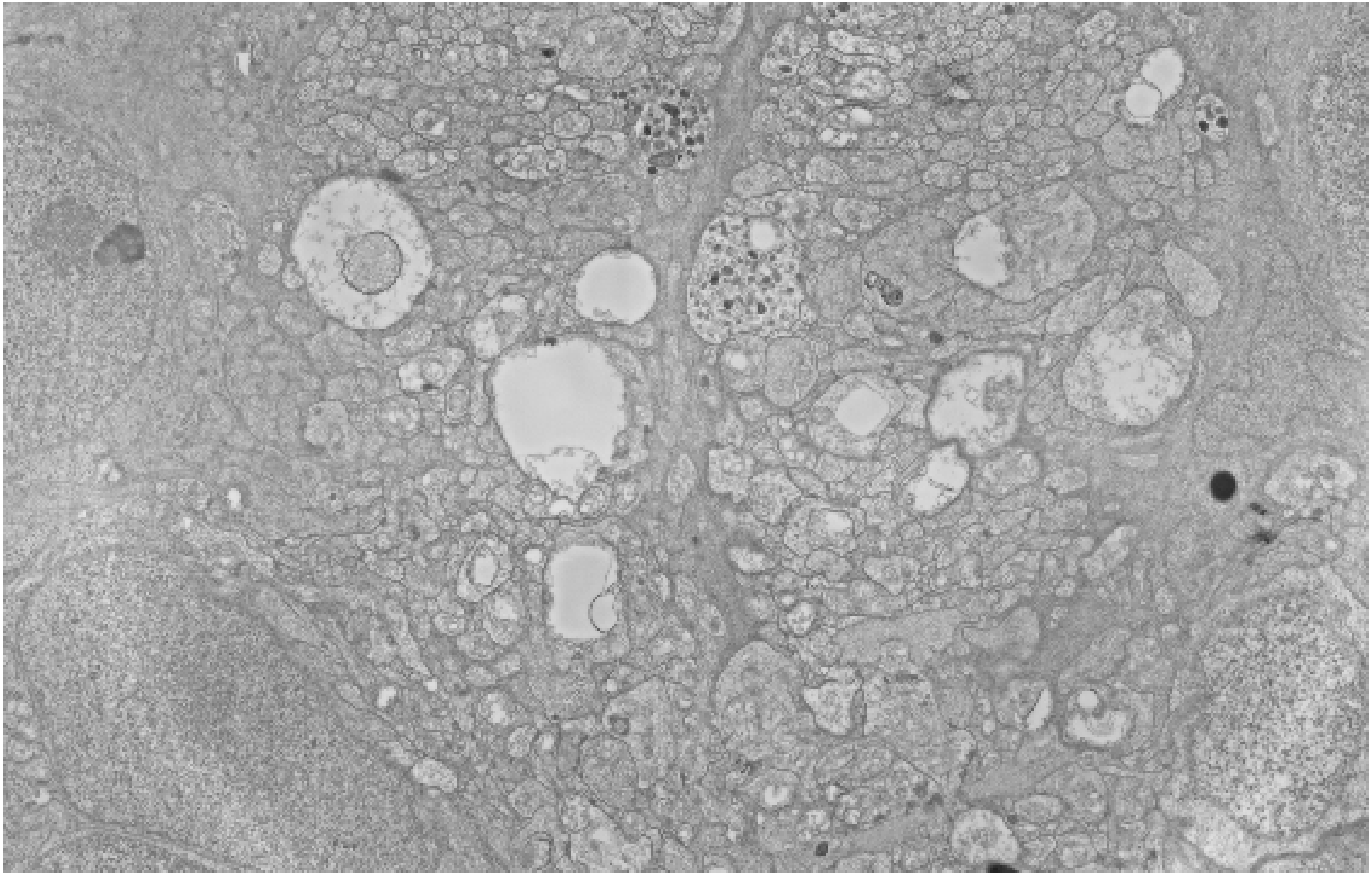
Nuclear Medicine
PET, SPECT, …



Serial Sectioning

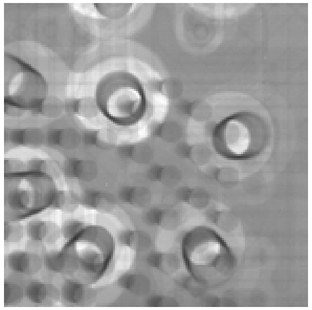


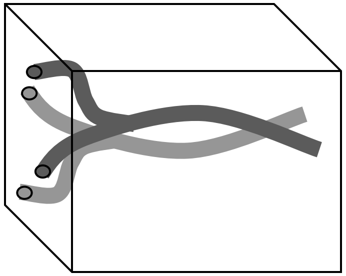
Serial Section Transmission Electron Microscopy

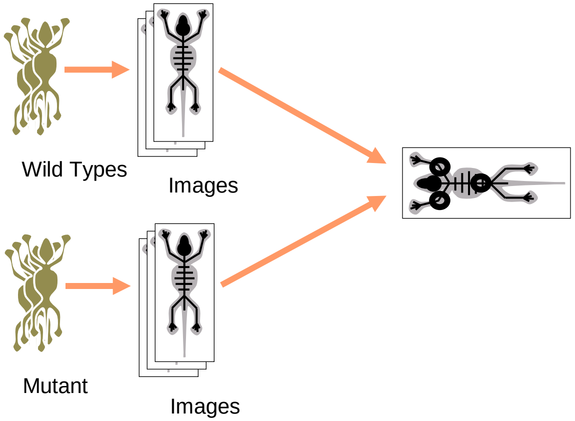


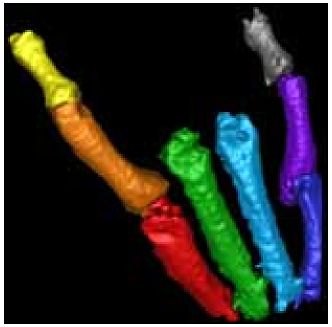
Examples

- Quality control of surface-mount packaging
- Retinal architecture from serial section TEM
- Image-based phenotyping







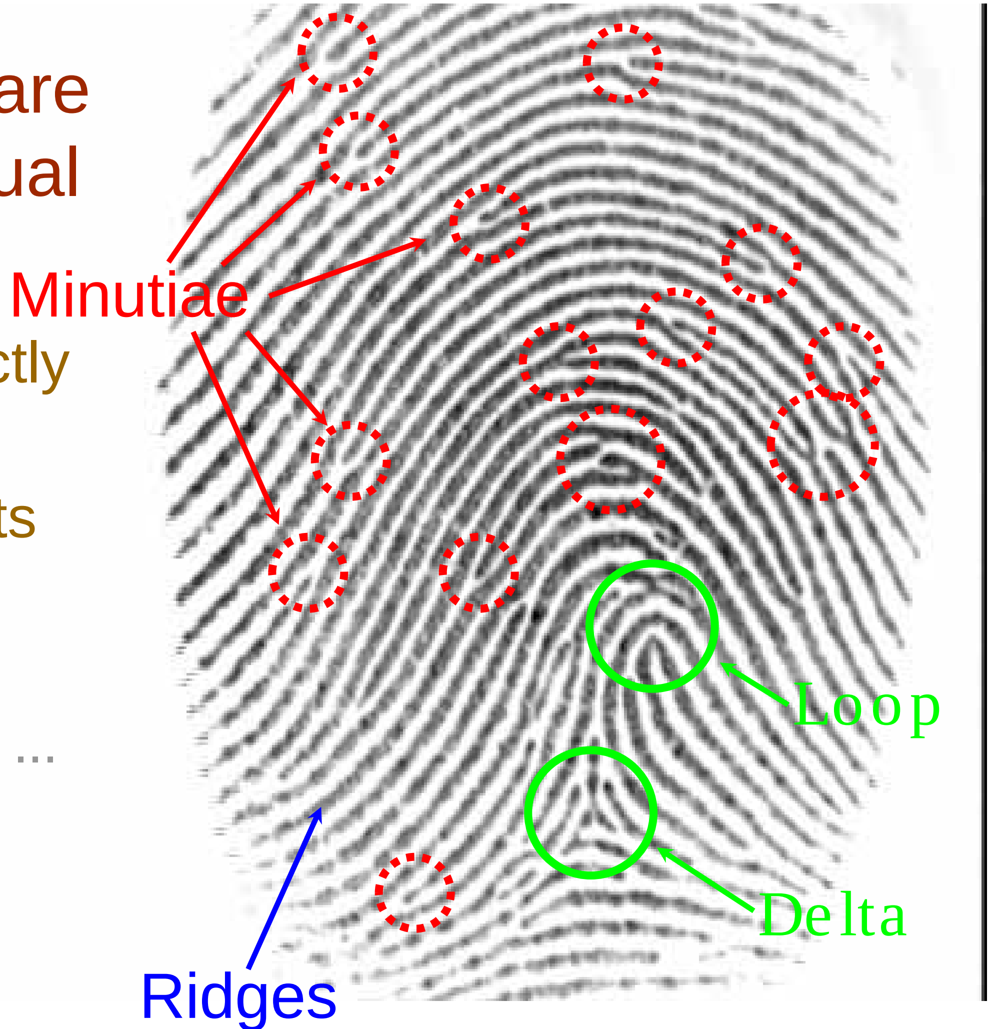


Fingerprint images

- Ink technique
  - spread ink
  - press on paper
  - capture with CCD camera or scanner
- Latent fingerprints
- Live-scan
  - Optical sensors
  - Capacitive sensor
  - Thermal sensor
  - Pizoelectric (pressure)

Fingerprint matching

- Fingerprint patterns are unique to the individual
- Matching
  - using the ridges directly is hard
  - often singularity points are used
    - Local: Minutiae
    - Global: Loop, delta, ...



Array vs. Matrix Operations

$$\underbrace{\begin{pmatrix} a & b \\ c & d \end{pmatrix}}_{A} \times \underbrace{\begin{pmatrix} x & y \\ w & z \end{pmatrix}}_{X} = \begin{pmatrix} ax+bw & ay+bz \\ cx+dw & cy+dz \end{pmatrix}$$

Matrix multiply
(MATLAB A*X)

$$\underbrace{\begin{pmatrix} a & b \\ c & d \end{pmatrix}}_{A} \times \underbrace{\begin{pmatrix} x & y \\ w & z \end{pmatrix}}_{X} = \begin{pmatrix} ax & by \\ cw & dz \end{pmatrix}$$

Array multiply
(MATLAB A.*X)

Images can be represented as matrices, but the operations refer to array operations unless otherwise specified

Arithmetic Operations on Images

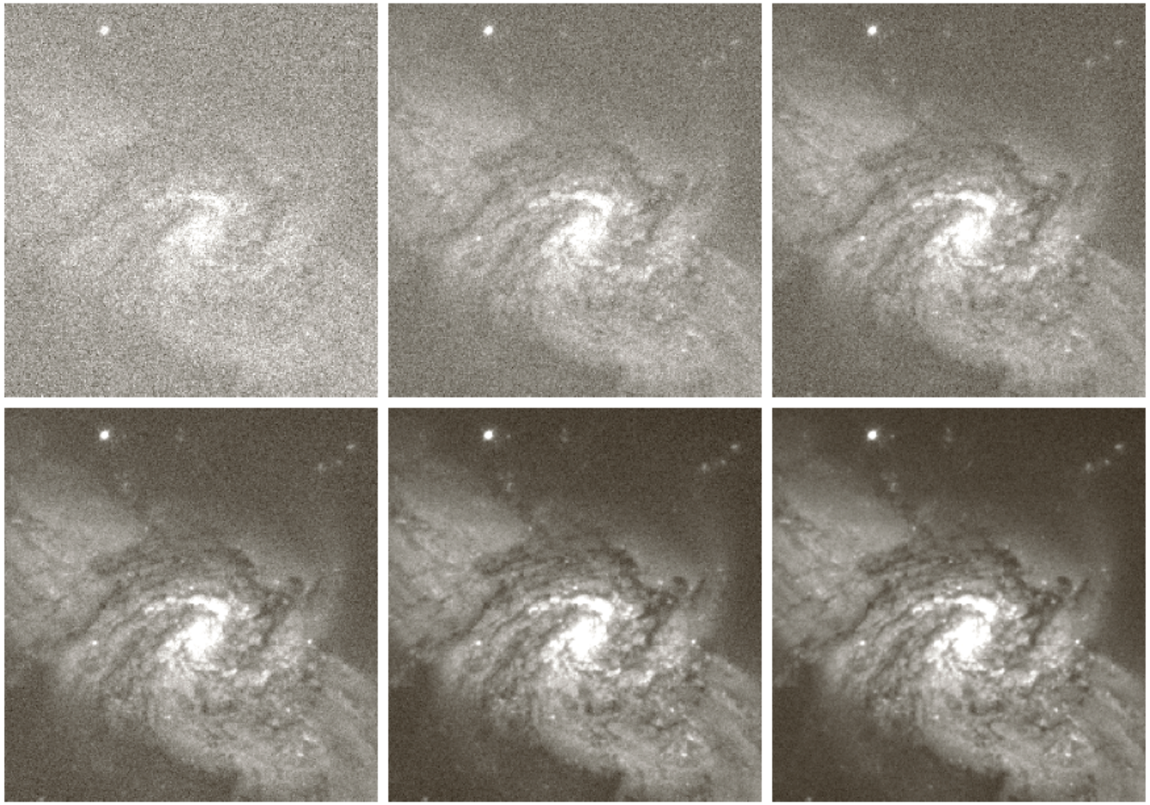
- Arithmetic operations on pixel values
  - Multiple images with the same domain
  - Image become arguments
    - Implied that the operation is applied pointwise across the domain
  - Addition, subtraction, multiply, divide, boolean

$$h = f + g \;\Rightarrow\; h(i,j) = f(i,j) + g(i,j)$$

$$\forall (i,j) \in \mathcal{D}$$

Arithmetic operations: f + g

Averaging (adding) multiple images can reduce noise



a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Arithmetic operations: f - g

Digital Subtractive Angiography

(DSA)

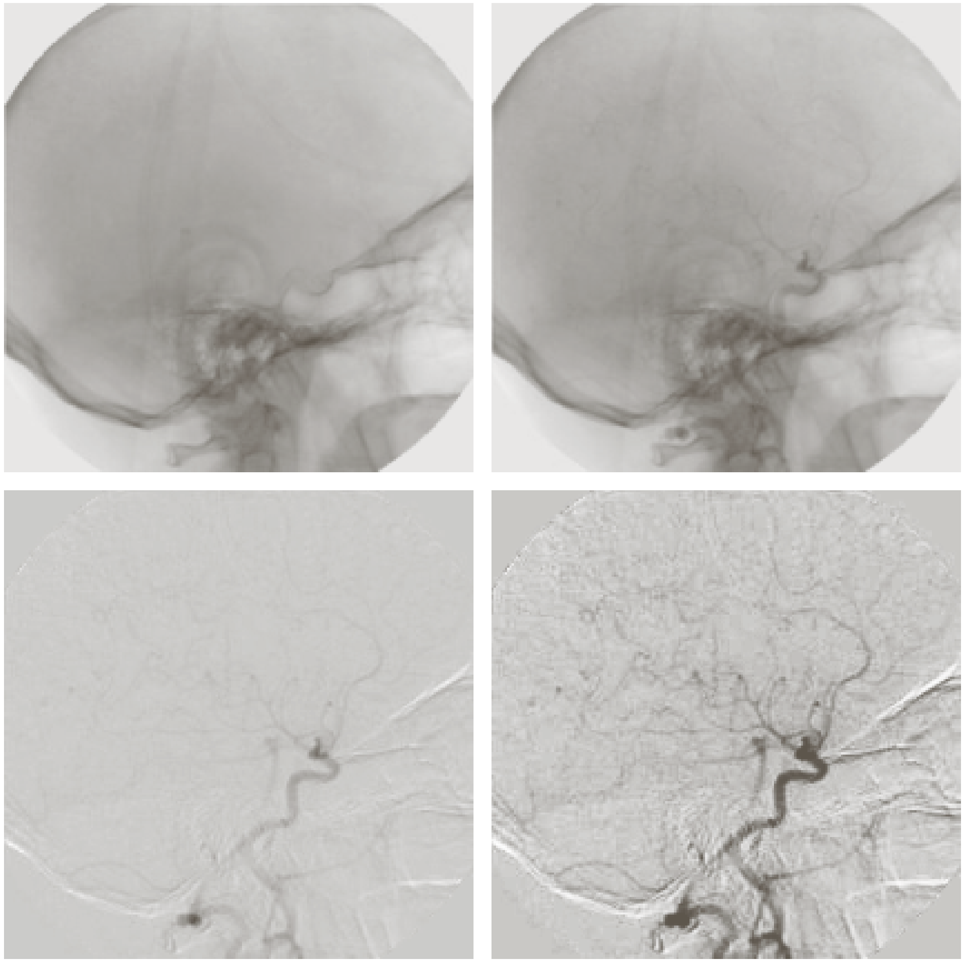
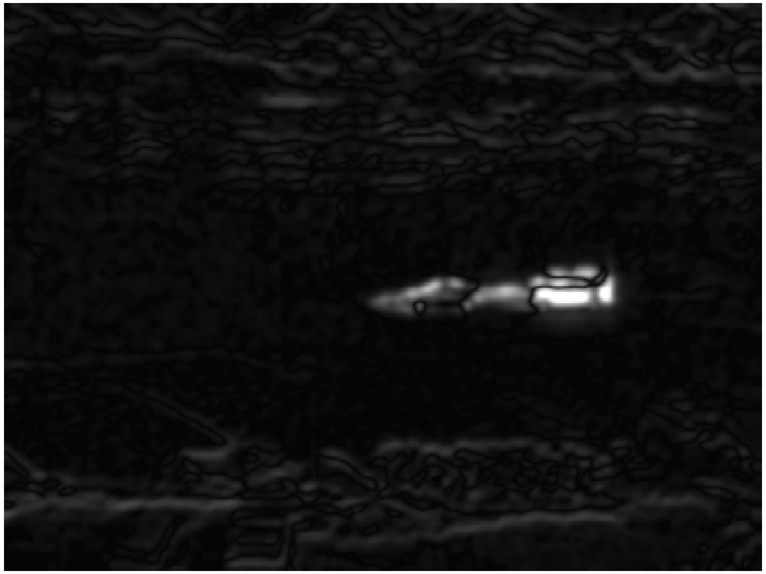


Image Subtraction: Motion Detection

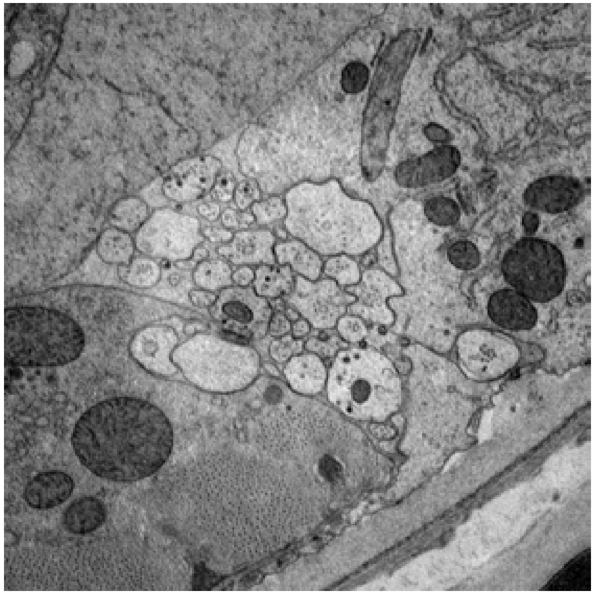


Arithmetic operations: f x g

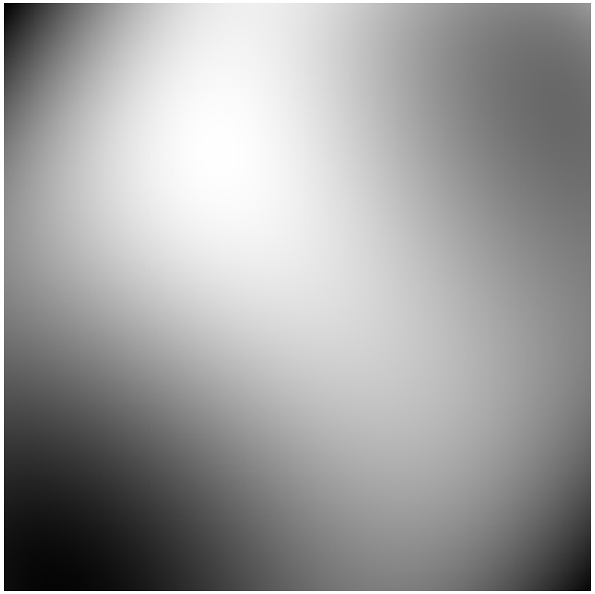
a b c

FIGURE 2.30 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

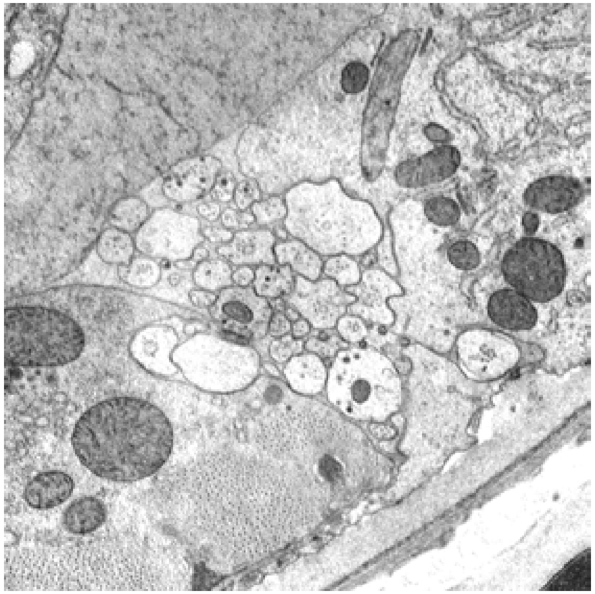
Arithmetic operations: f / g



Captured image



Illumination



Corrected image

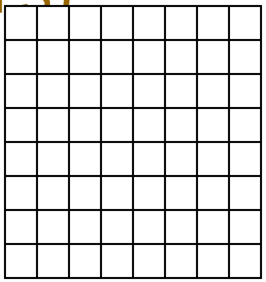
Operations on Cartesian Image Grids

- Grid resolution
- Neighborhoods
- Adjacency and connectivity
- Paths
- Connected components
- Flood fill

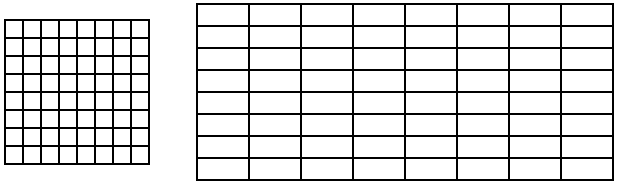
Image Coordinates and Resolution

- A single point on an image grid is a "pixel"
  - Sometimes this is just the location, sometimes also the value
- References to pixels
  - Single index (implied ordering) "i" or "f(i)"
  - Multiple index (gives position on logical grid) "i,j" or "f(i,j)"
- Physical coordinates $(x_{ij}, y_{ij}) = (r_x i + o_x, r_y j + o_y)$
  - Logical coordinates place the pixel in physical space
  - r - resolution (e.g. mm's)
  - o - origin

Resolution vs *size* vs *dimension*

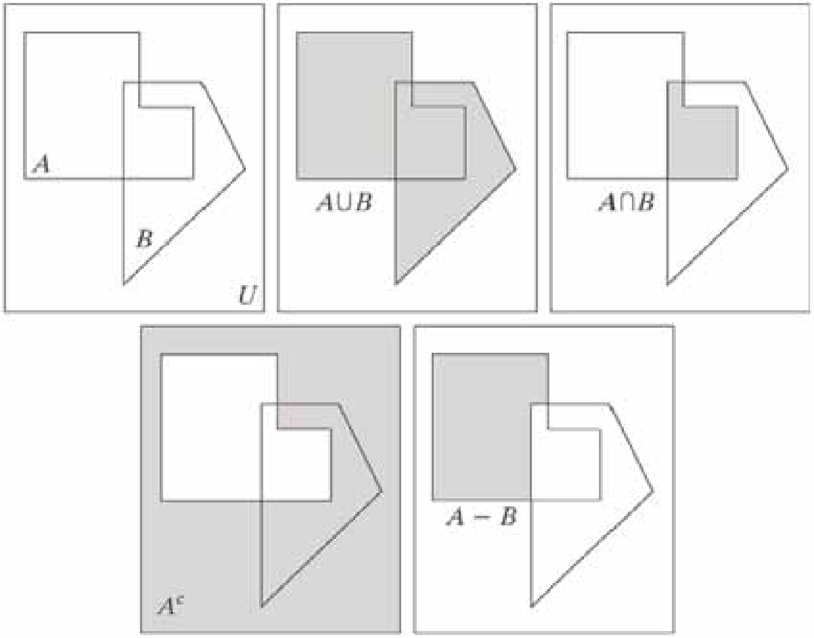


Different physical realizations of the same logical grid



Index Sets

- An index set is a collection of pixel locations
  - Used to specify subsets of an image
  - All boolean set operations apply
- Convention
    - Represent the set as an image with 0 indicating non membership and >0 indicating membership
    - Logical operations become arithmetic operations



Neighborhood

- *Neighborhood (N):* a set of *relative* indices that satisfy the symmetry condition
  - Symmetry:

$$(i,j) \in \mathcal{N} \;\Leftrightarrow\; (-i,-j) \in \mathcal{N}$$

- Applying neighborhoods:

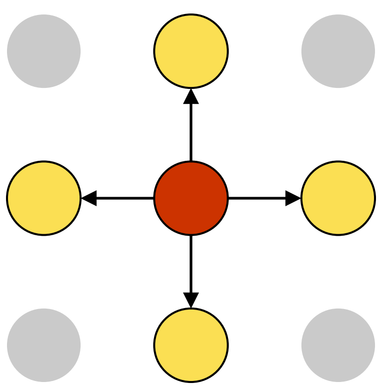
$$\mathcal{N}(i,j) = \{(k,l) | (k-i, l-j) \in \mathcal{N}\}$$

  - I.e. you translate neighborhoods to different locations
- Notice: $(p,q) \in \mathcal{N}(i,j) \;\Leftrightarrow\; (i,j) \in \mathcal{N}(p,q)$

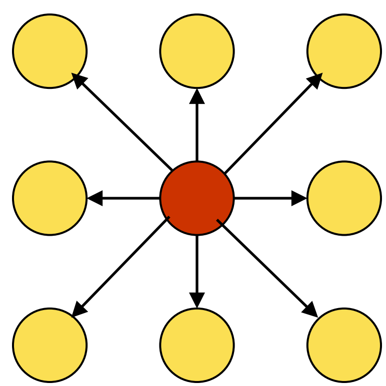
Adjacency

- Impose topological structure on the grid
- Local relationships between pixels
- Help to establish distances, paths, connectedness, etc.
- Typically adjacency is local and symmetric
- For 2D images we consider:

4 connected



8 connected



Denote

$$I \sim J$$

Paths

- *Path:* Ordered set of indices such that consecutive indices are adjacent

$$\mathcal{P} = (I_1, I_2, \ldots I_n) \text{ such that } \mathrm{I_i} \sim \mathrm{I_{i+1}}$$
$$\forall\, \mathrm{i} = 1, \ldots \mathrm{n} - 1$$

- *Noncylic path* – unique indices
- *Closed path* – noncyclic and first and last adjacent

Distances in Images

- Grid distance vs physical distance
  - Physical distance between pixels I and J

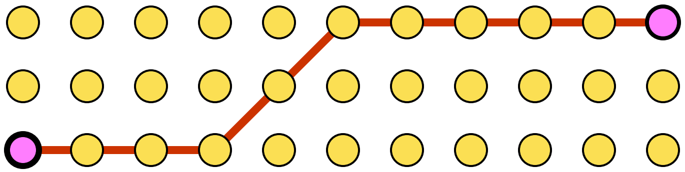
$$D(I,J) = \sqrt{(x_I - x_J)^2 + (y_I - y_J)^2}$$

Distances in Images

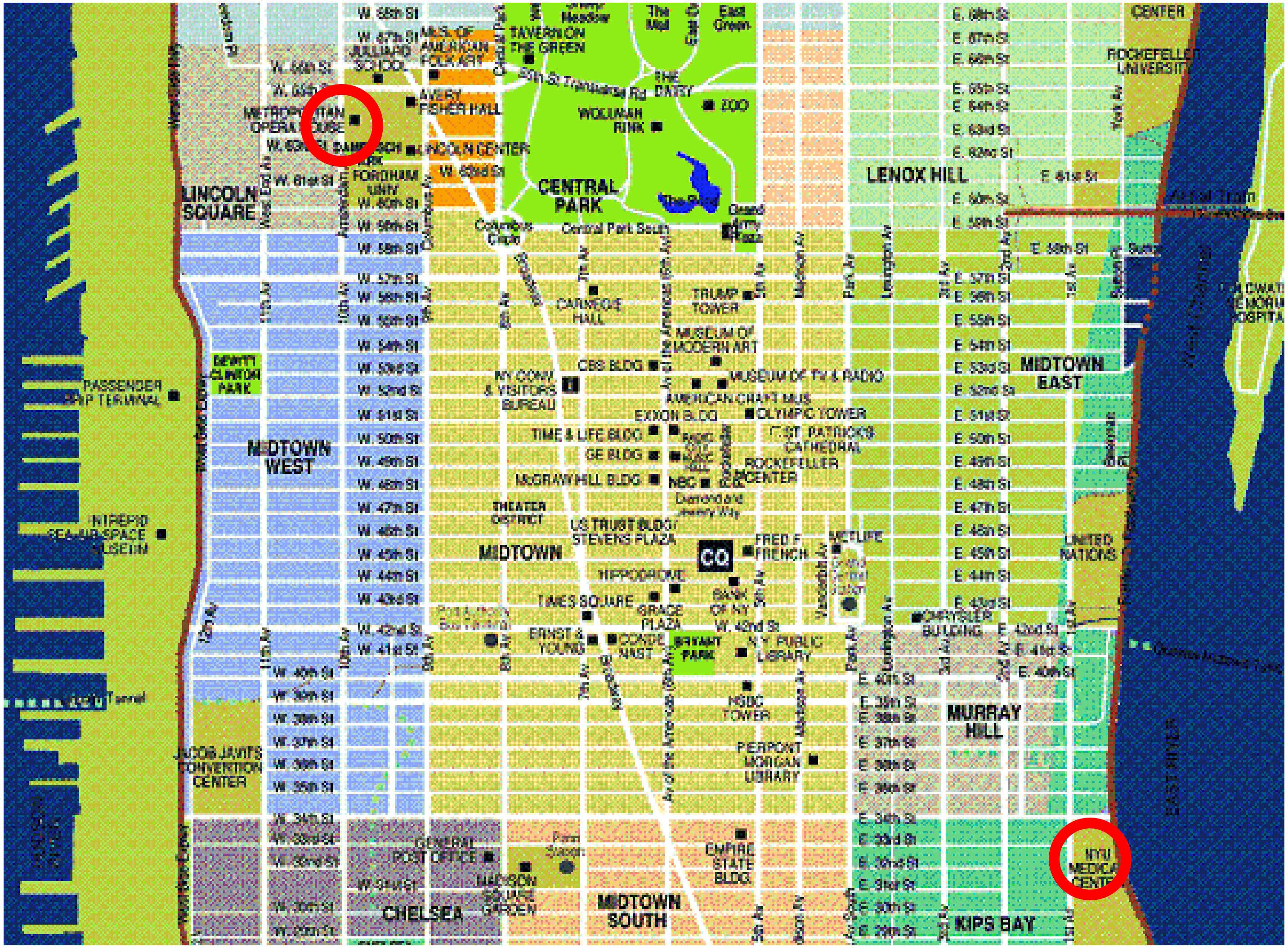
- Grid distance vs physical distance
  - Physical distance between pixels I and J

$$D(I,J) = \sqrt{(x_I - x_J)^2 + (y_I - y_J)^2}$$

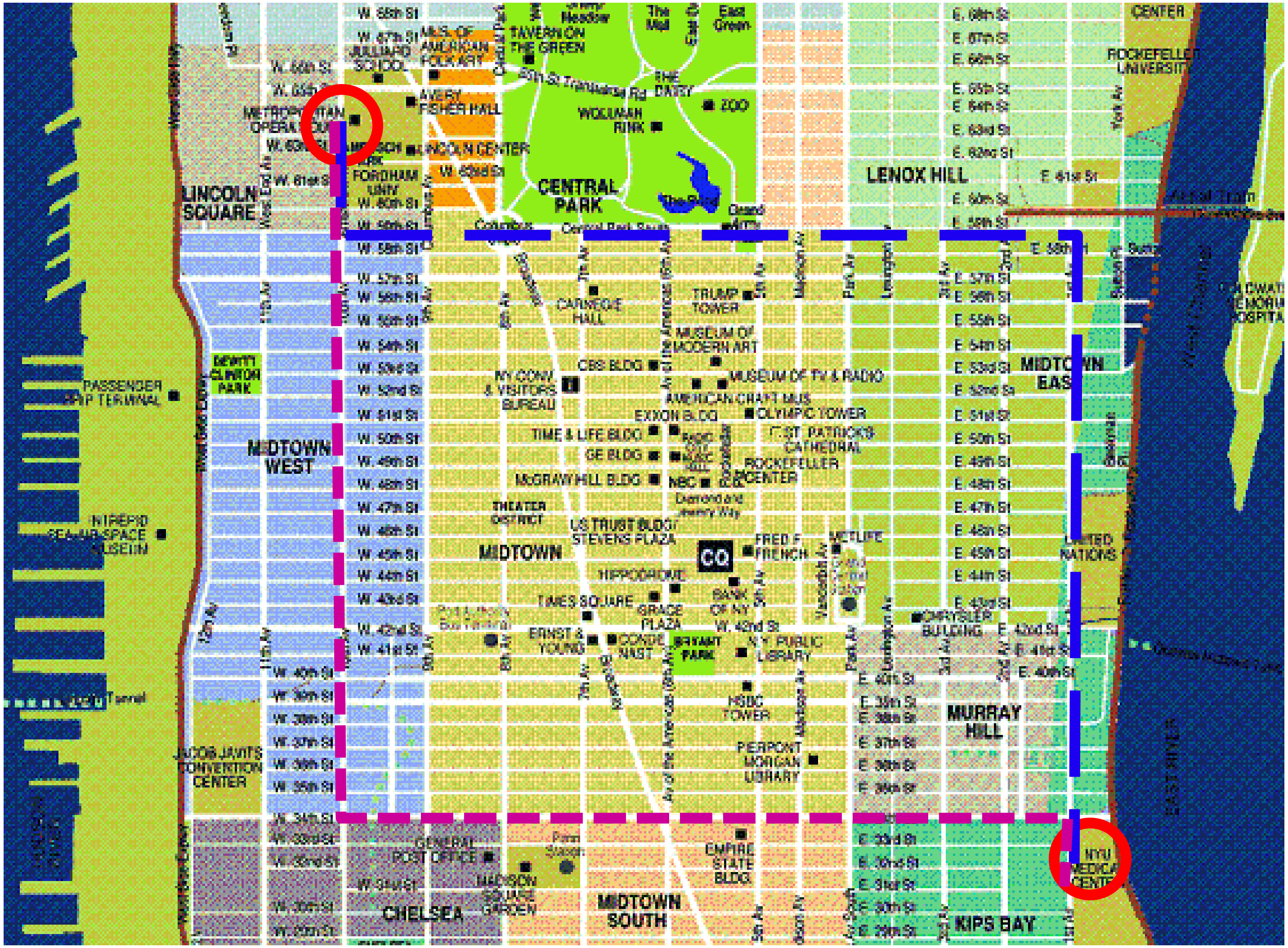
  - Grid distance: options
    - Grid Euclidean $D\left((i,k),(j,l)\right) = \sqrt{(i-j)^2 + (k-l)^2}$
    - Manhattan (city block) $D\left((i,k),(j,l)\right) = |i-j| + |k-l|$
    - Shortest path
      - Assign cost to each transition between adjacent pixels
      - Find path with shortest cost



Manhattan Distance / City Block Distance

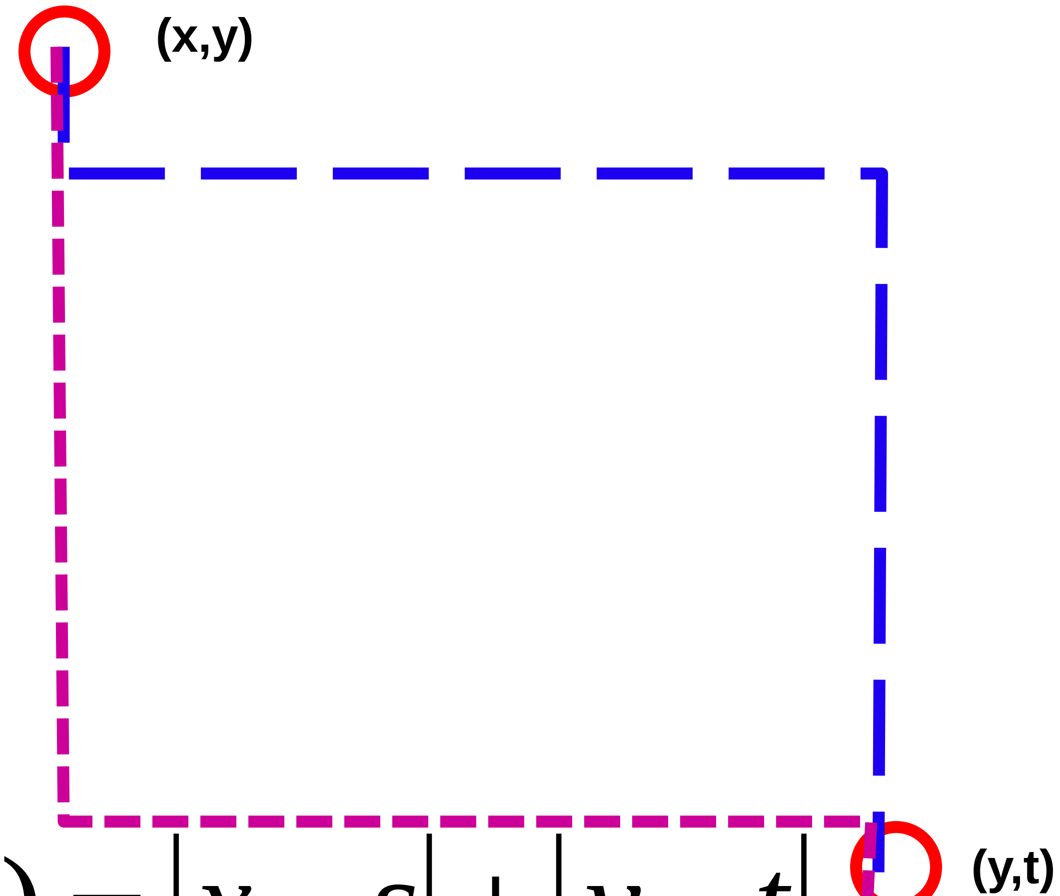


Manhattan Distance / City Block Distance



Manhattan Distance / City Block Distance

Which path is shorter?



$$D_4(p,q) = |x - s| + |y - t|$$

Pixels with D_4 distance from center

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

The pixels with $D_4 = 1$ are the 4-neighbors of (x, y).

Pixels with D_8 distance from center

The D_8 *distance* (called the *chessboard distance*) between p and q is defined as

$$D_8(p, q) = \max(|x - s|, |y - t|) \tag{2.5-3}$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

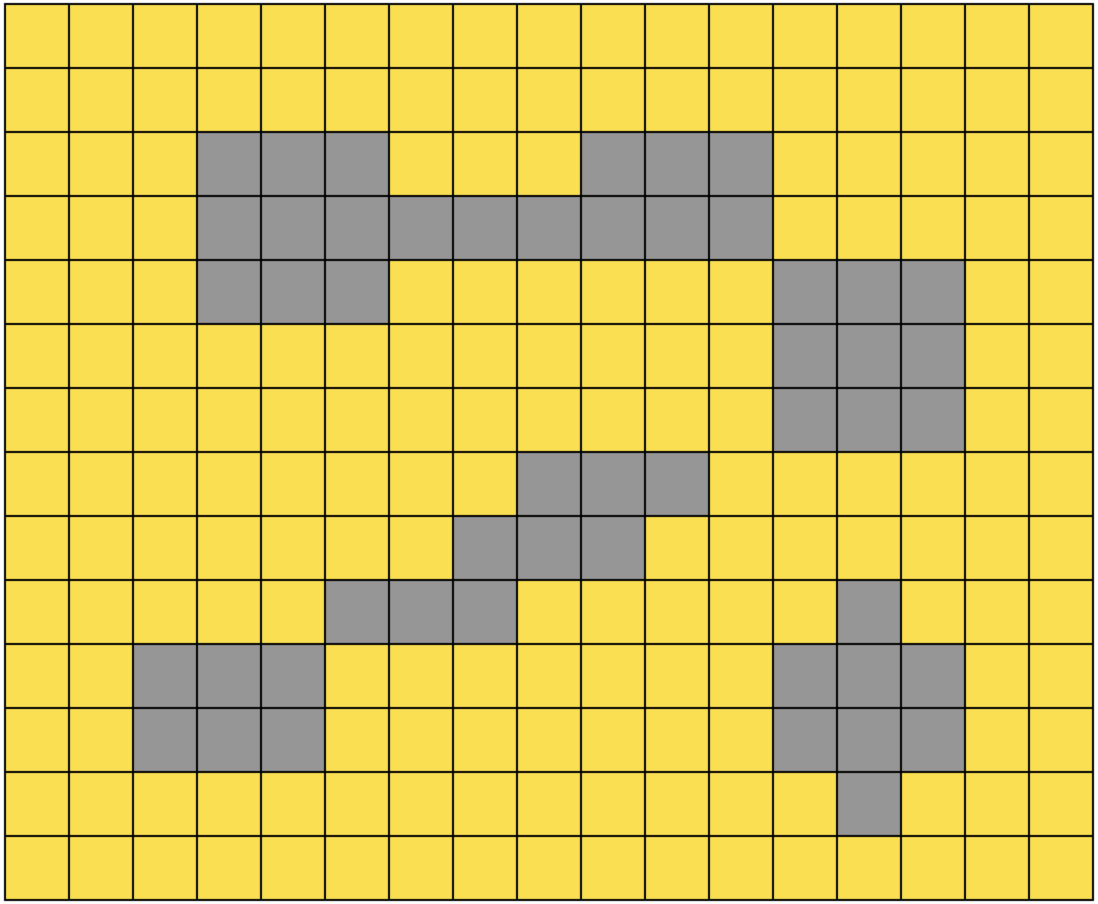
The pixels with $D_8 = 1$ are the 8-neighbors of (x, y).

Connected Component

- Consider image with a binary property
  - I.e. test on each index B(I) returns either true or false
- C*orrect path* : path for which every pixel satisfies B(I)
- *Connected component* (C) : set of pixels such that for every pair of pixels in C there exists a correct path between them

Connected Component

- How many distinct CC's are there?



A Simple Algorithm: Flood Fill

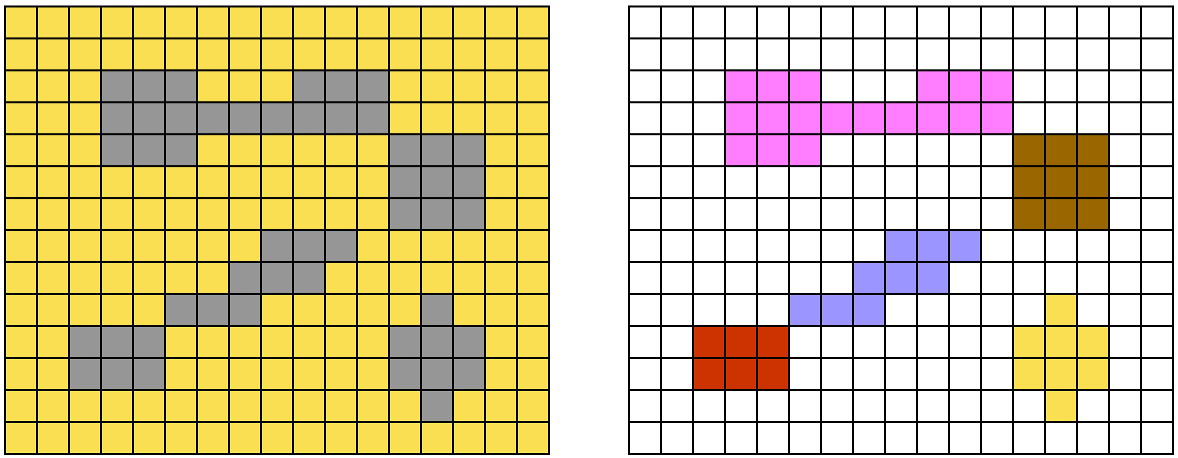
- Highlight regions in an image
- “Test(i, j)” - is value at pixel (i,j) between a and b
- Inputs: seed, image, test function
- Data structures: input array, output array, list of grid points to be processed

A Simple Algorithm: Flood Fill

- Empty list, clear output buffer (=0)
- Start at seed (i,j) and if Test(i,j), put (i,j) on list and mark out[i,j]=1
- Repeat until list of points is empty:
  - Remove point (i,j) from list
  - (Loop) for all 4 neighbors (i’,j’) of (i,j)
    - If (Test(i’,j’) and out[I’,j’]==0) put (i’,j’) on list and mark out[i’,j’]=1
- Properties
  - Guaranteed to stop
  - Worst case run time

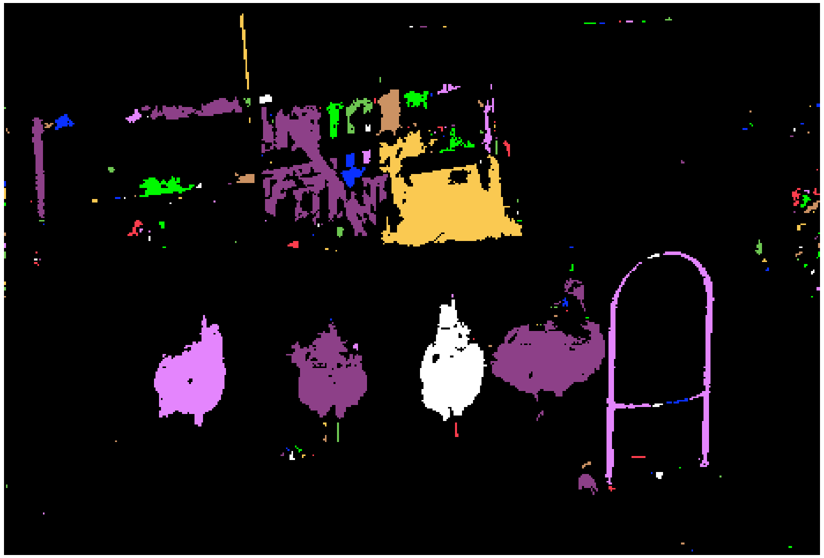
Connected Component Analysis

- Input: image and test
- Output: an integer image (label map) that has either "0" (failed test) or a positive integer associated with each distinct connected component



CC – Purpose

- When objects are distinguishable by a simple test (e.g. intensity threshold)
- Delineate distinct objects for subsequent processing
  - E.g Count the number, sizes, etc.
  - Statistics, find outliers irregular objects



CC Output/Extensions

- Output is typically a “label map”
- How to handle foreground/background, etc
- How to display