SSI: Fundamentals of Image processing

Xavier Lladó (llado@eia.udg.edu)
Arnau Oliver (aoliver@eia.udg.edu)

VICOROB – Universitat de Girona

Fundamentals of Image Processing

- Contents
  - Image
  - Histograms
  - Neighbourhood operations
  - Mathematical morphology
  - Using more than one image

But first, what is Image processing?

You can **support Wikipedia** by making a tax-deductible donation.

Log in / create account

article | discussion | edit this page | history

Image processing

From Wikipedia, the free encyclopedia

Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

Monochrome black/white image

Image processing usually refers to digital image processing, but optical and analog image processing are also possible. This article is about general techniques that apply to all of them.

Contents [hide]

Typical operations [edit]

Among many other image processing operations are:

The red, green, and blue color channels of a photograph by Sergei Mikhailovich Prokudin-Gorskii. The fourth image is a composite.

- Geometric transformations such as enlargement, reduction, and rotation
- Color corrections such as brightness and contrast adjustments, quantization, or conversion to a different color space
- Digital compositing or optical compositing (combination of two or more images). Used in filmmaking to make a "matte"
- Interpolation, demosaicing, and recovery of a full image from a raw image format using a Bayer filter pattern
- Image editing (e.g., to increase the quality of a digital image)
- Image differencing

1. Image, under an algorithmic and simple point of view

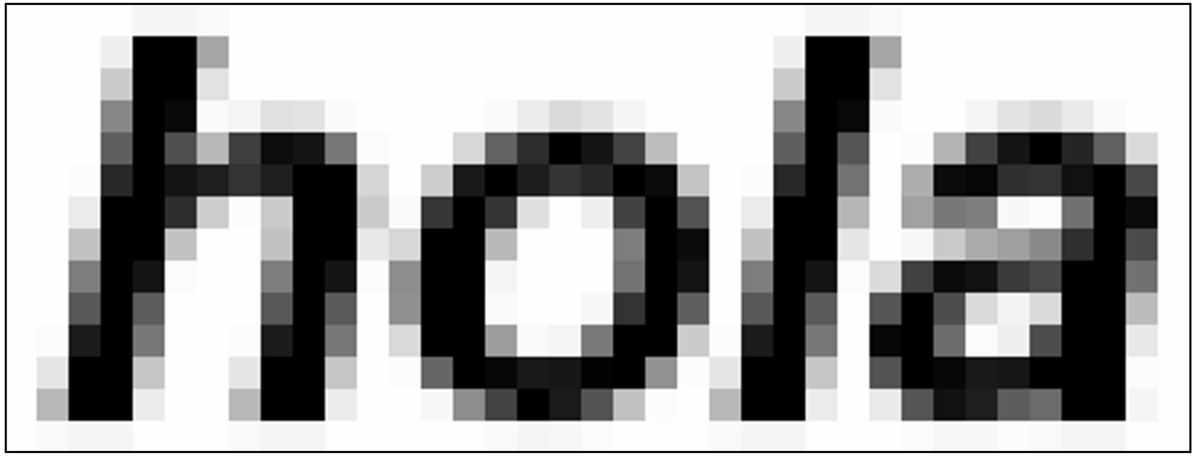
1 FIP

1.1 digital image: algorithmic pov

- 

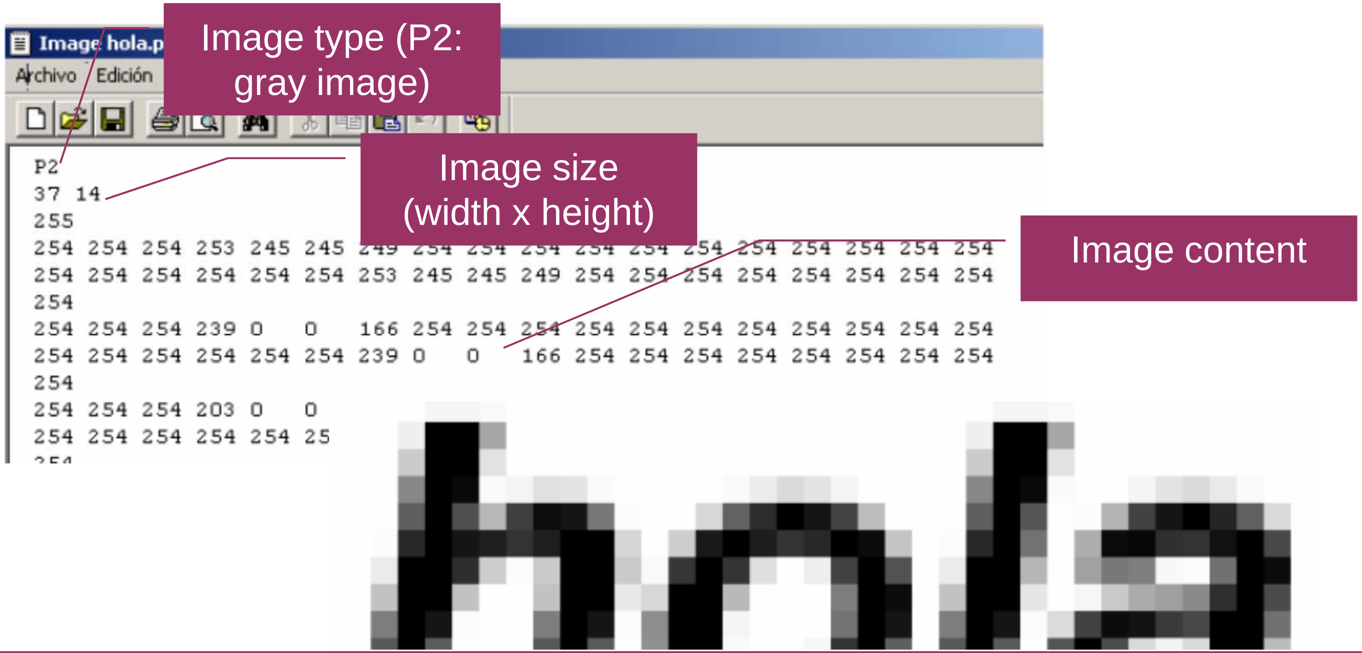
Original image

Zoom x1600



1 FIP
1.1 digital image: algorithmic pov

- Image representation
  - Java `byte [ ][ ] image = new byte[height][width]`
  - Matlab `image = zeros(width,height)`



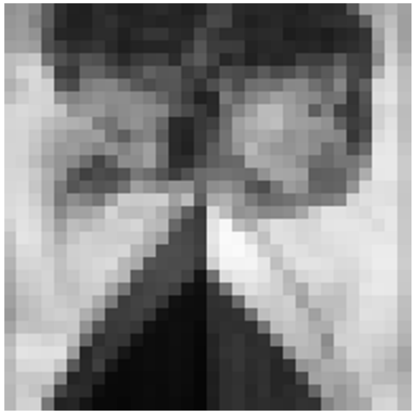
1 FIP

1.1 digital image: spatial resolution

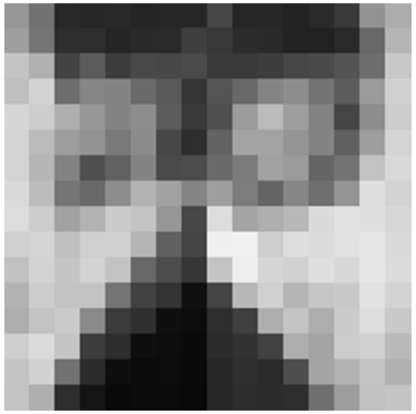
n=256 x 256

n=128 x 128

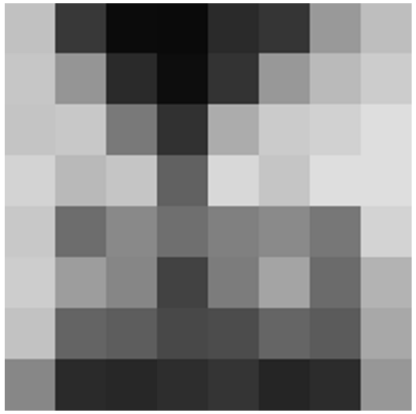
n=64 x 64



n=32 x 32



n=16 x 16



n=8 x 8

1 FIP
1.1 digital image: spatial resolution

n=256 x 256

n=128 x 128

n=64 x 64

n=32 x 32

n=16 x 16

n=8 x 8

1 FIP
1.1 digital image: quantization

- Number of grey-levels: $n = 2^z$

z = 8 bits
256 grey levels

z = 4 bits
16 grey levels

z = 2 bits
4 grey levels

z = 1 bit
Binary Image

1.1 digital images

- Color and multi-band images

• Grey scale image: 1 band
• Color Images: 3 bands (i.e. RGB or HLS)

Each band is represented by a pixel matrix where each pixel is represented by *z* bits

R G B

1. Image Characteristics

- Distance between pixels *(i,j)* and *(h,k)*
  - Euclidean

$$D_E((i,j),(h,k)) = \sqrt{(i-h)^2 + (j-k)^2}$$

  - City-block (Manhattan)

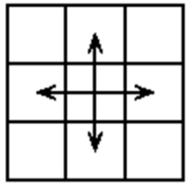
$$D_4((i,j),(h,k)) = |\, i-h \,| + |\, j-k \,|$$

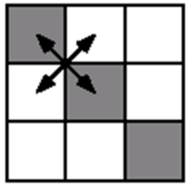
  - Chessboard

$$D_8((i,j),(h,k)) = max\{|\, i-h \,|,\ |\, j-k \,|\}$$

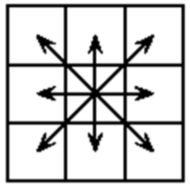
1. Image Characteristics

- Connectivity
  - 4-Neighbourhood

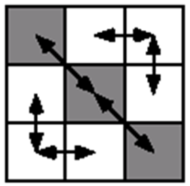


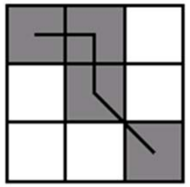


  - 8-Neighbourhood



  - Combinations





1. Image Characteristics

- Image operations
  - Single Image operations
    - At a pixel level: histogram, threshold and arithmetic operations
    - Using neighbourhood information, convolution, noise suppression, edge detection
  - Using more than one image
    - Arithmetic operations
    - Enhancement
    - Template matching

2. Histograms

2. Histogram

- Image grey-level occurrence

```
Initialise H(i) = 0 for all i
For each pixel (i,j)
  H(pixel(i,j)) ++;
```

- Considerations
  - Information about image characteristics and quality
  - Different images may have the same histogram (no spatial information)

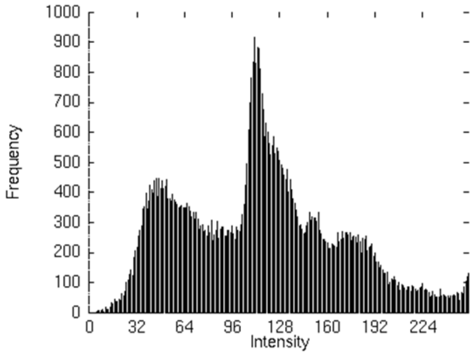
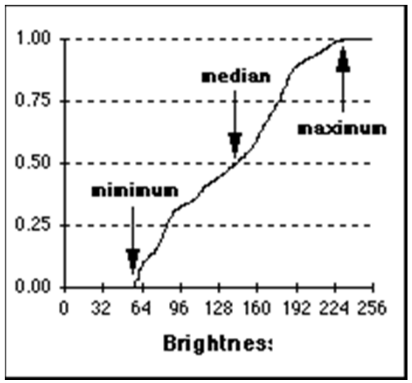
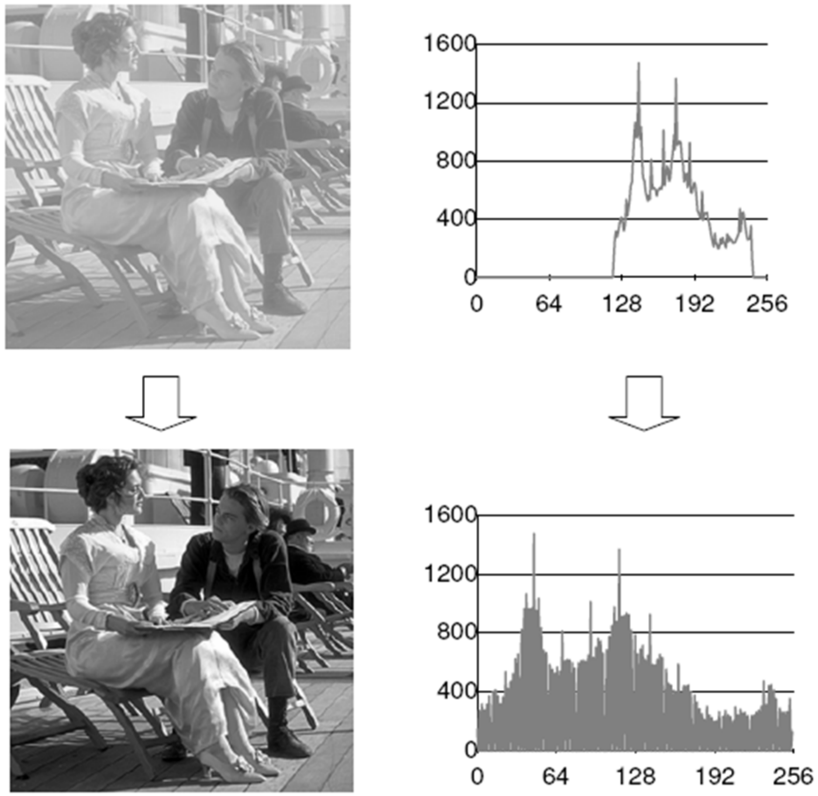


Figure 2.11 A brightness histogram.



2. Histogram

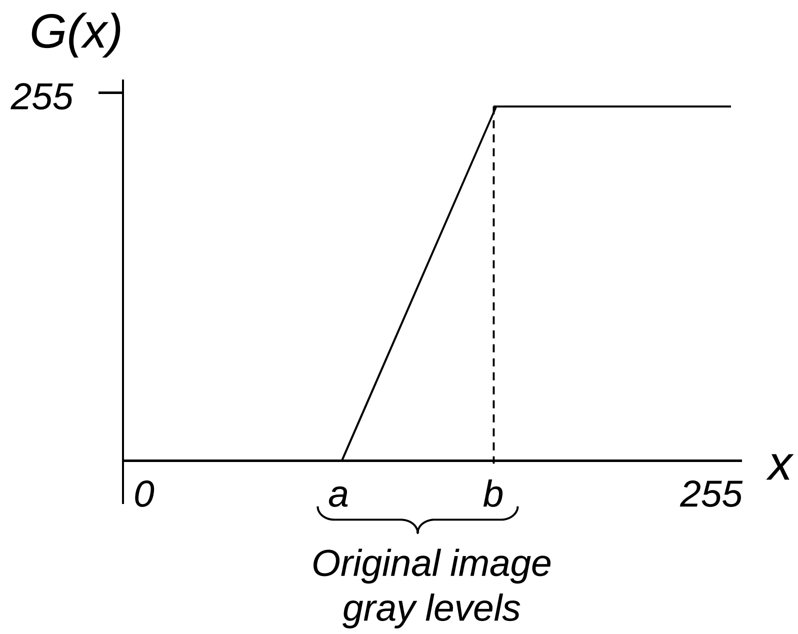
- Histogram Equalisation / Contrast stretching



2. Histogram

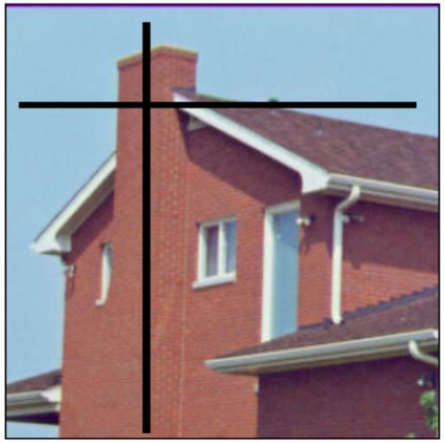
- Contrast maximization

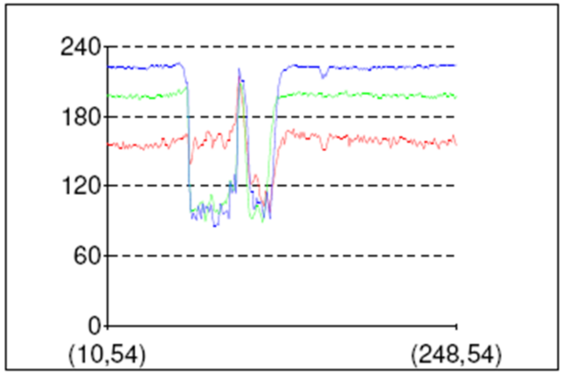
$$G(x,y) = 255 \cdot \left[\frac{I(x,y) - \min(I(x,y))}{\max(I(x,y)) - \min(I(x,y))} \right]$$

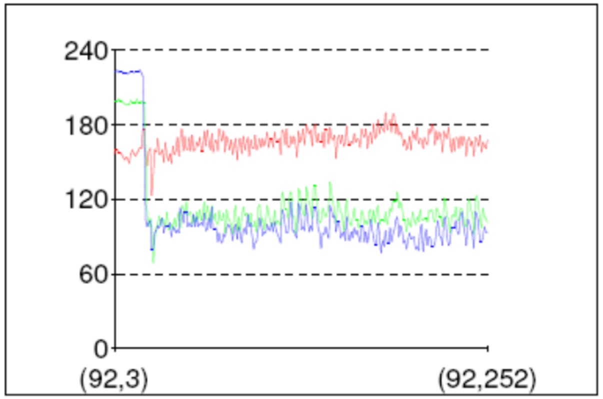


2. Histogram

- Line Histogram (1D)



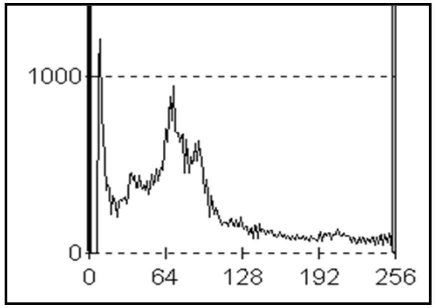




2. Histogram

- Thresholding

f



$histogram(f)$

if $f(x,y) > T$ then $g(x,y) = 255$
else $g(x,y) = 0$

where T is the threshold

g with T=50

g with T=75

g with T=128

g with T=240

2. Histogram
2.2 Thresholding: local threshold

Global Thresholding

Local/Adaptive Thresholding

7x7; T = mean

7x7; T = mean-7

75x75; T = mean-10

3. Neighbourhood Operations

3. Neighbourhood operations

3. Neighbourhood operations

- Neighbourhood operations (spatial filters, masks)
  - For each pixel of the image, a small window is computed for performing an operation. The central pixel changes its value using the result of this operation

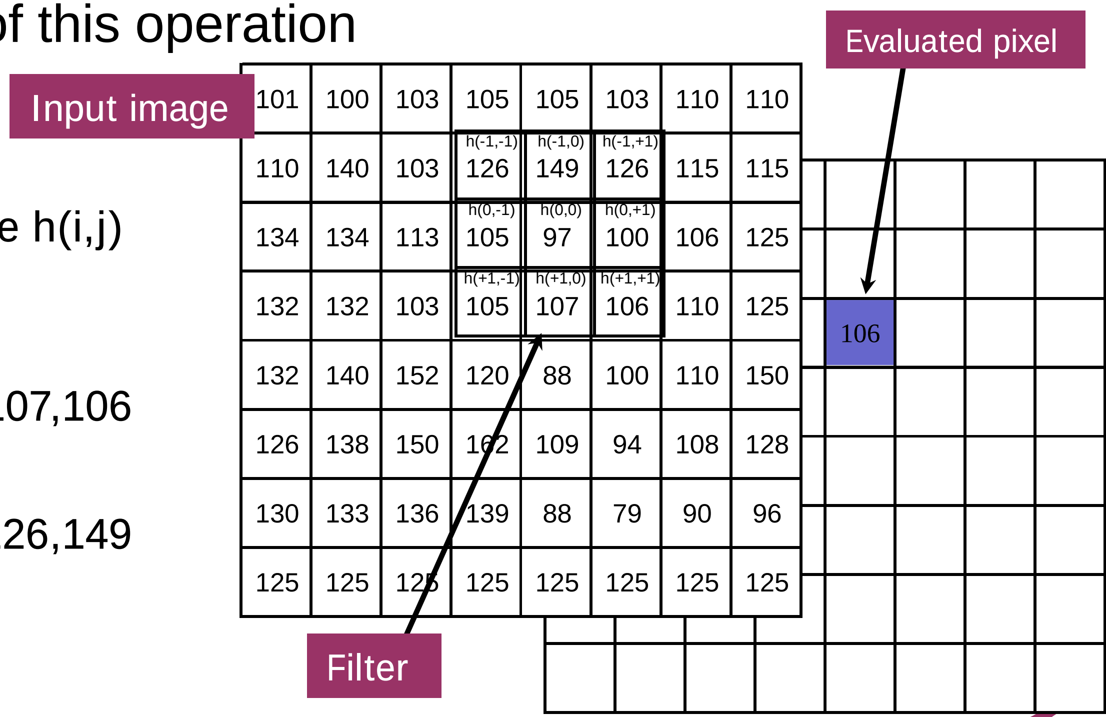
Example: Median Filter

Computes the median of all the h(i,j) belonging to the window.

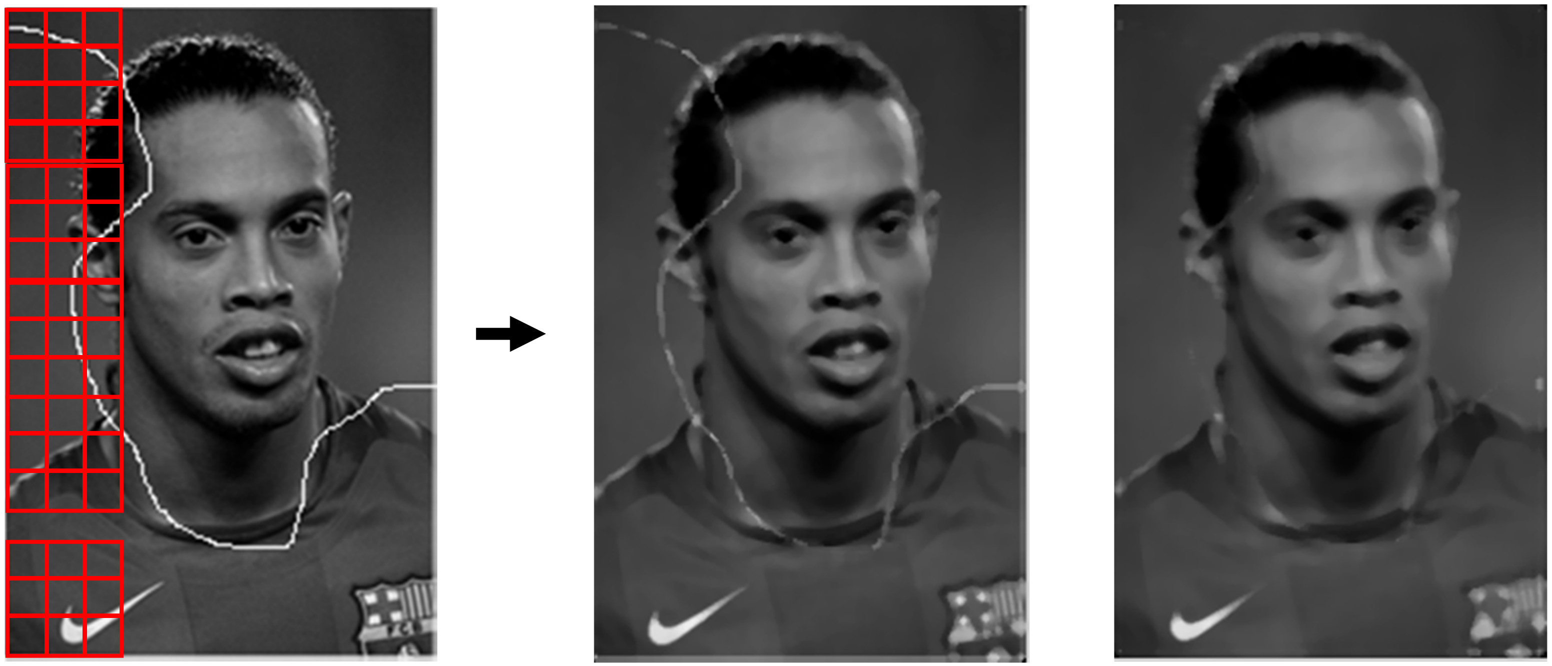
Neighbours= 126,149,126,105,97,100,105,107,106

Sorted list= 97,100,105,105,106,107,126,126,149

The median value is **106**



3. Neighbourhood operations



3. Neighbourhood operations

- Neighbourhood operations:
  - Noise suppression. Remove image noise, associated with removing high frequencies of the image
    - Edge blurring
  - Gradient operations. Detect regions of high gradient in the image. In the frequency space, remove low frequencies
    - Increase noise effects

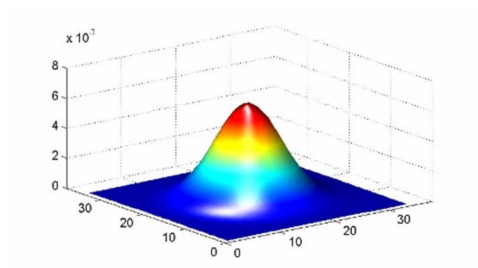
3. Neighbourhood operations
3.1 Noise suppression

- Average

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

- Gaussian Filter

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



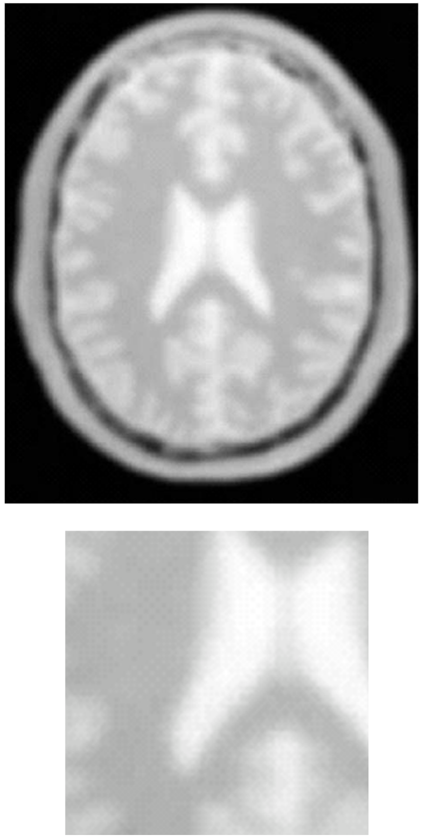
$$h = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Discriminant Filters
  - Smooth only if neighbour is considered noise (is inside an intensity range *[min,max]*)
  - Smoothing degree inversely depends on gradient magnitude (higher the gradient → more likely to be and edge → less smoothing is performed.) Example. Anisotropic Diffusion

- Examples

Original image

Gaussian Filter



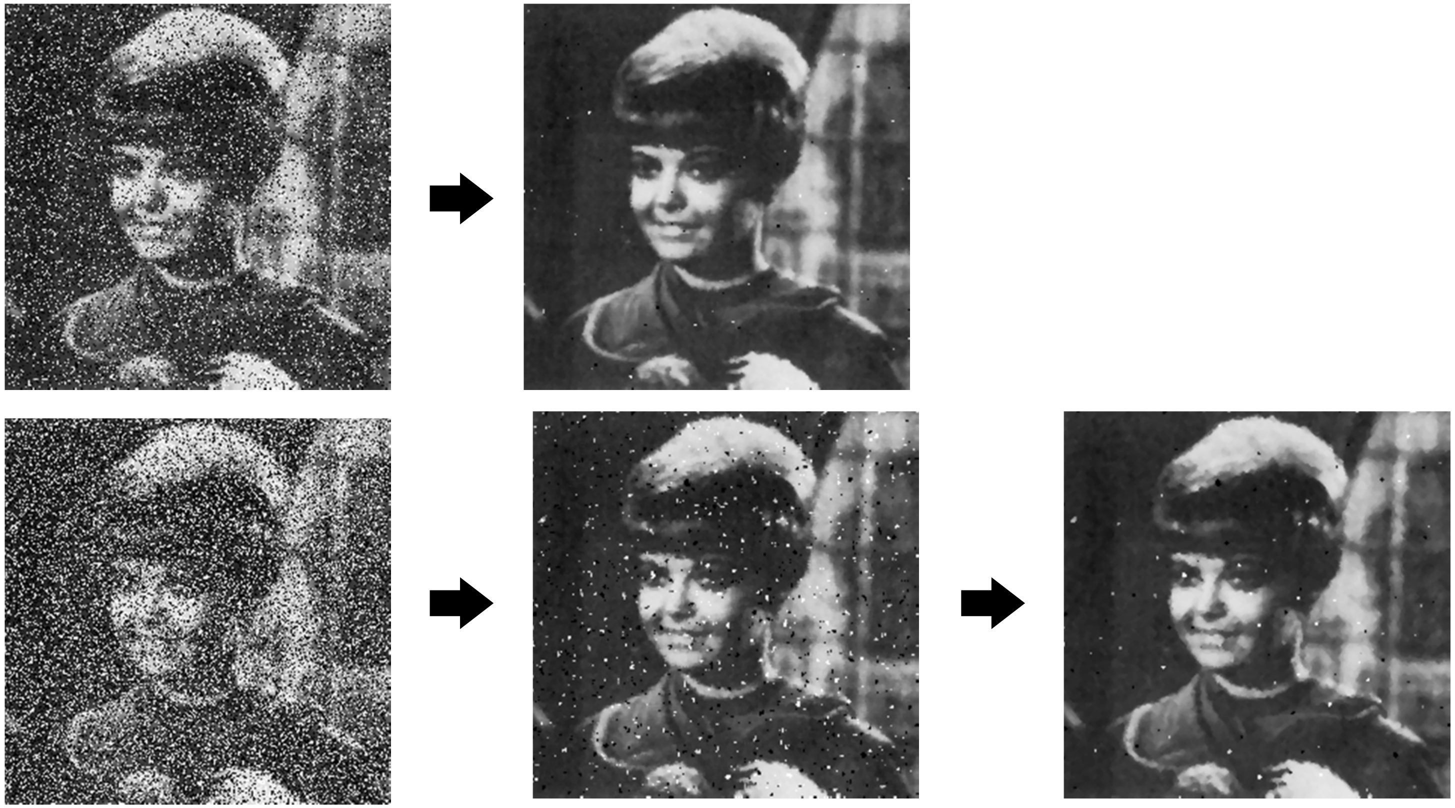
Anisotropic Diffusion

3. Neighbourhood operations
3.1 Noise suppression

- Median

20% noise

40% noise



source ***median 3x3*** ***median 3x3***

3. Neighbourhood operations
3.1 Edge detection

- Edges: Abrupt changes in the image intensity function
- Value changes = derivative

- Gradient
  - Magnitude

$$|L_{xy}| = \sqrt{L_x^2 + L_y^2}$$

$$L_x = \frac{\partial L}{\partial x} \quad L_y = \frac{\partial L}{\partial y},$$

  - Direction

Arctan (L_y / L_x)

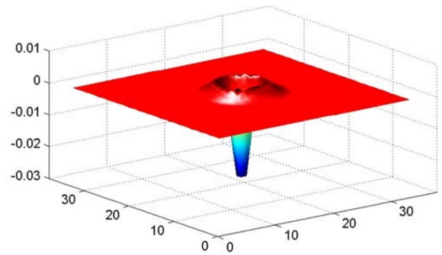
3. Neighbourhood operations
3.1 Edge detection

- 1st derivative
  - Roberts $$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$
  - Prewitt $$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$
  - Sobel $$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$
  - Robinson $$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} h_2 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} h_3 = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$
  - Kirsch $$h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} h_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} h_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}$$

3. Neighbourhood operations
3.1 Edge detection

- Second derivative. Laplacian of Gaussian (LOG)
  - Rotation invariant
  - Detecting zero crossings

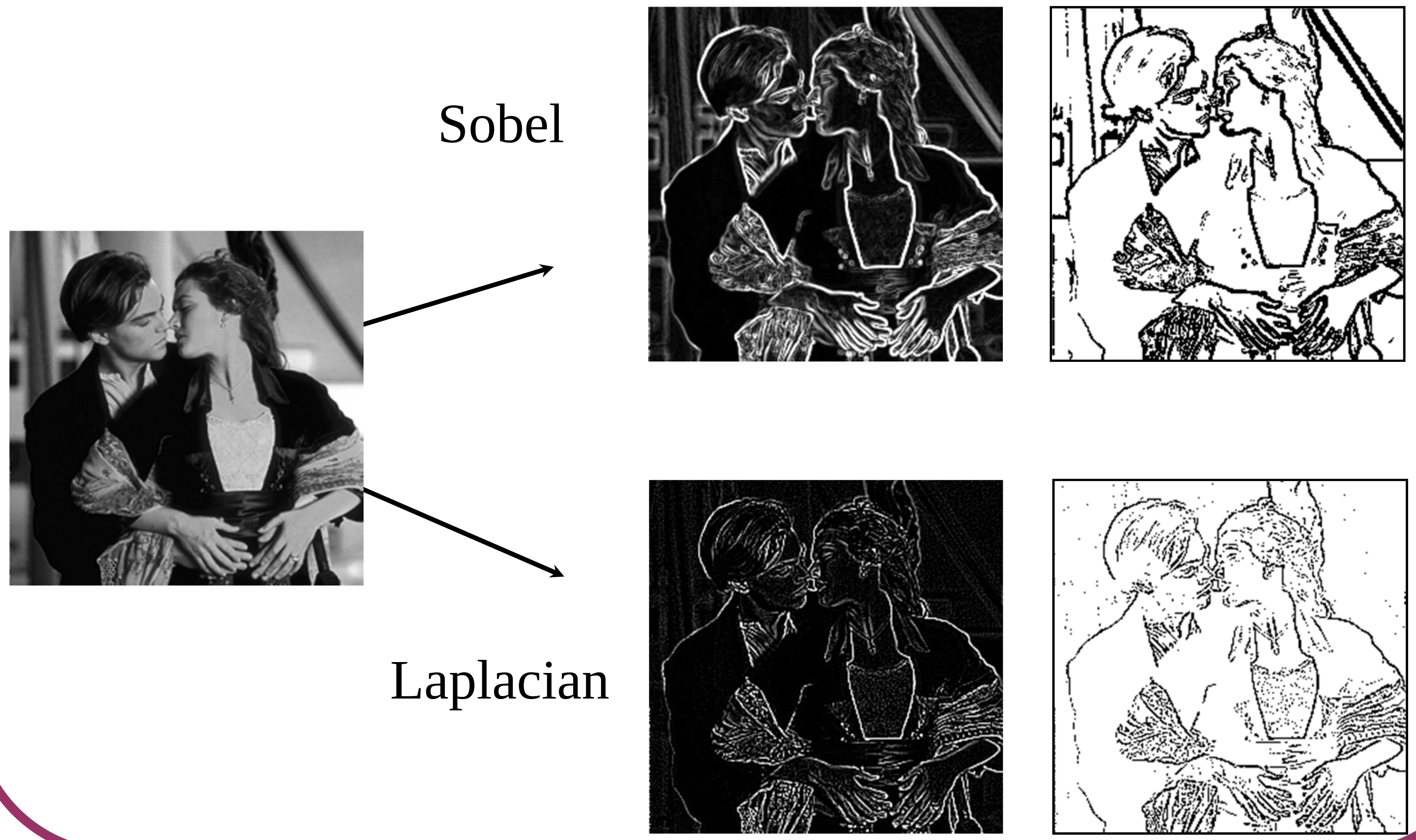
$$\nabla^2(x,y) = \frac{\partial^2 g(x,y)}{\partial x^2} + \frac{\partial^2 g(x,y)}{\partial y^2}$$



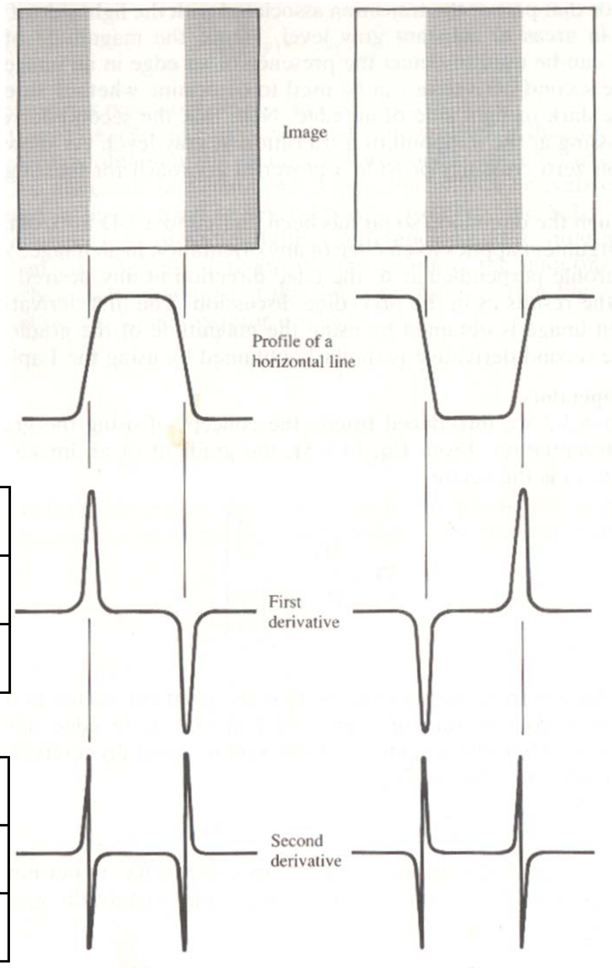
  - Laplacian masks

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

3. Neighbourhood operations
3.1 Edge detection



3. Neighbourhood operations
3.1 Edge detection: 1st & 2nd derivative comparison



-2	0	2
-2	0	2
-2	0	2

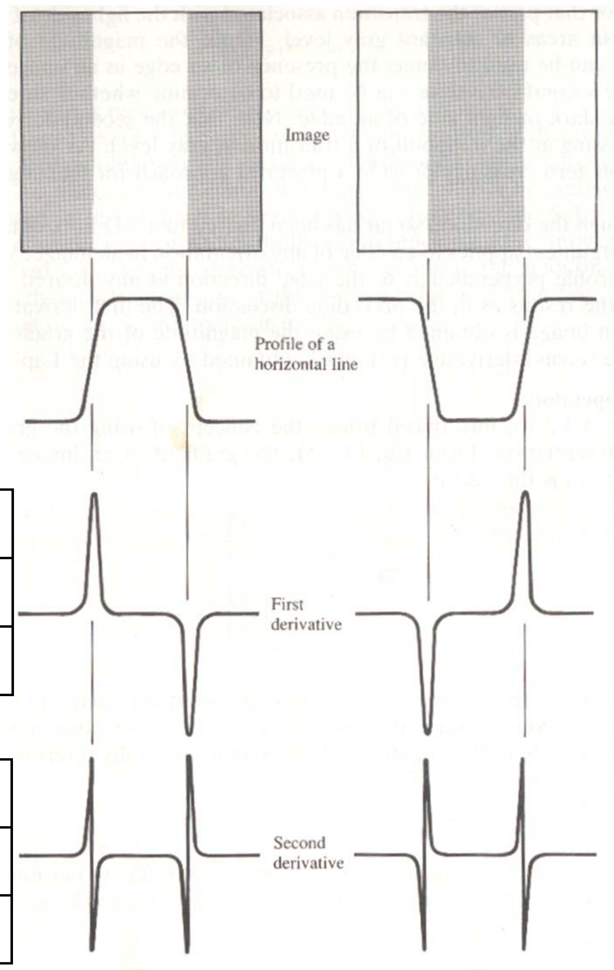
0	1	0
1	-4	1
0	1	0

0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255

1st derivative = -2*0+ -2*0+ -2*0+ 0*0+ 0*0+ 0*0+ 2*0+ 2*0+ 2*0 = 0

2ond derivative = 0*0+ 1*0+ 0*0+ 1*0 - 4*0+ 1*0+ 0*0+ 1*0+ 0*0 = 0

3. Neighbourhood operations
3.1 Edge detection: 1st & 2nd derivative comparison



0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255

-2	0	2
-2	0	2
-2	0	2

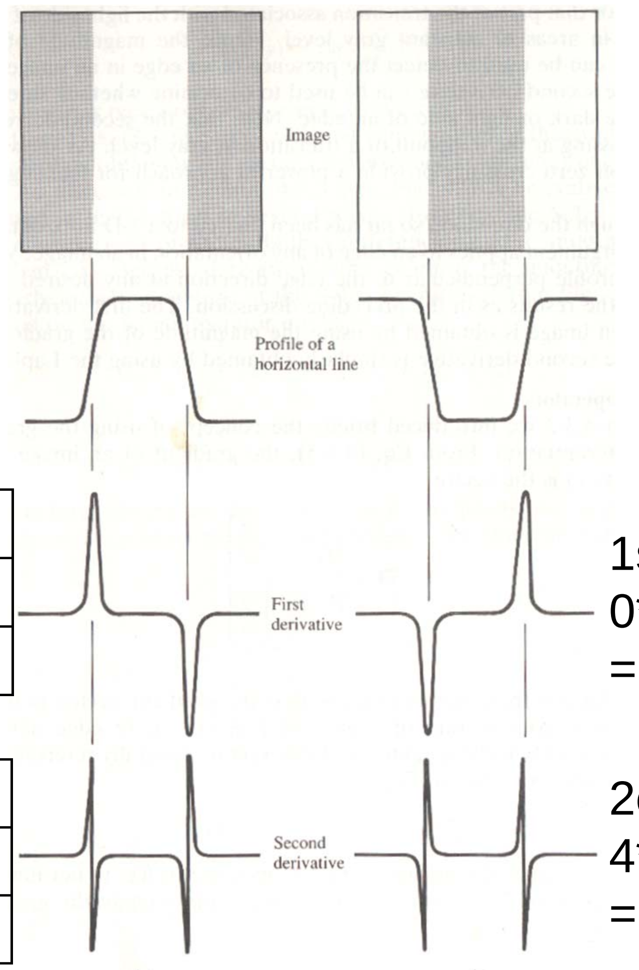
1st derivative = -2*0+ -2*0+ -2*0+ 0*0+ 0*0+ 0*0+2*255 +2*255 +2*255 = 6*255

0	1	0
1	-4	1
0	1	0

2ond derivative = 0*0+ 1*0+ 0*0+ 1*0 - 4*0 + 1*0+ 0*255 +1*255 +0*255 = 255

3. Neighbourhood operations
3.1 Edge detection: 1st & 2nd derivative comparison

0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255



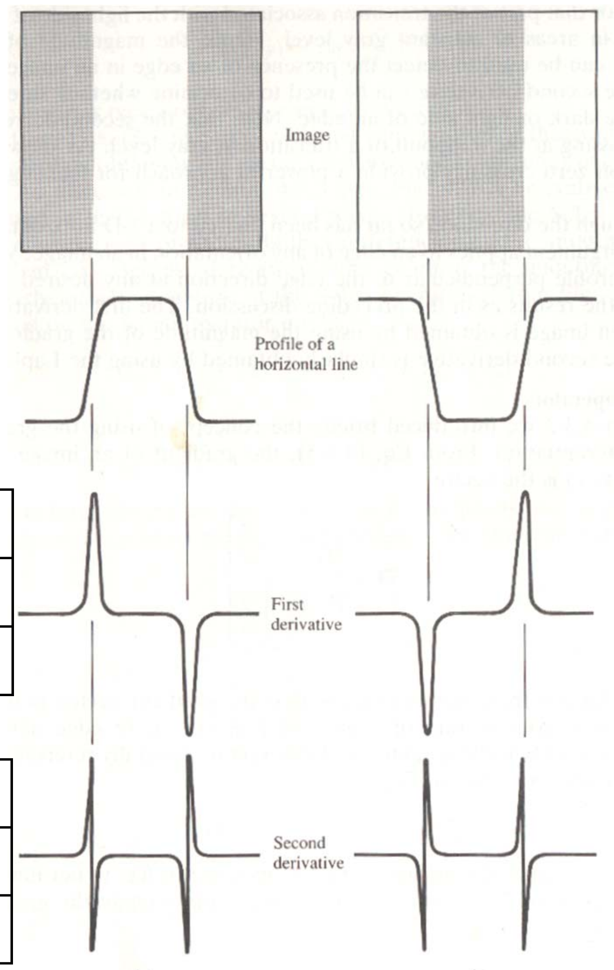
-2	0	2
-2	0	2
-2	0	2

1st derivative = -2*0+ -2*0+ -2*0+ 0*255+ 0*255+ 0*255+2*255 +2*255 +2*255
= 6*255

0	1	0
1	-4	1
0	1	0

2ond derivative = 0*0+ 1*0+ 0*0+ 1*255 - 4*255+ 1*255+0*255 + 1*255 +0*255
= 3*255 – 4*255 = –255

3. Neighbourhood operations
3.1 Edge detection: 1st & 2nd derivative comparison



0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255
0	0	0	0	255	255	255	255

-2	0	2
-2	0	2
-2	0	2

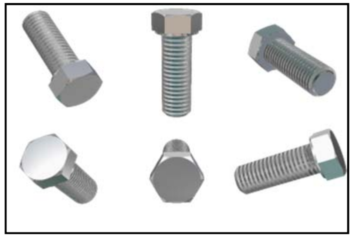
1st derivative = -2*255+ -2*255+ -2*255+ 0*255+ 0*255+ 0*255+2*255 +2*255 +2*255 = -6*255 + 6*255 = 0

0	1	0
1	-4	1
0	1	0

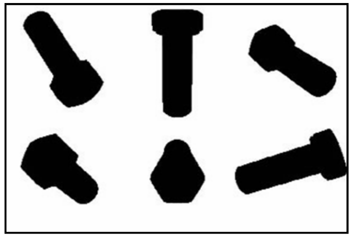
2ond derivative = 0*255+ 1*255+ 0*255+ 1*255 - 4*255+ 1*255 +0*255 + 1*255 +0*255 = 4*255 - 4*255 = 0

4. Mathematical Morphology

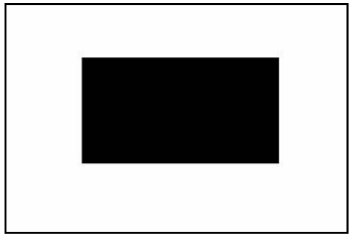
4. Morphology
4.1 Logical Operations



A

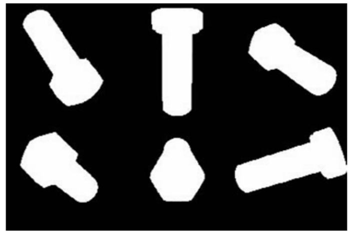


A1

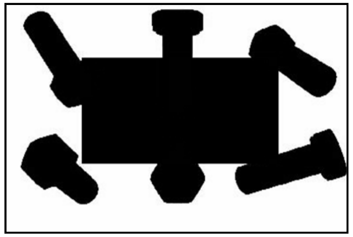


B

$A^c = \{\alpha \mid \alpha \notin A\}$

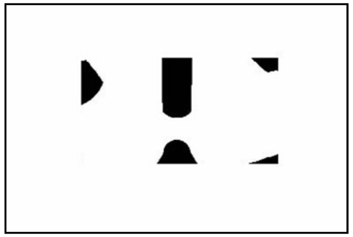


NOT (A1)



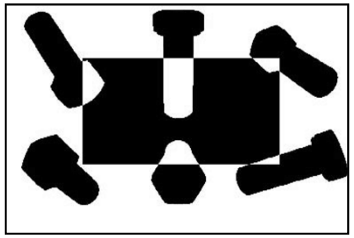
OR (A1,B)

$A \cup B = \{\alpha \mid \alpha \in A \vee \alpha \in B\}$

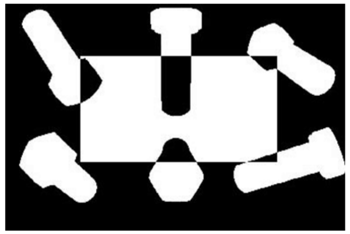


AND (A1,B)

$A \cap B = \{\alpha \mid \alpha \in A \wedge \alpha \in B\}$



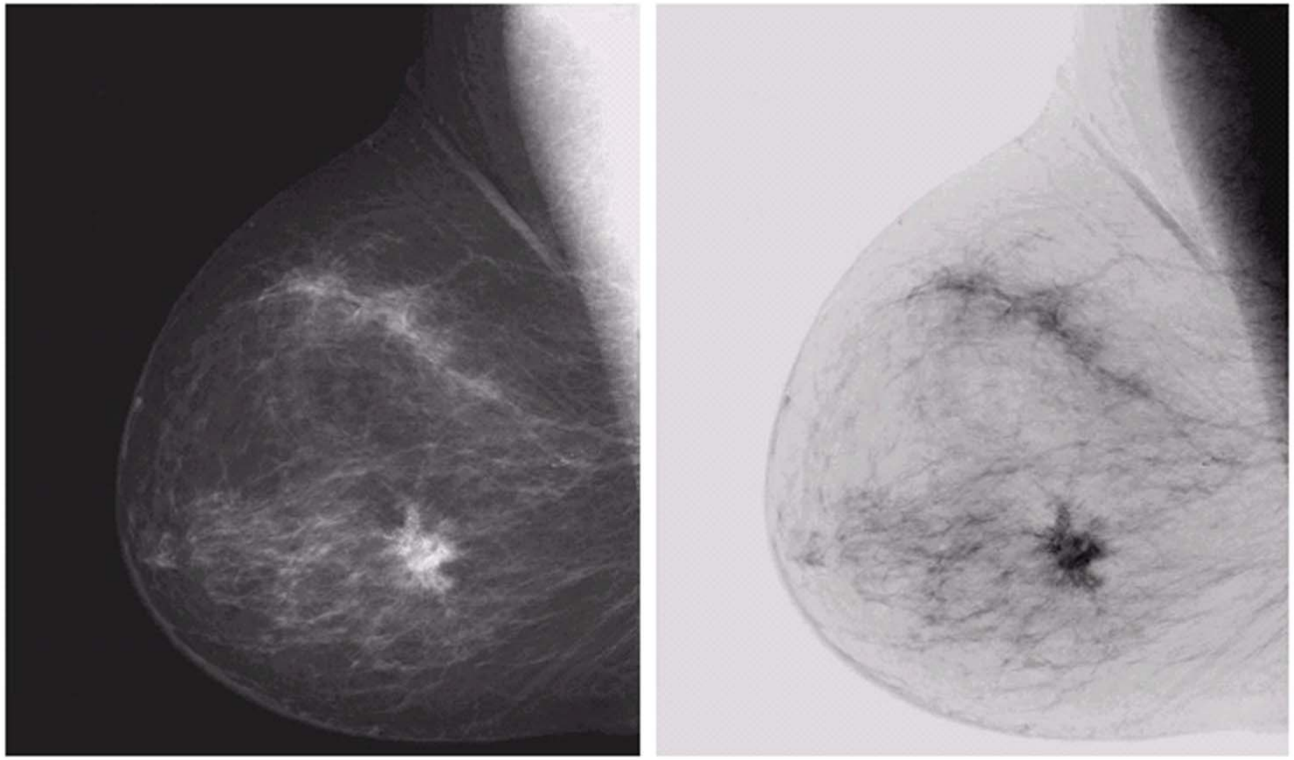
XOR (A1,B)



XNOR(A1,B)

4. Morphology

4.1 Logical Operations



a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

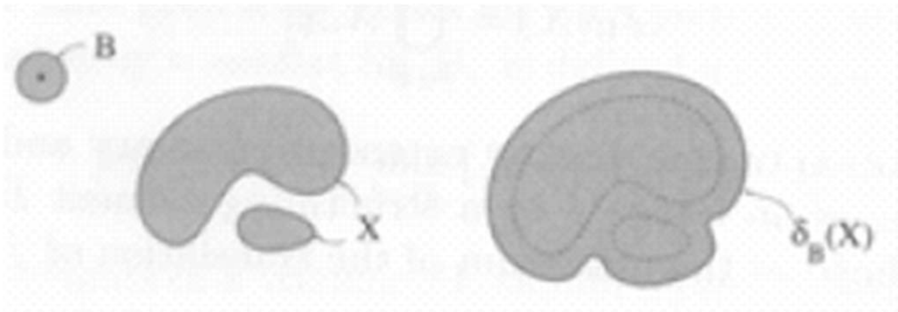
Example from Digital Image Processing by Gonzalez & Woods

4. Morphology
4.2 Dilate

- Dilation is regarded as a convolution of the SE in a region. Dilation of image A using an SE B is defined as

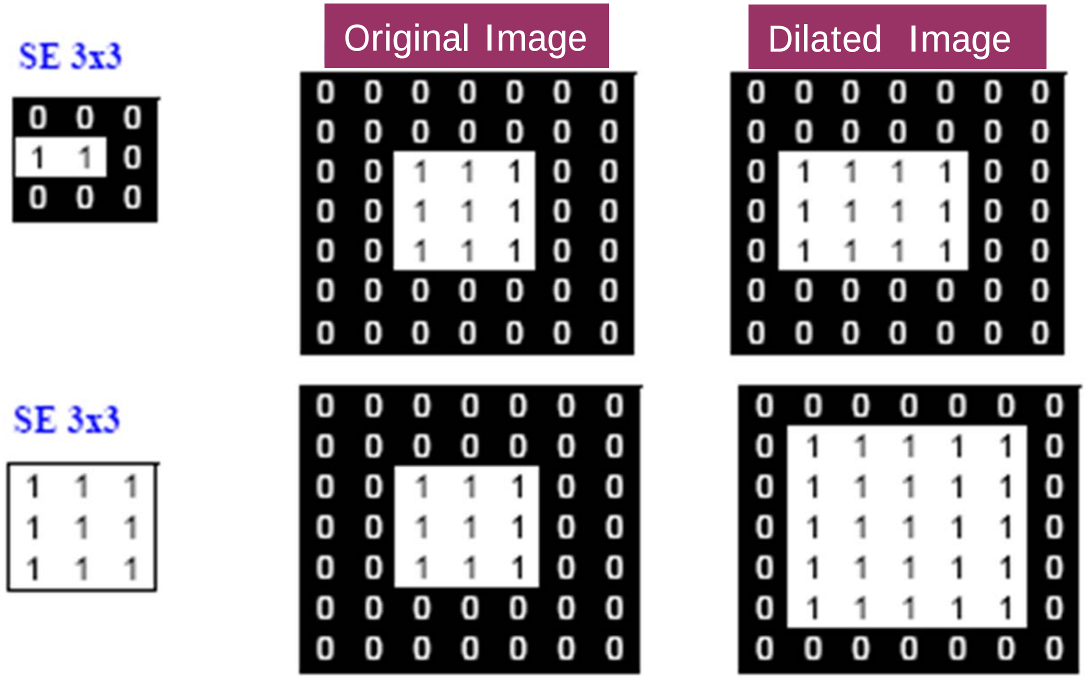
$$C = A \oplus B = \bigcup_{b \in B} (A)_b$$

where (A)b represents the translation of A by b. Intuitively, for each nonzero element $b_{i,j}$ of B, A is translated by i,j and summed into C using the "or" operator



4. Morphology
4.2 Dilate

- Dilate example



4. Morphology

4.2 Dilate

SE 3x3

1	1	1
1	1	1
1	1	1

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

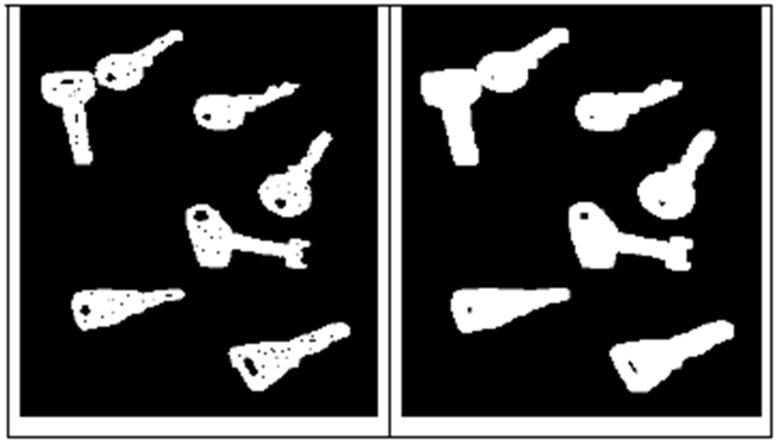
Original Image

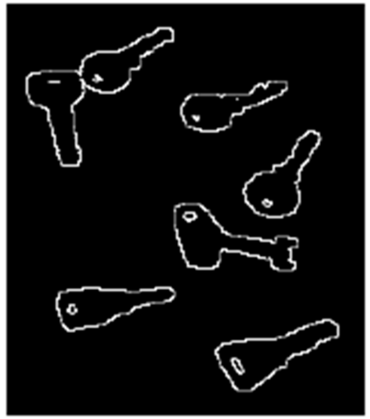
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	0	0
0	1	1	1	0	0	0	1	1	1	0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

Dilated Image

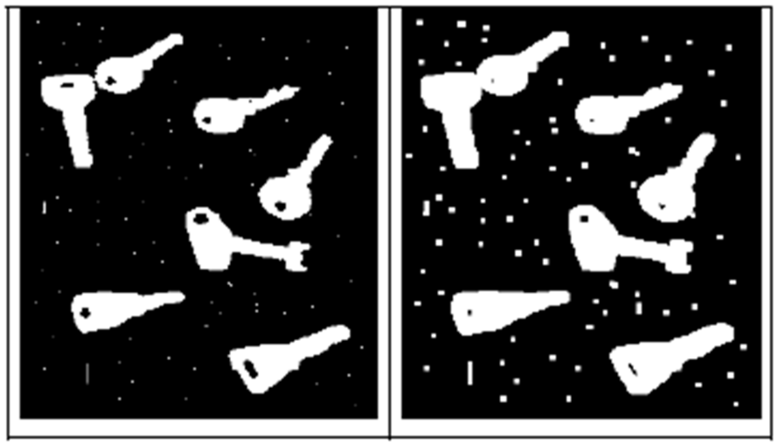
4. Morphology

4.2 Dilate





Dilate - original



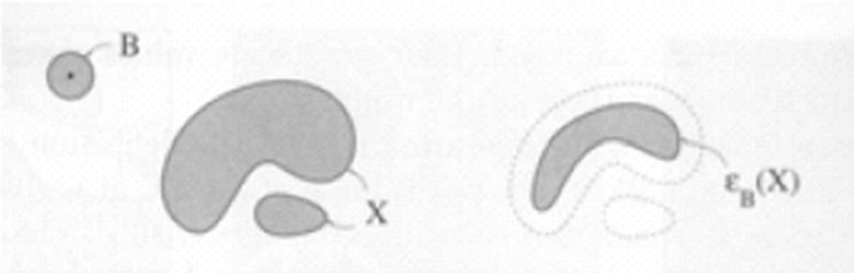
4. Morphology

4.2 Erode

- Erosion is defined as the intersection of a region with the structuring element. Erosion of image A using an SE B is defined as

$$C = A \otimes B = \bigcap_{b \in B} (A)_{-b}$$

where (A)-b represents the translation of A by b. For each nonzero element bi,j of B, A is translated by i,j and summed into C using the AND operator



4. Morphology
4.3 Erode

SE 3x3

0	0	0
1	1	0
0	0	0

Original Image

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

Eroded Image

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

SE 3x3

1	1	1
1	1	1
1	1	1

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

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

4. Morphology

4.3 Erode

SE

3x3

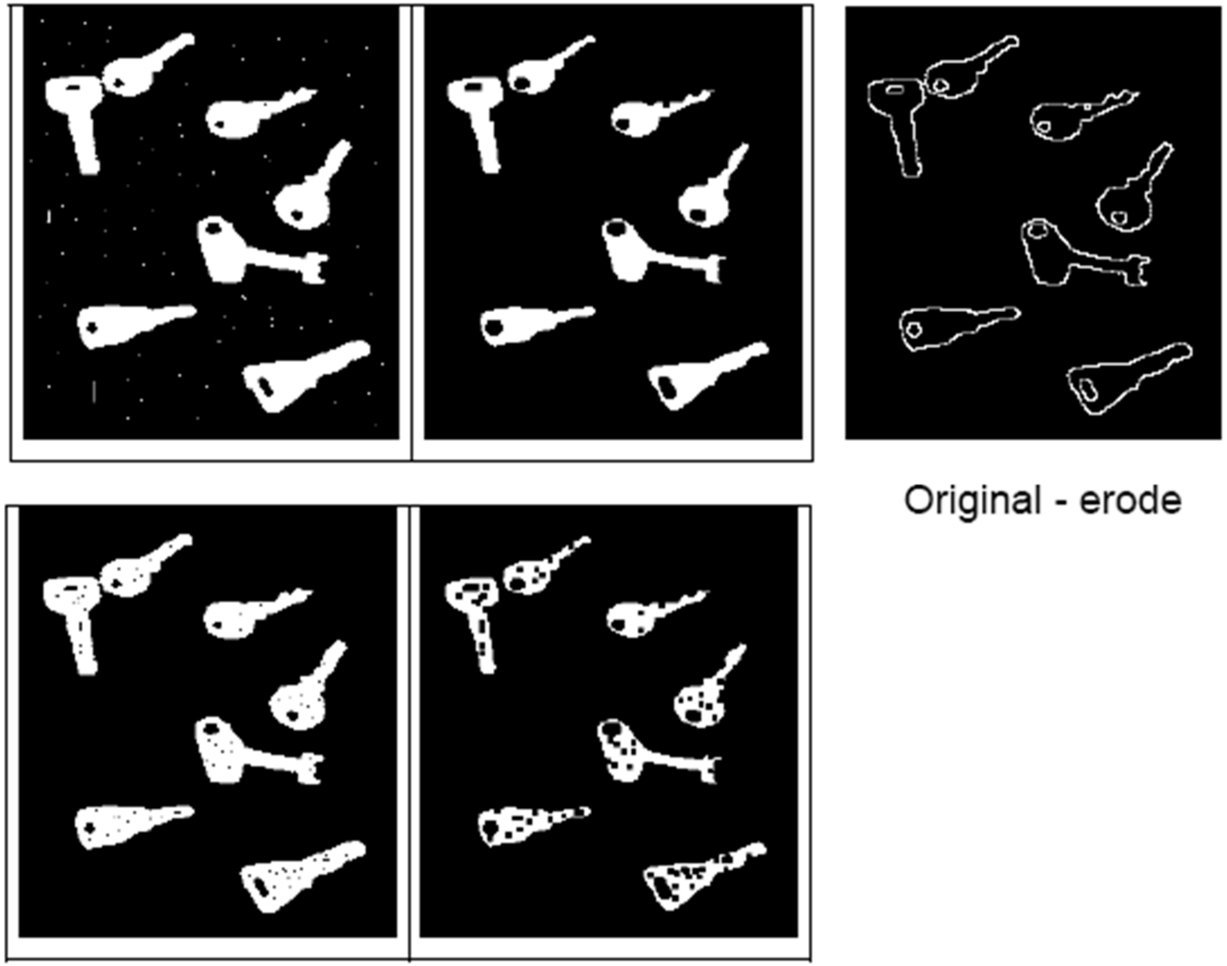
```
1 1 1
1 1 1
1 1 1
```

```
00000000000000000000
00111000000000000000
00111000001110000000
00111000001110000000
00000000001110000000
00000001110000111000
00000001110001111100
01110001110001111100
01110000000001111100
01110000000000111000
```

```
00000000000000000000
00000000000000000000
00010000000000000000
00000000000100000000
00000000000000000000
00000000000000000000
00000000100000010000
00000000000000111000
00100000000000010000
00000000000000000000
```

4. Morphology

4.3 Erode

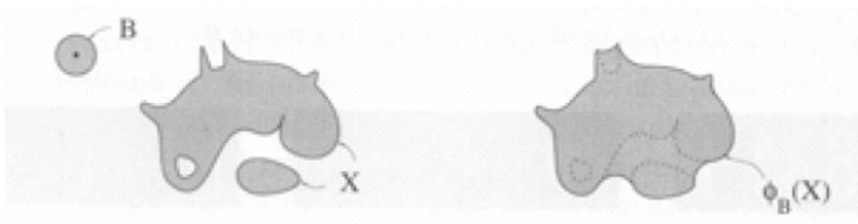


Original - erode

4. Morphology
4.4 Open and Close

- Combining dilate and erode to smooth while preserving region morphology
- **Open** = dilate (erode(A,B))
  - Removes small objects

- **Close** = erode (dilate(A,B))
  - Removes small holes



- Open and Close are dual of each other, not inverse!

5. Using more than one Image

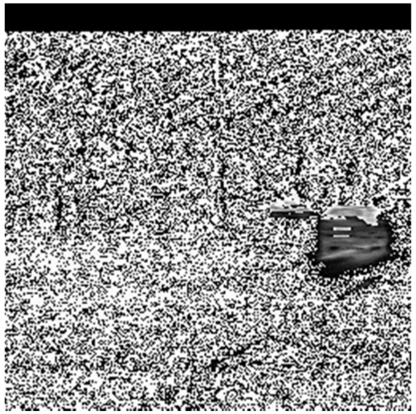
5. Using more than one image
5.1 Object detection

f1

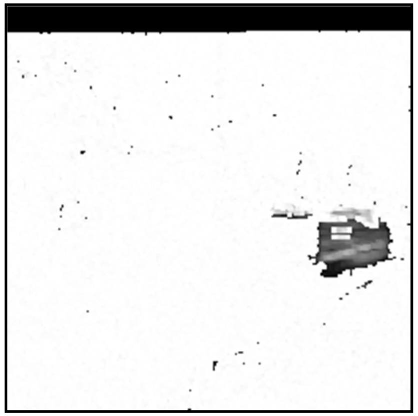
f2

Background subtraction

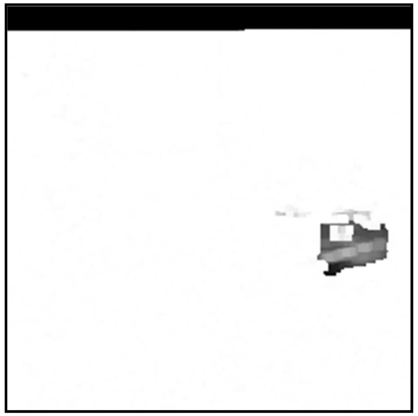
```
subtraction {                 for
each pixel do
        g(x,y)= f1(x,y) - f2(x,y)
}
```



g



dilate(g)



dilate(dilate(g))

5. Using more than one image
5.1 Object detection

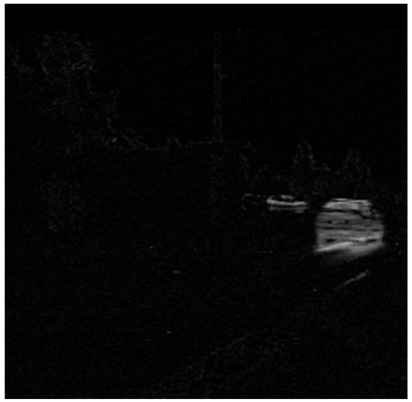
f1

f2

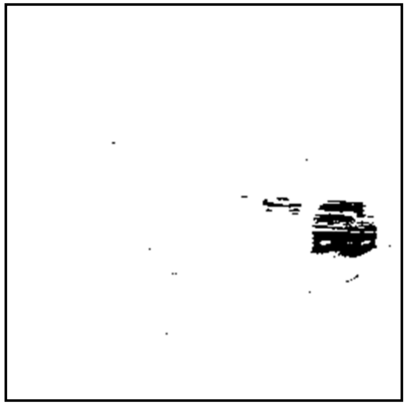
subtraction { for each pixel do

$$g(x,y) = |f1(x,y) - f2(x,y)|$$

}



g



g thresholded by 50

5. Using more than one image
5.2 Template Matching

Match-based consists on moving an image (sub-image or template) across the current image and computing a similarity measure at each position

How to compute similarity:

By subtracting? color, texture? Histogram comparison?

The trick:

Coarse-to-fine-search and subsampling: increasing the step size for moving the template across the image and subsampling the template

5. Using more than one image
5.2 Template Matching

Template – ideal representation of the object

Problems: Rotations, perspective and Illumination changes?
Additional techniques are required

-Pre-processing (for illumination,...)
-Registration (allow other transformations)

Templa

1. M ... ate around the image
2. Me ... sure its similarity

5. Using more than one image
5.2 Template Matching

- Applications: Where's Wally?

7. References...

- *Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods, 2002*
- *Computer Vision*. L. Shapiro, and G. Stockman, Prentice Hall. 2001
- *Pattern classification.* R.O. Duda, P.E. Hart, G. Stork. New York,John Wiley & Sons. 2001
- *Computer Vision: A modern approach.* D. Forsyth, J. Ponce. Prentice Hall, 2002
- http://homepages.inf.ed.ac.uk/rbf/CVonline/
- Matlab Image Processing Toolbox