

# Filtrage Mean Shift

Théorie, application,  
implémentation

*Thomas Grenier*

*Creatis*

# Plan

---

- I. Du filtrage Gaussien au filtrage *Mean Shift*  
*via la diffusion anisotrope et le filtrage bilatéral*
  
- II. *Mean Shift* et la théorie de l'estimation  
*pdf, espace des caractéristiques, noyaux, échelles*
  
- III. Applications  
*filtrage, segmentation, vidéo, à priori*
  
- IV. Solutions d'implémentations

I - Du filtrage Gaussien

au filtrage *Mean Shift*

# Introduction

- Sens de « filtrage »?

- (restauration)

- De-noising :

- Filtering :

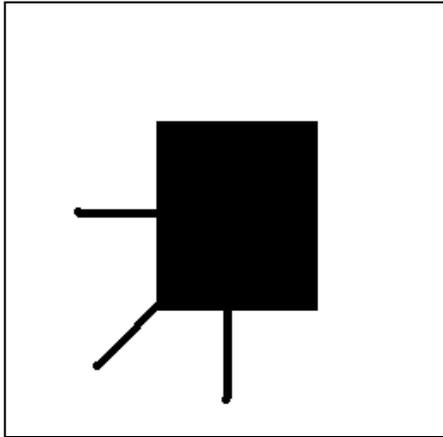
- Pour la suite filtrage:

- première étape d'un traitement,

- (ie. deuxième étape : segmentation, recalage, ...)



# Filtrage Moyenneur



original

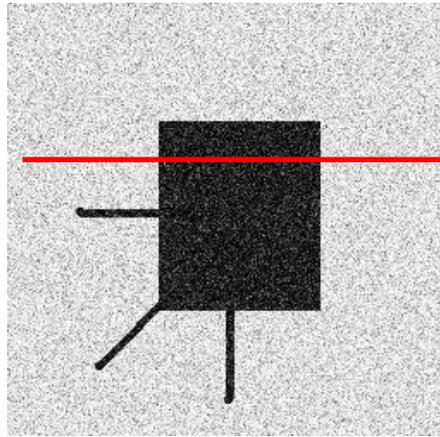
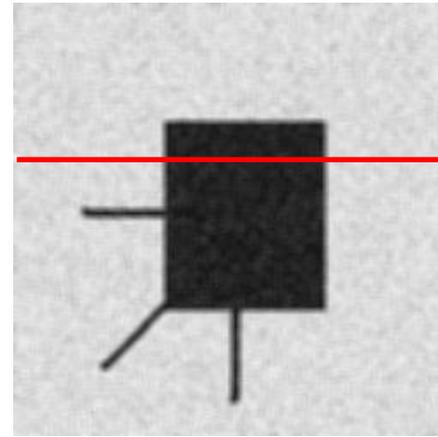
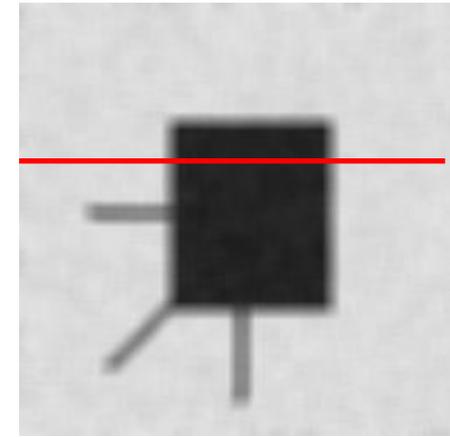


Image bruitée  
 $\sigma = 45$

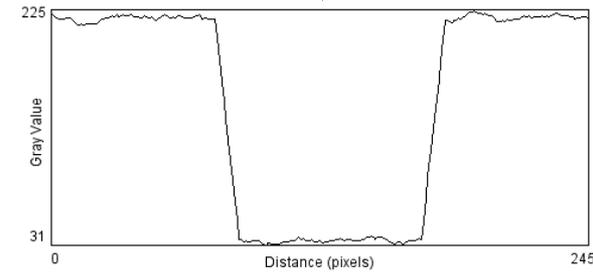
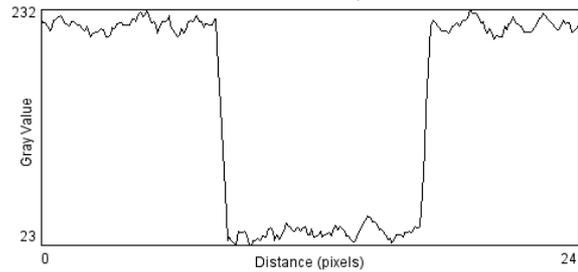
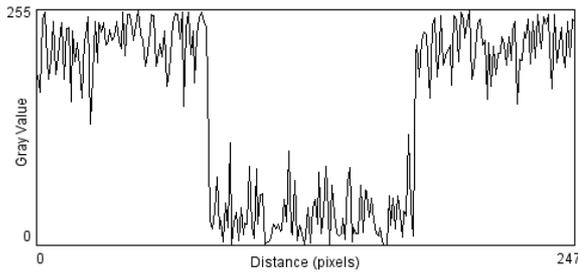


Moyenneur  
3x3



Moyenneur  
11x11

Profils :



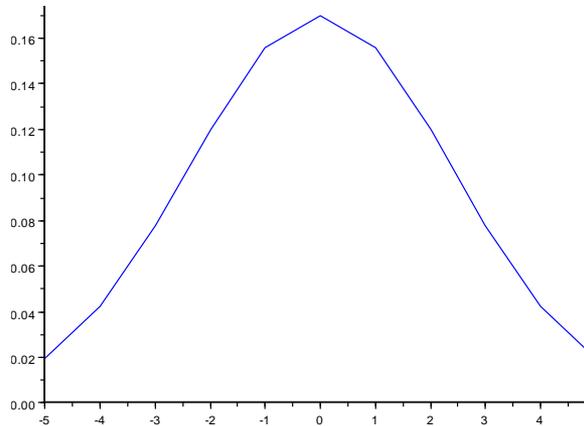
# Filtrage Gaussien

- Convolution par une fenêtre dont les coefficients sont fonctions de

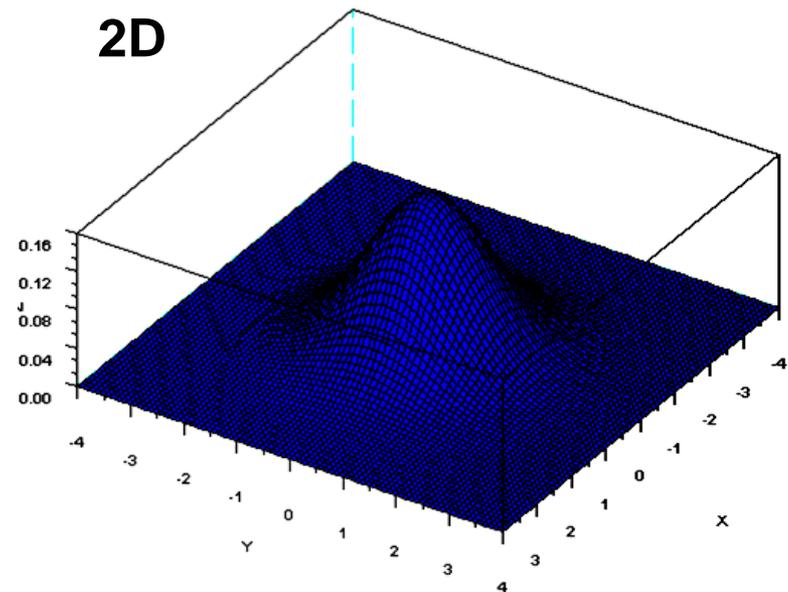
$$K_g(\mathbf{H}^{-1/2}\mathbf{x}) = (2\pi)^{-d/2} \cdot \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{H}^{-1}\mathbf{x}\right)$$

*Gaussienne généralisée*

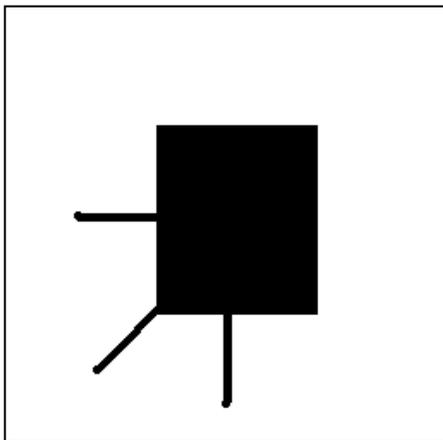
**1D**



**2D**



# Filtrage Gaussien



original

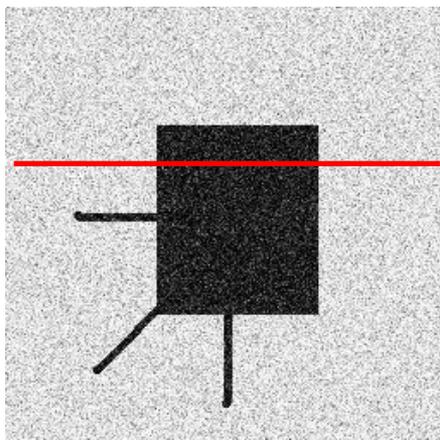
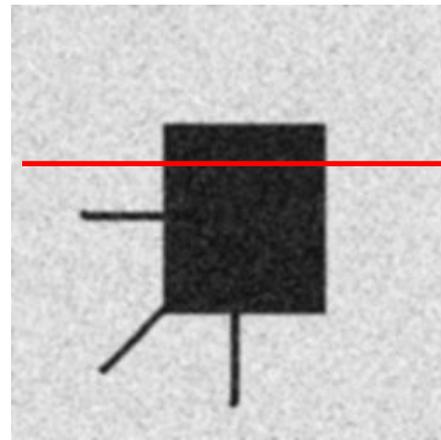
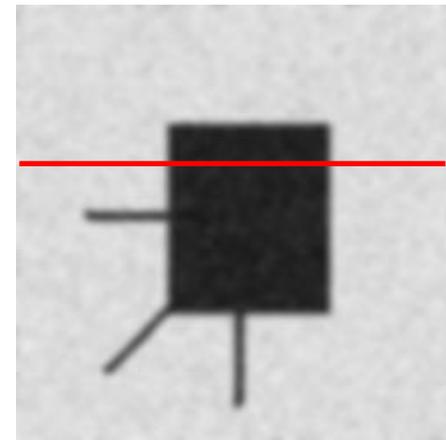


Image bruitée  
 $\sigma = 45$

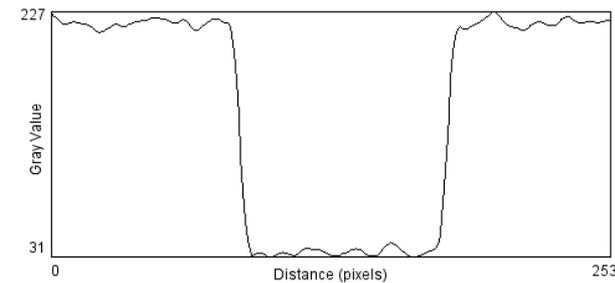
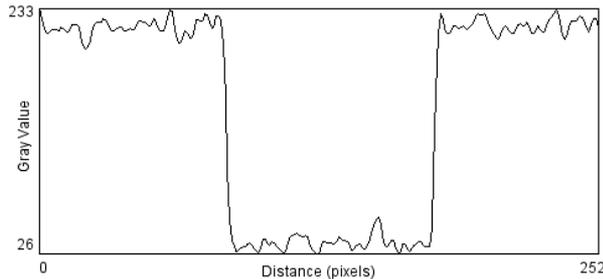
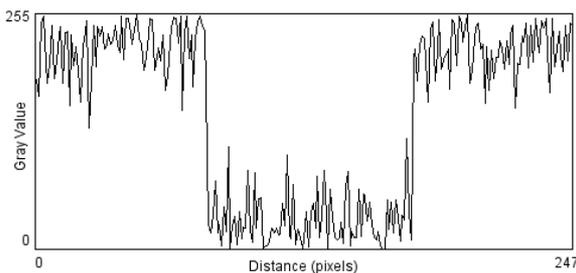


Filtr. Gauss.  
3x3



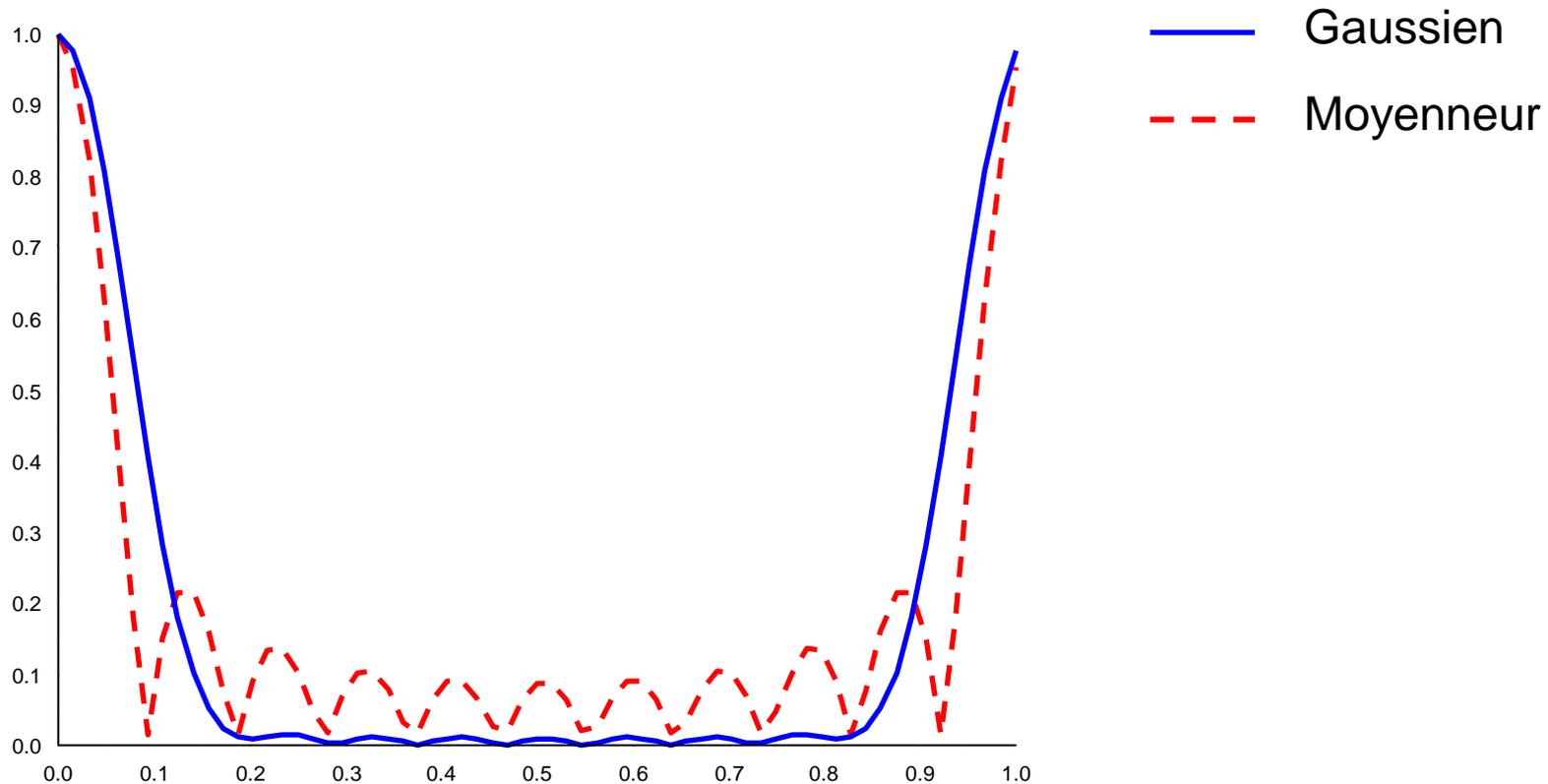
Filtr. Gauss.  
11x11

Profils :



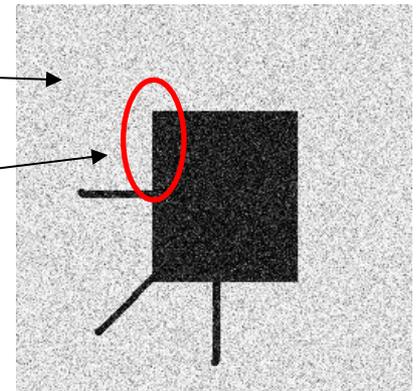
# Filtrage Gaussien vs. Moyenneur

- FFT pour 11 coefficients (puis zéro padding)



# Filtrage Gaussien & Moyenneur

- Problème de la localisation des contours...
  - ➔ Problème des transitions rapides (hautes fréquences)
- Transformer les filtres en passe bande ?
  - ❑ Détermination de la bande passante?
  - ❑ Amplification du bruit ?
  - ❑ Utilisation **localisée** du passe bande ?



# Filtrage gaussien et $PDE^*$

*\*Equations aux Dérivées Partielles*

- Equation de diffusion (isotrope) de la chaleur

$$\frac{\partial T(x,t)}{\partial t} = D \cdot \frac{\partial^2 T(x,t)}{\partial x^2}$$

1D

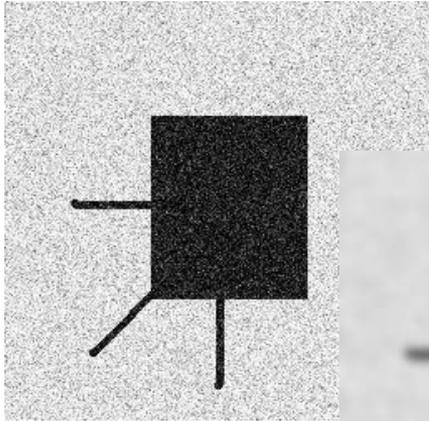
- Solution analytique (*fonction de Green*)

$$T(x,t) = \int_{-\infty}^{+\infty} f_0(x_0) \cdot G(x-x_0, t) \cdot dx_0$$

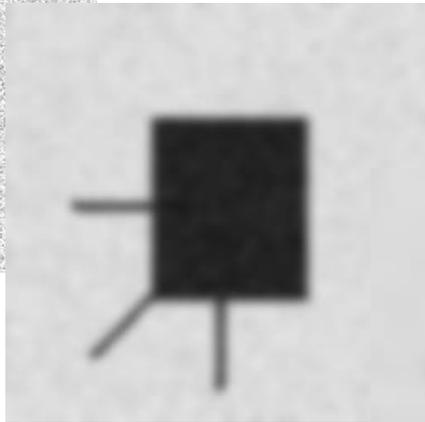
avec  $G(x-x_0, t) = \frac{1}{\sqrt{4\pi \cdot D \cdot t}} \cdot e^{-\frac{(x-x_0)^2}{4 \cdot D \cdot t}}$

➔ Convolutions successives  
par un filtre Gaussien !

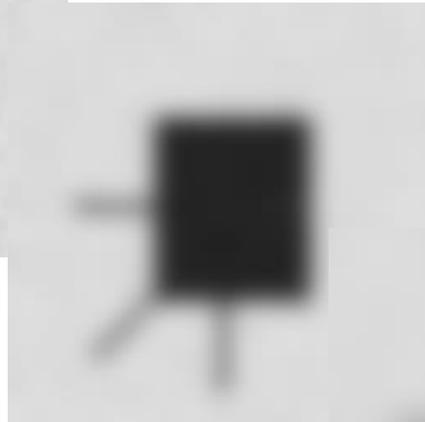
# Filtrage gaussien et *PDE*



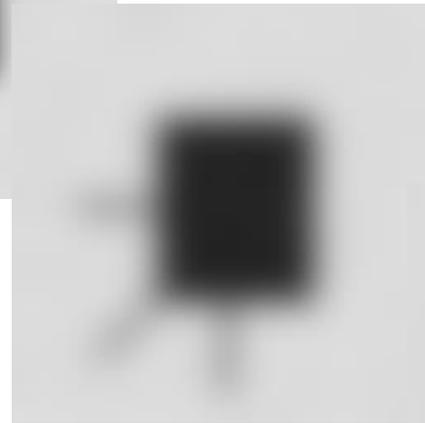
$t = 0$



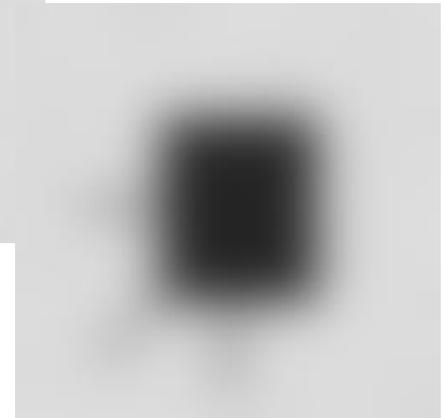
$t = 1$



$t = 8$



$t = 20$



$t = 60$

Diffusion en fonction du temps

# Filtrage anisotrope

- Idée : modifier l'équation de diffusion isotrope...

- Équation de diffusion isotrope

$$\frac{\partial I(\mathbf{x}, t)}{\partial t} = \text{div}(c \cdot \nabla I)$$

$I$  : image  $\mathbf{R}^s \rightarrow \mathbf{R}$

$s$  : nombre de dimensions spatiales

$\mathbf{x} = (x_1, x_2, \dots, x_s)^T$  : position spatiale

$t$  : paramètre artificiel de temps

$I(\mathbf{x}, t)$  : valeur de  $I$  à la position  $\mathbf{x}$  à l'instant  $t$

$c$  : constante (scalaire) de diffusion

# Filtrage anisotrope

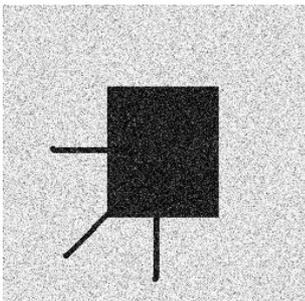
- ... en une équation anisotrope

$$\frac{\partial I(\mathbf{x}, t)}{\partial t} = \text{div}(g(\mathbf{x}, t) \cdot \nabla I)$$

n Dimensions

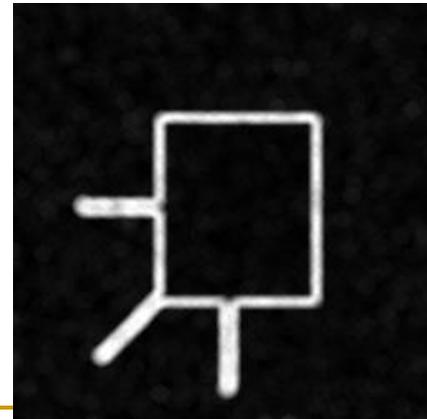
- Rôle de la fonction  $g(\mathbf{x}, t)$  ?
  - Pondérer l'amplitude du gradient en  $I(\mathbf{x}, t)$   
➔ **comment pondérer ?**

$I(\mathbf{x}, t)$



Norme du gradient

$$\|\nabla I(\mathbf{x}, t)\|$$



# Filtrage anisotrope

## ■ Equation de diffusion de Perona-Malik (1990)

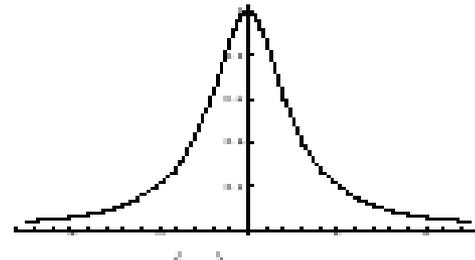
$$\frac{\partial I(\mathbf{x}, t)}{\partial t} = \operatorname{div}(g(\|\nabla I\|) \cdot \nabla I)$$

$g(u)$  : fonction de pondération,  $g(u) \rightarrow 0, u \rightarrow \infty$

$\|\nabla I\|$  : amplitude du gradient

$g(\|\nabla I\|)$  : fonction d'arrêt aux « bords »

$$g(u) = \frac{1}{1 + \frac{u^2}{K^2}}, \quad K = C^{ste} > 0$$



# Filtrage anisotrope

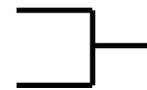
## ■ Formulation discrète de Perona-Malik

$$I_{\mathbf{x}}^{t+1} = I_{\mathbf{x}}^t + \frac{\lambda}{|\eta_{\mathbf{x}}|} \sum_{\mathbf{p} \in \eta_{\mathbf{x}}} g(\nabla I_{\mathbf{x},\mathbf{p}}) \nabla I_{\mathbf{x},\mathbf{p}}$$

$\eta_{\mathbf{x}}$  Voisinage de  $\mathbf{x}$

$\lambda$  Coefficient de diffusion

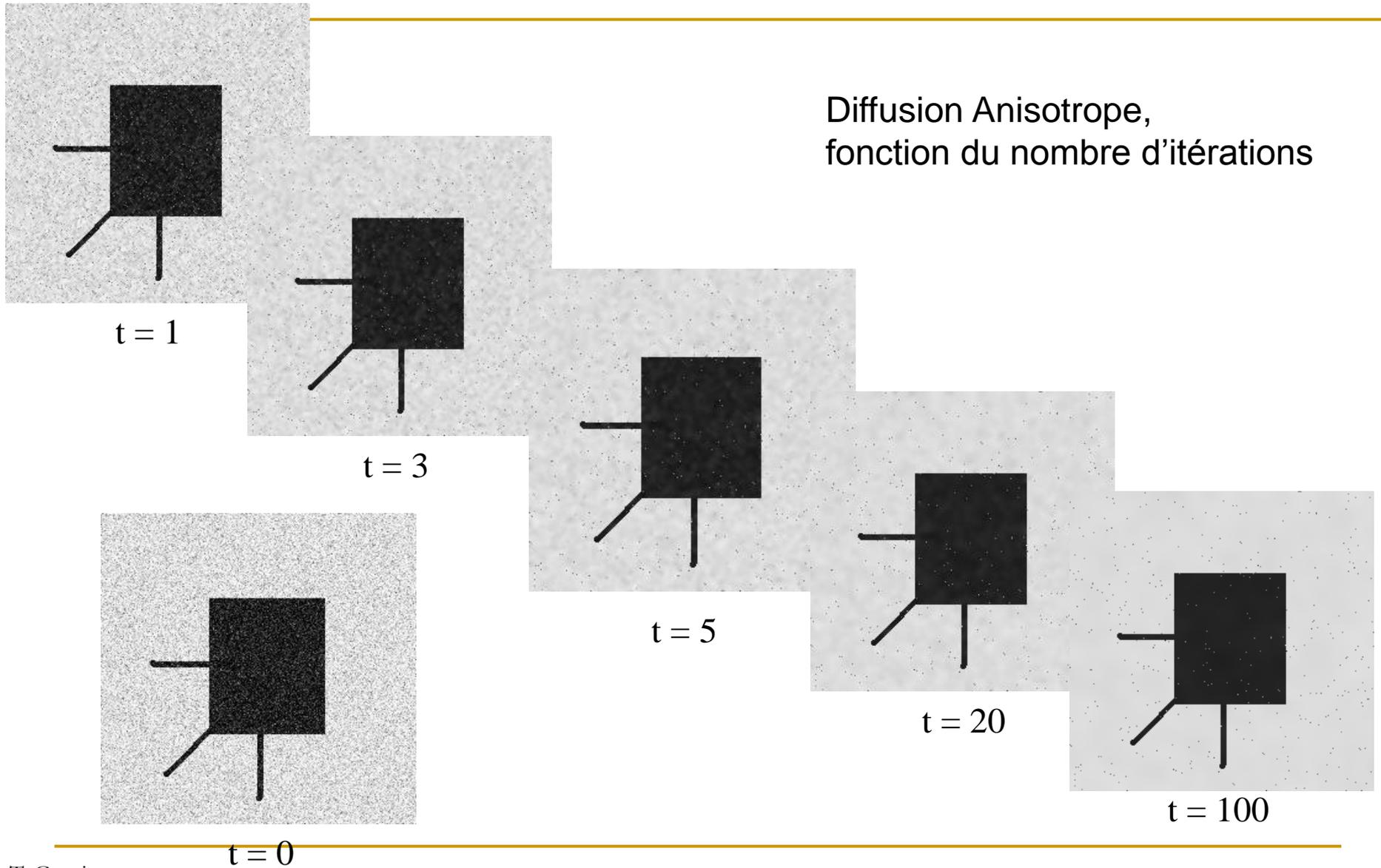
approximation du gradient :  $\mathbf{p} \in \eta_{\mathbf{x}}, \nabla I_{\mathbf{x},\mathbf{p}} = I_{\mathbf{p}} - I_{\mathbf{x}}^t,$



**Stabilité ...**

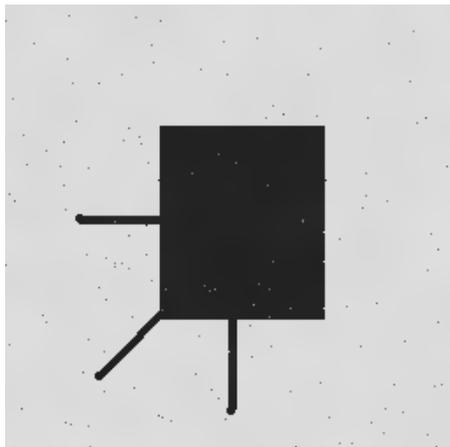
# Filtrage anisotrope

Diffusion Anisotrope,  
fonction du nombre d'itérations

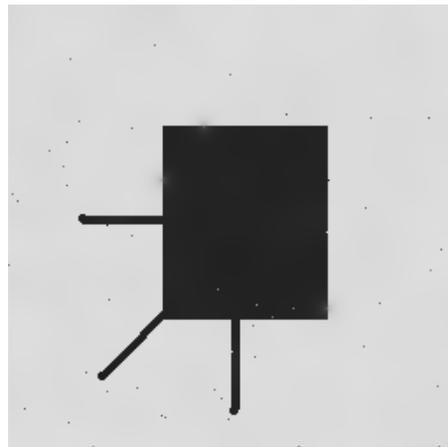


# Filtrage anisotrope

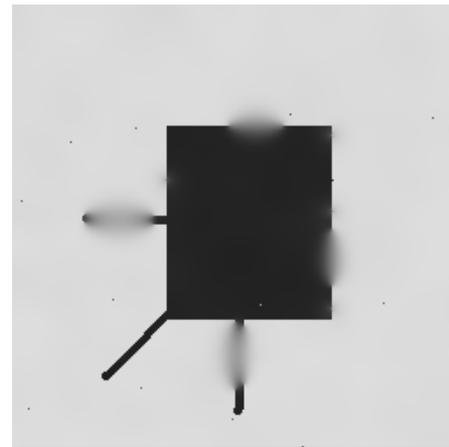
- Influence de la fonction de pondération  $g(x)$ 
  - Exemple: seuil de filtrage pour le gradient ( $t = 100$ )



Seuil=60



Seuil=70

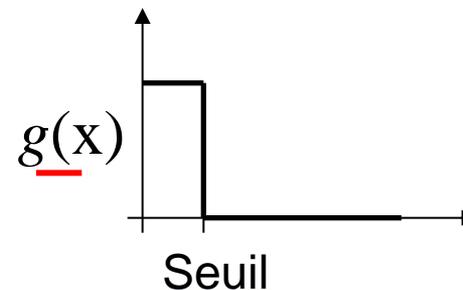


Seuil=80

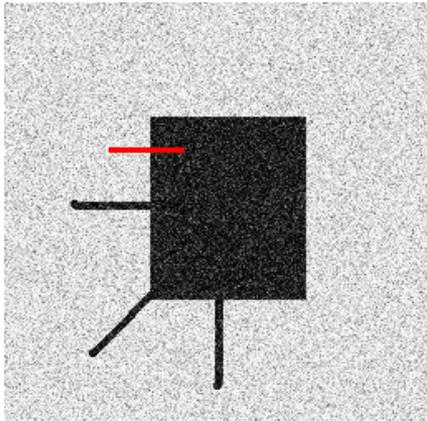


Seuil=255

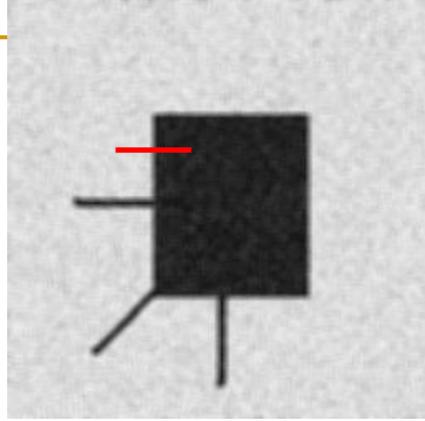
$$I_x^{t+1} = I_x^t + \frac{\lambda}{|\eta_x|} \sum_{p \in \eta_x} \underline{g}(\nabla I_{x,p}) \nabla I_{x,p}$$



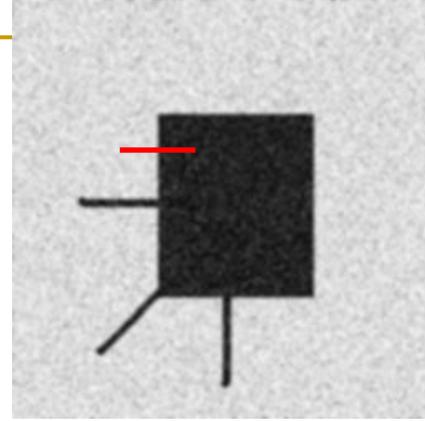
# Filtrage anisotrope



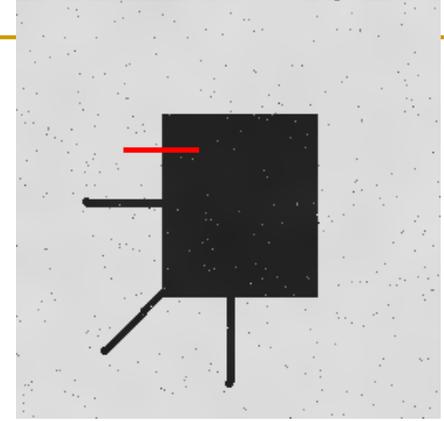
original



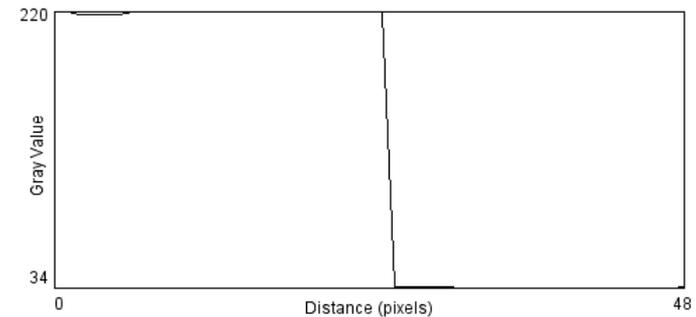
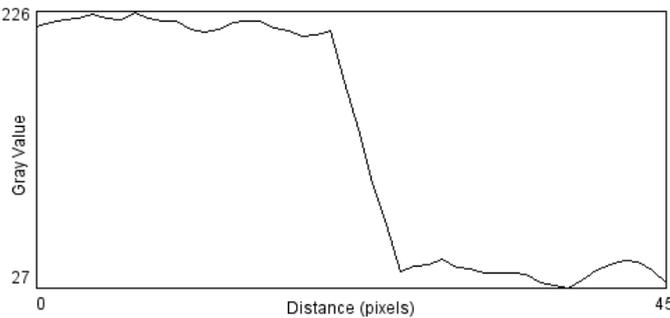
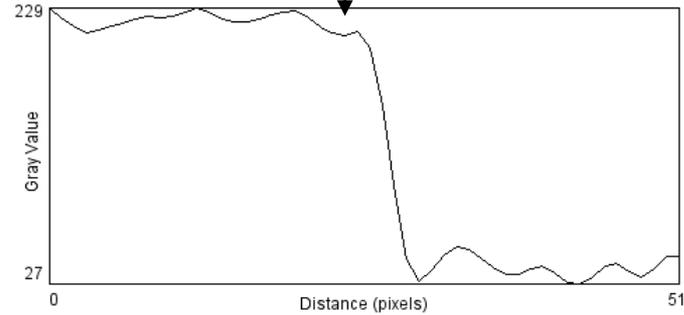
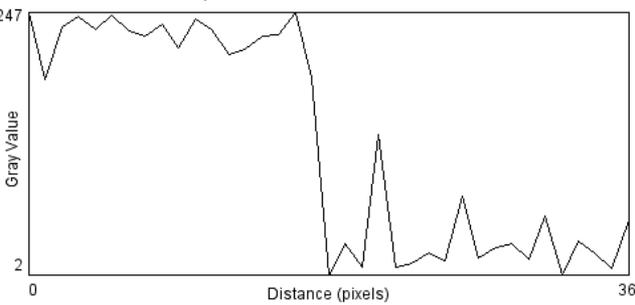
Filtr. Moyen.  
3x3



Filtr. Gauss.  
3x3



Filtr. Anisotrope  
(t=100)



# Filtrage anisotrope

- Temps ?
- Démo VTK+Python:

```
import vtk

# Create the image
reader = vtk.vtkBMPReader()
reader.SetFileName("../Data/Original_2D_Noise45.bmp")

diffusion = vtk.vtkImageAnisotropicDiffusion2D()
diffusion.SetInput(reader.GetOutput() )
diffusion.SetDiffusionFactor(0.1)
diffusion.SetDiffusionThreshold(80.0)
diffusion.SetNumberOfIterations(100)

ia = vtk.vtkImageActor()
ia.SetInput(diffusion.GetOutput())

writer = vtk.vtkBMPWriter()
writer.SetInput(diffusion.GetOutput() )
writer.SetFileName("Output.bmp")
writer.Write()
```

Vtk4, python2.4

# Filtrage anisotrope

## ■ Autre forme discrète (Barash 2002)

Image à l'itération  $t$

$$\mathbf{I}^{t+1}(\mathbf{x}) = \frac{\sum_{i=-1}^1 \sum_{j=-1}^1 \mathbf{I}^t(\mathbf{x}_{i,j}) \cdot w^t(\mathbf{x}_{i,j})}{\sum_{i=-1}^1 \sum_{j=-1}^1 w^t(\mathbf{x}_{i,j})}$$

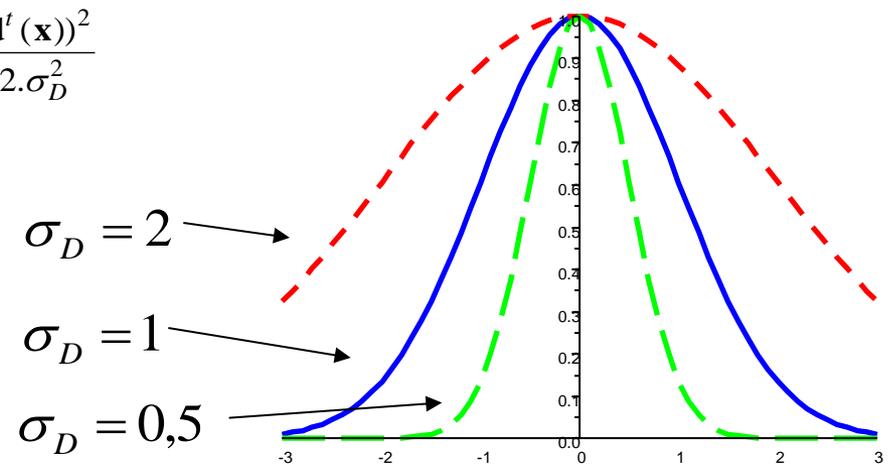
Avec :

$$\mathbf{x} = (x_1, x_2)$$

$$\mathbf{x}_{i,j} = (x_1 + i, x_2 + j)$$

$$d^t(\mathbf{x}) = \|\nabla \mathbf{I}^t(\mathbf{x})\|$$

$$w^t(\mathbf{x}) = e^{-\frac{(d^t(\mathbf{x}))^2}{2 \cdot \sigma_D^2}}$$



# Filtrage anisotrope

- Bilan:
  - Permet un filtrage adaptatif
    - Dépendant de la norme du gradient
    - Introduction d'une notion de distance
  - Filtrage de données scalaires (niveaux de gris)
  - Problème du choix des paramètres
    - Nombre d'itérations
    - Fonction de pondération (et seuil)
    - Coefficient de diffusion (problème de stabilité)
    - ...

# Filtrage bilatéral

- Filtrer des données couleurs ?
  - Filtrage bilatéral (Tomasi 1998)

$$\mathbf{I}^{t+1}(\mathbf{x}) = \frac{\sum_{i=-h}^h \sum_{j=-h}^h \mathbf{I}^t(\mathbf{x}_{i,j}) \cdot w^t(\mathbf{x}, \mathbf{x}_{i,j})}{\sum_{i=-h}^h \sum_{j=-h}^h w^t(\mathbf{x}, \mathbf{x}_{i,j})}$$

Avec:

$$\mathbf{I}^t(\mathbf{x}) = [I_R(\mathbf{x}), I_G(\mathbf{x}), I_B(\mathbf{x})]^T$$

$$w^t(\mathbf{x}, \mathbf{y}) = \underbrace{\exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2 \cdot \sigma_S^2}\right)}_{\text{Domaine spatial}} \cdot \underbrace{\exp\left(-\frac{\|\mathbf{I}^t(\mathbf{x}) - \mathbf{I}^t(\mathbf{y})\|^2}{2 \cdot \sigma_R^2}\right)}_{\text{Domaine des amplitudes}}$$

Domaine spatial

Domaine des amplitudes

# Filtrage bilatéral

## ■ Bilan:

- Permet un filtrage adaptatif
  - Dépendant de l'éloignement spatial des points
  - Dépendant de l'écart entre les amplitudes des points
  
- Filtrage de données vectorielles (RBG, ...)
  
- Problème du choix des paramètres
  - Nombre d'itérations
  - 2 fonctions de pondération (et seuil)
  - ... voisinage, ...

# Filtrage Mean Shift

- On continue de généraliser la forme itérative:

$$\mathbf{x}^t = \begin{bmatrix} \mathbf{x}_s^t \\ \mathbf{x}_r^t \end{bmatrix}$$

$$\mathbf{x}^{t+1} = \frac{\sum_{i=-h}^h \sum_{j=-h}^h \mathbf{x}_{i,j} \cdot g(\mathbf{x}_{i,j}, \mathbf{x}^t)}{\sum_{i=-h}^h \sum_{j=-h}^h g(\mathbf{x}_{i,j}, \mathbf{x}^t)}$$

$$\mathbf{x} \in \mathbf{R}^d$$

$d = s+r$ , domaine joint :  $s$  : spatial,

$r$  : « range » (valeur)

Exemple en 2D avec  
3 composantes couleurs:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix}, \mathbf{x}_{i,j} = \begin{bmatrix} x_1 + i \\ x_2 + j \\ r'_1 \\ r'_2 \\ r'_3 \end{bmatrix}, r'_i = I_i(x_1 + i, x_2 + j)$$

# Filtrage Mean Shift

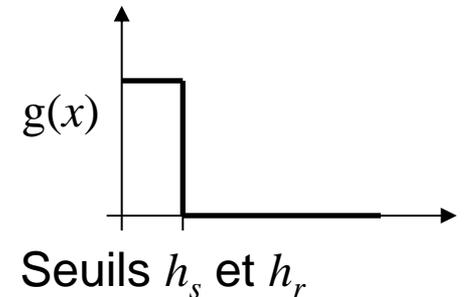
- On continue de généraliser la forme itérative:

$$\mathbf{x}^{t+1} = \frac{\sum_{i=-h}^h \sum_{j=-h}^h \mathbf{x}_{i,j} \cdot g(\mathbf{x}_{i,j} - \mathbf{x}^t)}{\sum_{i=-h}^h \sum_{j=-h}^h g(\mathbf{x}_{i,j} - \mathbf{x}^t)}$$

(Fukunaga 1975  
Comaniciu 1999)

$$\mathbf{x}^t = \begin{bmatrix} \mathbf{x}_s^t \\ \mathbf{x}_r^t \end{bmatrix}$$

$$g(\mathbf{x}) = g_s(\|\mathbf{x}_s\|^2) \cdot g_r(\|\mathbf{x}_r\|^2)$$

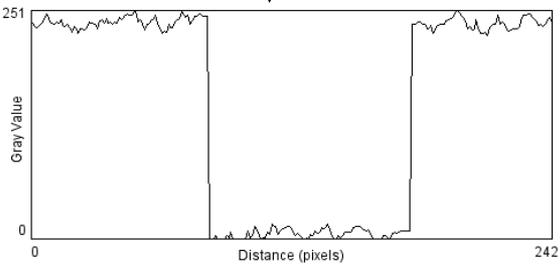
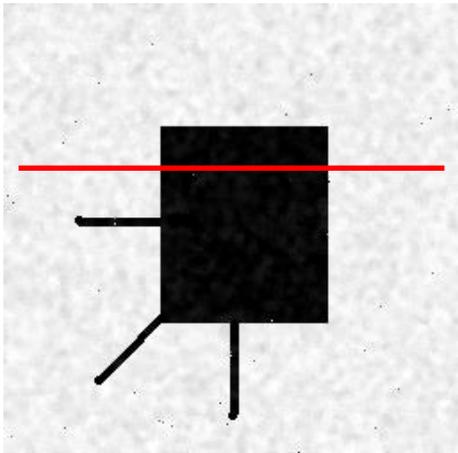


- ➔ **Processus itératif convergent !!**
- ➔ **Toutes les coordonnées évoluent !!**
- ➔ **2 seuils à régler:  $h_s, h_r$**
- ➔ **Calculs longs (plus longs que les précédents)**

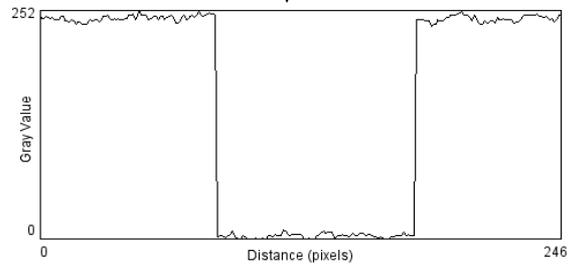
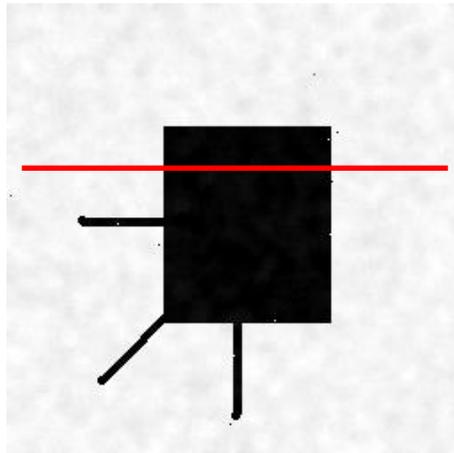
# Filtrage Mean Shift

## ■ Résultats

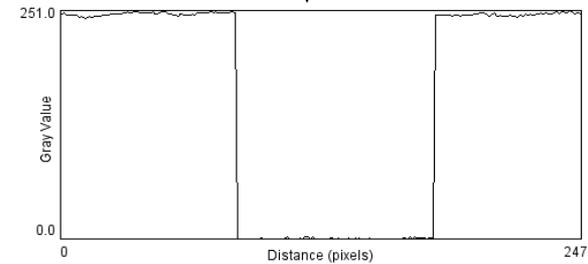
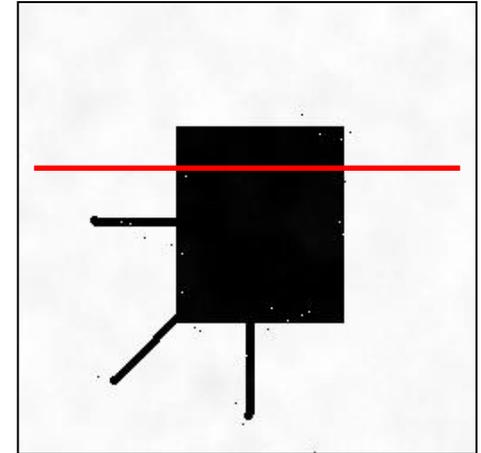
$h_s=3, h_r=70$



$h_s=5, h_r=70$



$h_s=10, h_r=70$



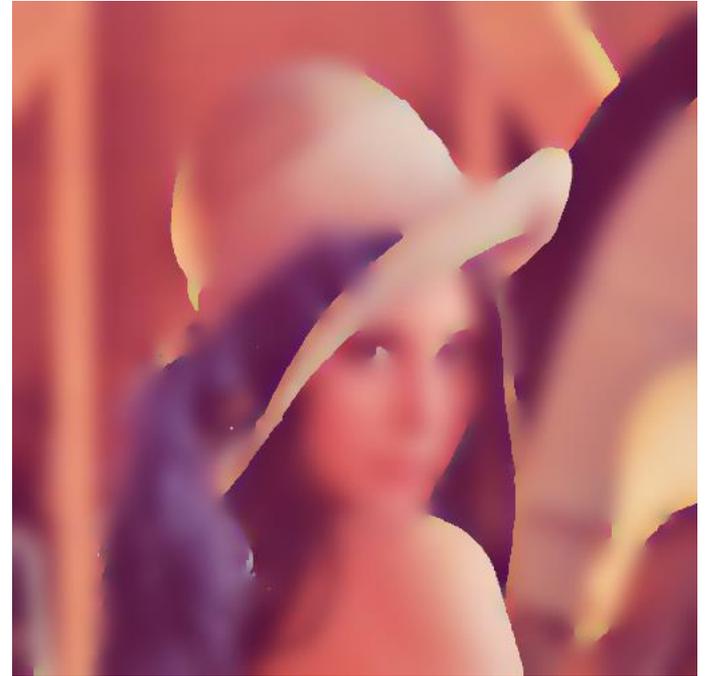
# Tests, image originale



# Tests Diffusion anisotrope



200 it. Seuil Gradient 10

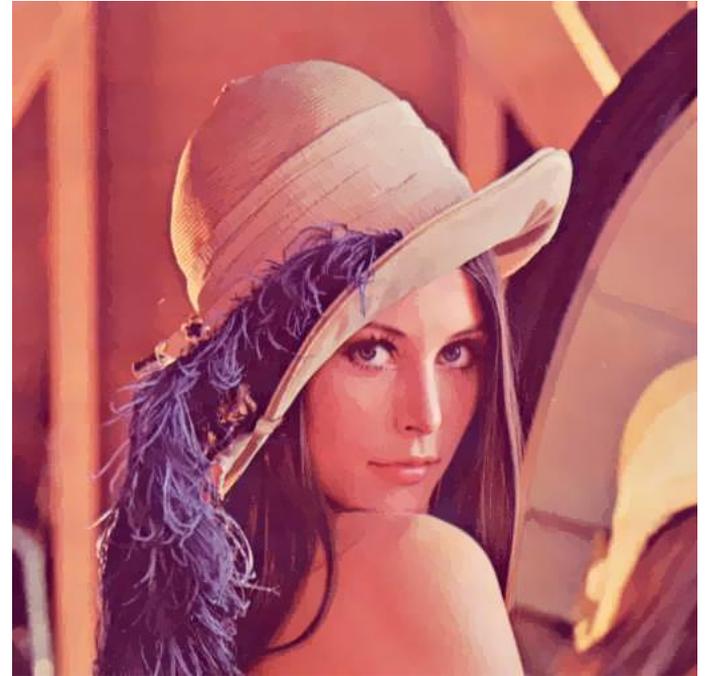


200 it.  
Seuil Gradient 30

# Tests Mean Shift



$h_s = 10, h_r = 30$



$h_s = 5, h_r = 10$

# II – Mean Shift et la théorie de l'estimation

# Définitions et notions préliminaires

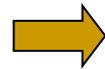
- Espace des caractéristiques
- Estimation non paramétrique de densité de probabilité par noyau
- Paramètres d'échelle

# Espace des caractéristiques

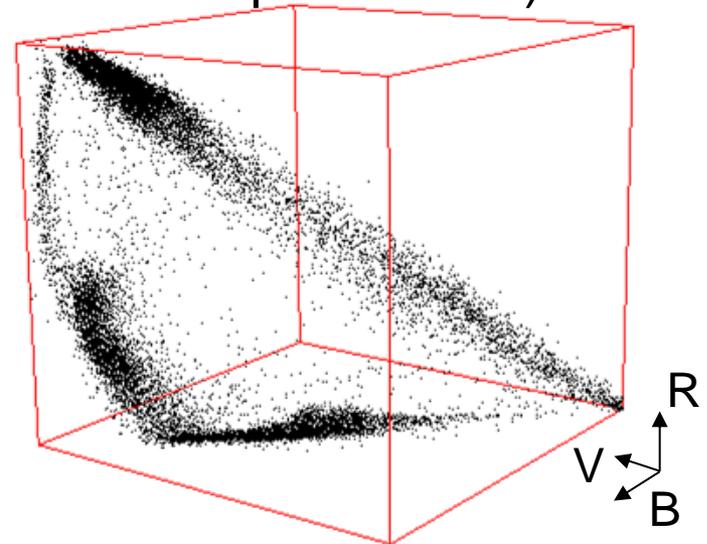
- Représenter les caractéristiques des données (images) dans un même espace
- Exemple : image couleur (caractéristique couleur)



Image couleur



$$\mathbf{x}_i = \begin{bmatrix} r \\ v \\ b \end{bmatrix}$$



Espace des couleurs (RVB)

# Espace des caractéristiques

[Schölkopf '99]

## ■ Généralisation et notations

- «  $c$  » caractéristiques

- Chaque caractéristique  $\mathbf{x}_j$  ( $j=1..c$ ) appartient à  $\mathbf{R}^{d_j}$
- L'espace des caractéristiques est un espace euclidien

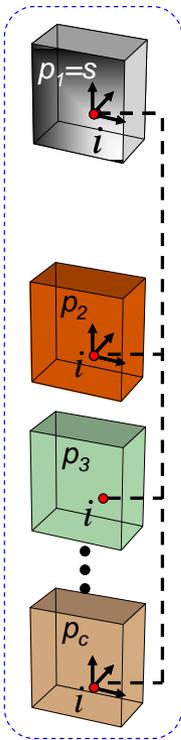
$$\mathbf{R}^d = \mathbf{R}^{d_1} \times \mathbf{R}^{d_2} \times \dots \times \mathbf{R}^{d_c}$$

$$d = d_1 + d_2 + \dots + d_c$$

- Un vecteur de l'espace des caractéristiques est noté

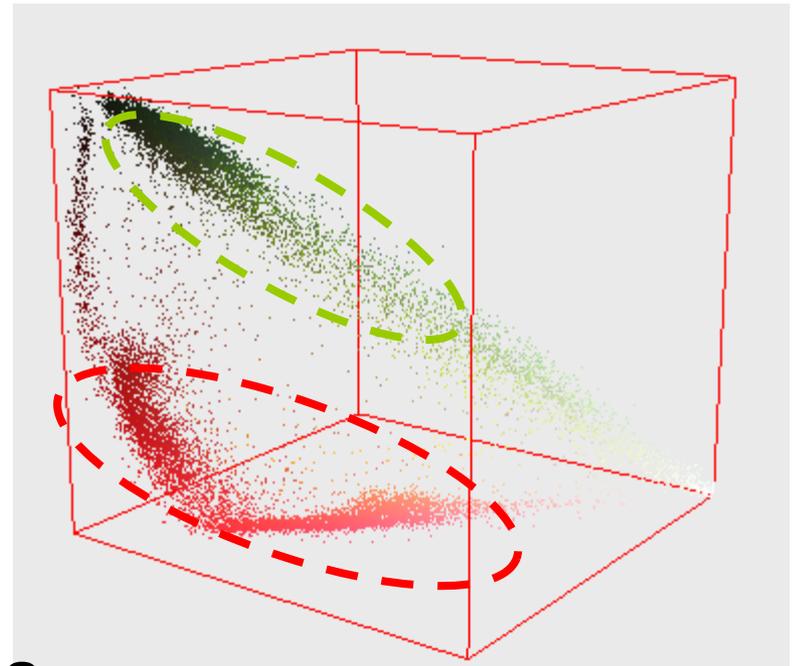
$$\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_j^T, \dots, \mathbf{x}_c^T]^T$$

- Chaque donnée d'entrée permet d'obtenir un vecteur  $\mathbf{x}_i$  dans l'espace des caractéristiques


 $\mathbf{x}_i$

# Espace des caractéristiques

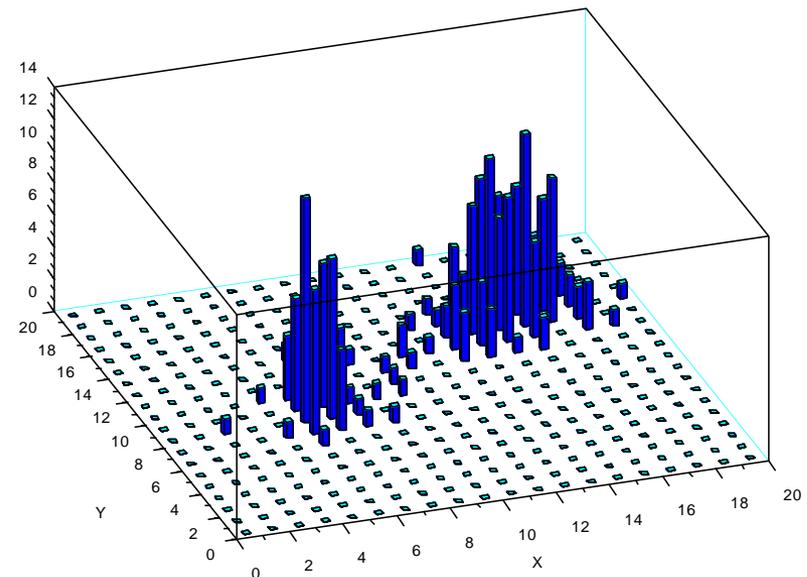
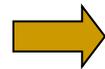
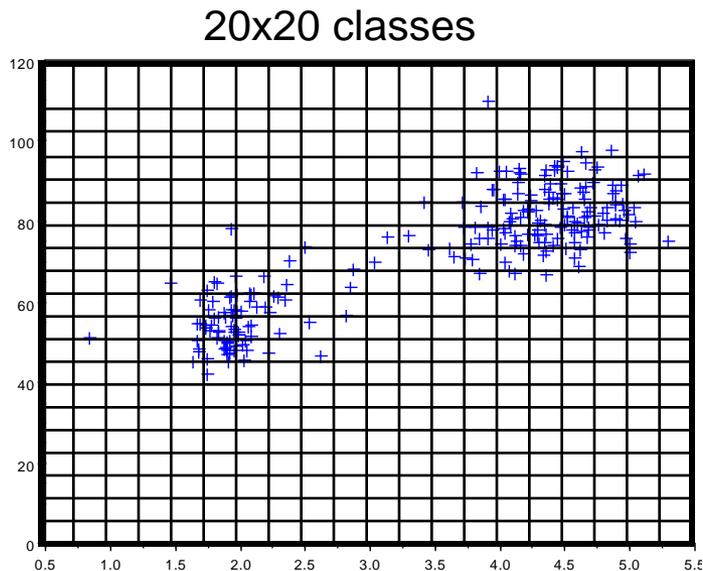
- Concentration de l'information dans l'espace des caractéristiques  
= information significative



- Estimation de densité dans l'espace des caractéristiques ?

# Estimation de densité

- Histogramme : une technique d'estimation de densité
  - Fixer le nombre de classes  $\Leftrightarrow$  **largeur des intervalles de valeurs**
  - Compter le nombre de points tombant dans chaque intervalle

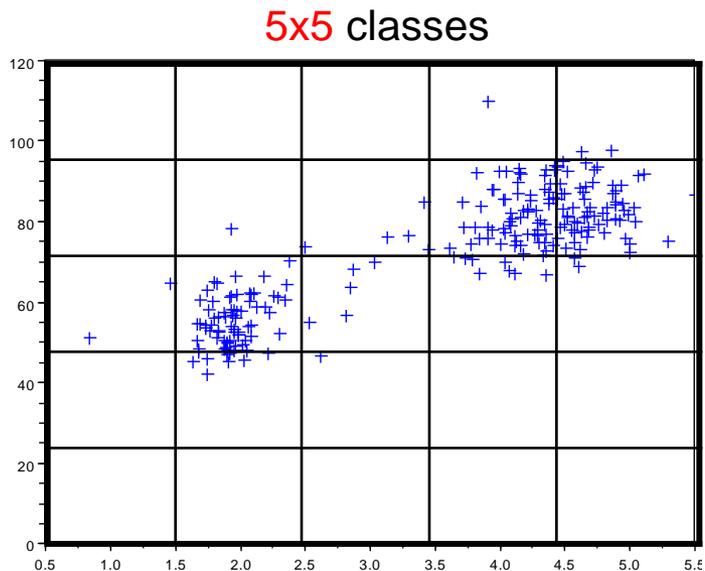


Espace des caractéristiques (2D)

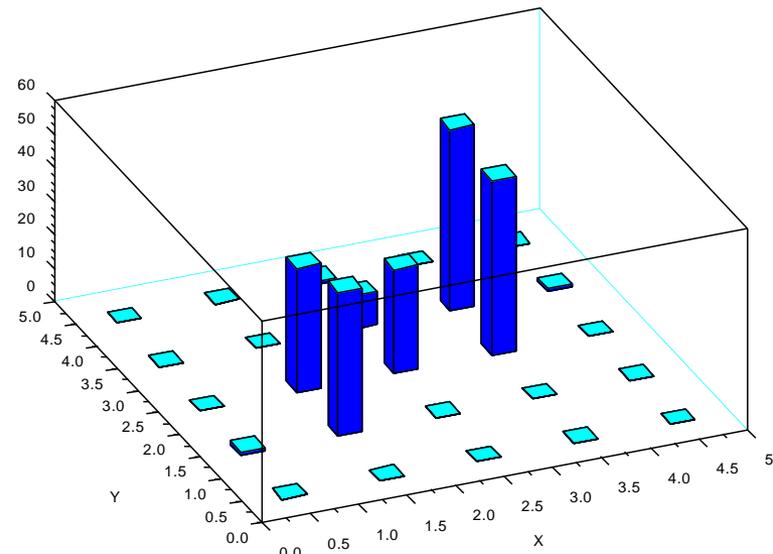
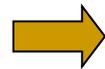
Histogramme (20x20 classes)

# Estimation de densité

- Histogramme : une technique d'estimation de densité
  - Fixer le nombre de classes  $\Leftrightarrow$  **largeur des intervalles de valeurs**
  - Compter le nombre de points tombant dans chaque intervalle



Espace des caractéristiques (2D)



Histogramme (5x5 classes)

# Estimation non paramétrique de densité par noyau

## ■ Estimateur de « Parzen »

[Parzen '62][Cacoullos '66]

$$\hat{f}(\mathbf{x}; \mathbf{H}) = \frac{1}{n} \sum_{i=1}^n K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)$$

$n$  : nombre de points  $\mathbf{x}_i$

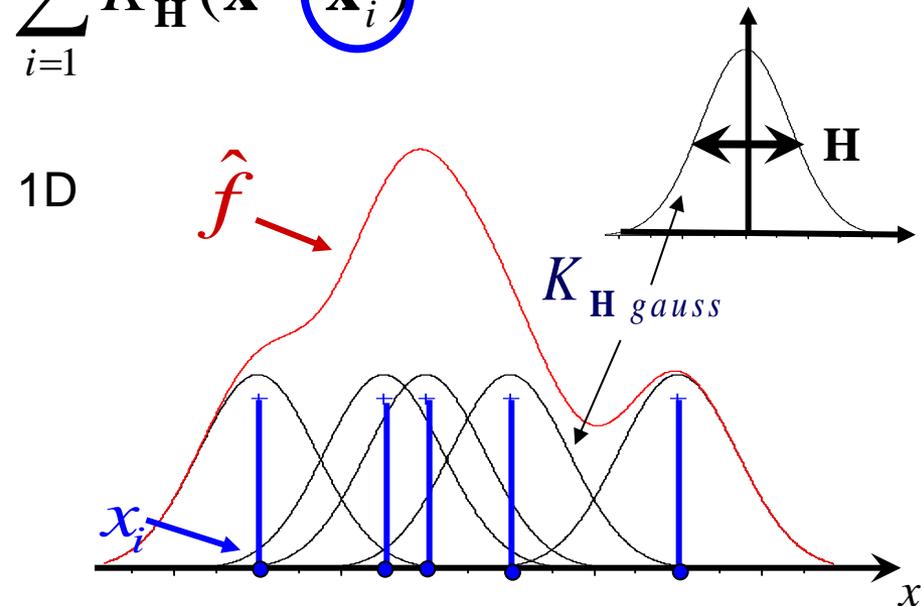
## ■ $K_{\mathbf{H}}(\mathbf{x})$ : noyau

Fonction de  $\mathbb{R}^d$  dans  $\mathbb{R}$

$$\int K_{\mathbf{H}}(\mathbf{x}) d\mathbf{x} = 1$$

## ■ $\mathbf{H}$ : matrice de paramètres d'échelle

matrice carrée ( $d \times d$ ), symétrique et définie positive  
(souvent diagonale)



# Estimation non paramétrique de densité par noyau

[Abramson '82]

## ■ Estimateur adaptatif

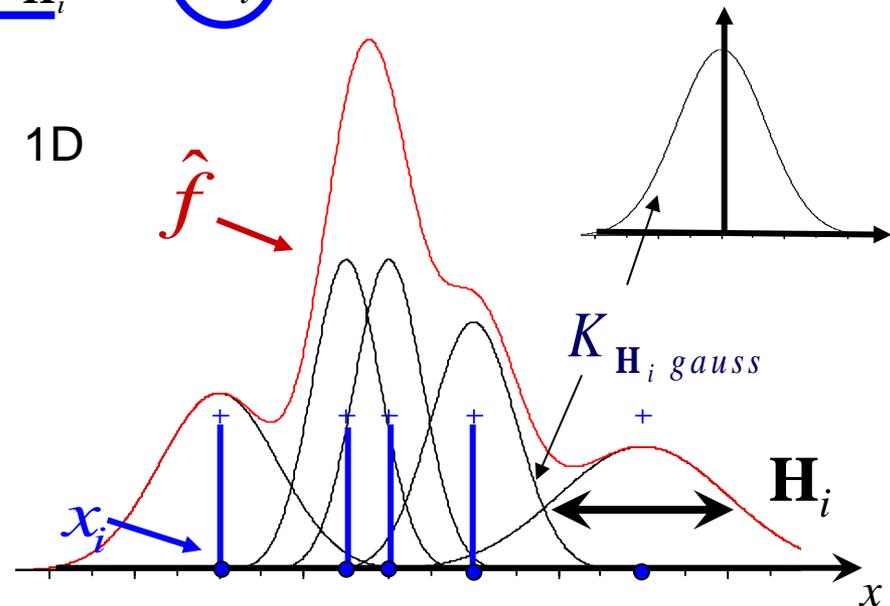
$$\hat{f}(\mathbf{x}; \mathbf{H}_i) = \frac{1}{n} \sum_{i=1}^n K_{\mathbf{H}_i}(\mathbf{x} - \mathbf{x}_i)$$

$n$  : nombre de points  $\mathbf{x}_i$

- $K_{\mathbf{H}_i}(\mathbf{x})$  : noyau  
Fonction de  $\mathbb{R}^d$  dans  $\mathbb{R}$

$$\int K_{\mathbf{H}_i}(\mathbf{x}) d\mathbf{x} = 1$$

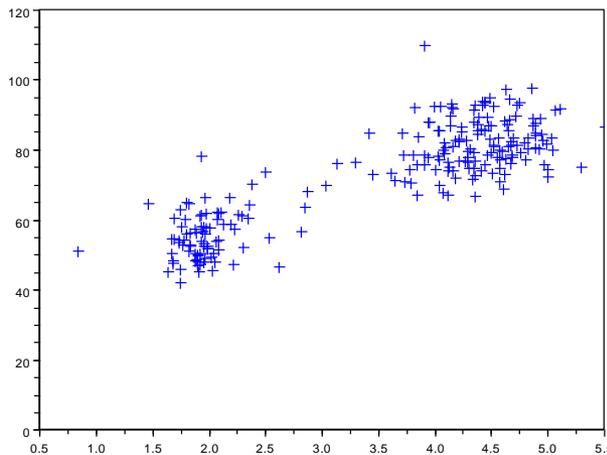
- $\mathbf{H}_i = \mathbf{H}(\mathbf{x}_i)$  : matrices de paramètres d'échelle (adaptatifs)  
matrices carrées ( $d \times d$ ), symétriques et définies positives  
(souvent diagonales)



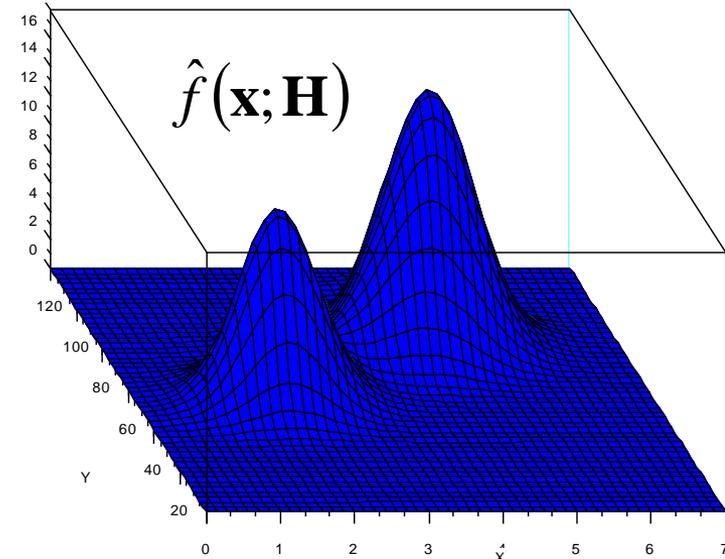
## non paramétrique de densité par noyau

### ■ Illustration 2D

$$K_{\mathbf{H}}^{\text{gauss}}(\mathbf{x}) = \text{img} \quad \mathbf{H} = \begin{bmatrix} 0,5^2 & 0 \\ 0 & 2^2 \end{bmatrix}$$

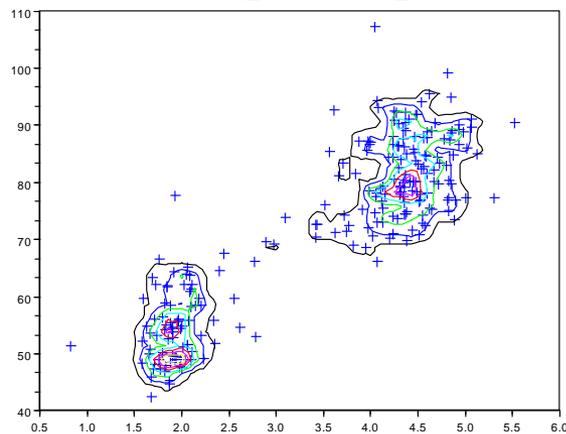


Espace des caractéristiques (2D)

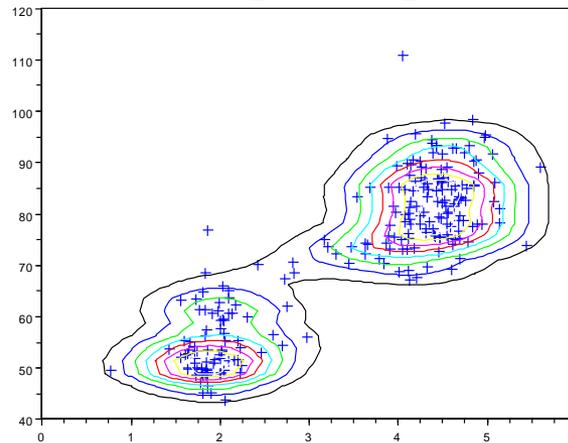


Estimation de densité (Parzen)

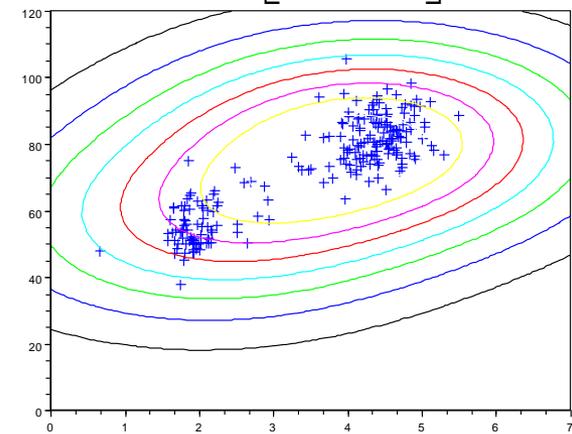
$$\mathbf{H} = \begin{bmatrix} 0,1^2 & 0 \\ 0 & 1^2 \end{bmatrix}$$



$$\mathbf{H} = \begin{bmatrix} 0,5^2 & 0 \\ 0 & 2^2 \end{bmatrix}$$



$$\mathbf{H} = \begin{bmatrix} 2^2 & 0 \\ 0 & 20^2 \end{bmatrix}$$

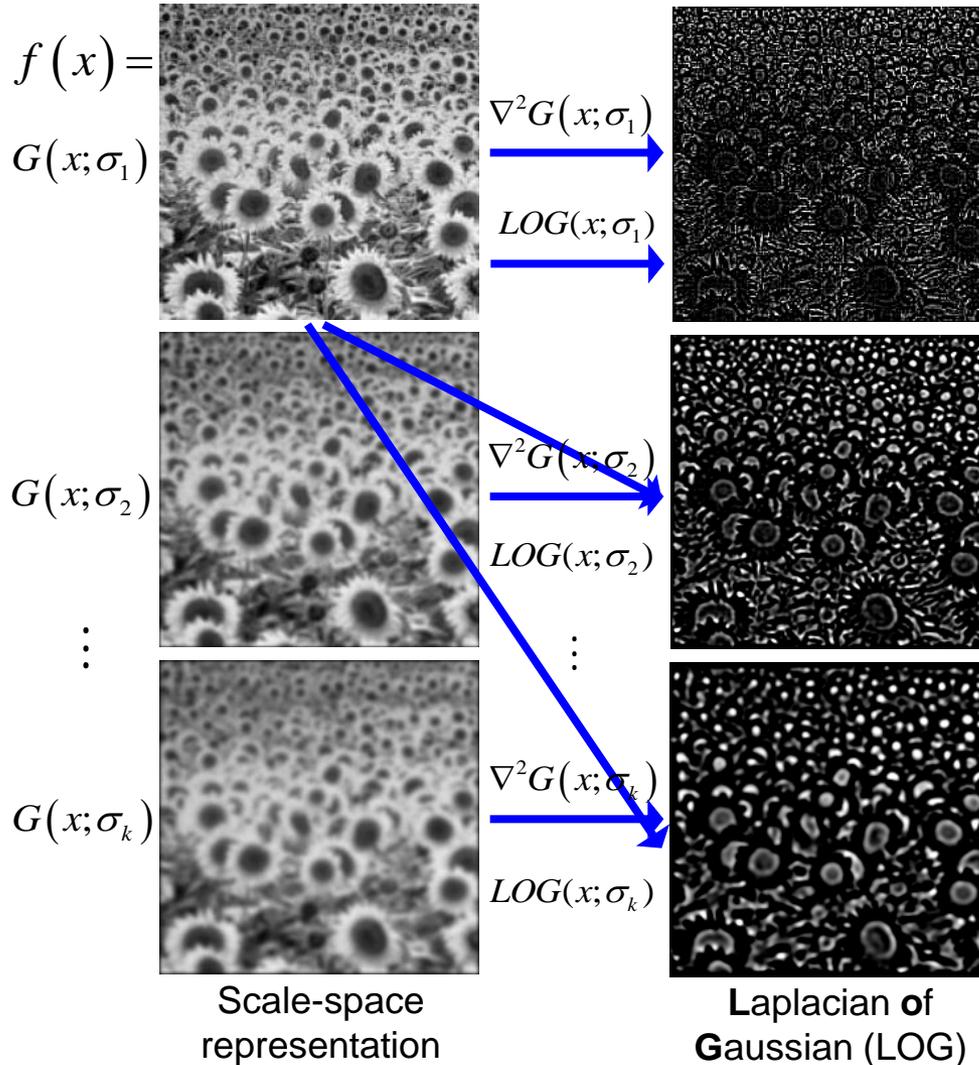


Iso-contours des estimations (avec  $K_{\mathbf{H} gauss}(\mathbf{x})$ )



# Paramètres d'échelle

[Lindeberg 1999]



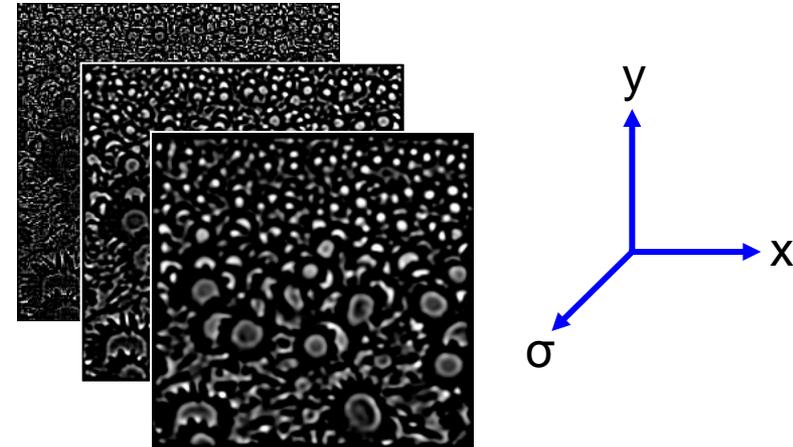
2D LOG filter with scale  $\sigma$

3D scale-space representation

$$LOG(x; \sigma) = \frac{2\sigma^2 - \|x\|^2}{2\pi\sigma^6} e^{-\frac{\|x\|^2}{2\sigma^2}}$$

$$\forall x \in f, \forall \sigma_{1..k} :$$

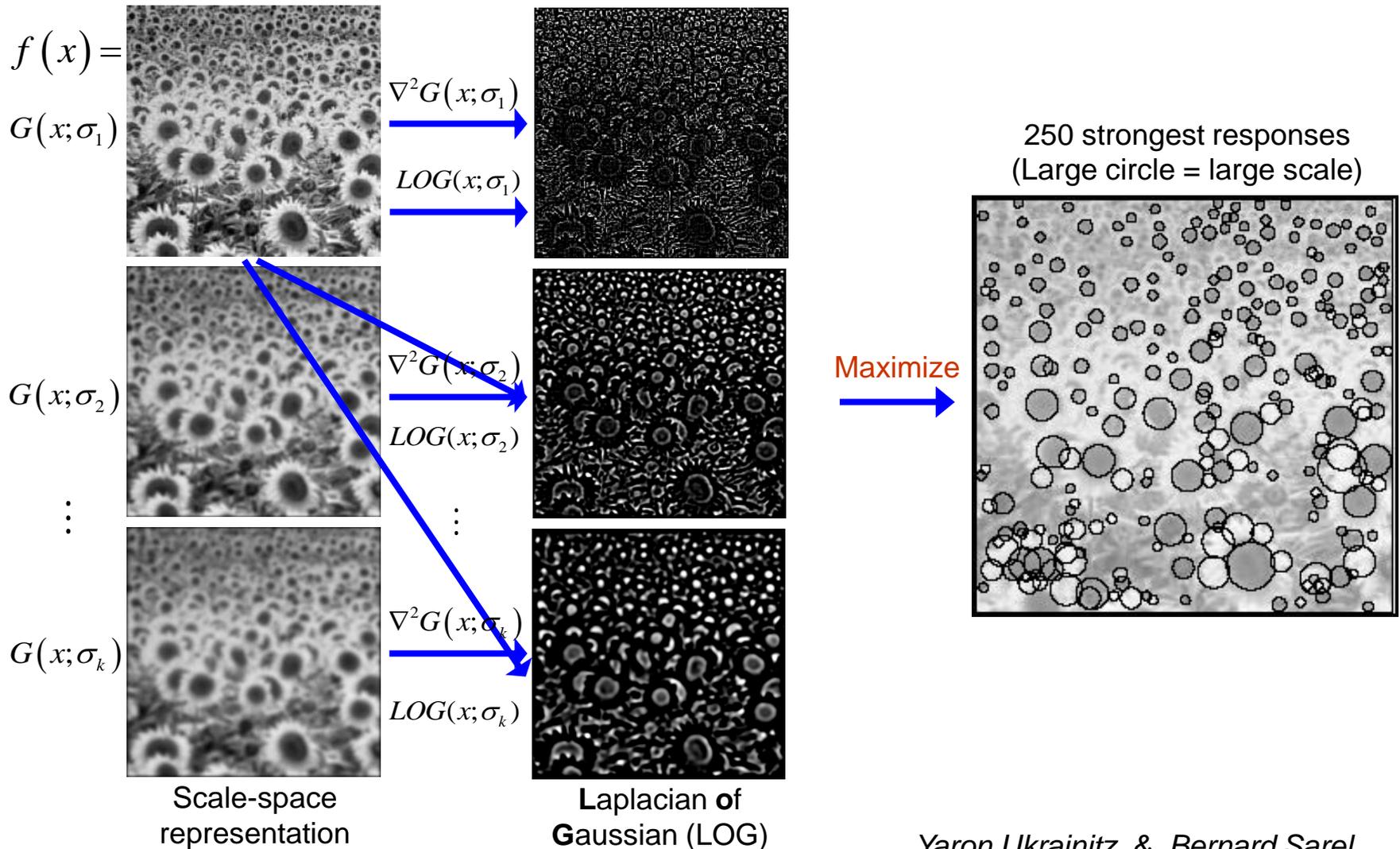
$$L(x, \sigma) = LOG(x; \sigma) * f(x)$$



Caractéristiques pertinentes:  
( $x, \sigma$ ) maximisant  $L$

# Paramètres d'échelle

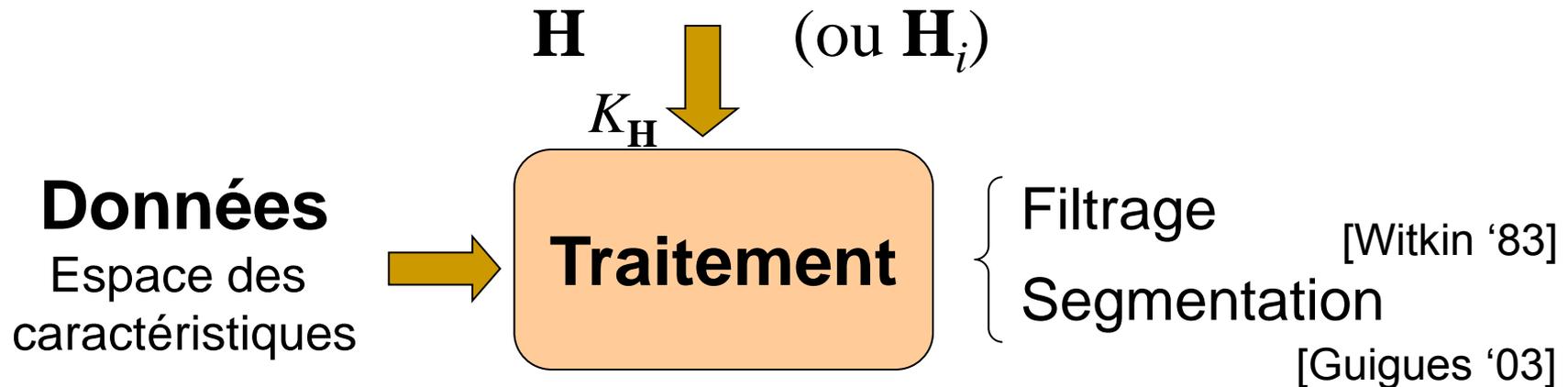
[Lindeberg 1999]



Yaron Ukrainitz & Bernard Sarel

# Intérêts pour les traitements

## Paramètres d'échelle (adaptatifs)



Choix de  
 $K$  et  $H$  ?

# Choix des paramètres d'échelle et du noyau

- Noyau et échelle optimaux
  - Calcul de l'Erreur Quadratique Moyenne Intégrée (EQMI)

$$EQMI(\hat{f}(\mathbf{x}; \mathbf{H})) = E \left[ \int_{-\infty}^{\infty} (\hat{f}(\mathbf{x}; \mathbf{H}) - f(\mathbf{x}))^2 . d\mathbf{x} \right]$$

- EQMI asymptotique ( $n$  très grand) et approximations
  - relations et hypothèses de convergence entre  $\mathbf{H}$  et  $n$
  - hypothèses sur  $K$
  - approximations de Taylor

➤ Solution:

$$\mathbf{H} = h^2 . \mathbf{I} \quad h_{opt} = \left[ \frac{d R(K)}{n \mu_2^2(K) \int (\nabla^2 f(\mathbf{x}))^2 d\mathbf{x}} \right]^{1/(d+4)}$$

$$R(K) = \int_{R^d} K^2(\mathbf{x}) . d\mathbf{x}$$
$$\mu_2(K) . \mathbf{I} = \int_{R^d} \mathbf{xx}^T . K(\mathbf{x}) d\mathbf{x}$$

dépend de  $K$

dépend de  $f$

Ajouter une hypothèse sur  $f$

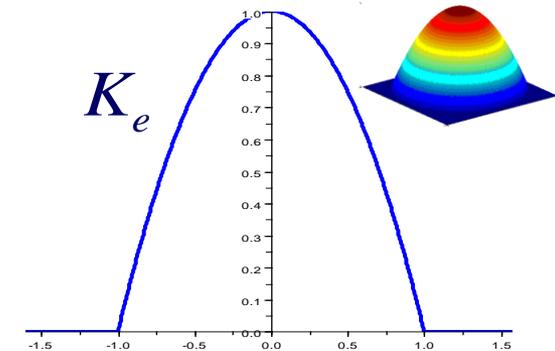
# Noyau et échelle optimaux

- Hypothèse :  $f$  gaussienne
  - Noyau optimal = **noyau d'Epanechnikov**

$$K_e(\mathbf{x}) = \begin{cases} \left(\frac{1}{2}\right) \cdot v_d^{-1} \cdot (d+2) \cdot (1 - \mathbf{x}^T \mathbf{x}) & \text{si } |\mathbf{x}^T \mathbf{x}| \leq 1 \\ 0 & \text{sinon} \end{cases}$$

- **Echelle optimale:**  $h_e = \left(8 \cdot v_d^{-1} \cdot n^{-1} \cdot (d+4) \cdot (2\sqrt{\pi})^d\right)^{\frac{1}{d+4}}$

- Pas d'hypothèse sur la densité  $f$ 
  - Solution dans le cas 1D



$v_d$  : volume hyper  
sphère unité

Choix d'un noyau  $K$

$$h_{opt} = \left[ \frac{R(K)}{n \cdot \mu_2^2(K) \cdot R(f'')} \right]^{1/5}$$

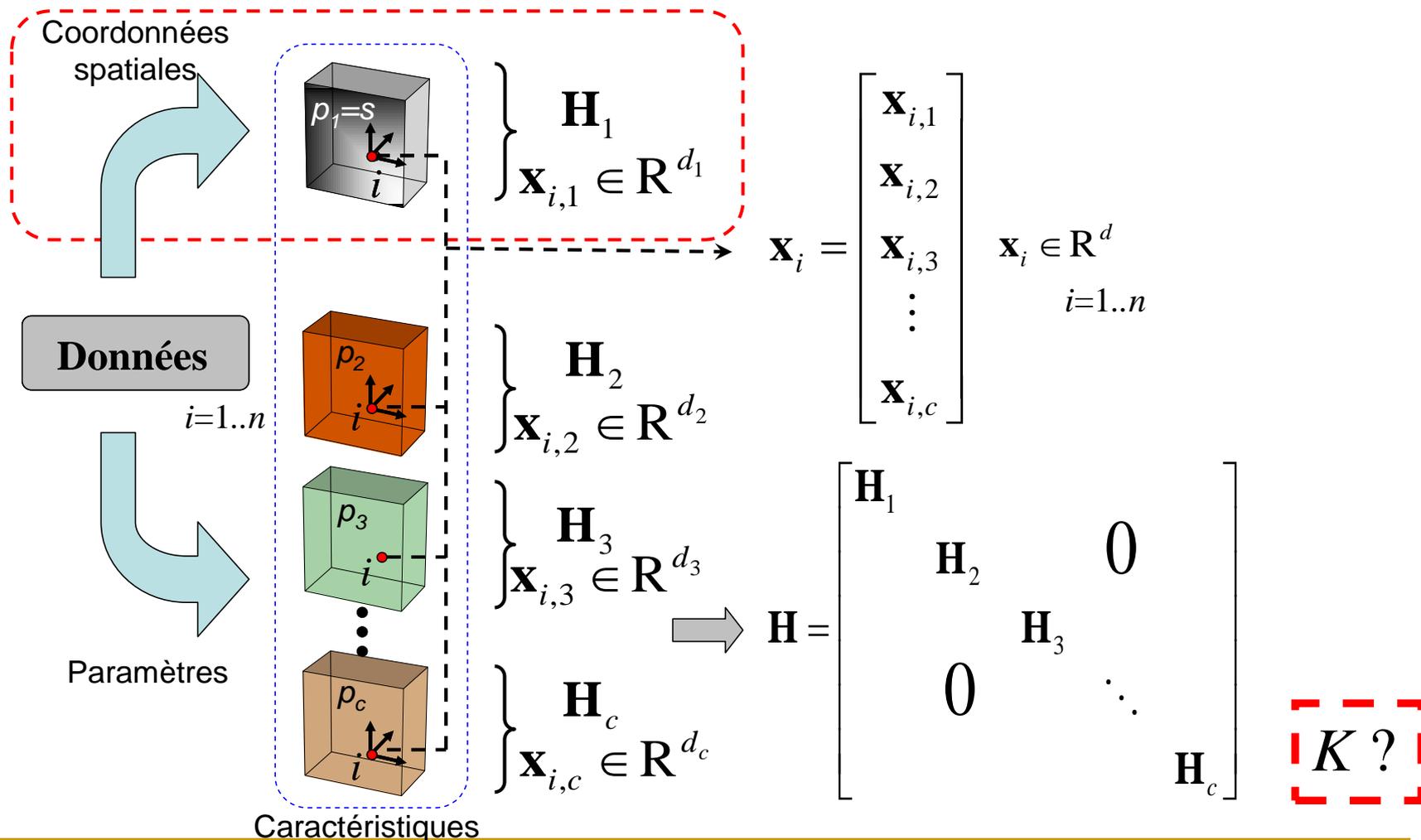
Estimations successives  
(estimation non paramétrique  
par noyau)



Méthode du *plug in* [Sheather '91]

# Paramètres d'échelle

## ■ Généralisation à l'espace des caractéristiques



# Noyau et paramètres d'échelle

- Noyau et échelle

$$K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2} \mathbf{x}) \quad \int_{\mathbb{R}^d} K(\mathbf{x}) d\mathbf{x} = 1$$

- Profil  $k$  du noyau sphérique  $K$

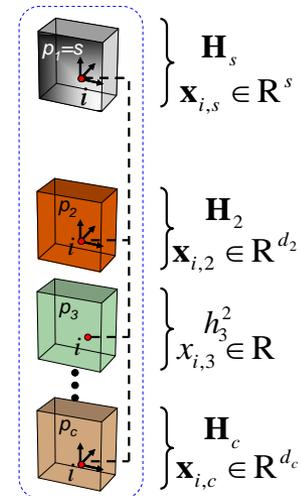
$$K(\mathbf{x}) = C_{k,d} \cdot k(\mathbf{x}^T \mathbf{x}) \quad k(u) \in \mathbb{R}$$

↑ Constante de normalisation

- Forme générale de  $K$  pour l'espace des caractéristiques

- produit de noyaux sphériques

$$K_{\mathbf{H}}(\mathbf{x}) = \left( \prod_{j=1}^c |\mathbf{H}_j| \right)^{-1/2} \cdot \prod_{j=1}^c C_{k_j, d_j} \cdot k_j(\mathbf{x}_j^T \mathbf{H}_j^{-1} \mathbf{x}_j)$$



# Définition et notions : Bilan

- Représentation des données d'entrée dans l'espace des caractéristiques (nuage de points)
- Définition de l'estimation de densité par noyau dans l'espace des caractéristiques:  $\hat{f}(\mathbf{x}; \mathbf{H})$
- Cette estimation dépend des paramètres d'échelle  $\mathbf{H}$  et du noyau  $K$  : étude du choix optimal  
(au sens du critère de l'EQMI)
  - Mise en œuvre en filtrage et segmentation d'images

# Principe Mean Shift

- **État de l'art**
  - Historique
  - Application
  
- **Fonctionnement**
  - Lien avec l'estimation non paramétrique
  - Principe de convergence
  - Illustrations

# Présentation du *mean shift*

## ■ Définition:

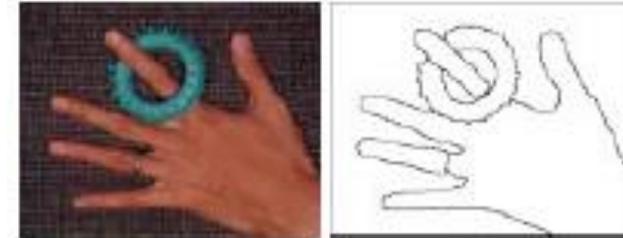
*estimateur non paramétrique du gradient de la densité de probabilité*

## ■ Premières références

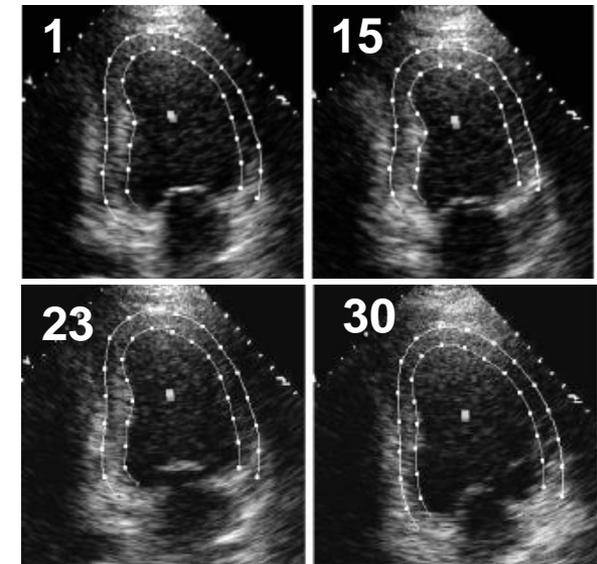
- ❑ inventé par Fukunaga (reconnaissance de forme) [Fukunaga '75]
- ❑ généralisation du formalisme par Cheng [Cheng '95]
- ❑ application au filtrage et à la segmentation d'image [Comaniciu '97] [Comaniciu '99]

# Présentation du *mean shift*

- Etat de l'art (surtout applications)
  - Filtrage et segmentation d'image
    - Images couleurs [Comaniciu '99]
    - Imagerie SAR [Cellier '05]
  - Suivi d'objet sur séquence vidéo
    - Suivi de joueur [Jaffre '03]
    - Suivi mouvement cardiaque en imagerie échographique [Zhou '05]
  - Maillage
    - Segmentation [Yamauchi '05]
    - Mean shift géodésique [Shamir '04]
- Grande variété d'applications :  
exploitation de l'espace des caractéristiques



[Comaniciu '99]

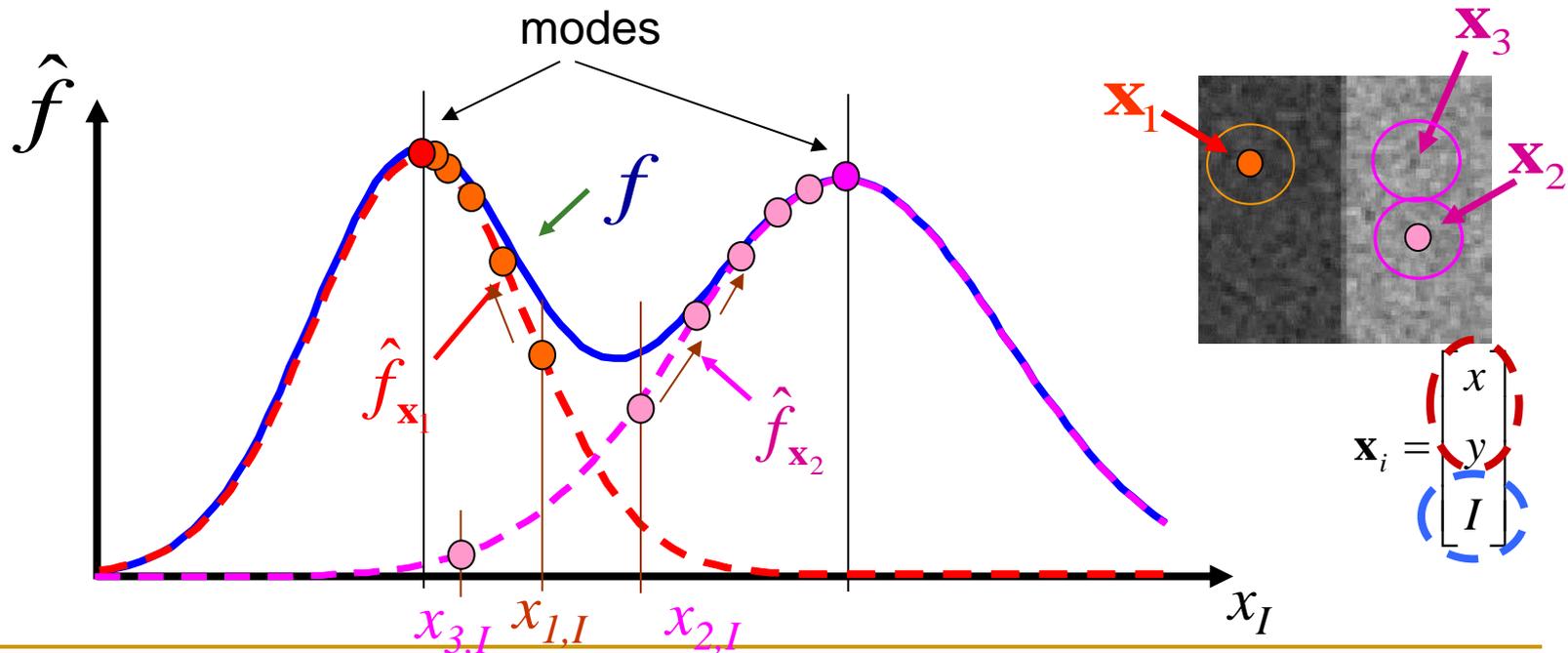


[Zhou '05]

# Principe : recherche des modes

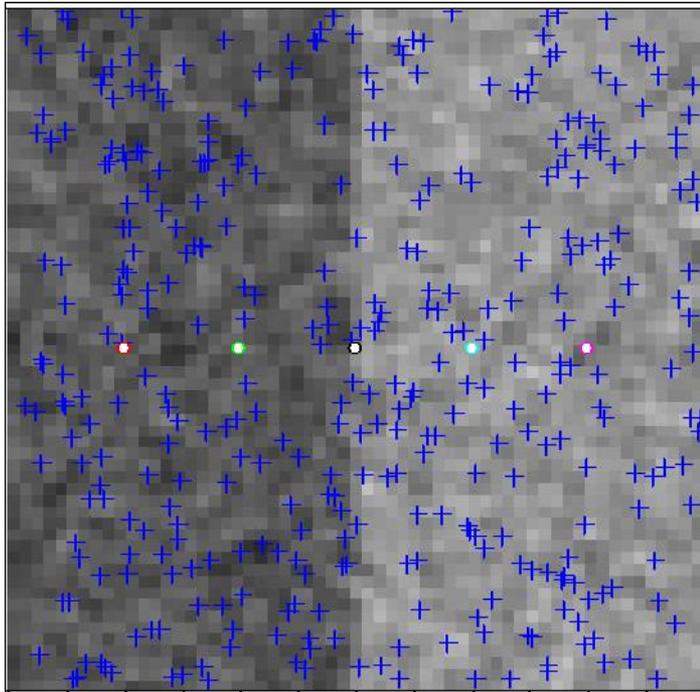
- Mode = *maximum d'une densité de probabilité*
- Un mode caractérise une densité de probabilité
- Une région est caractérisée par une densité de probabilité (plusieurs régions  $\Leftrightarrow$  plusieurs modes)
- Trouver à quel mode appartient une donnée  $\Leftrightarrow$  trouver à quelle région appartient la donnée

$$\nabla \hat{f}(\mathbf{x}) = \vec{0}$$

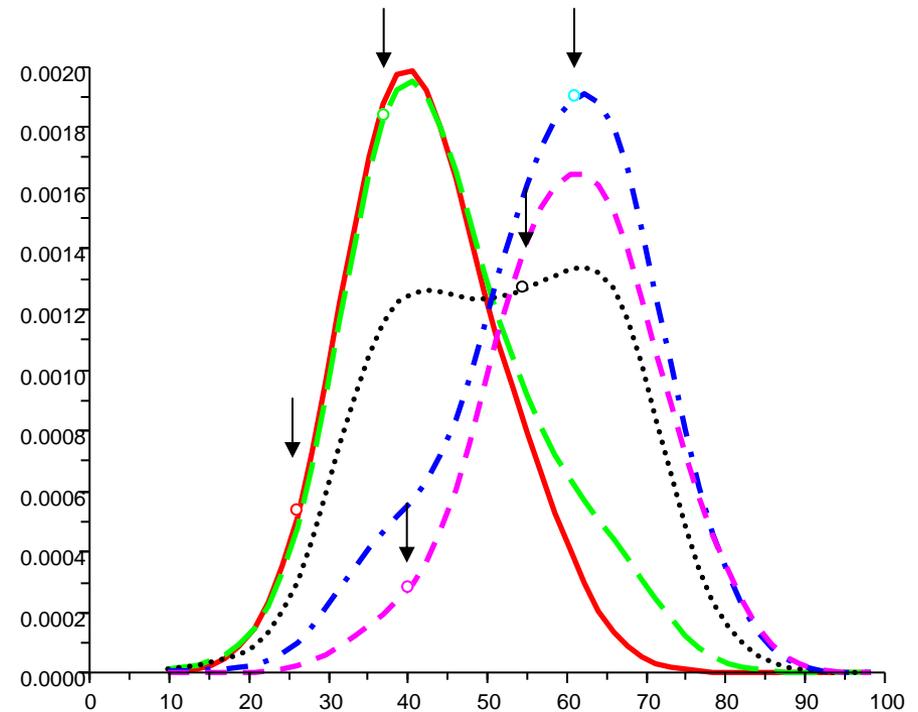


# Principe: recherche des modes

- Estimations locales,
  - basées sur un échantillon aléatoire



Image



Estimations de la pdf

# Principe : recherche des modes

- Gradient de l'estimation  $\hat{\nabla}f(x) \equiv \nabla\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \nabla K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)$

$$\nabla\hat{f}(\mathbf{x}) = \vec{0} \quad ?$$

Solution  
simple  
(littérature)



$$\mathbf{x} = \frac{\sum_{i=1}^n g(d^2(\mathbf{x}, \mathbf{x}_i, \mathbf{H})) \cdot \mathbf{x}_i}{\sum_{i=1}^n g(d^2(\mathbf{x}, \mathbf{x}_i, \mathbf{H}))}$$

Solution  
générale  
(proposition)



$$\mathbf{x}_j = \frac{\sum_{i=1}^n N(\mathbf{x}_j, \mathbf{x}_{i,j}, k_{\mathbf{H}}) \cdot \mathbf{x}_{i,j}}{\sum_{i=1}^n N(\mathbf{x}_j, \mathbf{x}_{i,j}, k_{\mathbf{H}})}$$

$j=1..c$

avec:

$$N(\mathbf{x}_j, \mathbf{x}_{i,j}, k_{\mathbf{H}}) = k_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i) \cdot \frac{g_j(d^2(.))}{k_j(d^2(.))}$$

$$k_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i) = \prod_{j=1}^c k_j(d^2(.))$$

$$g(u) = -k'(u)$$

$$d^2(.) = (\mathbf{x}_j - \mathbf{x}_{i,j})^T \mathbf{H}_j^{-1} (\mathbf{x}_j - \mathbf{x}_{i,j}) \quad : \text{ distance de Mahalanobis (au carré) de la caractéristique } j$$

# Principe : recherche des modes

## ■ Calcul itératif des racines du gradient

$$\nabla \hat{f}(\mathbf{x}) = \vec{0} \quad ?$$

Solution  
simple  
(littérature)



$$\mathbf{x}^{[0]} = \mathbf{x}$$

$$\mathbf{x}^{[t+1]} = \frac{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H})) \mathbf{x}_i}{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H}))}$$

Solution  
générale



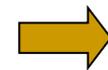
Pour chaque caractéristique  $j$

$$\mathbf{x}_j^{[t+1]} = \frac{\sum_{i=1}^n N(\mathbf{x}_j^{[t]}, \mathbf{x}_{i,j}, k_{\mathbf{H}}) \mathbf{x}_{i,j}}{\sum_{i=1}^n N(\mathbf{x}_j^{[t]}, \mathbf{x}_{i,j}, k_{\mathbf{H}})}$$

$j=1..c$

➤ Remarque:  
toutes les caractéristiques  
de  $\mathbf{x}$  évoluent

$$\mathbf{x}^{[t]} = \begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix}$$



$$\mathbf{x}^{[t+1]} = \begin{bmatrix} x' \\ y' \\ r' \\ g' \\ b' \end{bmatrix}$$

spatial

amplitude

# Procédures *mean shift*

- Deux procédures pour le calcul itératif : [Cheng '95]
  - *blurring* [Fukunaga '75]
  - *nonblurring* exploité par Comaniciu [Comaniciu '99]  
(convergence montrée par les auteurs pour les deux procédures)
  
- Procédure ***nonblurring*** de Comaniciu
  - $\{\mathbf{x}_i\}_{i=1..n}$  : ensemble des données d'entrée (dans  $\mathbb{R}^d$ )
  - $\{\mathbf{y}_i\}_{i=1..m}$  : ensemble de points de  $\mathbb{R}^d$
  - Pour chaque  $\mathbf{y}_i$ 
    - $\mathbf{y}_i \rightarrow \mathbf{y}^{[0]}$
    - Calculer :  $\mathbf{y}^{[t+1]} \rightarrow \mathbf{y}_i^{\text{mode}}$

$$\mathbf{y}^{[t+1]} = \frac{\sum_{i=1}^n g(d^2(\mathbf{y}^{[t]}, \mathbf{x}_i, \mathbf{H})) \mathbf{x}_i}{\sum_{i=1}^n g(d^2(\mathbf{y}^{[t]}, \mathbf{x}_i, \mathbf{H}))}$$

jusqu'à convergence  
 $\mathbf{y}^{[t+1]} \approx \mathbf{y}^{[t]}$

# Procédure *mean shift* nonblurring Principe Mean Shift

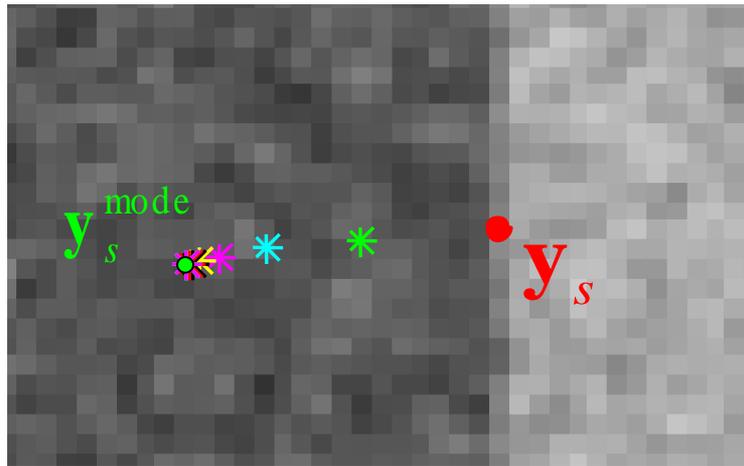
## ■ Illustration

Recherche du mode

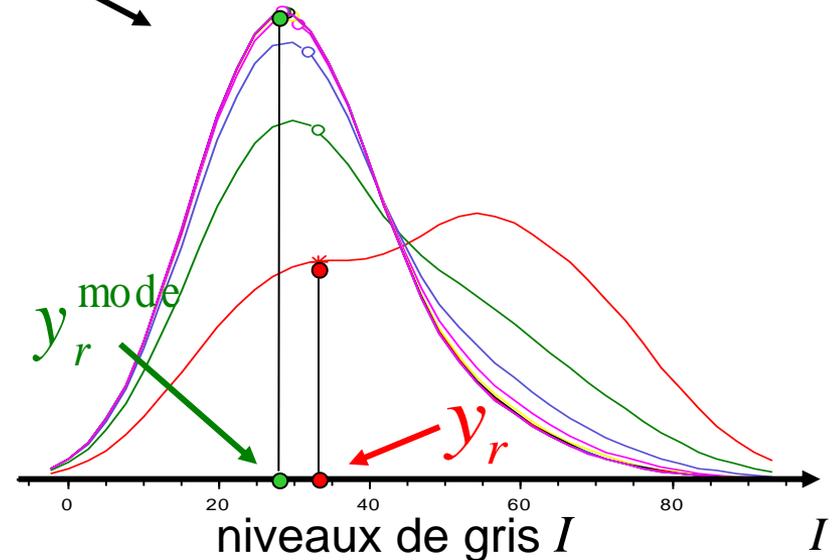
$$\mathbf{y} = \begin{bmatrix} x \\ y \\ I \end{bmatrix} \Rightarrow \mathbf{y}^{\text{mode}} = \begin{bmatrix} x' \\ y' \\ I' \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_s \\ \mathbf{y}_r \end{bmatrix} \Rightarrow \begin{cases} \mathbf{y}_s^{\text{mode}} = \begin{bmatrix} x' \\ y' \end{bmatrix} \\ \mathbf{y}_r^{\text{mode}} = I' \end{cases}$$

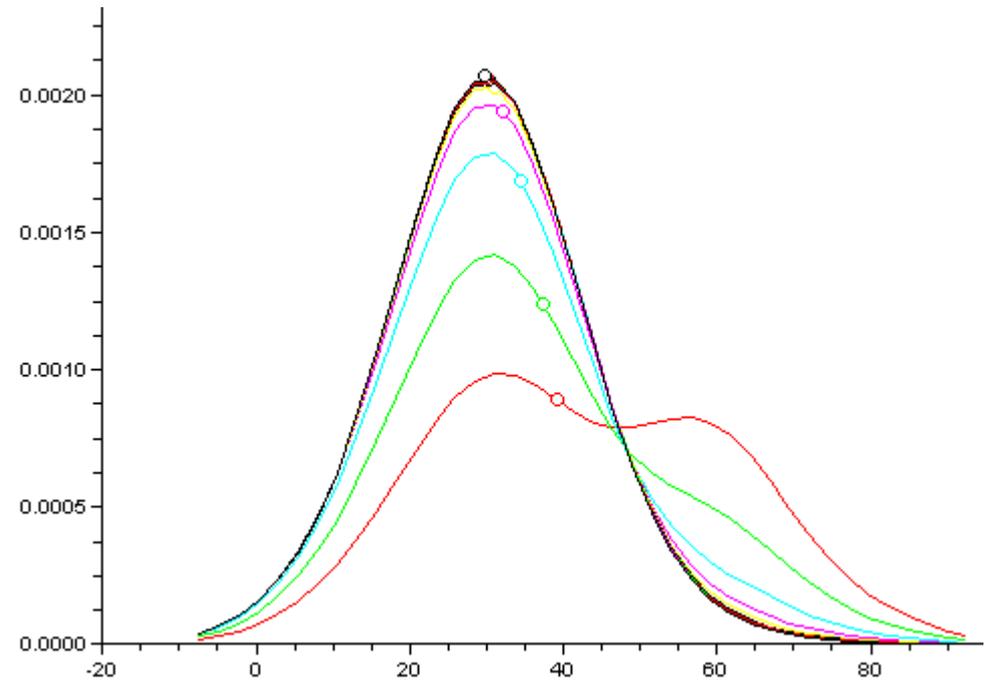
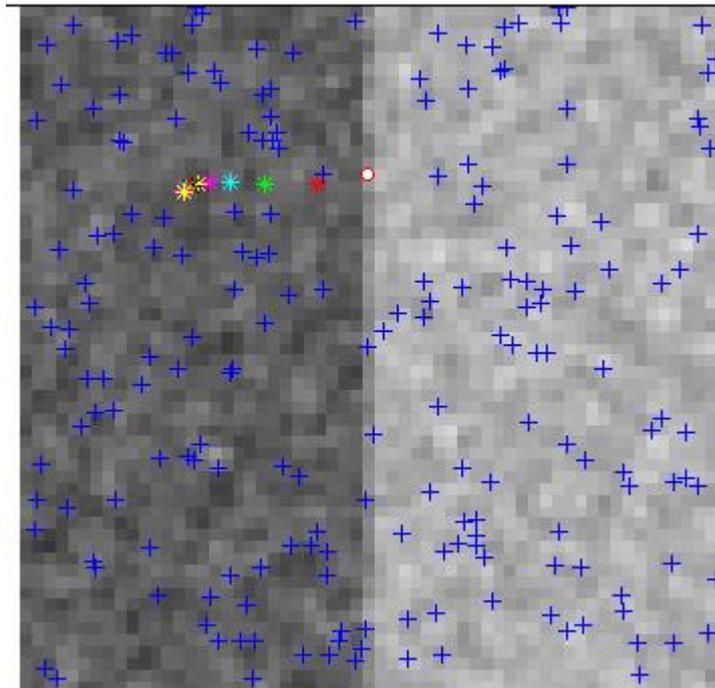
Noyaux gaussiens, paramètres d'échelle optimaux



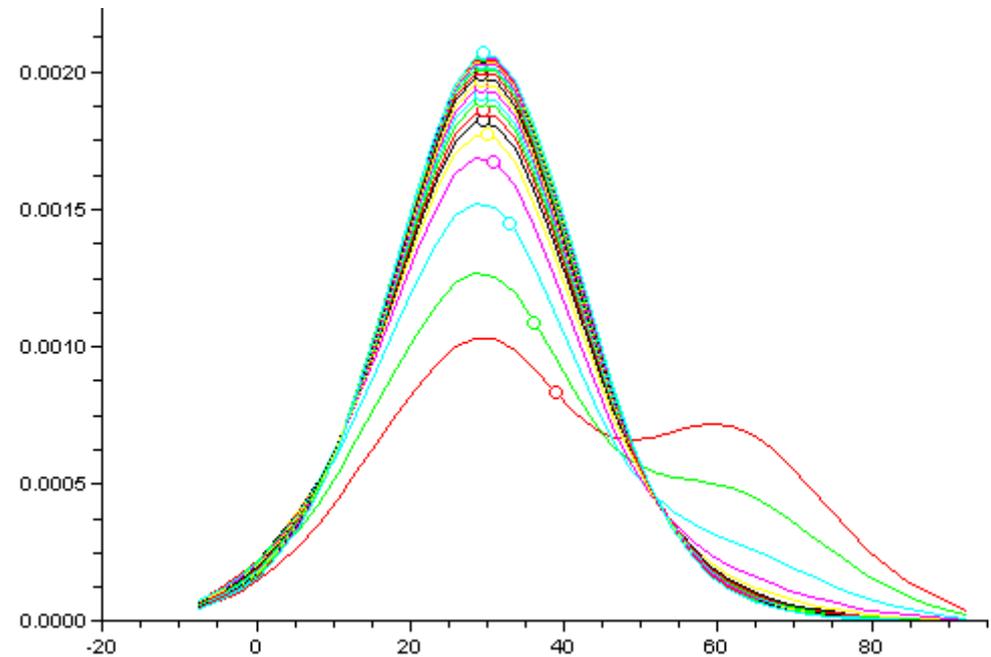
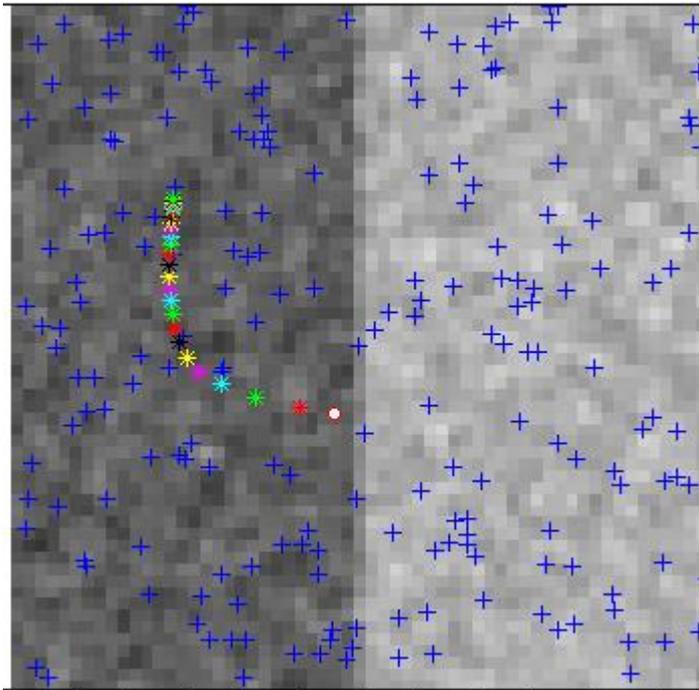
spatial  $x$  et  $y$



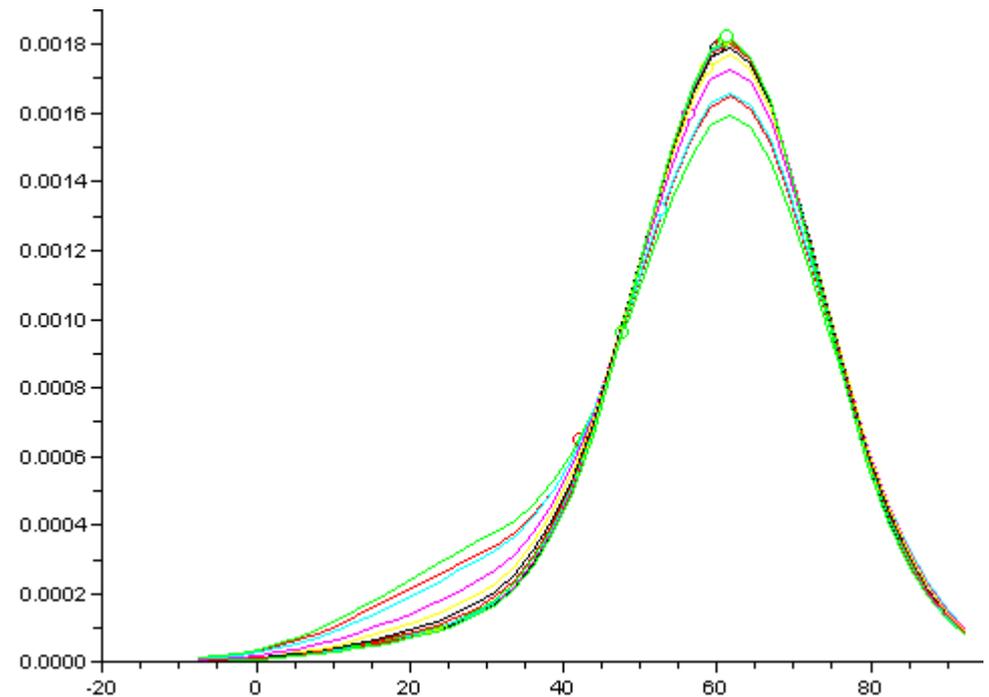
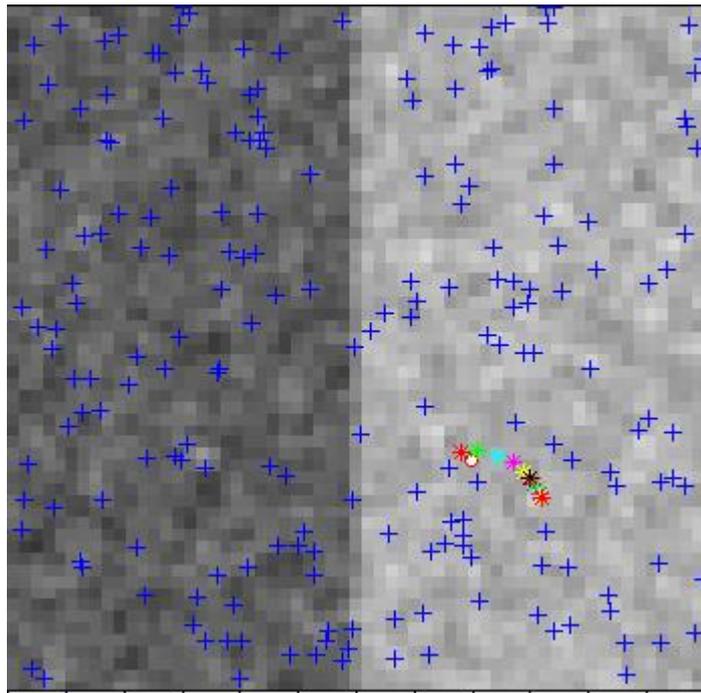
# Procédure *mean Shift* nonblurring Principe Mean Shift



# Procédure *mean Shift* nonblurring Principe Mean Shift



# Procédure *mean Shift* nonblurring Principe Mean Shift



# Procédures *mean shift*

- Deux procédures pour le calcul itératif : [Cheng '95]
  - *blurring* [Fukunaga '75]
  - *nonblurring* exploité par Comaniciu [Comaniciu '99]  
(convergence montrée par les auteurs pour les deux procédures)

- Procédure ***blurring*** de Fukunaga

- $\{\mathbf{x}_i\}_{i=1..n}$  : ensemble des données d'entrée (dans  $\mathbb{R}^d$ )
- Jusqu'à convergence  $\mathbf{x}_i^{[t+1]} \approx \mathbf{x}_i^{[t]}, \quad \forall i$ 
  - Calculer les  $\mathbf{x}^{[t+1]}$  à partir des  $\mathbf{x}^{[t]}$

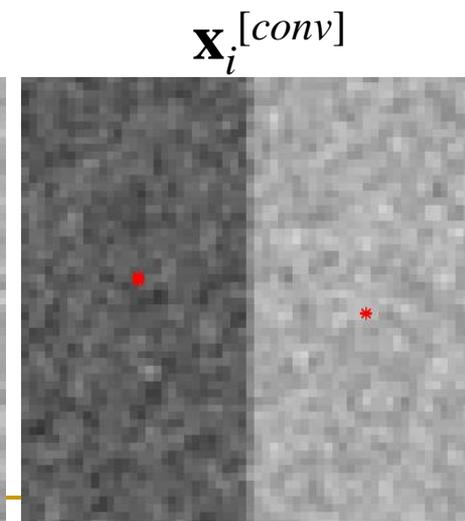
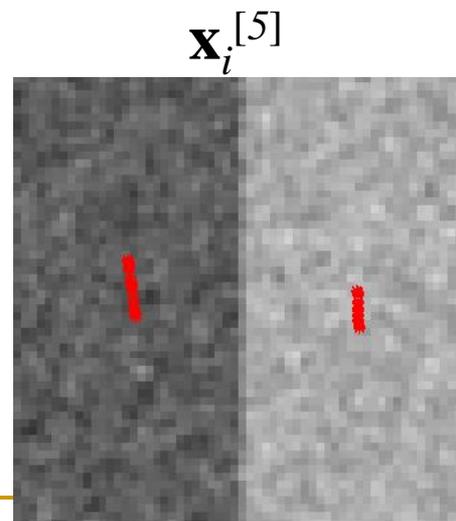
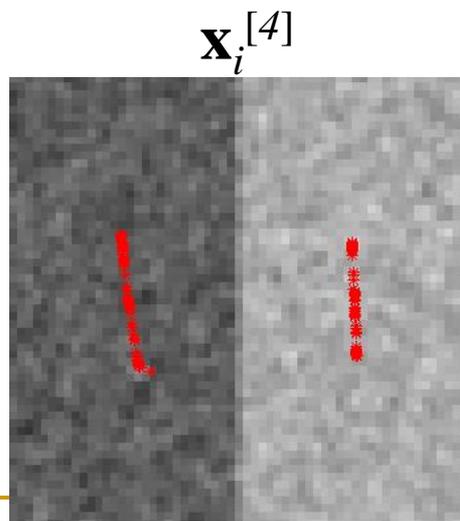
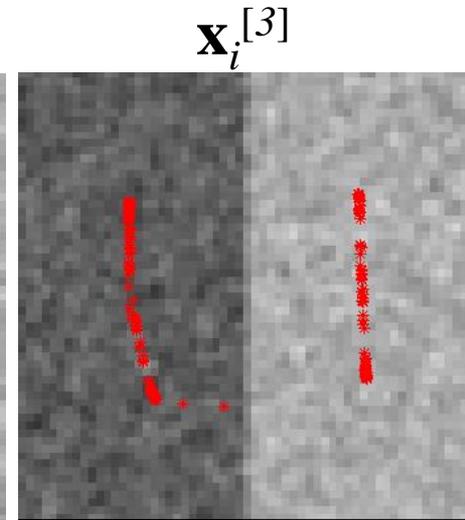
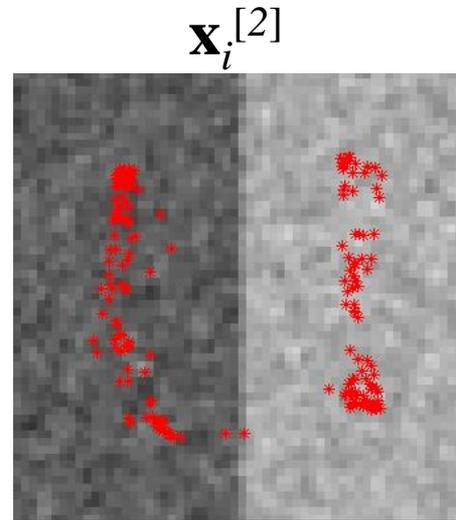
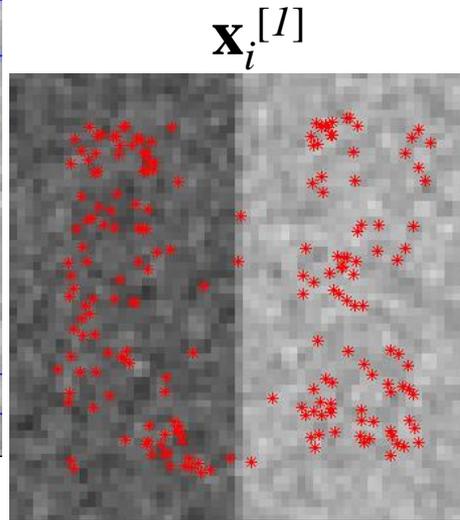
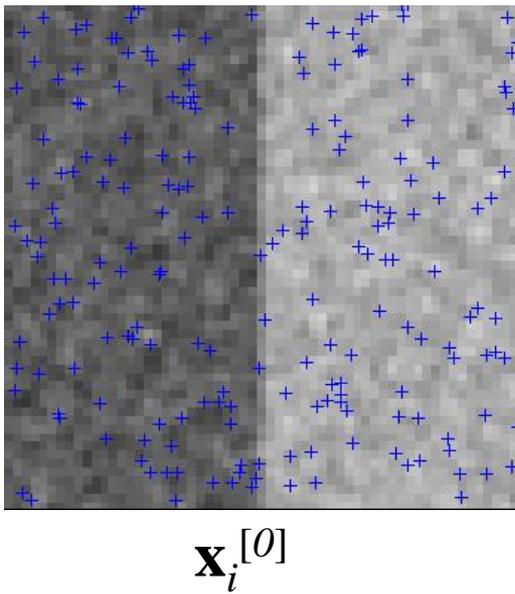
$$\mathbf{x}^{[t+1]} = \frac{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i^{[t]}, \mathbf{H})) \mathbf{x}_i^{[t]}}{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i^{[t]}, \mathbf{H}))}$$

plus dur...

→ La position des  $\mathbf{x}_i$  évolue

# Procédure *mean Shift* blurring

*Principe Mean Shift*

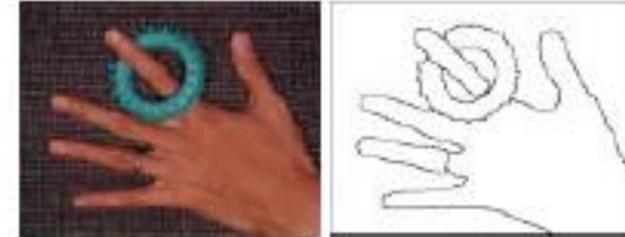


# III – Applications du Mean Shift

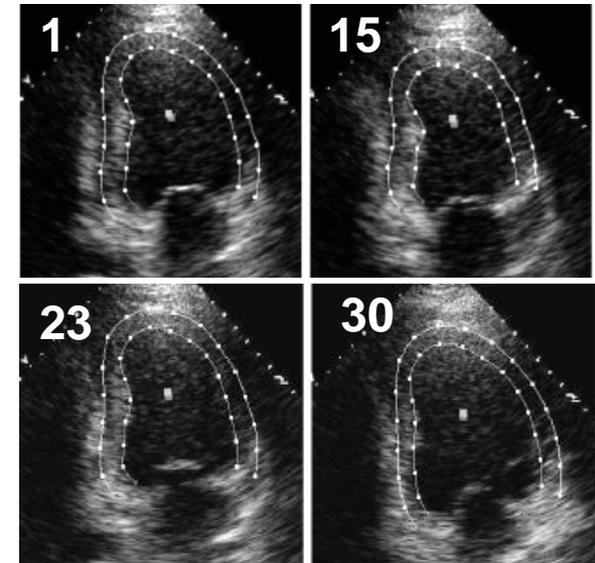
*Filtrage*  
*Segmentation*  
*Suivi vidéo*  
*A priori*

# Application du *mean shift*

- Grande variété d'application:  
Utilisation de l'espace des caractéristiques
  - Filtrage et segmentation d'image
    - Images couleurs [Comaniciu '99]
    - Imagerie SAR [Cellier '05]
  - Suivi d'objet sur séquence vidéo
    - Suivi de joueur [Jaffre '03]
    - Suivi mouvement cardiaque en imagerie échographique [Zhou '05]
  - Maillage
    - Segmentation [Yamauchi '05]
    - Mean shift géodésique [Shamir '04]



[Comaniciu '99]



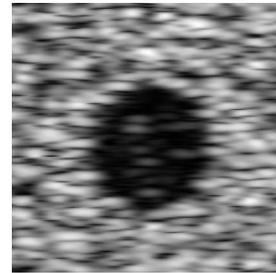
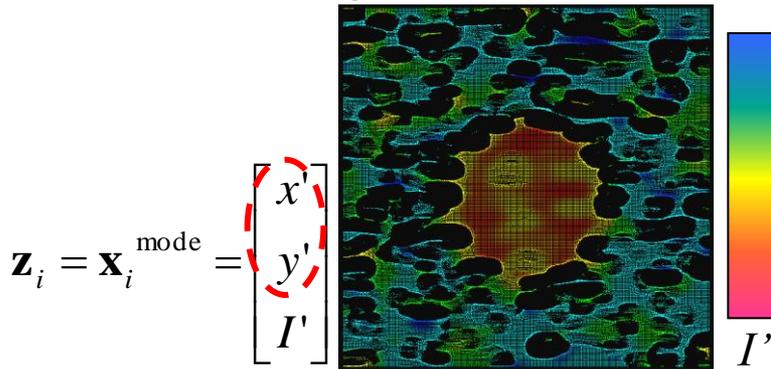
[Zhou '05]

# Algorithmes de filtrage *mean shift*

- Filtrage : *associer à chaque  $\mathbf{x}_i$  son mode  $\mathbf{x}_i^{\text{mode}}$*   
on notera  $\{\mathbf{z}_i\}$  : l'ensemble des points filtrés

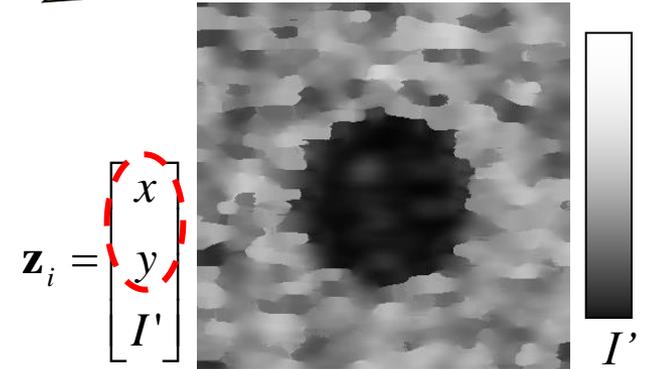
- Deux algorithmes

Filtrage Libre (MSL)



$$\mathbf{x}_i = \begin{bmatrix} x \\ y \\ I \end{bmatrix}$$

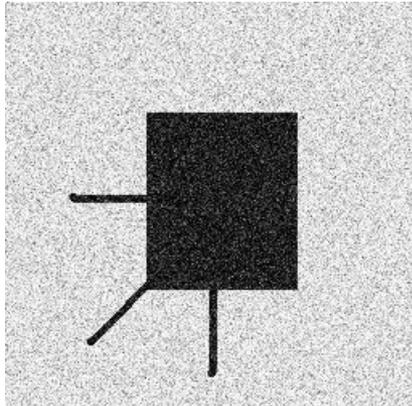
Filtrage Contraint (MSC)



- Variante : paramètres d'échelle adaptatifs

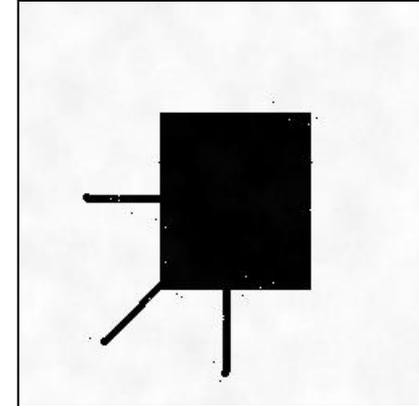
[Comaniciu '01][Grenier '05]

# Filtrage MS image synthétisée



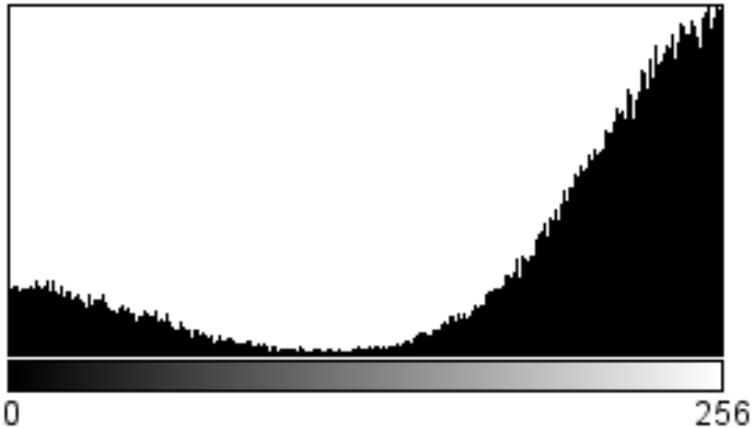
Original

Mean Shift  
 $h_s=10, h_r=70$

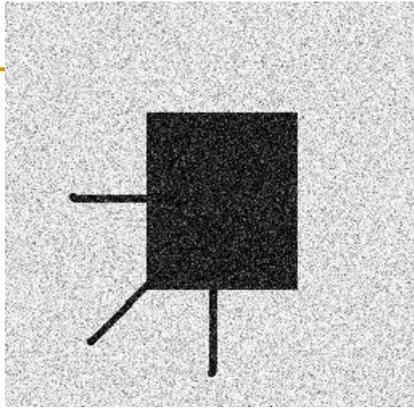


Filtrage Mean Shift Contraint

Histogrammes

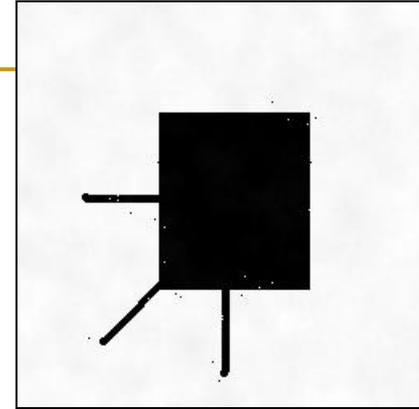


# Filtrage MS image synthétisée



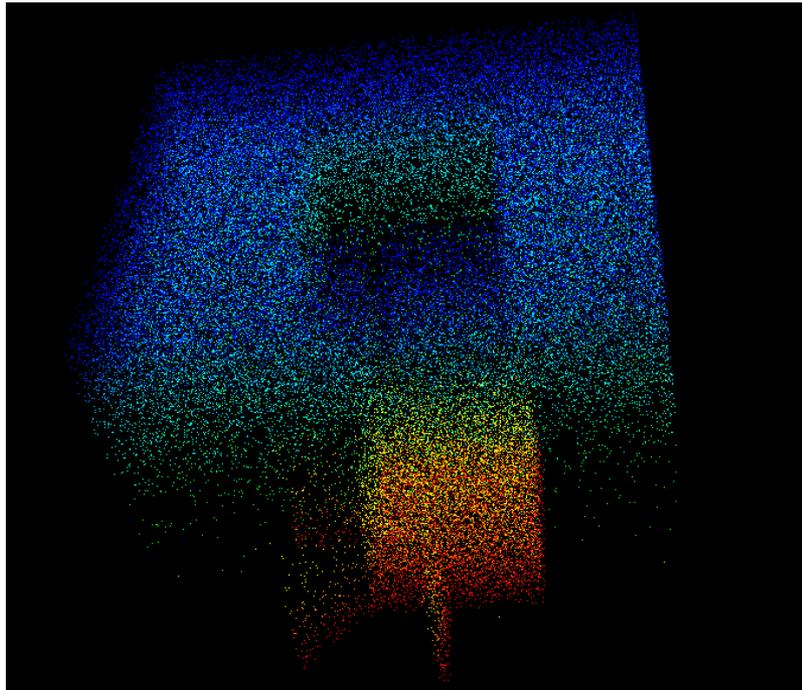
Original

Mean Shift  
→  
 $h_s=10, h_r=70$

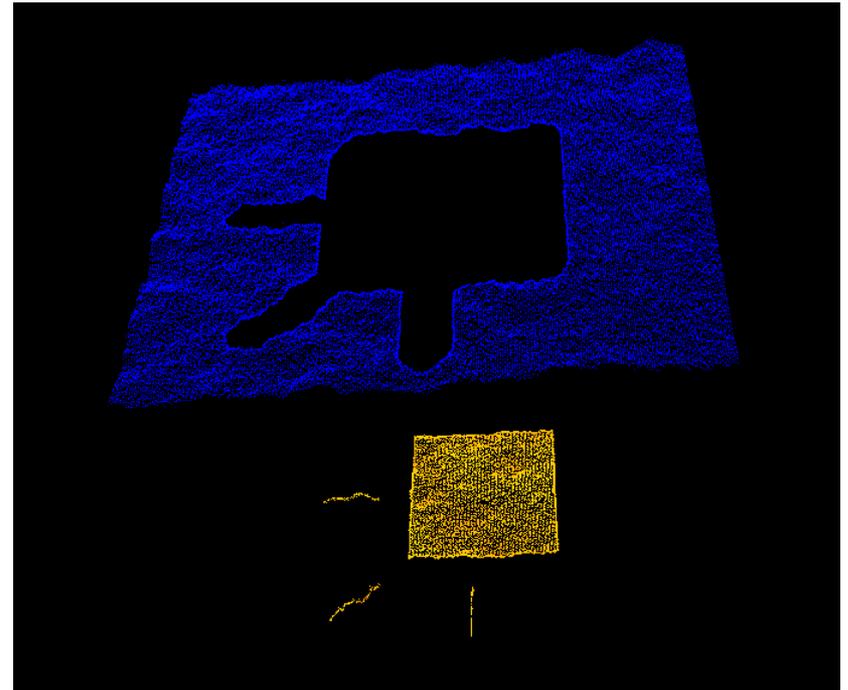


Mean Shift  
Contraint

Mean Shift Libre



3D



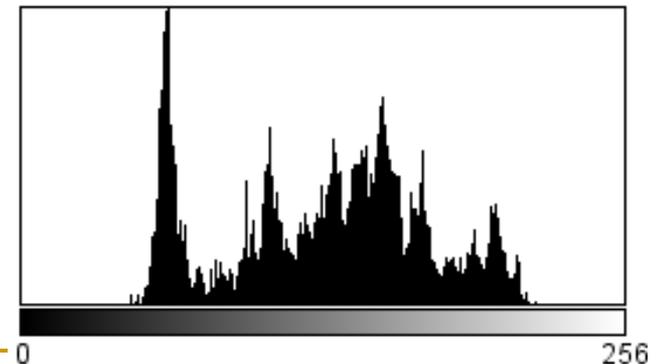
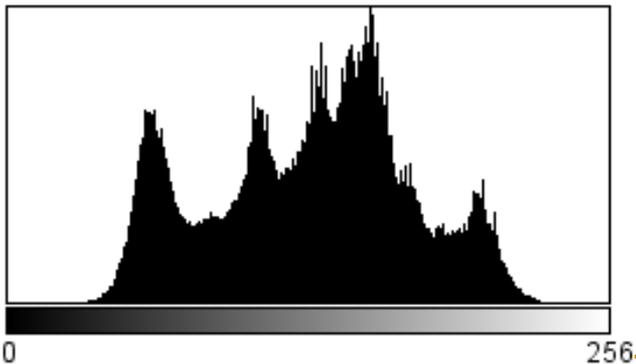
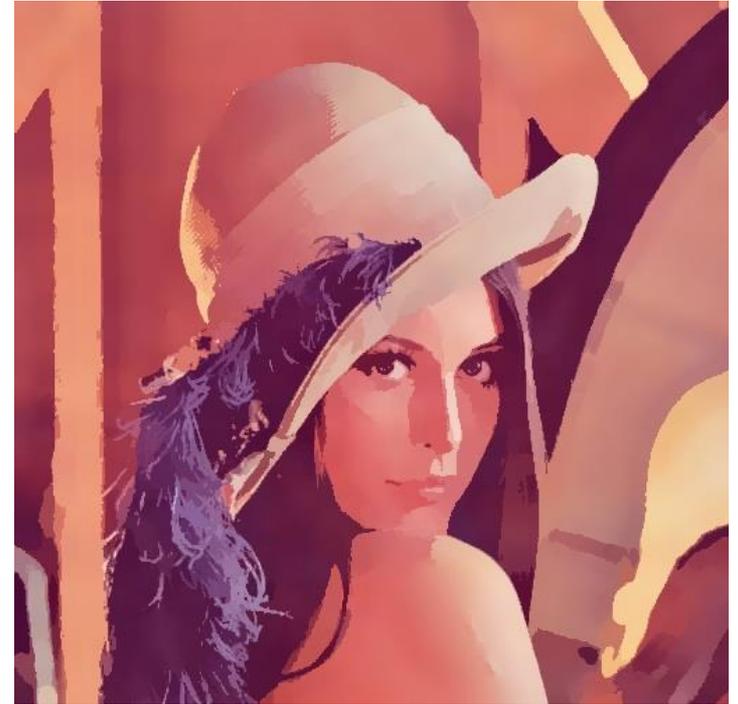
3D

# Filtrage MS image RGB (Léna)

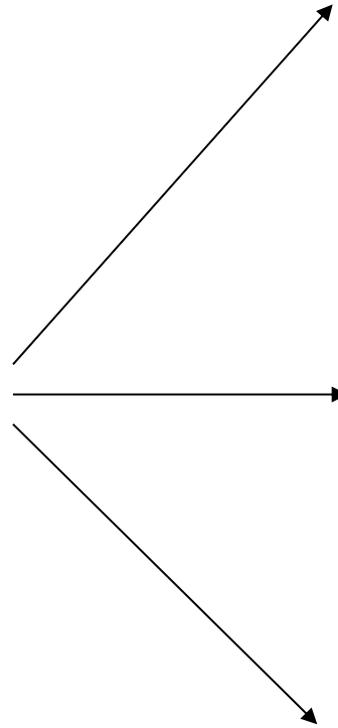


MSC

$h_s=10, h_r=30$



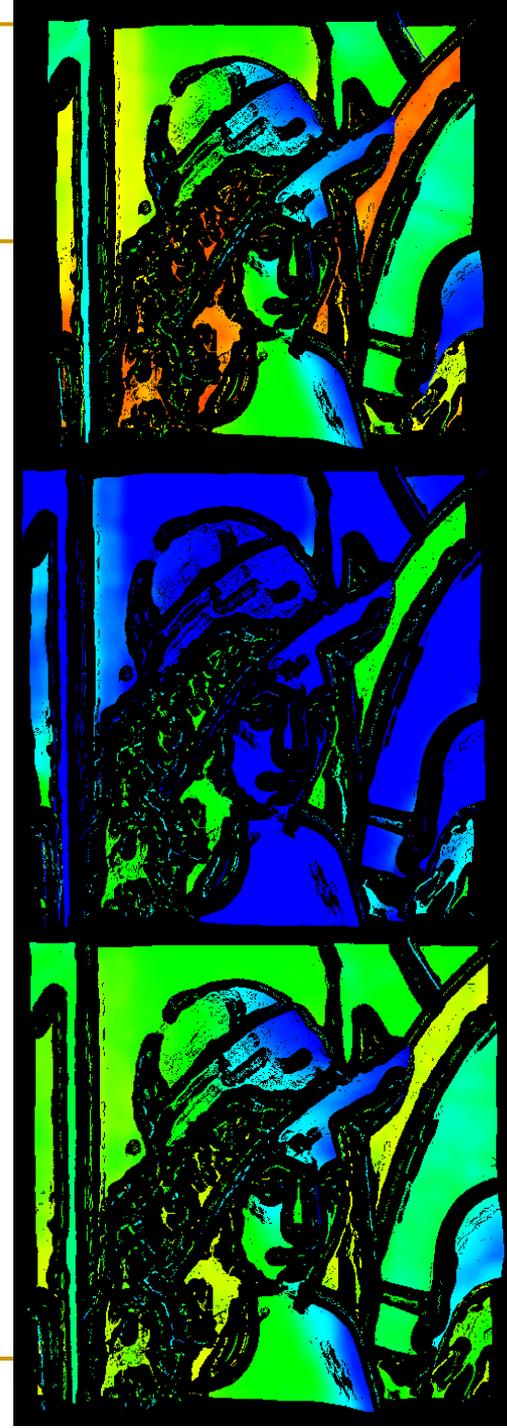
# Filtrage MS image RGB



**MSL**

$h_s=10, h_r=30$

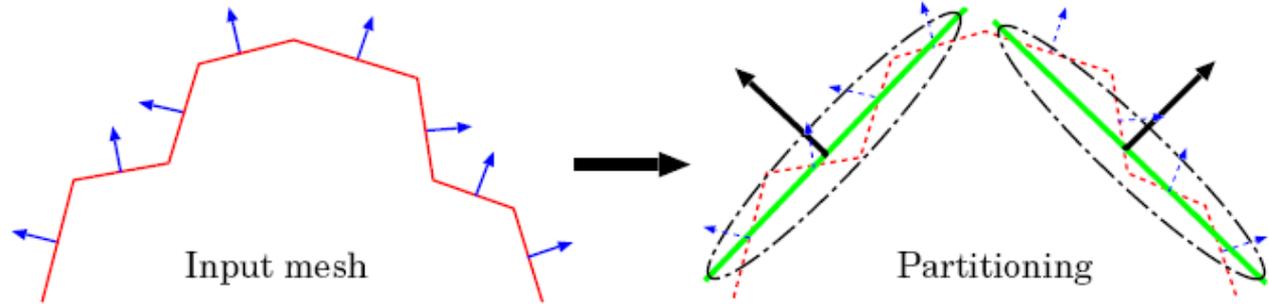
5 Dimensions



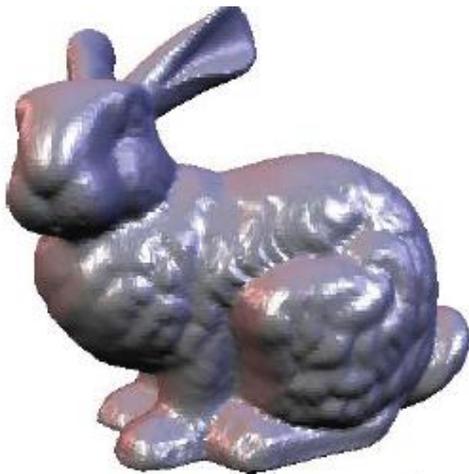
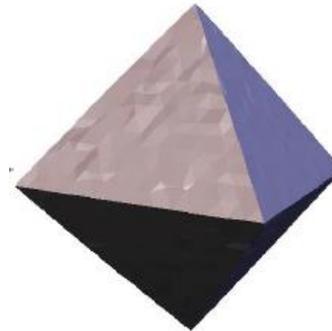
# Filtrage de ...

## ■ Maillage

[Yamauchi 2005]



Octaèdre bruité:



# Segmentation Mean Shift ?

- Pas de segmentation Mean Shift

mais des segmentations basées sur des prétraitements « Mean Shift »

*prétraitements* : filtrage, estimation de paramètres...

- Exemples de processus de *segmentation*:

- Filtrage → croissance de région (multidimensionnelle)
- Estimation de paramètres (i.e. d'un modèle géométrique) → mise en correspondance du modèle

# Segmentation après filtrage Mean shift

- Problème : segmenter les données dans l'espace des caractéristiques
  - Seuillage
  - Méthodes *de Clustering*
  - Croissance de région [Comaniciu 99]
  - ...
- Reste à définir la croissance de région pour l'espace des caractéristiques...

# Croissance de Région

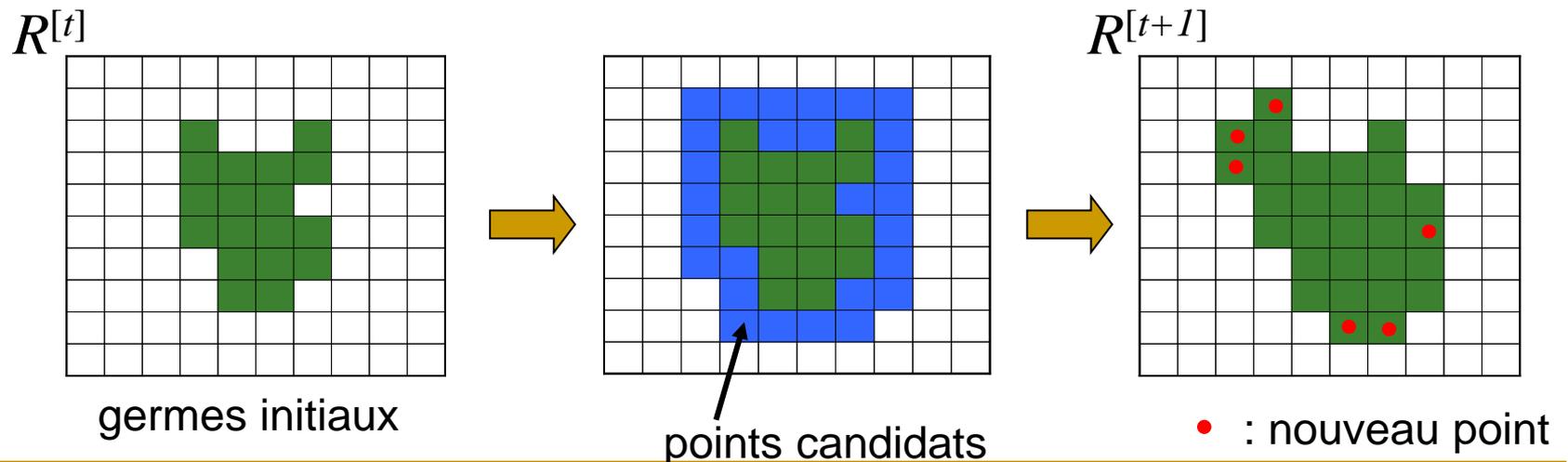
## ■ Présentation / rappel

□ Introduite par Zucker

[Zucker '76]

□ Principe

- désignation de germes initiaux
- agglomération des points situés à la **périphérie** de cette région respectant les conditions d'agglomération

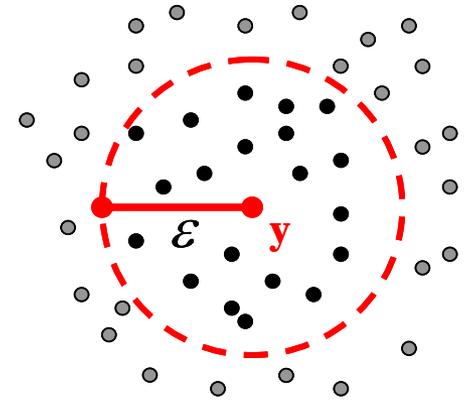


# Croissance de région multidimensionnelle

- Comment généraliser le principe de voisinage ?

- Définition d'un voisinage:

$$V(\mathbf{y}; \varepsilon) = \{\mathbf{x} \in X \mid d(\mathbf{x}, \mathbf{y}) \leq \varepsilon\}$$



- outils pour la distance:  
noyaux et échelles

- Matrices de paramètres d'échelle

- Adaptatives ou non

- $K$ : produit de noyaux

- $K(\cdot)$  fonction de n'importe quelle variable

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{H}_c \end{bmatrix}$$

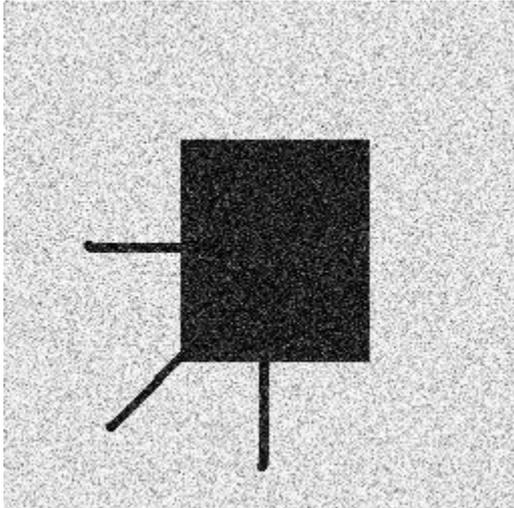
$$K_{\mathbf{H}}(\mathbf{x}) = \prod_{j=1}^c K_{\mathbf{H}_j}(\mathbf{x}_j)$$

# Croissance de région

## ■ Algorithme

- Choisir un germe dans l'espace des caractéristiques
  - À partir de ce germe regrouper itérativement les points respectant les critères d'homogénéité
- Quand on ne peut plus ajouter de nouveaux points:
  - choisir un nouveau germe parmi les points non regroupés dans une région et définir une nouvelle région.
- Répéter tant qu'il reste des points non affectés à une région
- *(supprimer les régions de faibles effectifs)*

# Exemple de croissance de Région



Original

Logiciel EDISON

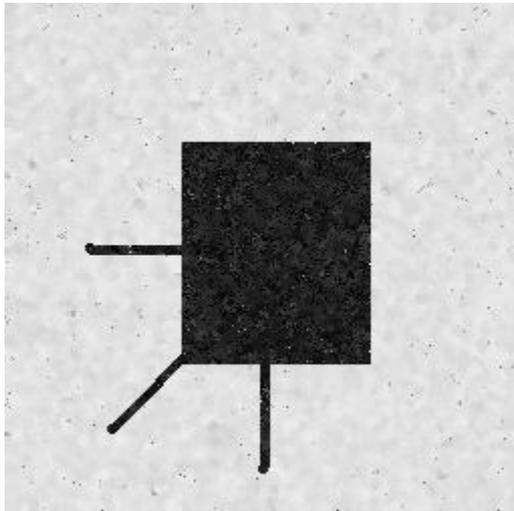
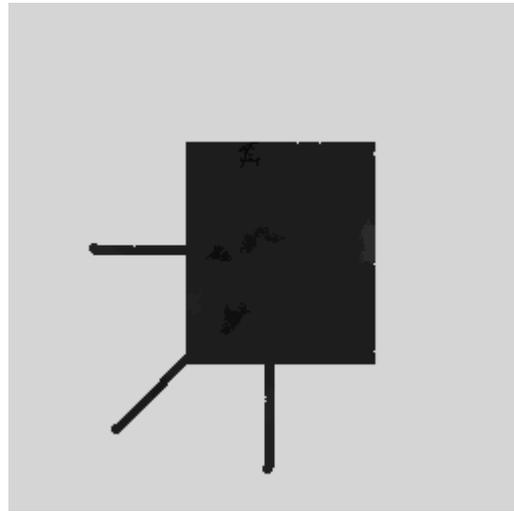
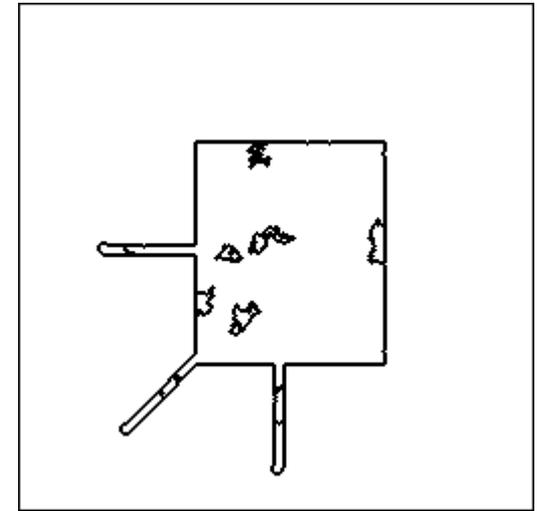


Image filtrée (MS\*)



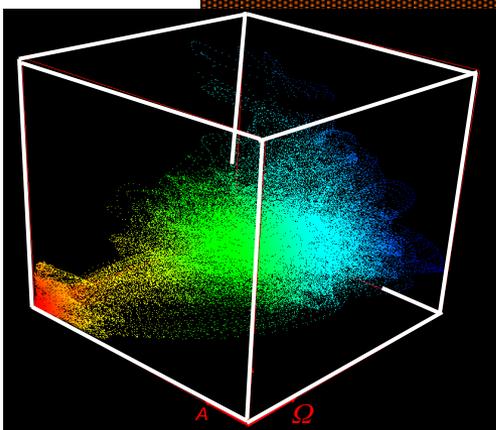
Segmentation



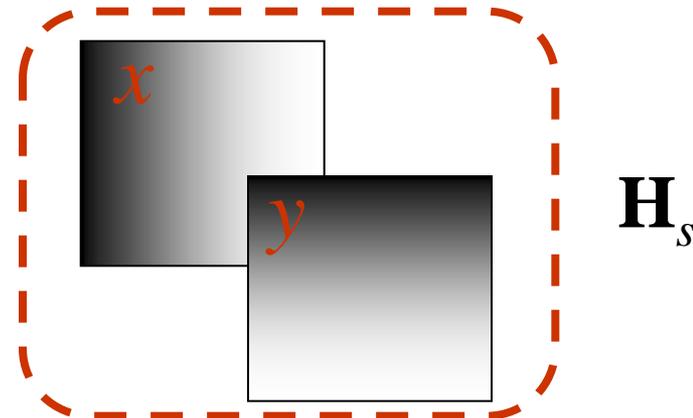
Frontières

# Croissance de région

initiale

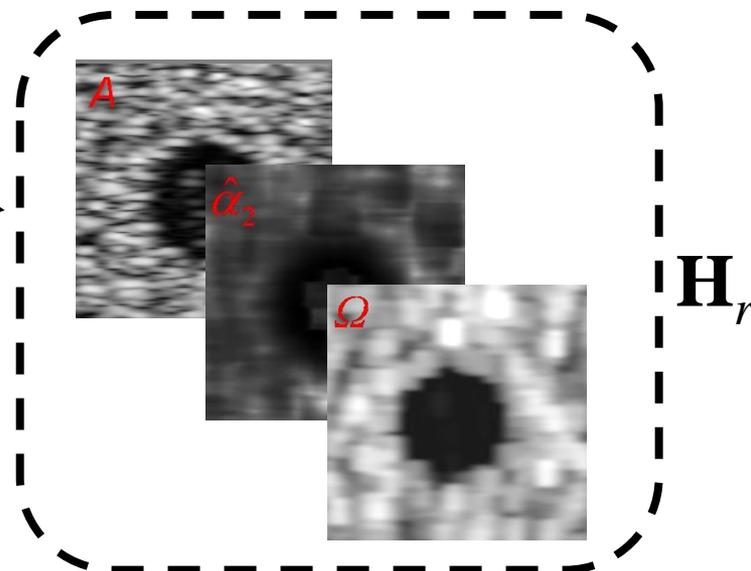


caractéristiques spatiales



$$\mathbf{z}_i = \begin{bmatrix} x \\ y \\ A \\ \hat{\alpha}_2 \\ \Omega \end{bmatrix}$$

caractéristiques des amplitudes



$K$  : Epanechnikov

$\mathbf{H}_s = 10^2 \cdot \mathbf{I}$  et  $\mathbf{H}_r = 70^2 \cdot \mathbf{I}$  :

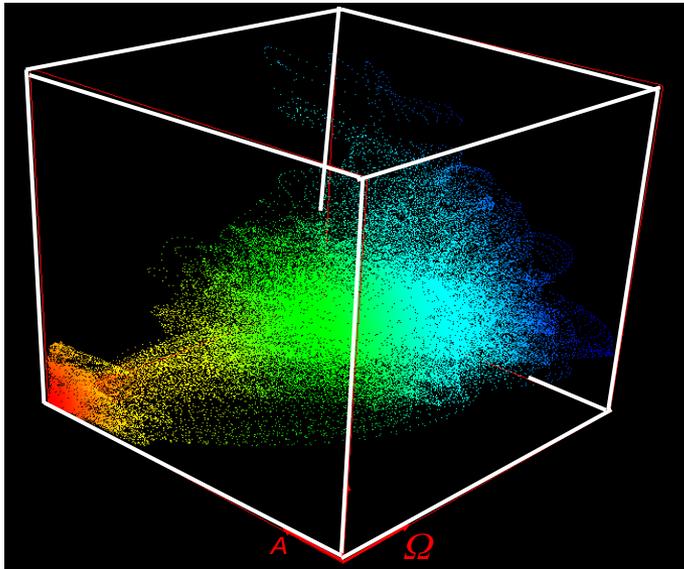
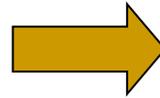
permettent d'obtenir un échantillon représentatif

# Croissance de région

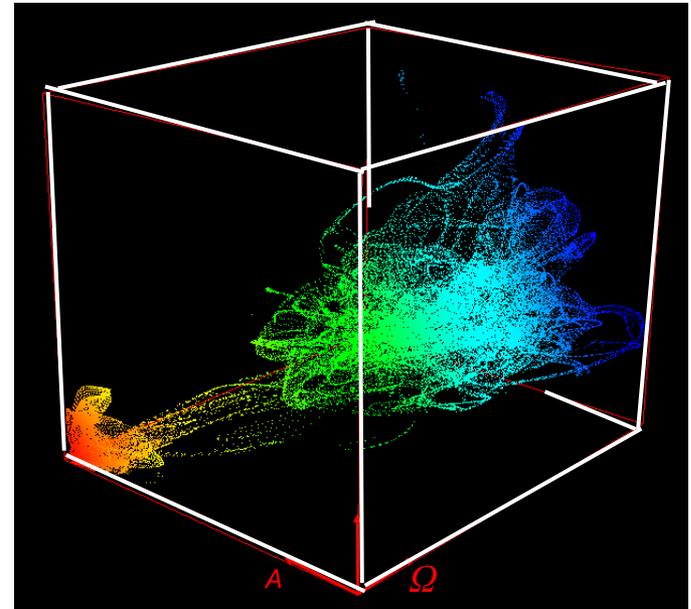


spatial

Filtrage  
Mean Shift



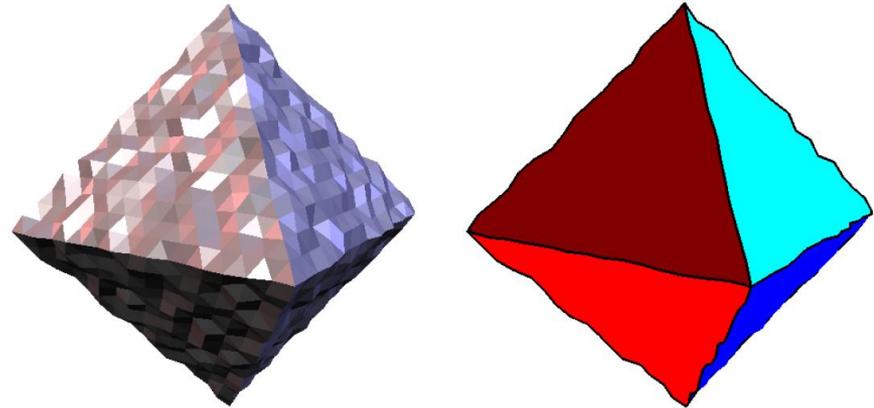
amplitudes



# Segmentation de ...

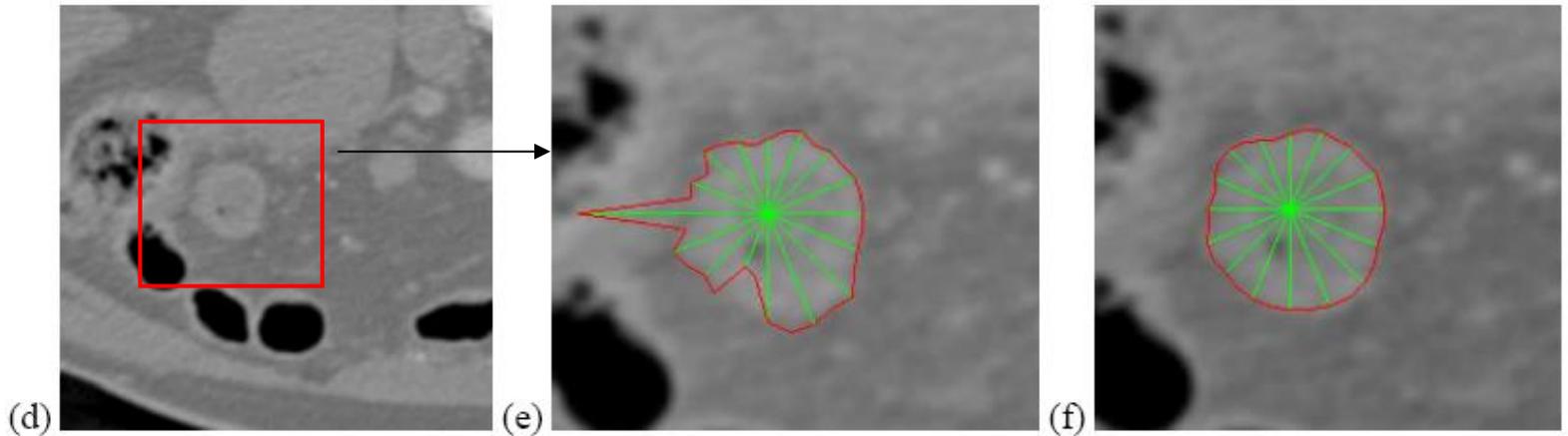
## ■ Maillages

[Yamauchi 2005]



## ■ Vaisseaux

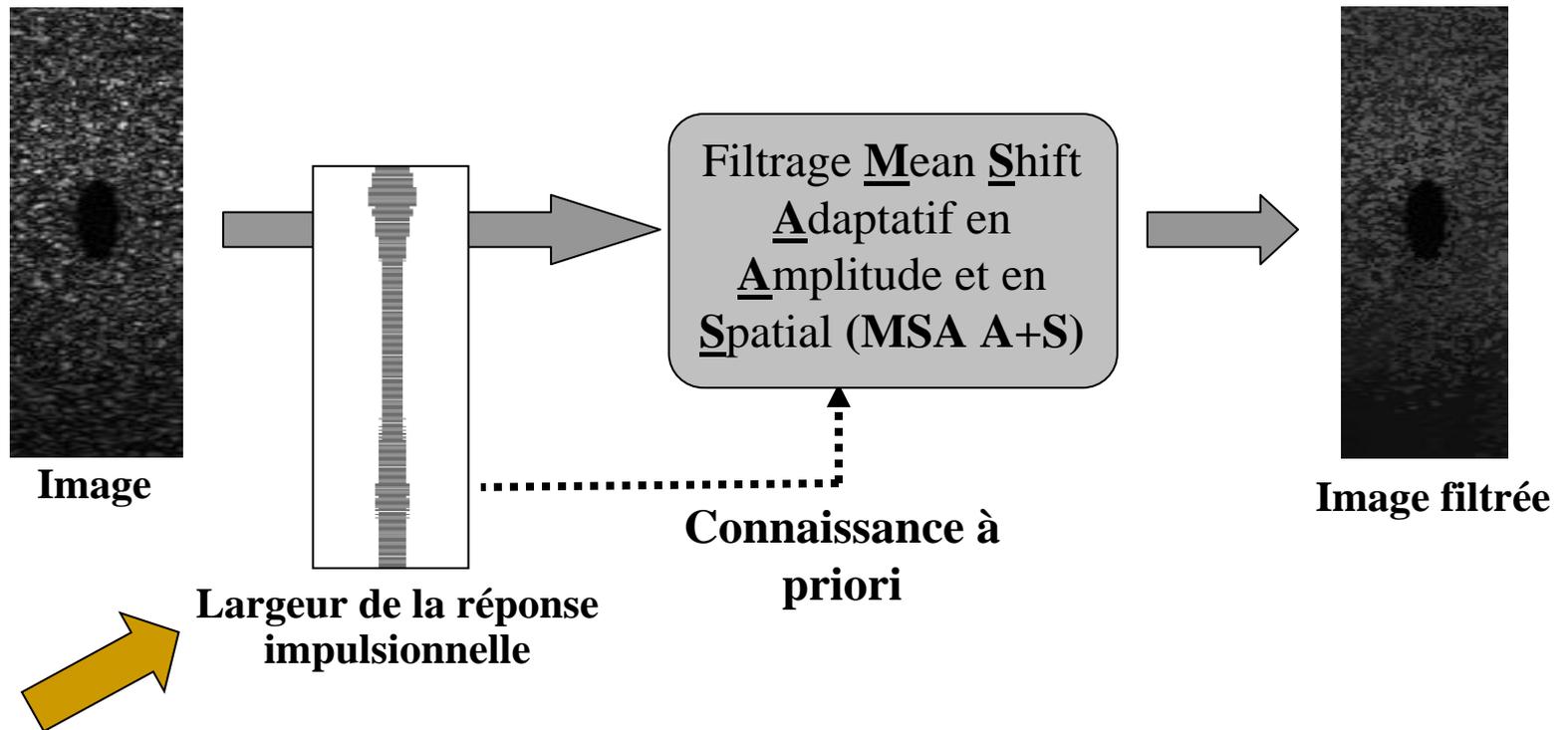
[Tek 2001]



# Ajout d'a priori ?

- Transformation de l'a priori dans un espace
  - Hough, pdf, échelle, polynôme, ...
- Définition d'une métrique adaptée à l'espace considéré
  - ML, Battacharya,
- Utilisation des Mean shift pour la recherche des maximum dans l'espace
- *Adaptation de l'échelle*
- *Modification de l'a priori*

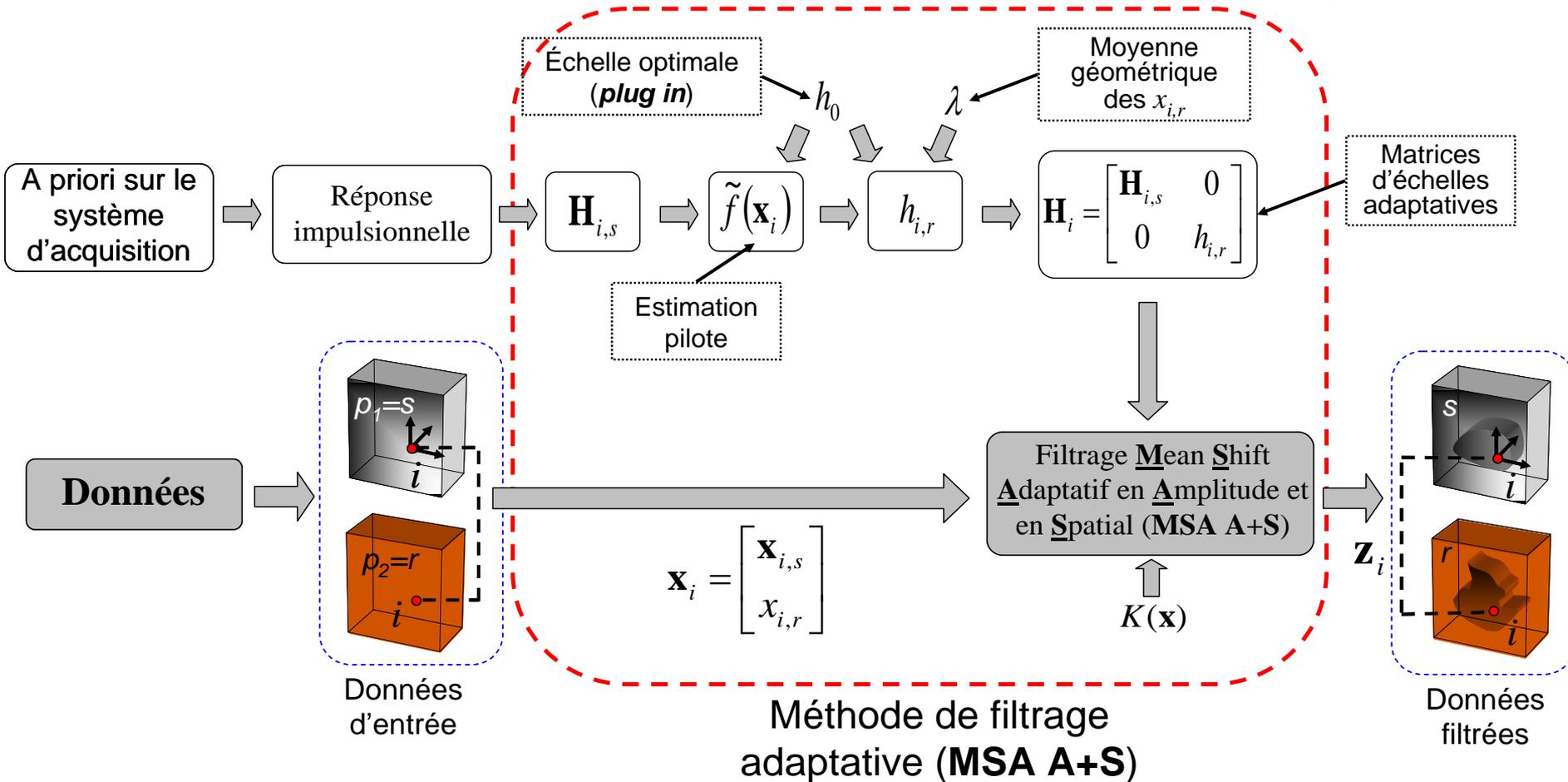
# Mean Shift Adaptatif avec a priori



# Mean Shift Adaptatif avec a priori

## ■ À priori : carte d'échelle

[Grenier 05]

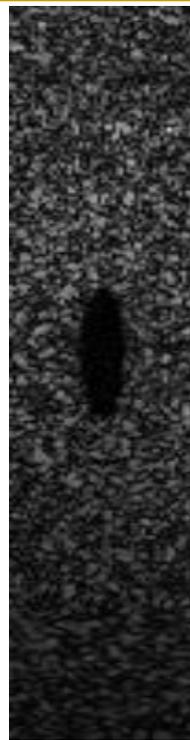
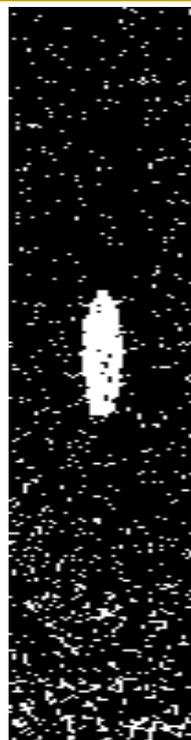
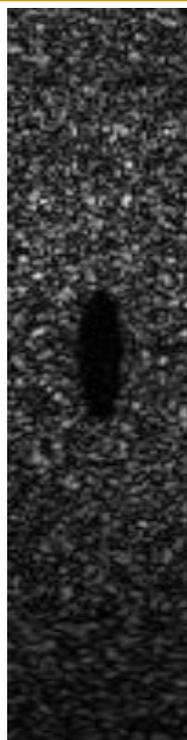


# Mean Shift Adaptatif avec a priori

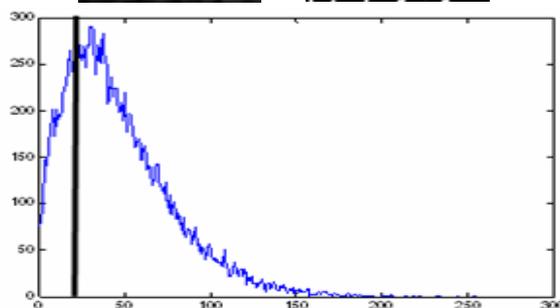
seuillage

seuillage

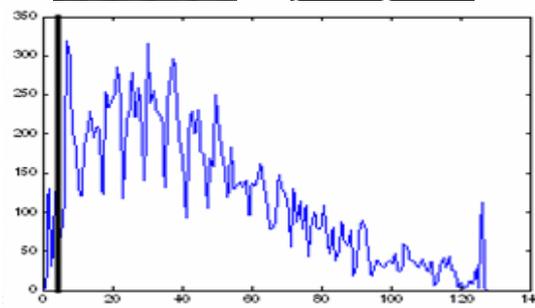
seuillage



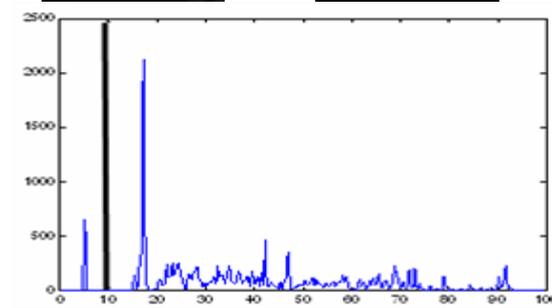
histogramme



Donnée originale



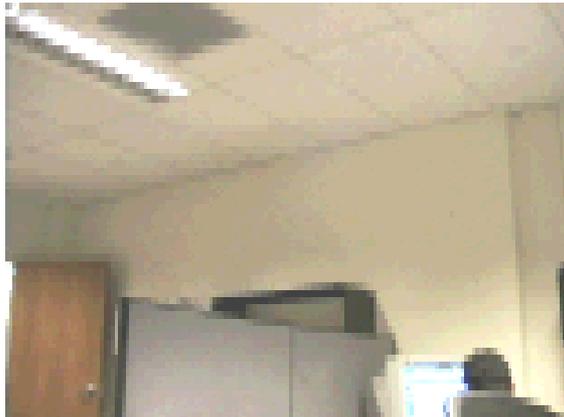
MS classique



MS avec a priori

# Tracking

- Modification du formalisme Mean Shift:  
→ recherche d'un motif dans une image



[Dorin Comaniciu, 2003]

# Tracking

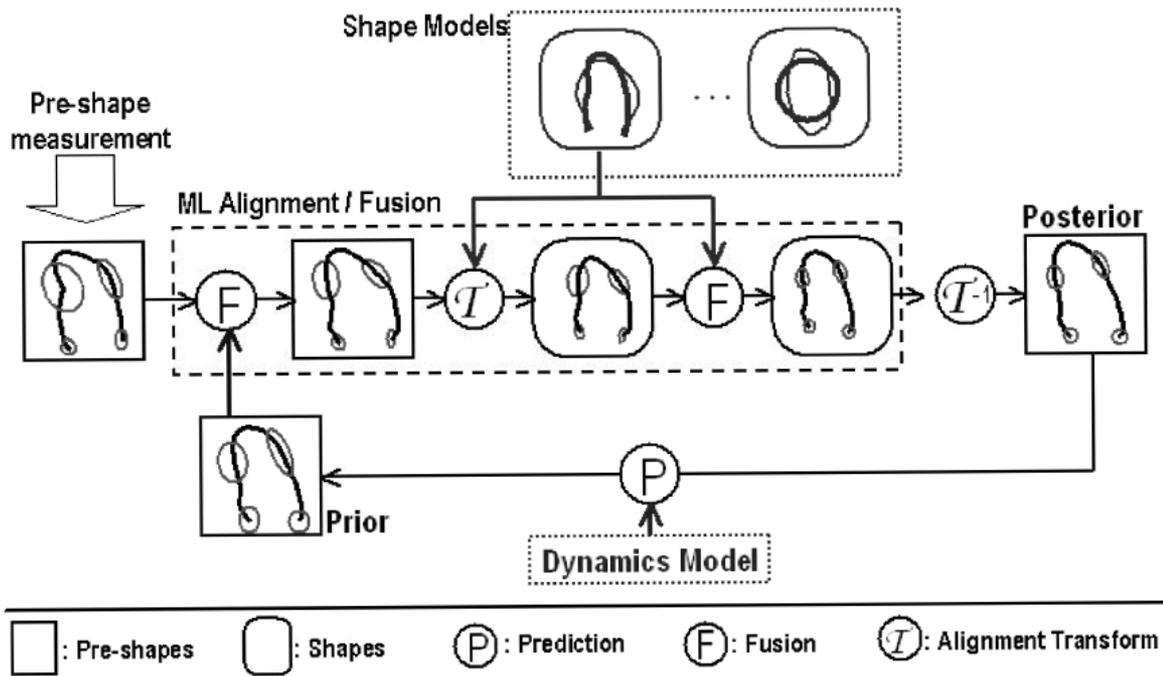
- Modification du formalisme Mean Shift:

→ recherche d'un motif dans une image avec Mean Shift  
Adaptatif et un a priori « multi-model »

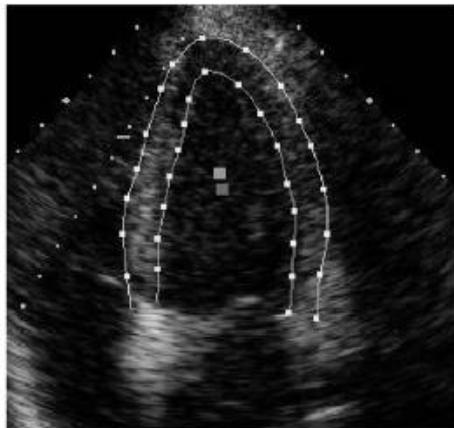
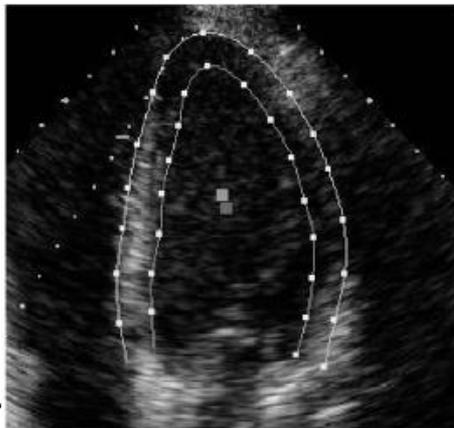
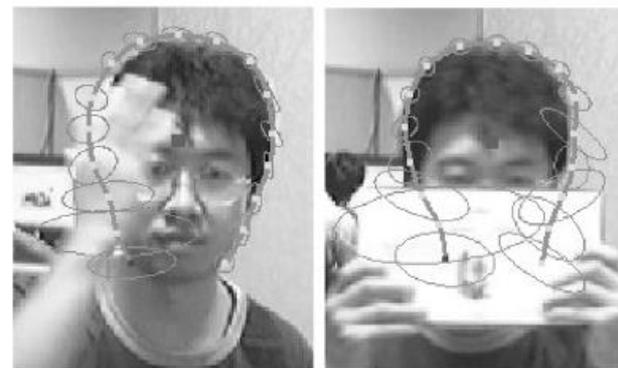
[Georgescu 2004]



# Encore un exemple



[Zhou 2005]



# IV – Solutions d'implémentations

# Problème

- Ce qu'il faut (*au minimum*) implémenter

Solution  
simple  
(littérature)

$$\mathbf{x}^{[t+1]} = \frac{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H})) \cdot \mathbf{x}_i}{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H}))}$$

ou

Solution  
générale

$$\mathbf{x}_j^{[t+1]} = \frac{\sum_{i=1}^n N(\mathbf{x}_j^{[t]}, \mathbf{x}_{i,j}, k_{\mathbf{H}}) \cdot \mathbf{x}_{i,j}}{\sum_{i=1}^n N(\mathbf{x}_j^{[t]}, \mathbf{x}_{i,j}, k_{\mathbf{H}})}$$

avec:

$$N(\mathbf{x}_j, \mathbf{x}_{i,j}, k_{\mathbf{H}}) = k_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i) \cdot \frac{g_j(d^2(\cdot))}{k_j(d^2(\cdot))} \quad k_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i) = \prod_{j=1}^c k_j(d^2(\cdot))$$

$$g(u) = -k'(u)$$

$$d^2(\cdot) = (\mathbf{x}_j - \mathbf{x}_{i,j})^T \mathbf{H}_j^{-1} (\mathbf{x}_j - \mathbf{x}_{i,j}) \quad : \text{distance de Mahalanobis (au carré) de la caractéristique } j$$

# Problème

## ■ Cout algorithmique théorique

- Temps de référence :

T = Calcul de Somme pondérée

$$\mathbf{x}^{[t+1]} = \frac{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H})) \mathbf{x}_i}{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H}))}$$

- Calcul de  $\mathbf{x}^{[t+1]}$  en  $O(n)$

- Pour un pixel :  $t=0 \dots t_{conv}$

- Ex. Filtrage Mean Shift nonblurring

Pour une image de  $p$  pixels

$$\text{FiltrageMeanShift} = O(p.n.t_{conv})$$

# Exo, filtrage Mean Shift *blurring*

- On dispose d'une image en niveaux de gris de  $n \times n$  pixels.
- Chaque pixel est représenté par un vecteur 3D (position spatiale, amplitude) dans la matrice  $E$ :
  - $E(x_1, x_2) = (x_1, x_2, I)^T$ , avec  $I$  le niveau de gris du pixel  $(x_1, x_2)$
- Ecrire le programme permettant de filtrer l'image  $E$  en utilisant la procédure Mean Shift blurring avec noyau gaussien,  $h_s, h_r, \varepsilon$

# Problème

## ■ Rappels pour calcul (*solution classique, non générale*)

Distance de Mahalanobis (au carré):

$$d^2(\cdot) = (\mathbf{x} - \mathbf{x}_i)^T \mathbf{H}^{-1} (\mathbf{x} - \mathbf{x}_i)$$

$$\mathbf{x}^{[t+1]} = \frac{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H})) \mathbf{x}_i}{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H}))}$$

Dérivée du profil:

$$g(u) = -k'(u)$$

$$K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2} \mathbf{x})$$

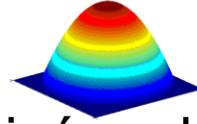
$$K(\mathbf{x}) = C_{k,d} \cdot k(\mathbf{x}^T \mathbf{x})$$

Ex. pour Epanechnikov:

$$K_e(\mathbf{x}) = \begin{cases} \left(\frac{1}{2}\right) \cdot v_d^{-1} \cdot (d+2) \cdot (1 - \mathbf{x}^T \mathbf{x}) & \text{si } |\mathbf{x}^T \mathbf{x}| \leq 1 \\ 0 & \text{sinon} \end{cases} \longrightarrow g_e(u) = \begin{cases} 1 & \text{si } |u| \leq 1 \\ 0 & \text{sinon} \end{cases}$$

# Dans la pratique

- Noyau : Epanechnikov



- sans prise en compte des dérivées des autres dimensions

- Noyau sphérique

- $g() =$  fonction porte (seuillage)

- Matrice d'échelle diagonale  $h_s, h_r$

- Utilisation de l'algorithme nonblurring

- Pas de modification des données servant à l'estimation (les  $\mathbf{x}_i$  sont constants)

- Estimation sur les points du voisinage  $n = (2h_s + 1)^s$

- Faible nombre d'itérations  $t_C$

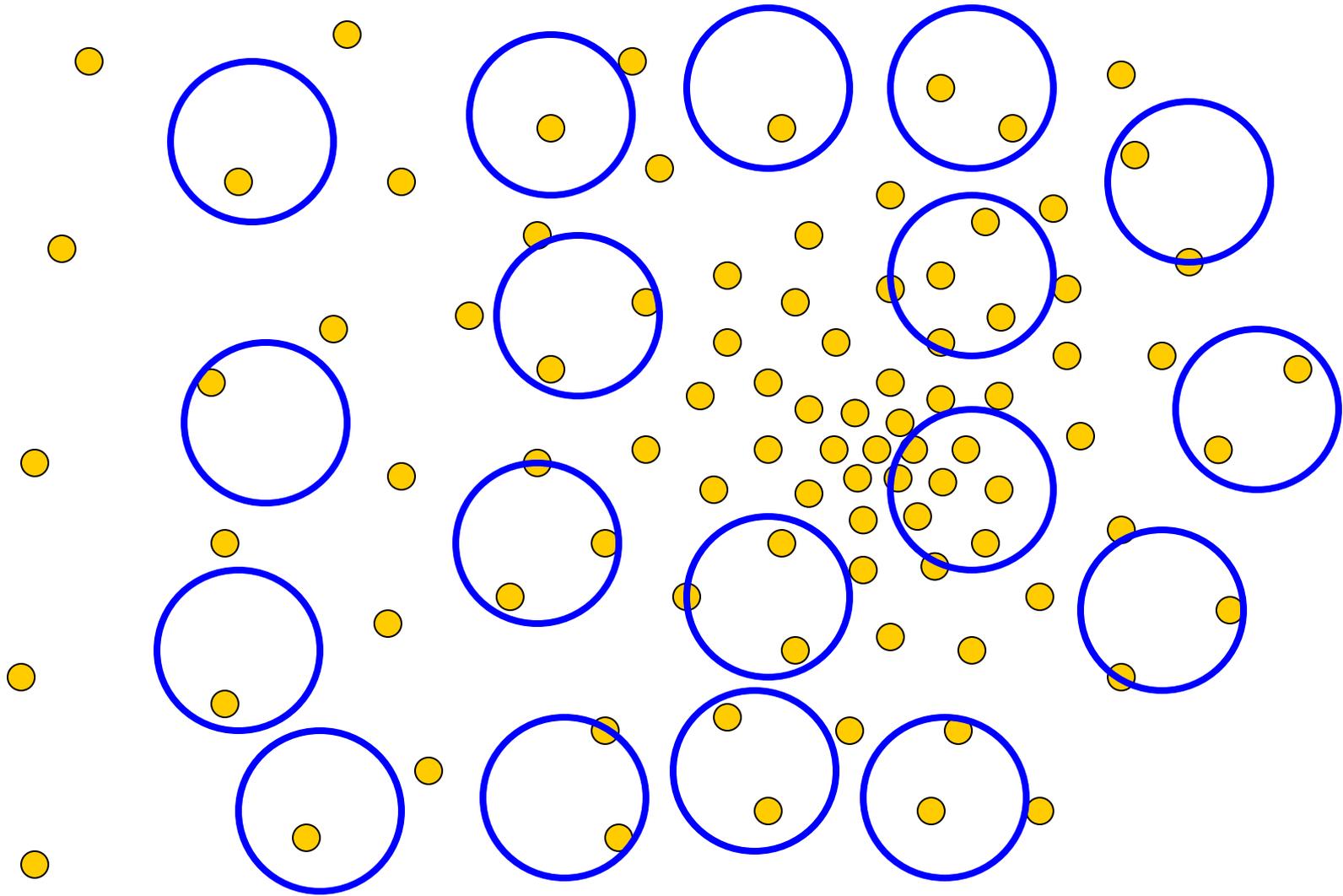
- Pas de recherche de la convergence exacte

$$t_C \approx \overline{t_{conv}}$$

- *Filtrage de quelques points de l'image*

$$\mathbf{x}^{[t+1]} = \frac{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H})) \mathbf{x}_i}{\sum_{i=1}^n g(d^2(\mathbf{x}^{[t]}, \mathbf{x}_i, \mathbf{H}))}$$

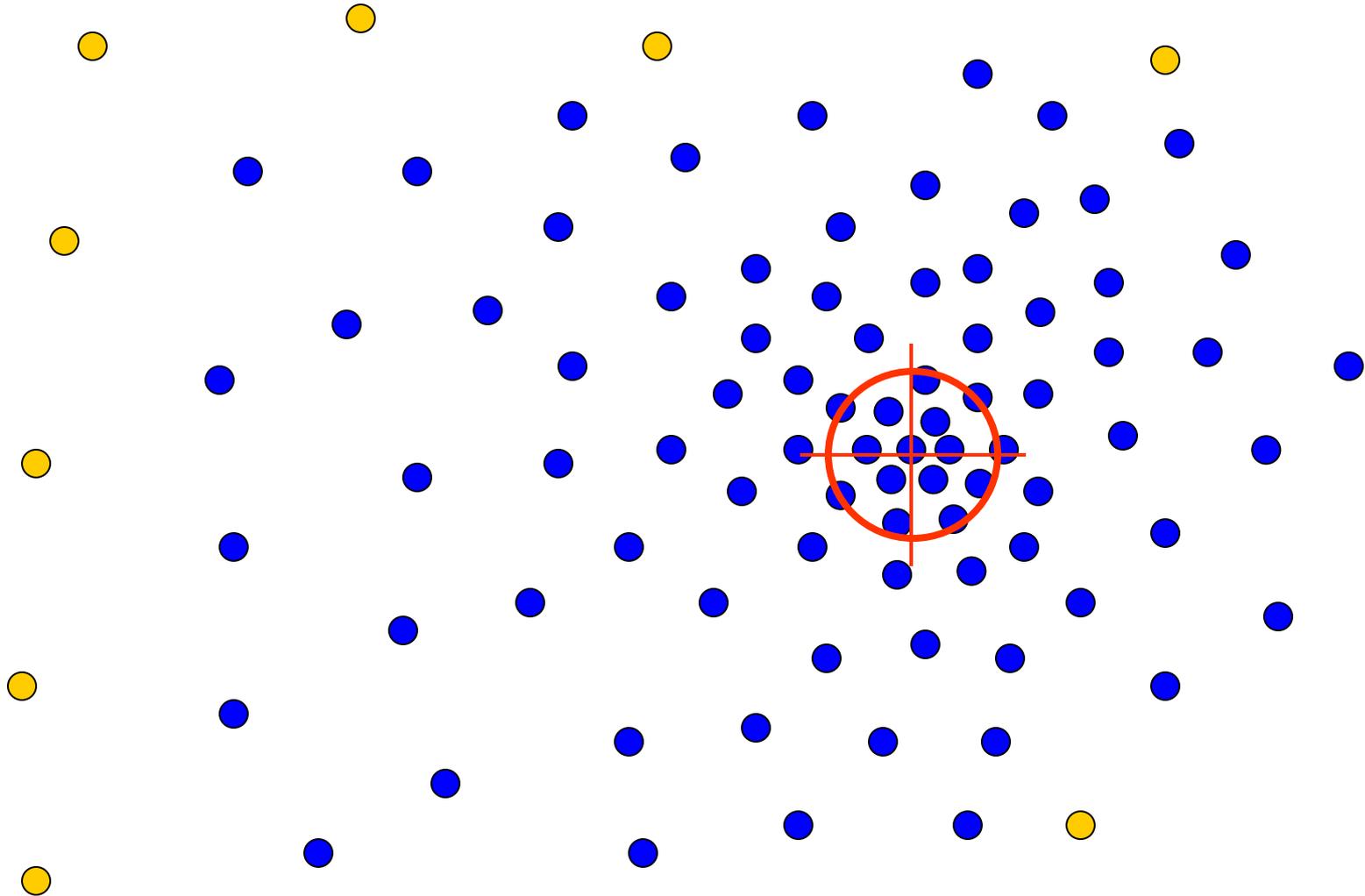
# Filtrage de quelques points



**Tessellate the space  
with windows**

**Run the procedure in parallel**

# Filtrage de quelques points



The blue data points were traversed by the windows towards the mode

# Dans la pratique

- Implémentation de versions approchées mais efficaces...
  - *Temps réel pour tracking (vidéo, séquence US)*
  - *Qq secondes pour filtrage d'image  $256^2$*