

# Cours de traitement d'images

Sur la base des cours de Catherine Achard

- 
- ▶ Image processing using Matlab ; Gonzalez et Woods
  - ▶ Digital Image Processing ; Gonzales & Woods
  - ▶ Introduction au traitement d'images ; Lingrand.
  - ▶ Traitement de l'image et de la Vidéo ; Rachid Belaroussi
  - ▶ Introduction au traitement d'images - Simulation sous Matlab ; Gilles Burel

# Introduction

# Applications

## Application: Détection de défauts

Cheveux d'anges



Inclusions

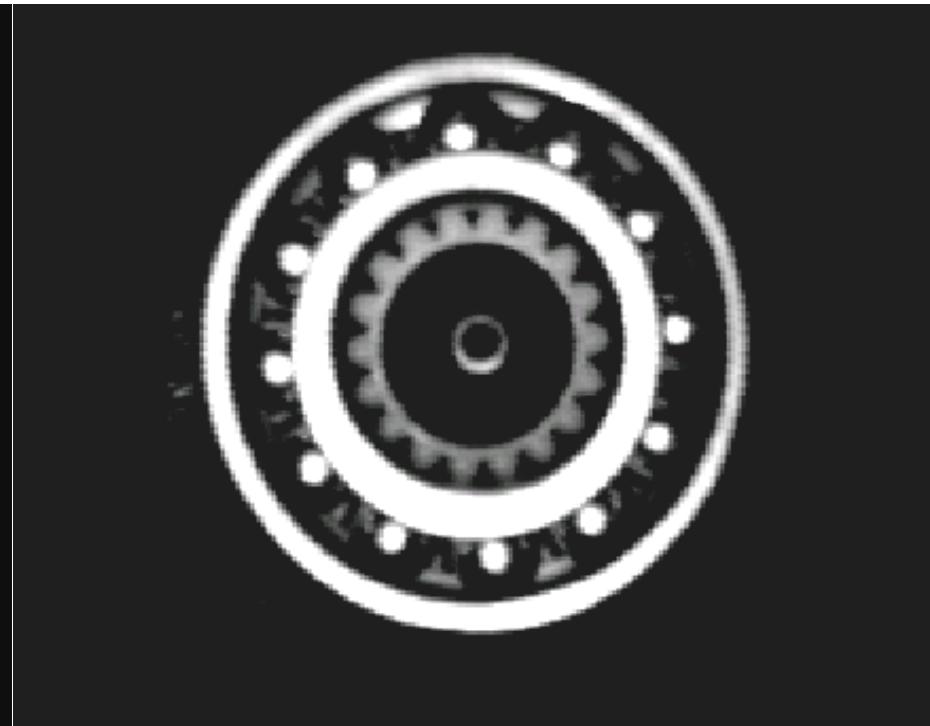
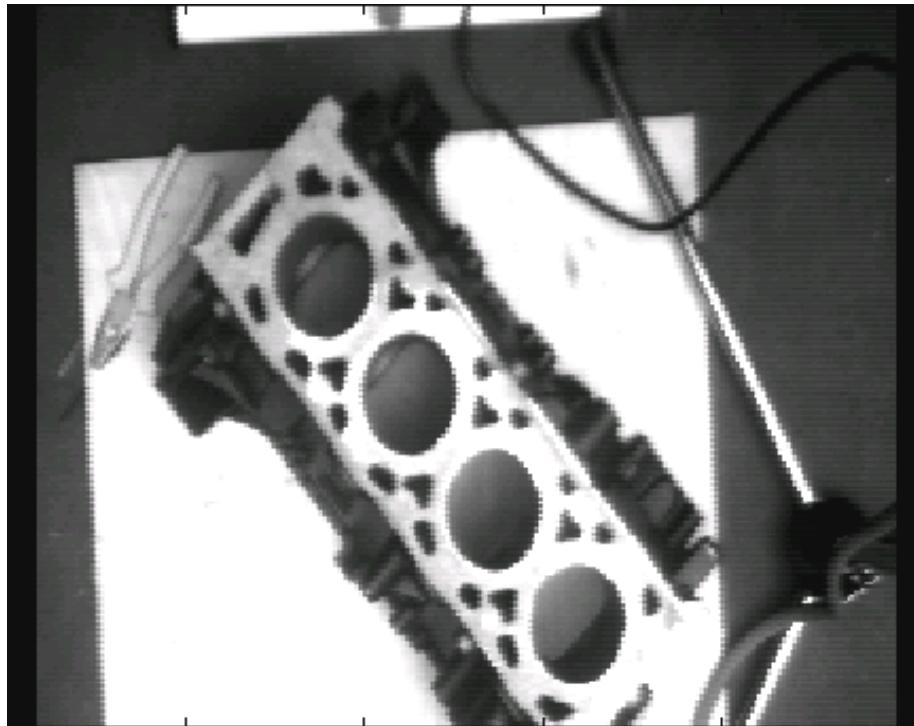


Cassure



## Application: Imagerie industrielle

---



## Application: Robotique autonome

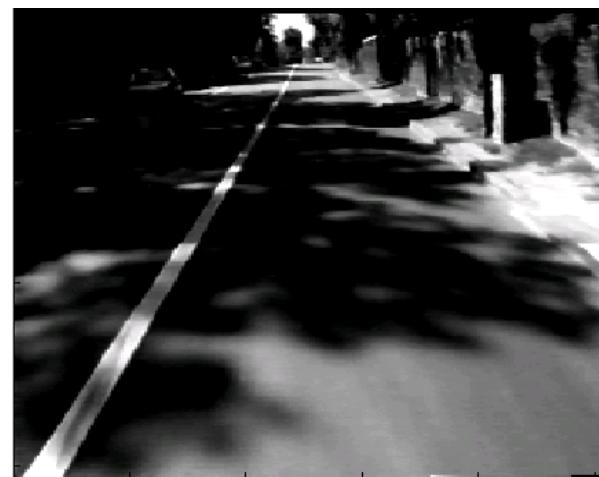
---



## Application: Robotique mobile



## Application: Guidage latéral de véhicules



## Application: Surveillance de tunnels



Media Tunnel par citilog qui fourni une détection d'incidents et analyse du trafic en temps réel dans les tunnels.

<http://www.citilog.com/product/fr/mediatunnel>

## Application: Analyse de crash test

La vision appliquée à l'analyse de mouvement.

Création de marqueurs vectoriels et suivi sur une séquence d'images.

Tracking automatique par reconnaissance de forme (pattern matching).-

Possibilité de prédictions des mouvements par calcul théorique.

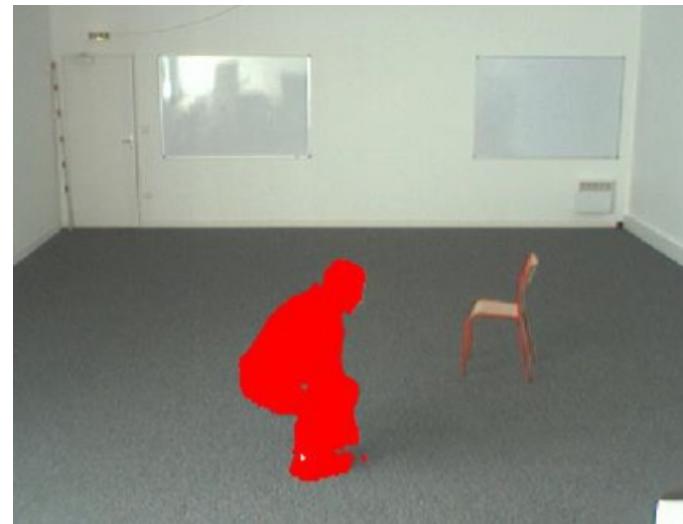
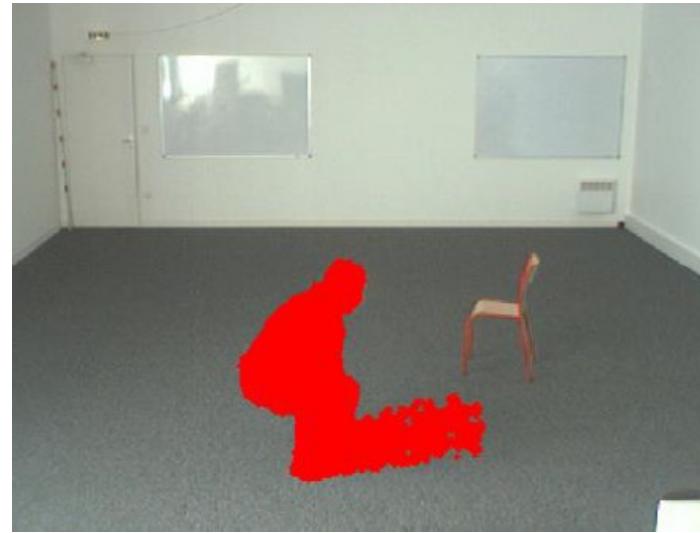
Possibilité de sauver la séquence et les marqueurs au format AVI.

Affichage des résultats sous forme graphique et tableurs Excel(vitesse de déplacement, accélération, trajectoire...).

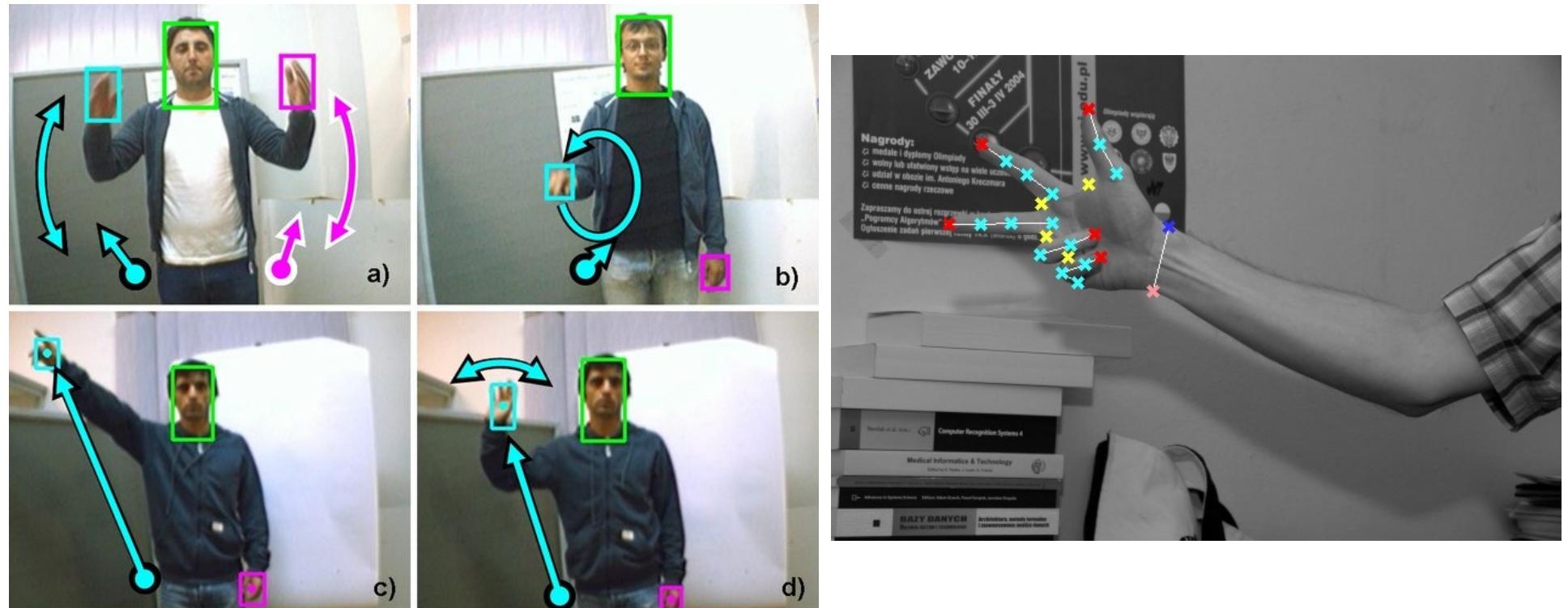
Support de caméras rapides



## Application: Déetecter des mouvements



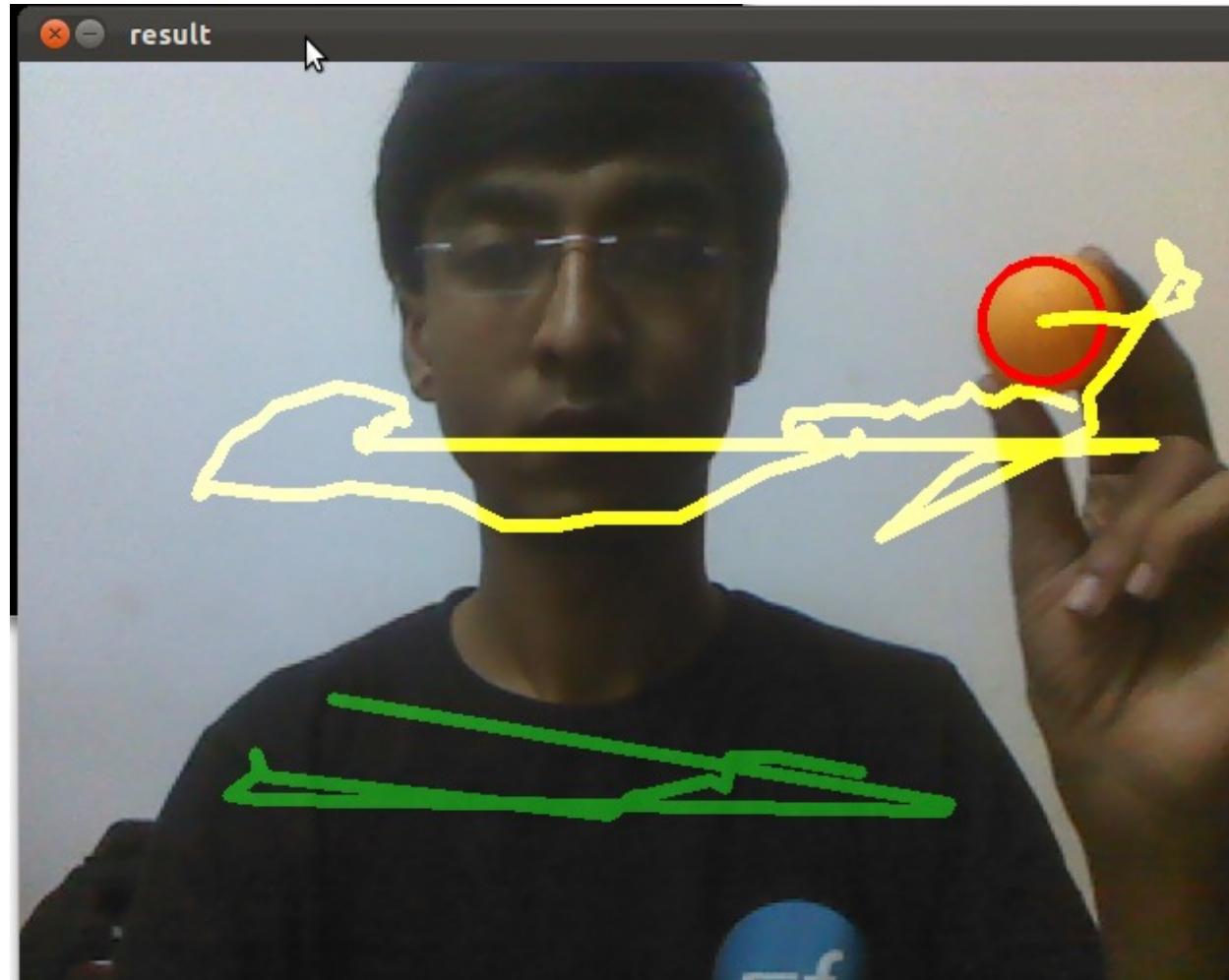
# Application: Reconnaître des mouvements



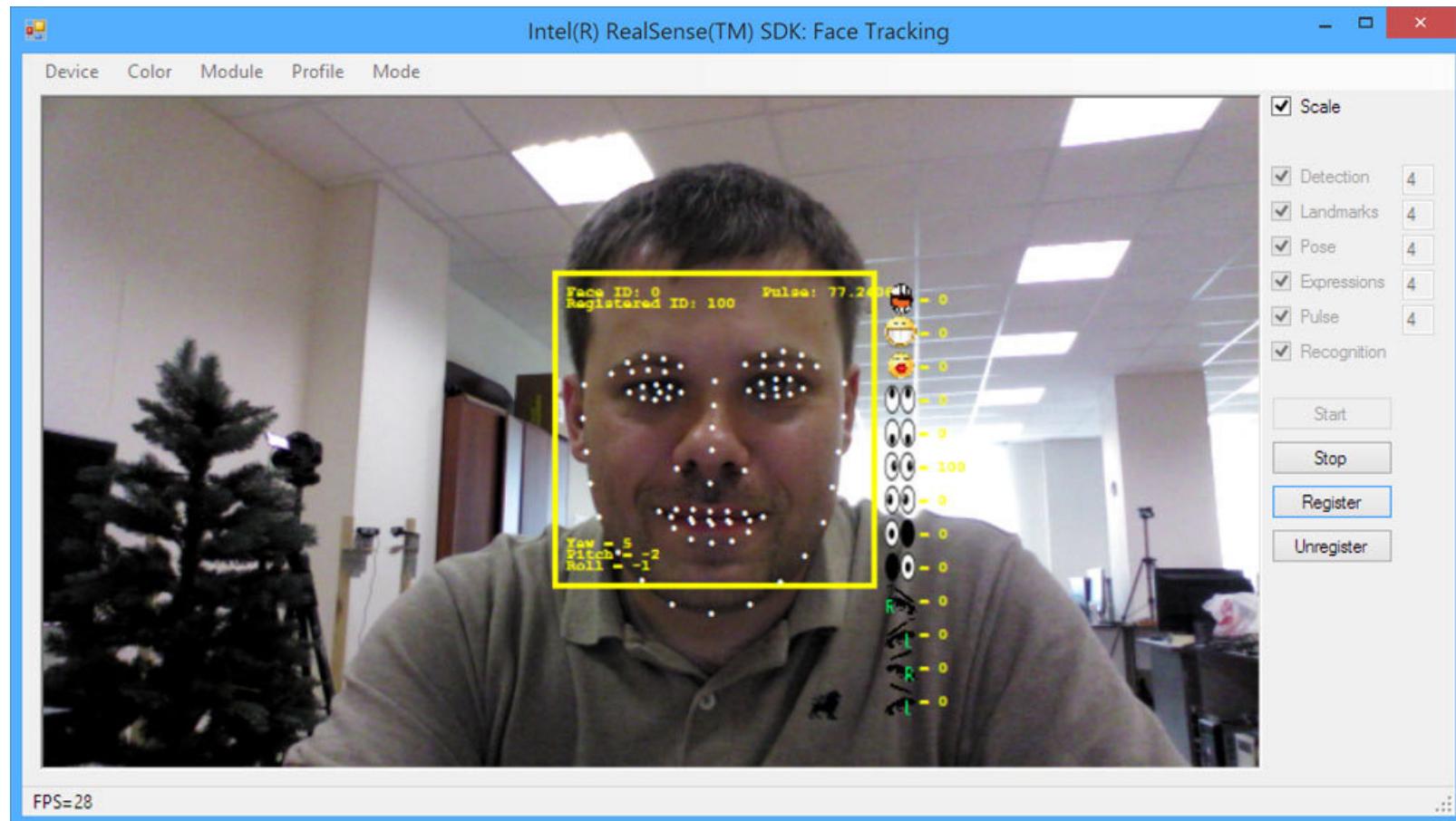
# Application: Détection et suivi de personnes



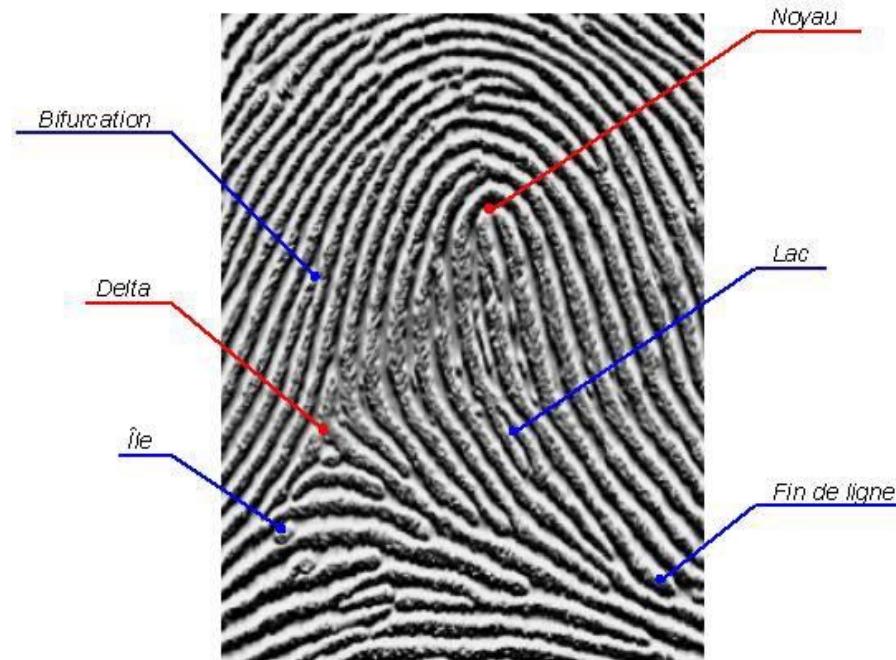
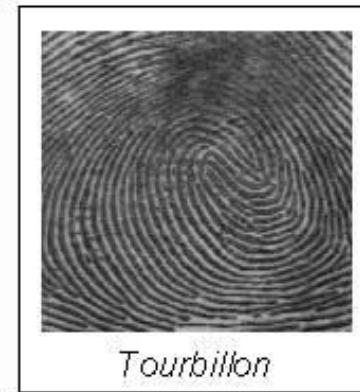
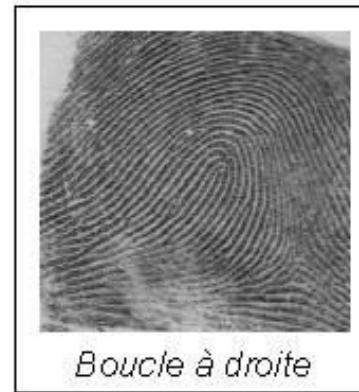
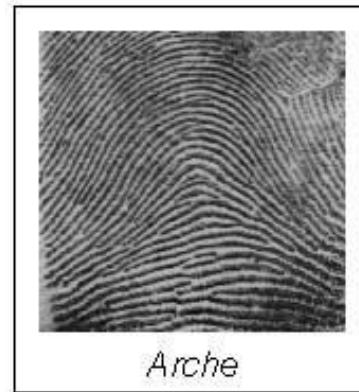
## Application: Suivi d'objets



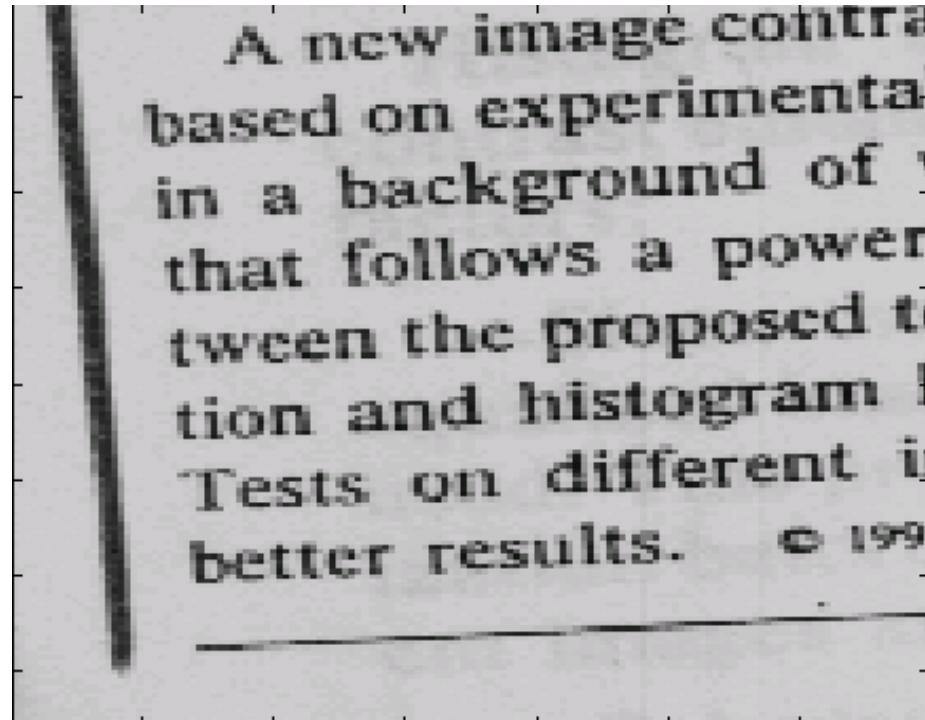
# Application: Détection de visages



# Application: Reconnaissance d'empreintes digitales



## Application: Traitement de l'écrit

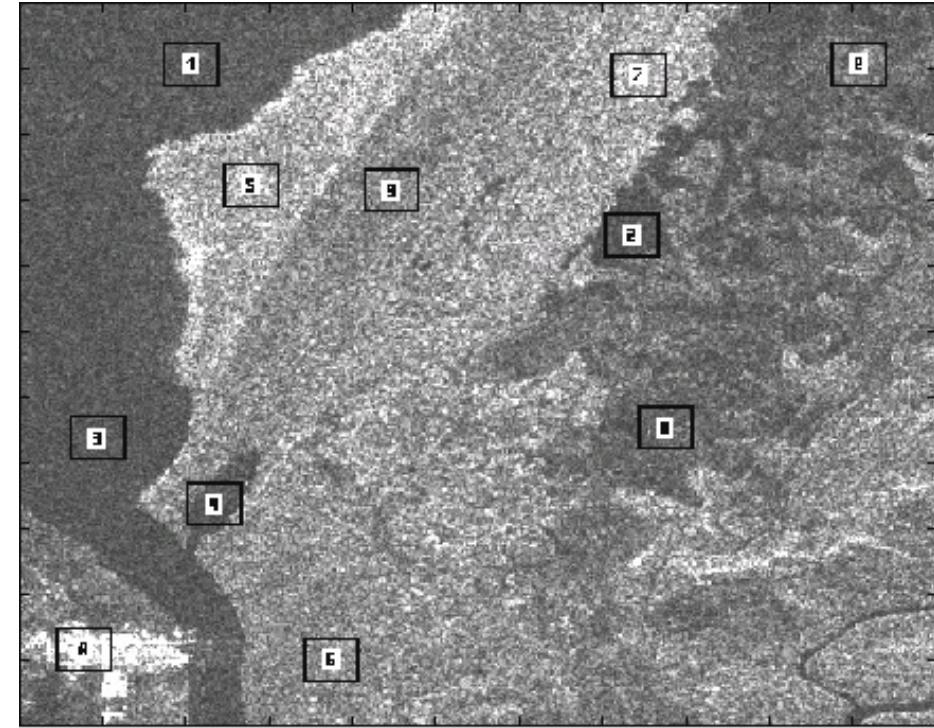


# Application: Aériennes et satellitaires

Imagerie classique(visible)



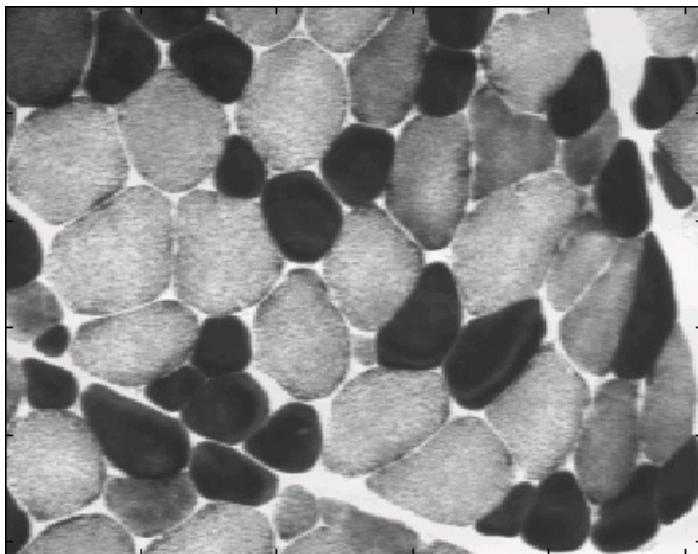
Imagerie radar (SAR)



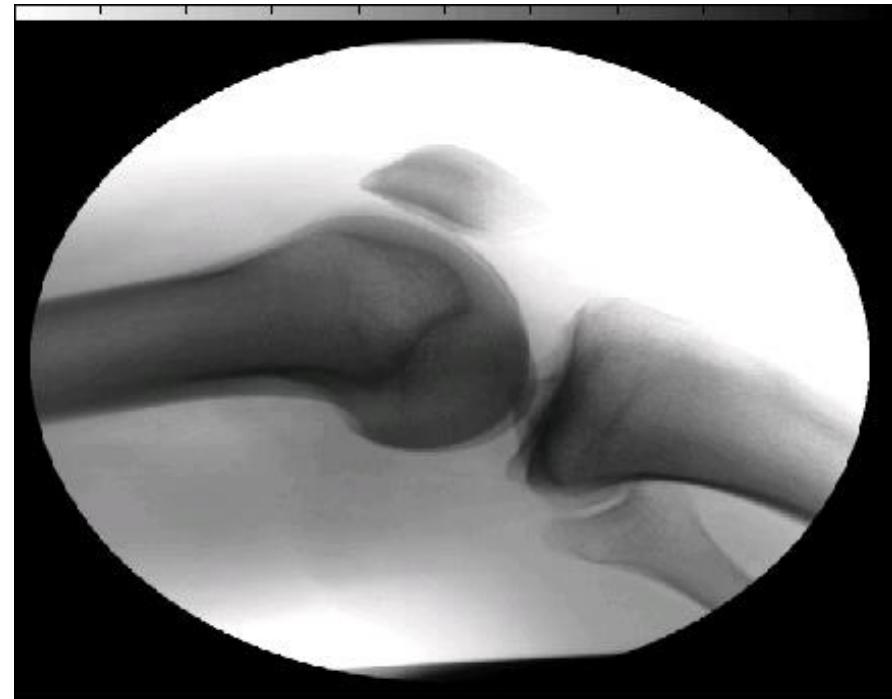
## Application: Imagerie bio-médicale

---

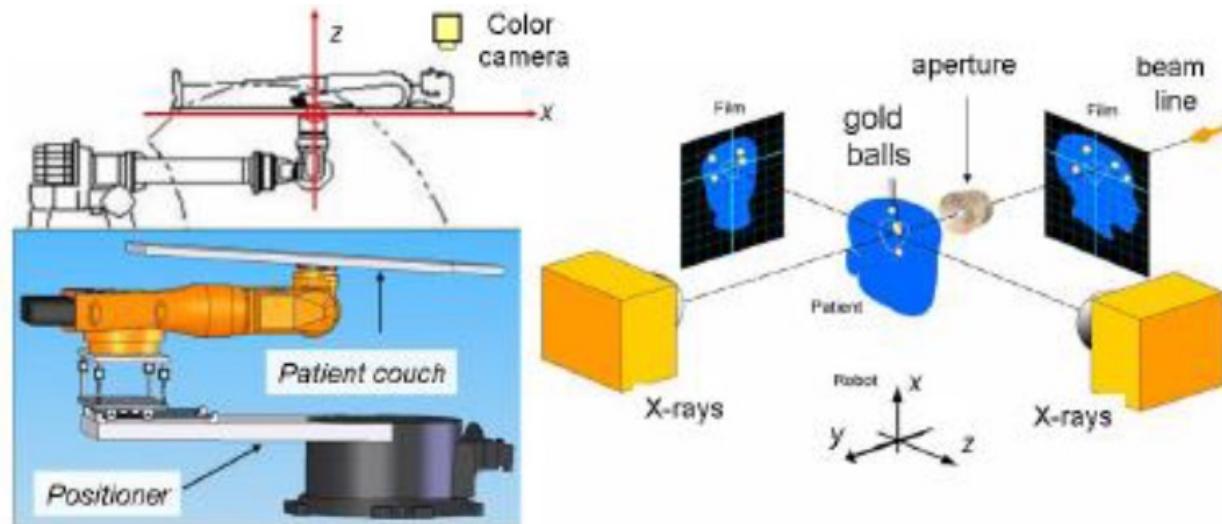
Microscopie optique



Radiologie

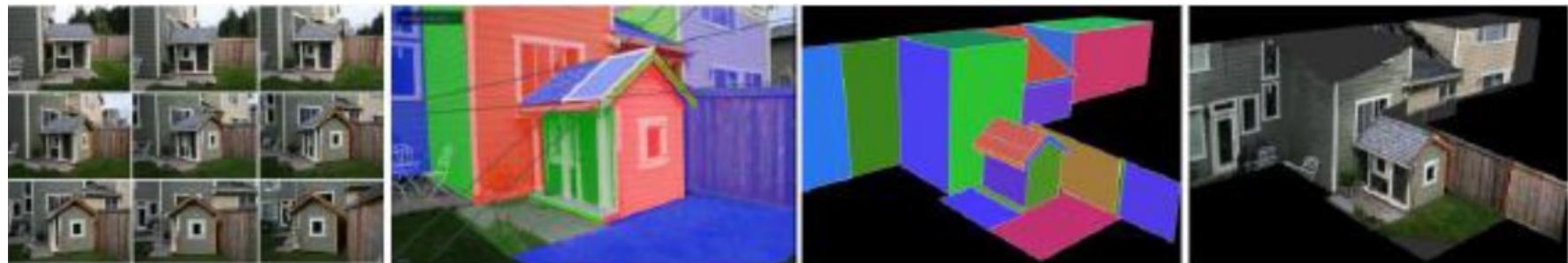


# Application: Robotique



Assistance et monitoring

## Application: Création d'images panoramiques



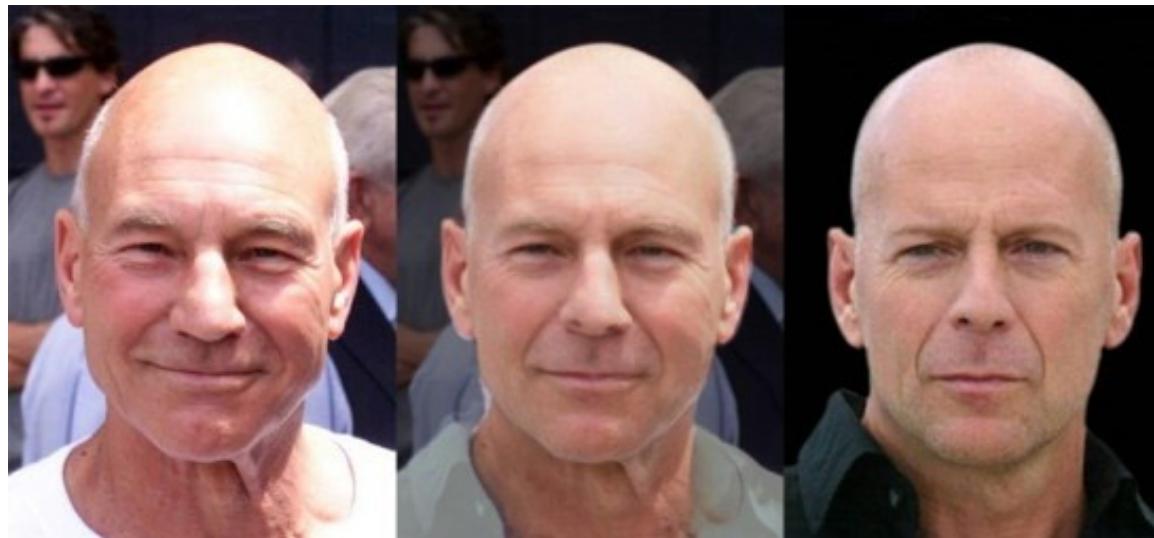
## Application: Correction d'exposition

---



## Application: Morphing

---



## Application: Télécommunication

---

Compression d'images, de vidéos. Transferts de données plus rapides.

Tatouage, une signature cachée dans les données de l'image. Cette approche a l'avantage de ne pas gêner la lecture de l'image par le simple spectateur tout en permettant une facile identification.

## Application: Renforcement de contraste



**Avant ajustement du contraste**



**Après ajustement du contraste**

(renforcement du contraste de l'élan et de l'herbe)

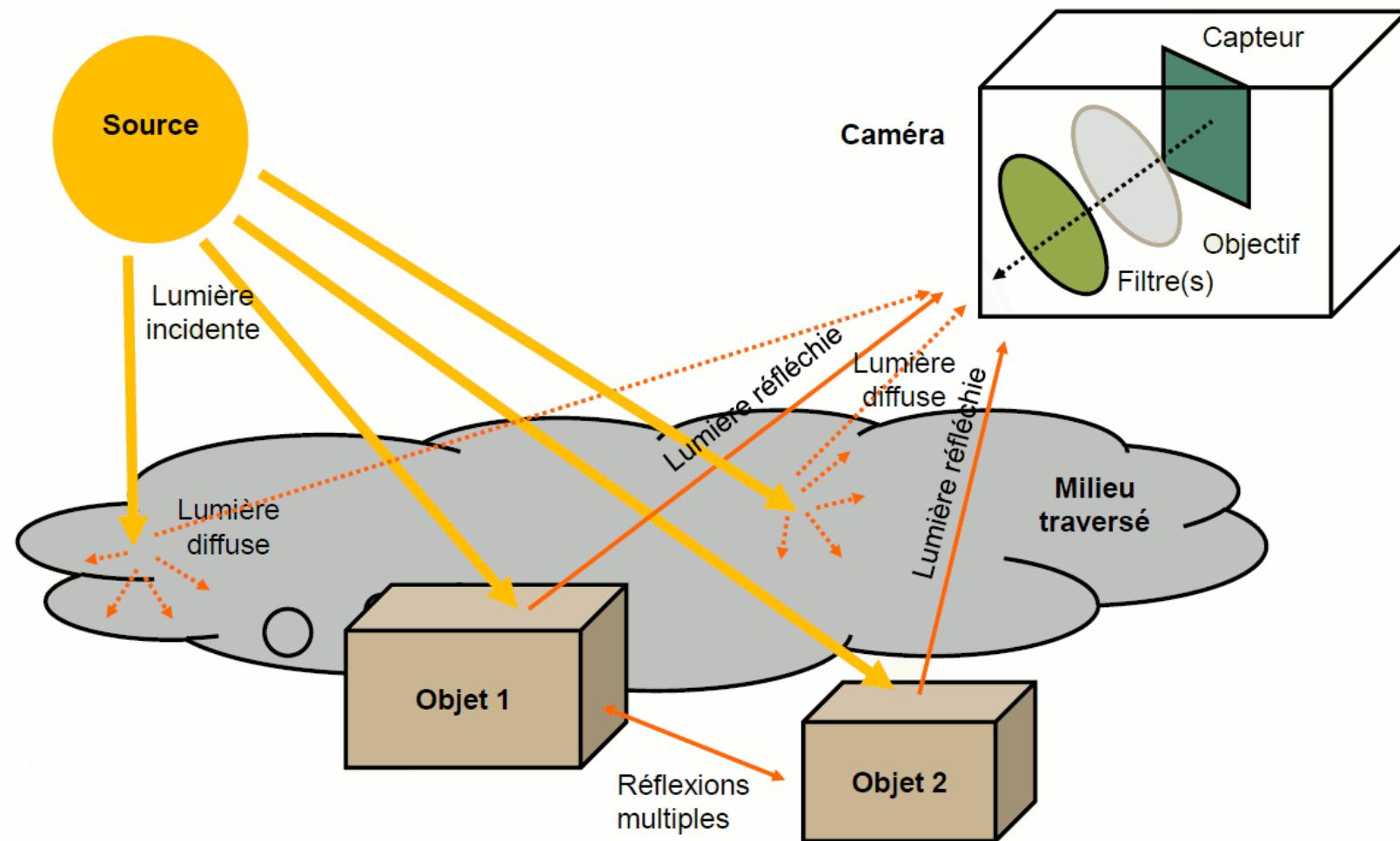
# Application: Photographie

Filtre anti yeux rouges

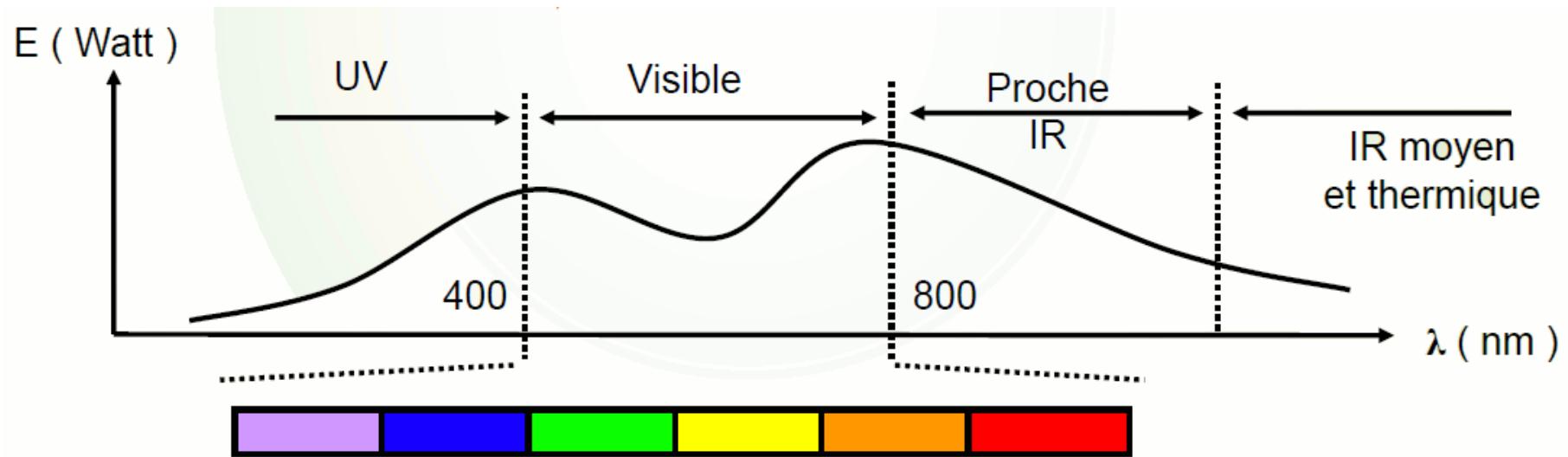


# Formation des images

# Phénomènes physiques

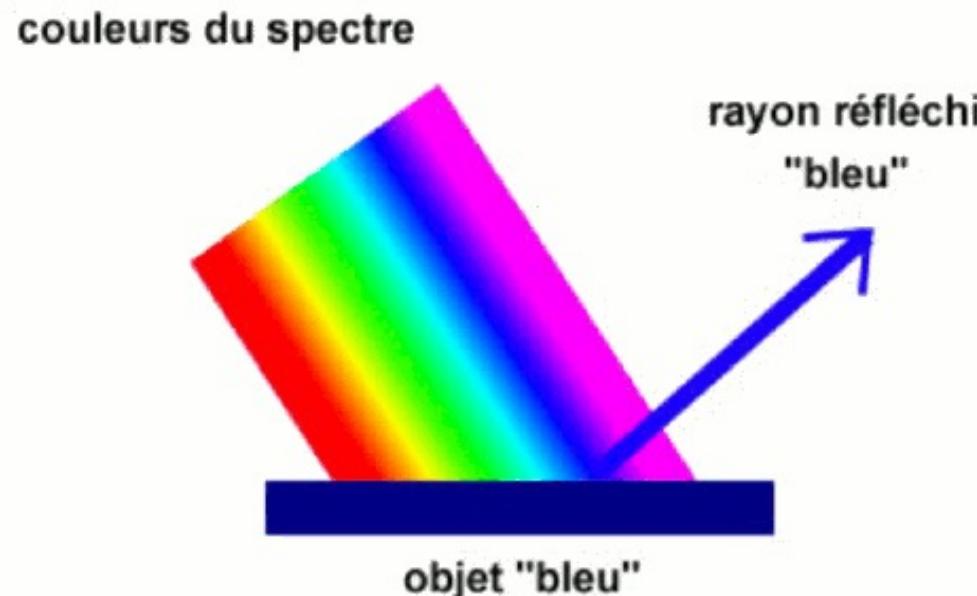


# Phénomènes physiques



Chaque longueur d'onde correspond à la perception d'une couleur

# Phénomènes physiques

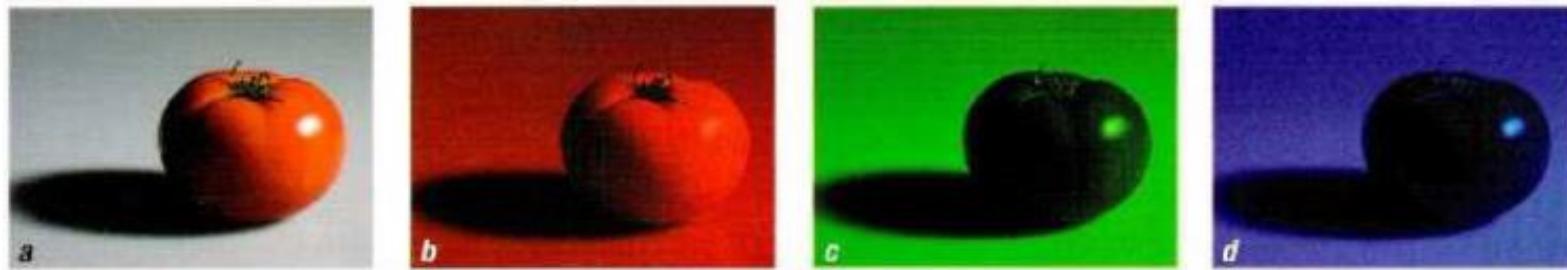


La lumière blanche est composée de toutes les couleurs

En fonction de la couleur de l'objet, certaines longueurs d'onde sont réfléchies et transmises (le bleu ici) tandis que d'autres sont absorbées (tout ce qui n'est pas bleu)

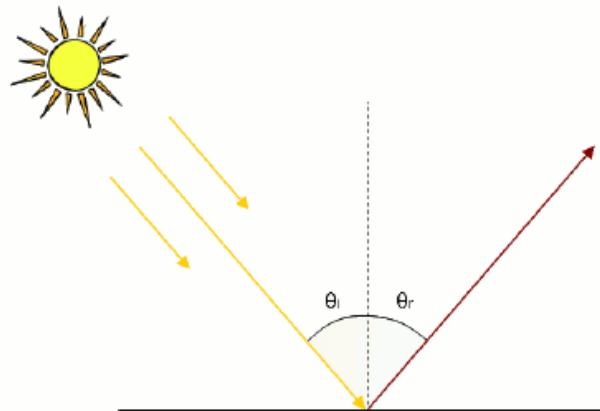
# Phénomènes physiques

---

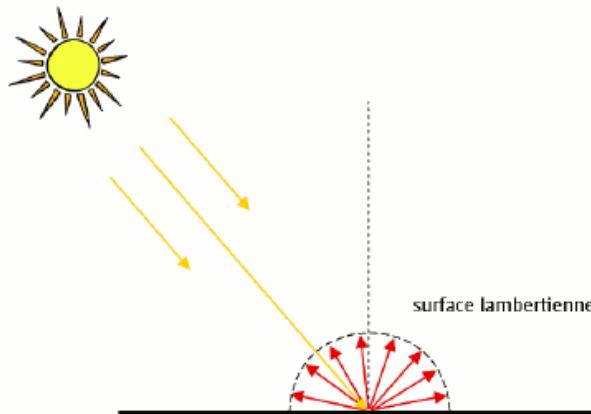


Une tomate éclairée successivement par la lumière blanche, rouge, verte et bleue.

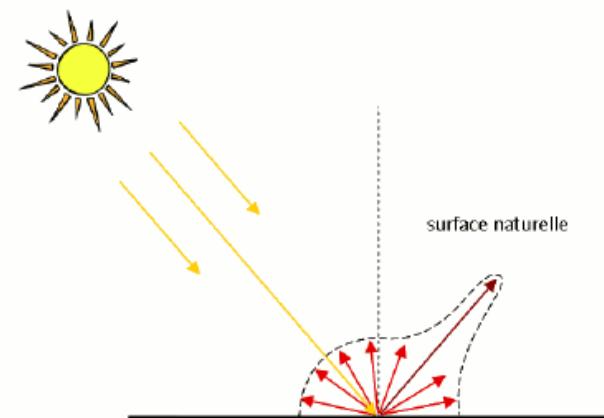
# Phénomènes physiques



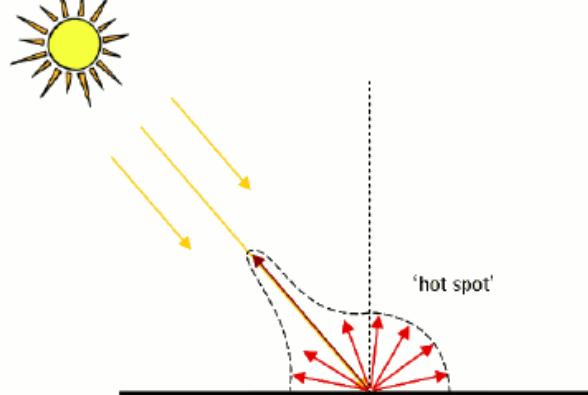
Réflexion spéculaire



Réflexion lambertienne



Réflexion Diffuse



# Phénomènes physiques

Pour la réflexion spéculaire

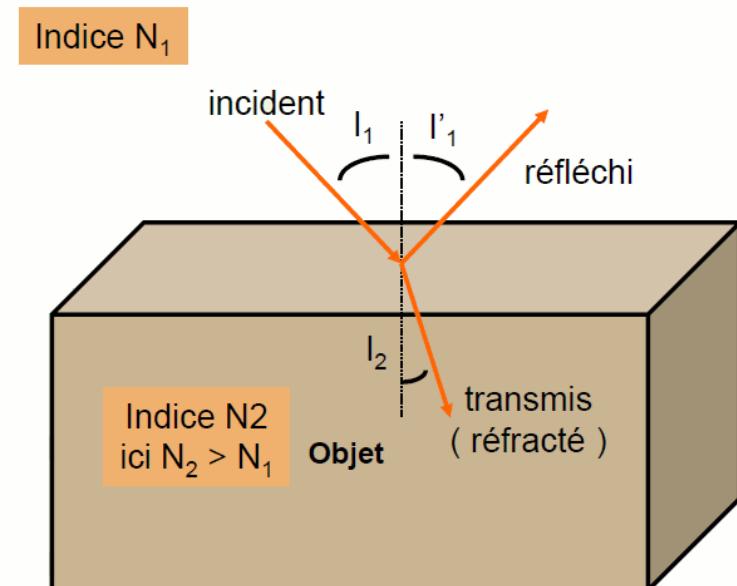
Loi de Descartes :

$$I_1 = I'_1 \text{ et } N_1 \sin(I_1) = N_2 \sin(I_2)$$

L'indice  $N_i$  est fonction de  $\lambda$  :

$N$  augmente si  $\lambda$  diminue

$N$  augmente si température diminue



Exemple :

$\lambda$  rouge >  $\lambda$  bleu donc  $N$  rouge <  $N$  bleu et  $I_2$  rouge >  $I_2$  bleu

# Phénomènes physiques

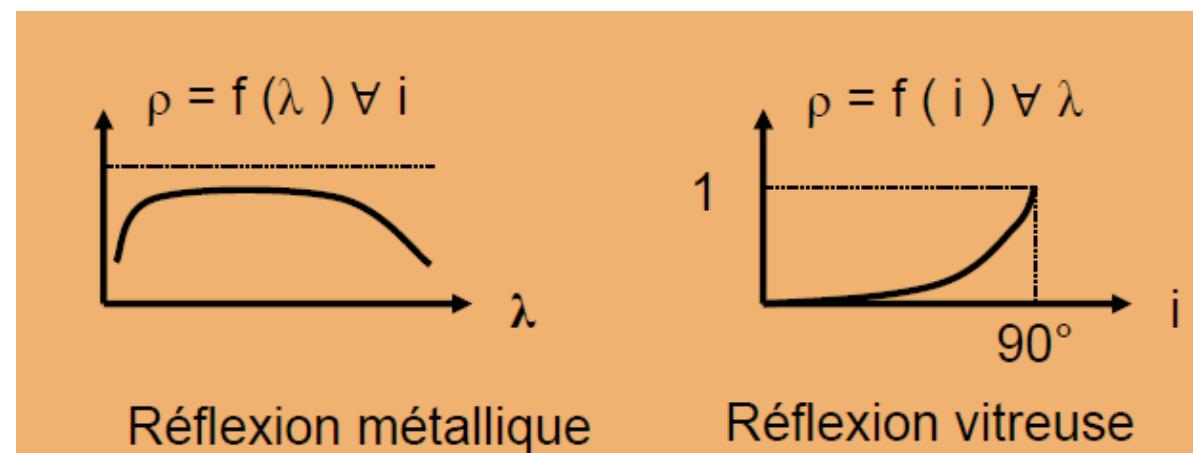
Réflexion / absorption / transmission =  $f(\lambda, i)$

Coefficients de réflexion et transmission :

$$\rho = E_{\text{réfléchi}} / E_{\text{incident}}$$

$$\tau = E_{\text{transmis}} / E_{\text{incident}}$$

$$\rho + \tau + \text{absorption} = 1$$



# Présentation des images

# Présentation des images

- ▶ Représentation des images
- ▶ Histogramme et histogramme cumulé
- ▶ Images couleur
- ▶ Convention et topologie

## Représentation des images

---

Une image numérique est associée à :

- Un échantillonnage spatial
- Une quantification de l'intensité lumineuse

Image = matrice de dimension  $N_{\text{lignes}} \times N_{\text{colonnes}}$

Chaque élément de la matrice contient la luminance du pixel codée en niveau de gris variant de 1 à N

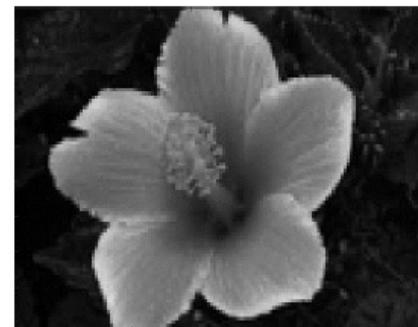
Souvent, on utilise 8 bits pour représenter les pixels, soit 256 niveaux de gris, avec 0 = noir et 255 = blanc

# Représentation des images

Effet de l'échantillonnage spatial



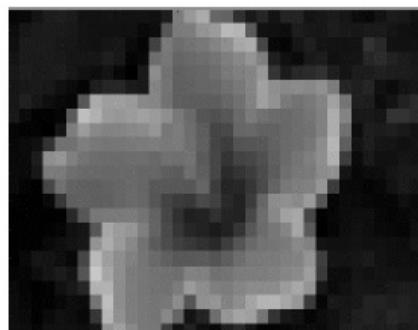
183x275 pixels



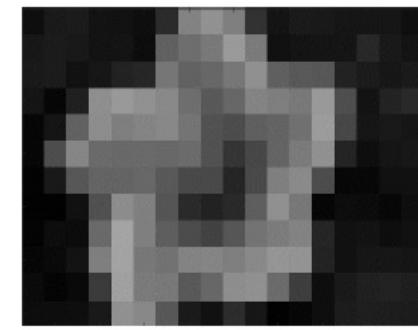
92x138 pixels



46x69 pixels



23x35 pixels



12x18 pixels

# Représentation des images

Effet de la quantification de la luminance en niveaux de gris



8 bits (256 ndg)



6 bits (64 ndg)



5 bits ( 32 ndg)



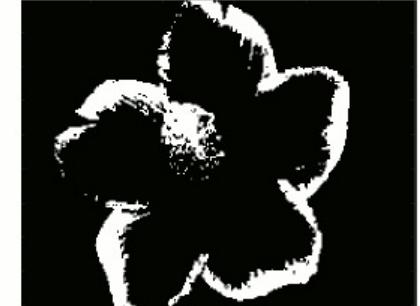
4 bits (16 ndg)



3 bits (8 ndg)



2 bits (4 ndg)

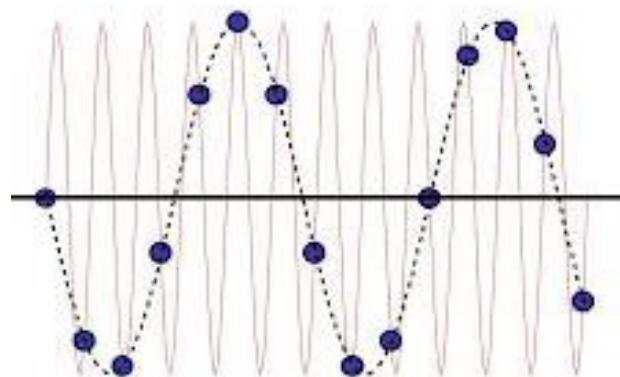


1 bit (2 ndg)

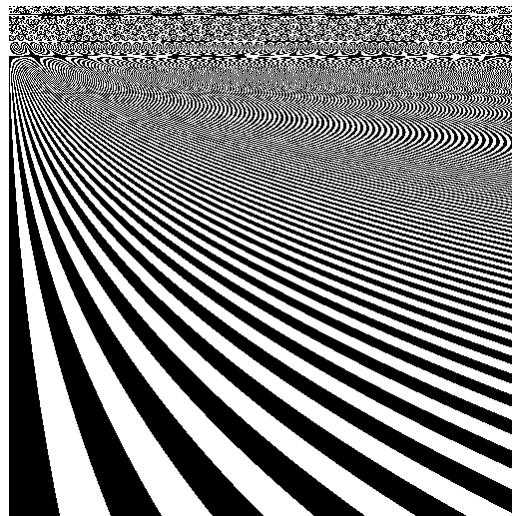
# Représentation des images

## Importance de l'échantillonnage

En 1D



En 2D

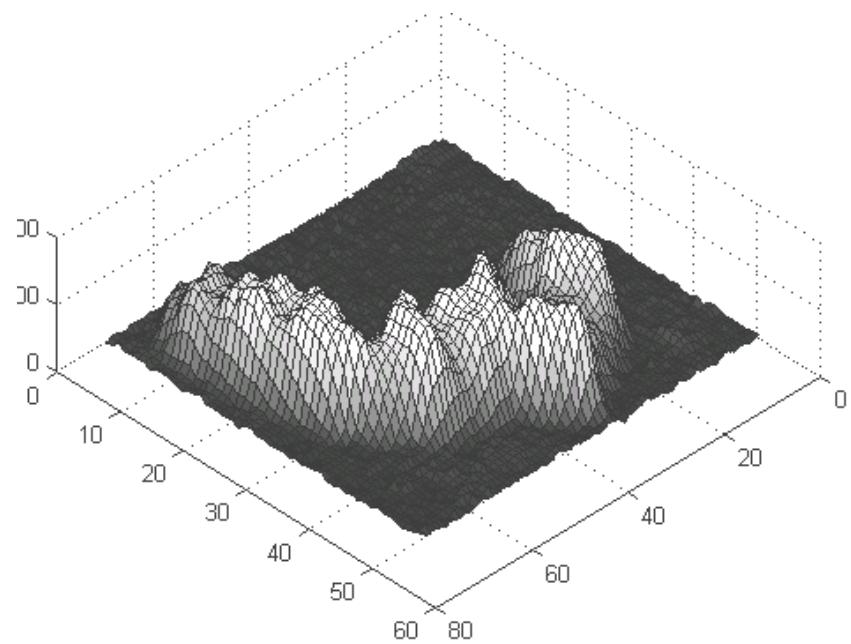


Attention quand on  
échantillonne sans vérifier  
Shannon !  
 $F_e \geq 2 \times F_{max}$

# Représentation des images

3 façons de représenter les images :

- Manière usuelle
- Comme une surface
- Visualiser les valeurs de la matrice



255	255	254	254	254	254	254	254
255	255	254	254	254	254	254	254
254	254	254	254	254	254	254	254
252	253	253	254	254	254	254	254
254	255	253	254	254	254	254	254
249	253	251	253	254	254	254	254
249	251	242	246	251	255	254	254
228	225	211	218	237	251	254	255
203	187	165	183	221	247	255	255
172	142	136	160	214	246	254	255
153	133	129	167	228	252	255	255
147	126	126	189	245	255	255	255
148	128	112	209	254	255	255	255
162	135	128	219	247	255	255	255

# Représentation des images

---

Lire une image sous Matlab :

```
I = imread('image.png');
```

Pour afficher une image en niveau de gris :

```
imagesc(I);  
colormap(gray);
```

Pour afficher une image comme une surface :

```
surf(I);
```

# Présentation des images

- ▶ Représentation des images
- ▶ Histogramme et histogramme cumulé
- ▶ Images couleur
- ▶ Convention et topologie

# Histogramme et histogramme cumulé

---

## Histogramme

C'est un vecteur de dimension 256

Chaque élément  $h(i)$  représente le nombre de pixels de l'image possédant le niveau de gris  $i$

Analogie avec la densité de probabilité des intensités lumineuses (à un facteur de normalisation près)

Sous Matlab, on calcule l'histogramme avec :

```
hist(I(:),[0:255])
```

# Histogramme et histogramme cumulé

---

## Histogramme cumulé

C'est un vecteur de dimension 256

Chaque élément  $hc(i)$  représente le nombre de pixels de l'image possédant le niveau de gris inférieur ou égal à  $i$

Analogie avec la fonction de répartition des niveaux de gris

Il peut être estimé à partir de l'histogramme en faisant une somme discrète

$$hc(i) = \sum_{j=0}^i h(j)$$

# Présentation des images

- ▶ Représentation des images
- ▶ Histogramme et histogramme cumulé
- ▶ Images couleur
- ▶ Convention et topologie

# Image couleur

Image couleur = 3 plans couleur

Pour la plupart des caméras : Rouge Vert Bleu (RVB)

Chaque plan est codé comme une image niveaux de gris, avec des valeurs allant de 0 à 255

Lorsque R=V=B, la couleur associée est un niveau de gris

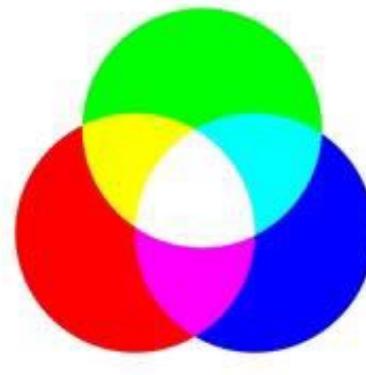
Pour passer d'une image couleur à une image niveau de gris on réalise :

$$I(y, x) = \frac{R(y, x) + V(y, x) + B(y, x)}{3}$$

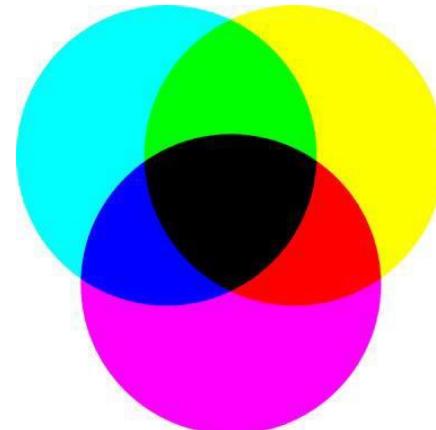


## Image couleur

Synthèse additive des couleurs utilisée pour les sources lumineuses chromatiques



Synthèse soustractive des couleurs utilisée pour les pigments colorés



## Image couleur

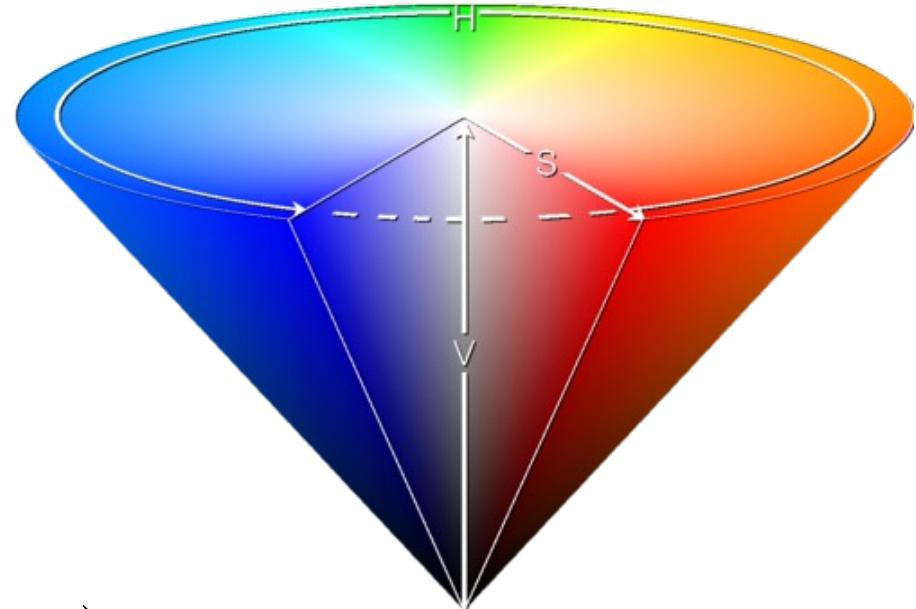
**H** : teinte (varie entre 0 et  $2\pi$ )

**S** : saturation (varie entre 0 et 1)

Une couleur très saturée qui possède une faible proportion de blanc se trouvera loin de l'axe des intensités

**V** : intensité lumineuse (varie entre 0 et 1)

Plus la valeur est faible, plus la couleur est sombre



# Image couleur

---

Pour passer de RGB vers HSV

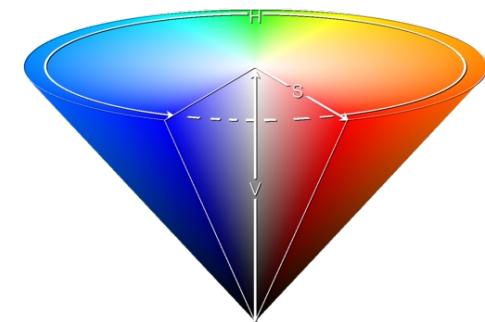
$$t = \begin{cases} 0, & \text{si } \max = \min \\ (60^\circ \times \frac{g-b}{\max - \min} + 360^\circ) \bmod 360^\circ, & \text{si } \max = r \\ 60^\circ \times \frac{b-r}{\max - \min} + 120^\circ, & \text{si } \max = g \\ 60^\circ \times \frac{r-g}{\max - \min} + 240^\circ, & \text{si } \max = b \end{cases}$$

$$s = \begin{cases} 0, & \text{si } \max = 0 \\ 1 - \frac{\min}{\max}, & \text{sinon} \end{cases}$$

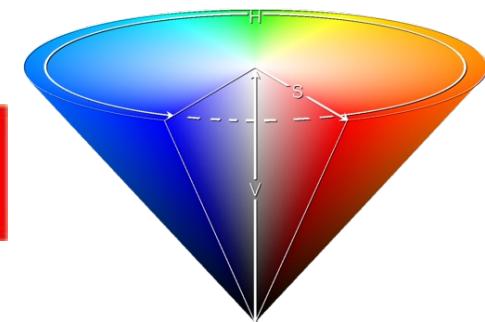
$$v = \max$$

## Image couleur

Dans les mêmes conditions ( $S = V = 1$ ), en faisant varier  $H$  de  $0$  à  $360^\circ$  on obtient un gradient de couleurs

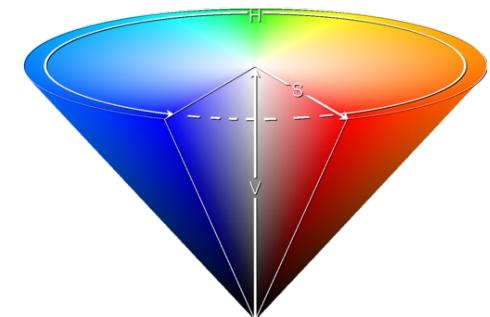


En prenant comme exemple la couleur rouge ( $H = 0$ ), en fixant  $V = 1$  (luminosité maximale) et en faisant varier  $S$  de  $0$  à  $1$  on obtient un gradient qui va du blanc jusqu'au rouge vif

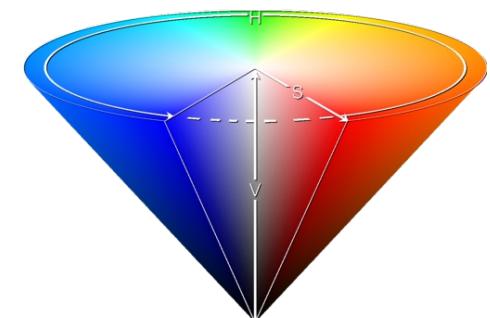


## Image couleur

En gardant la couleur rouge ( $H = 0$ ), en fixant  $S = 1$  (saturation maximale) et en faisant varier  $V$  de 0 à 1 on obtient un gradient qui va du noir au rouge vif



Si maintenant on se place au centre du cercle ( $S = 0$  et  $H$  quelconque) et que l'on fit varier  $V$  de 0 à 1 on obtient une échelle de gris



# Image couleur

---

Il existe d'autres systèmes de couleur

## Lab

- Un paramètre d'intensité (L) et de deux paramètres de chrominance
- Spécialement étudié pour que les distances calculées entre couleurs correspondent aux différences perçues par l'œil humain

## YUV

- La première représente la luminance et les deux autres représentent la chrominance
- YUV est utilisé dans les systèmes de diffusion télévisuelle PAL et NTSC
- Le modèle YUV se rapproche beaucoup plus de la perception humaine des couleurs que le standard RGB

## Image couleur

On peut aussi utiliser des couleurs indexées, chaque pixel est alors caractérisé par une valeur qui pointe vers un tableau

Lut niveaux de gris

	R	V	B
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
254	254	254	254
255	255	255	255

Lut JET utilisée par défaut par Matlab

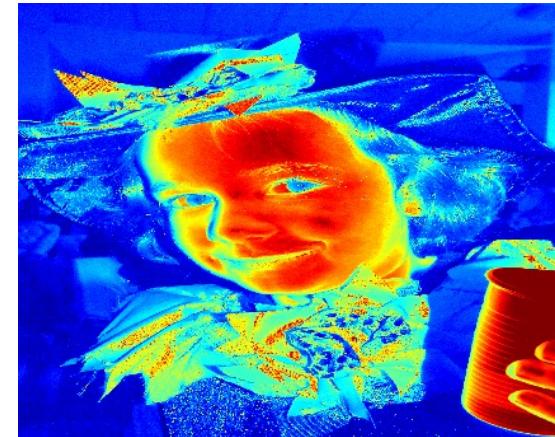
	R	V	B
0	0	0	144
1	0	0	160
2	0	0	176
3	0	0	192
4	0	0	208
5	0	0	224
254	144	0	0
255	128	0	0

# Image couleur

Niveaux de gris



Jet



Hot



Pink



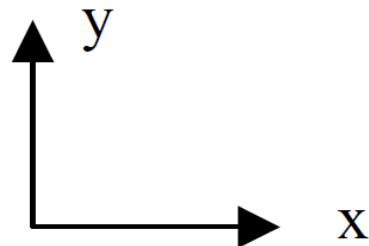
# Présentation des images

- ▶ Représentation des images
- ▶ Histogramme et histogramme cumulé
- ▶ Images couleur
- ▶ Convention et topologie

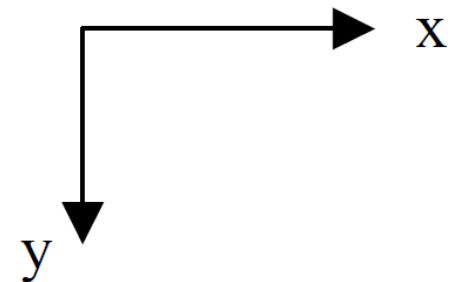
# Convention et topologie

Le système de convention des axes et le suivant :

En analogique



En discret



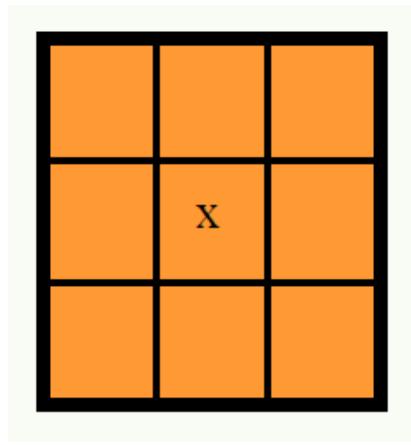
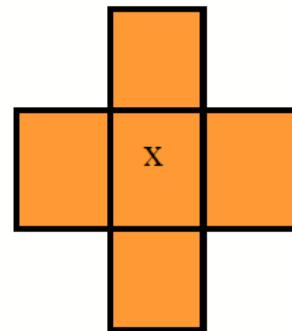
La ligne 0 d'une image est celle située en haut

# Convention et topologie

## Notions de topologie

L'image est représentée par un ensemble de pixels disposés sous la forme d'une grille

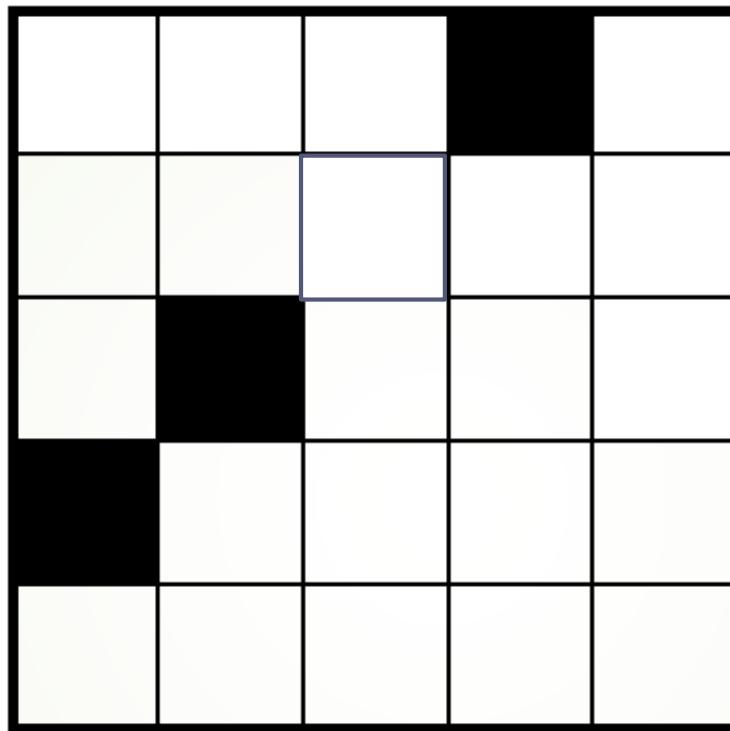
Quels sont les voisins d'un pixel de coordonées  $(x,y)$  ?



Selon les applications, on pourra se placer en 4 connexités ou en 8 connexités.

# Convention et topologie

## Notions de topologie



combien de régions?

On conserve la 8-connectivité des contours pour garder une 4-connectivité des régions lors des opérations de traitement d'images

---

# Amélioration et restauration d'images

## Diminution du bruit dans les images



# Amélioration et restauration d'images

## Diminution du bruit dans les images

SOS

**Amélioration d'images** : Modifier l'apparence d'une image pour qu'un observateur puisse plus facilement extraire des informations

**Restauration d'images** : Diminuer les dégradations, inverser un phénomène dégradant (supposé connu et modélisé)

**Diminution du bruit dans les images** : Réduire le bruit présent dans les images sans avoir de connaissance a priori sur celui-ci (aucun modèle). Ce prétraitement a pour but de faciliter les traitements ultérieurs dans la chaîne

# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

### Restauration

#### Filtrage

#### Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

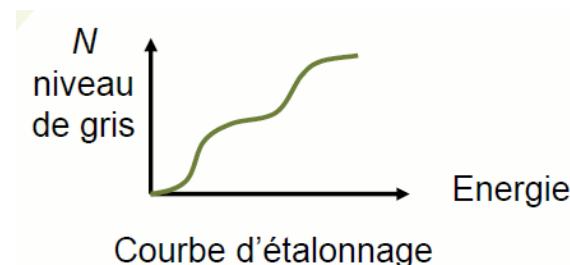
#### Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

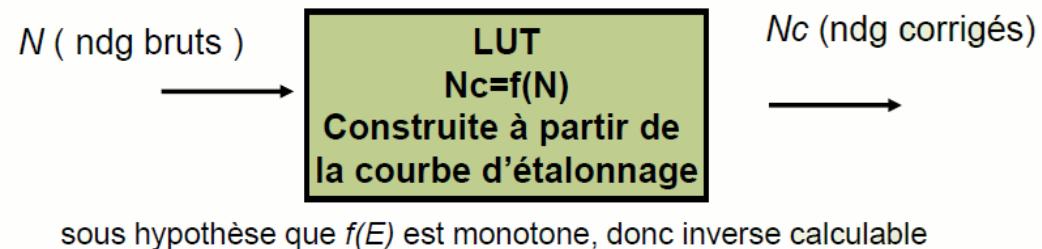
# Correction photométriques

Correction de capteur (gain, non linéarité, etc) → étalonnage expérimental

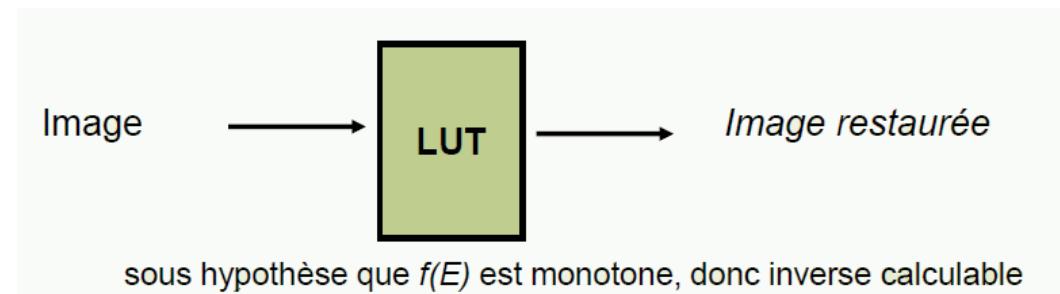
Etalonnage : modélisation du phénomène dégradant



Construction d'une LUT



Restauration



## Correction d'uniformité d'éclairage

L'exposition n'est pas homogène sur toute l'image → modélisation

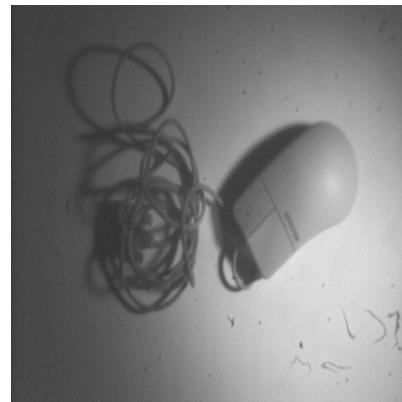
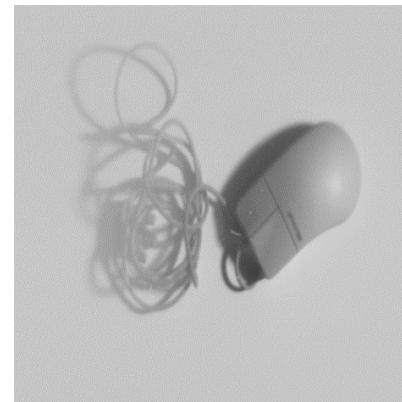


Image d'origine  $I_1$



Correction additive

$$I_r = I_1 - I_0 + \max(I_0)$$

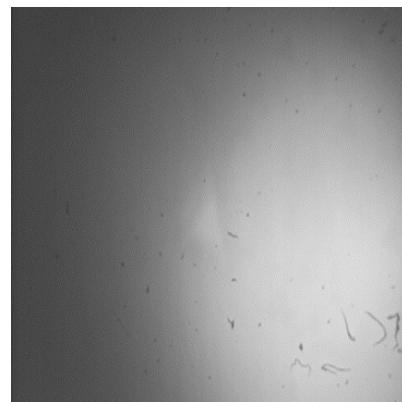
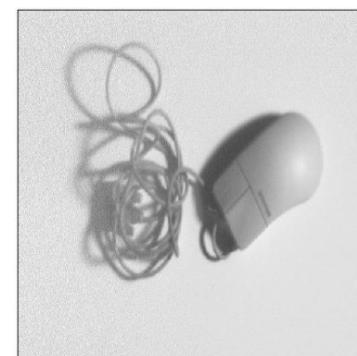


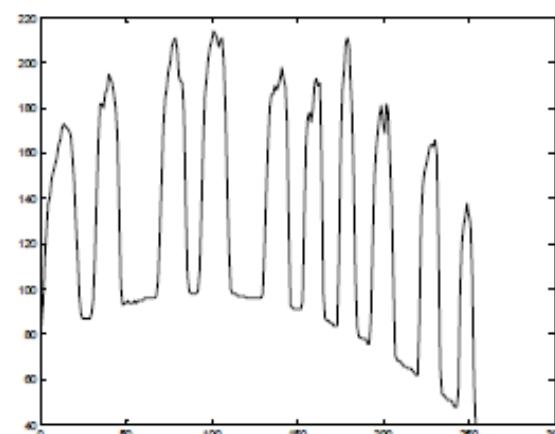
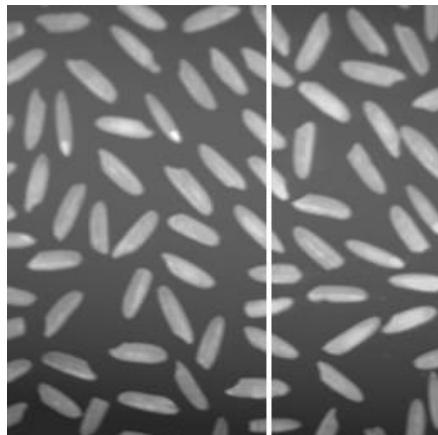
Image de référence  $I_0$



Correction multiplicative

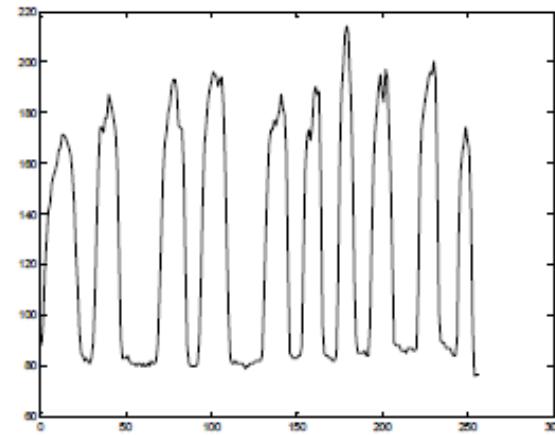
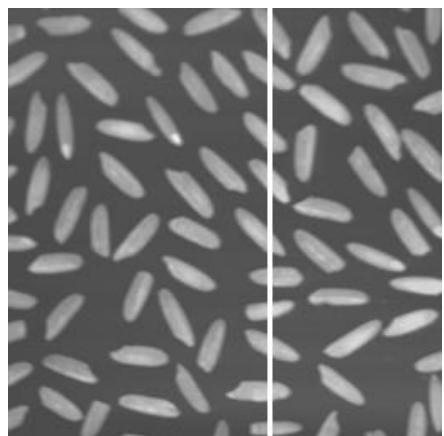
$$I_r = (I_1/I_0) \cdot \max(I_0)$$

# Correction d'uniformité d'éclairage



## Modélisation

Création d'une image représentant l'éclairage en prenant le minimum de chaque ligne (éclairage supposé constant le long des lignes)



## Restauration

On restaure une image avec un éclairage constant

$I_m = \text{Fond} + \text{moyenne du fond}$

On conserve un niveau moyen de fond)

# Corrections géométriques

## Transformations planes

### Translation

$$\begin{pmatrix} y' \\ x' \end{pmatrix} = [I|t] \begin{pmatrix} y \\ x \end{pmatrix} = I^* \begin{pmatrix} y \\ x \end{pmatrix} + t = \left[ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} | \begin{pmatrix} tx \\ ty \end{pmatrix} \right] \begin{pmatrix} y \\ x \end{pmatrix}$$

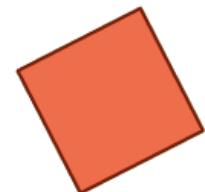
2 paramètres



### Transformation rigide (euclidienne)

$$\begin{pmatrix} y' \\ x' \end{pmatrix} = [R|t] \begin{pmatrix} y \\ x \end{pmatrix} = \left[ \begin{pmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{pmatrix} | \begin{pmatrix} tx \\ ty \end{pmatrix} \right] \begin{pmatrix} y \\ x \end{pmatrix}$$

3 paramètres



### Similarité

$$\begin{pmatrix} y' \\ x' \end{pmatrix} = [sR|t] \begin{pmatrix} y \\ x \end{pmatrix} \text{ avec } s \text{ scalaire}$$

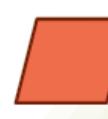
4 paramètres



### Projection affine

$$\begin{pmatrix} y' \\ x' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} (sR|t) \begin{pmatrix} y \\ x \end{pmatrix}$$

6 paramètres



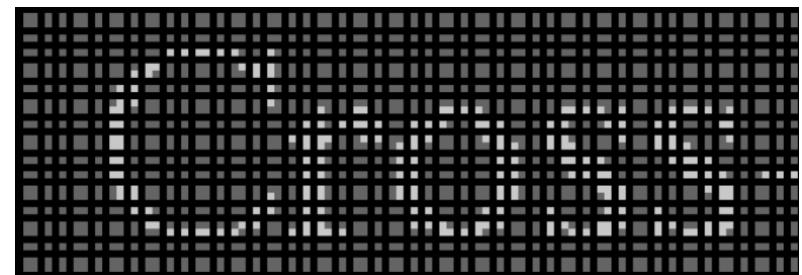
## Changement d'échelle

Exemple : zoom d'une image binaire avec  $\alpha = \alpha_x = \alpha_y = 1.7$



**Transformation directe** : calcul de la transformée d'un pixel de l'image initiale → phénomène de trous

$$I(y,x) \rightarrow I_z(\text{round}(\alpha.y), \text{round}(\alpha.x))$$



**Transformation inverse** : si matrice de transformation inversible, un pixel de l'image résultat est déterminé par son pixel correspondant dans l'image initiale

$$I_z(y,x) \leftarrow I(\text{round}(y/\alpha), \text{round}(x/\alpha))$$

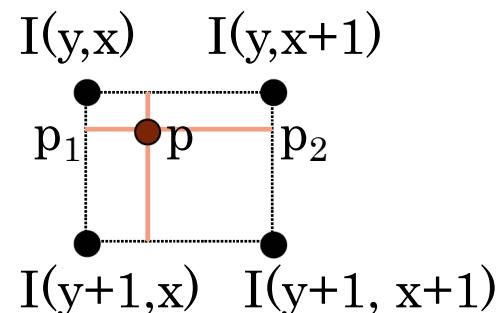
On prendra le voisin le plus proche



Il existe d'autres interpolations que le voisin le plus proche

# Changement d'échelle avec interpolation

**Interpolation bi-linéaire** : on prend en compte les 4 voisins proches du pixel de l'image d'origine pour déterminer l'intensité du pixel final



$$\begin{aligned}
I(p_1) &= (1-dy).I(y, x) + dy.I(y+1, x) \\
I(p_2) &= (1-dy).I(y, x+1) + dy.I(y+1, x+1) \\
I(p) &= (1-dx).I(p_1) + dx.I(p_2)
\end{aligned}$$

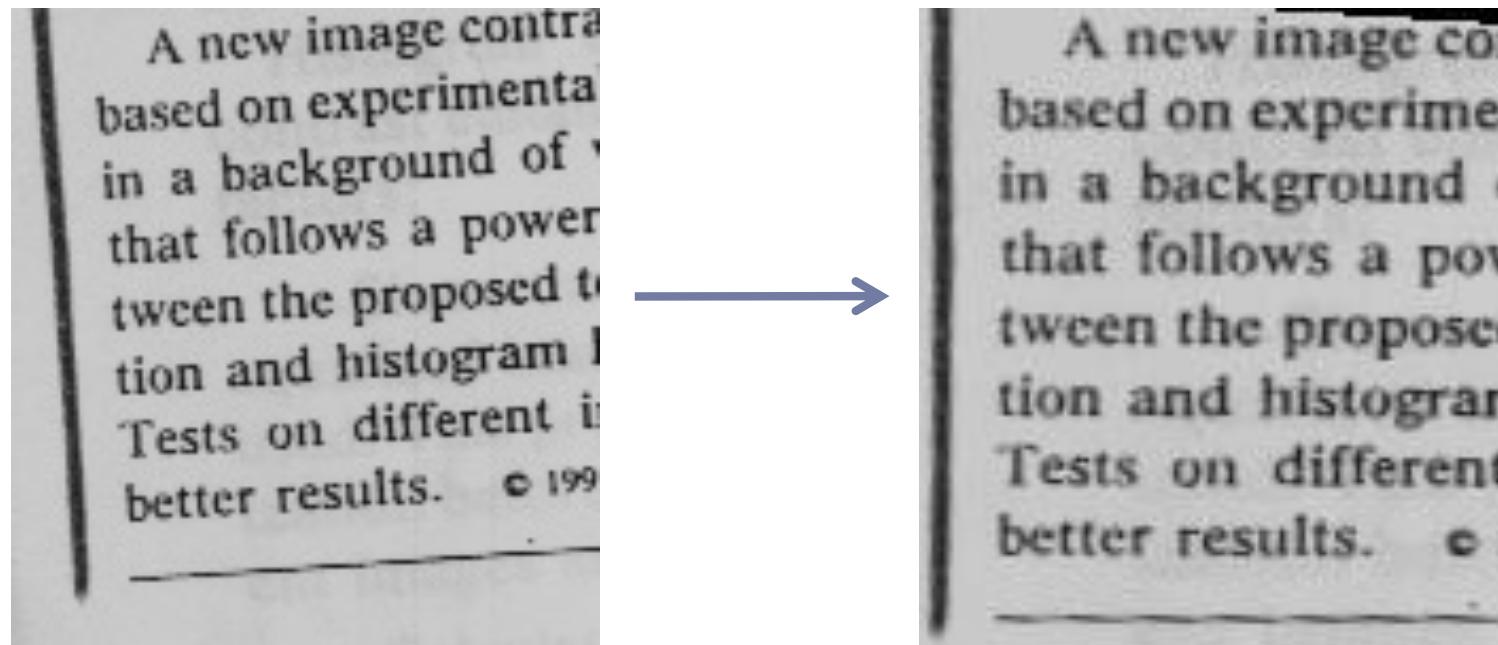
**Interpolation bi-cubique** : on prend en compte les 16 voisins proches du pixel de l'image d'origine pour déterminer l'intensité du pixel final



Et bien d'autres : Hermite, Mitchell, Lanczos...

# Redressage d'images

Redressement du texte : rotation + translation + zoom



# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

# Filtrage

## Rappel de traitement du signal

La convolution 2D de l'image  $I$  avec un filtre  $F$  est donnée par :

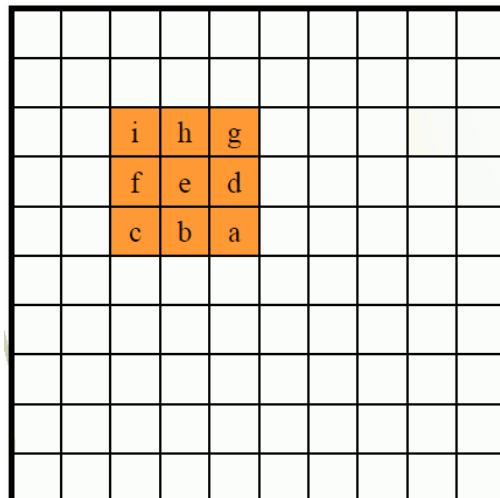
$$(I \otimes F)(x, y) = IF(x, y) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} I(x-n, y-m).F(n, m)$$

$$F = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Où  $F(n, m)$  est la réponse impulsionnelle du filtre.

L'élément central  $e$  correspond à la position  $(0,0)$  dans la matrice.

La convolution est le cumul des produits terme à terme à condition de tourner les coefficients du filtre de  $180^\circ$ .



$$(I \otimes F)(x, y) = I(x-1, y-1)i + I(x, y-1)h + I(x+1, y-1)g + \\ I(x-1, y)f + I(x, y)e + I(x+1, y)d + \\ I(x-1, y+1)c + I(x, y+1)b + I(x+1, y+1)a$$

En traitement d'image, par abus d'écriture on présente directement le masque inversé.

## Filtrage

Attention aux effets de bords de 1 pixel pour un filtre 3x3

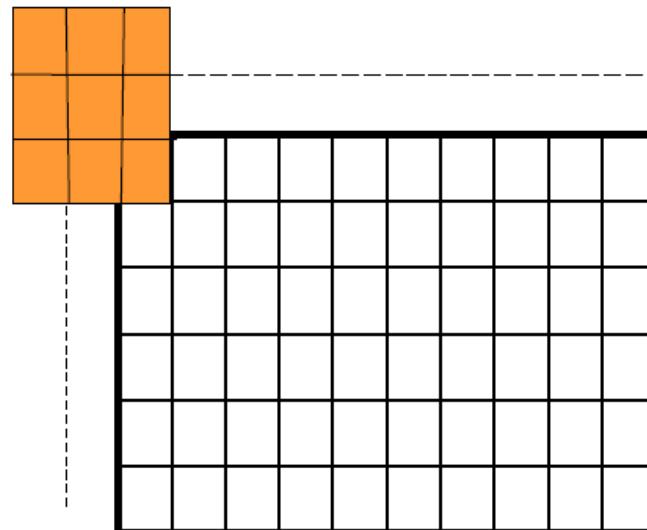
L'image convoluée est plus grande que l'image initiale

Sous Matlab on utilise :  $IF = conv2(I,F,'same')$

L'option *same* permet à IF d'être de la même dimension que I.

Pour gérer les bords, plusieurs méthodes existent pour agrandir l'image :

- Ajouter une bordure à zéro
- Faire un miroir de l'image
- Copier les derniers pixels



# Filtrage

---

## En fréquence

Comme pour un signal 1D, la transformée de Fourier du produit de convolution est le produit des transformées de Fourier.

Usuellement, les fréquences spatiales sont notées  $u$  et  $v$ .

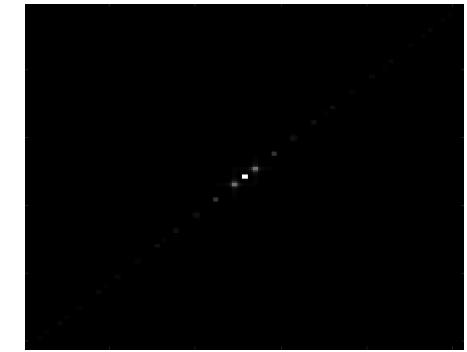
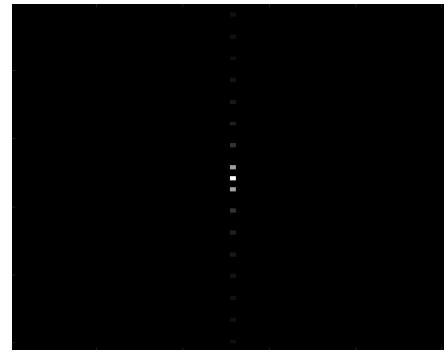
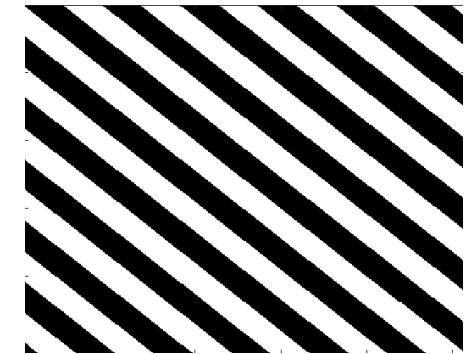
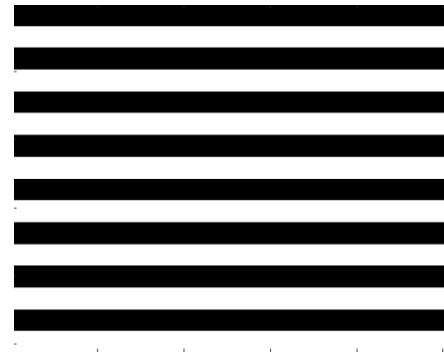
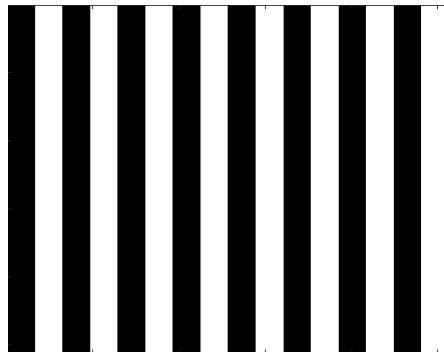
On a donc :

$$TF[I \otimes F(x, y)] = TF[I(x, y)]. TF[F(x, y)] = I(u, v). F(u, v)$$

En 2D les textures représentées dans cet espace vont produire des concentrations d'énergie dans le spectre de Fourier, traduisant une périodicité horizontale et verticale.

# Filtrage

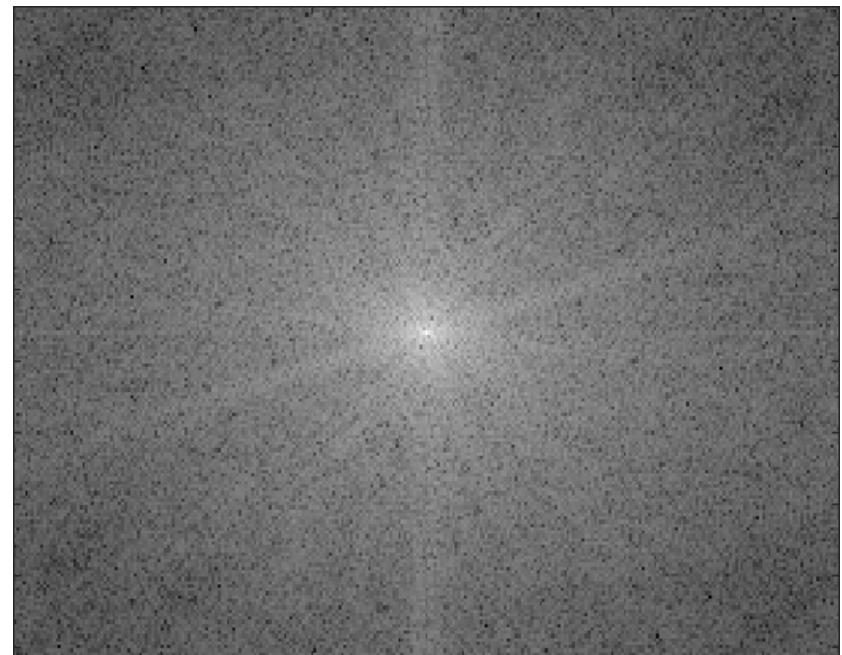
## Exemples



# Filtrage

---

## Exemples



# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

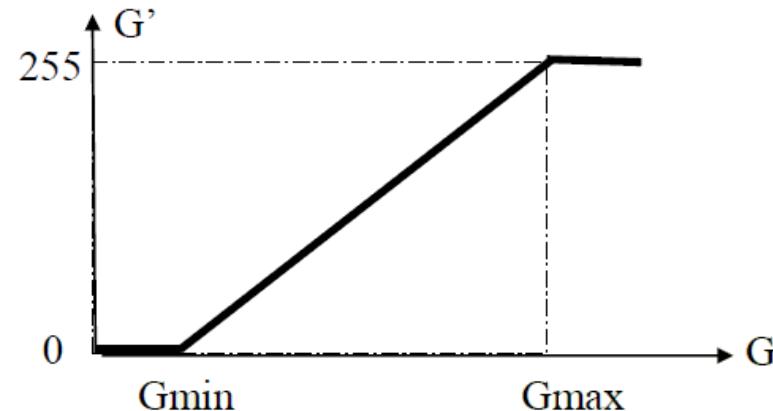
## Extension de dynamique

Luminance entre  $G_{min}$  et  $G_{max}$  d'une image → luminance entre 0 et 255.

Cette étendue réduite de niveaux de gris peut survenir suite à un temps de pose incorrecte ou à un éclairage de la scène trop faible.

La transformation mise en place est :

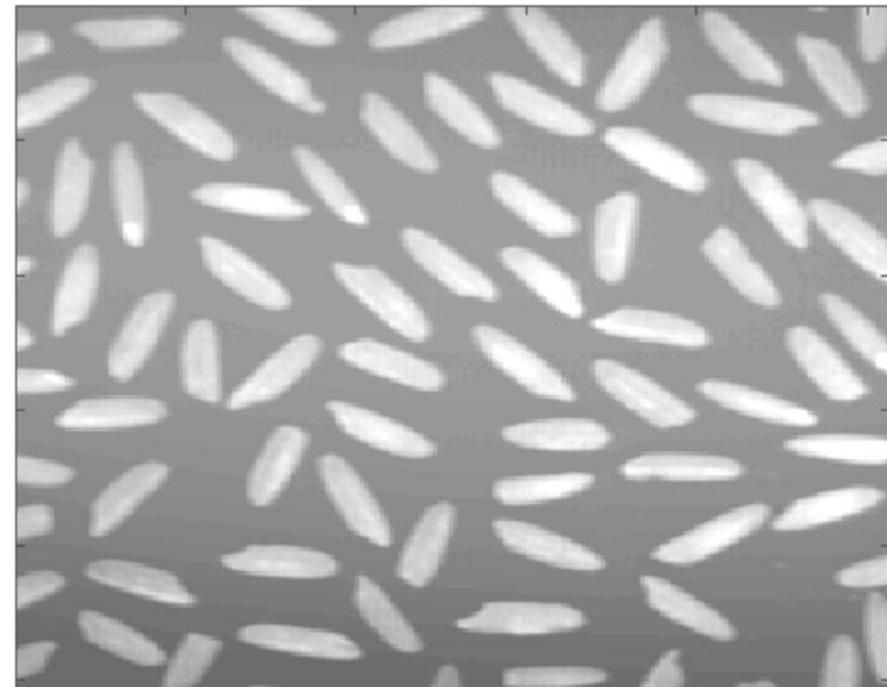
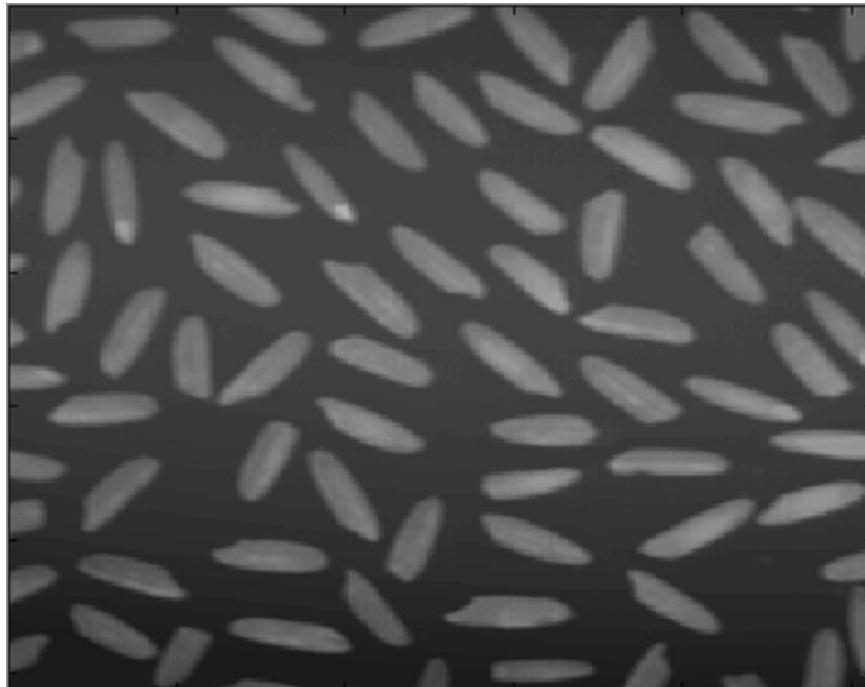
$$I_{new}(x, y) = \frac{I(x, y) - G_{min}}{G_{max} - G_{min}} * 255$$



# Extension de dynamique

---

Exemple



# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

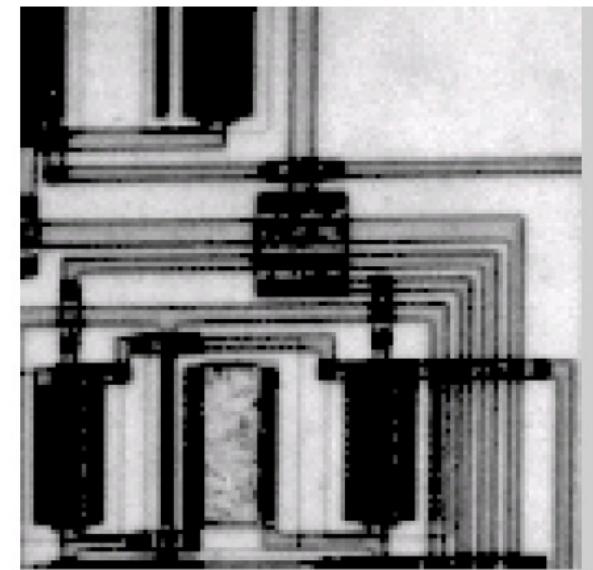
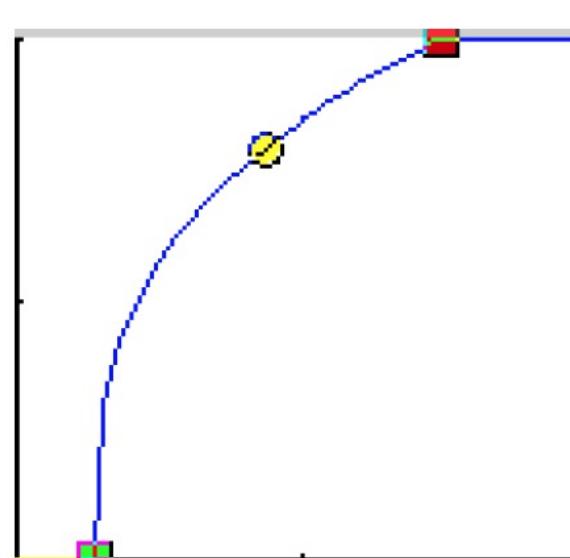
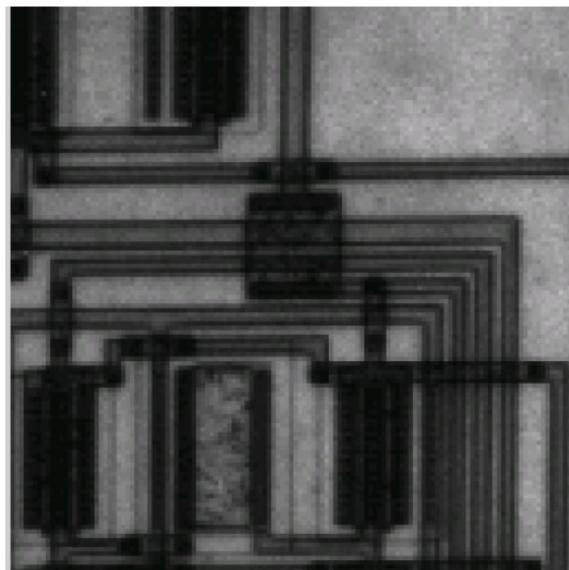
- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

## Correction d'exposition

On peut aussi être amené à vouloir renforcer certaine plage de niveaux de gris, au détriment d'autres plages pour mettre certains objets en valeur.

Dans ce cas, la transformation des luminances n'est plus linéaire.

Exemple, pour renforcer la présence des éléments d'un circuit électrique, on décide de renforcer la gamme des niveaux de gris sombres :



# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

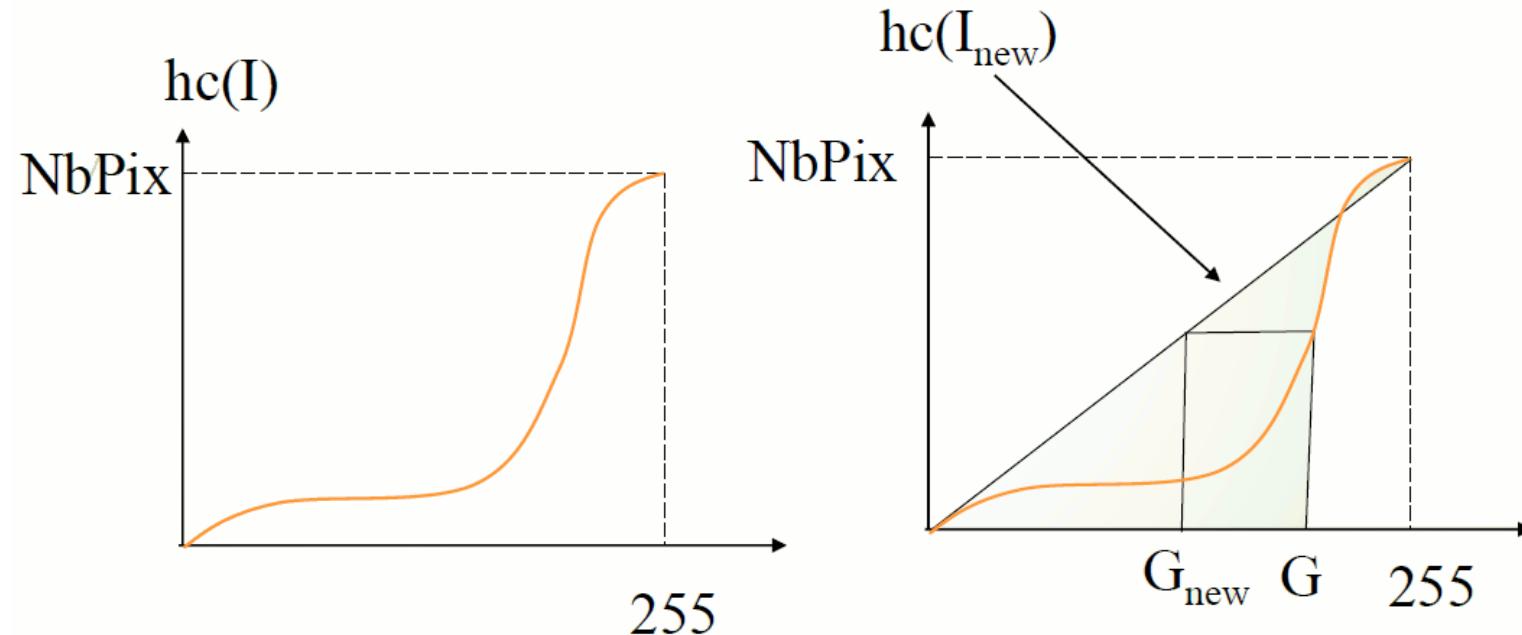
Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

## Egalisation d'histogramme

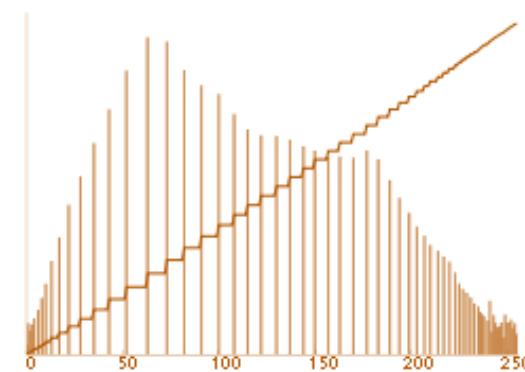
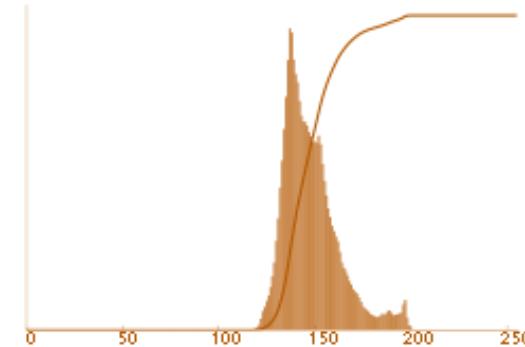
Cette opération à pour but de rendre l'histogramme le plus plat possible.

On souhaite ainsi que chaque niveau de gris soit également représenté dans l'image.



$$G_{new} = \frac{255}{NbPix} hc(G)$$

# Egalisation d'histogramme



## Egalisation d'histogramme adaptative

---

Le but est d'appliquer différentes égalisations en fonction des régions.

Appliquer l'égalisation d'histogramme dans des blocs de NxM pixels.

Défaut : rupture aux transitions.

Solution 1 : utiliser une fenêtre glissante et faire le calcul autour du pixel central, dans un bloc de taille MxN.

Défaut : très long.

Solution 2 : calculer les égalisations d'histogrammes dans des blocs non recouvrant de MxN pixels puis effectuer une interpolation bi-linéaire sur l'image pour lisser les différences entre blocs voisins.

# Egalisation d'histogramme adaptative



Egalisation globale



Image originale



Egalisation adaptive

# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

## Renforcement de la netteté

---

Ce traitement est utile pour supprimer les effets de flou dus, par exemple, à un bougé lors de la prise de vue.

On veut restaurer des contours francs sans toucher aux zones homogènes.

Flou = transition douce entre deux zones

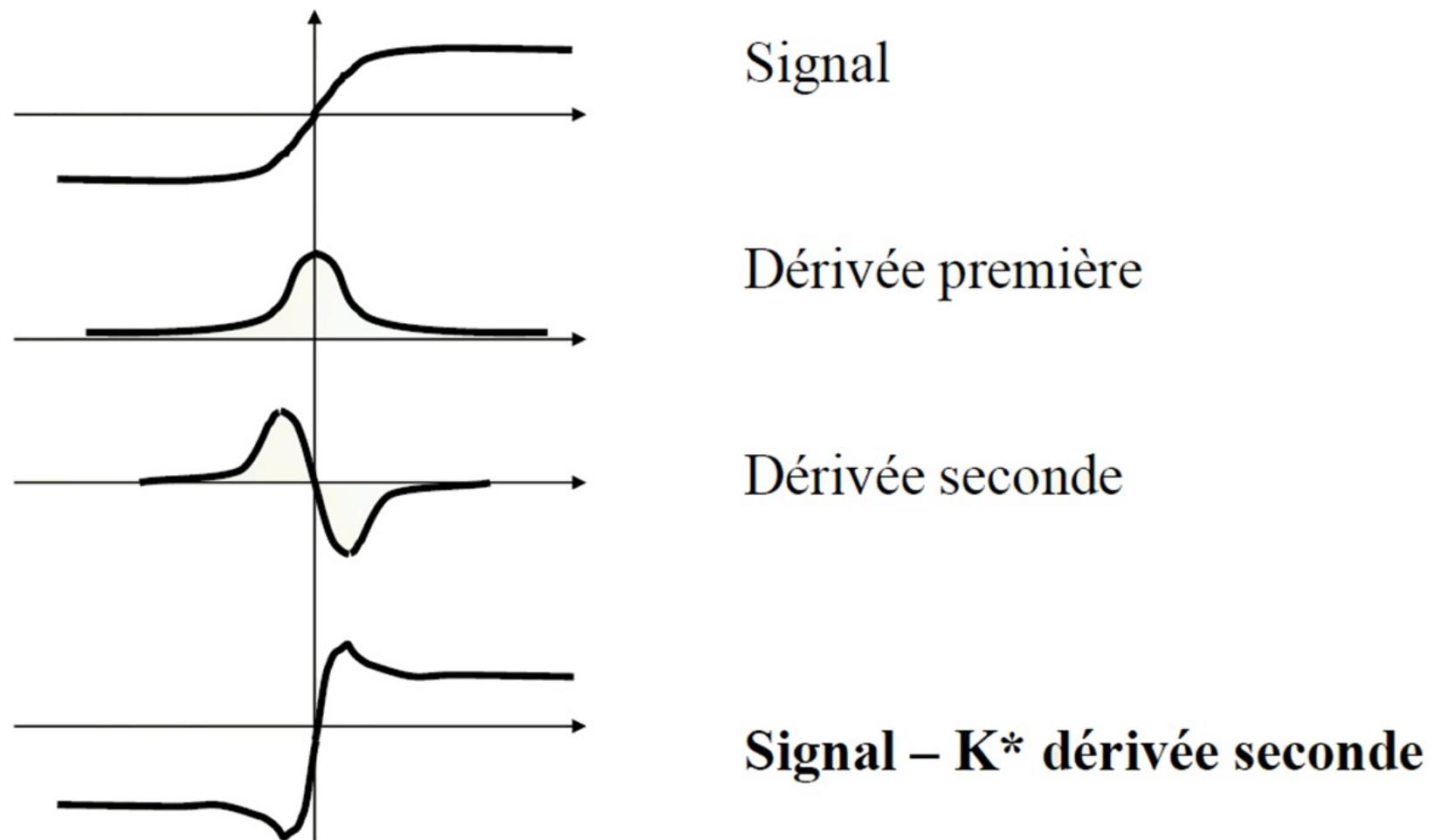
→ Amener à une transition plus rapide

Le traitement local prend en compte le voisinage de chaque pixel.

Le renforcement de la netteté est basé sur un calcul de Laplacien.

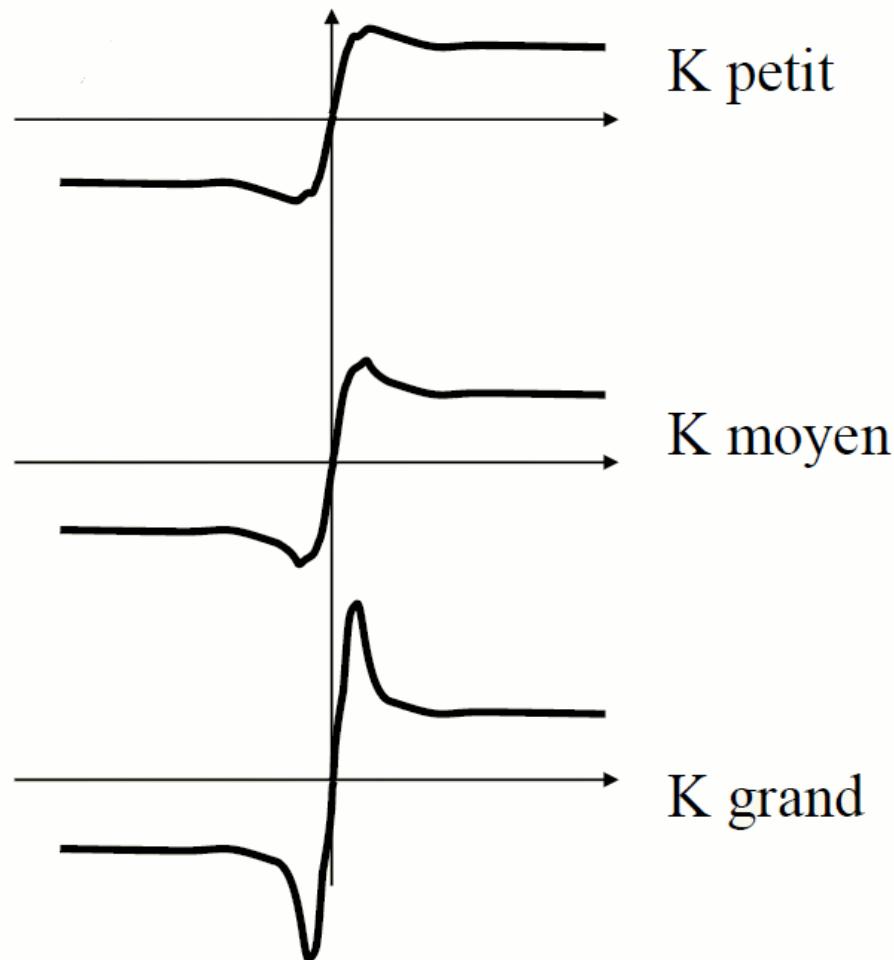
# Renforcement de la netteté

Exemple en 1D



# Renforcement de la netteté

Exemple en 1D



Selon les valeurs du paramètre  $K$ , on peut accentuer plus ou moins le phénomène

## Renforcement de la netteté

Considérons un signal discret 1D  $I(x)$ , sa dérivée première s'exprime par :

$$\text{et } I'\left(x - \frac{1}{2}\right) = I(x) - I(x-1)$$

$$I'\left(x + \frac{1}{2}\right) = I(x+1) - I(x)$$

Sa dérivée seconde s'exprime par :

$$I''(x) = I'\left(x + \frac{1}{2}\right) - I'\left(x - \frac{1}{2}\right) = I(x+1) - 2I(x) + I(x-1)$$

Dans le cas d'un signal 2D on utilise le Laplacien défini par :

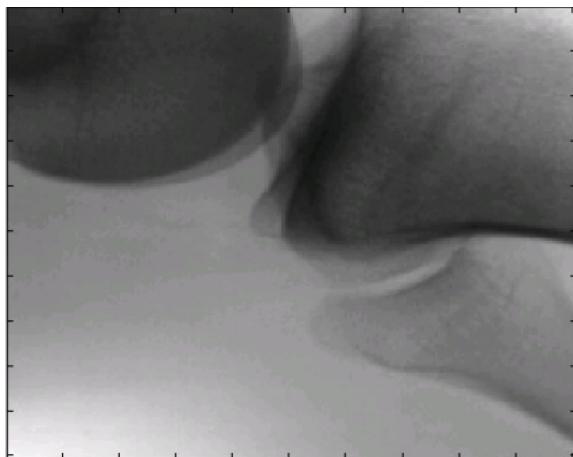
$$\Delta I(x, y) = \frac{\partial I^2(x,y)}{\partial x^2} + \frac{\partial I^2(x,y)}{\partial y^2} = I(x-1, y) + I(x+1, y) + I(x, y+1) - 4I(x, y)$$

Ce qui correspond à une convolution de l'image avec le masque :

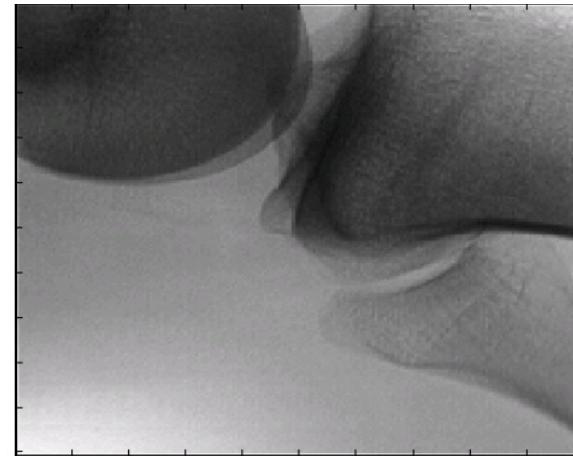
$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

# Renforcement de la netteté

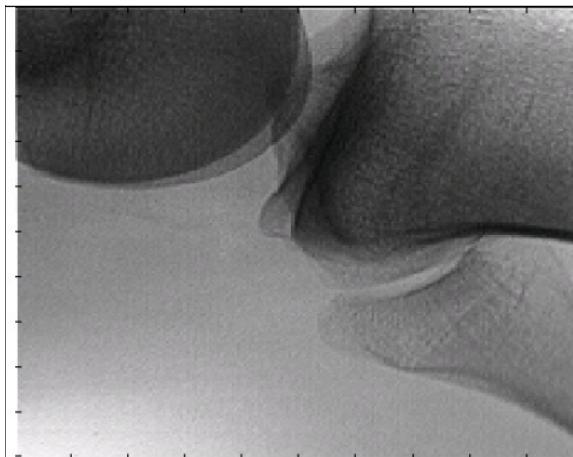
Image originale



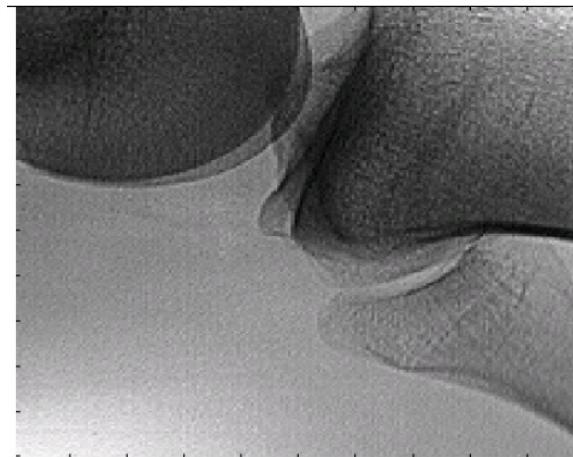
$K = 0.5$



$K = 1$



$K = 2$



## Renforcement de la netteté

---

Laplacien très sensible au bruit → effet de grain.

Utiliser des masques de plus grande dimension.

1	1	1	1	1
1	1	-3	1	1
1	-3	-8	-3	1
1	1	-3	1	1
1	1	1	1	1

Plus robuste au bruit + meilleure isotropie (même effet dans toutes les directions)

# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

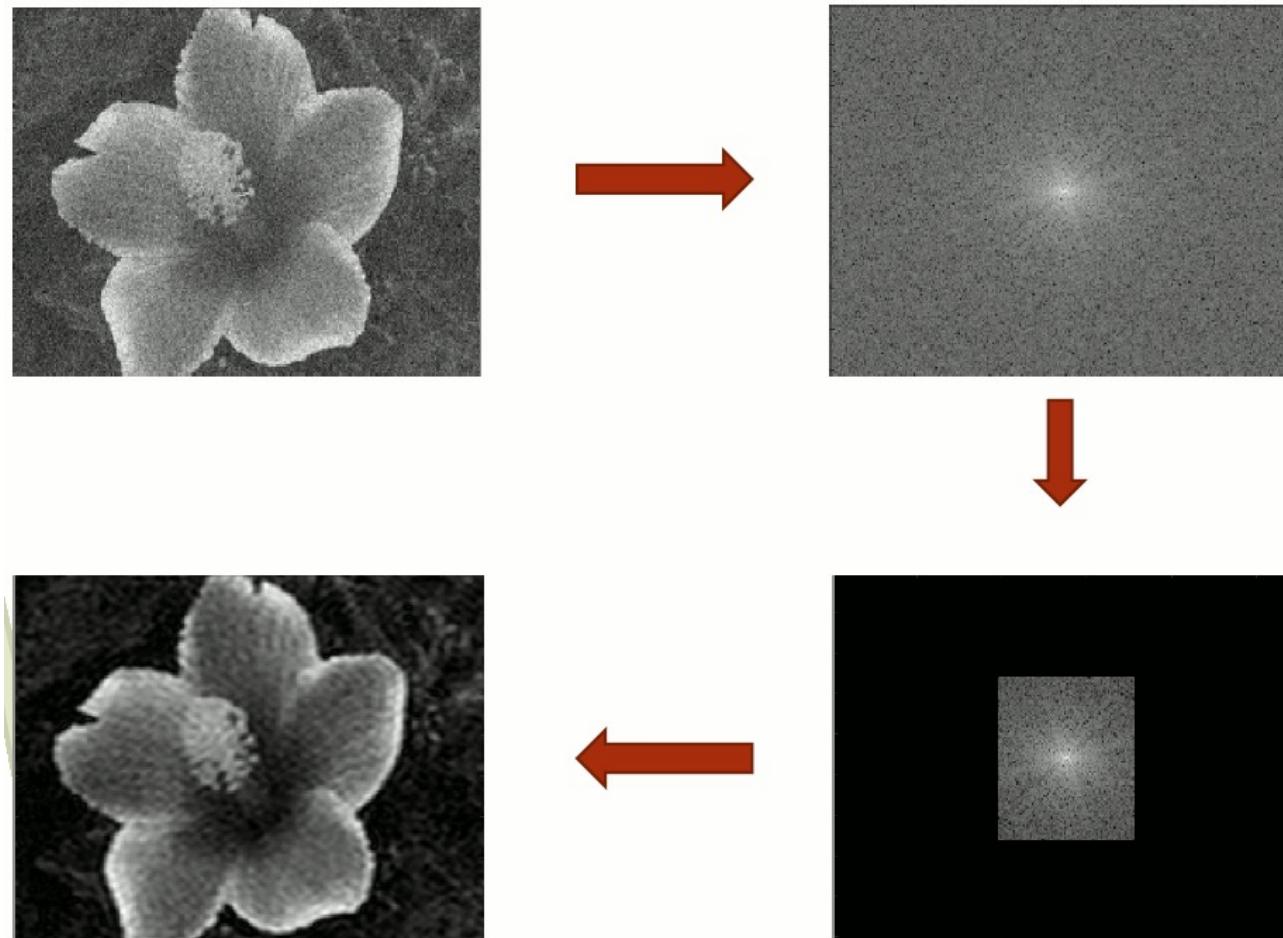
- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

# Filtrage dans le domaine fréquentiel

La taille de la fenêtre fréquentielle règle le degré de filtrage.



# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- **Filtre moyenneur**
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

## Filtrage linéaire - Moyenne

Réaliser une moyenne des niveaux de gris autour du pixel central.

Convolution de l'image avec un masque moyenieur :

$$IF(x,y) = I(x,y) * M(x,y)$$

On peut décider de lisser plus ou moins fort en choisissant une taille de masque plus ou moins grande.

$$M(x,y) = 1/9 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad M(x,y) = 1/25 \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

# Filtrage linéaire - Moyenne

## Remarque sur les filtres

S'il existe deux vecteurs  $F_1$  et  $F_2$  tels que :

$$F(x,y) = F_1(y)F_2(x) = F_1(y) * F_2(x) = F_2(x) * F_1(y)$$

On a alors :

$$\begin{aligned} I(x,y) * F(x,y) &= I(x,y) * (F_1(y) * F_2(x)) \\ &= (I(x,y) * F_1(y)) * F_2(x) \\ &= (I(x,y) * F_2(x)) * F_1(y) \end{aligned}$$

$$\begin{aligned} &\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} * (1 \quad 1 \quad 1) \\ &= (1 \quad 1 \quad 1) * \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \end{aligned}$$

Ceci permet de faire deux convolutions 1D plutôt qu'une convolution 2D.

Pour un filtre de taille  $N \times N$ , il y a  $2N$  multiplications plutôt que  $N^2$ .

Le filtre est alors dit séparable.

## Filtrage linéaire - Moyenne



Image originale



Image filtrée avec  
le masque 3x3



Image filtrée avec  
le masque 7x7

Méthode simple à mettre en œuvre mais effet de flou, contours dégradés.

# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- **Filtre gaussien**
- Filtre médian
- Filtre bilatéral
- Filtre de Nagao

# Filtrage linéaire – Filtre gaussien

Convoluter l'image avec un masque gaussien :

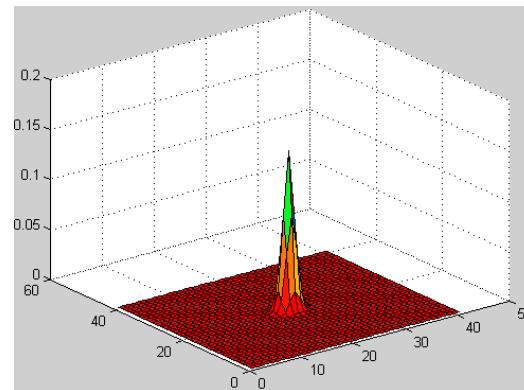
$$IF(x,y) = I(x,y) * G(x,y)$$

L'expression de la gaussienne en 2D est donnée par :

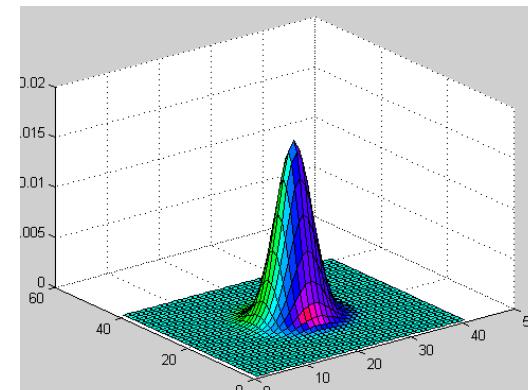
$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

L'intérêt du filtre gaussien est que l'on règle très facilement le degré de filtrage à travers le paramètre  $\sigma$ .

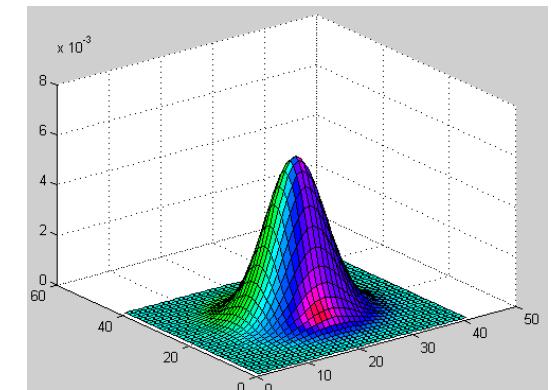
$\sigma = 1$



$\sigma = 2$



$\sigma = 3$



## Filtrage linéaire – Filtre gaussien

---

Tous ces masques ont été calculés sur un voisinage de taille 41x41 mais on voit sur les figures précédentes que la taille du masque peut être réduite ou augmentée en fonction de  $\sigma$ .

En théorie, la gaussienne a une étendue infinie, mais en pratique, on limite cette étendue  $[-3\sigma, 3\sigma]$  car la plupart de la puissance est dans cette zone (on trouve parfois  $[-4\sigma, 4\sigma]$ ).

Par rapport au filtre moyenneur, le filtre gaussien accorde une grande importance aux pixels proches du pixel central, et diminue cette importance au fur et à mesure que l'on s'éloigne de celui-ci.

Même défauts que filtre moyenneur, dégradation des contours, effet de flou.

Filtre linéaires : compromis filtrage/dégradation.

## Filtrage linéaire – Filtre gaussien

Les filtres gaussiens sont aussi séparables.

En continu on peut écrire :  $G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) = G(x) \cdot G(y)$

Où  $G(x)$  est la gaussienne 1D :  $G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$

Pour un masque 3x3 gaussien on aura :

$G(-1, -1)$	$G(0, -1)$	$G(1, -1)$
$G(-1, 0)$	$G(0, 0)$	$G(1, 0)$
$G(-1, 1)$	$G(0, 1)$	$G(1, 1)$

Même chose pour  $G(y)$ .

En discret on aura :  $G(x, y) = G(x)^T \cdot G(y) = G(x)^T * G(y) = G(y) * G(x)^T$

Plutôt que de réaliser une convolution 2D on aura intérêt à faire deux convolutions 1D

## Filtrage linéaire – Filtre gaussien

Remarque : pour une même taille de masque, plus  $\sigma$  augmente, plus le flou augmente.



Image originale



$\sigma = 1$



$\sigma = 2$



$\sigma = 3$



$\sigma = 4$



$\sigma = 5$

# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- **Filtre médian**
- Filtre bilatéral
- Filtre de Nagao

## Filtrage d'ordre – Filtre médian

Ce filtre étant non linéaire, il ne peut pas être réalisé avec une convolution 2D de l'image. Considérons un voisinage rectangulaire autour du pixel d'intérêt. Le filtre médian consiste à prendre la valeur des niveaux de gris séparant la population en deux effectifs égaux.

Exemple :


A 3x3 kernel, the central pixel is shaded gray. The values are:

10	10	7
9	12	10
12	10	7

7-7-9-10-10-10-10-12-12

La valeur médiane du masque est 10, le pixel central aura comme valeur finale 10.

Ce filtre donne de très bon résultats de part son principe sur le bruit impulsif (poivre et sel).

## Filtrage d'ordre – Filtre médian



Image originale



Filtre moyenneur 5x5

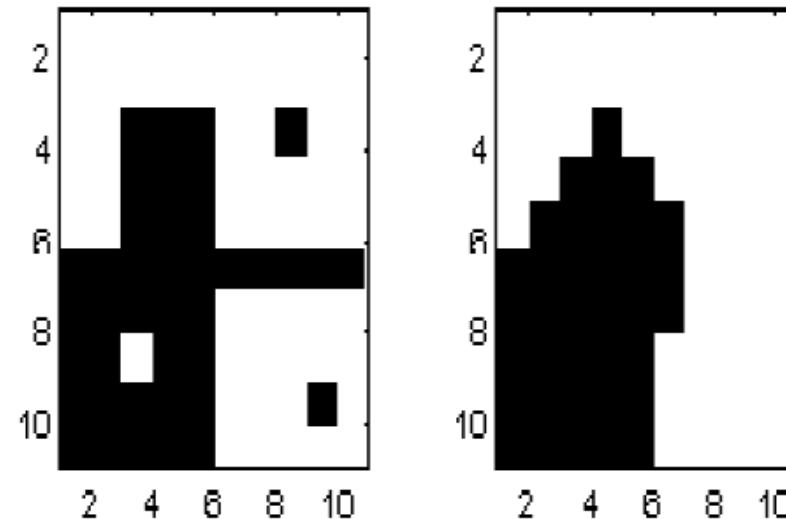


Filtre médian 5x5

# Filtrage d'ordre – Filtre médian

## Problèmes de la médiane :

- Supprime les détails fins.
- Enlève les points isolés.
- Couteuse en temps de calcul.



→ On va rechercher un opérateur qui remédie à ces inconvénients sans détériorer les contours.

## Filtrage d'ordre – Variantes paramétriques

---

But : Eviter l'élimination des détails, diminution temps de calculs, ne pas introduire de flou au niveau des contours.

### Filtre à moyenne seuillée :

Dans le masque autour du pixel p, calcul de la variance des niveaux de gris.

Si variance < seuil (critère de zone homogène) alors p = moyenne du masque, sinon on garde le niveau de gris d'origine.

### Filtre à moyenne pondérée :

Ce filtre est moins binaire que le précédent mais nécessite une estimation de la variance du bruit VB.

VC = variance courante dans le masque et M = moyenne du masque

$$p = (1 - K).M + K.p \text{ avec } K = (VC - VB)/VC$$

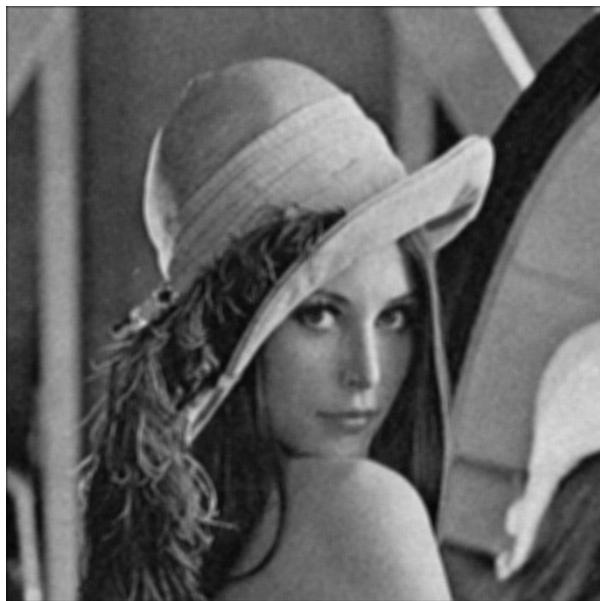
# Filtrage d'ordre – Variantes paramétriques

De gauche à droite :

Filtre moyenneur 5x5

Filtre moyenneur seuillé 5x5

Filtre moyenneur pondéré 5x5



# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- **Filtre bilatéral**
- Filtre de Nagao

## Filtre bilatéral

L'objectif est de préserver les discontinuités et lisser les zones homogènes.

Dans un filtre gaussien, seule la distance géométrique au point central du masque est considérée pour pondérer les valeurs.

Dans un filtre bilatéral, on prend aussi en compte la différence des intensités lumineuses au point central du masque.

$$IF(s) = \frac{1}{K(s)} \sum_{p \in S_m} f(|p - s|) \cdot g(I_p - I_s) \cdot I_p$$

Avec  $K(s) = \sum_{p \in S_m} f(|p - s|) \cdot g(I_p - I_s)$

$f$  une fonction symétrique décroissante, pondération spatiale

$g$  une fonction symétrique décroissante, pondération photométrique

$S_m$  une fenêtre de taille  $m$  autour du point  $s$  de l'image  $I$

$p$  un point de la fenêtre  $S_m$

## Filtre bilatéral

Original



Gaussien 7x7



Médian 3x3



Médian 7x7



Bilatéral 7x7



# Amélioration et restauration d'images

## Diminution du bruit dans les images

---

Restauration

Filtrage

Amélioration d'images

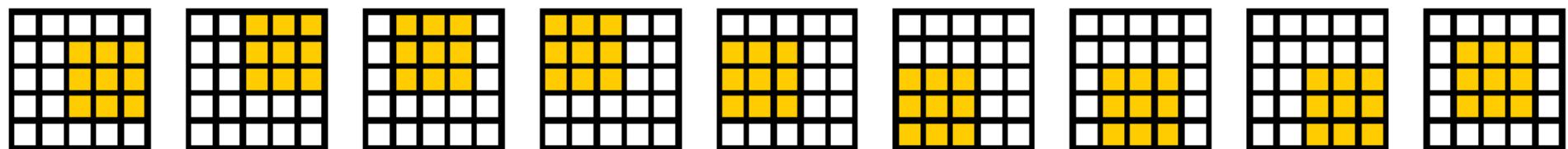
- Extension de dynamique
- Correction d'exposition
- Egalisation d'histogramme
- Renforcement de la netteté

Réduction du bruit

- Dans le domaine fréquentiel
- Filtre moyenneur
- Filtre gaussien
- Filtre médian
- Filtre bilatéral
- **Filtre de Nagao**

## Filtre de Nagao

Ce filtre est également appelé filtre à sélection de voisinage. Chaque masque 5x5 est divisée en 9 domaines.



Pour chaque domaine, on calcule sa moyenne et son écart type.

On cherche le domaine où la variance est la plus faible (zone avec le moins de contour) et on affecte la moyenne des niveaux de gris du domaine trouvé au pixel central du masque.

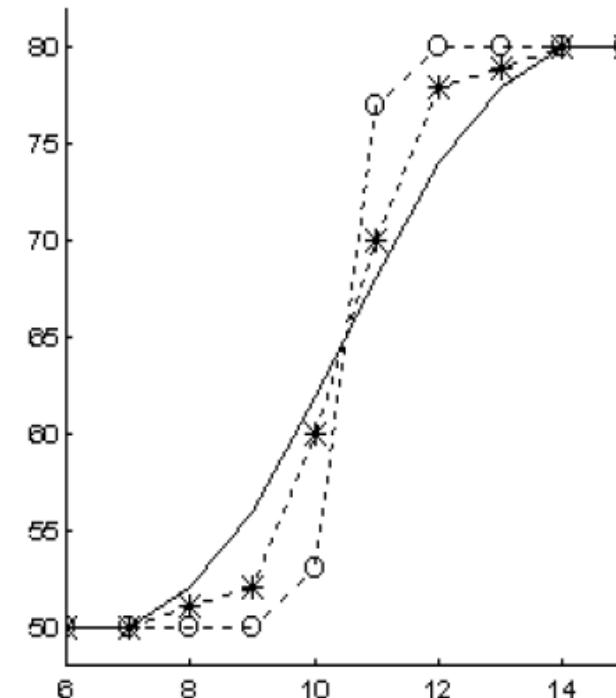
- Elimination des pixels hors norme (bruit impulsif)
- Préservation et amélioration du contraste
- Zones avant et après la transition sont plus homogènes

# Filtre de Nagao

## Etude comparative

Sur une zone de transition, nous affichons :

- Moyenne 5x5
- \* Médiane 5x5
- o Nagao

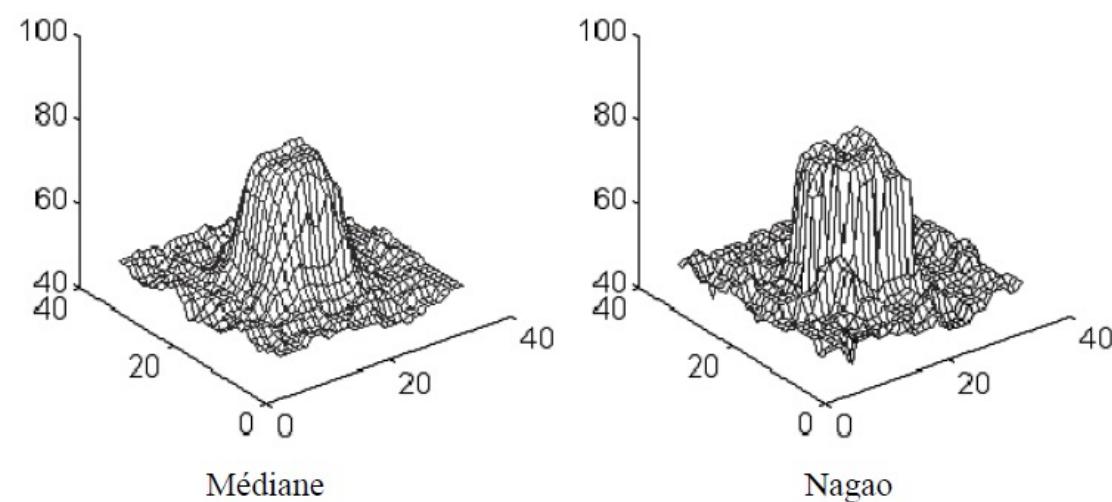
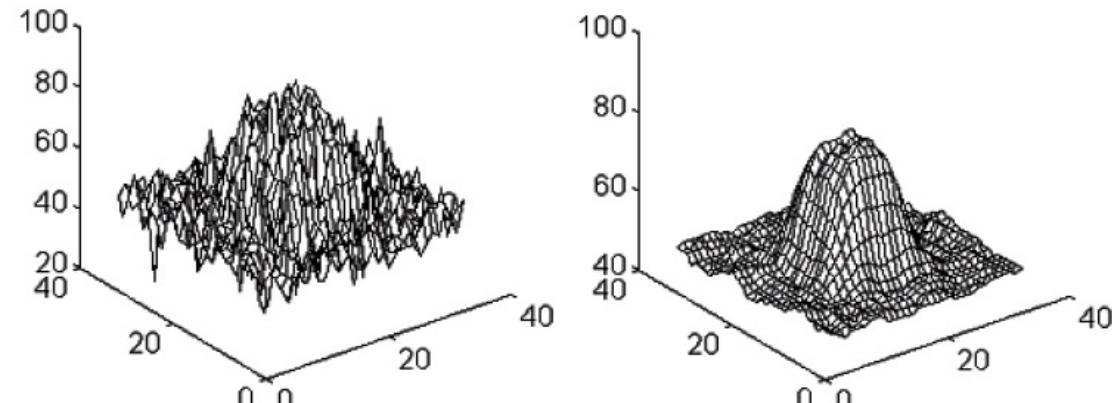


Du point de vue raideur de la transition, c'est le filtre de Nagao qui donne les meilleurs résultats.

# Filtre de Nagao

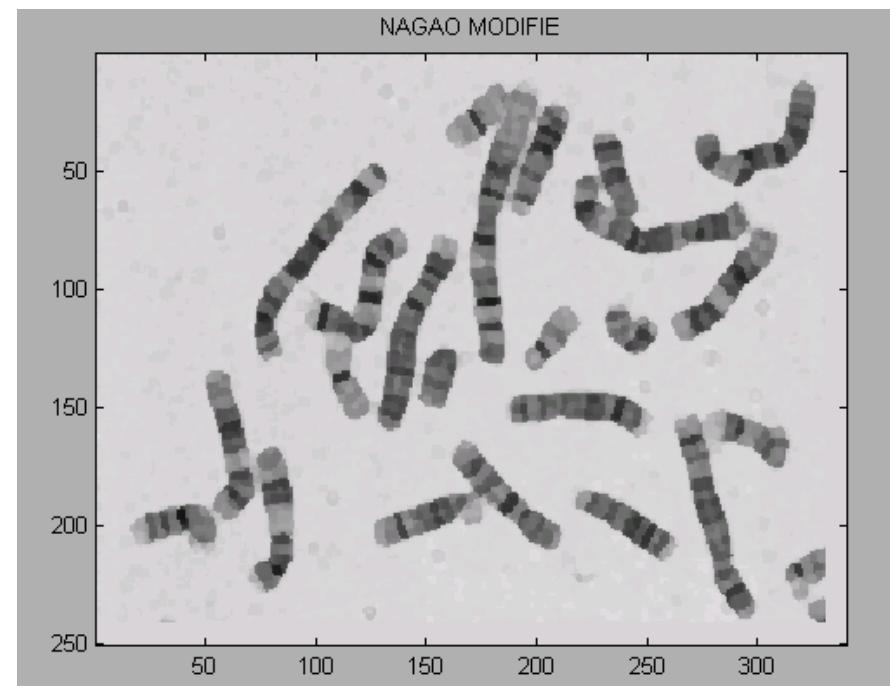
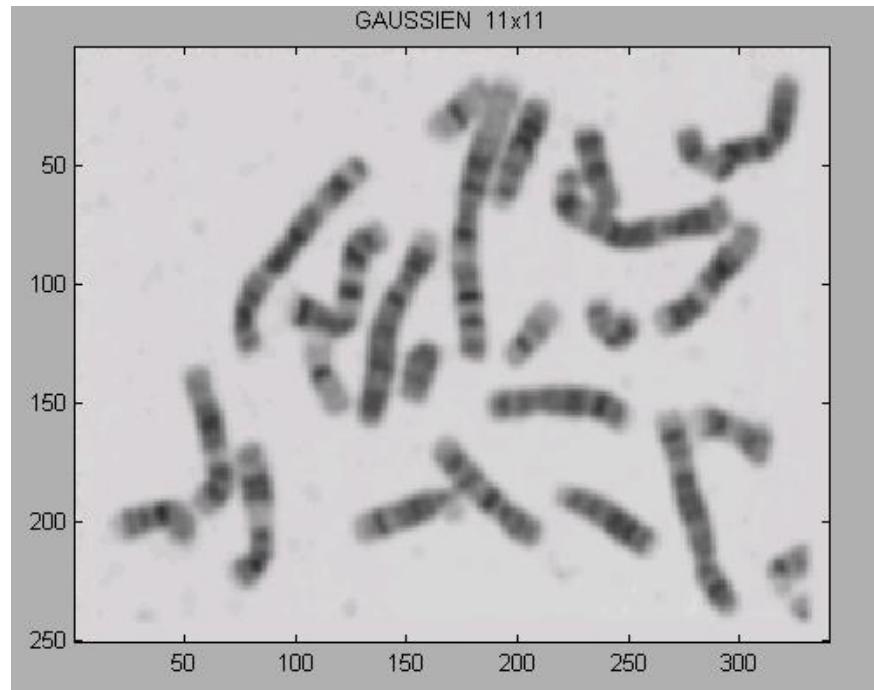
## Etude comparative

Résultats sous forme  
surfacique

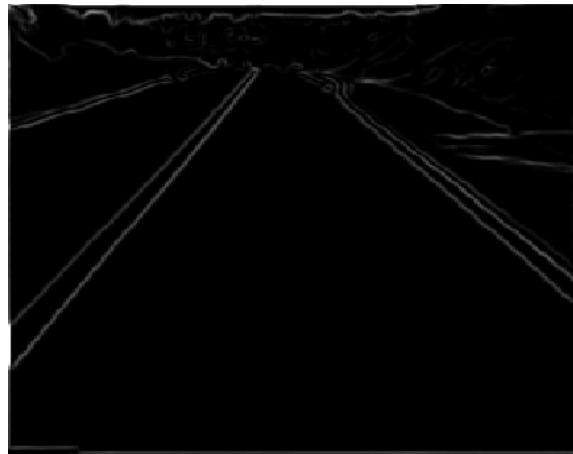


# Filtre de Nagao

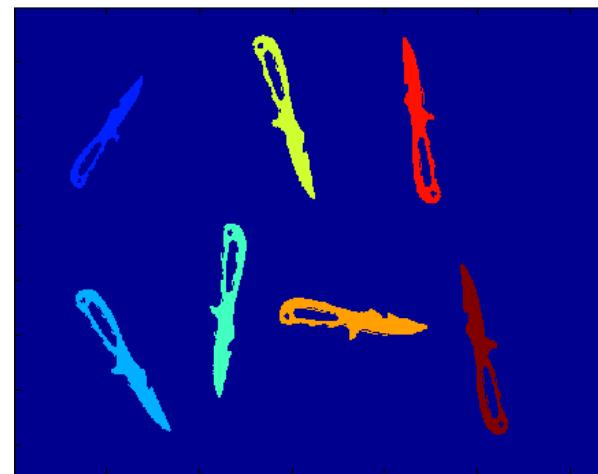
## Etude comparative



# Segmentation



Contours



Régions



Points

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

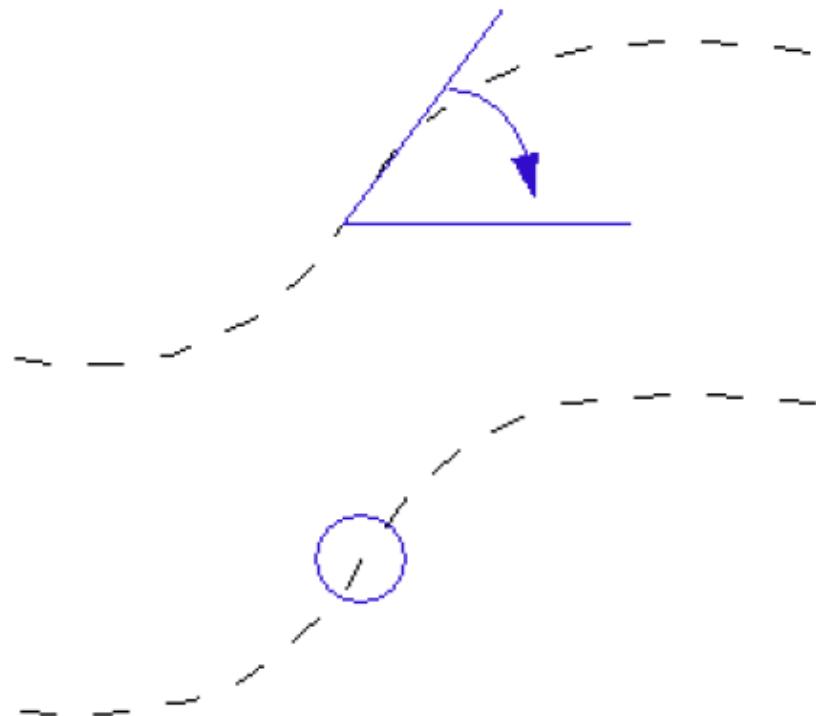
- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

- Par agrégation
- Par division
- Etiquetage

# Contours

Contours = forte discontinuités



Pente élevée de la transition  
→ maxima locaux du gradient

Changement de concavité  
→ passages par zéros de la  
dérivée seconde (laplacien)

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

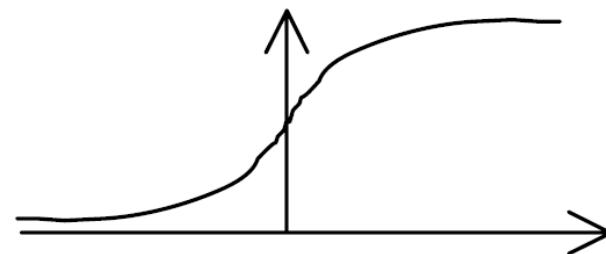
## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

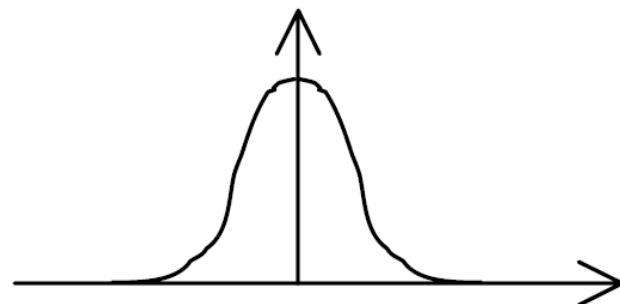
## Région

- Par agrégation
- Par division
- Etiquetage

# Approche gradient



Signal



Dérivée

→ Recherche des maxima locaux du gradient

## Approche gradient

---

Dans le cas 2D des images, le vecteur gradient est défini au point de coordonnées (x,y) par :

$$\nabla I(x, y) = \begin{pmatrix} \frac{\partial I(x, y)}{\partial x} \\ \frac{\partial I(x, y)}{\partial y} \end{pmatrix} = \begin{pmatrix} I_x(x, y) \\ I_y(x, y) \end{pmatrix}$$

Le module du gradient (amplitude du saut d'intensité) est défini par :

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

L'orientation du gradient (orthogonale à celle du contour) est définie par :

$$\Phi(x, y) = \arctan \left( \frac{I_y(x, y)}{I_x(x, y)} \right)$$

# Approche gradient



Image originale

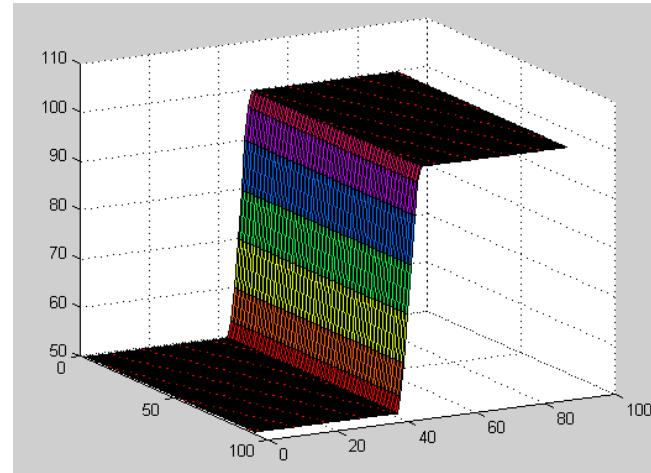


Image originale sous forme surfacique

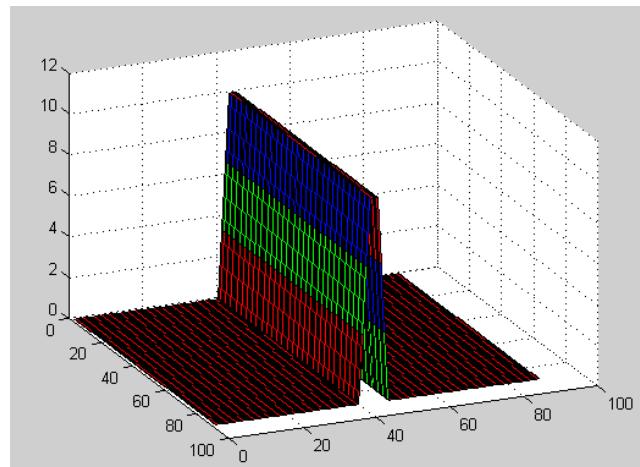


Image du module du gradient

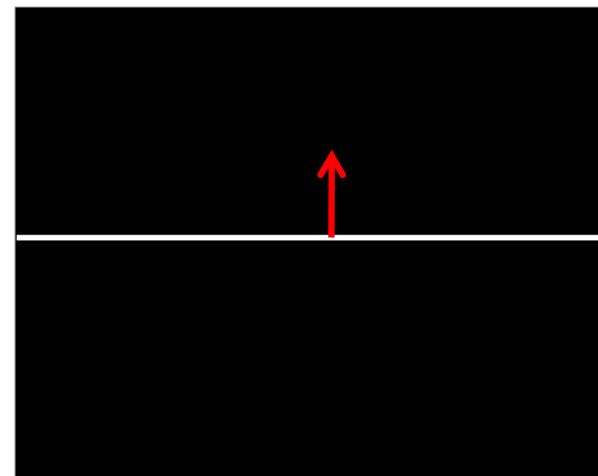


Image du gradient seuillé et orientation

# Approche gradient

## Calcul direct des dérivées

Il est réalisé de la façon la plus simple qu'il soit :

$$Ix(y,x) = I(y,x+1) - I(y,x) = (I^*Gx)(y,x)$$

$$Iy(y,x) = I(y+1,x) - I(y,x) = (I^*Gy)(y,x)$$

Ce qui correspond à une convolution avec  $Gx = \begin{bmatrix} -1 & 1 \end{bmatrix}$  et  $Gy = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

## Problème du point central

$$\rightarrow Gx = [-1 \quad 0 \quad 1] \text{ et } Gy = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Les dérivées sont très sensibles au bruit → filtrage préalable.

# Approche gradient

---

## Masque de Prewitt

Les masques dérivateurs sont maintenant :

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} * [-1 \quad 0 \quad 1]$$

Lissage des lignes  
et dérivée des colonnes

$$Gy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * [1 \quad 1 \quad 1]$$

Dérivée des lignes  
et lissage des colonnes

On combine à la fois un filtrage et une dérivée.

→ moins sensible au bruit que le calcul direct des dérivées.

# Approche gradient

## Masque de Sobel

Même principe mais avec un autre filtrage :

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * [-1 \quad 0 \quad 1]$$

$$Gy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * [1 \quad 2 \quad 1]$$

# Approche gradient

## Comparaison

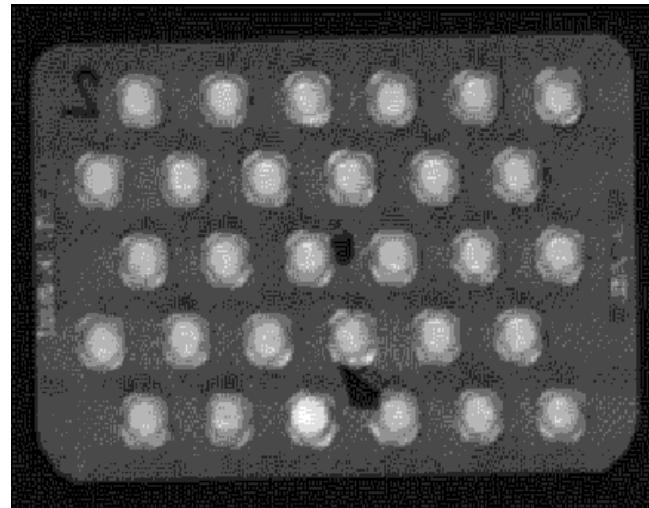
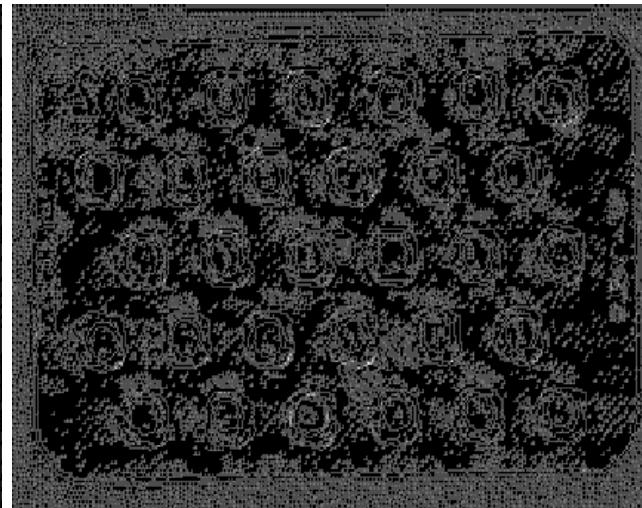
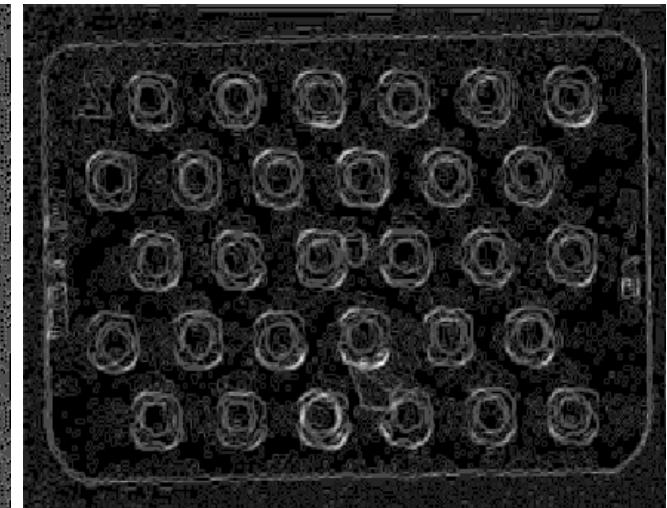


Image d'origine



Gradient simple



Gradient de Sobel

# Approche gradient

---

## Dérivée de Gaussienne

Le filtrage précédent la dérivation peut être réalisé grâce à un masque Gaussien  $G$ . Dans ce cas, le calcul de la dérivée en  $x$  se fait par :

$$\begin{aligned}
Ix(x, y) &= \frac{\partial(I(x, y) * G(x, y))}{\partial x} \\
&= I(x, y) * \frac{\partial G(x, y)}{\partial x} \\
&= I(x, y) * G(x, y) * dx(x, y) \\
&= I(x, y) * Gx(x, y)
\end{aligned}$$

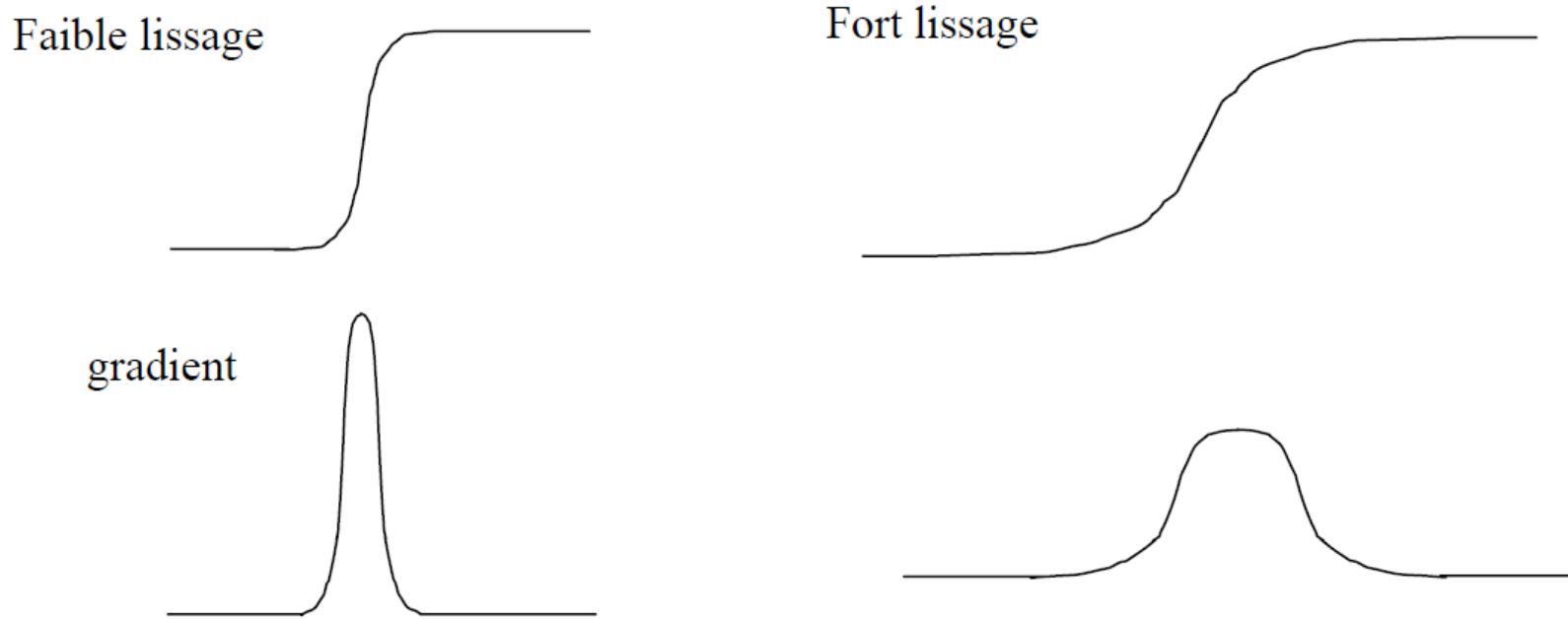
Plutôt que de calculer deux produits de convolution (couteux en temps de calcul), on peut calculer la dérivée de manière analytique :

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \text{ soit } \frac{\partial G(x, y)}{\partial x} = -\frac{x}{2\pi\sigma^4} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Le paramètre  $\sigma$  règle le degré de lissage.

# Approche gradient

## Dualité filtrage / raideur de la transition



Faible lissage = pic de haute amplitude et étroit

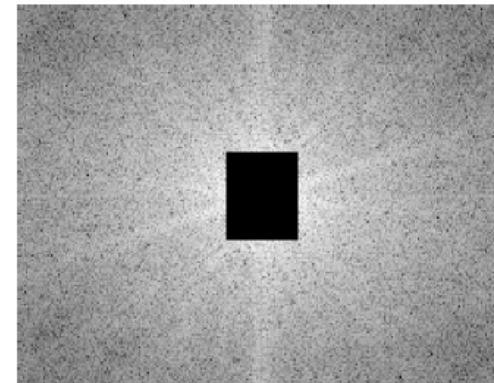
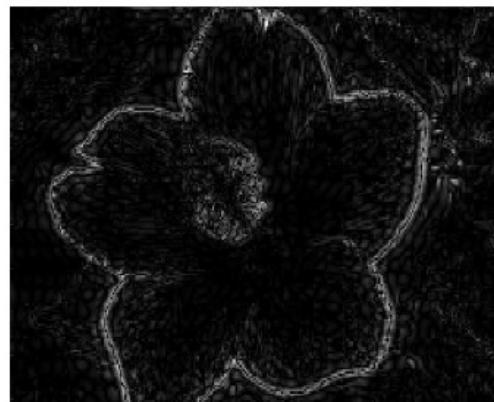
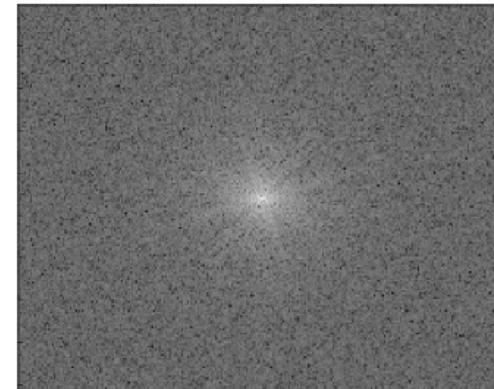
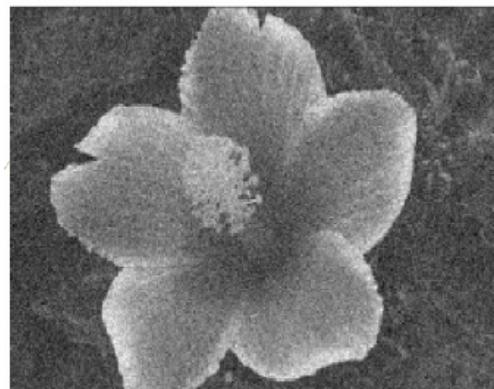
→ contours fins, beaucoup de bruit dans la dérivée

Fort lissage = pic de plus faible amplitude et plus large

→ contour épais, gradient moins perturbé par le bruit

# Approche gradient

## Filtrage passe-haut



# Approche gradient

---

## Seuillage

Une fois la norme du gradient calculée en chaque point de l'image, il faut seuiller cette norme pour décider si un pixel fait partie ou non d'un contour.

Tous les pixels possédant une norme supérieure à un seuil  $S$  sont déclarés appartenir à un contour.

Tout le problème réside alors dans le choix du seuil :

Seuillage trop bas = sur-détection, beaucoup de bruit et contours épais.

Seuillage trop bas = sous-détection, contours non fermés.

# Approche gradient

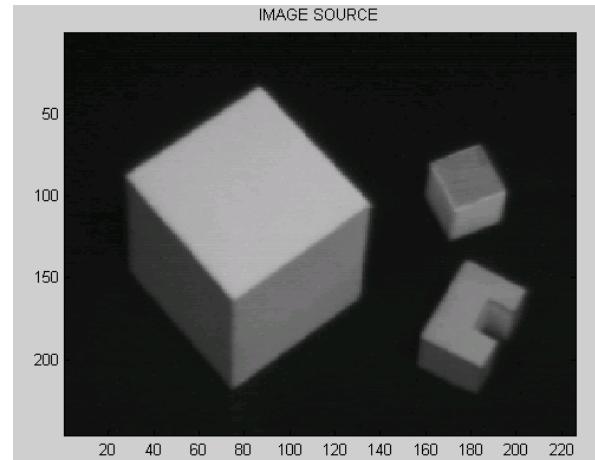
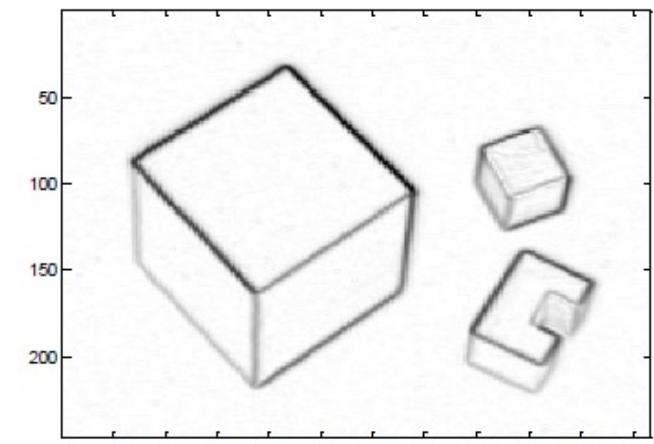
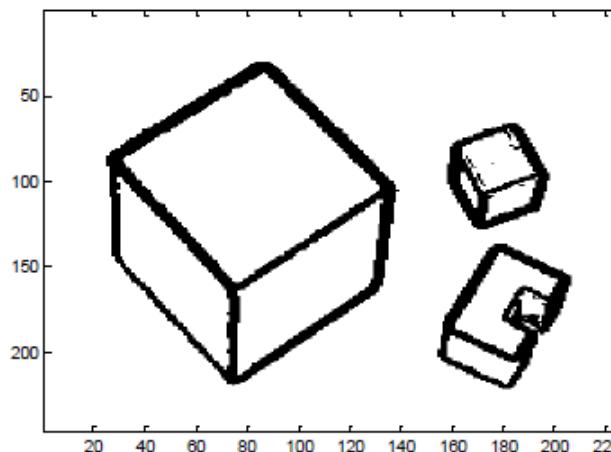


Image d'origine

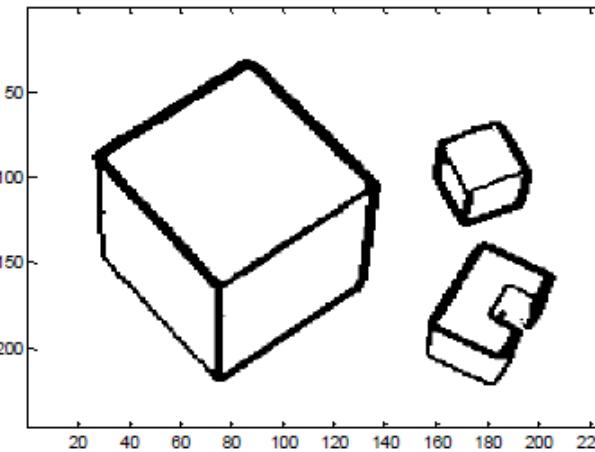


Norme du gradient

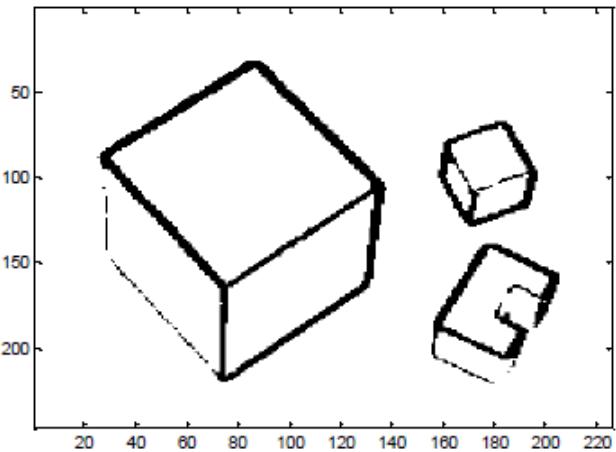
$S = 10$



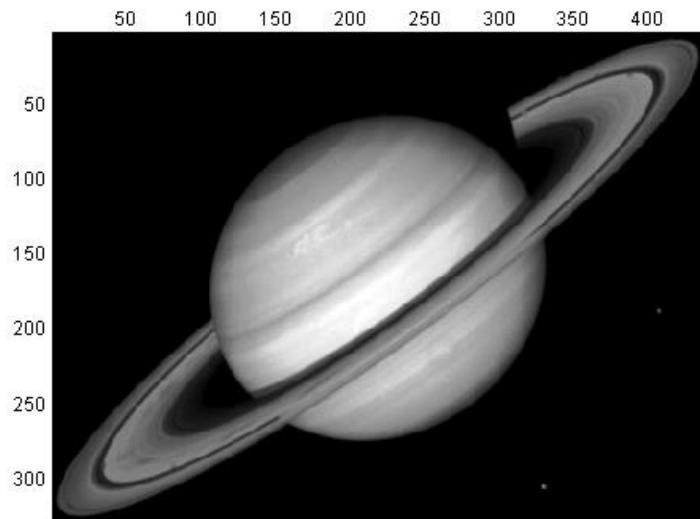
$S = 15$



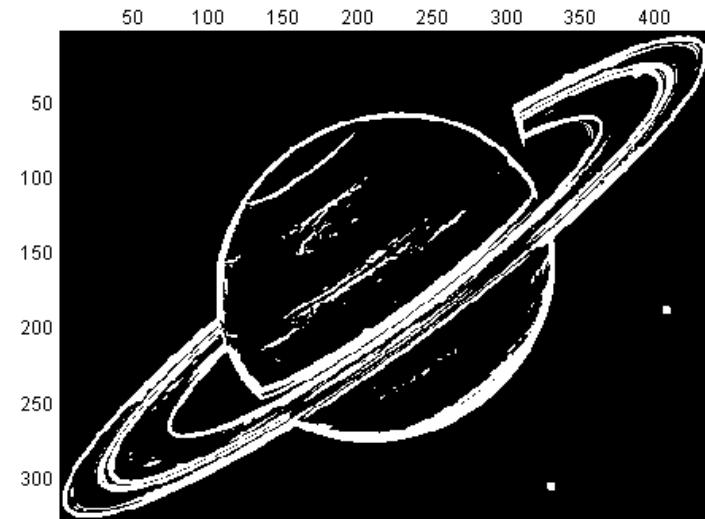
$S = 20$



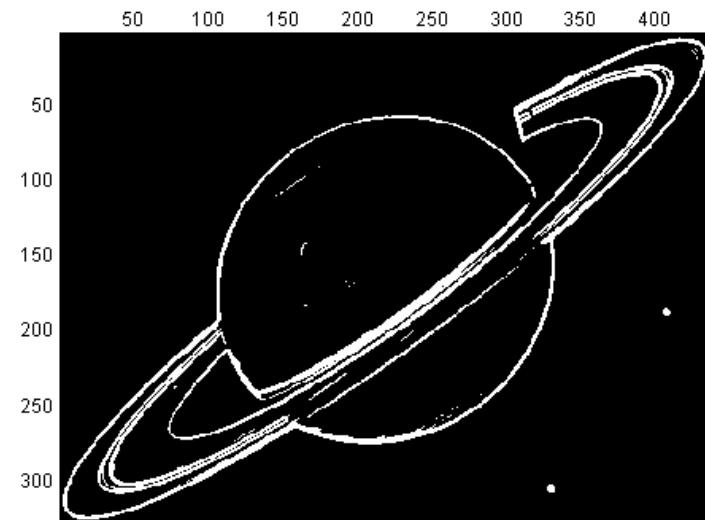
# Approche gradient



S = 15  
Contours épais



S = 25  
Lacunes  
Contours épais



## Approche gradient

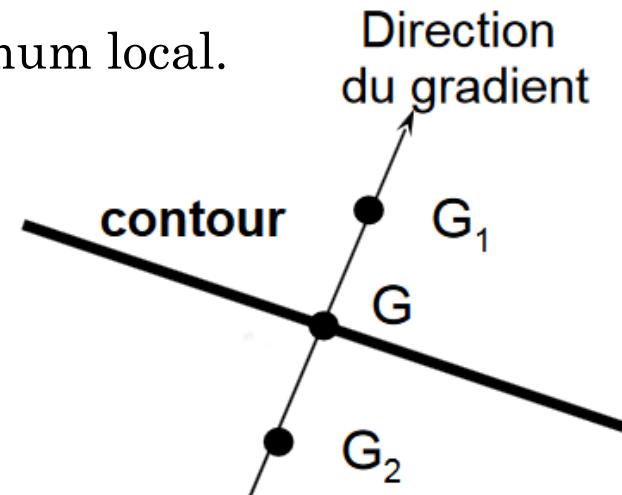
Pour obtenir des contours plus fins on peut rechercher les lignes de crêtes sur l'image du gradient.

On cherche alors les maxima locaux dans la direction du gradient.

$G_1$  et  $G_2$  deux voisins pris dans la direction du gradient.

Si  $|G| > |G_1|$  et  $|G| > |G_2|$ , alors  $G$  est un maximum local.

→ Contours non fermés, contours parasites.



# Approche gradient

---

## Seuillage par hystérésis

Deux seuils, un seuil haut et un seuil bas.

Si norme gradient > seuil haut → contour sûr (1)

Si norme gradient < seuil bas → pas de contour (0)

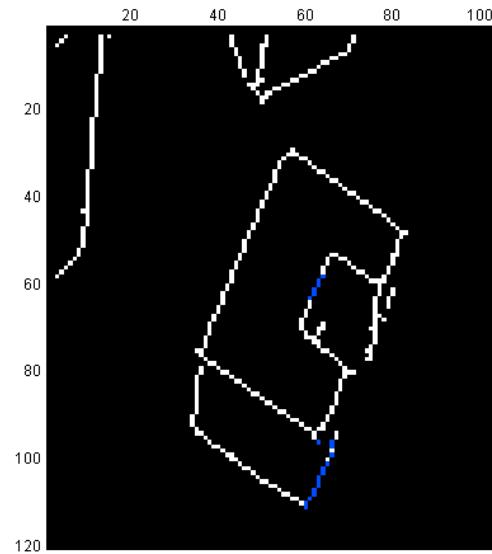
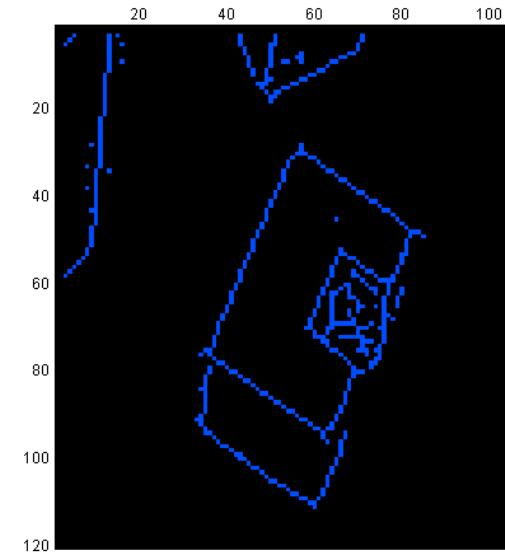
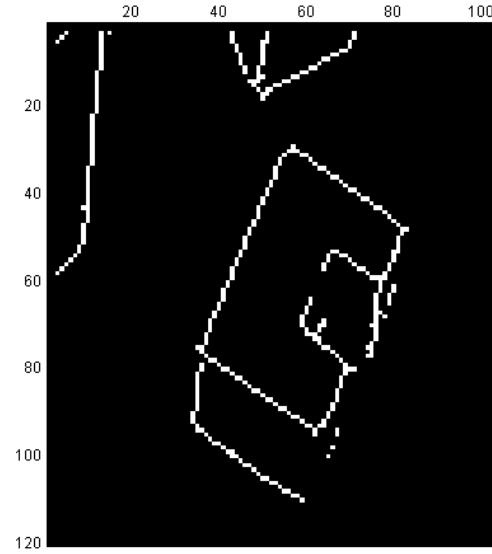
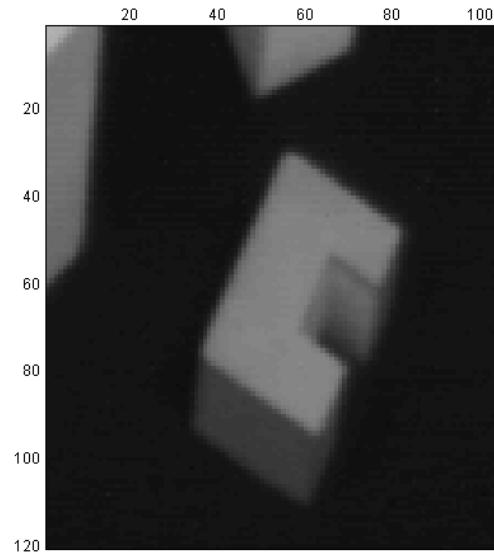
Si seuil bas < norme gradient < seuil haut → contour de fermeture (2)

Si contour 2 adjacent à contour 1 → contour 1 sinon contour 0

La recherche d'adjacence s'effectue à partir des points d'extrémités des contours sûrs (test de 8 connexité).

Ligne de crête + seuillage par hystérésis = détecteur de Canny.

# Approche gradient



# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

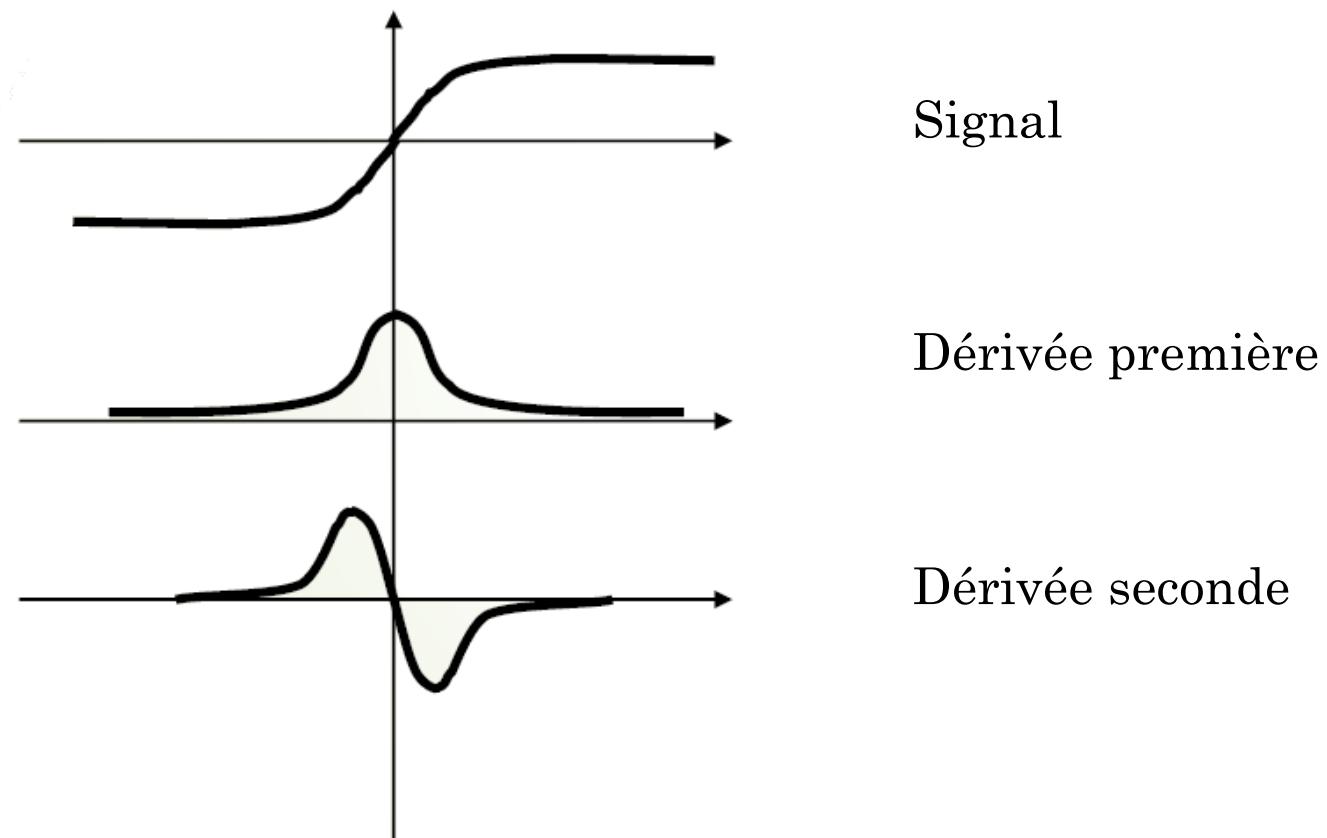
- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

- Par agrégation
- Par division
- Etiquetage

## Approche laplacien

Un contour peut aussi être détecté comme un passage par zéro de la dérivée seconde.



## Approche laplacien

---

Les dérivées secondes forment la matrice Hessienne qui est estimée en chaque pixel :

$$H = \begin{pmatrix} \frac{\partial^2 I(x, y)}{\partial x^2} & \frac{\partial^2 I(x, y)}{\partial x \partial y} \\ \frac{\partial^2 I(x, y)}{\partial x \partial y} & \frac{\partial^2 I(x, y)}{\partial y^2} \end{pmatrix}$$

La trace de la matrice Hessienne, qui est égale à la somme des valeurs propres, forme le Laplacien :

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

## Approche laplacien

Le Laplacien peut être estimé grâce à une convolution avec le masque (voir section renforcement de contraste) :

0	1	0
1	-4	1
0	1	0

Il peut aussi être estimé de manière robuste grâce aux dérivées de gaussiennes :

$$\Delta(I(x,y) * G(x,y)) = I(x,y) * \Delta G(x,y)$$

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

Et  $\Delta G(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} = \frac{1}{\pi\sigma^4} \frac{2\sigma^2 - x^2 - y^2}{2\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$

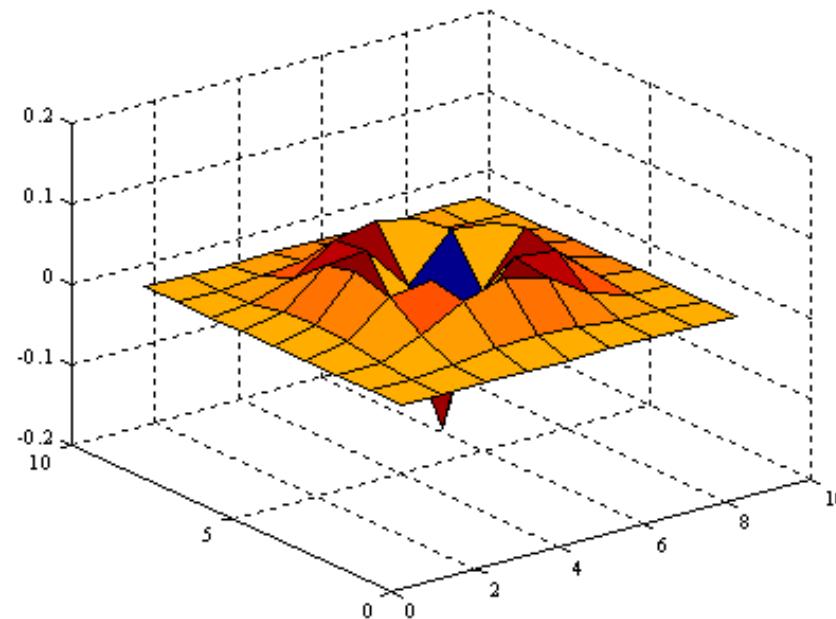
# Approche laplacien

## Dérivée de Gaussienne

La variance  $\sigma^2$  règle le degré de lissage.

La dimension du masque dépend de la variance.

Exemple :  $\sigma = 1$ , taille du masque :  $-4\sigma$  à  $4\sigma$ .



# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

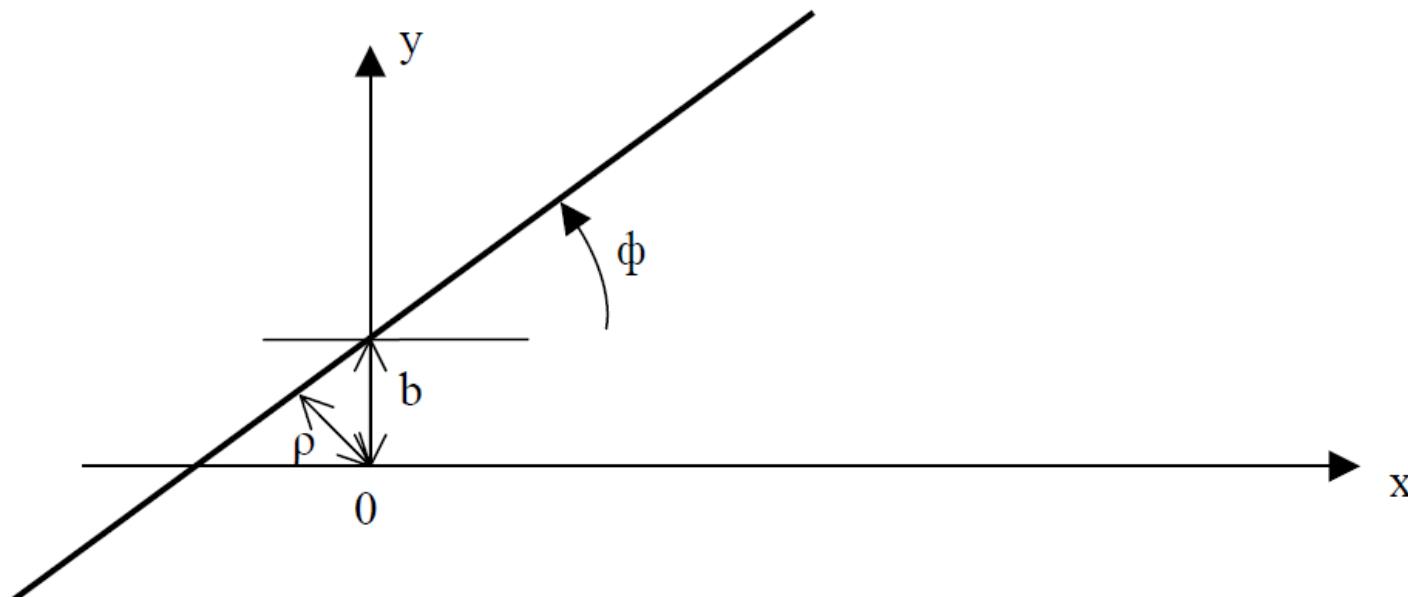
- Par agrégation
- Par division
- Etiquetage

## Transformée de Hough

Cette méthode a pour but de détecter des droites lorsque l'on a les points de contour.

Considérons une droite dans le plan  $(x,y)$ . Elle a pour équation :

$$y = a x + b \quad \text{ou} \quad x \sin \phi + y \cos \phi = \rho \quad (\text{équation 1})$$

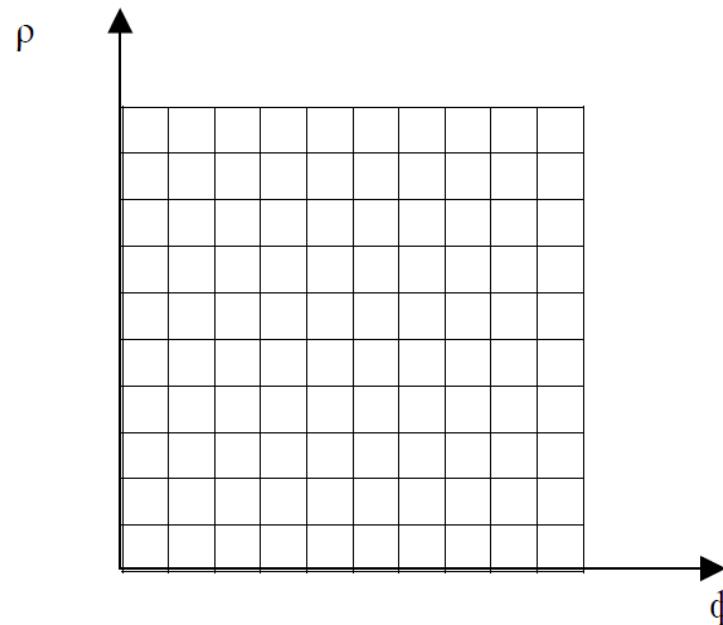


## Transformée de Hough

Si  $\phi \in [0, \pi]$ , les paramètres de la droite  $(\phi, \rho)$  sont uniques.

La transformée de Hough est une transformation qui permet de passer du plan image  $(x, y)$  au plan des paramètres.

Le plan image est parfaitement défini, le plan des paramètres devra être discréteisé en  $N \times M$  cellules :



## Transformée de Hough

---

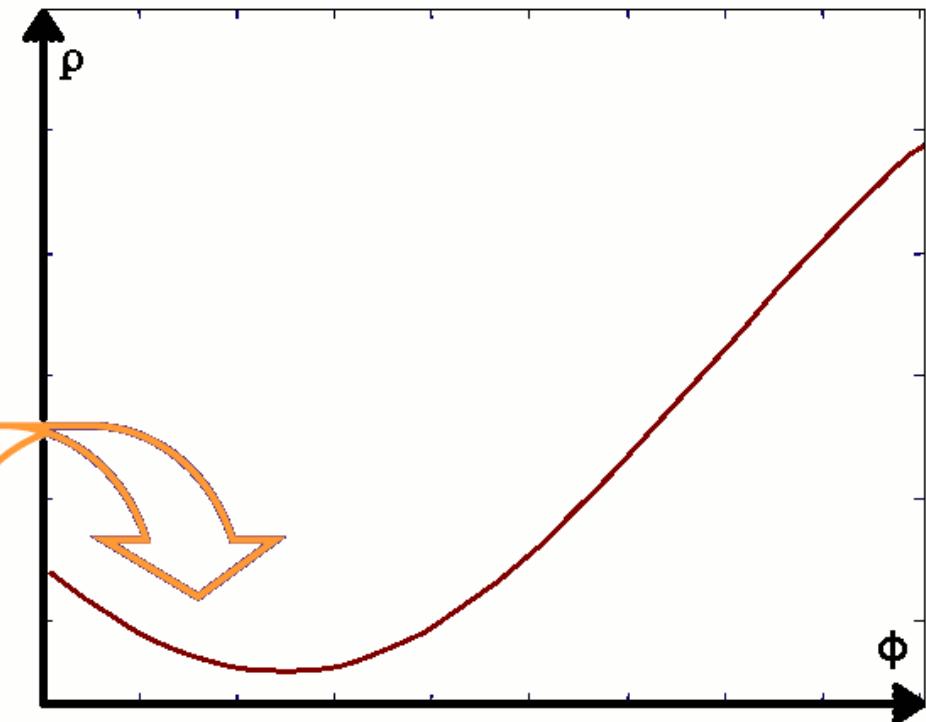
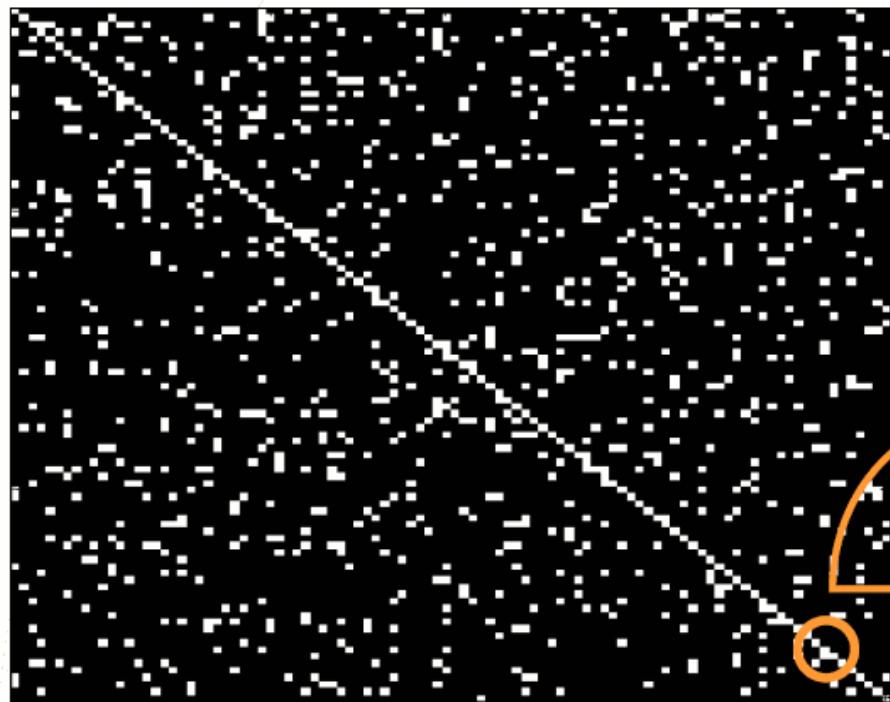
Considérons un point contour  $(x_i, y_i)$ .

Pour chaque orientation discrète  $\phi$ , on détermine  $\rho$  avec l'équation 1. On incrémentera alors de 1 la case  $(\phi, \rho)$  dans l'espace des paramètres.

Chaque point de contour vote pour  $M$  points (correspondants aux  $M$  orientations discrètes) dans l'espace des paramètres.

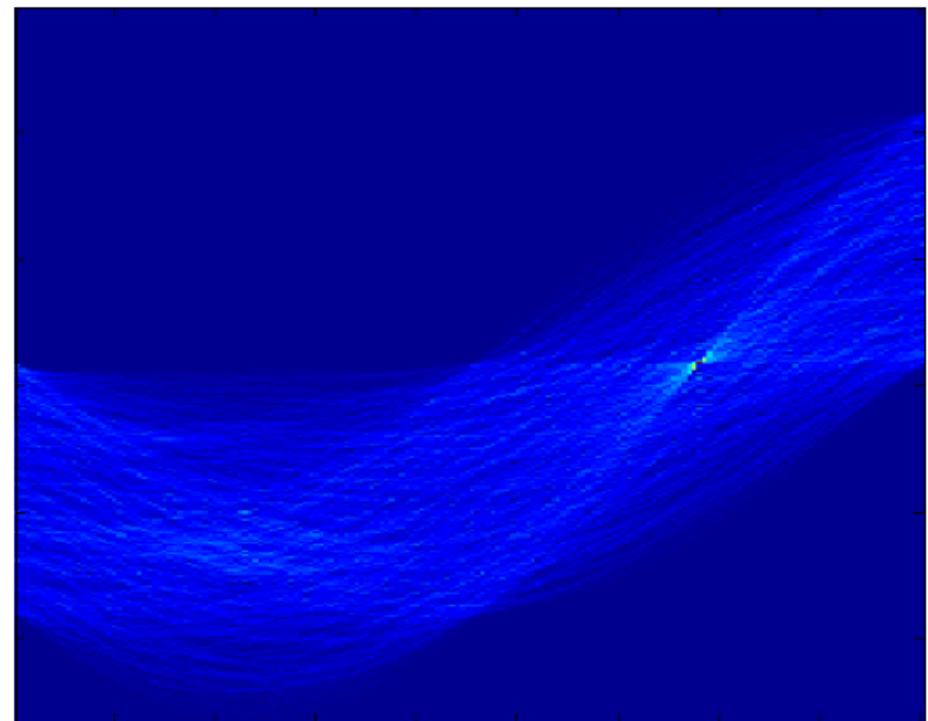
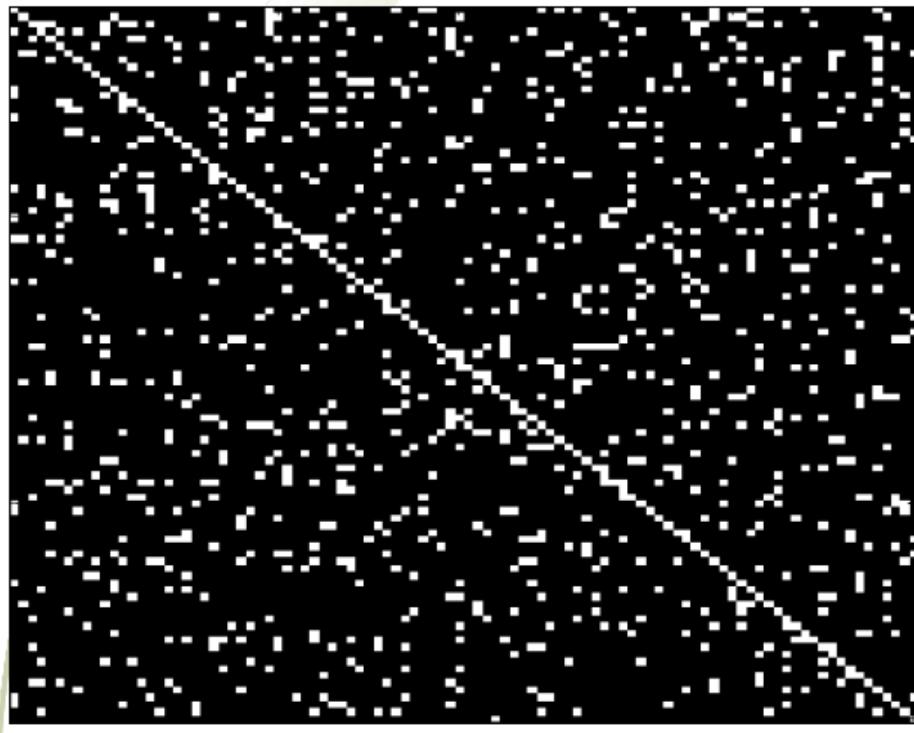
Si une droite est présente dans l'image, chaque point de contour de cette droite va voter pour la même case de l'espace des paramètres provoquant ainsi une accumulation. En la détectant, on a l'équation de la droite.

# Transformée de Hough



# Transformée de Hough

---



# Transformée de Hough

---

## Transformée de Hough généralisée

Une amélioration peut être amenée si on connaît en plus l'orientation des points de contour (facile à avoir : c'est l'orientation du gradient).

Pour chaque point contour on calcule  $\rho$  uniquement pour l'orientation perpendiculaire à celle du gradient. L'accumulation dans l'espace des paramètres est alors beaucoup plus nette.

La transformée de Hough peut être étendue à des courbes de formes quelconque.

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

- Par agrégation
- Par division
- Etiquetage

## Points d'intérêt

---



# Points d'intérêt

## Recherche de coins dans les images

### Pourquoi ?

- Faire de la mosaïque
- Faire de la calibration de caméra
- Faire de l'indexation
- Faire de l'appariement

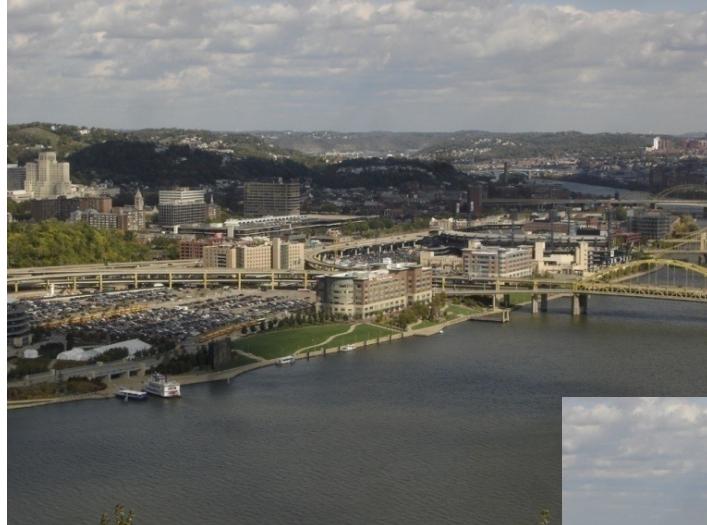


### Définition

Forte discontinuité du signal dans deux directions

# Points d'intérêt

## Détecteur de points d'intérêt



# Points d'intérêt

## Détecteur de points d'intérêt



# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

- Par agrégation
- Par division
- Etiquetage

# A partir des contours

Extraction de contours puis au choix :

- Courbure en chaque point de contour et extraire ceux  $>$  seuil
- Approximation polygonale des contours, intersections de segments = PI
- Recherche rupture dans la chaîne des orientations du gradient du contour

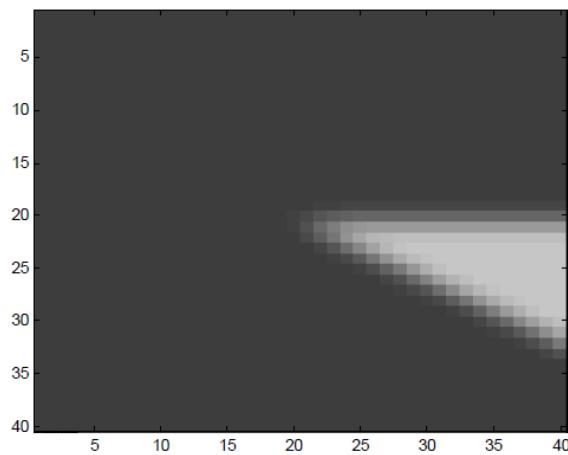
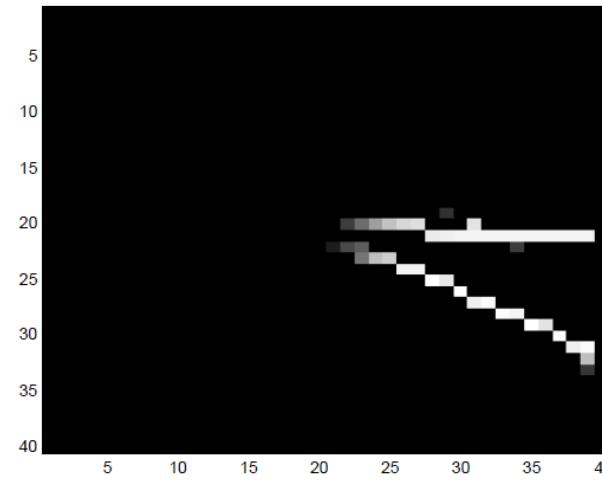


Image de synthèse d'un coin



Module du gradient

Pb : gradient mal défini près des coins.

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- **Détecteur de Moravec**
- Détecteur de Harris
- Mettre en correspondance les points

## Région

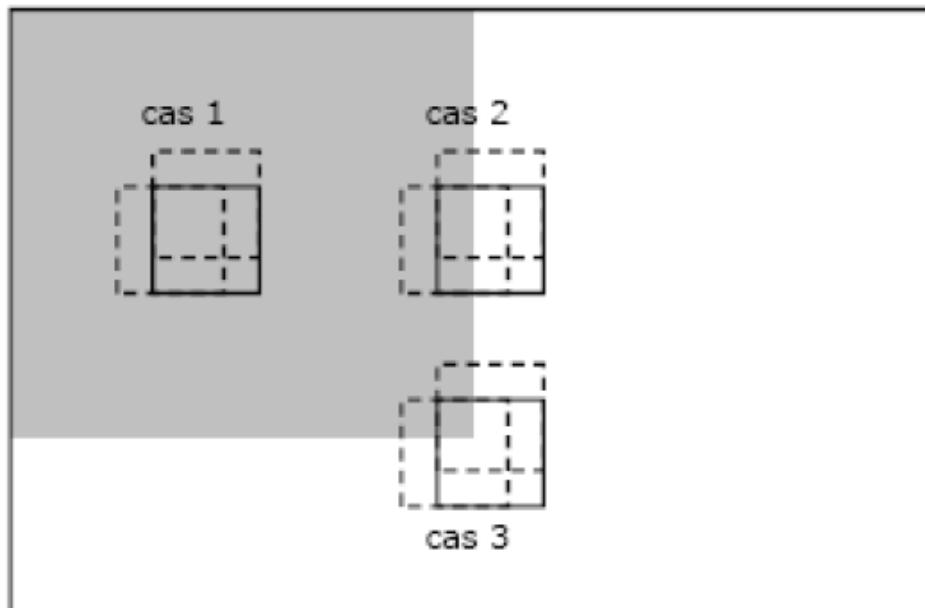
- Par agrégation
- Par division
- Etiquetage

## Détecteur de Moravec

Si l'on est proche d'un coin, il doit y avoir une forte dissemblance entre le signal et le signal décalé de  $(dx, dy)$  dans le voisinage d'un pixel.

Moravec propose ainsi d'utiliser :

$$E(dx, dy) = \sum_{x, y \in F} |I(x + dx, y + dy) - I(x, y)|^2$$



- Les 3 cas considérés par Moravec :
- **Cas 1** :  $E$  est faible dans toutes les directions
- **Cas 2** :  $E$  est faible pour des directions parallèles au contours et est fort sinon
- **Cas 3** :  $E$  est fort dans toutes les directions

## Détecteur de Moravec

---

Moravec recherche les maxima locaux du minimum de E en chaque pixel.

Principal défaut : son anisotropie.

Des coins seront détectés sur les contours de direction autre que les directions principales (pas invariant en rotation).

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- **Détecteur de Harris**
- Mettre en correspondance les points

## Région

- Par agrégation
- Par division
- Etiquetage

## Détecteur de Harris

---

Harris reprend la même mesure  $E(dx, dy)$  :

$$E(dx, dy) = \sum_{x,y} [I(x, y) - I(x + dx, y + dy)]^2$$

Pour un déplacement élémentaire  $(dx, dy)$  on a :

$$I(x + dx, y + dy) = I(x, y) + Ix dx + Iy dy$$

Et donc :  $E(dx, dy) = [dx, dy] \cdot M \cdot [dx, dy]^T$  avec  $M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$

$$A = Ix dx^2 * w \quad B = Iy dy^2 * w \quad C = Ix dx Iy dy * w \quad w = \exp^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

## Détecteur de Harris

---

Un coin est caractérisé par des grandes valeurs de E dans toutes les directions. Ceci correspond à des valeurs propres  $\lambda_1$  et  $\lambda_2$  importantes de la matrice M.

- Si  $\lambda_1 \approx 0$  et  $\lambda_2 \approx 0$ , le pixel est dans une zone homogène.
- Si  $\lambda_1 \approx 0$  et  $\lambda_2$  est grand et positif, le pixel est un pixel de contour.
- Si  $\lambda_1$  et  $\lambda_2$  sont grands et positifs, le pixel est un point d'intérêt.

Comme le calcul des valeurs propres est trop coûteux en temps de calcul, Harris et Stephens proposent d'utiliser :

$$H = \lambda_1\lambda_2 - c(\lambda_1 + \lambda_2)^2 = \det(M) - c \operatorname{Trace}(M)^2 = AB - C^2 - c(A + B)^2$$

Où  $c$  est une constante (0.05). Si on est proche d'un coin, H est grand.

## Détecteur de Harris

La détection est donc faire en recherchant les maxima locaux de  $H$ .

Carte de  $H$



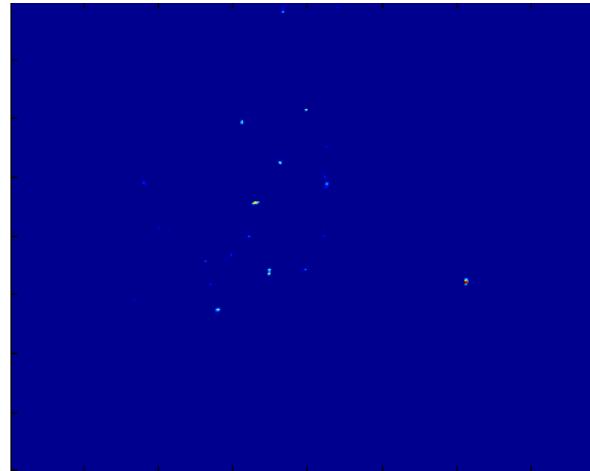
Image originale



Points détectés



Maxima



## Détecteur de Harris

En gardant tous les maxima locaux, beaucoup de bruit apparaît.

→ On peut garder les K plus grandes valeurs de maxima locaux.



K = 250



K = 500

Problème : la plupart des points sont localisés aux mêmes endroits.

## Détecteur de Harris

→ Détection adaptative des maxima locaux.

Un maxima est détecté s'il est plus grand d'au moins 10% que ses voisins dans un voisinage de taille  $r$ .



$K = 250$



$K = 250, r = 12$

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

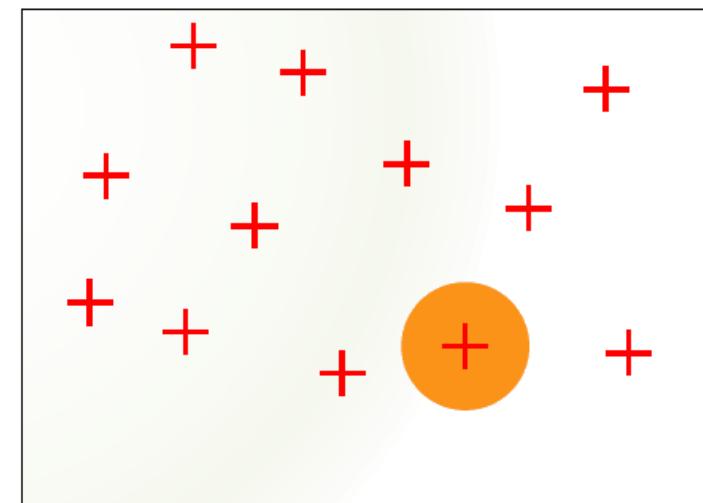
- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

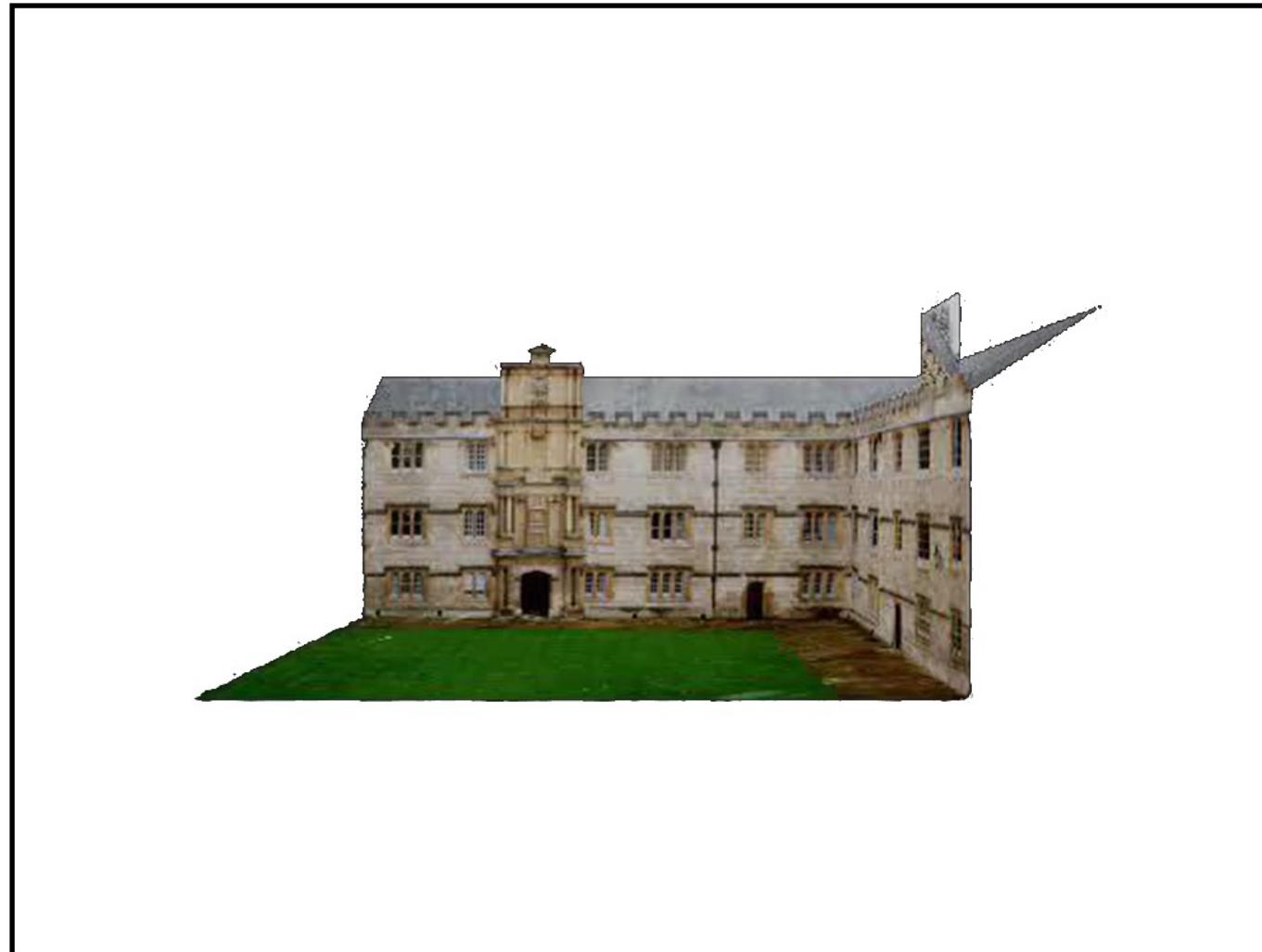
- Par agrégation
- Par division
- Etiquetage

## Mettre en correspondance les points

Un point est détecté dans I1. On prélève un voisinage autour de ce point (dont la taille dépend de l'échelle du point) et on va rechercher dans I2 un point qui a un voisinage similaire.



## Mettre en correspondance les points



## Mettre en correspondance les points

**Appariement des points d'intérêt par mesure de ressemblance**

Somme des différences au carrées (Sum of Squared Difference) :

$$SSD(x, y) = \sum_u \sum_v [I1(u, v) - I2(u + x, v + y)]^2$$

Somme des valeurs absolues des différences (Sum of Absolute Difference) :

$$SAD(x, y) = \sum_u \sum_v |I1(u, v) - I2(u + x, v + y)|$$

Somme des différences au carrées centrées (Zero mean Sum of Squared Difference) :

$$ZSSD(x, y) = \sum_u \sum_v \left[ \{I1(u, v) - \bar{I1}\} - \{I2(u + x, v + y) - \bar{I2}\} \right]^2$$

Somme des valeurs absolues des différences centrées :

$$ZSAD(x, y) = \sum_u \sum_v \left| \{I1(u, v) - \bar{I1}\} - \{I2(u + x, v + y) - \bar{I2}\} \right|$$

# Mettre en correspondance les points

Appariement des points d'intérêt par mesure de ressemblance

Corrélation normalisée :

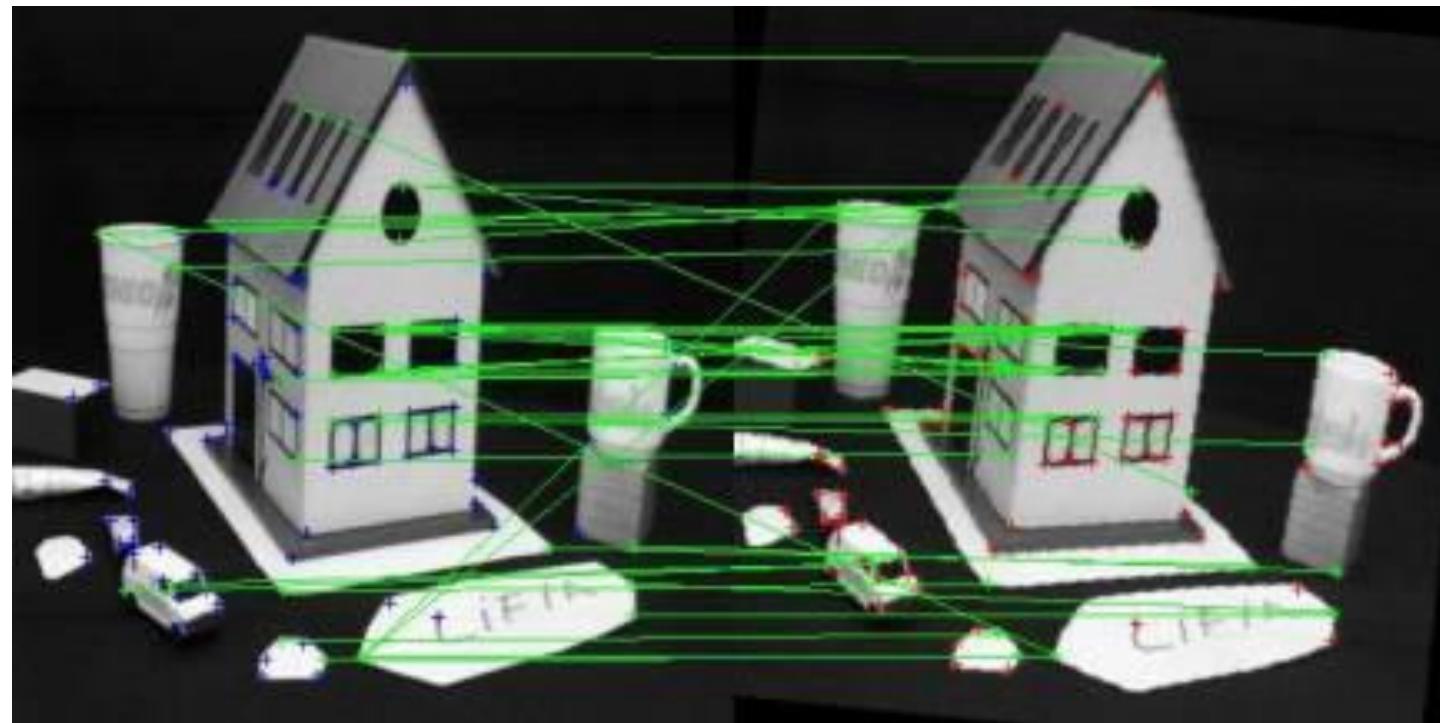
$$NCC(x, y) = \frac{\sum_{u} \sum_{v} I1(u, v) \cdot I2(u + x, v + y)}{\sqrt{\sum_{u} \sum_{v} I1(u, v)^2 \sum_{u} \sum_{v} I2(u + x, v + y)^2}}$$

Corrélation normalisée centrée :

$$ZNCC(x, y) = \frac{\sum_{u} \sum_{v} \{I1(u, v) - \bar{I1}\} \cdot \{I2(u + x, v + y) - \bar{I2}\}}{\sqrt{\sum_{u} \sum_{v} \{I1(u, v) - \bar{I1}\}^2 \sum_{u} \sum_{v} \{I2(u + x, v + y) - \bar{I2}\}^2}}$$

# Mettre en correspondance les points

## Exemple d'appariement



# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

- Par agrégation
- Par division
- Etiquetage

# Régions

---

## Région

Zone homogène au sens d'un critère donné appelé prédictat d'homogénéité  $P(R)$  qui peut dépendre du niveau de gris, de la couleur, de la texture,...

## Définition

Les régions doivent respecter des conditions :

- L'union de toutes les régions forme l'image
- Une région est composée de pixels connexes
- $P(R)$  est vrai pour toutes les régions
- $P(R_i \cup R_j)$  est faux pour tout  $i \neq j$

## Plusieurs attributs

Aire, périmètre, centre de gravité, axes principaux,...

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

- Par agrégation
- Par division
- Etiquetage

# Segmentation par agrégation de pixels

---

On regroupe les régions avec un double critère d'adjacence et d'homogénéité.

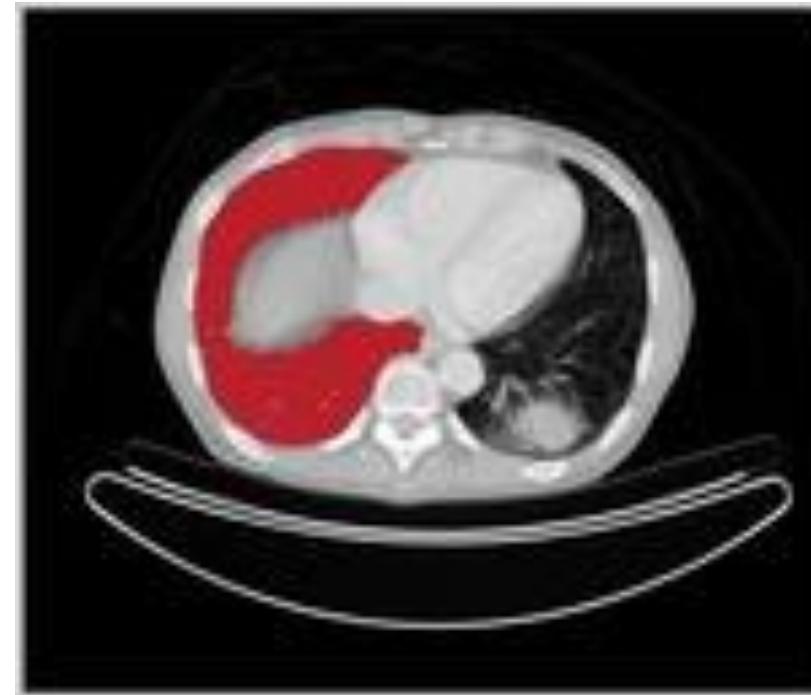
## Agrégation de pixels :

- On choisit un germe qui initialise la région R de ma moyenne  $\mu_R$ .
- On fait croître ce germe tant que des pixels de son voisinage  $I(x)$  vérifient le prédictat d'homogénéité ( $|I(x) - \mu_R| <$  seuil par exemple).
- Lorsqu'il n'y a plus de pixel candidat dans le voisinage, on choisit un nouveau germe et on itère le processus.

# Segmentation par agrégation de pixels

**Choix du germe :**

- Aléatoire
- Par ordre séquentiel, par balayage de l'image
- A partir du plus grand mode de l'histogramme



# Segmentation par agrégation de pixels

---

## Extension

Plusieurs prédictats d'homogénéité plus ou moins complexes peuvent être choisis, homogénéité locale, globale, utilisation de la variance,...

## Avantage

→ Très simple à mettre en œuvre

## Inconvénient

→ Très sensible au choix des germes  
→ Très sensible au prédictat d'homogénéité choisi

# Segmentation par agrégation de pixels

Critère d'homogénéité :  $\max(R;G;B) < \text{seuil}$



Image originale



Seuil = 6



Seuil = 7

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

- Par agrégation
- **Par division**
- Etiquetage

# Segmentation par division

---

## Division :

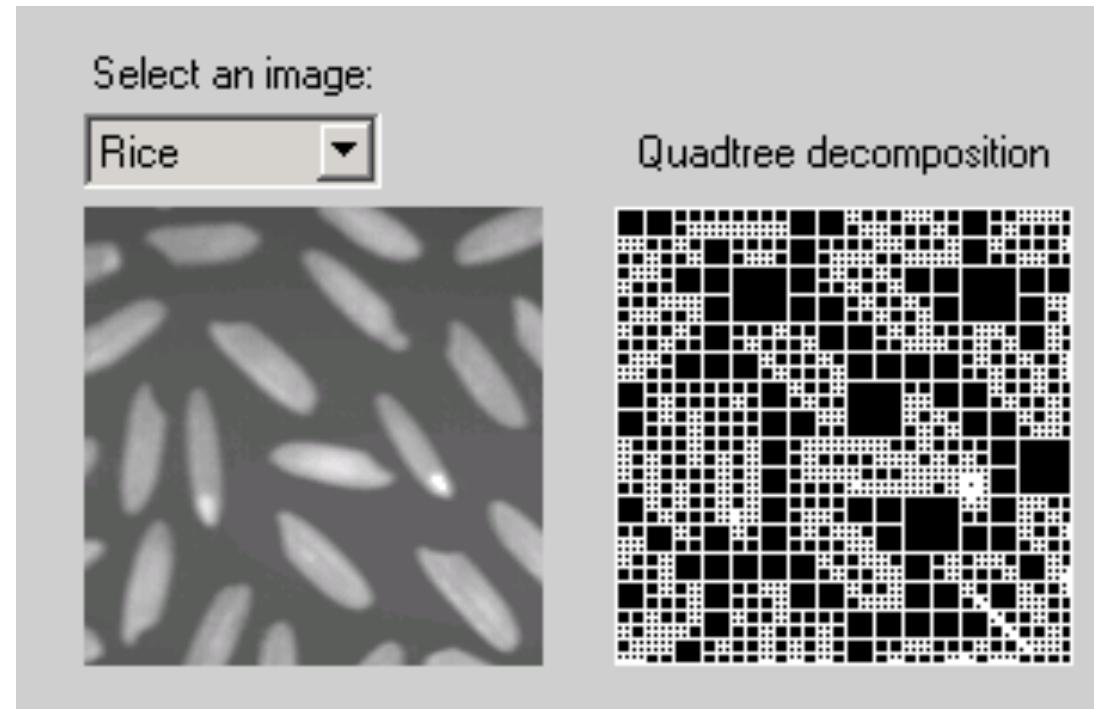
- Tous les pixels appartiennent à la même région.
- Toutes les régions qui ne vérifient pas le prédicat d'homogénéité sont divisées en plusieurs régions.
- Chaque région nouvellement créée est redivisée si elle ne vérifie pas le prédicat d'homogénéité.
- On itère jusqu'à stabilisation.

Un critère sur la taille des régions peut être ajouté.

# Segmentation par division

**Division géométrique :**

Exemple : toute région non homogène est divisée en 4

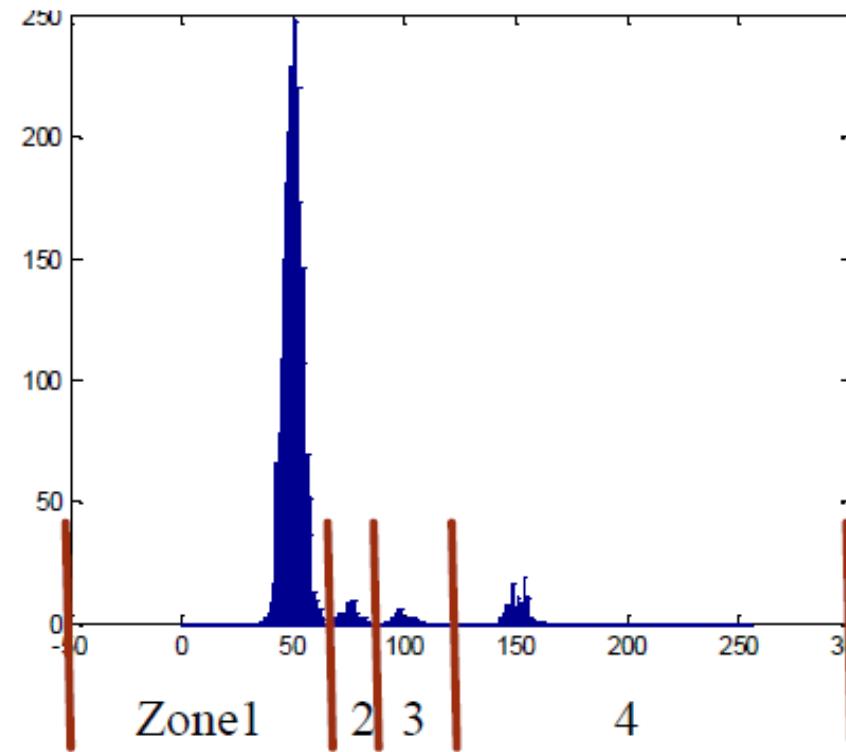
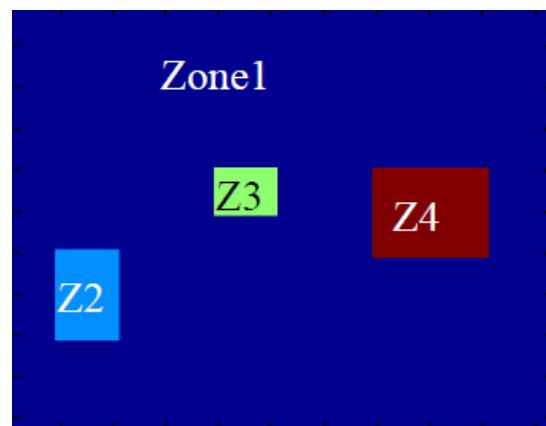
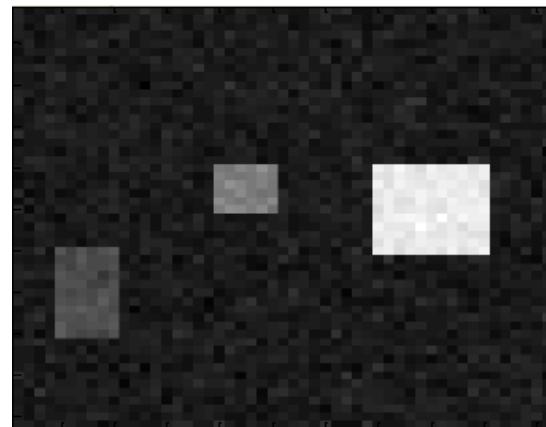


Ici 1564 régions

# Segmentation par division

Division avec l'histogramme :

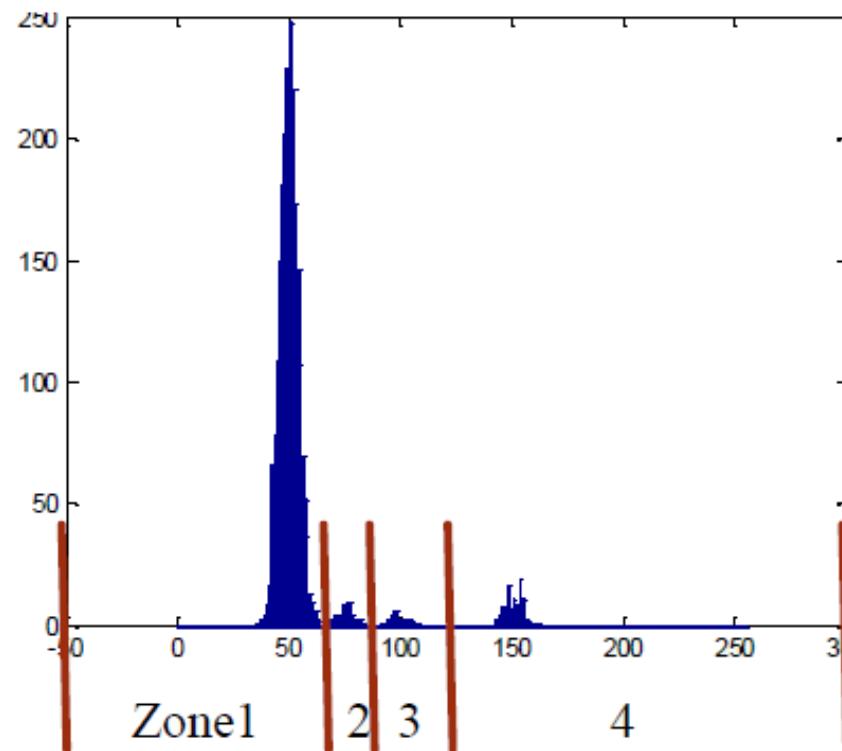
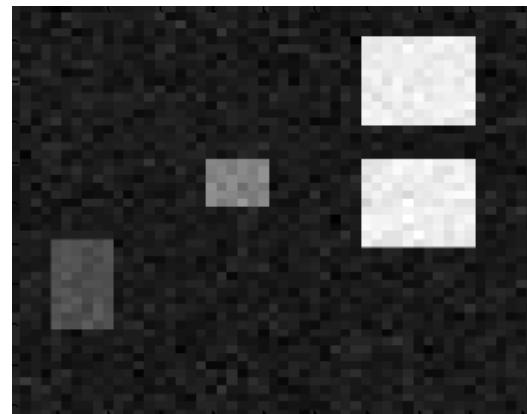
Les modes de l'histogramme servent à définir des seuils



# Segmentation par division

Division avec l'histogramme :

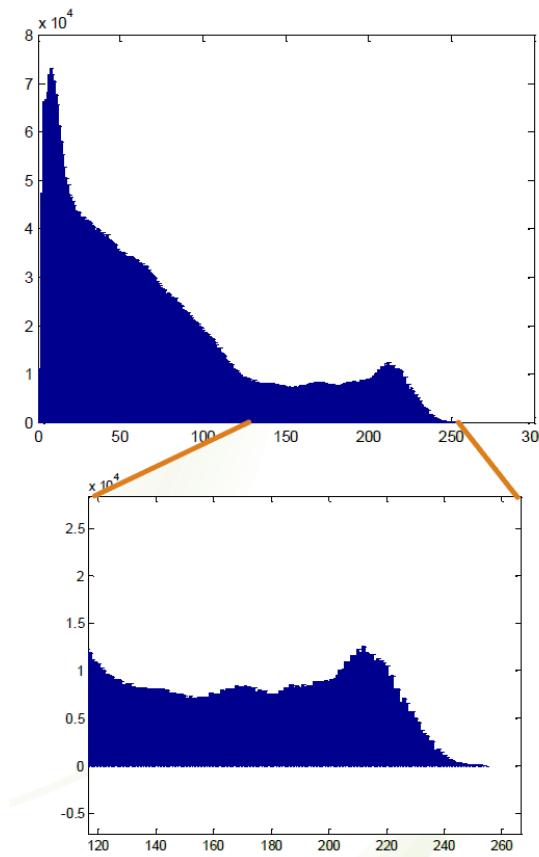
Attention les zones ne sont pas forcément adjacentes ! → Etiquetage



# Segmentation par division

Division avec l'histogramme :

Ne marche pas avec les images plus complexes



# Segmentation par division

---

**Division avec les k-means :**

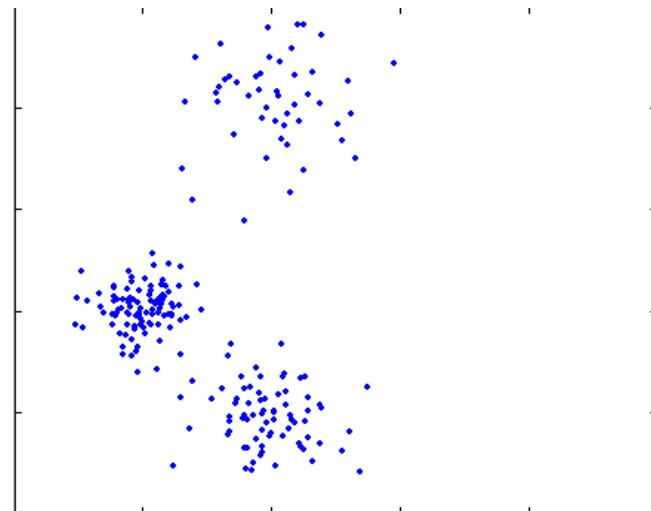
k-means partitionne un ensemble de N points en K classes. K est connu.

**Principe :**

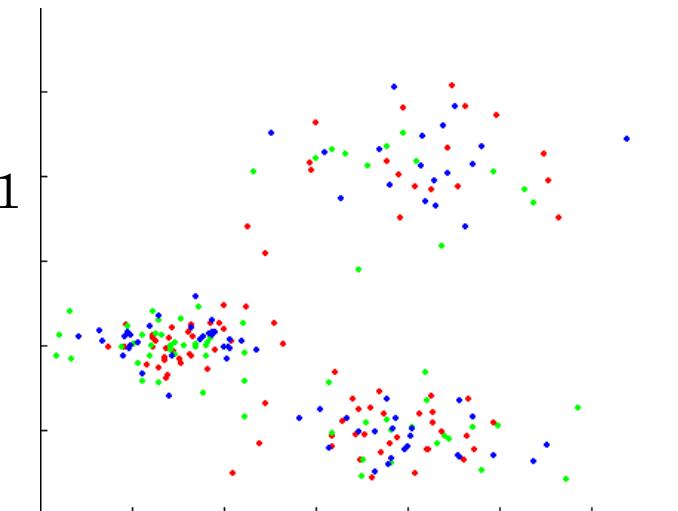
- On affecte aléatoirement chaque point à un des K groupes.
- On calcule la barycentre de chacun des groupes.
- On réaffecte chaque point au groupe dont le centre est le plus proche.
- On itère jusqu'à convergence.

# Segmentation par division

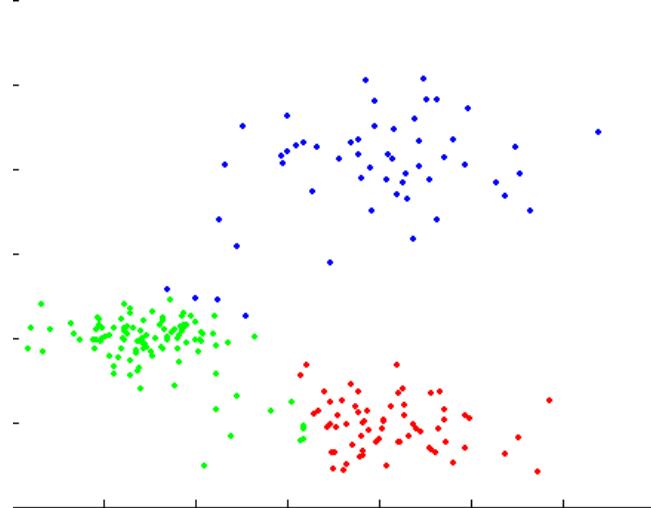
Division avec les k-means :



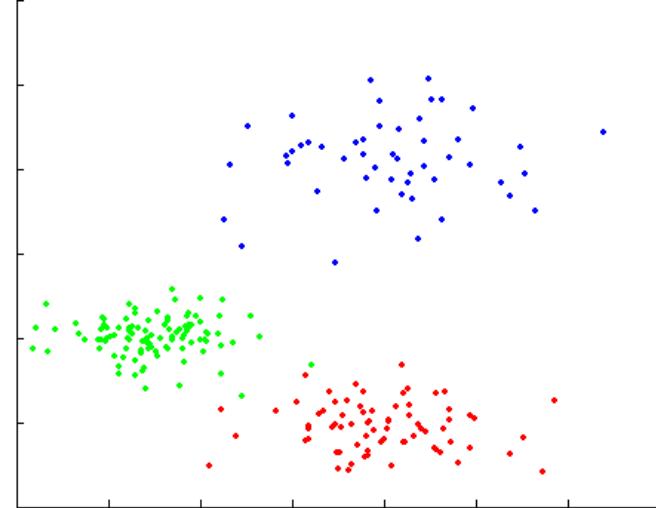
it = 0



it = 1



it = 2



it = 3

# Segmentation par division

## Division avec les k-means :

Pour des images couleurs, chaque point est un vecteur de dimension 3 (R,V,B)



Image originale

Régions non connexes  
→ Etiquetage

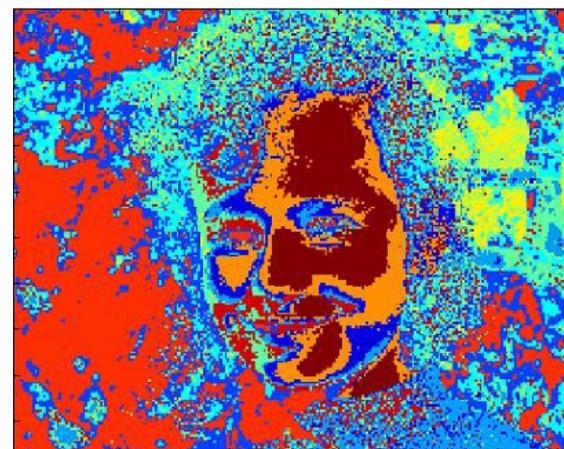


Image avec le numéro de la classe ( $K = 10$ )

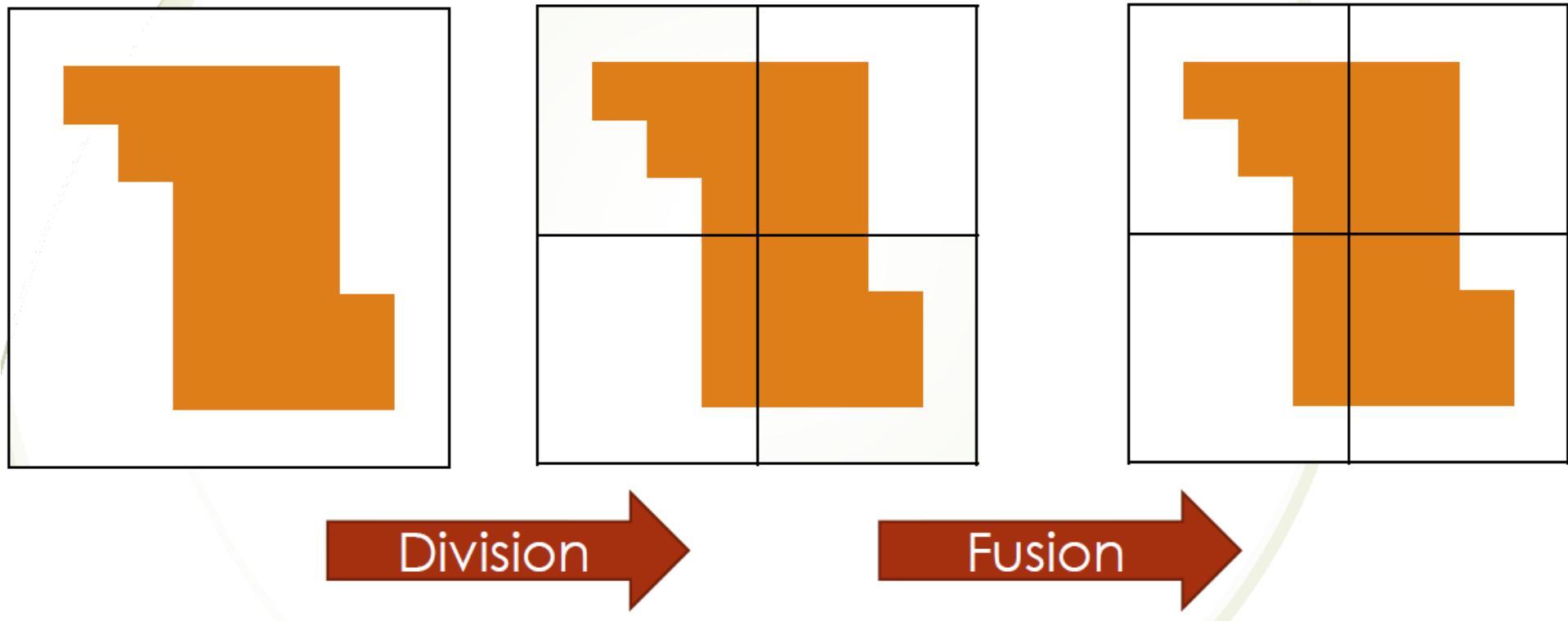


Image avec la couleur moyenne de la classe

## Segmentation par division/fusion

On alterne les étapes de division et de fusion :

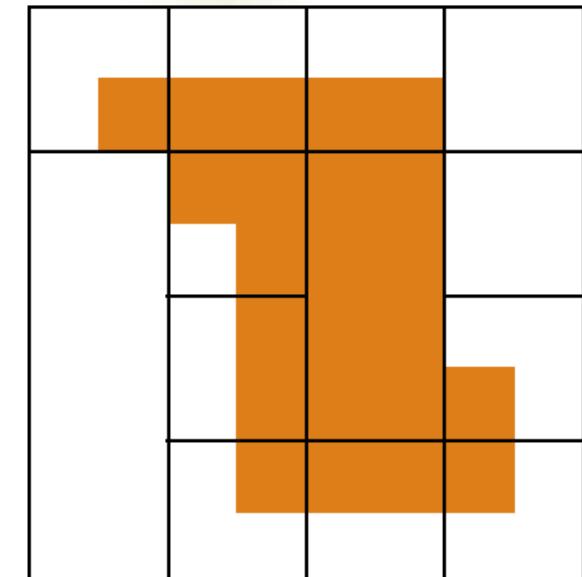
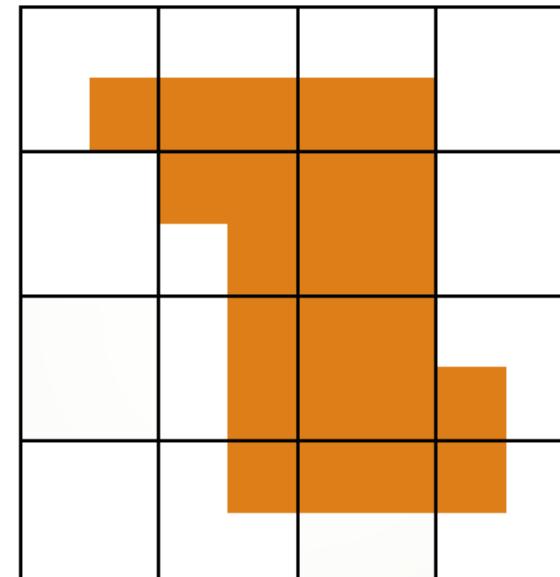
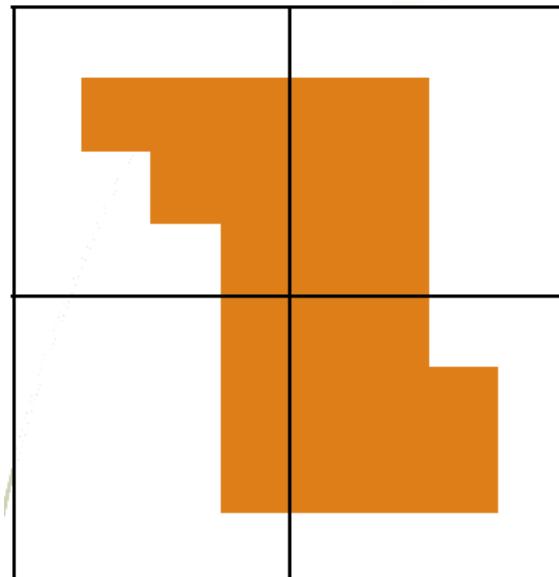
Etape 1



## Segmentation par division/fusion

On alterne les étapes de division et de fusion :

Etape 2



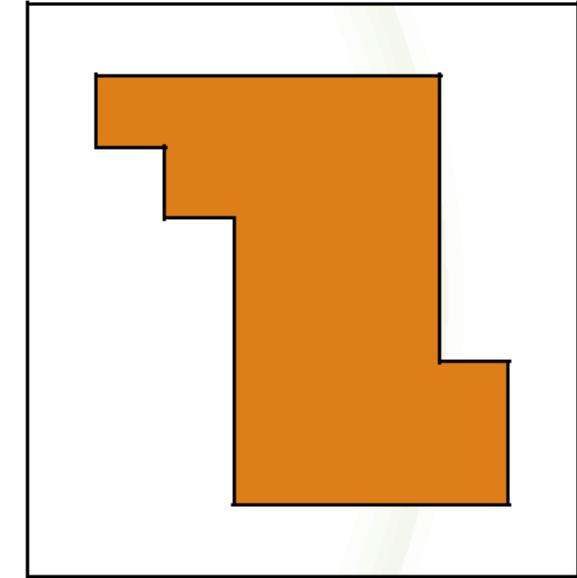
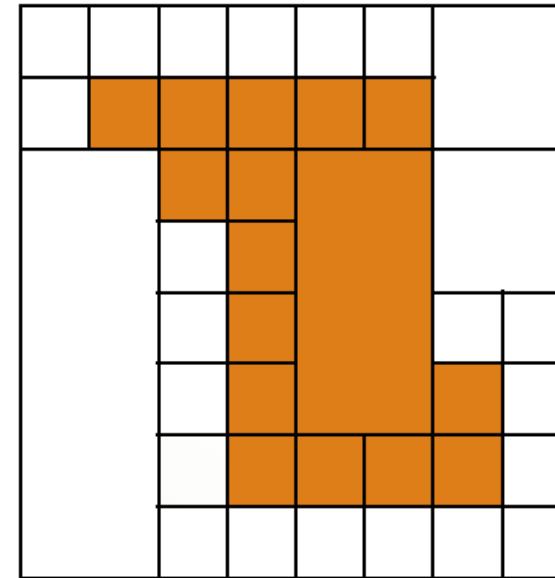
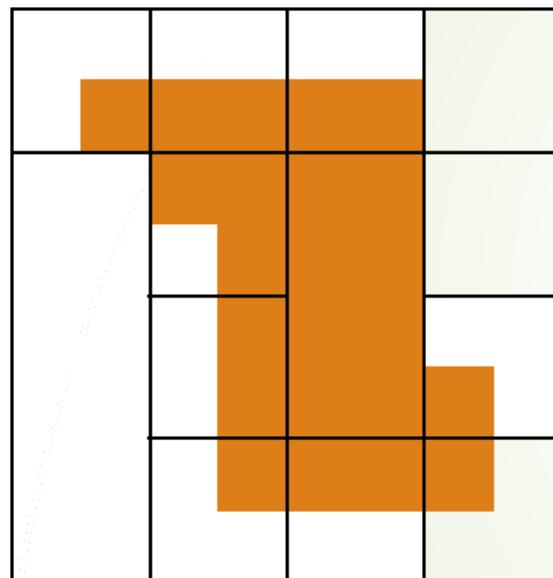
Division

Fusion

# Segmentation par division/fusion

On alterne les étapes de division et de fusion :

Etape 3



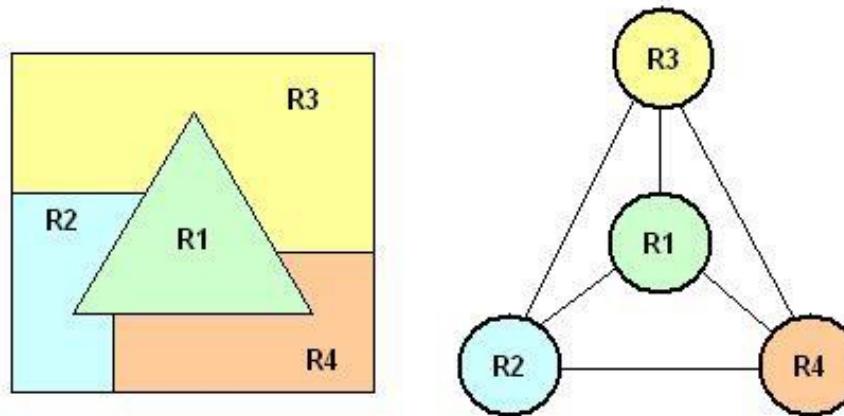
Division

Fusion

## Segmentation par division/fusion

On peut faire de même avec les autres types de division.

On utilise alors un graphe d'adjacence pour la fusion :



### Principe :

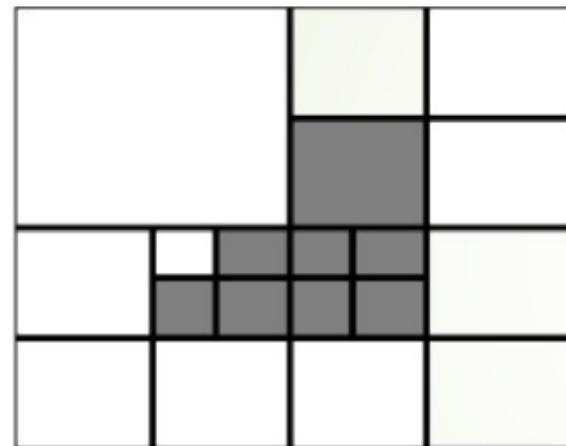
Un score de fusion est calculé pour tous les couples de nœuds adjacents.

Regroupement des deux meilleures régions si score > seuil.

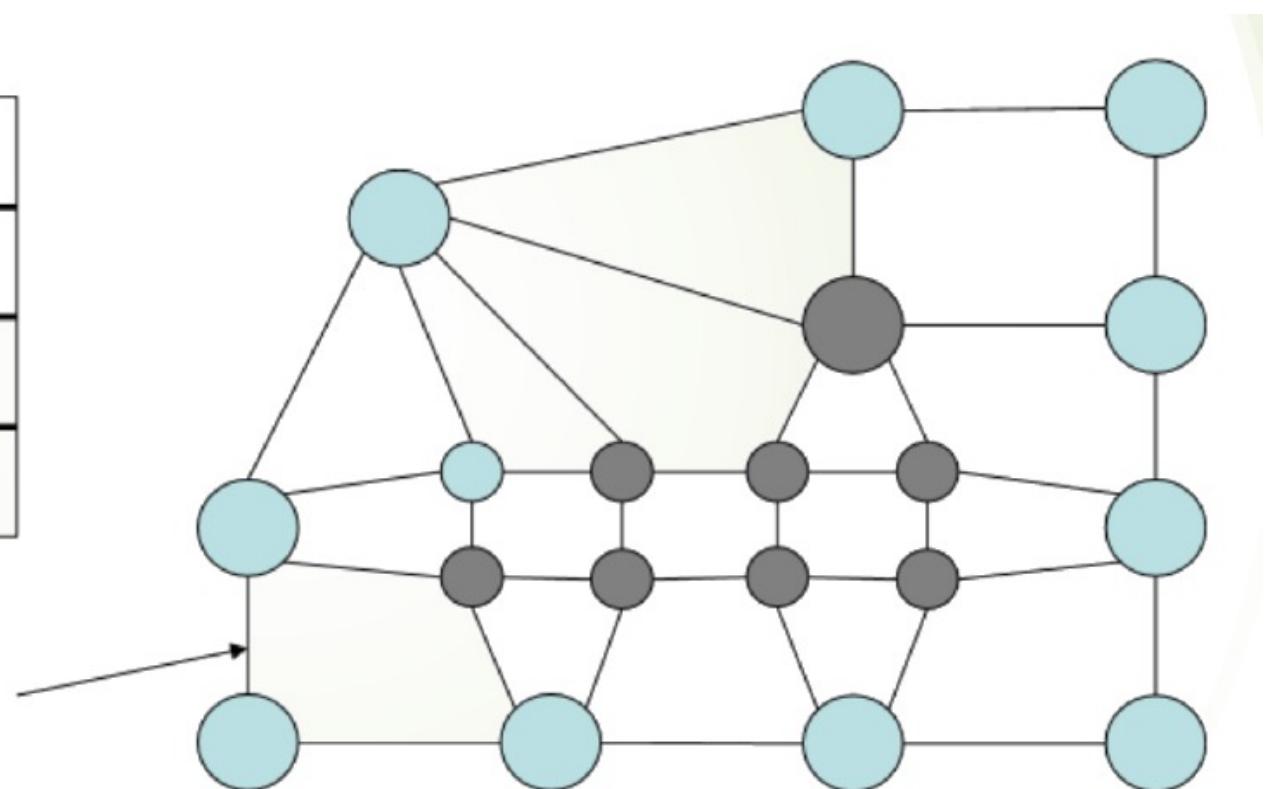
Mise à jour du graphe d'adjacence.

# Segmentation par division/fusion

Exemple



La distance en terme d'homogénéité de régions est portée par l'arrête valuée qui les relie dans le RAG



# Segmentation par division/fusion

## Inconvénients :

Difficile à mettre en place, régions un peu carrées

## Avantages :

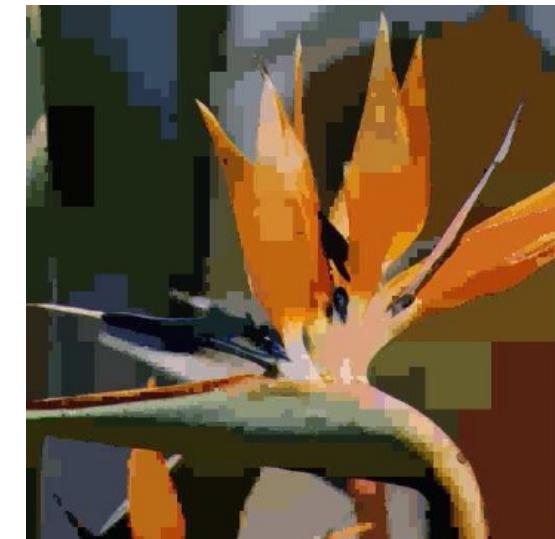
Ne dépend pas d'un germe initial, découpage plus logique que dans les approches par division



Image originale



Image en fausse couleur



Résultat avec couleur  
sur-imposée

# Segmentation

---

## Contours

- Approche gradient
- Approche laplacien
- Transformée de Hough

## Points d'intérêt

- A partir des contours
- Détecteur de Moravec
- Détecteur de Harris
- Mettre en correspondance les points

## Région

- Par agrégation
- Par division
- Etiquetage

## Etiquetage en composantes connexes



Image des attributs

Etiquetage de chaque région par un numéro → 7 régions

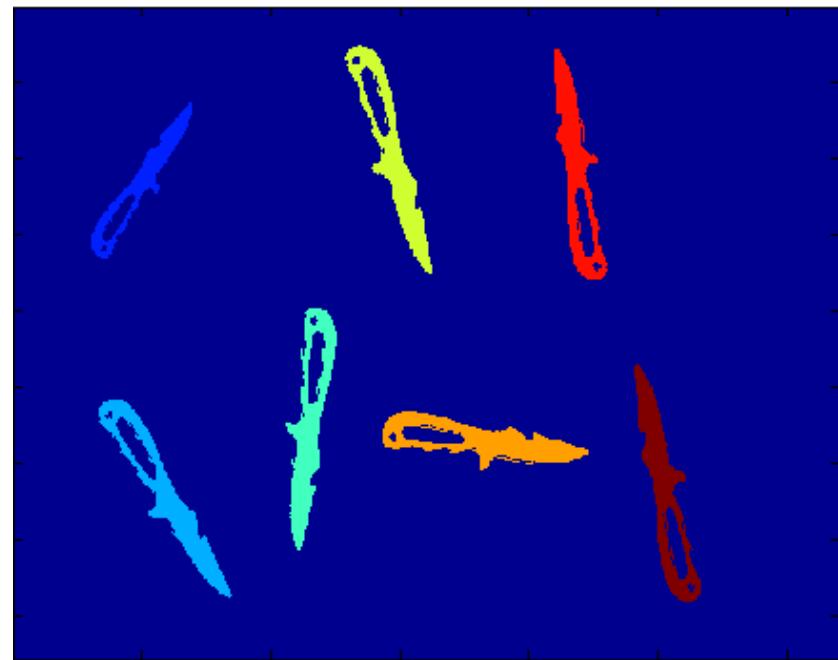


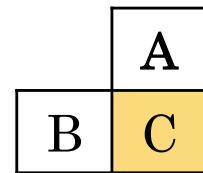
Image des étiquettes

## Etiquetage en composantes connexes

---

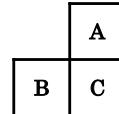
On veut regrouper tous les pixels connexes appartenant aux objets.

Pour cela, on balaye l'image et donne une étiquette à chaque nouveau pixel.

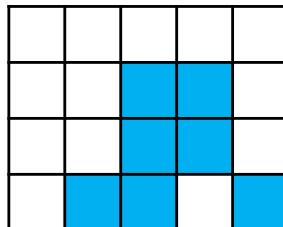


Pixels A et B déjà étiquetés  
Pixel C à étiqueter

# Etiquetage en composantes connexes



Etiquette : numéro identique attribué à chaque pixel d'une même composante connexe.



Principe : balayage de l'image par un motif 3 pixels (A,B,C).

NB : initialiser bordures haute / gauche à 0.

$N = Nb$  étiquettes (initialisé à 0).

Si  $C = 0$  : pixel de fond, il reste invariant

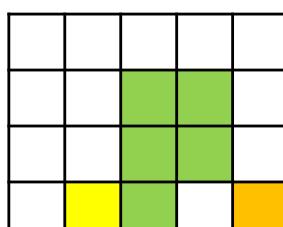
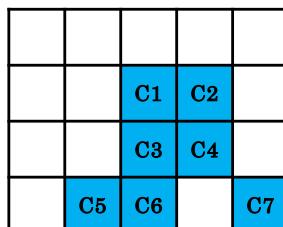
Si  $C \neq 0$  : Si  $A = B = 0$ , alors  $C = N+1$

Si  $A = 0$  et  $B \neq 0$ , alors  $C = B$  et inversement

Si  $A \neq 0$  et  $B \neq 0$  et  $A = B$ , alors  $C = A$

Si  $A \neq 0$  et  $B \neq 0$  et  $A \neq B$ , alors  $C = \min(A, B)$

et équivalence  $\max(A, B) \sim C$



C1	C2	C3	C4	C5	C6	C7
A=0, B=0 $N=N+1$ $C=N=1$	A=0, B=1 $C=B=1$	A=1, B=0 $C=A=1$	A=B=1 $C=A=1$	A=0, B=0 $N=N+1$ $C=\min(A, B)=1$ et équivalence $\max(A, B) \sim C$ $2 \sim 1$	A=1, B=2 $C=\min(A, B)=1$ et équivalence $\max(A, B) \sim C$ $2 \sim 1$	A=0, B=0 $N=N+1$ $C=N=3$

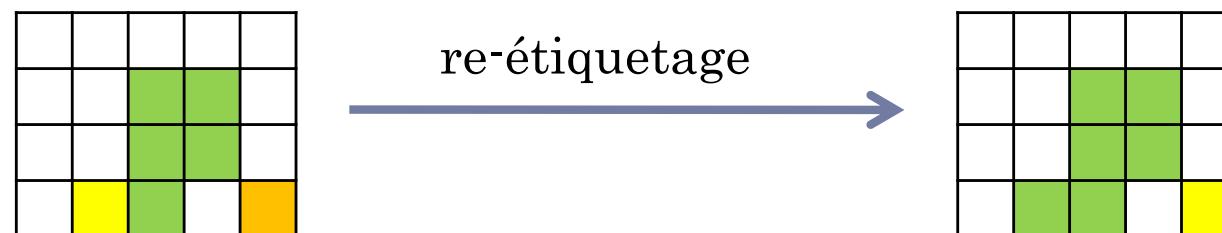
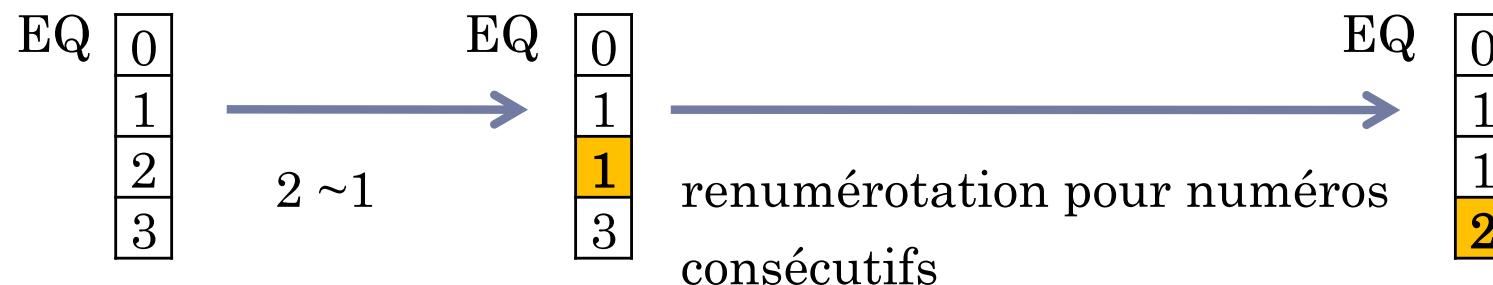
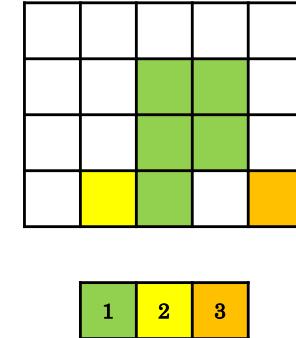
## Etiquetage en composantes connexes

Gestion des équivalences.

En cas d'équivalences (exemple  $2 \sim 1$ ), deux solutions :

- retour en arrière et réécriture de tous les « 2 » → lent !
- mémorisation dans une table puis second balayage

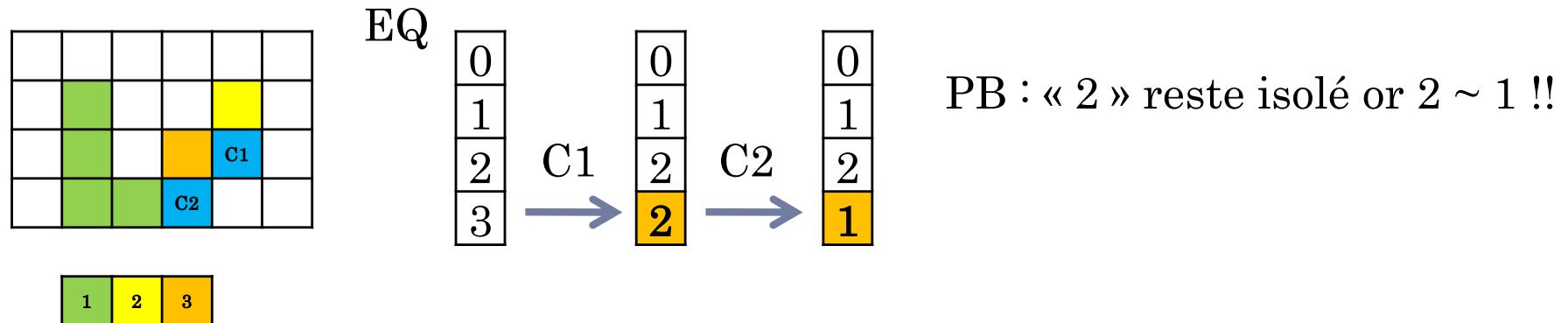
Chaque ligne de la table est initialisée par son numéro :



## Etiquetage en composantes connexes

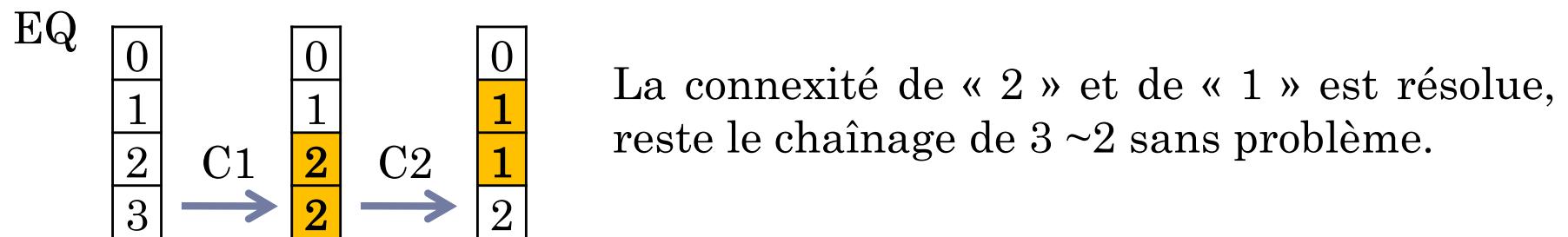
Rupture de chaîne d'équivalence.

En C1 :  $3 \sim 2$  puis en C2 :  $3 \sim 1 \rightarrow C = \min(A, B)$  et  $EQ(\max(A, B)) = C$



Solution : nouvelle formule complétée :

Si  $A \neq 0$  et  $B \neq 0$  et  $A \neq B \rightarrow A = EQ(A)$ ,  $C = \min(A, B)$ ,  $EQ(A) = EQ(B) = C$



# Etiquetage en composantes connexes

---

## Algorithme évolué d'étiquetage

- Mise à jour de la table de correspondance :

Vraies régions tant que  $EQ(i) = i$ . On les numérote par ordre croissant.

Pour les autres régions  $EQ(i) = EQ(EQ(i))$ .

- Deuxième balayage :

A chaque pixel d'étiquette  $i$ , on affecte l'étiquette  $EQ(i)$ .

# Caractéristiques extraites des régions

---

## Orientation

$$\theta = \frac{1}{2} \arctan\left(\frac{2M_{11}}{M_{20} - M_{02}}\right)$$

Moments centrés pour être invariants en translation

$$\mu_{pq} = \sum_i \sum_j (x_i - \bar{x})^p (y_j - \bar{y})^q I(x_i, y_j)$$

Moments normalisés pour être invariants en changement d'échelle

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\chi}} \quad \text{avec} \quad \chi = \frac{p+q}{2} \quad \text{et } p+q>2$$

# Morphologie mathématique

# Morphologie mathématique

---

## Binarisation d'images

- Choix manuel du seuil
- Minimisation de la variance intra-classe : méthode d'Otsu
- Seuillage entropique
- Minimisation de l'erreur de classification bayesienne

## Notion de morphologie mathématique

## Squelettisation

# Morphologie mathématique

---

## Binarisation d'images

- Choix manuel du seuil
- Minimisation de la variance intra-classe : méthode d'Otsu
- Seuillage entropique
- Minimisation de l'erreur de classification bayesienne

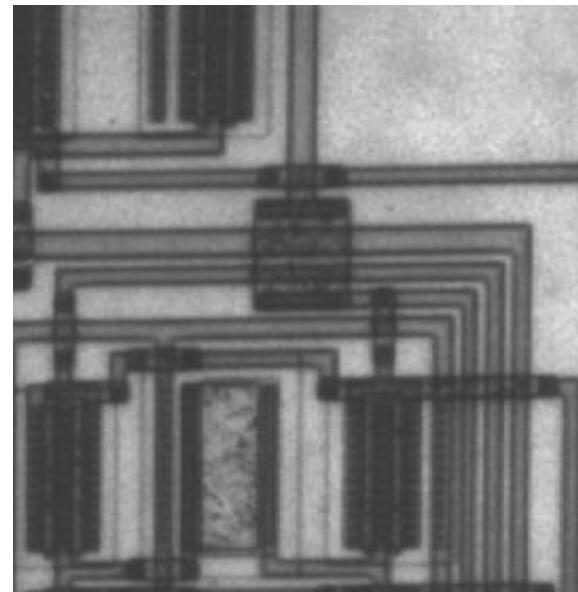
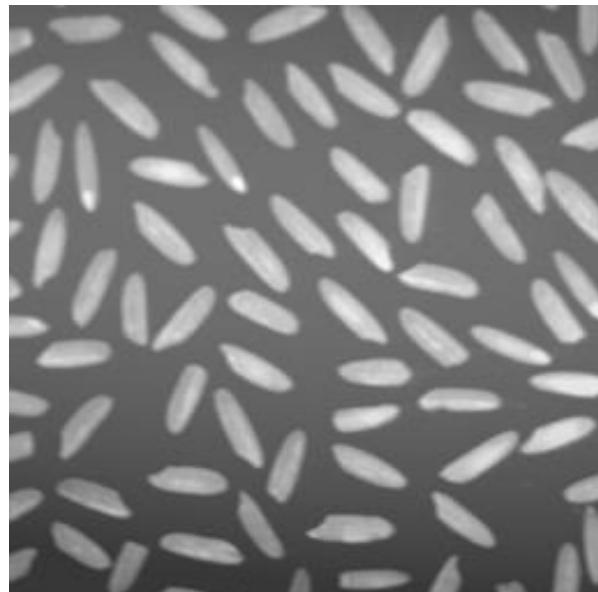
## Notion de morphologie mathématique

## Squelettisation

# Choix manuel du seuil

## Images bimodales

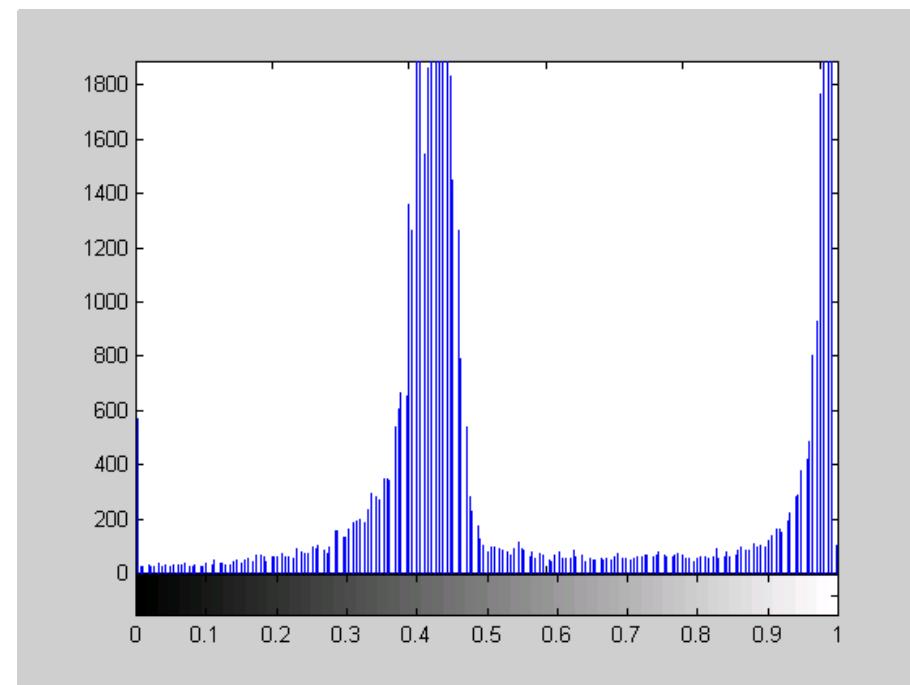
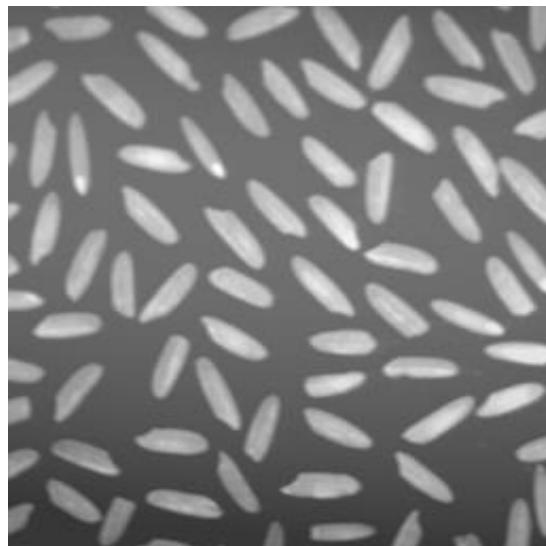
Cas particulier important par ses applications : industrie, temps réel, ...



→ Choix du seuil pour séparer les objets du fond

# Binarisation avec l'histogramme

Minimum entre les deux pics → seuil de binarisation



# Morphologie mathématique

---

## Binarisation d'images

- Choix manuel du seuil
- Minimisation de la variance intra-classe : méthode d'Otsu
- Seuillage entropique
- Minimisation de l'erreur de classification bayesienne

## Notion de morphologie mathématique

## Squelettisation

## Méthode d'Otsu

---

But : minimiser la variance intra-classe :

$$\omega_1\sigma_1 + \omega_2\sigma_2$$

Où :  $\omega_i$  est la probabilité d'être dans la classe i.

$\sigma_i$  est l'écart type des niveaux de gris de la classe i.

Minimiser la variance intra-classe revient à maximiser :

$$\text{Critère} = \omega_1\omega_2 [\mu_1 - \mu_2]$$

Où  $\mu_i$  est la moyenne des niveaux de gris de la classe i.

## Méthode d'Otsu

---

Algorithme d'Otsu :

- Parcourir tous les seuils possibles d'intensité
- Pour chaque seuil calculer le critère
- Choisir le seuil qui maximise le critère

# Morphologie mathématique

---

## Binarisation d'images

- Choix manuel du seuil
- Minimisation de la variance intra-classe : méthode d'Otsu
- **Seuillage entropique**
- Minimisation de l'erreur de classification bayesienne

## Notion de morphologie mathématique

## Squelettisation

## Seuillage entropique

---

Le seuillage entropique provient de la théorie de l'information. On recherche à maximiser l'entropie  $E$  qui mesure la quantité d'information portée par chaque classe.

$$E = -p_1 \log_2(p_1) - p_2 \log_2(p_2)$$

Où  $p_i$  représente la probabilité des pixels de la classe  $i$ .

On cherche le seuil qui maximise cette entropie.

# Morphologie mathématique

---

## Binarisation d'images

- Choix manuel du seuil
- Minimisation de la variance intra-classe : méthode d'Otsu
- Seuillage entropique
- Minimisation de l'erreur de classification bayesienne

## Notion de morphologie mathématique

## Squelettisation

# Classification bayesienne

---

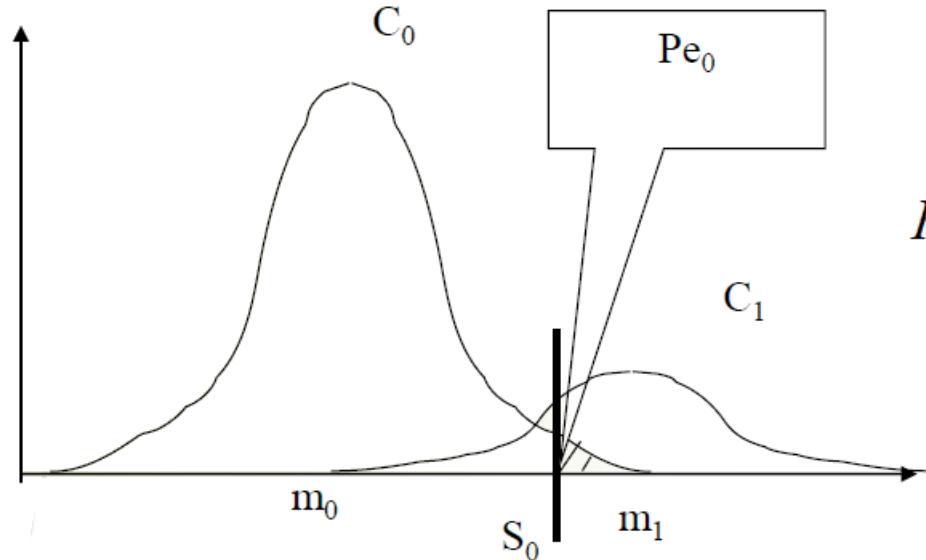
Hypothèses :

- Ndg répartis en 2 classes  $C_0$  et  $C_1$
- $N_0$  et  $N_1$  : effectifs
- Chaque classe modélisée par une loi gaussienne
- $m_0$  et  $m_1$  : moyennes des gaussiennes
- $V_b$  : variance des gaussiennes (même variances car bruit stationnaire)

On cherche le seuil  $S_0$  qui minimise la probabilité d'erreur :

$$\text{Perreur} = Pe_0 \cdot P(C_0) + Pe_1 \cdot P(C_1)$$

# Classification bayesienne



$$Pe_0 = \frac{1}{\sqrt{2 \pi V_b}} \int_{S_0}^{\infty} \exp\left(-\frac{(S - m_0)^2}{2 V_b}\right) dS$$

$$Pe_1 = \frac{1}{\sqrt{2 \pi V_b}} \int_{-\infty}^{S_0} \exp\left(-\frac{(S - m_1)^2}{2 V_b}\right) dS$$

# Classification bayesienne

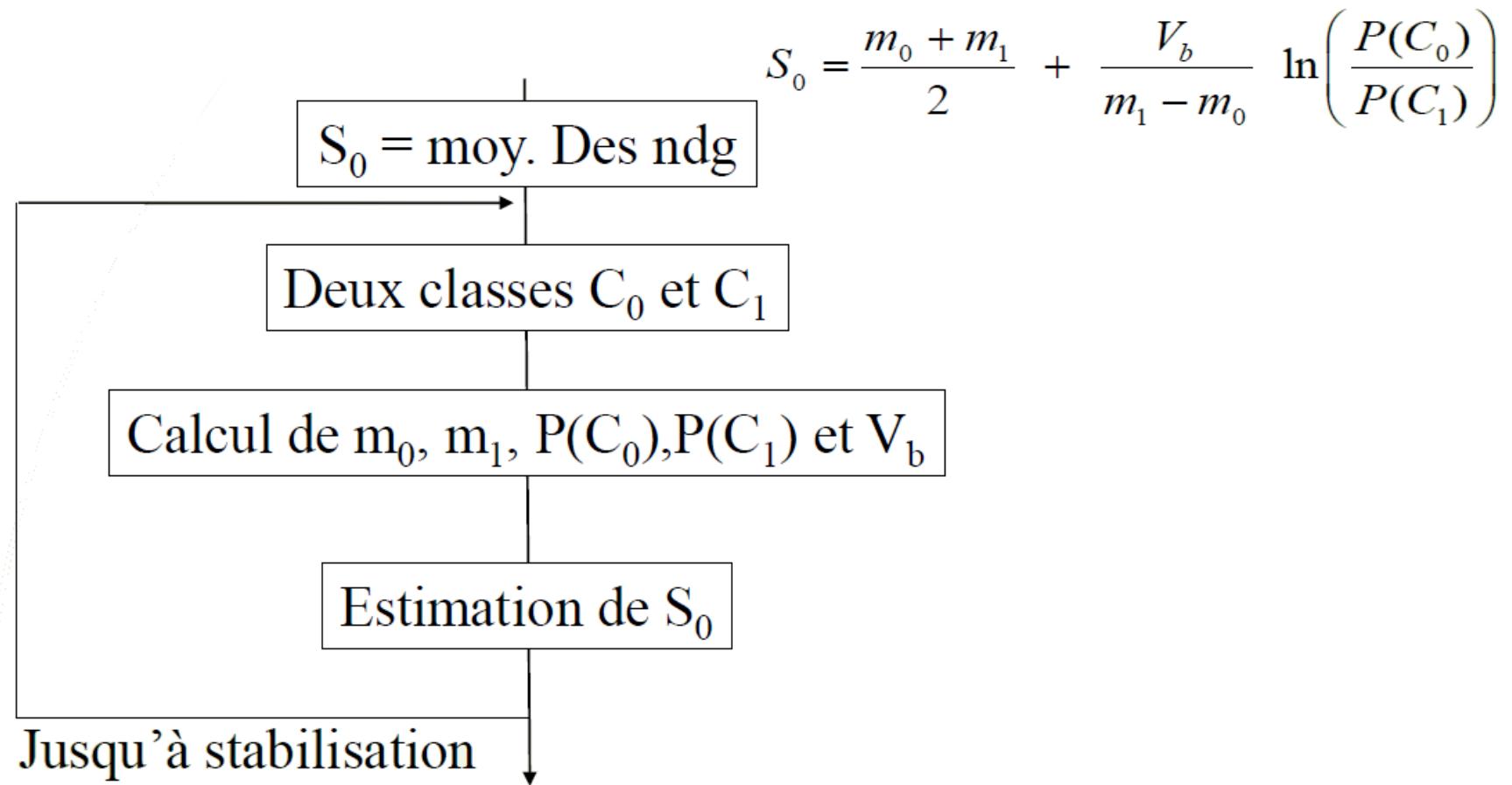
---

$$\frac{dP\text{erreur}}{dS} = \frac{1}{\sqrt{2 \pi V_b}} \left[ -P(C_0) \exp\left(-\frac{(S_0 - m_0)^2}{2 V_b}\right) + P(C_1) \exp\left(-\frac{(S_0 - m_1)^2}{2 V_b}\right) \right] = 0$$

$$\ln(P(C_0)) - \frac{(S_0 - m_0)^2}{2 V_b} = \ln(P(C_1)) - \frac{(S_0 - m_1)^2}{2 V_b}$$

$$\Rightarrow S_0 = \frac{m_0 + m_1}{2} + \frac{V_b}{m_1 - m_0} \ln\left(\frac{P(C_0)}{P(C_1)}\right)$$

# Classification bayesienne



# Morphologie mathématique

---

## Binarisation d'images

- Choix manuel du seuil
- Minimisation de la variance intra-classe : méthode d'Otsu
- Seuillage entropique
- Minimisation de l'erreur de classification bayesienne

## Notion de morphologie mathématique

## Squelettisation

# Morphologie mathématique

C'est une théorie ensembliste

Notations et définitions :

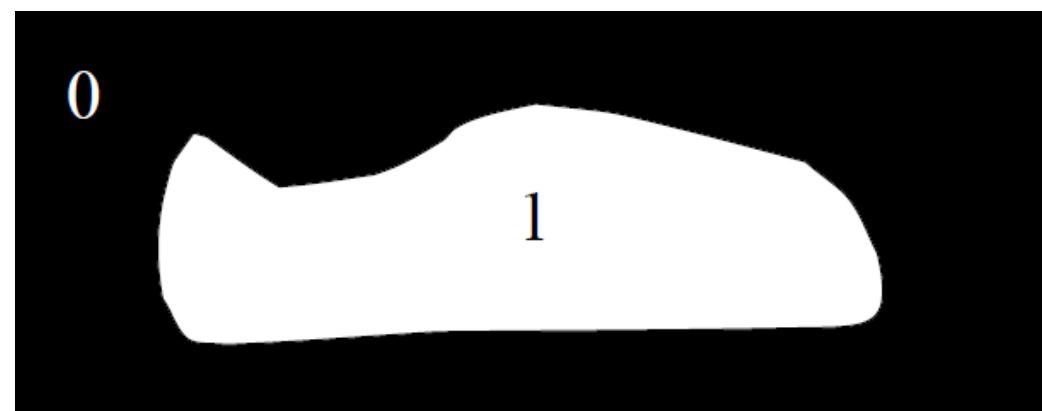
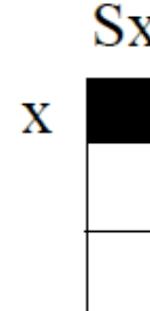
Objets :  $X = \{ x / x = 1 \}$

Fond :  $\sim X = \{ x / x = 0 \}$

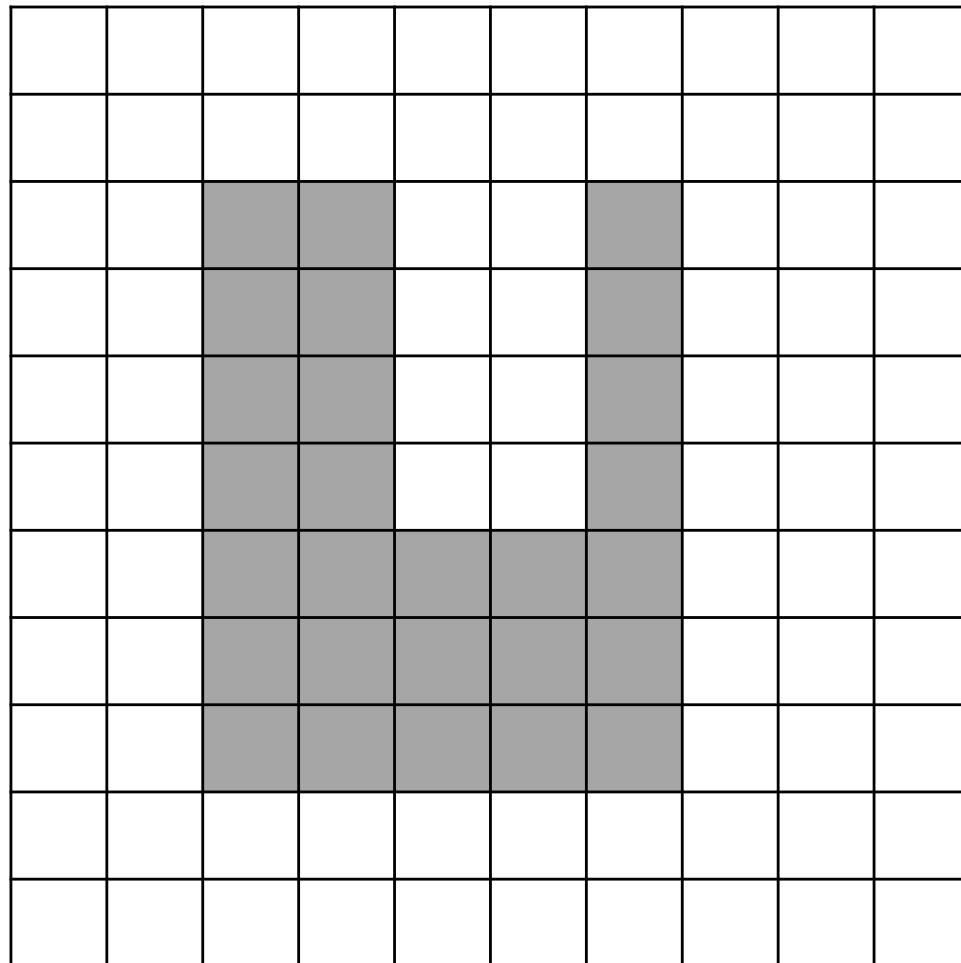
Élément structurant :

$S_x$  caractérisé par ses dimensions et sa forme

Son centre  $x$  est le point d'application de l'opération locale



# Morphologie mathématique



## Dilatation morphologique

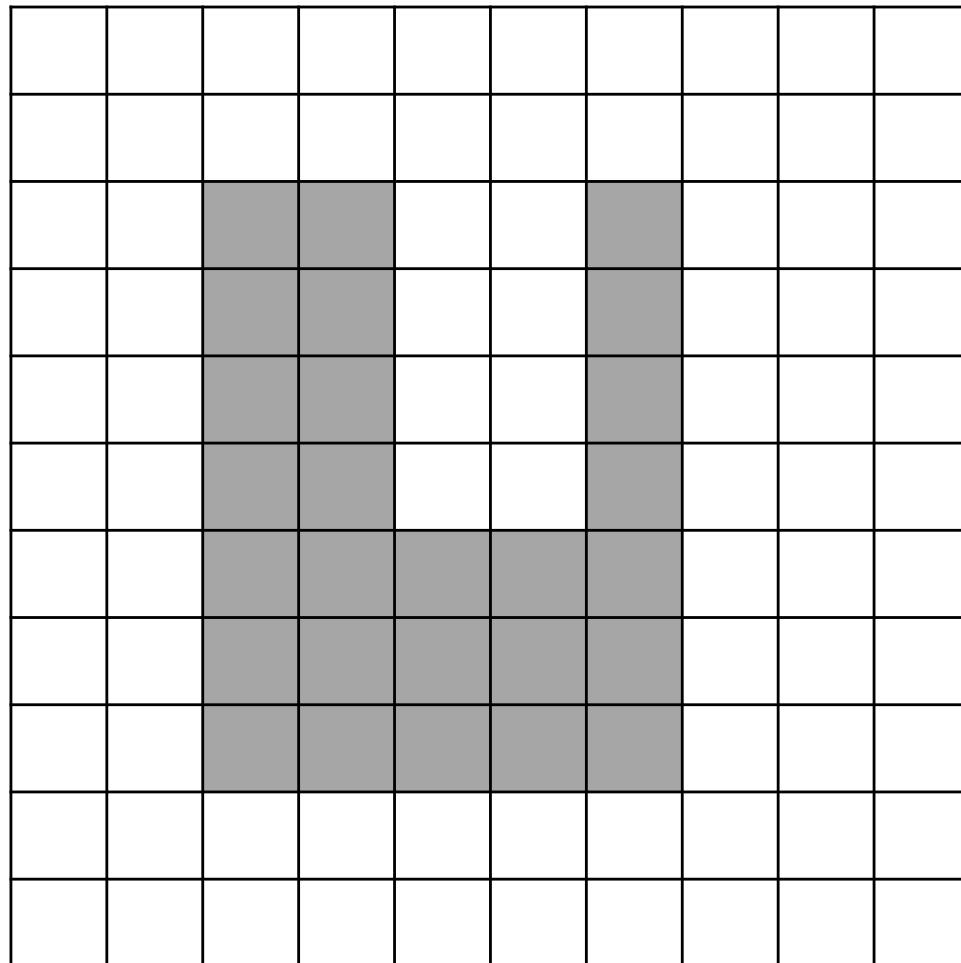
$$\begin{aligned} X \oplus S &= \{ x / Sx \cap X \neq \emptyset \} \\ &= \{ x / (Sx \cap X) \subset X \} \end{aligned}$$

### Exercice :

Dilater l'objet ci-contre avec l'élément structurant :



# Morphologie mathématique



## Erosion morphologique

$$\begin{aligned} X \ominus S &= \{ x / (S_x \cap \sim X) \neq \Phi \} \\ &= \{ x / S_x \subset X \} \end{aligned}$$

### Exercice :

Eroder l'objet ci-contre avec l'élément structurant :



# Morphologie mathématique

---

## Interprétation avec un élément structurant particulier

$S_x$  : disque centré au point d'application, de rayon 1

2 formes : →  $S_{x8}$  si 8-connectivité V8

→  $S_{x4}$  si 4-connectivité V4

**Dilatation** :  $\forall x = 0$ , si  $\exists 1$  point de  $VN = 1 \rightarrow x = 1$

→ Croissance de 1 pixel

**Erosion** :  $\forall x = 1$ , si  $\exists 1$  point de  $VN = 0 \rightarrow x = 0$

→ Contraction de 1 pixel

# Algorithme basé sur la convolution

---

## Dilatation

$$\begin{aligned} X \oplus S &= \{ x / Sx \cap X \neq \emptyset \} \\ \rightarrow X \oplus S &= (S * \text{Image}) \neq 0 \end{aligned}$$

## Erosion

$$\begin{aligned} X \ominus S &= \{ x / (Sx \cap \sim X) \neq \emptyset \} \\ \rightarrow X \ominus S &= (S * \text{Image}) = \text{Card}(S) \end{aligned}$$

# Cascade d'opérations

---

## Ouverture

$$X \circ S = (X \ominus S) \oplus S$$

## Fermeture

$$X \bullet S = (X \oplus S) \ominus S$$

A quoi ça sert ???

# Cascade d'opérations

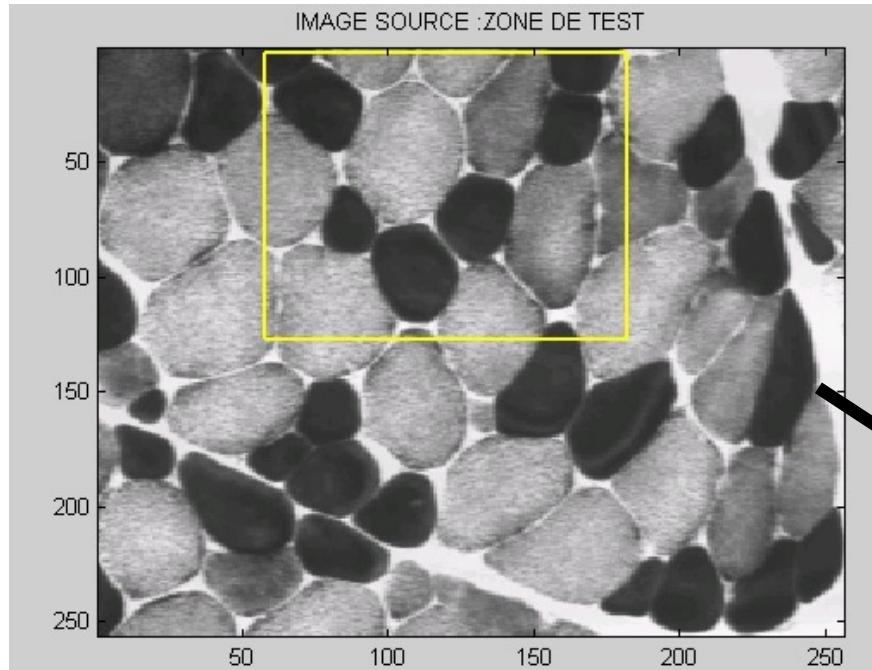
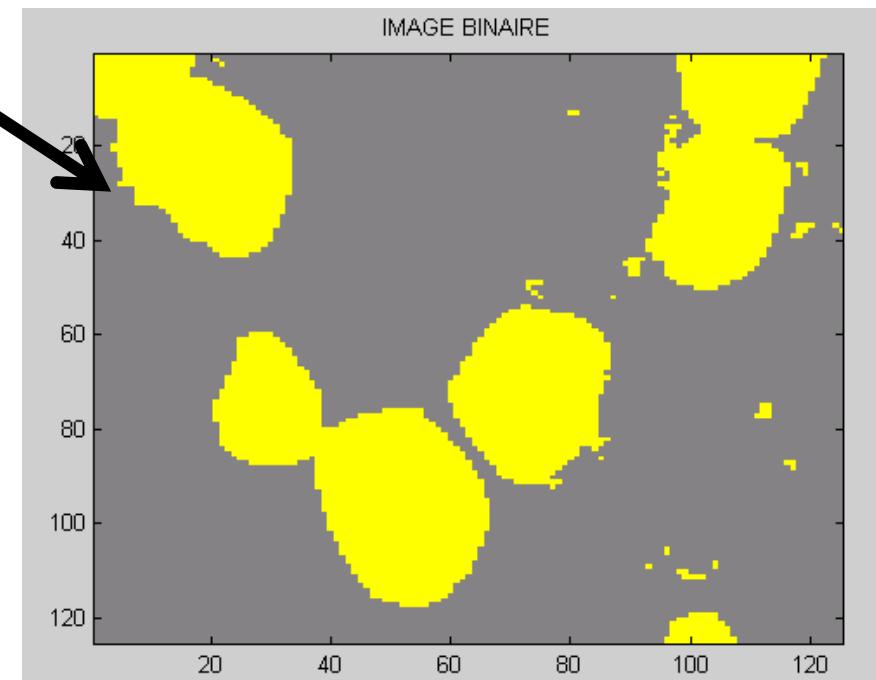


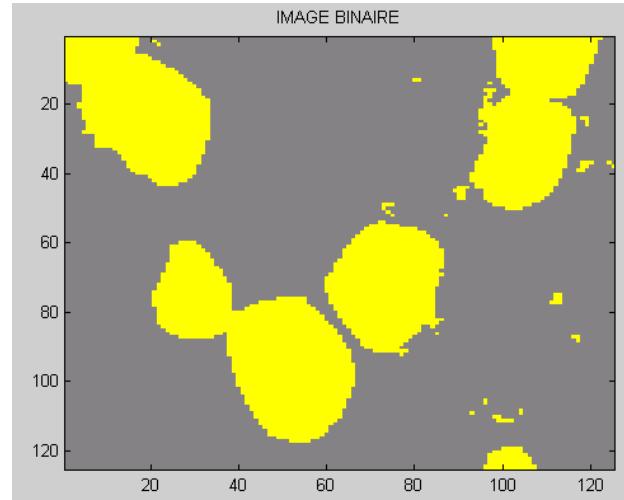
Image source

Binarisation

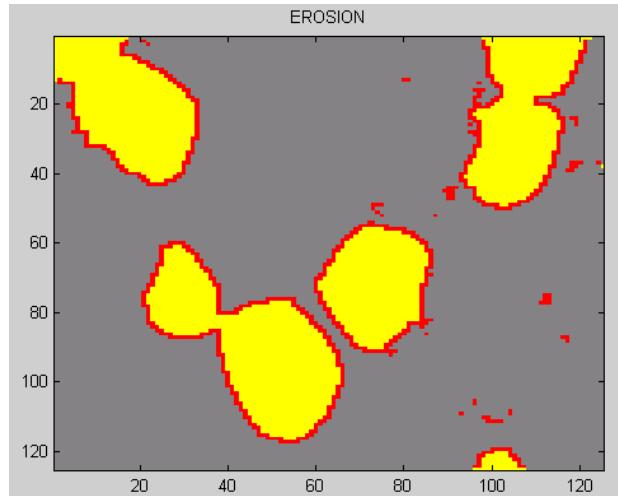


# Exemple

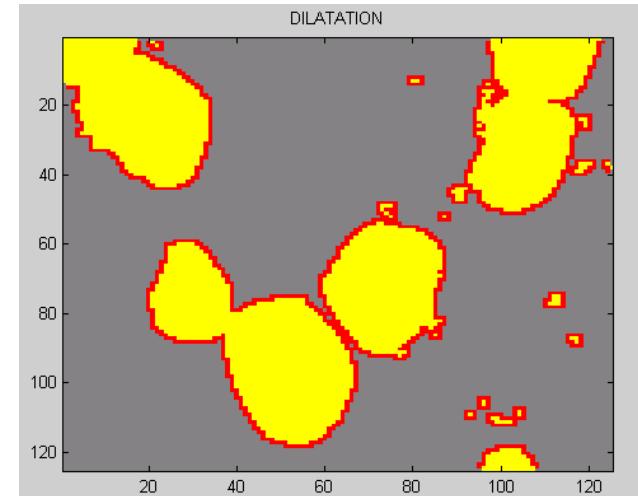
Image source



Erosion

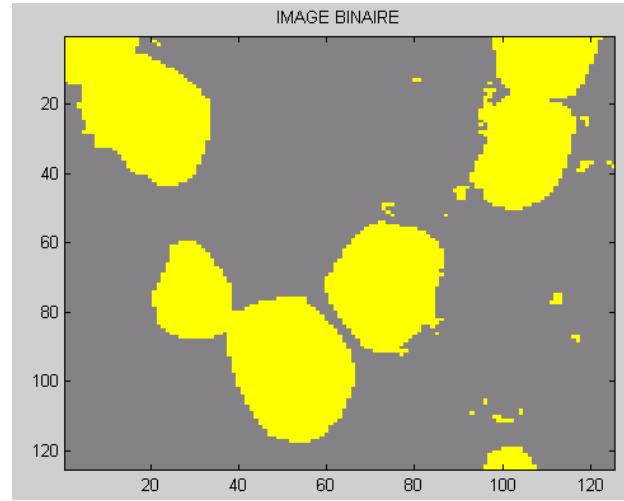


Dilatation

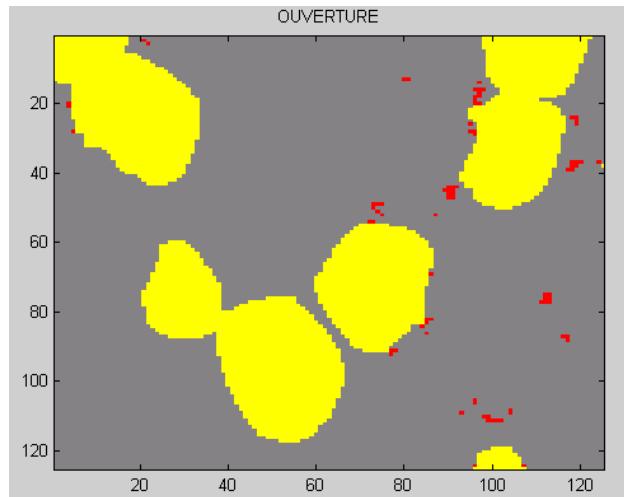


# Exemple

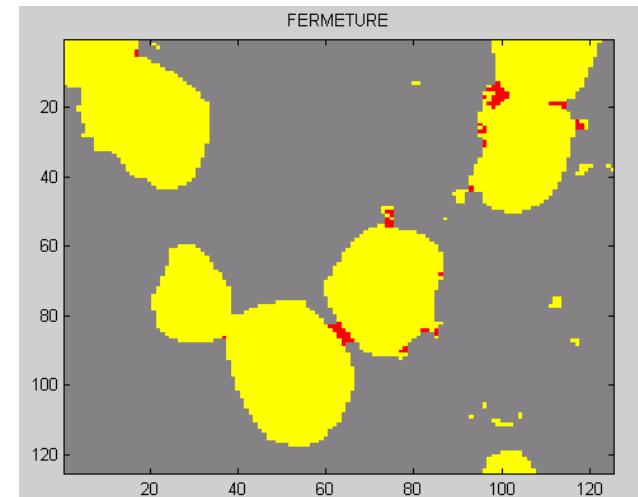
Image source



Ouverture  
(érosion puis dilatation)



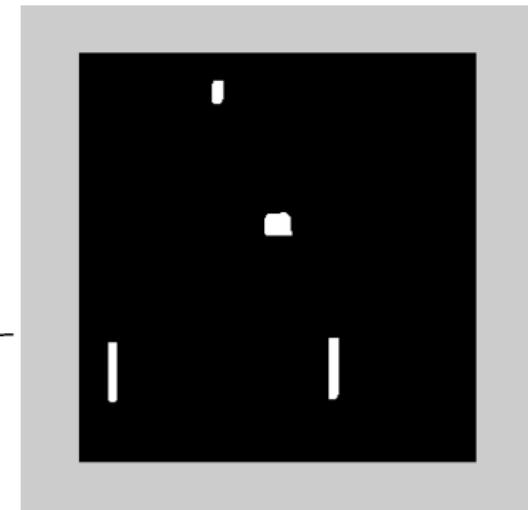
Fermeture  
(dilatation puis érosion)



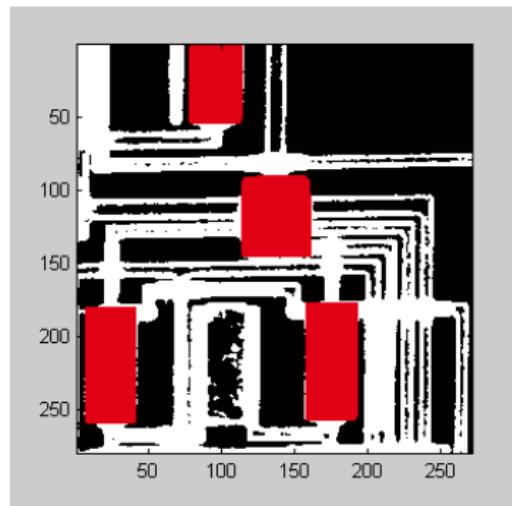
## Exemple : détection des composants



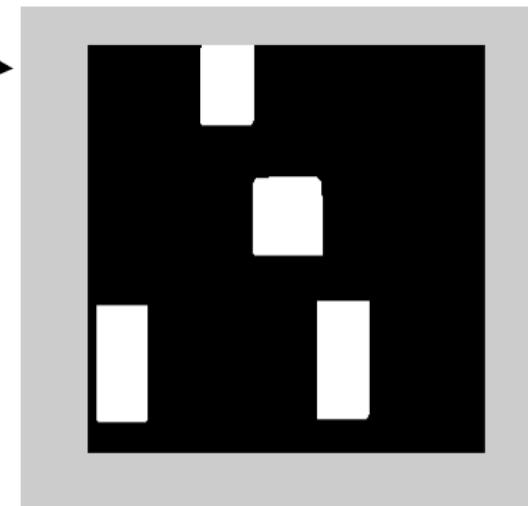
Érosion de l'image  
L'élément structurant  
est de taille 40x30



Dilatation avec le  
même élément  
structurant

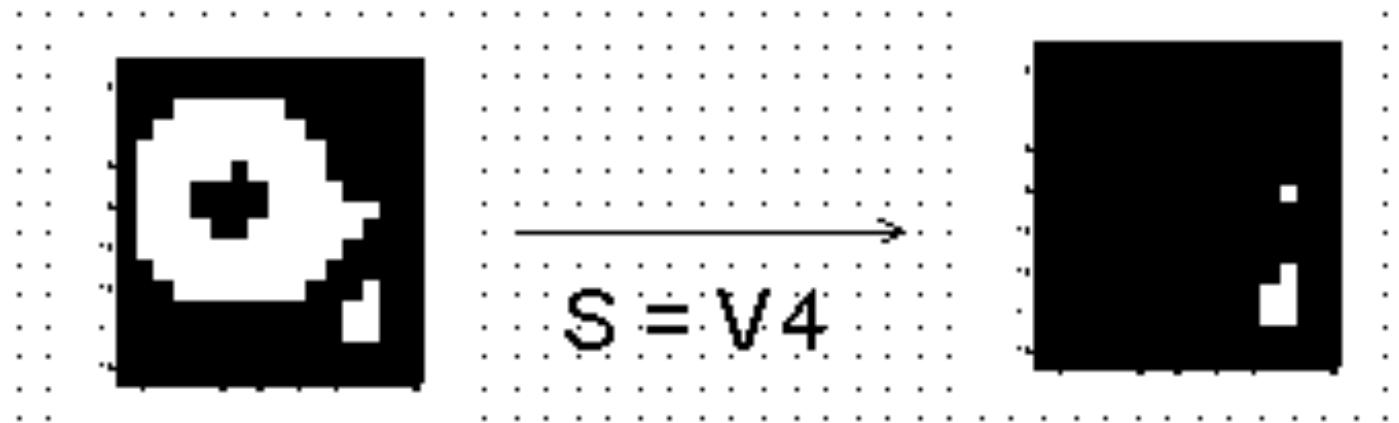


Report sur l'image initiale



## Sélection d'objets de petite taille face à S

$$X - (X \circ S)$$

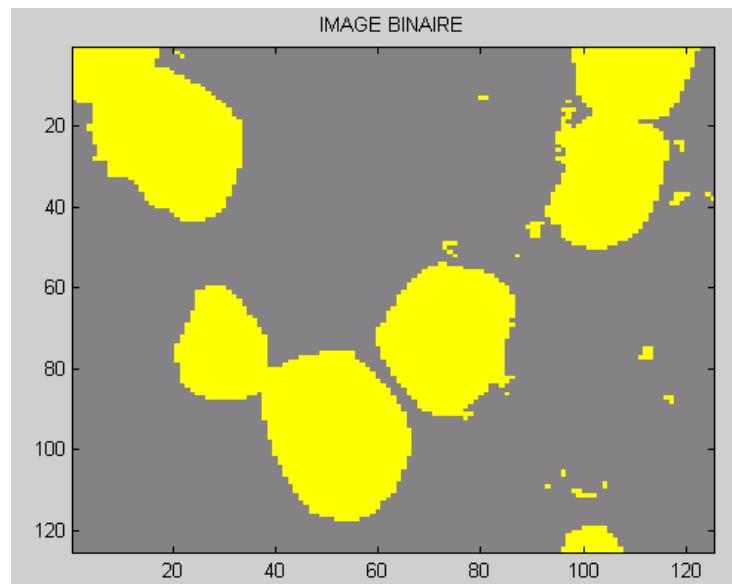


# Contours intérieurs ou extérieurs

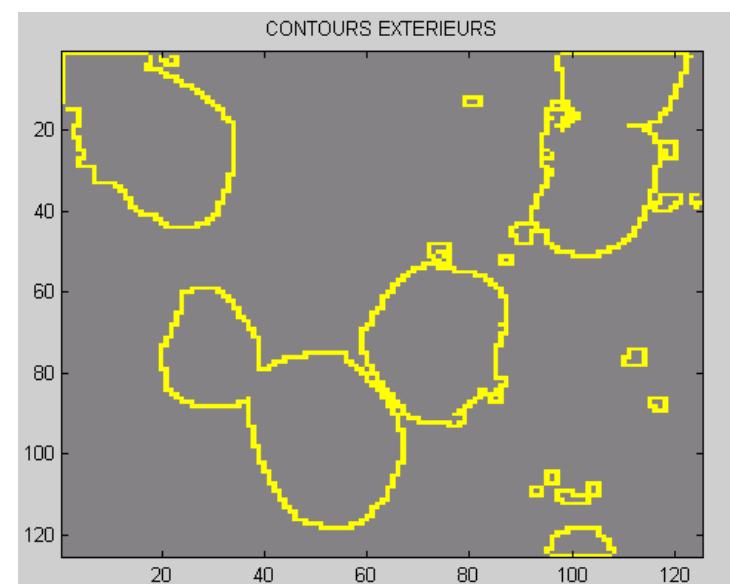
$X - (X \ominus S) \rightarrow$  Contour intérieur

$(X \oplus S) - X \rightarrow$  Contour extérieur

Image binaire



Contours extérieurs



## Amincissement et épaissement

---

**Amincissement** : Erosion avec un disque de rayon 1 et conservation de la connexité.

**Epaississement** : Dilatation dans les mêmes conditions.

## Amincissement

L'élimination du pixel central ne modifie pas la connexité locale.

Point bordure « nord » :

V8

$0$	$0$	$0$
$1$	$1$	$1$
$x$	$1$	$1$

$x$	$0$	$0$
$1$	$1$	$0$
$x$	$1$	$x$

$x$  : état indifférent 0 ou 1

Pour tous les pixels à 1 de bordure nord vérifiant une des configurations précédentes, on pourra mettre le pixel central à 0.

Les autres configurations correspondant aux bordures E, W, S s'en déduisent par rotation de 90°.

**Attention** : Les bandes de largeur 2 pixels 0 1 1 0 sont détruites du fait du parallélisme du traitement → les détecter et ne pas modifier le point central pour et idem en vertical.

# Amincissement

Application : amélioration de contours d'objets minces (caractères, ...)



Gris : image source

Blanc : 1 puis 2 amincissements

# Epaississement

---

La mise à 1 du point central, point de bordure « nord » étend la connexité locale.

$x$	$x$	1
$x$	0	1
$x$	1	1

1	$x$	$x$
1	0	$x$
1	1	$x$

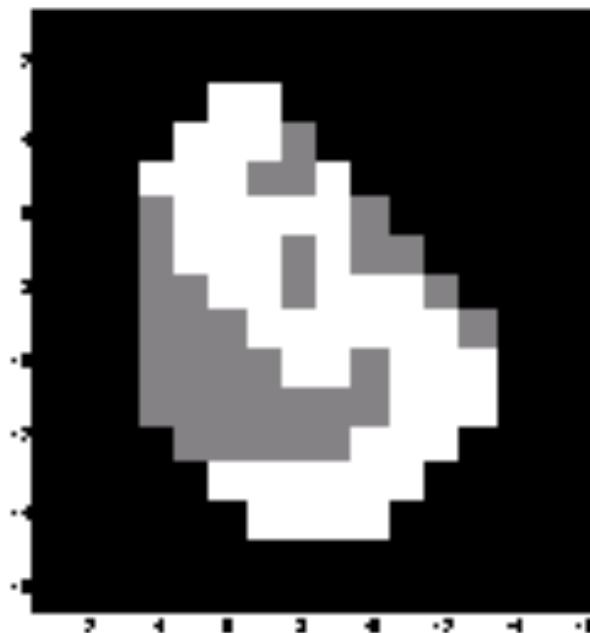
$x$	$x$	$x$
1	0	1
$x$	1	$x$

E, W, S s'en déduisent par rotation de 90°.

Attention la connexité globale n'est pas conservée.

# Epaississement

**Application :** Détermination itérative (jusqu'à stabilisation) de l'enveloppe convexe d'un objet isolé.



Blanc : image source

Gris : enveloppe convexe

# Morphologie mathématique

---

## Binarisation d'images

- Choix manuel du seuil
- Minimisation de la variance intra-classe : méthode d'Otsu
- Seuillage entropique
- Minimisation de l'erreur de classification bayesienne

## Notion de morphologie mathématique

## Squelettisation

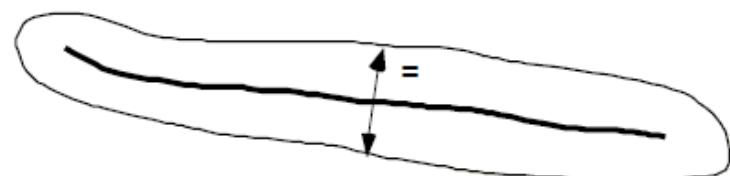
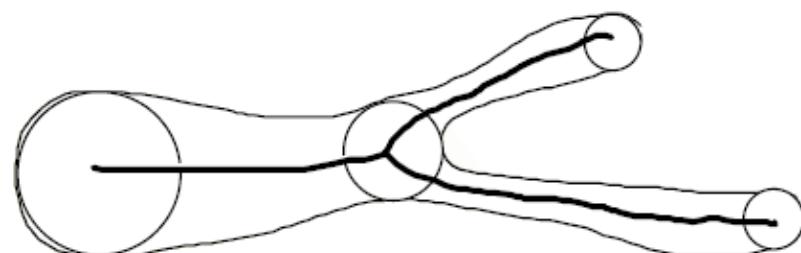
# Squelettisation

Graphe caractéristique de la forme d'un objet binaire qui peut permettre de le reconstituer (transformation réversible).

Deux définitions similaires dans  $\mathbb{R}^2$  :

→ Lieu des centres des disques de rayon minimum inscrits dans l'objet. Le rayon en chaque point permet de reconstruire l'objet.

→ Axe médian, ensemble des points équidistants de 2 bords de l'objet. Cette définition correspond à l'état stable obtenu par amincissements itératifs



# Squelettisation

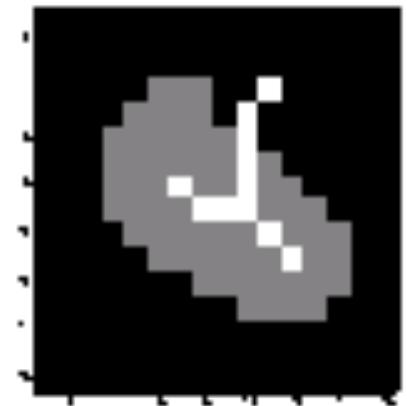
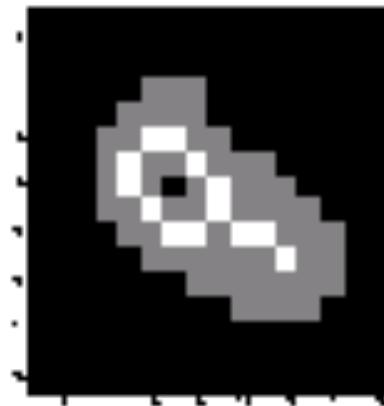
**Propriétés :**

Invariance présumée en rotation.

Sensibilité aux détails (irrégularité des bords, lacunes) → pré-traitement

**Exemples :**

Gris : image source, blanc : squelette



# Squelettisation

**Pré-traitements et lissage des irrégularités :**

Elimination des lacunes en 4 connexité :

$X$	$X$	$X$
1	0	1
$X$	1	$X$

Lissage des points saillants :

0	0	0
0	1	0
$X$	$X$	0

0	0	0
0	1	0
$X$	1	$X$

0	0	0
0	1	0
0	$X$	$X$

# Squelettisation

Itérations d'amincissement jusqu'à stabilisation

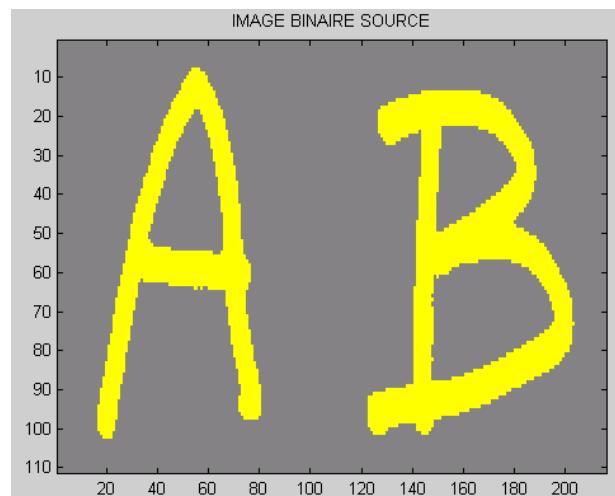
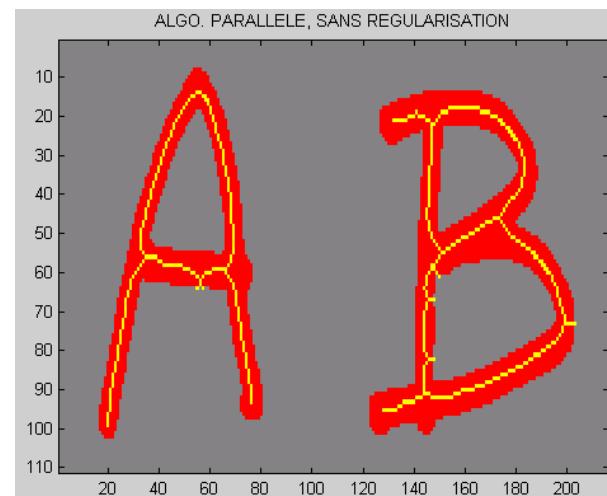
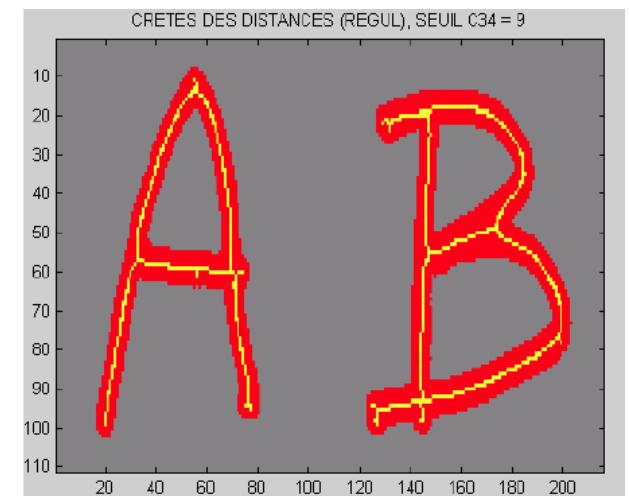


Image originale

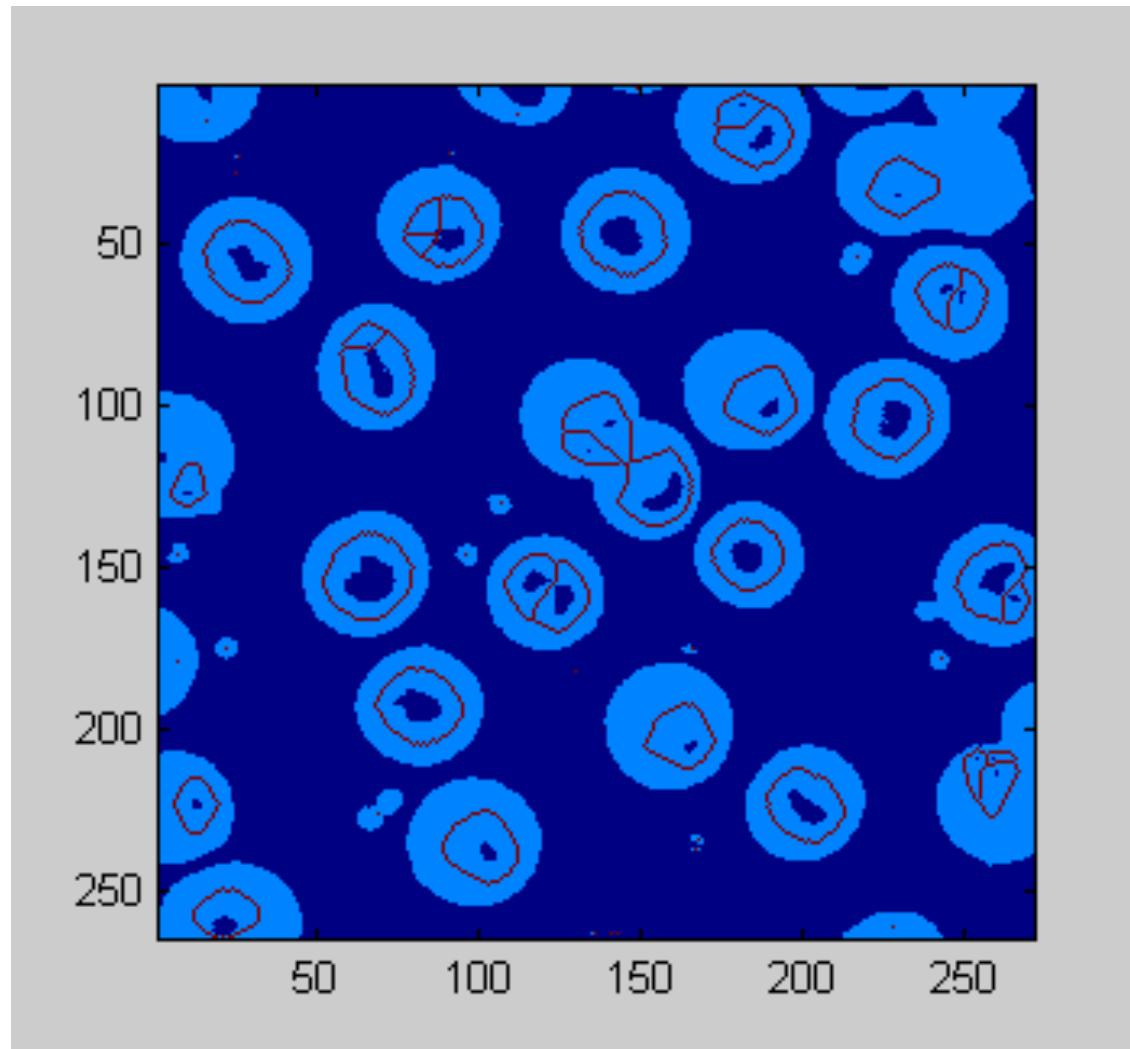


Sans régularisation



Avec régularisation

## Squelettisation



Chaque composante sans trou est réduite à 1 pixel

Les trous sont entourés d'une courbe

# Squelettisation par maxima locaux des distances au fond

## Principe

- Calcul des distances des points objet au fond (chamfer).
- Seuillage éventuel pour éliminer des irrégularités locales.
- Recherche des maxima locaux directionnels (lignes de crêtes).

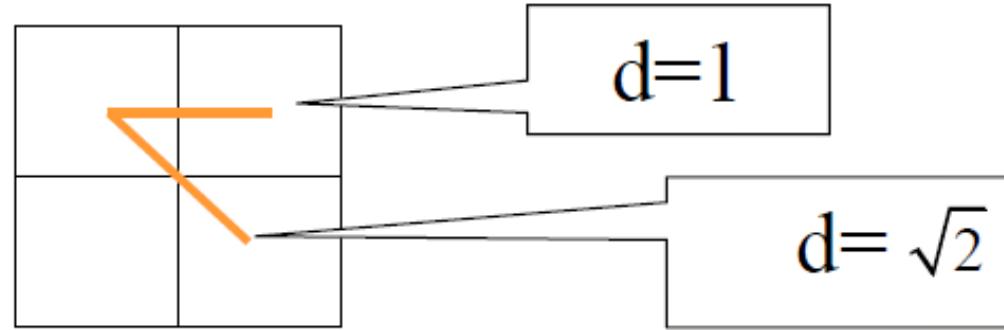
0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0
0	0	0	1	2	1	0	0
0	1	1	1	2	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

Maxima locaux directionnels  
→ Squelette non connexe.

## Avantages

Temps de calcul indépendant de l'épaisseur des objets (2 passes pour le calcul des distances + 1 pour rechercher les maxima directionnels

# Squelettisation par maxima locaux des distances au fond



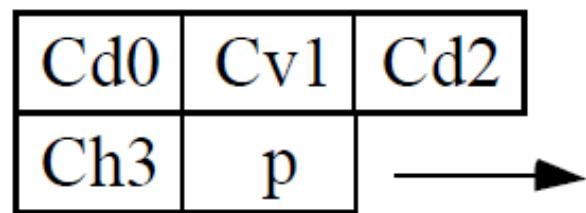
- Deux balayages : sens vidéo puis inverse vidéo.
- Cumul des distances de proche en proche.
- Pour simplifier, on ramène à des valeurs entières :

$$\begin{cases} 1 \rightarrow 2 \\ \sqrt{2} \rightarrow 3 \end{cases} \rightarrow 2 \text{ fois la distance euclidienne} \rightarrow \text{erreur de 13\% par défaut}$$

$$\begin{cases} 1 \rightarrow 3 \\ \sqrt{2} \rightarrow 4 \end{cases} \rightarrow 3 \text{ fois la distance euclidienne} \rightarrow \text{erreur de 8\% par excès}$$

# Algorithme de Chamfer

Balayage en sens vidéo :

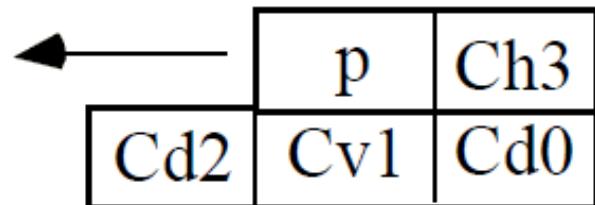


$$D_p = \min(D(i) + C(i))$$

$$i=0..3$$

Avec  $C_v = C_h = 2$  ou  $3$   
 $C_d = 3$  ou  $4$  respectivement

Balayage en inverse vidéo :



$$D_p = \min(D_p, D(i) + C(i))$$

$$i=0..3$$

Remarque : Si  $C_h = C_v = C_d = 1 \rightarrow D_{inf}$

Si  $C_h = C_v = 1$  et  $C_d = inf \rightarrow D_1$

# Le mouvement

# Morphologie mathématique

---

## Application

Détection par différences d'images successives

Détection avec une image de référence

Filtrage temporel

Modélisation gaussienne

Modélisation par KDE

Post-traitement

## Application

---

La détection du mouvement est utilisée dans de nombreuses applications :

**La télésurveillance** : trafic routier, intrusion dans une zone, analyse du comportement de personnes, surveillance à distance de personnes âgées, étude du geste sportif,...

**La robotique** : détection d'obstacle, analyse du mouvement d'un robot, suivi de cibles,...

**La compression** : détecter les zones immobiles dans un séquence pour les coder qu'une seule fois,...

**Imagerie bio-médicale** : suivre des amers dans le corps humain,...

---

# Détection des zones en mouvement



# Morphologie mathématique

---

Application

Détection par différences d'images successives

Détection avec une image de référence

Filtrage temporel

Modélisation gaussienne

Modélisation par KDE

Post-traitement

# Détection par différences d'images successives

---

Comment détecter les zones en mouvement dans une scène ?

Par différence d'images entre deux images successives :

$$D_t(\mathbf{x}) = \begin{cases} 0 & \text{si } |I_t(\mathbf{x}) - I_{t-1}(\mathbf{x})| < \text{seuil} \\ 1 & \text{sinon} \end{cases} \quad \text{avec } \mathbf{x} = [x, y]$$

Où  $I_t(\mathbf{x})$  est l'image à l'instant  $t$

## Détection par différences d'images successives



Temps t



Temps t+1

## Détection par différences d'images successives



Image de différence



Image de différence seuillée

Problèmes :

- Si le mouvement est de faible amplitude, on détecte seulement les contours de l'objet en mouvement
- Si le mouvement a une forte amplitude, on détecte deux fois la forme : là où elle a disparue et là où elle est apparue

# Morphologie mathématique

---

Application

Détection par différences d'images successives

Détection avec une image de référence

Filtrage temporel

Modélisation gaussienne

Modélisation par KDE

Post-traitement

## Détection avec une image de référence



$$D_t(\mathbf{x}) = \begin{cases} 0 & \text{si } |I_t(\mathbf{x}) - F(\mathbf{x})| < \text{seuil} \\ 1 & \text{sinon} \end{cases} \quad \text{avec } \mathbf{x} = [x, y]$$

Où  $F(x)$  est l'image de référence

## Détection avec une image de référence



Image de différence



Image de différence seuillée

- Meilleurs résultats
- Problème lorsque l'objet a une couleur similaire au fond
- Problème lorsque le fond évolue

# Morphologie mathématique

---

Application

Détection par différences d'images successives

Détection avec une image de référence

Filtrage temporel

Modélisation gaussienne

Modélisation par KDE

Post-traitement

## Filtrage temporel médian

---

L'image de référence  $F(x)$  est composée de la médiane des  $N$  dernières images

Problème : nécessite de stocker les  $N$  dernières images en mémoire

## Filtrage temporel

---

L'image de référence est initialisée avec la première image

Puis mise à jour de l'image de référence :

$$F_{t+1} = \alpha I_t(x) + (1 - \alpha) F_t(x)$$

Où  $\alpha$  est le paramètre d'apprentissage (généralement entre 0.01 et 0.1)

Problème : quelle valeur de  $\alpha$  choisir ? Quel seuil choisir pour la détection ?

# Morphologie mathématique

---

Application

Détection par différences d'images successives

Détection avec une image de référence

Filtrage temporel

Modélisation gaussienne

Modélisation par KDE

Post-traitement

## Modélisation gaussienne

---

Chaque pixel est modélisé par une gaussienne (  $\mu_t(x)$ ,  $\sigma_t(x)$  )

Mise à jour des paramètre par :

$$\mu_t(x) = \alpha I_t(\mathbf{x}) + (1 - \alpha) \mu_{t-1}(\mathbf{x})$$
$$\sigma_t(x) = \sqrt{\alpha [I_t(\mathbf{x}) - \mu_t(x)]^2 + (1 - \alpha) \sigma_{t-1}^2(\mathbf{x})}$$

Un pixel est détecté comme objet si

$$|I_t(\mathbf{x}) - \mu_t(x)| > k\sigma_t$$

$k$  est un paramètre (typiquement 2.5)

# Modélisation gaussienne

---

## Avantages

Peu de place en mémoire ( $\mu_t(x)$ ,  $\sigma_t(x)$ )

Seuil adaptatif en fonction des pixels

## Inconvénient

Hypothèse gaussienne valide ?

# Modélisation par mixture de gaussiennes

---

## Idée

Chaque pixel est modélisée par une densité de probabilité gaussiennes (de 3 à 5)

Intérêt pour une distribution des ndg multimodale (feuille d'un arbre)

## Objectif

Déterminer si un pixel appartient à l'arrière plan ou l'avant plan

Méthode de Stauffer et Grimson (1998)

# Modélisation par mixture de gaussiennes

---

La probabilité d'observer la valeur  $I_t$  d'un pixel est :

$$P(I_t) = \sum_{i=1}^K \omega_{i,t} * \eta(I_t, \mu_{i,t}, \Sigma_{i,t})$$

Où :

- $\omega_{i,t}$  est le poids de la gaussiennes i à l'instant t (vraisemblance)
- K est le nombre de gaussiennes
- $\mu_{i,t}$  et  $\Sigma_{i,t}$  sont la moyenne et l'écart type de la gaussienne i
- $\eta()$  est la densité de probabilité gaussienne :

$$\eta(I_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2} |\Sigma_{i,t}|^{1/2}} e^{-\frac{1}{2}(I_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (I_t - \mu_{i,t})}$$

Souvent,  $\Sigma_{i,t} = \sigma_{i,t} \text{Id}$ , pour diminuer les temps de calcul (3 canaux couleur indépendants)

## Modélisation par mixture de gaussiennes

---

Le pixel courant  $I_t$  appartient à une/plusieurs des gaussiennes ?  
(valeur comprise dans 2.5 fois l'écart type d'une des gaussienne)

Si oui les paramètres sont mis à jours :

$$\begin{aligned}\mu_t &= (1 - \rho)\mu_{t-1} + \rho X_t \\ \sigma_t^2 &= (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \\ \rho &= \alpha\eta(X_t, \mu, \sigma)\end{aligned}$$

Sinon la gaussienne la moins probable (poids le plus faible) est supprimée et remplacée par la valeur courante du pixel, associé à un faible poids et une grande variance.

## Modélisation par mixture de gaussiennes

---

Les poids de toutes les gaussiennes sont mis à jour avec :

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha M_{i,t}$$

Où  $\alpha$  est le coefficient d'apprentissage

$M_{i,t}$  = 1 si le pixel appartient à la gaussienne i, 0 sinon

Les poids sont ensuite renormalisés pour que leur somme fasse 1

## Modélisation par mixture de gaussiennes

---

Pour la détection de mouvement, les gaussiennes sont rangées par ordre décroissant de  $\omega/\sigma$

Les premières  $B$  distributions sont considérées comme du fond :

$$B = \arg \min_b \left( \sum_{i=1}^b \omega_i > T \right)$$

Si le pixel appartient à une de ces gaussiennes, il est statique, sinon, il est détecté en mouvement

# Morphologie mathématique

---

Application

Détection par différences d'images successives

Détection avec une image de référence

Filtrage temporel

Modélisation gaussienne

Modélisation par KDE

Post-traitement

## Modélisation par KDE

---

Approximation du fond par un histogramme des valeurs récentes

→ Ne marche pas car pas assez de valeurs pour avoir une bonne estimation

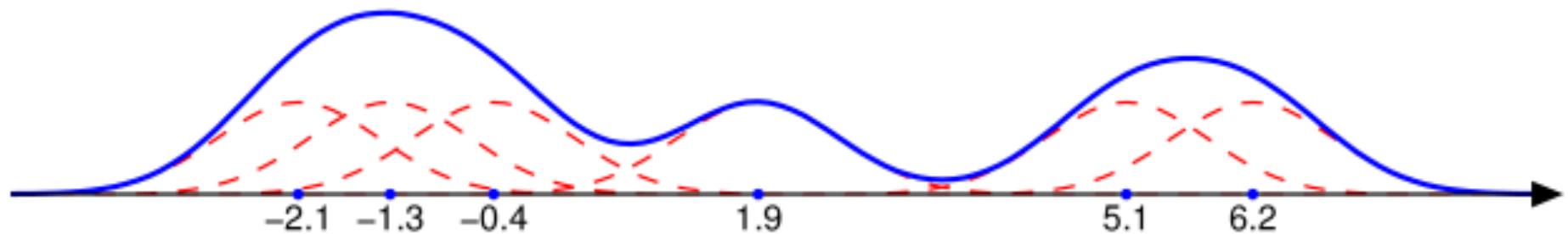
→ Modélisation non paramétrique en utilisant des fonctions noyaux. Soient  $x_1, x_2, \dots, x_N$ , les  $N$  dernières intensités lumineuses d'un pixel. La probabilité d'observer une nouvelle intensité  $x_t$  au temps  $t$  est donnée par:

$$P(x_t) = \frac{1}{N} \sum_{i=1}^N K_\sigma(x_t - x_i)$$

Où  $K_\sigma$  est la fonction noyau de largeur  $\sigma$

Elgammal, Harwood et Davis (2002)

## Modélisation par KDE



[http://en.wikipedia.org/wiki/Kernel\\_density\\_estimation](http://en.wikipedia.org/wiki/Kernel_density_estimation)

## Modélisation par KDE

---

Cette estimation s'étend facilement à la couleur :

$$P(x_t) = \frac{1}{N} \sum_{i=1}^N \prod_{j=1}^d K_{\sigma_j}(x_{tj} - x_{ij})$$

Si on choisit un noyau gaussien, la densité s'estime par :

$$P(x_t) = \frac{1}{N} \sum_{i=1}^N \prod_{j=1}^d \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_{tj}-x_{ij})^2}{\sigma_j^2}}$$

En utilisant cette estimation de la densité de probabilité, le pixel est considéré comme objet si :

$$P(x_t) < th$$

# Morphologie mathématique

---

Application

Détection par différences d'images successives

Détection avec une image de référence

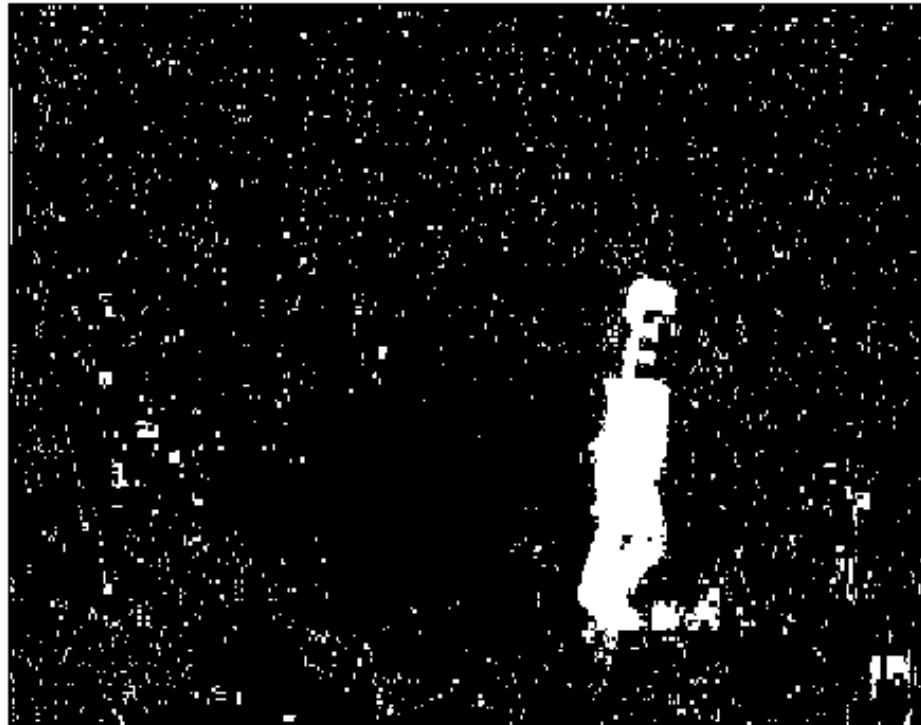
Filtrage temporel

Modélisation gaussienne

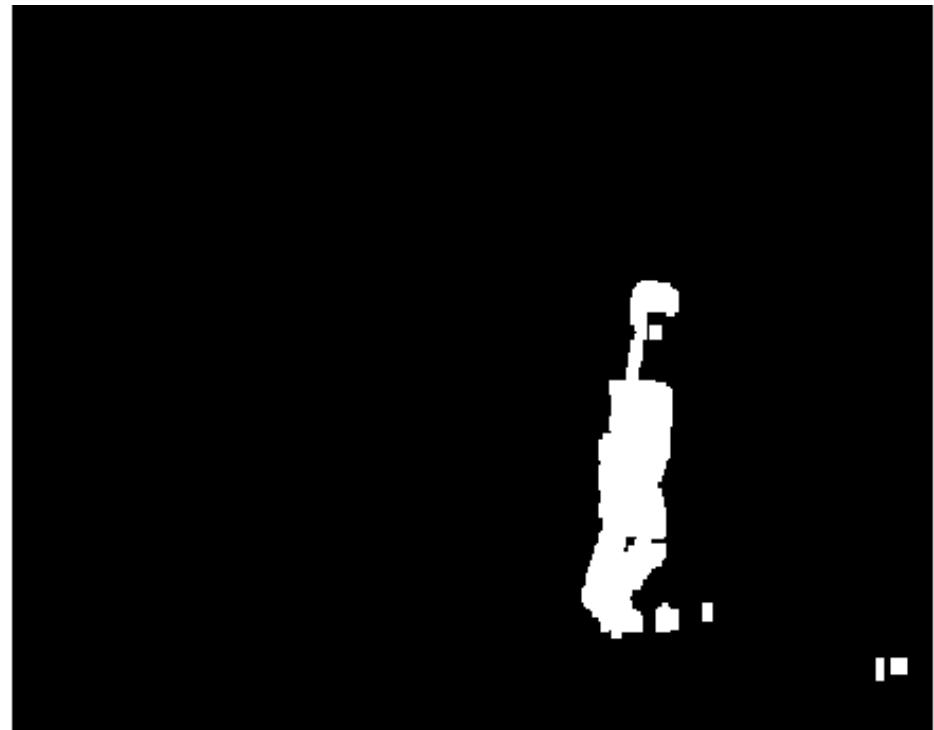
Modélisation par KDE

**Post-traitement**

## Post-traitements



Opérations de morphologie mathématiques pour diminuer le bruit



## Problème des ombres

