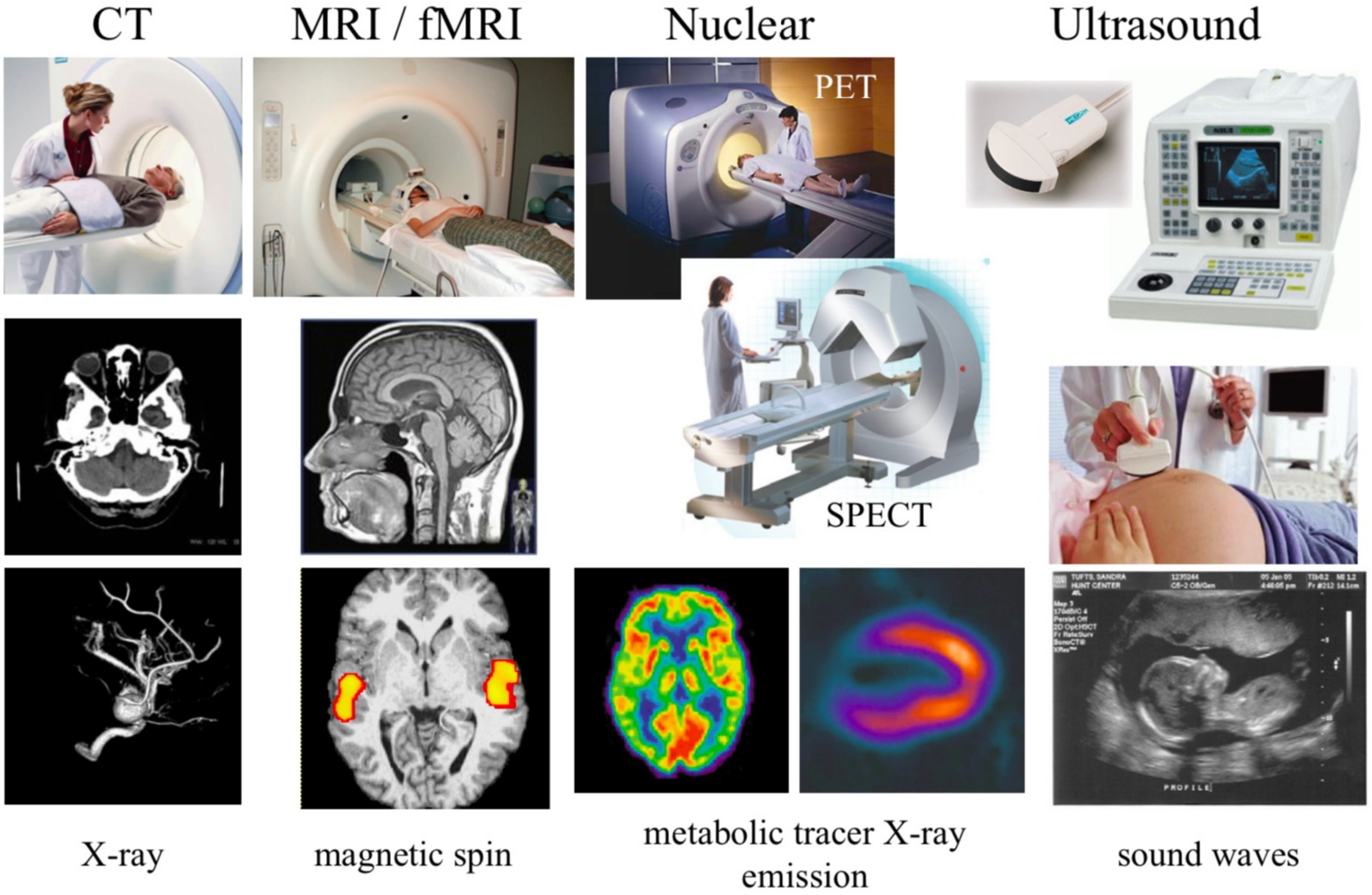
Intelligent Analysis of Biomedical Images

Presenter: Mohammad H. Rohban, Ph.D.

Fall 2023

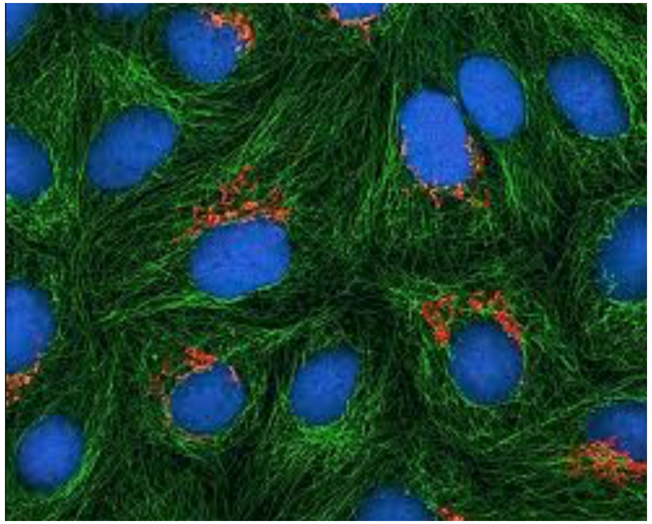
Courtesy: Some slides are adopted from CSE 377 Stony Brook University and CS 473 U. Waterloo

Medical Imaging Modalities

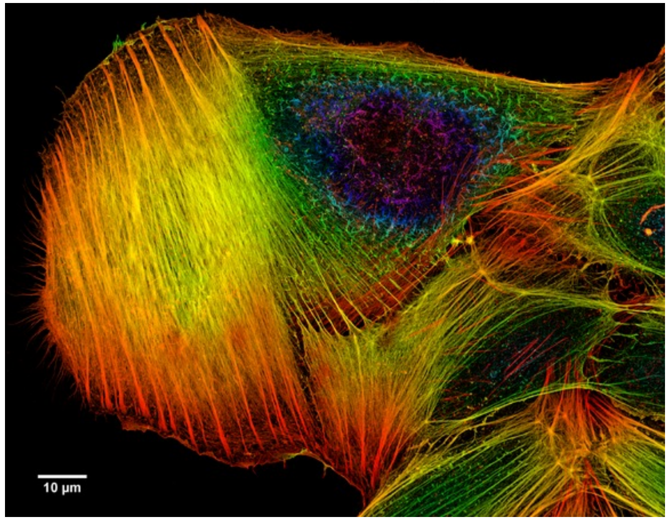


Biomedical Images: Going Deeper

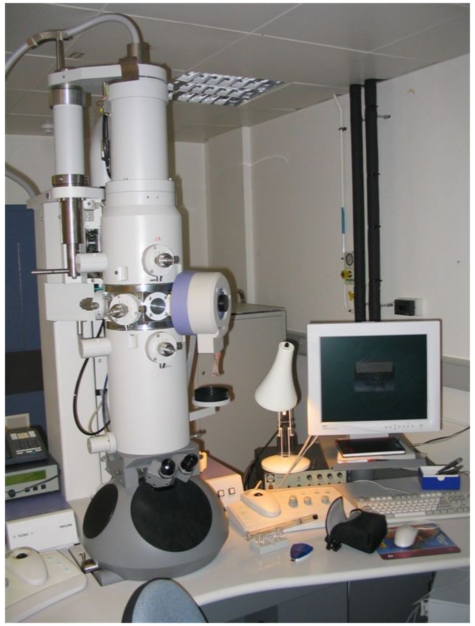
Fluorescence Microscope

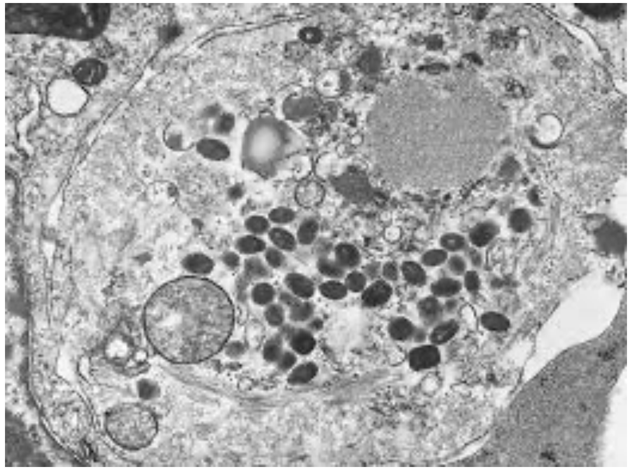


Confocal Microscope

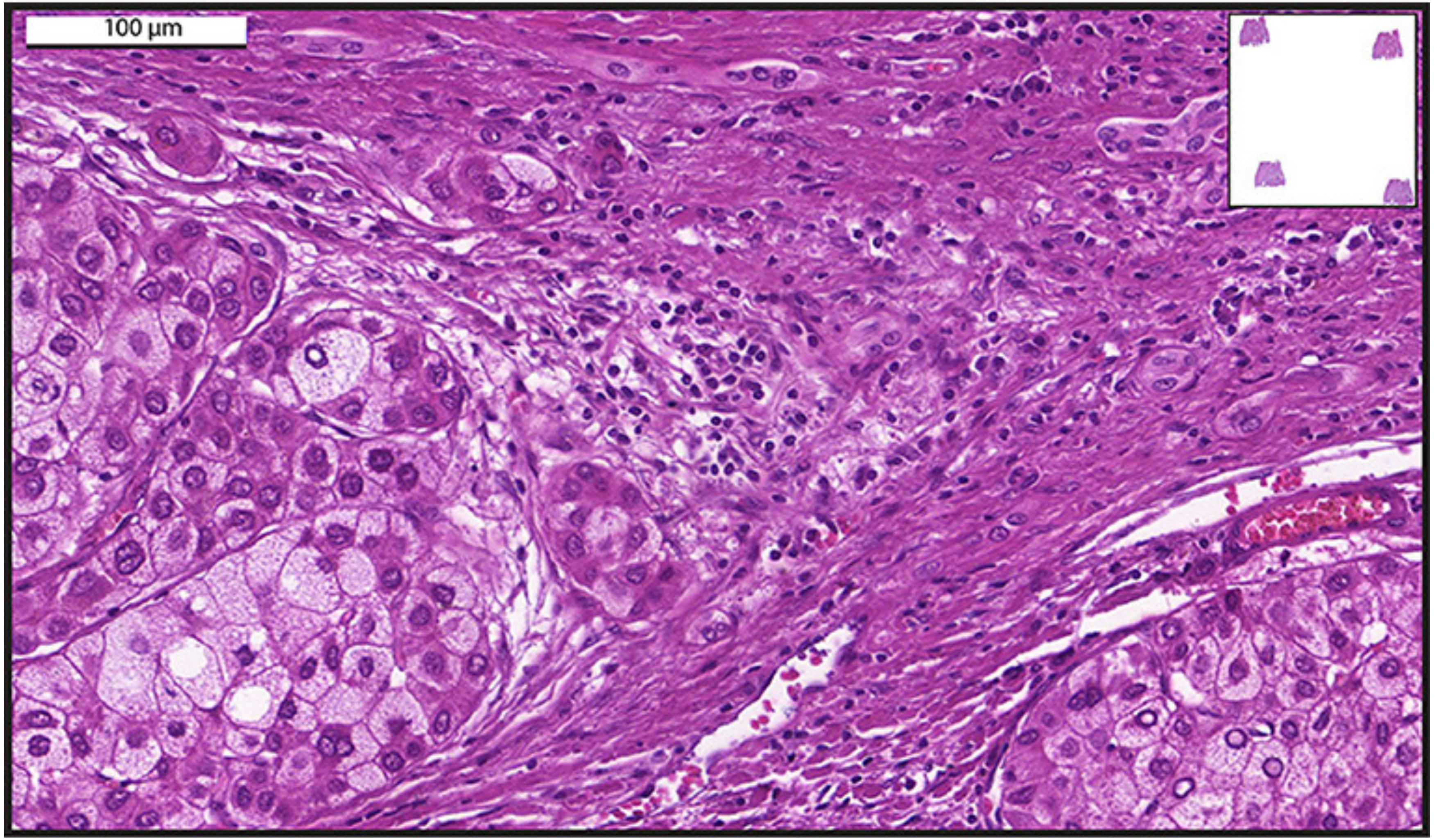


Electron Microscope





Histopathology Images



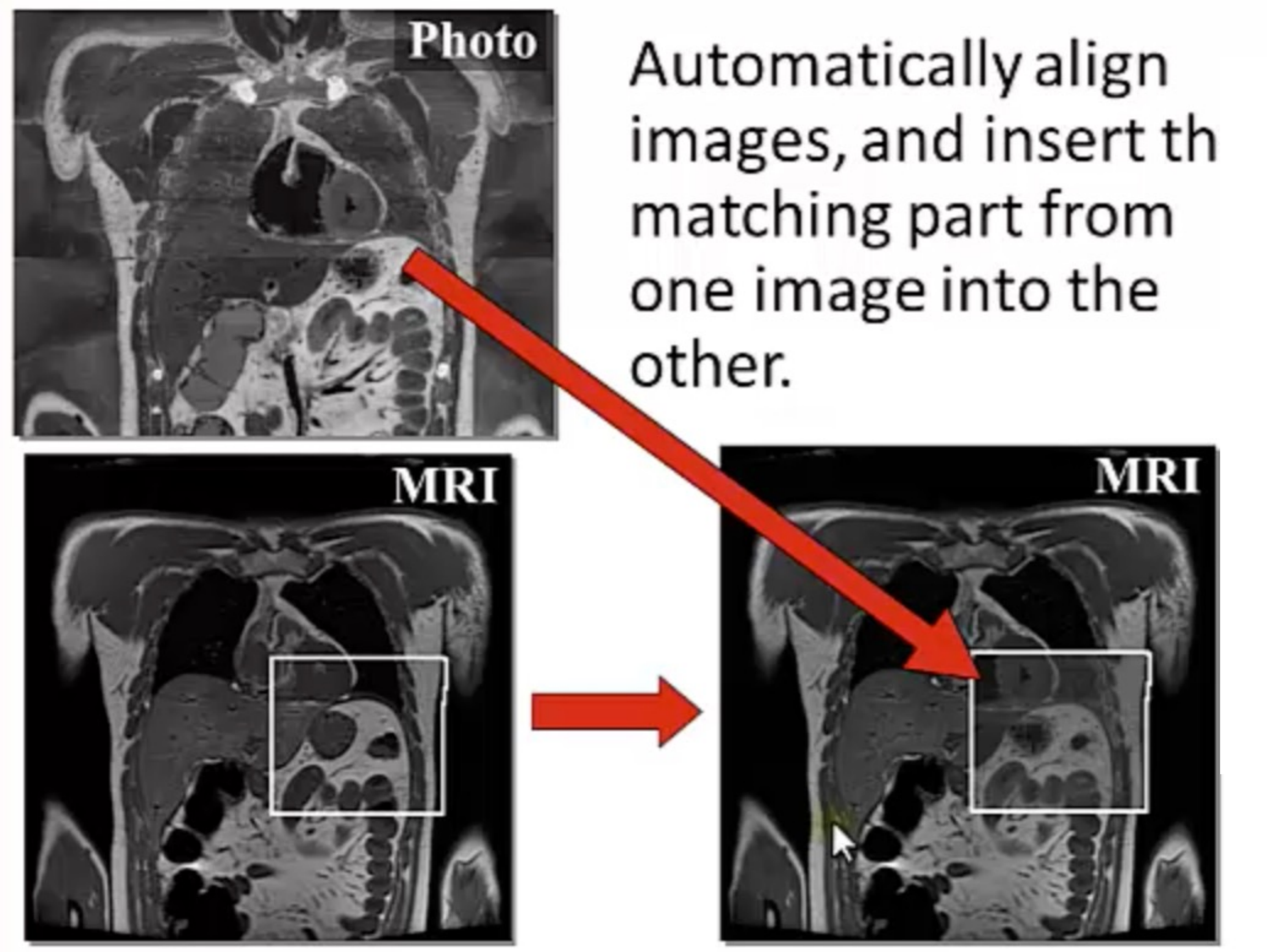
Main Aims

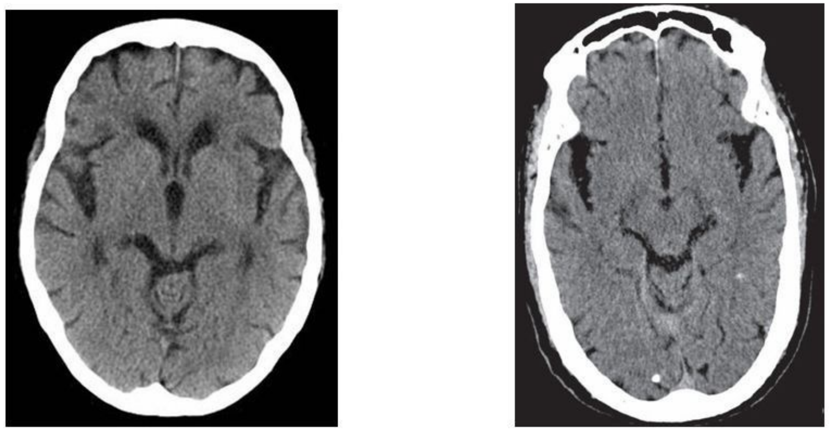
- Prevent/Predict/Diagnose/Treat a disease
- Study biological systems
  - Diagnose a disease
  - Characterize effects of genes/small molecules on cells
- Images are **traditionally** processed by humans.
- We want to make this process **automatic** in this course.
  - More realistically, help the expert to achieve his/her goal.

Biomedical Image Processing

- Image Registration
- Segmentation
- Shape Analysis
- Image Enhancement (denoising and deblurring)
- Detection/Prediction
- Tracking
- Profiling

Image Registration



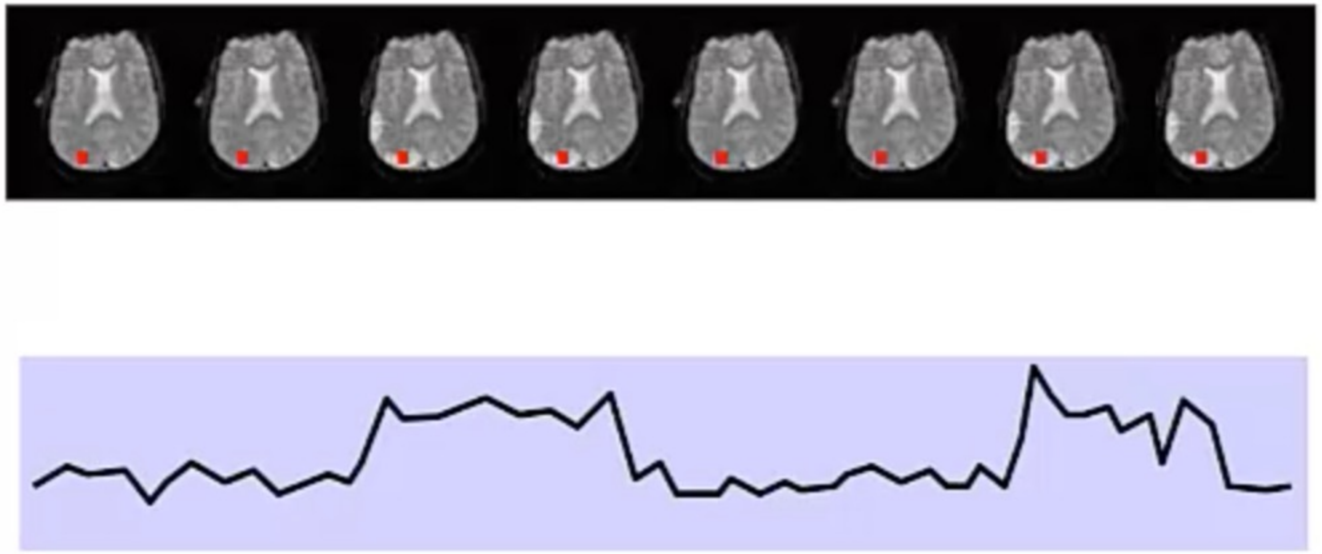


Patient 1 Patient 2

X-Ray CT

Image Registration: fMRI (cont.)

Regions of activity show signal changes.



This only works if the images are properly registered.

Image Registration: Multimodal Fusion

- Some analyses require the information of multiple images.

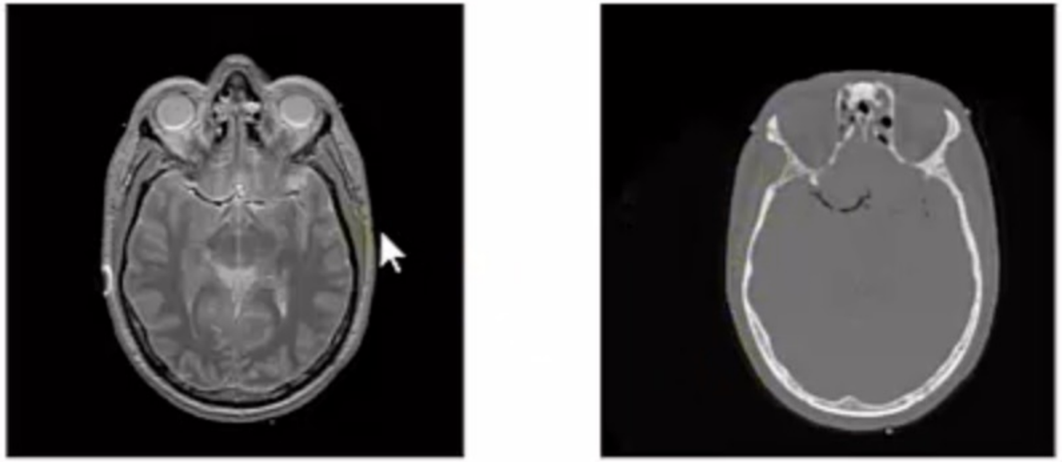
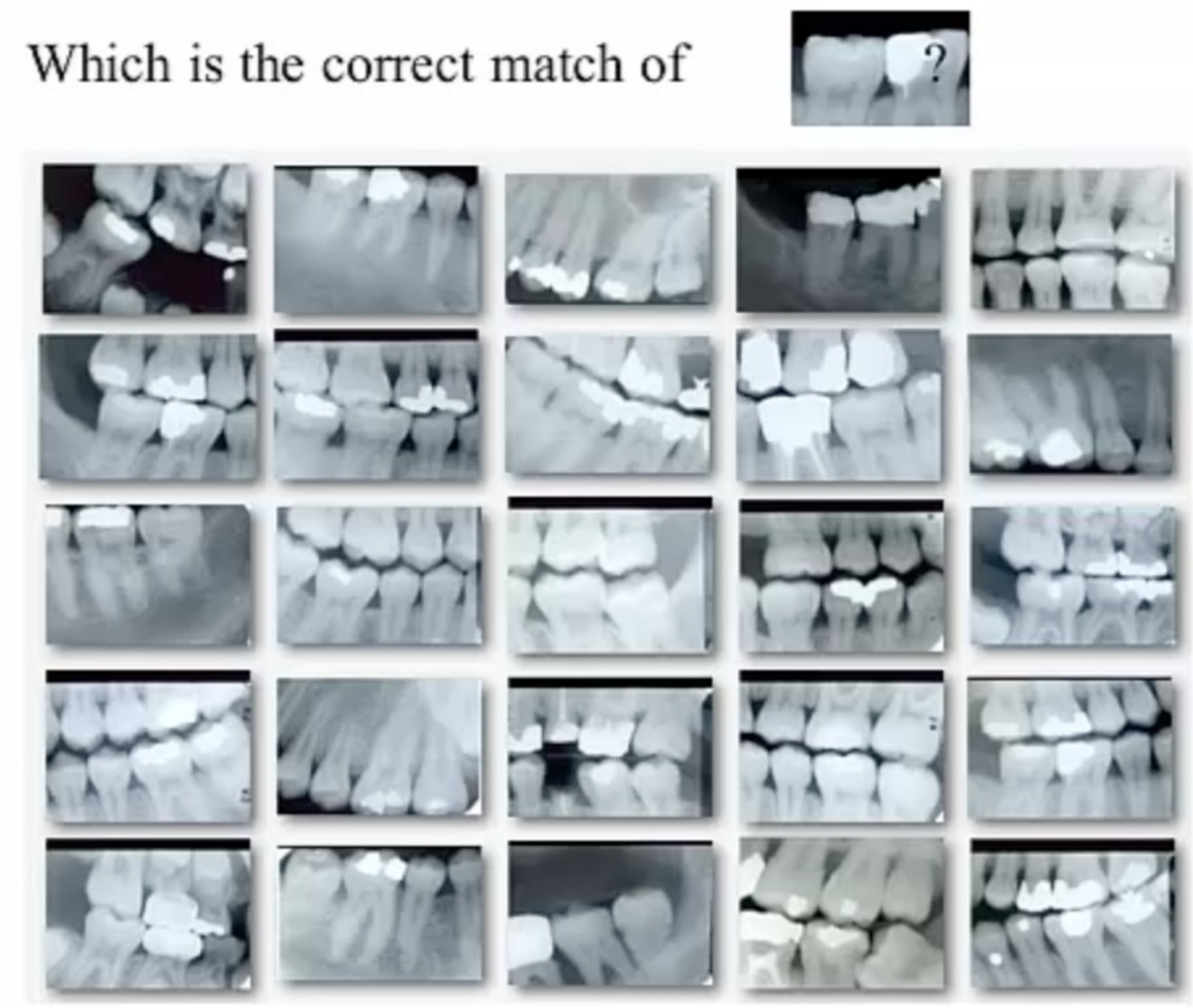
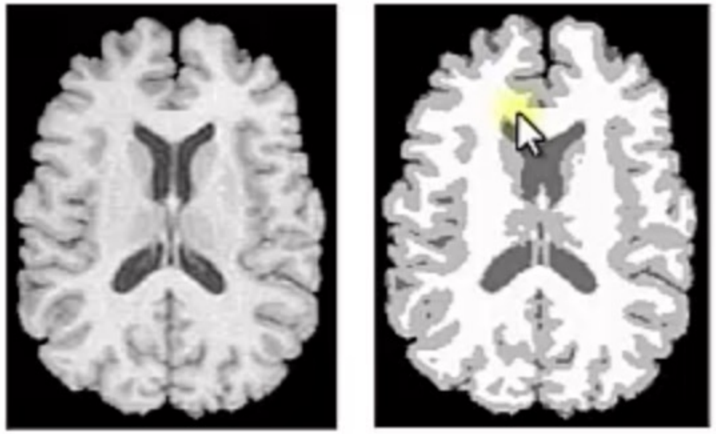


Image Registration: Forensic Identification



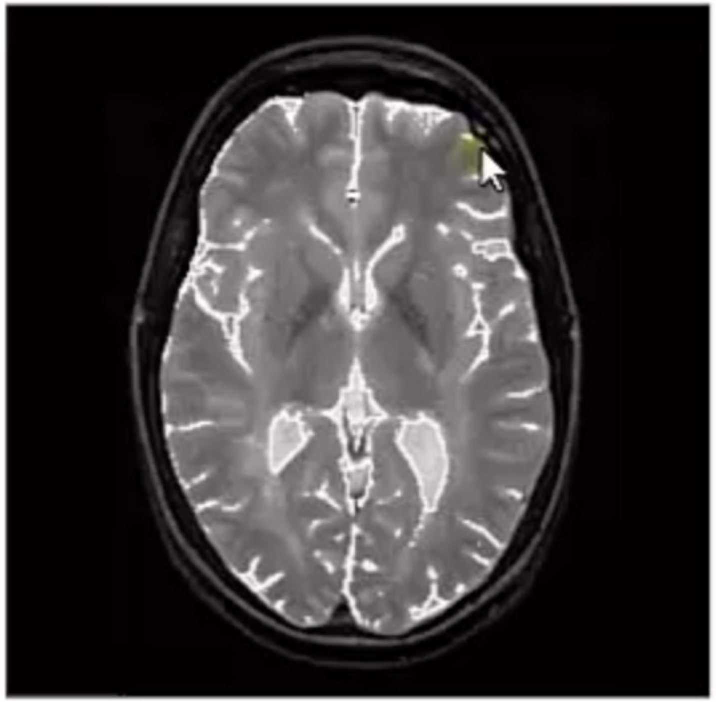
Segmentation

Automatically classify tissue type (brain, liver, bone, muscle, etc.)



Segmentation: Volume Studies

Brain Parenchymal Fraction (fraction of brain volume that is not cerebrospinal fluid) changes as degenerative diseases progress, such as Multiple Sclerosis, Alzheimers, and Parkinson's disease.

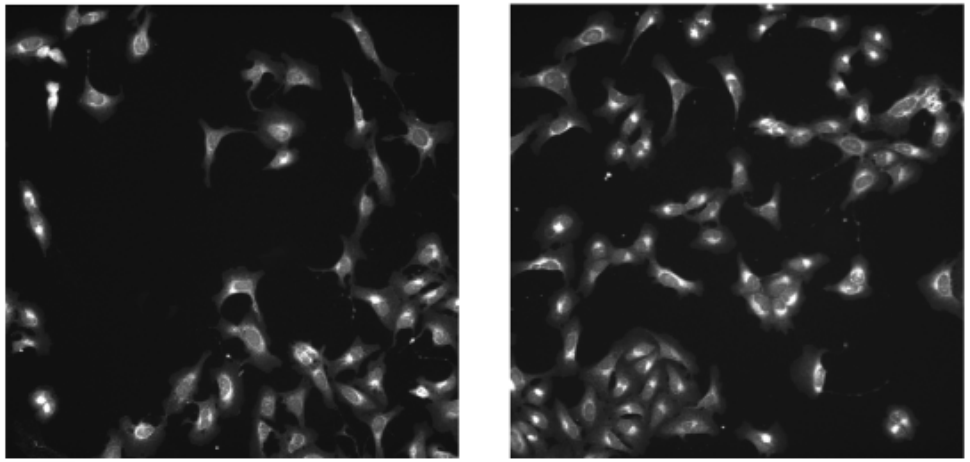


Segmented CSF

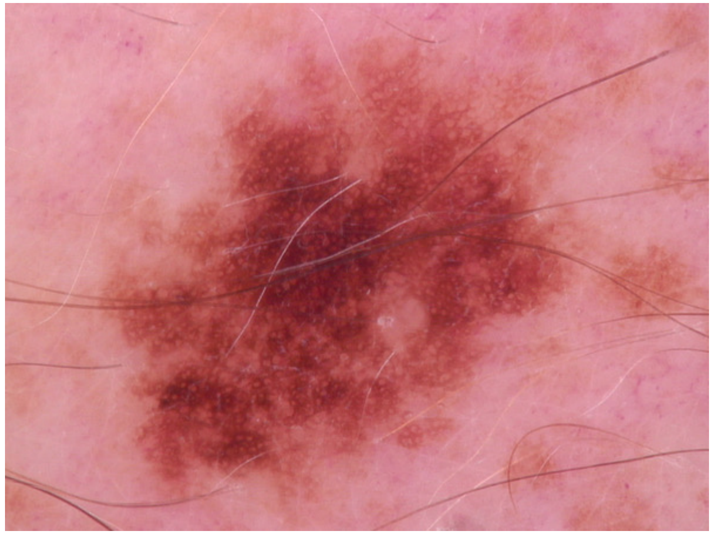
Segmentation: Region of Interest (ROI) Processing

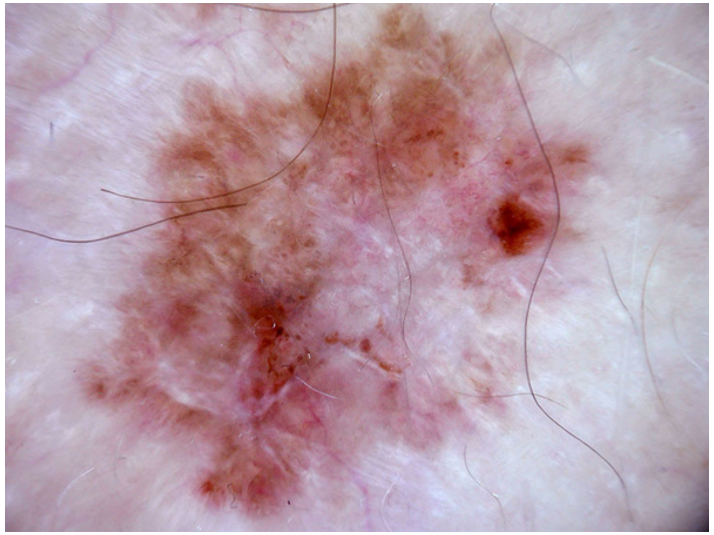
When looking for lung nodules, don't bother looking outside the lungs.

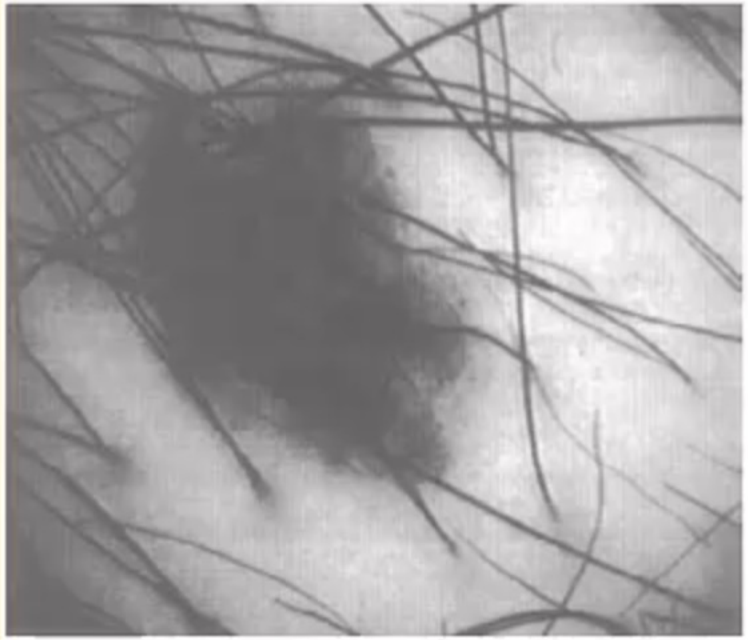
In processing the cell images, the in-between space doesn't matter



Shape Analysis







Mole border delineation
to detect skin cancer.

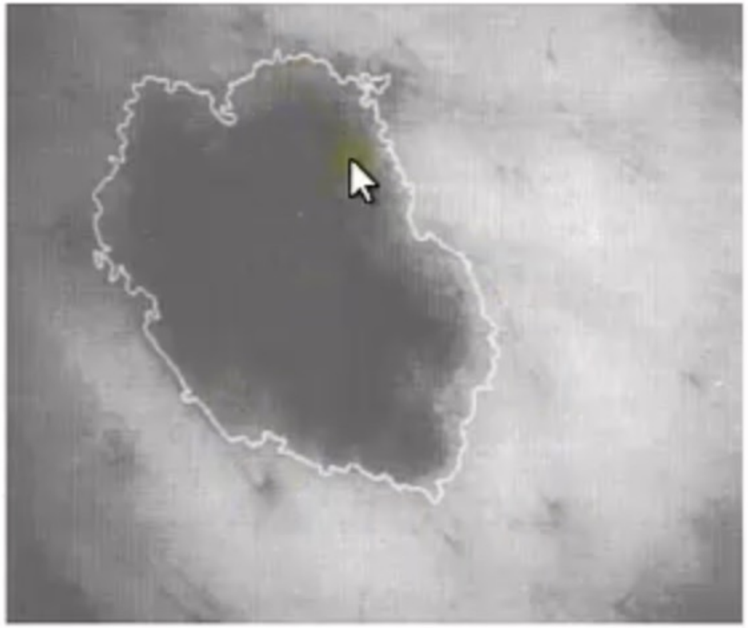
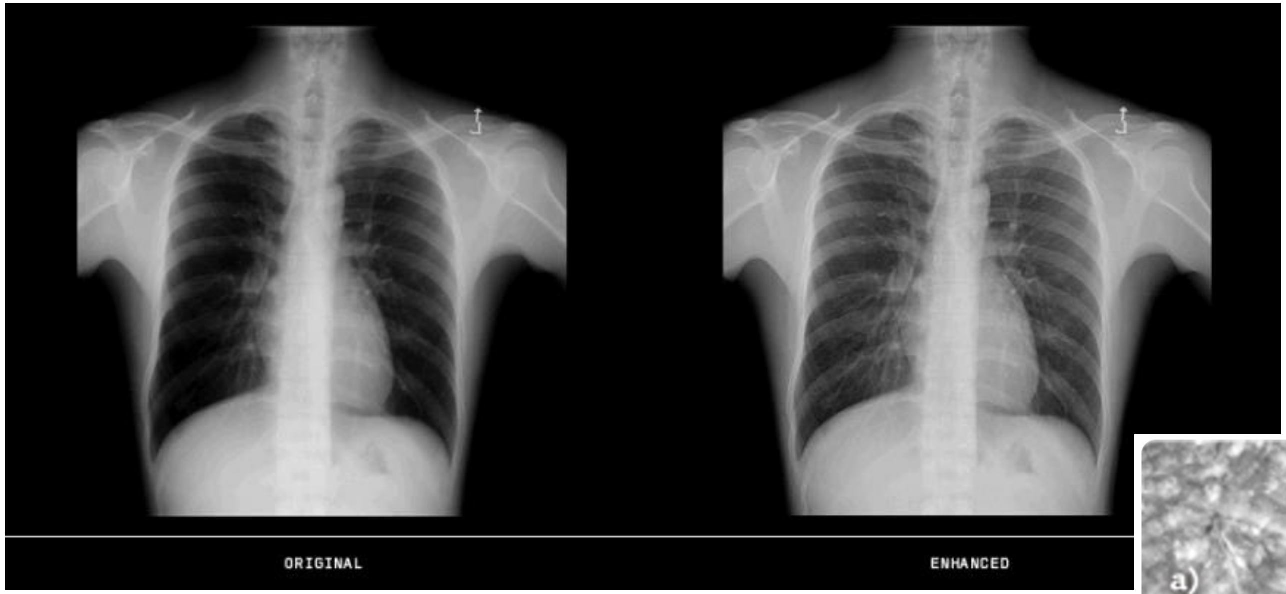
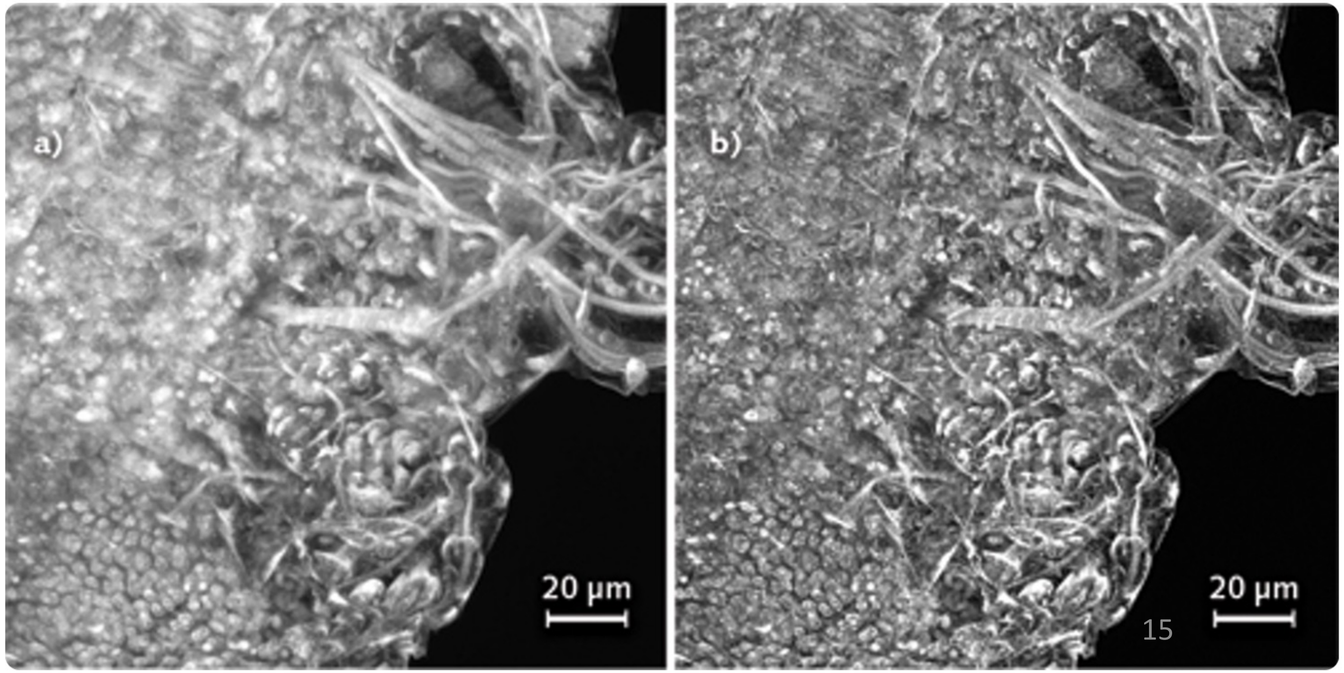


Image Enhancement



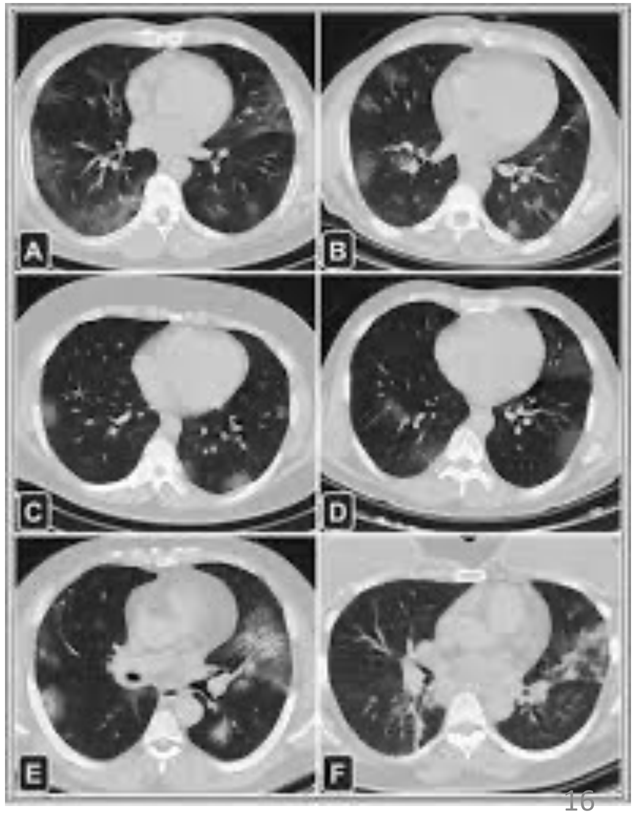


Diagnosis/Prediction

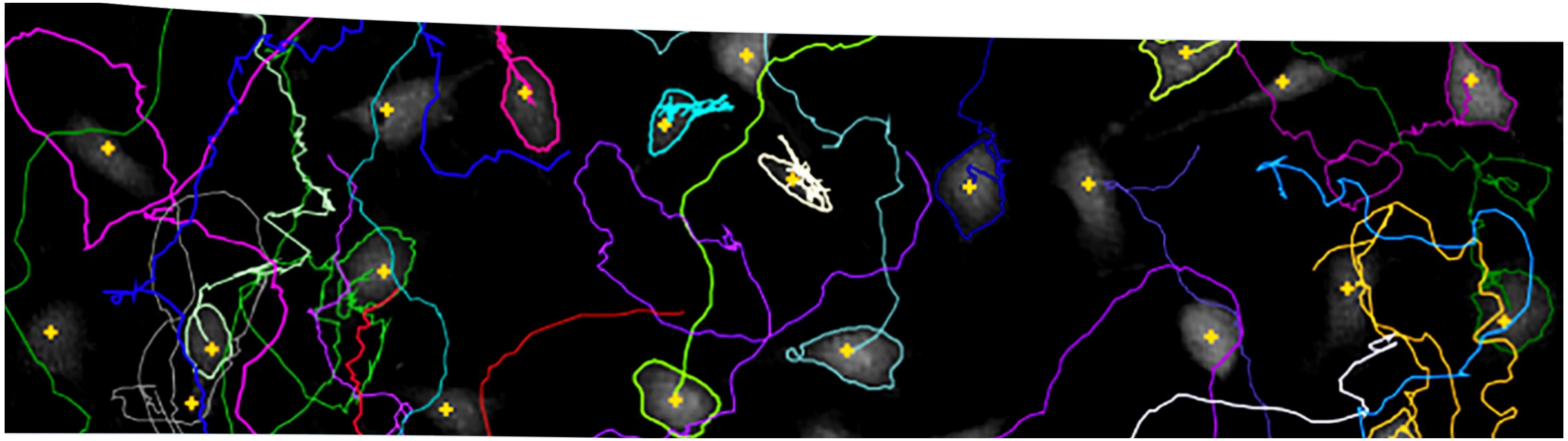
Radiomics Gathered from CT Scans Predict Outcomes from COVID-19 Pneumonia

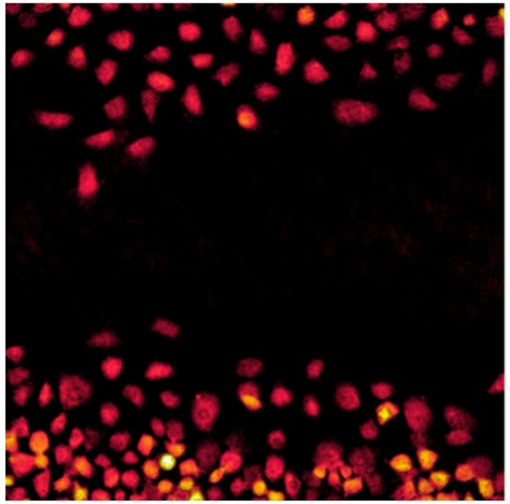
July 28, 2020

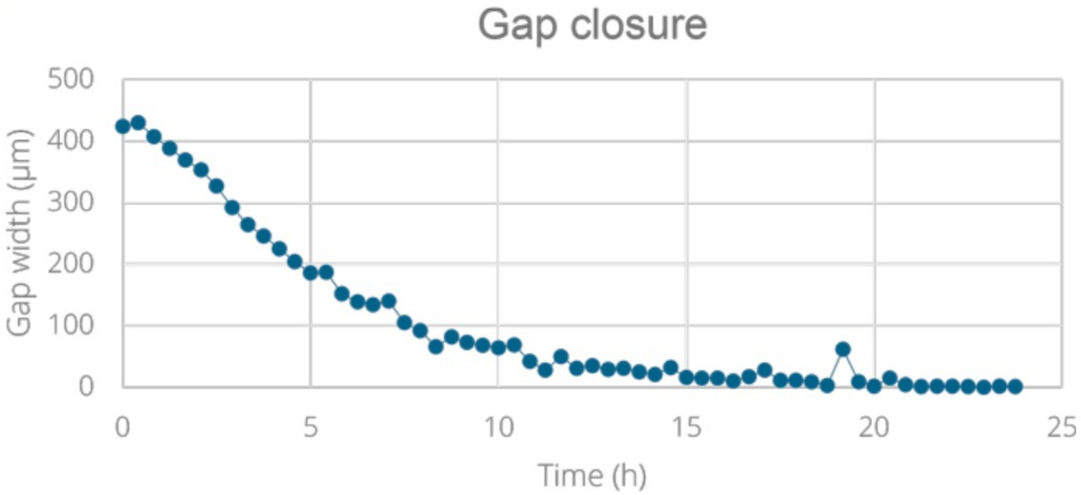
Whitney J. Palmer



Cell Tracking







The HoloMonitor Wound Healing Assay provides kinetic gap closure analysis showing how gap width and cell covered area change over time.

Morphological Profiling

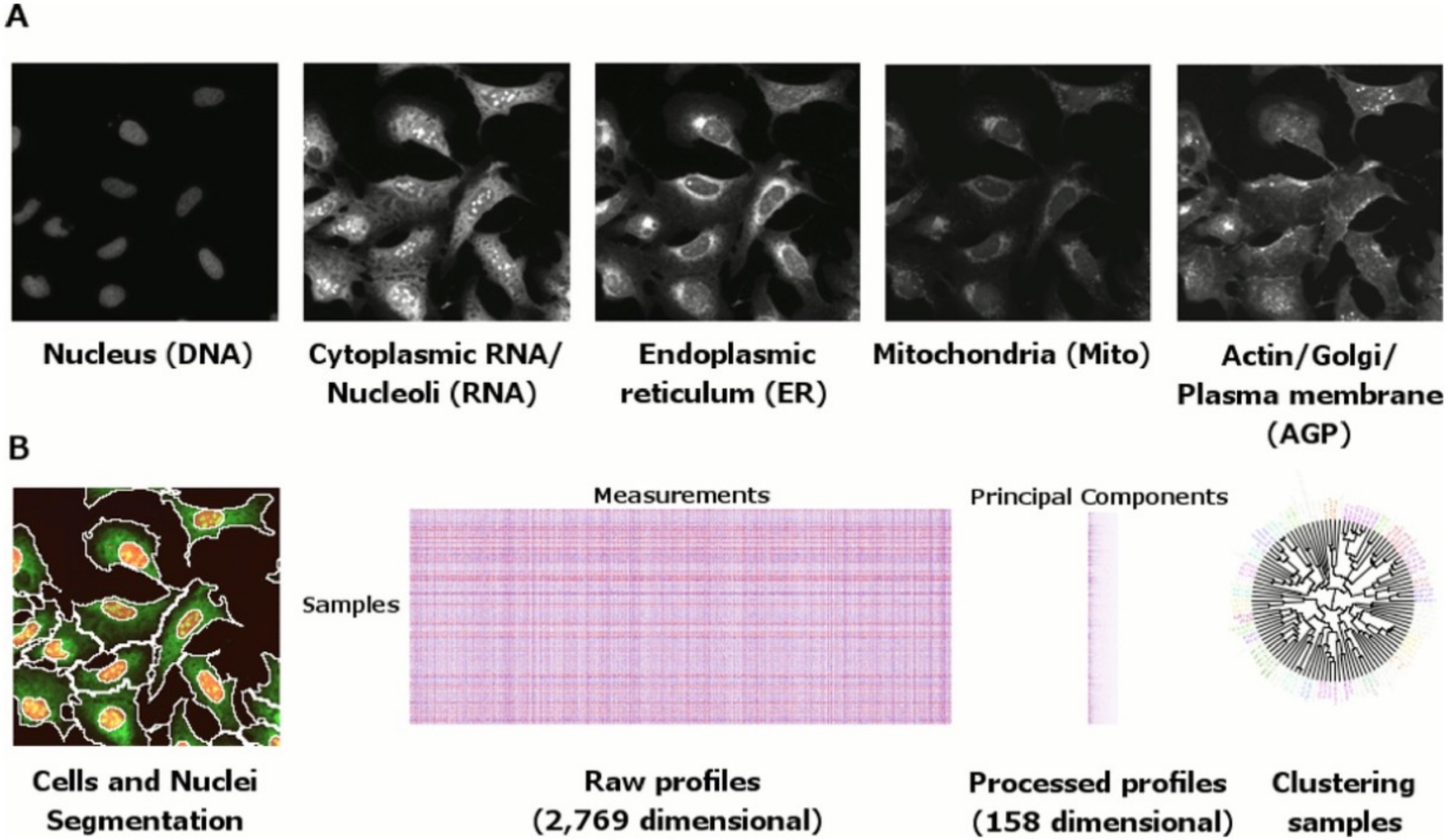
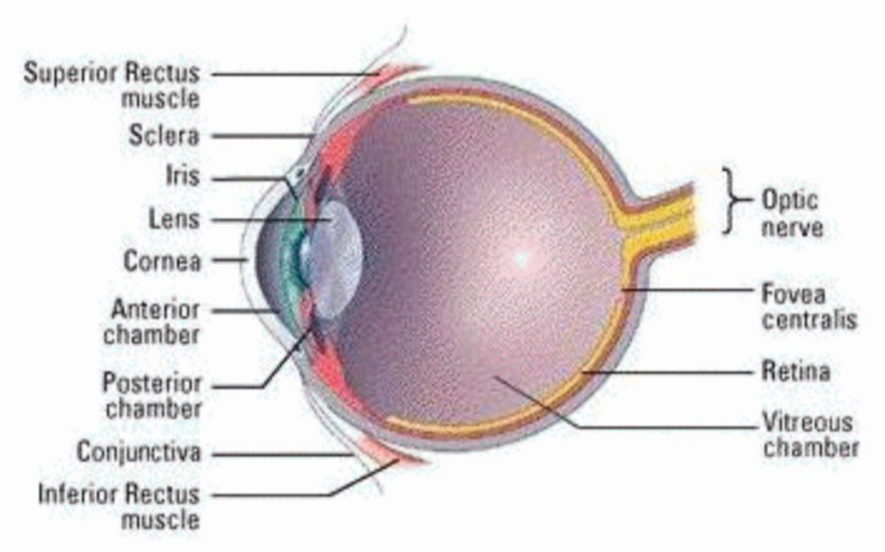


Image Processing Basics: Human eye

Two types of receptors on retina: rods and cones



Rods:

- spread all over the retinal surface (75 - 150 million)
- low resolution, no color vision, but very sensitive to low light

Cones:

- a dense array around the central portion of the retina, the fovea centralis (6 - 7 million)
- high-resolution, color vision, but require brighter light

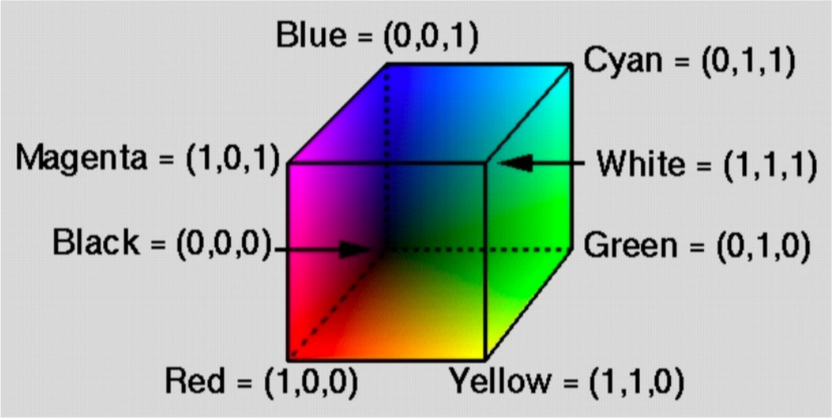
Color Perception

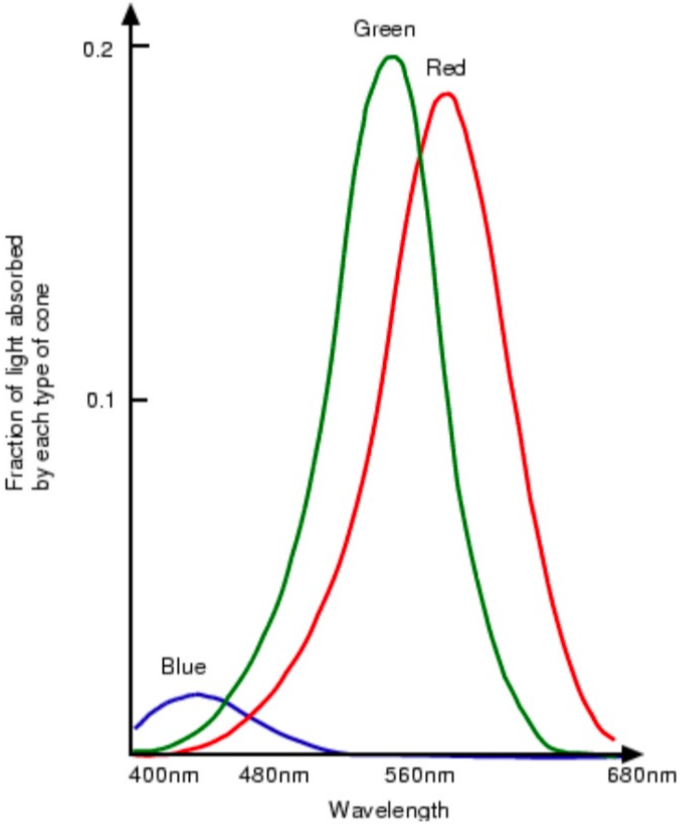
Tristimulus Theory:

- the eye has three types of color receptors: Red, Green, Blue.

Color reproduction:

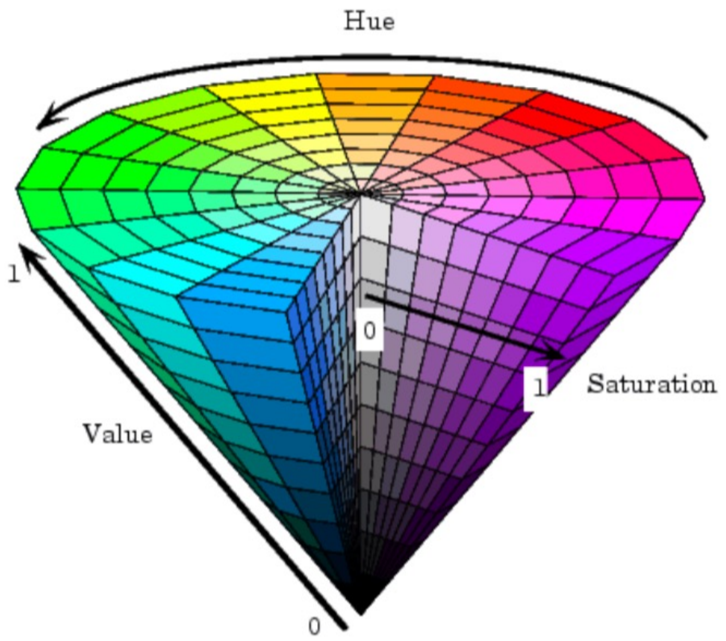
- one can generate (almost) any color on a monitor by mixing three primaries, RGB
- CRT monitor have 3 color guns: RGB



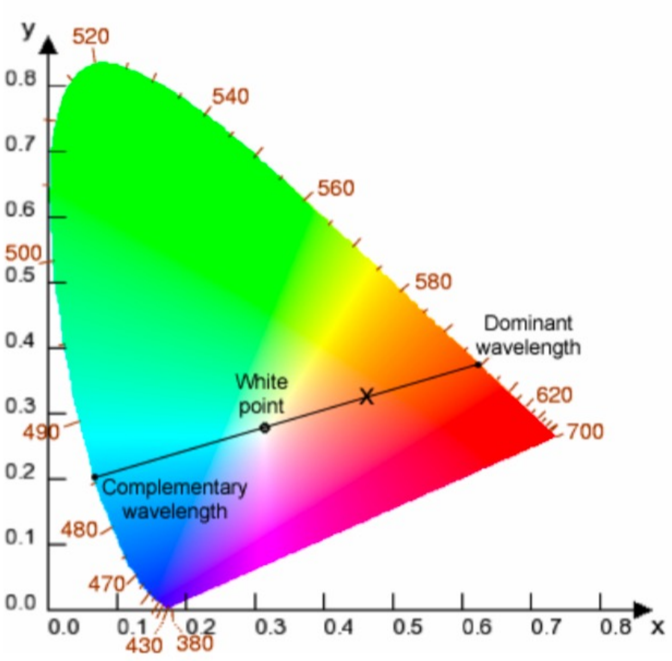


Color Spaces

HSV



CIE



*H*ue: color
*S*aturation: peak from white light
*V*alue: overall integral across all λ

CIE Lαβ: equal distances mean equal perceptive differences

Digital Image

Image:

- 2D matrix of pixels

Image resolution:

- number of pixels along each matrix dimension

300 150 75 37

resolution

Each pixel has a value:

- a single value if greylevel image
- a triple RGB if color image

Dynamic Range

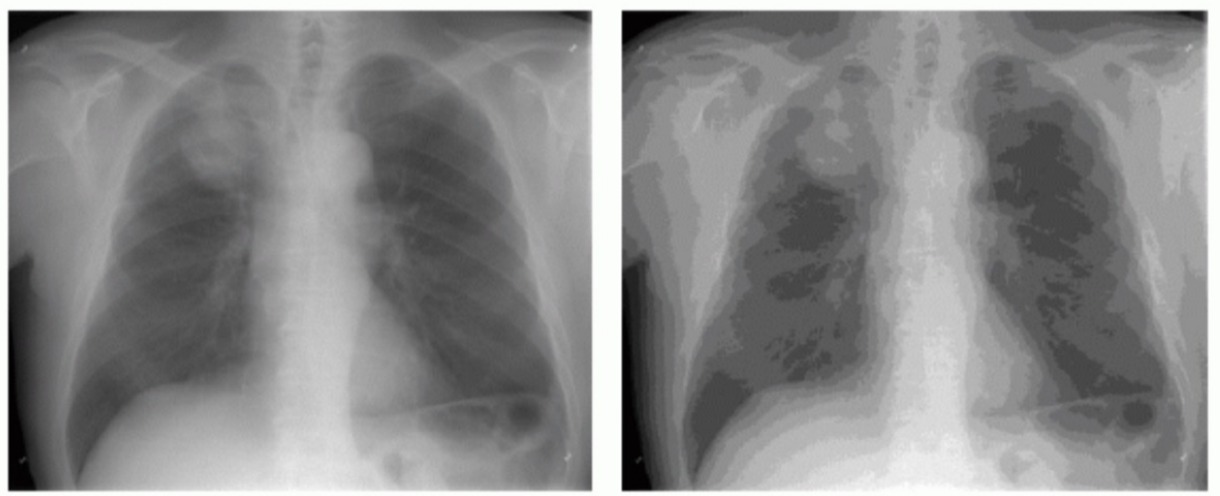
Each pixel is represented by a number of bits

Quantization:

- process of discretizing a continuous value into bits

Minimal number of bits = 6 (64 greylevels or 4 levels for R,G,B)

- most medical digital images have 12 bits (4096 grey levels)



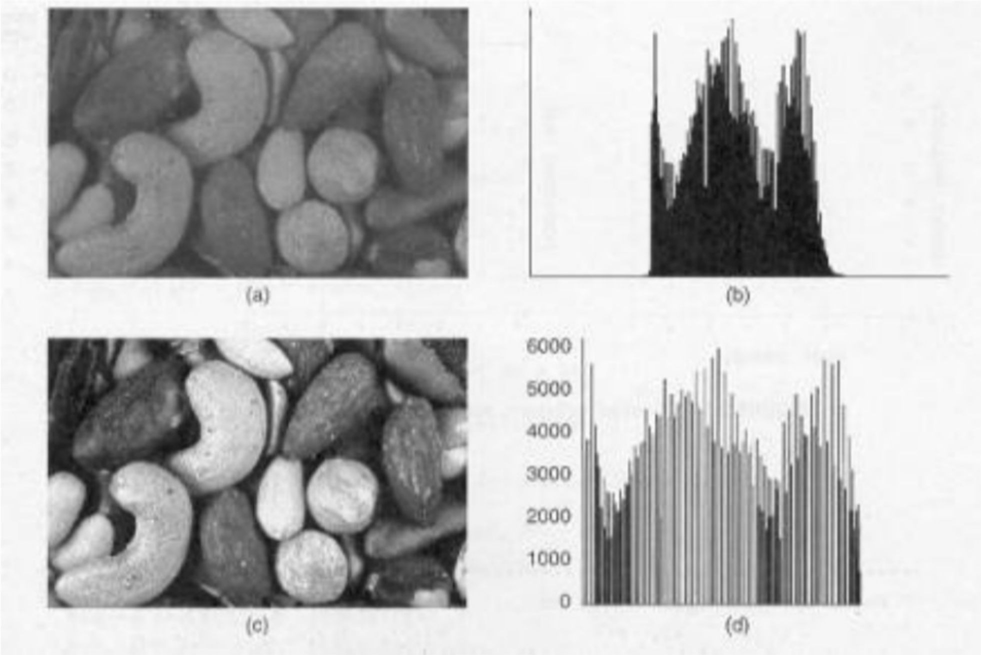
8 bits 4 bits

- not enough bits leads to quantization artifacts and loss of resolution

Histogram

A histogram counts the number of pixels at each greylevel

- $h(v)$ = number of pixels having grey value v / total number of pixels



partial bandwidth

full bandwidth

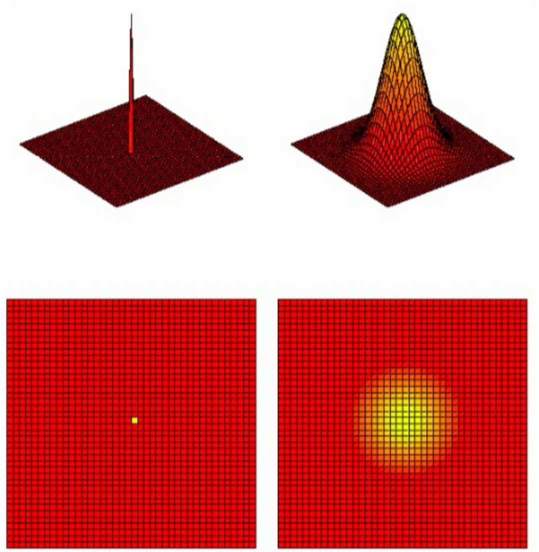
Good contrast requires a histogram with full bandwidth

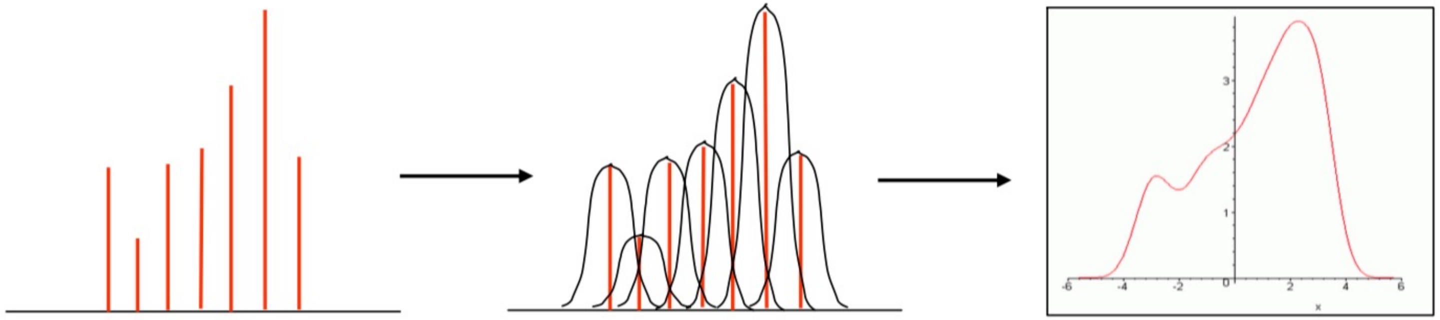
Point Spread Function

Each pixel is not a sharp spike, but represented by a point spread function (PSF)

The PSFs overlap and form a continuous function (for the eye)

Smaller PSFs give sharper images





Contrast

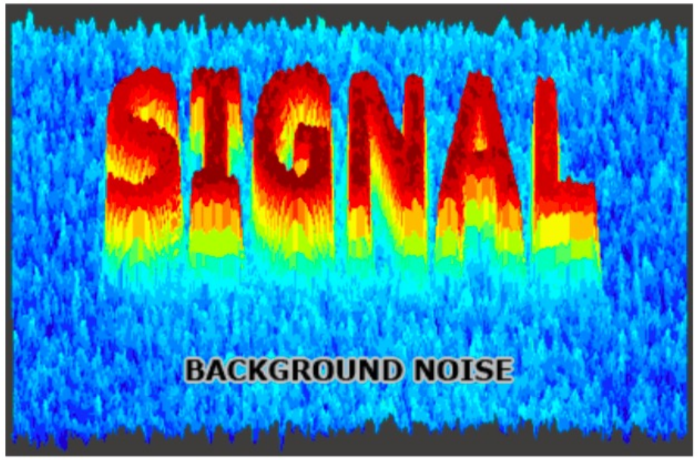
Difference of brightness in adjacent regions of the image

- grey-level (luminance) contrast
- color contrast

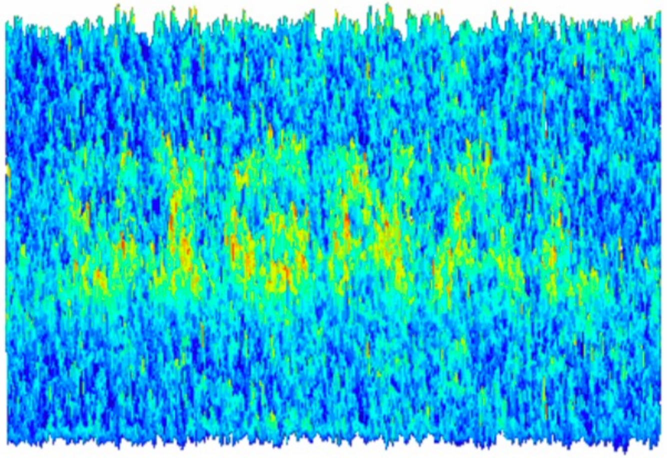
Signal-to-noise ratio

Signal-to-Noise ratio (SNR) = S_{RMS} / N_{RMS}

- RMS: root mean square



high SNR



low SNR

Image Operations

Provide the clinician with some means to:

- enhance contrast of local features
- remove noise and other artifacts
- enhance edges and boundaries
- composite multiple images for a more comprehensive view

There are two basic operations: global and local

Global operations:

- operate on the entire set of pixels at once
- examples: brightness and contrast enhancement

Local operations:

- operate only on a subset of pixels (in a pixel neighborhood)
- examples: edge detection, contouring, image sharpening, blurring

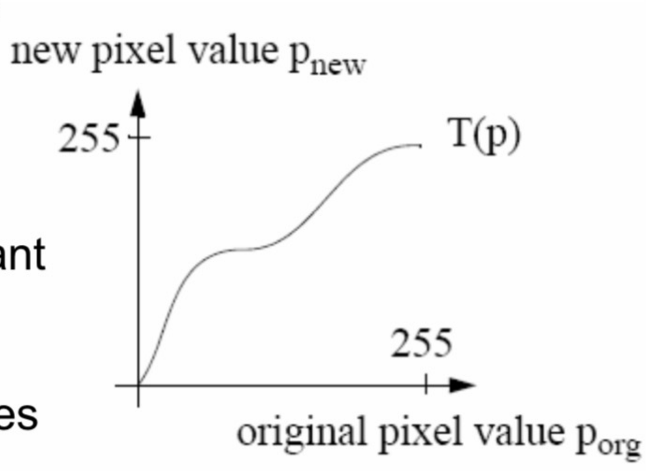
Gray level transformation (basics)

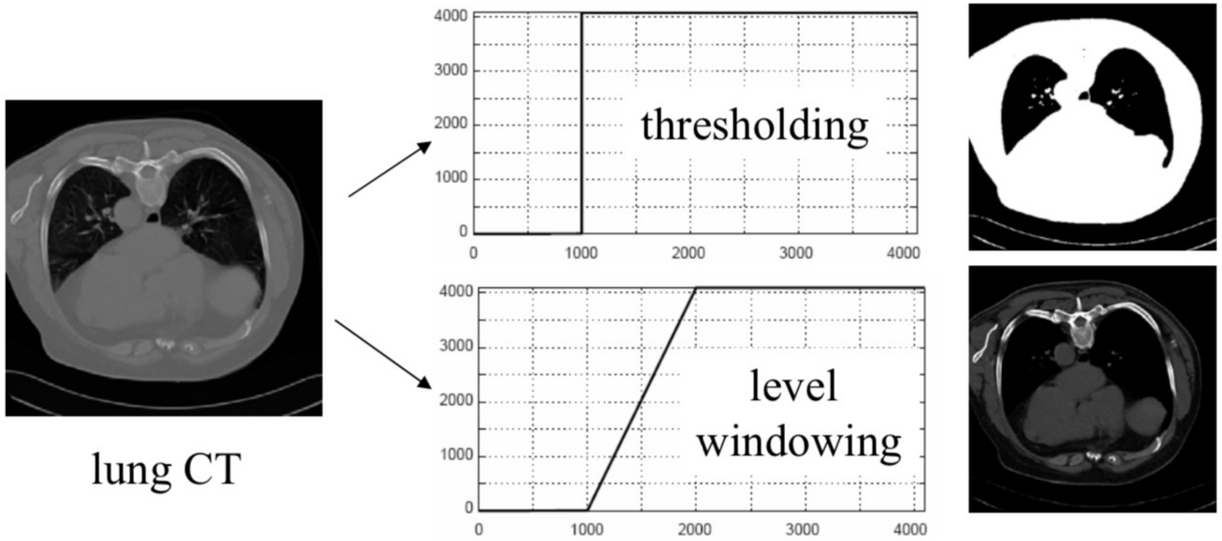
We only have a fixed number of grey levels that can be displayed or perceived

- need to use this ‘real estate’ wisely to bring out the image features that we want

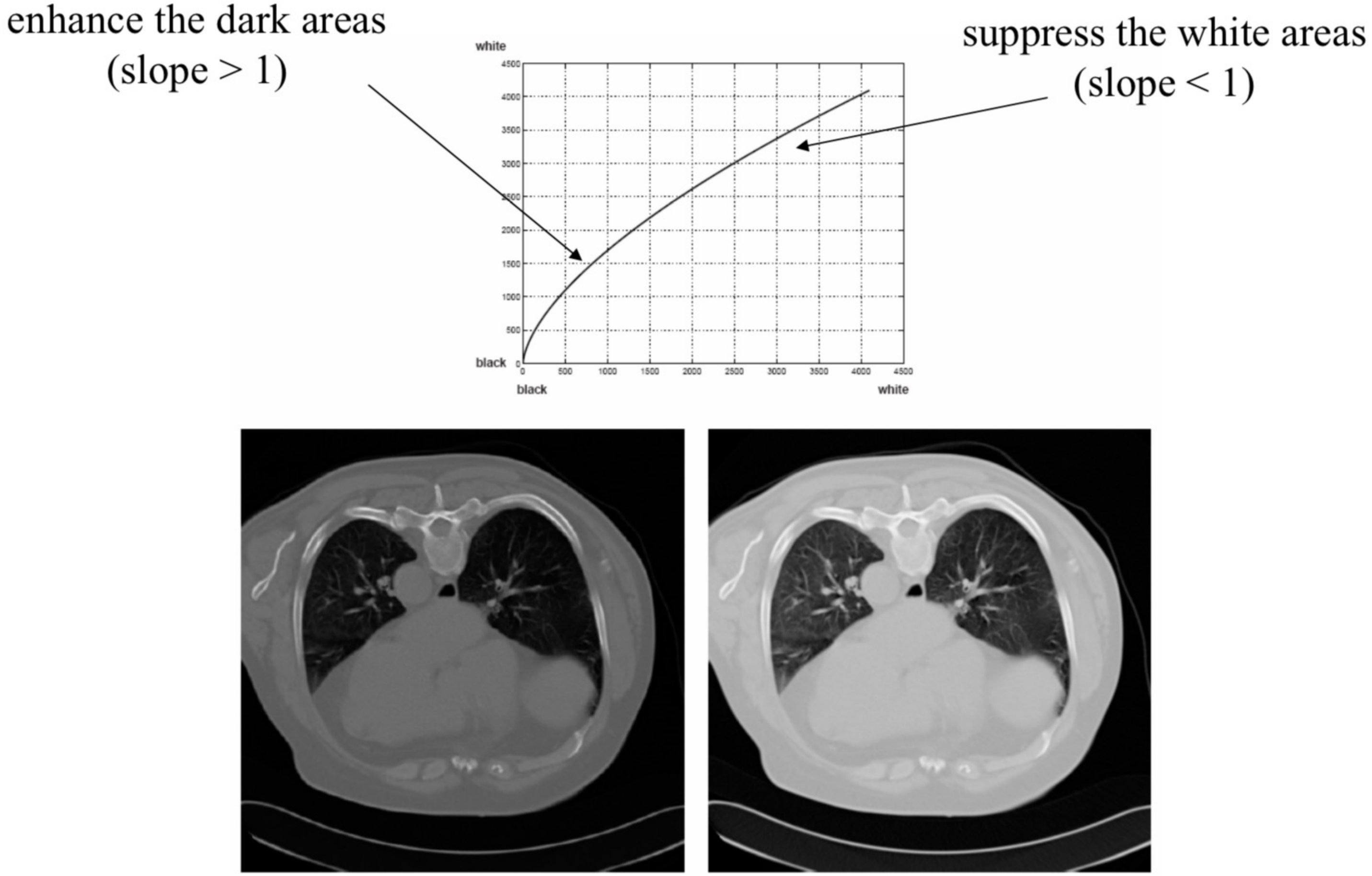
Use *intensity transformations* T_p

- enhance (remap) certain intensity ranges at the cost of compressing others





Gray level transformation (enhancement)

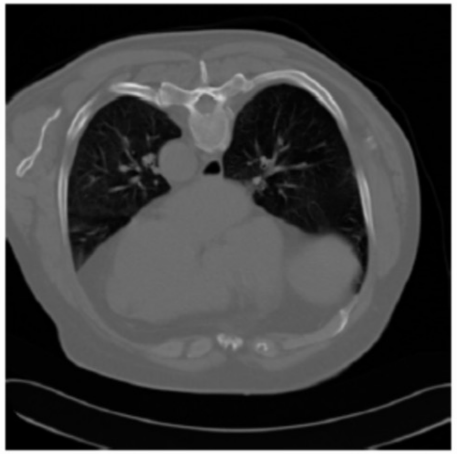


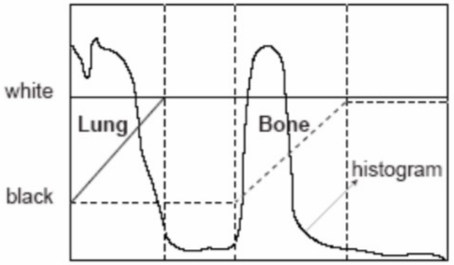
original enhanced

Gray level transformation (windowing)

Dedicate full contrast
to either bone or lungs

original lung
CT image





bi-modal
histogram

bone window

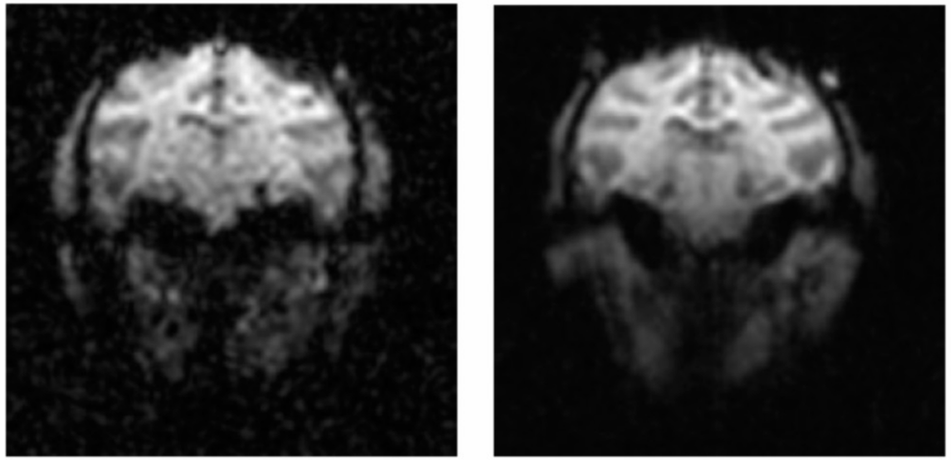
lung window

Noise averaging

Assume a pixel value p is given by: p = *signal* + *noise*

- E(*signal*) = *signal*
- E(*noise*) = 0, when noise is random

Thus, averaging (adding) multiple images of a steady noisy object will eliminate, or at least reduce, the noise



original

after averaging 16 subsequently acquired images

Eliminating Background

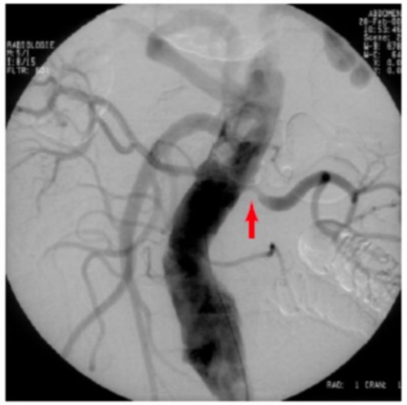
In angiography, radio-opaque contrast agents (injected into the bloodstream) are used to enhance the perfused vessels

An X-ray image is taken when the radio-opaque bolus of blood is coming through

- however, the background reduces the contrast of the dye
- subtracting the (constant) background from the (dynamic) radiographic image leaves just the perfused structures (angio image)

after injection
(radio-image)

background
(mask image)



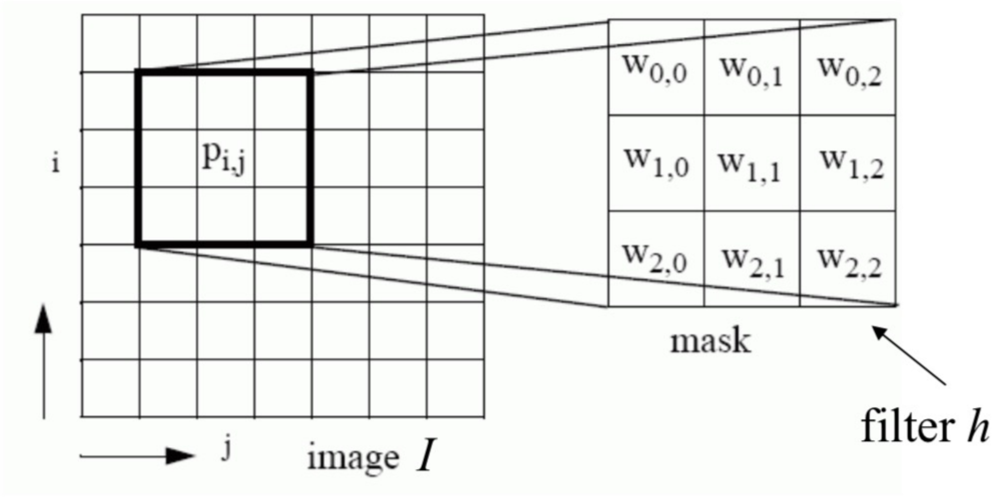
just the bolus
(angio image)

Discrete Filters

We say *discrete filters* since they operate on a discretized signal, the image

- to implement discrete filters we use discrete convolution

$h * I$



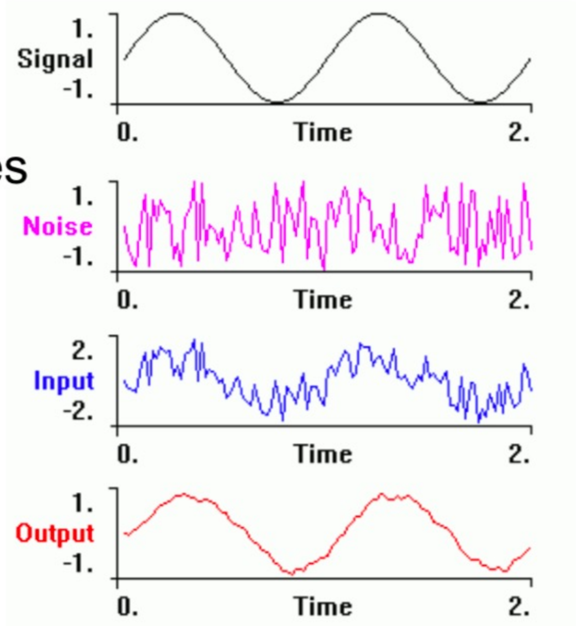
Procedure:

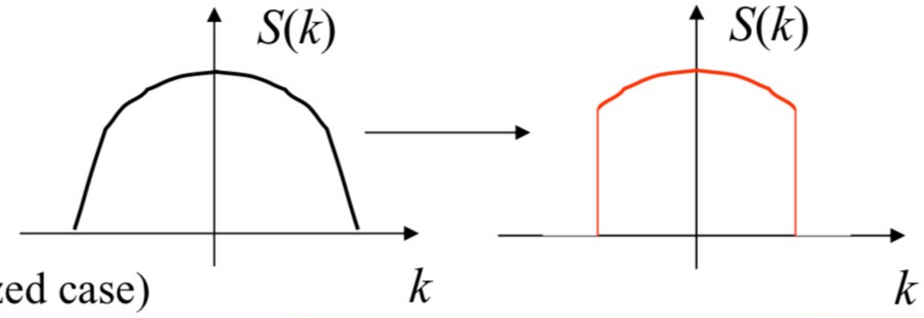
- place a weight matrix or *mask* at each pixel location p_{ij}
- this mask weighs the pixel's neighborhood and determines the output pixel's value
- important: do not replace the computed values into the original image, but write to an output image

Lowpass filters

Smoothing (averaging):

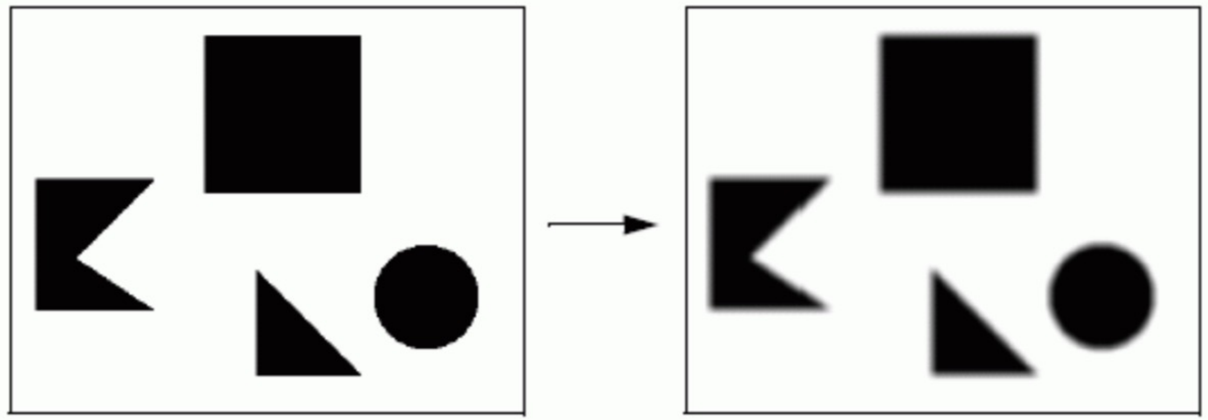
- also called *low-passing:* keeps the low frequencies, but reduces the high frequencies
- removes noise and jagged edges
- but also blurs the signal





(idealized case)

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$



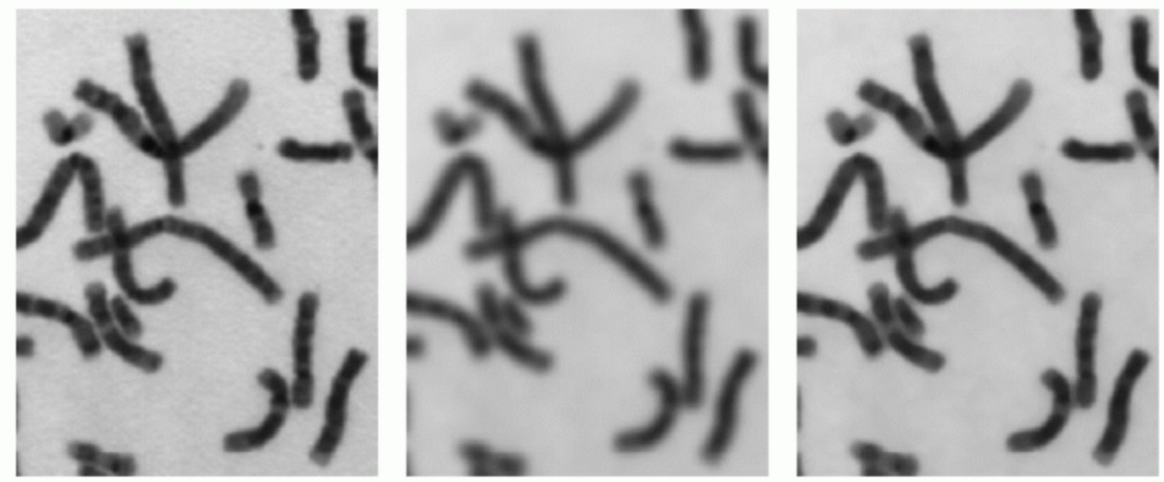
Median Smoothing

A non-linear filter, best used to remove speckle noise

- a regular smoothing filter would blur the speckles (and the signal)
- the median filter will elimate the speckle and leave the signal as is

Procedure:

- convolve with a mask as usual
- but this time, for each mask position, sort the values under the mask
- pick the median and write to the output image
- the speckle pixel will be an outlier and not be selected as the median



original smoothed median filtered

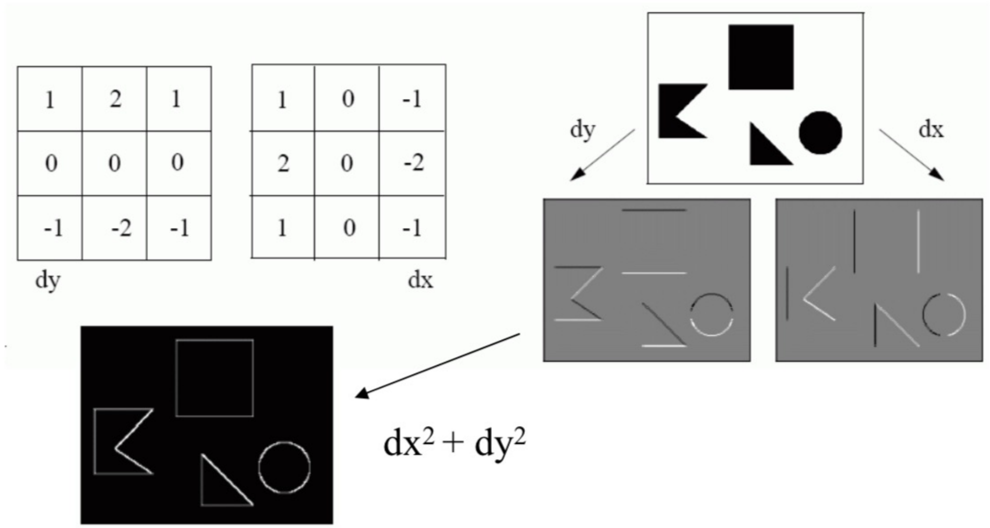
Highpass filters

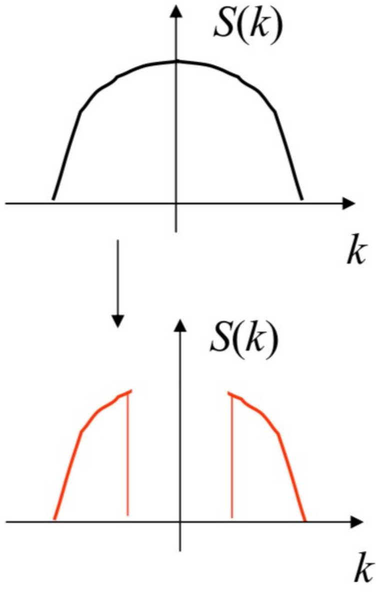
Edge detector / enhancer:

$$\nabla I = \nabla h * I \qquad \text{first derivative (gradient)}$$

$$\nabla^2 I = \nabla^2 h * I \qquad \text{second derivative (Laplacian)}$$

- also called *high-passing:* keeps the high frequencies, but reduces the low frequencies
- enhances edges and contrast
- but also enhances noise and jagged edges



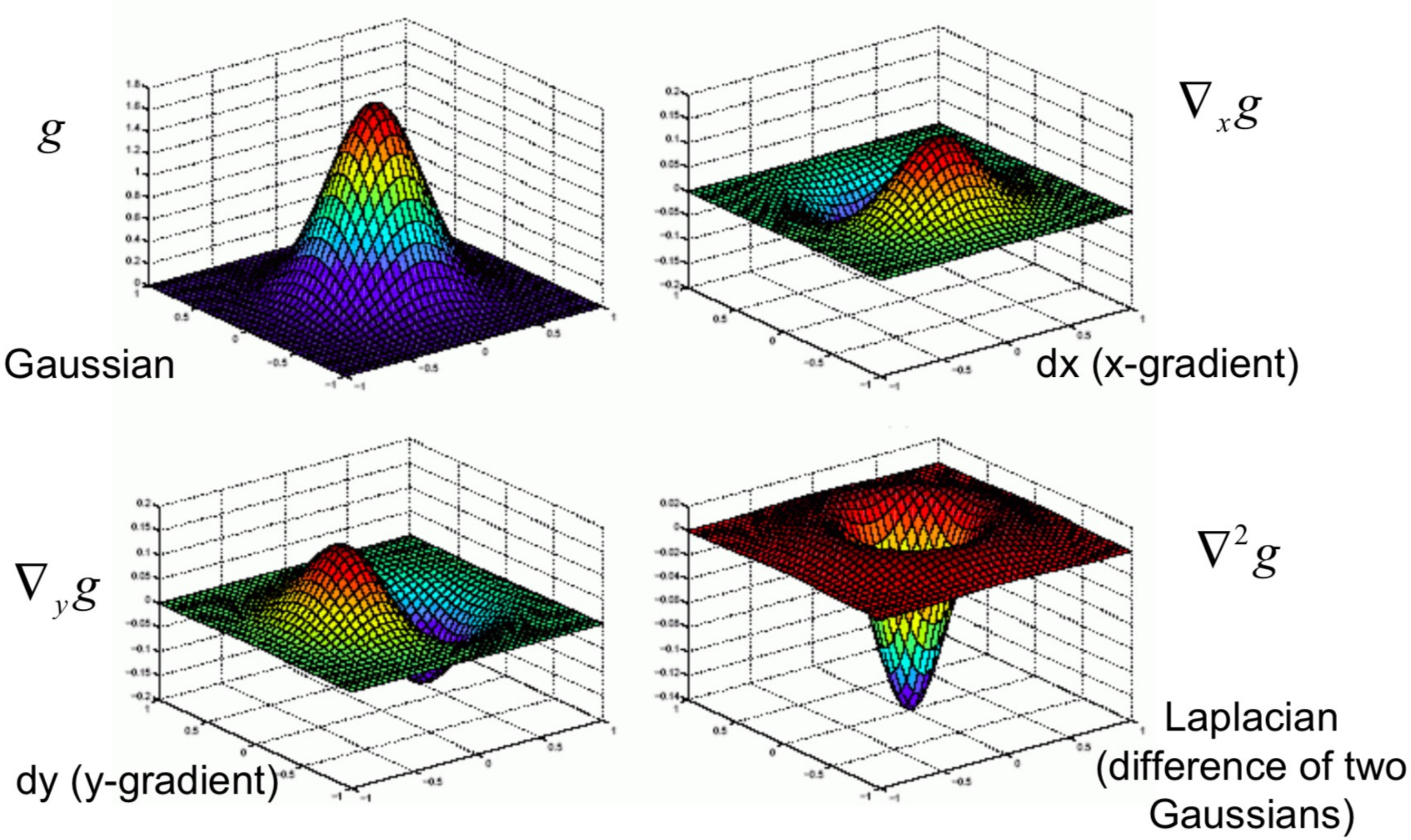


(idealized case)

Gaussian kernel

The Gaussian kernel is a popular filter function

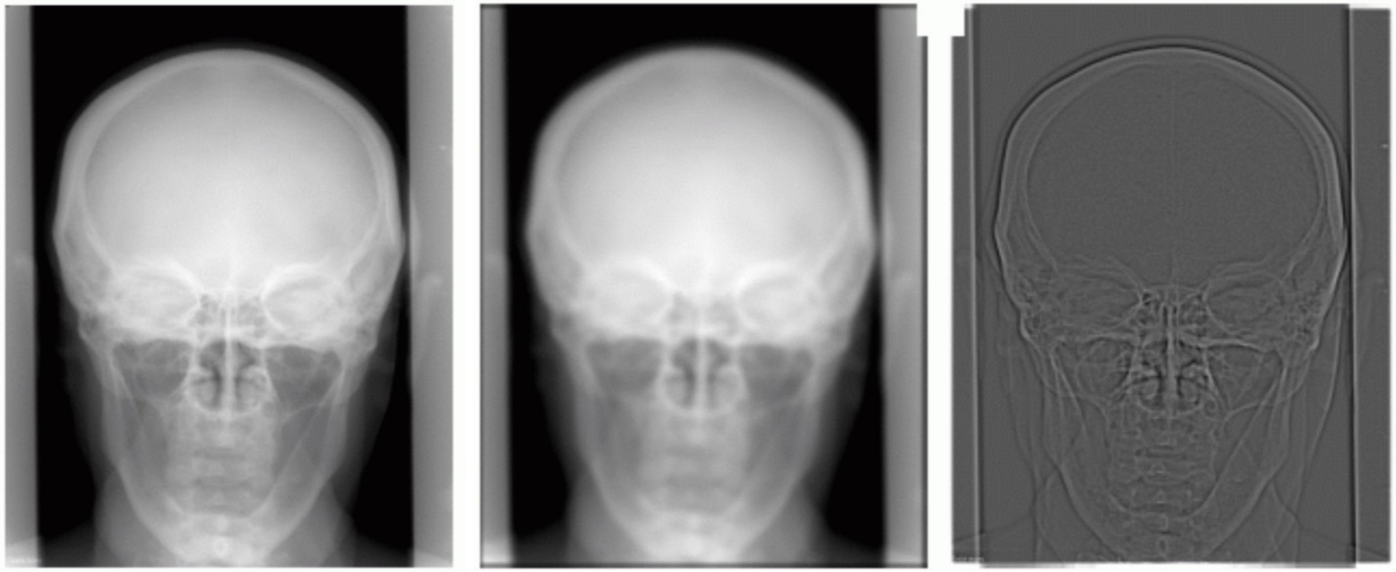
- see book for 3x3 convolution masks



Highpass filters

Several useful effects can be achieved by subsequent filtering with different masks (kernels) and/or multi-image operations

Subtracting a smoothed image from the original image leaves the edges (the high frequencies):



original I

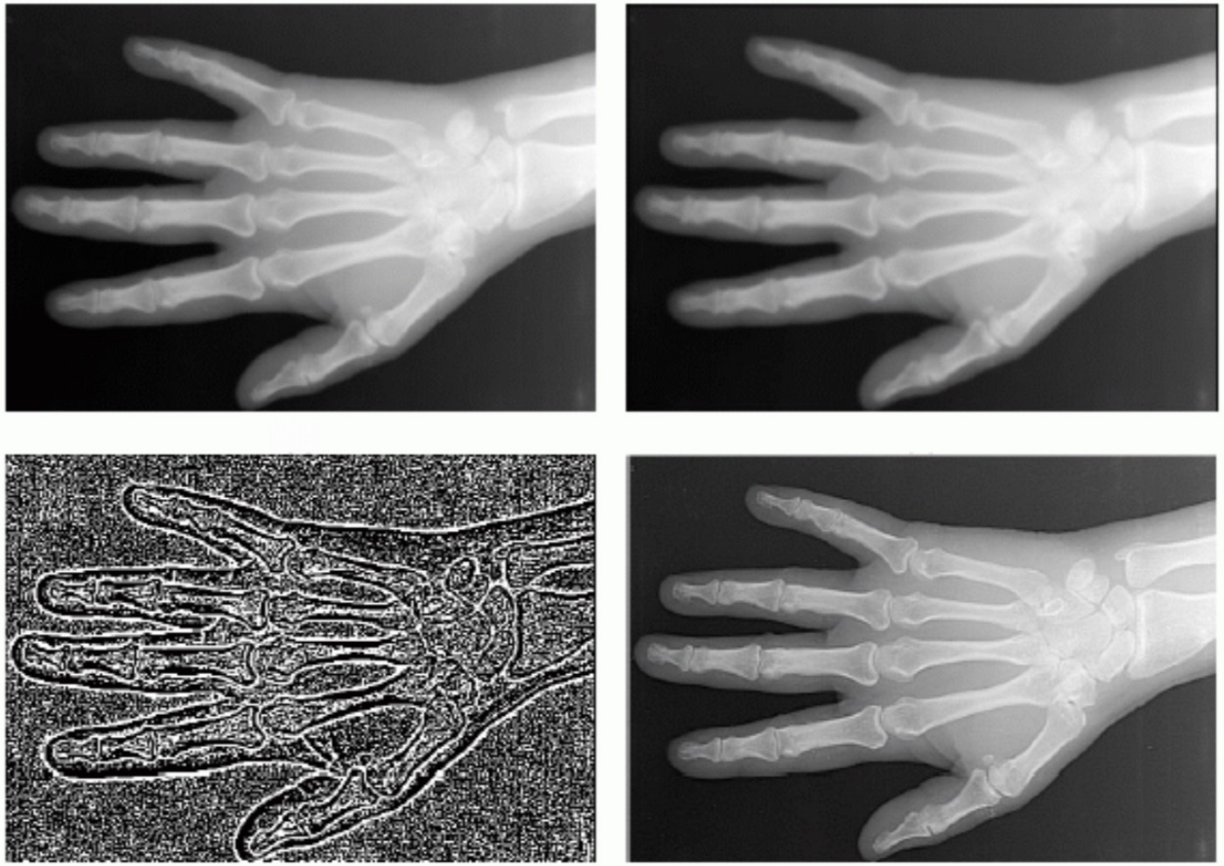
smoothed $g * I$

original - smoothed $I - g * I$

Unsharp masking

Places the enhanced edges on top of a smoothed original

I



$g * I$

$I - g * I$

$g * I + (1+\alpha)(I - g * I)$

Shortcomings

Windowing enhances contrast only for a specific range of grey levels (not sensitive to edges)

- strong edges with already good contrast are further enhanced

Edge enhancement (such as sharp masking) only boosts features within a certain frequency band

- this frequency band is determined by filter size -- features outside that band are not enhanced (cannot see many scales at the same time)
- all grey value variations (within that band) are enhanced, even if they already had good contrast

original small filter: small detail large filter: large-scale variations

Shortcomings (cont.)

One more example: digital radiograph of a foot

original

edge enhanced

window/level operation

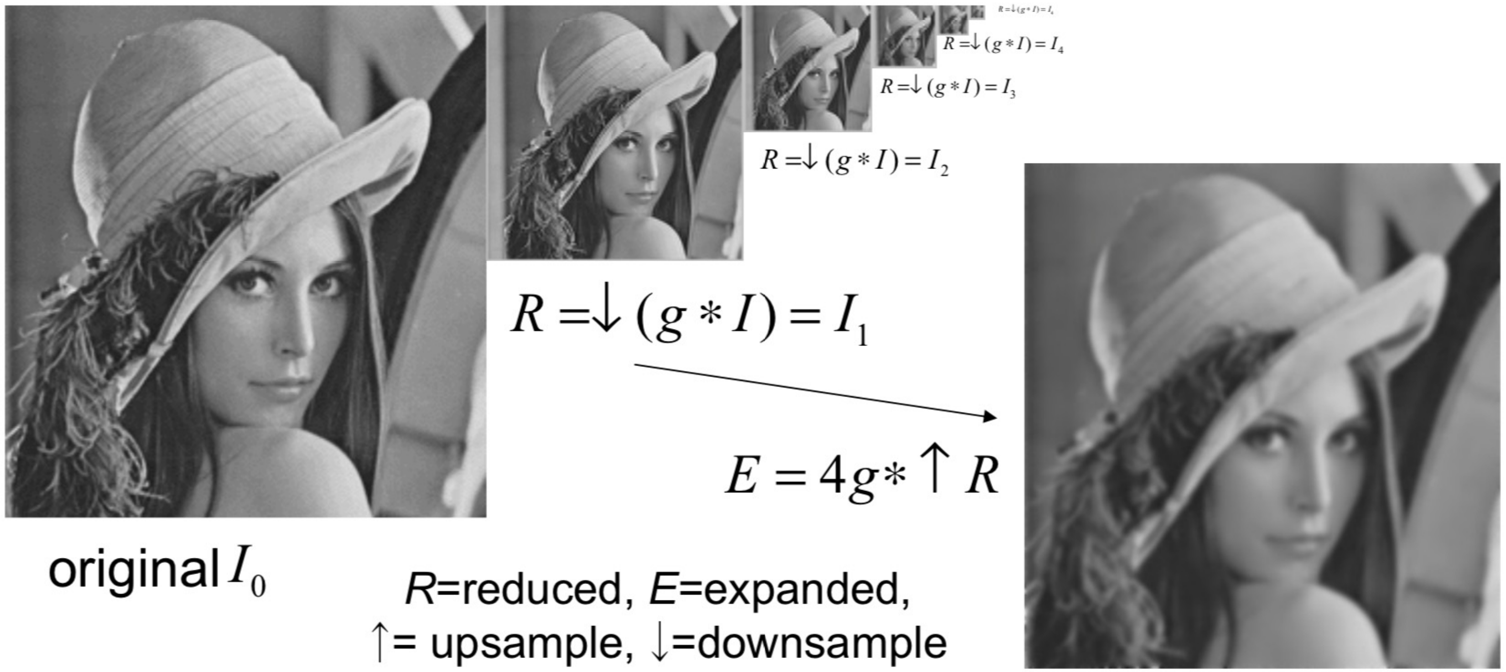
Multi-scale image enhancement

Designed to overcome these shortcomings

- enhancements will be visible at all scales at the same time
- this requires a pyramid of detail images that are added together

Image pyramid of lowpassed images

- a hierarchy of images, repeatedly lowpassed at scales of power of 2



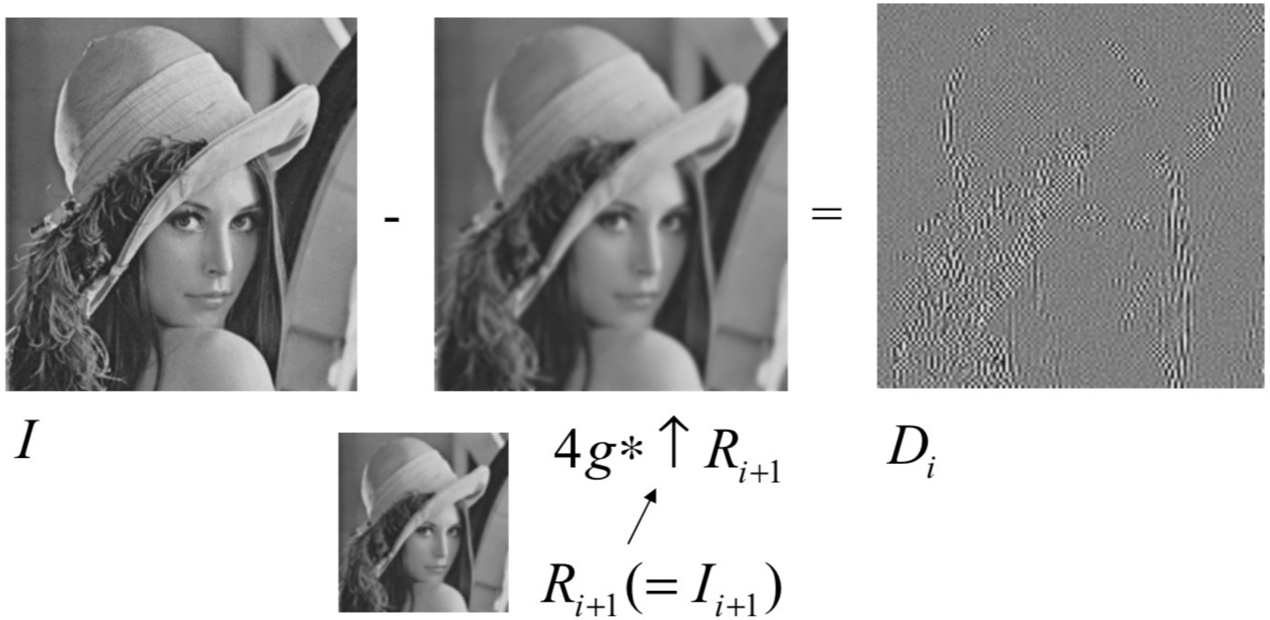
Detail images

We have seen detail enhancement by high-pass filtering

- the result is called a *detail image*

We can create an image pyramid of detail images

- constructed by subtracting the smoothed image at the corresponding pyramid level from the original: $D_i = I - I_i * g$
- this gives us the detail D_i at scale i



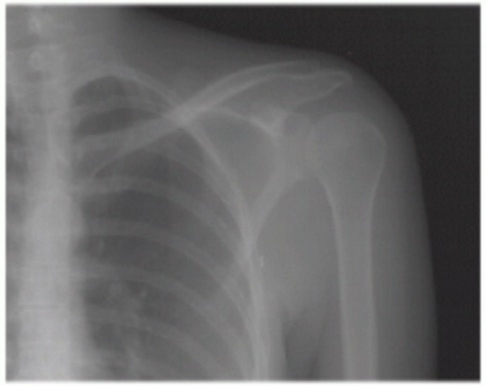
Detail Pyramid

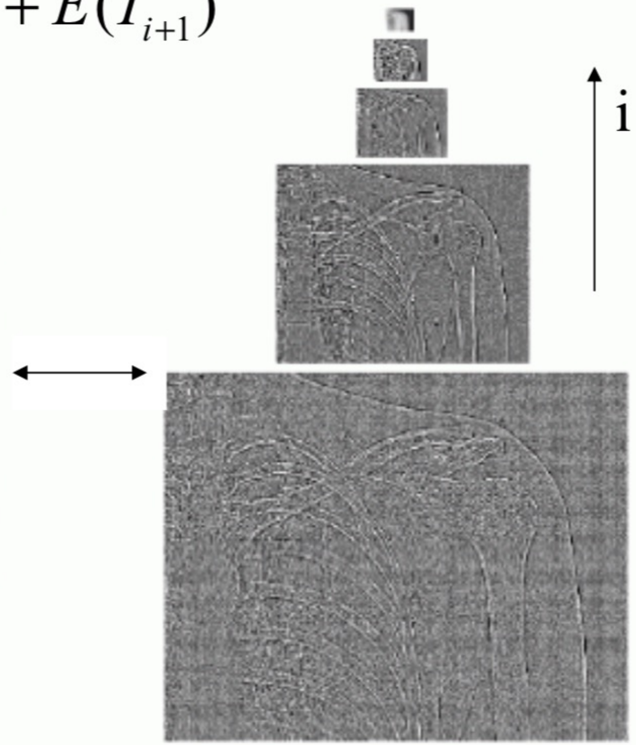
A representation of the details occuring at multiple levels of scale is called *detail pyramid*

We can reconstruct the image at level *i* by adding the expanded image at level (*i*+1) to the detail at level *i*:

$$I_i = D_i + E(I_{i+1})$$

By adding all the details we can assemble the original image:

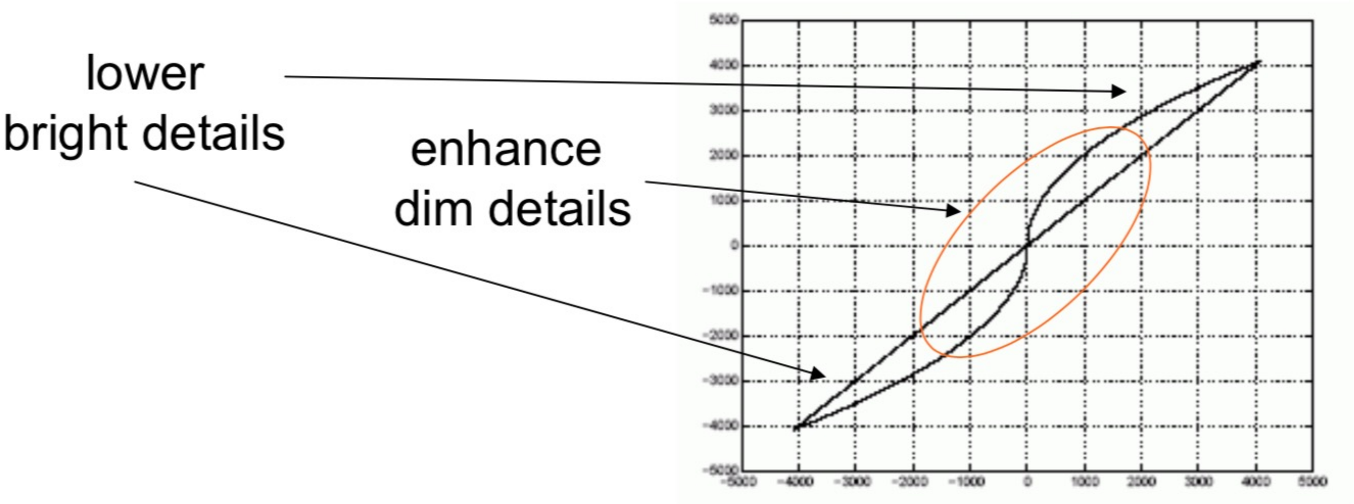




Multi-scale enhancement

Strategy:

- create pyramid of detail images D_i
- apply a non-linear grey-scale transformation to each of the D_i
- this emphasizes the low-contrast details (previously invisible)
- it de-emphasizes the high-contrast details (to just noticeable levels)



- finally, re-assemble the image by adding these transformed detail images recursively

Example

This strategy has been employed in the MUSICA algorithm

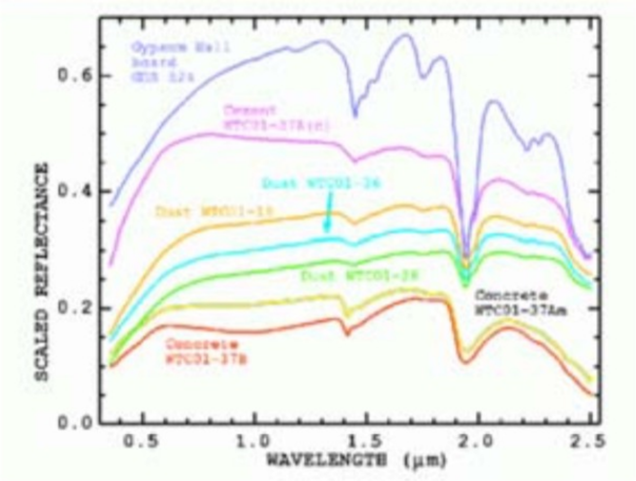
- developed by the company Agva Gevaert
- routinely in used in digital radiography in hospitals worldwide

edge enhanced

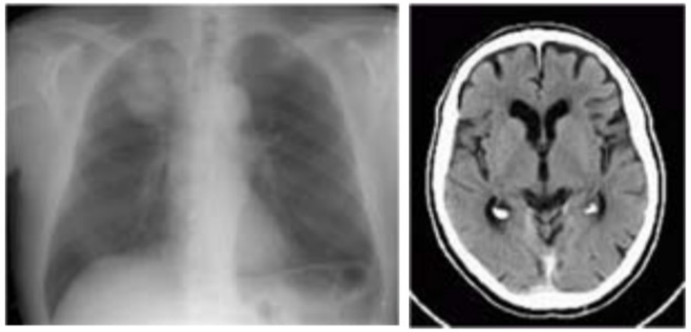
window/level operation

MUSICA

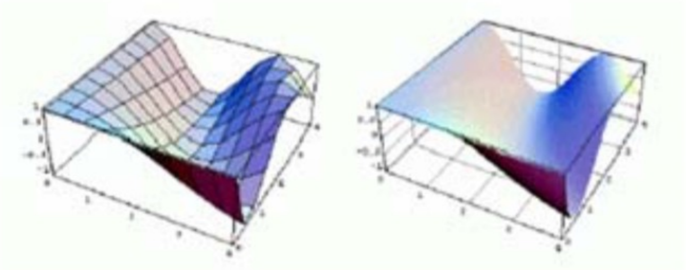
Dimensions



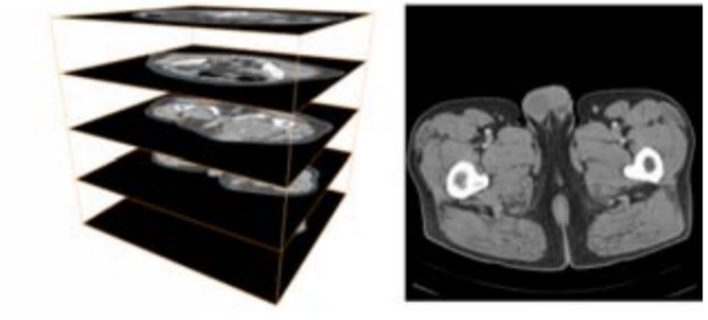
1D signal $f(x)$



2D signal $f(x, y)$



2D signal, shown as height field



3D signal $f(x, y, z)$

4D signal $f(x, y, z, t\text{=time})$
example: 3D heart in motion

Even/Odd Functions

Signal is even if $s(-x) = s(x)$

- denote as s_e

$$\int_{-\infty}^{+\infty} s_e(x)dx = 2\int_{0}^{+\infty} s_e(x)dx$$

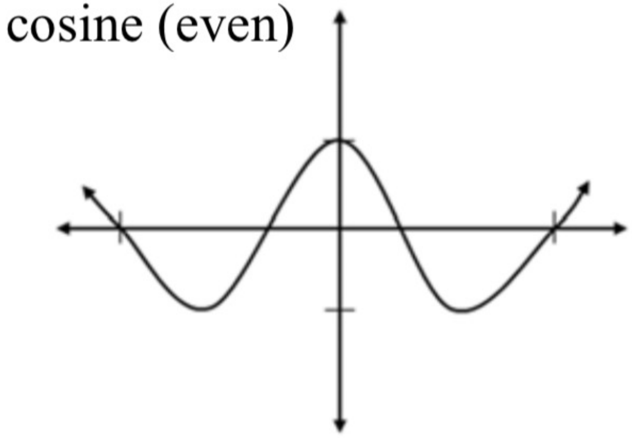
Signal is odd if $s(-x) = -s(x)$

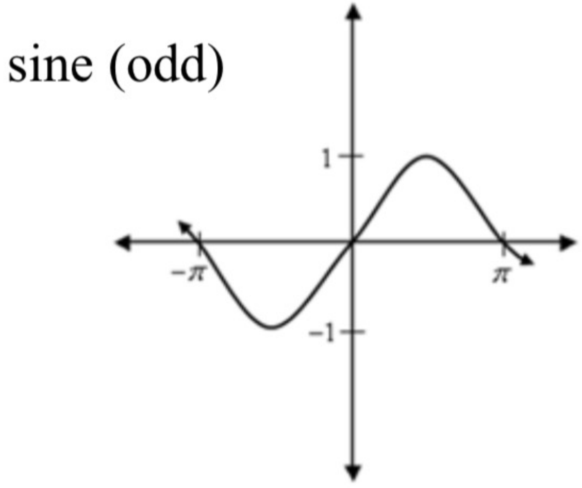
- denote as s_o

$$\int_{-\infty}^{+\infty} s_o(x)dx = 0$$

Can write any signal as a sum of its even and odd part:

$$s(x) = [\frac{s(x)}{2} + \frac{s(-x)}{2}] + [\frac{s(x)}{2} - \frac{s(-x)}{2}]$$
$$= s_e(x) + s_o(x)$$





Periodic Signals

A signal is periodic if $s(x+X) = s(x)$

- we call X the period of the signal
- if there is no such X then the signal is aperiodic

Sinusoids are periodic functions

- sinosoids will play an important role in this course

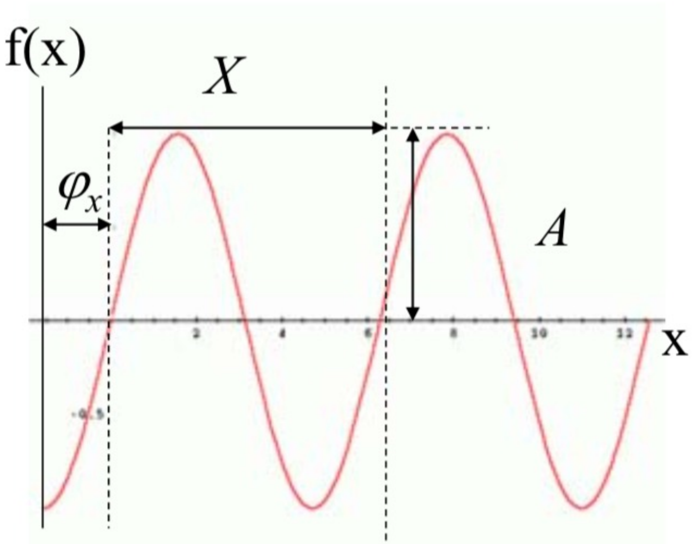
Write as:

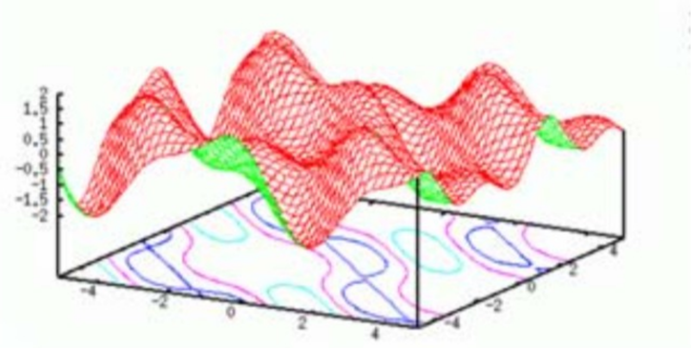
$$A \sin(\frac{2\pi x}{X} + \varphi_x)$$

- where φ_x is the phase shift and A is the amplitude

Sinusoids can combine

- they can also occur in higher dimensions:





Complex Numbers

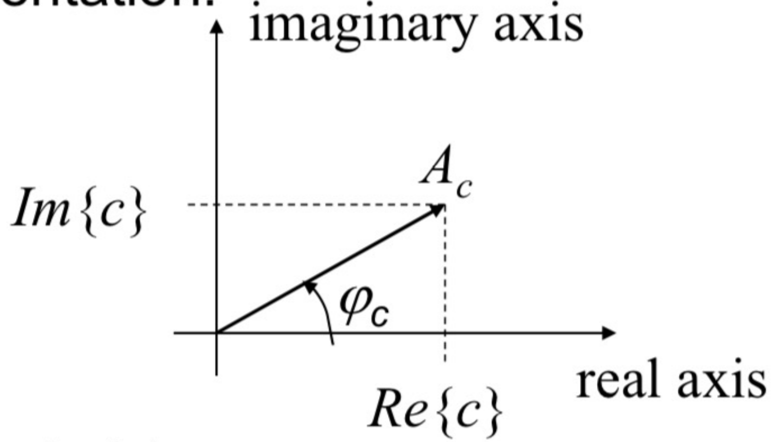
A complex number *c* has a real and and an imaginary part:

- $c = Re\{c\} + i\, Im\{c\}$ (cartesian representation) $i = \sqrt{-1}$
- here, *i* always denotes the complex part

We can also use a polar representation:

$$A_c = \sqrt{\mathrm{Re}\{c\}^2 + \mathrm{Im}\{c\}^2}$$

$$\varphi_c = \tan^{-1}\left(\frac{\mathrm{Im}\{c\}}{\mathrm{Re}\{c\}}\right)$$



Now think of *c* as a periodic signal *s*(*x*):

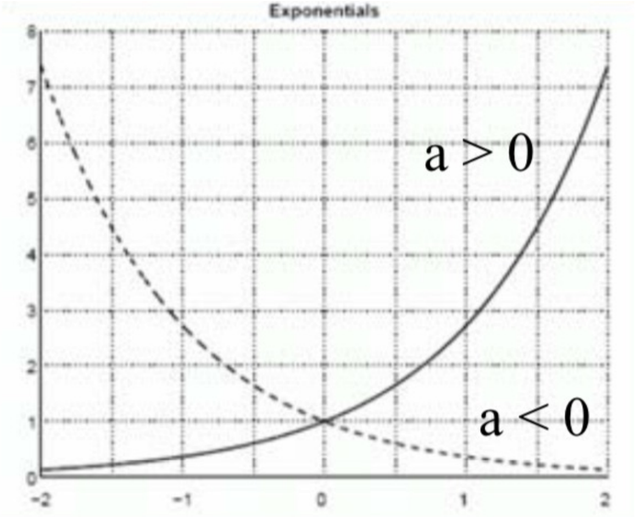
- then the pointer (A_c, φ_c) rotates with period *X*, that is, it completes one rotation after each integer multiple of *X*
- if there is a phase shift φ_x then the pointer simply is already located at (A, φ_x) when $x=0$
- considering *c* a 2D vector: $Re\{c\} = A_c cos(\varphi_c)$ and $Im\{c\} = A_c\, sin(\varphi_c)$

Important Signals (1)

Exponential *exp*

$$\exp(ax) = e^{ax}$$

- when $a > 0$ then *exp* increases with increasing x
- when $a < 0$ then *exp* approximates 0 with increasing x

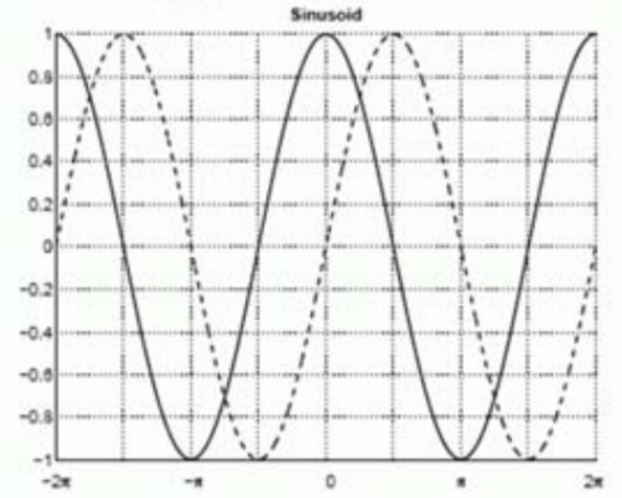


Complex exponential / sinusoid:

$$Ae^{i(2\pi kx+\phi)} = A(\cos(2\pi kx+\phi)+i\sin(2\pi kx+\phi))$$

As before

- the *cos* term is the signal's real part
- the *sin* term is the signal's imaginary part
- *A* is the amplitude, φ the phase shift, *k* determines the frequency



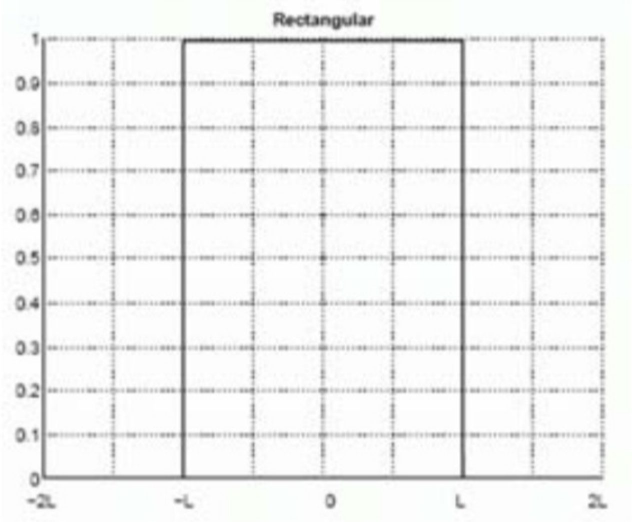
Important Signals (2)

Rectangular function:

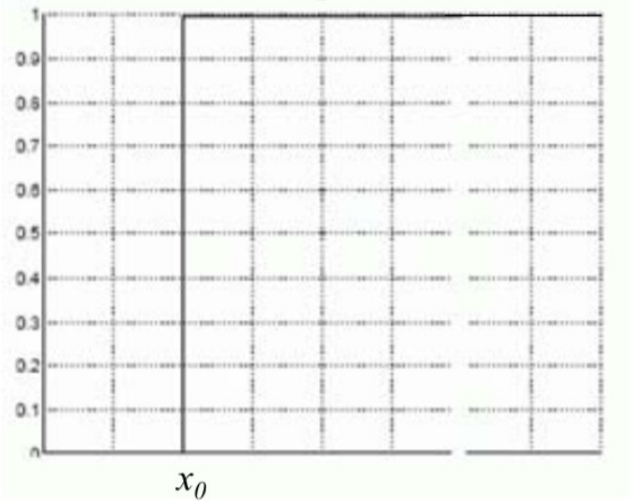
$$\Pi(\frac{x}{2L}) = 1 \quad \text{for} \quad |x| < L$$

$$= \frac{1}{2} \quad \text{for} \quad |x| = L$$

$$= 0 \quad \text{for} \quad |x| > L$$



step



Step function:

$$u(x - x_0) = 0 \quad \text{for} \quad x < x_0$$

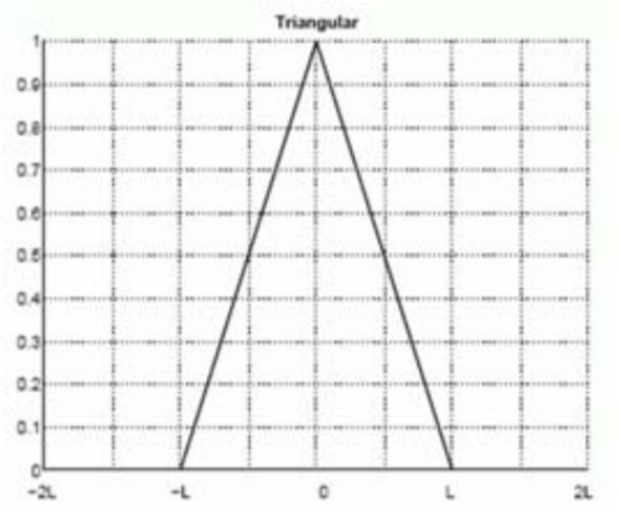
$$= \frac{1}{2} \quad \text{for} \quad x = x_0$$

$$= 1 \quad \text{for} \quad x > x_0$$

Important Signals (3)

Triangular function:

$$Tri\left(\frac{x}{2L}\right) = 1 - \frac{|x|}{L} \quad \text{for} \quad |x| < L$$
$$= 0 \qquad\qquad \text{for} \quad |x| > L$$



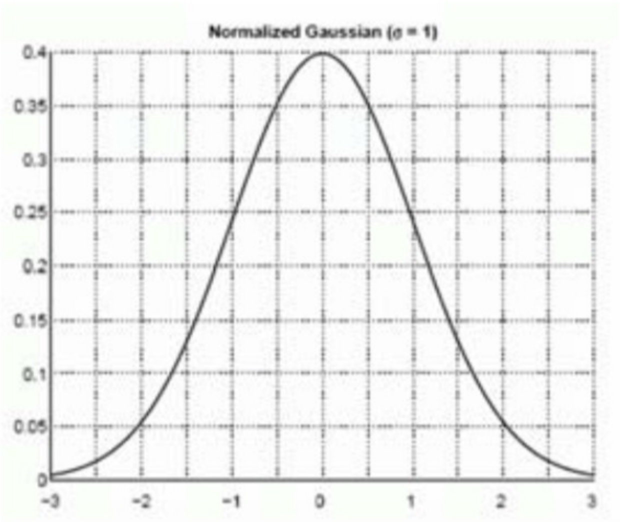
Normalized Gaussian:

$$G_n(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

μ is the mean

σ is the standard deviation

- normalized means that the integral for all *x* is 0

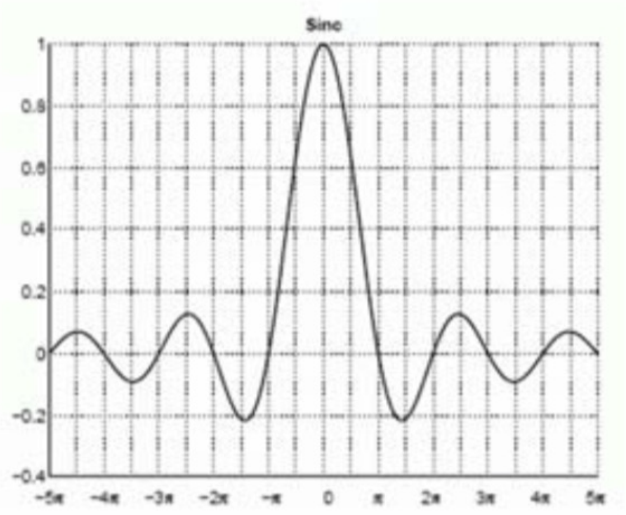


Important Signals (4)

Sinc function:

$$sinc(x) = \frac{\sin(\pi x)}{\pi x}$$

- sinc(0) = 1 (L'Hopital's rule)



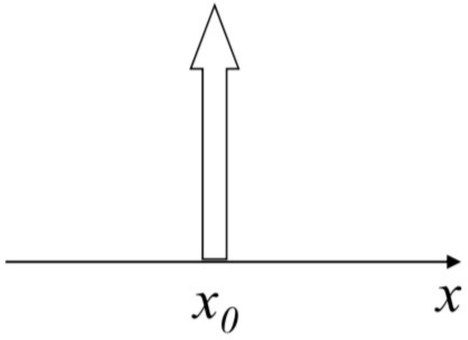
Dirac impulse:

$$\delta(x - x_0) = 0 \quad \text{for } x \neq x_0$$

$$\int_{-\infty}^{+\infty} \delta(x - x_0)dx = 1$$

- an important property is its sifting property:

$$\int_{-\infty}^{+\infty} s(x)\delta(x - x_0)dx = s(x_0)$$



a "needle" spike of infinite height at x=x_0

Linear Systems (1)

System response *L:*

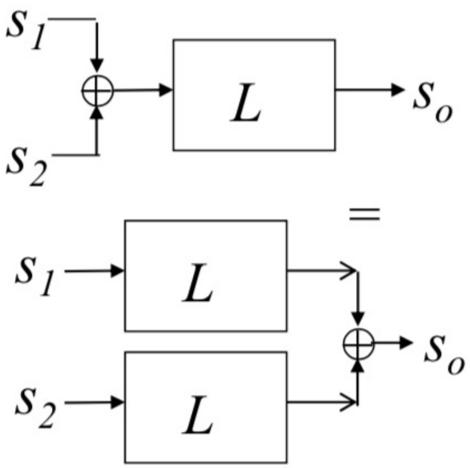
$$s_o = L\{s_i\}$$

- might be a function of time *t* or space *x*

$$s_o(t) = L\{s_i(t)\} \quad \text{or} \quad s_o(x) = L\{s_i(x)\}$$

Finding the mathematical relationship between in- and output is called *modeling*

Linear systems fulfill *superposition principle*:



$$L\{c_1 s_1 + c_2 s_2\} = c_1 L\{s_1\} + c_2 L\{s_2\} \quad \forall c_1, c_2 \in \Re$$

where s_1, s_2 are arbitrary signals

- for example, consider an amplifier with gain A:

$$L\{c_1 s_1 + c_2 s_2\} = A(c_1 s_1 + c_2 s_2)$$
$$= c_1 A s_1 + c_2 A s_2 = c_1 L\{s_1\} + c_2 L\{s_2\}$$

Linear Systems (2)

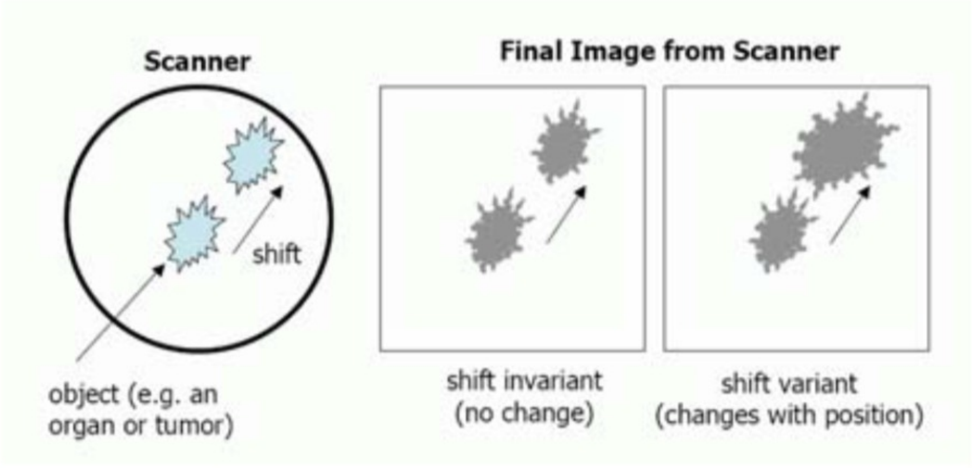
An example for a non-linear system:

$$L\{c_1 s_1 + c_2 s_2\} = (c_1 s_1 + c_2 s_2)^2$$
$$\neq (c_1 s_1)^2 + (c_2 s_2)^2$$

Time-invariance (shift-invariance = LSI):

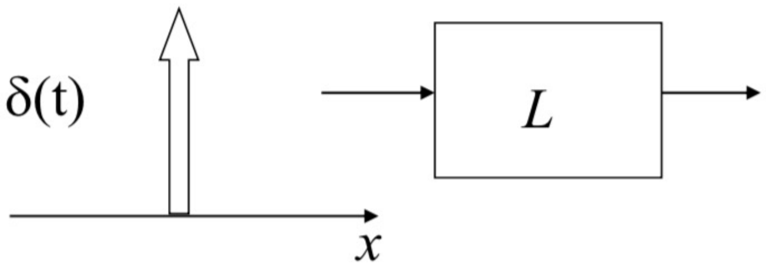
- properties of *L* do not change over time (spatial position), that is:

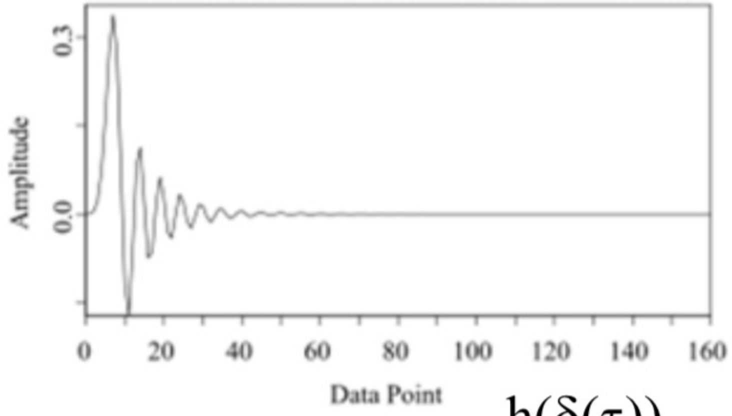
$$s_o(x) = L\{s_i(x)\} \text{ then } s_o(x-X) = L\{s_i(x-X)\}$$



Impulse Response (1)

A system's response to a Dirac impulse is called *impulse response h*:





h(δ(τ))

Start with:

$$s_i(x) = \int_{-\infty}^{+\infty} s_i(\xi)\delta(x-\xi)d\xi$$

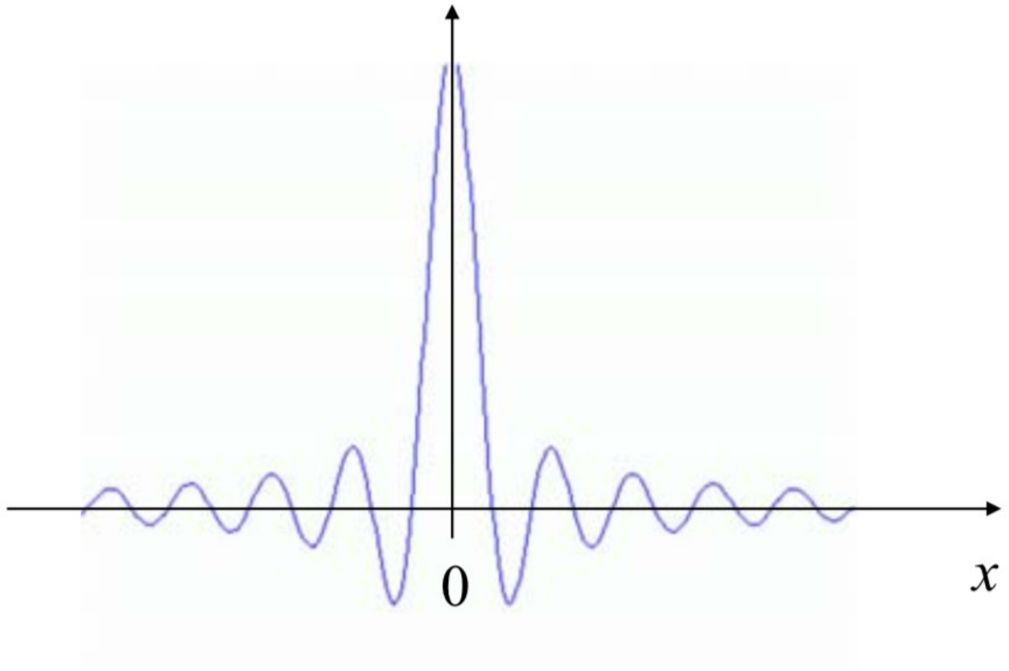
Then write:

$$s_o(x) = L\{s_i\} = \int_{-\infty}^{+\infty} s_i(\xi)L\{\delta(x-\xi)\}d\xi = \int_{-\infty}^{+\infty} s_i(\xi)h(x-\xi)d\xi$$

Impulse Response (2)

In practice we use non-causal impulse responses

- appear symmetric in their waveform



Convolution

The expression

$$s_o(x) = \int_{-\infty}^{+\infty} s_i(\xi)h(x-\xi)d\xi = s_i * h$$

is called *convolution*, defined as:

$$s_1(x) * s_2(x) = \int_{-\infty}^{+\infty} s_1(\xi)s_2(x-\xi)d\xi = \int_{-\infty}^{+\infty} s_1(x-\xi)s_2(\xi)d\xi$$

Procedure:

for each *x* do:

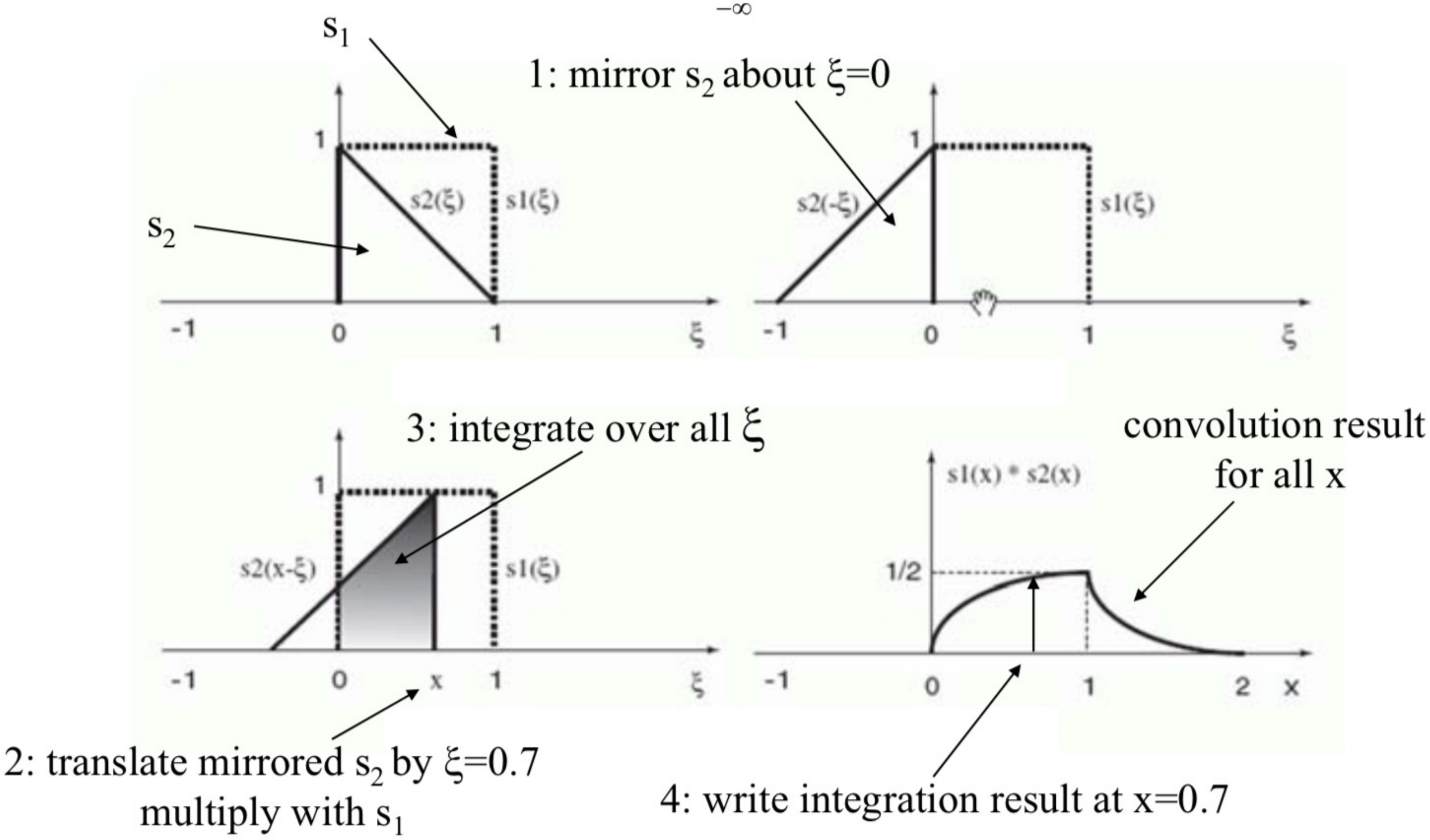
1: mirror s_2 about $\xi = 0$ (change ξ το $-\xi$)
2: translate mirrored s_2 by $\xi = x$
3: multiply s_1 and mirrored s_2
4: integrate the resulting signal

See next slides for an example and detailed explanation…

Convolution: Example

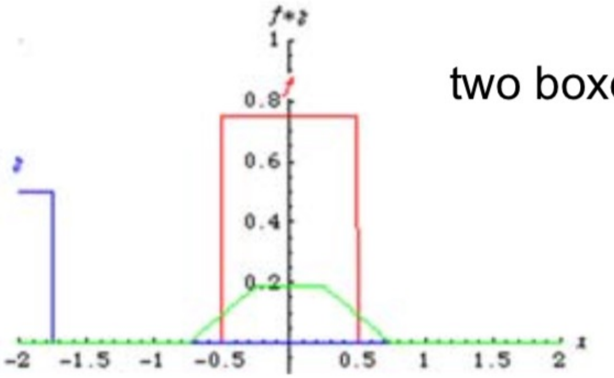
Example x=0.7:

$$s_1(x) * s_2(x) = \int_{-\infty}^{+\infty} s_1(\xi) s_2(x-\xi) d\xi$$

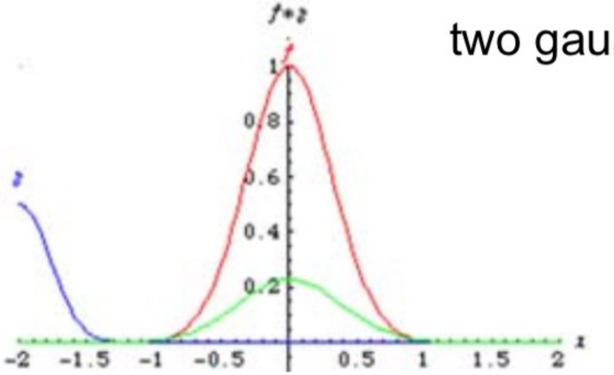


Convolution: More Examples

Animated gifs:
- red, blue: convolved signals
- green: convolution result



two boxes

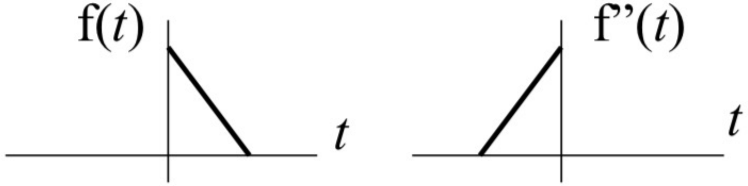


two gaussians

Convolution: Detailed Explanation

Mirroring:

- when you take a function f(t) and mirror it about the y-axis then you get a new function f"(t) = f(-t)



For convolution:

- you have two functions: $f_1(t)$ and $f_2(t)$
- you would like to compute:

$$f(x) = \int_{-\infty}^{+\infty} f_1(t) f_2(x-t)\,dt$$

- but in this form: t increases in f_1 and decreases in f_2, which is not convenient
- to fix this, you mirror $f_2(x\text{-}t)$ into $f_2''(t\text{-}x) = f_2(-(x\text{-}t))$
- now the convolution writes:

$$f(x) = \int_{-\infty}^{+\infty} f_1(t) f_2''(-(x-t))\,dt = \int_{-\infty}^{+\infty} f_1(t) f_2''(t-x)\,dt$$

- at this point you need $f_2''(t)$ which is obtained by mirroring $f_2(t)$: $f_2''(t) = f_2(-t)$
- now you can do the intuitive right-sliding of f_2" for growing x

Convolution Properties

Also defined for multi-dimensional signals:

$$s_1(x, y) * s_2(x, y) = \int_{-\infty}^{+\infty} s_1(x-\xi, y-\zeta) s_2(\xi, \zeta) d\xi d\zeta$$

Some important properties:

- commutativity:

$$s_1 * s_2 = s_2 * s_1$$

- associativity:

$$(s_1 * s_2) * s_2 = s_1 * (s_2 * s_3)$$

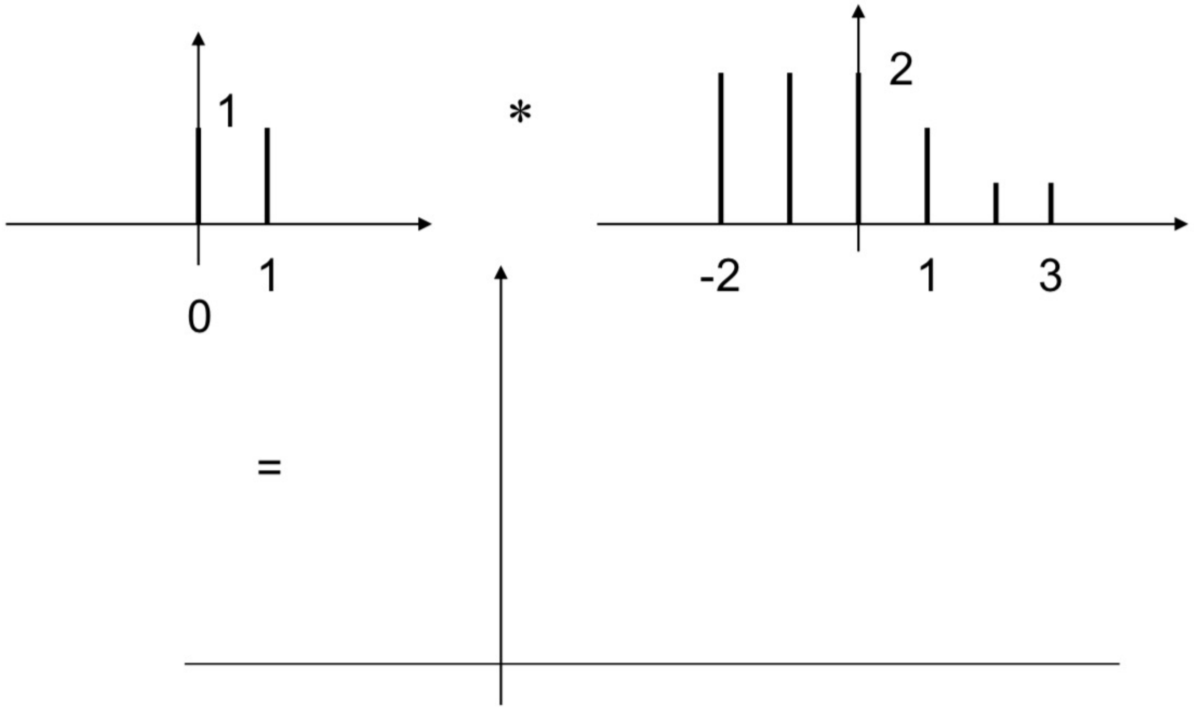
- distributivity:

$$s_1 * (s_2 + s_3) = s_1 * s_2 + s_1 * s_3$$

Discrete Signals

Typically, signals are only available in discrete form

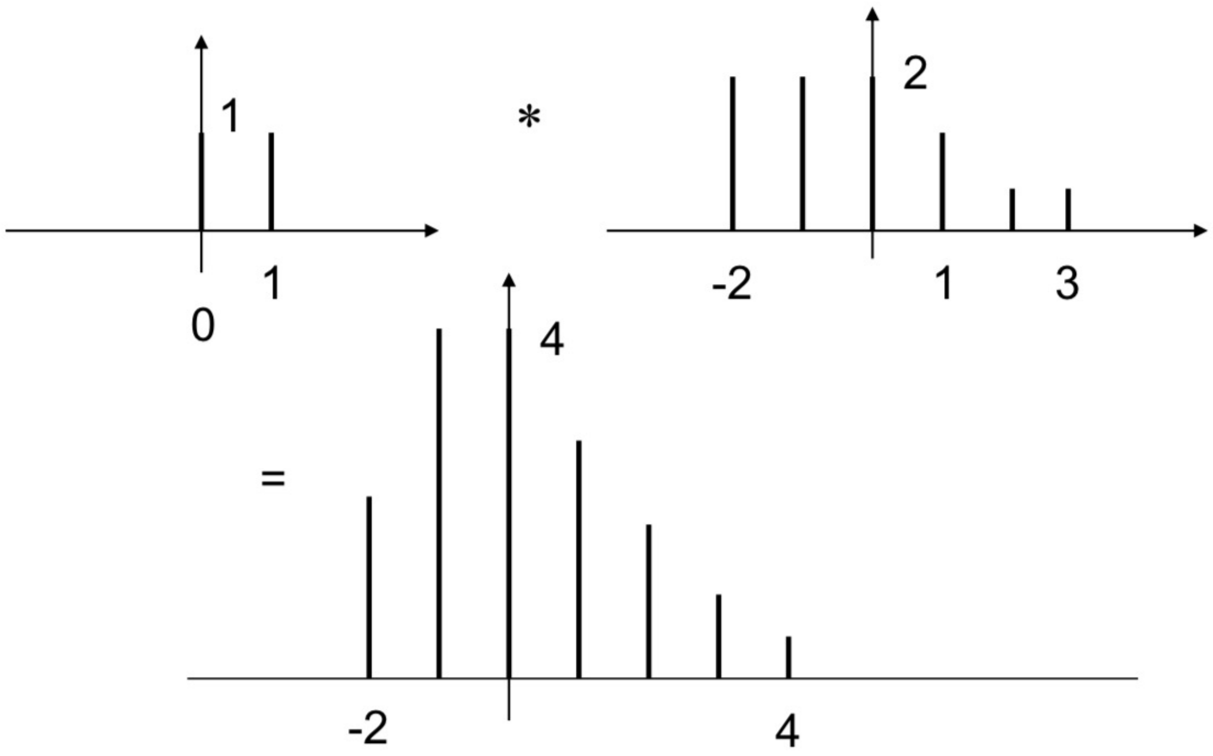
- reconstruction into a continuous signal (for visualization, etc) occurs by overlapping point spread functions (see previous lecture)
- but all computer processing (convolution and others) is done on the discrete representations



Discrete Signals

Typically, signals are only available in discrete form

- reconstruction into a continuous signal (for visualization, etc) occurs by overlapping point spread functions (see previous lecture)
- but all computer processing (convolution and others) is done on the discrete representations



LSI System Responses (1)

Now assume the input is a complex sinusoid with $Ae^{2\pi ikx}$ then:

for now, assume $\varphi=0$

$$s_0(x) = \int_{-\infty}^{+\infty} Ae^{2\pi ik(x-\xi)} h(\xi)d\xi$$

$$= Ae^{2\pi ikx} \int_{-\infty}^{+\infty} e^{-2\pi ik\xi} h(\xi)d\xi$$

$$= Ae^{2\pi ikx} H = S_i H$$

H is called the *Fourier Transform* of *h*(*x*):

$$H = \int_{-\infty}^{+\infty} e^{-2\pi ik\xi} h(\xi)d\xi$$

- *H* is also often called the *transfer function* or *filter*
- the Fourier transform will be discussed in detail shortly

LSI System Responses (2)

H scales, and maybe phase-shifts, the input sinusoid S_i

In essence, we have now two alternative representations:

- determine the effect of *L* on s_i by convolution with *h*: $s_i * h$
- determine the effect of *L* on s_i by multiplication with *H:* $S_i \cdot H$

$$s_i * h \leftrightarrow S_i \cdot H$$

Since convolution is expensive for wide *h*, the multiplication may be cheaper

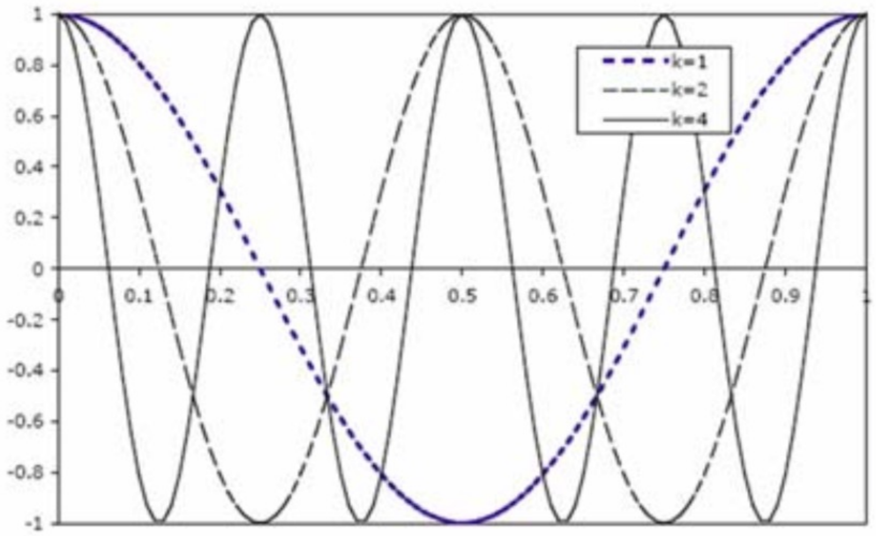
- but we need to perform the Fourier transforms of s_i and *h*
- in fact, there is a “sweetspot”
- more later…

Complex Sinusoids Revisited (1)

Recall the factor *k* in the complex sinusoid:

$$Ae^{i(2\pi kx+\phi)} = A\cos(2\pi kx+\phi)+i\sin(2\pi kx+\phi)$$

- as *k* increases, so does the frequency of the oscillation



- note: the higher *k*, the higher the signal resolution, that is, one can represent smaller signal details (signals that vary more quickly)

Signal Synthesis with Sinusoids

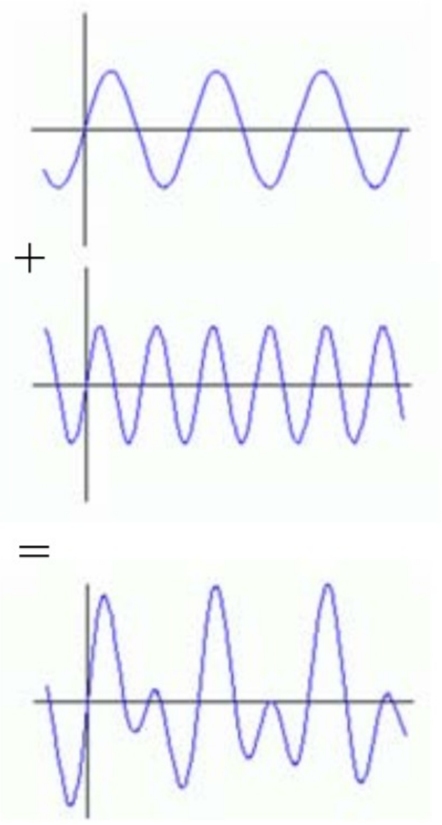
Any periodic signal can be created by a combination of weighted and shifted sinusoids at different frequencies

$$s_o(x) = \int_{-\infty}^{+\infty} A_k \cos(2\pi kx + \phi_k) + i\sin(2\pi kx + \phi_k)dk$$

$$= \int_{-\infty}^{+\infty} A_k e^{i(2\pi kx+\phi_k)} dk = \int_{-\infty}^{+\infty} A_k e^{i\phi_k} e^{i2\pi kx} dk$$

$$= \int_{-\infty}^{+\infty} S_i(k) e^{2\pi ikx} dk$$

- A_k is the amplitude and φ_k is the phase shift



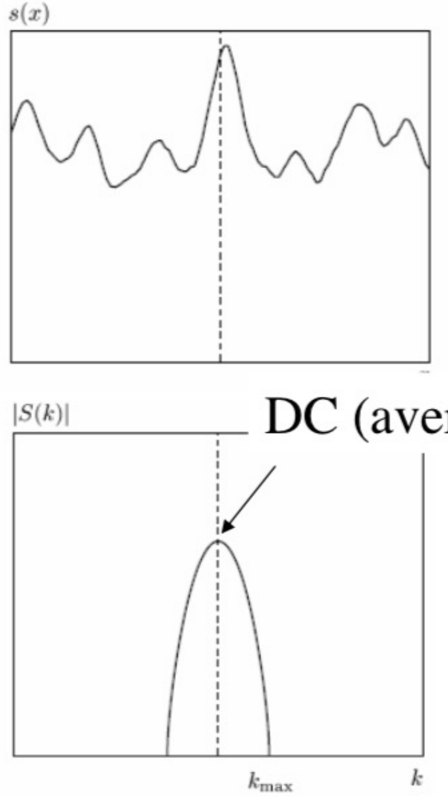
Incorporating the transfer function, now one for each k:

$$s_0(x) = \int_{-\infty}^{+\infty} S_i(k) e^{2\pi ikx} H(k) dk$$

Introduction

Theory developed by Joseph Fourier (1768-1830)

The Fourier transform of a signal s(x) yields its *frequency spectrum S*(*k*)



DC (average) term

$s(x)$

$S(k)$

forward transform

$$S(k) = F\{s(x)\} = \int_{-\infty}^{+\infty} s(x) e^{-2\pi ikx} dx$$

inverse transform

$$s(x) = F^{-1}\{S(k)\} = \int_{-\infty}^{+\infty} S(k) e^{2\pi ikx} dk$$

Extension to Higher Dimensions

The Fourier transform generalizes to higher dimensions

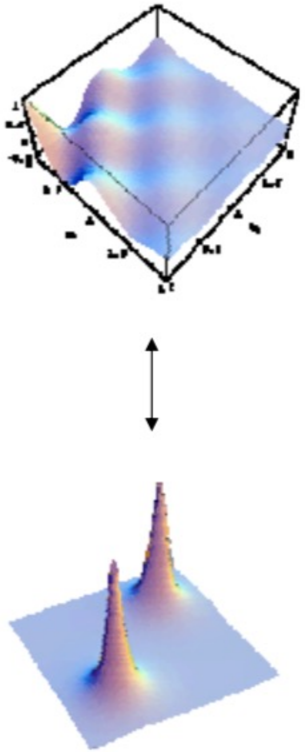
Consider the 2D case:

forward transform

$$S(k,l) = F\{s(x,y)\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} s(x,y)e^{-2\pi i(kx+ly)}dxdy$$

inverse transform

$$s(x,y) = F^{-1}\{S(k,l)\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} S(k,l)e^{2\pi i(kx+ly)}dkdl$$

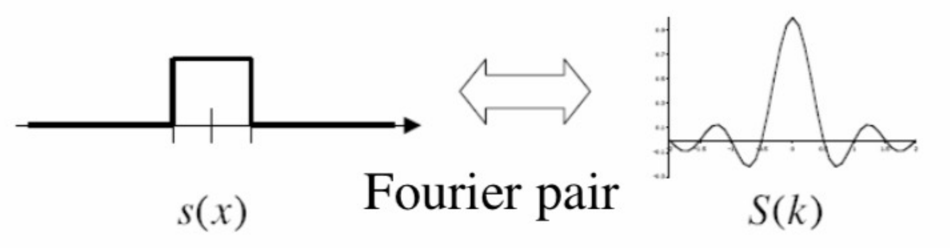


Calculation: Rect Function

$$S(k) = F\{A\Pi(\frac{x}{2L})\} = \int_{-\infty}^{+\infty} A\Pi(\frac{x}{2L})e^{-i2\pi kx}dx = \int_{-L}^{+L} Ae^{-i2\pi kx}dx$$

$$= -\frac{A}{2\pi ki}(e^{-i2\pi kL} - e^{i2\pi kL}) = \frac{A}{2\pi k}2\sin(2\pi kL)$$

$$= 2AL\ \mathrm{sinc}(2\pi kL)$$

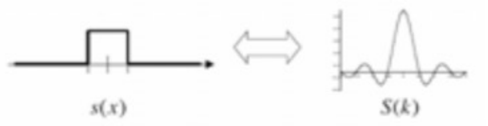


We see that a finite signal in the *x*-domain creates an infinite signal in the *k*-domain (the frequency domain)

- the same is true vice versa

Properties

Scaling:



$$S(k) = 2AL\ \mathrm{sinc}(2\pi kL)$$

- consider the rect (box): the greater L…
  … the higher the spectrum (factor AL)
  … the narrower the spectrum (factor L)
- the scaling rule is therefore:

$$F\{s(ax)\} = \frac{1}{|a|} S(\frac{k}{a})$$

a>1 shrinks s
a<1 stretches s

Symmetry: $F\{S(x)\} = s(-k)$

Linearity: $F\{as_1(x) + bs_2(x)\} = F\{as_1(x)\} + F\{bs_2(x)\}$

Translation: $F\{s(x - x_0)\} = S(k)e^{-2\pi i x_0 k}$

phase shift

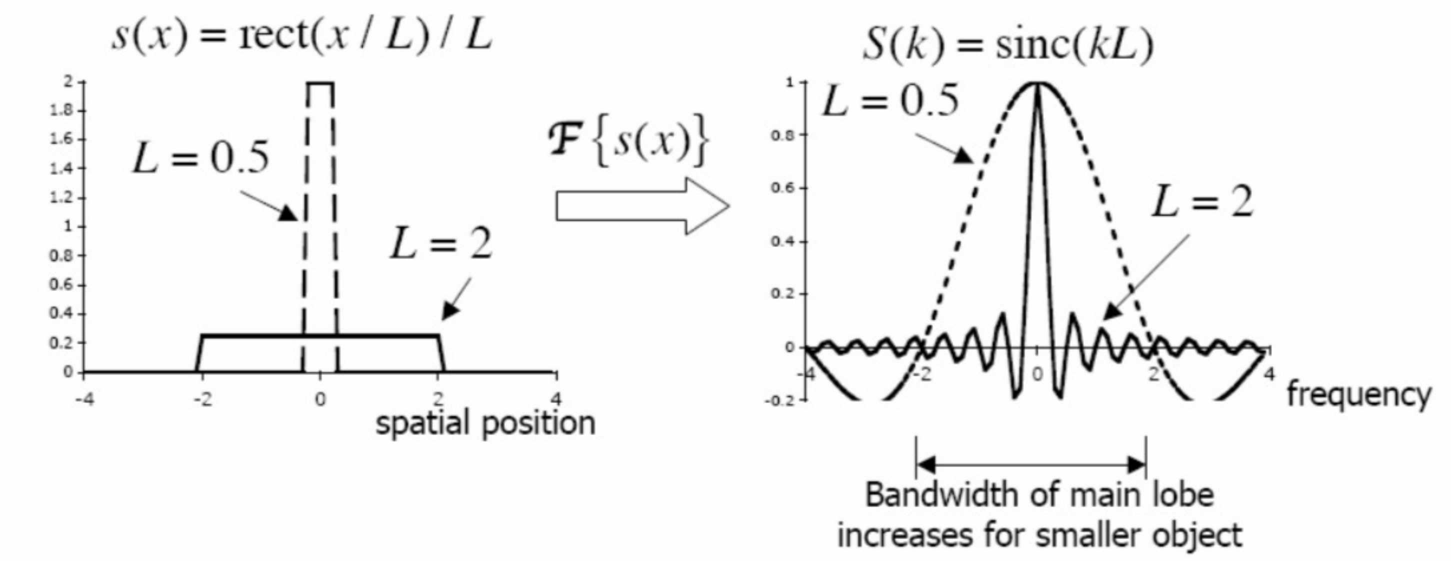
Convolution: $F\{s_1(x) * s_2(x)\} = S_1(k) \cdot S_2(k)$

$F\{s_1(x) \cdot s_2(x)\} = S_1(k) * S_2(k)$

Scaling Property

The rect function provides good insight into the relationship of fine detail and frequency bandwidth

- a thin rect can represent/resolve fine detail (think of a signal being represented as an array of thin rects
- a thin rect gives rise to a wide frequency lobe
- this illustrates that signals with more detail will have broader frequency spectra
- or, in turn, signals with thin frequency spectra will have low spatial resolution



Influence of Transfer Function H

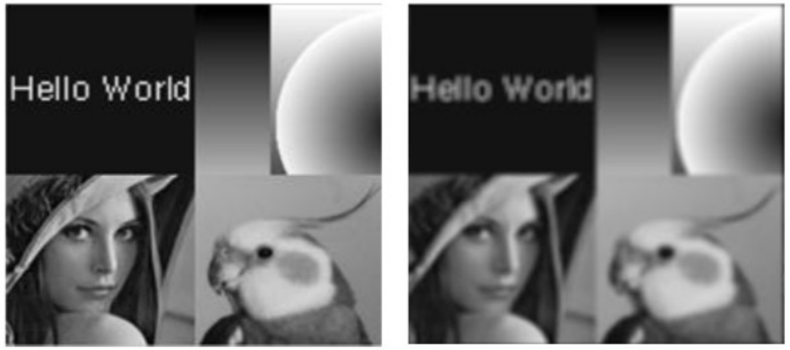
We know (from the last lecture) that:

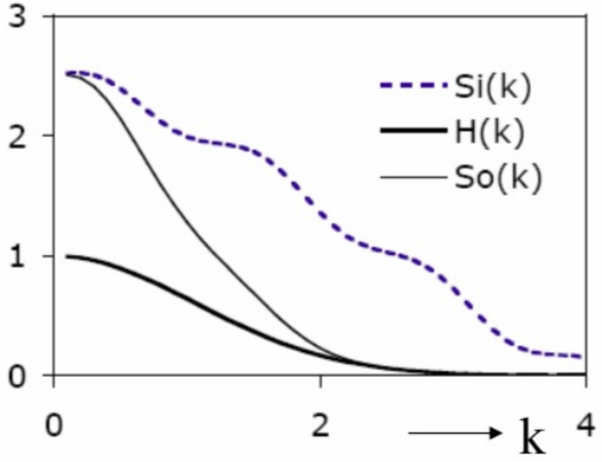
$$s_0(x) = \int_{-\infty}^{+\infty} S_i(k) e^{2\pi ikx} H(k) dk$$

$$s_o(x) = s_i(x) * h(x) \leftrightarrow S_i(k) \cdot H(k) = S_o(k)$$

Let's look at a concrete example:

- *H* is a *lowpass (blurring) filter*: it reduces the higher frequencies of *S* more than the lower ones



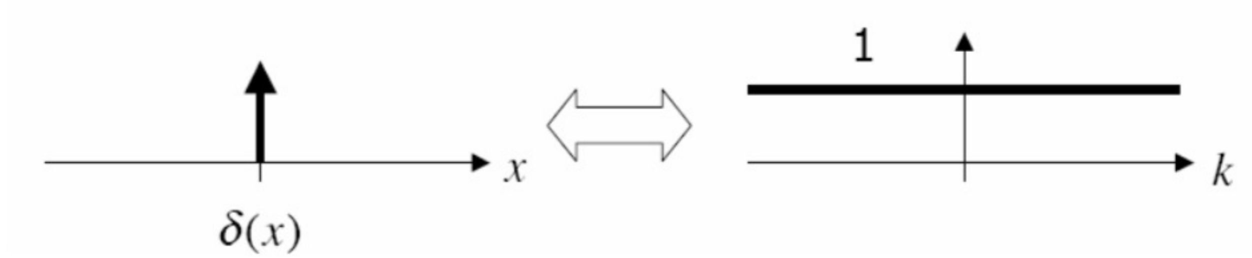


after application of *H*

Calculation: Dirac Impulse

For s(x)=δ(x):

$$S(k) = F\{\delta(x)\} = \int_{-\infty}^{+\infty} \delta(x) e^{-i2\pi kx} dx = e^{-i2\pi k0} = 1$$



Recall that the Dirac is an extremely thin rect function

- the frequency spectrum is therefore extremely broad (1 everywhere)

This illustrates a key feature of the Fourier Transform:

- the narrower the $s(x)$, the wider the $S(k)$
- sharp objects need higher frequencies to represent that sharpness

Important Fourier Pairs: Sinusoids

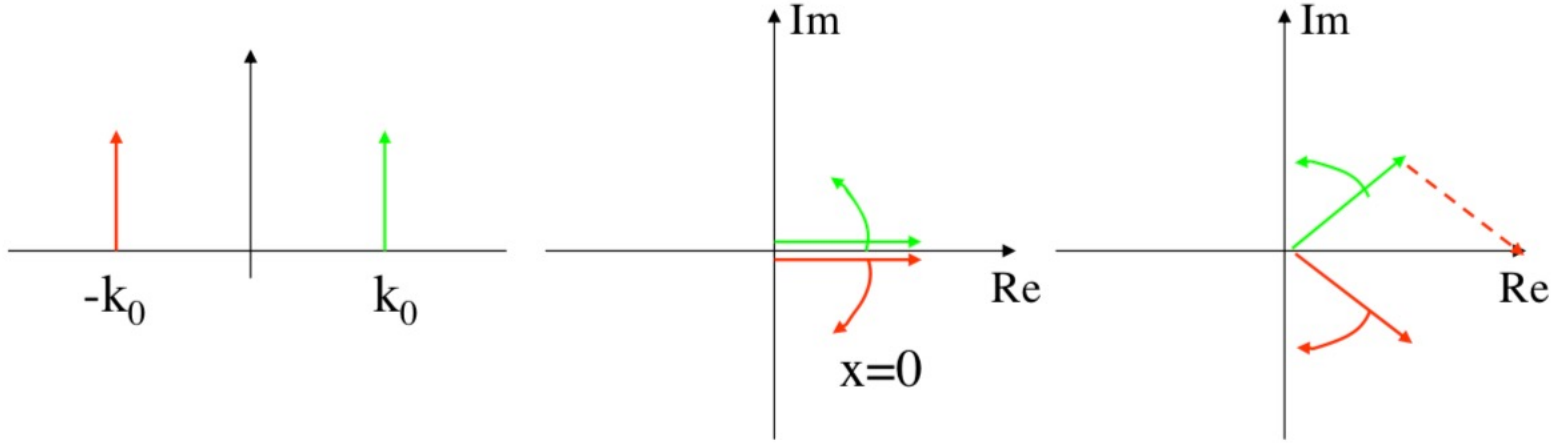
Sinusoids of frequency k_0 give rise to two spikes in the frequency domain at $\pm k_0$

$$\cos(2\pi k_0 x) \leftrightarrow (\delta(k+k_0)+\delta(k-k_0))/2$$

$$\sin(2\pi k_0 x) \leftrightarrow i(\delta(k+k_0)-\delta(k-k_0))/2$$

Recall the pointer analogon in the complex plane

for the cos(): the real signal is given by the addition of the two vectors (divided by 2), projected onto the real axis



Important Fourier Pairs: Sinusoids

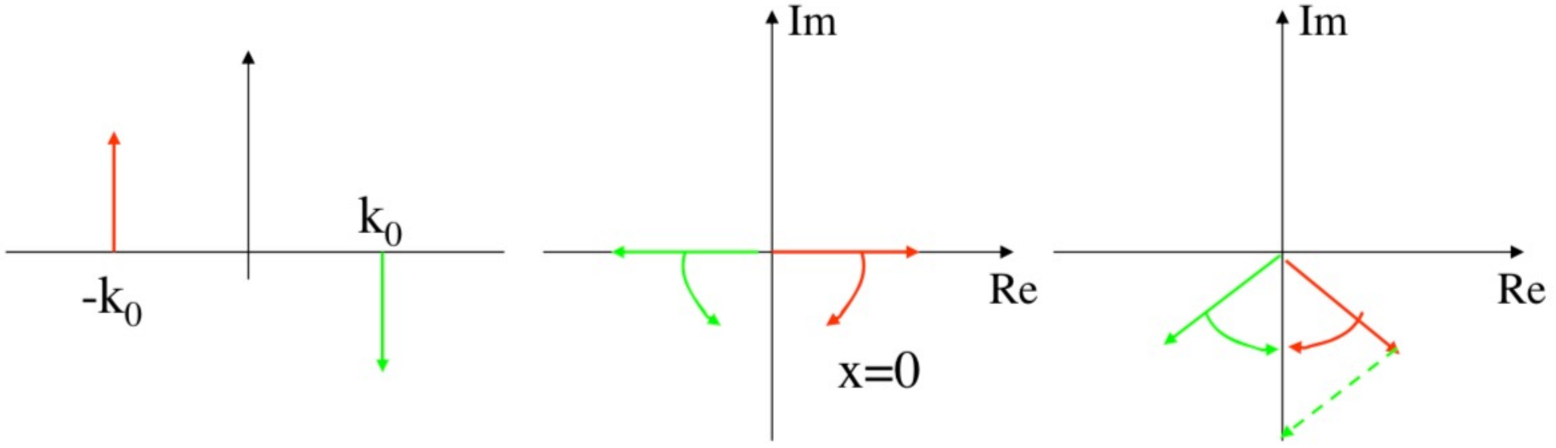
Sinusoids of frequency k_0 give rise to two spikes in the frequency domain at $\pm k_0$

$$\cos(2\pi k_0 x) \leftrightarrow (\delta(k+k_0)+\delta(k-k_0))/2$$

$$\sin(2\pi k_0 x) \leftrightarrow i(\delta(k+k_0)-\delta(k-k_0))/2$$

Recall the pointer analogon in the complex plane

for the sin(): the real signal is given by the addition of the two vectors (divided by 2), projected onto the imaginary axis (note the i in the equation)



More Important Fourier Pairs

$$\delta(x) \leftrightarrow 1$$

$$1 \leftrightarrow \delta(k)$$

$$\cos(2\pi k_0 x) \leftrightarrow (\delta(k+k_0) + \delta(k-k_0))/2$$

$$\sin(2\pi k_0 x) \leftrightarrow i(\delta(k+k_0) - \delta(k-k_0))/2$$

$$\Pi(\frac{x}{2L}) \leftrightarrow 2L\mathrm{sinc}(2\pi Lk)$$

$$\Lambda(\frac{x}{2L}) \leftrightarrow L\mathrm{sinc}^2(\pi Lk)$$

$$e^{\frac{-x^2}{2\sigma^2}} \leftrightarrow e^{-2\sigma^2 k^2}$$

the Gaussian width is
inversely related

Some Notes

In the 2D transform, if *f*(*x*,*y*) is separable, that is, f(*x*,*y*)=*f*(*x*)*f*(*y*), one may write:

$$S(k,l) = F\{s(x,y)\} = \int_{-\infty}^{+\infty} s(y)e^{-2\pi ily}(\int_{-\infty}^{-\infty} s(x)e^{-2\pi ikx}dx)dy$$

$$s(x,y) = F^{-1}\{S(k,l)\} = \int_{-\infty}^{+\infty} S(l)e^{-2\pi ily}(\int_{-\infty}^{-\infty} s(k)e^{-2\pi ikx}dk)dl$$

- this comes in handy sometimes

Some Notes

Sometimes the factor $2\pi k$ is used as ω:

$$s_0(x) = \int_{-\infty}^{+\infty} S_i(\omega) e^{i\omega x} H(\omega) d\omega$$

So far, we have only discussed the continuous space with (potentially) infinite spectra and signals

- that is where it makes sense to use ω
- but in reality we deal with finite, discrete signals (here *k* matters)
- we shall discuss this next

Fourier Transform of Discrete Signals: DTFT

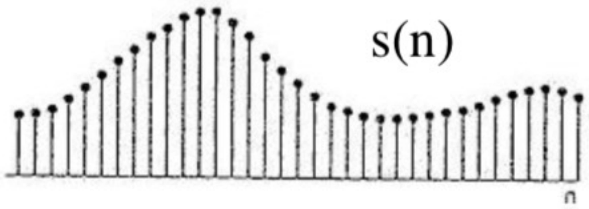
Discrete-Time Fourier Transform (DTFT)

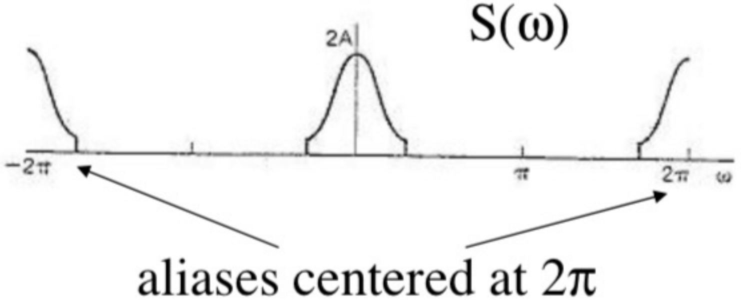
- assumes that the signal is discrete, but infinite

$$S(\omega) = \sum_{n=-\infty}^{+\infty} s(n) e^{-i\omega n}$$

$$s(n) = \int_{-\pi}^{+\pi} S(\omega) e^{i\omega n}$$

- the frequency spectrum is <u>continuous</u>, but is <u>periodic</u> (has *aliases*)





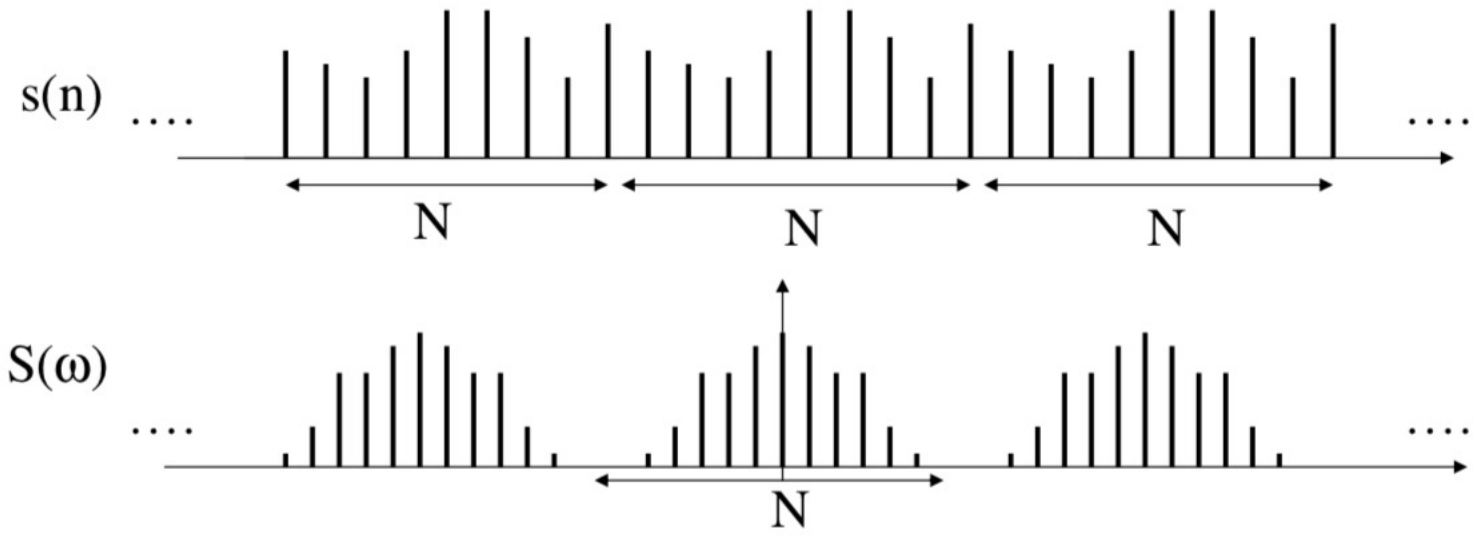
Fourier Transform of Discrete Signals: DFT

Discrete Fourier Transform (DFT)

- assumes that the signal is discrete and finite

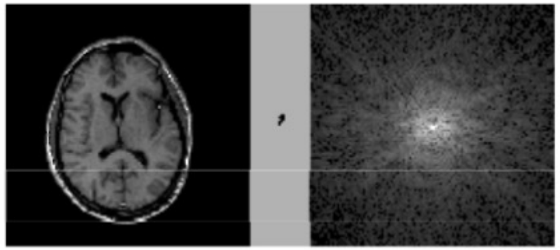
$$S(k) = \sum_{n=0}^{N-1} s(n) e^{\frac{-i2\pi kn}{N}} \qquad s(n) = \frac{1}{N}\sum_{n=0}^{N-1} S(k) e^{\frac{i2\pi kn}{N}}$$

- now we have only N samples, and we can calculate N frequencies
- the frequency spectrum is now <u>discrete</u>, and it is <u>periodic</u> in N



Fourier Transform in Higher Dimensions

The 2D transform:



$$S(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} s(n,m)e^{\frac{-i2\pi(kn+lm)}{NM}}$$

$$s(n,m) = \frac{1}{NM}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} S(k,l)e^{\frac{i2\pi(kn+lm)}{NM}}$$

Separability:

$$S(k,l) = \frac{1}{NM}\sum_{m=0}^{M-1} e^{\frac{-i2\pi lm}{M}} P(k,m) \quad \text{where } P(k,m) = \sum_{n=0}^{N-1} s(n,m)e^{\frac{-i2\pi kn}{N}}$$

$$s(n,m) = \frac{1}{NM}\sum_{l=0}^{M-1} e^{\frac{-i2\pi lm}{M}} p(n,l) \quad \text{where } \mathrm{p}(n,l) = \sum_{k=0}^{N-1} S(n,m)e^{\frac{-i2\pi kn}{N}}$$

- if M=N, complexity is $2\cdot O(2N^3)$

Fast Fourier Transform (1)

Recursively breaks up the FT sum into odd and even terms:

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{\frac{-i2\pi kn}{N}} = \sum_{n=0}^{N/2-1} s(2n) e^{\frac{-i2\pi k2n}{N}} + \sum_{n=0}^{N/2-1} s(2n+1) e^{\frac{-i2\pi k(2n+1)}{N}}$$

$$= \sum_{n=0}^{N/2-1} s_{even}(n) e^{\frac{-i2\pi kn}{N/2}} + e^{\frac{-i2\pi k}{N}} \sum_{n=0}^{N/2-1} s_{odd}(n) e^{\frac{-i2\pi kn}{N/2}}$$

Results in an O(n·log(n)) algorithm (in 1D)

- $O(n^2 \cdot \log(n))$ for 2D (and so on)

Fast Fourier Transform (1)

Gives rise to the well-known butterfly architecture: