Introduction to Computer Vision
Image processing

Dr. Sergio Escalera
University of Barcelona

UNIVERSITAT DE BARCELONA

Low-level processes involve primitive operations such as image preprocessing to reduce noise, contrast enhancement, and image sharpening. ***Input and outputs are images***

Mid-level processes involve tasks such as segmentation (partitioning an image into regions), description of these regions and classification or recognition of those. ***Input is an image, but outputs are attributes*** extracted from those images (e.g. edges, contours or identity)

High-level processes involves **"making sense"** of a collection of these objects (e.g. scene understanding).

Image processing, image analysis and computer vision

There are no clear boundaries among these concepts, but we may consider a three level paradigm:

Image Processing

Low-level processes involve primitive operations such as image preprocessing to reduce noise, contrast enhancement, and image sharpening. ***Input and outputs are images***

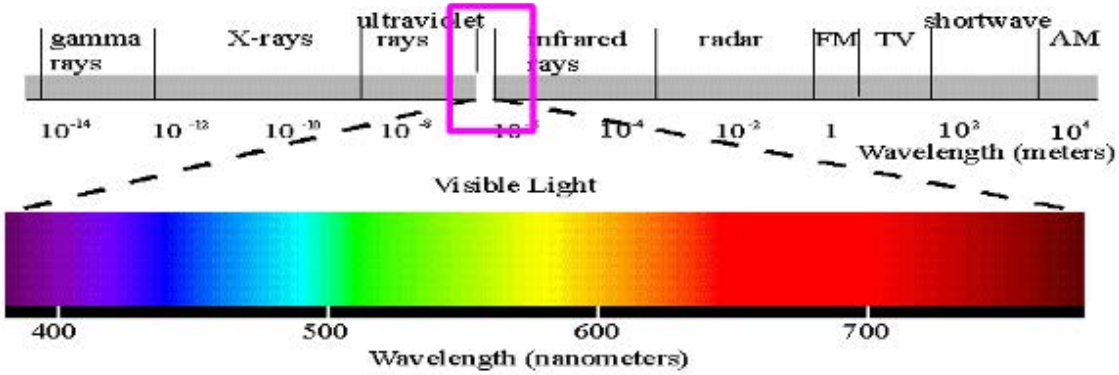
Mid-level processes involve tasks such as segmentation (partitioning an image into regions), description of these regions and classification or recognition of those. ***Input is an image, but outputs are attributes*** extracted from those images (e.g. edges, contours or identity)

High-level processes involves **"making sense"** of a collection of these objects (e.g. scene understanding).

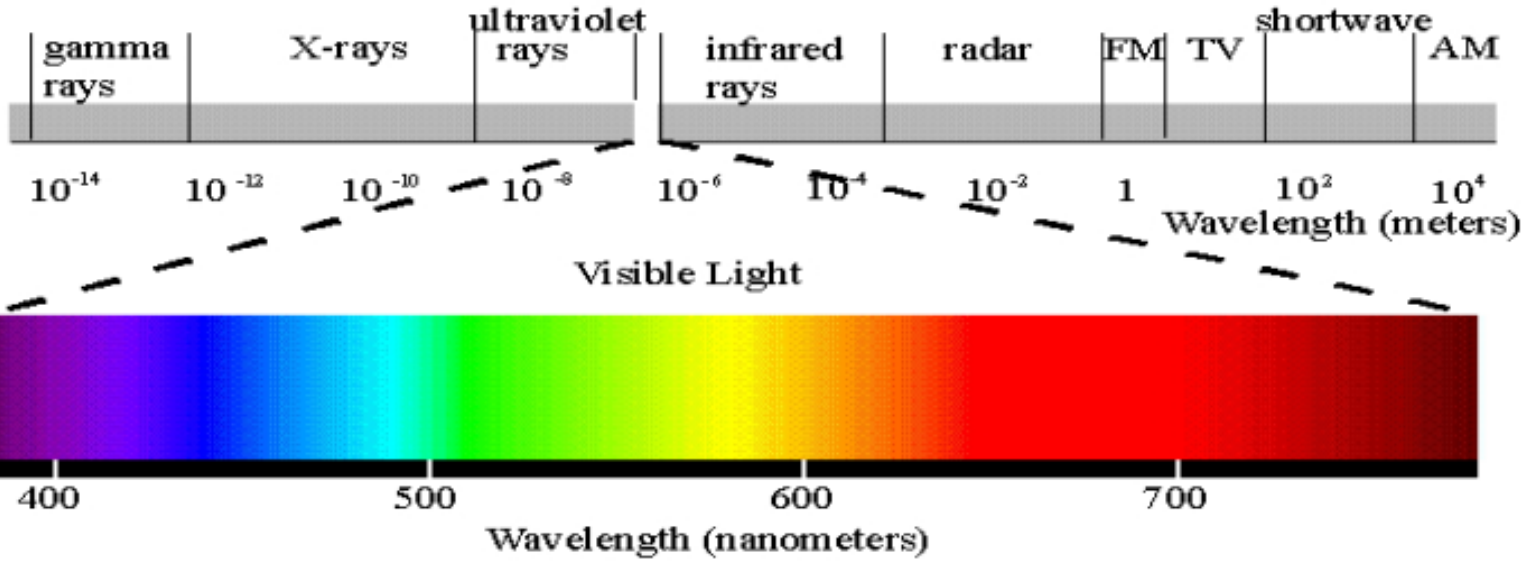
Computer Vision

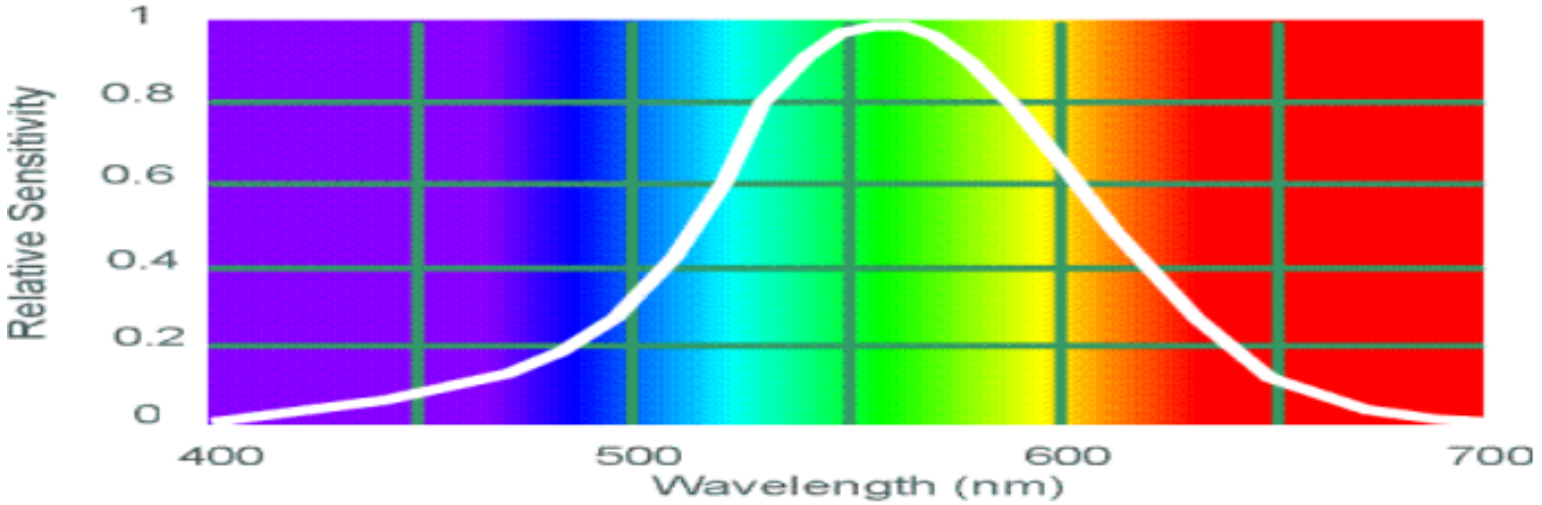
ELECTROMAGNETIC SPECTRUM

Images, images and … images !!!



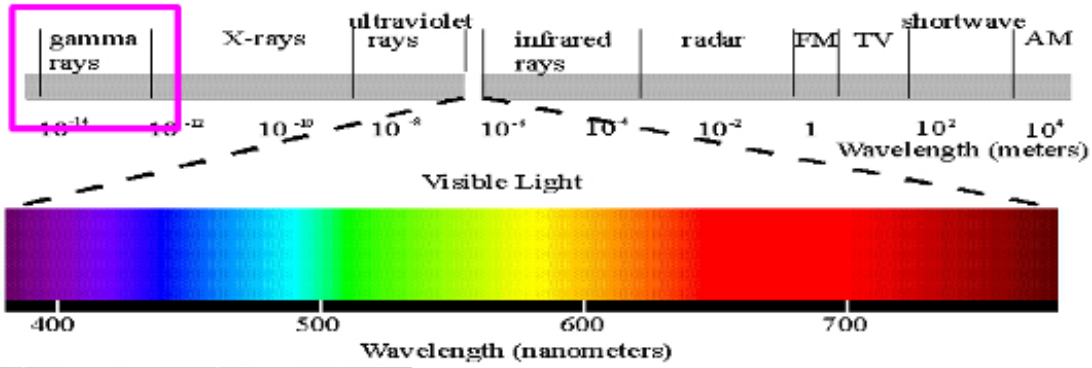
ELECTROMAGNETIC SPECTRUM

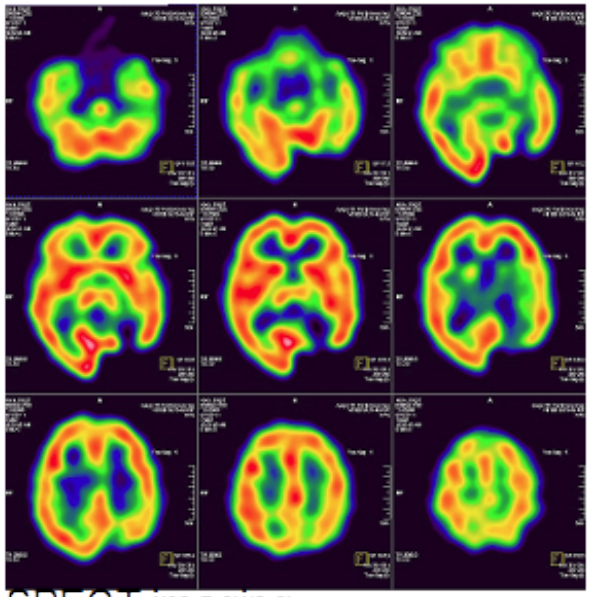




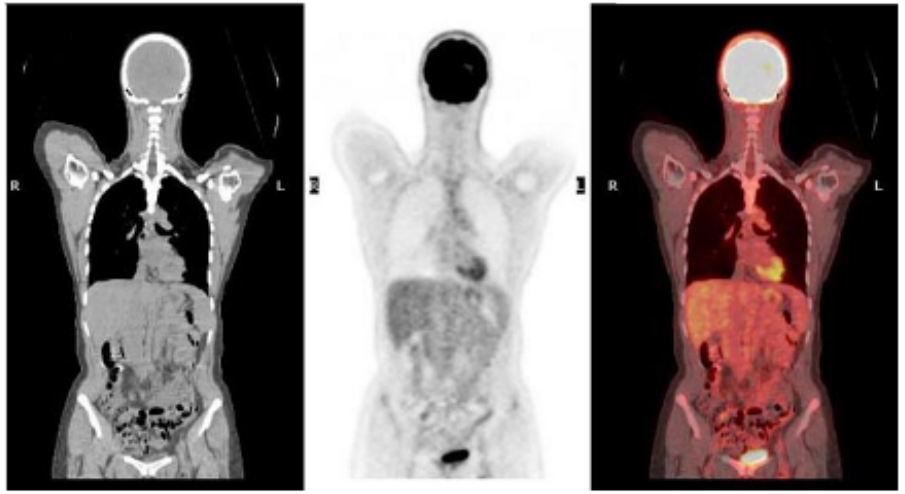
Human Luminance Sensitivity Function

Images, images and … images !!!





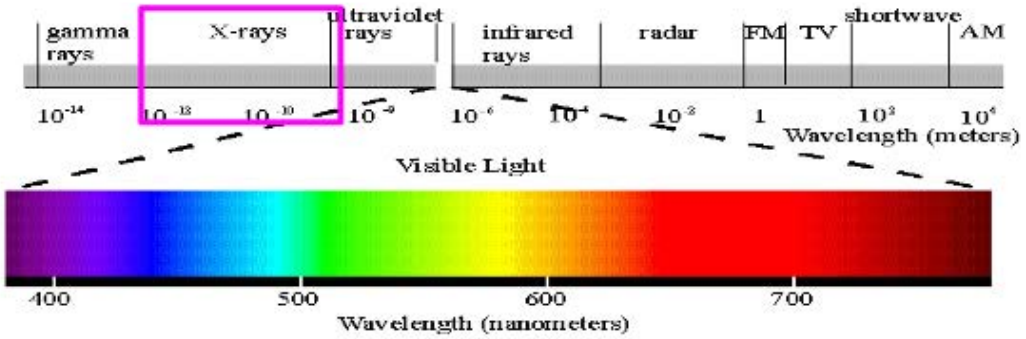
SPECT imaging
(single-photon emission computed tomography)



Gamma PET (Positront emision tomography)

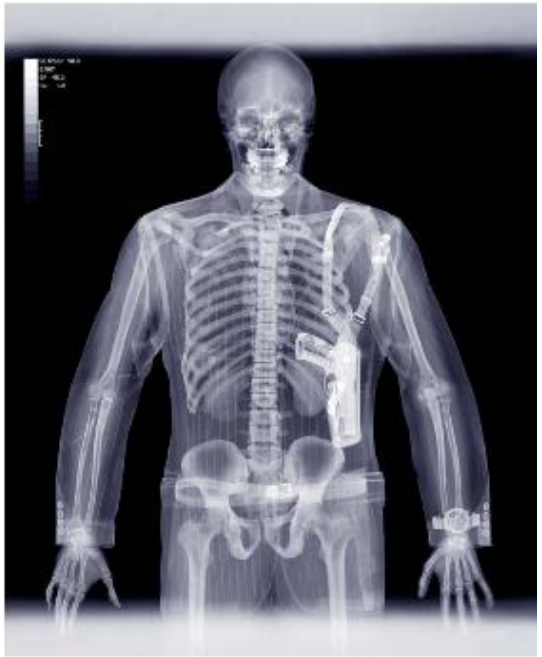
Associated to functionalities of the scanned areas. A gamma-ray emiting radiotracer is injected and monitored using a gamma camera.

Images, images and … images !!!

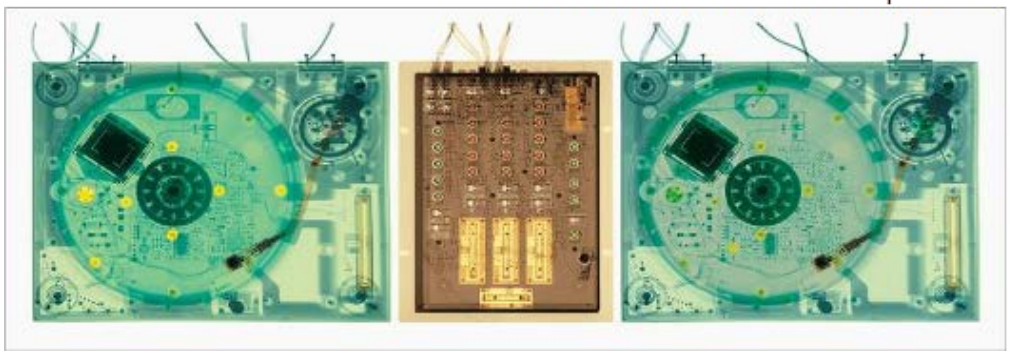


Airport Scanning

Traumatology

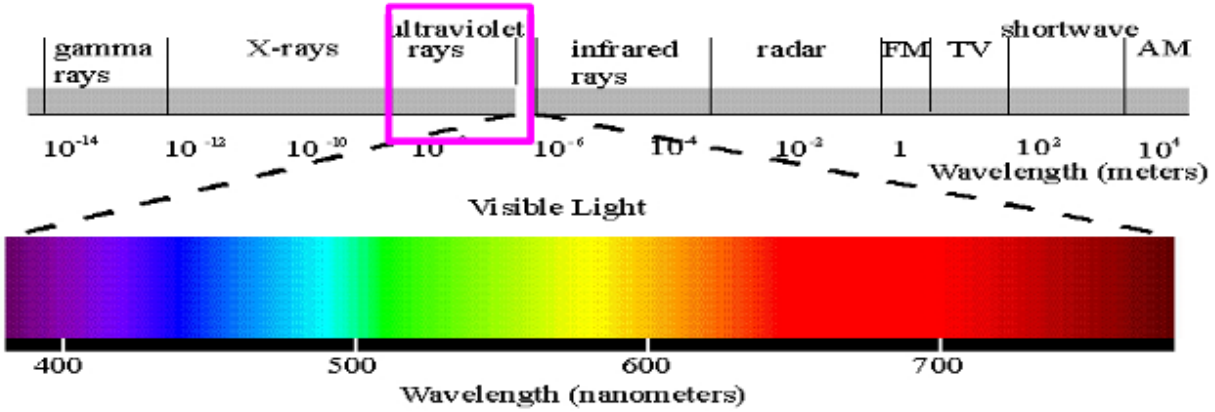


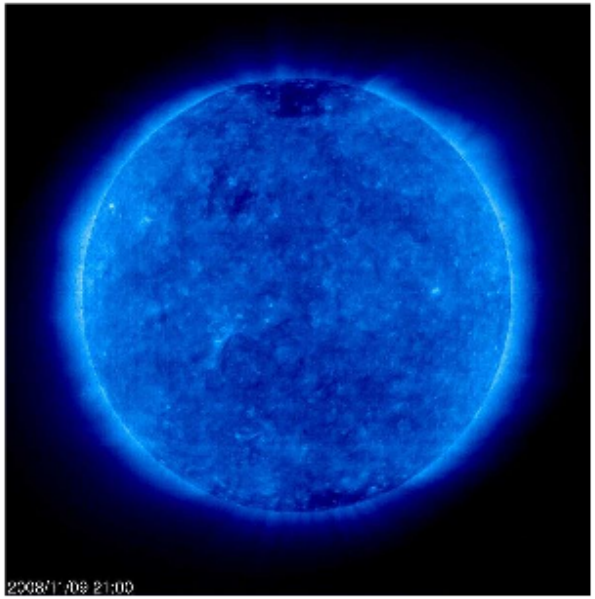
Fault detection in HW pieces

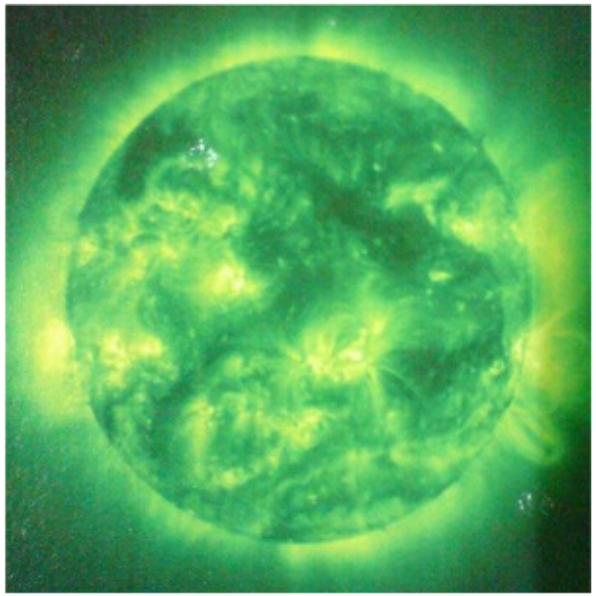


Electromagnetic waves

Images, images and … images !!!

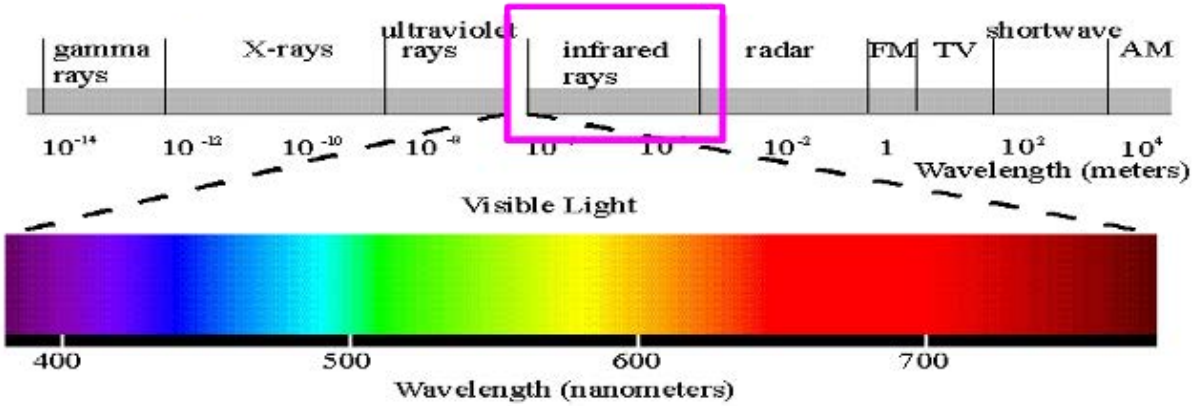


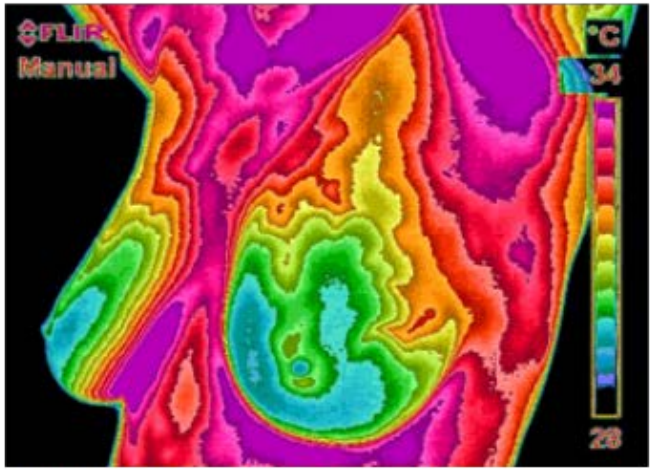




Solar flare studies
Extreme Ultraviolet Imaging Telescope (EIT)

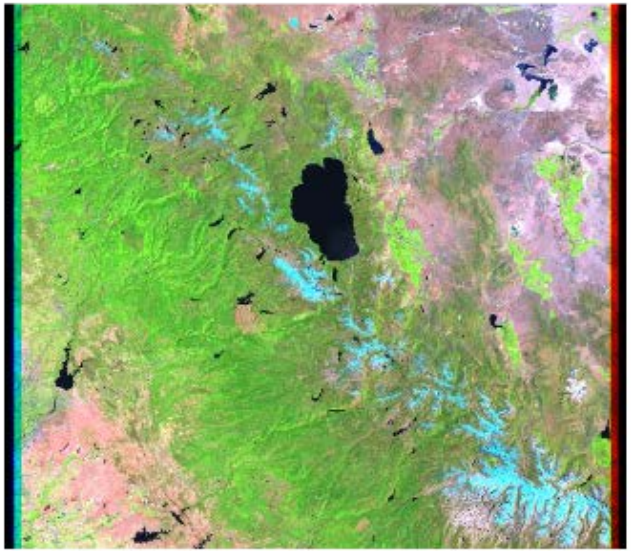
Images, images and … images !!!





Heat due to blood flow
for cancer detection

Landsat projects

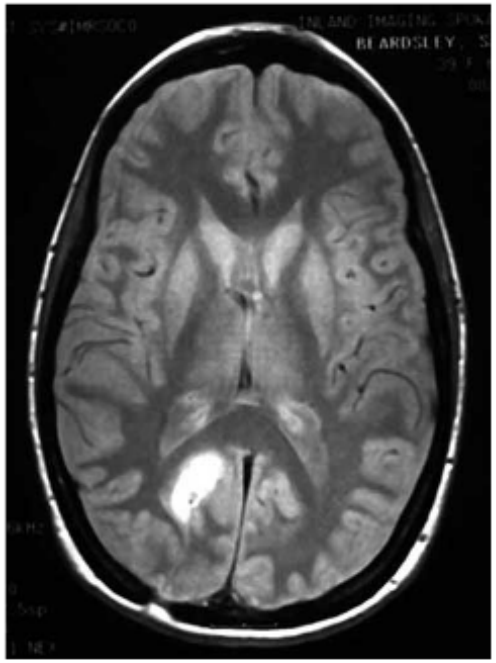


Images, images and … more!!!

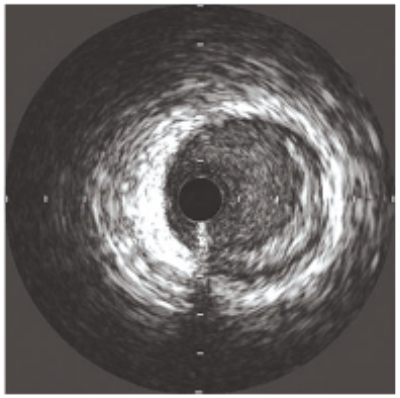
But not restricted to electromagnetic waves …

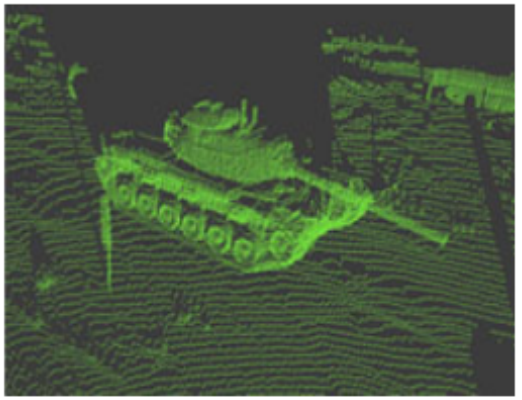
… in general any bidimensional scanning producing some magnitude value associated to a measure of a quantity of interest can be interpreted as an image.

MRI (magnetic resonance)



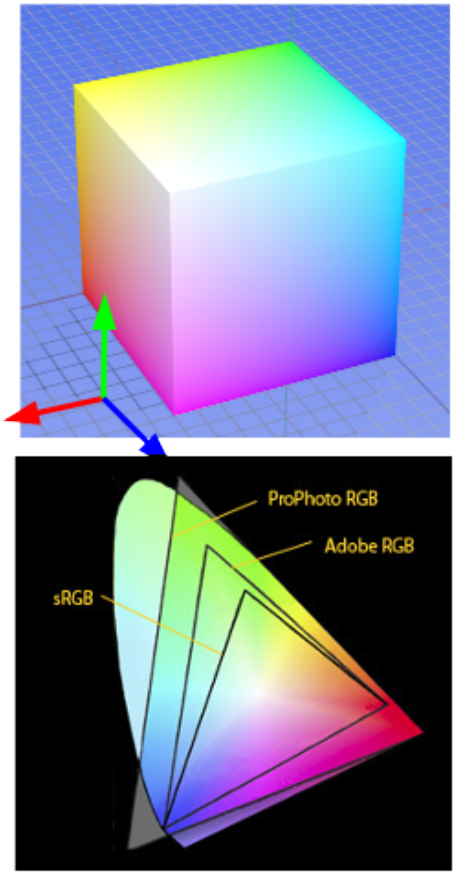
Ultrasounds (acoustic waves reflection)





LADAR (Laser depth)

COLOR SPACES: sRGB

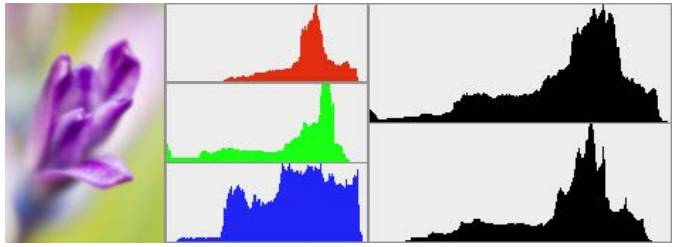


sRGB Gamut

R
(G=0,B=0)

G
(R=0,B=0)

B
(R=0,G=0)



Some drawbacks

- Strongly correlated channels
- Non-perceptual

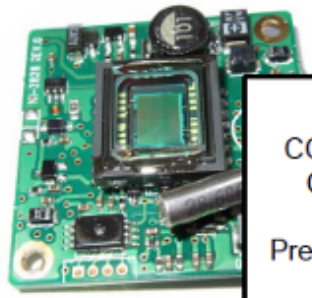
Image from: http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png

Modern image acquisition pipeline

Scene Radiance

Starting point:
reality (in radiance)

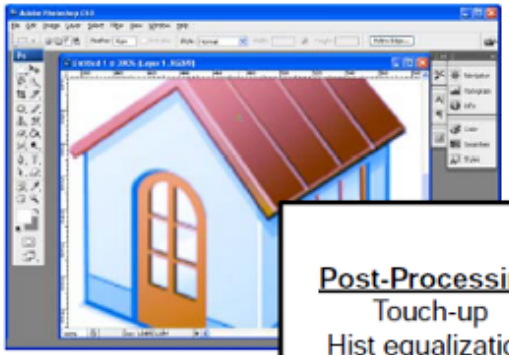
Pre-Camera
Lens Filter
Lens
Shutter
Aperture



In-Camera
CCD response (RAW)
CCD Demosaicing
Gamut Mapping
Preferred color selection
Tone-mapping

Camera Output: sRGB

Even if we stopped here, the original CCD response potentially has had many levels of processing.



Post-Processing
Touch-up
Hist equalization
Spatial warping
Etc . . .

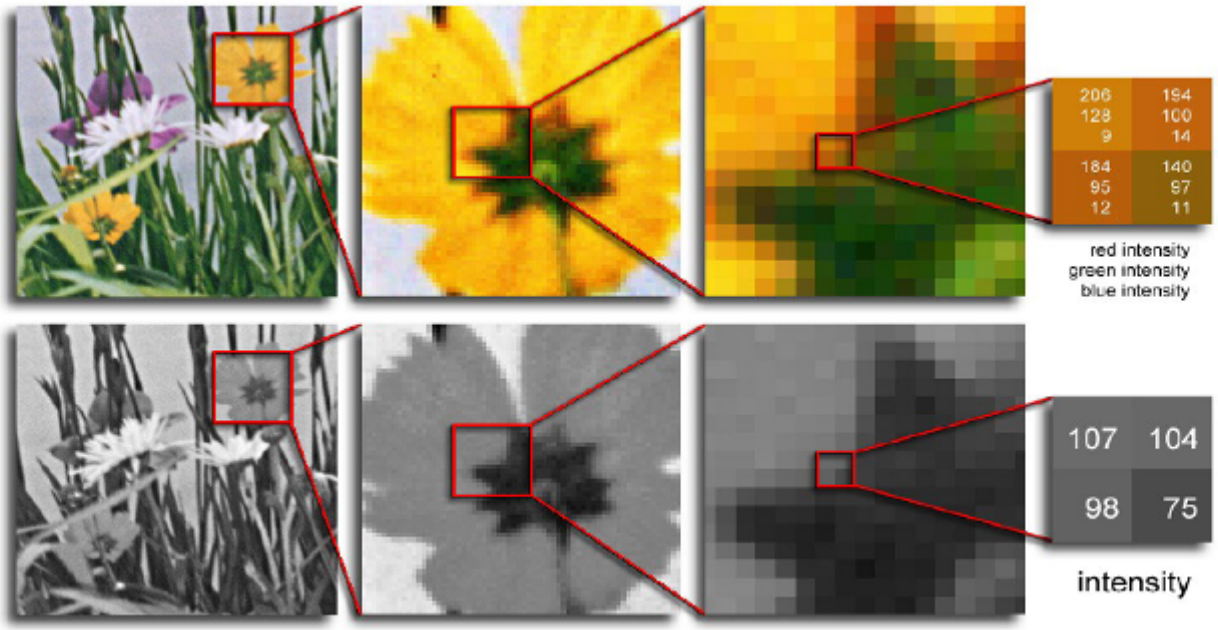
Final output

Ending point:
better than reality (in RGB)

Image representation

An image is defined as a 2-dimensional function ***f(x,y)*** where **x** and **y** are *spatial* coordinates, and the amplitude of ***f*** is called *intensity* or *gray level.*

When *x,y* and *f* are finite and discrete quantities, we call the image a digital image.



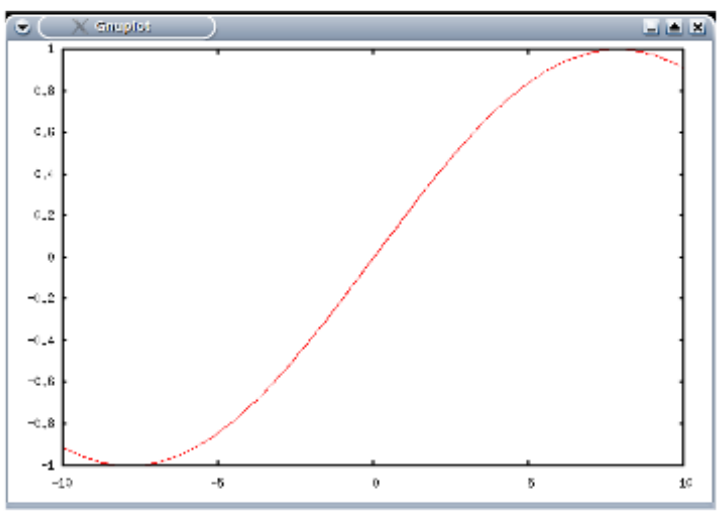
Digitalization

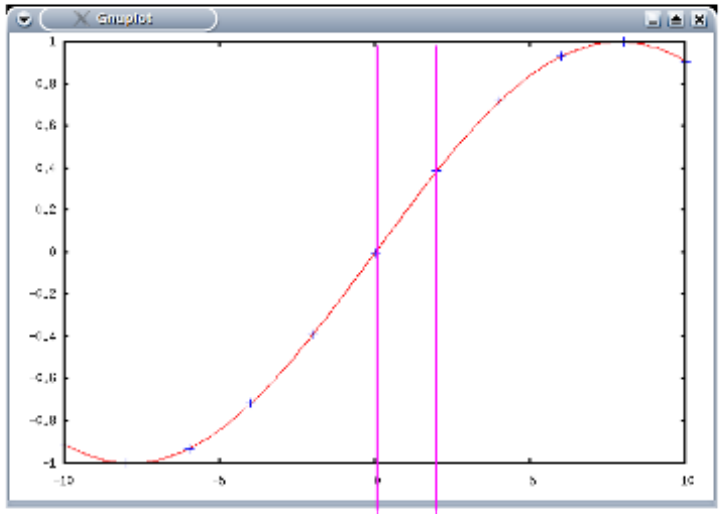
Real data is defined in a continuous domain. However, computers can only handle discrete sets of values.
The way to represent continuous data is to **dicretize and quantify.**

Discretization

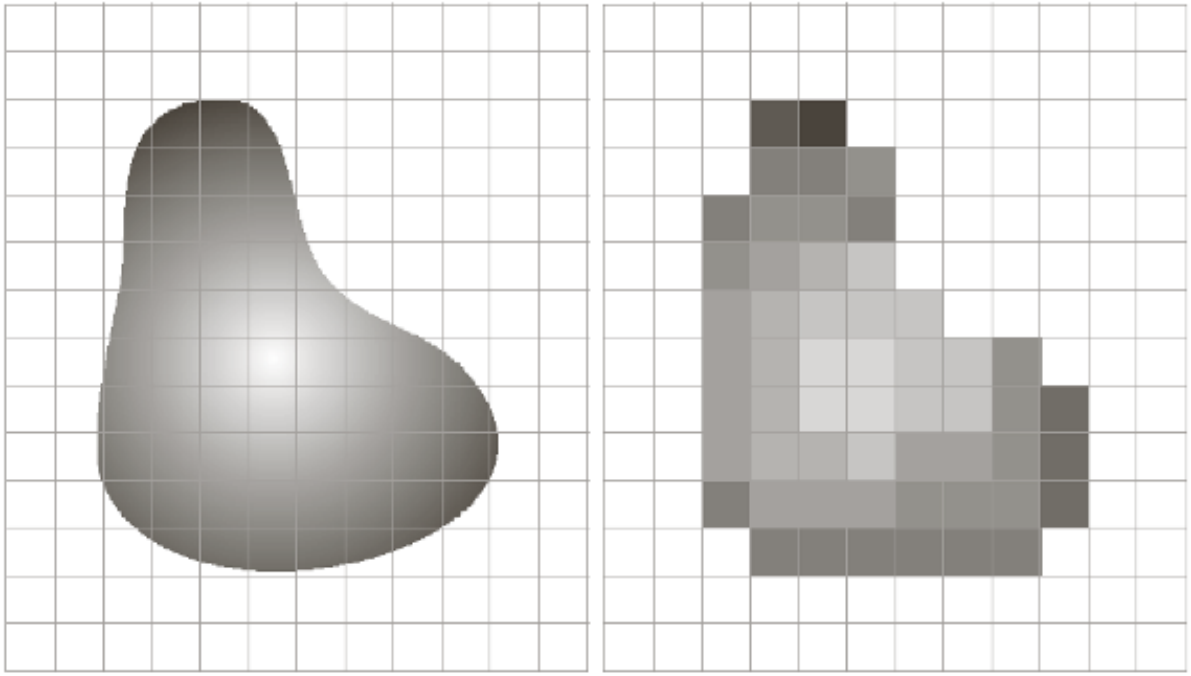
Associated to the concept of sampling and characterized by the sampling frequency or sampling period.

$$x(t) \rightarrow x(n \cdot T_s), \quad \forall n \in \{1 \dots T\}$$





2D sampling and quantization



a b

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

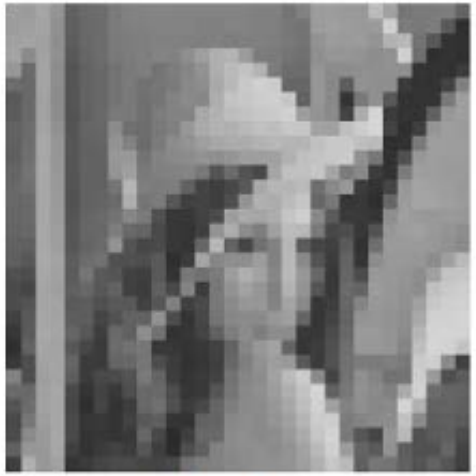
The spatial sampling is related to image resolution (e.g. 640x480).
And quantization is related to intensity or color depth (the number of bits per pixel).

Resolution

512x512

64x64

128x128



32x32

Quantization

The number of gray levels associated to *b* bits is

$$L = 2^b$$

64 levels

16 levels

4 levels

2 levels

Brightness

It is a measure of the luminance (global light intensity value) of the image

An estimation of the brightness value: $$B = \frac{1}{N} \sum_{x,y} f(x,y)$$

Contrast

Measure of the perceived intensity difference between two regions.

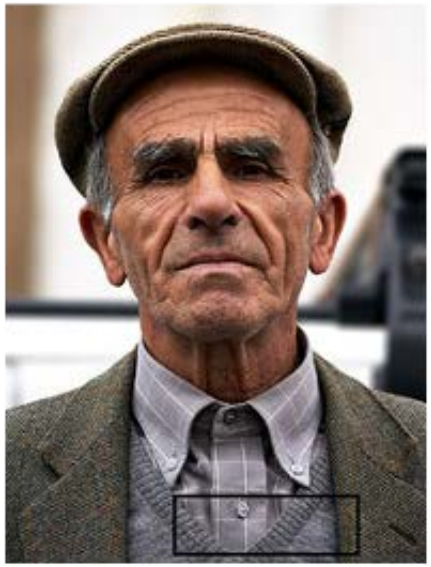
A simple definition of contrast:

$$C = \frac{|f_a - f_b|}{f_a + f_b}$$

Sharpness and Acutance

Acutance is the edge contrast of the image

Sharpness describes the clarity of detail in an image and depends on resolution and acutance



Low res & High ac

High res & Low ac

High res & High ac

Noise

Different types of noise usually categorized by their model:
Additive, multiplicative, impulsive, ...

Examples:

Additive White Gaussian Noise

$$f(x,y) = I(x,y) + n, \quad n \in \mathcal{N}(0,\sigma)$$

Salt and Pepper Noise (Random impulses of extreme values)

Gaussian noise with different sigma

Salt and Pepper

Introduction

The term spatial domain refers to the plane of the image.

Spatial methods manipulate pixels themselves.

Basic operation on stand-alone pixels

$$g(x,y) = T[f(x,y)]$$

Operations on pixels based on general image information

$$g(x,y) = T[f(x,y), \theta]$$

Operations taking into account the neighborhood

$$g(x,y) = T[f(x,y), \mathcal{N}(x,y)]$$

INTUITIVE MECHANIC OF SPATIAL FILTERING

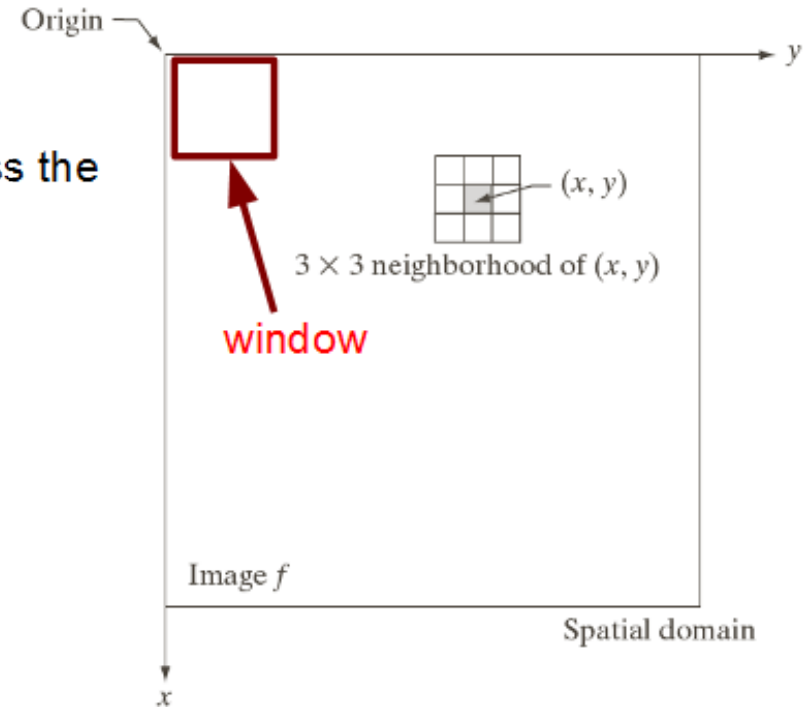
We borrow the name ***filtering*** from the concepts of frequency filtering, where filtering refers to accepting (passing) or rejecting certain frequency components.

So, what must we do if we want to process a pixel according to its neighborhood?

Lay a window around the pixel of interest and operate with those values.

And what do we do if we want to process **the** full image?

Repeat this process for each pixel



INTUITIVE MECHANIC OF SPATIAL FILTERING

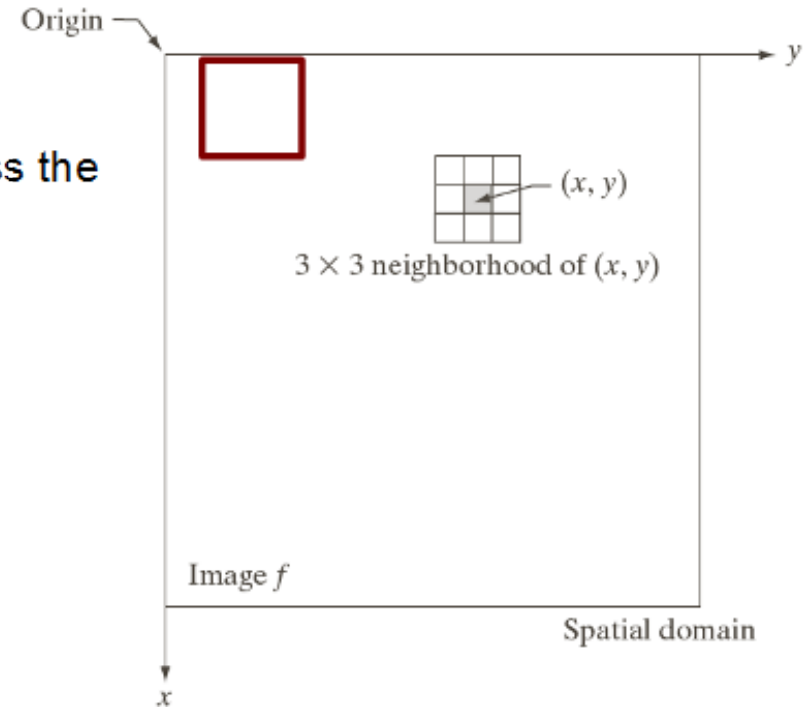
We borrow the name ***filtering*** from the concepts of frequency filtering, where filtering refers to accepting (passing) or rejecting certain frequency components.

So, what must we do if we want to process a pixel according to its neighborhood?

Lay a window around the pixel of interest and operate with those values.

And what do we do if we want to process **the** full image?

Repeat this process for each pixel



INTUITIVE MECHANIC OF SPATIAL FILTERING

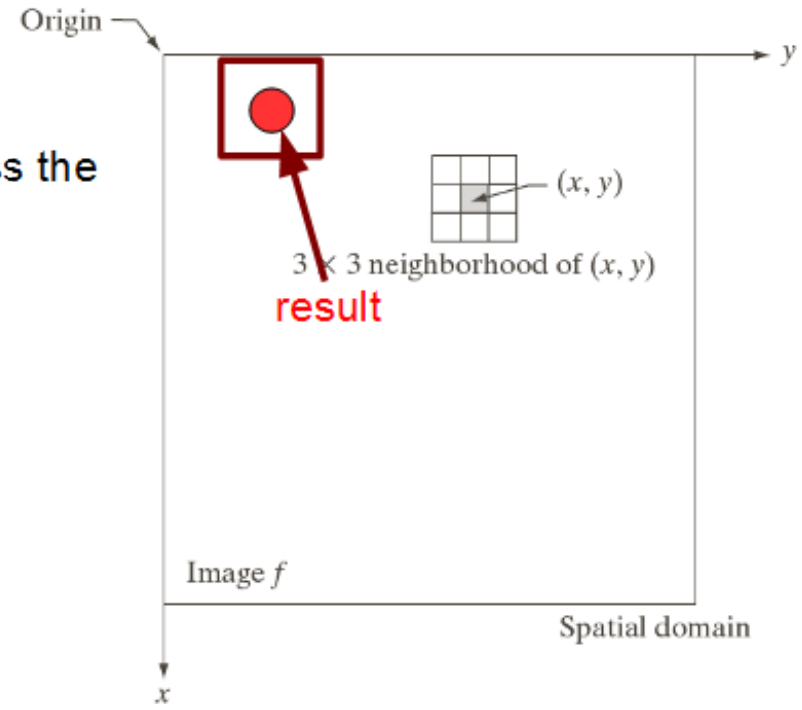
We borrow the name ***filtering*** from the concepts of frequency filtering, where filtering refers to accepting (passing) or rejecting certain frequency components.

So, what must we do if we want to process a pixel according to its neighborhood?

Lay a window around the pixel of interest and operate with those values.

And what do we do if we want to process **the** full image?

Repeat this process for each pixel



What for?

Enhance images
- Denoise
- Resize
- Increase contrast

Extract features
- Texture
- Edges
- Salient points

Detect patterns
- Template matching

A SIMPLE EXAMPLE

We want to smooth an image. For this task we may use the mean value in a 3x3 window.

f(x-1,y-1)	f(x-1,y)	f(x-1,y+1)
f(x,y-1)	f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

$$\frac{1}{9}(f(x-1,y-1)+f(x-1,y)+f(x-1,y+1)+\ldots$$
$$+f(x,y-1)+f(x,y)+f(x,y+1)+\ldots$$
$$+f(x+1,y-1)+f(x+1,y)+f(x+1,y+1))$$

A SIMPLE EXAMPLE

We want to smooth an image. For this task we may use the mean value in a 3x3 window.

f(x-1,y-1)	f(x-1,y)	f(x-1,y+1)
f(x,y-1)	f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

OBSERVE THAT THE SAME RESULT CAN BE OBTAINED USING A WINDOW WITH WEIGHTS AND ADDING THE ELEMENT MULTIPLICATION

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

MASK

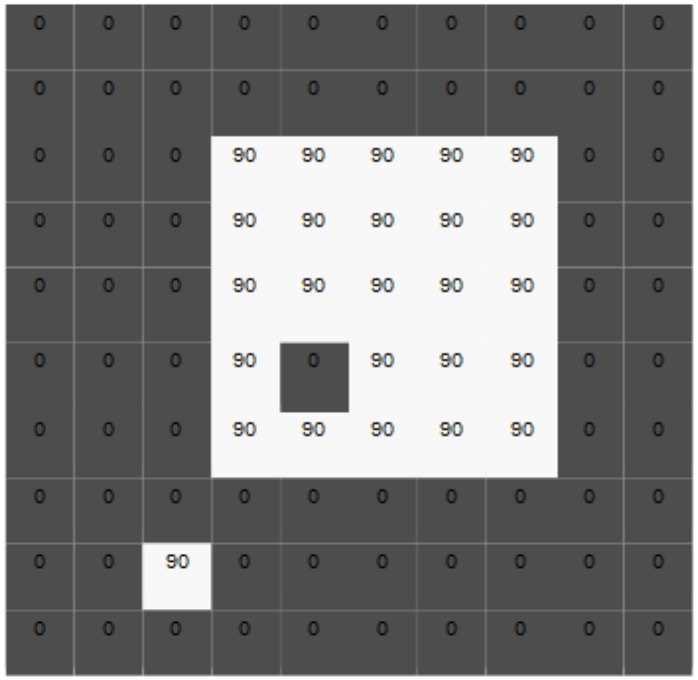
$$\frac{1}{9} \sum_{i,j \in \{-1,0,1\}} w(i,j) f(x+i, y+j)$$

EXAMPLE: BOX FILTER

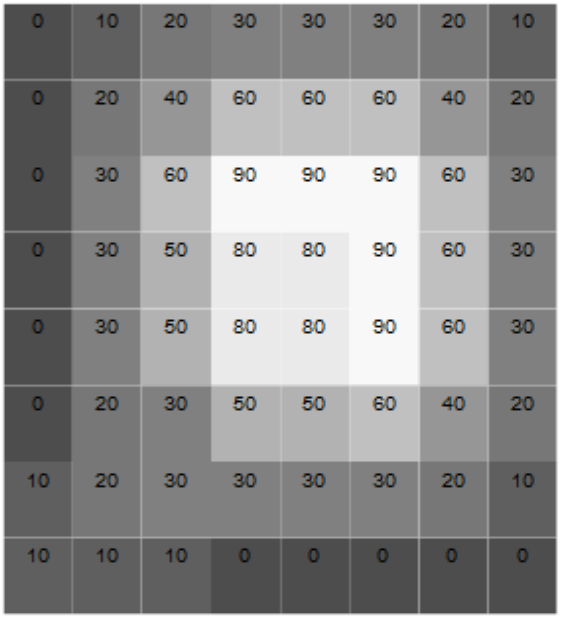
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$w(i,j)$

$f(x,y)$



$h(x,y)$



$$h(x,y) = \sum_{i,j\in\{-1,0,1\}} w(i,j)f(x+i,y+j)$$

EXAMPLE: BOX FILTER

What does it do?

- Replaces each pixel with an average of its neighborhood

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

$w(i,j)$

- Achieve smoothing effect (remove sharp features)

PRACTICE WITH FILTERS

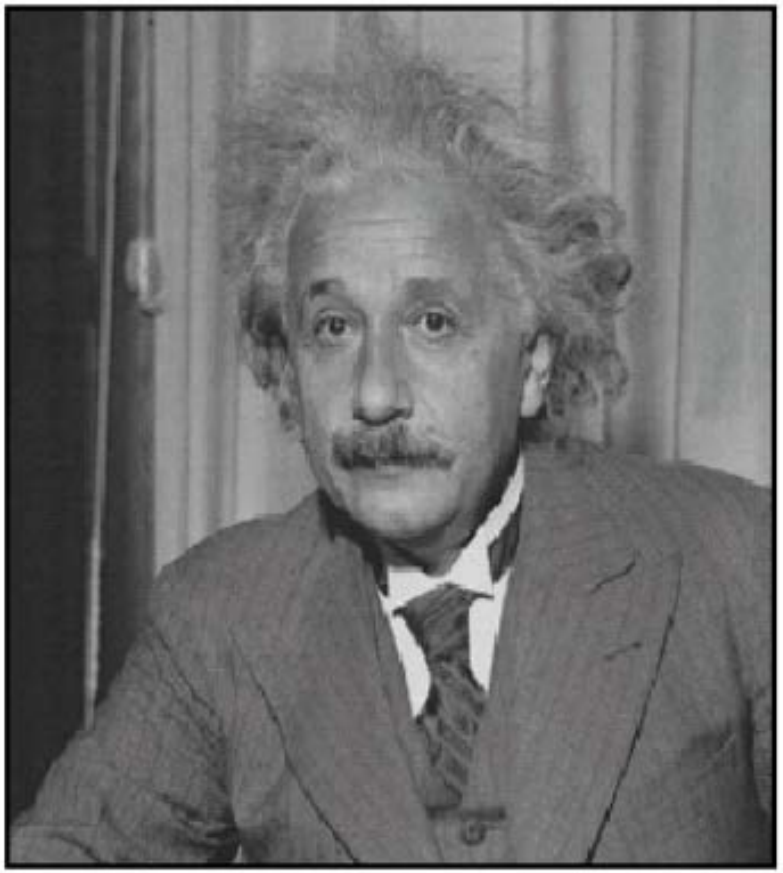
Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

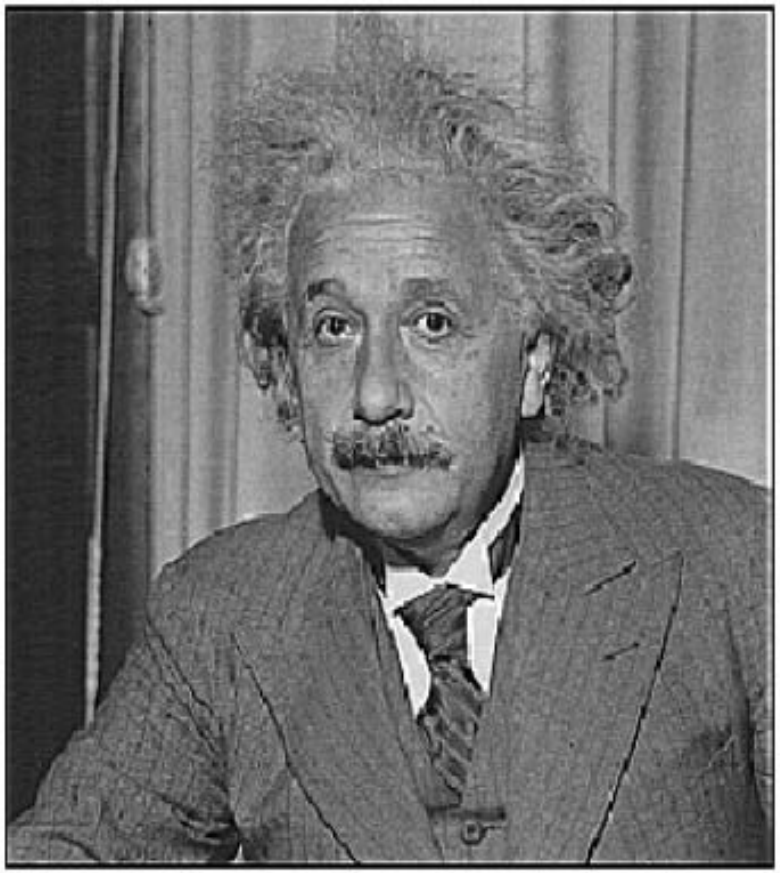
Sharpening filter

- Accentuates differences with local average

SHARPENING

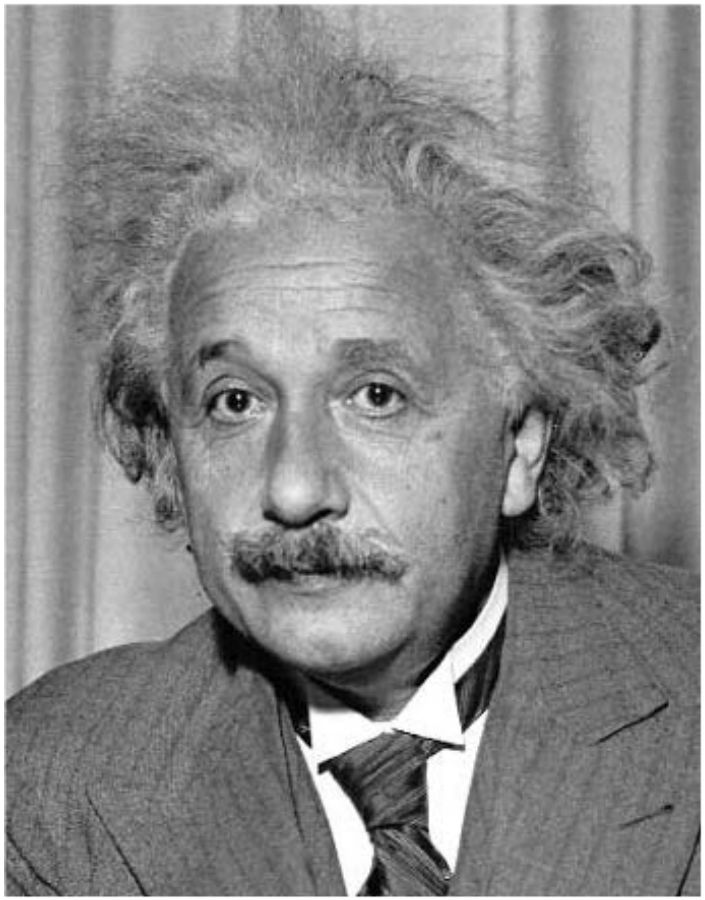


before



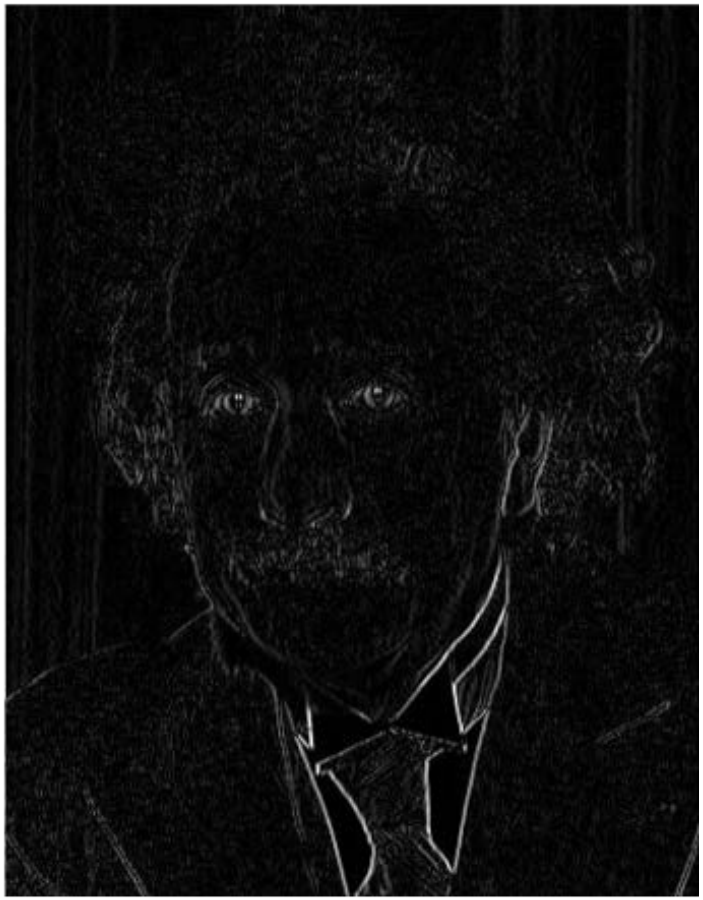
after

PRACTICE WITH FILTERS



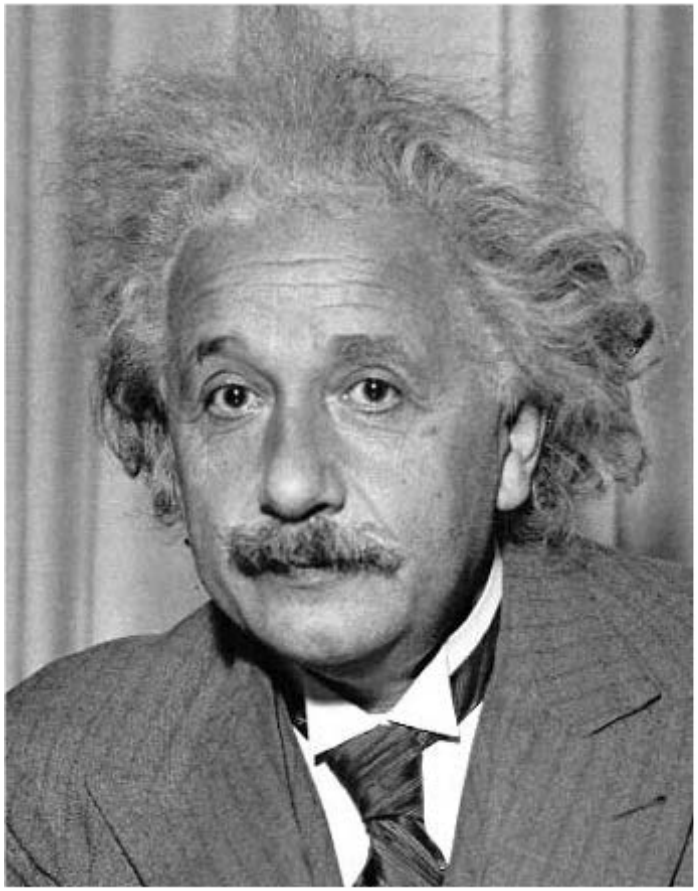
1	0	-1
2	0	-2
1	0	-1

Sobel



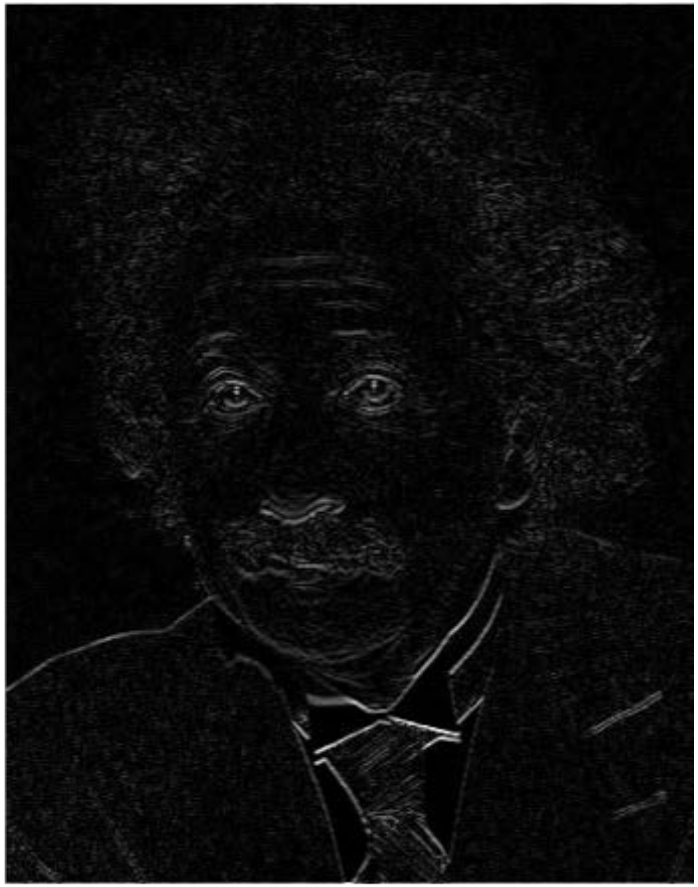
Vertical Edge
(absolute value)

PRACTICE WITH FILTERS



1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)

KEY PROPERTIES OF LINEAR FILTERS

Linearity:

filter(f1 + f2) = filter(f1) + filter(f2)

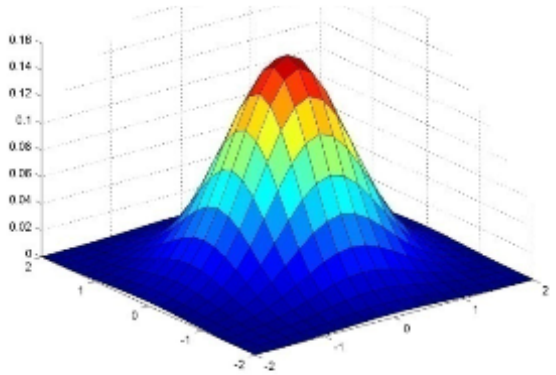
Shift invariance: same behavior regardless of pixel location

filter(shift(f)) = shift(filter(f))

Any linear, shift-invariant operator can be represented as a convolution

AN IMPORTANT FILTER: GAUSSIAN

- Weight contributions of neighboring pixels by nearness

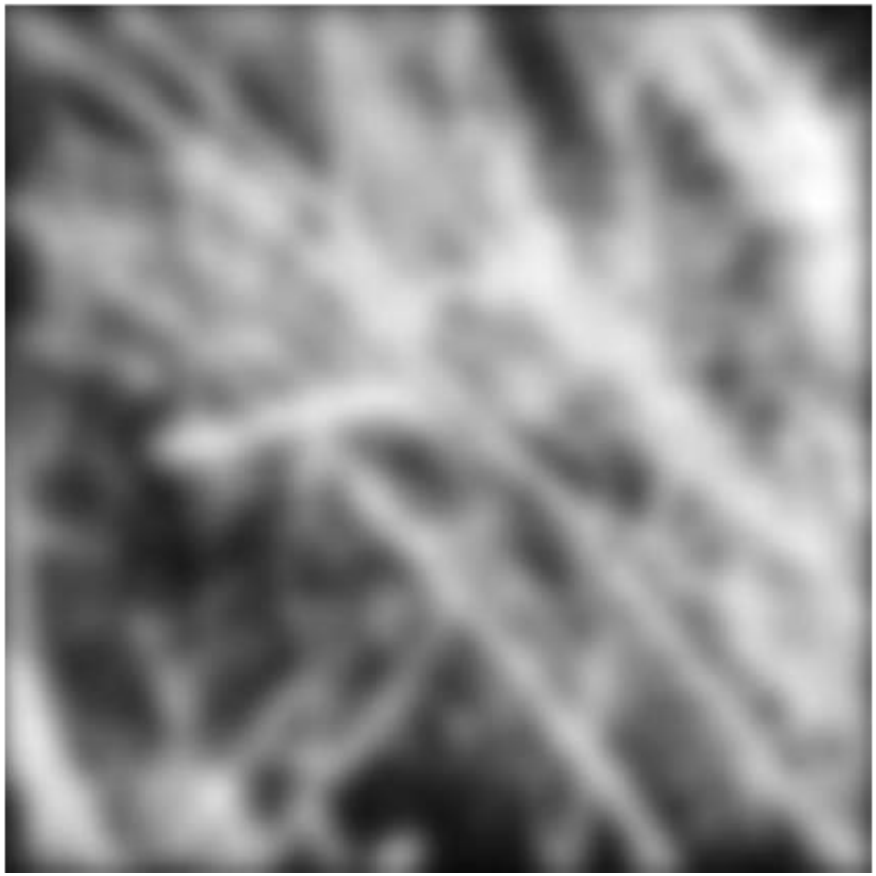


0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

SMOOTHING WITH A GAUSSIAN FILTER



APPLICATION OF FILTERS: TEXTURES

WHAT IS A TEXTURE?

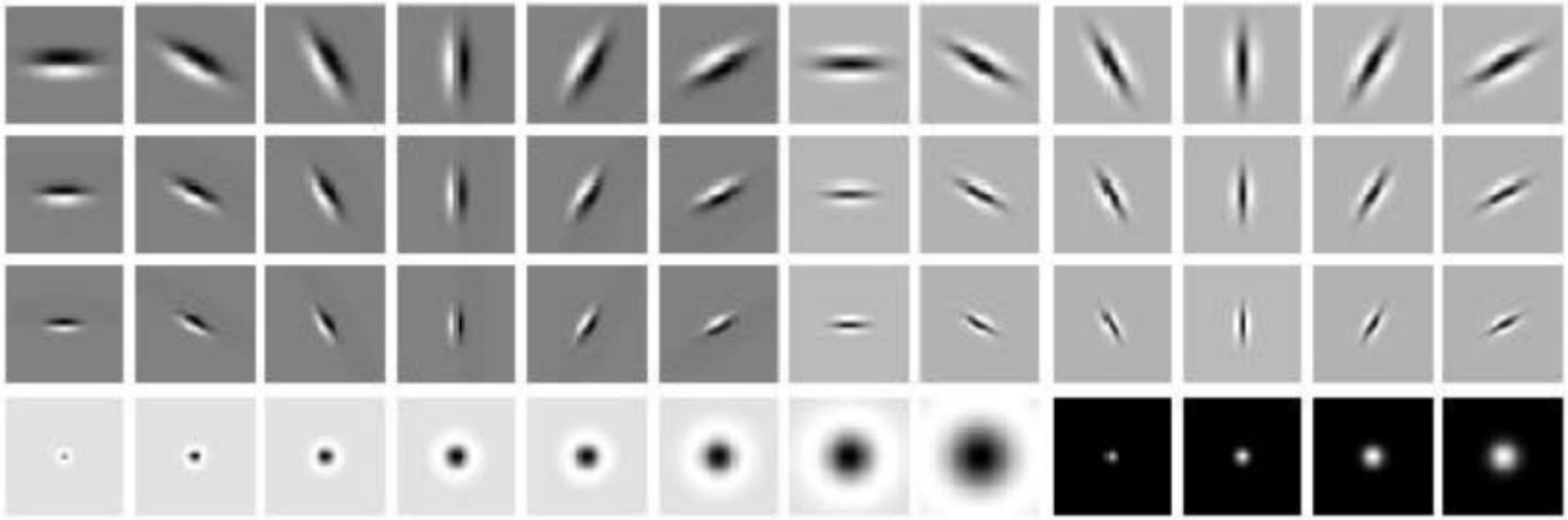
Regular or stochastic patterns caused by bumps, grooves, and/or markings

HOW CAN WE REPRESENT TEXTURE?

Compute responses of blobs and edges at various orientations and scales

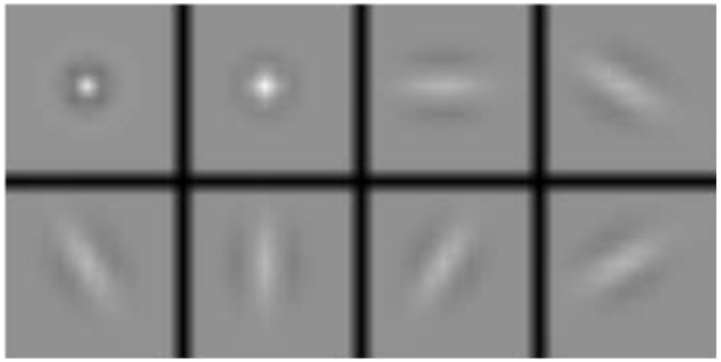
OVERCOMPLETE REPRESENTATION: FILTER BANKS

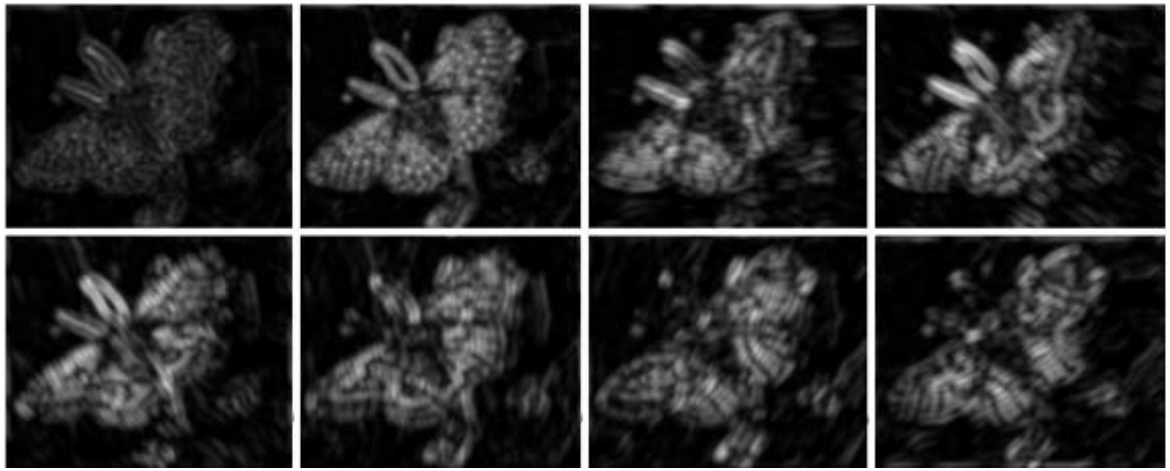
LM Filter Bank



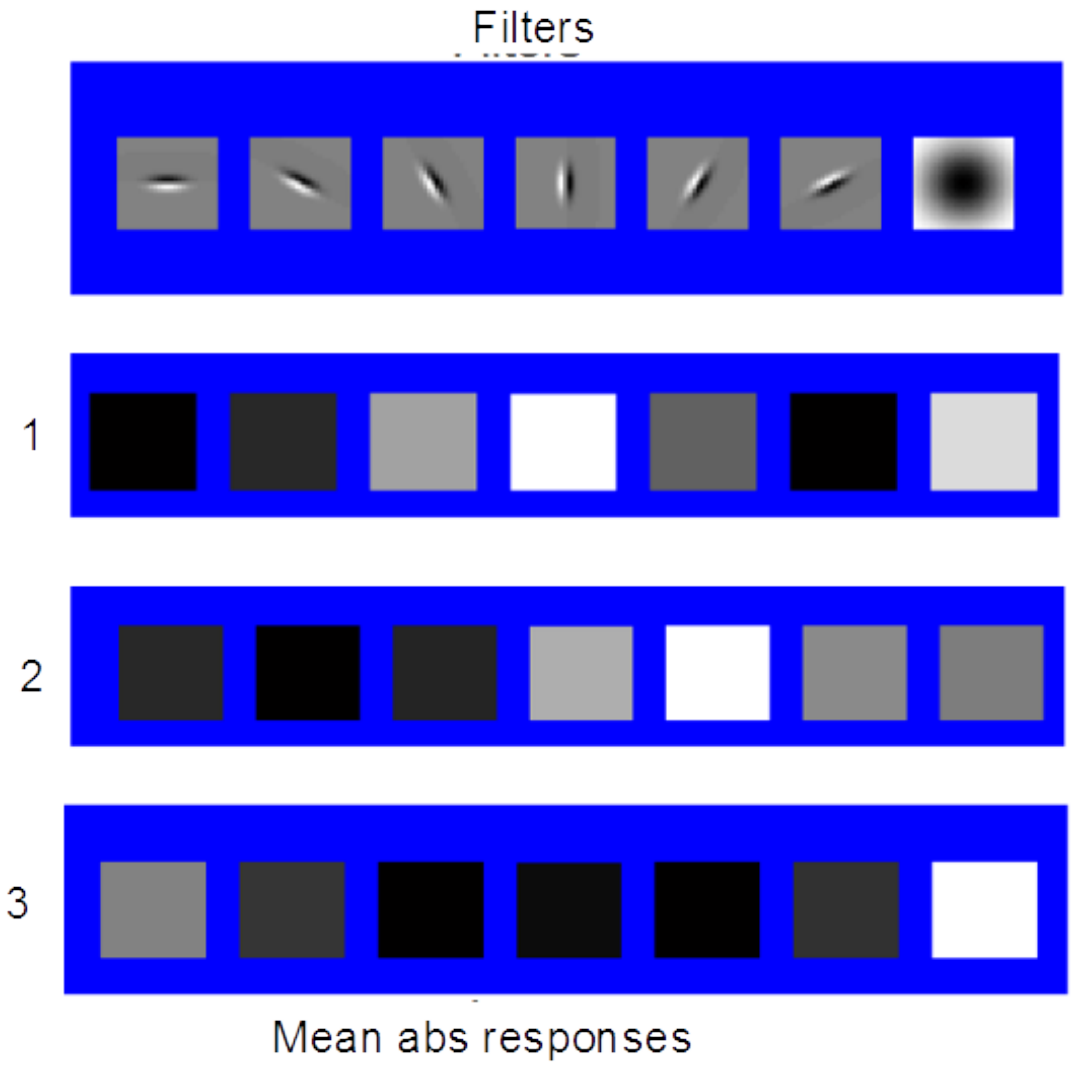
HOW TO USE IT...

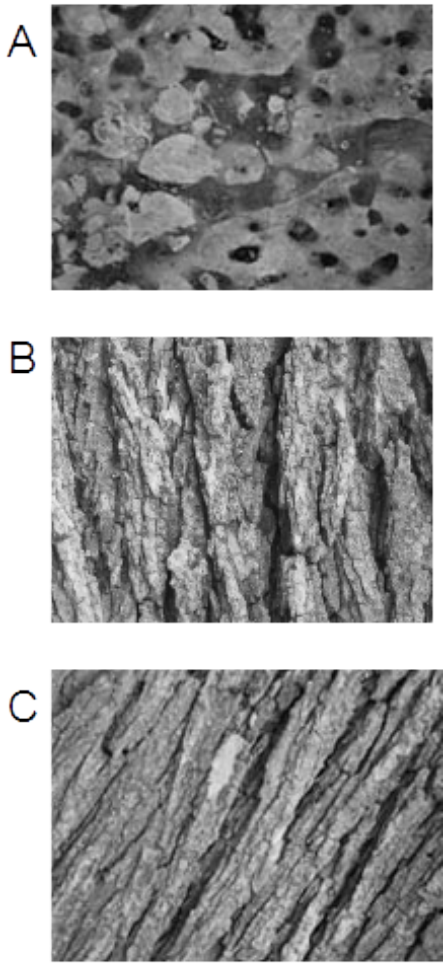
- Process image with each filter and keep responses (or squared/abs responses)



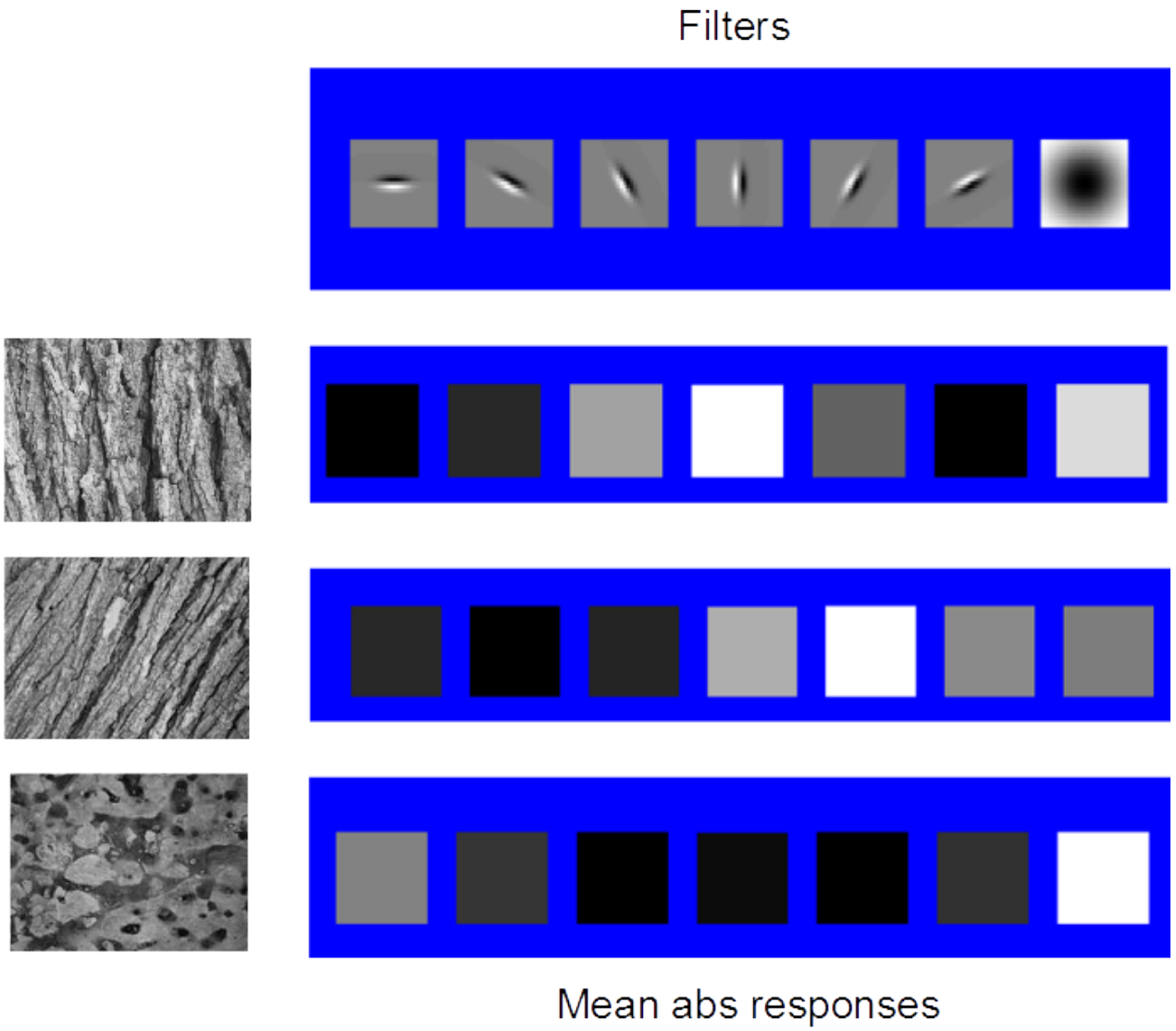


MATCH TEXTURE WITH RESPONSE



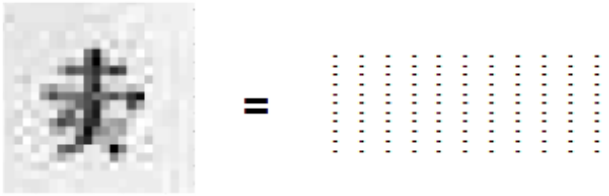


MATCH TEXTURE WITH RESPONSE

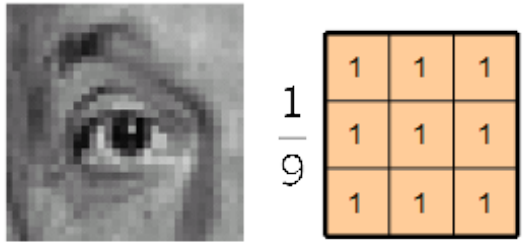


TO REMEMBER

- Image is a matrix of numbers



- Linear filtering is sum of dot product at each position
  - Can smooth, sharpen, translate (among many other uses)

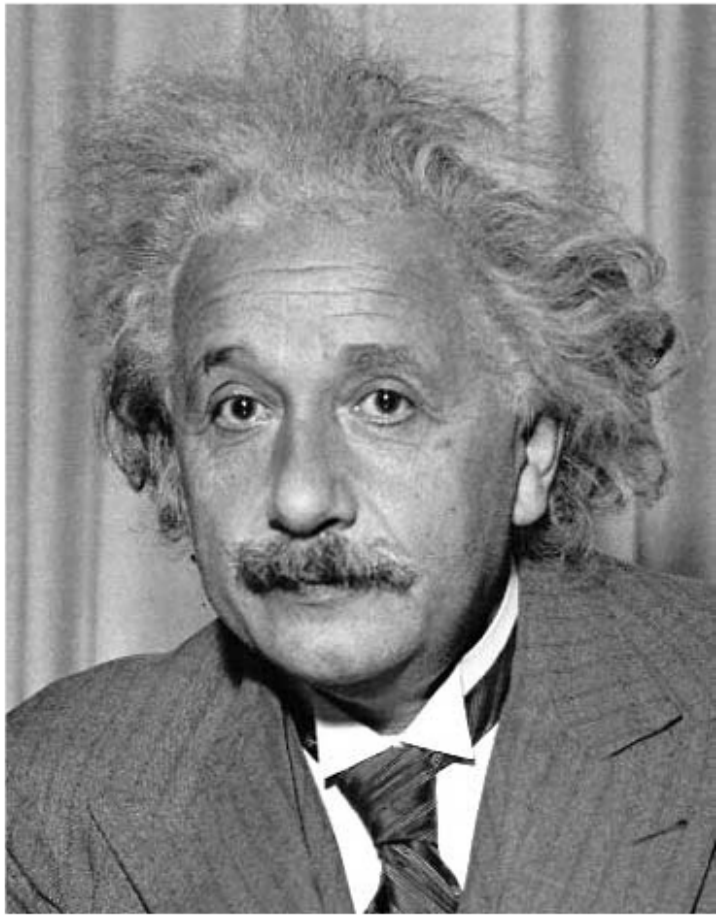


- Be aware of details for filter size, extrapolation, cropping

TEMPLATE MATCHING

- Goal: find  in image

- Main challenge: What is a good similarity or distance measure between two patches?
  - Correlation
  - Zero-mean correlation
  - Sum Square Difference
  - Normalized Cross Correlation



MATCHING WITH FILTERS

- Goal: find  in image
- Method 3: normalized cross-correlation

f = image
g = filter

Filter mean

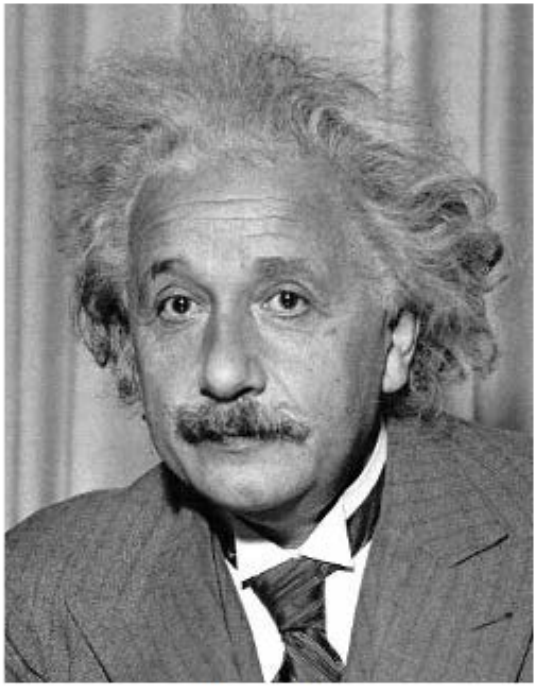
Image patch mean

$$h(m,n) = \frac{\sum_{k,l}(g(k,l) - \bar{g})(f(m+k,n+l) - \bar{f}_{m,n})}{\sqrt{\sum_{k,l}(g(k,l) - \bar{g})^2 \sum_{k,l}(f(m+k,n+l) - \bar{f}_{m,n})^2}}$$

Matlab: `normxcorr2(template, im)`

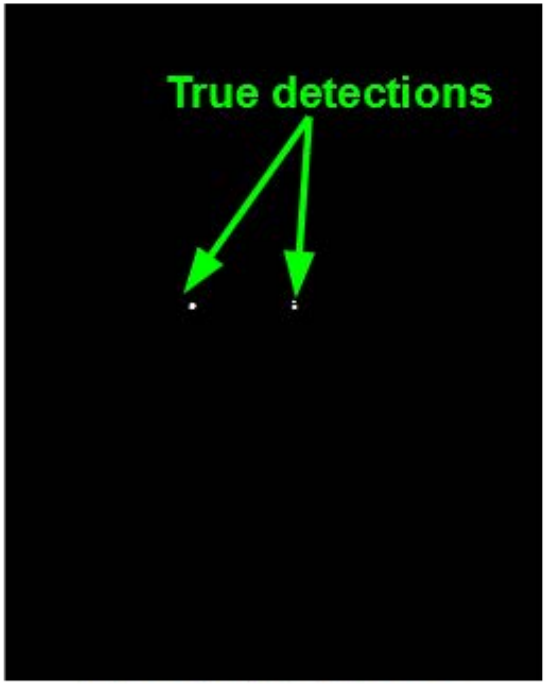
MATCHING WITH FILTERS

- Goal: find  in image
- Method 3: normalized cross-correlation



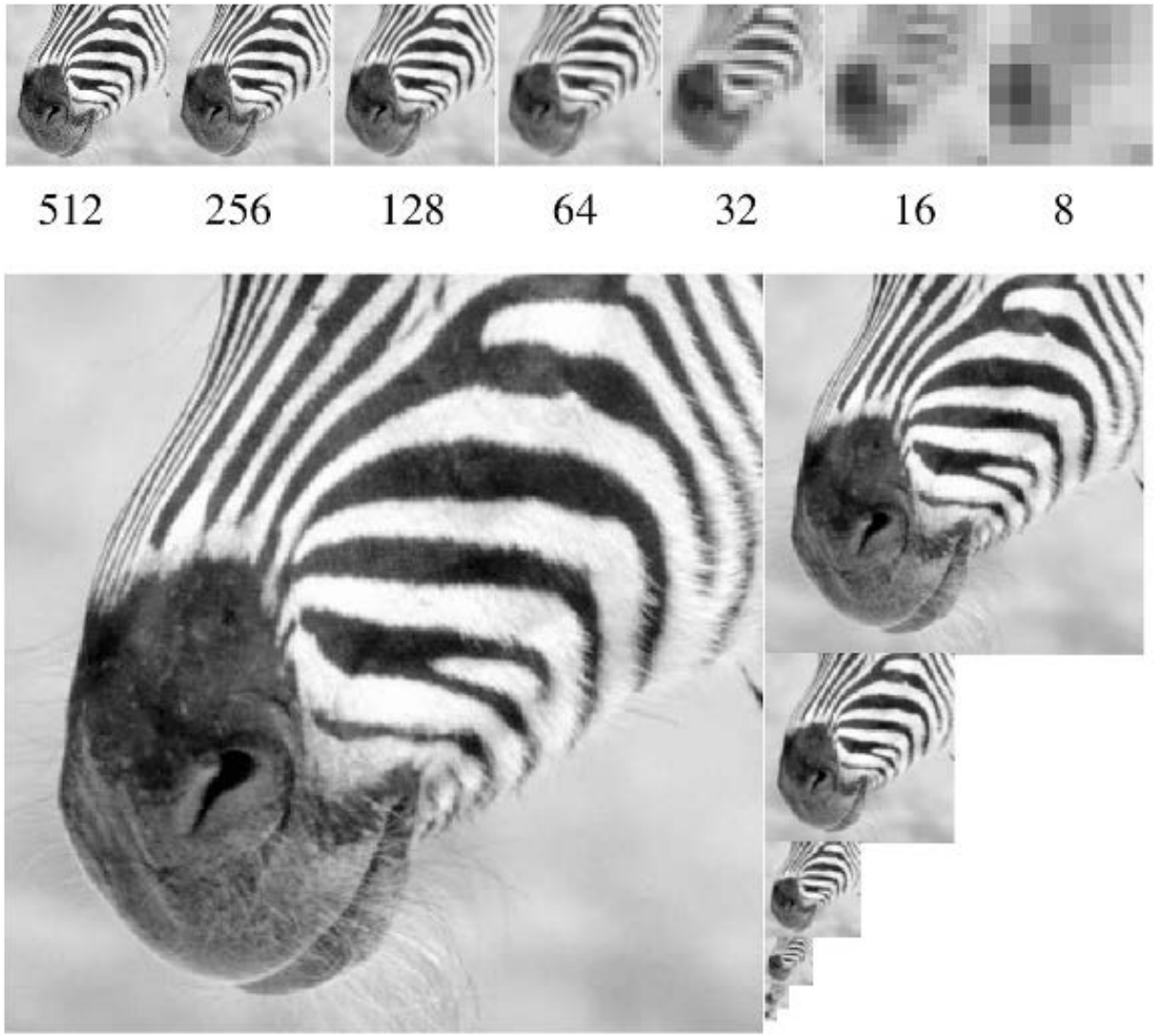
Input

Normalized X-Correlation



Thresholded Image

GAUSSIAN PYRAMID

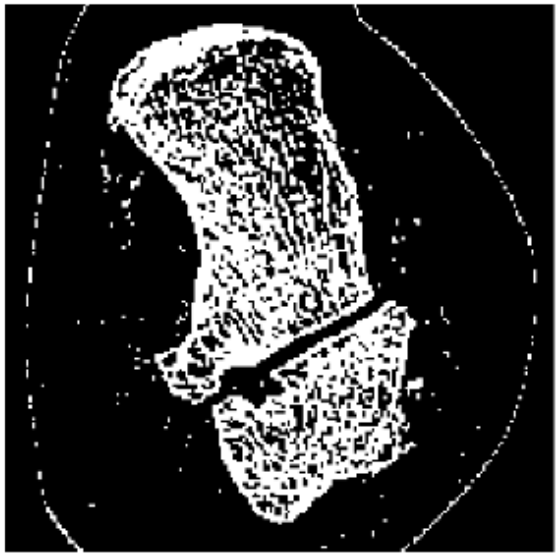


Source: Forsyth

SEGMENTATION

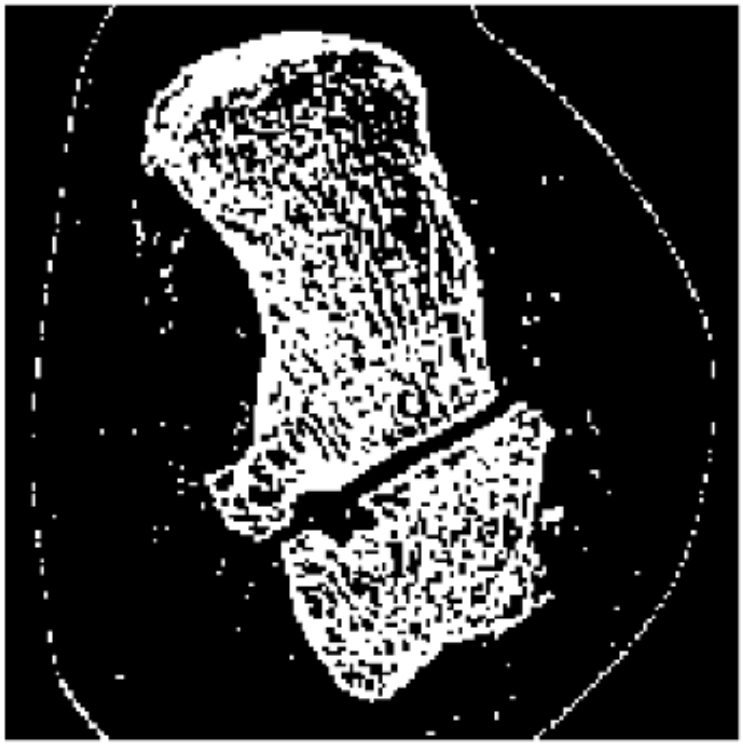
- Separating object from background in a grayscale picture
  - A simple method: thresholding by pixel (voxel) color
    - All pixels (voxels) with color above a threshold are set to 1

Grayscale picture



Thresholded binary picture

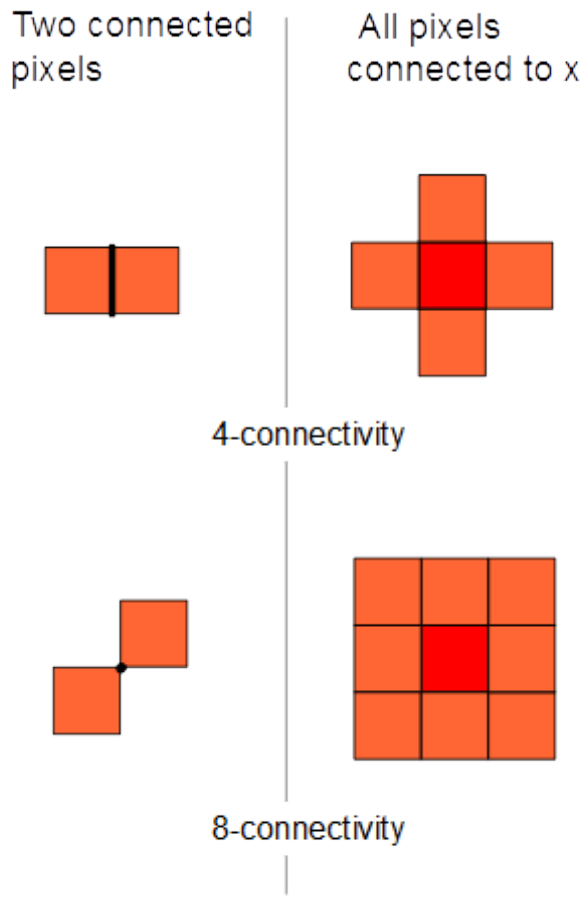
Connected Components



Take the largest 2 components of the object

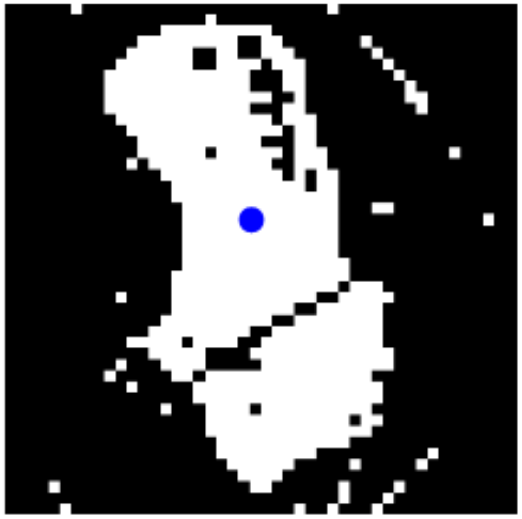
Connectivity (2D)

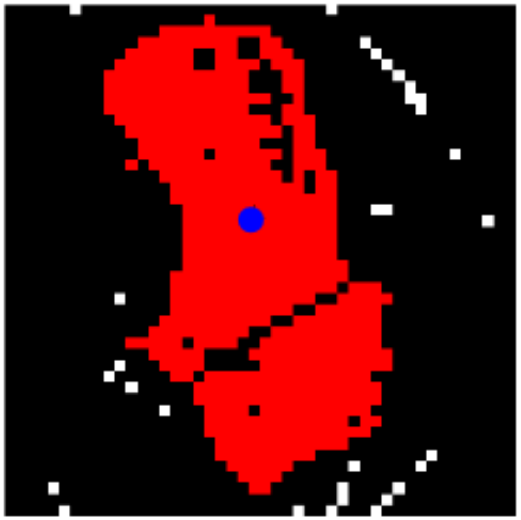
- Two pixels are connected if their squares share:
  - A common edge
    - 4-connectivity
  - A common vertex
    - 8-connectivity



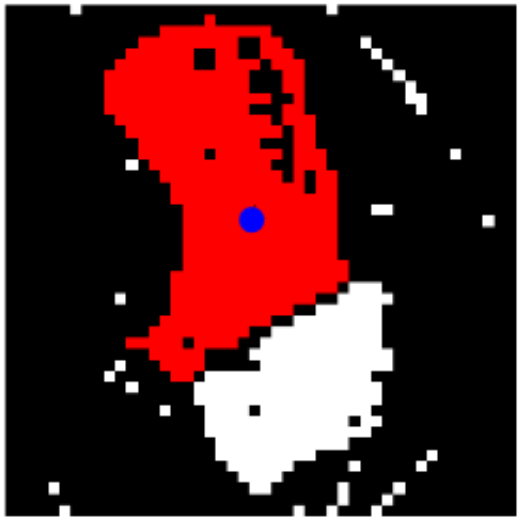
Connectivity (2D)

- What is the component connected to the blue pixel?





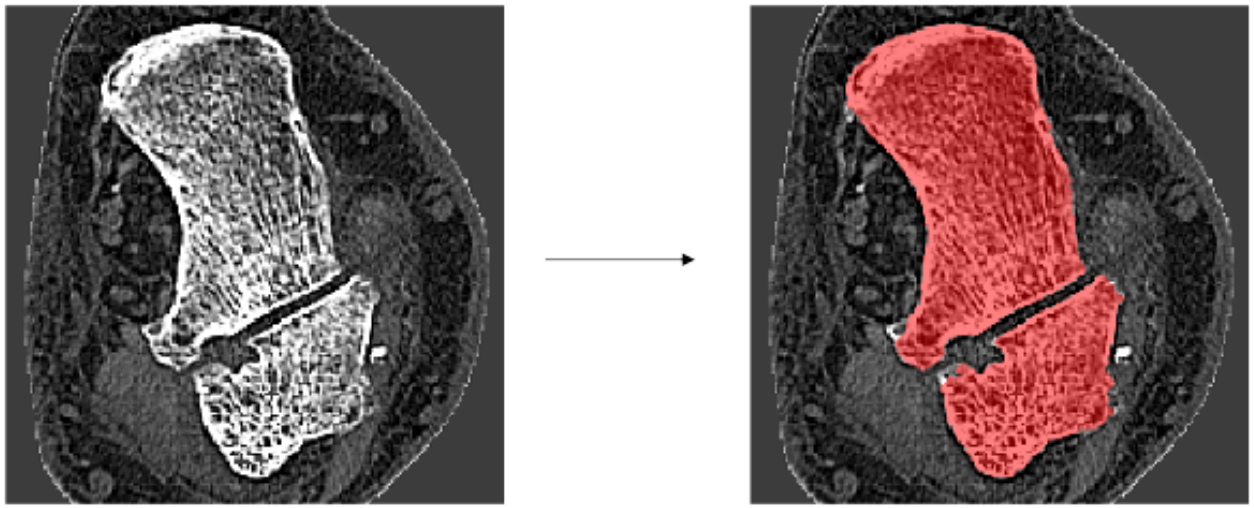
8-connectivity



4-connectivity

EXAMPLE

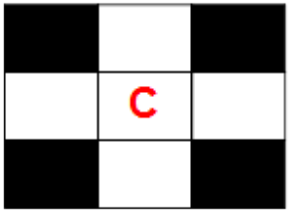
- A simple 2D segmentation routine
  - Initial segmentation using thresholding
  - Using connected components and opening/closing to “clean up” the segmentation.

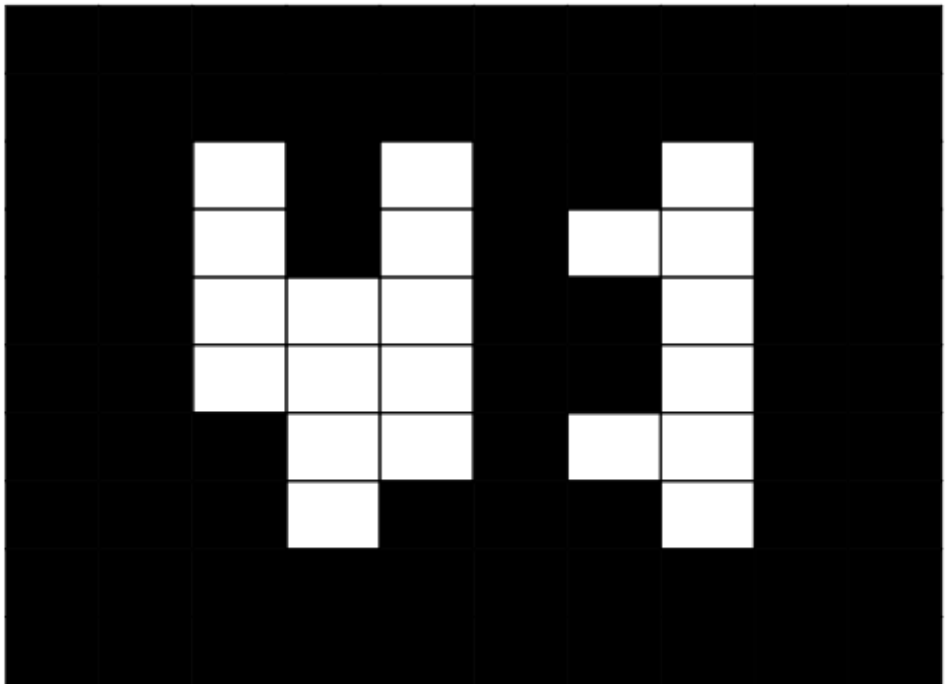


Mathematical morphology operations

Operations are defined by an structuring element and the image we want to process

Structuring element is defined by a neighborhood and a center



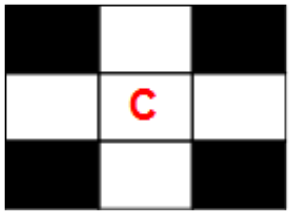


Basic operations: Dilation

Let p=(p_x,p_y) the coordinates of all white pixels in image I and s=(s_x,s_y) the coordinates of a white pixel in the structuring element with respect to the center:

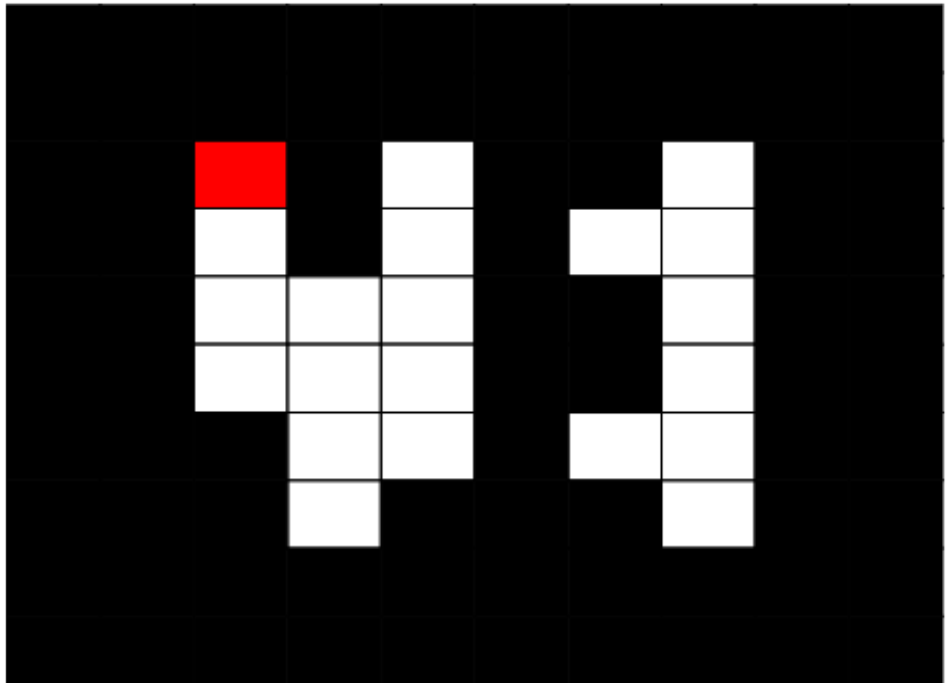
The dilation is defined as the union of the translations p + s for all s in the structuring element.

$$D = \{p + s,\ \forall s \in S,\ \forall p \in P\}$$



Example:
S = {(0,-1),(0,0),(-1,0),
(0,+1),(+1,0)}

p = (3,3)

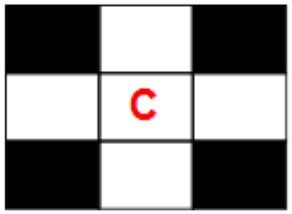


Basic operations: Dilation

Let p=(p_x,p_y) the coordinates of all white pixels in image I and s=(s_x,s_y) the coordinates of a white pixel in the structuring element with respect to the center:

The dilation is defined as the union of the translations p + s for all s in the structuring element.

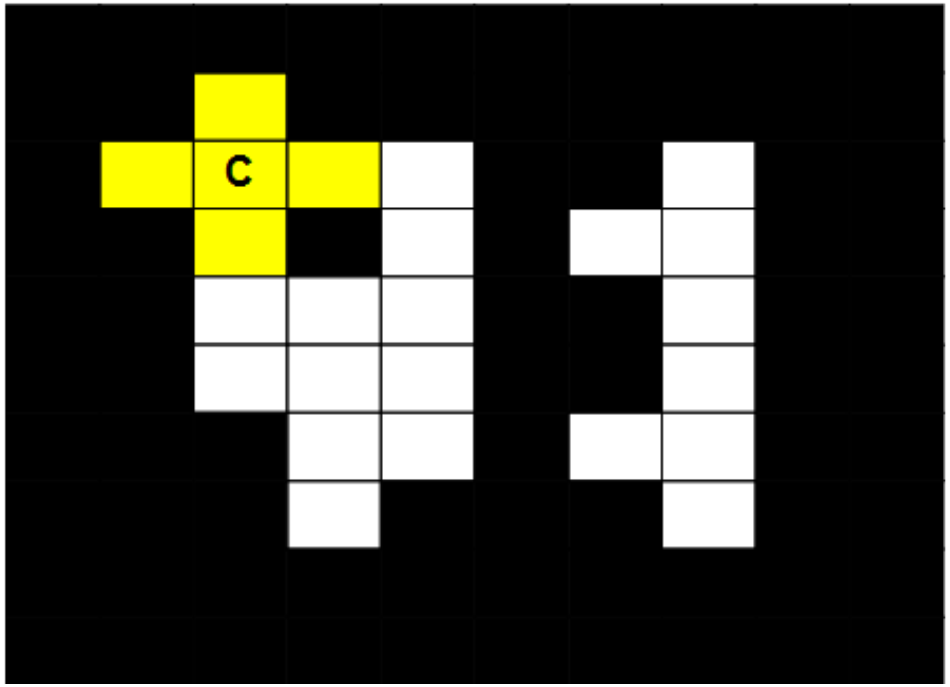
$$D = \{p + s,\ \forall s \in S,\ \forall p \in P\}$$



Example:
S = {(0,-1),(0,0),(-1,0),(0,+1),(+1,0)}

p = (3,3)

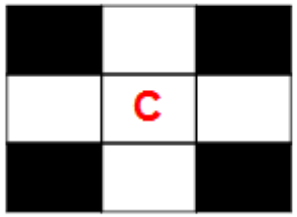
p+s = {(3,2),(3,3),(2,3),(3,4),(4,3)}



Basic operations: Dilation in practice

Put the SE over all white pixels of the image

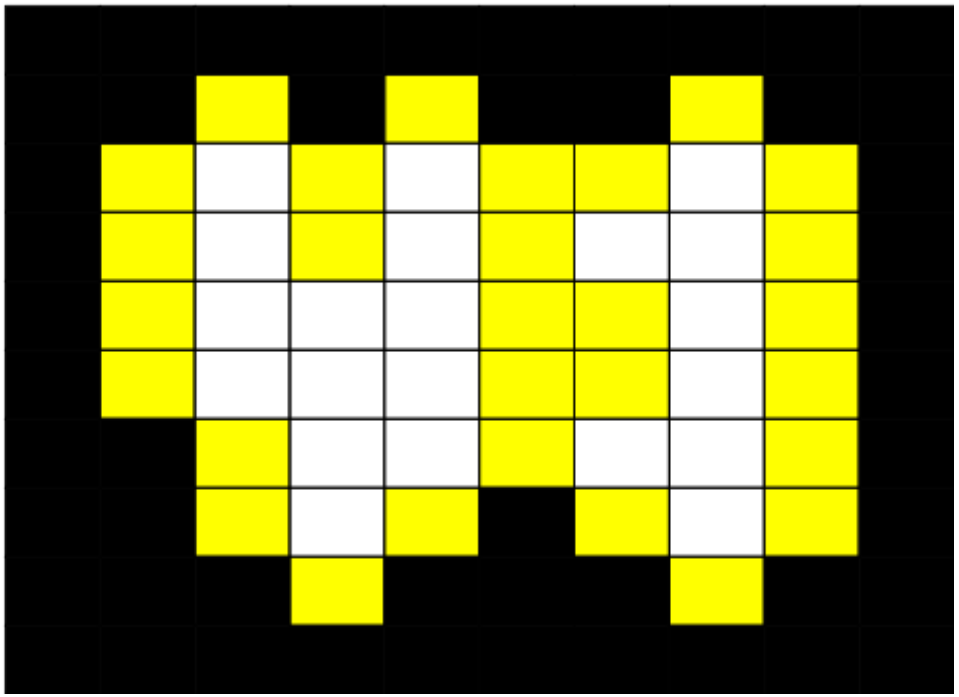
$$D = \{p + s,\ \forall s \in S,\ \forall p \in P\}$$



Example:
S = {(0,-1),(0,0),(-1,0),(0,+1),(+1,0)}

p = (3,3)

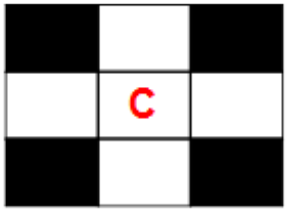
p+s = {(3,2),(3,3),(2,3),(3,4),(4,3)}

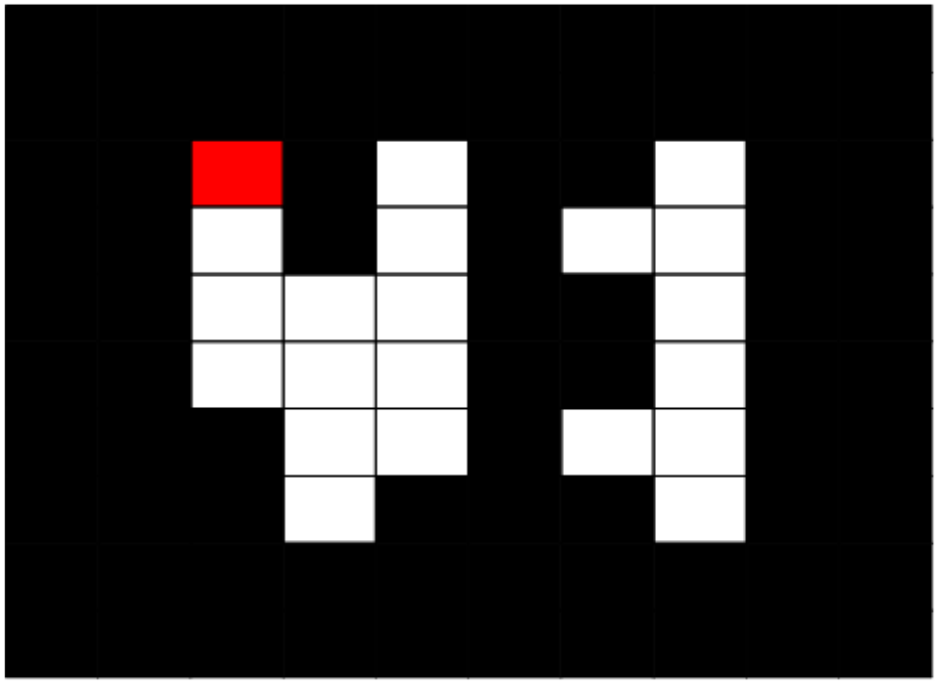


Basic operations: Erosion

Let p=(p_x,p_y) the coordinates of a pixel in the image and P the set of the coordinates of all white pixels in image I. Let s=(s_x,s_y) the coordinates of pixel in the structuring element and S the set of the coordinates of all white pixel in the structuring element with respect to the center:

The erosion is defined as the elements $E = \{p |\ p + s \in P,\ \forall s \in S\}$



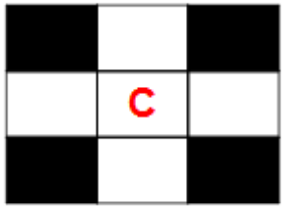


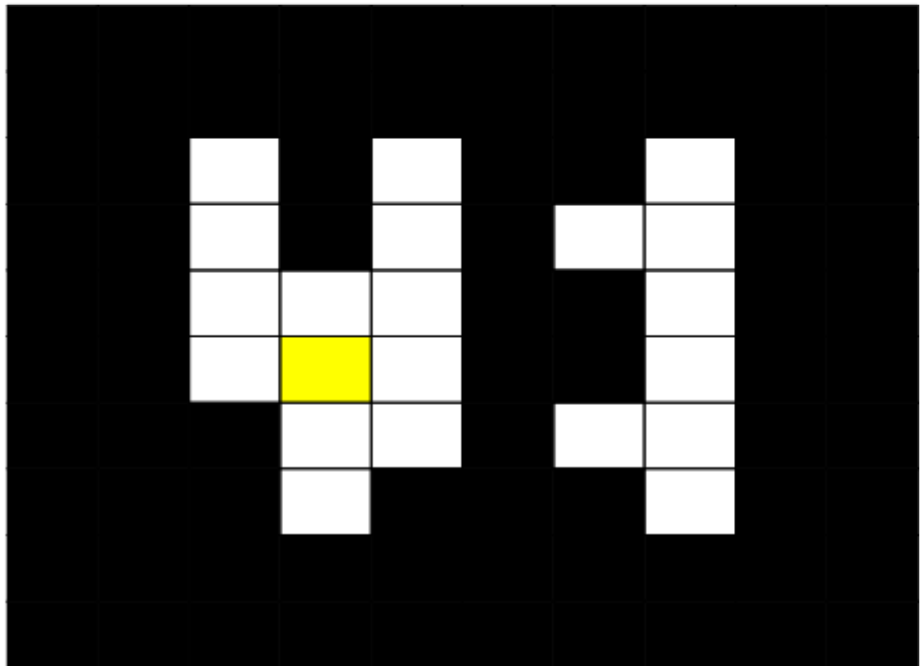
Example:
S = {(0,-1),(0,0),(-1,0),(0,+1),(+1,0)}

p = (3,3)

Basic operations: Erosion in practice

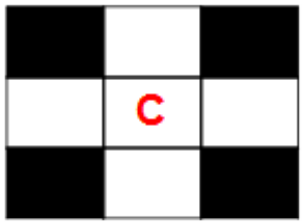
"Keep all the points in which the structuring element fits in the white region"
(only true if the center belongs to the inside of the structuring element)

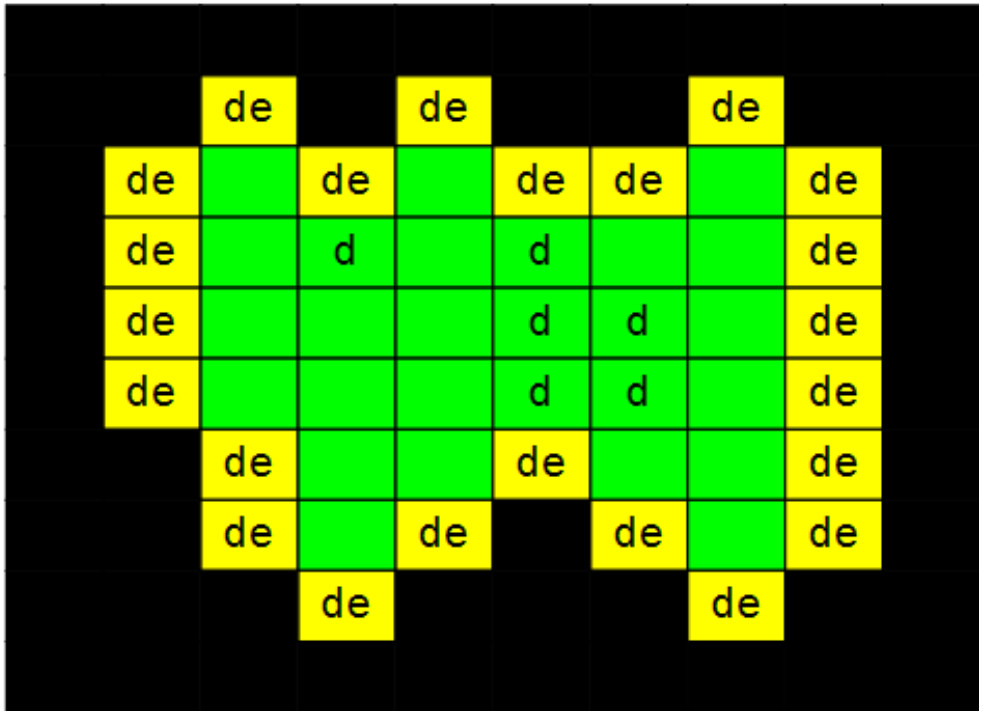




Composing operations:
Closing = Dilation + Erosion

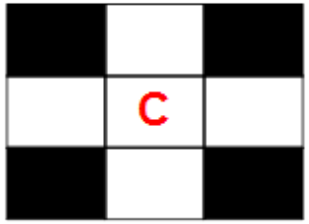
Recovers the original objects closing holes, and joining structures smaller than the SE

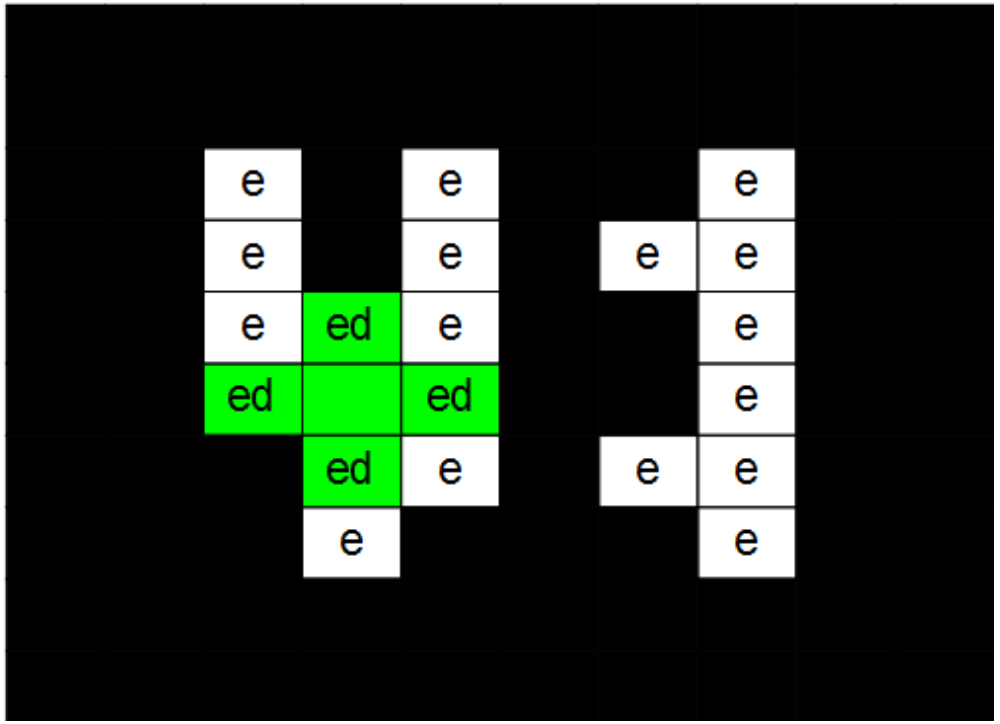




Composing operations:
Opening = Erosion + Dilation

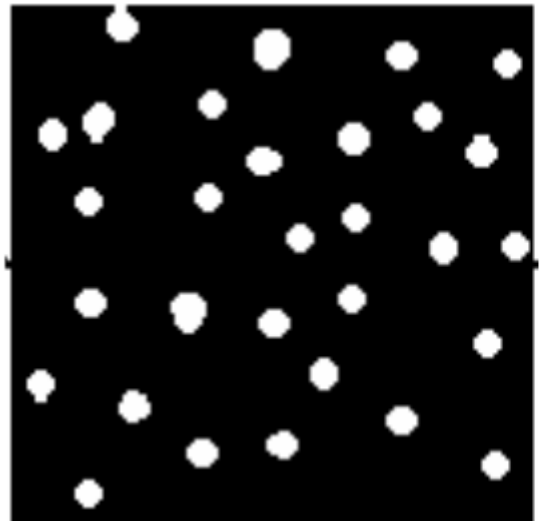
Removes parts of the elements not conforming the SE shape





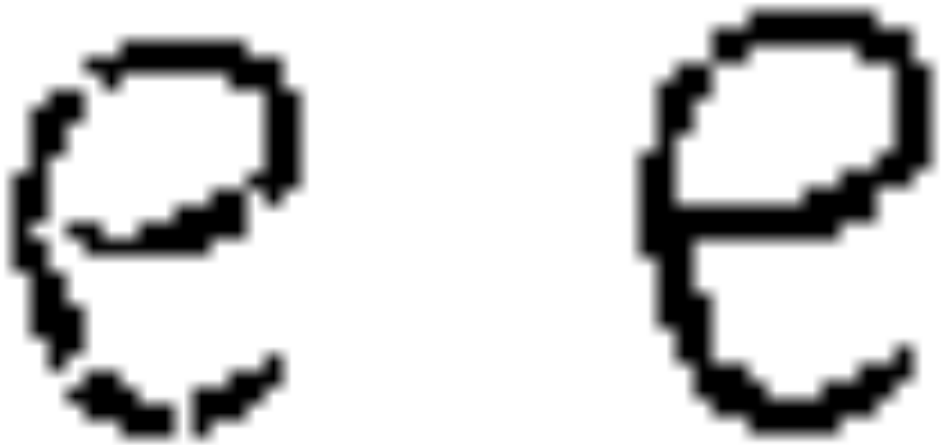
Aplicacions

Separation of structures: If we know that we are looking for disk structures of 7 pixels of diameter, which operation would you use?



Aplicacions

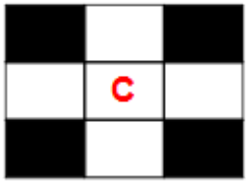
Restauration of binary scanned letters.



Which operation did I use?

Morphological Gradient: Contour detection

Dilation(I)-I External contour
I – erosion(I) Internal contour
Dilation(i)-Erosion(i) Both contours



Inversa de I