Data Understanding and pre-processing Images

Anna Monreale and Francesca Naretto
Computer Science Department

Introduction to Data Mining, 2nd Edition
Chapter 1 & Data Exploration (Additional Resources)

Also for images: know your data

- For preparing data for data mining task it is essential to have an overall picture of your data

- Gain insight in your data
  - with respect to your project goals
  - and general to understand properties

- Find answers to the questions
  - How is the data quality?
  - What about outliers?

Which is the type of data?

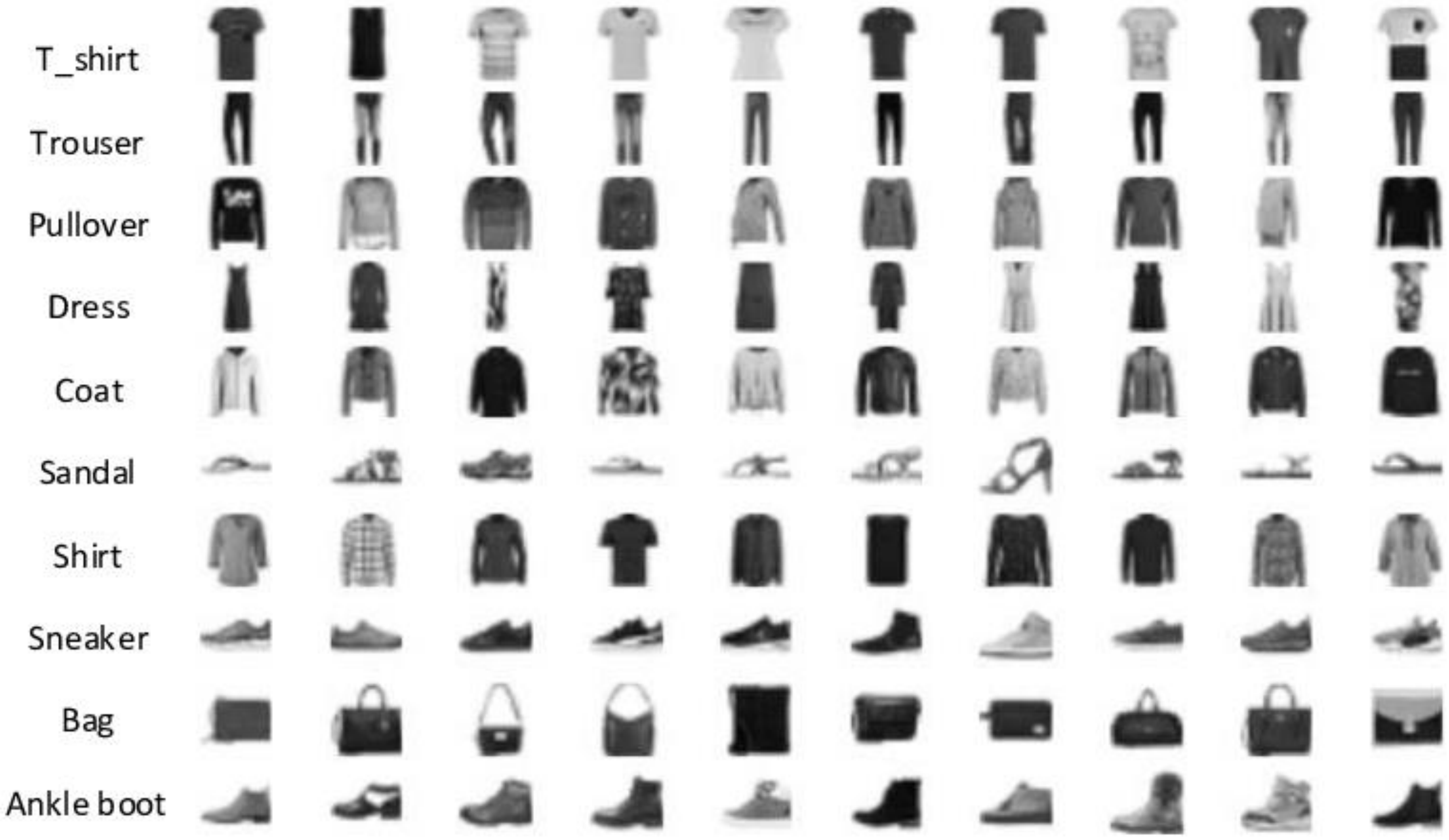
Types of data sets

In this case, we can have 2 kinds of data:

- **Images**
- **Video**
  - **Collection of images in sequence**

Types of data sets

Types of data sets



Types of data sets

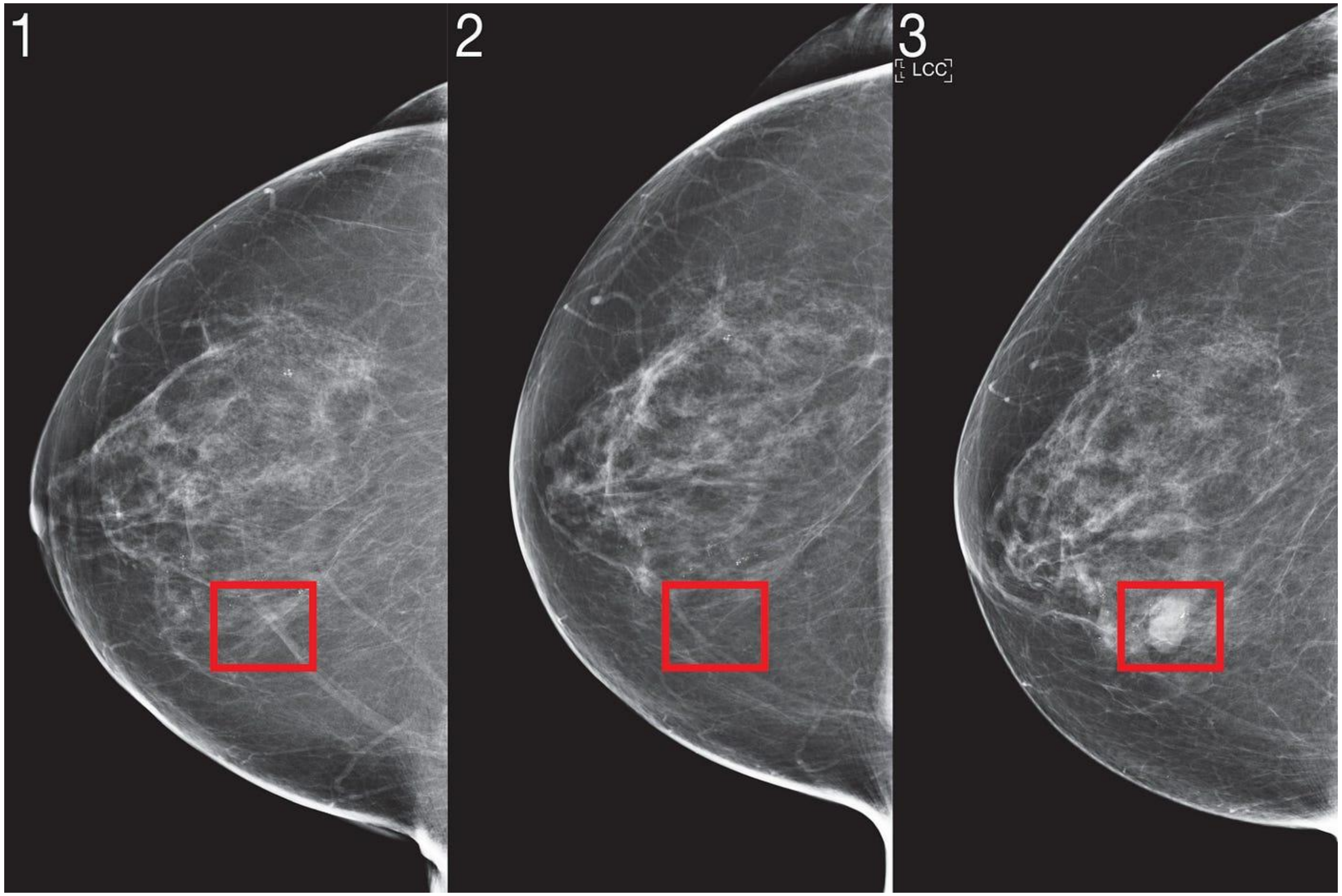
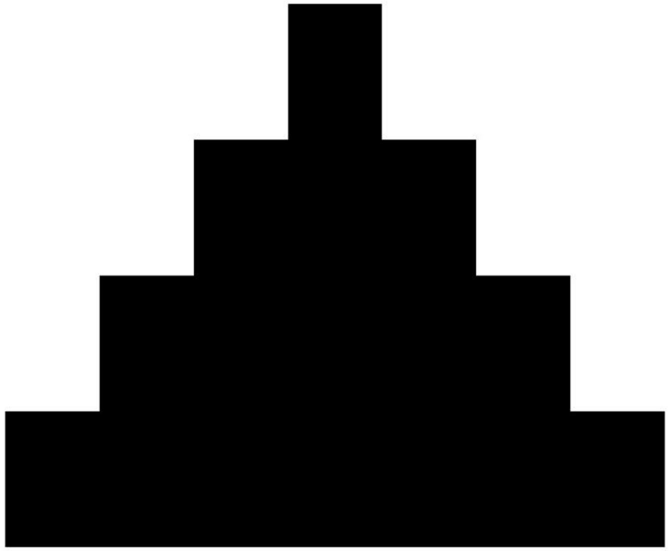
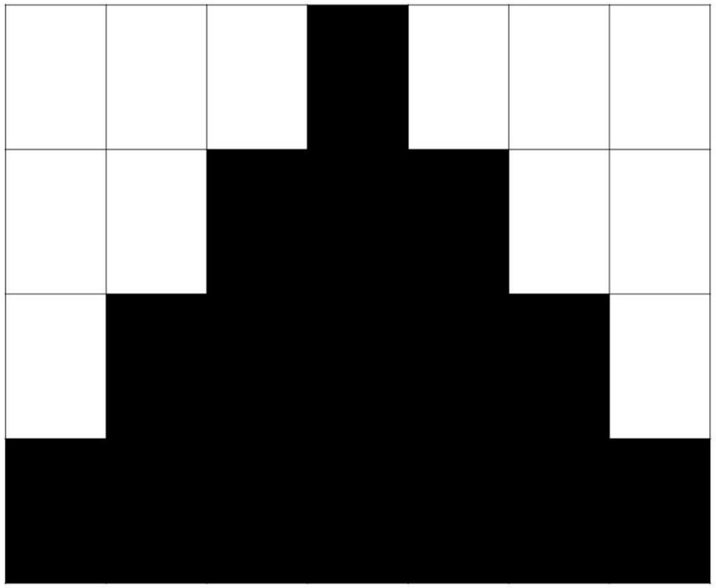


Image characteristics

- Different dimensions
  - Small images (like MNIST)
  - Big images (like ImageNet)
- Colored vs. black and white
  - Ct scans
  - radiography
- High/low resolution
- Different quality of images
- Created by different sources
  - Machines
  - Humans

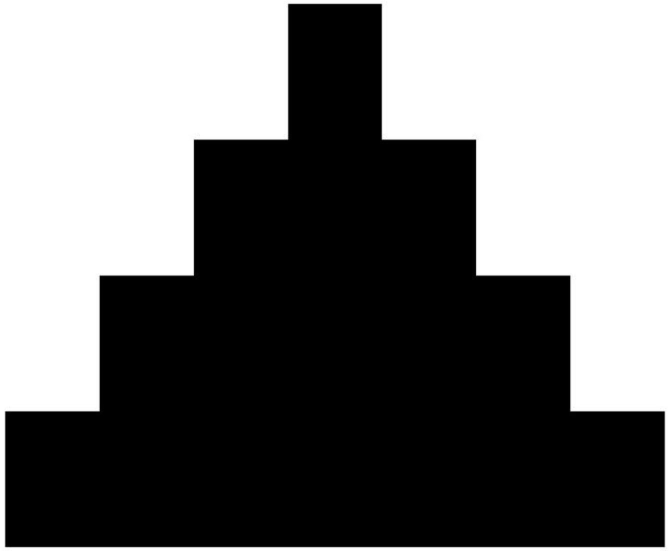
Images are matrices





0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1

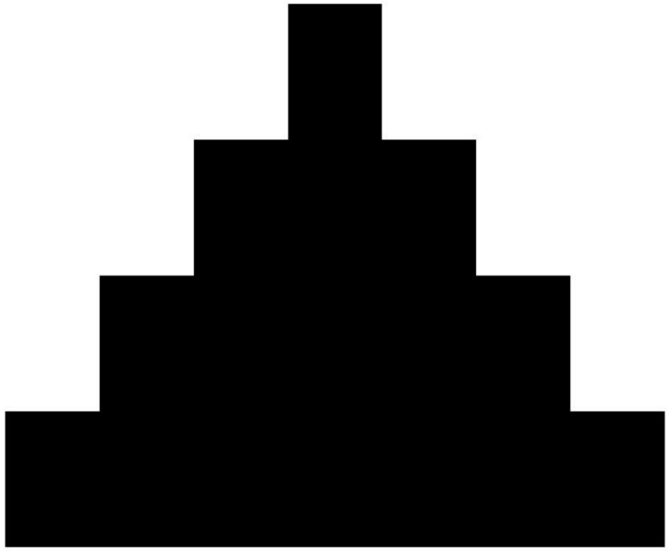
Images are matrices



What if we have shadows of grey?

0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1

Images are matrices



What if we have shadows of grey?

We can have float numbers between 0 and 1

0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1

Images are matrices

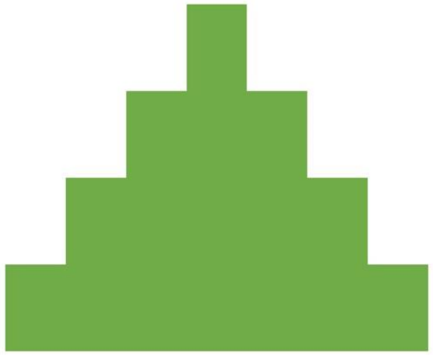
What if we have colors?
A number is not enough to represent all the colors.

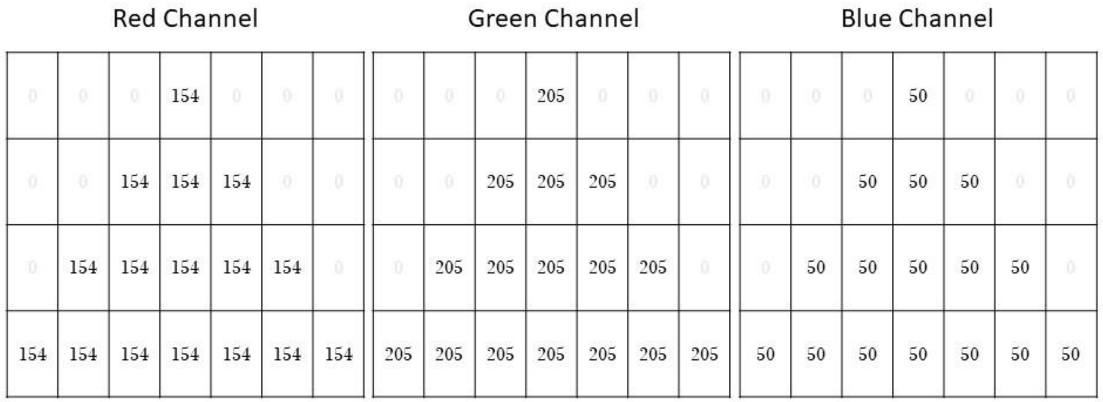
How to represent colors

Sytems based on the components of the colors.

- RGB represents the color as the amount of RED, GREEN and BLUE. Red, Green and Blue are called channels and they can have a value between 0 and 255.
- CYMK represents the color in terms of cyan, magenta, yellow and black.

How to represent colors





Red Channel

0	0	0	154	0	0	0
0	0	154	154	154	0	0
0	154	154	154	154	154	0
154	154	154	154	154	154	154

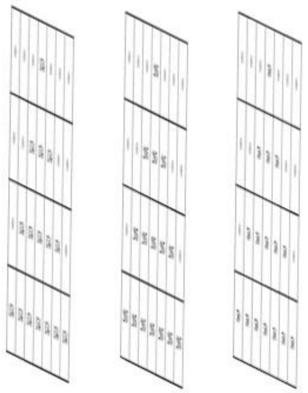
Green Channel

0	0	0	205	0	0	0
0	0	205	205	205	0	0
0	205	205	205	205	205	0
205	205	205	205	205	205	205

Blue Channel

0	0	0	50	0	0	0
0	0	50	50	50	0	0
0	50	50	50	50	50	0
50	50	50	50	50	50	50

3 Matrix Stack

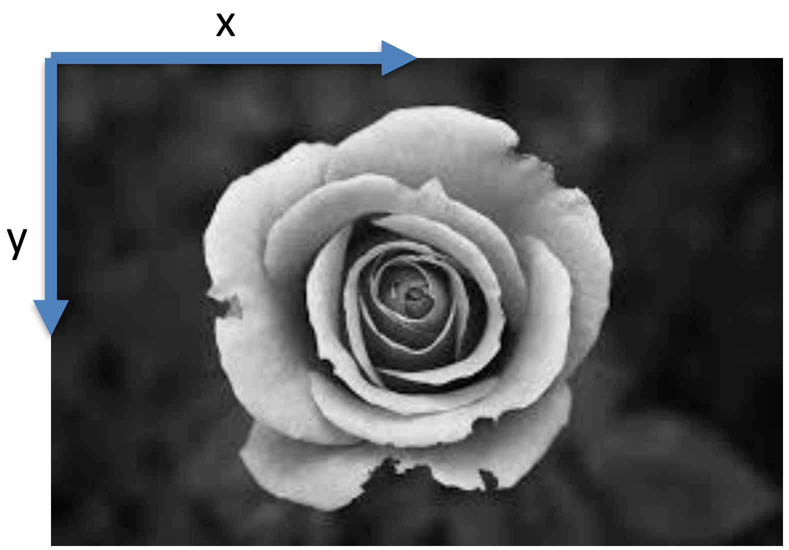


Pros and Cons

To create an image, we need to stack 3 matrices (considering RGB).
With high resolution images (like 4k) the matrices are big and we need a lot of storage space.
Just think the space a video needs!
Nowadays we have a lot of storage space, however...
There are many ways to represent the colors and each of them may have pros and cons.

An image as a function

We can think at the image as a function: *f(x,y)*



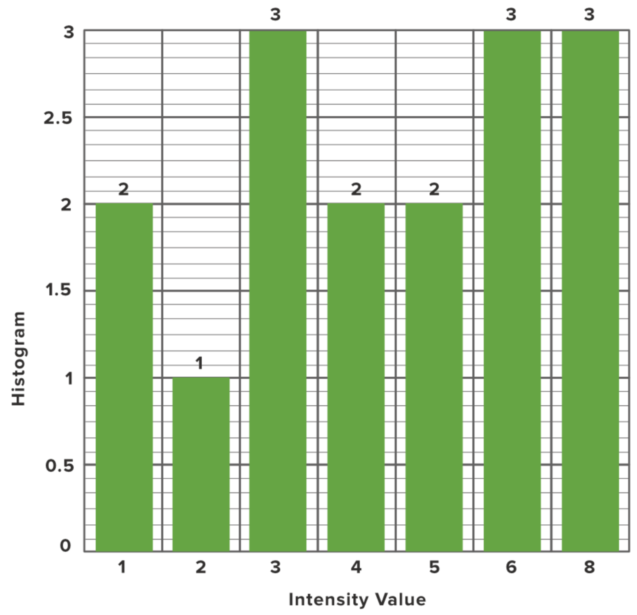
And all the operations we do on the image can be considered trasnformations of functions

Histograms

- Histograms are an important tool in many context, including the images.
- They provide a global description of the image
- If we consider pictures in the shadows of grey, we can have the relative frequencies of different gray levels

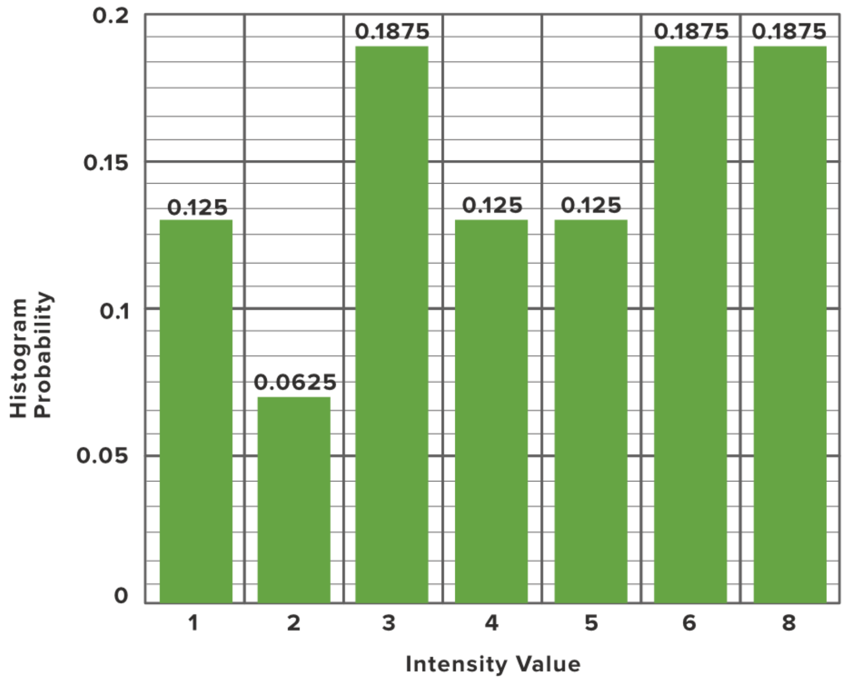
Histograms

- X-axis has the gray levels/intesity values
- Y-axis is the number of pixels in each grey level



Histograms

- X-axis has the gray levels/intesity values
- Y-axis is the probability of occurence of the grey level



Data Quality

When working with data, we need to take into account its quality.

With images, we need to assess the quality:

- Resolution of the image
- Semantic of the image

In fact, if we consider, as an example, images that are recording the movements of an animal, it is important to notice if the animal is present in the images, where it is, if it is clearly visible etc. and not just the resolution of the picture.

Data Quality – the same applies for tabular and time series

- What kinds of data quality problems?
- How can we detect problems with the data?
- What can we do about these problems?

- Examples of data quality problems:
  - Wrong data
  - Duplicate data
  - Noise and outliers

Data Quality issues

- **Techical accuracy:** The picture is techinally sound. No problems with resolution etc.
- **Semantic accuracy:** The picture is ok but the subject may have problems.
  - **Example**: The subject is not visible in the picture or the subject is cut.
- **Unbalanced data:** The data set might be biased to one type of records.
  - **Example**: Defective goods are a very small fraction of all.

Image processing

Transform the image to a clearer one:

- Noise removal
- Fast moving object, motion blur
- Object out of focus

Recovering information you need from the image:

- Enhance the edges
- Enhance the corners

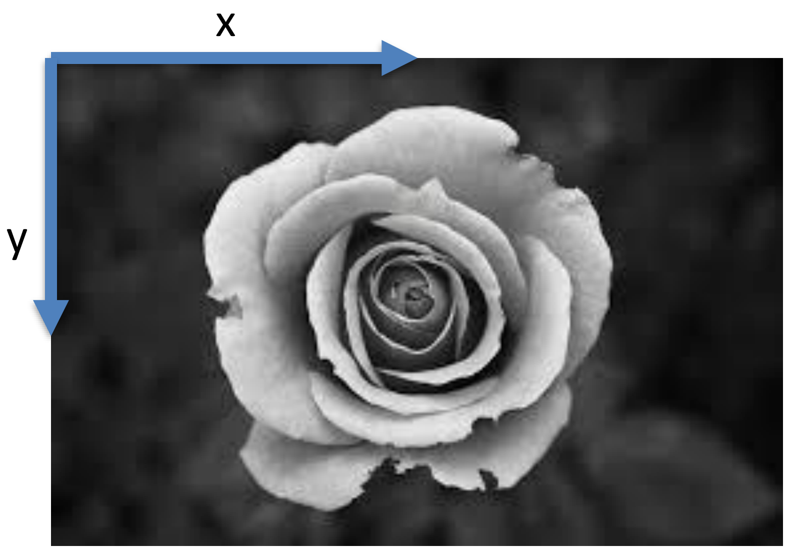
Image processing

Different kinds of image processing:

- Pixel processing
- Linear shift invariance system
- Linear image filters
- Non linear image filters

An image as a function

We can think at the image as a function: *f(x,y)*
f(x,y) is the image intensity in position *(x,y)*.



And all the operations we do on the image can be considered trasnformations of functions

Pixel processing

A transformation T of intensity *f* at each pixel to intensity *g*:

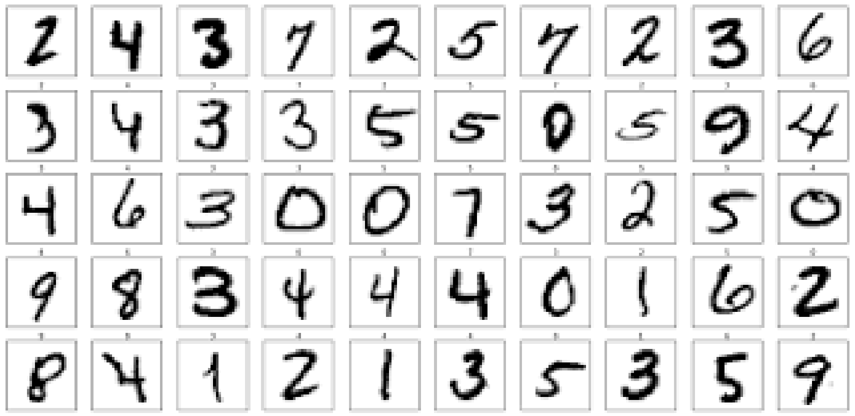
$$g(x, y) = T\big(f(x, y)\big)$$

At every pixel the operation is independent w.r.t. the other pixels.

You can:

- Darken your image (f -128)
- Lighten yout image (f +128)
- Invert it (255 - f)
- Lower/increase the contrast of the image (f/2 or f*2)

Detect object in the image by pixel processing



- To detect the object in each image
- We first convert the image in black and white
- We define a **threshold** number
- We identify the obejct as the composition of all the pixels above the threshold.

Detect object in the image

- How to choose the threshold?
- What to do when the borders are not really clear?

Simple thresholding is not enough

A possible solution is the OTSU method.

It is an automatic threshold determination mechanism: it assumes that in the image there are 2 classes and the objective is to separate them by minimizing the intra-class variance. By separating these 2 classes we are separating background and object.

- Non parametric
- Unsupervised
- Global model

Otsu method

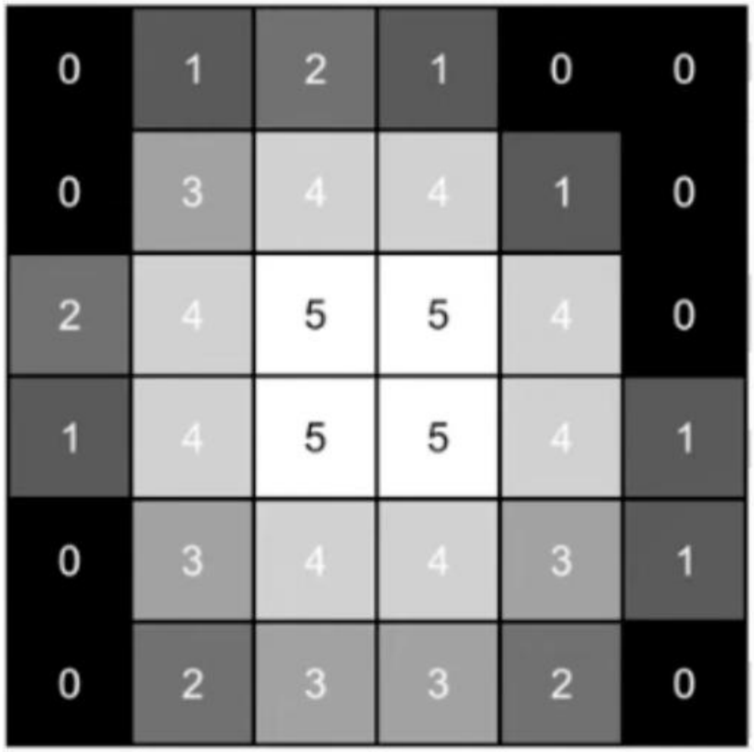
This method calculates the threshold intensity It, which maximizes the between class variance σ_B^2:

$$\sigma_B^2 = W_b * W_f\left(\mu_b + \ \mu_f\right)^2$$

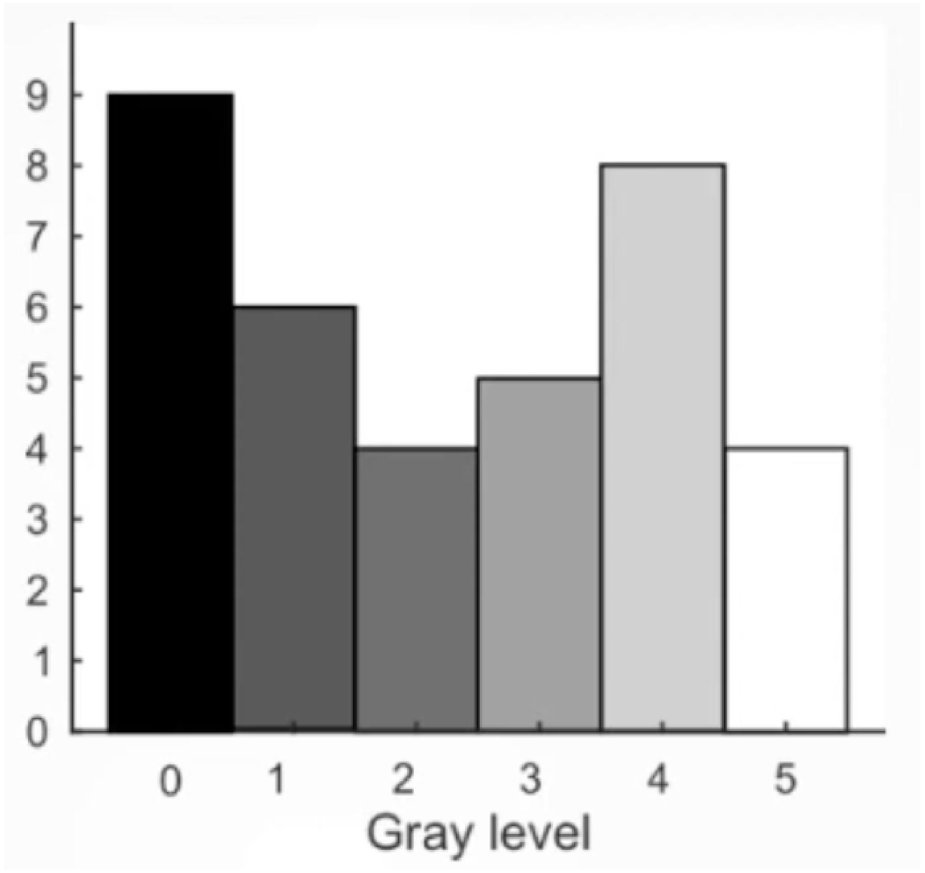
Where

- W_b is the number of pixels in background/tot. Number of pixels
- W_f is the number of pixels in foreground/tot. Number of pixels
- $\mu_b$is the mean intensity of the background

Otsu method

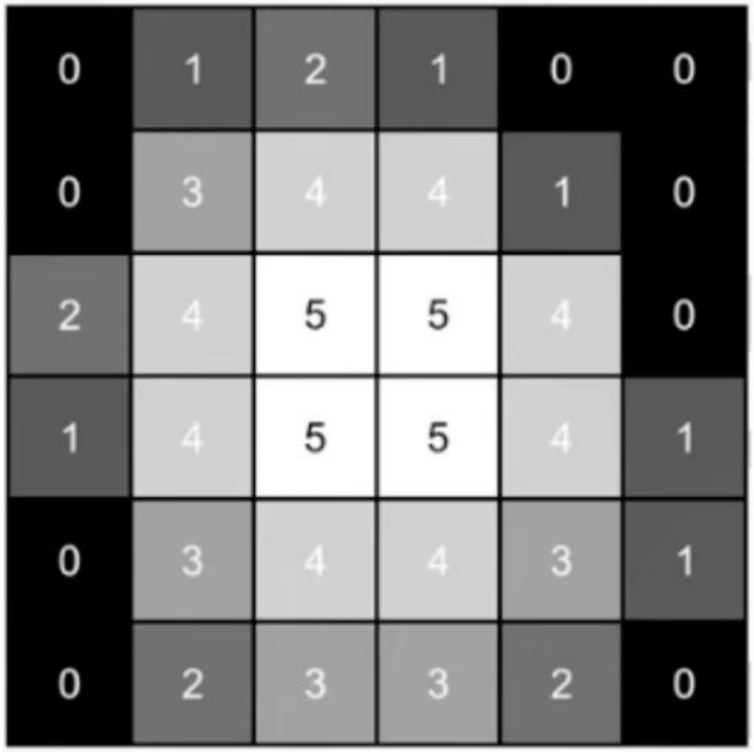


0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

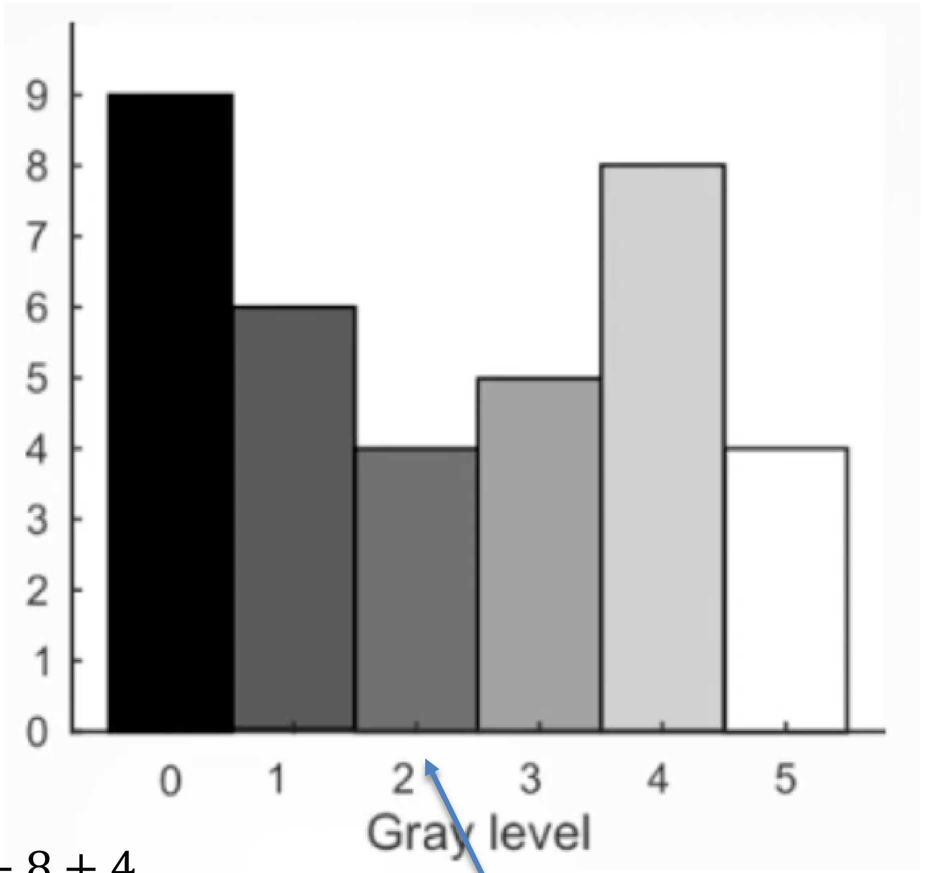


Otsu method

$$\sigma_B^2 = W_b * W_f\left(\mu_b + \ \mu_f\right)^2$$



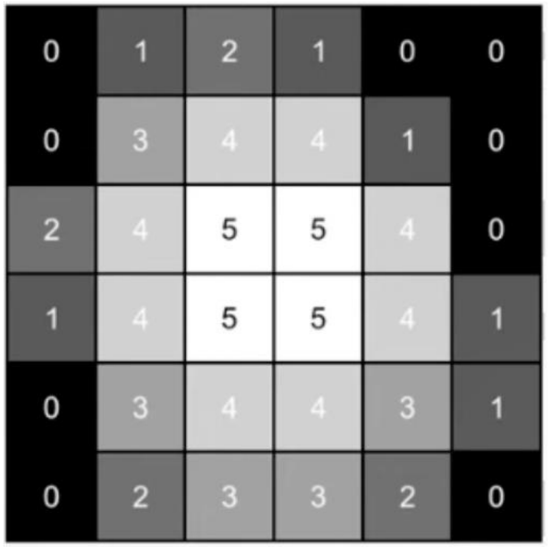
0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0



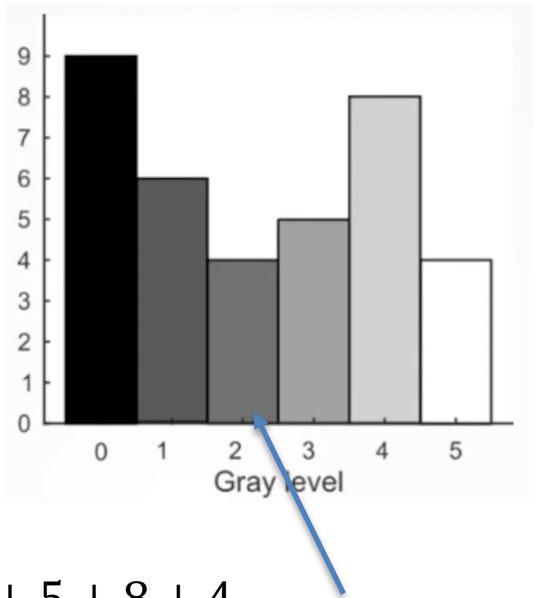
$$W_b = \frac{9+6}{36} = 0.42 \qquad W_f = \frac{4+5+8+4}{36} = 0.58$$

$$\mu_b = \frac{9*0+6*1}{9+6} = 0.4 \qquad \mu_f = \frac{4*2+5*3+8*4+4*4}{4+5+8+4} = 3.57$$

Otsu method



0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

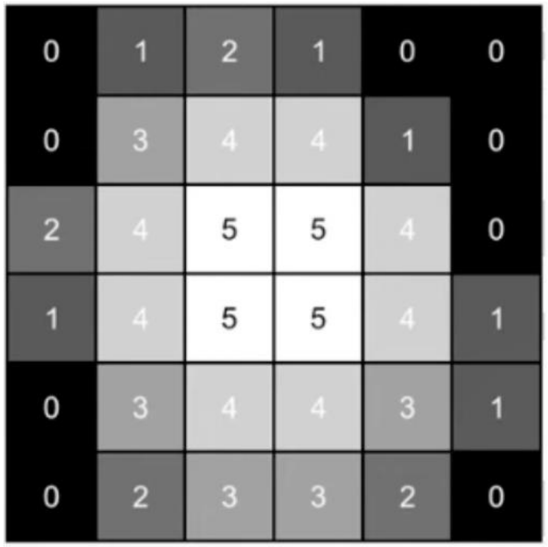


$$W_b = \frac{9+6}{36} = 0.42 \qquad W_f = \frac{4+5+8+4}{36} = 0.58$$

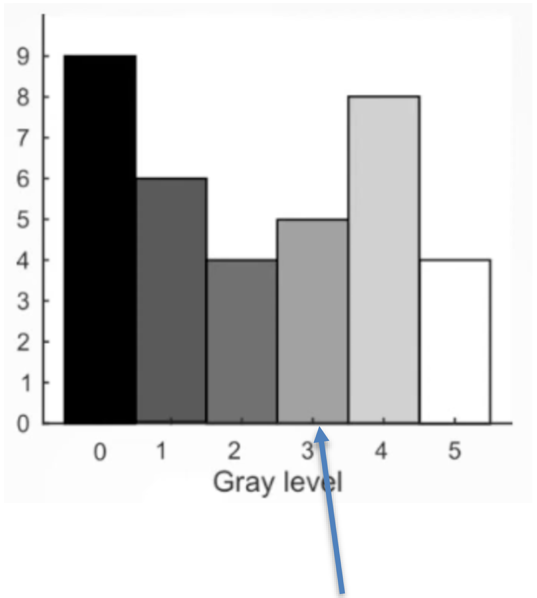
$$\mu_b = \frac{9*0+6*1}{9+6} = 0.4 \qquad \mu_f = \frac{4*2+5*3+8*4+4*4}{4+5+8+4} = 3.57$$

$$\sigma_B^2 = W_b * W_f\left(\mu_b + \mu_f\right)^2 = 2.44$$

Otsu method

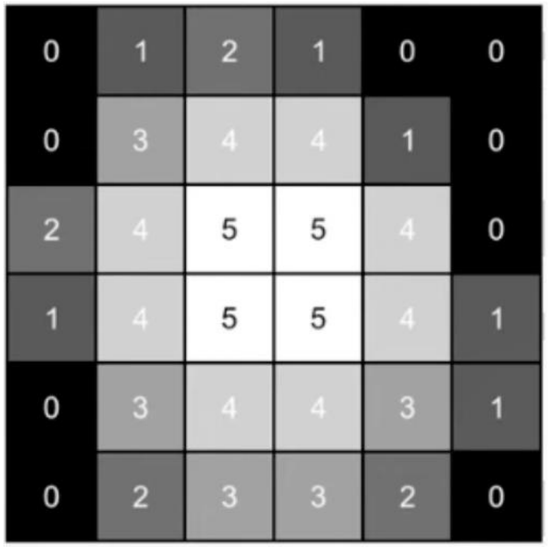


0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

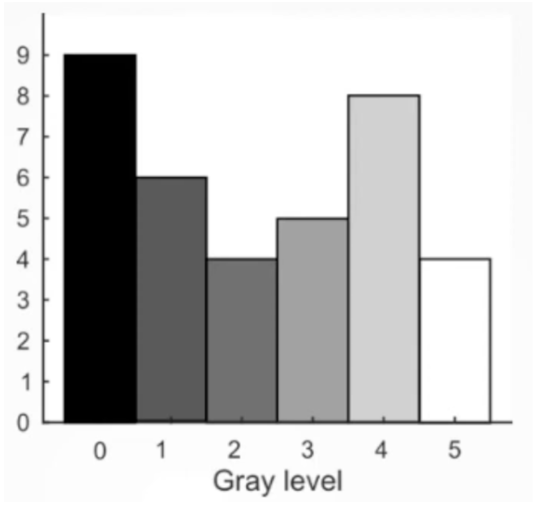


We repeat the calculus for all the possible thresholds

Otsu method

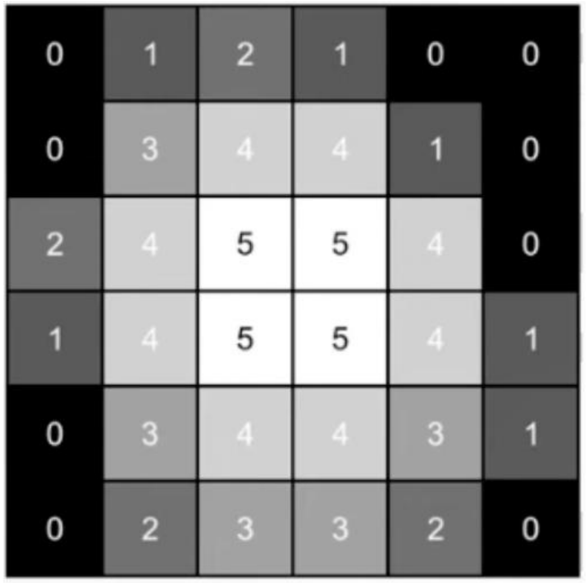


0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0

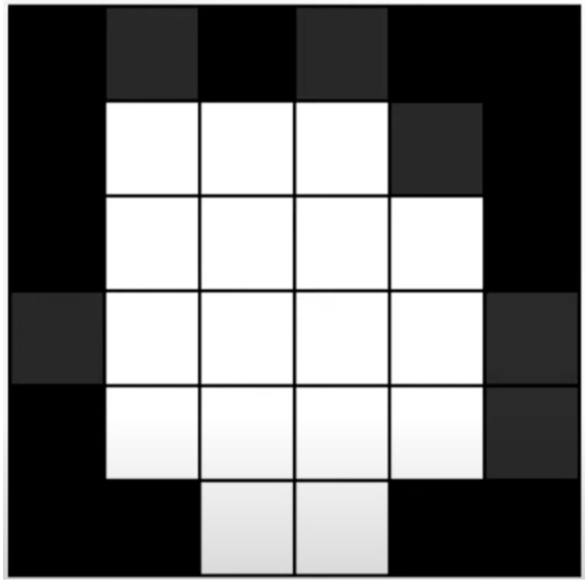


I_t	0	1	2	3	4	5
W_b	0	0.25	0.42	0.53	0.67	0.89
μ_b	0	0	0.40	0.74	1.21	1.91
W_f	1	0.75	0.58	0.47	0.33	0.11
μ_f	2.25	3.00	3.57	3.94	4.33	5.00
σ_b^2	0	1.69	2.44	2.56	2.17	0.95

Otsu method



0	1	2	1	0	0
0	3	4	4	1	0
2	4	5	5	4	0
1	4	5	5	4	1
0	3	4	4	3	1
0	2	3	3	2	0



Global thresholding: problem

Global models works well when the background and the subjects have distinct intensity distributions.

When there are different lighting conditions or complex intensity distributions, global thresholding models may not be enough.

Local thresholding

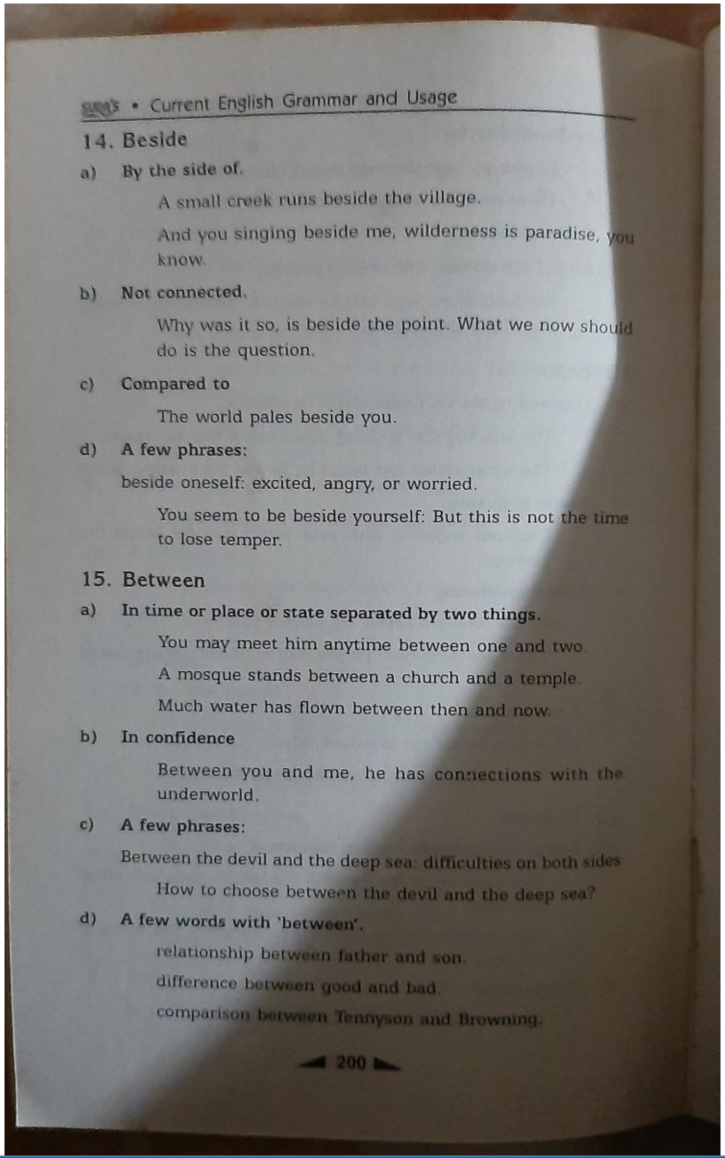
- Always thresholding but it considers smaller regions in the image, the vicinity of a point
- Works better w.r.t. global methods when the light changes
- More sensible to noise and rumors
- More computational resources

Mean and Gaussian adaptive thresholding

It calculates the threshold value for each sub-region by taking the average intensity (or weighted average) of all pixels within a region.

OpenCV is one of the most popular libraries for this kind of image manipulation.

Mean thresholding example



Sura's • Current English Grammar and Usage

14. Beside

a) By the side of.

A small creek runs beside the village.

And you singing beside me, wilderness is paradise, you know.

b) Not connected.

Why was it so, is beside the point. What we now should do is the question.

c) Compared to

The world pales beside you.

d) A few phrases:

beside oneself: excited, angry, or worried.

You seem to be beside yourself: But this is not the time to lose temper.

15. Between

a) In time or place or state separated by two things.

You may meet him anytime between one and two.

A mosque stands between a church and a temple.

Much water has flown between then and now.

b) In confidence

Between you and me, he has connections with the underworld.

c) A few phrases:

Between the devil and the deep sea: difficulties on both sides

How to choose between the devil and the deep sea?

d) A few words with 'between'.

relationship between father and son.

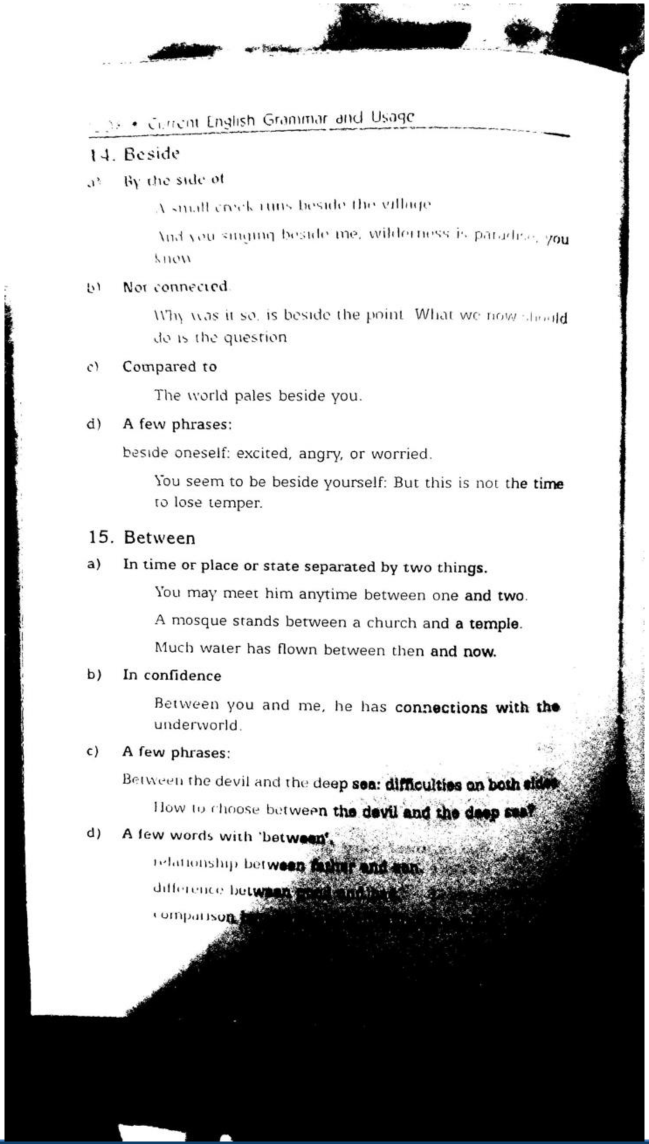
difference between good and bad.

comparison between Tennyson and Browning.

◄ 200 ►

Mean thresholding example

Global thresholding



[illegible] • Current English Grammar and Usage

14. Beside

a) By the side of

A small creek runs beside the village

And you singing beside me, wilderness is paradise, you know

b) Not connected

Why was it so, is beside the point. What we now should do is the question

c) Compared to

The world pales beside you.

d) A few phrases:

beside oneself: excited, angry, or worried.

You seem to be beside yourself: But this is not the time to lose temper.

15. Between

a) In time or place or state separated by two things.

You may meet him anytime between one and two.

A mosque stands between a church and a temple.

Much water has flown between then and now.

b) In confidence

Between you and me, he has connections with the underworld.

c) A few phrases:

Between the devil and the deep sea: difficulties on both sides

How to choose between the devil and the deep sea?

d) A few words with 'between'.

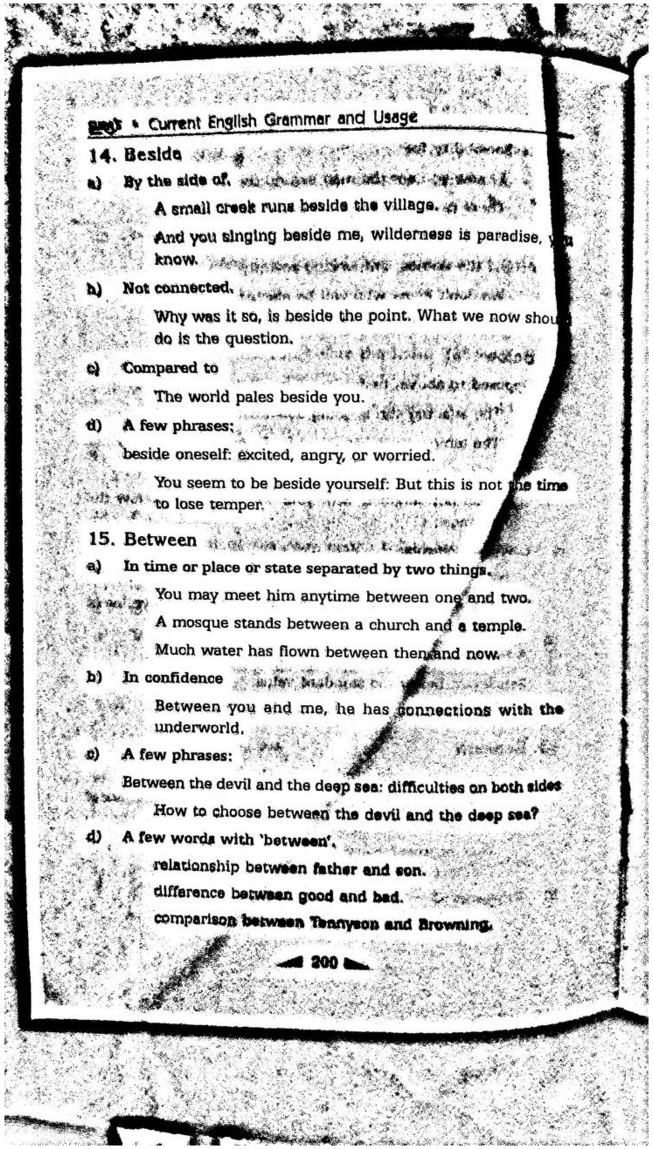
relationship between father and son.

difference between [illegible]

comparison [illegible]

Mean thresholding example

Local thresholding



Current English Grammar and Usage

14. Beside

a) By the side of.

A small creek runs beside the village.

And you singing beside me, wilderness is paradise, you know.

b) Not connected.

Why was it so, is beside the point. What we now shou do is the question.

c) Compared to

The world pales beside you.

d) A few phrases:

beside oneself: excited, angry, or worried.

You seem to be beside yourself: But this is not the time to lose temper.

15. Between

a) In time or place or state separated by two things.

You may meet him anytime between one and two.

A mosque stands between a church and a temple.

Much water has flown between then and now.

b) In confidence

Between you and me, he has connections with the underworld.

c) A few phrases:

Between the devil and the deep sea: difficulties on both sides

How to choose between the devil and the deep sea?

d) A few words with 'between'.

relationship between father and son.

difference between good and bad.

comparison between Tennyson and Browning.

200

Aritmetric Mean adaptive thresholding

Objective: analyze the image statistically in local regions

1. Choose a dimension for the sub-regions
2. For each sub-regions, all the pixels contribute equally to the final value
3. Apply the formula: $T = mean(I_L) - C$, where I_L is the local sub-region of the image, C is a constant value that can be used to fine tune T (threshold)

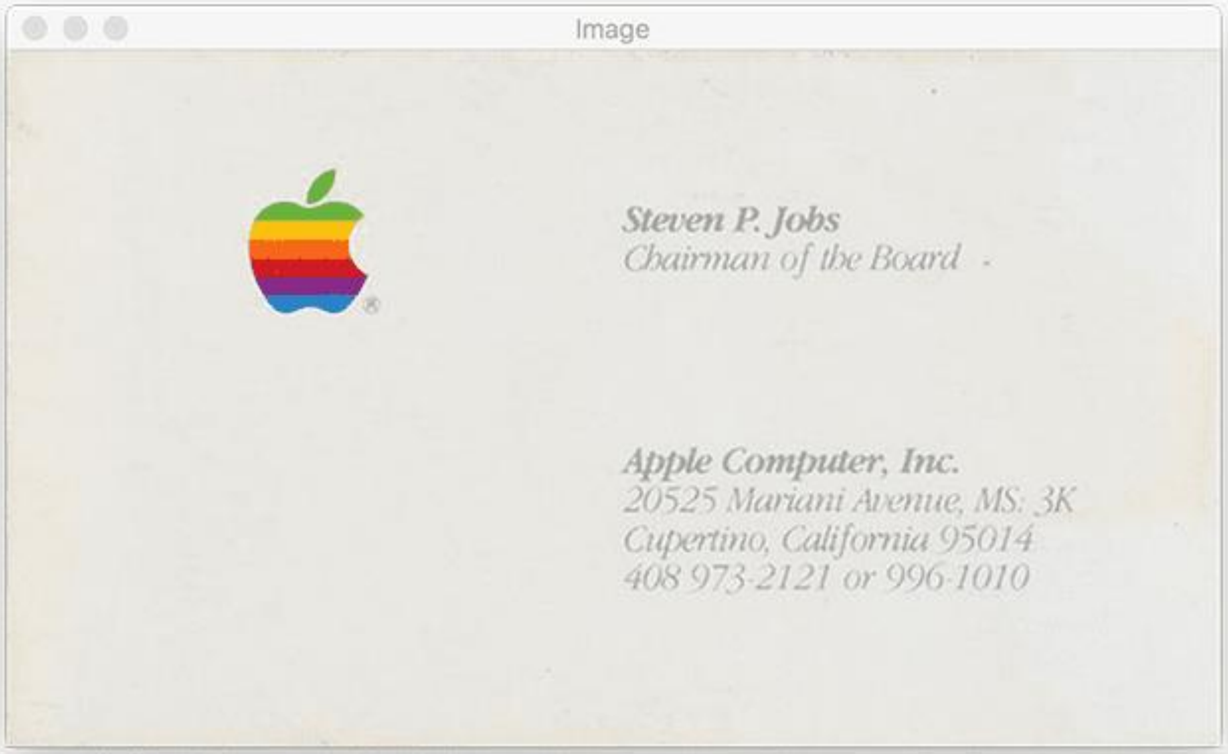
Gaussian adaptive thresholding

Objective: analyze the image statistically in local regions

1. Choose a dimension for the sub-regions
2. Each sub-region has a center pixel with (x,y) as coordinate
3. Apply the formula: $T = mean(I_L) - C$, where I_L is the local sub-region of the image, C is a constant value that can be used to fine tune T (threshold), the mean is the gaussian mean, in which pixels farther away from the (x,y) coordinate center of the image contributes less.

Thresholding wrap up

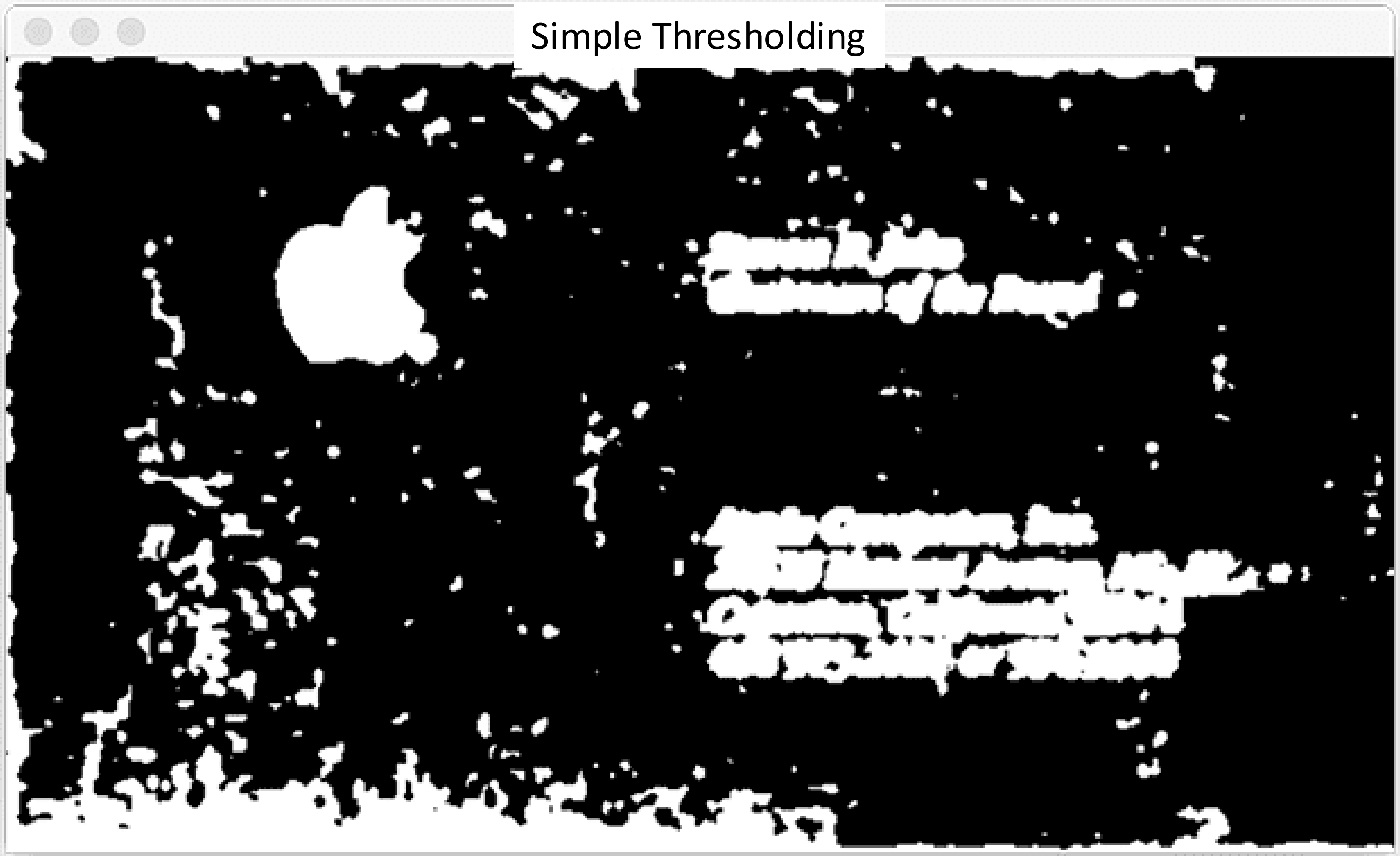
Original image



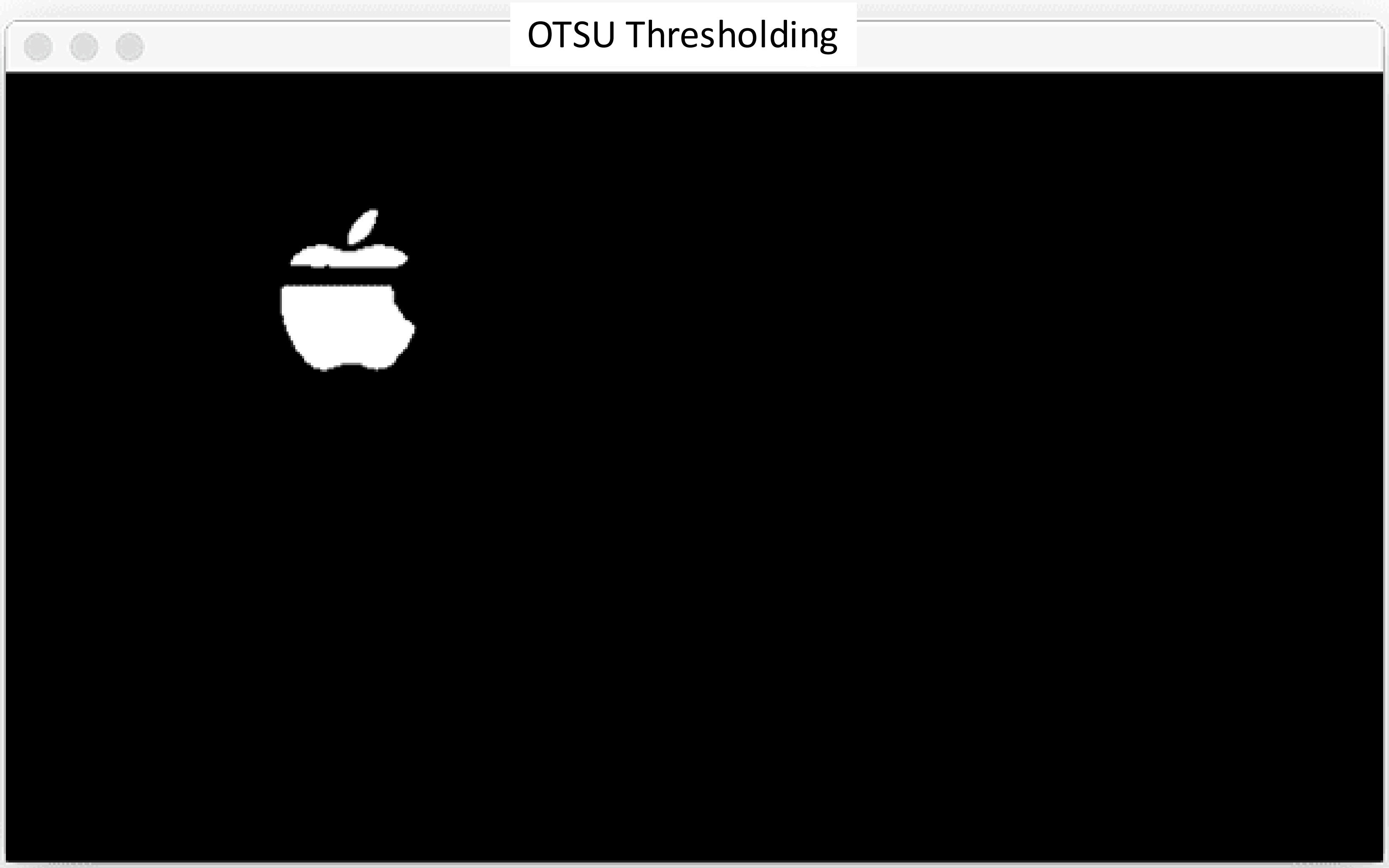
Steven P. Jobs
Chairman of the Board

Apple Computer, Inc.
20525 Mariani Avenue, MS: 3K
Cupertino, California 95014
408 973-2121 or 996-1010

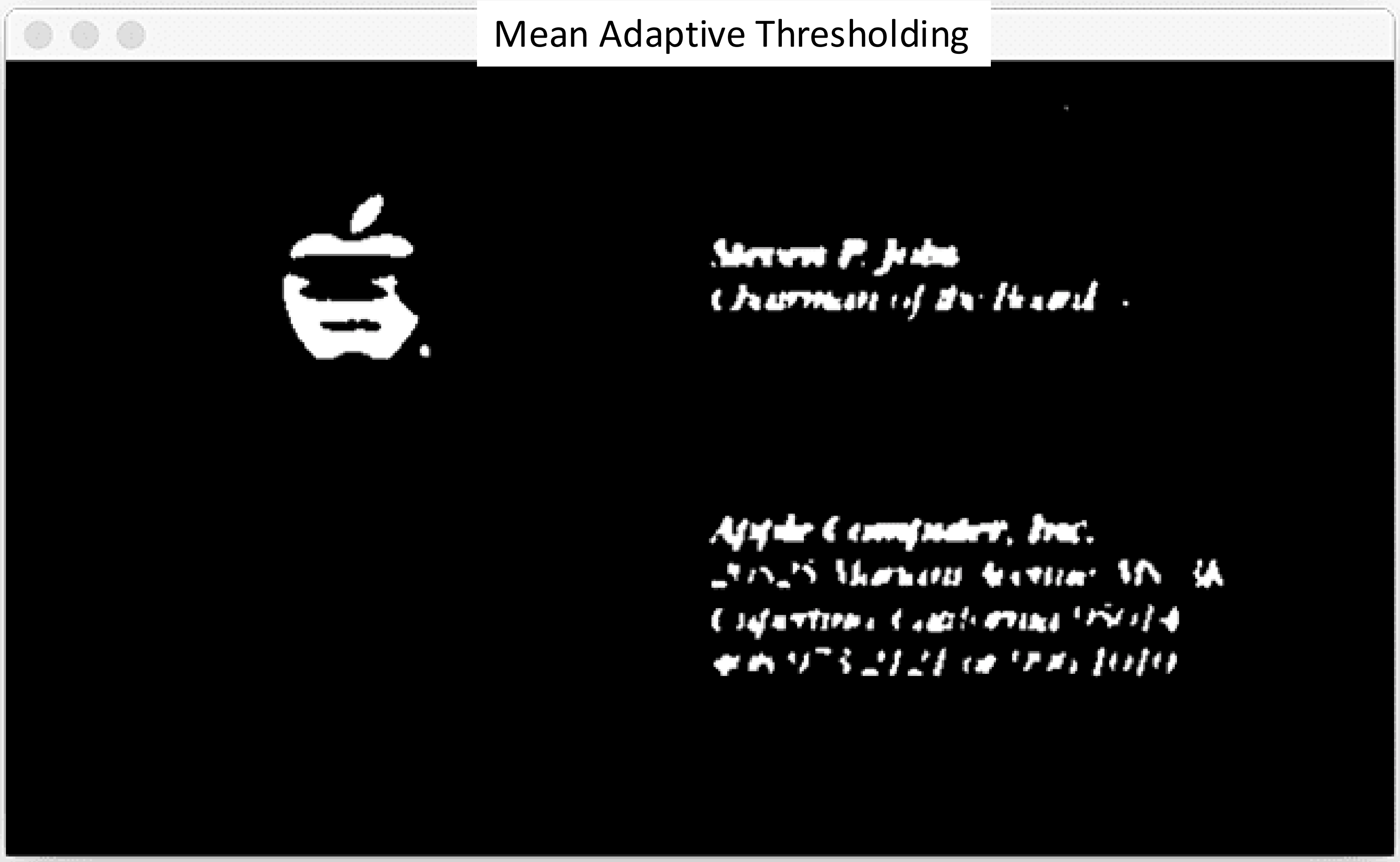
Thresholding wrap up



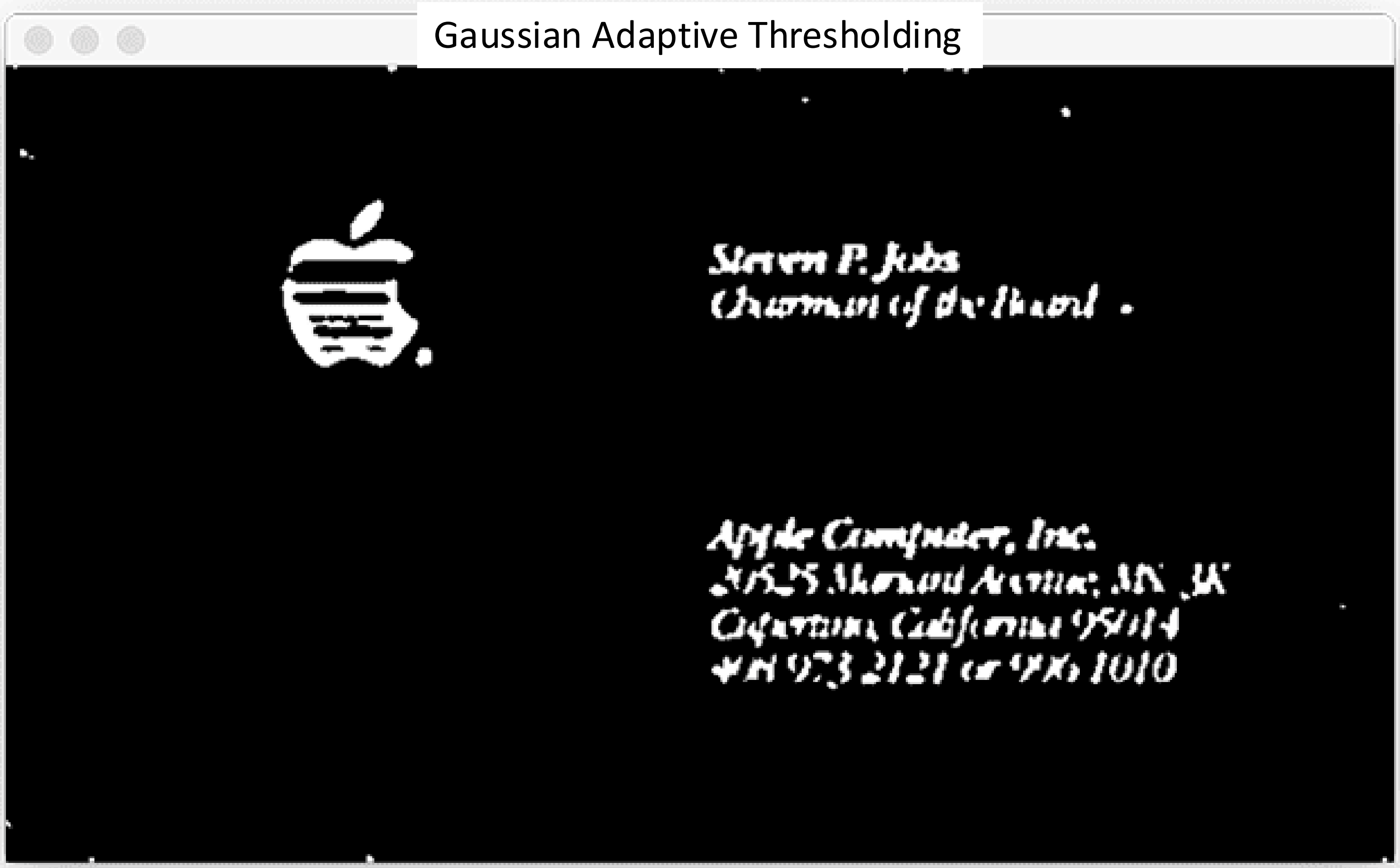
Thresholding wrap up



Thresholding wrap up



Thresholding wrap up



Steven P. Jobs
Chairman of the Board

Apple Computer, Inc.
20525 Mariani Avenue, MS 3K
Cupertino, California 95014
408 973 2121 or 996 1010

Thresholding wrap up

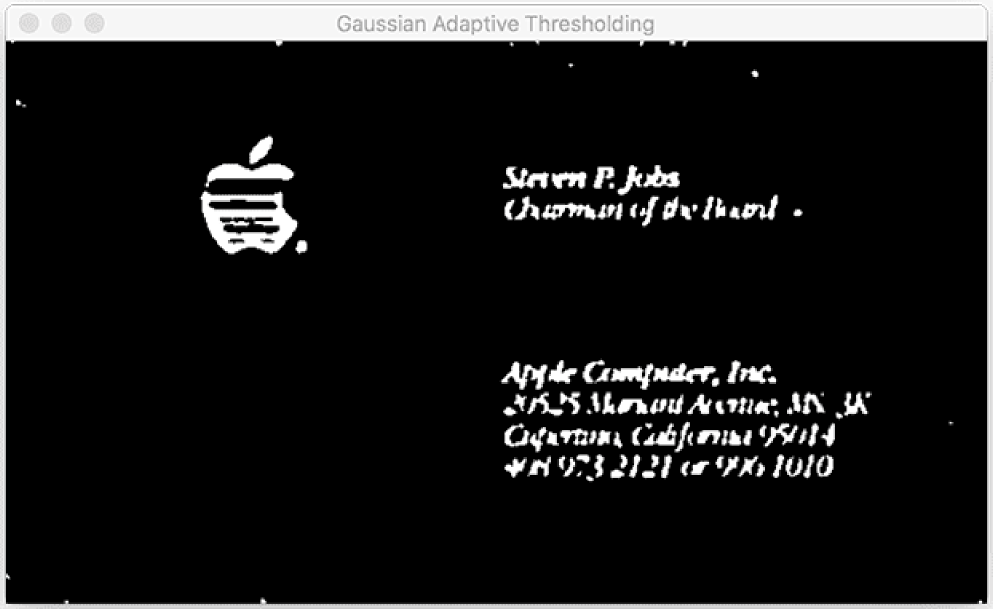
We saw several methods to detect objects in an image based on thresholding operations.
We have:

1. Global methods, in which the threshold is one for the entire image
2. Local methods, in which the image is split in sub-regions, each with its own threshold

Pros/Cons:
Global methods may fail when we have different lights in the image
Local methods require the selection of the dimension of the sub-image, how to do it?

Thresholding wrap up

Thresholding is not enough, even after the application of it, the image is not perfect.
We can do more to make the image better.
For starters, we can apply morphological operations.



Morphological operations

These kinds of operations looks at the shapes and can be applied to clean the image:

- Translation
- Reflection
- Erosion
- Dilation
- Closing
- Morphological gradient
- Black hat
- Top hat

Morphological operations

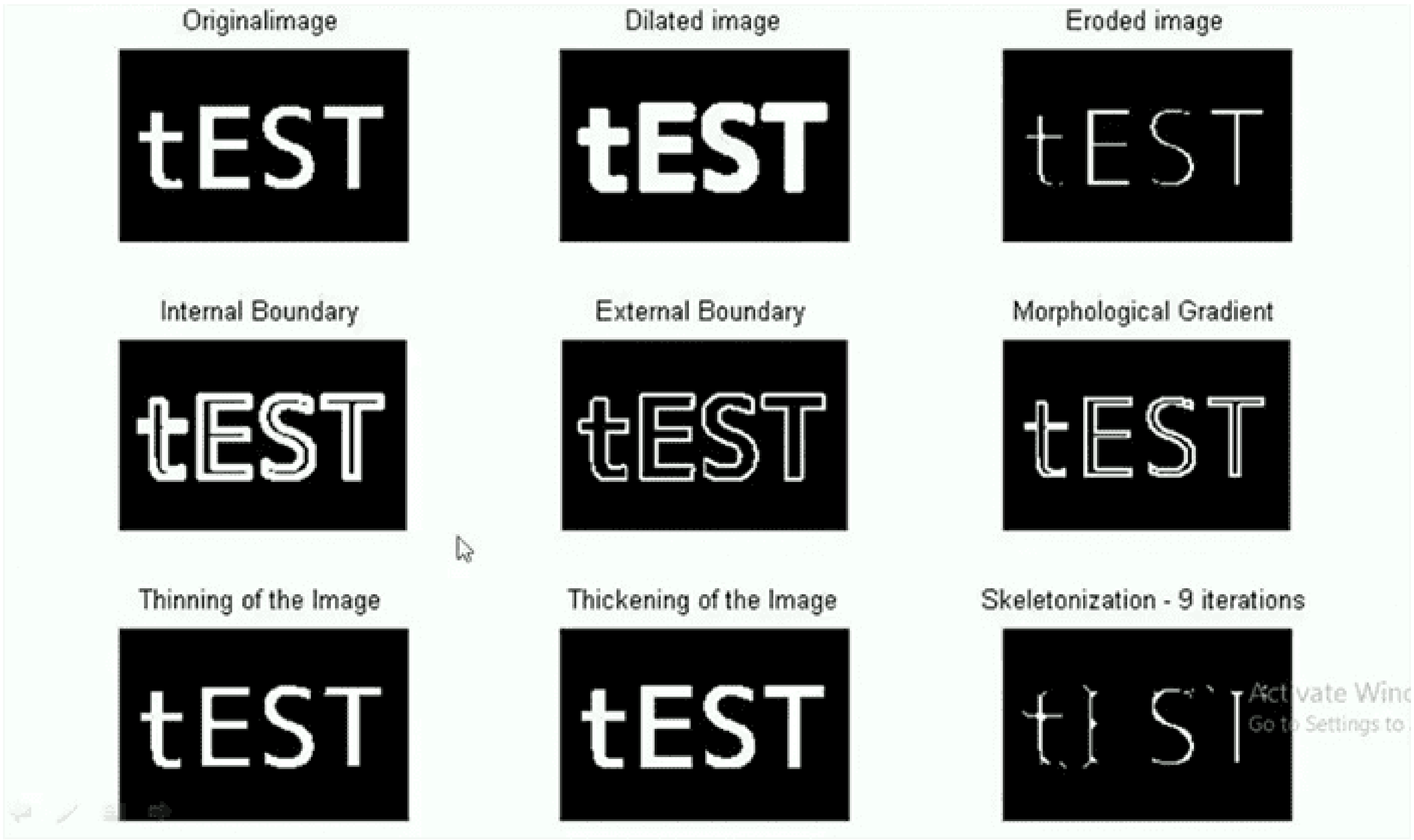
They can be applied to black and white images or gray scale.

They are applied to the shapes and the structures of the images.

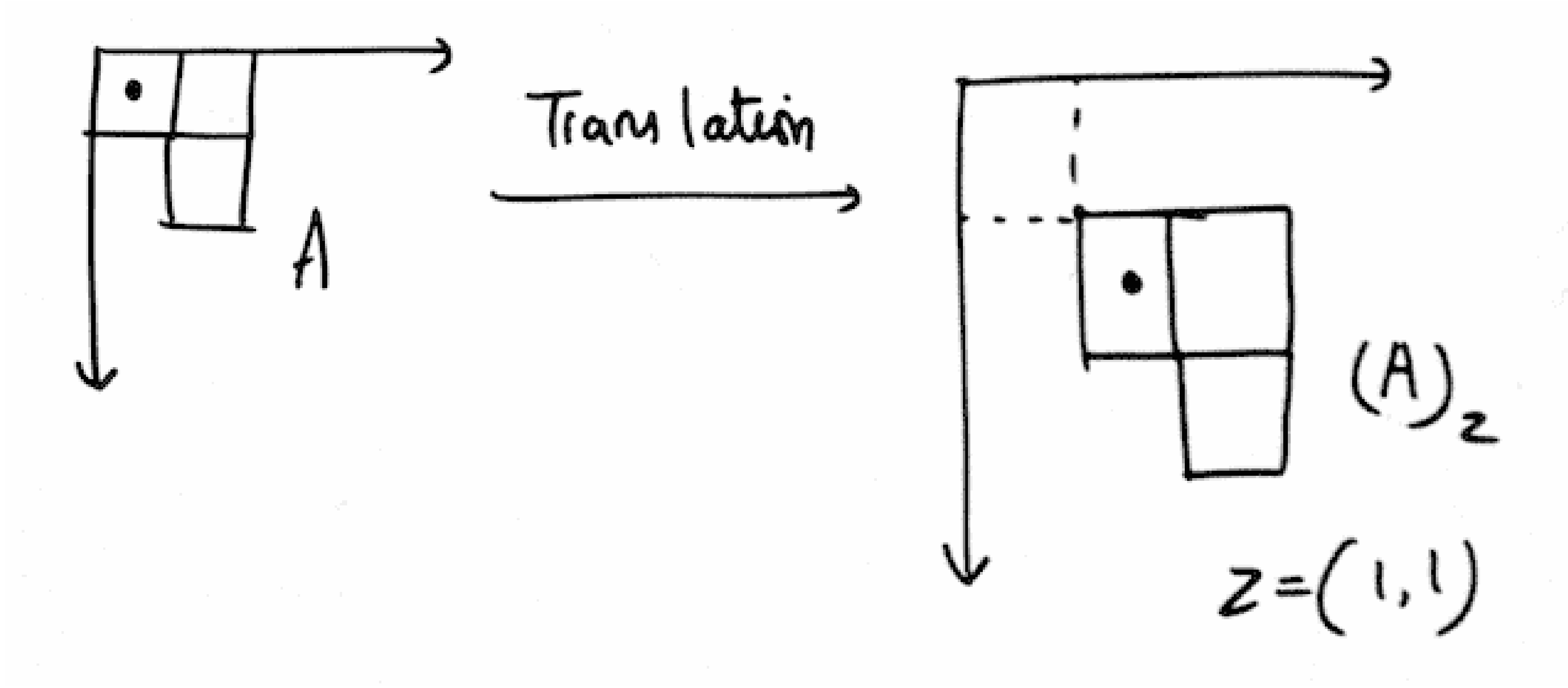
We can:

- Increase the size of the objects
- Decrease the size of the objects
- Close gaps
- Open gaps

Morphological operations

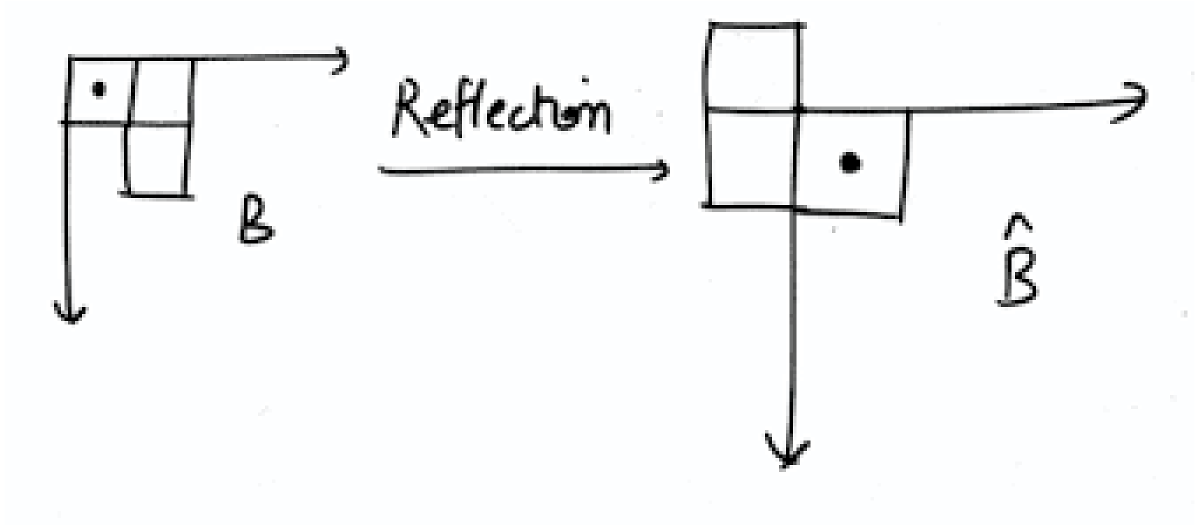


Translation



From (x,y) -> (x+z1, y+z2)

Reflection



From (x,y) -> (-x, -y)

Linear Shift Invariance System (LSIS)

$$f(x) \rightarrow LSIS \rightarrow g(x)$$

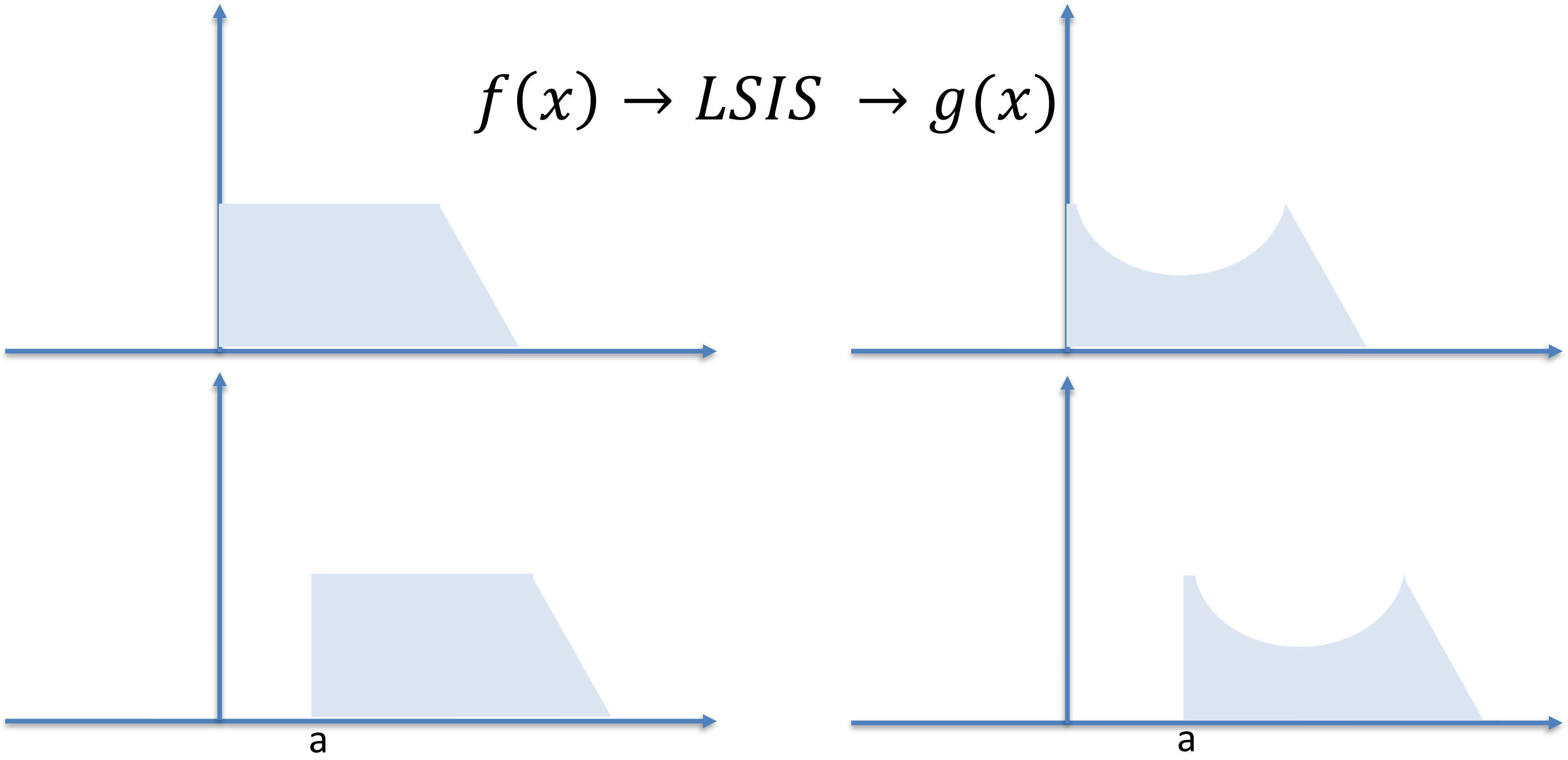
$$f_1(x) \rightarrow LSIS \rightarrow g_1(x) \qquad\qquad f_2(x) \rightarrow LSIS \rightarrow g_2(x)$$

$$\alpha f_1 + \beta f_2 \rightarrow LSIS \rightarrow \alpha g_1 + \beta g_2$$

It is linear, meaning that the sum of the input gives you the sum of the output.

Linear Shift Invariance System (LSIS)

$$f(x) \rightarrow LSIS \rightarrow g(x)$$



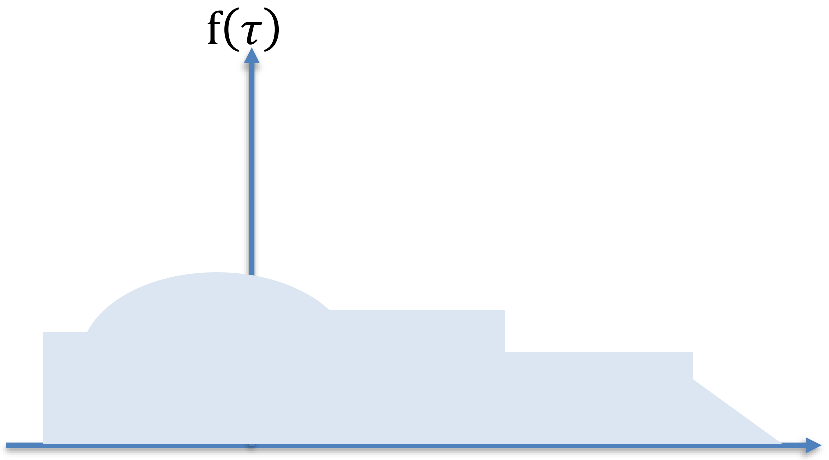
$$f(x-a) \rightarrow LSIS \rightarrow g(x-a)$$

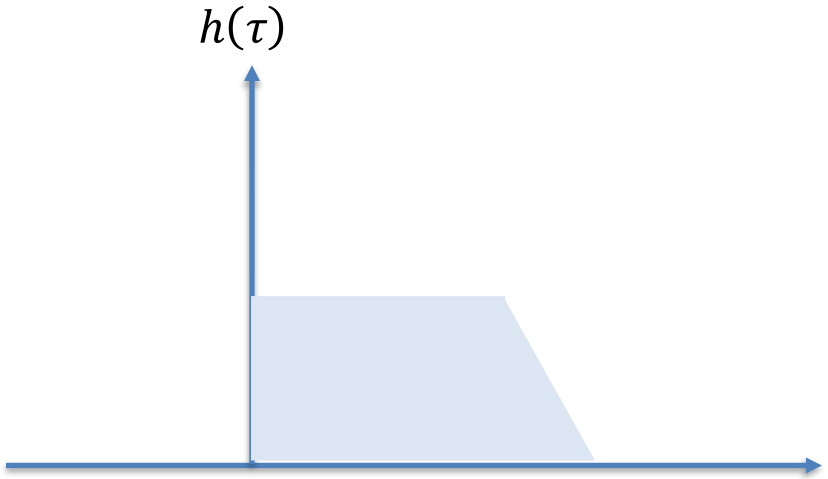
Any LSIS is a convolution!

Convolution

Convolution of 2 functions f(x) and h(x) is:

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)d\tau$$



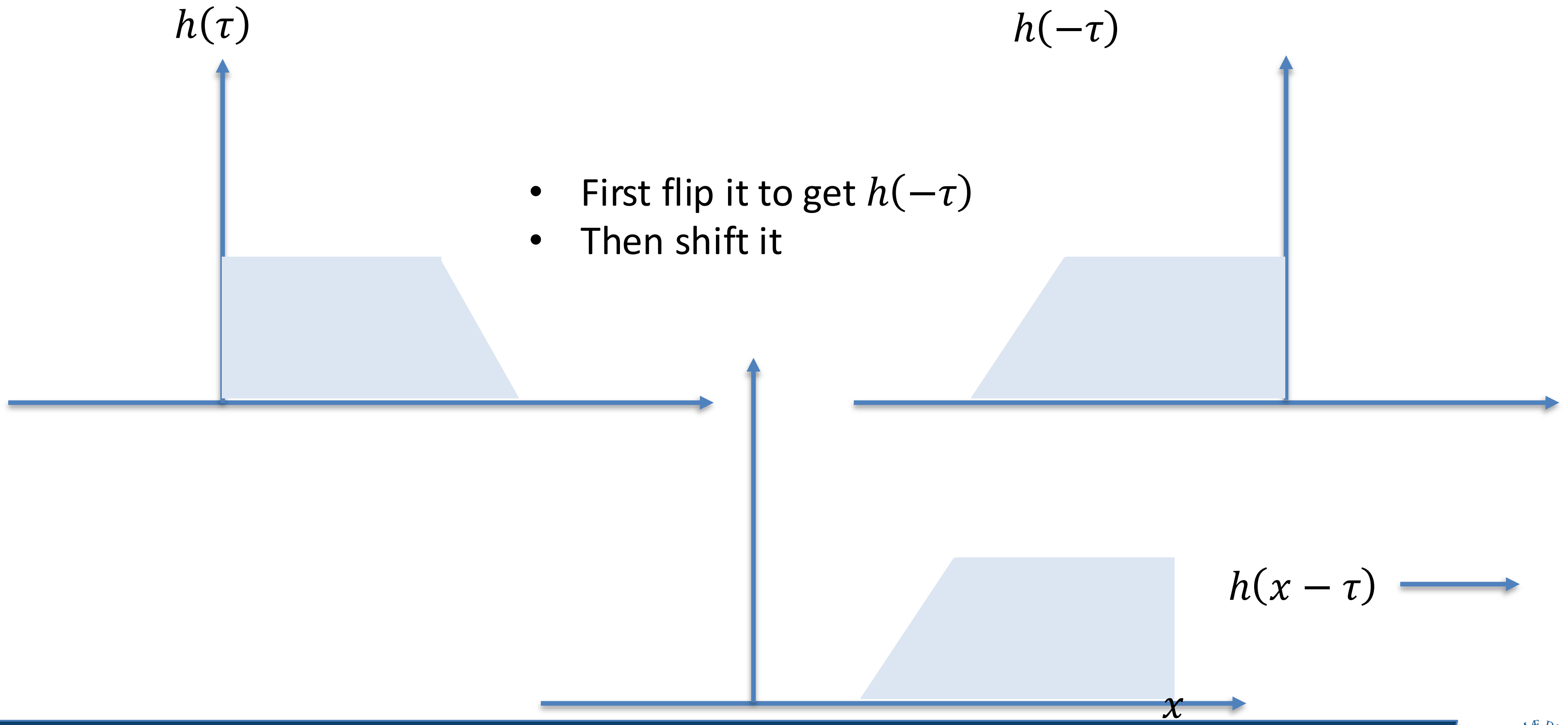


To apply the integral, I first need to obtain: $h(x-\tau)$

Convolution

Convolution of 2 functions f(x) and h(x) is:

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)d\tau$$

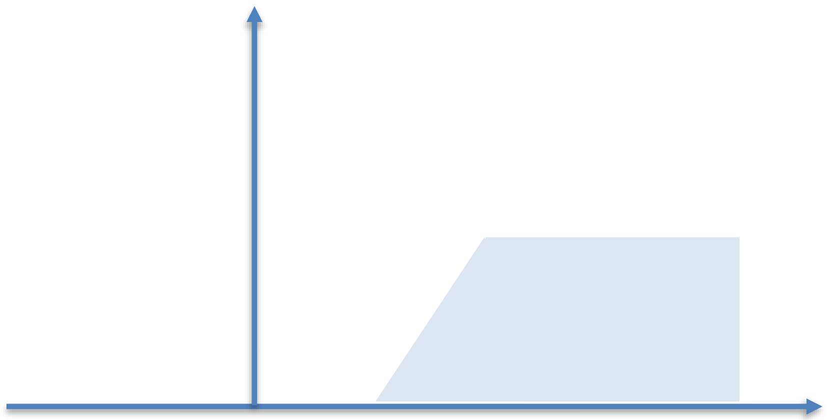


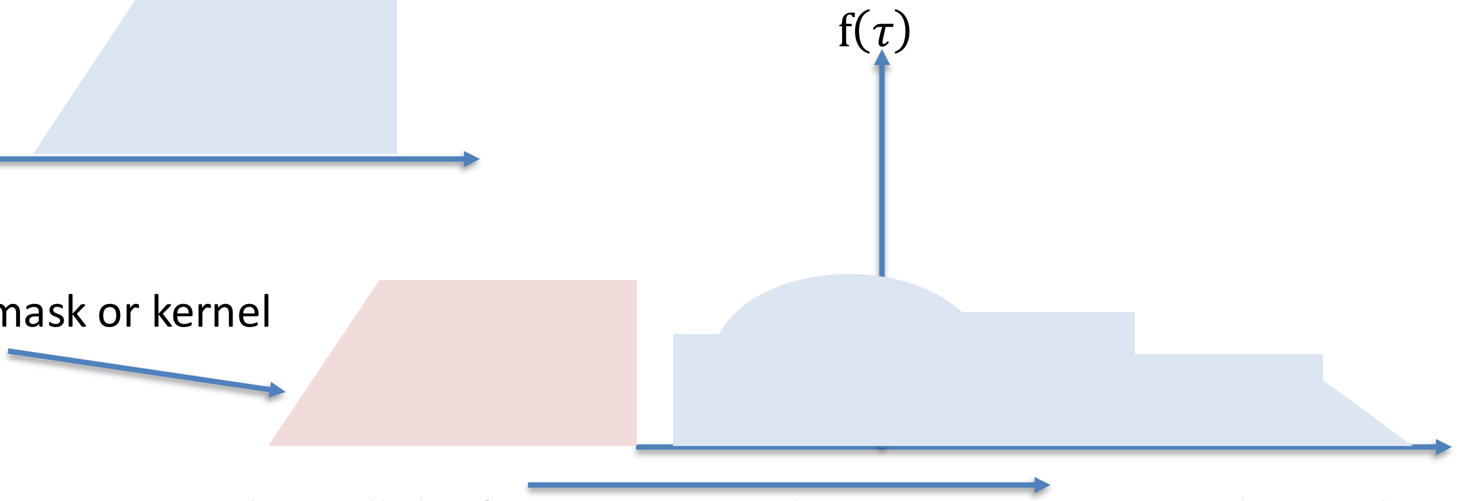
- First flip it to get $h(-\tau)$
- Then shift it

Convolution

Convolution of 2 functions f(x) and h(x) is:

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)d\tau$$





Often called mask or kernel

Apply to all the function to get the product and integral at each point

Convolution

Any LSIS implies convolution and convolution implies LSIS.

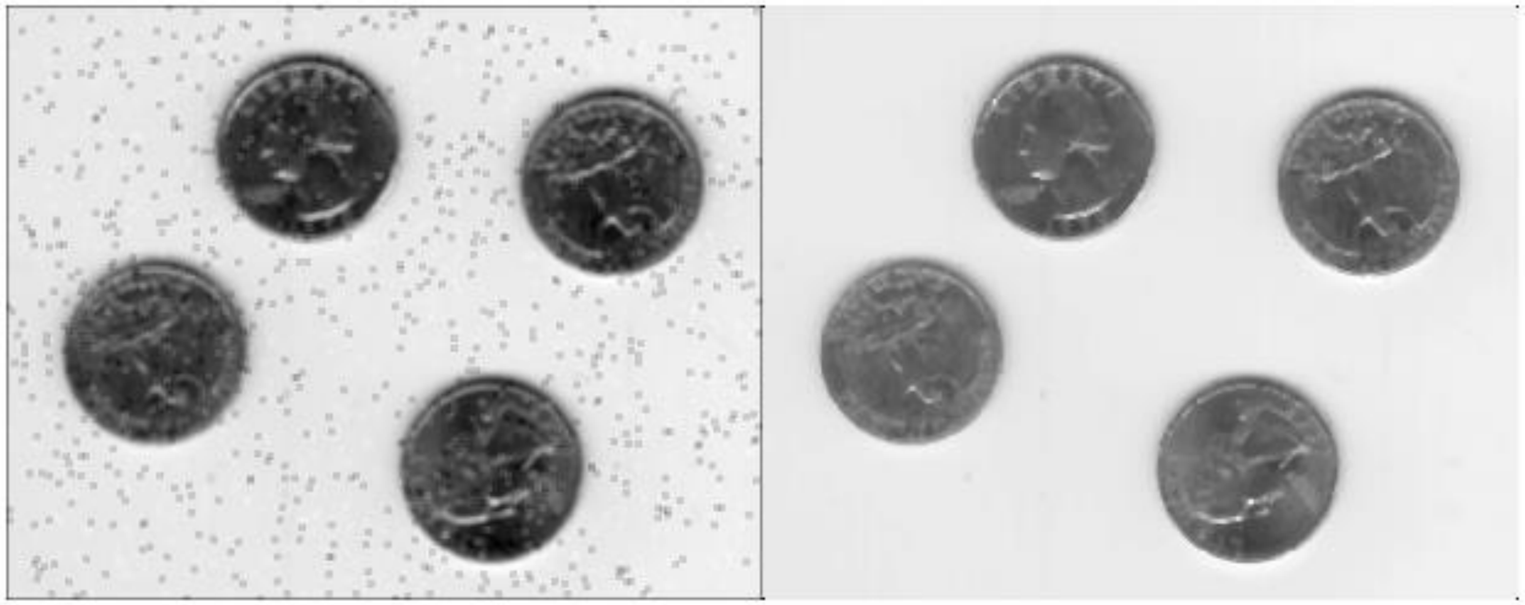
Properties:

- Commutative
- Associative
- Cascaded system, instead of applying h() and g() one after the other, we can do h()*g() (convoluted)

Filtering

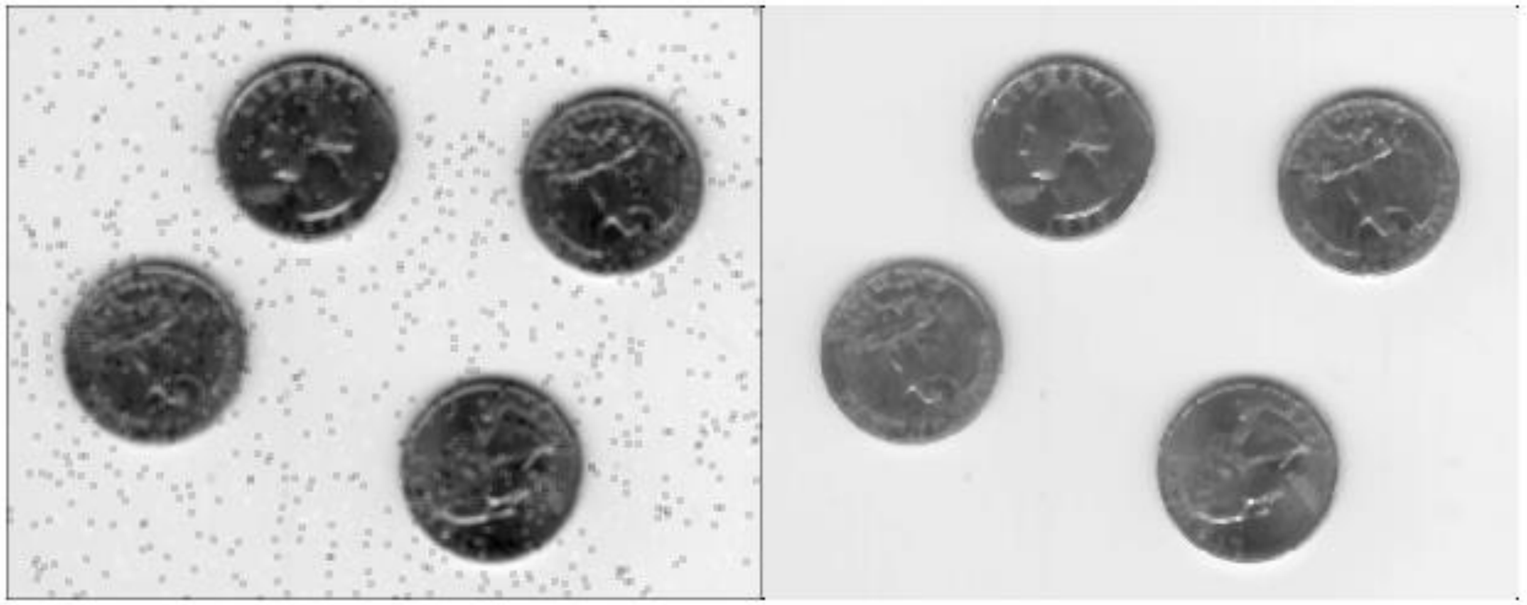
It is a technique for modifying or enhancing an image, such as smoothing, sharpening, edge enhancement.

They are neighbourhood operations, meaning that the final value of a pixel depend on its vicinity.



Noise removal

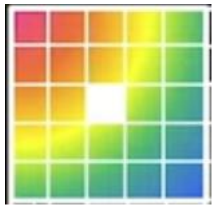
- Filtering strategies based on statistics
- Linear filtering
- Median filtering
- Adaptive filtering

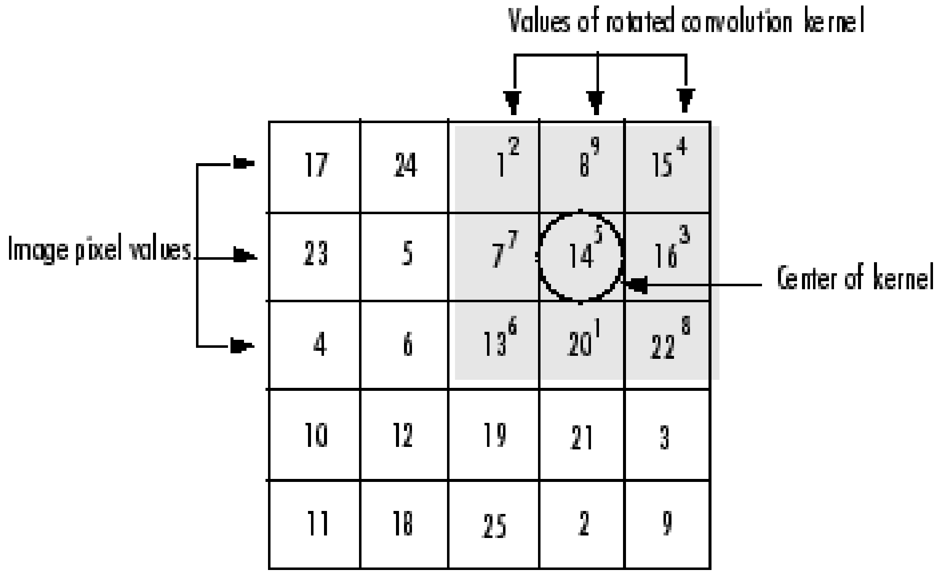


Linear filtering: Convolution

To perform linear filtering we apply the convolution: it is a neighbourhood operation in which each pixel is the weighted sum of the close pixels. The filter, also called convolution kernel, is the matrix of weights exploited to perform the filtering.

```
h = [8   1   6
     3   5   7
     4   9   2]
```



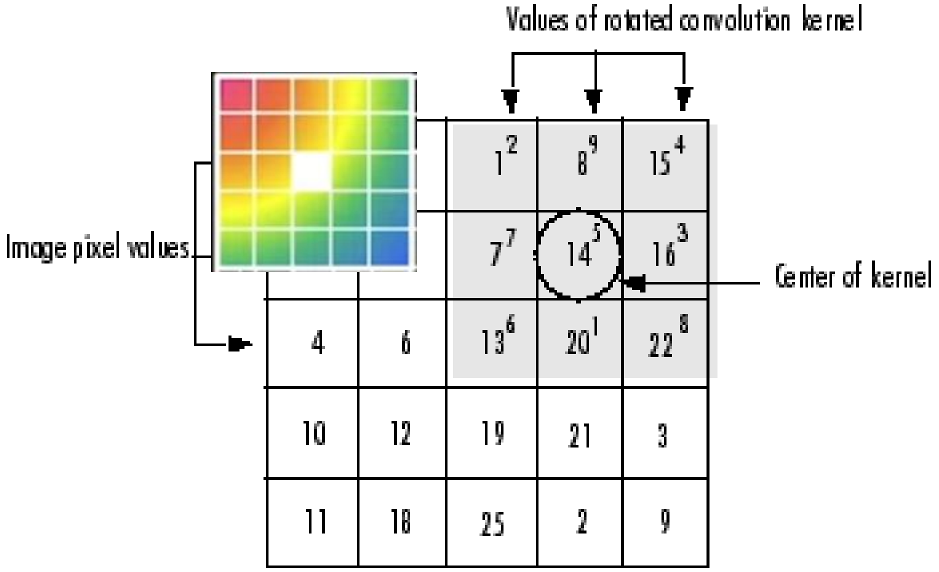


$$1 \cdot 2 + 8 \cdot 9 + 15 \cdot 4 + 7 \cdot 7 + 14 \cdot 5 + 16 \cdot 3 + 13 \cdot 6 + 20 \cdot 1 + 22 \cdot 8 = 575$$

Linear filtering: Convolution

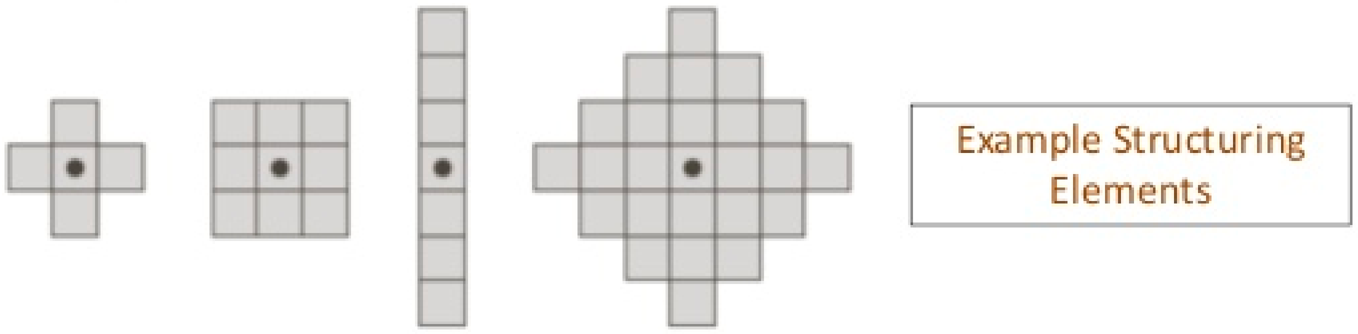
Problem with the border:

- Ignore the border
- Pad the input image with a costant value
- Pad with reflected image



Structuring element

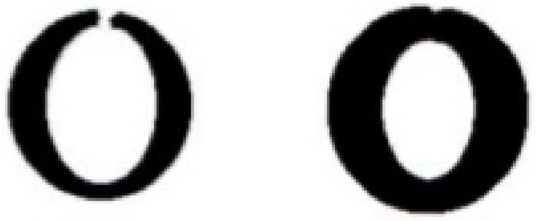
- Structuring elements are small shapes or sub-images used to examine the image, applying erosion or dilation.
- They may have any shape and/or size
- To process the image, we move the SE across the image pixel by pixel

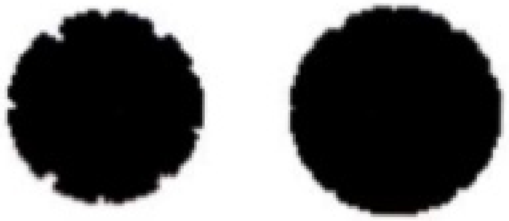


Dilation

- It enlarge the borders of the object/ increase the size of the foreground objects
- It can repair breaks
- It can repair intrusions
- It covers gaps due to high resolution

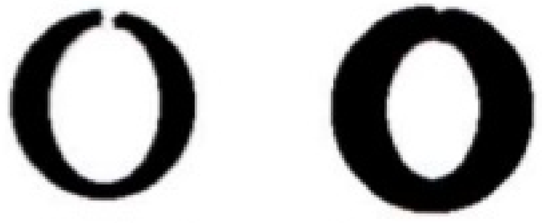
How to: if at least one of the pixels in the structuring element is >0.

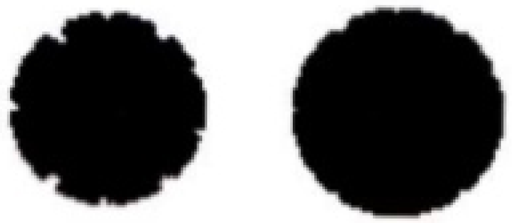




Dilation - properties

- It is commutative
- Associative
- If the kernel contains the origin, the dilation is a transformation that extends the original set
- It is a monotonic transformation

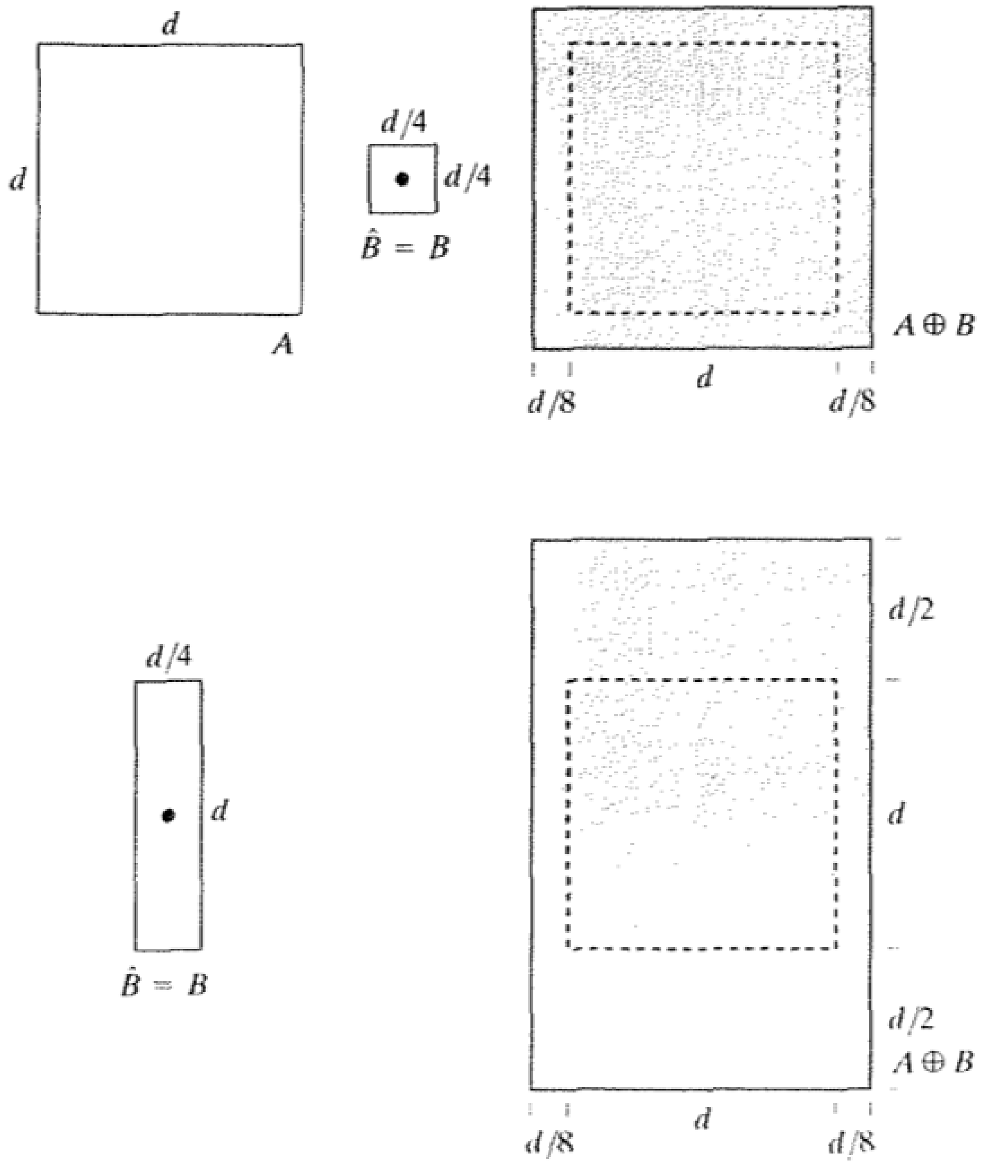




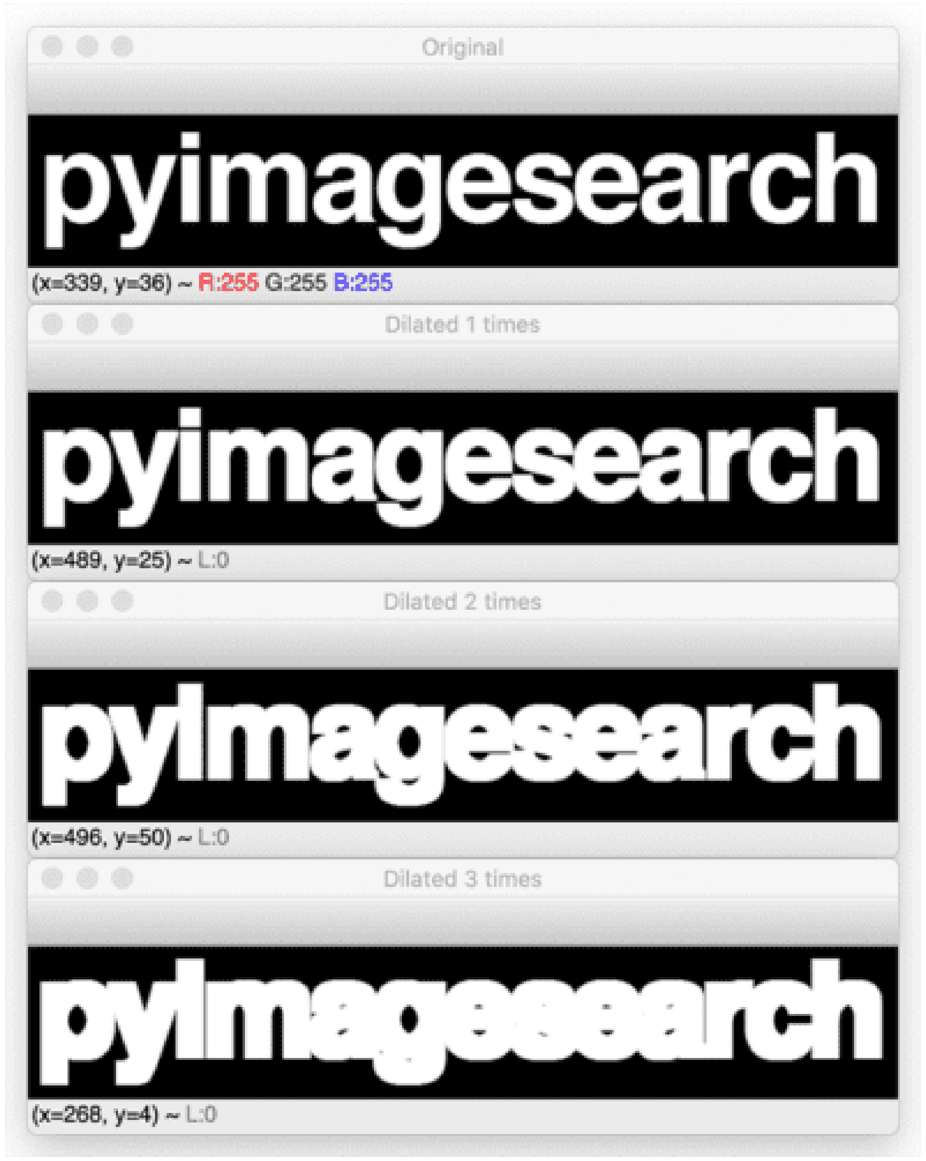
Dilation

a b c
d e

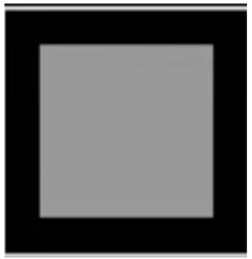
FIGURE 9.4
(a) Set A.
(b) Square structuring element (dot is the center).
(c) Dilation of A by B, shown shaded.
(d) Elongated structuring element.
(e) Dilation of A using this element.

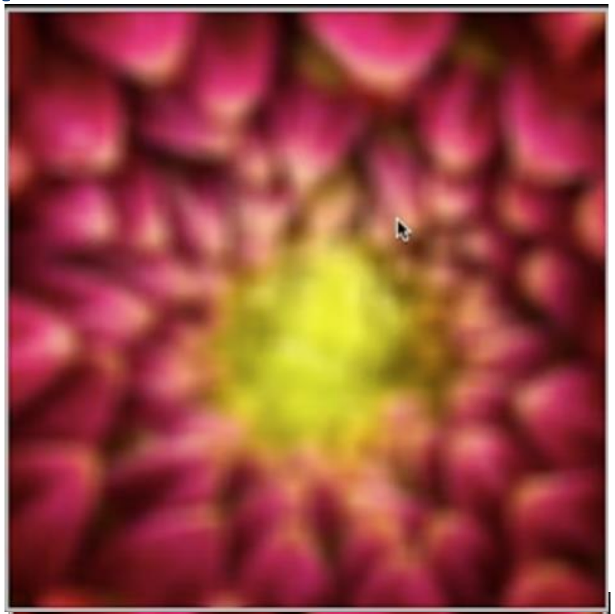


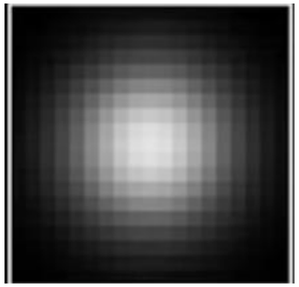
Dilation

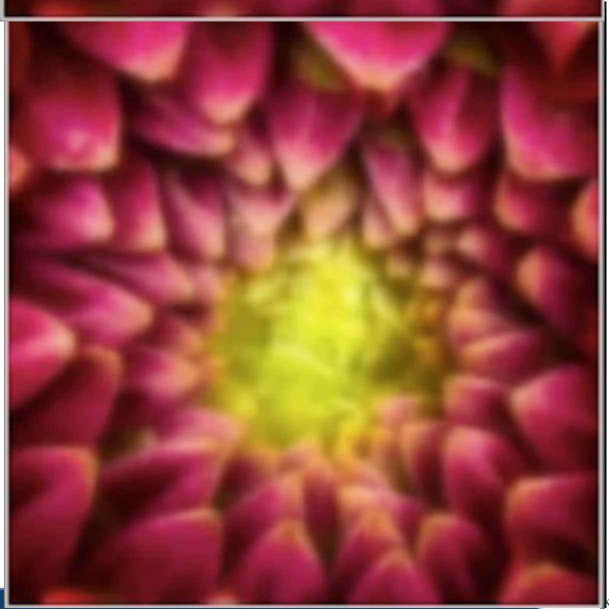

Original
pyimagesearch
(x=339, y=36) ~ R:255 G:255 B:255
Dilated 1 times
pyimagesearch
(x=489, y=25) ~ L:0
Dilated 2 times
pyimagesearch
(x=496, y=50) ~ L:0
Dilated 3 times
pyimagesearch
(x=268, y=4) ~ L:0

Linear filtering: Convolution

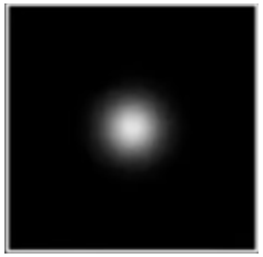

Box filter



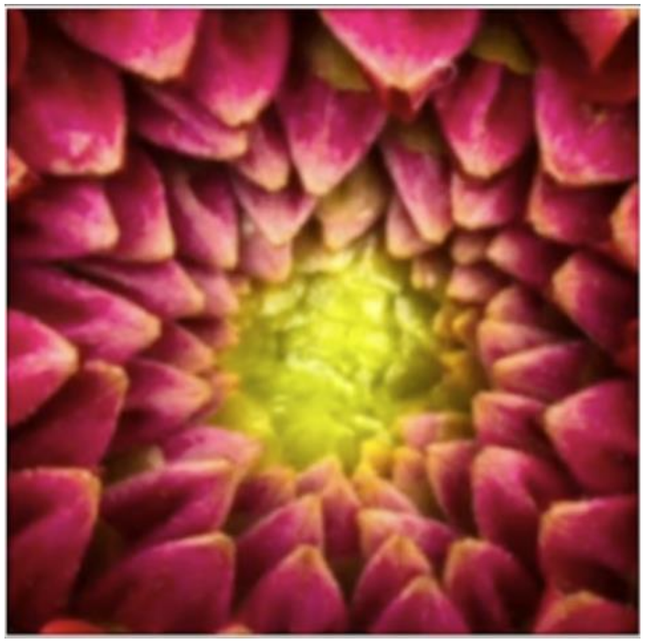

Fuzzy filter



Gaussian kernel - smoothing



Gaussian kernel

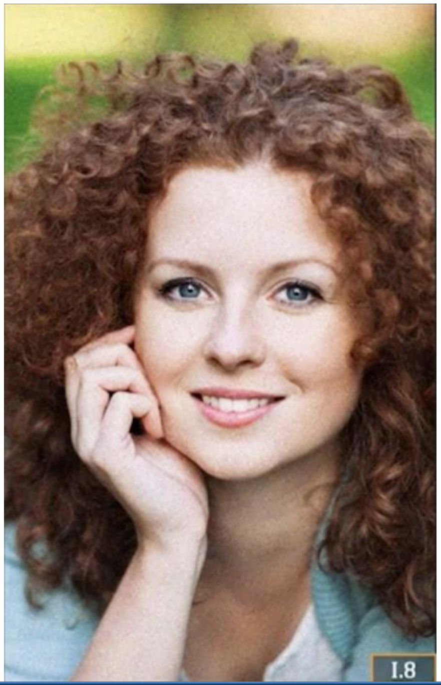


As you increase the value of sigma (the variance of the Gaussian kernel), you get smooth and smother images.

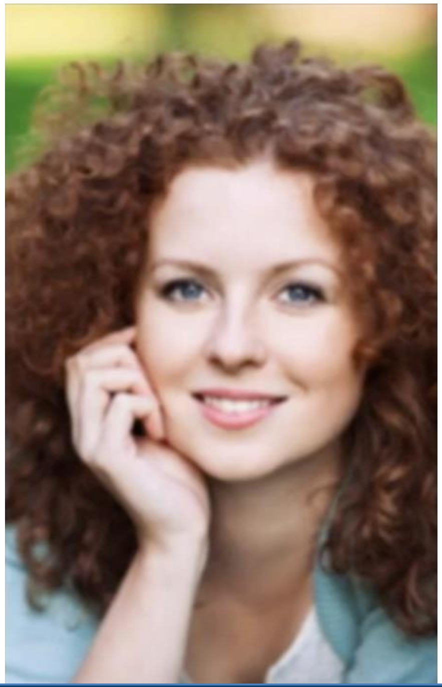
Gaussian kernel - smoothing

It smooths everything and this may be good for noise removal, but sometimes we do not want exactly this. In some cases, we want to take into account different brightness of the pixels as well.
We need a filter that changes along the way.
Hence, we need something a little bit more complicated: non linear filters.
One of the possible solutions is Bilateral filter: spatial + brightness gaussian.

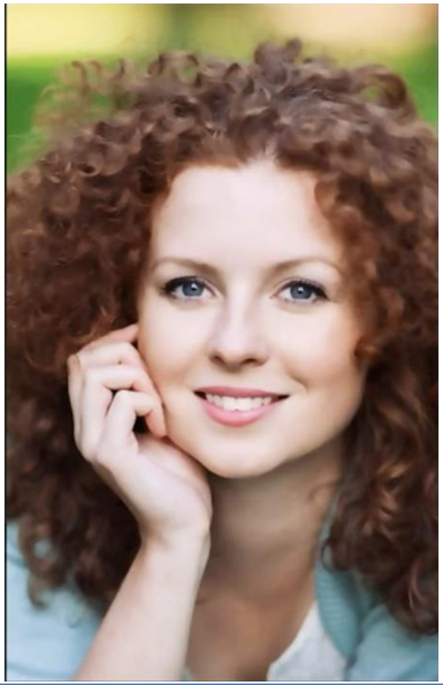
Original



Gaussian filter

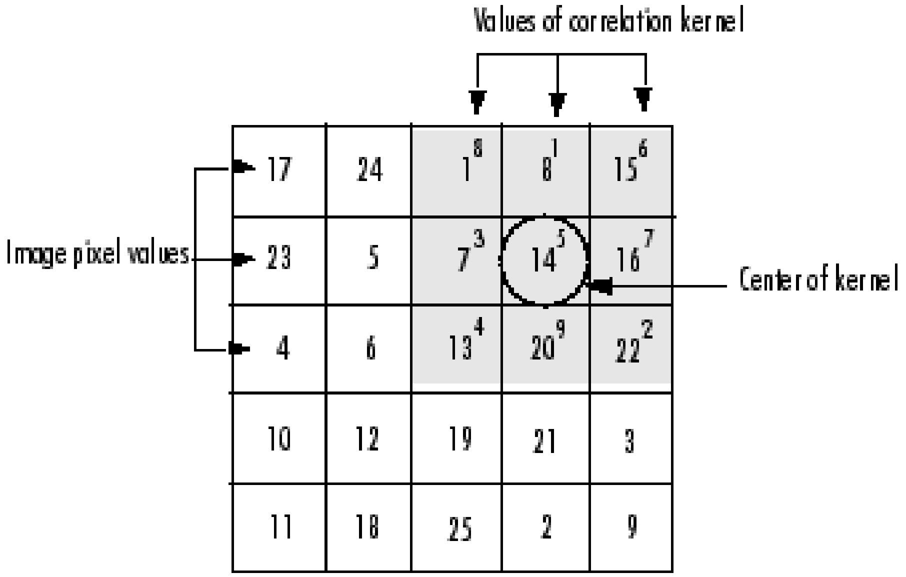


Bilateral filter



Linear filtering: Correlation

```
h = [8   1   6
     3   5   7
     4   9   2]
```



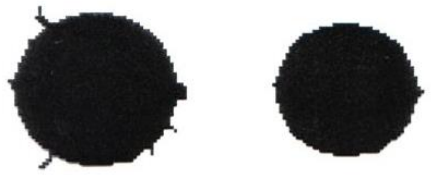
17	24	1 (8)	8 (1)	15 (6)
23	5	7 (3)	14 (5)	16 (7)
4	6	13 (4)	20 (9)	22 (2)
10	12	19	21	3
11	18	25	2	9

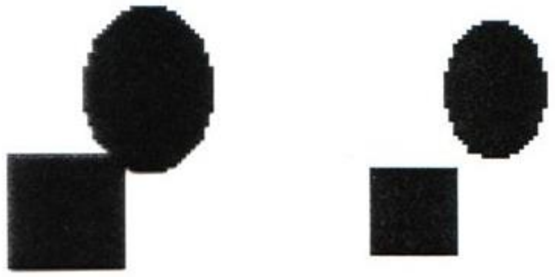
$$1 \cdot 8 + 8 \cdot 1 + 15 \cdot 6 + 7 \cdot 3 + 14 \cdot 5 + 16 \cdot 7 + 13 \cdot 4 + 20 \cdot 9 + 22 \cdot 2 = 585$$

In this case we do not flip the kernel before applying it

Erosion

- To remove irrelevant details from the object shape
- Shrinks the image

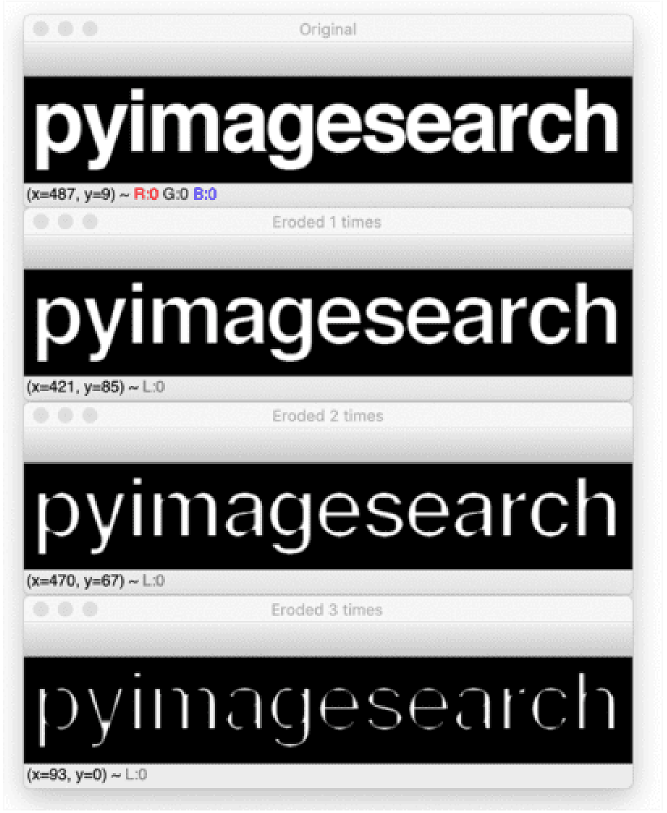




Erosion

Objective: the pixels near the boundary of an object are discarded

How to do: each foreground pixel is kept as foreground only if all the pixels in the structuring element have values greater than 0.



Erosion and dilation

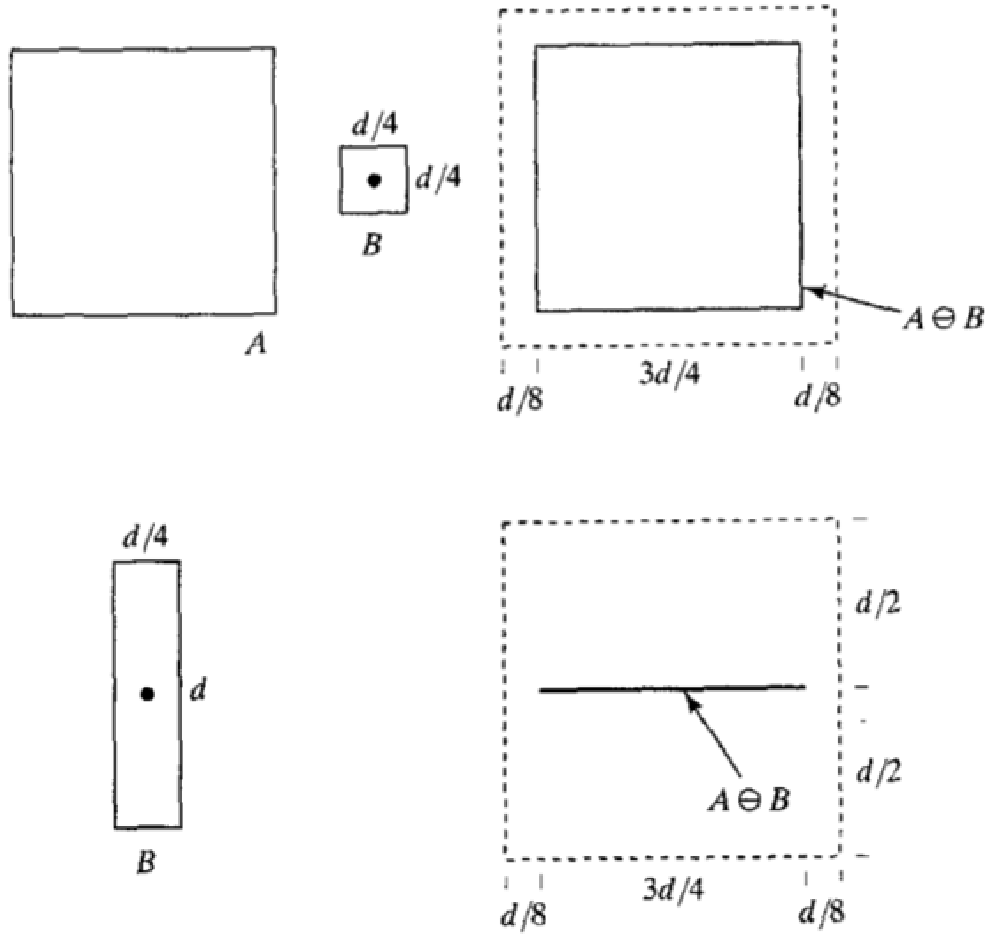
Dilation: if at least one of the pixels in the structuring element is > 0.

Erosion: each foreground pixel is kept as foreground only if all the pixels in the structuring element have values > 0.

Erosion: properties

- It is not commutative
- It is associative if the kernel can be decomposed in dilation components
- It is the opposite of the dilation: if the origin is in the kernel, the operation is anti-extensive and the final eroded set is part of the original one
- It is a monotonic transformation

Erosion



a b c
d e

FIGURE 9.6 (a) Set A. (b) Square structuring element. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A using this element.

Combinations

If our goal is to remove noise or irrelevant details and fill holes without affecting the remaining parts of the image, the solution may be to combine erosion and dilation.

- Erosion removes the irrelevant details
- Dilation fills the gaps in the image

It is possible to apply erosion and then dilation, or viceversa.
Dilation followed by erosion:
Smooth contour, eliminate small holes, fill gaps in the countor
Erosion followed by dilation:
Eliminate protrusions, smooth contour

Combinations

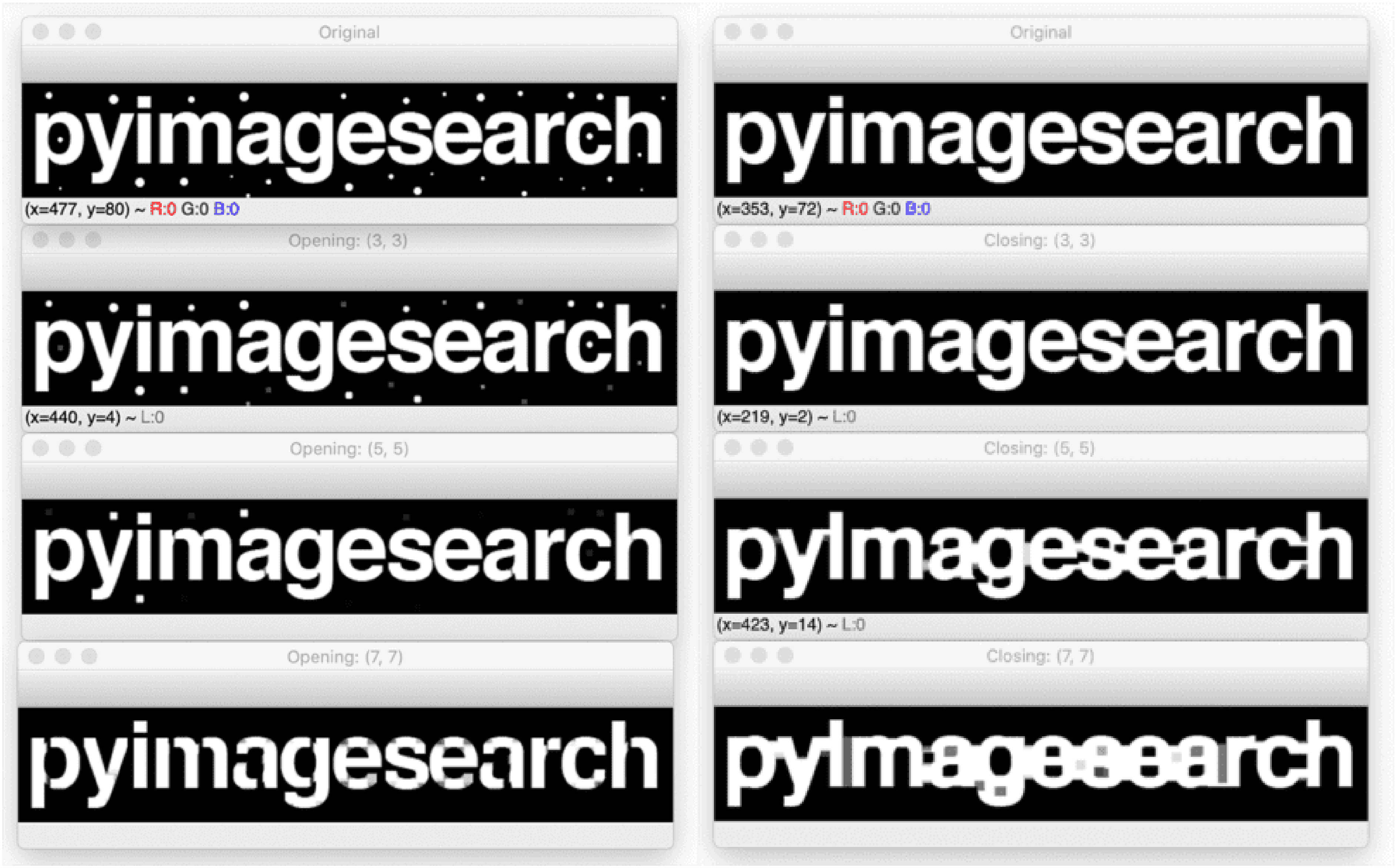
Dilation followed by erosion is a CLOSING

Smooth contour, eliminate small holes, fill gaps in the countor

Erosion followed by dilation is an OPENING

Eliminate protrusions, smooth contour

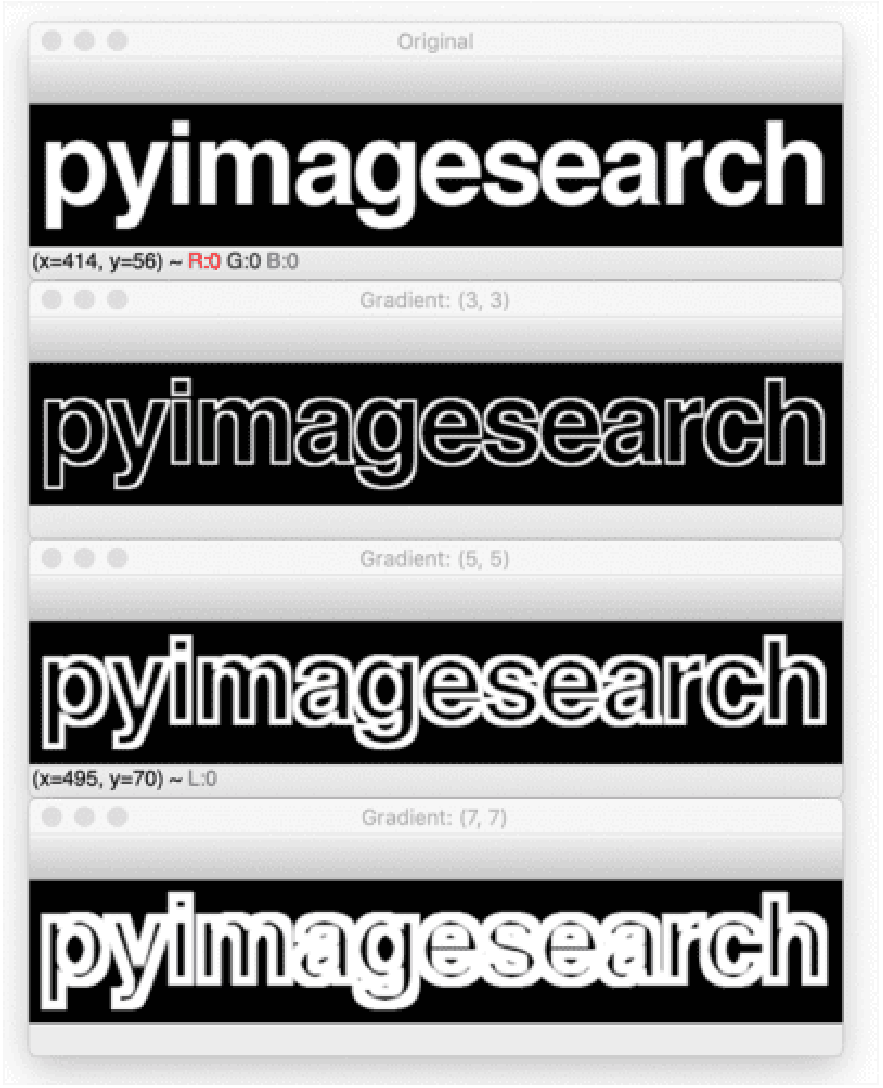
Combinations



Morphological gradient

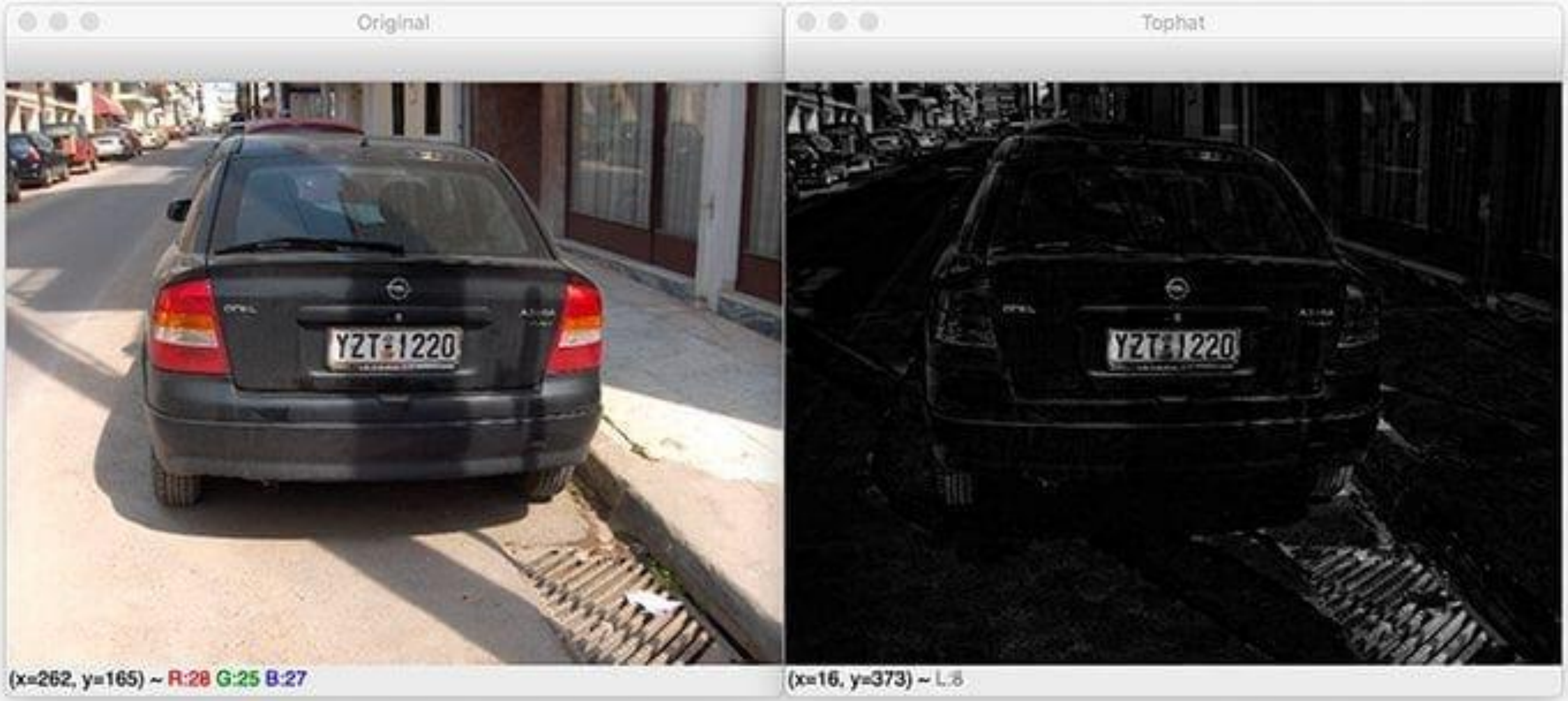
It is the difference between a dilation and erosion.

It enhance the borders/outlines



Top hat/white hat

It is the difference between the original image in input and the opening operation.
It gives you bright regions on dark background.



Examples

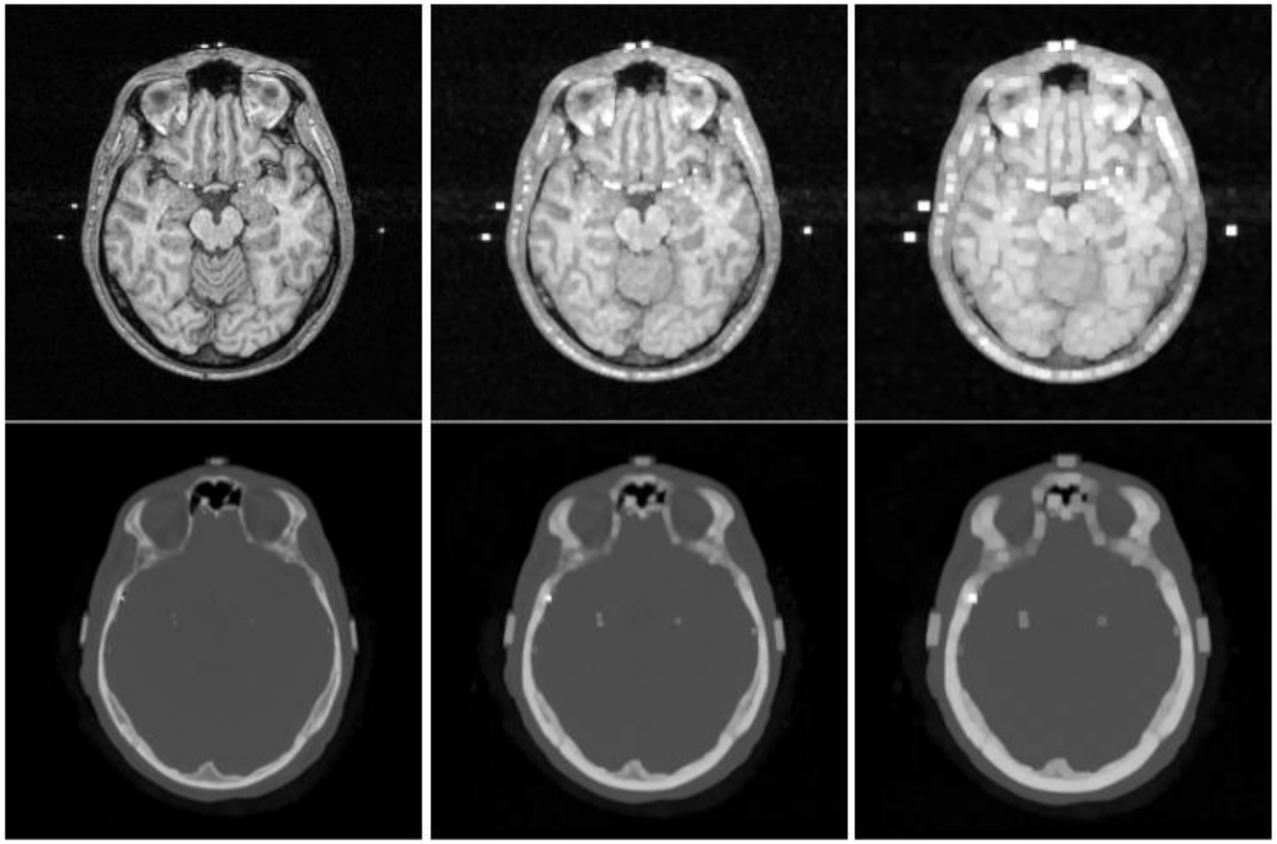


Figure 6.7 Example of dilation by a 3×3 square structuring element (middle column) and a 5×5 square structuring element (right column), applied to a 256×256 MR image (top row) and a 256×256 CT image (bottom row).

Fourier Transform

The Fourier transformation can be applied to images for converting the images into sine waves of variuos frequencies. By converting the images in this way, it is possible filtering out unwanted information.

Applications:

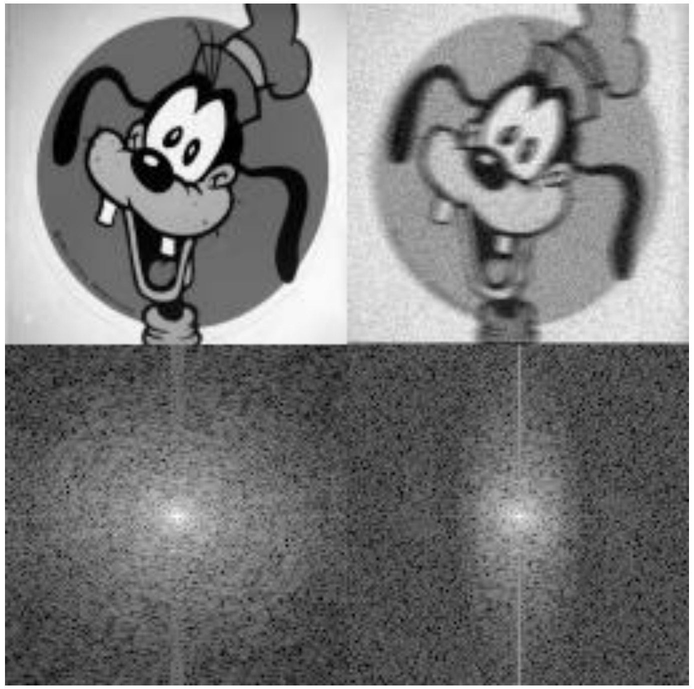
- Image filtering
- Image reconstruction
- Image compression

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}$$

One dimensional discrete Fourier transform

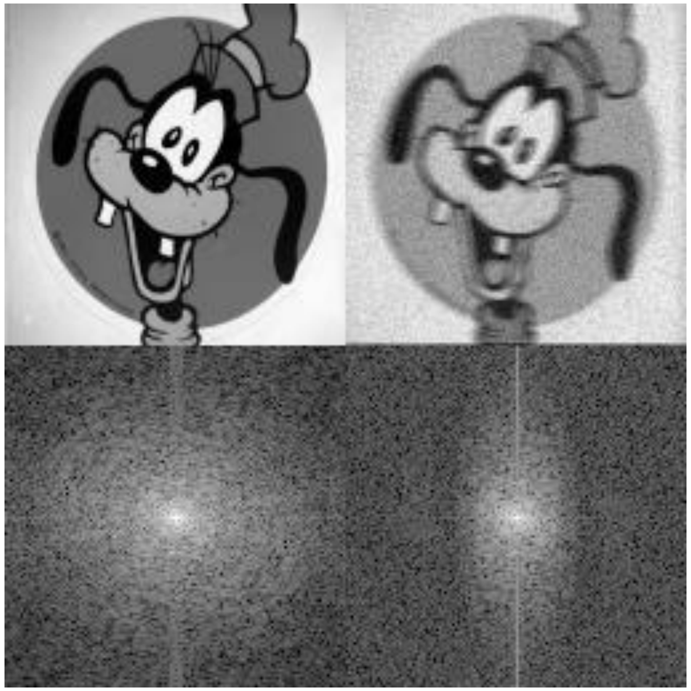
Fourier Transform

The one dimensional transformation has to be applied two times for each image: once on the horizontal axes, once on the vertical one.

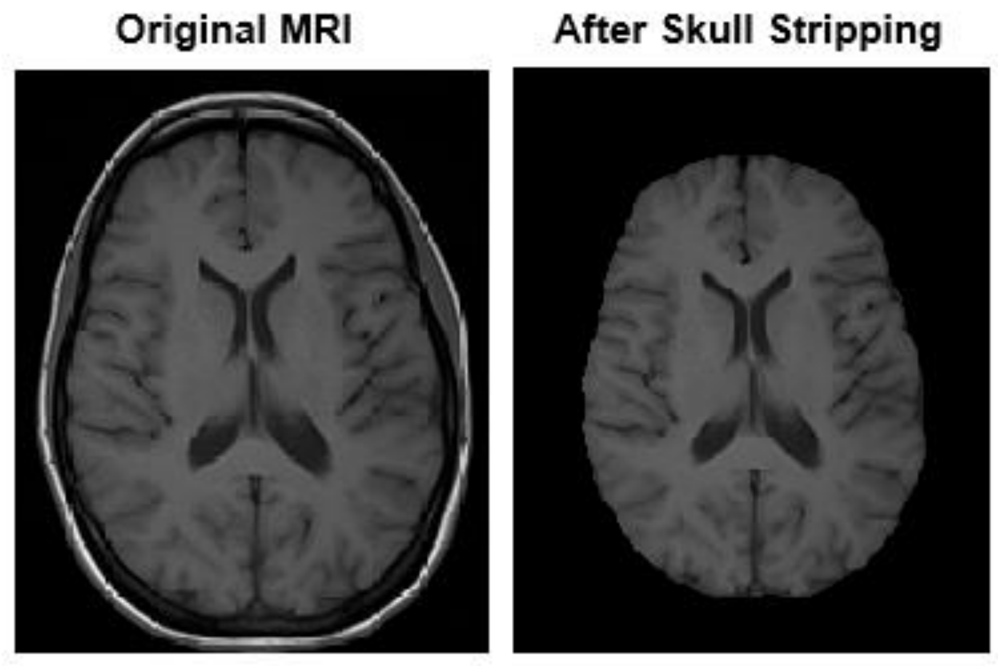


Fourier Transform: applications

- Scaling
- Shifting
- Image reconstruction
- Image compression
- Image filtering



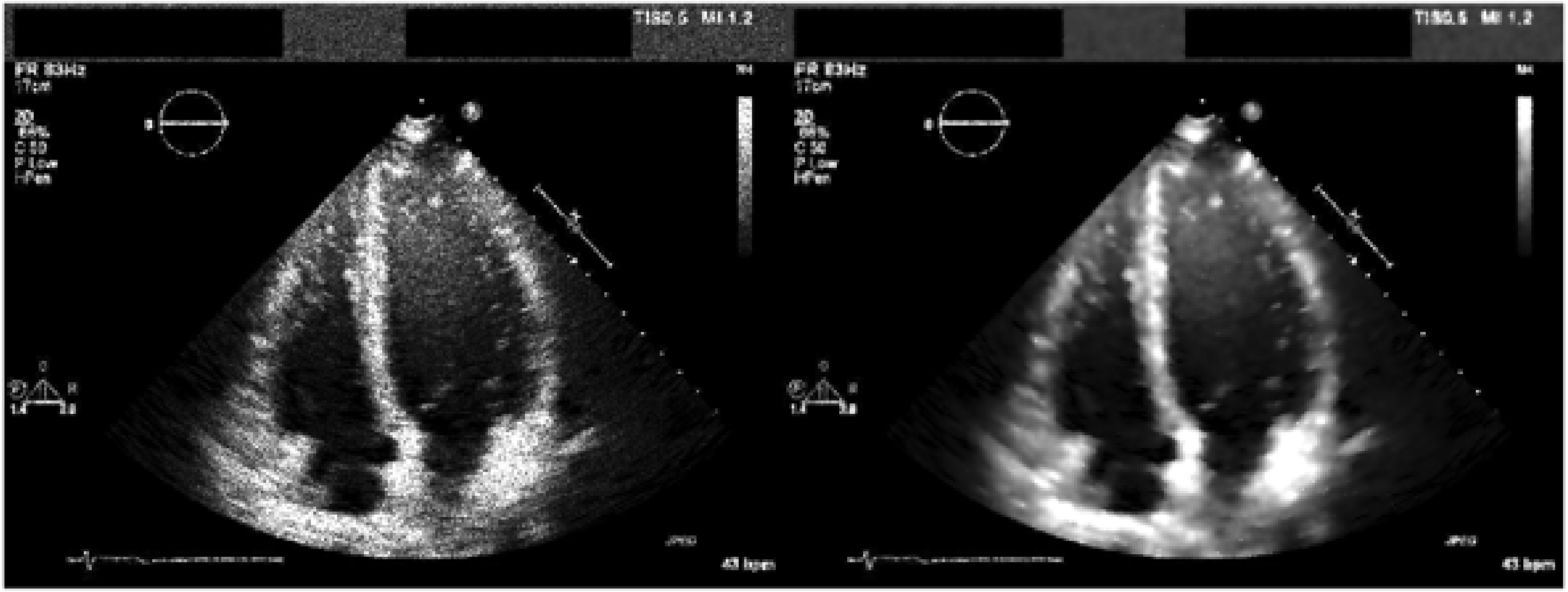
Medical images: background removing



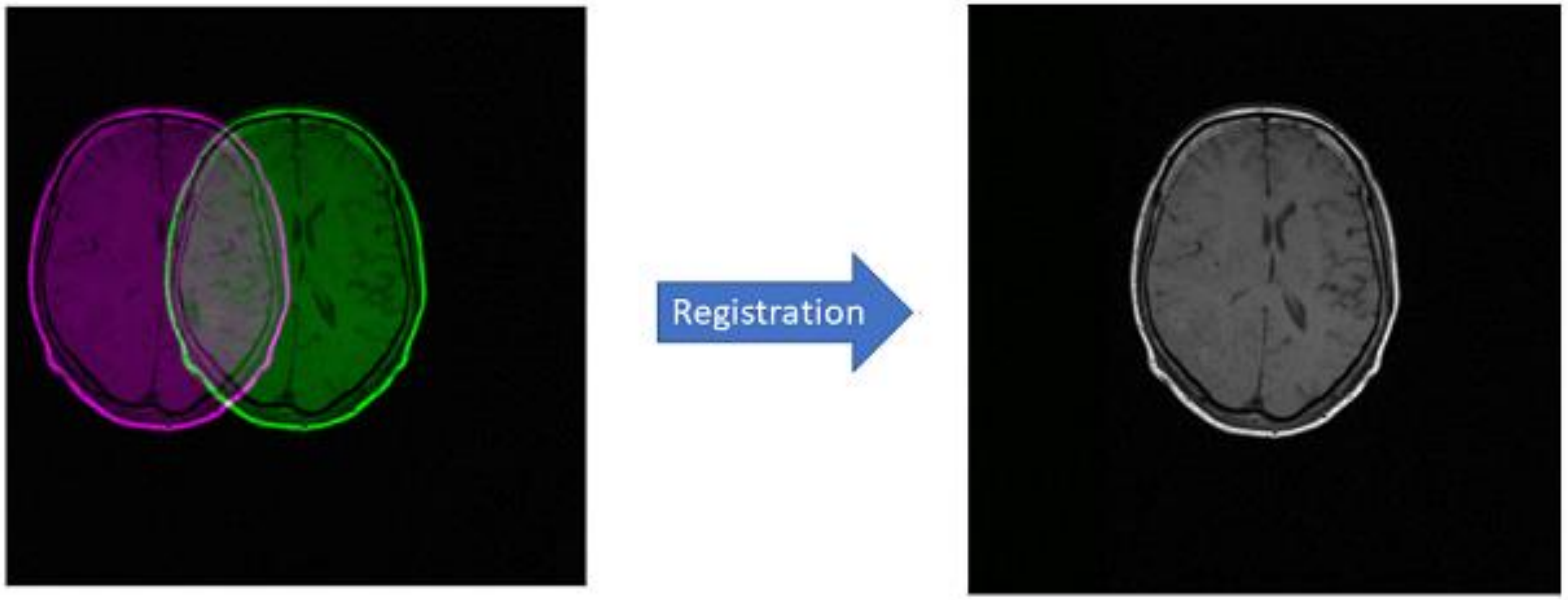
Medical images: noise removal

Ultrasound with speckle noise

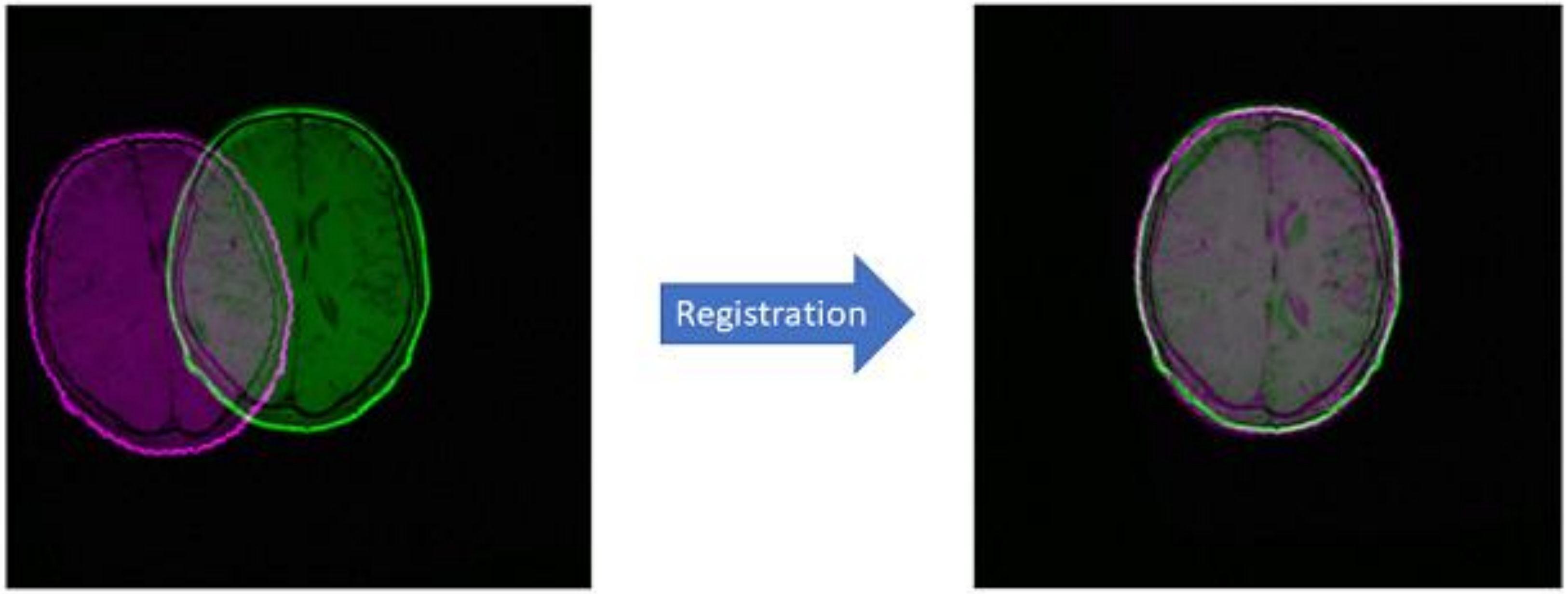
Ultrasound filtered using `specklefilt`



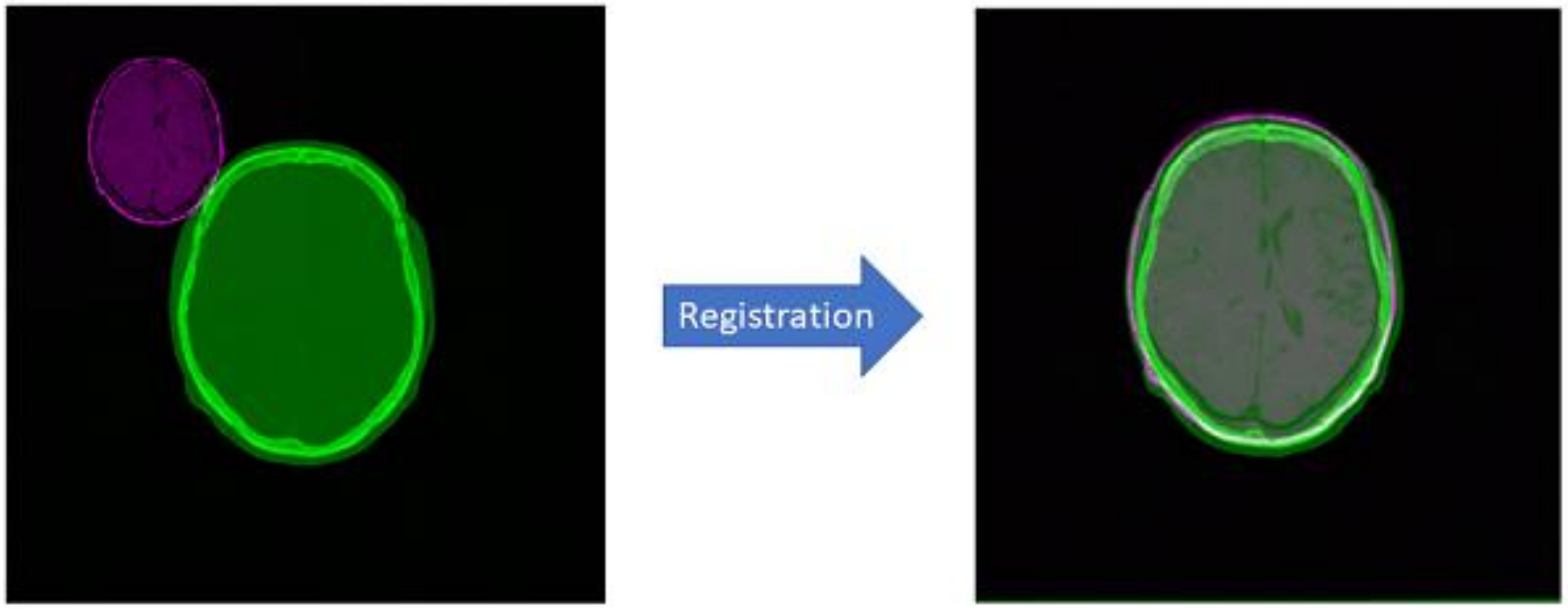
Medical images: registration via translation



Medical images: registration via rotation



Medical images: registration via similarity



How to evaluate similarity in images?

We want a number that tells us how similar two images are:

- Feature based approaches
- Structural similarity
- Histogram based similarity
- Deep learning based approaches

Histogram based approaches

These methods capture the distribution of pixels in an image.

By comparing histograms, you can measure similarity.

PROS: it is not affected by images of different dimensions.

- Histogram Intersection
- Histogram correlation

Structural similarity Index (SSIM)

This method considers luminance, contrasts and structure of the images.

-1 means the two images are different, 1 they are identical.

CONS: it is affected by different dimensions of the images. Pre-processing, like background subtraction or noise removing is fundamental.

Scale Invariant Feature Transform (SIFT)

It is a feature based approach. It looks at the edges, key points, corners and so on. After having identified the most salient points of the image, they are compared to evaluate the similarity.