

# Medical Imaging : Image Filtering & Segmentation

Hervé Delingette

Epione Team

[Herve.Delingette@inria.fr](mailto:Herve.Delingette@inria.fr)



# Course Schedule

[https://www-sop.inria.fr/asclepios/cours/MS\\_DS\\_2023\\_2024/](https://www-sop.inria.fr/asclepios/cours/MS_DS_2023_2024/)

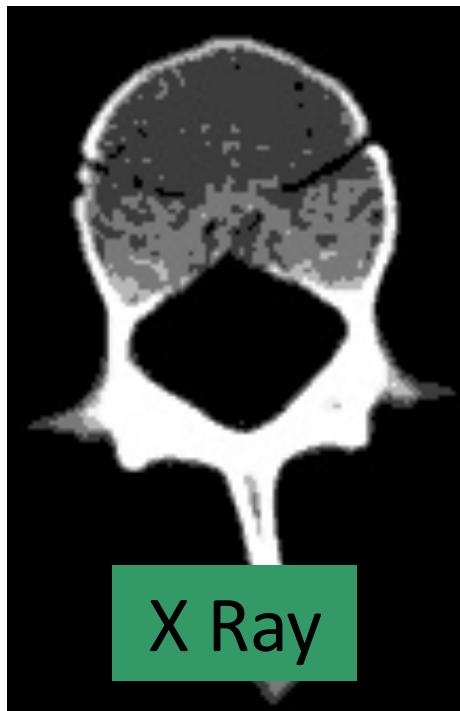
- Tuesday January 09, 14:00-17:00 [Hervé Delingette] Introduction to Medical Image Acquisition
- Tuesday January 16 13:30-16:30 [Xavier Pennec] Medical Image Registration
- Tuesday January 23 13:30-16:30 [Hervé Delingette] Image Filtering and Image segmentation based on clustering
- Tuesday January 30 13:30-16:30 [Xavier Pennec] Statistics on Riemannian manifolds and Lie groups
- Monday January 29 13:30-16:30[Hervé Delingette] Connexity and Shape Constrained Image segmentation
- Tuesday February 13 13:30-16:30 [Xavier Pennec] Non linear and diffeomorphic Registration
- Tuesday February 20, 13:30-16:30 [Hervé Delingette] Shape constraints in Image Segmentation
- Tuesday February 20, 9:30-12:30 [Xavier Pennec] Computational Anatomy
- Tuesday March 5 2023 13:30-15:00, [XP & HD] EXAM

# 1. Medical Image Representation & Visualization

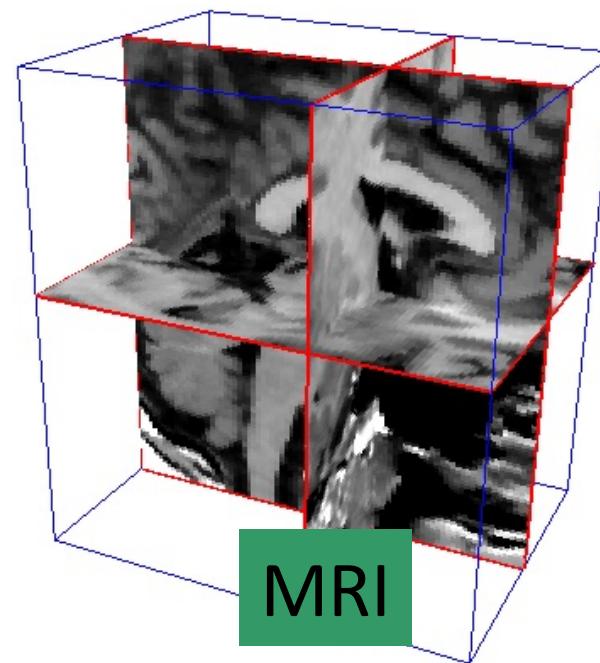
- 1.1 Image representation : discrete or continuous
- 1.2 Image Visualization

# Medical Imaging Classification (1)

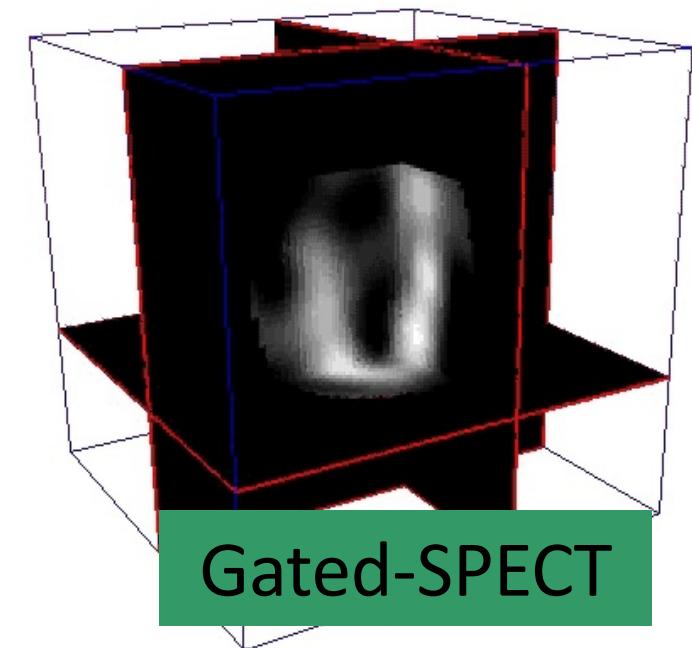
- Dimensionality



2D



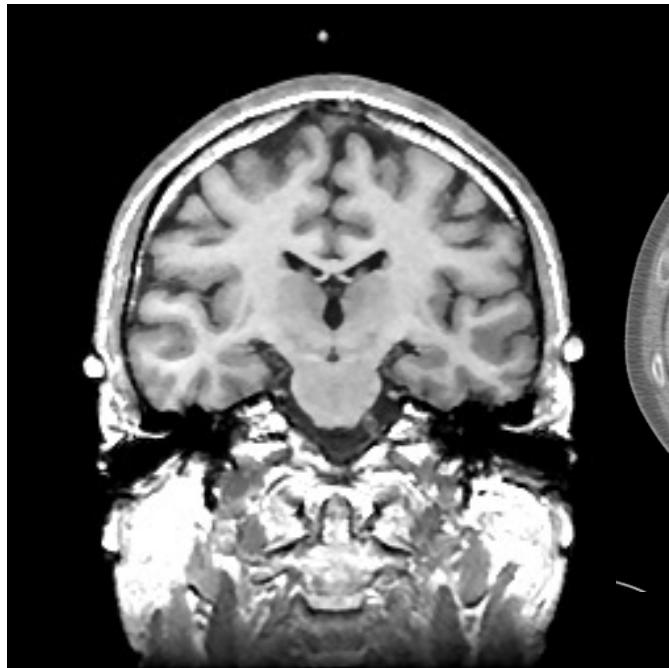
3D



4D (3D+T)

# Medical Imaging Classification (2)

- Anatomical vs functional Imagery

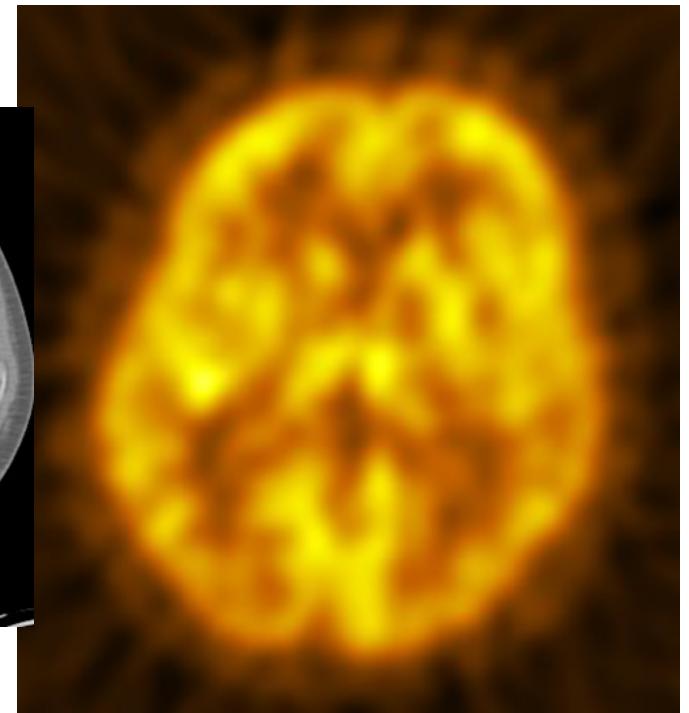


MRI

Anatomical



CT with  
contrast agent



PET scan

Functional

# Medical Image Processing vs Computer Vision

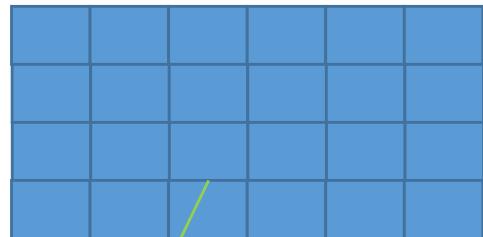
	Computer Vision	Medical Image Processing
	<p>Projective Geometry</p> <p>Occluding Objects</p> <p>Intensity depends on lighting</p>	<p>Complex Image Formation</p> <p>Large Datasets</p> <p>Patient Images</p>
	<p>Easy to acquire</p> <p>Low dimensionality</p>	<p>Cartesian Geometry</p> <p>Statistics Information</p> <p>Intensity links to physics</p> <p>Patient Images</p>

# Discrete Image Representation (1)

- Domain is considered as a 2D/3D regular grid

2D Array  $I$

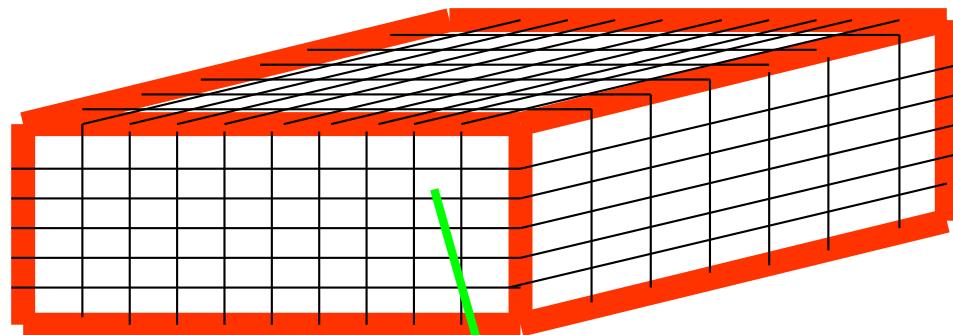
$I[\text{col}][\text{row}]$



Pixel=picture element

3D Array  $I$

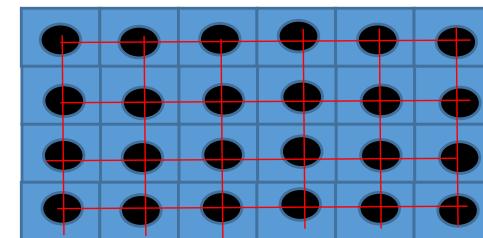
$I[\text{plane}][\text{col}][\text{row}]$



Voxel=  
Volume Element

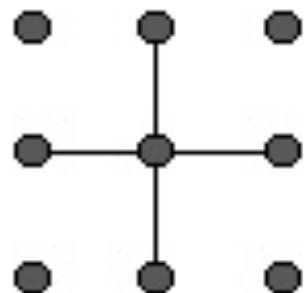
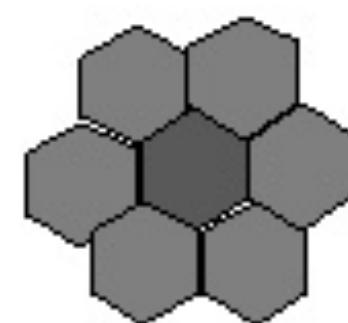
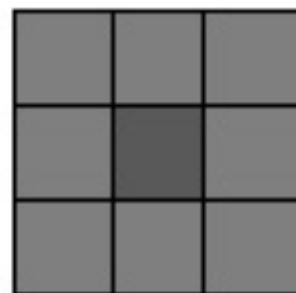
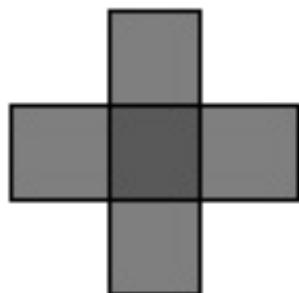
# Discrete Image Representation (2)

- Pixel / Voxel values can be :
  - Discrete :
    - Integer : char (MRI), signed short (CT-scan)
    - Labels of structures
  - Continuous :
    - Float / double
- Images can be seen as a graph
  - Nodes are pixel / voxel centers
  - Edges between adjacent elements
  - Grid Duality

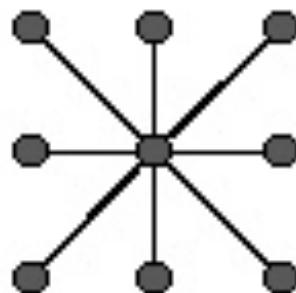


# Neighborhood

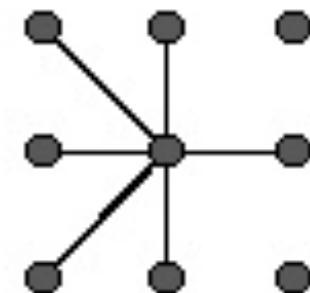
- Different types of neighborhood in 2D



4 -neighborhood



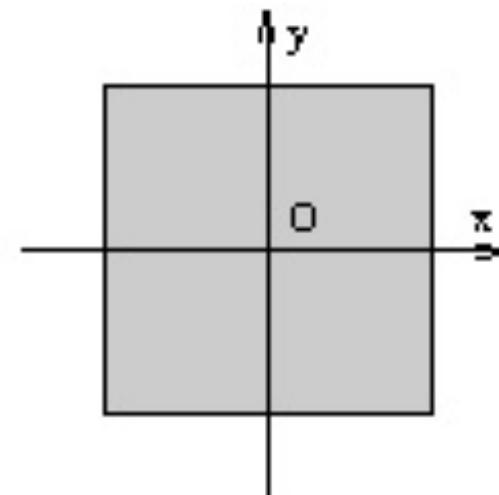
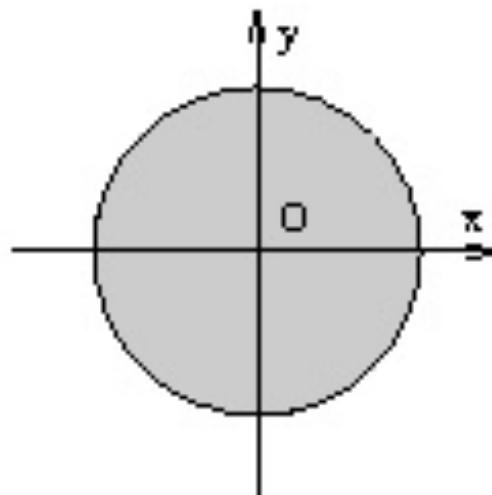
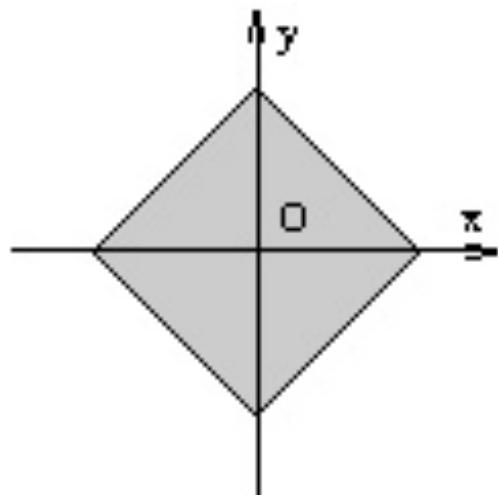
8 -neighborhood



6 -neighborhood

# Neighborhood

- Some generalizes to higher dimensions
- Corresponds to a choice of metric norm



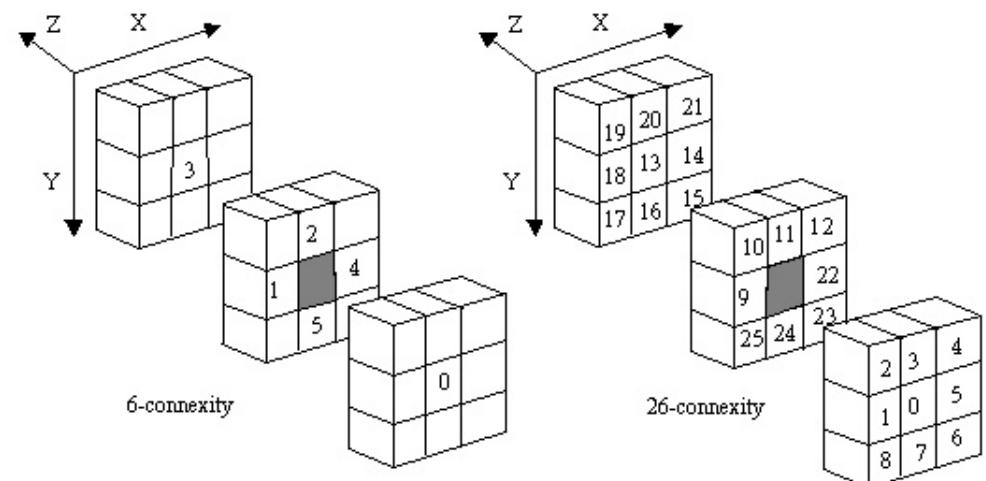
$$D_1(x, y) = \sum_{i=1}^n |y_i - x_i|$$

$$D_2(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

$$D_\infty(x, y) = \max_{i=1 \dots n} |y_i - x_i|$$

# Neighborhood

- 3 types of neighborhood for a 3D image :
  - 6-neighborhood : adjacency through faces
  - 18-neighborhood : adjacency through faces and edges
  - 26-neighborhood : adjacency through faces and edges and vertices



# Continuous Image representation

- Image seen as 2D or 3D Fields :  
 $I(x), x \in \mathbb{R}^n, n = 2,3$
- Requires to define interpolation functions

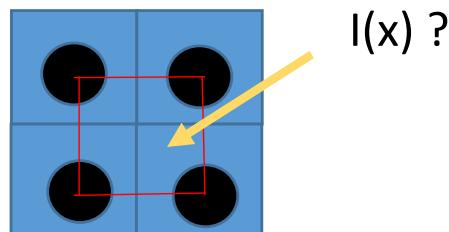
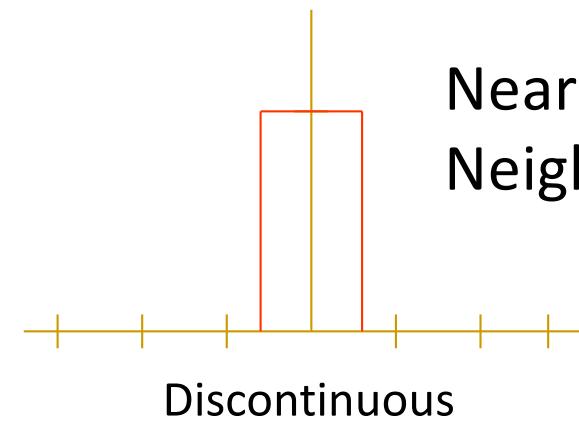
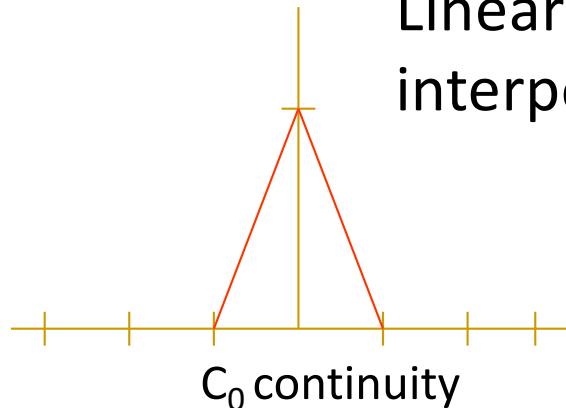


Image Domain \ Image Value	Discrete	Continuous
Discrete	Array of Int	<del>Field of Integer</del>
Continuous	Array of Float	Field of Float

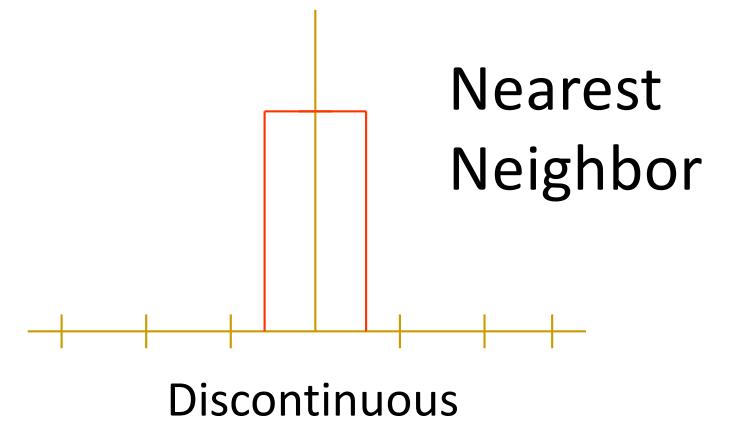
# 1D Interpolation functions



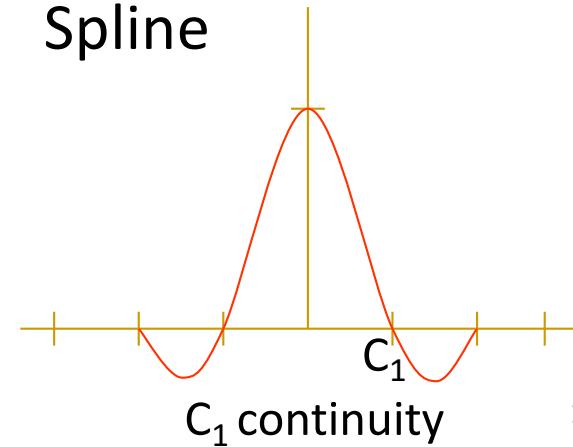
Nearest  
Neighbor



Linear  
interpolation



$C_{00}$  continuity

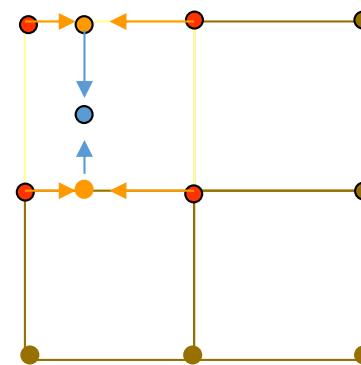
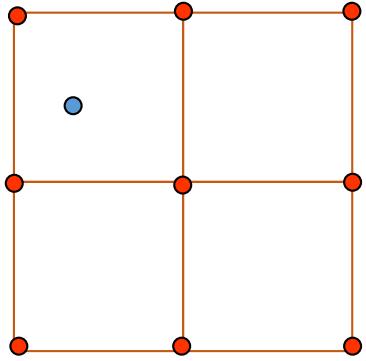


Spline

$C_1$  continuity

# Bilinear Interpolation (2D Field)

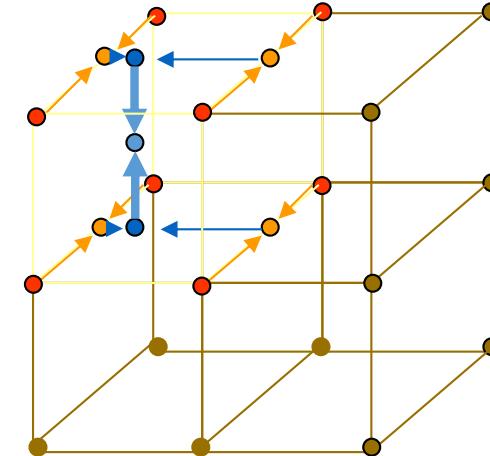
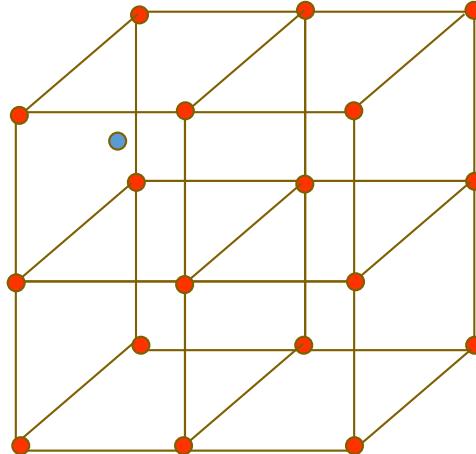
- Bilinear Interpolation : 3 linear interpolations



$$I(u, v) = (1 - u)(1 - v)I_{i,j} + u v I_{i+1,j+1} + \\ (1 - u)v I_{i,j+1} + u (1 - v)I_{i+1,j}$$

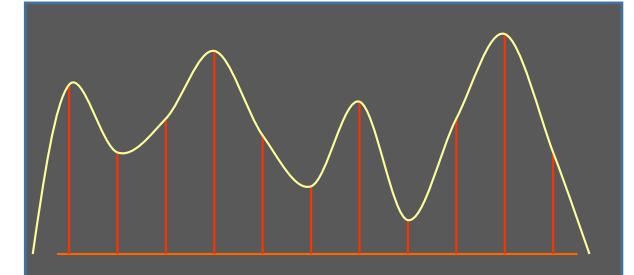
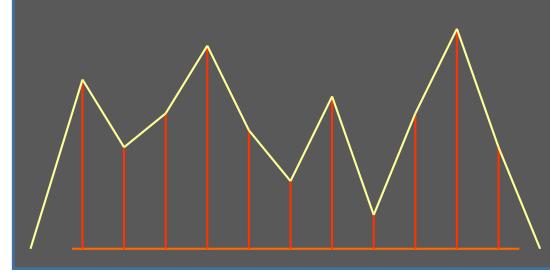
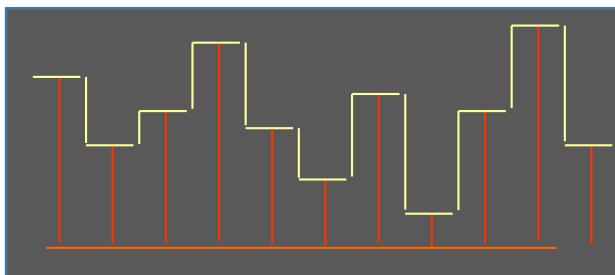
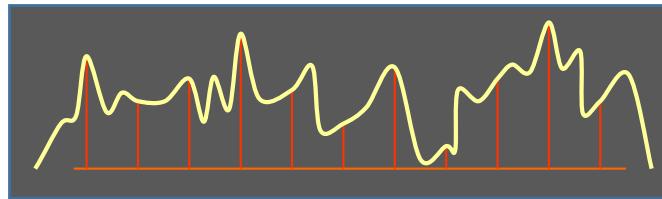
# Trilinear Interpolation (3D Field)

- 7 linear interpolations



$$\begin{aligned} I(u, v, w) = & (1 - u)(1 - v)(1 - w)I_{i,j,k} + u v w I_{i+1,j+1,k+1} + \\ & (1 - u)v w I_{i,j+1,k+1} + u (1 - v)w I_{i+1,j,k+1} + \\ & (1 - u)v (1 - w)I_{i,j+1,k} + u (1 - v)(1 - w)I_{i+1,j,k} + \\ & u v (1 - w)I_{i+1,j+1,k} + (1 - u) (1 - v)w I_{i,j,k+1} \end{aligned}$$

# Image interpolation



Hervé Delingette

# Medical Image Format

- Industrial standard :
  - **DICOM** : Digital Imaging and COmmunications in Medicine
  - More a communication standard for interoperability than an image format
- Academic standard :
  - Must support volumetric images, generic voxel format (short, double, array of double), voxel size, metadata
  - ITK based : MHA, MHD
  - NIFTI : Neuroimaging Informatics Technology Initiative

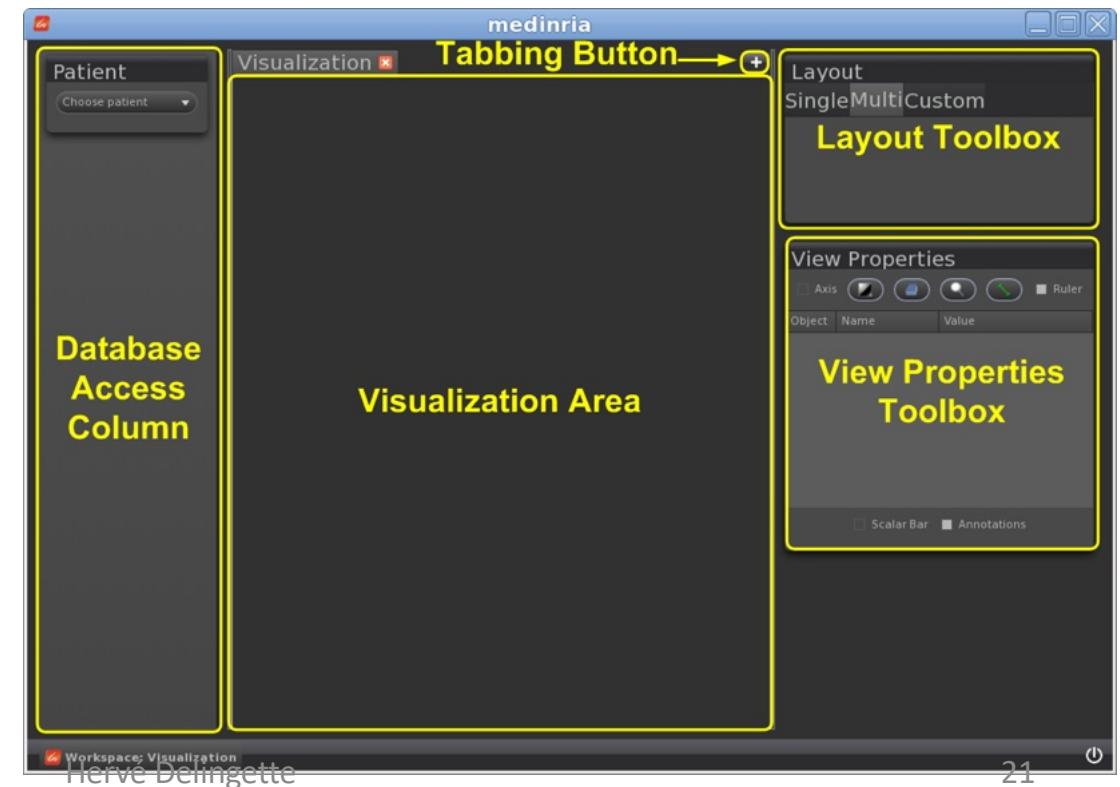
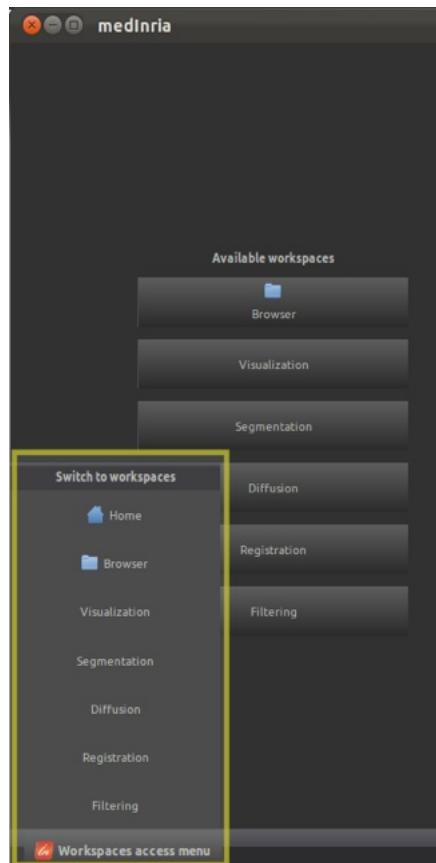
# 1. Medical Image Representation & Visualization

- 1.1 Image representation : discrete or continuous
- 1.2 Image Visualization

# Visualizing Medical Images

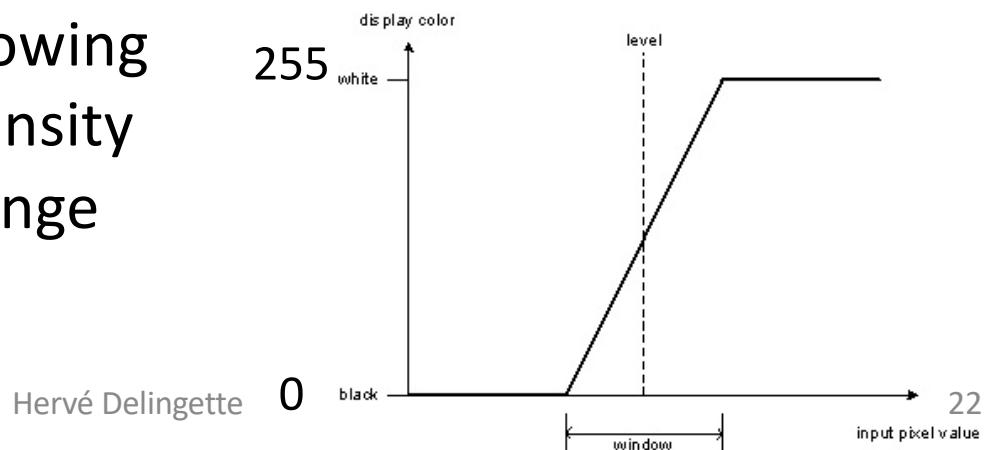
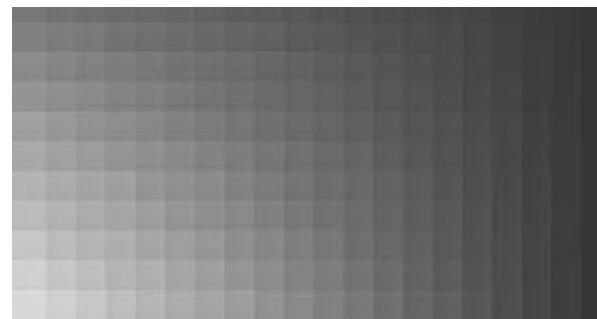
- MedInria : <https://med.inria.fr>
  - Free & Multiplatform, plugin based.

Quick Access  
Menu



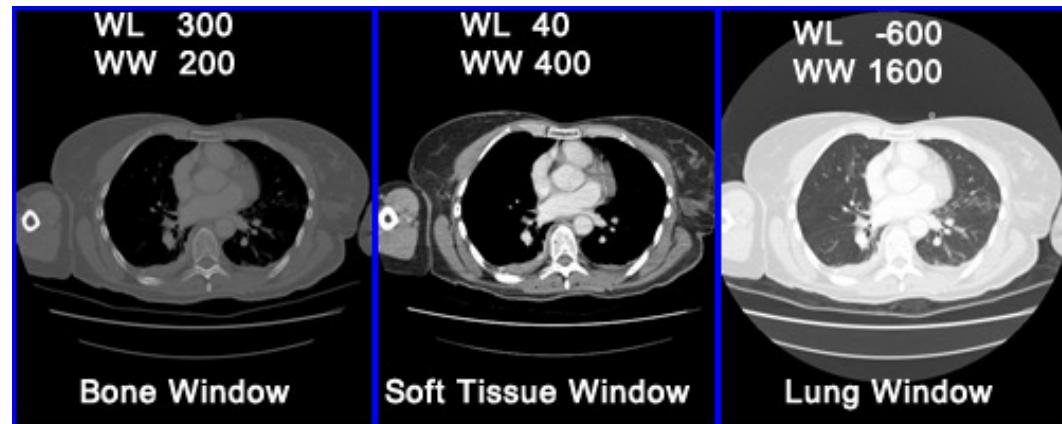
# Visualizing Medical Images

- Windowing
  - CT images are coded on 2 bytes ( $2^{16}$  values).
  - The human eye can only see a limited (200 ?) number of shades of grey !
- Need to perform windowing i.e. map a range of intensity values in the [0,255] range

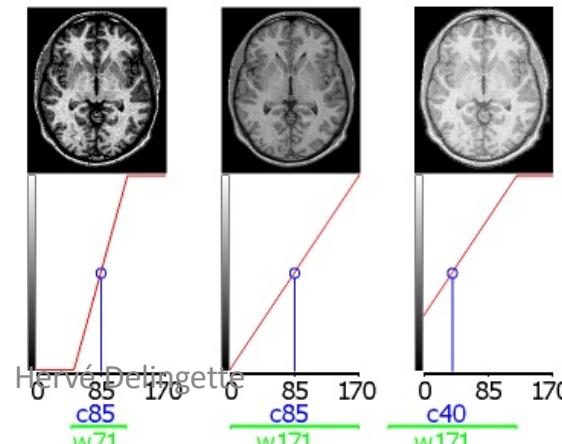


# Visualizing Medical Images

- Windowing
  - Predefined windows on CT as Hounsfield units are absolute



Brain MRI Image



# Visualizing Medical Images

- Volumetric Images

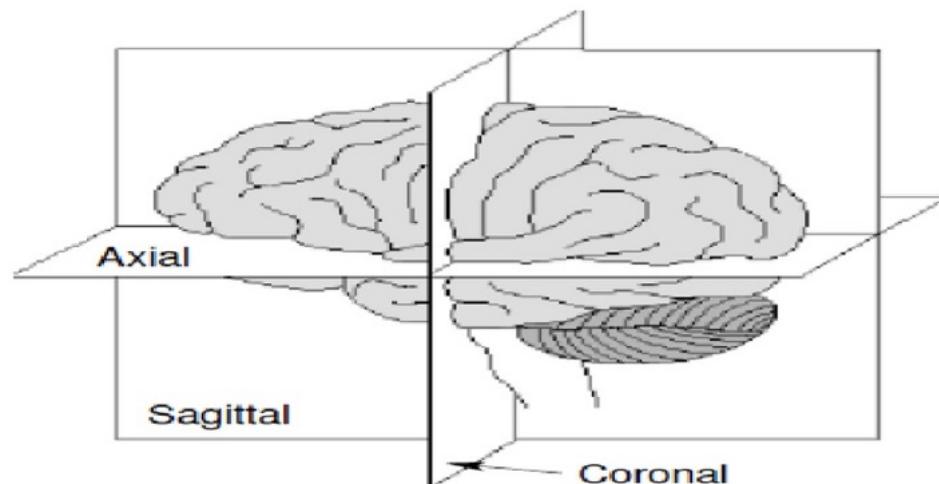
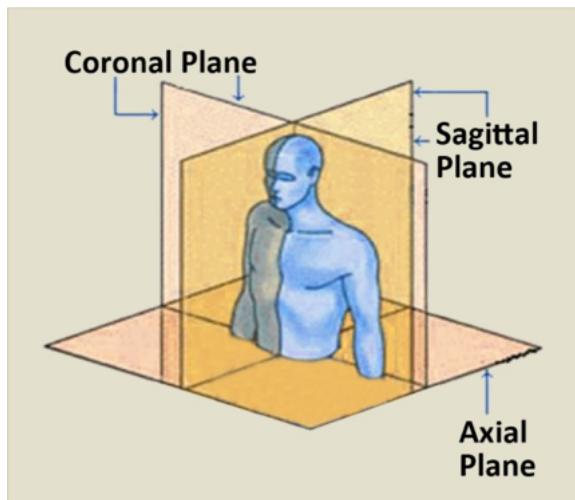
How to visualize a 3D image on a 2D screen ?

Multiplanar Reformatting



# Visualizing Medical Images

- Volumetric Images
  - Radiological Convention to name the 3 orthogonal slices.
  - Axial, coronal and sagittal



# 2.0 Image Filtering

- **2.1 Taxonomy of Filters**
  - Linear vs Non-Linear
  - Separable vs Non-Separable
  - Recursive Filters
- **2.2 Smoothing Filters :**
- **2.3 Gradient Filters :**
- **2.4 Image Contour Extraction**
- **2.5 Mathematical Morphology**

# 2.0 Image Filtering

- 2.1 Taxonomy of Filters
  - Linear vs Non-Linear
  - Separable vs Non-Separable
  - Recursive Filters
- 2.2 Smoothing Filters :
  - Isotropic vs Anisotropic
  - Recursive
  - Non-Local
- 2.3 Gradient Filters :
  - Discrete vs Continuous
- 2.4 Image Contour Extraction
- 2.5 Mathematical Morphology

# Image Filtering

- Image Filtering consists in extracting local information : basis for feature extraction
- Different types of filters:
  - **Linear** := Convolution with a function (filter)
  - **Non-linear**
  - **Separable** : a  $nD$  filter may be written as the composition of several 1D filters
  - **Non-separable**

# Image Convolution

- convolution in the continuous domain

$$(I * f)(M_0) = \int_{\text{Image}} I(M)f(M_0 - M)dM$$

- convolution in the discrete domain

$$(I * f)(M_0) \approx \sum_{\text{Image}} I(M)f(M_0 - M)$$

- Notation

$$\bar{I} = I * \begin{pmatrix} A_{00} & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & A_{2M,2M} \end{pmatrix}$$

Matrix of size  $(2M+1) \times (2M+1)$

$$\bar{I}(x, y) = \sum_{i=-M}^M \sum_{j=-M}^M I(x+i, y+j) * A(M+i, M+j)$$

# Properties of Convolution

- Transitive combinaison of convolution filters

$$I * (f * g) = (I * f) * g$$

- Derivatives of convolution filters

$$\begin{aligned}(I * f)^{(n)} &= I^{(n)} * f \\ &= I^{(k)} * f^{(n-k)} = I * f^{(n)}\end{aligned}$$

- This property is not verified for discrete convolutions

# Separable Convolution Filters

- Separable filters lead to efficient implementation
  - In dimension  $n$ , the application of a convolution filter  $f$  of size  $p^n = p \times p \times \cdots \times p$  on an image  $I$  of size  $d^n$ , requires :
    - number of multiplications :  $d^n p^n$
    - number of additions :  $d^n(p^n - 1)$
  - if  $f$  is separable along  $n$  directions, it requires :
    - nombre de multiplications :  $n(d^n p)$
    - nombre d'additions :  $n(d^n(p-1))$

# Convolution

## ■ Derivation and Separation

$$\begin{aligned}\frac{\partial(I * (f^x(x) * f^y(y)))}{\partial x} &= I * \frac{\partial(f^x(x) * f^y(y))}{\partial x} \\ &= I * f_x^x(x) * f^y(y)\end{aligned}$$

# Example : Mean Filter

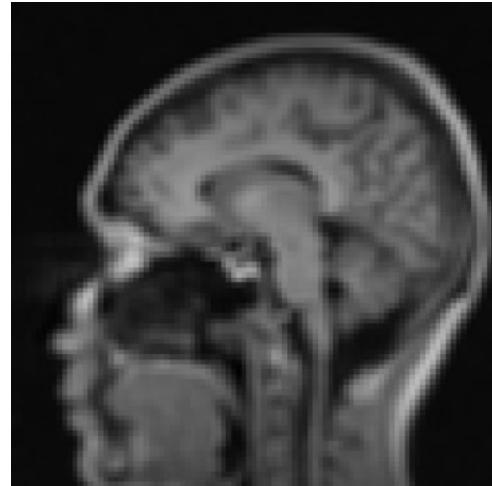
$$\bar{I}(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 I(x+i, y+j)$$

■ Can be written as :

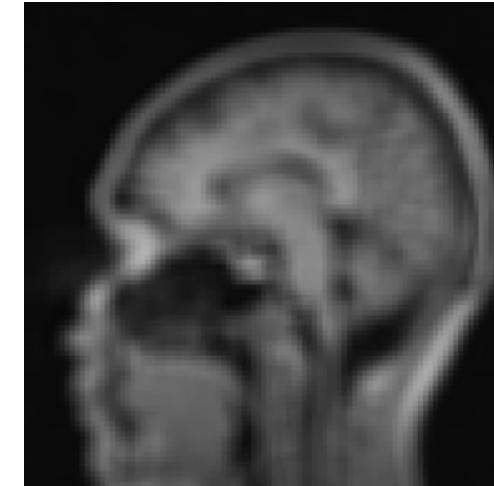
$$\bar{I} = I * \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$



Input MR Image



MR Image with  
3x3 Mean Filter



MR Image with  
5x5 Mean Filter

## Example : Mean Filter

### ■ Separable

$$\bar{I}(x, y) = \frac{1}{3} \sum_{i=-1}^1 \left( \frac{1}{3} \sum_{j=-1}^1 I(x+i, y+j) \right)$$

### ■ Can be written as

$$\bar{I} = \left( I * \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \right) * \frac{1}{3} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

# Non Linear Separable Filters

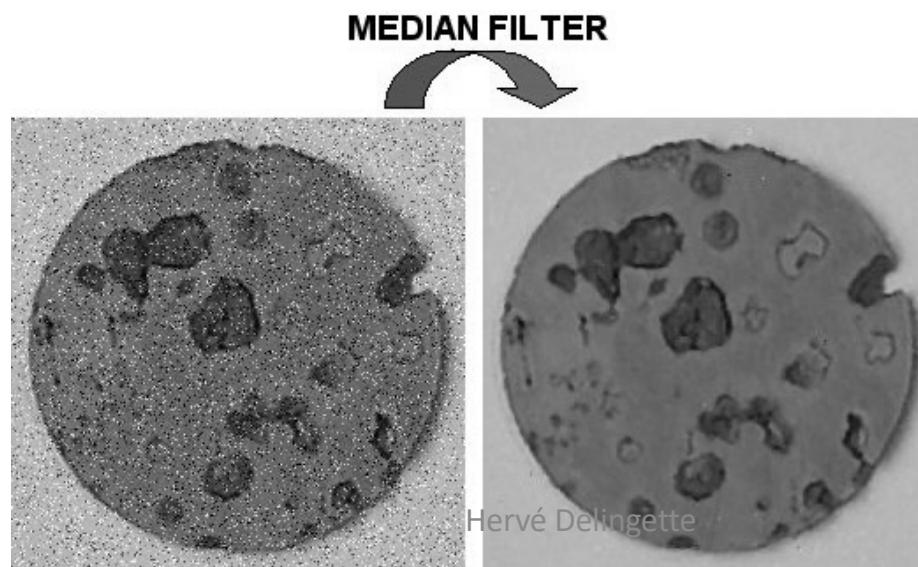
- maximum and minimum filters

$$I \oplus \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = \max_{i=-1\dots 1, j=-1\dots 1} I(x+i, y+j)$$
$$= \max_{j=-1\dots 1} \left( \max_{i=-1\dots 1} I(x+i, y+j) \right)$$

# Non Linear Non Separable Filters

- Median Filter
  - ⇒ Remove Impulse noise
    - For a given voxel  $M$ , values of its  $n$  neighbors are sorted
    - Output value for  $M$  is the value of rank  $n/2$

Use moving window with fast sort algorithms (*quicksort*, ...)



# Recursive Filters

- Limitation of discrete convolution :
  - Finite support
  - Computation load increases with support size
- Discrete 1D convolution filter of size M implemented as:

$$y[n] = \sum_{m=n-M}^{n+M} a[n-m]x[m]$$

*aka* Finite Impulse Response  
FIR Filter

- Recursive Filters lead to infinite support filters

$$y[n] = \sum_{m=n-M}^n a[n-m]x[m] + \sum_{m=n-M}^{n-1} b[n-m]y[m]$$

- Need to combine causal and anti-causal filters

*aka* Infinite Impulse Response  
IIR Filter

# 2.0 Image Filtering

- 2.1 Taxonomy of Filters
- 2.2 **Smoothing Filters**
  - Isotropic vs Anisotropic
  - Recursive
  - Non-Local
- 2.3 Gradient Filters :
  - Discrete vs Continuous
- 2.4 Image Contour Extraction
- 2.5 Mathematical Morphology

# Image Smoothing

- Mean Filter, Triangular Filter
- Gaussian Filter  $G(\mu; \sigma)$ :
  - infinite support therefore inefficient computation
  - Equivalent to solving an isotropic diffusion problem
    - 1D problem

$$\frac{\partial I}{\partial t} = k \frac{\partial^2 I}{\partial x^2}$$

$$\text{Minimize } E = \frac{1}{2} k \int \|\nabla I\|^2 d\Omega$$

$I(x, t)$



$$I(x, t) \approx \int \exp\left(-\frac{(x-y)^2}{4kt}\right) I(y, 0) dy = G(\sqrt{4kt}) * I(x, 0)$$

# Binomial Filters

$$\left[ \frac{1}{2} \quad \frac{1}{2} \right]^n$$

- Separable and finite support
- Iterative convolution of a simple filter

$$n=2 \quad \frac{1}{4} [1 \quad 2 \quad 1]$$

$$n=3 \quad \frac{1}{8} [1 \quad 3 \quad 3 \quad 1]$$

$$n=4 \quad \frac{1}{16} [1 \quad 4 \quad 6 \quad 4 \quad 1]$$

- Coefficient of rank  $k$  in a filter of size  $n$  given by (see Pascal Triangle)  
$$\frac{1}{2^n} \binom{n}{k} = \frac{n!}{2^n k!(n-k)!}$$
- Converge towards a Gaussian distribution  $G\left(\frac{n}{2}; \frac{\sqrt{n}}{2}\right)$
- Separable Filters

$$\frac{1}{4} [1 \quad 2 \quad 1] * \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

# Gaussian Recursive Filtering

- Approximate Gaussian Filter with recursive filters
- Two approaches
  - Sum of causal & anti-causal filters

Deriche, R., 1992, Recursively implementing the Gaussian and its derivatives:  
Proceedings of the 2nd International Conference on Image Processing,  
Singapore, p. 263–267

- Product of causal & anti-causal filters

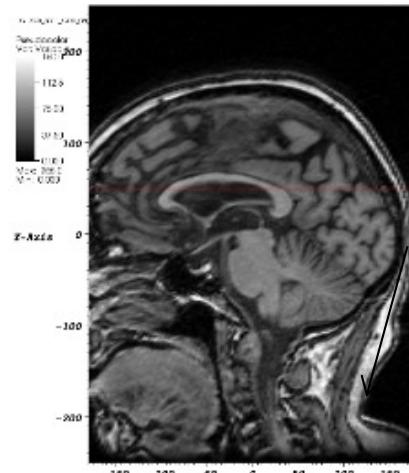
van Vliet, L., Young, I., and Verbeek, P. 1998, Recursive Gaussian derivative filters:  
Proceedings of the International Conference on Pattern Recognition, Brisbane, p. 509–514.

# Anisotropic Diffusion

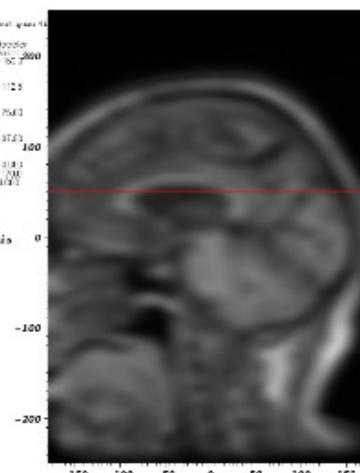
- Isotropic diffusion blurs edges in images

Source  
[vis.lbl.gov](http://vis.lbl.gov)

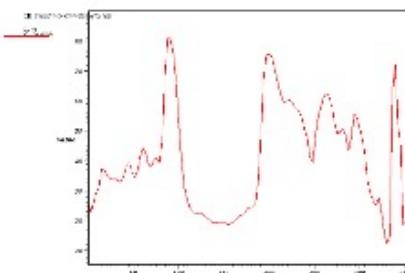
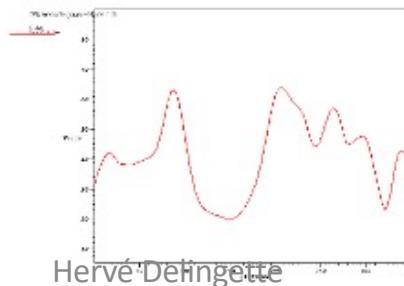
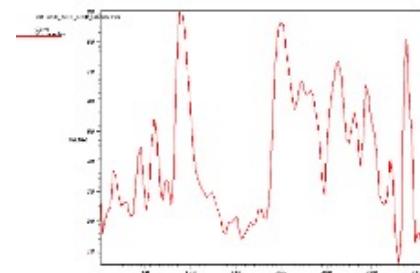
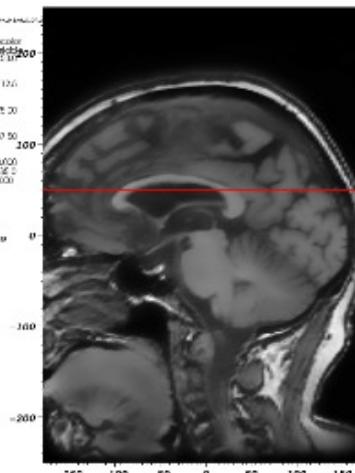
Input Image



Isotropic  
Diffusion



Anisotropic  
Diffusion



Hervé Delingette

# Anisotropic Diffusion

- Isotropic diffusion in 3D

$$\frac{\partial I}{\partial t} = k \left( \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} + \frac{\partial^2 I}{\partial z^2} \right) = k \Delta I = k \operatorname{div} \nabla I$$

- Anisotropic diffusion in 3D

$$\frac{\partial I}{\partial t} = \operatorname{div} c(x, y) \nabla I = \nabla c \cdot \nabla I + c(x, y) \Delta I$$

- $c(x, y)$  is the diffusion coefficient

# Diffusion coefficient

- Choose  $c(x, y, z) = c(\|\nabla I\|)$

- Minimize  $E = \frac{1}{2} \int g(\|\nabla I\|^2) d\Omega$

$$c(x, y, z) = \frac{dg(\|\nabla I\|)}{d\|\nabla I\|}$$

- Choice of the diffusion coefficient :
  - Should be high in homogeneous regions
  - Show be low around edges

- Perona-Malik

$$c(\|\nabla I\|) = e^{-\left(\frac{\|\nabla I\|}{K}\right)^2}$$
$$c(\|\nabla I\|) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2}$$

# Non-Local Means

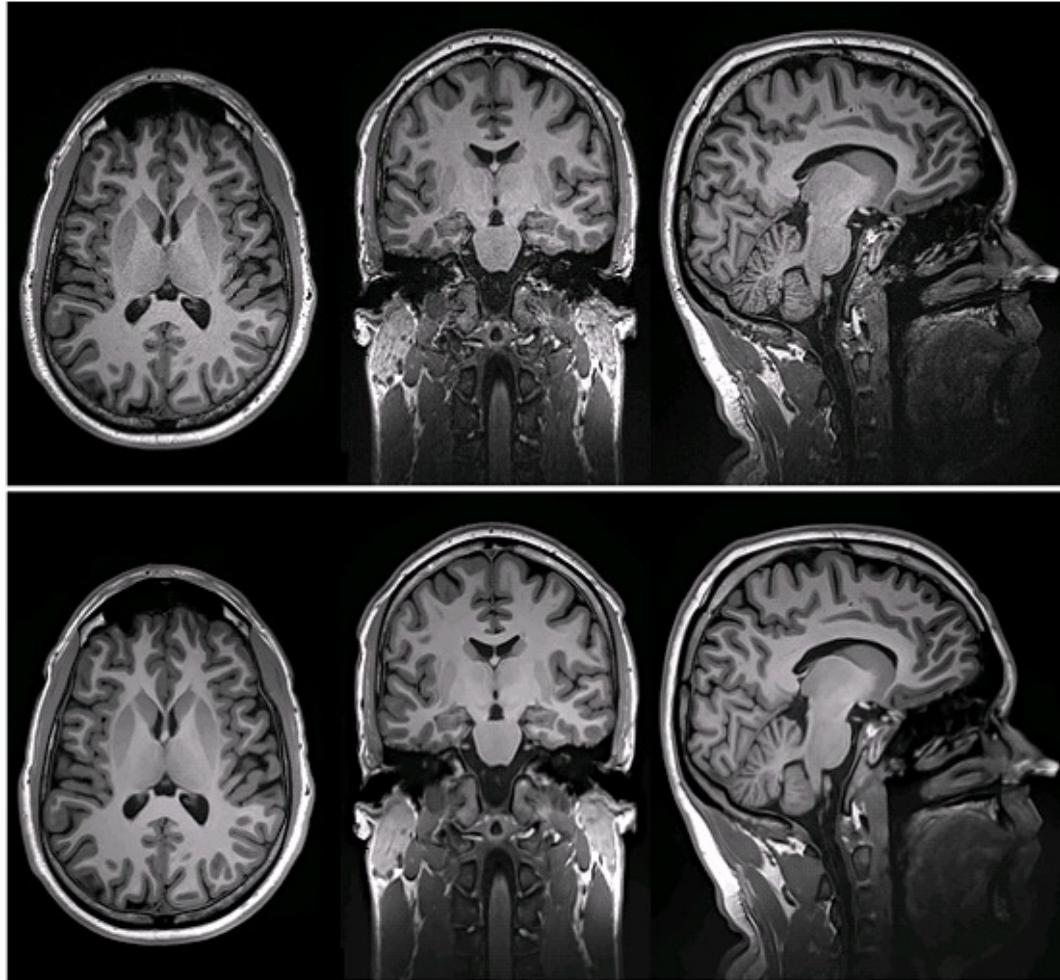
- Denoising by computing the voxel value as the weighted sum of neighboring pixels

$$\tilde{I}(x, y, z) = \frac{1}{c(x, y, z)} \sum_{i=-n}^n \sum_{j=-n}^n \sum_{k=-n}^n w_{ijk}(x, y, z) I(x+i, y+j, z+k)$$
$$c(x, y, z) = \sum_{i=-n}^n \sum_{j=-n}^n \sum_{k=-n}^n w_{ijk}(x, y, z) \quad w_{ijk}(x, y, z) = \exp\left(-\frac{|I(x, y, z) - I(x+i, y+j, z+k)|^2}{h^2}\right)$$

- Can be performed at the block level

Buades, B. Coll., and J. Morel, “A non local algorithm for image denoising,” In Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR), vol. 2, 2005, pp. 60–65.

# Adaptive Non Local Means



Manjón JV, Coupé P, Martí-Bonmatí L, Collins DL, Robles M. Adaptive non-local means denoising of MR images with spatially varying noise levels. *J Magn Reson Imaging*. 2010 Jan;31(1):192-203

# 2.0 Image Filtering

- 2.1 Taxonomy of Filters
- 2.2 Smoothing Filters
- 2.3 **Gradient Filters**
  - Discrete vs Continuous
- 2.4 Image Contour Extraction
- 2.5 Mathematical Morphology

# Computing Image Gradient

## ■ Through Finite Differences

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

$$\Rightarrow \text{First Order (Gradient) 1D Filter } (h = 1) \quad \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

- Gradient 2D Filters along x and y directions

$$\frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Separable filters based on 1D  
gradient and mean filters

$$\frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

# Sobel Filters

- Sobel Filters
  - Separable Filters
  - Reduce noise by convolving with low-pass filter (remove high frequencies) : binomial filters

- First Order along x

$$\frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- First Order along y

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \frac{1}{2} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

# Computing 2<sup>nd</sup> Derivatives

$$f''(x) \approx \frac{\frac{f(x+h_1+h_2) - f(x+h_1-h_2)}{2h_2} - \frac{f(x-h_1+h_2) - f(x-h_1-h_2)}{2h_2}}{2h_1}$$

⇒ Second order filters for  $h_1 = h_2 = 1/2$        $[1 \quad -2 \quad 1]$

- Laplacian Filters

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

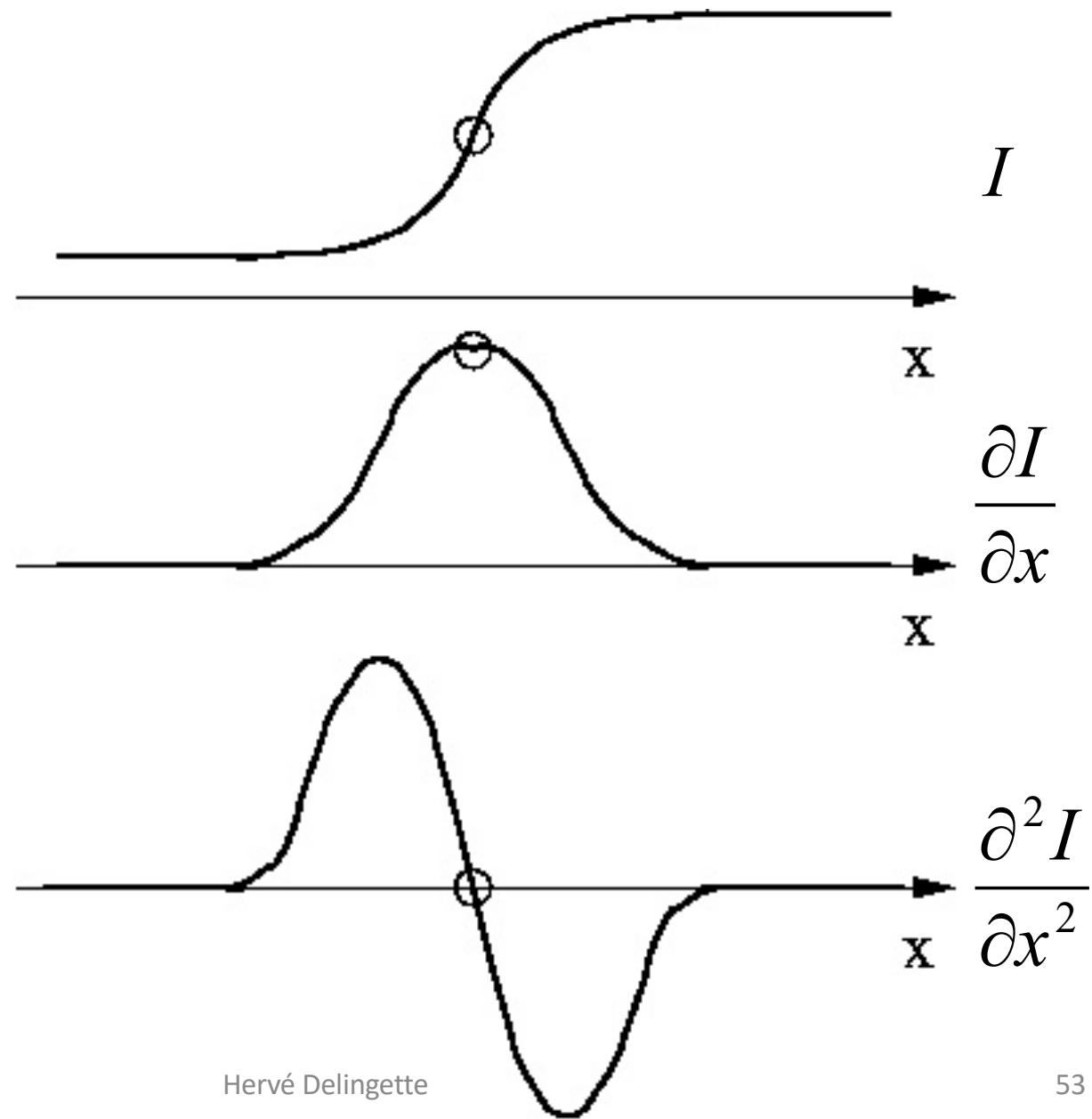
# 2.0 Image Filtering

- 2.1 Taxonomy of Filters
- 2.2 Smoothing Filters
- 2.3 Gradient Filters
- 2.4 **Image Contour Extraction**
  - Definition of visible contours
  - Implementation of contour detection
  - Optimal contour Detection
- 2.5 Mathematical Morphology

# Contour Detection

- Qualitatively :

Find points where  
 $I(x,y)$  varies very  
quickly with  $(x,y)$ .



# Contour Detection

- 2 detection criteria
  - Find Maxima of gradient in the gradient direction

$$\|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

maximum in the direction  $\overrightarrow{\nabla I}$

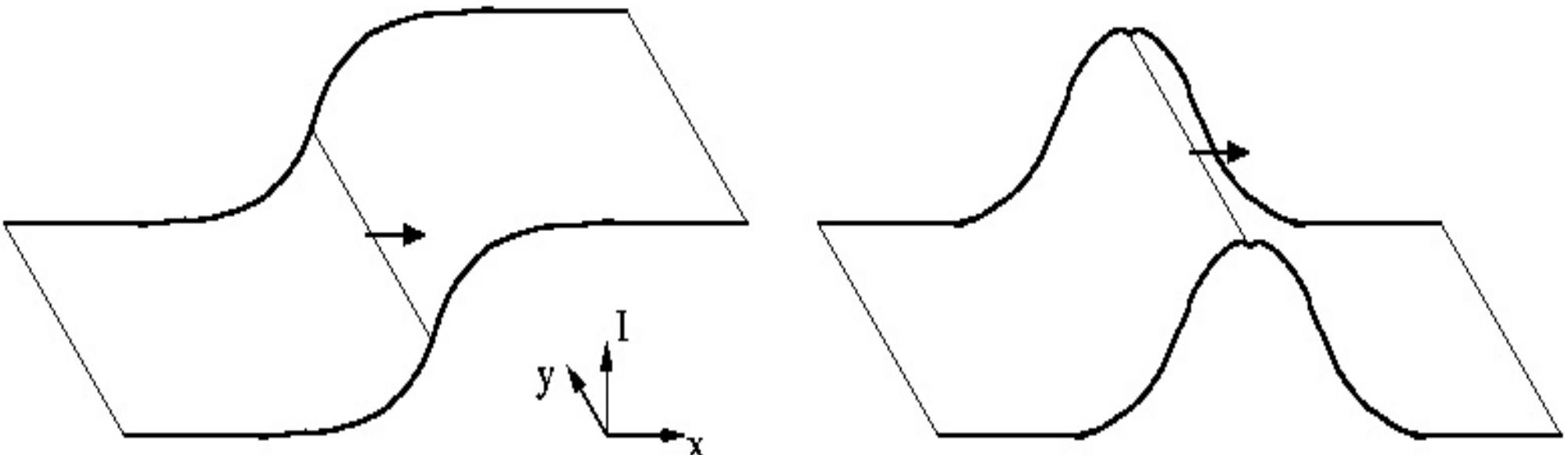
- Zero crossing of Laplacian

$$\Delta I = \nabla^2 I = I_{xx} + I_{yy} = 0$$

# Gradient Maxima

- Find Maxima of gradient in the gradient direction

$$\frac{\partial \|\nabla I\|}{\partial \vec{n}} = 0 \text{ with } \vec{n} = \frac{\vec{\nabla} I}{\|\vec{\nabla} I\|}$$



# Gradient Maxima

- Find Maxima of gradient in the gradient direction

$$\frac{\partial \|\nabla I\|}{\partial \vec{v}} = \vec{v} \cdot \vec{\nabla} \|\nabla I\|$$

$$\vec{\nabla} \|\nabla I\| = \begin{pmatrix} \frac{I_x I_{xx} + I_y I_{yx}}{\|\nabla I\|} \\ \frac{I_x I_{xy} + I_y I_{yy}}{\|\nabla I\|} \end{pmatrix} = \frac{H(i) \vec{\nabla} I}{\|\nabla I\|}$$

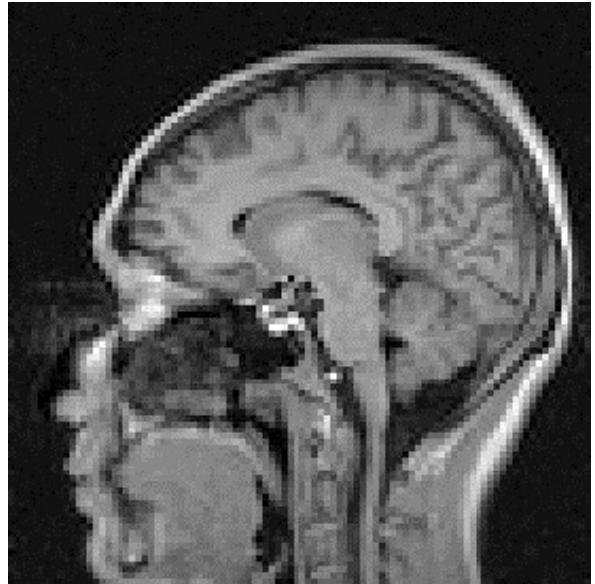
# Contour Detection

- Find Maxima of gradient in the gradient direction

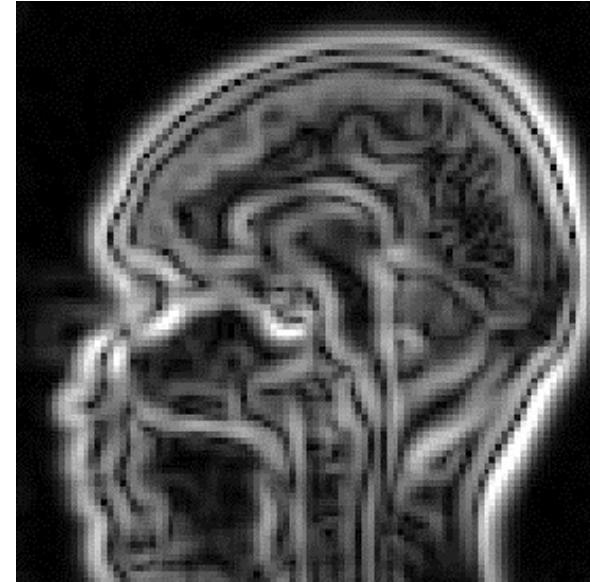
$$\frac{\partial \|\nabla I\|}{\partial \vec{n}} = \frac{\vec{\nabla}I \cdot H(I) \vec{\nabla}I}{\|\nabla I\|^2}$$

$$\frac{\partial \|\nabla I\|}{\partial \vec{n}} = 0 \leftrightarrow \vec{\nabla}I \cdot H(I) \vec{\nabla}I = 0$$

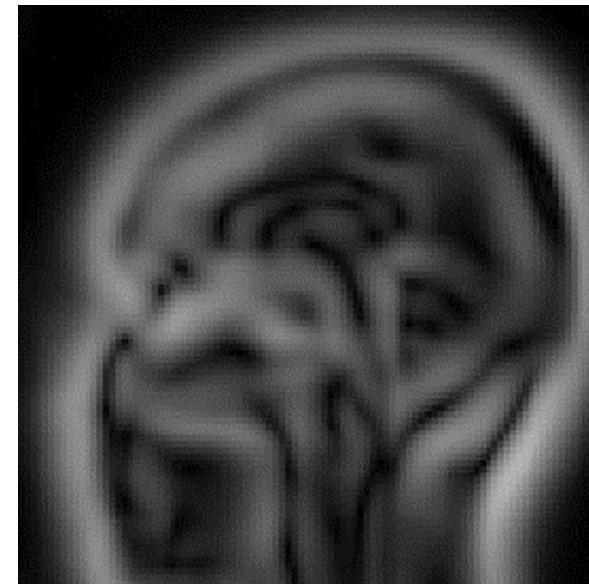
- Laplacian  $\Delta I = 0 \Leftrightarrow \text{trace}(H(I)) = 0$



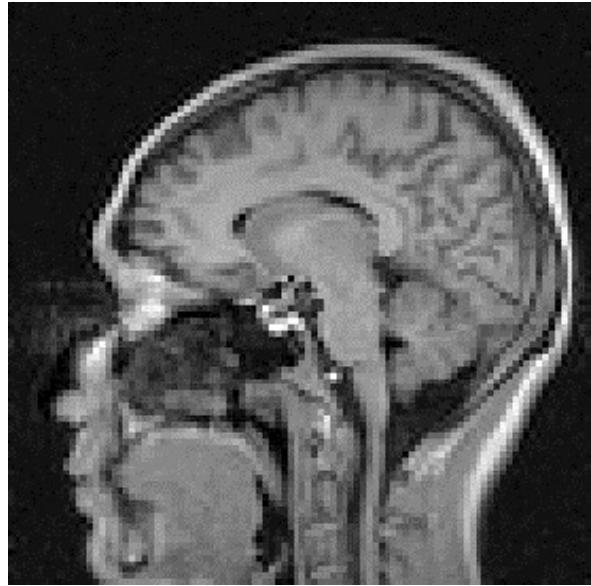
Original Image



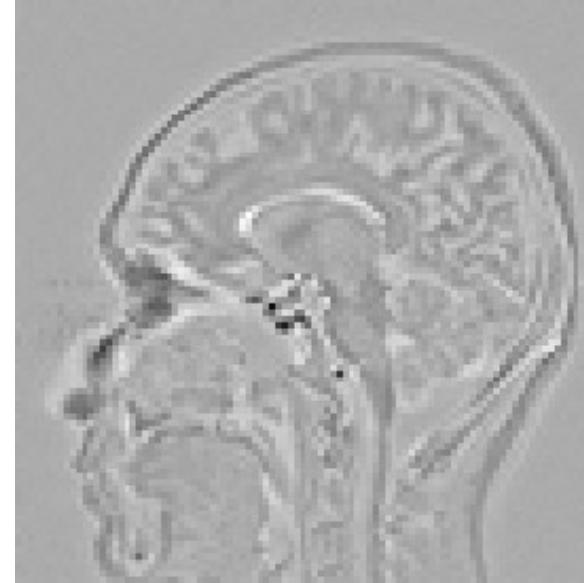
Norm of Gradient  $\sigma = 3$



Hervé Delingette Norm of Gradient  $\sigma = 9$  58



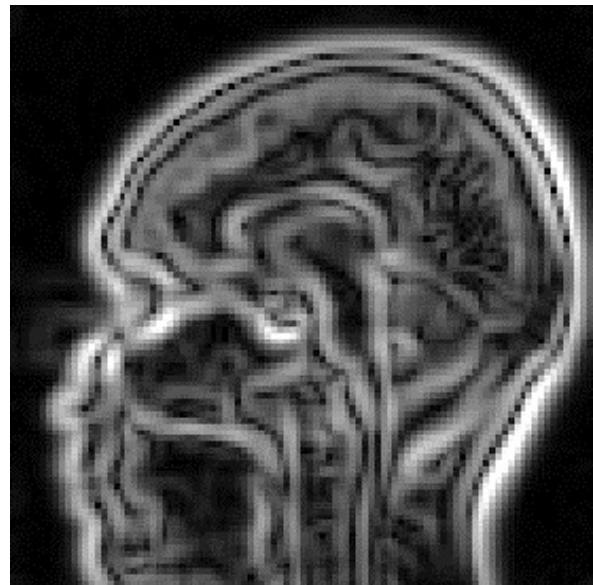
Original Image



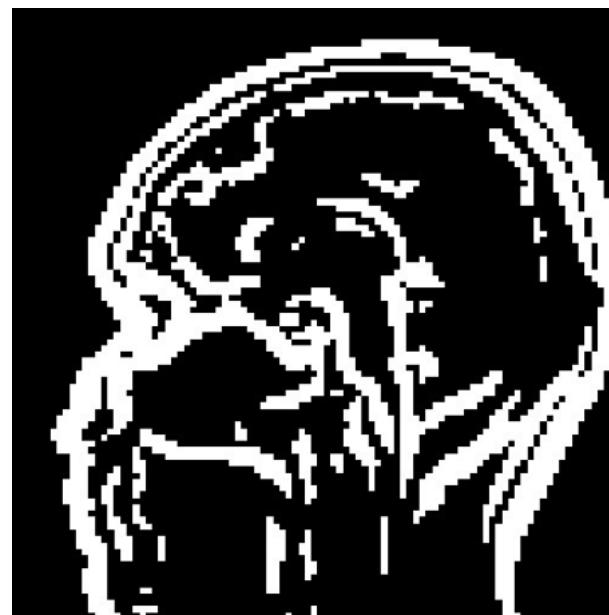
Laplacian  $\sigma = 3$



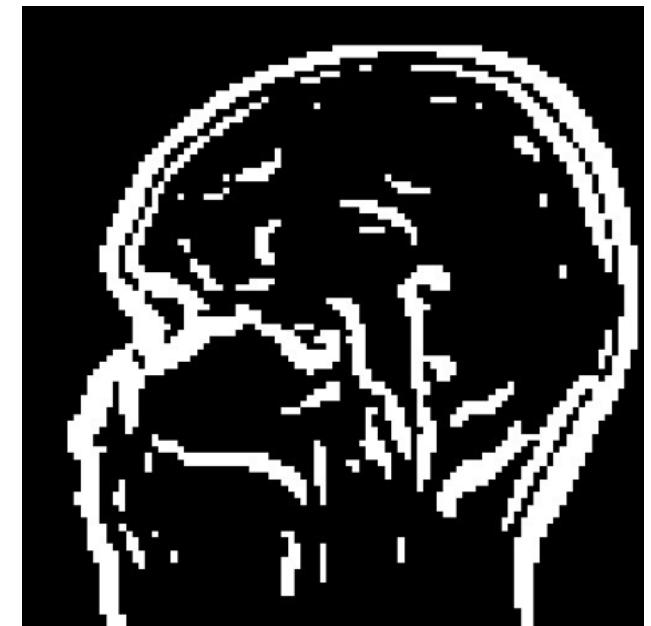
Norm of Gradient  $\sigma = 3$



Threshold= 51



Threshold= 60



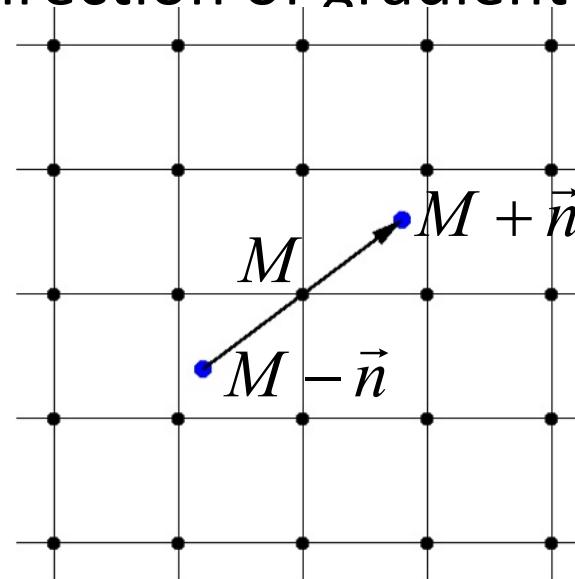
# Gradient Extrema

1. Compute Image Gradient
2. Extract extrema gradient in the direction of gradient

$$M_1 = M + d \frac{\vec{\nabla} I}{\|\nabla I\|}$$

$$M_2 = M - d \frac{\vec{\nabla} I}{\|\nabla I\|}$$

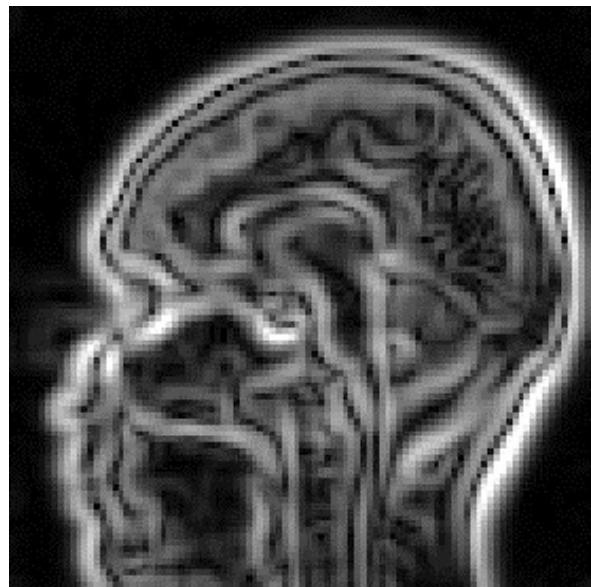
Extremum Criterion



3. Threshold Gradient Extrema

$$\|\nabla I(M)\| > \|\nabla I(M_2)\| \quad \text{et} \quad \|\nabla I(M)\| \geq \|\nabla I(M_1)\|$$

Norm of Gradient  $\sigma = 3$



Threshold= 51



Extrema of Gradient  
in Gradient direction



Heiko Delingette

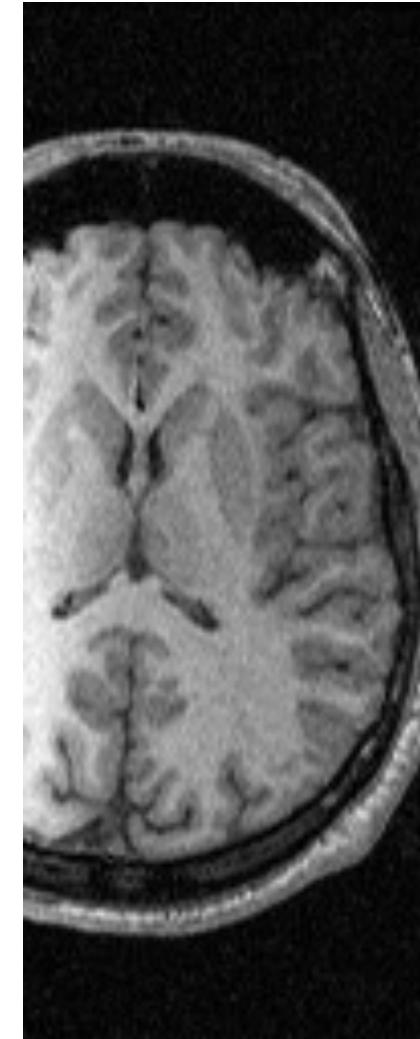
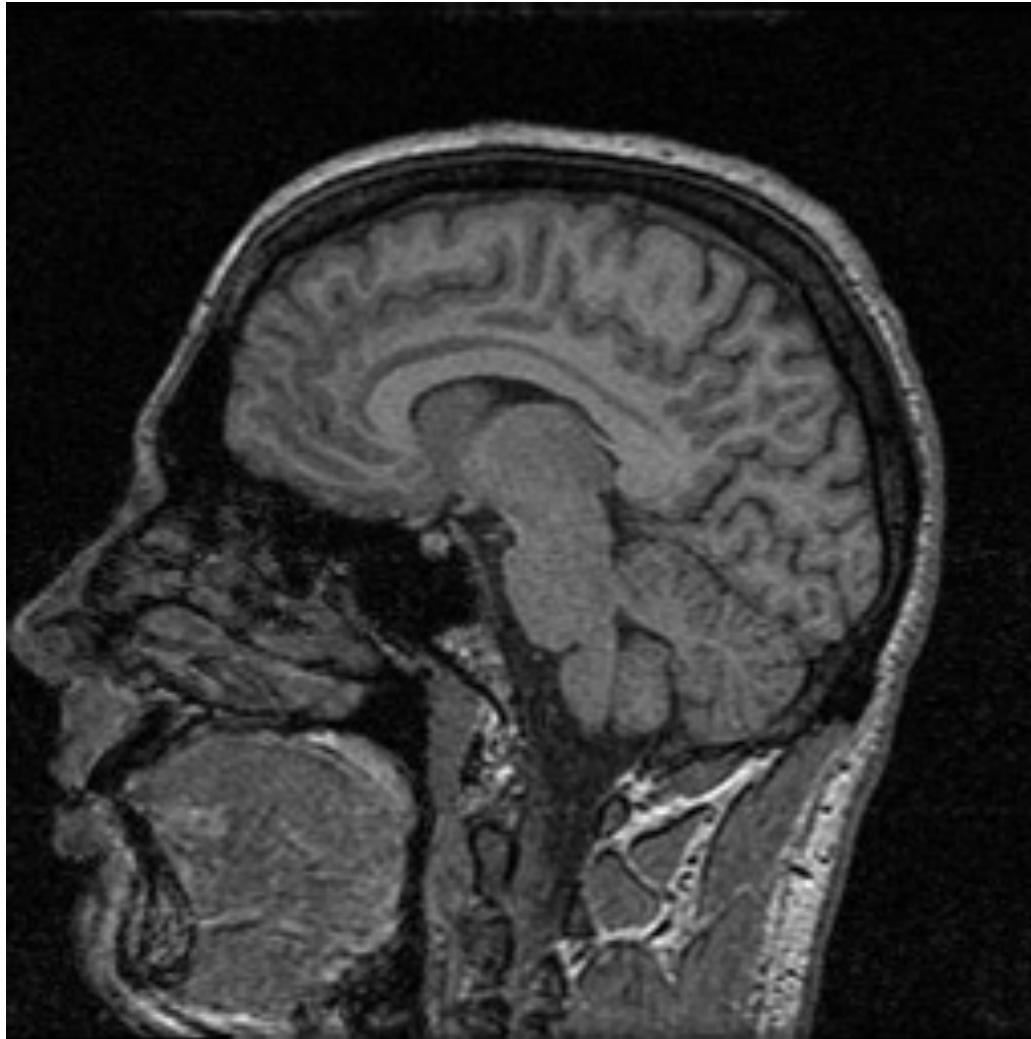
# Thresholding

- Simple Thresholding       $\|\vec{\nabla I}(M)\| \geq s$
- Hysteresis Thresholding
  - Select connected components for which
    - All points verify  $\|\vec{\nabla I}(N)\| \geq s_{\text{bas}}$
    - At least one point P verify  $\|\vec{\nabla I}(P)\| \geq s_{\text{haut}}$
    - (optional) Size is greater than t

# Optimal Contour Detection

- Contours from previous approach are difficult to evaluate
- Define contours as maxima of the convolution of  $I(x)$  with filter  $f$
- Canny defined an ideal filter that should optimize the following 3 criteria on an ideal contour
  - **detection** : filter should be able to provide high values next to the contour
  - **localisation** : contour should be localised with accuracy
  - **unique response** : contour should provide a unique maximum around contour

# Contour Detection 2D / 3D



# Contour Detection 2D



# Contour Detection 3D

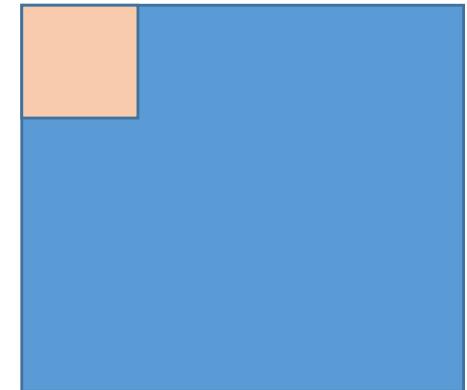


# 2.1 Mathematical Morphology

- Overview :
- Definitions of Neighborhood
- Transformations on Binary Images
  - Boolean
  - Hit or Miss
  - Erosion
  - Dilation
- Extension to Grey-level Images
- Distance maps

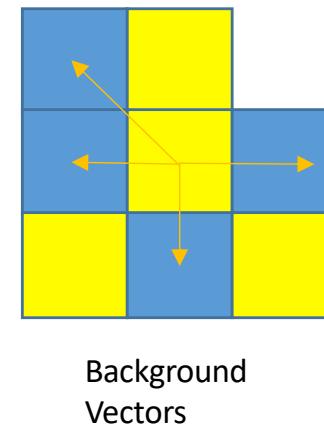
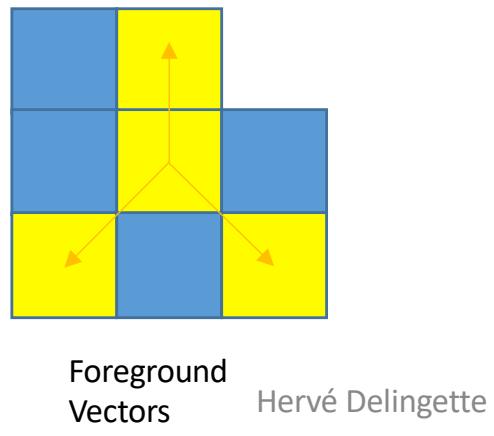
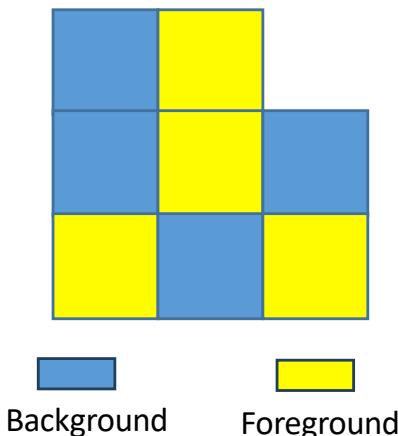
# Mathematical Morphology Principles

- Similar principle as convolution (sliding window)
  - Apply on a binary or grey level image
  - Based on a structuring element (a small template)
- Iteratively center template on a pixel and compute resulting value
- but with non-linear operations:



# Definition of Structuring Element

- We consider binary images : pixels are either 0 or 1
- Structuring element = specific pattern to be detected.
- Pattern can include both foreground or background pixels
- SE can be defined with vectors since sliding pattern



# Notations (Binary)

$E_a$  Affine space

$E_v$  Associated vector space

$I$  : Binary Image :  $K \subset E_a \rightarrow \{0,1\}$

$X$  : **Foreground Object**

$$X = \{x \in K / I(x)=1\}$$

$X^c$  : **Background Object** complementary of  $X$  with respect to  $E_a$ ,

$B$  : structuring element

=  $\{ b \in E_v \}$  set of vectors

=  $\{ x \in E_a / Ox = b \}$  set of points

# Boolean Operations

- Union :  $X \cup Y = \{ x / x \in X \text{ or } x \in Y \}$
- Intersection :  $X \cap Y = \{ x / x \in X \text{ and } x \in Y \}$
- Inclusion :  $X \subset Y \Leftrightarrow \forall x \in X \Rightarrow x \in Y$
- Non symmetric difference  
$$X \setminus Y = \{ x / x \in X \text{ and } x \notin Y \}$$
- Symmetric Difference  
$$X \setminus\setminus Y = \{ x / x \in ((X \cup Y) \setminus (X \cap Y)) \}$$
- Complement

$$(X^c)_E = \{ x / x \in E \text{ and } x \notin X \}$$

# Translated & Symmetric

- Translated of  $X$  by vector  $b$

$$X_b = \{x + \vec{b} ; x \in X\}$$

- $B$  centered in  $x$  (a.k.a translated of  $B$  by  $x$ )

$$B_x = \{x + \vec{b} ; \vec{b} \in B\}$$

- Symmetric of  $B$

$$\breve{B} = \{-\vec{b}; \vec{b} \in B\}$$

# Hit or miss transformation

- Partition of a structuring element  $B$

$$B = B^1 \cup B^2 \text{ et } B^1 \cap B^2 = \emptyset$$

- Hit or miss transformation of set  $X$

$$X \otimes B = \{ x / B^1_x \subset X \text{ et } B^2_x \subset X^c \}$$

- $x \in X \otimes B$

$\Leftrightarrow B^1_x$  is in foreground and  $B^2_x$  is in background

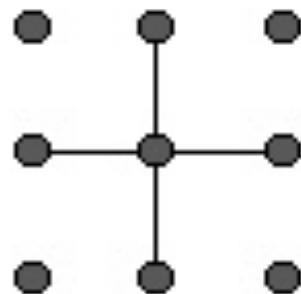
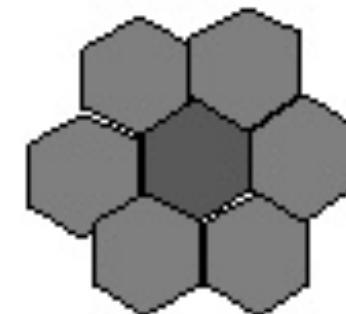
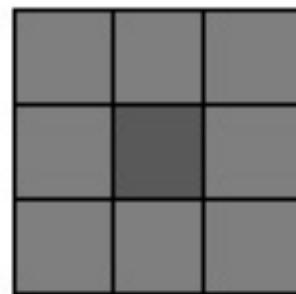
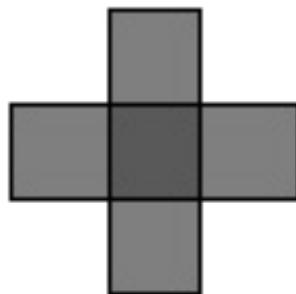
$\Leftrightarrow B$  describe the local configuration of the background and foreground objects

# 2.1 Mathematical Morphology

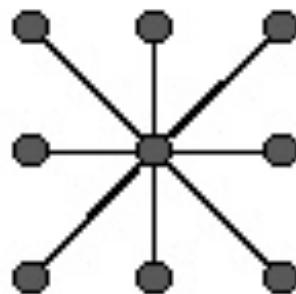
- Overview :
- • Definitions of Neighborhood
  - Transformations on Binary Images
    - Boolean
    - Hit or Miss
    - Erosion
    - Dilation
  - Extension to Grey-level Images
  - Distance maps

# Neighborhood

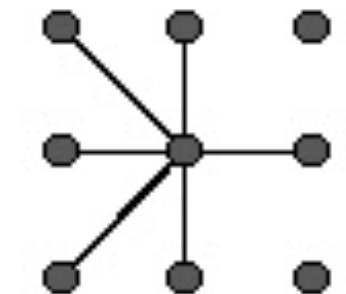
- Different types of neighborhood in 2D



4 -neighborhood



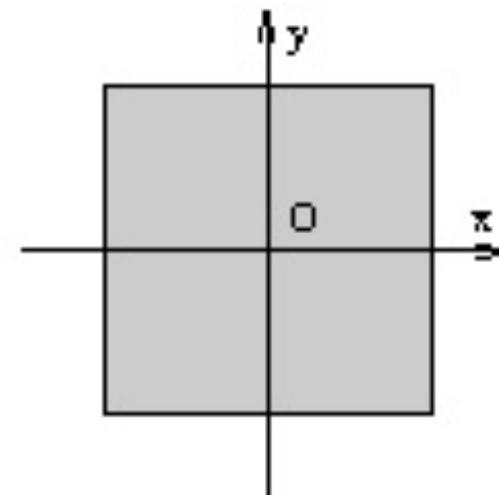
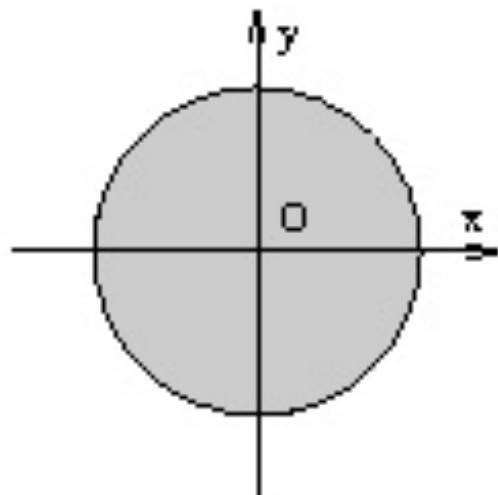
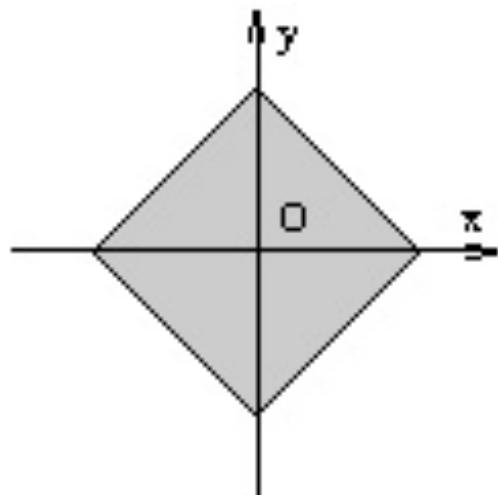
8 -neighborhood



6 -neighborhood

# Neighborhood

- Some generalizes to higher dimensions
- Corresponds to a choice of metric norm



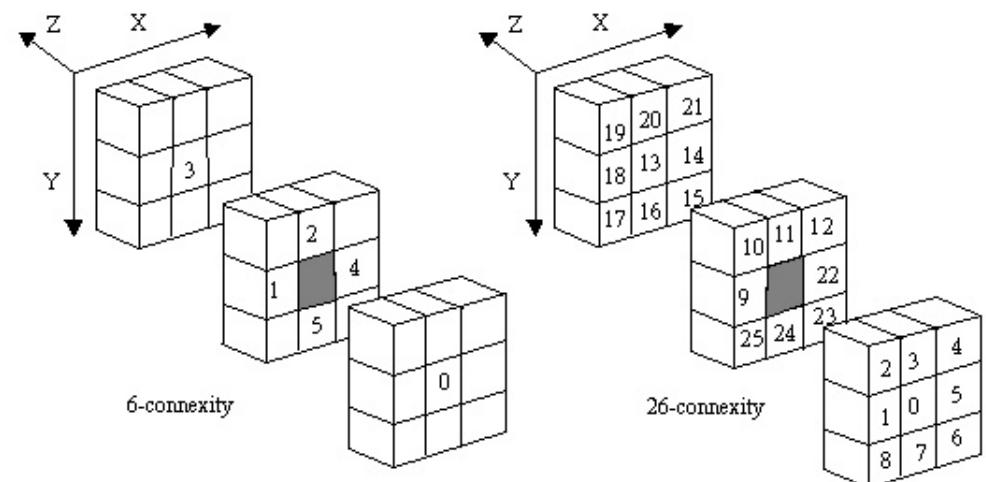
$$D_1(x, y) = \sum_{i=1}^n |y_i - x_i|$$

$$D_2(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

$$D_\infty(x, y) = \max_{i=1 \dots n} |y_i - x_i|$$

# Neighborhood

- 3 types of neighborhood for a 3D image :
  - 6-neighborhood : adjacency through faces
  - 18-neighborhood : adjacency through faces and edges
  - 26-neighborhood : adjacency through faces and edges and vertices

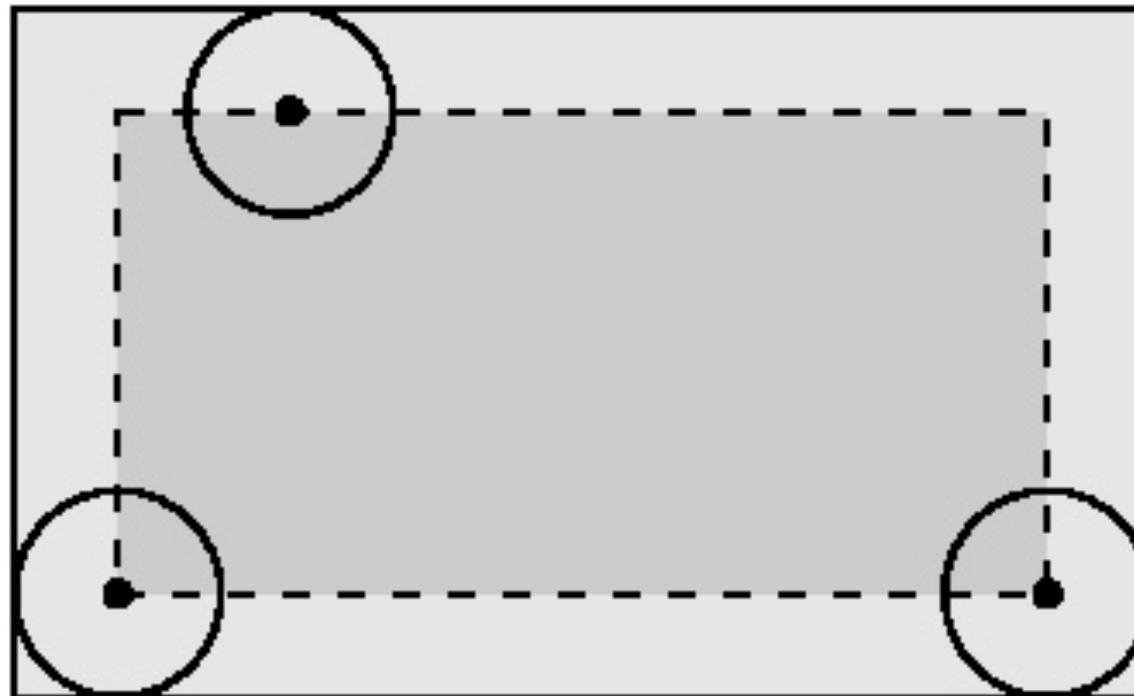


# Mathematical Morphology

- Technique to manipulate digital shapes
  - Unknown image
  - Known shape : **structuring element**
- Boolean operation between image & structuring element
$$\cup, \cap, \subset, \in, \neq \emptyset, \dots$$
- See work by Serra, Schmitt, ...

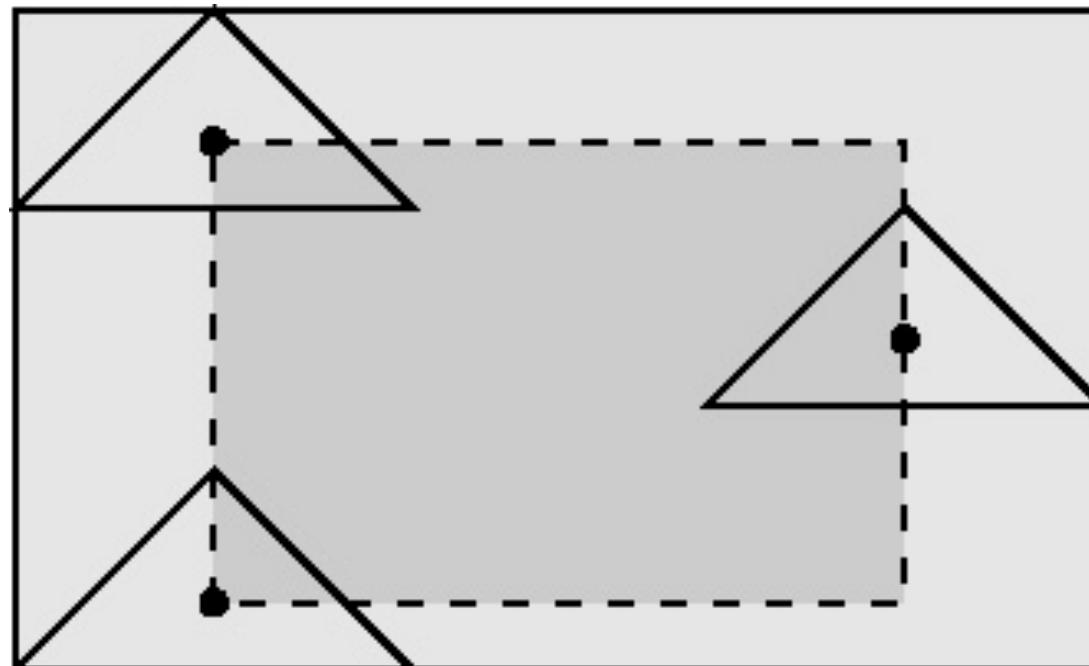
# Erosion

- Specific case of HoM transformation when  $B^2 = \emptyset$   
Erosion of  $X$  by  $B$  =  $\text{Ero}(X, B) = \{ y \in E_a / B_y \subset X \}$



# Erosion

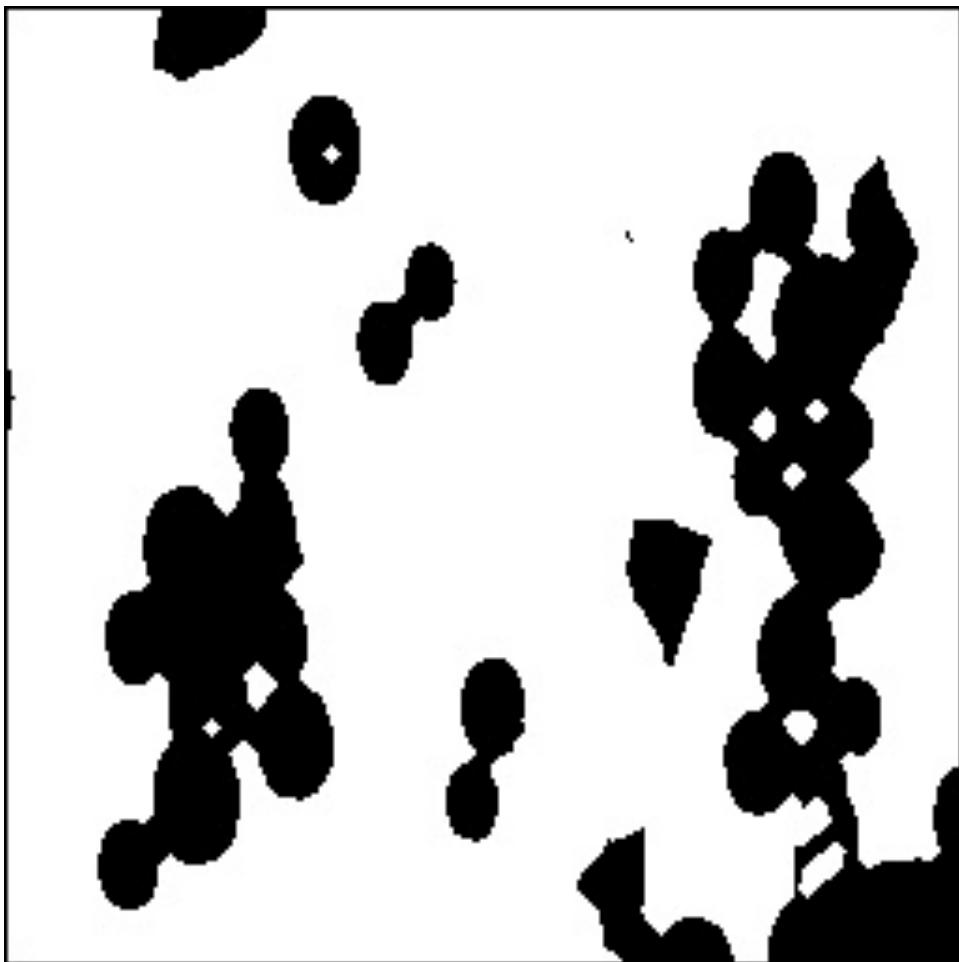
- Specific case of HoM transformation when  $B^2 = \emptyset$   
Erosion of  $X$  by  $B$  =  $\text{Ero}(X, B) = \{ y \in E_a / B_y \subset X \}$



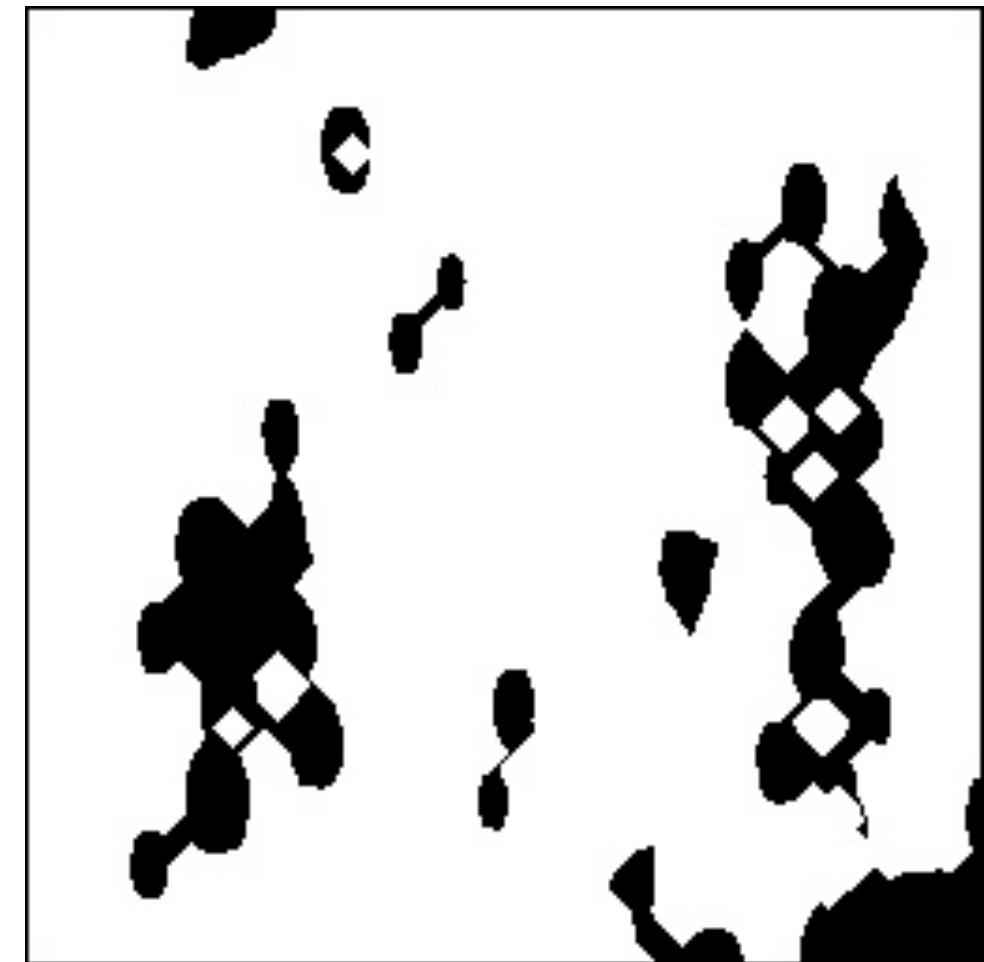
# Definitions (Binary)



# Erosion



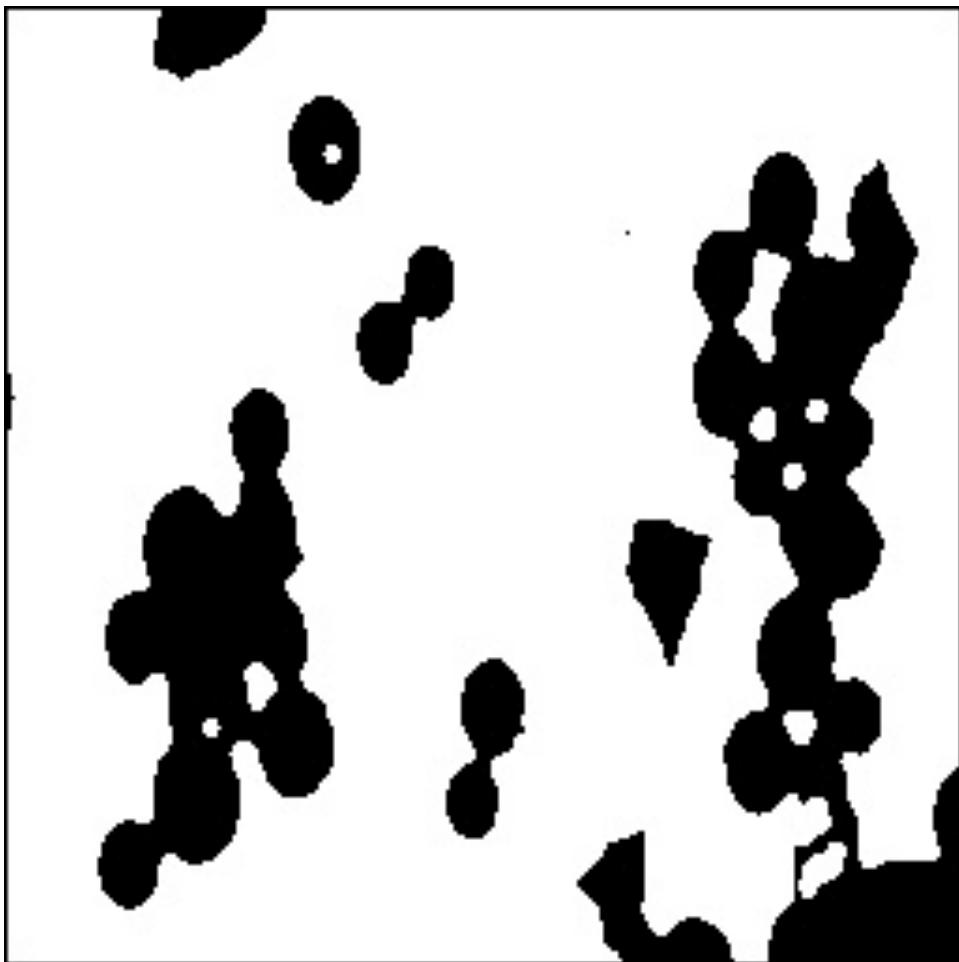
iterations = 2



4-structuring element

iterations = 5

# Erosion



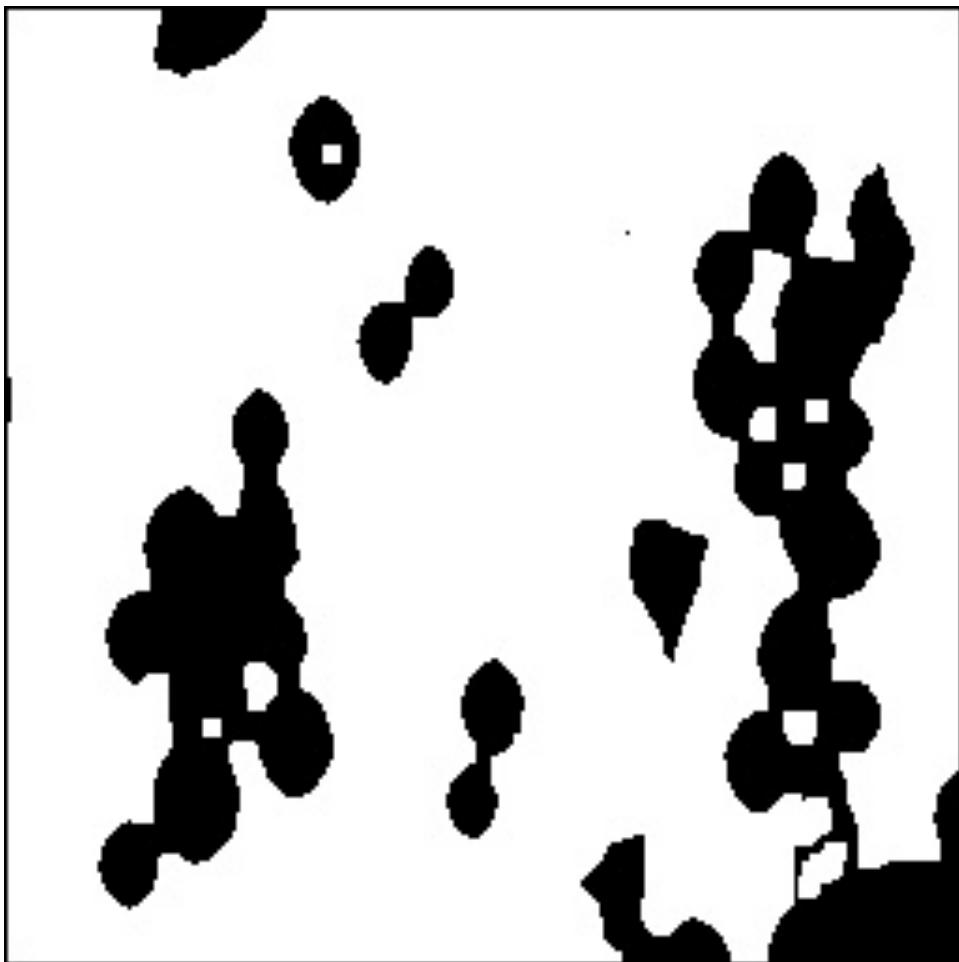
itérations = 2



6-structuring element

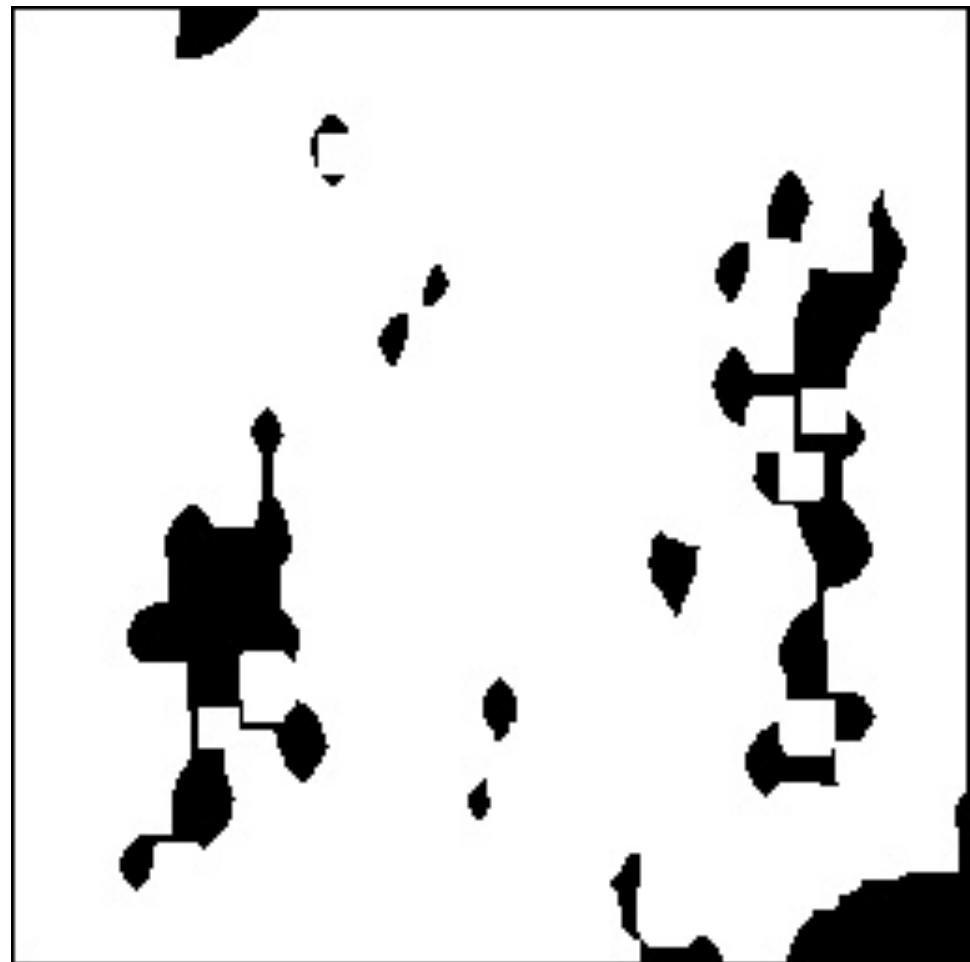
itérations = 5

# Erosion



itérations = 2

8-structuring element

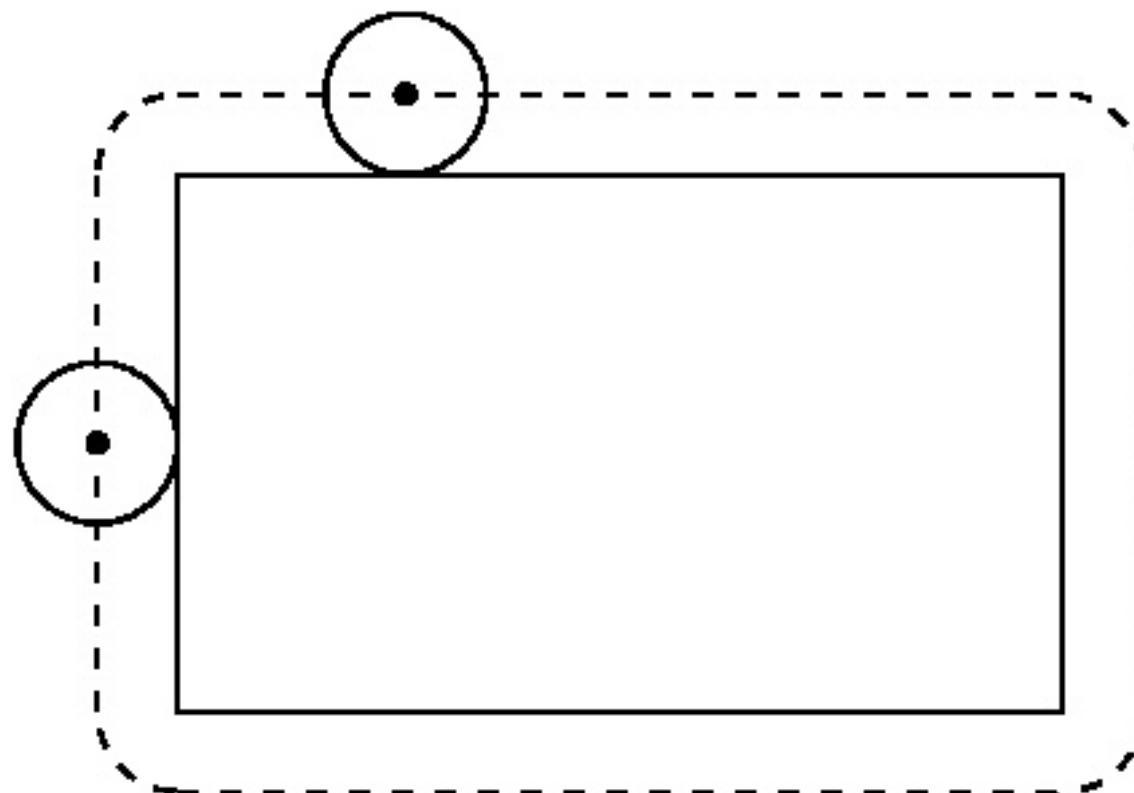


itérations = 5

# Dilation

- Dilation of  $X$  by  $B$

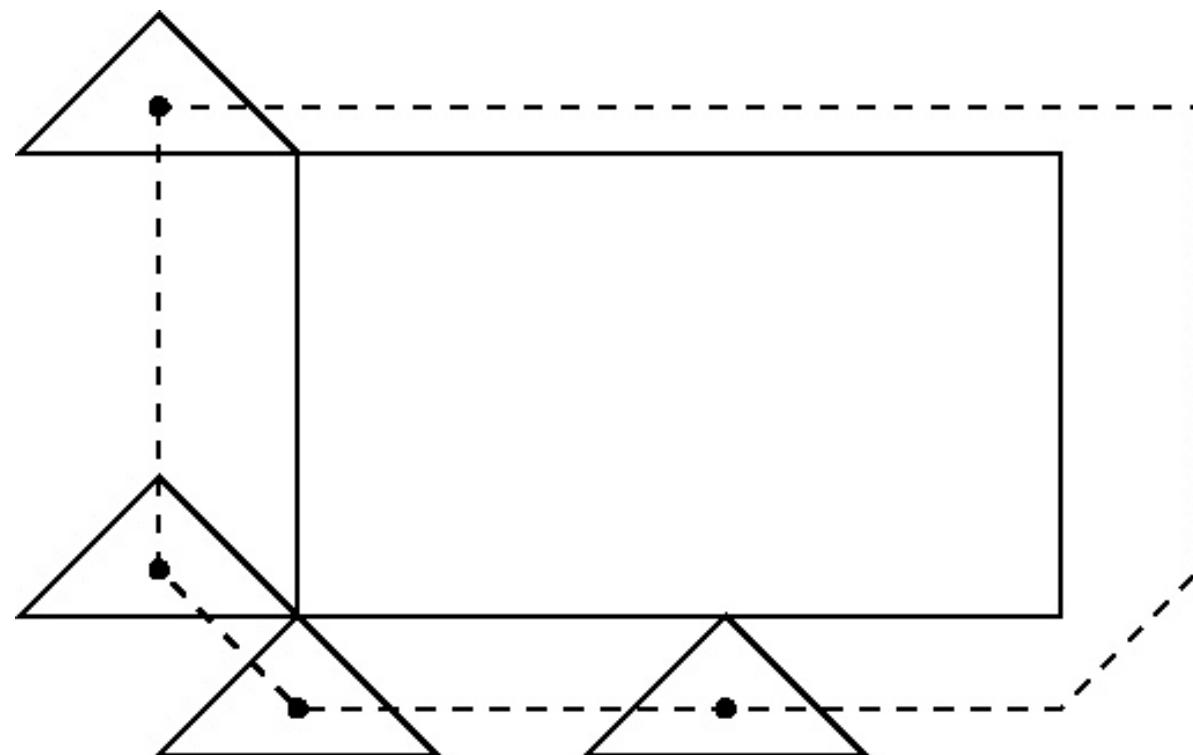
$$\text{Dil}(X, B) = \{ y \in E_a / B_y \cap X \neq \emptyset \}$$



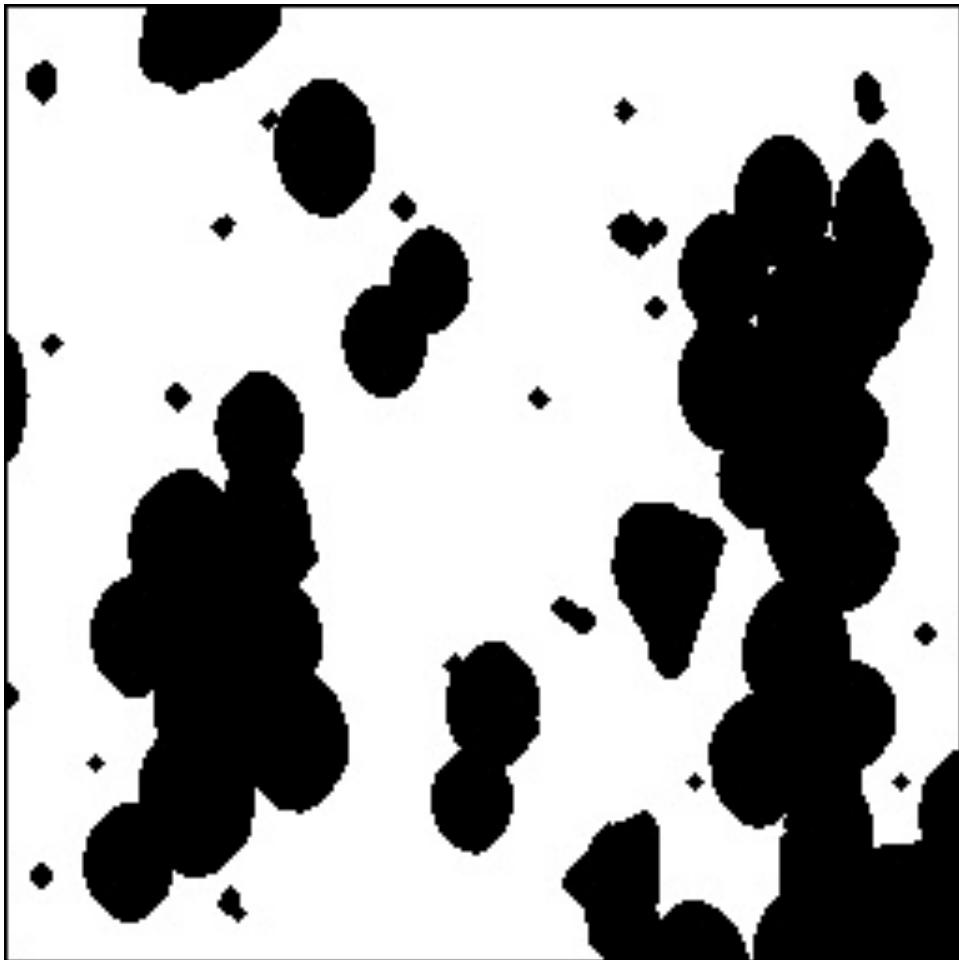
# Dilation

- Dilation of  $X$  by  $B$

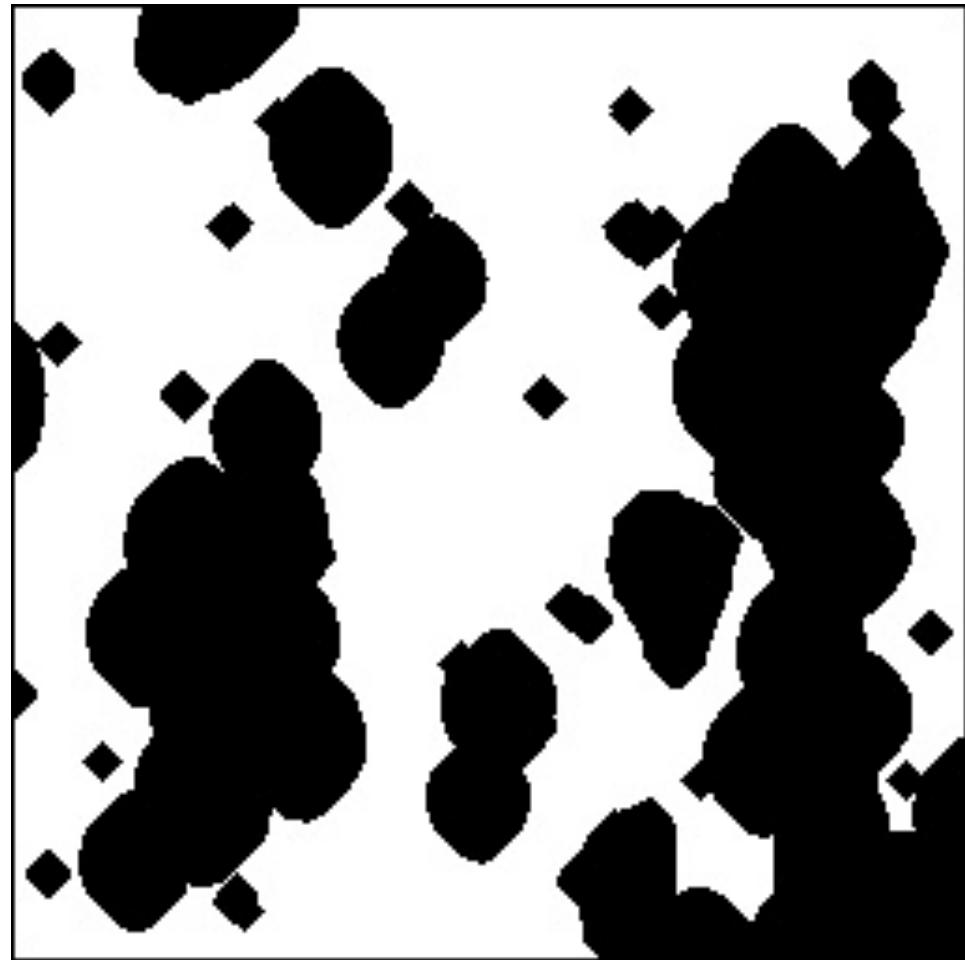
$$\text{Dil}(X, B) = \{ y \in E_a / B_y \cap X \neq \emptyset \}$$



# Dilation



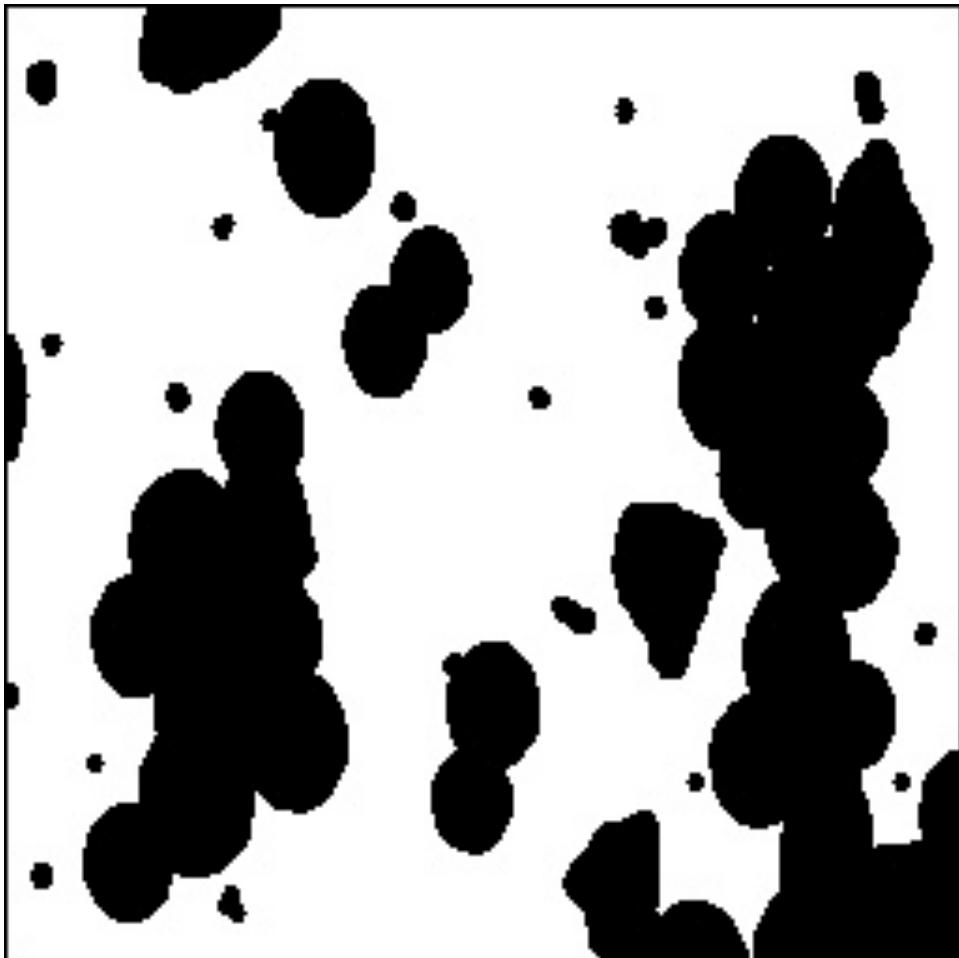
iterations = 2



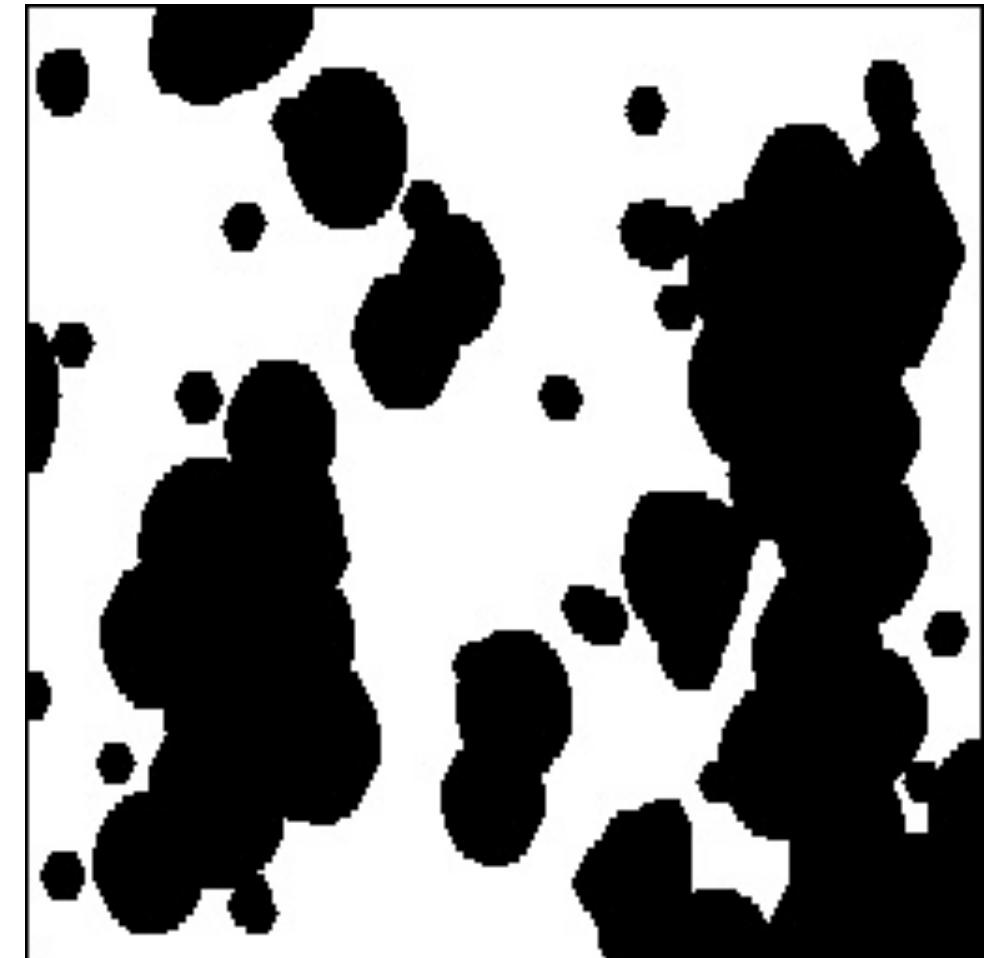
4-structuring element

iterations = 5

# Dilation



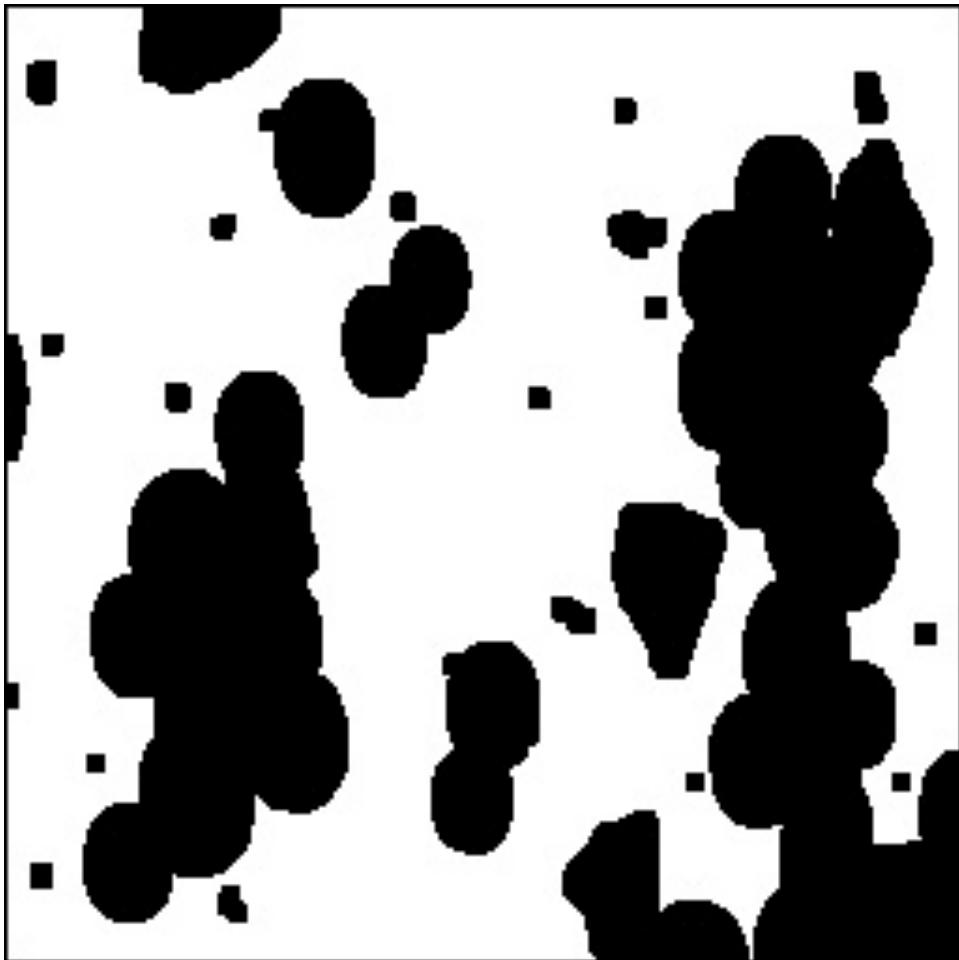
iterations = 2



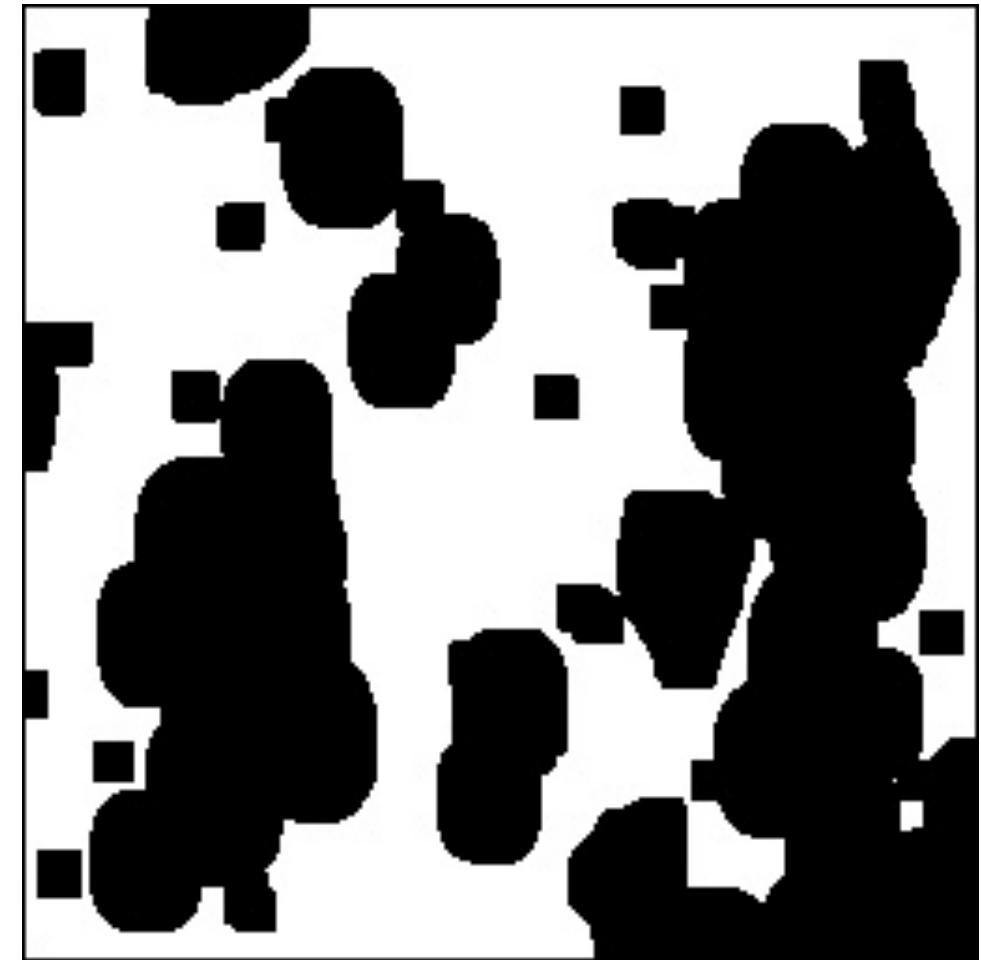
6-structuring element

iterations = 5

# Dilation



iterations = 2

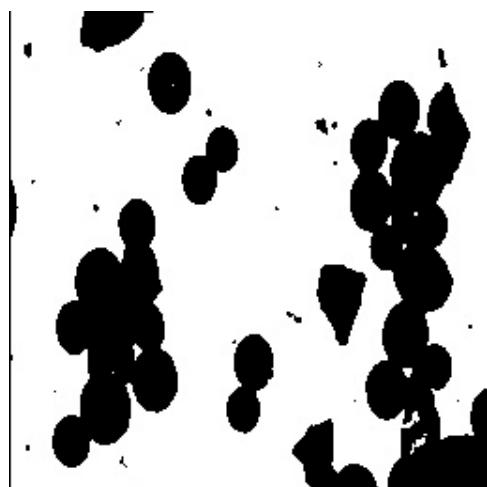


8-element structurant

iterations = 5

# Duality Erosion / Dilation

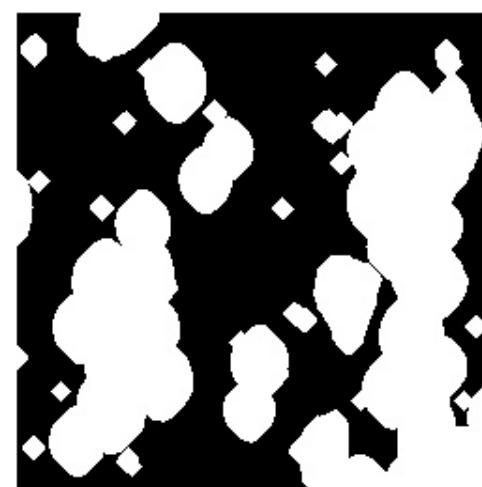
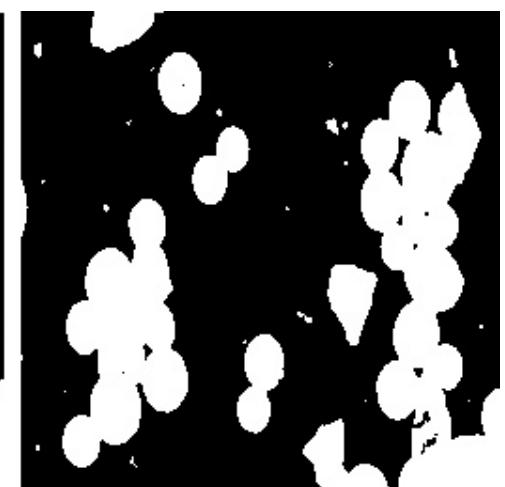
$$(X \oplus \breve{B})^c = X^c \ominus \breve{B}$$

 $X$ 

Original image

 $X \oplus \breve{B}$ 

Dilated Image

 $X^c \ominus \breve{B}$ Erosion of  
The complement of X $X^c$ The complement of  
The dilated image

# 2.1 Mathematical Morphology

- Overview :
- • Definitions of Neighborhood
  - Transformations on Binary Images
    - Boolean
    - Hit or Miss
    - Erosion
    - Dilation
  - Extension to Grey-level Images
  - Distance maps

# Extension to grey-level images

- Let  $f$  be a grey level image defined on  $K$  with values  $\mathbb{R}$ .

- Let  $g$  be the structuring element.

- One writes  $\check{g}(x) = g(-x)$

- Dilation :

$$(f \oplus \check{g})(x) = \max_{y \in B_x} \{f(y) + g(y - x)\}$$

- Erosion :

$$(f \ominus \check{g})(x) = \min_{y \in B_x} \{f(y) - g(y - x)\}$$

# Extension to grey-level images

- Correspondance with binary mathematical morphology  $B$

$$B(x) = \begin{cases} 0 & \text{si } x \in B \\ -\infty & \text{si } x \in B^c \end{cases}$$

- In this case, erosion and dilation write as :

$$(f \oplus \check{B})(x) = \max_{y \in B_x} \{f(x + y)\}$$

$$(f \ominus \check{B})(x) = \min_{y \in B_x} \{f(x + y)\}$$

- Dilation and Erosion are still dual of each other.

# Extension to grey-level images

dilation



4-structuring  
element  
(2 iterations)



8-structuring  
element  
(2 iterations)



erosion

# Morphological Gradient

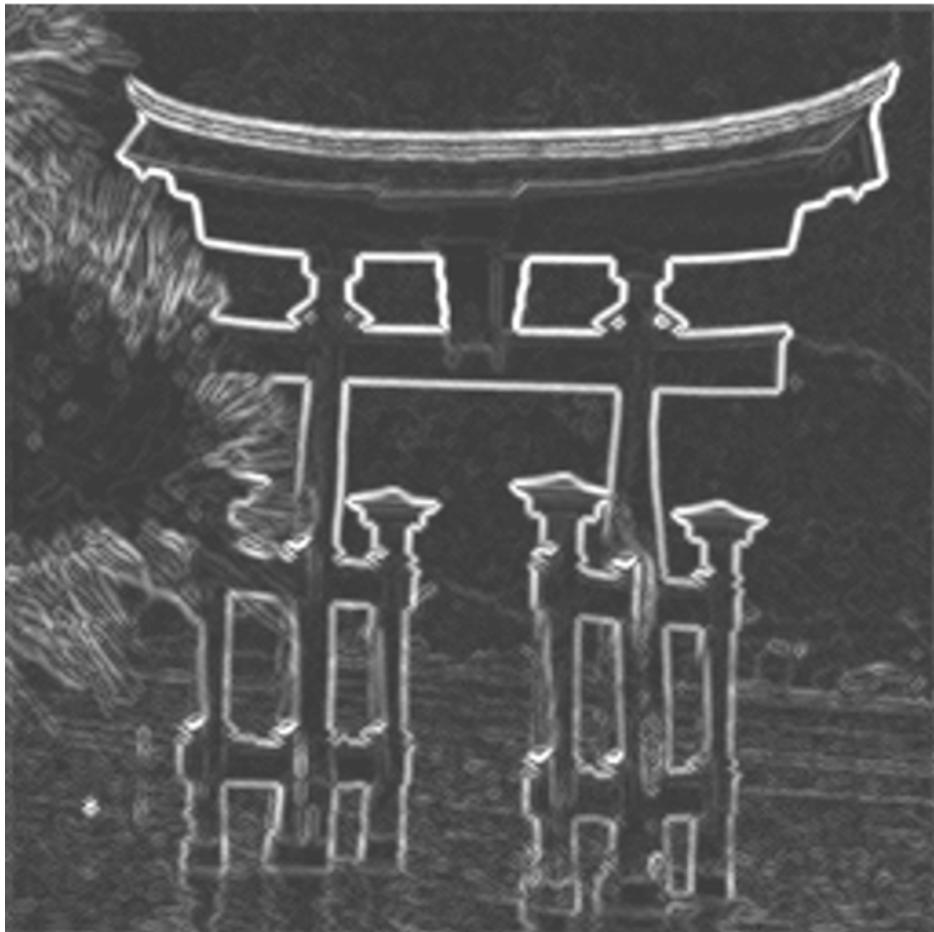
- Function gradient (when it exists) may be formalized as :

$$\lim_{r \rightarrow 0} \frac{f \oplus \breve{B}(r) - f \ominus \breve{B}(r)}{2r}$$

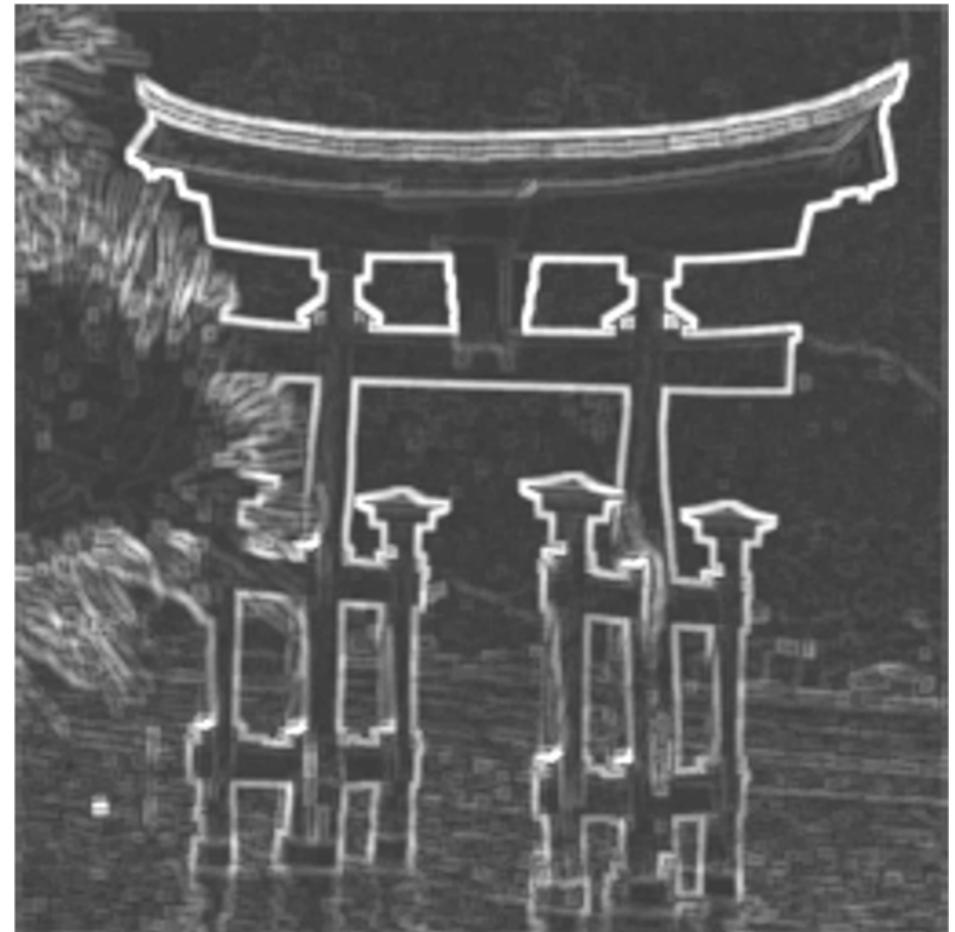
Where  $B(r)$  is a bowl of radius  $r$

- Morphological Gradient       $B = \breve{B}$        $f \oplus \breve{B} - f \ominus \breve{B}$
- Superior Gradient                   $f \oplus \breve{B} - f$
- Inferior Gradient                   $f - f \ominus \breve{B}$

# Morphological Gradient



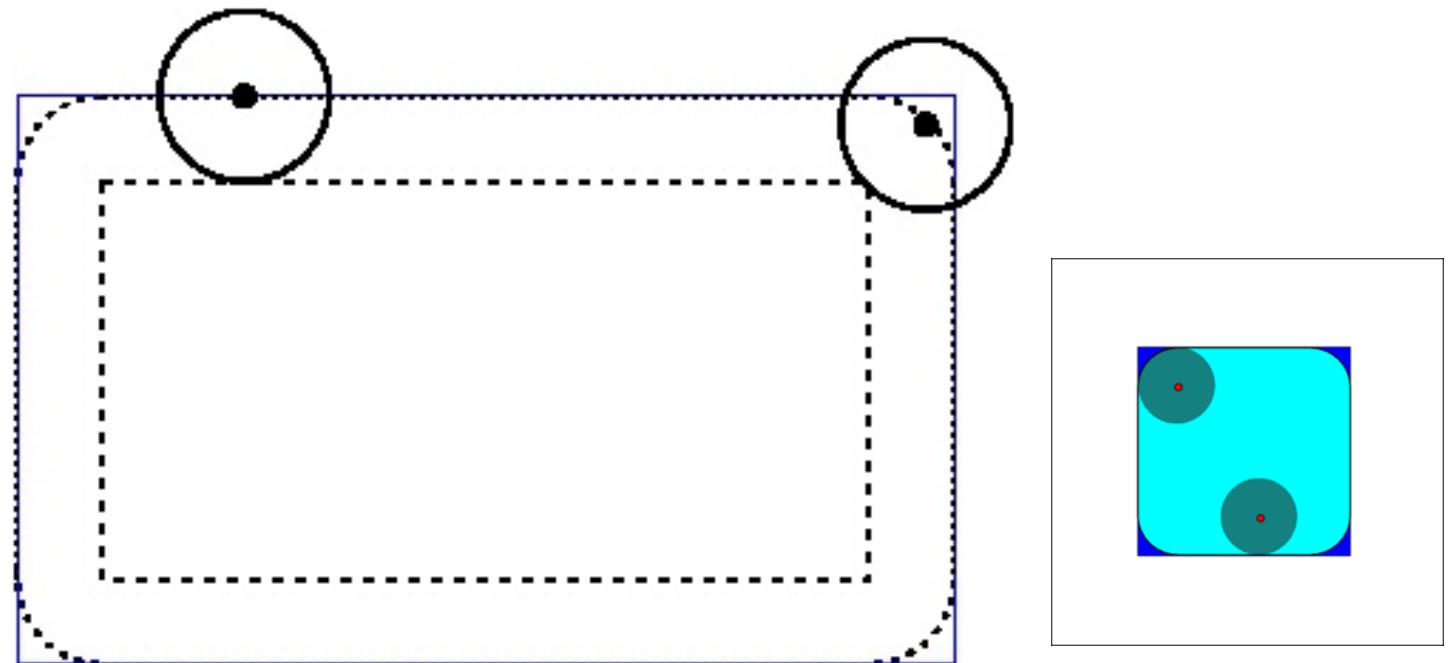
4-structuring element



8-structuring element

# Opening

- $X$  = Foreground Object
- $B$  = Structuring Element
- Opening defined as  $X \circ B = (X \ominus B) \oplus B$



Opening of a set is the dilated of the eroded set (if  $B$  is symmetric)



4 iterations



4-Neighborhood

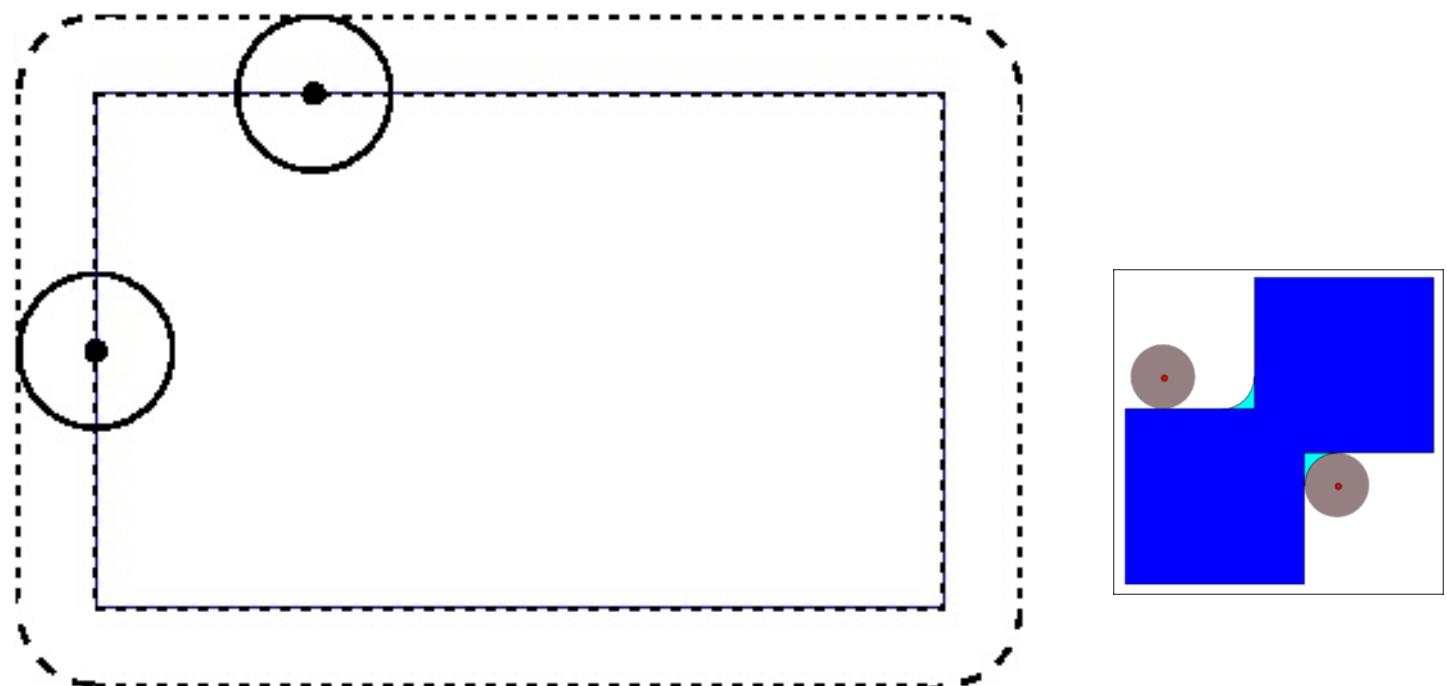


8-Neighborhood



# Closure

- Closure defined as  $X \cdot B = (X \oplus \check{B}) \ominus B$



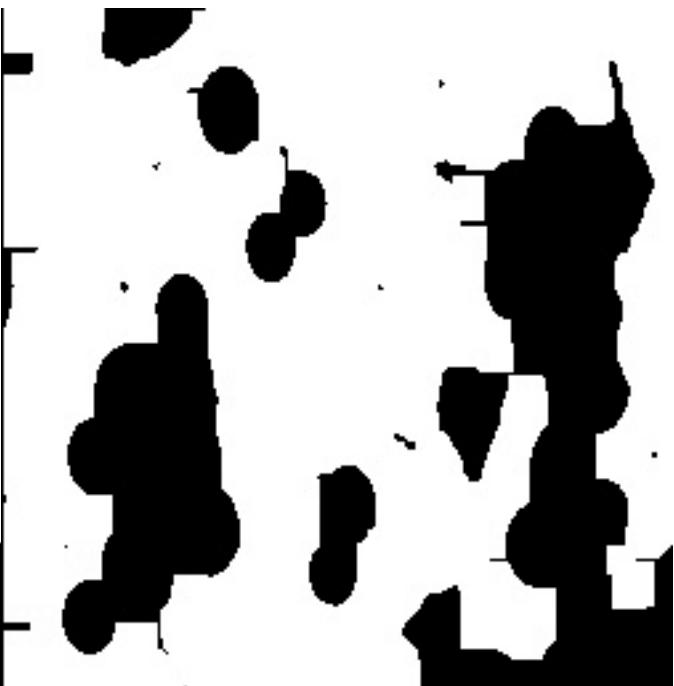
Closure of a set is the eroded of the dilated set (if  $B$  is symmetric)



4 itérations



4-élément structurant



8-élément structurant



# Link with Partial Differential Equation

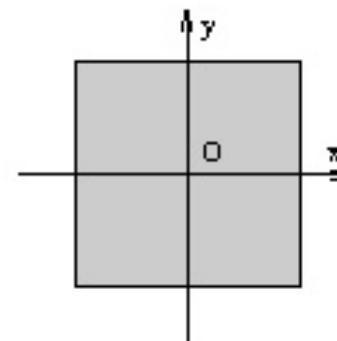
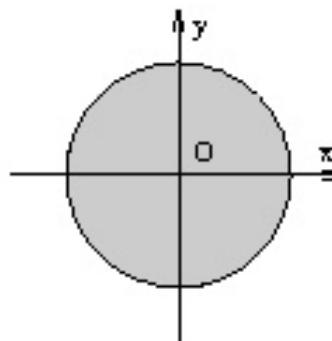
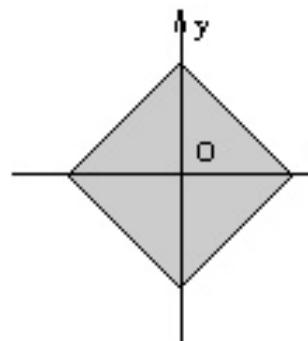
- Erosion

$$\frac{\partial I}{\partial t} = -\|\nabla I\|_p$$

- Dilation

$$\frac{\partial I}{\partial t} = \|\nabla I\|_p$$

Choice of p-norm



# Distance Maps

- Allow
  - implementation of dilation / erosion of large size (with thresholding)
  - Compute easily the skeletonization of a binary shape (as regional maxima of distance function)