

BASES DU TRAITEMENT D'IMAGE

Généralités

Quelques exemples

- Robotique
- Médecine
- Industrie
- Cinéma

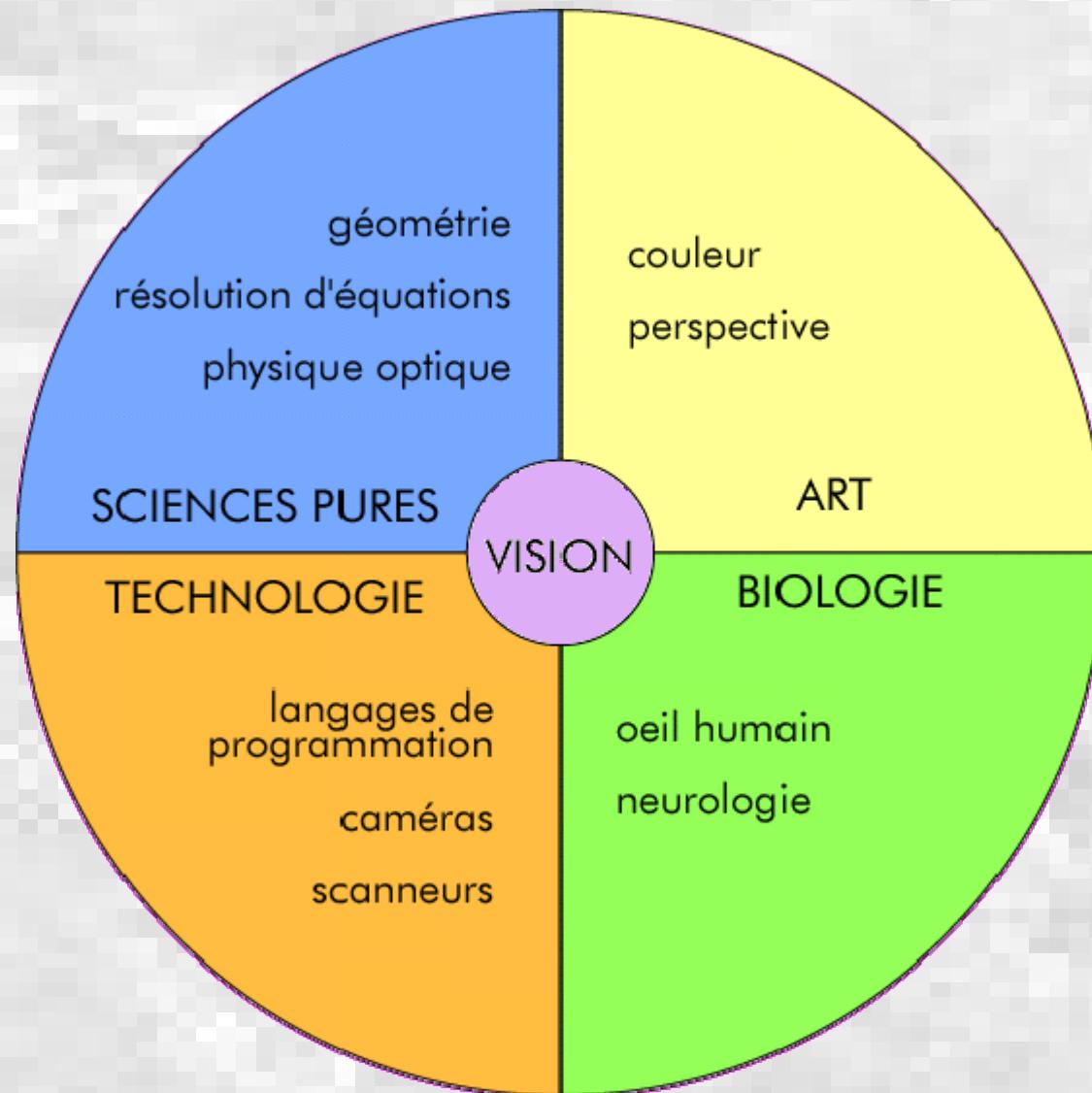


Copyright © 1998 Tristar Pictures, Inc.

Source: Laboratoire de robotique mobile de l'Université de Sherbrooke
Source: Matrox Imaging

Généralités

A Principe



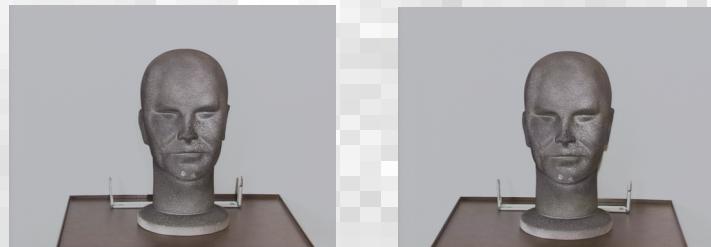
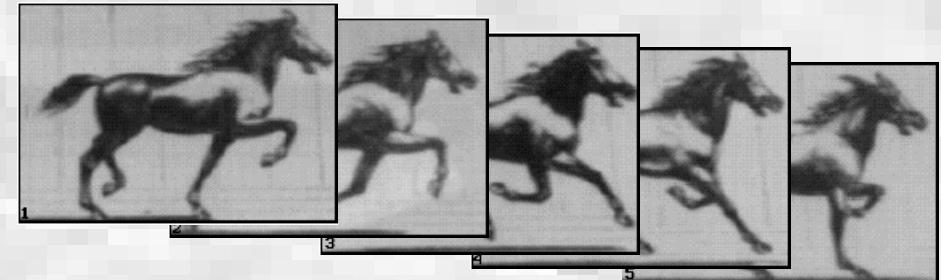
Généralités

A Principe

« Traiter » une image : Extraire une information utile de l'image.

Film

- mesure du déplacement
- poursuite de cible



Paire stéréoscopique

- mesure de la profondeur
- Reconstruction 3D



Echographie

- Localisation/mesure

Généralités

B Exemples de traitement

reconnaissance de forme

NUMERISATION

Image continue \Rightarrow image discontinue et quantifiée

PRETRAITEMENT

Restauration de l'image

SEGMENTATION

Partition de l'image

ATTRIBUTS DE FORME

Détermination de paramètres géométriques représentatifs.

IDENTIFICATION

Confrontation des attributs précédents avec une référence.

Généralités

B Exemples de traitement

Tracking

NUMERISATION

PRETRAITEMENT

Estimation du champ de déplacement

SEGMENTATION

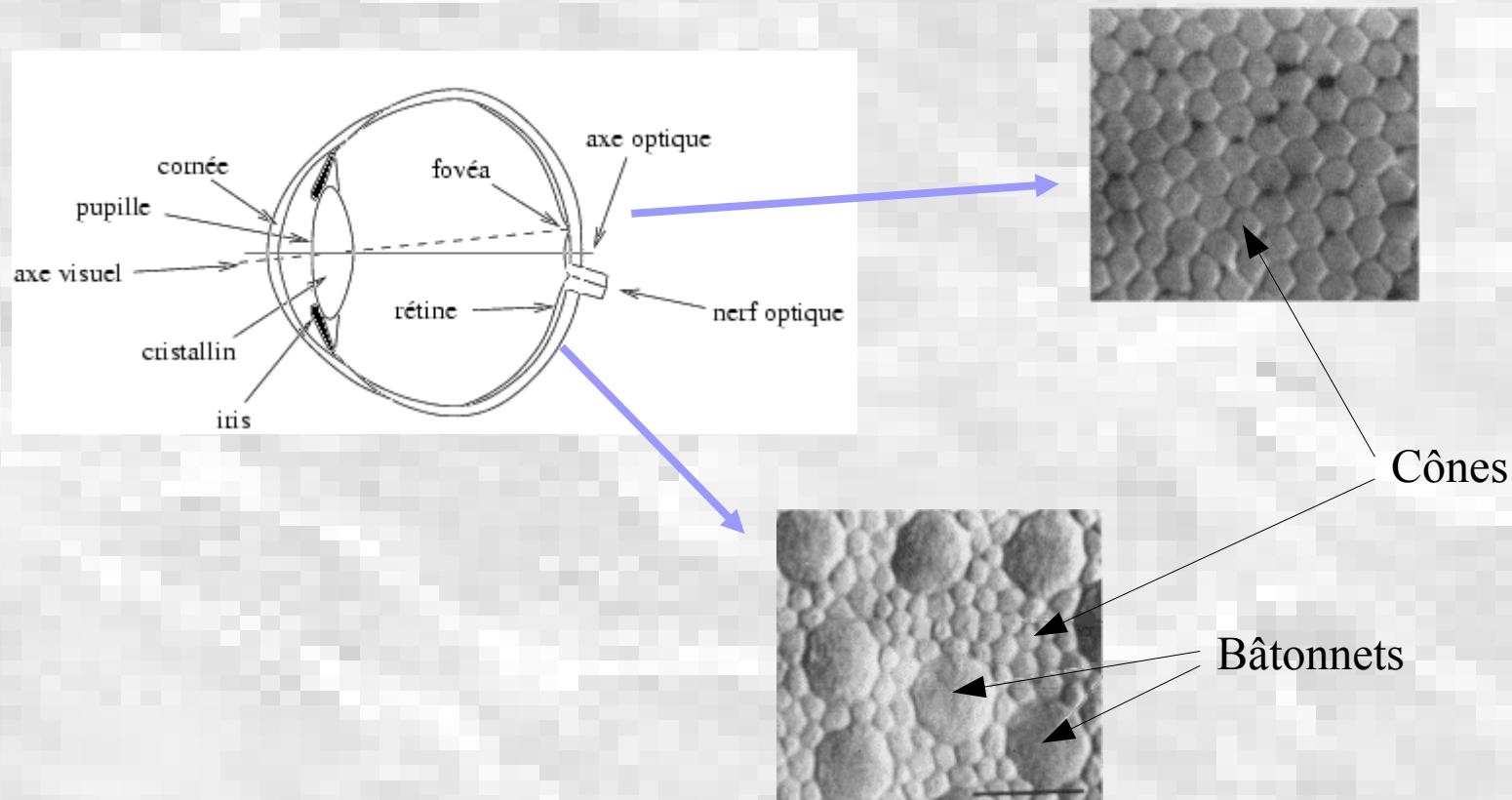
Segmentation au sens du mouvement de l'image

POURSUITE

Acquisition & représentation des images

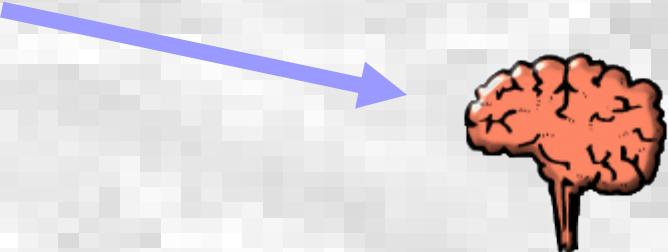
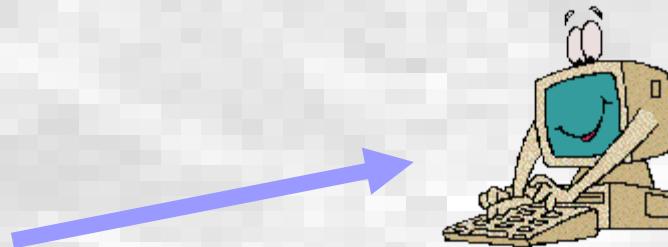
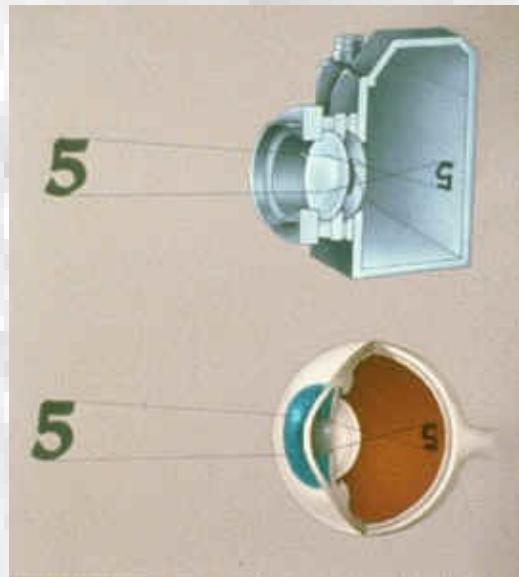
Acquisition & représentation

A Système visuel humain



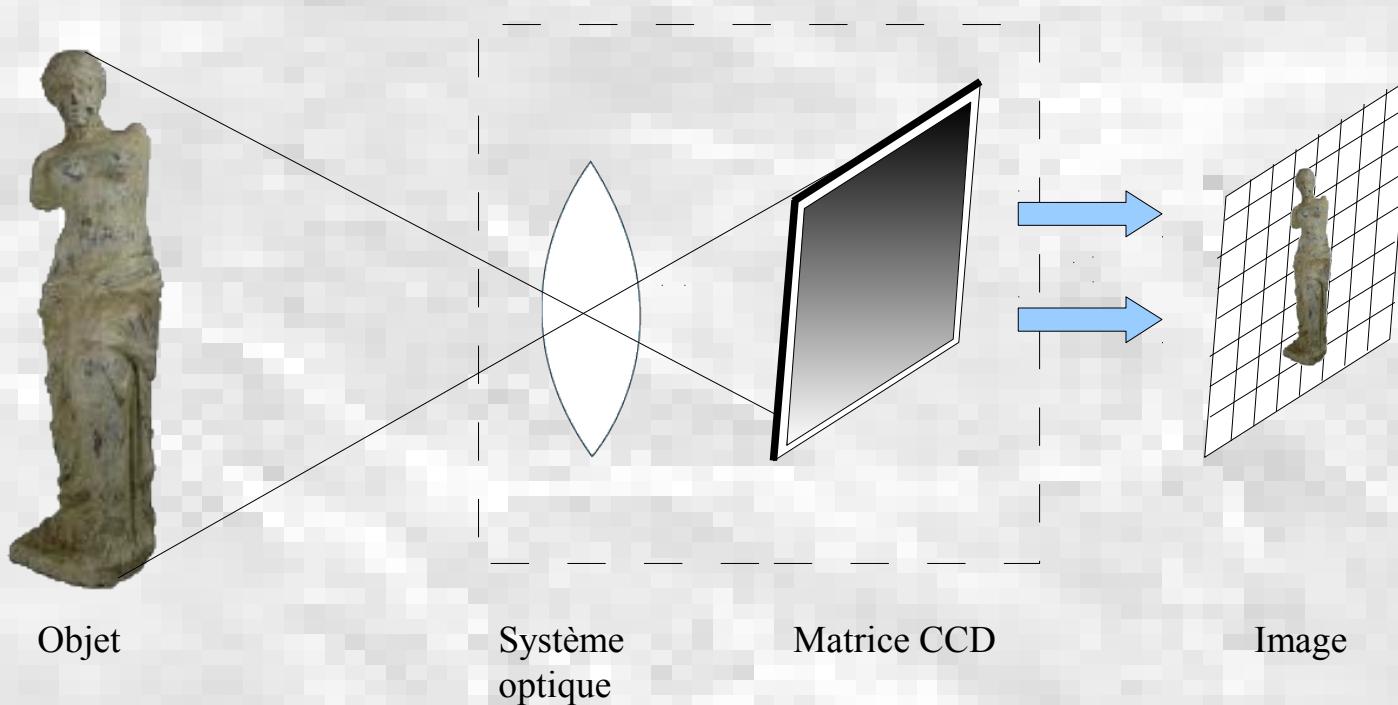
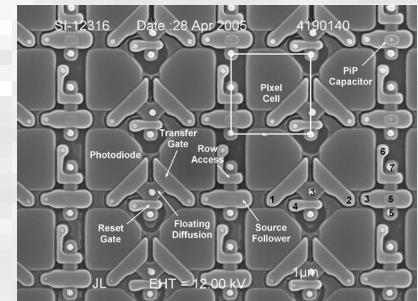
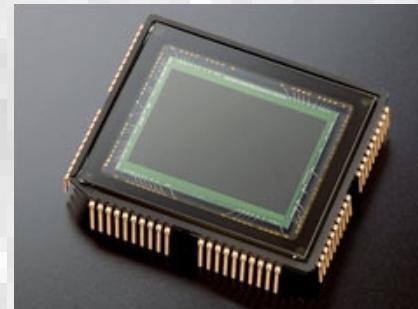
Acquisition & représentation

B Cas de l'ordinateur

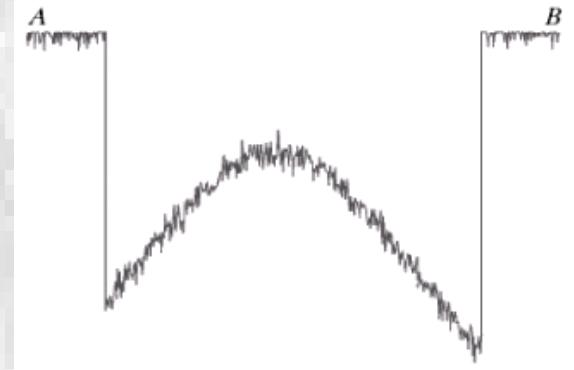
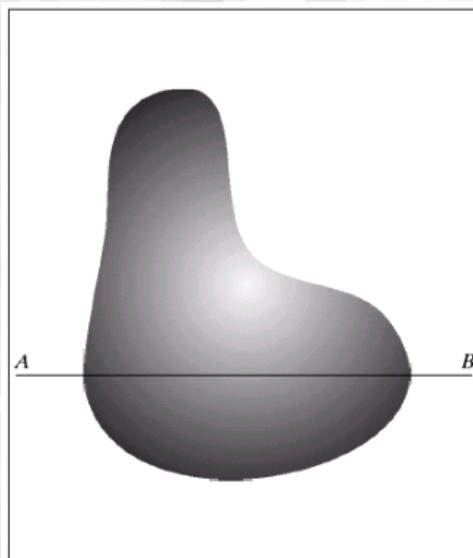


Acquisition & représentation

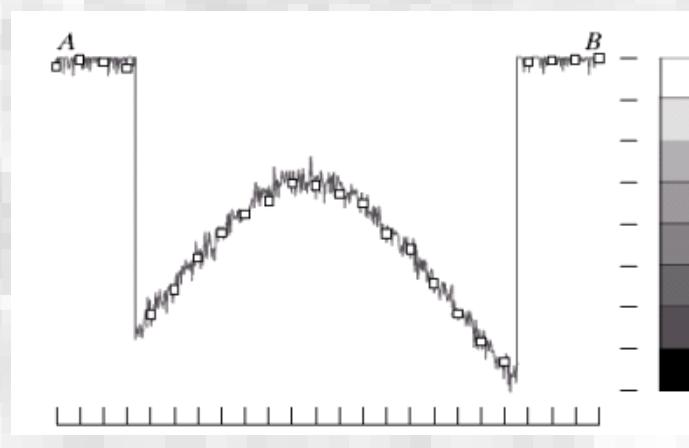
B Cas de l'ordinateur



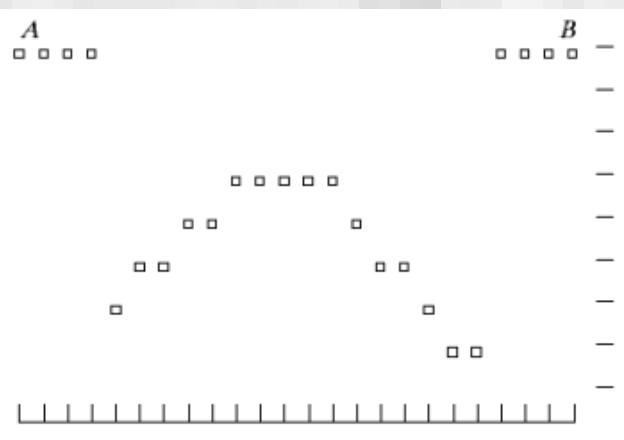
Acquisition & représentation B Cas de l'ordinateur



Signal source



Échantillonnage
Quantification

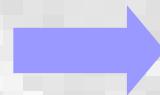
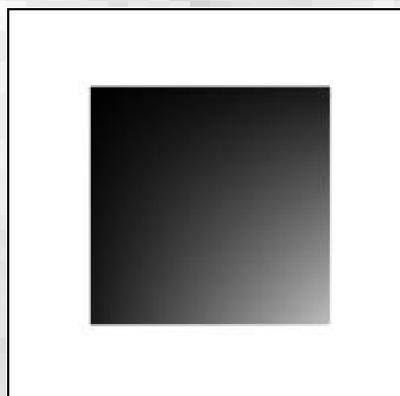


Signal
échantillonné
enregistré

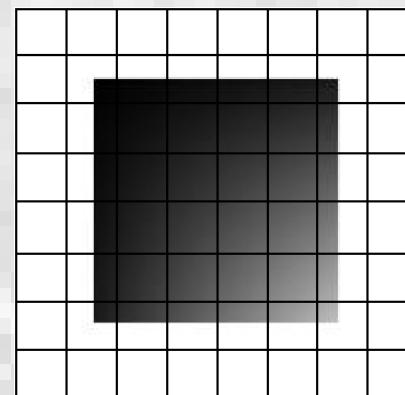
Acquisition & représentation

C Images en niveaux de gris

Image physique (continue)

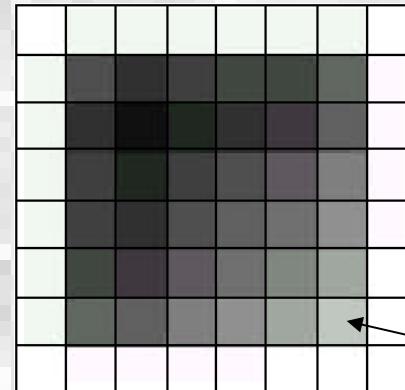


Échantillonnage



Quantification

255	255	255	255	255	255	255	255	255
255	82	26	40	49	50	98	255	
255	26	0	31	40	42	76	255	
255	40	31	56	59	77	106	255	
255	49	40	59	68	103	125	255	
255	50	42	77	103	124	146	255	
255	98	76	106	125	146	177	255	
255	255	255	255	255	255	255	255	255

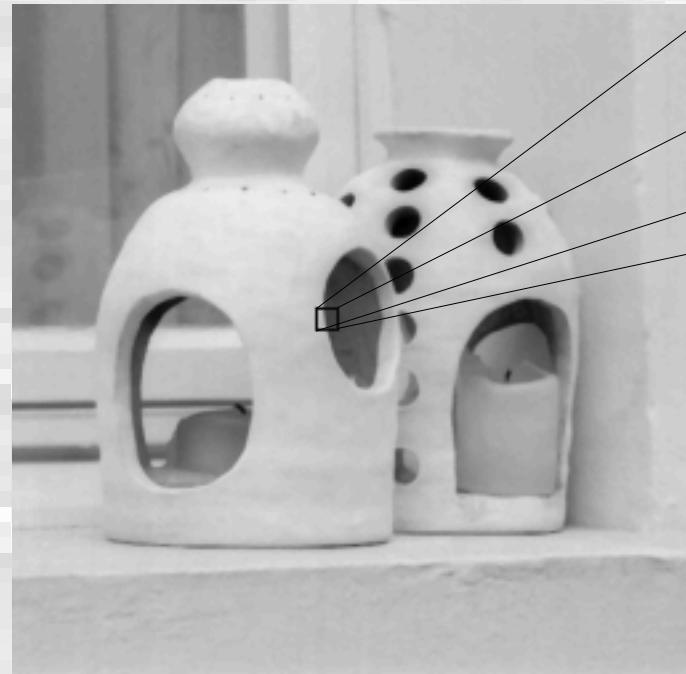


Pixel (« picture element »)

Acquisition & représentation

C Images en niveaux de gris

Images en niveaux de gris



211	198	106	71	71	71	71	71
207	203	119	71	71	71	71	75
207	203	132	71	71	75	71	71
207	203	150	71	71	75	71	71
203	203	163	75	71	71	75	71
207	203	176	80	71	71	71	71
207	207	194	97	75	71	71	75

Exemple :

l'image en niveaux de gris :

1 pixel \Leftrightarrow 1 octet (0 - 255)

Résolution



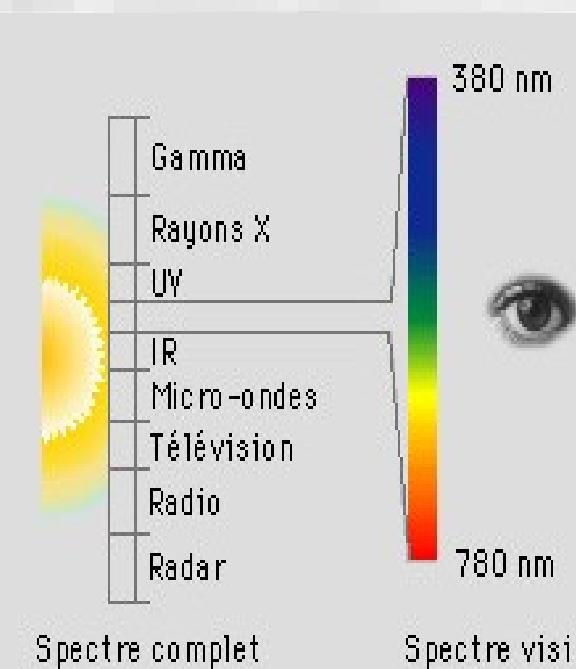
Quantification



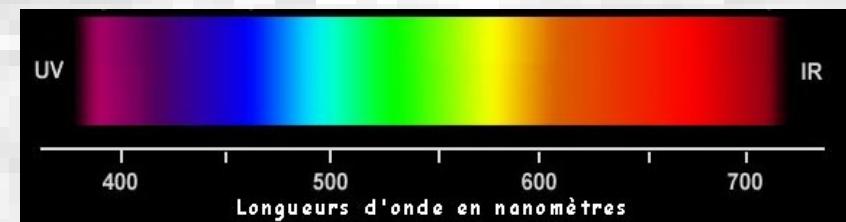
Acquisition & représentation

D Perception de la couleur

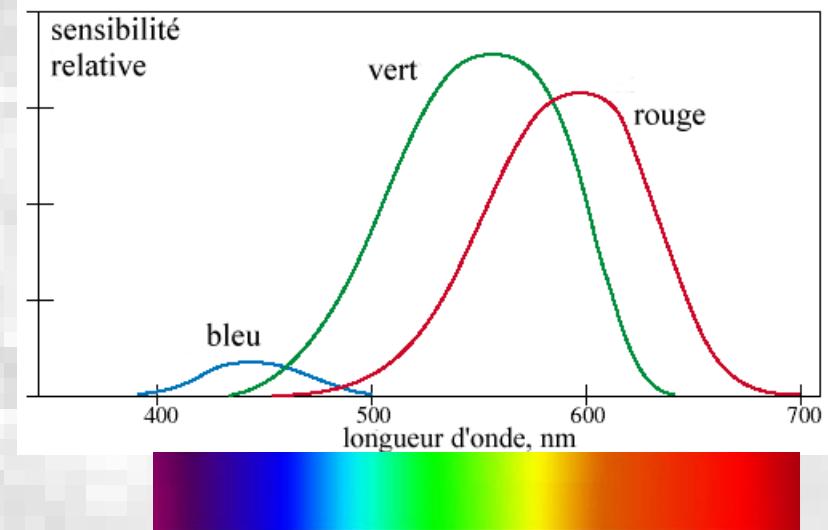
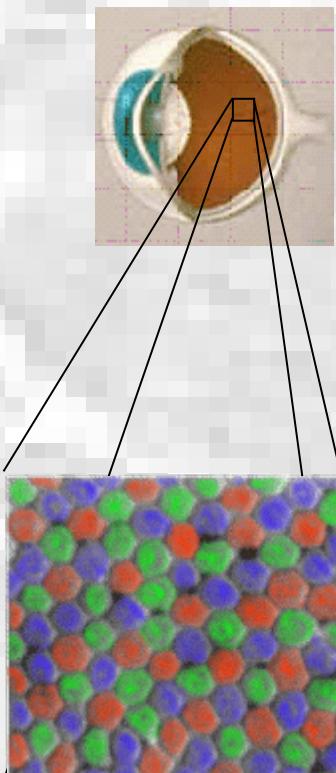
spectre complet



spectre visible

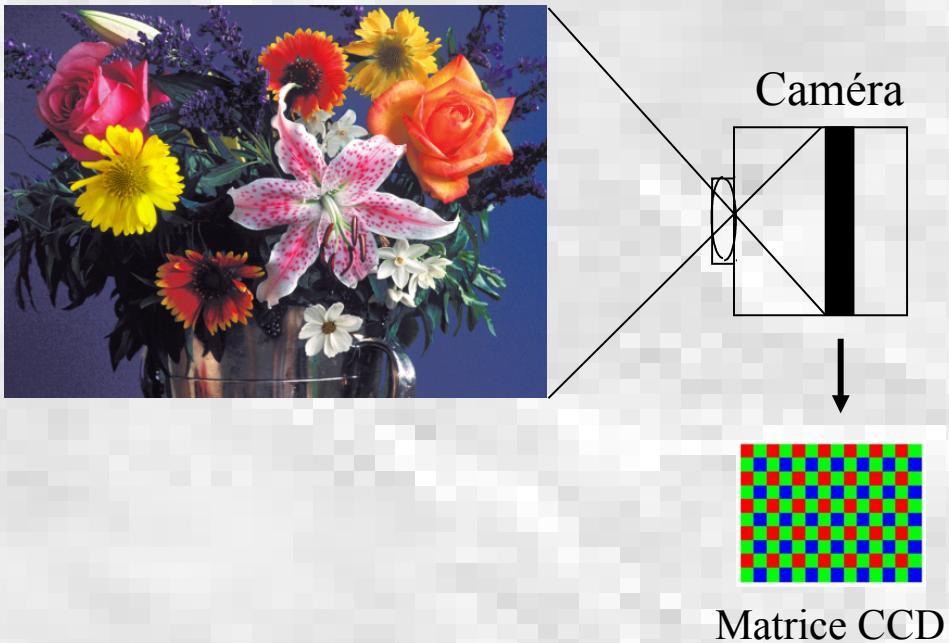


Perception de la couleur



Acquisition & représentation

E Images en couleur



Acquisition & représentation

E Images en couleur



155	158	159	155	155
134	132	131	131	131
148	147	146	146	142
158	161	159	160	160
223	217	202	192	185

Composante rouge

175	175	176	174	175
167	168	166	167	167
172	173	172	173	170
168	170	169	171	171
218	212	199	190	183

Composante verte

192	192	192	190	191
192	191	190	191	191
196	193	191	194	190
185	187	186	188	190
216	213	200	192	188

Composante bleue

Image couleur :

1 pixel \Leftrightarrow 3 octets
rouge (0 - 255)
vert (0 - 255)
bleu (0 - 255)

Filtrage linéaire

Filtrage linéaire

A Principe

- Principe :
 - Éliminer dans une image certaines composantes fréquentielles.
- Processus général :
 - La nouvelle image est une somme pondérée (combinaison linéaire) de l'image d'origine.
- Propriétés :
 - La sortie est linéairement dépendante de l'entrée.
 - La sortie est une fonction invariante par translation de l'entrée (si l'entrée est décalée de n pixels => la sortie est décalée de n pixels).
- Applications :
 - Filtrage linéaire (débruitage).
 - Calcul du gradient de l'image (déttection de contours).
 - Identifier des textures (segmentation en région).

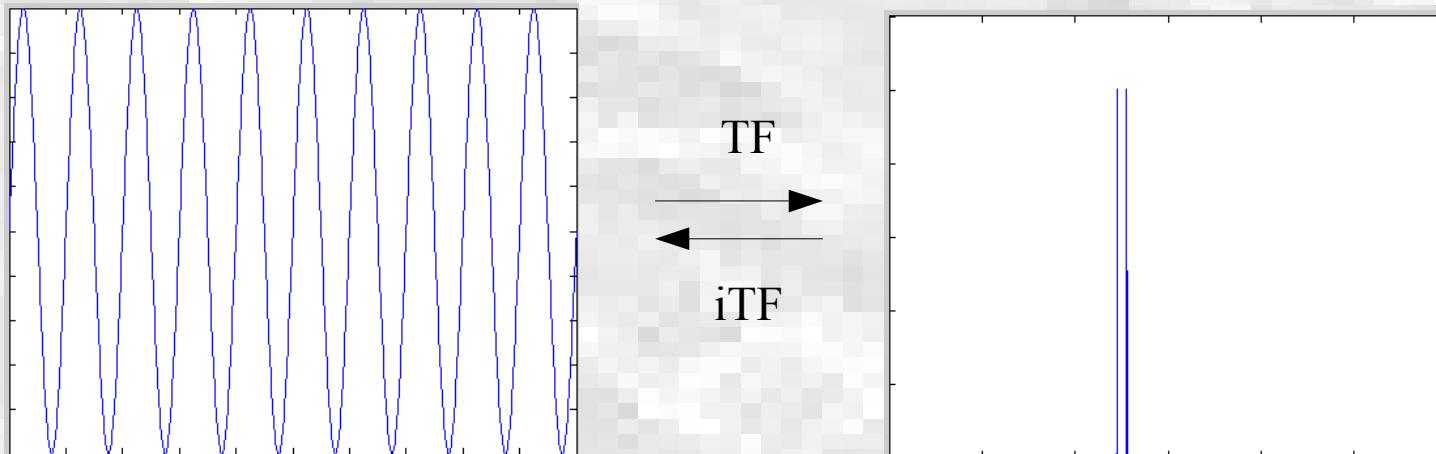
Filtrage linéaire

B Rappel sur la transformée de Fourier 1D

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i ux} dx \quad \xrightleftharpoons[\text{iTF}]{\text{TF}} \quad f(x) = \int_{-\infty}^{+\infty} F(u) e^{2\pi i ux} du$$

N.B. : $F(u)$ est complexe.

On définit donc : $F(u) = ||F(u)|| \cdot e^{i\theta(u)}$ avec : $\begin{cases} ||F(u)|| & : \text{Spectre fréquentiel.} \\ \theta(u) & : \text{Spectre de phase.} \end{cases}$



Filtrage linéaire

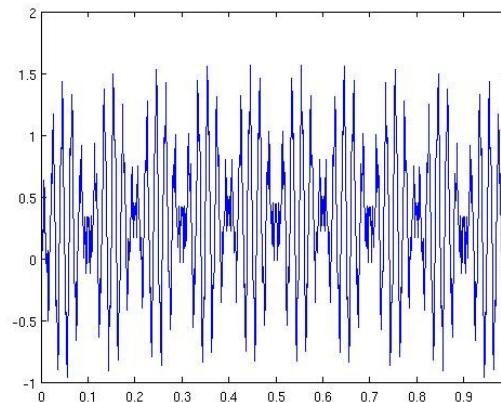
B Rappel sur la transformée de Fourier 1D

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i ux} dx \xrightleftharpoons[\text{iTF}]{\text{TF}} f(x) = \int_{-\infty}^{+\infty} F(u) e^{2\pi i ux} du$$

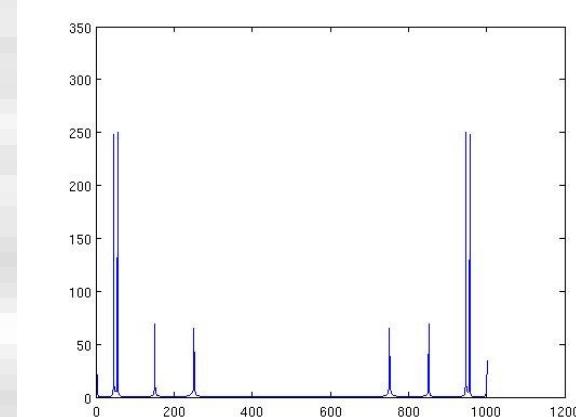
N.B. : $F(u)$ est complexe.

On définit donc : $F(u) = ||F(u)|| \cdot e^{i\theta(u)}$ avec :

$ F(u) $: Spectre fréquentiel.
$\theta(u)$: Spectre de phase.



$$\xrightleftharpoons[\text{iTF}]{\text{TF}}$$



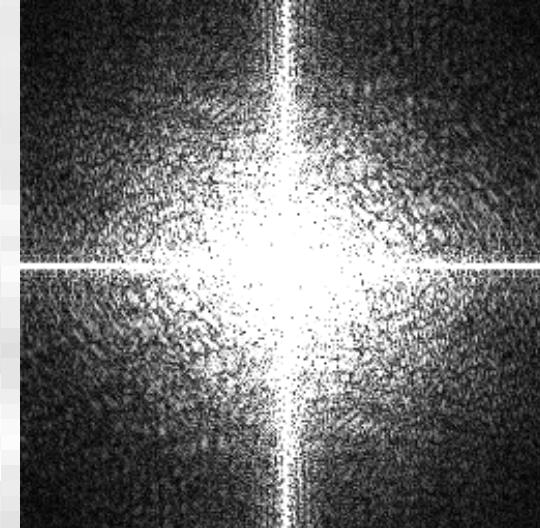
Filtrage linéaire

C La transformée de Fourier 2D

$$F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-2\pi i(ux+vy)} dx dy \xrightleftharpoons[\text{iTF}]{\text{TF}} f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v) e^{2\pi i(ux+vy)} du dv$$



$$\xrightleftharpoons[\text{iTF}]{\text{TF}}$$



Filtrage linéaire

C La transformée de Fourier 2D

Propriétés :

Séparabilité

$$F(u, v) = \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} f(x, y) e^{-2\pi i ux} dx \right] e^{-2\pi i vy} dy$$

La TF de $f(x, y)$ peut donc se réaliser en 2 étapes :

$$i) \text{ TF 1D de } f(x, y) \text{ pour tout } y \text{ fixé} \quad \Rightarrow \quad f'(u, y)$$

$$ii) \text{ TF 1D de } f'(u, y) \text{ pour tout } u \text{ fixé} \quad \Rightarrow \quad F(u, v)$$

N.B. : Si l'image $f(x, y)$ est séparable : $f(x, y) = f_x(x)f_y(y)$

$$\text{alors : } F(u, v) = F_x(u).F_y(v)$$

Filtrage linéaire

C La transformée de Fourier 2D

Propriétés :

Linéarité

$$\text{Soit : } \quad TF[f_1(x,y)] = F_1(u,v) \quad \text{et} \quad TF[f_2(x,y)] = F_2(u,v)$$

$$\forall (a,b) \in \mathbb{R}^2 \quad \quad \quad TF[a.f_1(x,y) + b.f_2(x,y)] = a.F_1(u,v) + b.F_2(u,v)$$

Homothétie

$$\text{Soit : } \quad TF[f(x,y)] = F(u,v)$$

$$TF[f(ax, by)] = \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

Filtrage linéaire

C La transformée de Fourier 2D

Expression discrète :

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

Filtrage linéaire

C La transformée de Fourier 2D

Exemple :

Soit : $f(x,y) = A.rect_{[a,b]}(x, y)$

avec $rect_{[a,b]}(x, y) = \begin{cases} 1 & |x| < a/2 \text{ et } |y| < b/2 \\ 0 & \text{sinon} \end{cases}$

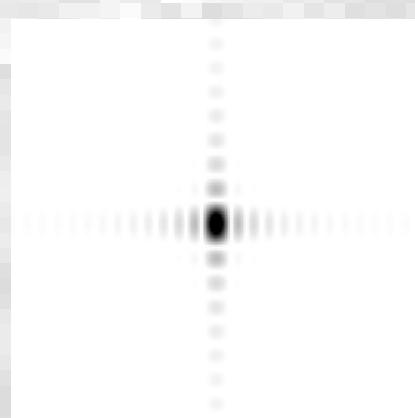
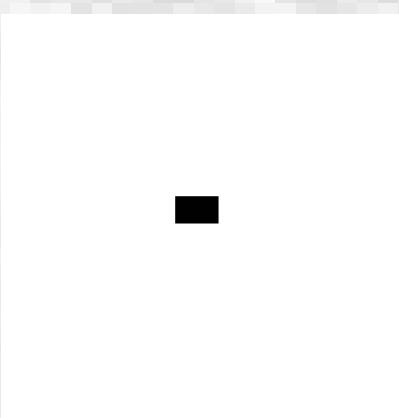
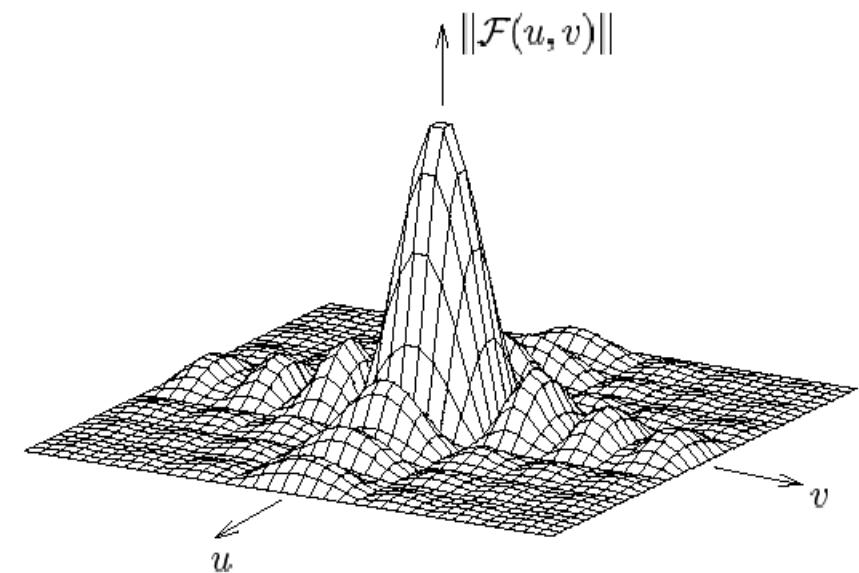
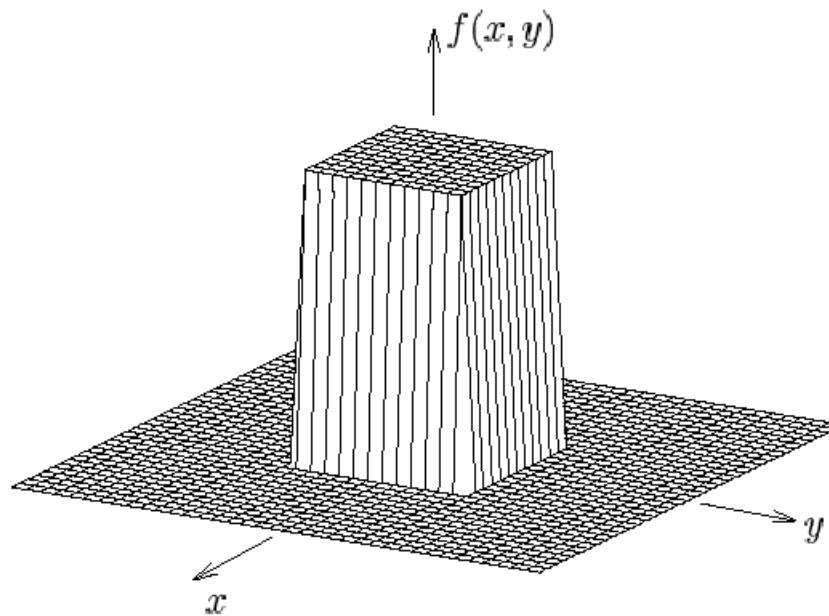
$$F(u, v) = A \int_{-a/2}^{a/2} e^{-2\pi i ux} dx \int_{-b/2}^{b/2} e^{-2\pi i vy} dy$$

$$F(u, v) = Aab \left(\frac{\sin(\pi au)}{\pi au} \right) \left(\frac{\sin(\pi bv)}{\pi bv} \right)$$

Filtrage linéaire

C La transformée de Fourier 2D

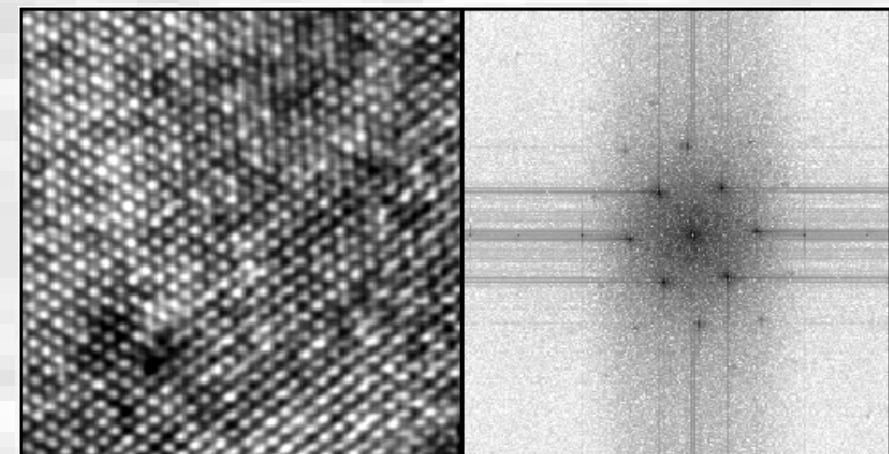
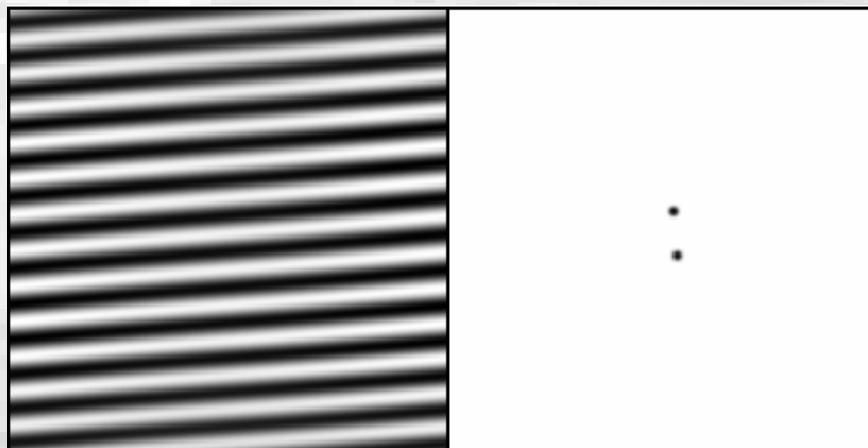
Exemple :



Filtrage linéaire

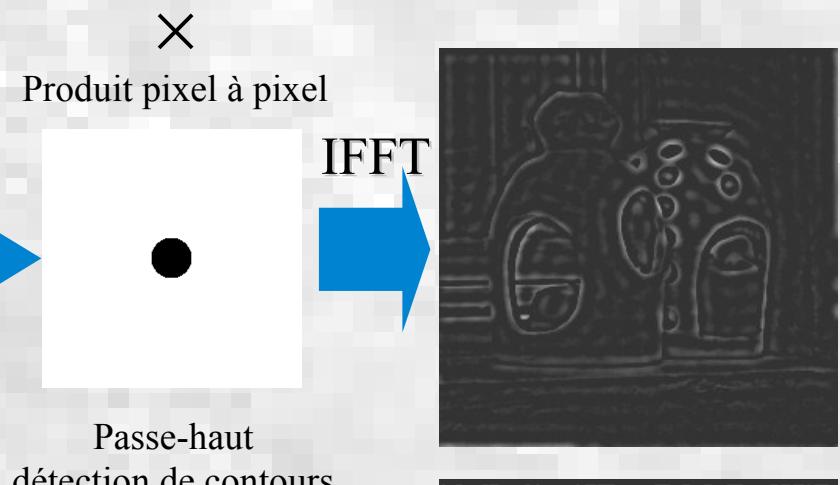
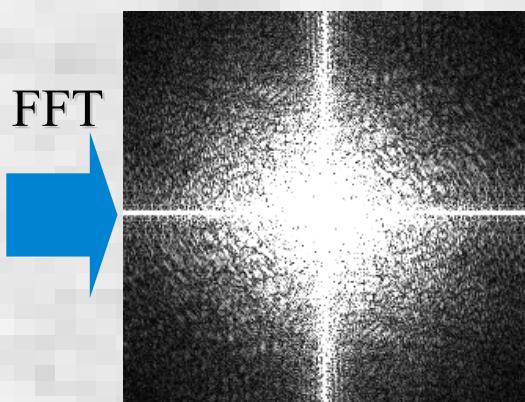
C La transformée de Fourier 2D

Exemple :

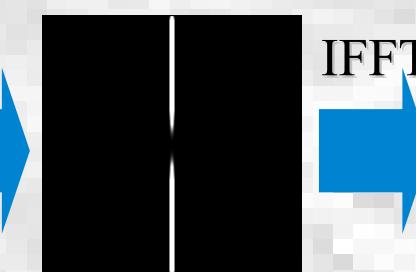


Filtrage linéaire

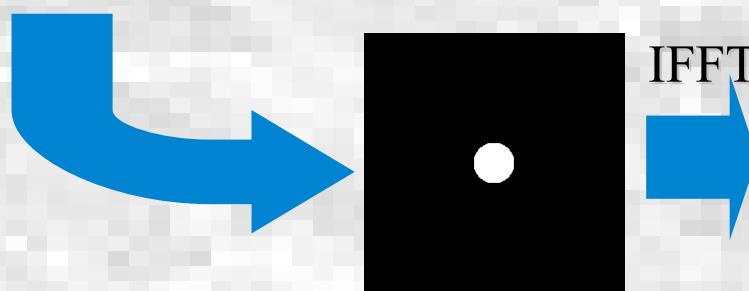
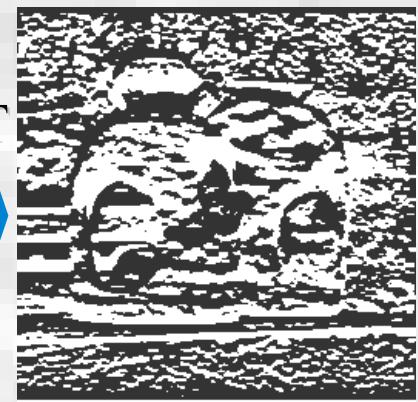
C La transformée de Fourier 2D



Passe-haut
détection de contours



Quelconque
détection de texture



Passe-bas
débruitage



Filtrer une image \Leftrightarrow Convoluer par un masque

Domaine fréquentiel :

$$\text{TF}(\mathbf{I}_f)(u, v) = \text{TF}(f)(u, v) \cdot \text{TF}(I)(u, v)$$

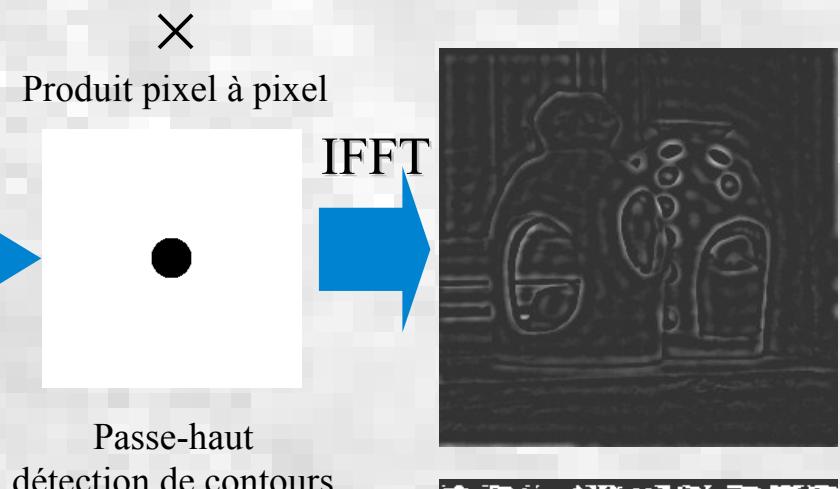
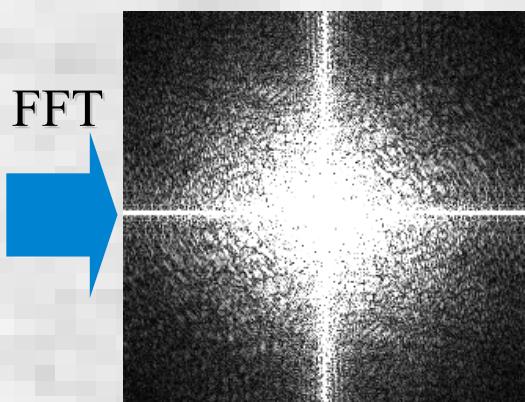
Domaine spatial :

$$I_f(x, y) = f(x, y) \otimes I(x, y)$$

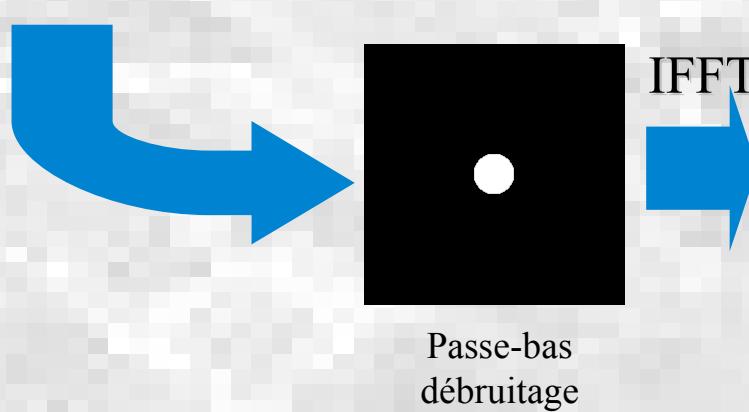
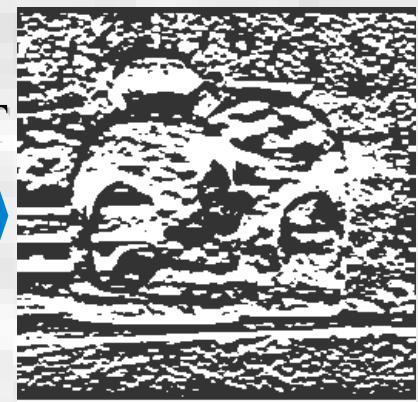
$$I_f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-x', y-y') I(x', y') dx' dy'$$

Filtrage linéaire

C La transformée de Fourier 2D



Passe-haut
détection de contours



Passe-bas
débruitage



Filtrage linéaire

C La transformée de Fourier 2D



⊗
Produit de convolution

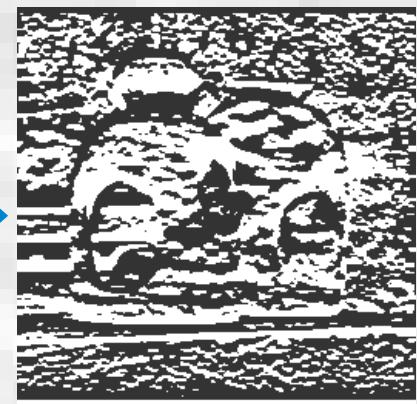
$$\begin{matrix} \mathbf{H} \\ \mathbf{H} \end{matrix}$$



Passe-haut
détection de contours



$$\begin{matrix} \mathbf{H} \\ \mathbf{M} \end{matrix}$$



Quelconque
détection de texture



$$\begin{matrix} \mathbf{H} \\ \mathbf{B} \end{matrix}$$



Passe-bas
débruitage

Expression discrète du produit de convolution.

$$I_f(i, j) = \sum_{i'=-K/2}^{K/2} \sum_{j'=-K/2}^{K/2} f(i-i', j-j') I(i', j')$$

f : Filtre.

I : Image d'origine.

I_f : Image filtrée.

I_f est une combinaison linéaire de I .

Filtrage linéaire

D Produit de convolution

I(11,10)	I(11,11)	I(11,12)	I(11,13)	I(11,14)
I(12,10)	I(12,11) f(3,3)	I(12,12) f(3,2)	I(12,13) f(3,1)	I(12,14)
I(13,10)	I(13,11) f(2,3)	I(13,12) f(2,2)	I(13,13) f(2,1)	I(13,14)
I(14,10)	I(14,11) f(1,3)	I(14,12) f(2,3)	I(14,13) f(1,1)	I(14,14)
I(15,10)	I(15,11)	I(15,12)	I(15,13)	I(15,14)

f(1,1)	f(1,2)	f(1,3)
f(2,1)	f(2,2)	f(2,3)
f(3,1)	f(3,2)	f(3,3)

Retournement



Superposition

f(3,3)	f(3,2)	f(3,1)
f(2,3)	f(2,2)	f(2,1)
f(1,3)	f(2,3)	f(1,1)

$$I_f(13,12) = I(12,11)f(3,3) + I(12,12)f(3,2) + \dots + I(14,13)f(1,1)$$

Filtrage linéaire

D Produit de convolution

Exemple : Filtre de la moyenne

Exemple : filtre 3x3

$$I_f(u,v) = [I(u-1,v-1) + \dots + I(u+1,v+1)] / 9$$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Image bruitée (SNR 2,25)

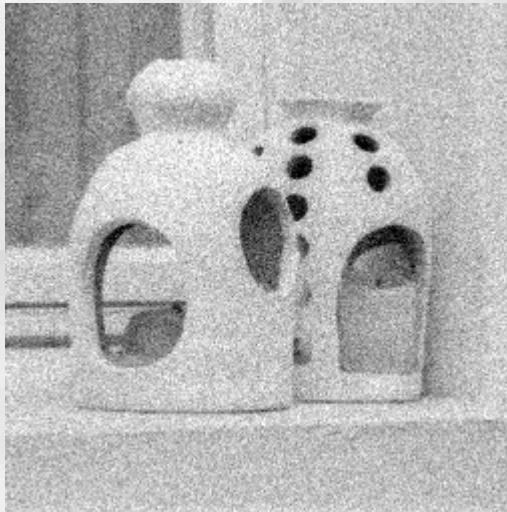


Image filtrée (SNR 9,59)

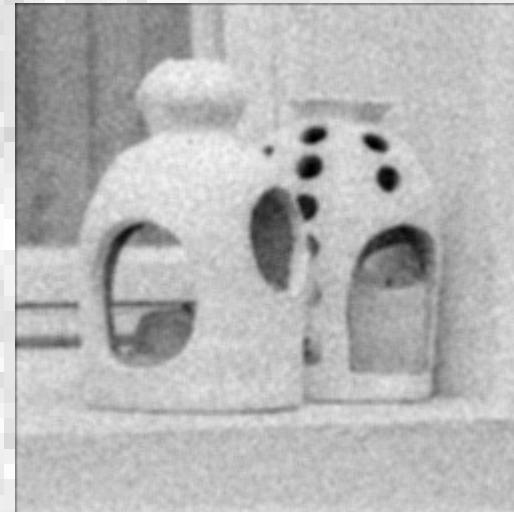


Image originale

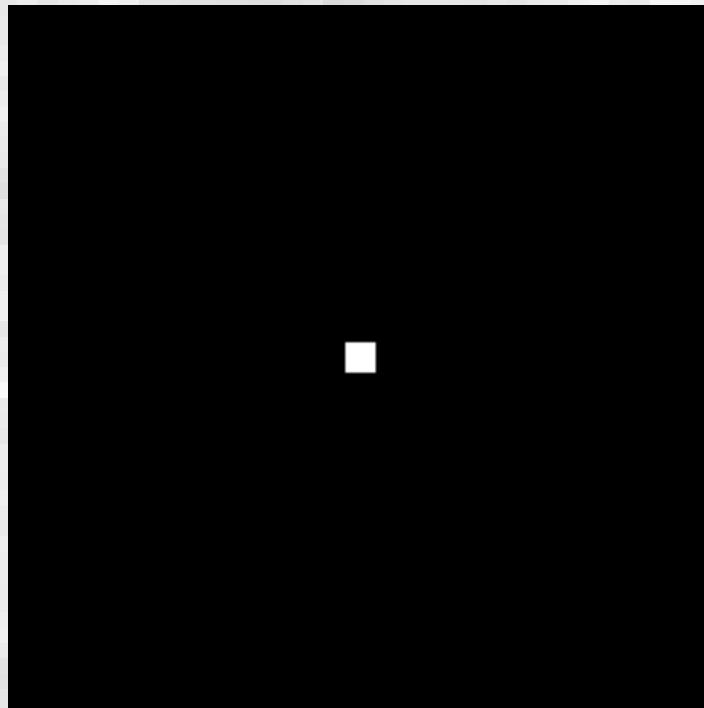


Filtrage linéaire

D Produit de convolution

Exemple : Filtre de la moyenne

Masque de 10x10

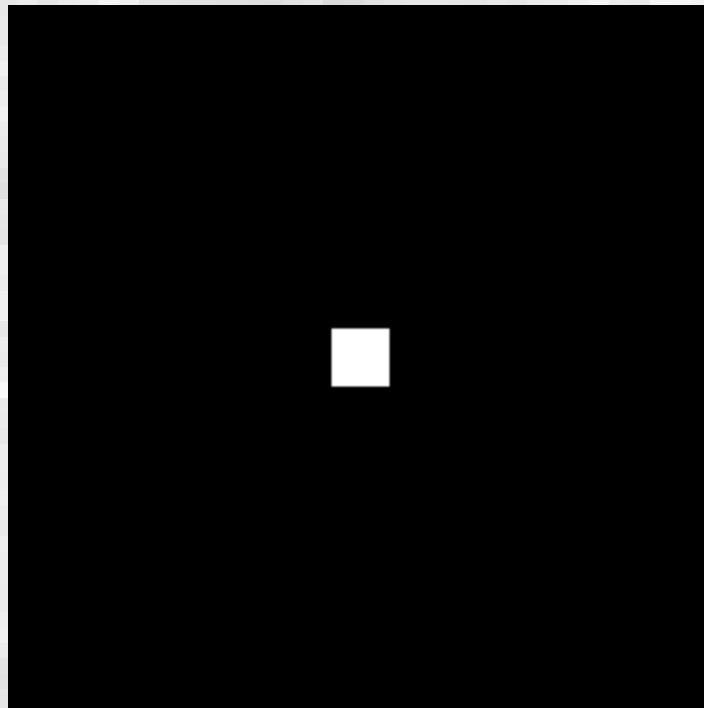


Filtrage linéaire

D Produit de convolution

Exemple : Filtre de la moyenne

Masque de 20x20

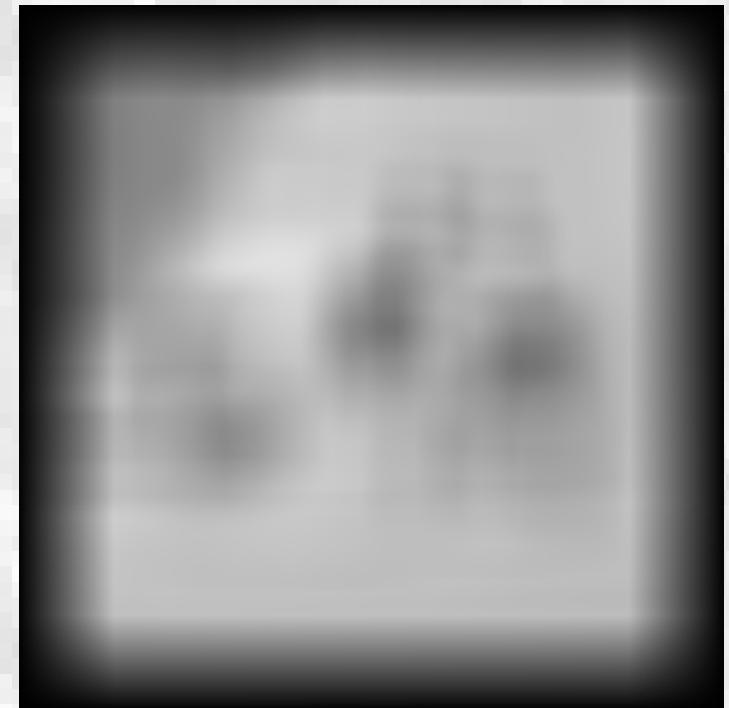
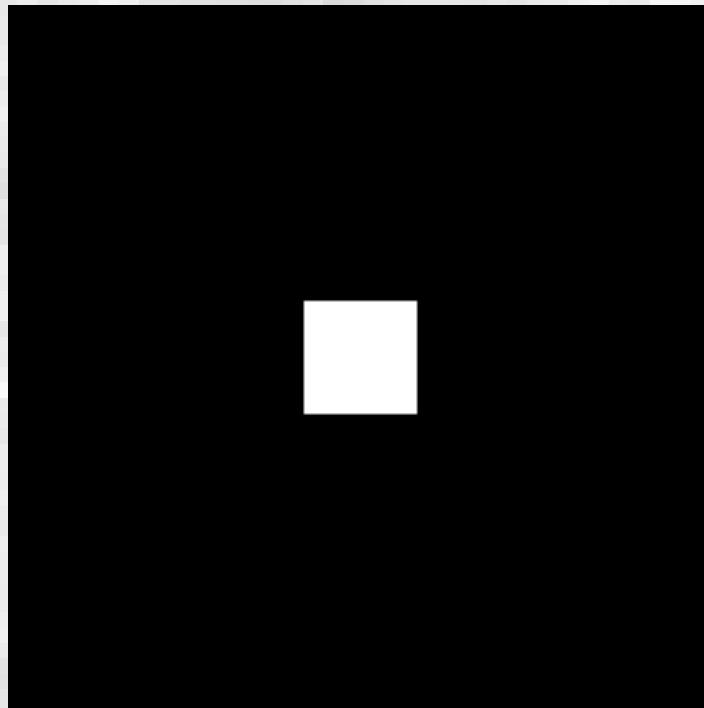


Filtrage linéaire

D Produit de convolution

Exemple : Filtre de la moyenne

Masque de 40x40



Filtrage linéaire

D Produit de convolution

Exemple : Filtre Gaussien

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

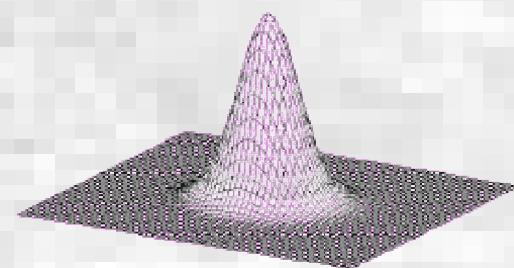


Image bruitée (SNR 2,25)

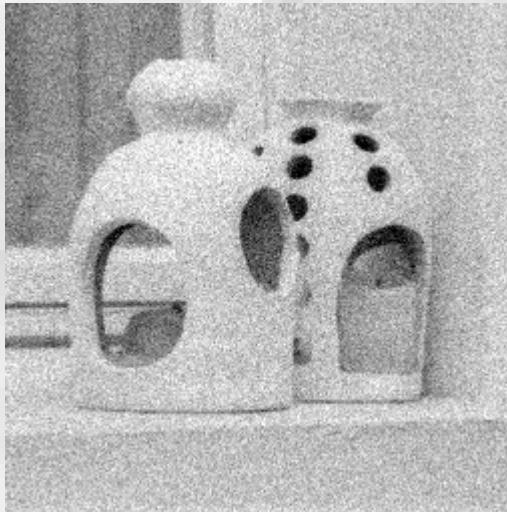


Image filtrée (SNR 11,45)

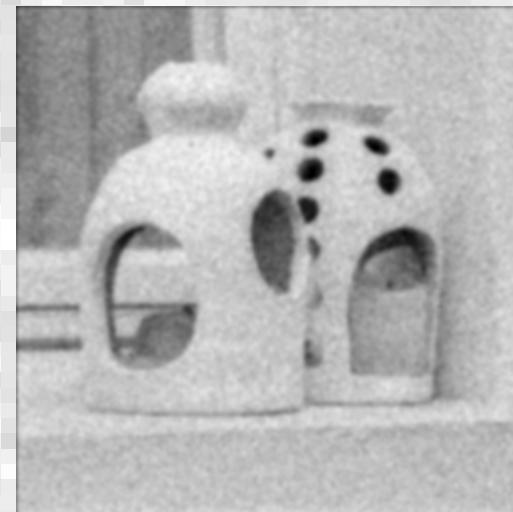


Image originale



Exemple : Filtre Gaussien

Problème : le support est infini.

On choisit généralement une taille de support de 4σ à 6σ .

4σ correspond à 85%
 6σ correspond à 99,9% } de l'énergie du filtre.

Avantage : Le filtre est séparable

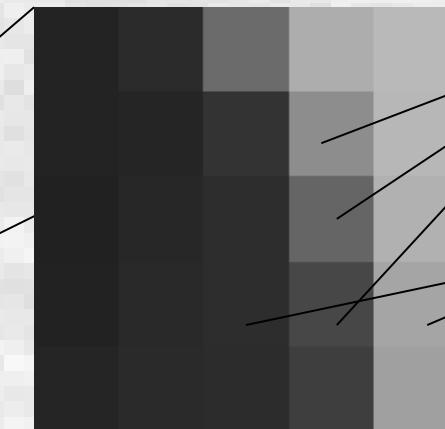
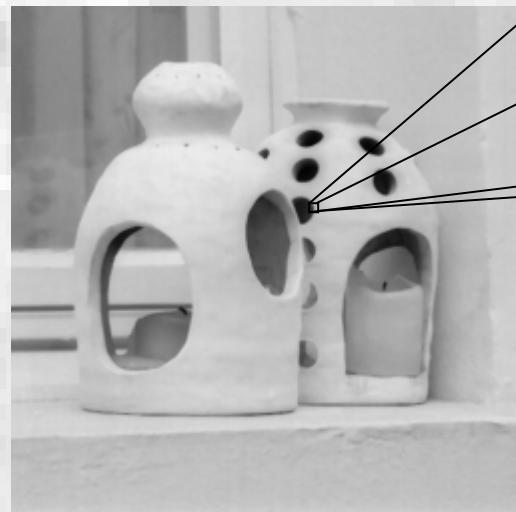
$$TF(H) = \exp\left(\frac{\sigma^2(\omega_x^2 + \omega_y^2)}{2}\right) = \exp\left(\frac{\sigma^2 \omega_x^2}{2}\right) \exp\left(\frac{\sigma^2 \omega_y^2}{2}\right)$$

$$TF(I_f) = TF(I).TF(H_x).TF(H_y) \quad \Rightarrow \quad I_f = I \otimes H_x \otimes H_y$$

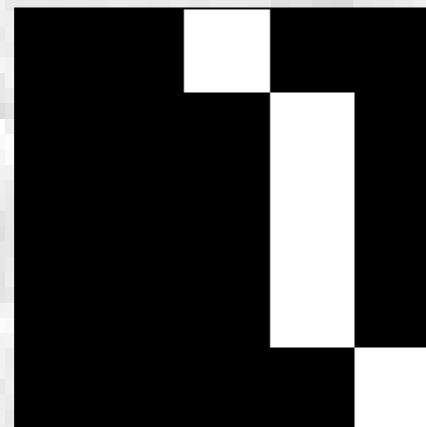
Détection de contour

Détection de contour

A Principe

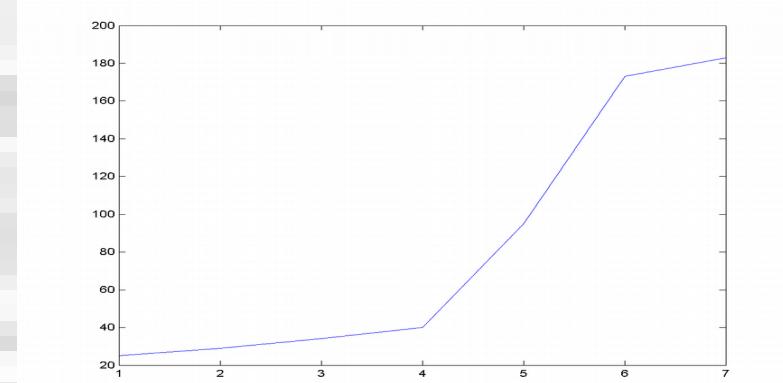
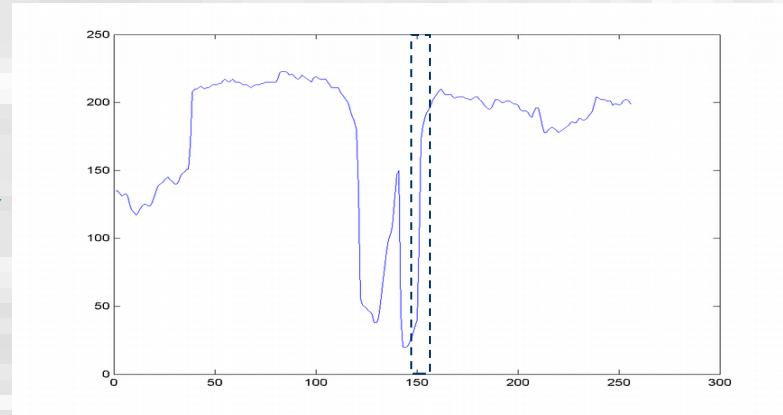


Pixels contour → 1
Pixels objet → 0



Détection de contour

A Principe



Détection de contour

A Principe

Détection des variations importantes du NdG.

Deux approches :

Maxima du gradient (dérivée première)

Zéros du laplacien (dérivée seconde)

Détection de contour

B Approche gradient

$$I_x(x, y) = \frac{\partial I(x, y)}{\partial x}$$

$$I_y(x, y) = \frac{\partial I(x, y)}{\partial y}$$

Norme :

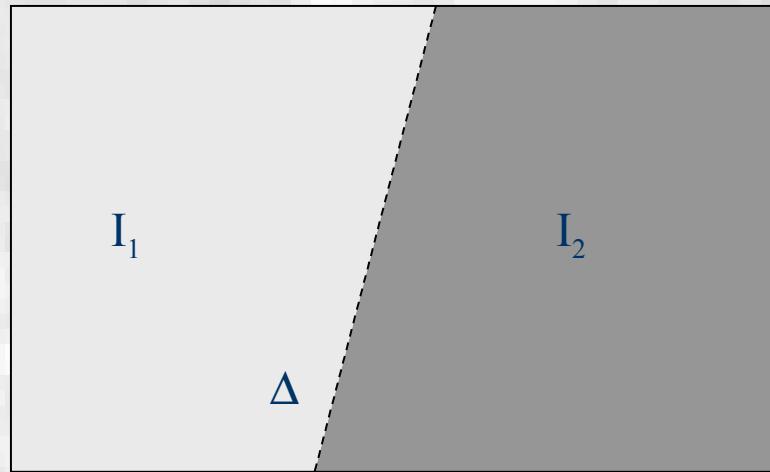
$$G(x, y) = \sqrt{I_x(x, y)^2 + I_y(x, y)^2}$$

Orientation :

$$\varphi(x, y) = \operatorname{Arctg} \left(\frac{I_y(x, y)}{I_x(x, y)} \right)$$

Détection de contour

B Approche gradient



$$\Delta : -x \cdot \sin \theta + y \cdot \cos \theta + r = 0$$

$$I(x,y) = I_1 + (I_2 - I_1) \cdot u(-x \cdot \sin \theta + y \cdot \cos \theta + r)$$

$$u(t) = \begin{cases} 1 & \text{si } t \geq 0 \\ 0 & \text{si } t < 0 \end{cases} \quad u'(t) = \delta(t)$$

$$G_x = \cos(\theta + \pi/2)(I_2 - I_1)\delta(-x \cdot \sin \theta + y \cdot \cos \theta + r)$$

$$G_y = \sin(\theta + \pi/2)(I_2 - I_1)\delta(-x \cdot \sin \theta + y \cdot \cos \theta + r)$$

Le gradient est nul partout sauf sur le contour.

$$\text{Norme : } |G| = I_2 - I_1$$

$$\text{Direction : } \phi = \theta + \pi/2$$

Détection de contour

B Approche gradient

Approximation par différences finies

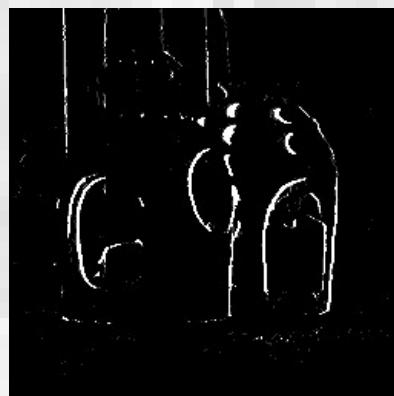
$$I_i(i, j) = \frac{\Delta I}{\Delta i} = I(i+1, j) - I(i, j)$$



-



=



$$I_j(i, j) = \frac{\Delta I}{\Delta j} = I(i, j+1) - I(i, j)$$



-



=



Détection de contour

B Approche gradient

Approximation par différences finies

$$I_i(i, j) = \frac{\Delta I}{\Delta i} = I(i+1, j) - I(i, j)$$

$$I_j(i, j) = \frac{\Delta I}{\Delta j} = I(i, j+1) - I(i, j)$$

⇒ Convolution

$$I_i(i, j) = I(i, j) \otimes \begin{array}{|c|c|}\hline -1 & 1 \\ \hline\end{array}$$

D_x

$$I_j(i, j) = I(i, j) \otimes \begin{array}{|c|}\hline -1 \\ \hline 1 \\ \hline\end{array}$$

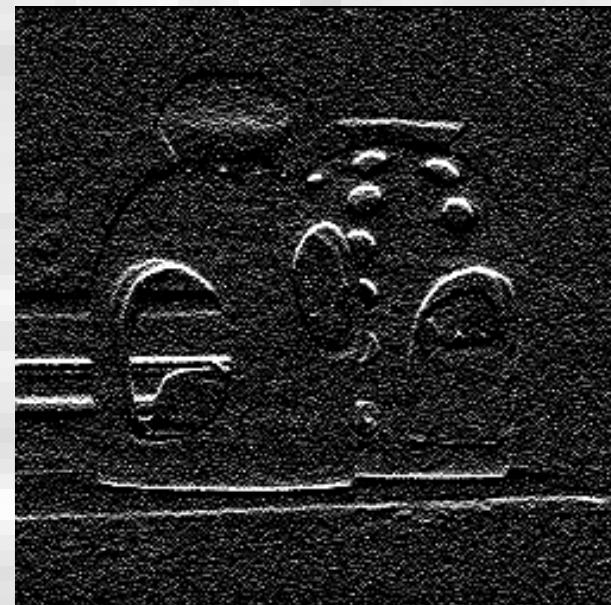
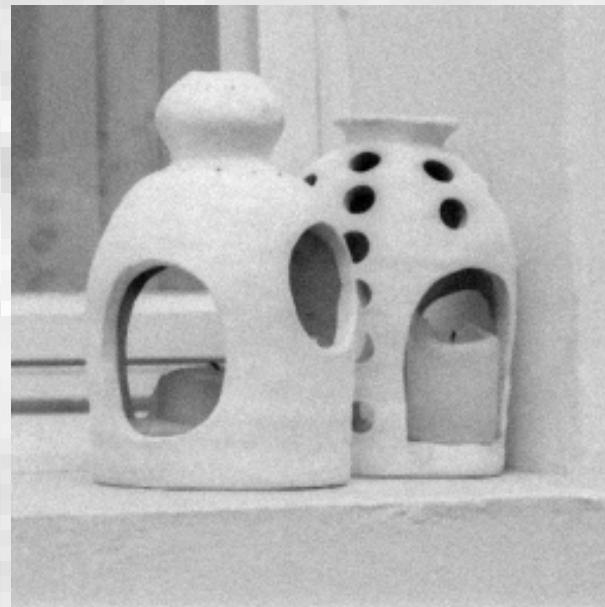
D_y

Problème : Le bruit est amplifié.

Détection de contour
B Approche gradient

Différences finies : Influence du bruit.

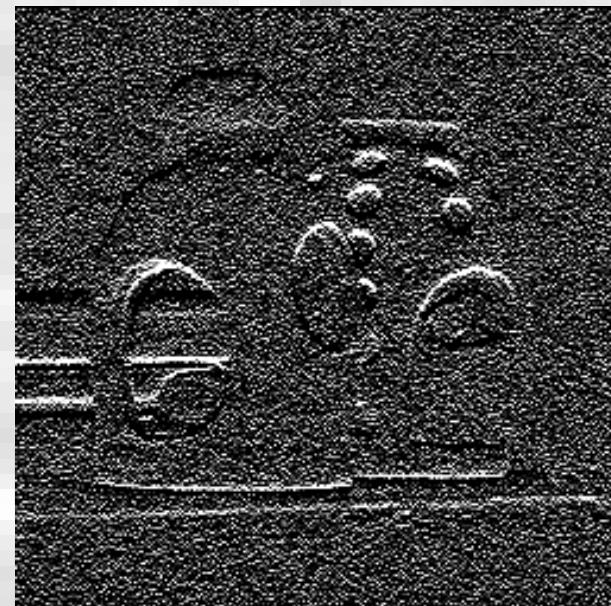
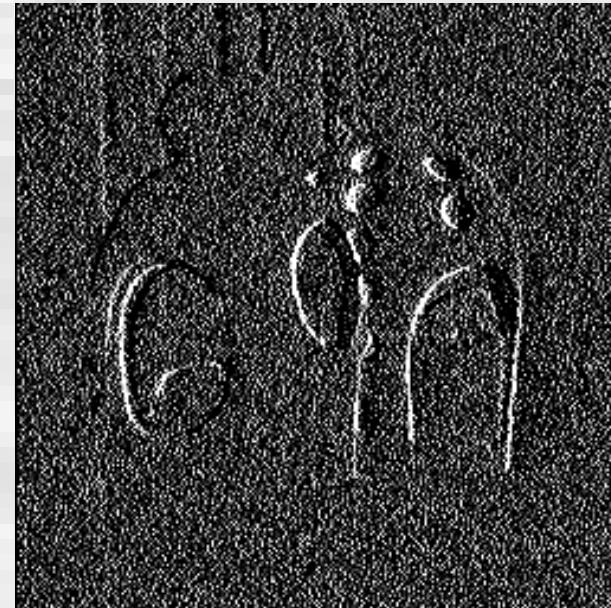
Écart type : 5



Détection de contour
B Approche gradient

Différences finies : Influence du bruit.

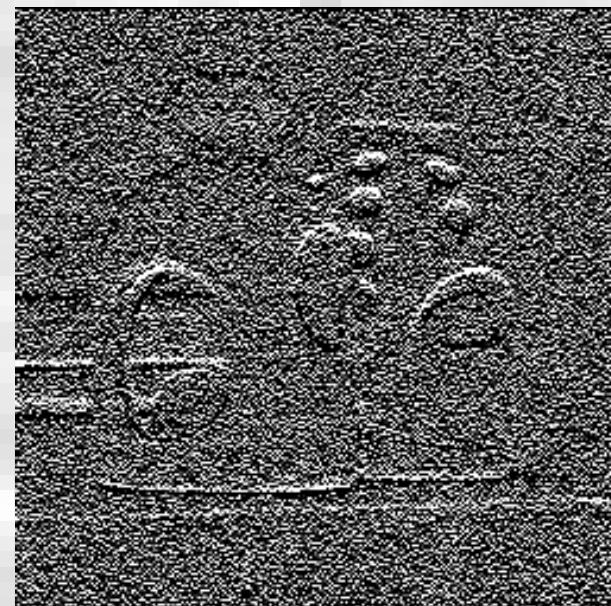
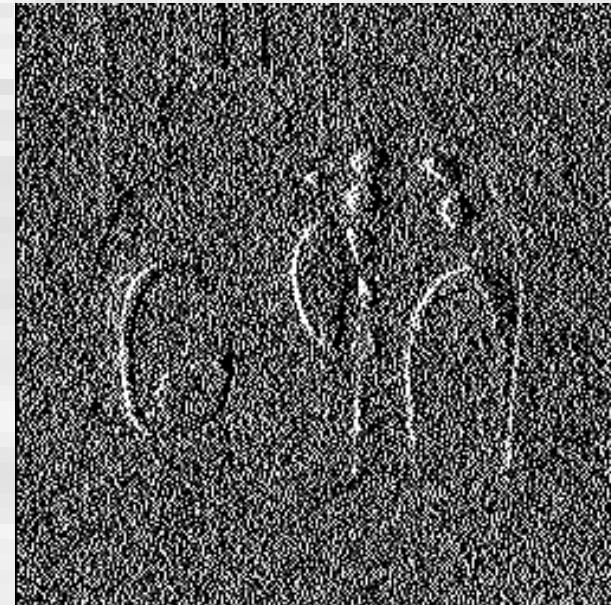
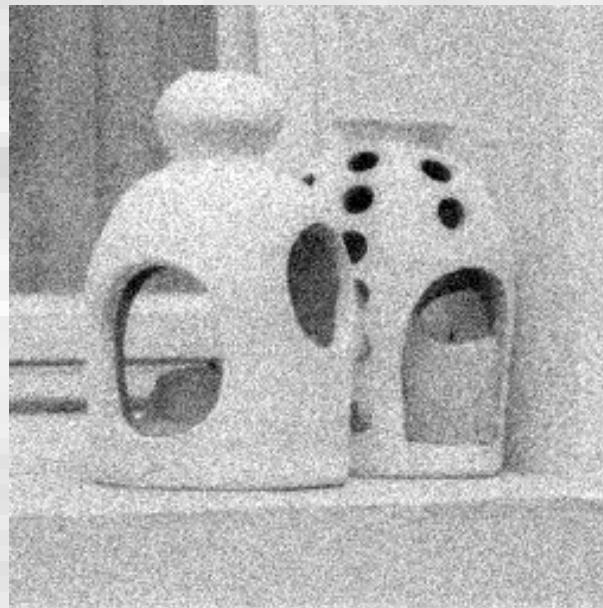
Écart type : 10



Détection de contour
B Approche gradient

Différences finies : Influence du bruit.

Écart type : 20



Détection de contour

B Approche gradient

Solution : Combiner un filtrage de l'image à la dérivation.

$$I_x = \frac{\partial I_f}{\partial x} = \frac{\partial(I \otimes H)}{\partial x} = I \otimes H \otimes D_x$$

$$I_y = \frac{\partial I_f}{\partial y} = \frac{\partial(I \otimes H)}{\partial y} = I \otimes H \otimes D_y$$

Détection de contour

B Approche gradient

Solution : Combiner un filtrage de l'image à la dérivation.

$$I_x = \frac{\partial I_f}{\partial x} = \frac{\partial(I \otimes H)}{\partial x} = I \otimes (H \otimes D_x)$$

$$I_y = \frac{\partial I_f}{\partial y} = \frac{\partial(I \otimes H)}{\partial y} = I \otimes (H \otimes D_y)$$

Détection de contour
B Approche gradient

Opérateurs de Prewitt ($c = 1$) et Sobel ($c = 2$)

$$H \otimes D_x = f_x =$$

1	0	-1
c	0	$-c$
1	0	-1

$$H \otimes D_y = f_y =$$

1	c	1
0	0	0
-1	$-c$	-1

Remarques :

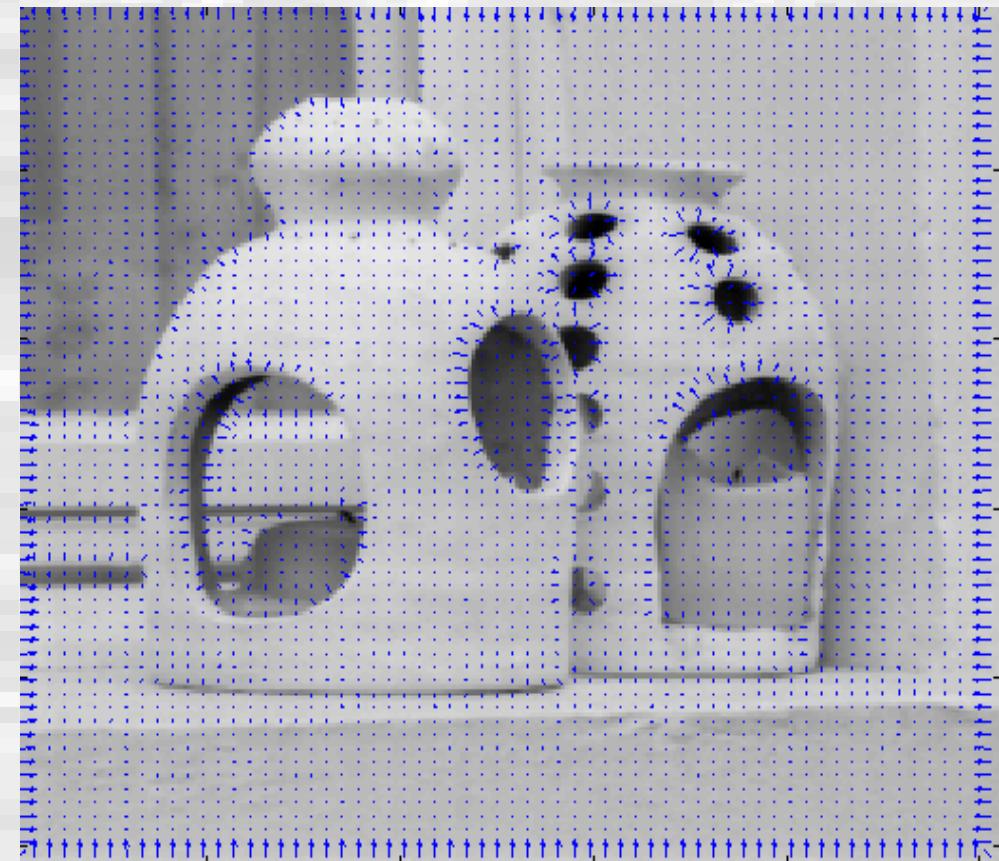
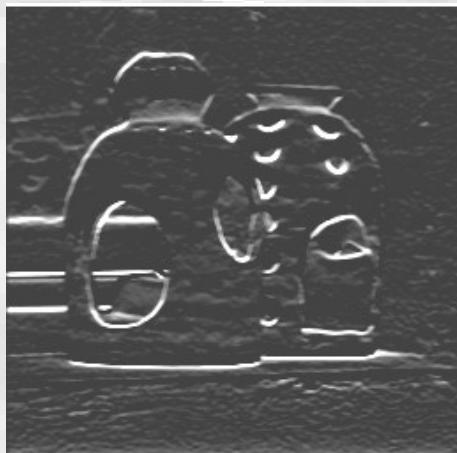
- La dérivation et le filtrage passe-bas sont réalisés dans des directions complémentaires.
- Les filtres sont séparables :

$$f_x = \begin{bmatrix} 1 \\ c \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad f_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & c & 1 \end{bmatrix}$$

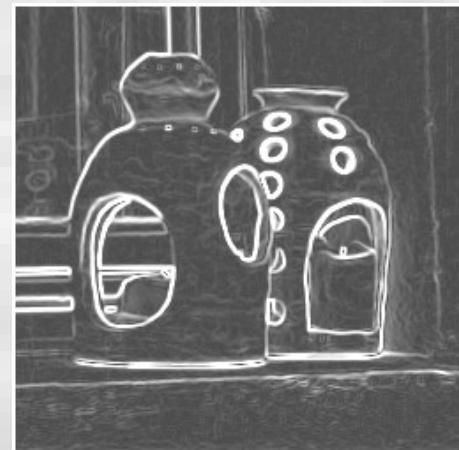
Réduction du temps de calcul de p^2 à $2p$ opérations.

Détection de contour

B Approche gradient



Exemple :
masques de
Prewitt



Détection de contour

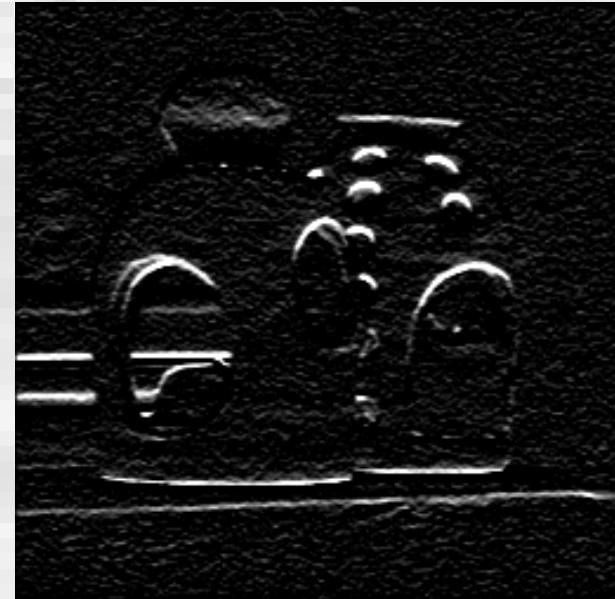
B Approche gradient

Méthode de Sobel : Influence du bruit.

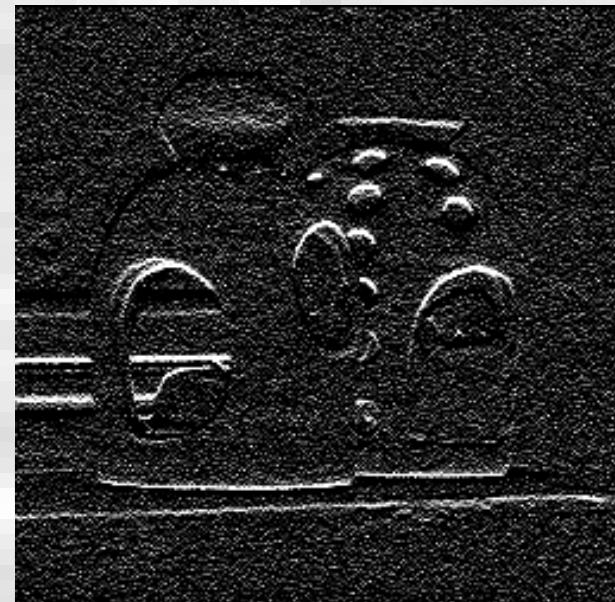
Écart type : 5



Sobel



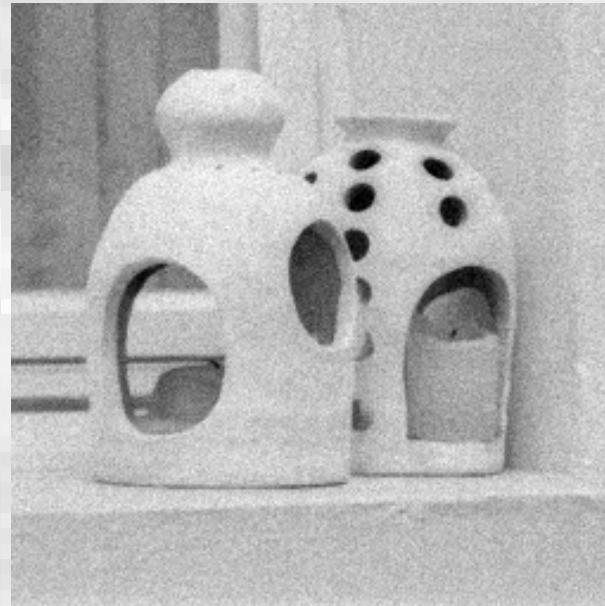
Diff. finies



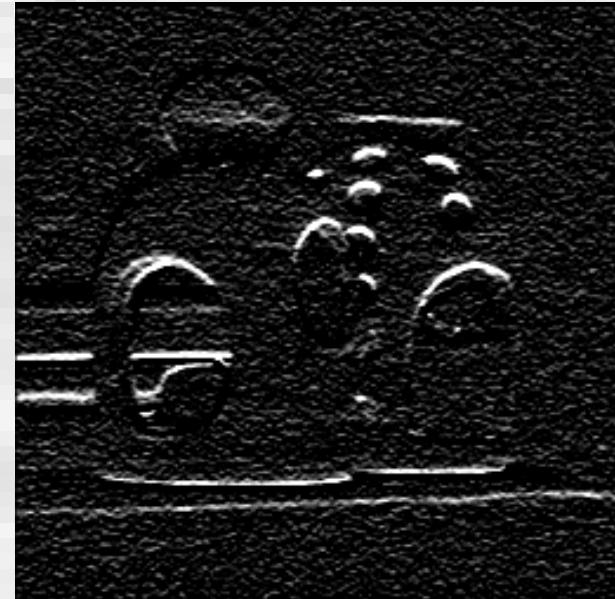
Détection de contour
B Approche gradient

Méthode de Sobel : Influence du bruit.

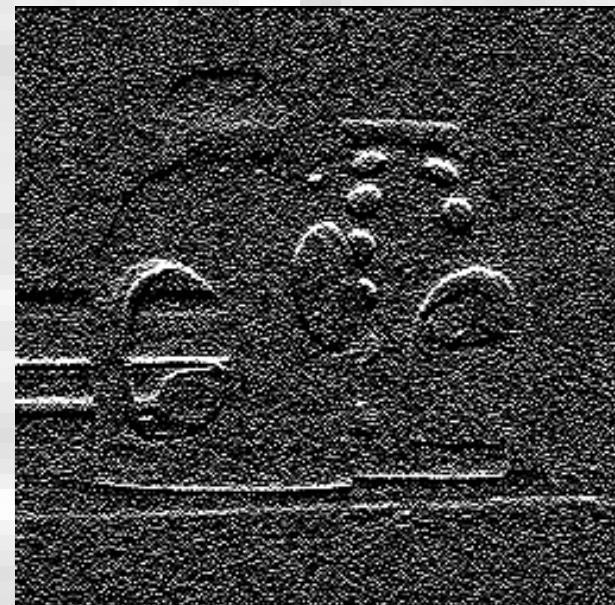
Écart type : 10



Sobel



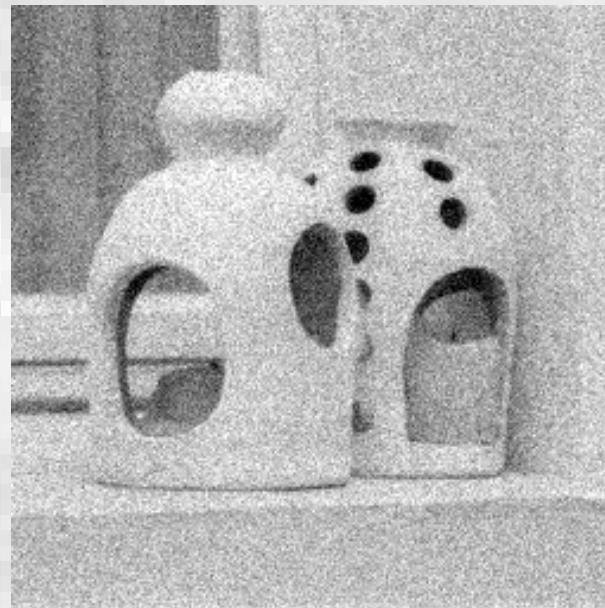
Diff. finies



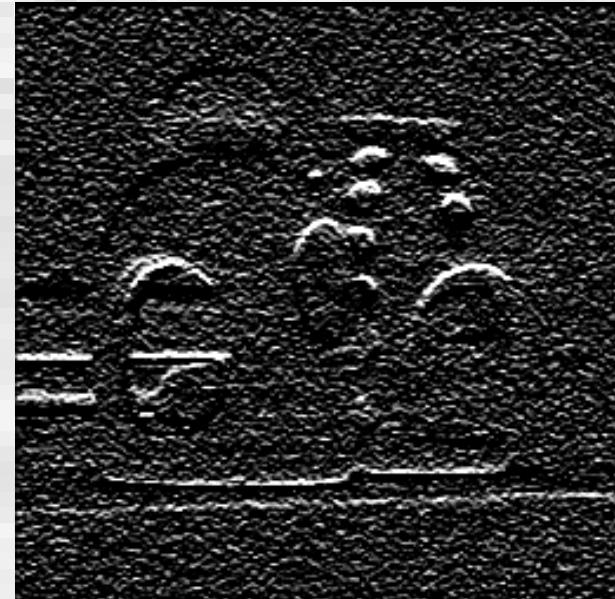
Détection de contour
B Approche gradient

Méthode de Sobel : Influence du bruit.

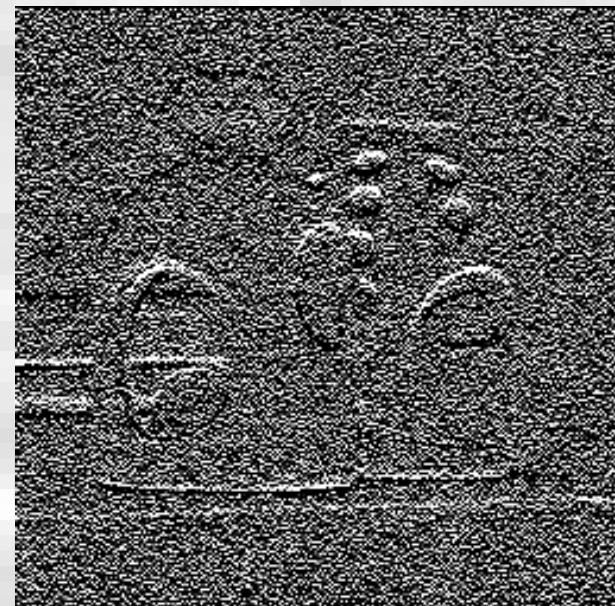
Écart type : 20



Sobel



Diff. finies



Opérateurs de Kirsch

On définit un jeu de masques directionnels :

$$H_0 = \begin{matrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{matrix}$$

$$H_1 = \begin{matrix} c & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -c \end{matrix}$$

$$H_2 = \begin{matrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{matrix}$$

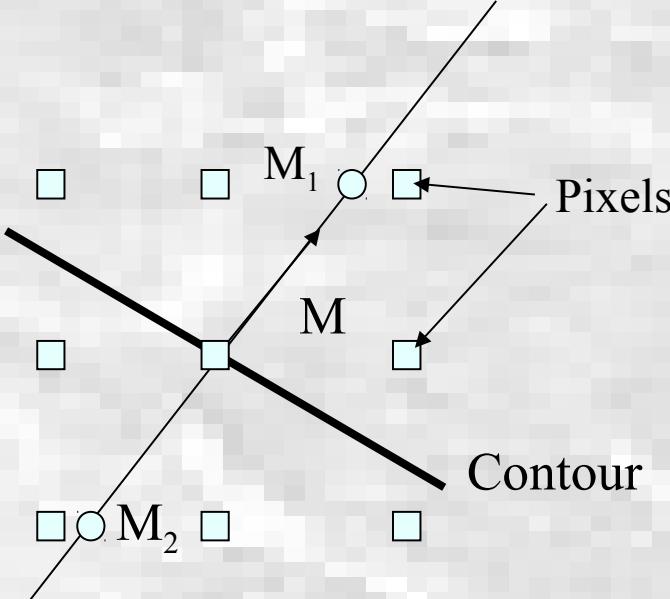
• • •

Les autres masques sont obtenus par rotation de $\pi/4$.

Le gradient retenu est donné par : $\text{Max} (H_i \otimes I)$

L'orientation est donnée par : $\arg.\text{Max} (H_i \otimes I).\pi/4$

Localisation du contour : Algorithme de Canny



Etape 1 : Extraction des maxima locaux

$|G_r|$ en $M(i,j)$ à $|G_r|$ en M_1 et M_2

M est un maximum local du gradient si :

$|G_r(M)| > |G_r(M_1)|$ et $|G_r(M)| \geq |G_r(M_2)|$

$|G_r(M_1)|$ et $|G_r(M_2)|$ sont calculées par interpolation linéaire.

Etape 2 : Seuillage à hystérésis

On définit un seuil haut S_h et un seuil bas S_b .

Ne sont retenus que les maxima vérifiant l'une des 2 conditions :

$$|G(M)| > S_h$$

et

$$|G(M)| > S_b \text{ et } \text{ il existe } N \text{ connexe à } M \text{ tq : } |G(N)| > S_h$$

Détection de contour

C Approche laplacien

Laplacien scalaire : $\Delta I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$

Il peut donc être calculé à partir des opérateurs de dérivation.

Exemple :

1
-1

et

1	-1
---	----

Conduisent à :

0	1	0
1	-4	1
0	1	0

Détection de contour

C Approche laplacien

La localisation du contour est réalisée de la façon suivante :

- Création d'une image de polarité:

$$Ip(M) = \begin{cases} 0 & \text{si } \Delta(M) > 0 \\ 1 & \text{si } \Delta(M) \leq 0 \end{cases}$$

- Création d'une image Iz des passages par zéro :

$$Iz(M) = \begin{cases} 1 & \text{si } M \text{ correspond à une transition (0 1) ou (1 0) de } Ip \\ 1 & \text{si } \Delta(M) \leq 0 \end{cases}$$

Le résultat est ensuite affiné en éliminant les points de passage par zéro associés à un gradient faible.

Détection de contour

D Approche globale

Approches locales : On détecte les pixels isolément les uns des autres.

Problèmes : Contours non fermés.

Approche globales : On recherche le contour complet.

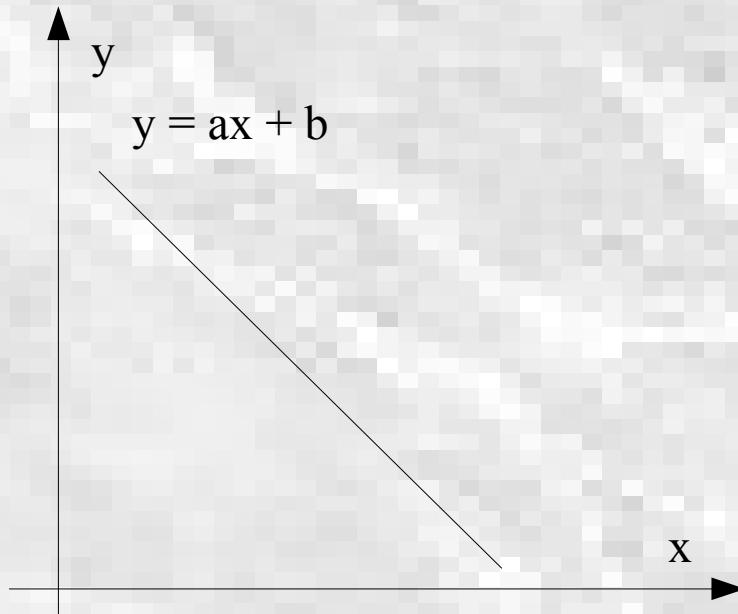
Méthode : Transformée de Hough

Transformée de Hough

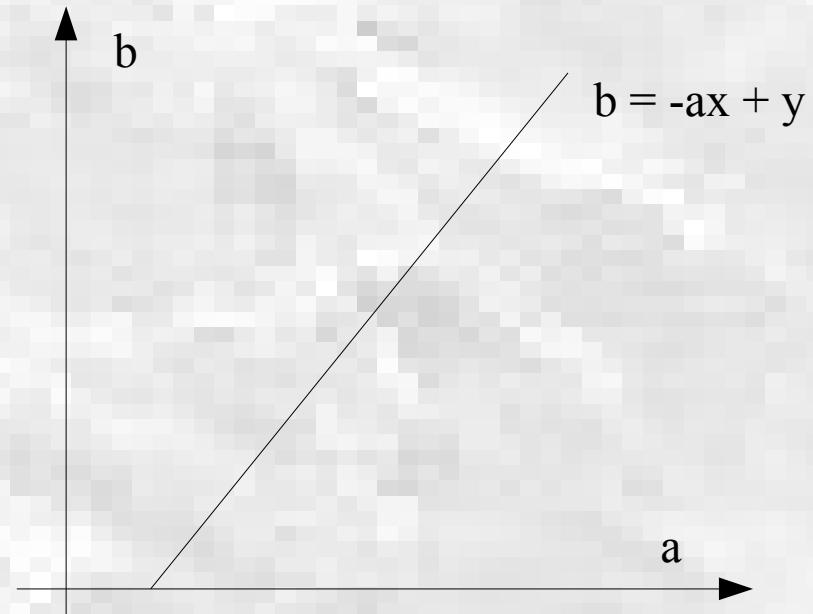
Principe : Dualité variables - paramètres

L'équation d'une droite dans le plan est : $y = a x + b$

Paramètres : (a, b) Variables : (x, y)

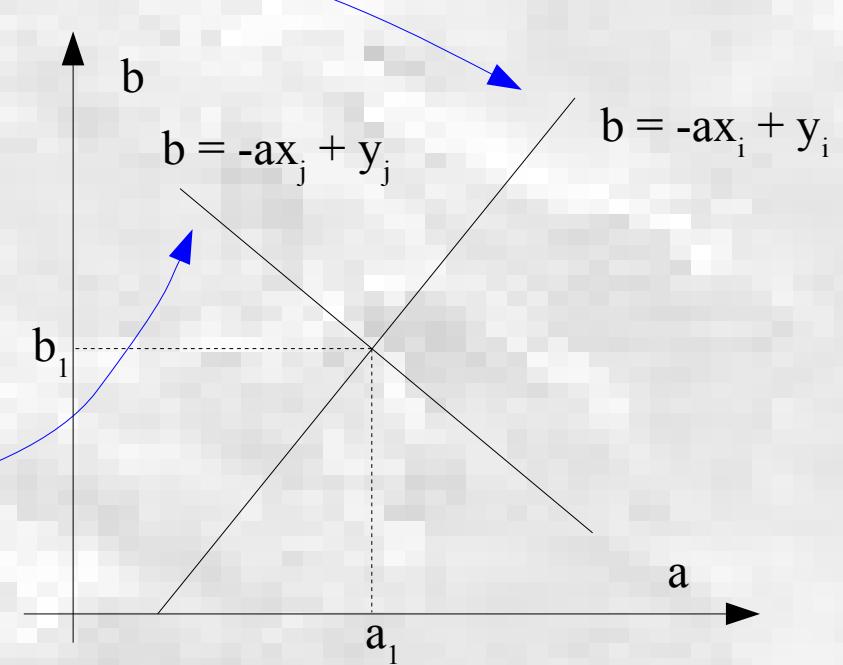
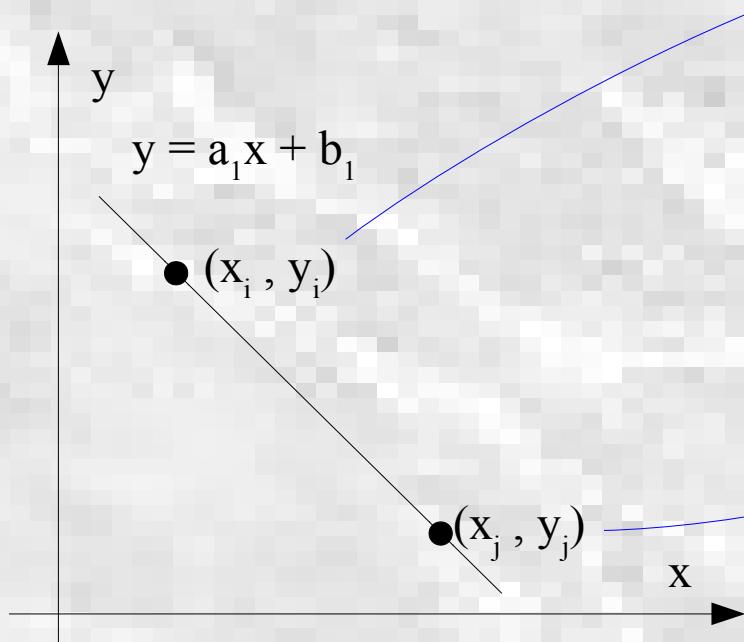


Paramètres : (x, y) Variables : (a, b)



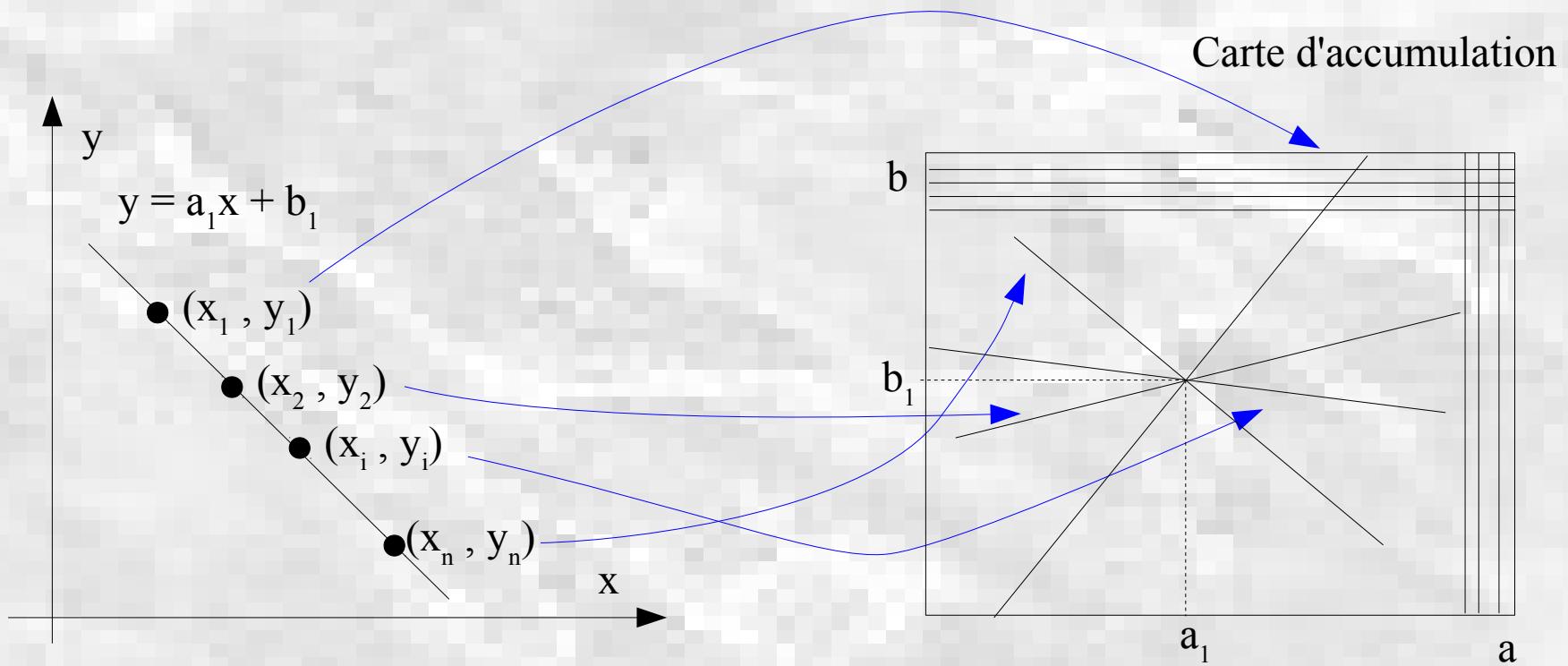
Transformée de Hough

Tous les points (x, y) d'une droite vérifient l'équation de la droite : $y = ax + b$



Transformée de Hough

Utilisation d'une carte d'accumulation (matrice (a,b))



Transformée de Hough

Algorithme de la transformée de Hough :

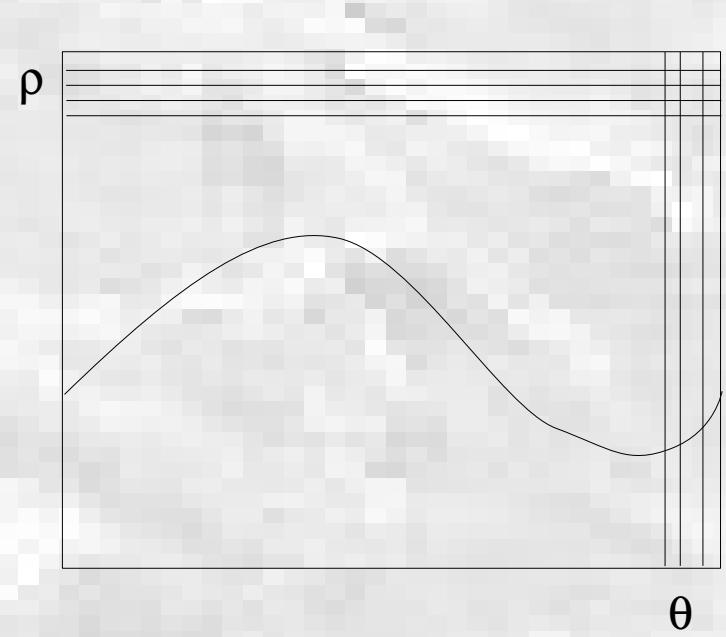
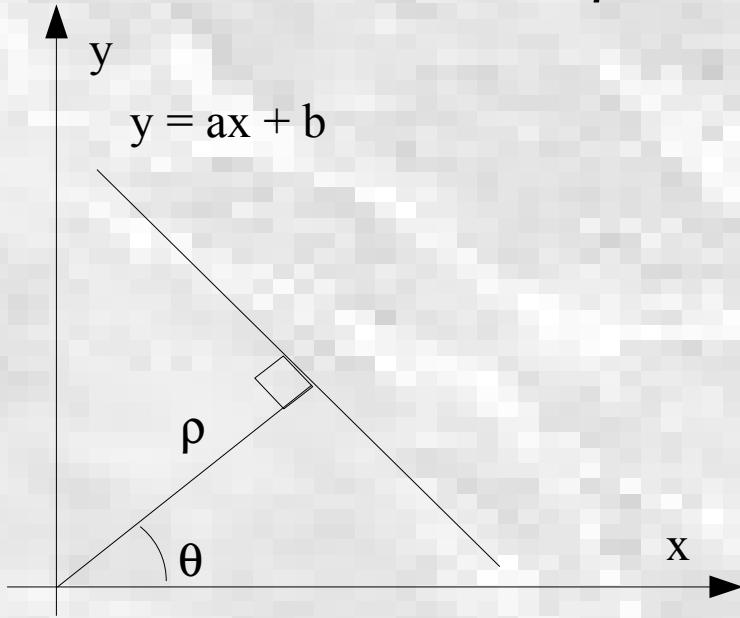
1. Déetecter les pixels des contours de l'image (Canny, Laplacien, ...).
2. Pour chaque point de contour tracer la droite dans la carte d'accumulation.
3. DéTECTER les maximums de la carte d'accumulation, ils correspondent aux droites de l'image.

Transformée de Hough

Problème : Pour les droites verticales : $a = \infty$

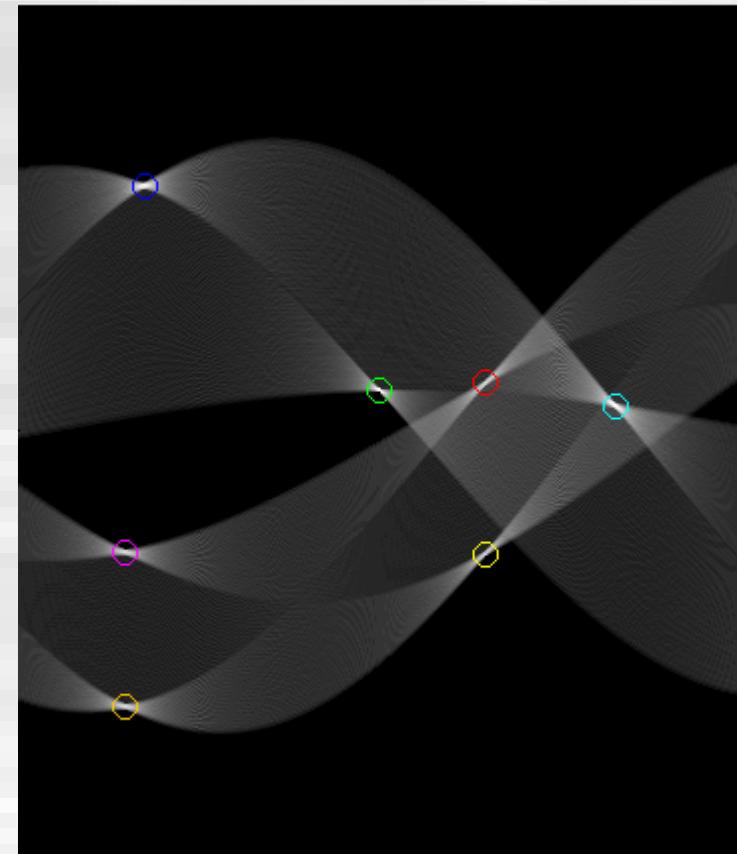
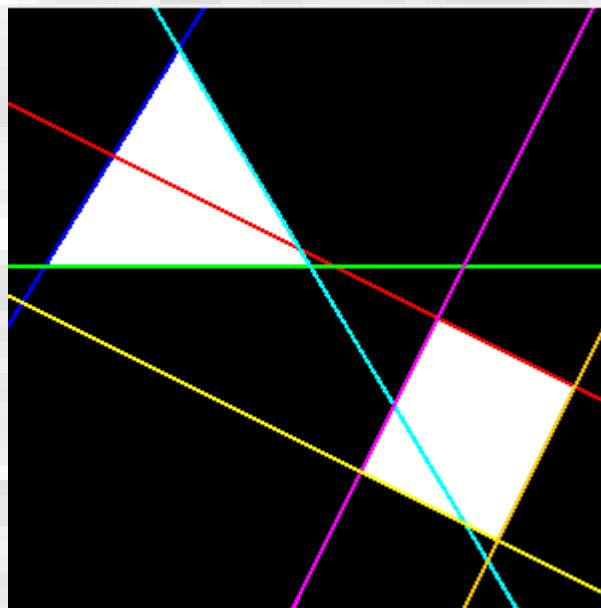
Solution : Représentation polaire : (ρ, θ)

$$\rho = x \cos \theta + y \sin \theta$$



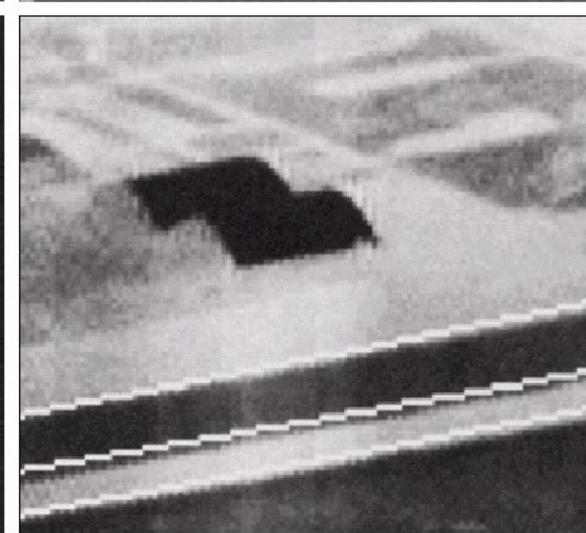
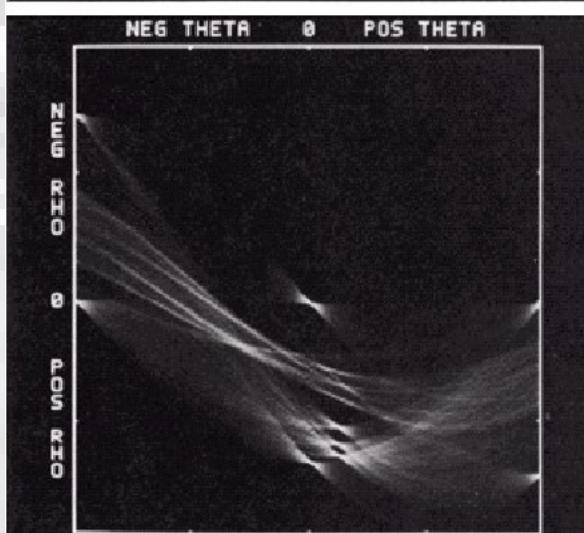
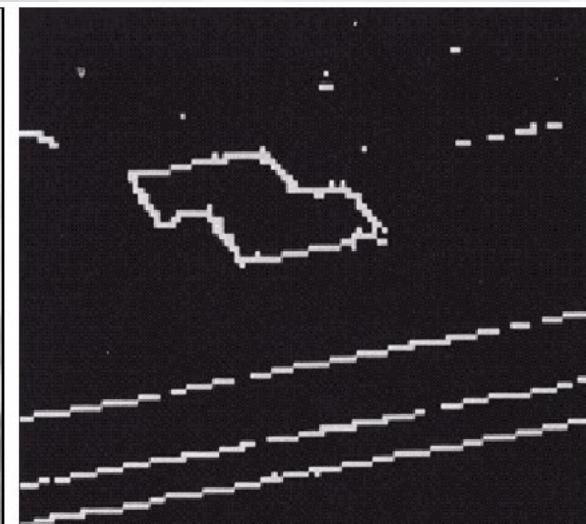
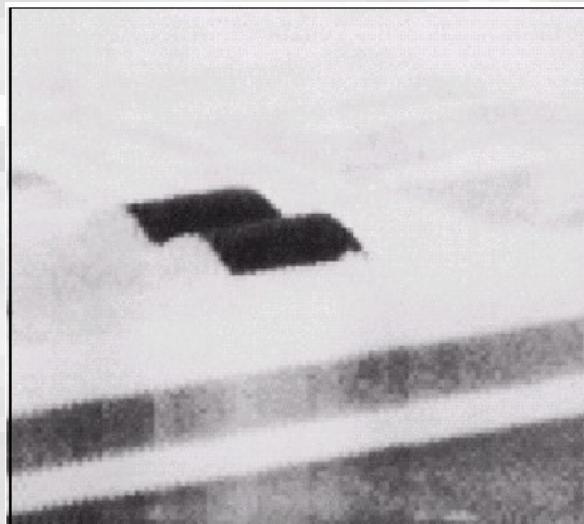
Détection de contour
D Approche globale

Transformée de Hough



Détection de contour
D Approche globale

Transformée de Hough



Segmentation en régions

Segmentation en régions A Principe

Principe : La segmentation consiste à attribuer à chaque pixel un label (valeur symbolique) qui correspond à l'objet auquel il appartient.

Exemple 1 : Cas d'une image NdG à deux classes (objet + fond).



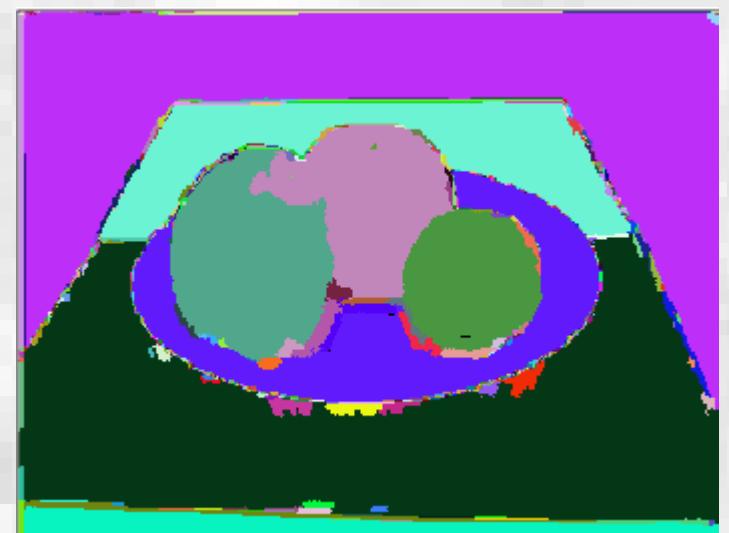
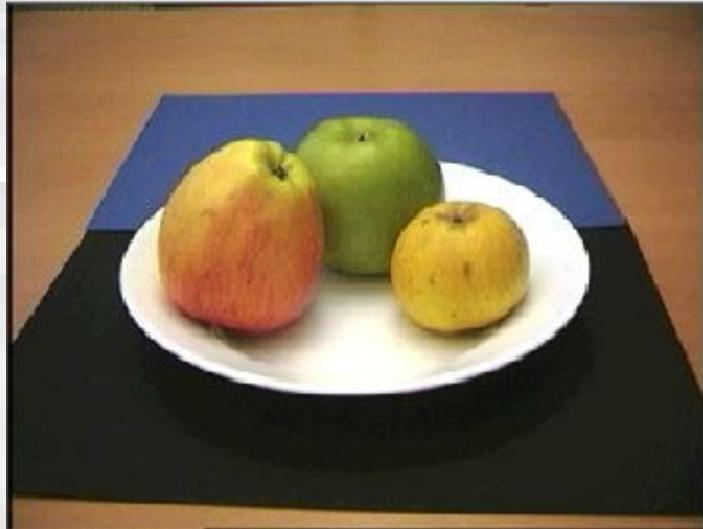
Pixel d'une pièce → 0

Pixel du fond → 1

Segmentation en régions A Principe

Principe : La segmentation consiste à attribuer à chaque pixel un label (valeur symbolique) qui correspond à l'objet auquel il appartient.

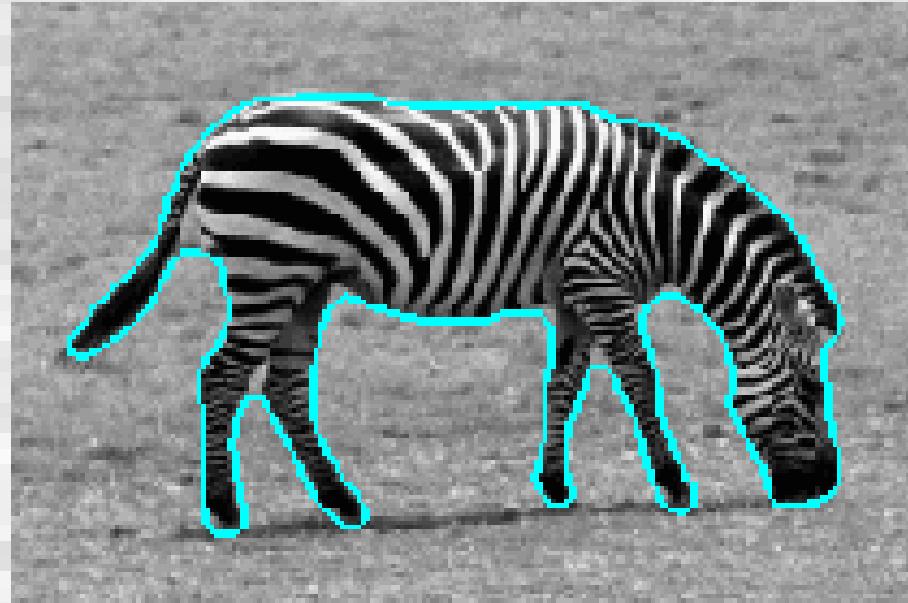
Exemple 2 : Cas d'une image couleur plus de deux classes.



Segmentation en régions A Principe

Principe : La segmentation consiste à attribuer à chaque pixel un label (valeur symbolique) qui correspond à l'objet auquel il appartient.

Exemple 3 : Cas d'une image texturée (couleur ou NdG).

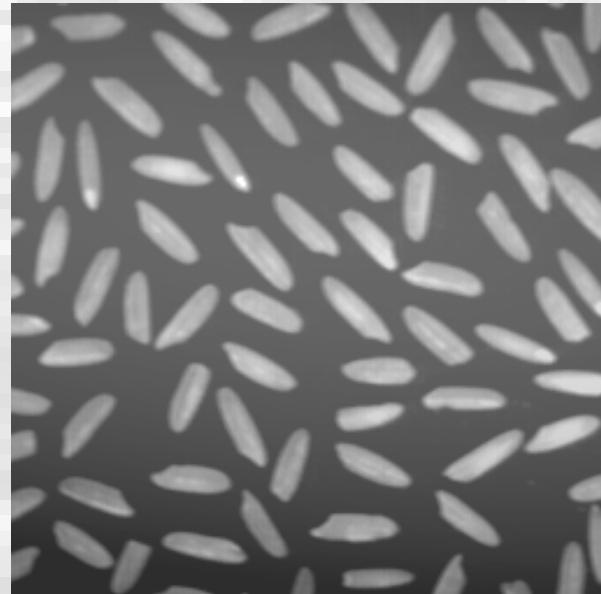


Segmentation en régions

B Seuillage global

On définit un seuil S .

$$I_b(i,j) = \begin{cases} 1 & \text{si } I(i,j) > S \\ 0 & \text{si } I(i,j) \leq S \end{cases}$$



Seuil : 100



Seuil : 150



Seuil : 200

Segmentation en régions B Seuillage global

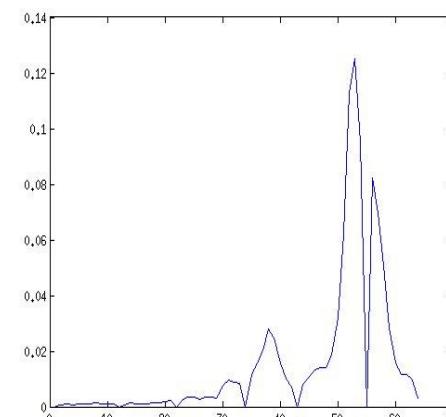
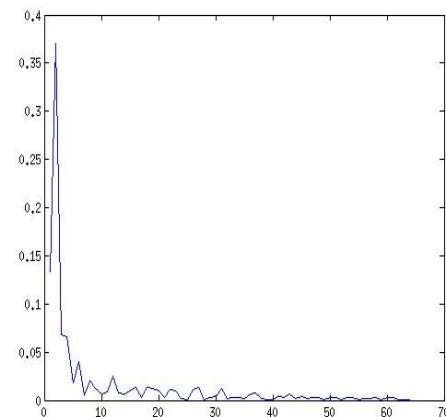
Histogramme

Définition : Graph donnant la fréquence d'apparition d'un NdG.

Image sombre

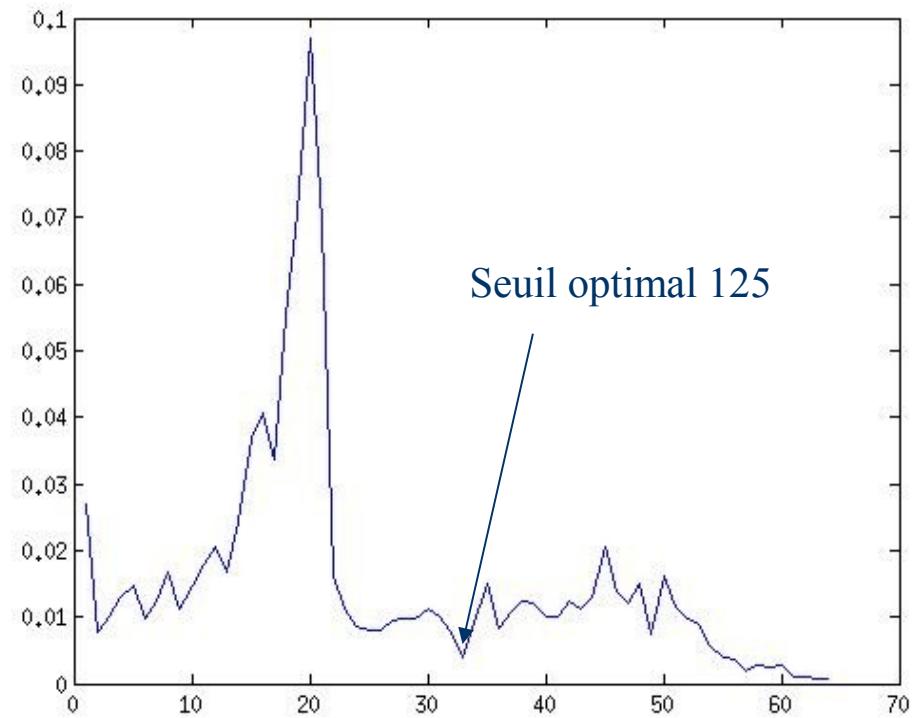
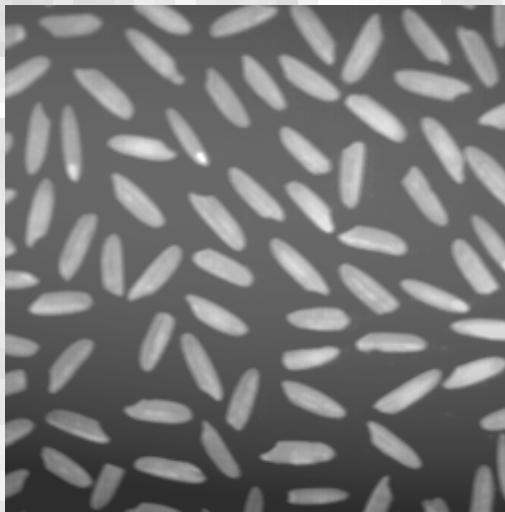


Image claire



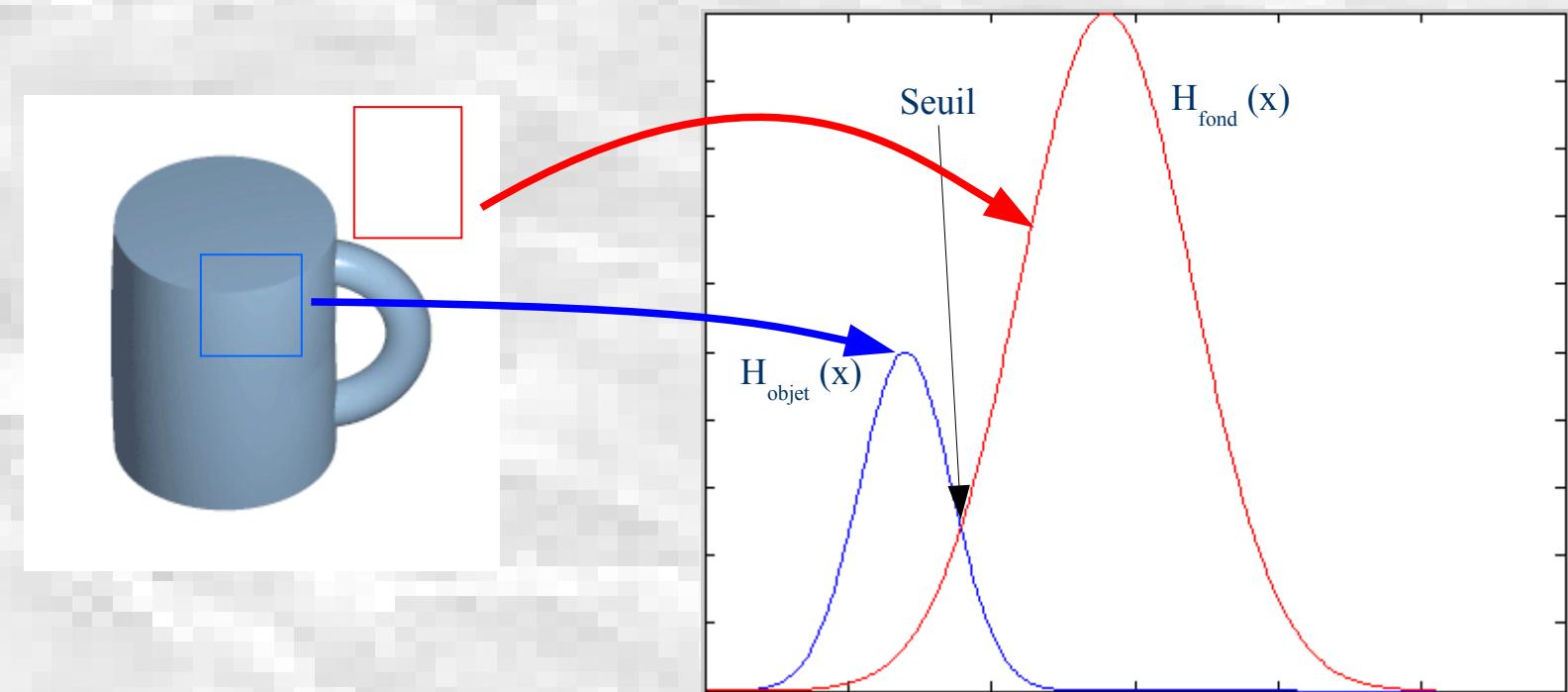
Segmentation en régions
B Seuillage global

Histogramme



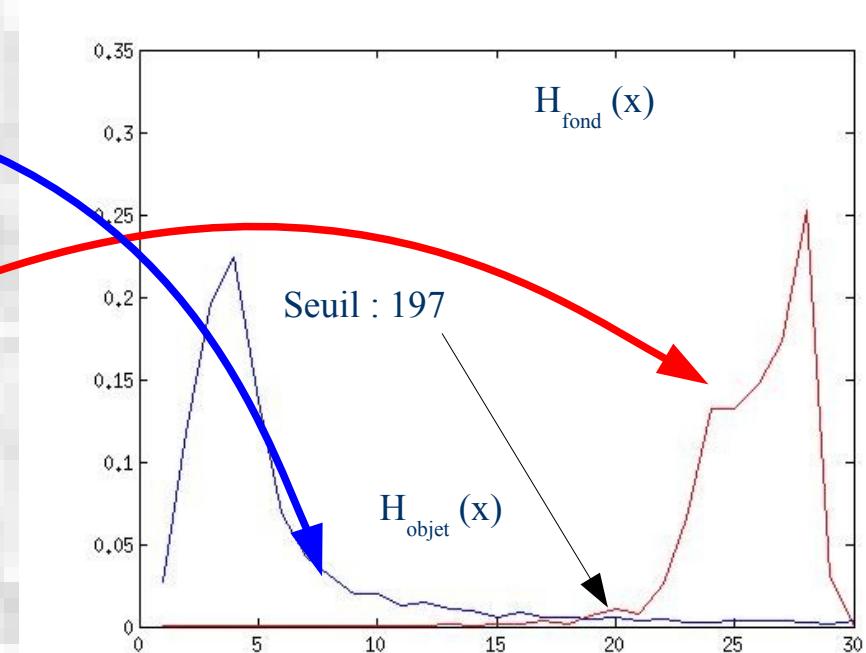
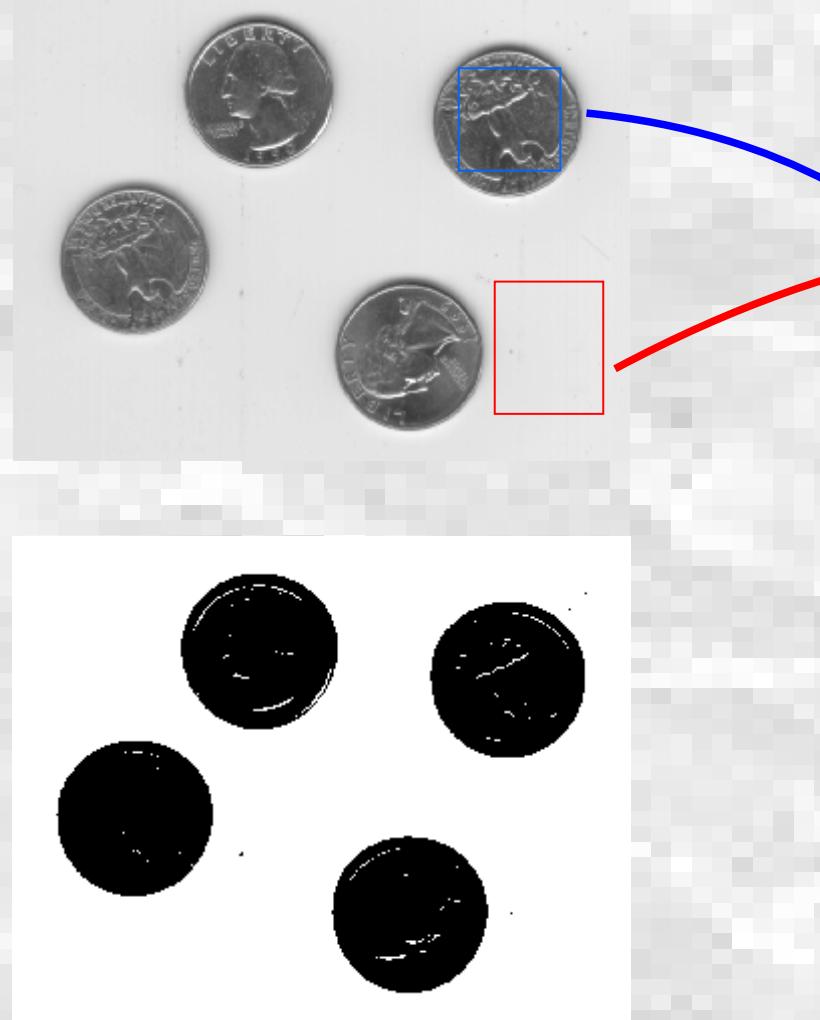
Segmentation en régions
B Seuillage global

Approche supervisée



Segmentation en régions
B Seuillage global

Approche supervisée



Approche non supervisée

Algorithme d'Otsu

On balaie toutes les valeurs de seuil possible T .
Pour chaque T on calcul :

Les moyennes :

$$\mu_1 = \frac{1}{T} \sum_{i=0}^{T-1} h(i) \quad \mu_2 = \frac{1}{256-T} \sum_{i=T}^{255} h(i)$$

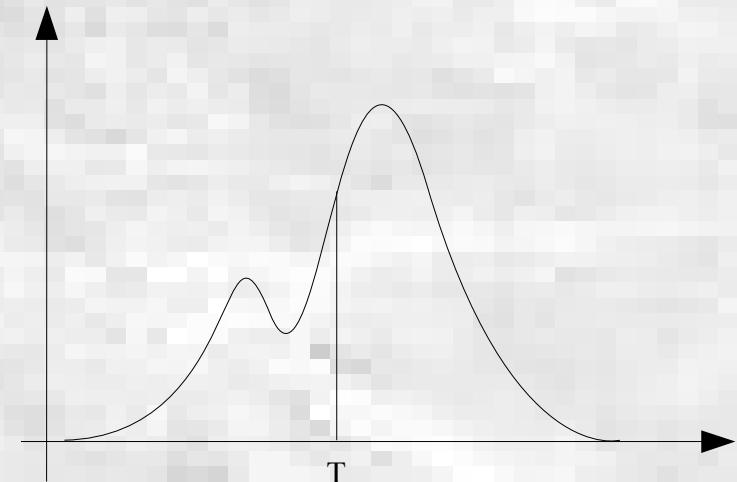
Les variances :

$$\sigma_1^2 = \frac{1}{T} \sum_{i=0}^{T-1} (h(i) - \mu_1)^2 \quad \sigma_2^2 = \frac{1}{256-T} \sum_{i=T}^{255} (h(i) - \mu_2)^2$$

Le poids de chaque classe :

$$P_1 = \sum_{i=0}^{T-1} h(i)$$

$$P_2 = \sum_{i=T}^{255} h(i)$$



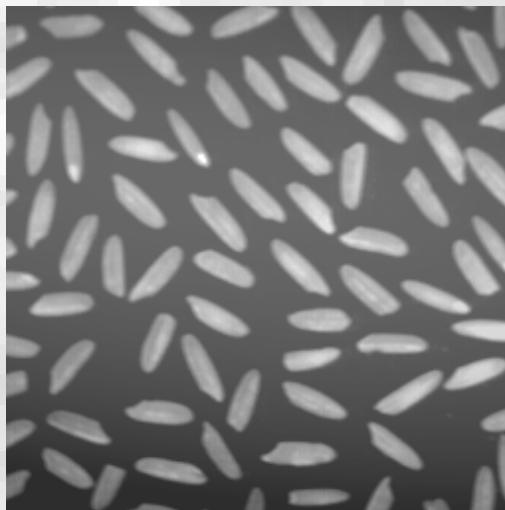
La variance Intra-Classe est donnée par :

$$\sigma_w^2 = P_1 \sigma_1^2 + P_2 \sigma_2^2$$

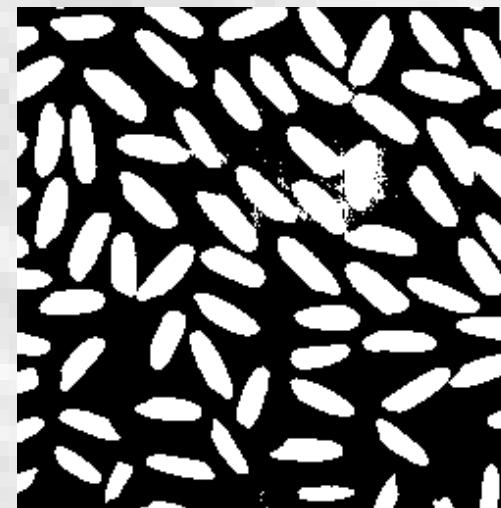
Le seuil optimal est celui donnant la valeur la plus faible de σ_w

Approche non supervisée

Algorithme d'Otsu



Histogramme



Seuil = 125

Otsu



Seuil = 145

Approche non supervisée

Algorithme des k-means

algorithme itératif en 3 étapes qui débute avec une valeur initiale de T aléatoire.

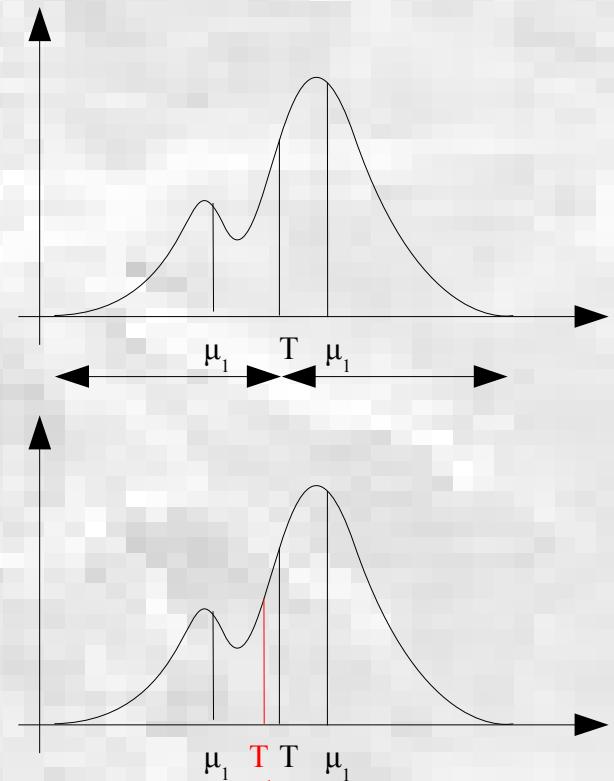
1. Pour un T donné on calcule les deux valeurs moyennes :

$$\mu_1 = \frac{1}{T} \sum_{i=0}^{T-1} h(i) \quad \mu_2 = \frac{1}{256-T} \sum_{i=T}^{255} h(i)$$

2. On calcul le nouveau T comme la valeur moyenne de μ_1 et μ_2 :

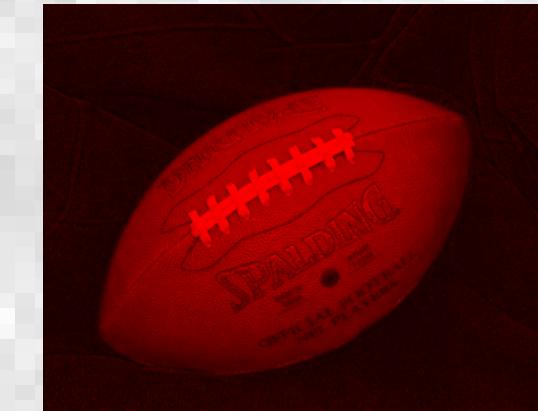
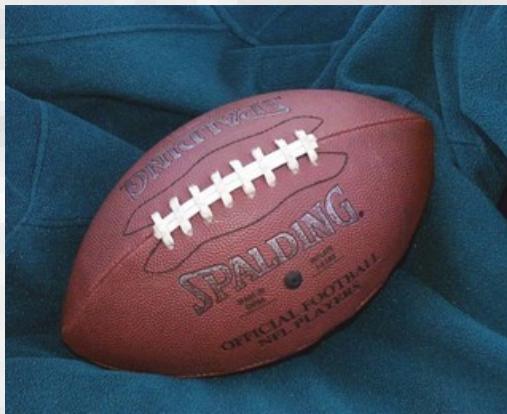
$$T = \frac{\mu_1 + \mu_2}{2}$$

3. Si T ne varie plus (convergence) c'est la valeur finale sinon retour en 1.



Segmentation en régions

C Cas des images couleur



3 valeurs par pixel : les composantes rouge, verte et bleue.



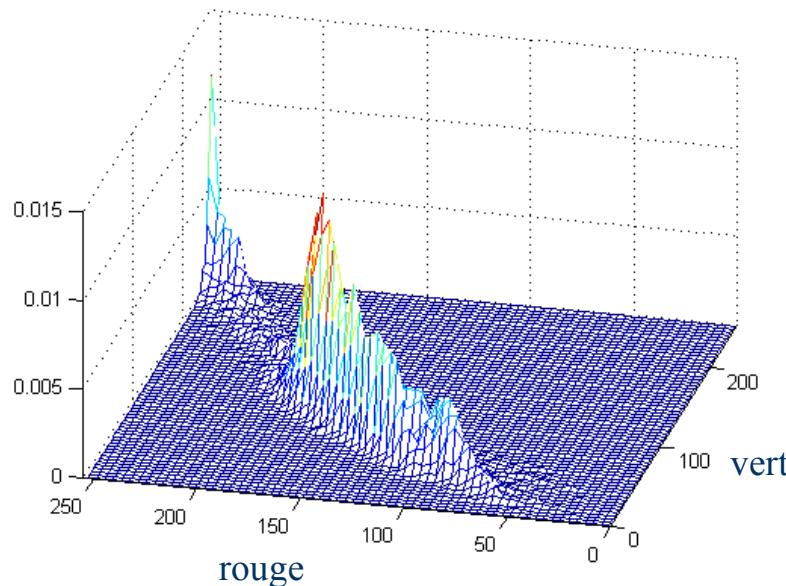
Segmentation en régions

C Cas des images couleur

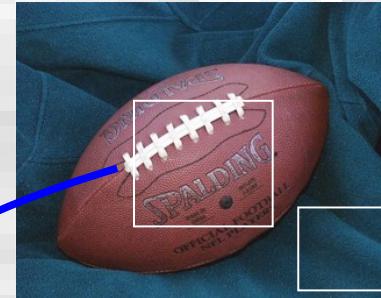
On applique la règle :

$$c = \arg\max(H_{\text{objet}}(c), H_{\text{fond}}(c))$$

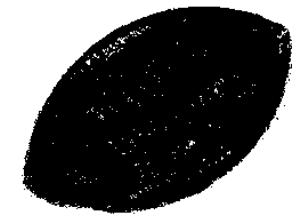
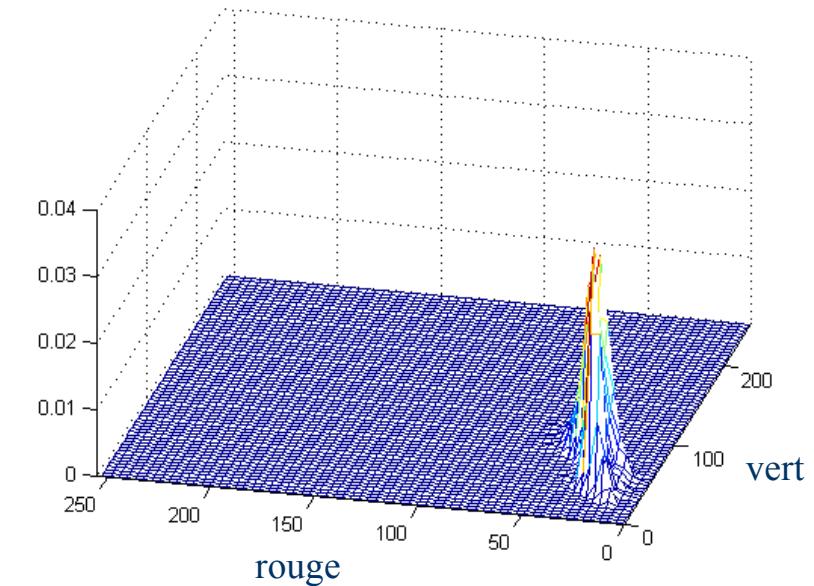
H_{balle}



Approche supervisée



H_{fond}



Approche supervisée

Méthode de Parzen

Principe : La méthode de Parzen (ou du noyau) consiste à pondérer de manière continue la contribution de chaque prototype à l'estimation de $p(x_0)$ en fonction de la distance de l'observation correspondante au point x_0 .

$$p(x_0) = \frac{1}{M} \sum_{m=1}^M \frac{1}{V[D(x_0)]} \cdot \phi\left(\frac{x_0 - x_m}{h}\right)$$

Où les x_m sont les M prototypes disponibles et $D(x_0)$ représente l'hypercube de côté h :

$$V[D(x_0)] = h^N \quad N \text{ nombre d'attributs}$$

h est choisi en suivant la règle :

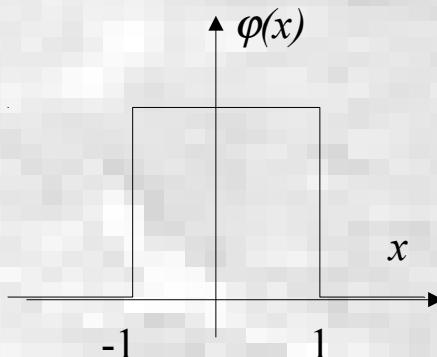
$$h = \frac{h_0}{\sqrt{M}} \quad \text{ou} \quad h = \frac{h_0}{\ln(M)}$$

Approche supervisée

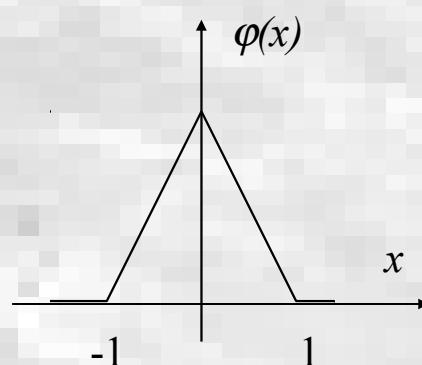
Méthode de Parzen

La fonction $\varphi(\cdot)$ est appelée le noyau de l'estimateur. Elle peut être de différentes formes :

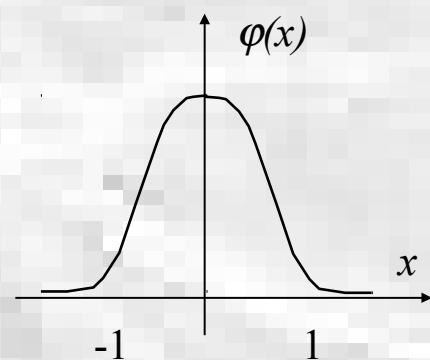
Cubique



Triangulaire



Gaussien



N.B. :

- La forme cubique conduit à un estimateur de type histogramme.
- Pour que l'estimation reste homogène à une probabilité $\varphi(\cdot)$ doit vérifier :

$$\varphi(u) \geq 0 \quad \forall u \quad \text{et} \quad \int \varphi(u) du = 1$$

Approche supervisée

Méthode des kppv

Principe : Dans la méthode des kPPV on se fixe un nombre d'échantillons k , et on fait croître le domaine D de l'estimation autour de x_0 jusqu'à ce qu'il contienne les k échantillons.

L'estimation est ensuite réalisée en appliquant :

$$p(x_0) = \frac{k/M}{V[D]}$$

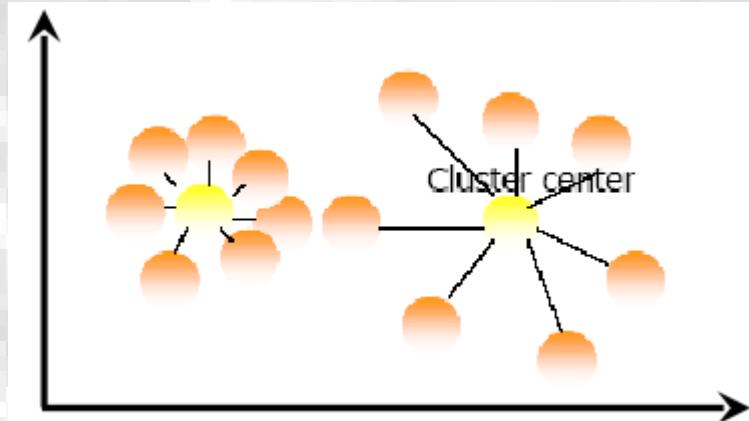
L'intérêt de cette méthode est qu'elle ajuste la taille du volume au nombre d'échantillons disponibles :

- Si $p(x_0)$ est grand (beaucoup d'échantillons au voisinage de x_0) D sera petit.
- Si $p(x_0)$ est faible D sera grand ce qui réduira le bruit d'estimation.

Algorithme des k-means multidimensionnel

Principe :

1. Chaque classe est associée à un noyau (cluster) qui représente une classe.
2. Chaque noyau est déterminé comme le centre de gravité des éléments de la classe.



Algorithme des k-means multidimensionnel

Algorithme :

1. *Choisir M noyaux (centres de classes).*
2. *Nouvelle partition : Chaque vecteur est associé au noyau le plus proche .*
3. *Calculer les centres de gravité des nouvelles classes.*
4. *Répéter 2 et 3 jusqu'à la convergence (Ex : les noyaux ne varient plus).*

La première étape peut être réalisée en choisissant M points au hasard. Généralement ce choix est déterminant pour le résultat final.

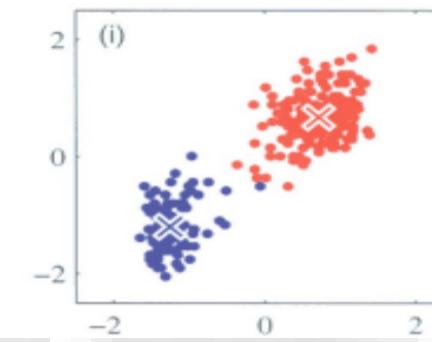
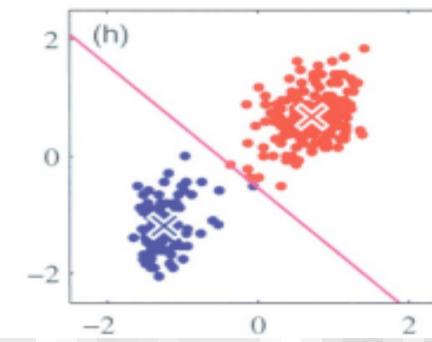
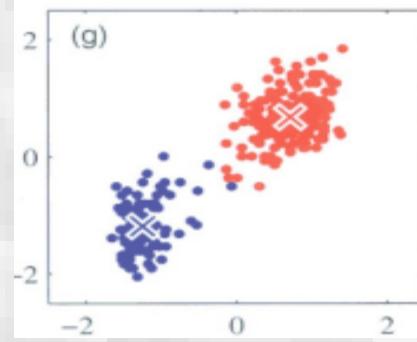
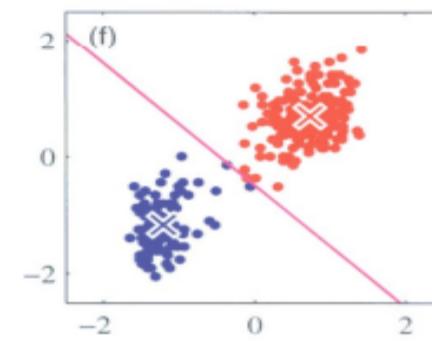
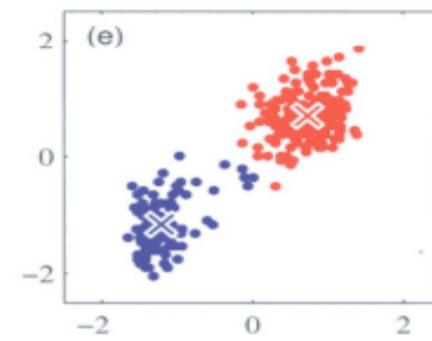
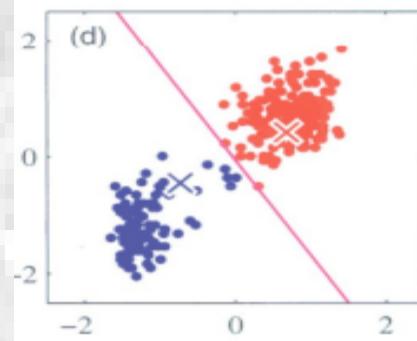
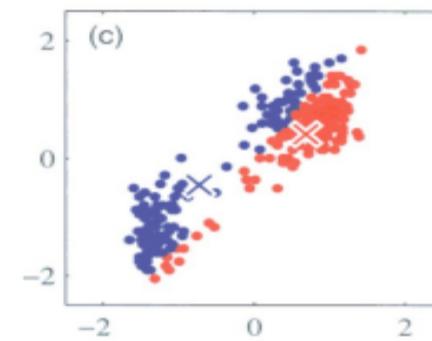
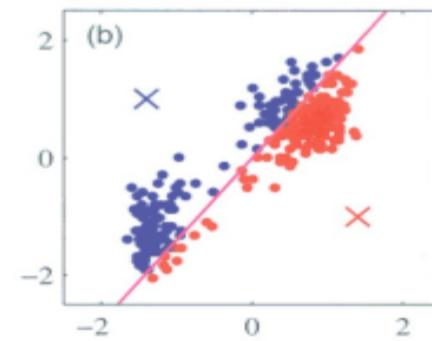
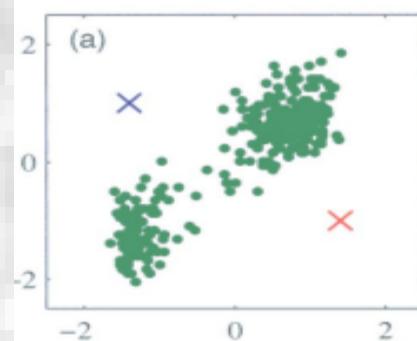
Si le nombre de classes n'est pas connu ajouter :

5. *Revoir le nombre de classes en regroupant les classes peu remplies, en éclatant les classes trop importantes ou en éliminant les points ou classes aberrantes.*

Segmentation en régions
C Cas des images couleur

Approche non supervisée

Algorithme des k-means



Segmentation en régions

D Croissance de région

On segmente l'image en région en détectant les zones homogènes selon une mesure particulière.

Parmi les différents critères utilisés on trouve :

- Le niveau de gris
- la texture
- la couleur

Principe :

La méthode nécessite de définir :

- Un critère H d'homogénéité sur une région (le niveau de gris moyen, une propriété de texture, ...).
- Un critère de voisinage (4-voisinage, 8-voisinage,...).
- Une région initiale R (généralement $R_{init} = 1$ pixel).

On étend la région R en incluant les pixels situés sur la frontière et dont la mesure est proche de celle de R (la variation de couleur est inférieure un seuil δ). En répétant cette procédure jusqu'à ce qu'il n'y ai plus de pixels dont la mesure est assez proche sur la frontière, on obtient une région de couleur homogène maximale autour du pixel de départ.

La méthode débute avec un (ou plusieurs) pixel(s) initial(aux) (germes).

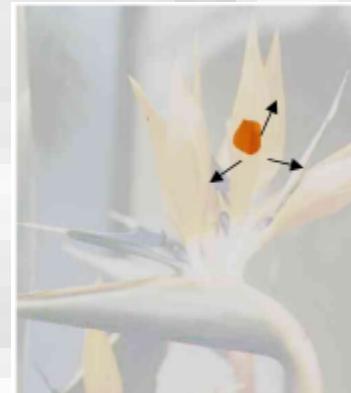
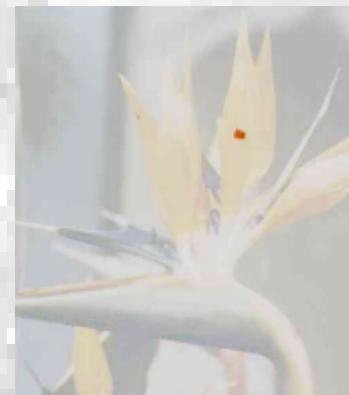
Segmentation en régions

D Croissance de région

Principe :

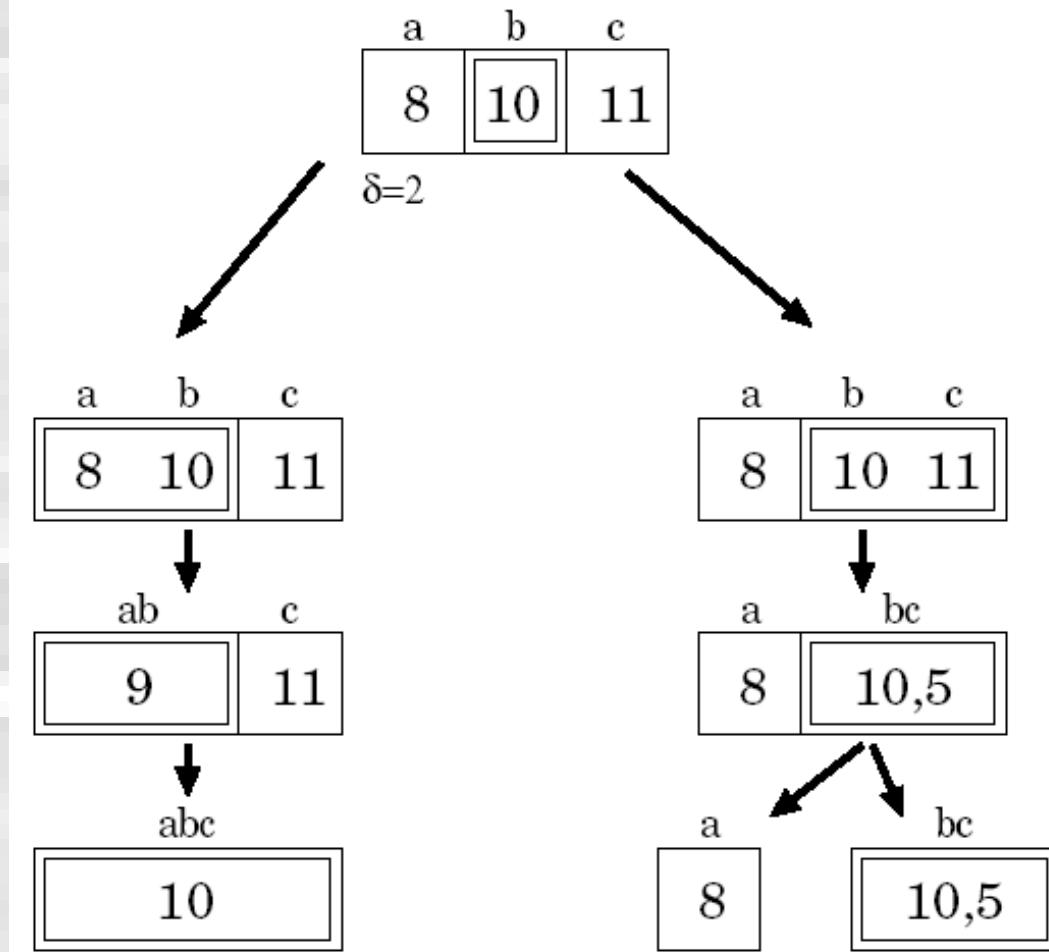
On démarre à partir d'un germe et on l'étend en ajoutant les pixels voisins qui satisfont au critère d'homogénéité.

Le germe peut être initialisé par un humain ou de façon automatique (dans une zone uniforme).



Segmentation en régions

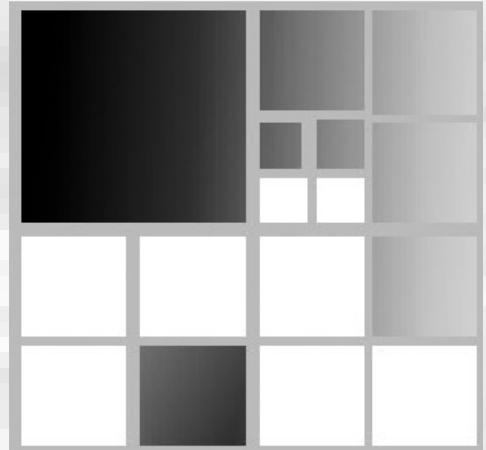
D Croissance de région



Segmentation en régions E « Split & merge »

- Etape de division (split)
 - Diviser récursivement tout bloc non homogène (ex : variance des NdG)
 - La division d'un bloc donne 4 sous-blocs.

- Etape de fusion (merge)
 - Regrouper les blocs adjacents représentant des régions homogènes



Segmentation en régions E « Split & merge »

Etape « split » :

L'image est découpée selon une structure de type « *quadtree* » (arbre quaternaire).

Algorithme :

1. R_{init} = l'image entière.
2. Si R est homogène \Rightarrow fin
Sinon découper R en 4 secteurs (R_1, R_2, R_3, R_4)
3. Exécuter 2. récursivement sur $R1, R2, R3, R4$

La région initiale est stockée comme un noeud dans un graphe et les sous parties comme des fils de ce noeud.

Segmentation en régions E « Split & merge »

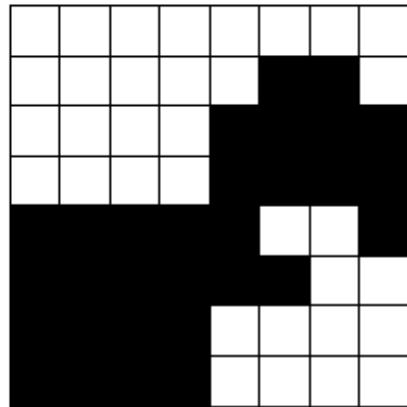
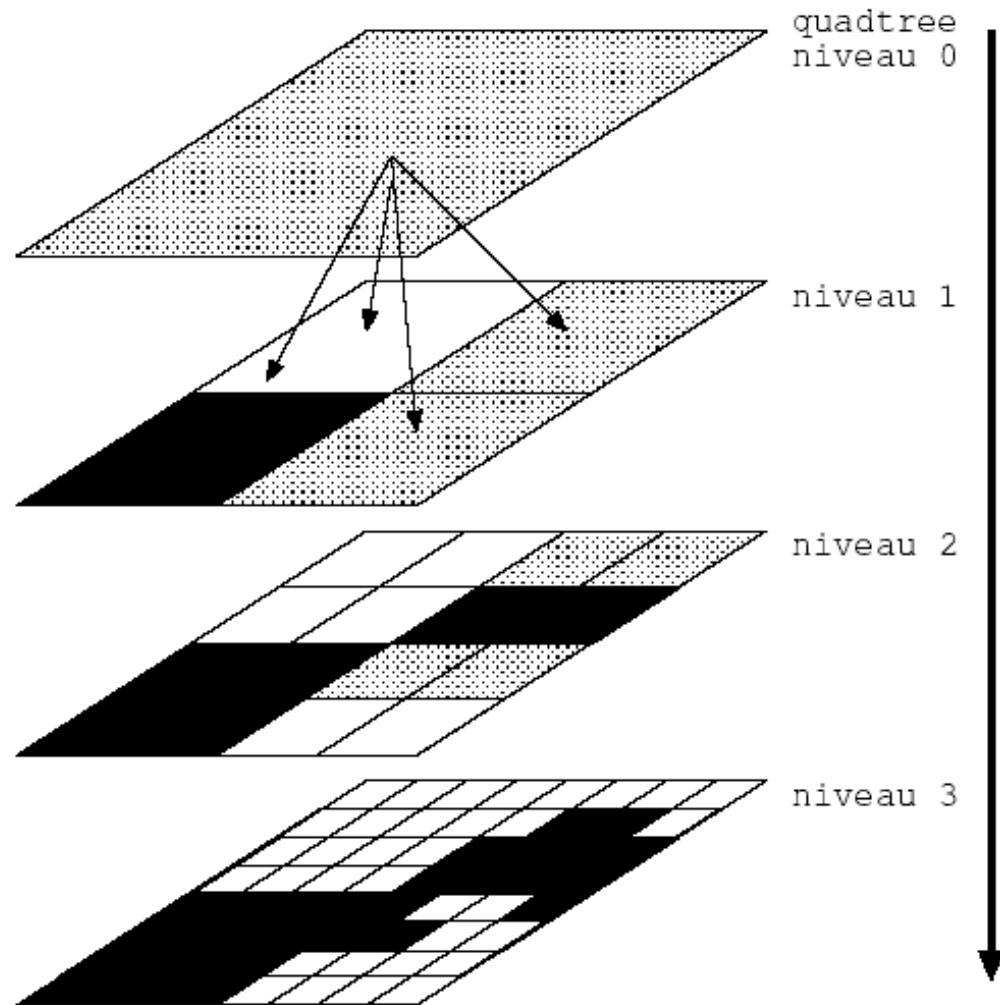
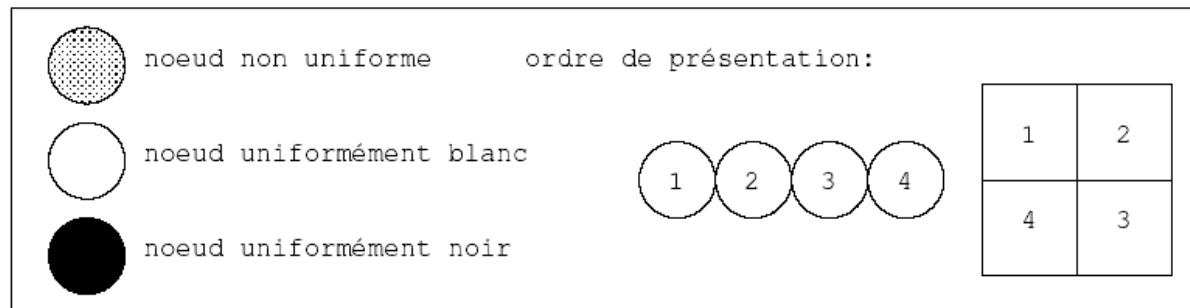
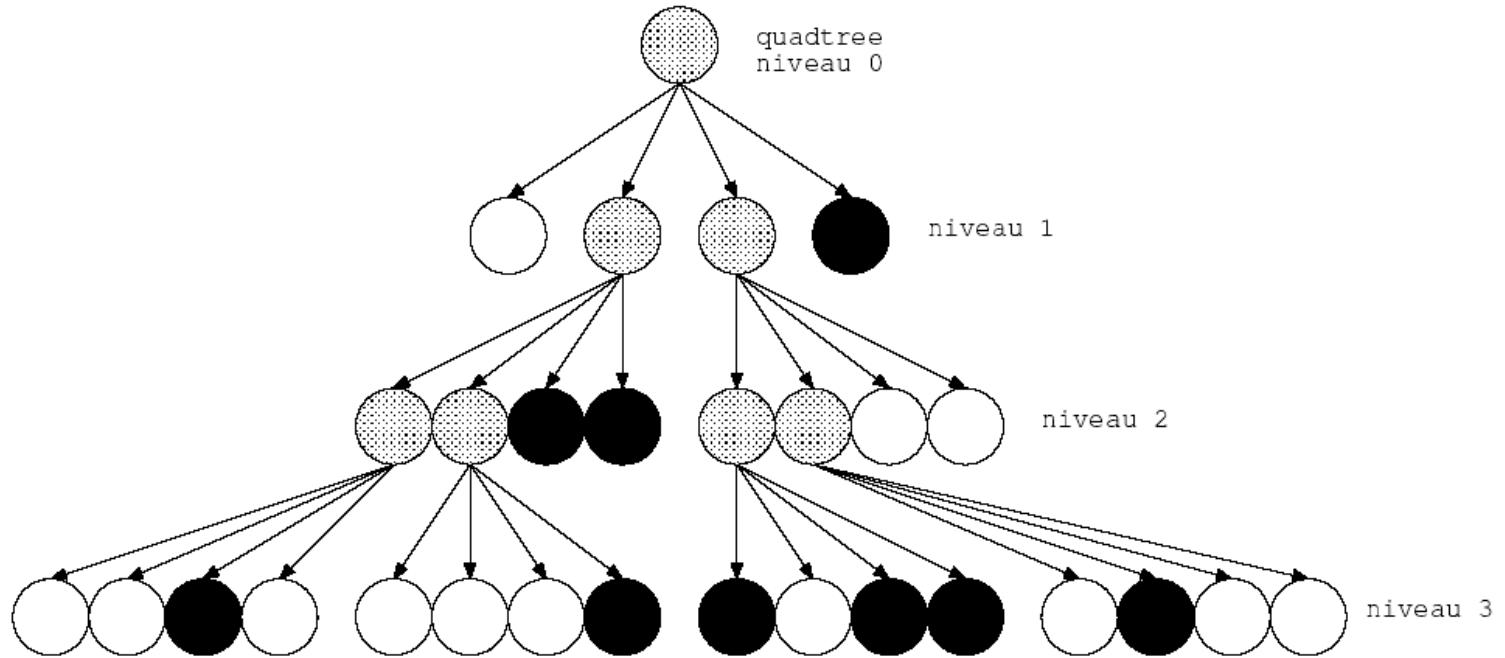


image 8x8 original (1bpp)



Segmentation en régions

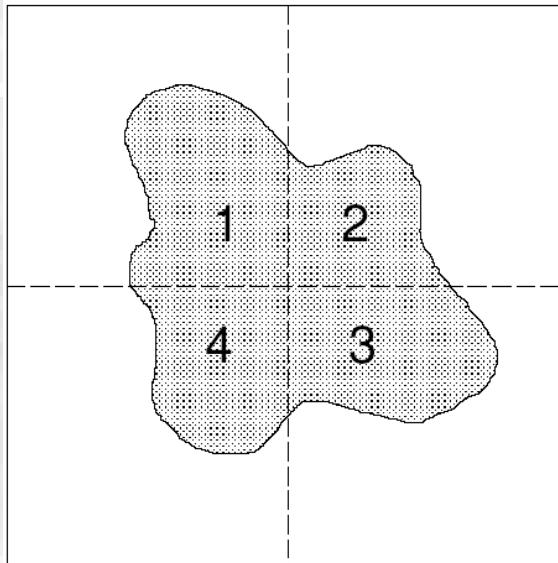
E « Split & merge »



Segmentation en régions E « Split & merge »

Etape « Merge » :

Après l'étape de « split » on obtient une image sur-segmentée : chaque feuille correspond à une région homogène, mais une région homogène peut-être éclatée entre plusieurs feuilles.



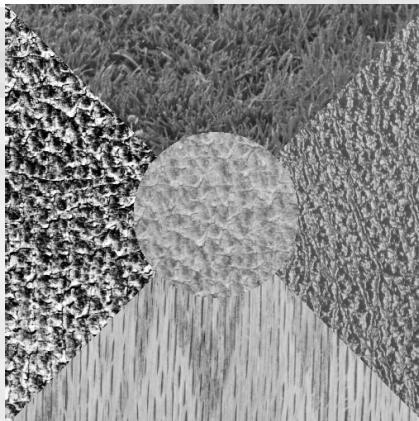
On associe une étape « merge » à l'étape « split » afin de fusionner les régions ainsi obtenues.

Analyse de texture

Analyse de texture

A Principe

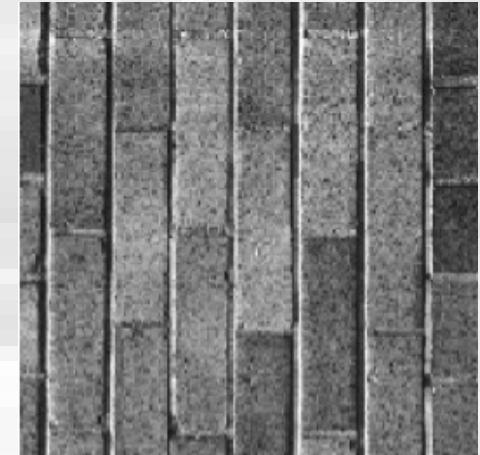
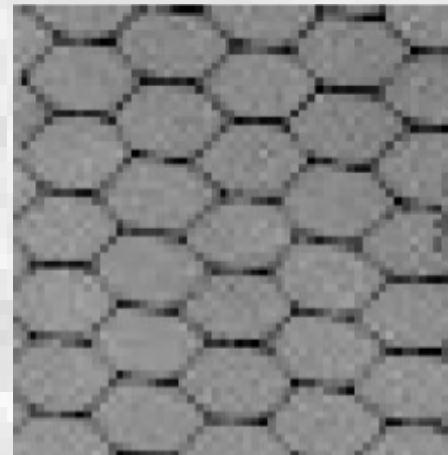
Une image texturée comporte des zones où un motif se répète de façon plus ou moins régulièrement.



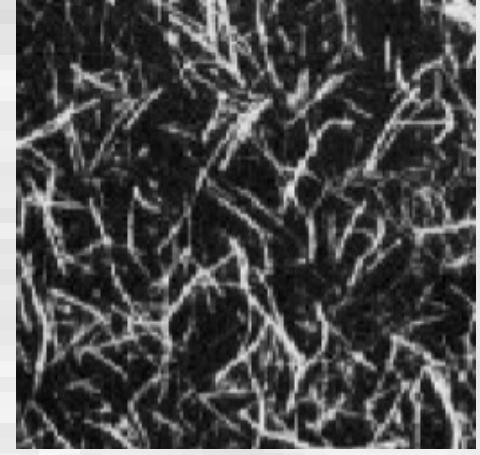
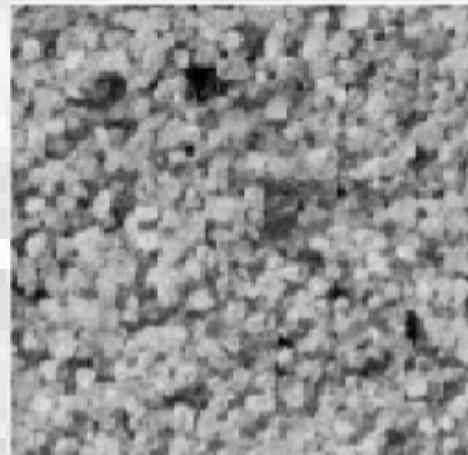
Analyse de texture

A Principe

Textures périodiques



Textures non périodiques

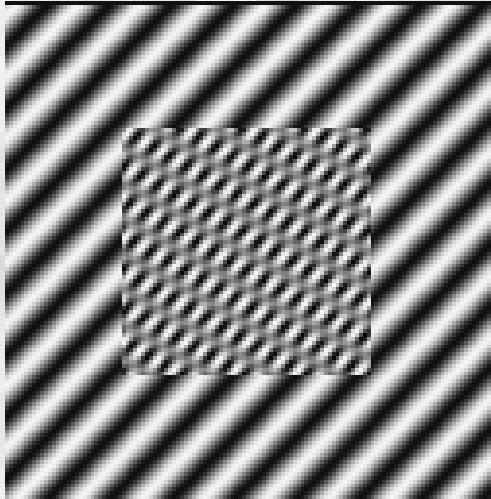


Analyse de texture

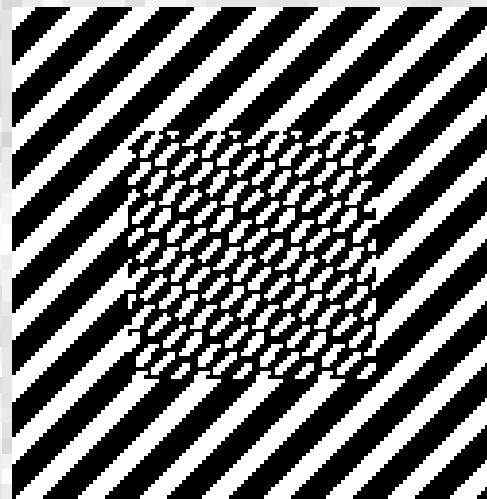
A Principe

Les techniques de segmentation basées sur l'intensité du niveau de gris sont inefficaces.

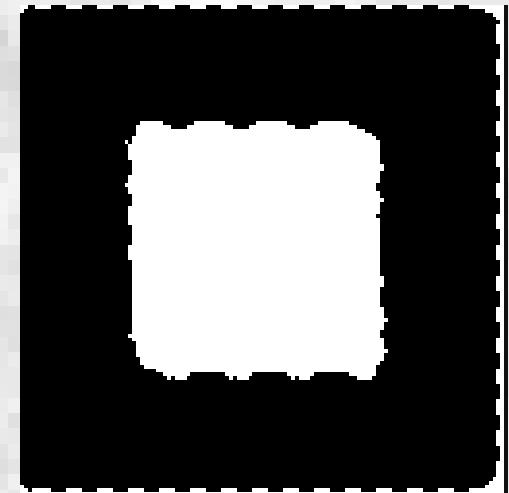
Image texturée



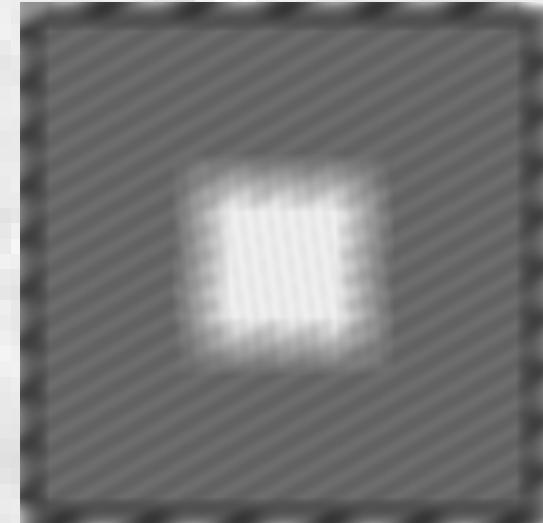
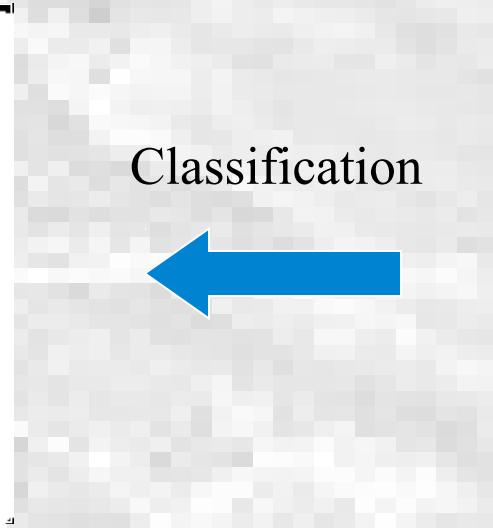
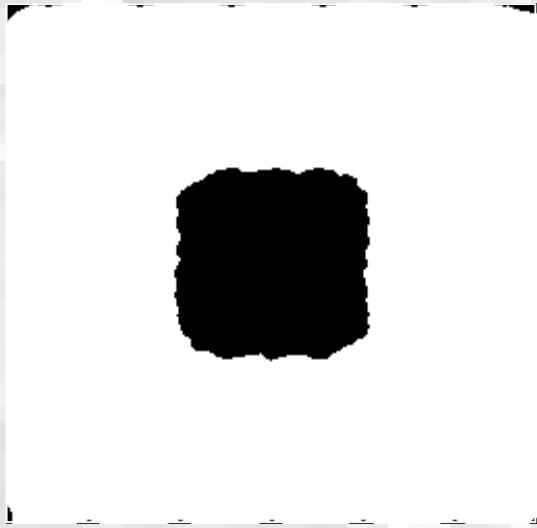
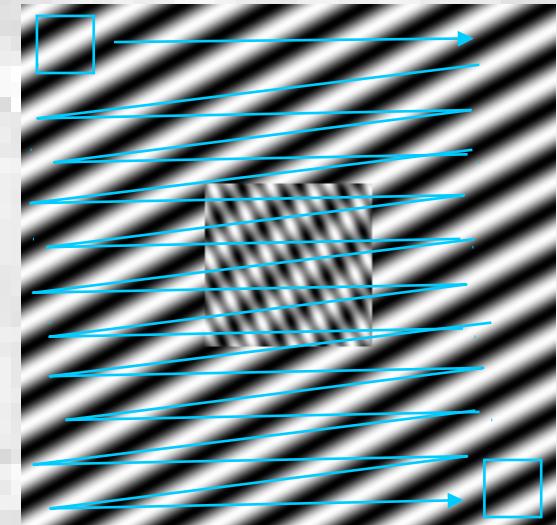
Segmentation
basée sur le NdG



Segmentation basée sur
un attribut de texture



Analyse de texture A Principe



Analyse de texture A Principe

Plusieurs approches :

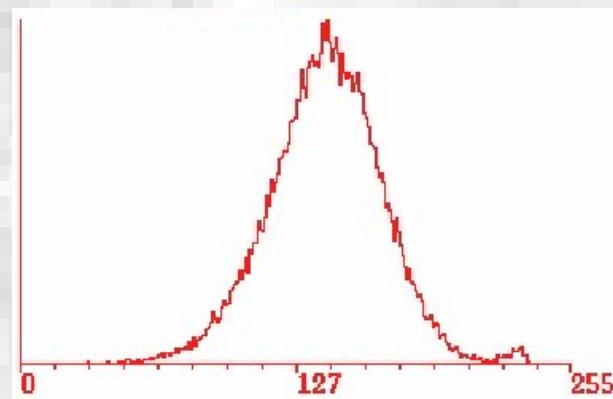
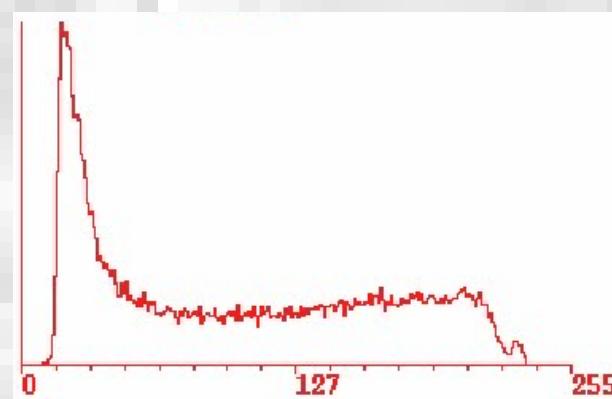
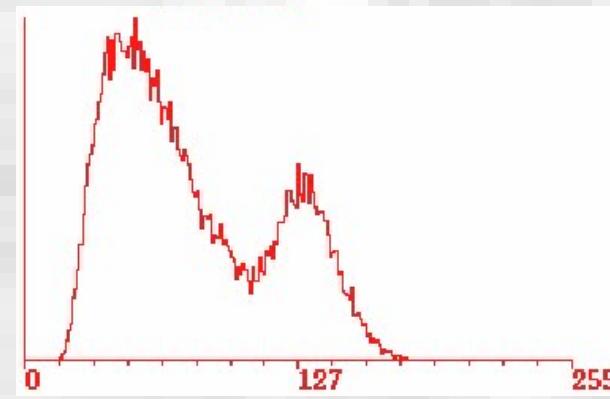
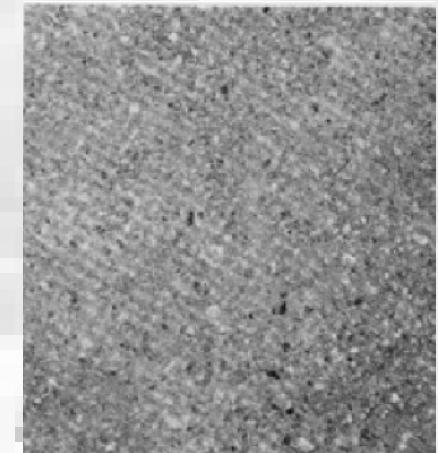
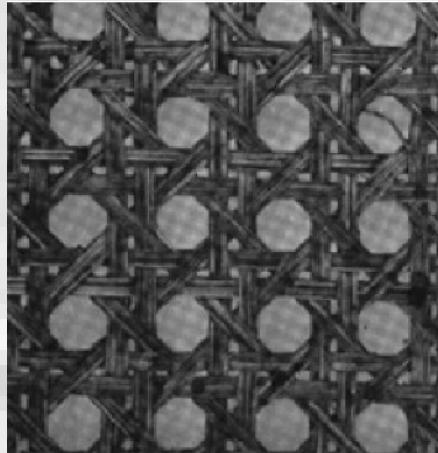
Analyse des propriétés statistiques de l'image
Utilisation de l'histogramme

Utilisation de la matrice de co-occurrences

Analyse des propriétés fréquentielles

Analyse de texture

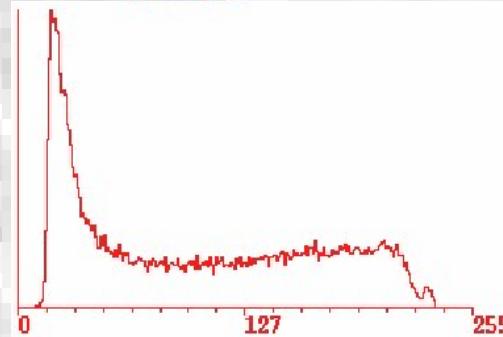
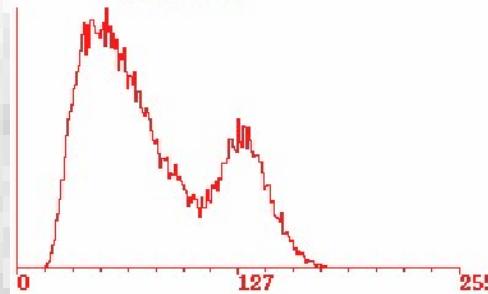
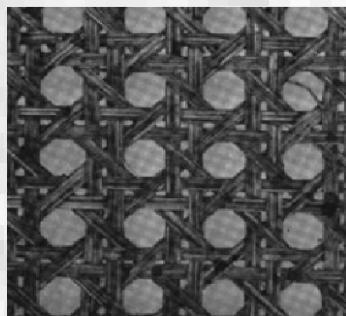
B Analyse de l'histogramme



Analyse de texture

B Analyse de l'histogramme

Comparaison de textures en utilisant l'histogramme.



Distance du Chi² entre les histogrammes :

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^K \frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

Analyse de texture

C Moments de l'image

Une façon simple de caractériser une texture est d'utiliser les moments du niveau de gris de l'image :

Moyenne :
$$\frac{1}{N} \sum_i \sum_j I(i, j)$$

Variance :
$$\frac{1}{N} \sum_i \sum_j (I(i, j) - \bar{I})^2$$

Asymétrie :
$$\frac{1}{N} \sum_i \sum_j \left(\frac{I(i, j) - \bar{I}}{\sigma} \right)^3$$

Kurtosis :
$$\frac{1}{N} \sum_i \sum_j \left(\frac{I(i, j) - \bar{I}}{\sigma} \right)^4$$

Analyse de texture

D Matrice de co-occurrence

La matrice de cooccurrence MC_t d'une région R se définit à partir d'un vecteur de translation t de la façon suivante :

$$MC_t(a,b) = \text{card}\{(s,s+t) \in \mathbb{R}^2 \text{ tq } I(s) = a, I(s+t) = b\}$$

où $I(s)$ représente l'intensité lumineuse d'un point s de l'image.

Image

1	2	1	3	4
2	3	1	1	4
3	3	2	1	1

MC_t avec $t = (0,1)$

$\begin{matrix} b \\ a \end{matrix}$	1	2	3	4
1	2	2	0	0
2	0	0	2	0
3	1	0	1	0
4	1	0	0	1

Analyse de texture

D Matrice de co-occurrence

A partir de cette matrice, on définit les attributs suivants :

L'homogénéité :
$$\frac{1}{Nc^2} \sum_a \sum_b (MC_t(a, b))^2$$

Le contraste :
$$\frac{1}{Nc \cdot (L-1)^2} \sum_{k=0}^{L-1} \sum_{|a-b|=k} k^2 \cdot MC_t(a, b)$$

La directivité :
$$\frac{1}{Nc} \sum_a MC_t(a, a)$$

L'uniformité :
$$\frac{1}{Nc^2} \sum_a MC_{t^2}(a, a)$$

Nc : Nombre de pixels de la région de l'étude

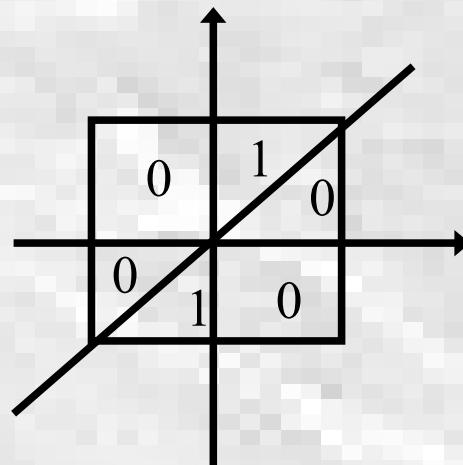
L : Nombre de niveaux de gris sur lesquels est calculée la matrice MC_t .

Analyse de texture

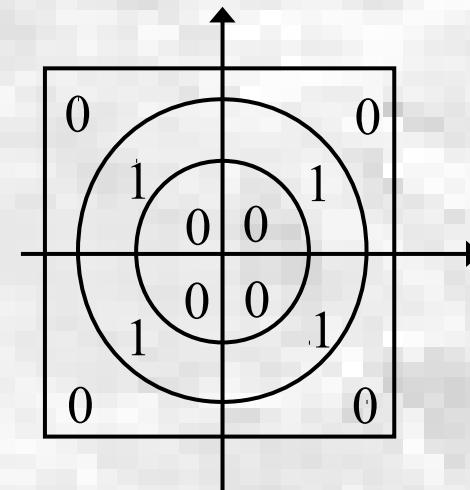
E Analyse fréquentielle

Principe : L'espace de Fourier est découpé en secteurs angulaires ou en couronnes.

On définit des filtres présentant des « 0 » et des « 1 » sur les différentes zones.



Filtre pour extraire une orientation fréquentielle.



Filtre pour extraire une amplitude fréquentielle.

Analyse de texture

E Analyse fréquentielle

L'attribut associé au domaine D s'écrit :

$$x = \iint_D (F(u, v) \cdot H(u, v))^2 dudv$$

Dans le domaine spatial x s'écrit :

$$x = \iint_J (f(x, y) * h(x, y))^2 dxdy$$

J étant un voisinage du pixel considéré de taille suffisante pour que la notion de variance ait un sens.

Analyse de texture

E Analyse fréquentielle

On montre que les filtres de Law peuvent se construire en multipliant 2 vecteurs.

Dans le cas de filtre 5x5 on a les vecteurs suivants :

$$\begin{aligned} E5 &= [-1 \quad -2 \quad 0 \quad 2 \quad 1] \\ L5 &= [1 \quad 4 \quad 6 \quad 4 \quad 1] \\ R5 &= [1 \quad -4 \quad 6 \quad -4 \quad 1] \\ S5 &= [-1 \quad 0 \quad 2 \quad 0 \quad -1] \\ W5 &= [-1 \quad 2 \quad 0 \quad -2 \quad 1] \end{aligned}$$

Exemples :

$$E5L5 = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$R5R5 = \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

Analyse de texture

E Analyse fréquentielle

Algorithme :

1. Convolution de l'image avec le jeu de masques :

$$\begin{aligned}x(i, j) &= (h * f)(i, j) \\&= \sum_{k=-a}^a \sum_{l=-a}^a h(k, l) \cdot f(i+k, j+l)\end{aligned}$$

2. Extraction des attributs de texture à l'aide d'un paramètre statistique de type variance discrète ou écart moyen absolu (ABSAVE).

Variance :

$$v(i, j) = \frac{1}{(2n+1)^2} \sum_{k=i-n}^{i+n} \sum_{l=j-n}^{j+n} [x(k, l) - m(i, j)]^2$$

ABSAVE :

$$s(i, j) = \frac{1}{(2n+1)^2} \sum_{k=i-n}^{i+n} \sum_{l=j-n}^{j+n} |x(k, l) - m(i, j)|$$

Estimation en traitement d'image

Estimation

A Moindres carrés linéaires

Soit un ensemble de k mesures $\{b_i\}$ supposées linéairement dépendantes d'un vecteur de paramètres x :

$$b_i = s_i(x)$$

Si $s_i(\cdot)$ est linéaire en x on a :

$$b_i = A_{i1}x_1 + A_{i2}x_2 + \dots + A_{in}x_n = A_i^T x$$

$A_i = (A_{i1}, \dots, A_{in})^T$: un vecteur de paramètres connus.

On pose $A = (A_1 \dots A_k)$ $b = (b_1 \dots b_k)^T$ et $x = (x_1 \dots x_n)^T$

Le problème se formalise par :

$$b = Ax$$

Estimation

A Moindres carrés linéaires

Cas $k = n$

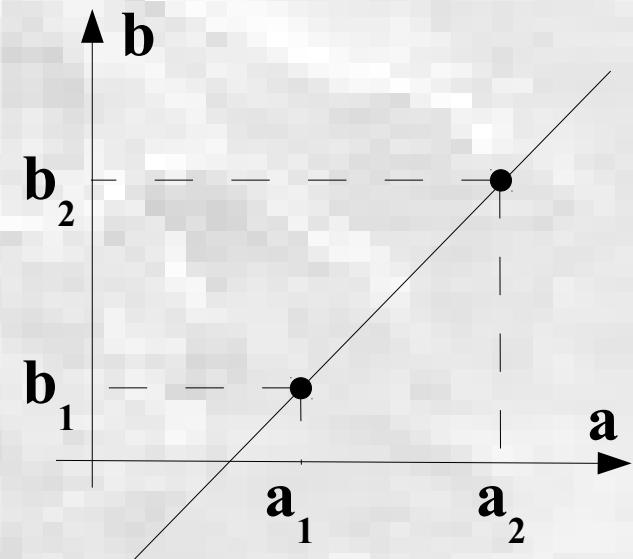
la solution est donnée par :

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad (\text{ si } \mathbf{A} \text{ est de rang } n)$$

Exemple : $n = 2$

Estimation de : $\mathbf{b} = \mathbf{a} \cdot \mathbf{x}_1 + \mathbf{x}_2$

$$\begin{cases} \mathbf{b}_1 = \mathbf{a}_1 \cdot \mathbf{x}_1 + \mathbf{x}_2 \\ \mathbf{b}_2 = \mathbf{a}_2 \cdot \mathbf{x}_1 + \mathbf{x}_2 \end{cases}$$



Estimation

A Moindres carrés linéaires

Cas $k > n$ Pas de solution.

Solution approchée :

$$\vec{\nabla} C = \vec{0} \quad [\mathbf{A}^T \mathbf{A}] \mathbf{x} - \mathbf{A}^T \mathbf{b} = 0$$

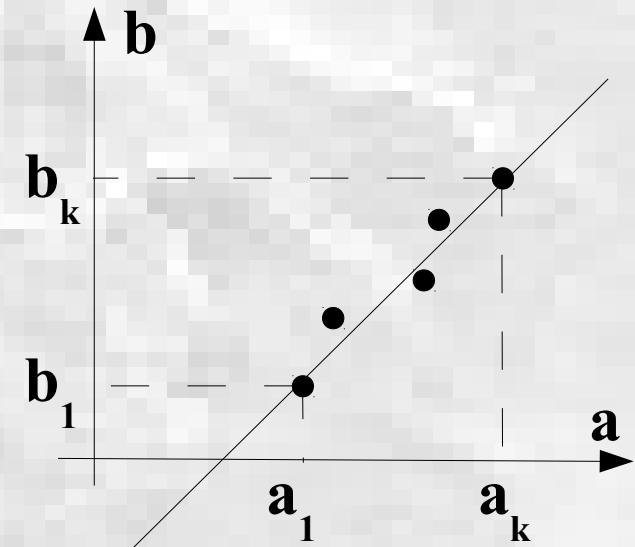
$$\mathbf{x} = [\mathbf{A}^T \mathbf{A}]^{-1} \mathbf{A}^T \mathbf{b} \quad \text{si } [\mathbf{A}^T \mathbf{A}] \text{ inversible}$$

Exemple : $n = 2$

Estimation de : $\mathbf{b} = \mathbf{a} \cdot \mathbf{x}_1 + \mathbf{x}_2$

$$\begin{cases} \mathbf{b}_1 = \mathbf{a}_1 \cdot \mathbf{x}_1 + \mathbf{x}_2 \\ \vdots \\ \mathbf{b}_k = \mathbf{a}_k \cdot \mathbf{x}_1 + \mathbf{x}_2 \end{cases}$$

$$\left\{ \begin{array}{l} \hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{Arg Min}} \mathbf{C} \\ \mathbf{C} = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 = \sum_{i=1}^k (\mathbf{b}_i - \mathbf{a}_i^T \mathbf{x})^2 \end{array} \right.$$



Estimation

B Application à la restauration

On suppose l'image localement modélisée par une constante :

$$x_i = c \quad x_i \in D$$

Le problème des moindres carrés se simplifie :

$$\begin{aligned}c &\text{ est un scalaire} \\y &= 1.\end{aligned}$$

On obtient : $c = \frac{1}{k} \sum_{i=1}^k x_i \Rightarrow$ Moyenne de l'image

Estimation

C Application à l'estimation du mouvement

Problème : Évaluation du déplacement des pixels entre 2 images.

Soit $I(i, j, t)$ la séquence.

Si le déplacement pendant la durée Δt du point (i, j) à l'instant t est (u, v) alors : $I(i, j, t) = I(i + u, j + v, t + \Delta t)$

Approximation linéaire : $I(i+u, j+v, t+\Delta t) \approx I(i, j, t) + \frac{\partial I}{\partial i} \cdot u + \frac{\partial I}{\partial j} \cdot v + \frac{\partial I}{\partial t} \cdot \Delta t$

On obtient : $\frac{\partial I}{\partial i} \cdot \frac{u}{\Delta t} + \frac{\partial I}{\partial j} \cdot \frac{v}{\Delta t} + \frac{\partial I}{\partial t} = 0$

Gradient de l'image (Sobel, ...):

$$\left(\frac{\partial I}{\partial i}, \frac{\partial I}{\partial j} \right)$$

Dérivée temporelle :

$$\frac{\partial I}{\partial t} \approx \frac{I(t+dt) - I(t)}{\Delta t}$$

Estimation

C Application à l'estimation du mouvement

$$\frac{\partial I}{\partial i} \cdot \frac{u}{\Delta t} + \frac{\partial I}{\partial j} \cdot \frac{v}{\Delta t} + \frac{\partial I}{\partial t} = 0$$

On suppose le vecteur de déplacement $(u, v)^T$ est constant sur un voisinage Ω (constitué de k pixels) du pixel considéré.

On pose :

$$c = \left(\frac{u}{\Delta t}, \frac{v}{\Delta t} \right) \quad \underline{x} = \begin{pmatrix} -\frac{\partial I_1}{\partial t} \\ \vdots \\ -\frac{\partial I_k}{\partial t} \end{pmatrix} \quad Y = \begin{pmatrix} \frac{\partial I_1}{\partial i} & \frac{\partial I_k}{\partial i} \\ \frac{\partial I_1}{\partial j} & \dots \\ \frac{\partial I_k}{\partial j} & \frac{\partial I_k}{\partial j} \end{pmatrix}^T$$

On a donc : $\underline{x} = Y \underline{c} + b$

$$\hat{c} = [Y^T Y]^{-1} Y^T \underline{x}$$

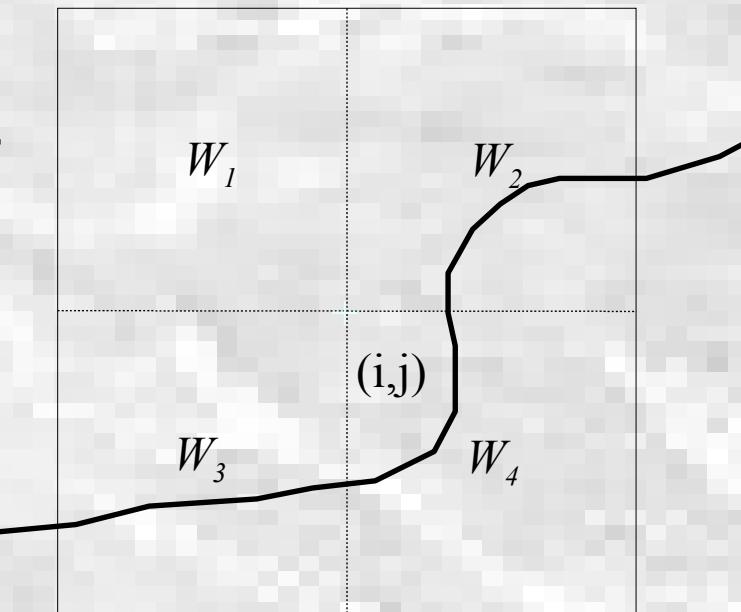
Estimation

C Application à l'estimation du mouvement



Estimation

D Restauration à préservation de contour

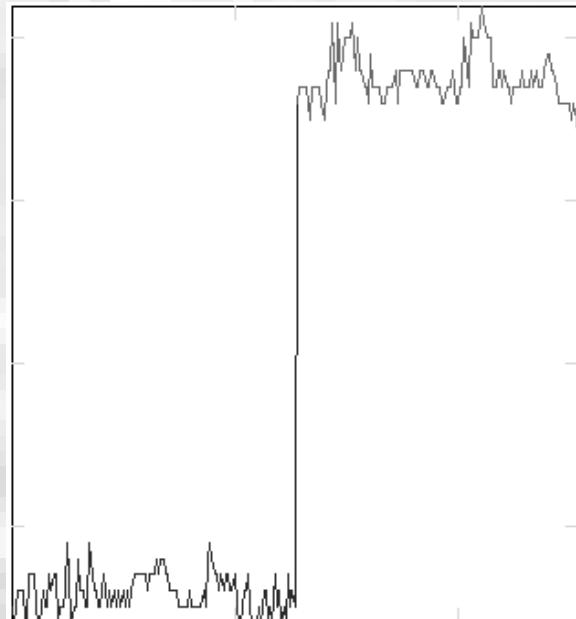
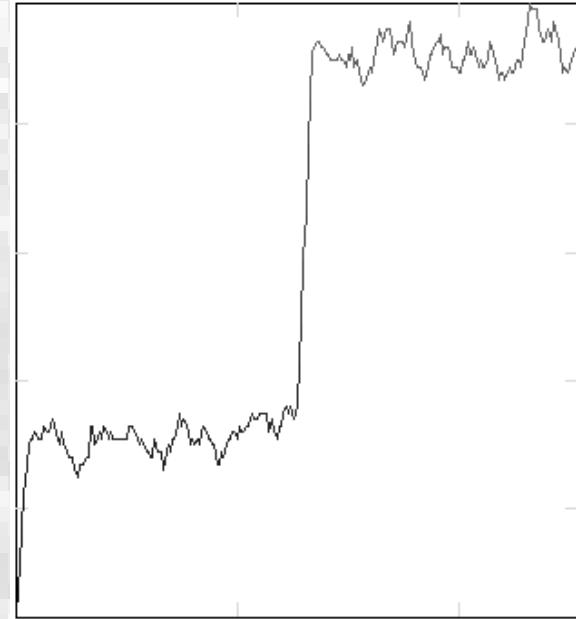
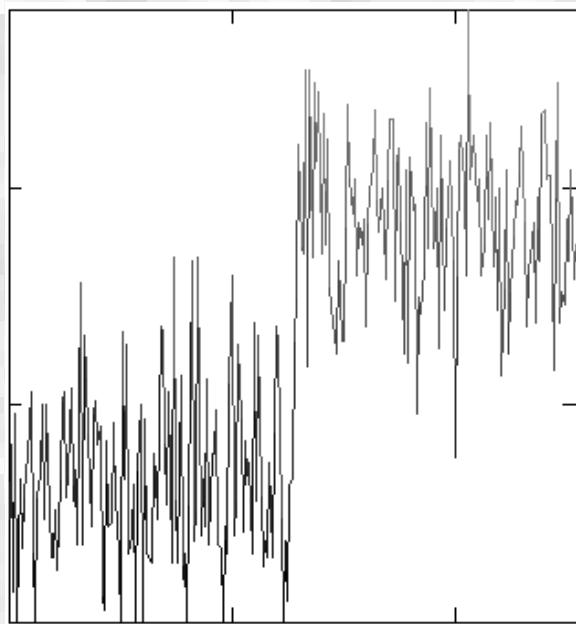


On définit une fenêtre locale au pixel divisée en quatre quadrants. La variance locale est calculée pour chaque sous-fenêtre.

La sous-fenêtre la plus homogène (i.e. dont la variance est la plus faible) est alors utilisée pour calculer la valeur finale filtrée associée au pixel (i,j) .

Estimation

D Restauration à préservation de contour



Estimation

E Filtres d'ordre

Les filtres d'ordre (également appelés **Order Statistic Filters** ou **L-filters**) constituent une catégorie d'estimateurs robustes appliqués à la restauration.

Ils se définissent de la façon suivante :

A partir du vecteur d'observation :

on réalise le vecteur des données triées :

tel que :

$$\underline{x} = (x_1, \dots, x_k)^T$$

$$\underline{x}' = (x'_1, \dots, x'_k)^T$$

$$x'_1 \leq x'_2 \leq \dots \leq x'_k$$

c est ensuite calculée en appliquant :

$$c = \underline{a}^T \cdot \underline{x}' \quad \underline{a} : \text{vecteur de pondération.}$$

\underline{a} est généralement choisi afin d'atténuer les valeurs éloignées de la valeur médiane.

Exemple : Le vecteur : $a[i] = \begin{cases} 1 & i = n \\ 0 & i \neq n \end{cases}$ défini le filtre de la médiane.

avec $k = 2n+1$ la dimension de \underline{a}



image originale

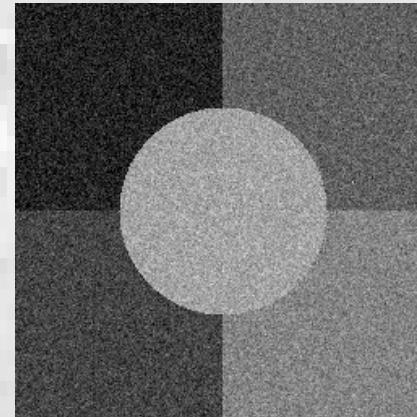
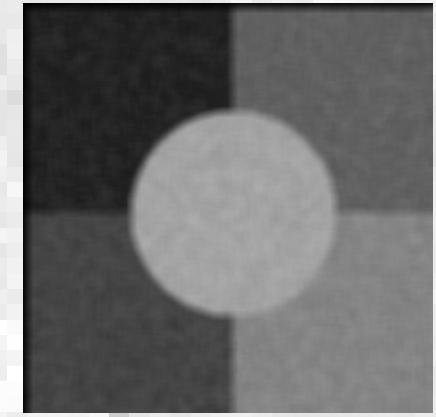
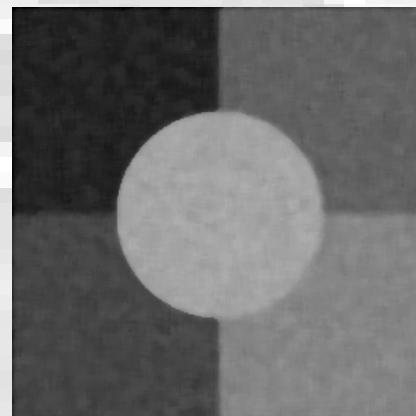


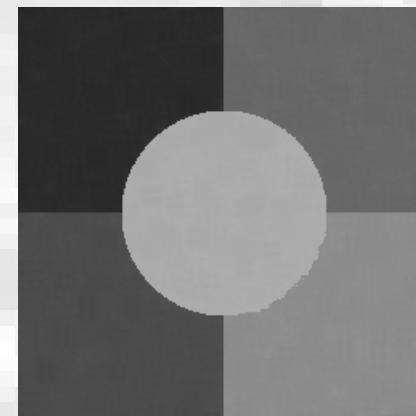
image bruitée



filtrage moyenneur



filtre médian



EPNSQ