ComVis1U2

ComVis1

ComVis1U2

- Spatial filtering (linear and non-linear)
- Mathematical Morphology

ComVis1U2

ComVis1

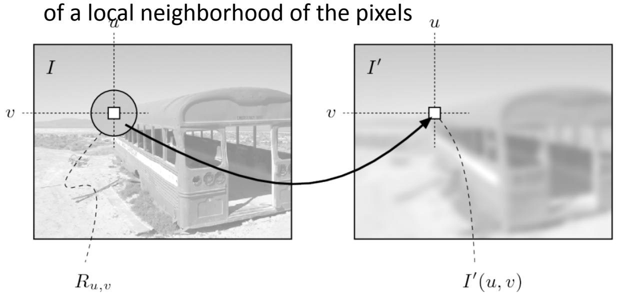
ComVis1U2

- **Spatial filtering (linear and non-linear)**
- Mathematical Morphology

Image Filtering: Spatial Methods

Spatial Methods

Def: modify the pixels in an image based on some function of a local neighborhood of the pixels



$$\mathrm{I}_2(\mathbf{x}) = f(\mathrm{I}_1(\mathbf{x}), V_{\mathrm{I}_1}(\mathbf{x}))$$

Image Filtering: Spatial Methods

Spatial Methods

Two categories:

✓linear based filters,

✓non-linear based filters.

Image Filtering: Spatial Methods

Linear based filters:

✓The simplest

✓Replace each pixel by a linear combination of its neighbors.

✓ The prescription for the linear combination is called the "convolution kernel"

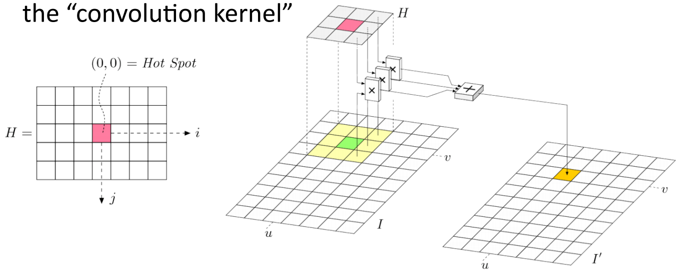


Image Filtering: Spatial Methods

Linear based filters: a convolution operation

✓The simplest

✓Replace each pixel by a linear combination of its neighbors.

✓ The prescription for the linear combination is

10	5	3
4	5	1
1	1	7

Local image data

0	0	0
0	0.5	0
0	1	0.5

kernel

	7	

Modified image data

Image Filtering: Spatial Methods

10	5	3
4	5	1
1	1	7

Local image data

0	0	0
0	0.5	0
0	1	0.5

kernel

	7	

Modified image data

Let W be the kernel (matrix) of size [-n,n]x[-m,m]

$$\mathrm{I}_2(\mathbf{x}) = \sum_{\mathbf{u}\in W} W(\mathbf{u})\mathrm{I}_1(\mathbf{x}+\mathbf{u})$$

Image Filtering: Spatial Methods

How to cope with image boundary?

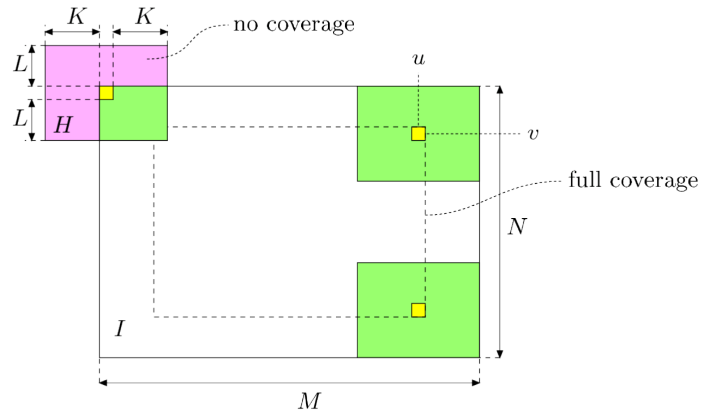


Image Filtering: Spatial Methods

How to cope with image boundary?

CROP

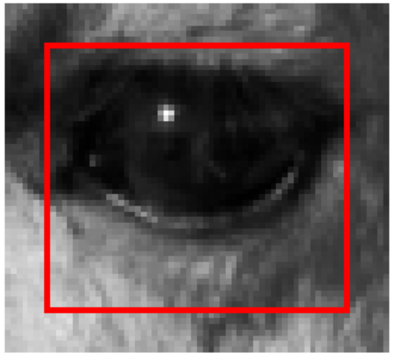


Image Filtering: Spatial Methods

How to cope with image boundary?

PAD

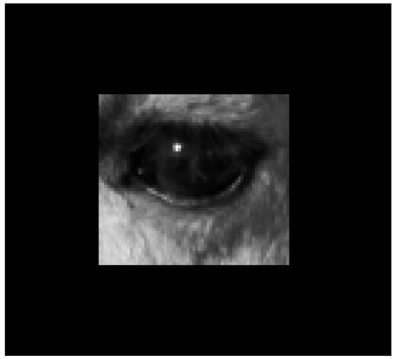


Image Filtering: Spatial Methods

How to cope with image boundary?

EXTEND

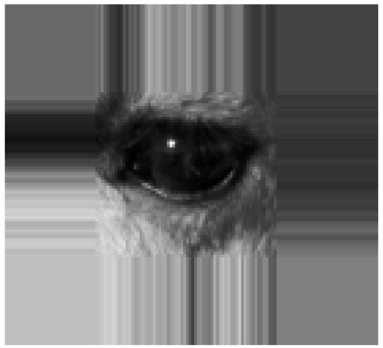


Image Filtering: Spatial Methods

Spatial Methods

Def: modify the pixels in an image based on some function of a local neighborhood of the pixels

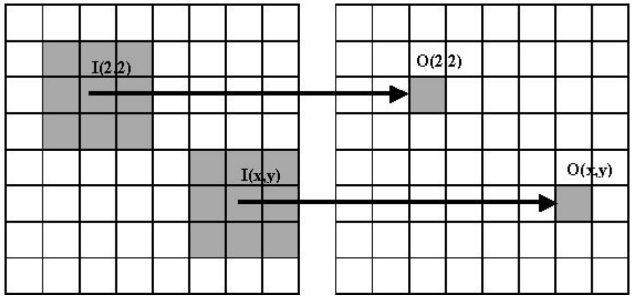


Image Filtering: Spatial Methods

Some classic kernels (average operators)

$$W_1 = \frac{1}{9}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix}$$

$$W_1 = \frac{1}{10}\begin{pmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}$$

$$W_1 = \frac{1}{16}\begin{pmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{pmatrix}$$

Image Filtering: Spatial Methods

Some classic kernels (average operators)

Average (neighbourhood) 5x5

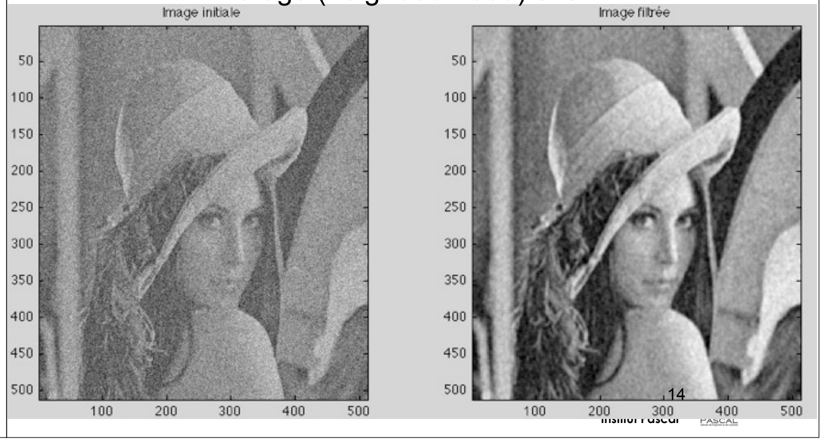


Image Filtering: Spatial Methods

Classic kernels (Gaussian filter)

$$W_1(i,j) = C\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right)$$

$\frac{1}{159}$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

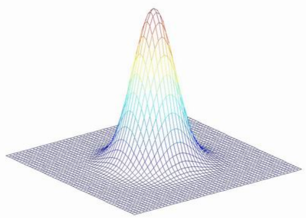


Image Filtering: Spatial Methods

Classic kernels (Gaussian filter)

Gaussian (sig = 3 and support = 5x5)

Gradient approximation kernels (Sobel filter)

Horizontal

$$W_1 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Vertical

$$W_2 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$W_2 = W_1^T$$

Gradient approximation kernels (Sobel filter)

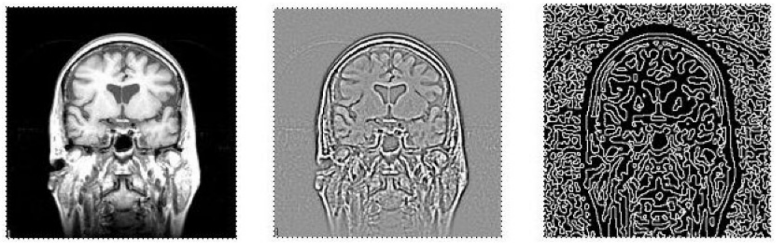
Gradient approximation kernels (Laplacian filter)

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Approximated by:

$$W_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \qquad W_1 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Gradient approximation kernels (Laplacian filter)



(A) Original MR image

(B) Laplacian results

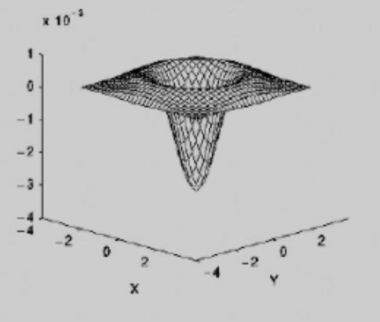
(C) Extraction of the zero crossing of the Laplacian (object edges)

Image Filtering: Spatial Methods

Gradient approximation kernels (Laplacian filter)

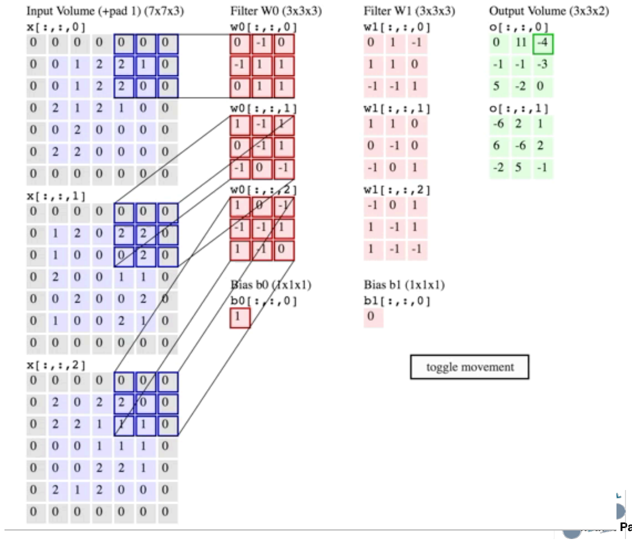
1	1	1
1	8	1
1	1	1

-1	2	-1
2	4	2
-1	2	-1



Deep Convolutional Neural Network (DCNN)

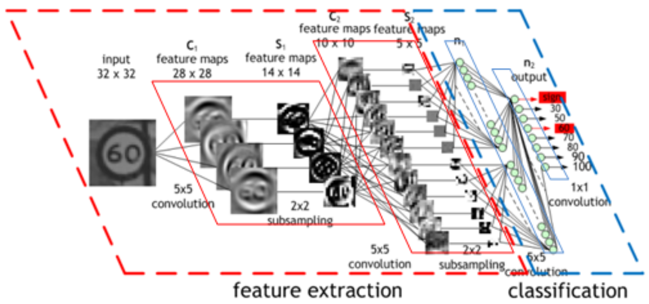
Convolution



Deep Convolutional Neural Network (DCNN)

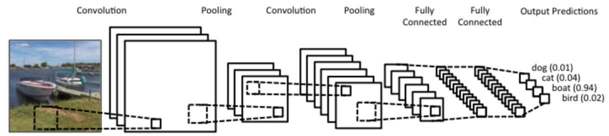
Convolution

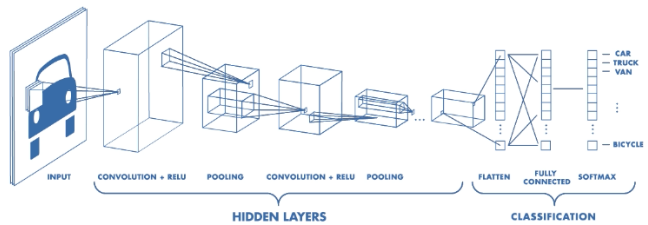
DCNN for trafic sign recognition



Deep Convolutional Neural Network (DCNN)

Convolution





Deep Convolutional Neural Network (DCNN)

Convolution

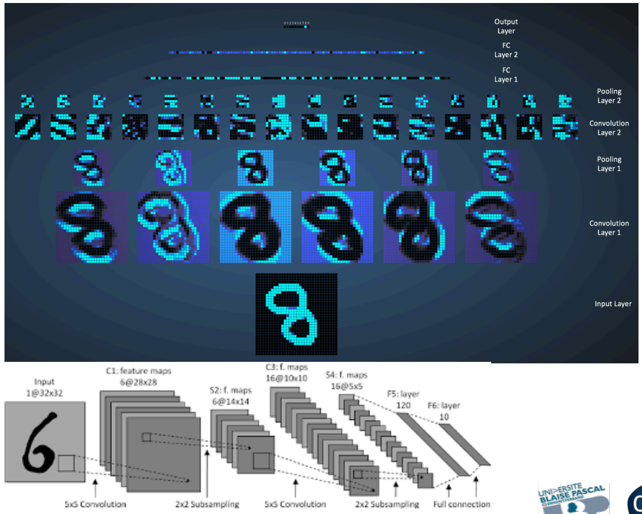


Image Filtering: Spatial Methods

Linear filters: Some properties (separable filters)

$$\mathtt{I}_2(x,y) = \mathtt{I}_1(x,y) * \mathtt{H}_{xy}(x,y)$$

$$\text{If } \mathtt{H}_{x,y}(x,y) = \mathtt{H1}_x(x) * \mathtt{H2}_y(y)$$

$$\text{Then } \mathtt{I}_2(x,y) = [\mathtt{I}_1(x,y) * \mathtt{H1}_x(x)] * \mathtt{H2}_y(y)$$

Image Filtering: Spatial Methods

Linear filters: Some properties (separable filters)

$$\begin{pmatrix} a & b & a \end{pmatrix} * \begin{pmatrix} a \\ b \\ a \end{pmatrix} = \begin{pmatrix} a^2 & ab & a^2 \\ ab & b^2 & ab \\ a^2 & ab & a^2 \end{pmatrix}$$

Image Filtering: Spatial Methods

Non linear filters

- Mathematical Neighborhood Operators

- Calculation within the kernel is defined by non-linear mathematical and statistical operations

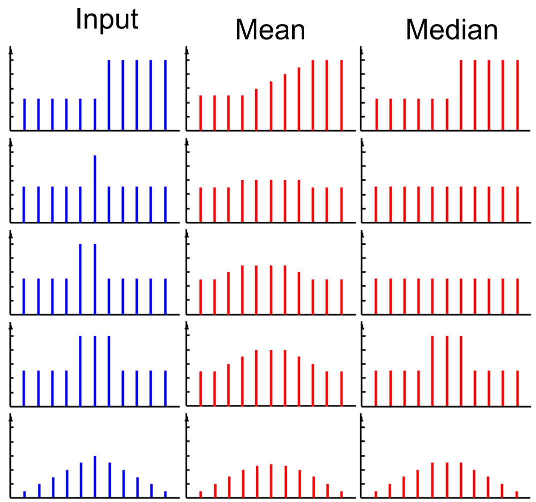
- Minimum
- Maximum
- Median
- Range
- Majority
- Standard deviation, ...

Median filter

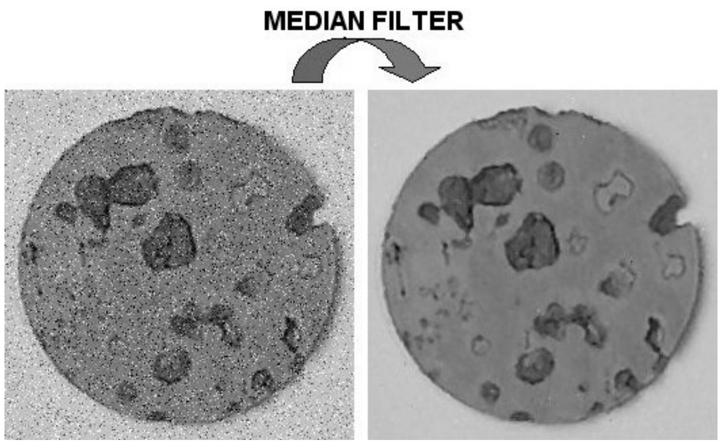
- Robust Filter
- Non-linear operation
- Each pixel is modified according to the median value of it neighbourhood

Median filter (support = 5)

Can be computed using a quick sort algorithm



Median filter



ComVis1

ComVis1U2

- Spatial filtering (linear and non-linear)
- Mathematical Morphology

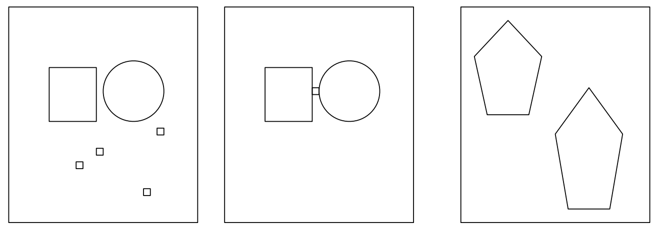
Binary image filtering: Mathematical morphology

MM was originally developed for binary images, and was later extended to grayscale functions and images

born in 1964 from the collaborative work of **Georges Matheron** and **Jean Serra**, at the *École des Mines* de Paris, France

Binary image filtering: Mathematical morphology

What can we do with MM ?



Remove noise separate shapes compare shapes

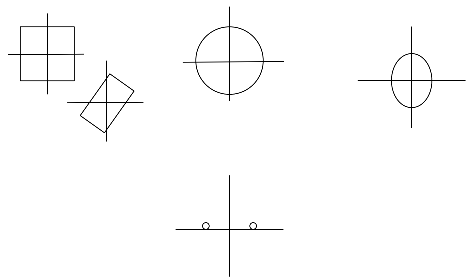
Binary image filtering: Mathematical morphology

Main idea: probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image.

This simple "probe" is called structuring element, and is itself a binary image (i.e., a subset of the space or grid).

Binary image filtering: Mathematical morphology

Some structuring elements



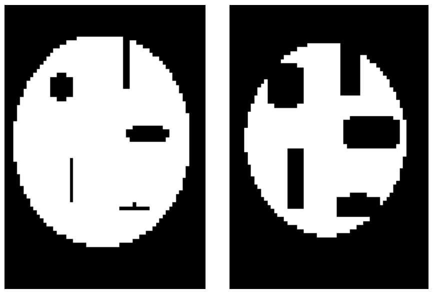
Binary image filtering: Mathematical morphology

Basic operators: erosion

Erosion of the binary image A by the structuring element B:

$$A \ominus B = \{z \in E | B_z \subseteq A\}$$

translation of B by z



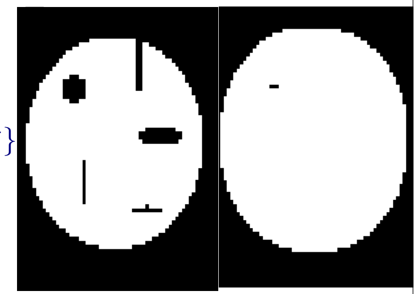
Binary image filtering: Mathematical morphology

Basic operators: dilation

Dilation of the binary image A by the structuring element B:

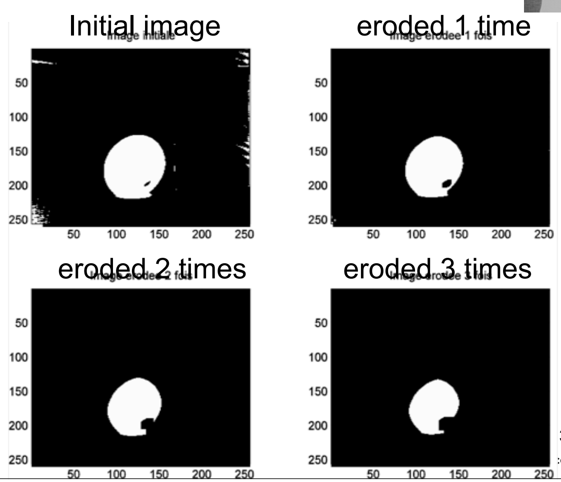
$$A \oplus B = \{z \in E | (B^s)_z \cap A \neq \varnothing\}$$

symmetric of B



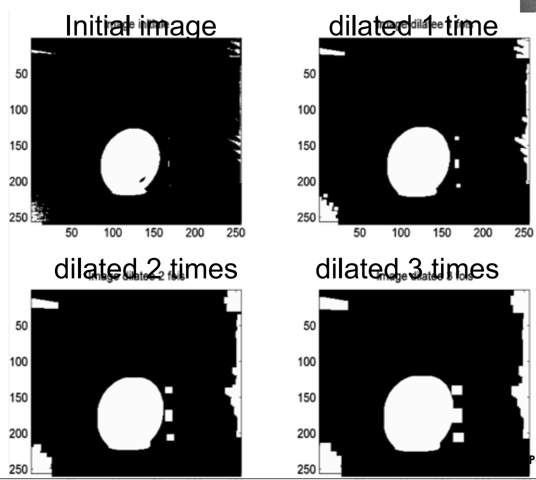
Binary image filtering: Mathematical morphology

Example



Binary image filtering: Mathematical morphology

Example



Binary image filtering: Mathematical morphology

Basic operators: opening

$$A \circ B = (A \ominus B) \oplus B$$

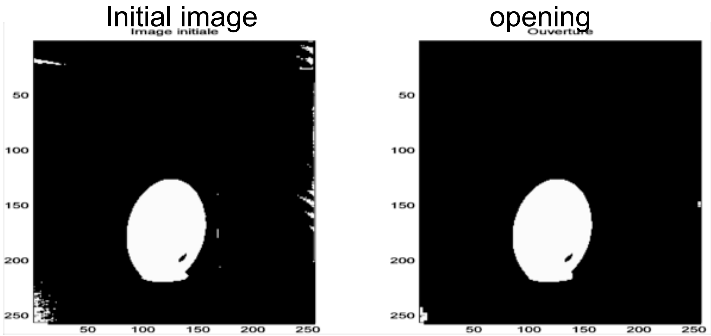
Binary image filtering: Mathematical morphology

Basic operators: closing

$$A \bullet B = (A \oplus B) \ominus B$$

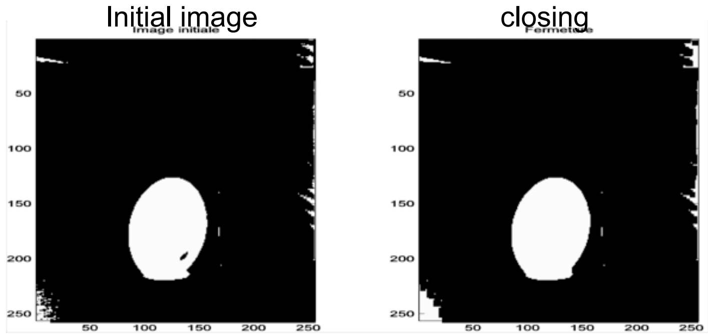
Binary image filtering: Mathematical morphology

Basic operators: opening

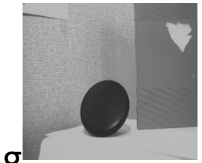


Binary image filtering: Mathematical morphology

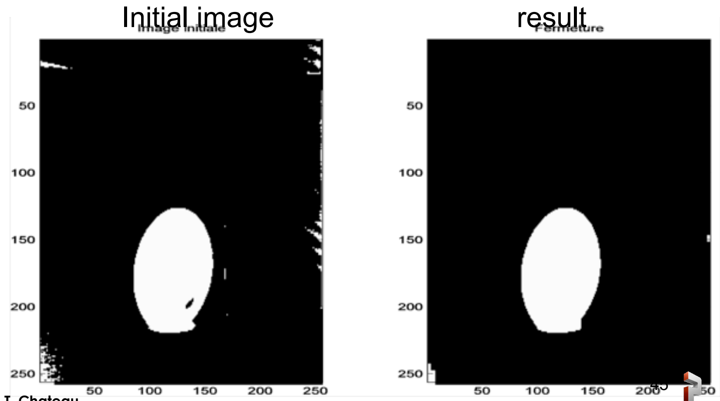
Basic operators: closing



Binary image filtering: Mathematical morphology



Basic operators: opening+closing



Binary image filtering: Mathematical morphology

Some properties

$$A \ominus B \subset A \subset A \oplus B$$

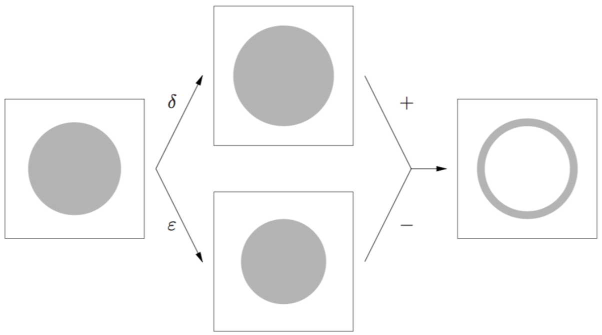
$$A \circ B \subset A \subset A \bullet B$$

$$A \circ (A \circ B) = A \circ B$$

$$A \bullet (A \bullet B) = A \bullet B$$

Binary image filtering: Mathematical morphology

Edge detection



Binary image filtering: Mathematical morphology

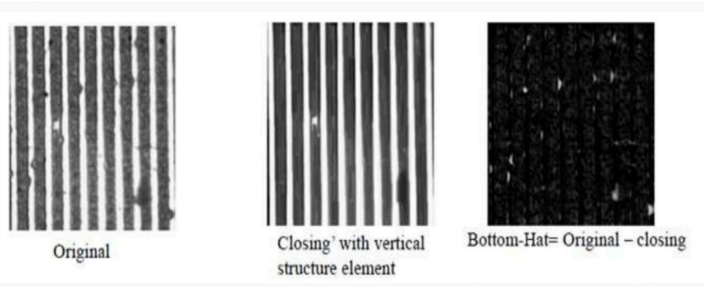
Top hat operation: detect structures of a certain size (white objects on a black background)

$$T_w(f) = f - f \circ b$$

Binary image filtering: Mathematical morphology

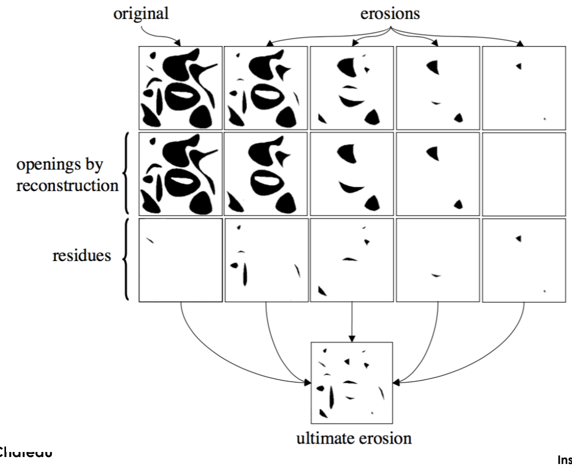
Bottom(black) hat operation: detect structures of a certain size (black objects on a white background)

$$T_b(f) = f \bullet b - f$$



Binary image filtering: Mathematical morphology

Ultimate Erosion



Binary image filtering: Mathematical morphology

References

J. Serra, Image Analysis and Mathematical Morphology, Academic Press, New-York, 1982.

J. Serra (Ed.), Image Analysis and Mathematical Morphology, Part II: Theoretical Advances, Academic Press, London, 1988.

P. Soille, Morphological Image Analysis, Springer-Verlag, Berlin, 1999.

Image Processing

Exercices

Filtering

Compute the filtered image for :

1) $W_1 = \frac{1}{9}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix}$

2) a median filter (3x3 support)

Conclude

2	2	8	8	8
2	2	8	0	8
2	0	8	8	8
2	2	8	0	8
2	2	8	8	8

Exercices

Mathematical Morphology

Propose a binary structured element and a set of morphological transformations to remove the noise and close the square

Noise

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