

Traitement d'image

Diapositives du cours

Christophe Ducottet

Telecom Saint-Etienne
25, rue Docteur Rémy Annino
42000 Saint-Etienne
Email : ducottet@univ-st-etienne.fr

version FI2-2020-09

Séance 1

Traitement d'image

- Cours 18h (12 séances), TD 12h (4 séances)
 - Evaluations
 - QCM de 10min portant sur le TD1 et le TD2 (20% x 2)
 - organisé au début du CM qui suit le TD c'est à dire au début du CM6 (pour le TD1) et du CM8 (pour le TD2)
 - Evaluation écrite finale à la fin (60%)
- Projet analyse d'image 12h (4 séances)

Objectifs du cours

- Etudier les principales étapes allant de l'image numérique à la mesure de paramètres
 - La chaîne de traitement d'image
 - Les post-traitements
 - La mesure de paramètres
 - La segmentation : différentes approches
 - Points d'intérêt et mise en correspondance
 - Exemples et illustrations

Plan

1. Introduction
2. Pré-traitements
3. Seuillage automatique
4. Mesure de paramètres
5. Segmentation par classification
6. Détection de contours
7. Points d'intérêts et mise en correspondance

Aujourd'hui

1. Introduction

1.1 Traitement d'image et segmentation

1.2 Modélisation et représentation d'une image

1.3 Approches utilisées segmentation

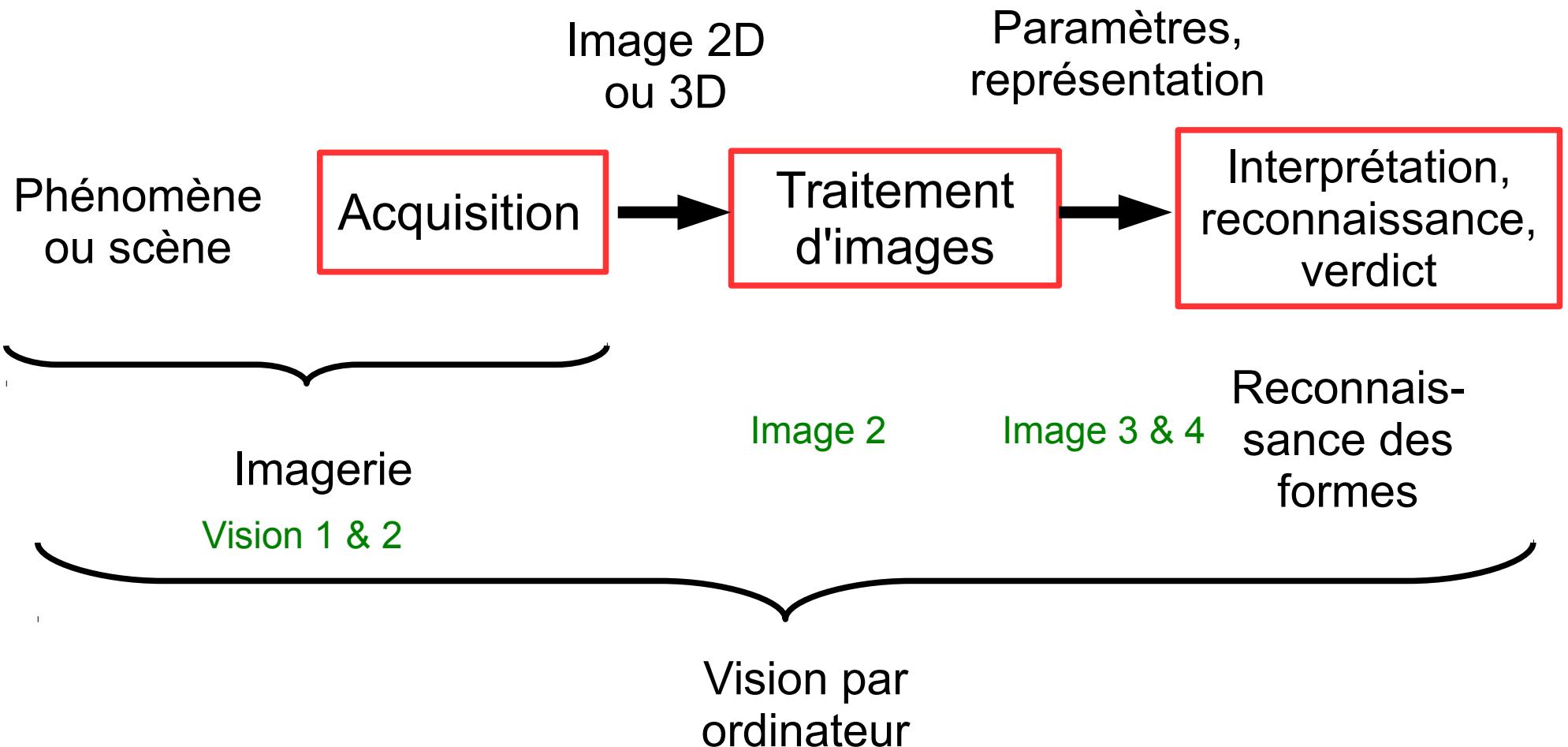
1.1 Traitement d'image et segmentation

- **Vocabulaire**
 - Vision par ordinateur (computer vision) ou vision artificielle (machine vision)
 - Ensemble des méthodes allant de l'acquisition jusqu'à l'interprétation du contenu, la reconnaissance d'éléments, le verdict
 - Traitement ou analyse d'image (image processing or analysis)
 - On part de l'image numérique jusqu'à la mesure de paramètres
 - Imagerie (imaging)
 - Concerne principalement la partie acquisition et reconstruction de l'image

1.1 Traitement d'image et segmentation

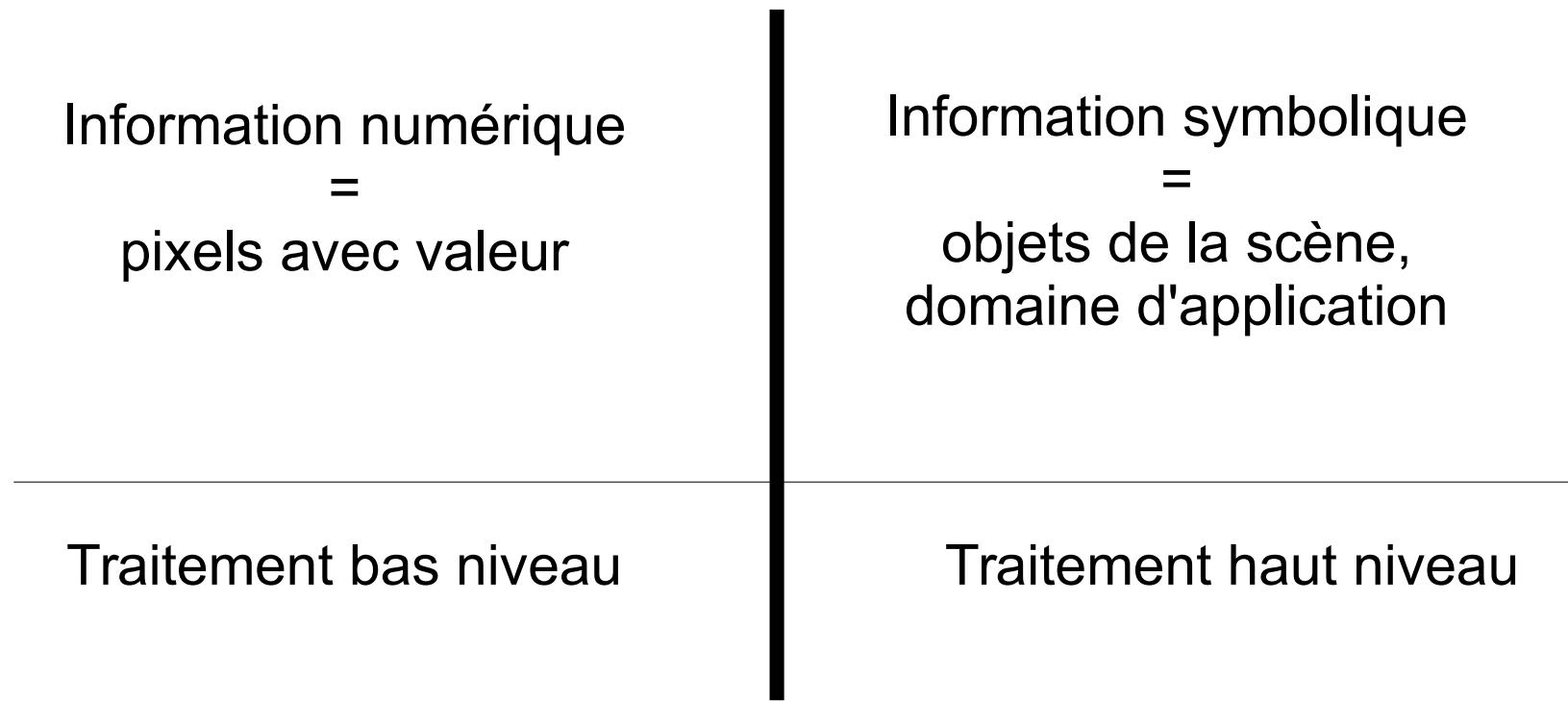
- **But de la vision par ordinateur**
 - Remplacer l'observateur humain en proposant un traitement numérique (automatique) pour :
 - Contrôle industriel (vision artificielle)
 - Surveillance, assistance aux personnes
 - Aide au diagnostic en imagerie médicale, satellitaire,...
 - Pilotage de véhicules ou de robots
 - Reconnaissance de caractère (OCR)
 - Recherche ou classification des contenus multimédias
 - ...

Système de vision par ordinateur



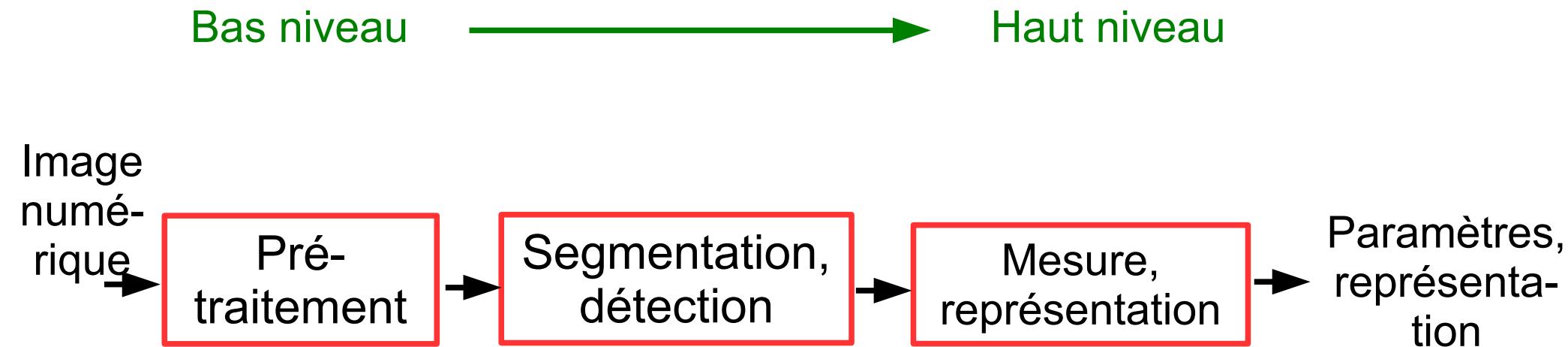
Vers l'interprétation

- Difficulté : passer des pixels à l'interprétation



Fossé sémantique

Chaîne de traitement d'image

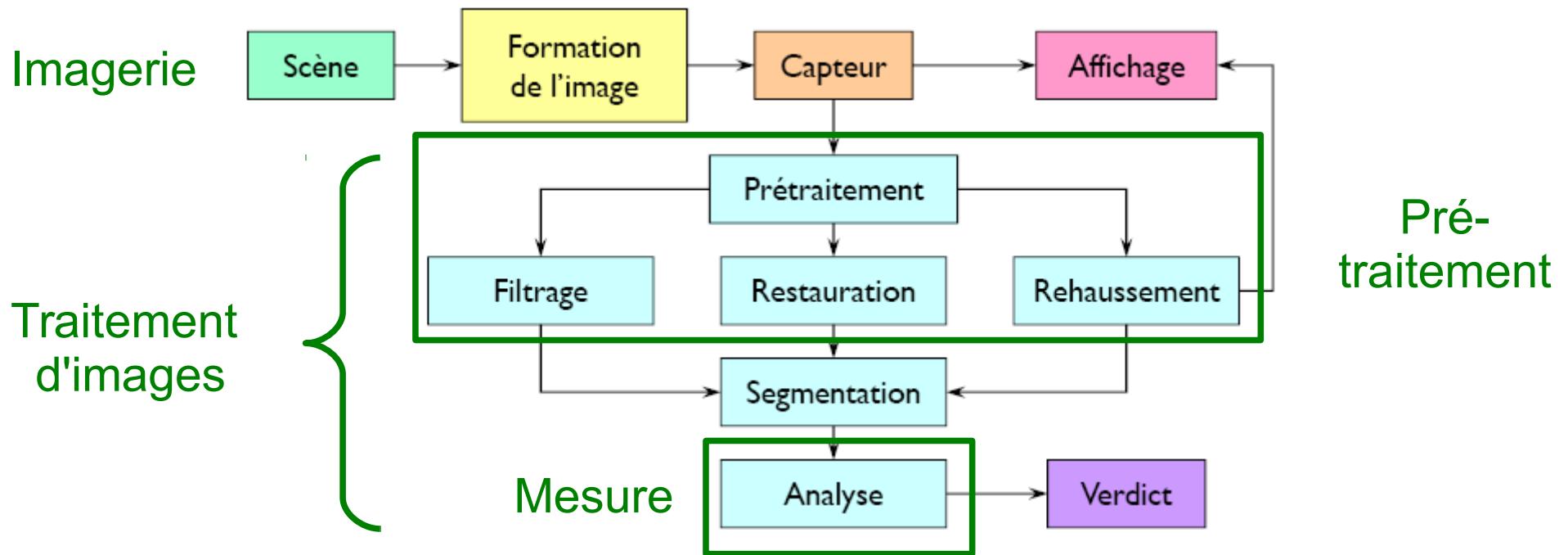


- Correction défauts
- Amélioration, réduction du bruit
- Réhaussement de contraste

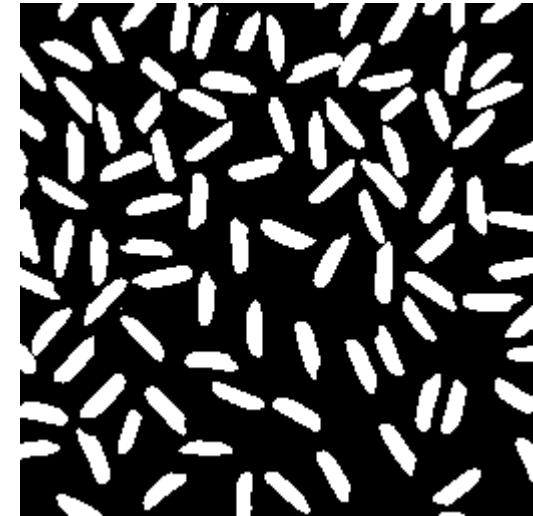
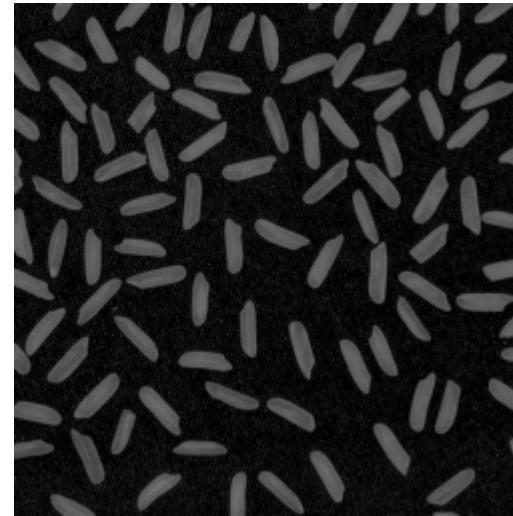
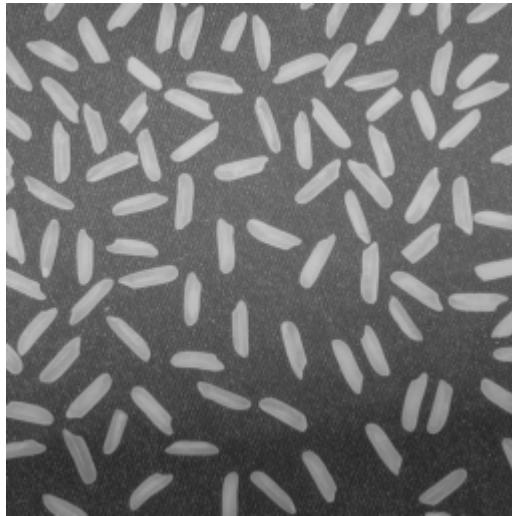
- Partition en régions
- Détection de zones d'intérêt

- Mesures sur les régions
- Extraction de paramètres
- Représentation simplifiée

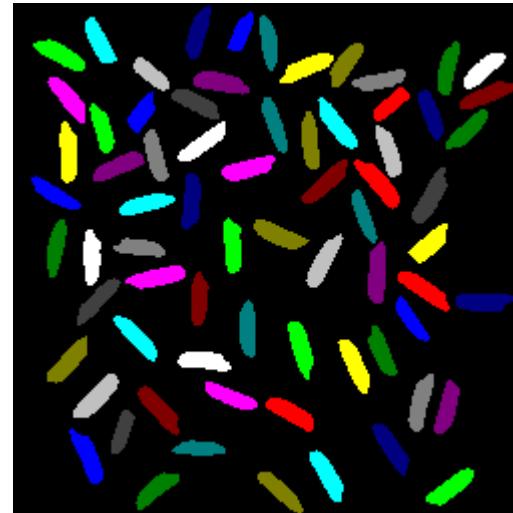
Lien avec le cours de 1ère année



Exemple de chaîne de traitement



Pré-traitement



Label	Aire	Perimetre
1	197	64,2843
2	140	51,4558
3	179	61,799
4	189	61,1127
5	208	64,7696

M
E
S
U
R
E

Post-traitements

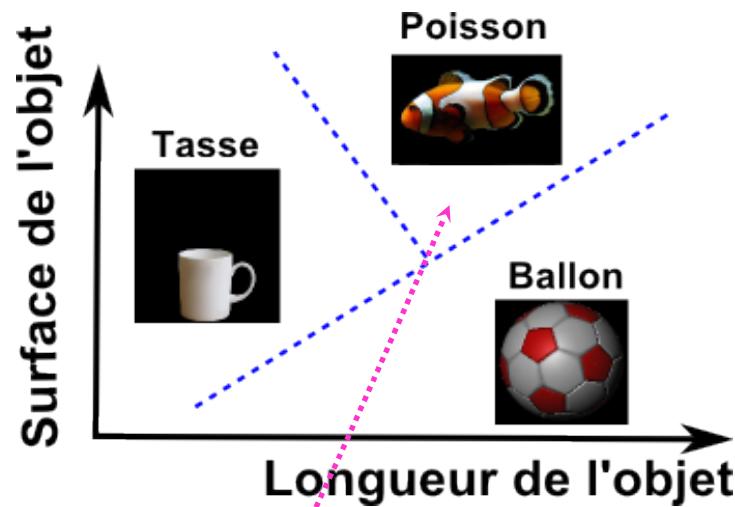
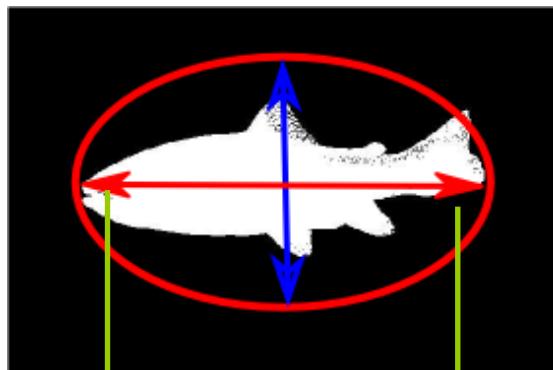
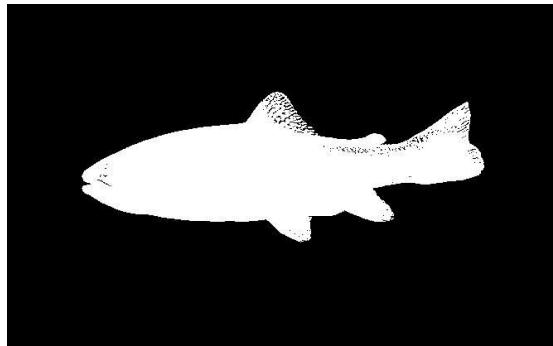
Etiquetage

Mesures

Après le traitement...

- Interprétation, reconnaissance, verdict
 - Tests sur les paramètres mesurés
 - Exemples :
 - si aire > seuil alors pièce non conforme
 - si nbre_objet < seuil alors ...
 - Méthodes de reconnaissance des formes
 - Classification, apprentissage automatique
 - Cf. cours 2ème semestre (FI-Image3)

Reconnaissance des formes



Segmentation : une étape cruciale

- **But**
 - Créer une partition de l'image en régions
 - Chaque région a une étiquette (classe)
 - Typiquement 2 étiquettes : fond et objet (image binaire)
 - Peut se limiter à détecter des éléments caractéristiques (points, contours, régions,...)
- **Difficulté**
 - Ne pas perdre l'information utile (irréversible)

Exemple d'image couleur

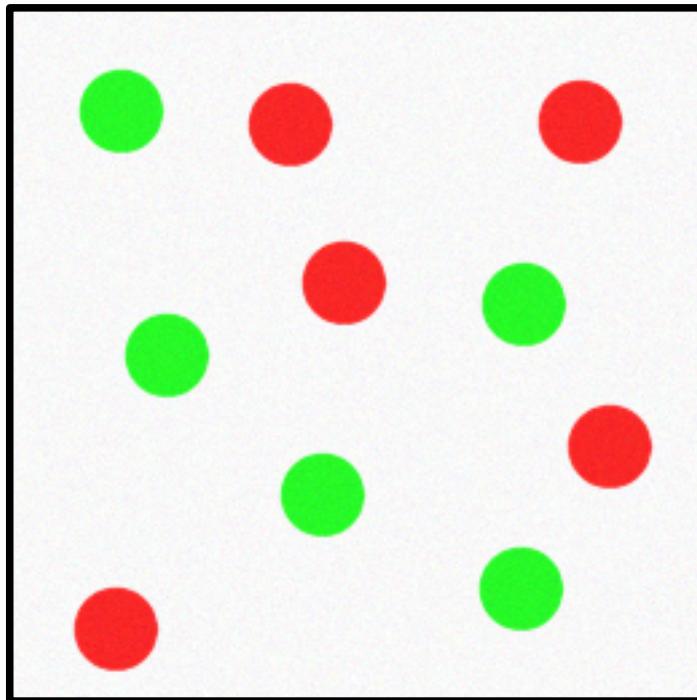


Image couleur originale

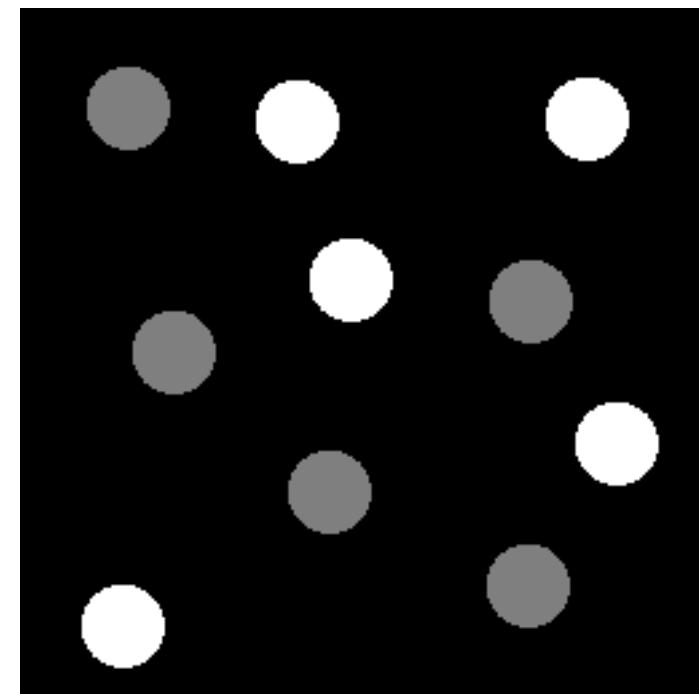
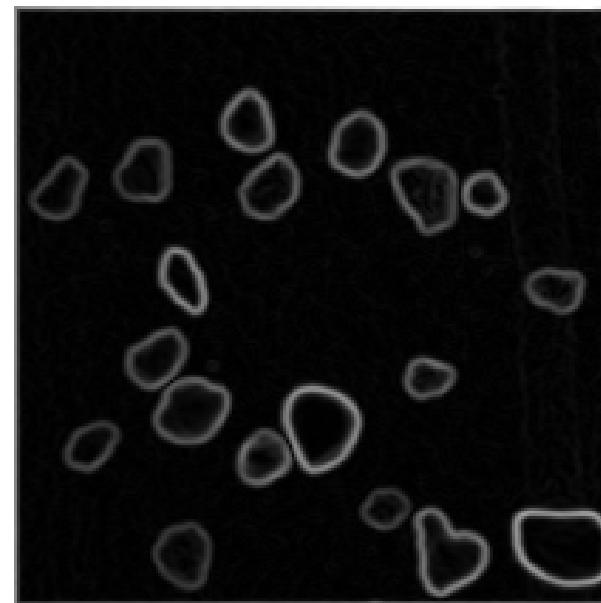
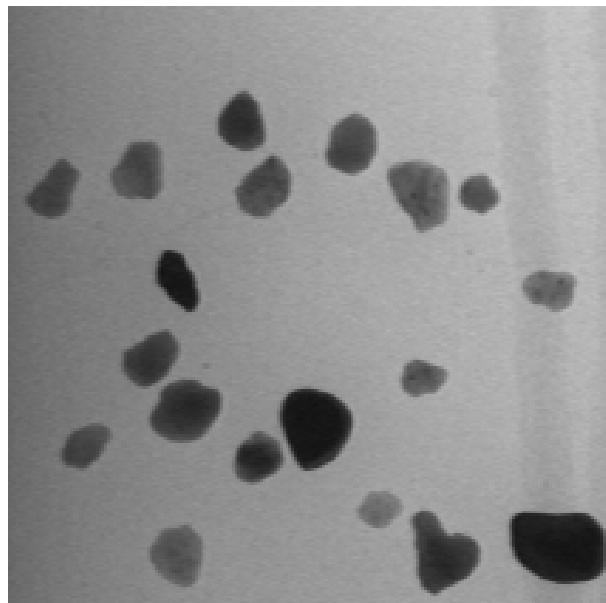
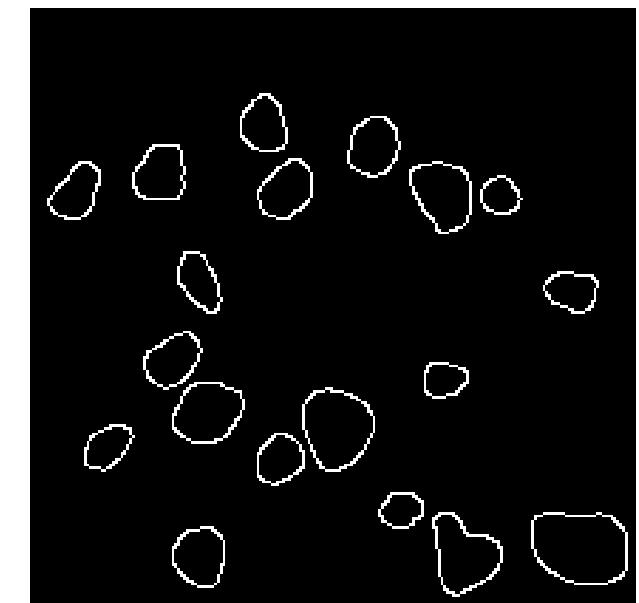


Image segmentée en 3 classes

Exemple de détection de contours



Module du gradient
= dérivée de l'image



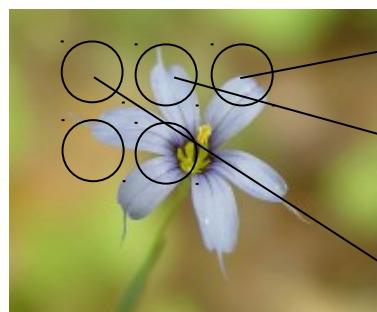
Seuillage des maxima
locaux du gradient

- **Les 2 approches de la segmentation**
 - Exploiter la notion de ressemblance entre pixels
 - Approche région
 - Regrouper des points dont les propriétés sont similaires
 - Exploiter la notion de différences entre pixels
 - Approche frontière
 - Recherche des points de variation rapides ou de discontinuité

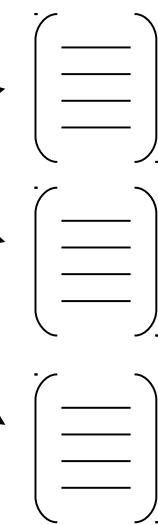
Mise en garde

- Pas de chaîne de traitement « universelle »
 - On doit s'adapter en fonction du problème
 - Certains éléments ne sont pas toujours nécessaires ni souhaitables (prétraitements)
- Evolution du domaine
 - L'étape de reconnaissance intervient de plus en plus pendant le traitement
 - La segmentation n'est pas toujours nécessaire

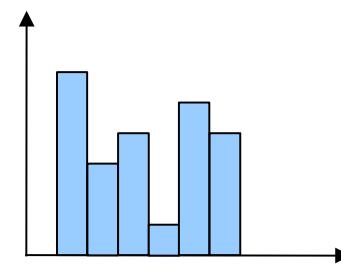
Approches basées sur des descriptions locales (patch)



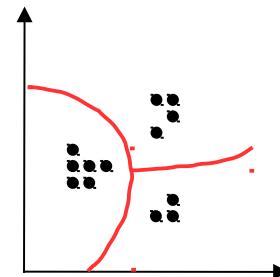
Sélection de points



Description locale



Représen-tation statistique



Apprentissage automatique

Classification d'image

- Challenge Pascal VOC 2007



aeroplane

cat



chair

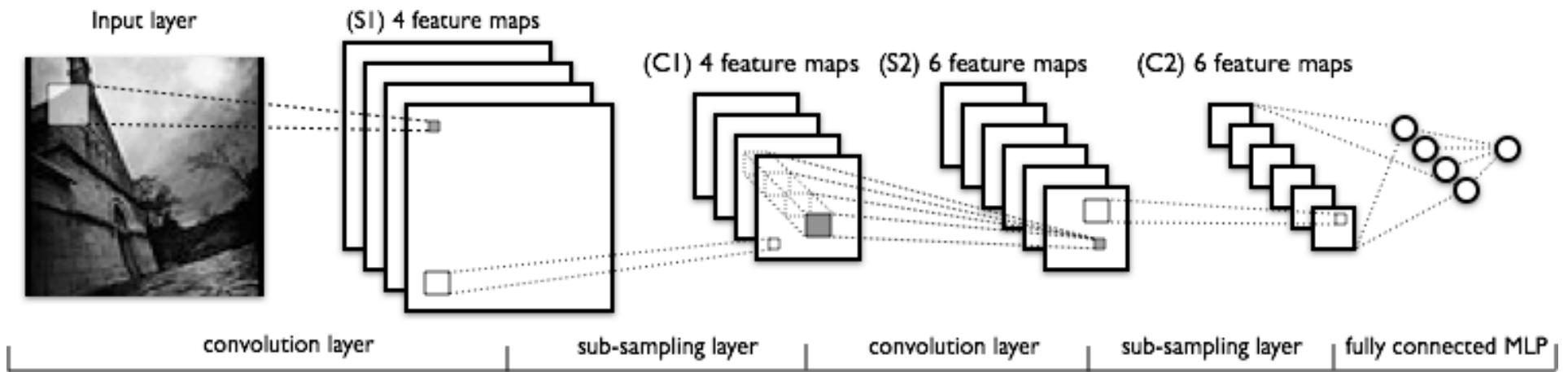
sheep



bicycle

potted plant

Réseaux de neurones convolutionnels (CNN)



Voir cours "Image et machine learning"
(Bloc Image4)

1.2 Modélisation et représentation

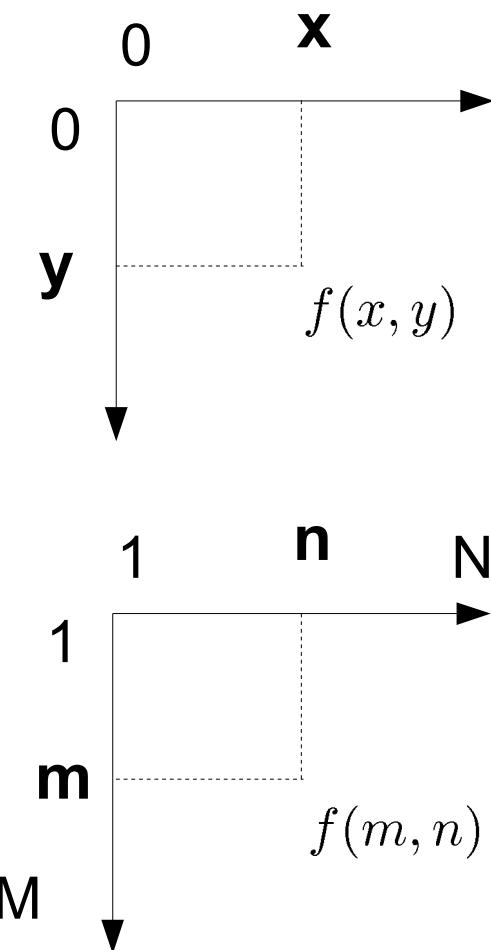
- Pour traiter une image numérique
 - On utilise des modèles de représentation de l'image
 - 3 modèles courants
 - Fonctionnel : Image = fonction
 - Probabiliste : Image = processus aléatoire
 - Morphologique : Image = surface topologique

Modèle fonctionnel

- Image = fonction (ou signal) 2D

- Image numérique échantillonnée

$f(m, n)$ = valeur numérique



Modèle fonctionnel

- **Image couleur**

$f(m, n)$ = vecteur de 3 composantes, (R, V, B) par exemple

- **Codage de l'information numérique**

- **Image numérisée par un capteur**

- Généralement 8 bits : $0 \rightarrow 255$
 - Peut être 10 ou 12 bits

- **Image issue d'un traitement**

- Codage numérique : int, float, double, ...

Modèle fonctionnel

- Exemples d'utilisation de ce modèle
 - Opérations arithmétiques sur les images
 - Traitement du signal : filtrage, transformée de Fourier
 - Calcul de dérivées (détection de contours)
 - Outils d'analyse numérique (résolution d'équations, minimisation,...)

Modèle probabiliste

- Image = processus aléatoire

$\forall (m, n) f(m, n)$ =variable aléatoire $X(m, n)$

$f(m, n)$ =réalisation d'un processus aléatoire

Exemple



Modèle probabiliste

- Moments statistiques (réalisations)

- 1er ordre : espérance mathématique

$$\mu_f(m, n) = E[f(m, n)]$$

- 2ème ordre : variance

$$\sigma_{mn}^2 = E[(f(m, n) - \mu_f(m, n))^2]$$

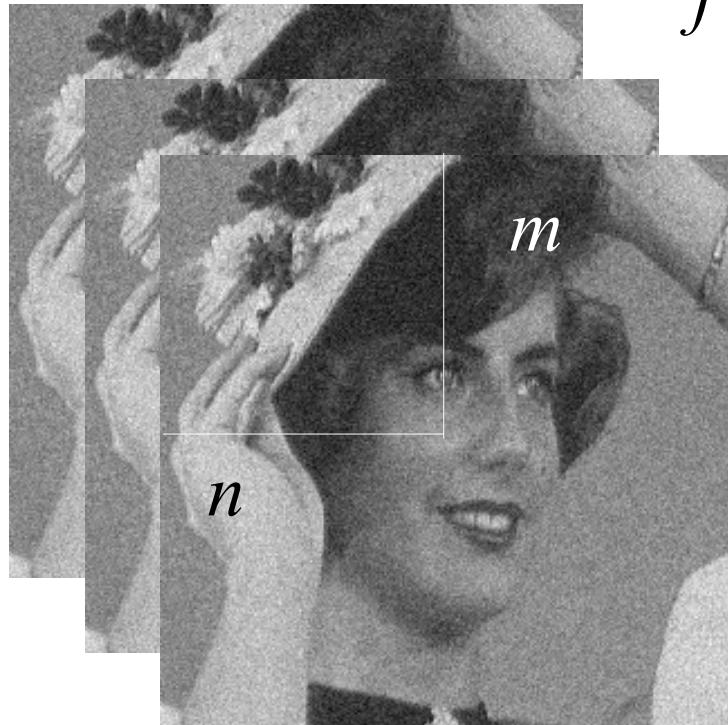
- Autocorrélation

$$\gamma(m, n, k, l) = E[f(m, n)f(k, l)]$$

Modèle probabiliste

- Signification de l'opérateur $E[]$

K réalisations de f



$$f_3 \\ f_2 \\ f_1$$

$$\mu_f(m, n) = \frac{1}{K} \sum_{k=1}^K f_k(m, n)$$

Modèle probabiliste

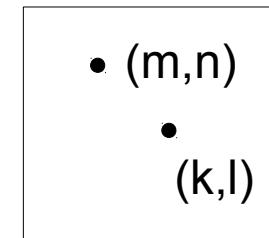
- Densité

- Densité du 1er ordre

$$h_{m,n}(z) = P(f(m, n) = z)$$

- Densité du 2ème ordre

$$h_{m,n,k,l}(z_1, z_2) = P(f(m, n) = z_1, f(k, l) = z_2)$$



Modèle probabiliste

- Moments d'espace et histogramme
 - Pas d'accès aux grandeurs statistiques
 - Une seule réalisation
 - Espérance => moyenne d'espace
 - Hypothèse de stationnarité et d'ergodicité

Modèle probabiliste

- Moments d'espace

$$\mu_f = \frac{1}{K} \sum_{(m,n) \in D} f(m,n) \quad \sigma_f = \frac{1}{K} \sum_{(m,n) \in D} (f(m,n) - \mu_f)^2$$

Nombre de pixels

$$\gamma(k,l) = \frac{1}{K} \sum_{(m,n) \in D} f(m,n) f(m+k, n+l)$$

Stationnarité « locale »

$$\mu_f(m,n) = \frac{1}{(2a+1)^2} \sum_{k=-a}^{k=a} \sum_{l=-a}^{l=a} f(n+k, m+l)$$

Modèle probabiliste

- Histogramme

Normalisé

$$h^*(z) = \frac{1}{K} \underset{(m,n) \in D}{\text{Freq}} (f(m,n) = z)$$

Analogue à la densité de probabilité

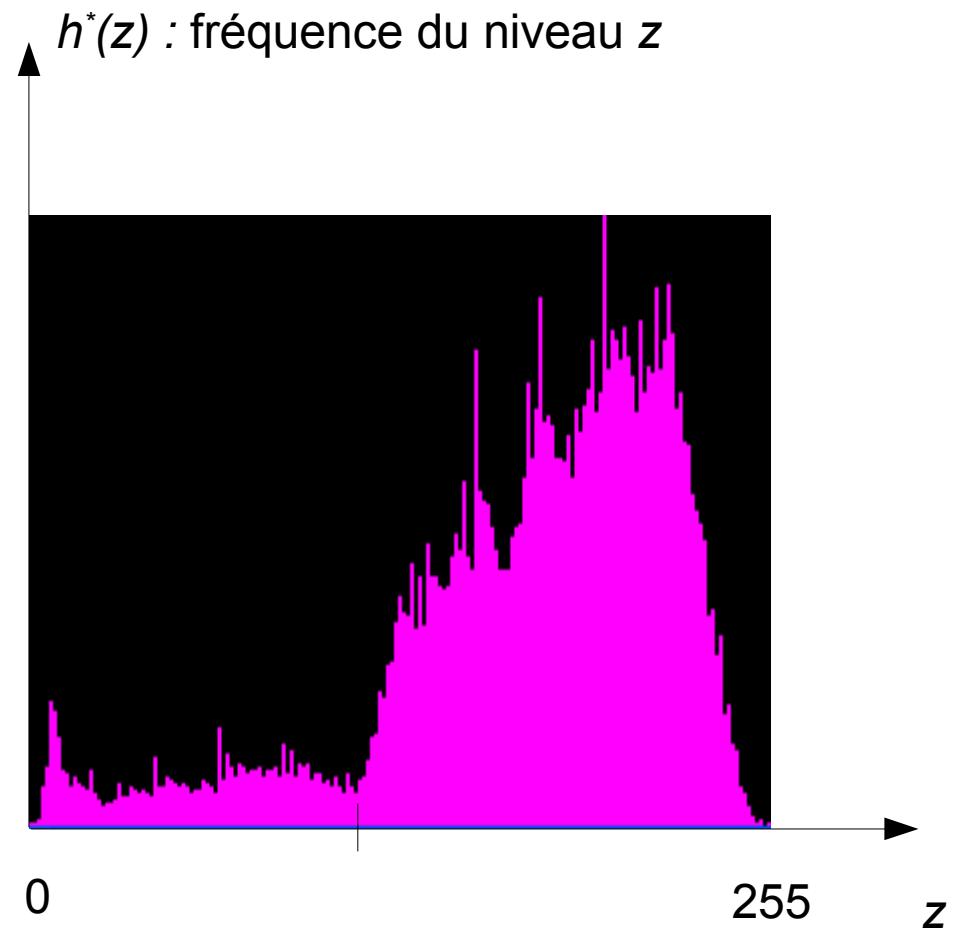
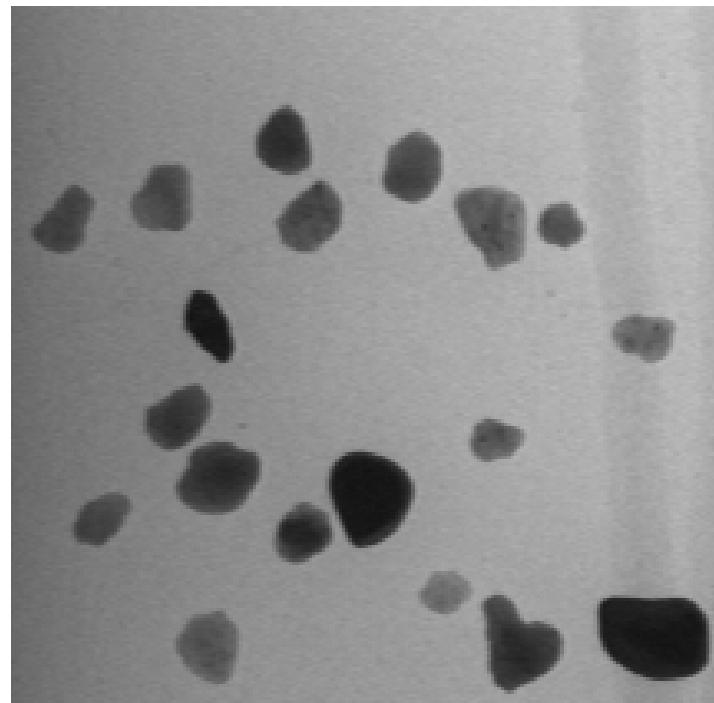
- Co-occurrence

- (m,n)
- $(m+k,n+l)$

$$h_{k,l}(z_1, z_2) = \frac{1}{K} \underset{(m,n) \in D}{\text{Freq}} (f(m,n) = z_1, f(m+k,n+l) = z_2)$$

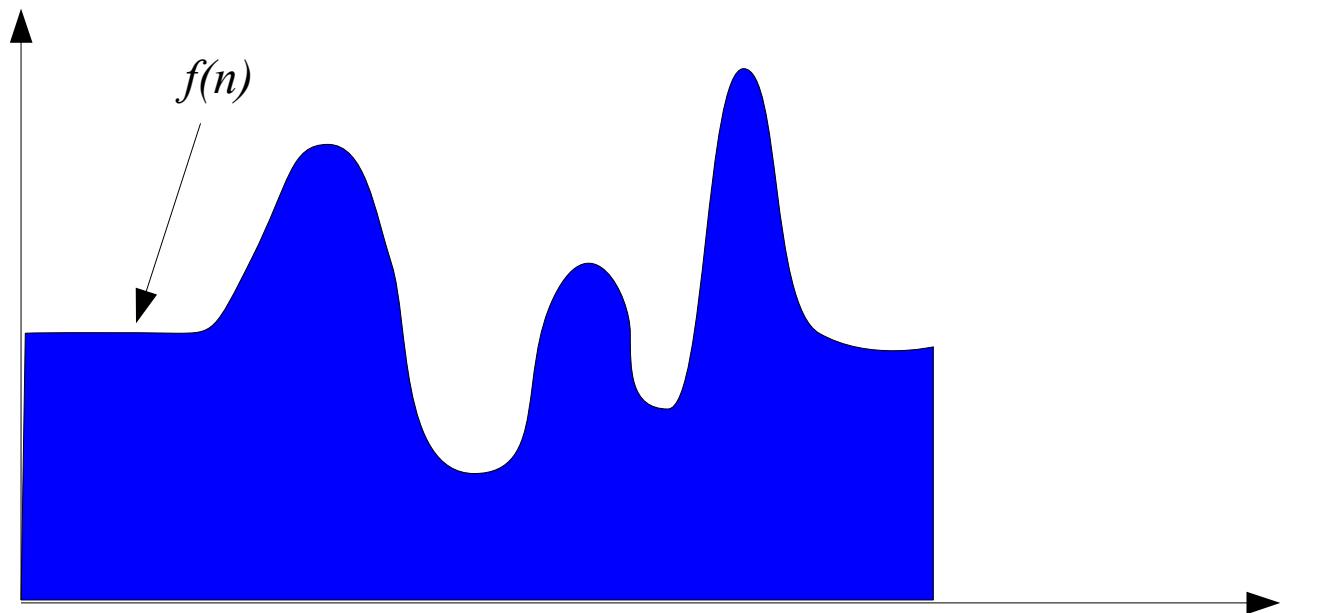
Caractérisation de la texture

Histogramme



Modèle morphologique

- Image = ensemble des points du sous-graphe de la fonction
 - Opérations de base
 - Dilatation, érosion,...



Principales commandes Matlab

```
% Lecture d'une image  
f=imread('Cailloux.bmp');  
  
% Affichage d'une image  
imshow(f);  
% Affichage avec outils d'analyse  
imtool(f);  
% Affichage de la table de couleurs  
colorbar;  
  
% Conversion vers double  
g=double(f);  
% Affichage avec mise à l'échelle  
imshow(g,[]); colorbar;  
% Remise à l'échelle  
g=g/255;  
imshow(g);
```

Principales commandes matlab

```
% Histogramme  
imhist(f);  
  
% Valeurs de l'histogramme  
[h z]=imhist(f);  
bar(z,h);  
  
% Image couleur  
fc=imread('BillesColorees.bmp');  
fr=fc(:,:,1); % plan rouge  
fv=fc(:,:,2); % plan vert  
fb=fc(:,:,3); % plan bleu  
  
% Conversion en niveaux de gris  
fg=rgb2gray(fc);
```

Exercice 1

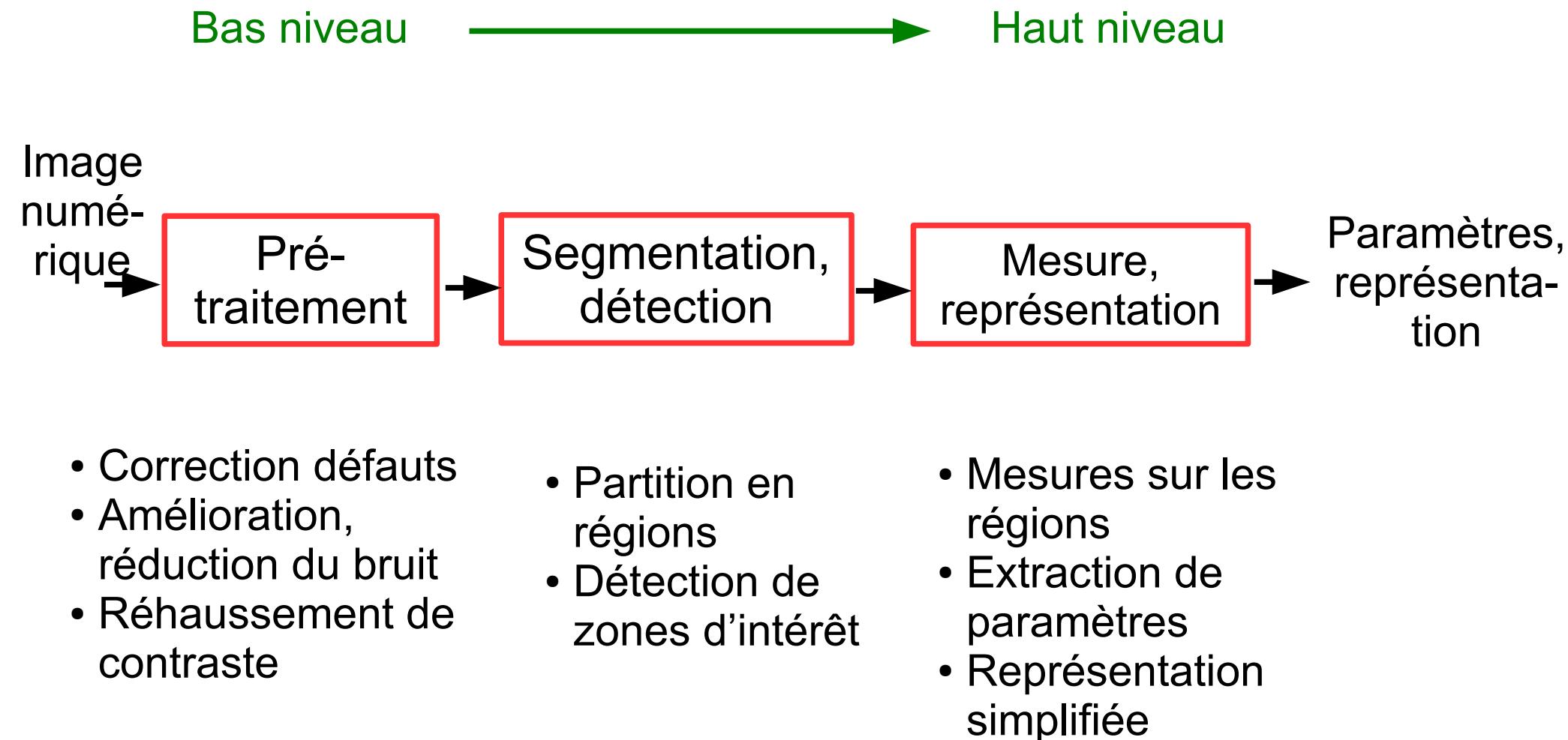
- Modèle probabiliste
 - Simuler plusieurs réalisations d'une image en ajoutant du bruit blanc à une image connue
 - Calculer la moyenne et la variance statistique en chaque pixel
 - Calculer la moyenne et la variance sur une réalisation
 - Calculer la moyenne et la variance locale en chaque pixel (faire varier la taille du voisinage)
 - Comparer les résultats obtenus

Séance 2

Plan

1. Introduction
2. Pré-traitements
3. Seuillage automatique
4. Mesure de paramètres
5. Segmentation par classification
6. Détection de contours
7. Points d'intérêts et mise en correspondance

Chaîne de traitement d'image



Les questions à se poser

- 2.1 Les corrections géométriques ?
- 2.2 Y a-t-il du bruit sur l'image ?
- 2.3 Y a-t-il une dérive de fond ?
- 2.4 Faut-il corriger ou normaliser l'histogramme ?

2.1 Corrections géométriques

- **Centrer l'image sur la zone d'intérêt**
 - Translater, faire une rotation, extraire une zone de travail
→ pas de difficulté particulière
- **Changer la définition**
 - La résolution est-elle suffisante ?
 - Est-il nécessaire de garder une image de 20 Mega-pixels ?

Quelle taille (définition) nécessaire ?

- Qu'est-ce que je veux mesurer ou estimer sur l'image ?
- Calculer la résolution nécessaire
 - La définition est la taille en pixels de l'image
 - La résolution est le nombre de pixels par unité de longueur

$$r = \frac{\text{nombre de pixels}}{\text{taille de référence}} \text{ (pixel / mm)}$$

- La taille apparente du pixel

$$\alpha = \frac{1}{r}$$

- La résolution minimale est :

$$r_{min} = \frac{1}{\text{taille du plus petit détail à mesurer}}$$

- La taille (définition) minimale de l'image est :

$$d_{min} = r_{min} \times \text{taille_du_champ}$$

- Facteur d'échelle à appliquer

$$a = \frac{\text{taille finale}}{\text{taille initiale}}$$

Rééchantillonnage

- Les corrections géométriques nécessitent un rééchantillonnage de l'image

- Translation non entière

$$f_t(m, n) = f(m - t_y, n - t_x)$$

- Changement de résolution

$$f_a(m, n) = f(m/a, n/a)$$

Matlab : imresize, imrotate, imtranslate
→ choix de la méthode d'interpolation

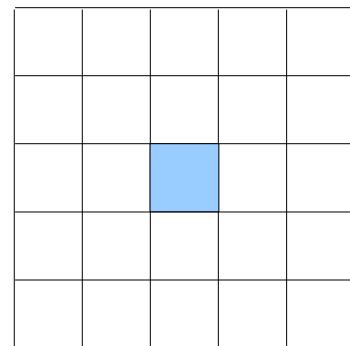
→ Nécessite une interpolation

Translation non entière : principe du calcul

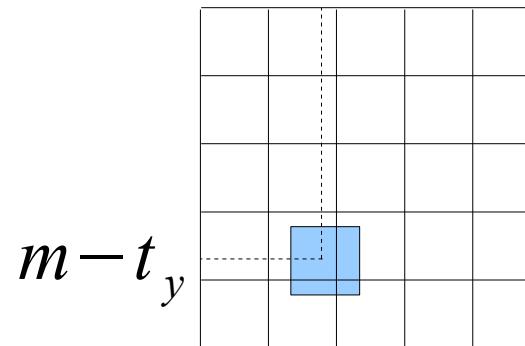
$$f_t(m, n) = f(m - t_y, n - t_x) \quad f(m', n')$$

n

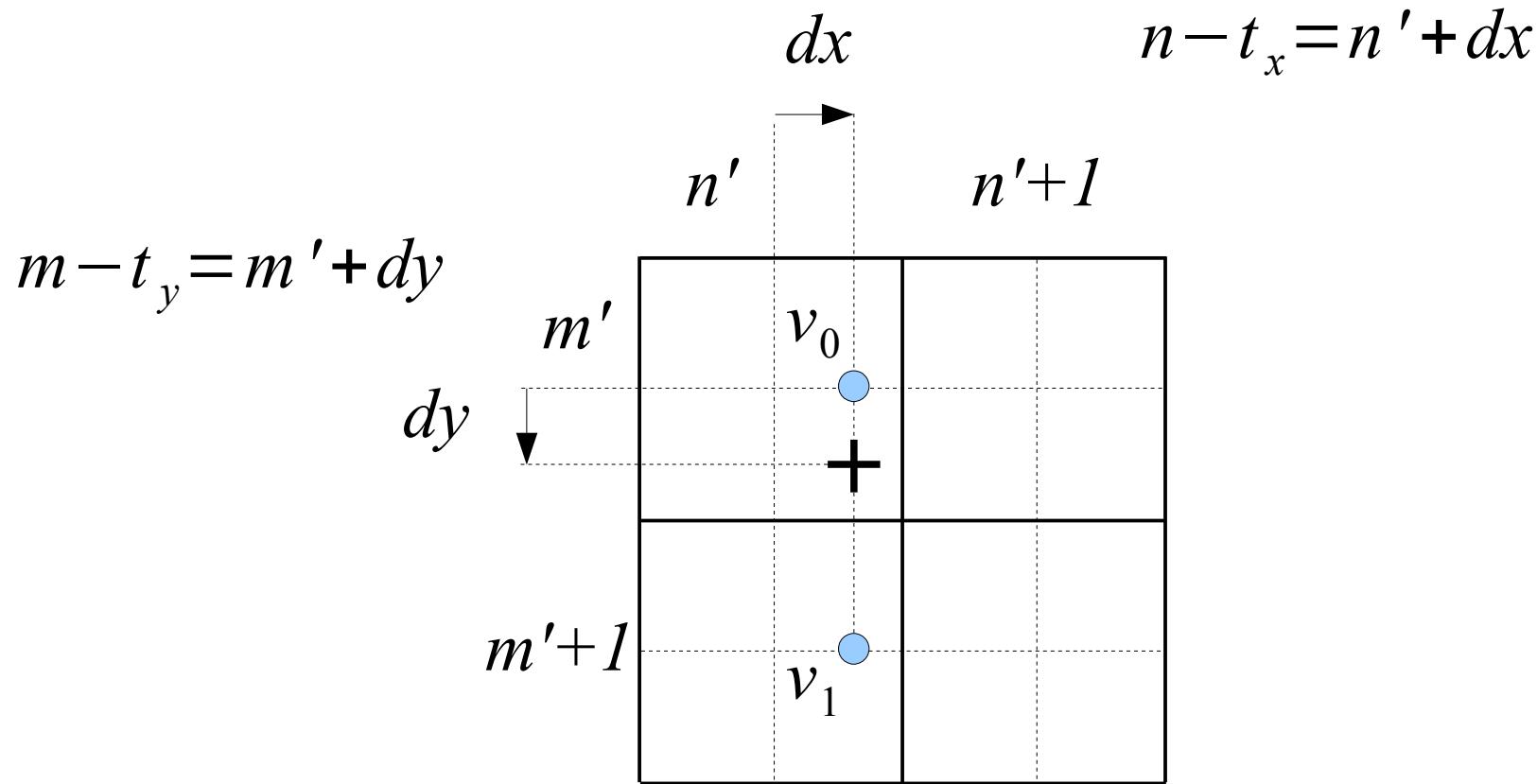
m



$n - t_x$



Interpolation bilinéaire



$$v_0 = (1 - dx) f(m', n') + dx f(m', n' + 1)$$

$$v_1 = (1 - dx) f(m' + 1, n') + dx f(m' + 1, n' + 1)$$

$$f_t(m, n) = (1 - dy) v_0 + dy v_1$$

Les questions à se poser

2.1 Les corrections géométriques ?

2.2 Y a-t-il du bruit sur l'image ?

2.3 Y a-t-il une dérive de fond ?

2.4 Faut-il corriger ou normaliser
l'histogramme ?

Sources de bruit

- Capteur et circuit électronique
 - Du à la faible illumination ou au bruit thermique
 - Généralement blanc et gaussien
- Bruit quantique
 - Blanc poissonnien
- Erreurs de numérisation
 - Poivre et sel
- Erreurs de quantification
 - Uniforme

Estimation du bruit

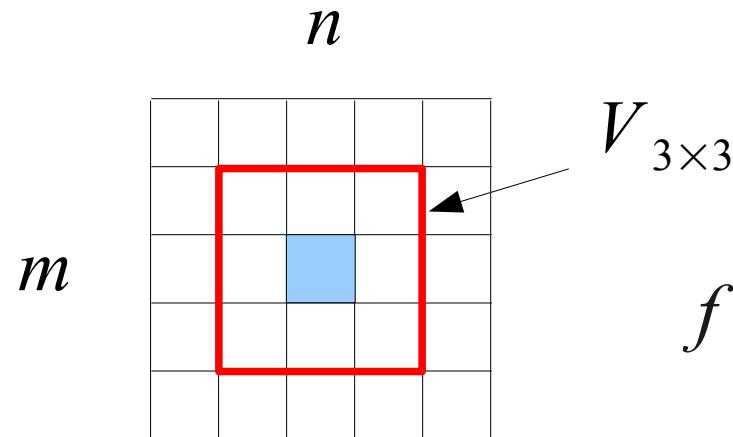
- Si on a plusieurs réalisations
 - Estimation statistique sur les réalisations
- Si on n'a qu'une seule image
 - Hypothèse d'ergodicité et statistiques spatiales locales
- Rapport signal sur bruit

$$SNR = \frac{\frac{1}{NM} \sum_{m=1}^M \sum_{n=1}^N f^2(m, n)}{\sigma^2}$$

$$SNR_{dB} = 10 \log_{10}(SNR)$$

Réduction de bruit

- Opérateurs de voisinage



$$f'(m, n) = \text{Oper}_{(k, l) \in V_{2a+1 \times 2a+1}} [f(m+k, n+l)]$$

$$V_{2a+1 \times 2a+1} = \{(k, l) \mid k = -a, \dots, 0, \dots, a, l = -a, \dots, 0, \dots, a\}$$

Filtre linéaire

- Opérateur de convolution discret

- Formule directe

Matlab : conv2 ou
imfilter

$$\begin{aligned}g(m, n) &= \sum_{k=-a}^a \sum_{l=-a}^a f(m+k, n+l) h(-k, -l) \\&= \sum_{k=-a}^a \sum_{l=-a}^a f(m-k, n-l) h(k, l) \\&= (f * h)(m, n)\end{aligned}$$

- Calcul par transformée de Fourier rapide

$$g(m, n) = \text{FFT}^{-1}(\text{FFT}(f) \text{FFT}(h))$$

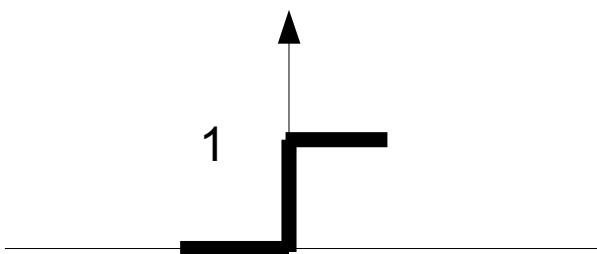
Filtres classiques

- Moyenneur

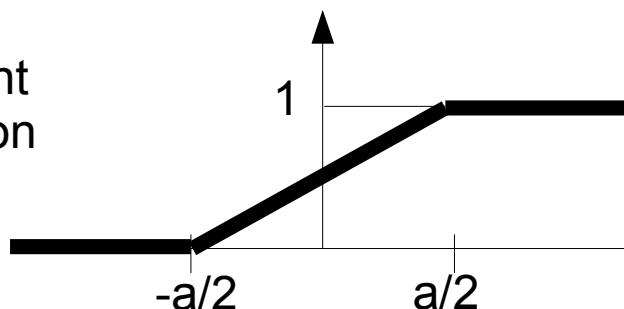
Modèle additif : $f^* = f + \eta$

$$h(x, y) = \frac{1}{a^2} \quad (x, y) \in [-a/2, a/2] \times [-a/2, a/2]$$

$$f(x, y) = H(x) \quad \begin{aligned} g &= f * h \\ &= 1/2 + x/a \quad x \in [-a/2, a/2] \end{aligned}$$



Élargissement de la transition



Réduction du bruit : $\text{var}(\eta * h) = \frac{\sigma_\eta^2}{a^2}$

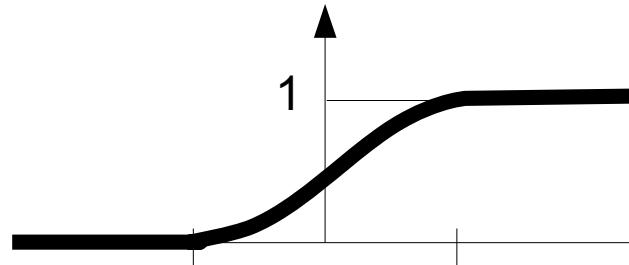
```
Matlab :  
h=fspecial('average',[3 3]);  
g=imfilter(f,h) ;
```

Filtres classiques

- **Gaussien**

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

$$\text{var}(\eta * h) = \frac{\sigma_\eta^2}{4\pi\sigma^2} \quad g = \text{erf}\left(\frac{x}{\sigma}\right)$$



Matlab :
`h=fspecial('gaussian',3,0.5);
g=imfilter(f,h);
% g=imgaussfilt(f,0.5);`

Filtrage non linéaire

- **Filtrage d'ordre**

Si on a une fenêtre de taille : $(2a + 1) \times (2a + 1)$
centrée en (m,n) , avec : $L = (2a + 1)^2$

On numérote les pixels de 1 à L : $p = \{p_k\}_{k=1,\dots,L}$

On trie les éléments de a : $p_{(1)} \leq p_{(2)} \leq \dots \leq p_{(L)}$

La nouvelle valeur du pixel est :
 $p_{(k)}$ = valeur de rang k
 = filtrage d'ordre de rang k

- **Filtrage médian**

Valeur médiane = filtrage d'ordre de rang $(L+1)/2$

La nouvelle valeur du pixel est : $p_{((L+1)/2)}$

Moins efficace pour la suppression d'un bruit gaussien

Préserve les transitions

Matlab : `g=medfilt2(f,[3 3]);`

Réduction de bruit



Image 256x256
Bruit SNR 17 dB



Filtre moyenneur 3x3

Réduction de bruit



Filtre Gaussien sigma=1



Filtre moyenneur 3x3

Réduction de bruit



Filtre médian 3x3



Filtre moyenneur 3x3

Réduction de bruit



Bruit impulsional « poivre et sel »



Filtre moyenneur 3x3

Réduction de bruit



Bruit impulsional « poivre et sel »



Filtre médian 3x3

Les questions à se poser

2.1 Les corrections géométriques ?

2.2 Y a-t-il du bruit sur l'image ?

2.3 Y a-t-il une dérive de fond ?

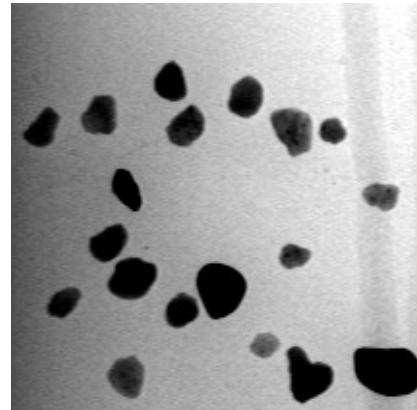
2.4 Faut-il corriger ou normaliser
l'histogramme ?

2.3 Dérive de fond

- L'objet d'étude est superposé à une scène fixe



- Il y a une dérive de luminance



Différence d'image



Détection de changement : différence absolue



Correction d'une dérive de fond

Les questions à se poser

- 2.1 Les corrections géométriques ?
- 2.2 Y a-t-il du bruit sur l'image ?
- 2.3 Y a-t-il une dérive de fond ?
- 2.4 Faut-il corriger ou normaliser l'histogramme ?

2.4 Transformation de l'histogramme

- Expansion de dynamique (étalement)
 - On cherche à utiliser toute la plage des niveaux de gris (0 - Z)

$$z_{min} = \min_{m,n} f(m,n)$$

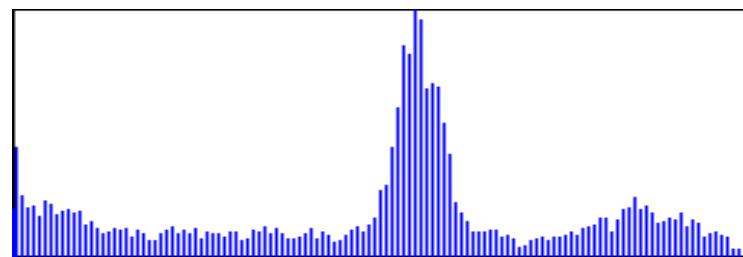
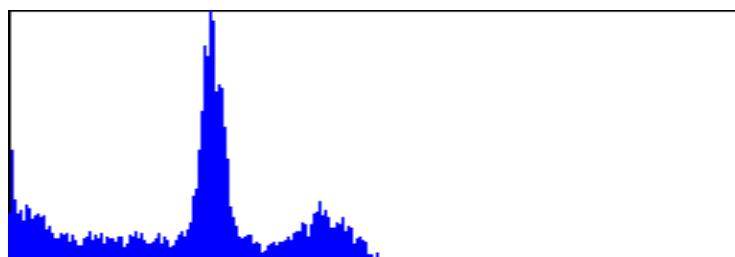
$$z_{max} = \max_{m,n} f(m,n)$$

$$g(m,n) = (f(m,n) - z_{min}) \frac{Z}{z_{max} - z_{min}}$$

Matlab

```
J = imadjust(I,[low_in; high_in],[low_out; high_out]);
```

Expansion de dynamique



Expansion de dynamique

- Problème des valeurs singulières
 - Le min ou le max peut être atteint pour un petit nombre de pixels (bruit)
 - Instabilité de la transformation
- Autres choix possibles
 - Utiliser les centiles de l'histogramme
 - $z_{min} = 1^{\text{er}} \text{ centile}$, $z_{max} = 99^{\text{ème}} \text{ centile}$
 - Utiliser un pourcentage de la classe d'effectif maximum comme seuil de l'histogramme

```
Matlab  
J = imadjust(I);
```

Transformation de l'histogramme

- **Egalisation d'histogramme**
 - On cherche à rendre l'histogramme plat

image $I \rightarrow$ fonction de gris $f \rightarrow$ histogramme $h_f(z)$

image $J \rightarrow$ fonction de gris $g \rightarrow$ histogramme $h_g(z)$

On cherche une transformation $T : z \rightarrow z'$

telle que :

$$g(m, n) = \underset{z}{T}[f(m, n)]$$

Soit l'histogramme cumulé : $H(z) = \sum_{u=0} h_f(z)$

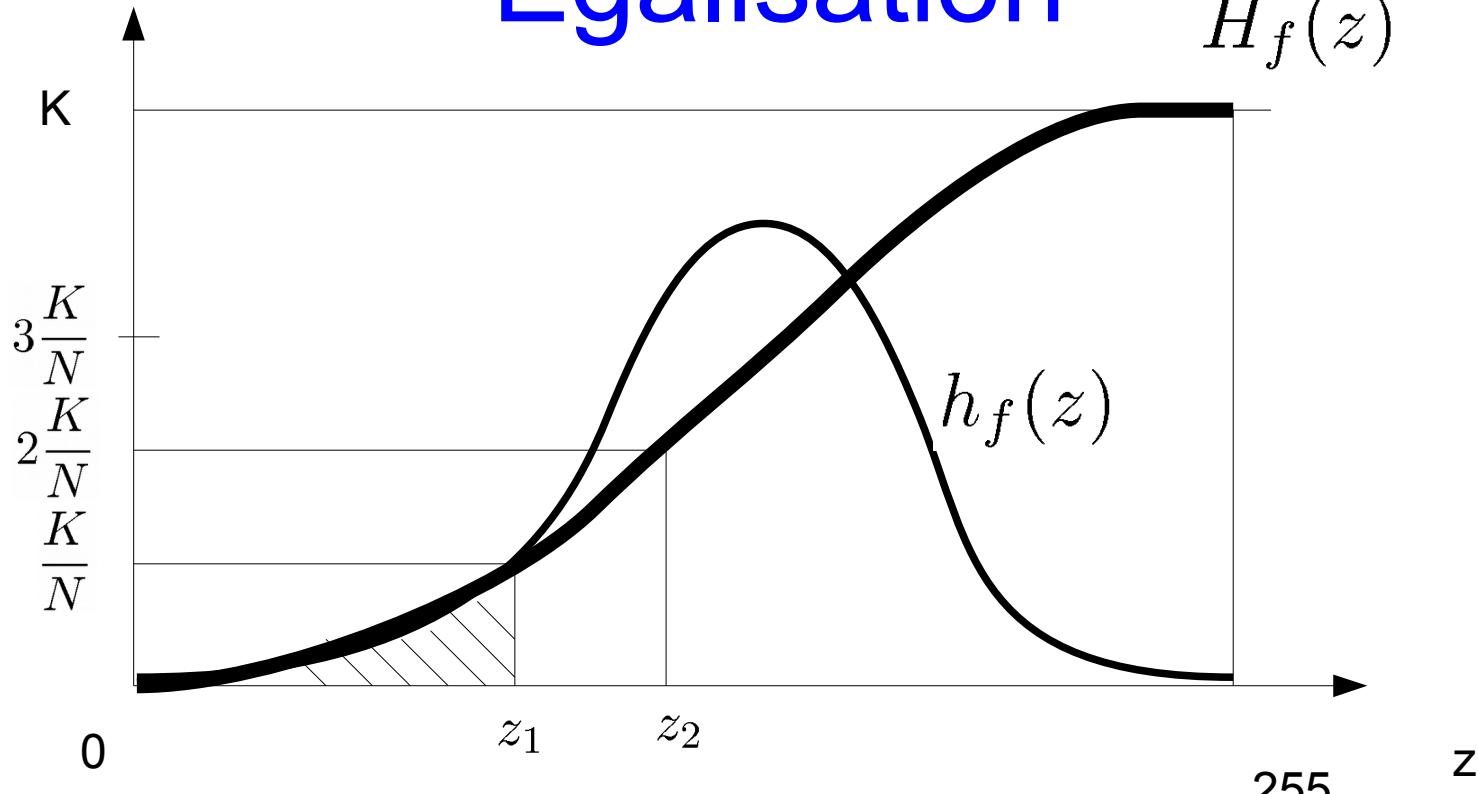
On montre que :

$$T(z) = \frac{255}{K} H(z)$$

Nombre de pixels

Matlab
`J = histeq(I);`

Egalisation



$$z' = 1 \rightarrow \frac{K}{N} = \sum_{z=0}^{z_1} h_f(z) = H_f(z_1) \quad \text{remplissage 1ère classe}$$

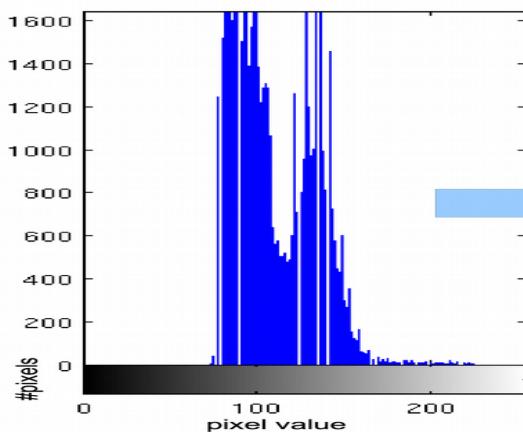
$$z' = 2 \rightarrow 2\frac{K}{N} = \sum_{z=0}^{z_2} h_f(z) = H_f(z_2) \quad \text{remplissage 2ème classe}$$

donc : $\frac{z'K}{N} = H_f(z) \Rightarrow z' = \frac{N}{K}H_f(z)$

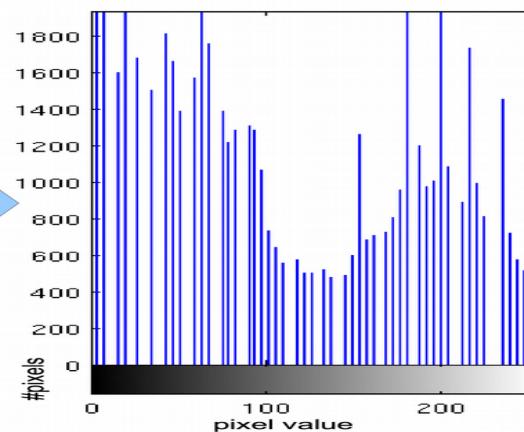
Egalisation



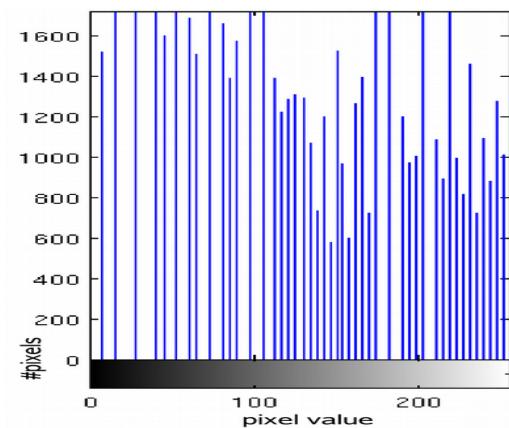
Original



Etalement



Egalisation



Exercice 2

- **Filtrage**
 - Tester les 3 filtres moyenne, gaussien et médian pour une image bruitée (bruit gaussien et bruit poivre et sel)
 - Calculer le SNR avant et après filtrage

Séance 3

Plan

1. Introduction
2. Pré-traitements
3. Seuillage automatique
4. Mesure de paramètres
5. Segmentation par classification
6. Détection de contours
7. Points d'intérêts et mise en correspondance

3. Seuillage automatique

3.1 Introduction

3.2 Maximisation de la variance

3.3 Maximisation de l'entropie

3.4 Evaluation du résultat

3.5 Correction des dérives de fond

3.1 Introduction

- **Principe**
 - Classer les pixels en exploitant leur similarité
→ approche région
 - Chaque pixel peut avoir 1 ou plusieurs attributs
(niveau de gris, composantes couleur, texture,...)
 - 1 seul attribut → méthode mono-dimensionnelle
 - Plusieurs attributs → méthodes de classification (ch 5)
 - Ici : méthodes mono-dimensionnelles

Méthodes mono-dimensionnelles

- Types de méthodes
 - Global
 - Le seuil est global
 - Le calcul ne dépend que de l'histogramme de l'image
 - Local ou dynamique
 - Le seuil dépend du point traité
 - Le calcul dépend du résultat d'opérateurs locaux

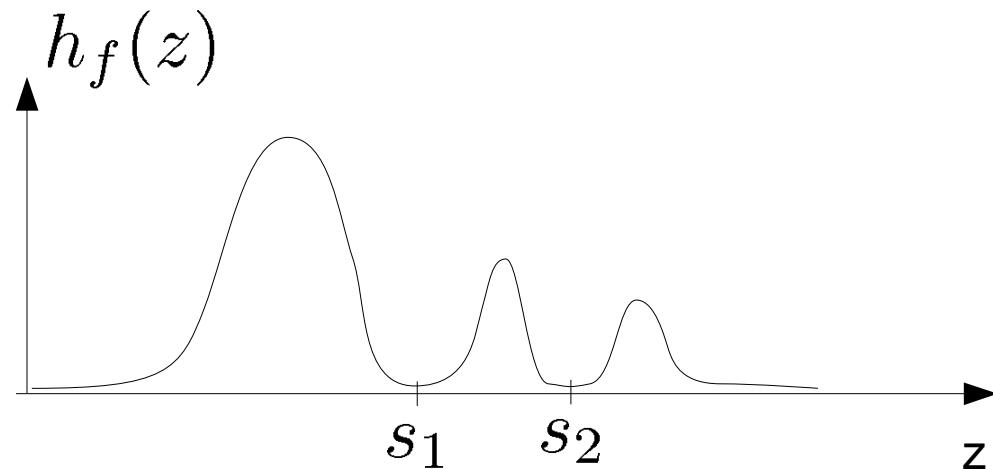
Méthodes globales

- Classes à partir de plages de niveaux de gris
 - Cas 2 classes : binarisation → 1 seuil
 - Cas n_c classes → $n_c - 1$ seuils

$$\begin{aligned} f_s[i, j] &= 0 && \text{si } f[i, j] \leq s_1 \\ &= 1 && \text{si } s_1 < f[i, j] \leq s_2 \\ &\dots \\ &= k && \text{si } s_k < f[i, j] \leq s_{k+1} \\ &\dots \\ &= n_c - 1 && \text{si } s_{n_c - 1} < f[i, j] \end{aligned}$$

Choix des seuils

- Utilisation de l'histogramme
 - Max et min locaux de l'histogramme
 - On peut lisser l'histogramme
 - On peut calculer le log
 - Méthodes automatiques -> présentées ici



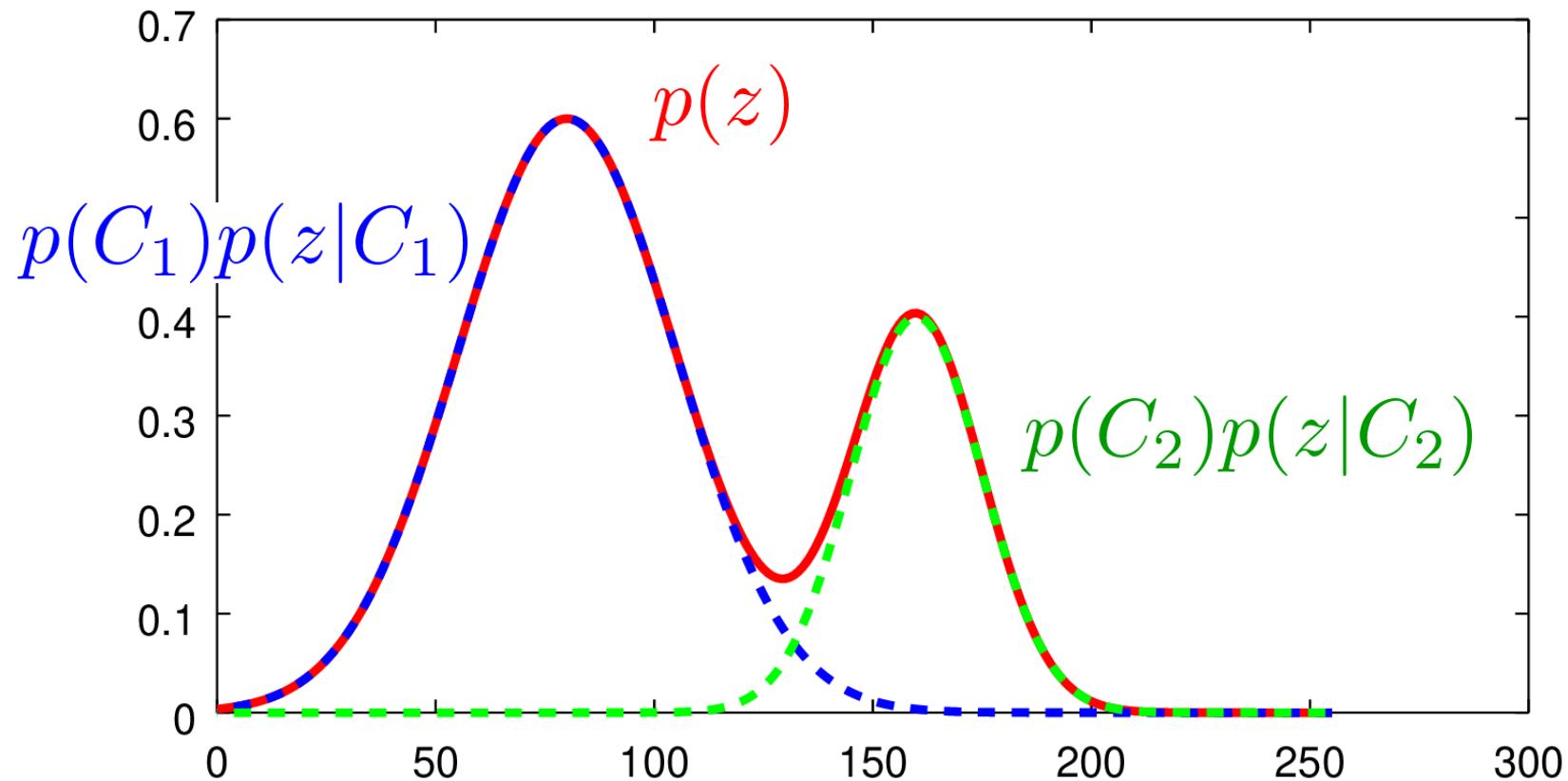
Limitations de ces méthodes

- Sensible à une dérive des niveaux
- Limité aux images en niveaux de gris
- Pour les images couleur
 - Appliquer sur une ou plusieurs composantes couleur
 - Choisir le bon espace
 - Combiner les résultats obtenus
- Utiliser plusieurs composantes simultanément
 - cf. chapitre 5

Principe du seuillage global automatique

- Trouver le seuil "optimal" en se basant uniquement sur les niveaux de gris de l'image
 - Calculer l'histogramme de l'image
 - Choisir un critère d'évaluation de la qualité du seuil
 - Calculer le seuil qui optimise le critère de qualité utilisé

Histogramme interprété comme une densité de probabilité



$$p(z) = p(C_1)p(z|C_1) + p(C_2)p(z|C_2)$$

Estimation des densités

si $h(z)$ est l'histogramme des niveaux de gris

$$p(z) = \frac{1}{K} h(z) = \text{probabilité d'un niveau de gris}$$

$$P(C_1) = \sum_{z=0}^{s_1} p(z) = \frac{K_1}{K} \quad P(C_2) = \sum_{z=s_1+1}^{s_2} p(z) = \frac{K_2}{K}$$

proba qu'un pixel $\in C_1$

$$m_1 = \frac{\sum_{z=0}^{s_1} z p(z)}{P(C_1)}$$

niveau de gris moyen de C_1

Maximisation de la variance interclasse

- Principe (Otsu 79)
 - Minimiser la variance intraclasse
 - càd regrouper les classes autour de leur centre
 - Maximiser la variance interclasse
 - càd séparer au mieux les classes

Variance interclasse

- Recherche des seuils maximisant :

$$\begin{aligned} V(s_1, \dots, s_{n_c-1}) &= \sum_{k=1}^{n_c} P(C_k)(m_k - m)^2 \\ &= \frac{1}{K} \left(\underbrace{\sum_{k=1}^{n_c} K_k m_k^2 - K m^2}_{\text{à maximiser}} \right) \end{aligned}$$

- Revient à minimiser la variance intra-classes

$$V_1(s_1, \dots, s_{n_c-1}) = \sum_{k=1}^{n_c} \sum_{z=s_{k-1}+1}^{s_k} (z - m_k)^2 p(z)$$

Minimisation

- Pour 1 seuil
 - Recherche « force brutale »
- Pour plusieurs seuils
 - Recherche exhaustive (<4 classes)
 - Algorithmes itératifs

Matlab (1 seuil)

```
I = imread('coins.png');  
level = graythresh(I);  
BW = im2bw(I,level);  
% BW = imbinarize(I);  
imshow(BW)
```

Matlab (multiseuil)

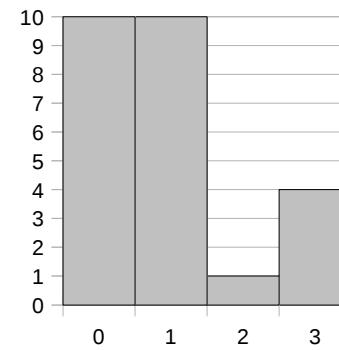
```
thresh = multithresh(I,2);  
seg_I = imquantize(I,thresh);
```

Matlab (TD)

```
seg_I=tse_imthreshold(I,2,'variance');
```

Exemple (1/4)

- Soit une image f :
 - de taille 5x5 pixels
 - avec 4 niveaux de gris (0-3)
 - histogramme $h(z)$ donné par :
- Trouver le seuil pour segmenter cette image par maximisation de la variance interclasse



Exemple simple (2/4)

On a deux classes définies par :

$$\begin{aligned} C_1 &= \{(m, n) | f(m, n) \leq s\} \\ C_2 &= \{(m, n) | f(m, n) > s\} \end{aligned}$$

La variance interclasse est :

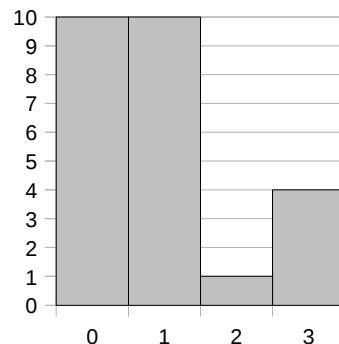
$$\begin{aligned} V(s) &= P(C_1)(m_1 - m)^2 + P(C_2)(m_2 - m)^2 \\ &= \frac{1}{K} (K_1 m_1^2 + K_2 m_2^2 - K m^2) \end{aligned}$$

Il faut maximiser :

$$\begin{aligned} E(s) &= K_1 m_1^2 + K_2 m_2^2 \\ &= \frac{1}{K_1} (\sum_{z=0}^s z h(z))^2 \\ &\quad + \frac{1}{K_2} \left(\sum_{z=s+1}^3 z h(z) \right)^2 \end{aligned}$$

Exemple simple (3/4)

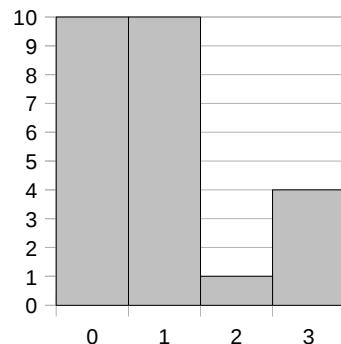
s	K1	K1 m1	K2	K2 m2	E(s)
0					
1					
2					
3					



$$\begin{aligned}
 K_1 &= \sum_{z=0}^s h(z) \\
 K_1 m_1 &= \sum_{z=0}^s z h(z) \\
 K_2 &= \sum_{z=s+1}^3 h(z) \\
 K_2 m_2 &= \sum_{z=s+1}^3 z h(z)
 \end{aligned}$$

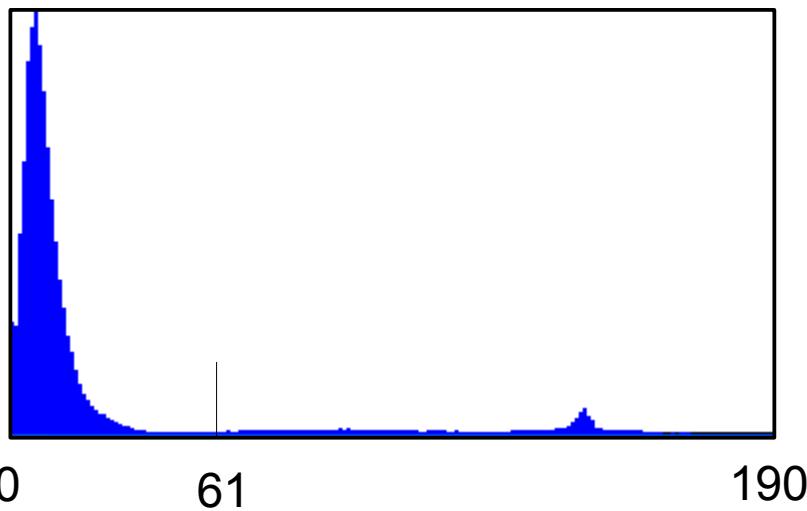
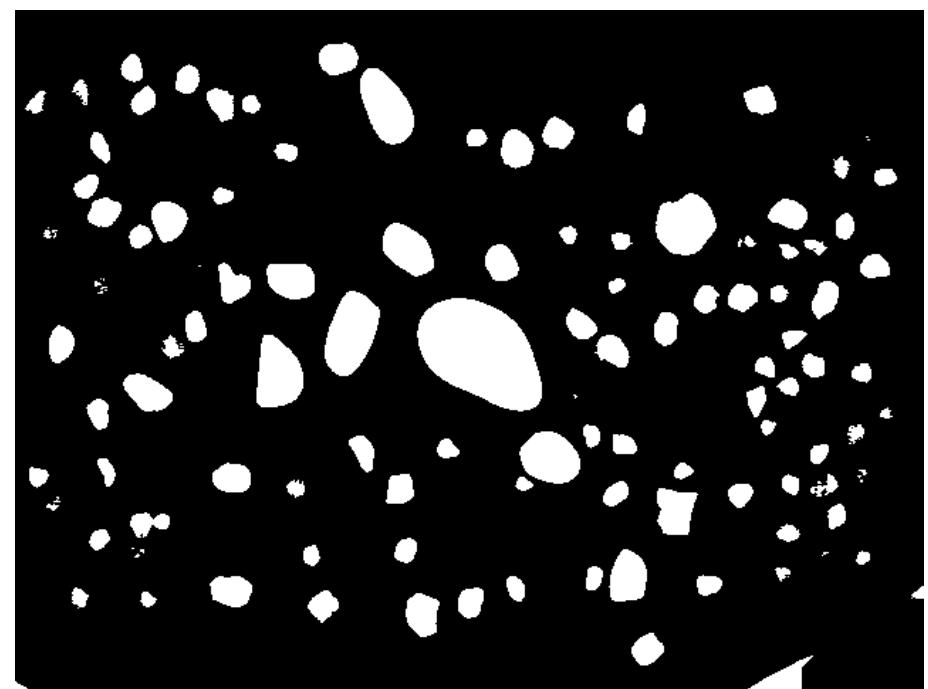
Exemple simple (4/4)

0	10	0	15	24	38,4
1	20	10	5	14	44,2
2	21	12	4	12	42,8571
3	25	24	0	0	23



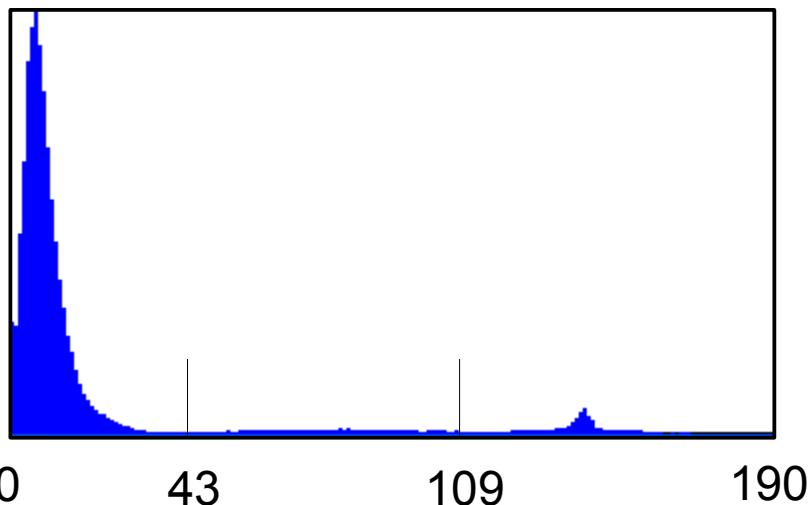
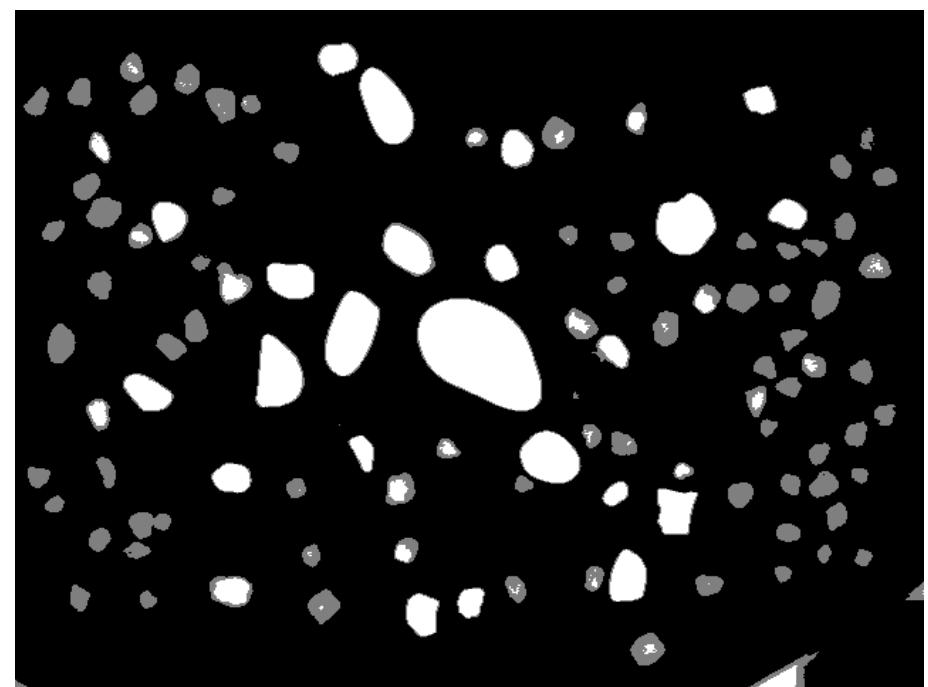
$$\begin{aligned} K_1 &= \sum_{z=0}^s h(z) \\ K_1 m_1 &= \sum_{z=0}^s z h(z) \\ K_2 &= \sum_{z=s+1}^3 h(z) \\ K_2 m_2 &= \sum_{z=s+1}^3 z h(z) \end{aligned}$$

Sur une image (2 classes)



Maximisation variance 2 classes
 $s=61$

Sur une image (3 classes)



Maximisation variance 3 classes
 $s_1=43, s_2=109$

Maximisation de l'entropie

- Principe (Kapur 1985)
 - Entropie = mesure de la quantité d'information (en nb de bits)
 - découpage donnant des classes portant le maximum d'information
 - histogramme de chaque classe le plus étalé possible

Maximisation de l'entropie

- Entropie d'une classe C_k

$$H(C_k) = - \sum_{z=s_{k-1}+1}^{s_k} p(z|C_k) \log_2 p(z|C_k)$$

$$\begin{aligned} p(z|C_k) &= \text{proba d'avoir le ng } z \text{ dans la classe } C_k \\ &\approx \frac{p(z)}{p(C_k)} = \frac{p(z)K}{K_k} \end{aligned}$$

- Pour 2 seuils, il faut maximiser

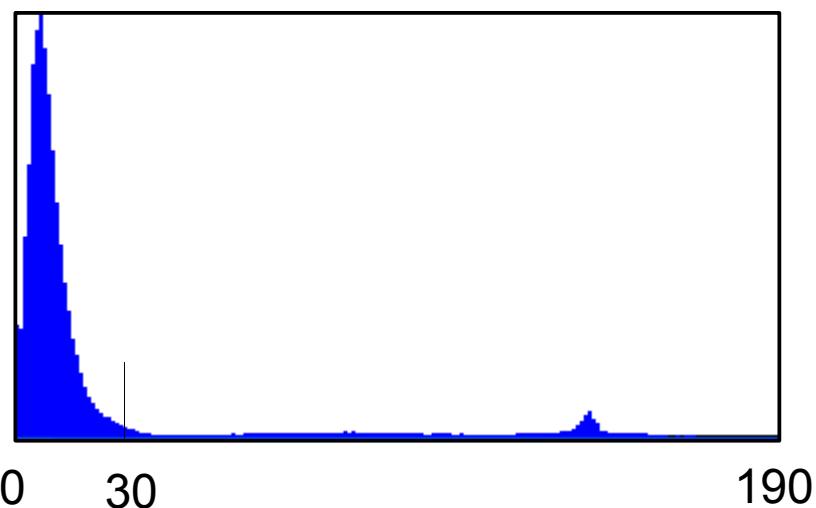
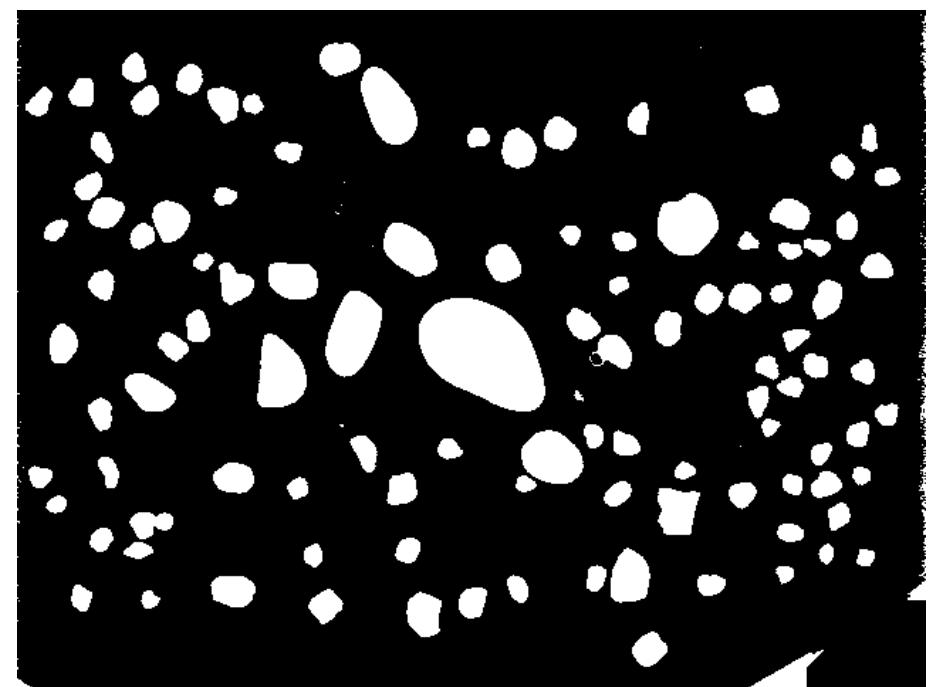
$$\begin{aligned} H(s) &= H(C_1) + H(C_2) \\ &= - \sum_{z=0}^s \frac{p(z)K}{K_1} \log_2 \left(\frac{p(z)K}{K_1} \right) \\ &\quad - \sum_{z=s+1}^{n_g} \frac{p(z)K}{K_2} \log_2 \left(\frac{p(z)K}{K_2} \right) \end{aligned}$$

Minimisation

- Pour 1 ou plusieurs seuils
 - Recherche « force brutale »
 - Algorithme en $O(n_g^{n_c - 1})$

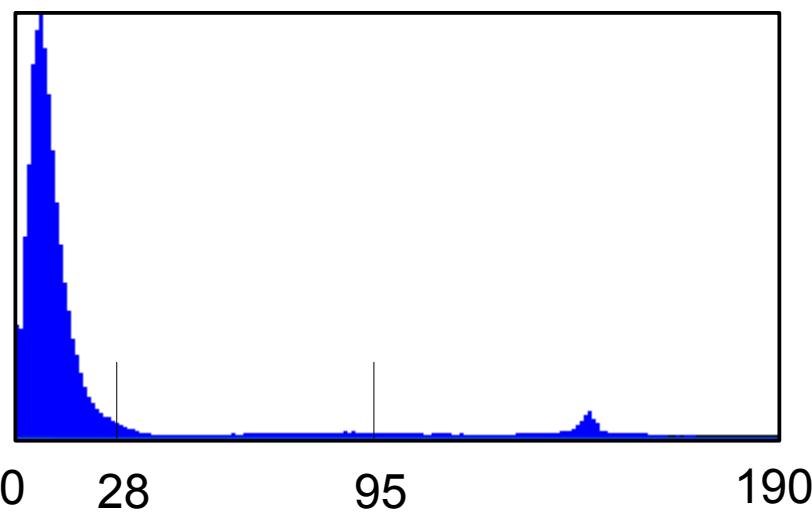
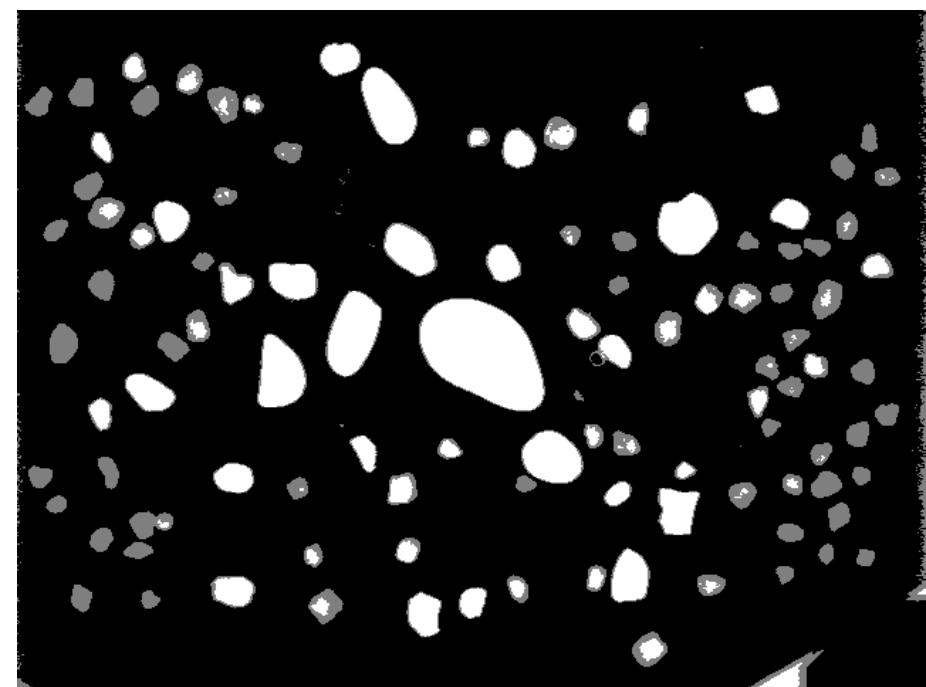
```
Matlab (TD)
seg_l=tse_imthreshold(l,2,'entropy') ;
```

Maximisation entropie 2 classes



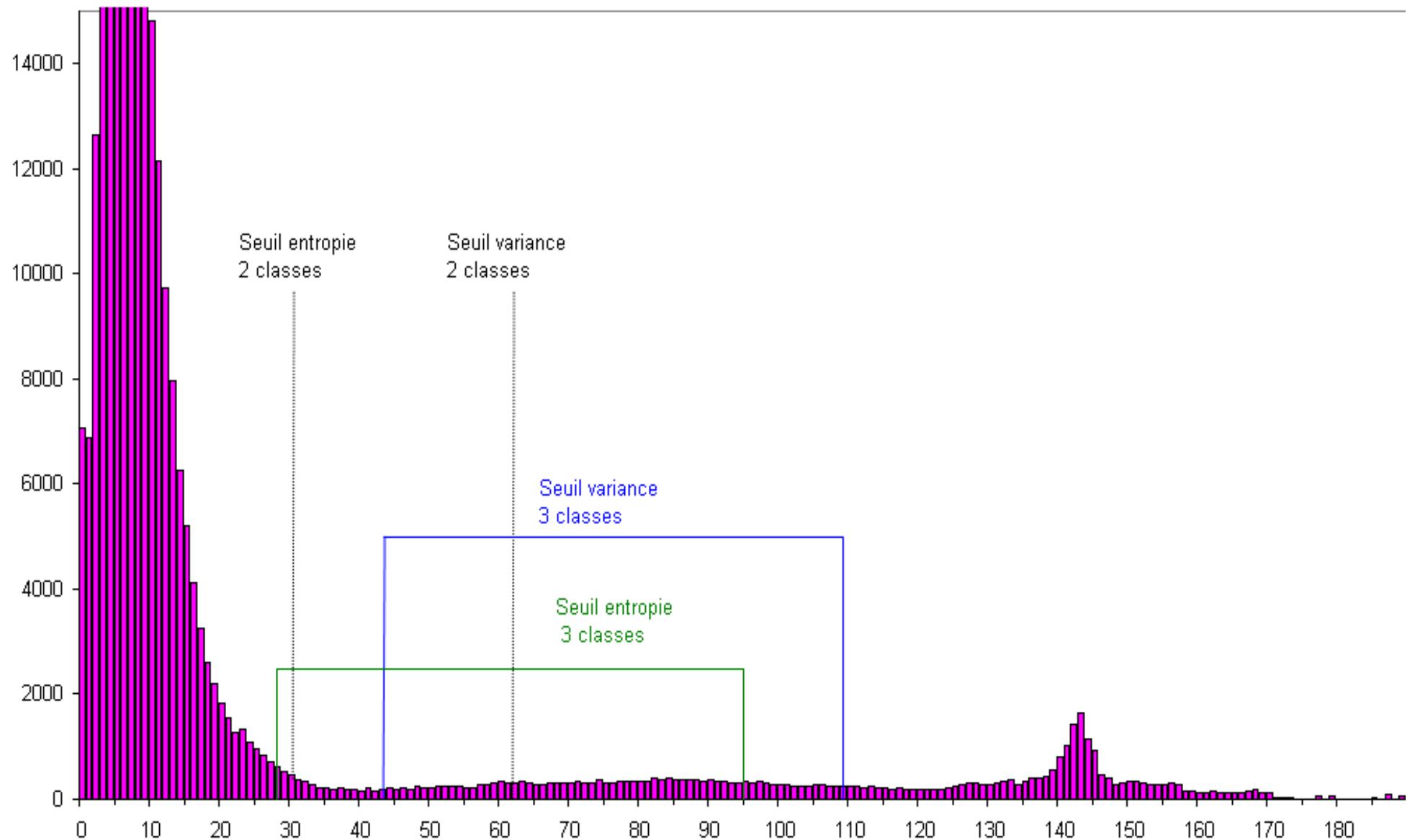
Maximisation entropie 2 classes
 $s=30$

Maximisation entropie 3 classes



Maximisation entropie 3 classes
 $s_1=28, s_2=95$

Comparaison des seuils



Comparaison entre les 2 méthodes

- Maximisation de la variance
 - Adapté pour des distributions gaussiennes
 - Classes de même effectif
 - 1 paramètre à choisir : le nombre de classes
- Maximisation de l'entropie
 - Adapté pour des distributions uniformes
 - Classes d'effectif différent
 - 1 paramètre à choisir : le nombre de classes

3. Seuillage automatique

3.1 Introduction

3.2 Maximisation de la variance

3.3 Maximisation de l'entropie

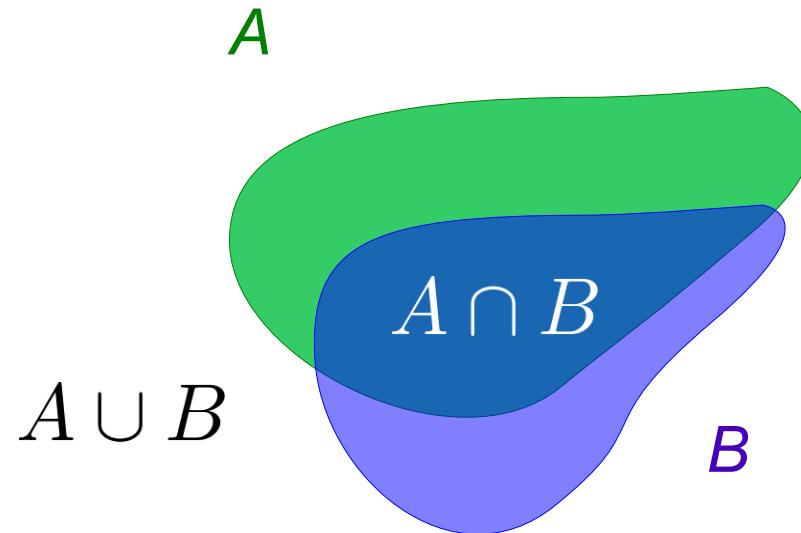
3.4 Evaluation du résultat

3.5 Correction des dérives de fond

Evaluer le résultat d'une segmentation

- Principe
 - Partir d'une segmentation de référence (vérité terrain)
 - Comparer le résultat à la vérité terrain
 - L'idéal est de disposer d'une série d'images de références dans différentes conditions
- Mesures de comparaison possible
 - Distances entre formes binaires
→ différence symétrique

Distance différence symétrique



$$D(A, B) = \frac{A \cup B - A \cap B}{A \cup B}$$

Exercice 3

- Evaluation du seuillage par maximisation de la variance
 - On donne une image et sa vérité terrain (cf. portail)
 - Binariser l'image avec un seuillage automatique par maximisation de la variance (2 classes).
 - Recommencer avec 3 classes en fusionnant les 2 dernières classes.
 - Comparer avec la vérité terrain et conclure

Séance 4

3. Seuillage automatique

3.1 Introduction

3.2 Maximisation de la variance

3.3 Maximisation de l'entropie

3.4 Segmentation de l'histogramme

3.5 Correction des dérives de fond

Correction d'une dérive de fond

- Soustraction du fond (modèle additif)

$$f^* = f + f_0 \quad \Rightarrow \quad f = f^* - f_0$$

↗
image avec
dérive ↗
 fond

f_0 obtenue par une acquisition du fond seul

- Reconstruction du fond
 - Moyenne sur une série d'images
 - Ouverture morphologique (chapeau haut de forme)

Chapeau haut de forme

- Rappels de morphologie (élément struct. plat)

- Erosion

$$\epsilon_t(f)(x) = \min_{\substack{x+y \in D \\ y \in T}} (f(x+y))$$

- Dilatation

$$\delta_t(f)(x) = \max_{\substack{x+y \in D \\ y \in T}} (f(x+y))$$

- Ouverture

$$\gamma_t(f) = (\delta_t \circ \epsilon_t)(f)$$

- Fermeture

$$\phi_t(f) = (\epsilon_t \circ \delta_t)(f)$$

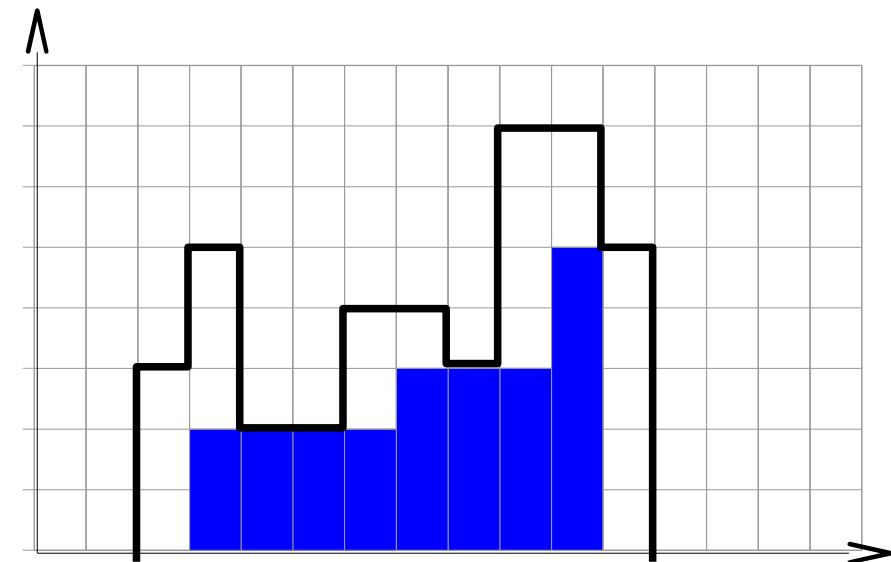
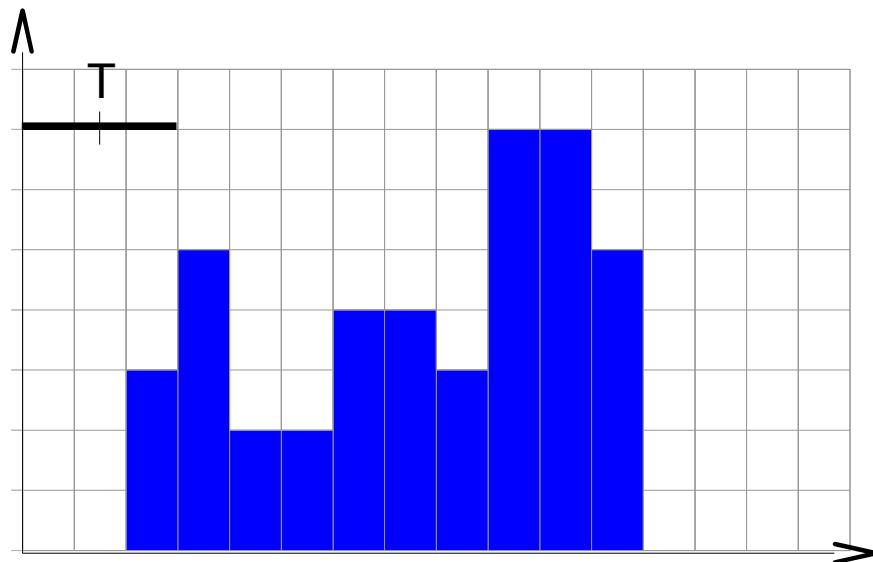
Matlab

```
s=strel('disk',5);
fe=imerode(f,s);
fd=imdilate(f,s);
fo=imopen(f,s);
fc=imclose(f,s);
```

Erosion en niveau de gris

- Erosion pour chaque niveau (élément plat)

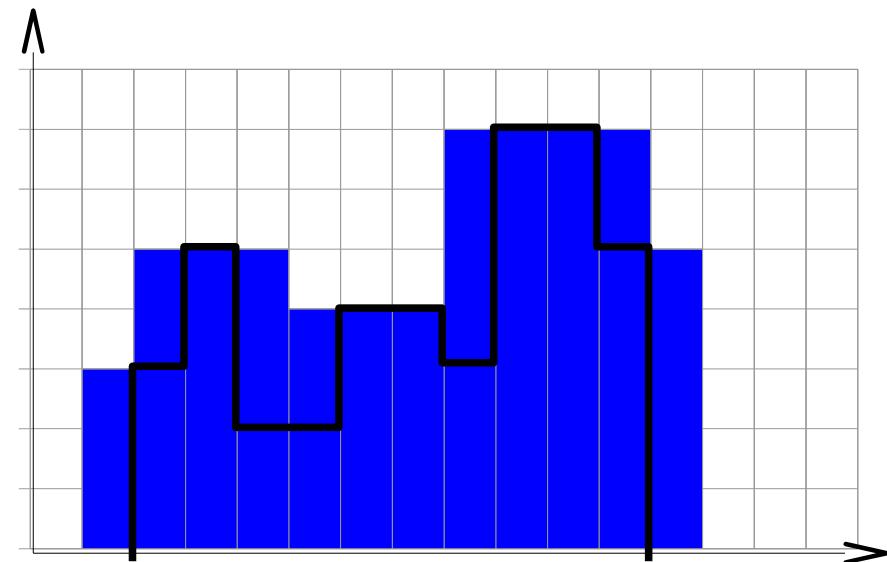
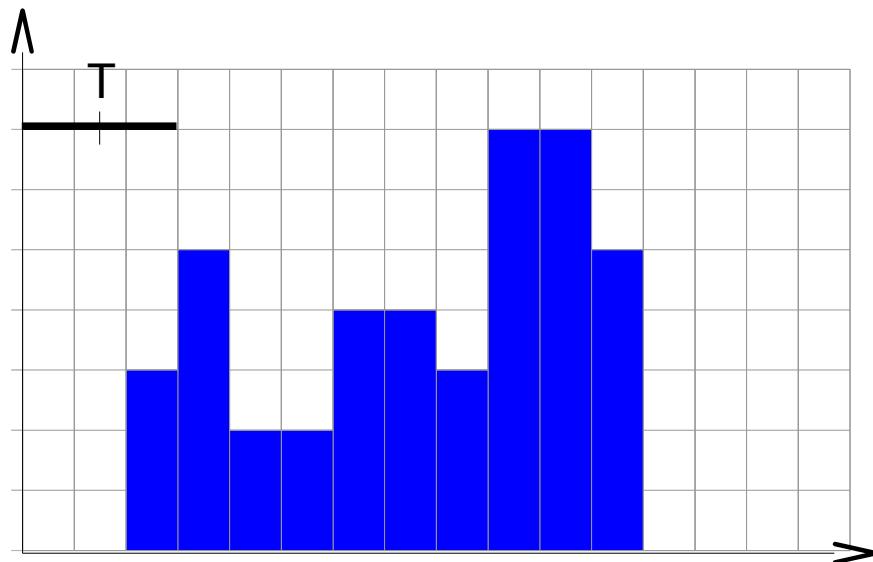
$$\epsilon_t(f)(x) = \min_{\substack{x+y \in D \\ y \in T}} (f(x+y))$$



Dilatation en niveau de gris

- Dilatation pour chaque niveau (élément plat)

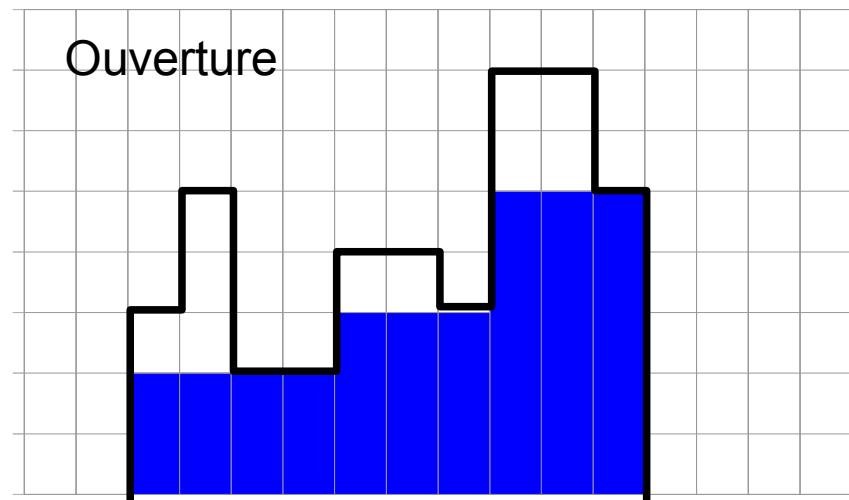
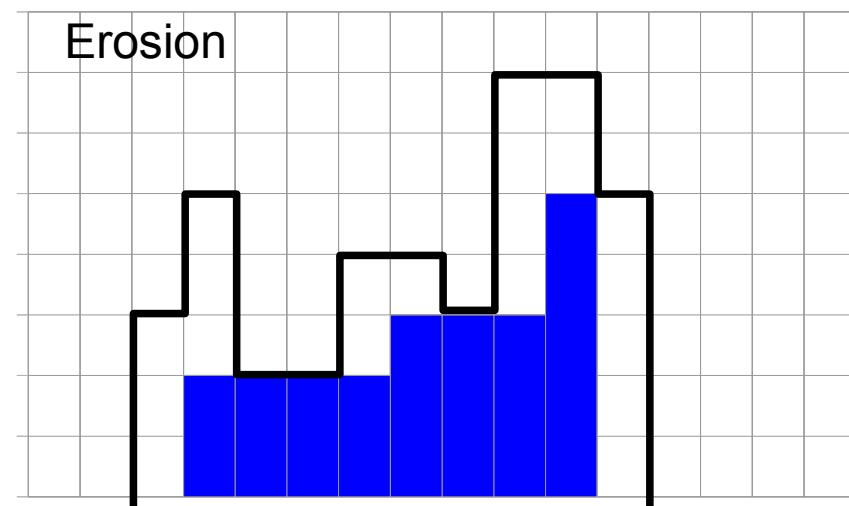
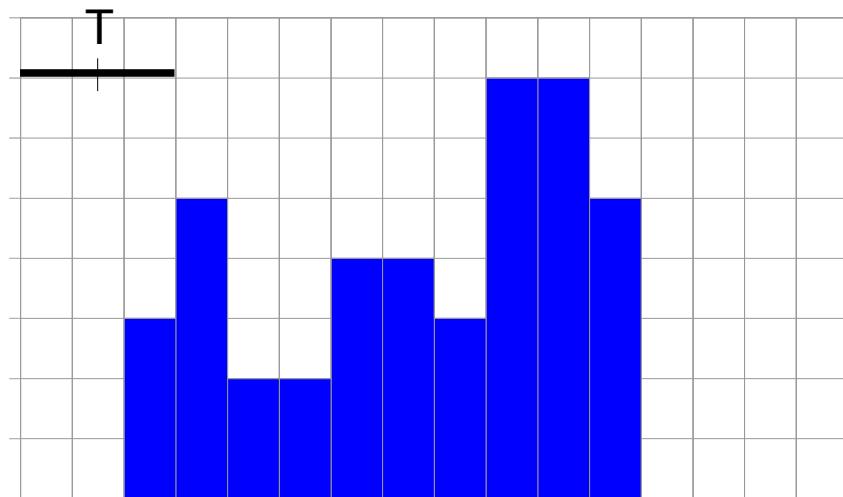
$$\delta_t(f)(x) = \max_{\substack{x+y \in D \\ y \in T}} (f(x+y))$$



Rappel de morphologie

- Ouverture

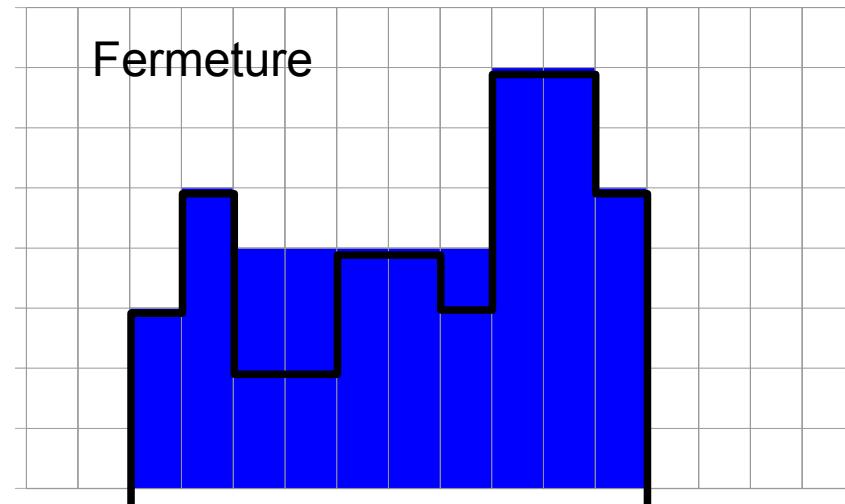
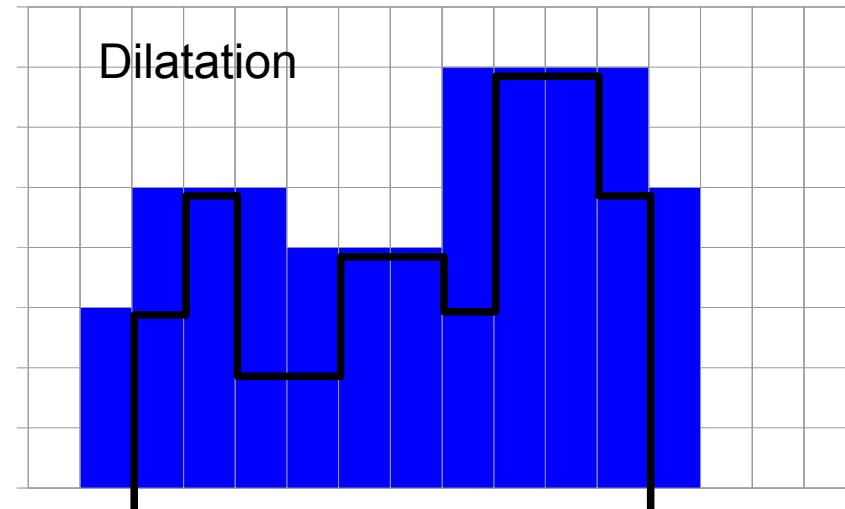
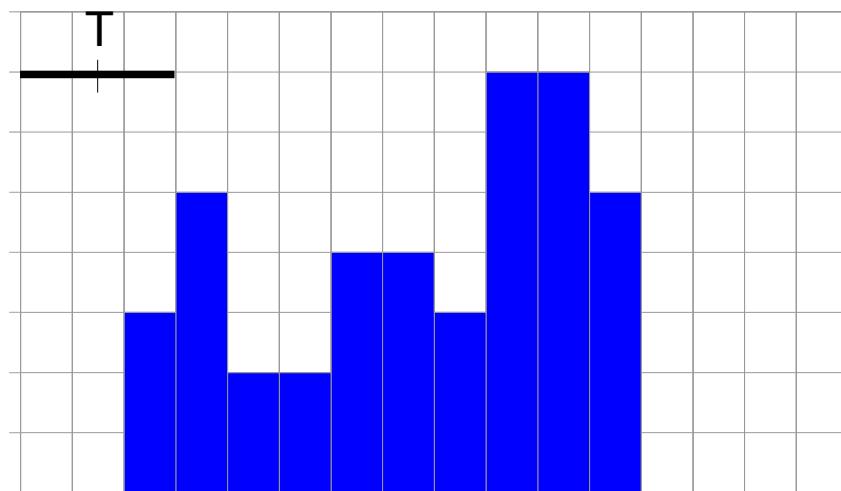
$$\gamma_t(f) = (\delta_t \circ \epsilon_t)(f)$$



Rappel de morphologie

- Fermeture

$$\phi_t(f) = (\epsilon_t \circ \delta_t)(f)$$



Chapeau haut de forme

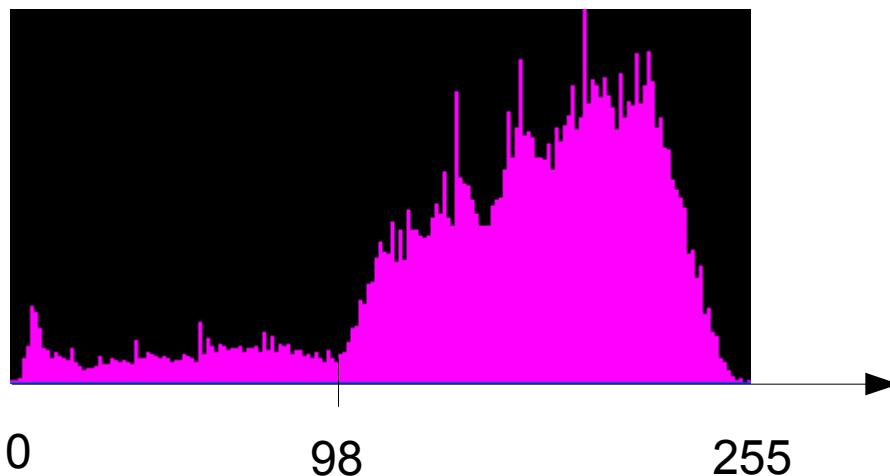
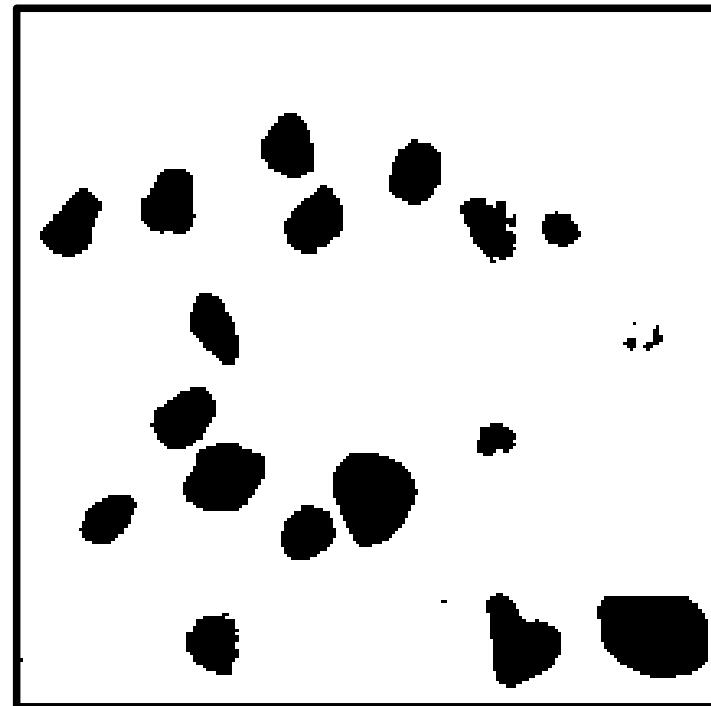
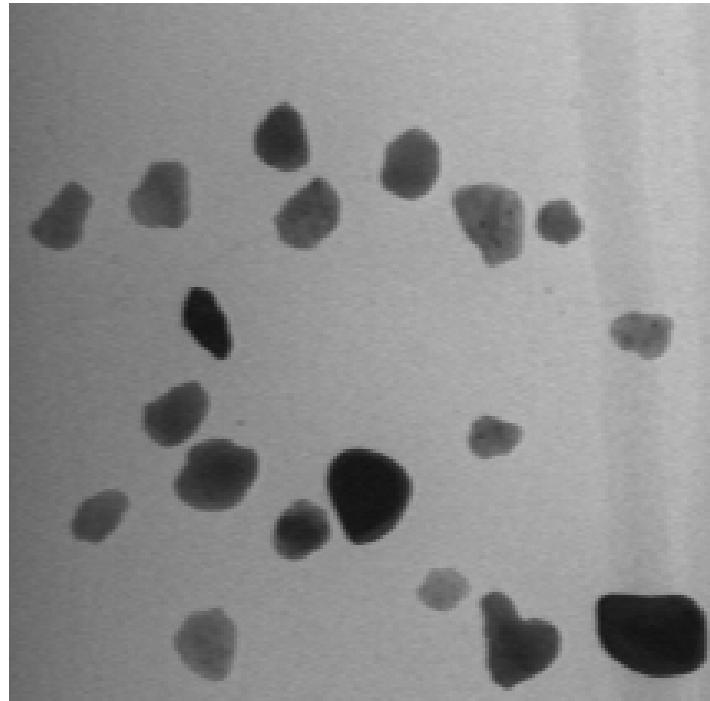
- Top Hat

$$f - \gamma_t(f)$$

- Bottom Hat

$$\phi_t(f) - f$$

Dérive de fond



Seuillage à 98

Correction de la dérive

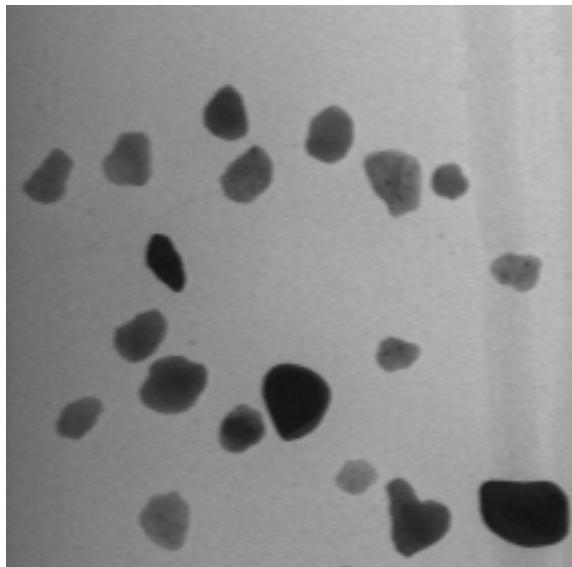
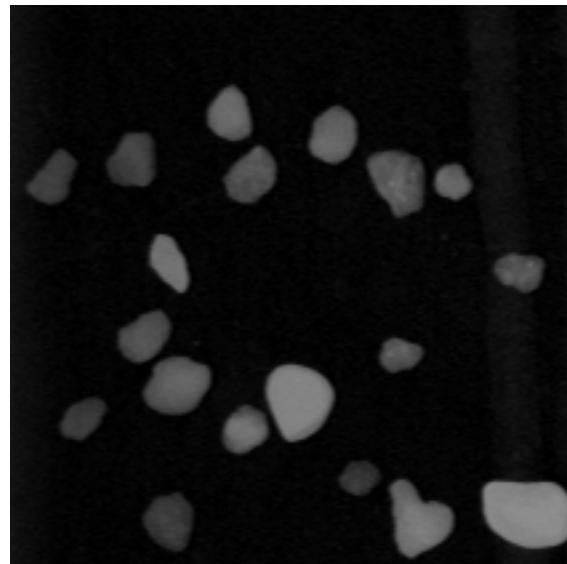


Image 214x212



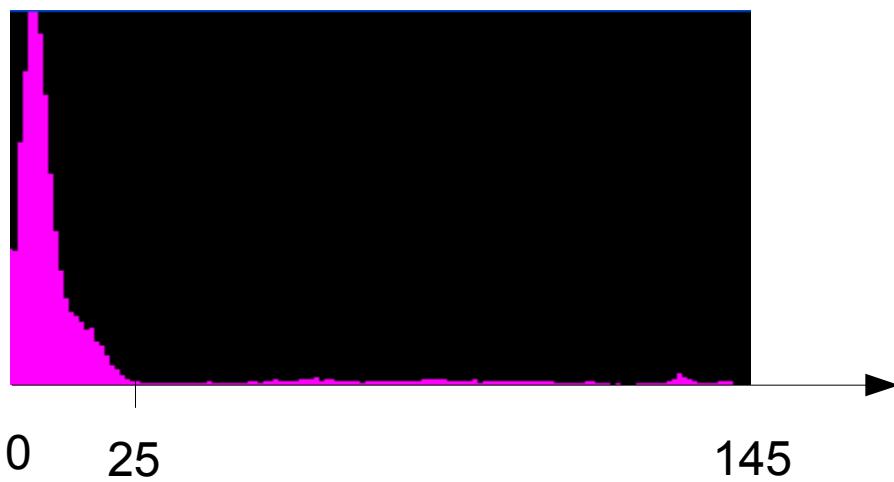
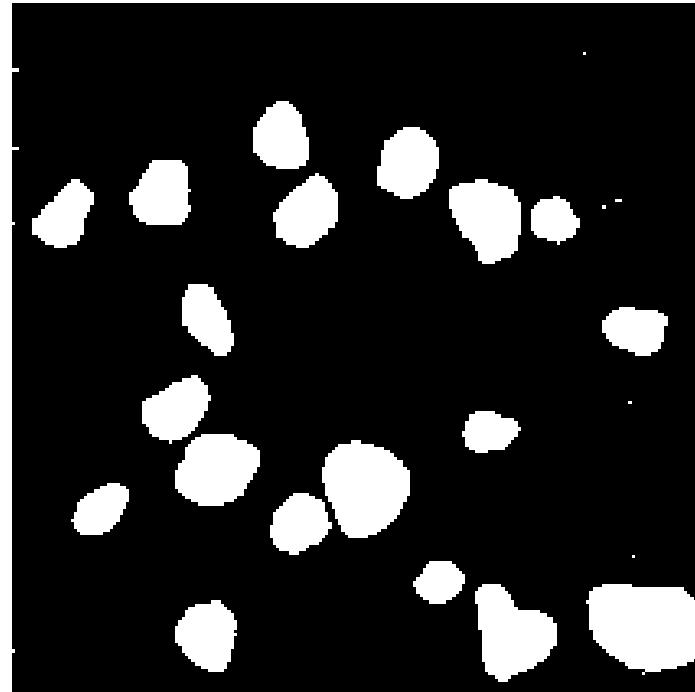
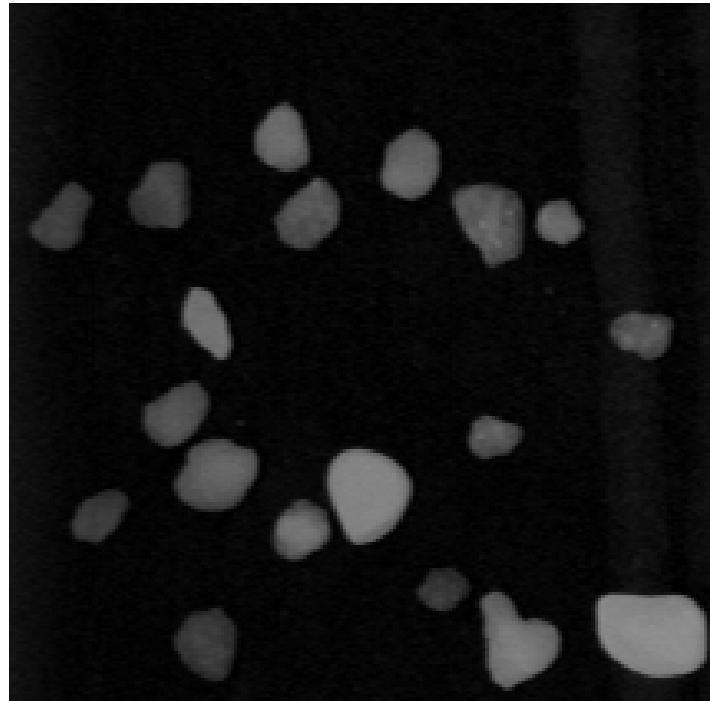
Fermeture de taille 20 : fond

```
Matlab  
s=strel('disk',20);  
g=imclose(f,s);  
h=g-f;
```



Fermeture – image : bottom hat

Seuillage



Seuillage à 25

Conclusion sur les méthodes de seuillage

- Avantages
 - Méthodes généralement simples
 - Maximisation de la variance et de l'entropie
 - Calcul rapide
 - Robustes
- Inconvénients
 - Paramètres à fixer
 - nombre de classes
 - Sensibles à une dérive de fond
 - Pas de méthode générale

Plan

1. Introduction
2. Pré-traitements
3. Seuillage automatique
4. Mesure de paramètres
5. Segmentation par classification
6. Détection de contours
7. Points d'intérêts et mise en correspondance

4 Mesure de paramètres

4.1 Post-traitements

4.2 Etiquetage

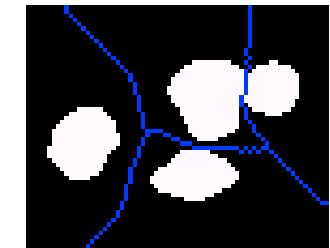
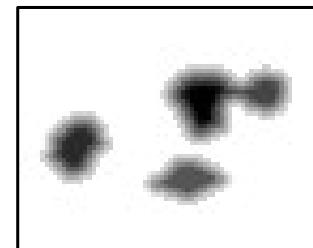
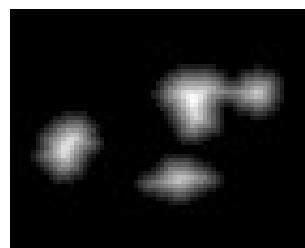
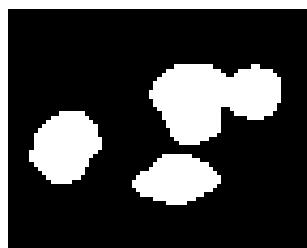
4.3 Mesures

Post-traitements

- Après segmentation il reste des défauts
 - Petites régions ou présence de trous
 - Défauts de bords
 - Régions collées
- Utilisation d'opérations morphologiques
 - Ouverture et fermeture
 - Suppression des régions touchant le bord
 - Reconstruction avec marqueur du bord imclearborder
 - Séparation de régions
 - Carte des distances + Ligne de partage des eaux tse_imsplitobjects

Séparation de régions

- Principe
 - Transformation distance
 - Inverse de la transformation distance
 - Ligne de partage des eaux



Transf.
distance

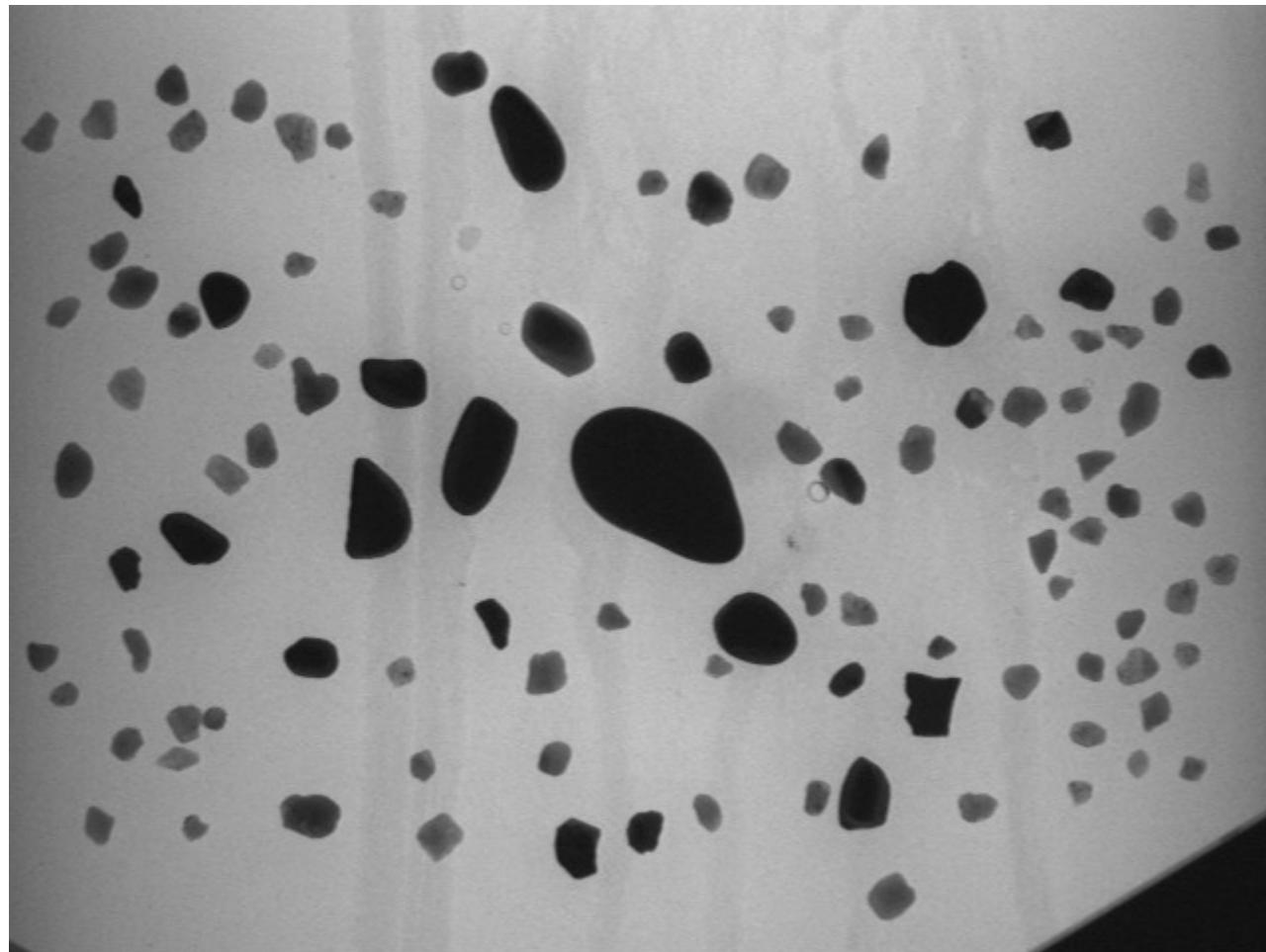
Inverse
transf.
distance

Ligne de
partage des
eaux

Séparation

Exemple complet

- Image originale



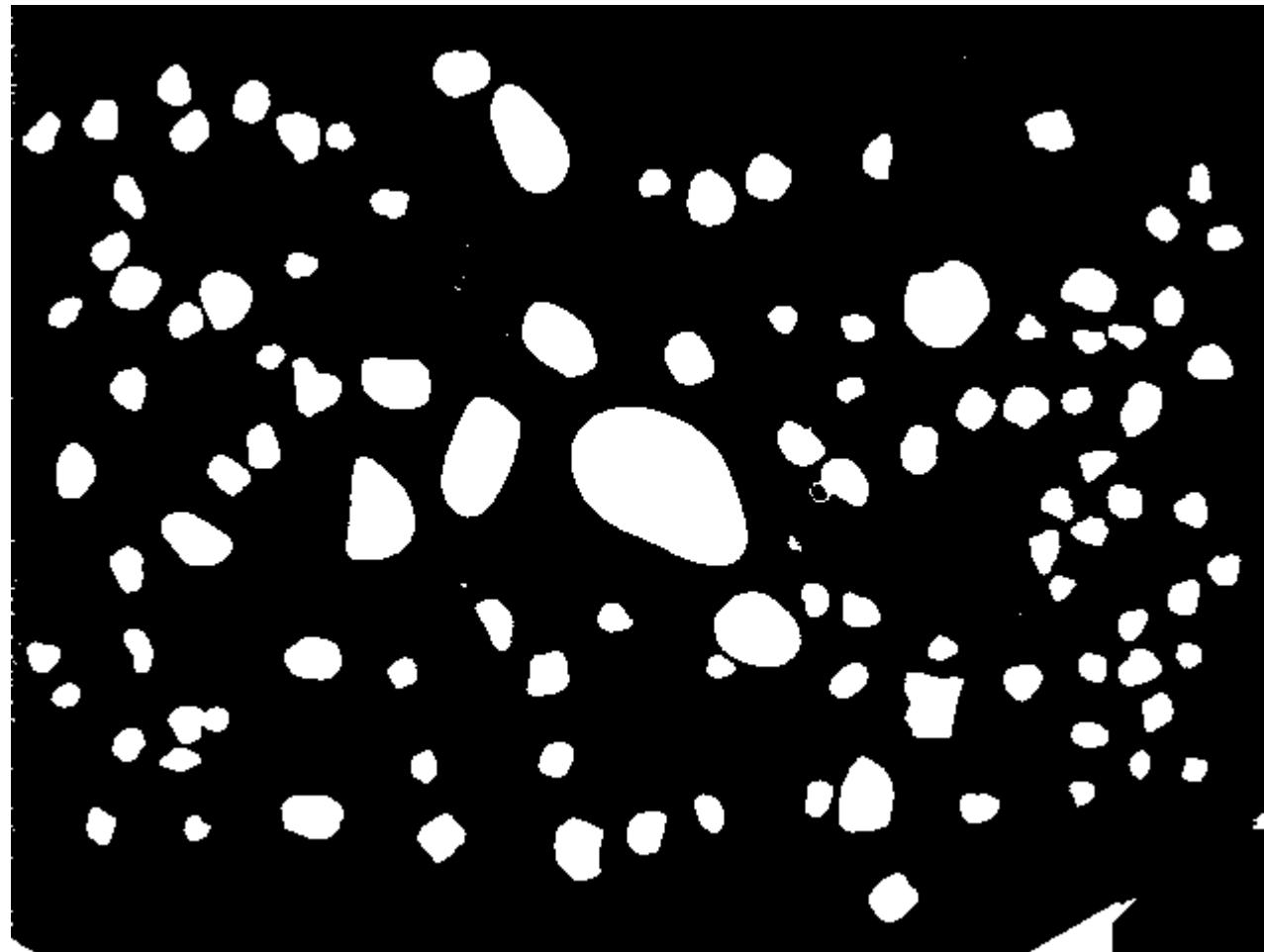
Exemple complet

- Suppression du fond (Bottom Hat 30)



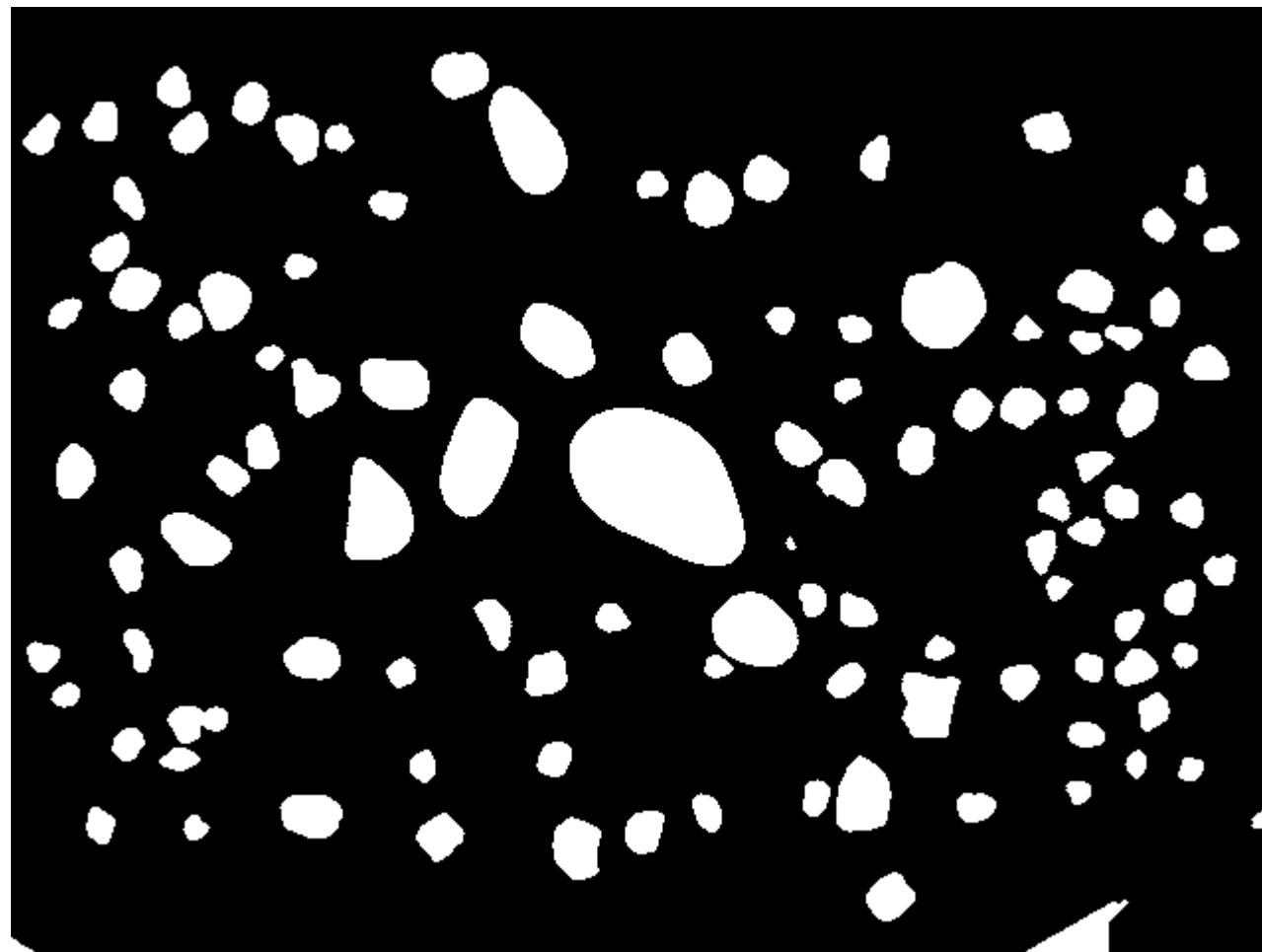
Exemple complet

- Seuillage maximisation d'entropie



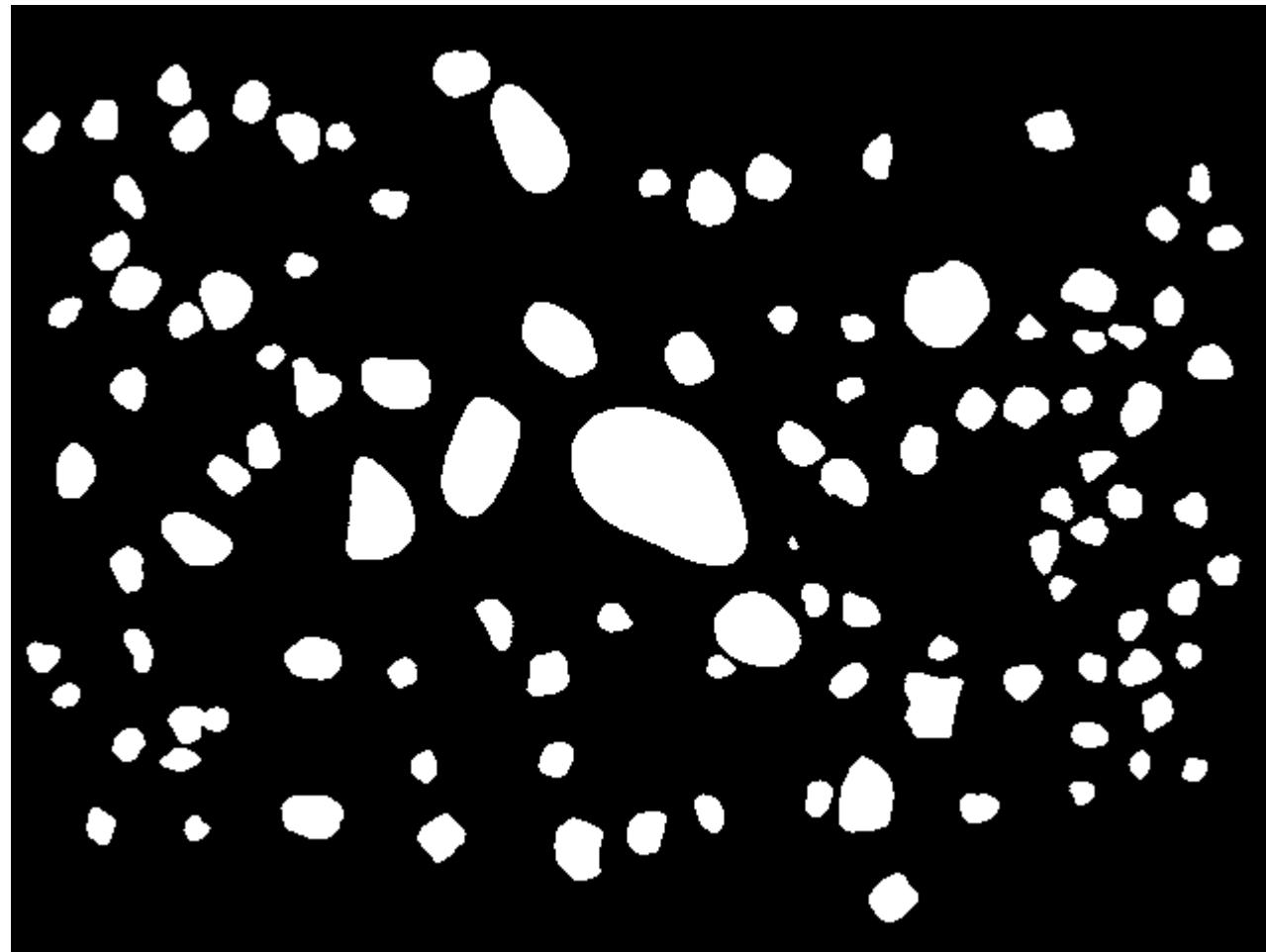
Exemple complet

- Ouverture rayon 1



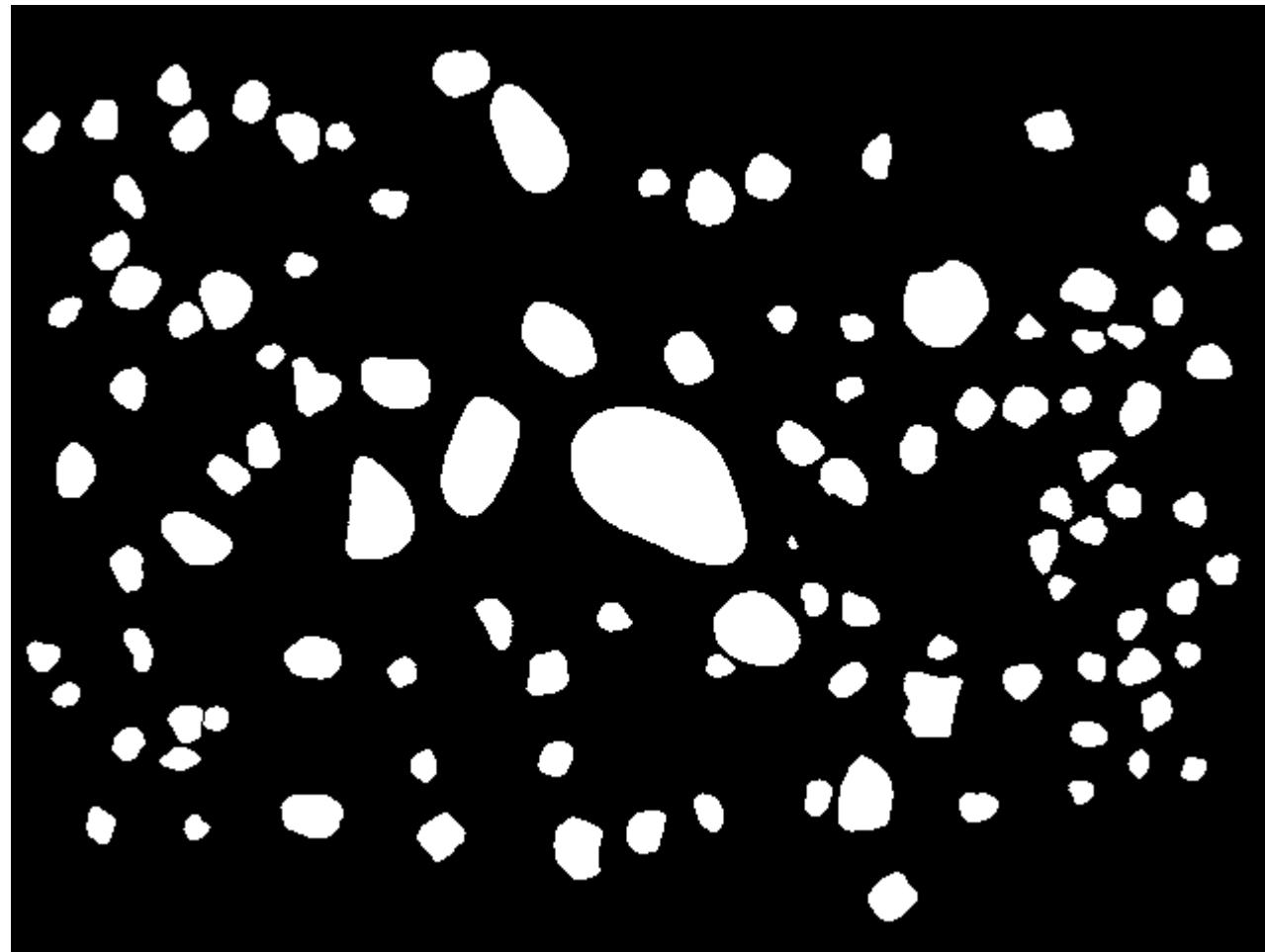
Exemple complet

- Suppression des objets touchant les bords



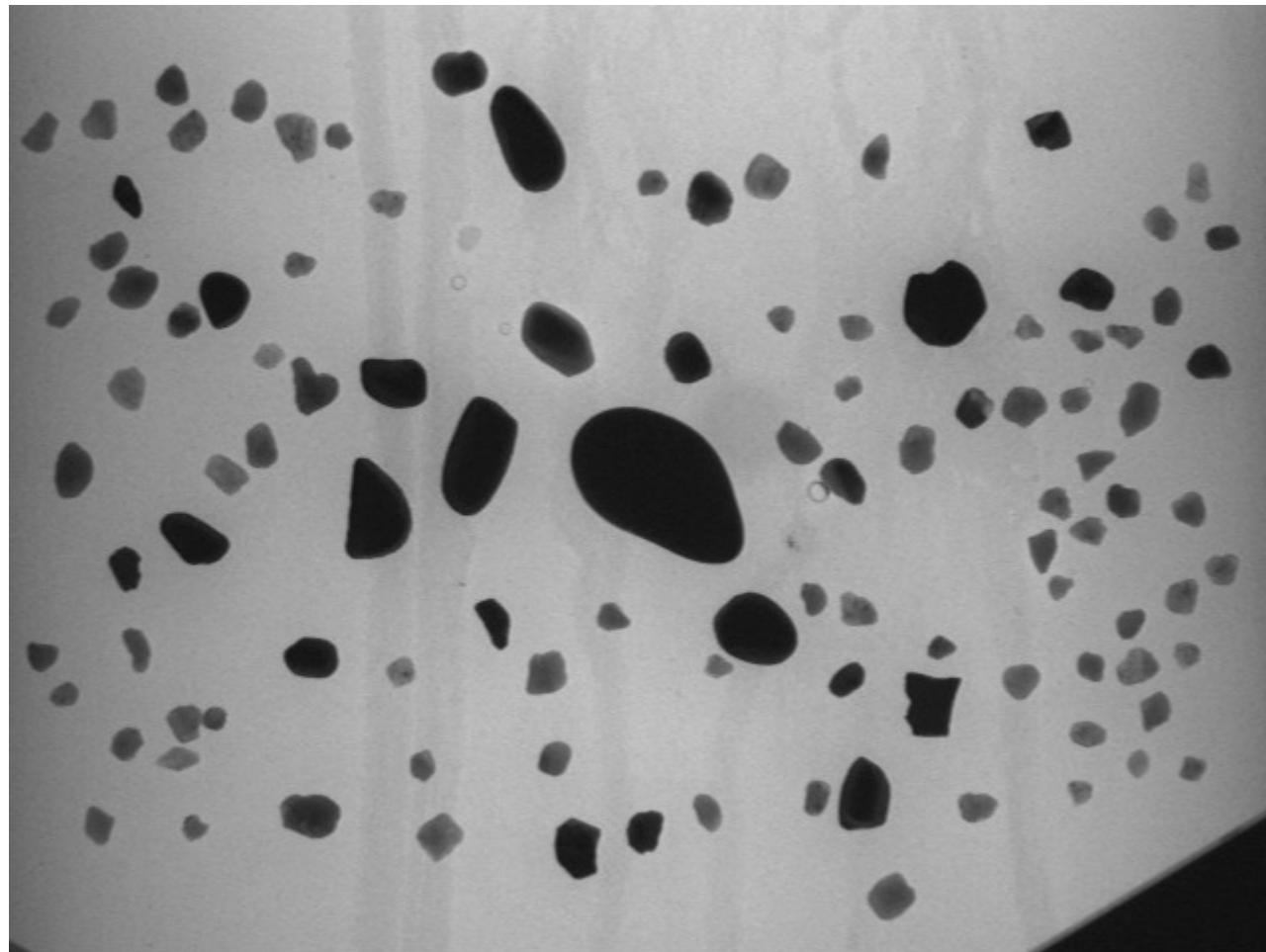
Exemple complet

- Séparation des objets



Exemple complet

- Image originale



Séance 5

4 Mesure de paramètres

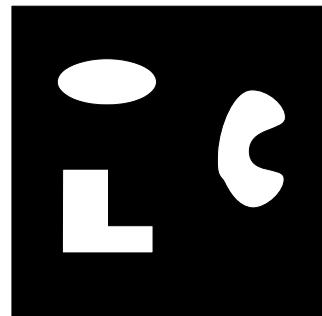
4.1 Post-traitements

4.2 Etiquetage

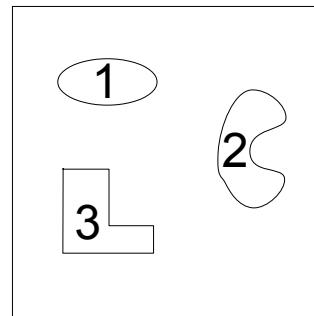
4.3 Mesures

Mesures

- **Etiquetage en composantes connexes**
 - On passe de l'image segmentée à l'image étiquetée



f_s



f_e

f_e Image d'étiquettes

Permet de passer
des classes aux
régions

Matlab
`[fe,n]=bwlabel(fs)`

- Algorithme en 2 passes en mettant à jour une table d'équivalence de niveaux T[256]
- **Mesure sur chaque région**

Algorithme d'étiquetage

Variables

$T[256]$: table d'équivalence

k : étiquette courante

Initialisation

Pour $n=1$ à 256 faire $T[n]=n;$

$k=1;$

Boucle principale

Pour chaque pixel de f tel que $f_s[i,j] \neq 0$ faire

Si les prédécesseurs de $f_s[i,j]$ sont 0 alors $\{f_e[i,j] = k; k++; \}$

Sinon si les prédécesseurs ont l'étiquette ε alors $f_e[i,j]=\varepsilon;$

Sinon

$f_e[i,j]=\min\{ T[f_e[i',j']] \mid (i',j') \in \text{prédécesseurs de } (i,j) \};$

mise à jour de T : les étiquettes des prédécesseurs sont équivalentes à l'étiquette min

Algorithme d'étiquetage (suite)

Mise à jour de la table T

Chaque indice est affecté à son étiquette définitive

Second balayage

Mise à jour de l'image étiquette en fonction de la table T pour avoir la plus petite étiquette d'une classe donnée

Exemple d'étiquetage

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

Image de départ

Premier passage

Mise à jour

T 1 2 3 4 5 6 7 8

1	2	3	4	5	6	7	8

En V4

Table d'équivalence

Exemple d'étiquetage

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

Image de départ

1	1	0	2	0
1	0	3	2	0
1	1	1	0	4
0	0	0	5	4
6	6	0	5	0

Premier passage

1	1	0	1	0
1	0	1	1	0
1	1	1	0	4
0	0	0	4	4
6	6	0	4	0

Mise à jour

T 1 2 3 4 5 6 7 8

En V4

1	2	3	4	5	6	7	8
1	2		4				
		1					

Table d'équivalence

4 Mesure de paramètres

4.1 Post-traitements

4.2 Etiquetage

4.3 Mesures

Mesure des régions

- **Principe**
 - Calculer des paramètres pour chaque région étiquetée
 - Aire, périmètre, centre de gravité, fenêtre circonscrite,...
- **Suivant le paramètre calculé on peut soit :**
 - Utiliser directement f_e (+ éventuellement f ou f_s)
 - Un seul parcours de f_e
 - Mise à jour de tableaux de mesure
 - Chaîner le contour des objets (ie pour le périmètre)
 - Codage de Freeman

```
Matlab  
mes=regionprops(fe,'Area');  
a=[mes.Area];
```

Mesures de régions



Num	Ox	Oy	Label	X0	Y0	TX	TY	Aire	Xg	Yg	Perimetre	Long	Larg	Angle	Dmax	Dequ	Elong	Compa	Excent	I1_AxePr	I2_AxePr
0	74	30	1	74	30	17	21	278	82,17	40,79	60,87	21,75	17,47	80,02	21,92	18,81	1,25	0,94	0,05	28,06	17,99
1	112	38	2	112	38	19	22	322	121,07	48,55	65,7	22,4	19,14	-67,63	22,62	20,25	1,17	0,94	0,04	31,2	21,17
2	36	48	3	36	48	19	20	303	45,64	58,03	64,87	22,37	19,68	-69,39	22,84	19,64	1,14	0,9	0,03	29,8	20,49
3	80	52	4	80	52	20	23	315	90,08	63,76	66,77	24,06	18,22	-52,64	23,94	20,03	1,32	0,89	0,09	34,47	18,55
4	6	54	5	6	54	19	21	250	15,62	64,8	61,94	22,5	16,11	-54,51	22,84	17,84	1,4	0,82	0,19	33,17	12,97

Différents types de mesures

- Mesures statistiques

- moments 1er ordre, 2nd ordre, histogramme,...
 - on a besoin de $f[i,j]$ et de $f_e[i,j]$
- Peuvent être calculés en 1 passe

- Mesures géométriques

- aire, périmètre, centre de gravité, diamètres, paramètres de forme,...
- Certains peuvent être calculés en 1 balayage (aire, cdg,...)
- Certains nécessitent le chaînage du contour de l'objet (périmètre)

Mesures géométriques 1

- **Aire ('Area')**

$S(R) = K$ = nombre de pixels de la région R

- **Périmètre ('Perimeter')**

$$\begin{aligned} P(R) &= \text{somme des distances entre pixels consutifs} \\ &= \text{nb_codes_pairs} + \sqrt{2} \text{ nb_codes_impairs} \end{aligned}$$

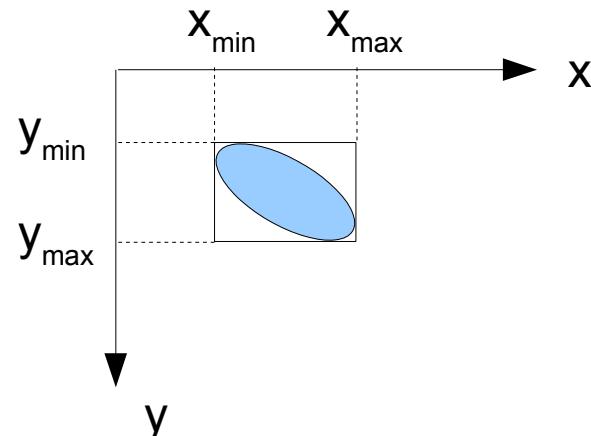
- **Centre de gravité ('Centroid')**

si (x_k, y_k) = ensemble des pixels de la région R

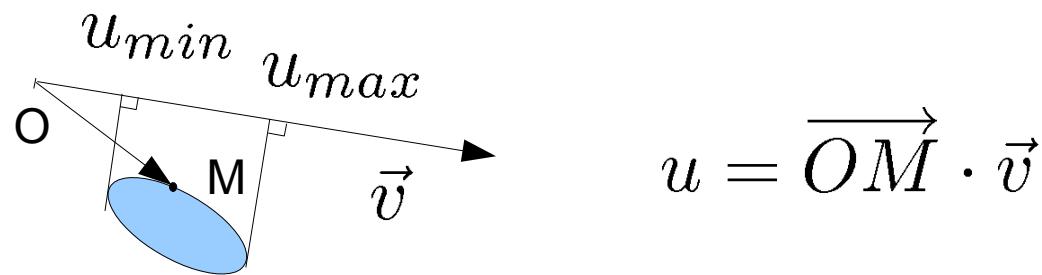
$$x_G = \frac{1}{K} \sum_{k=1}^K x_k \quad y_G = \frac{1}{K} \sum_{k=1}^K y_k$$

Mesures géométriques 2

- Fenêtre circonscrite ('BoundingBox')



- Diamètre dans une direction



$$u = \overrightarrow{OM} \cdot \vec{v}$$

Mesures géométriques 3

- Diamètre max

$$\begin{aligned} D_{max} &= \max\{d(M, N) | M, N \in R\} \\ &= \max\{d(M, N) | M, N \in Fr(R)\} \end{aligned}$$

- Diamètre équivalent ('EquivDiameter')

$$D_{equ} = \sqrt{\frac{4S(R)}{\pi}}$$

- Compacité (circularité)

$$\text{Comp}(R) = \frac{4\pi S(R)}{P^2(R)}$$

=1 si cercle
=0 si découpé ou allongé

Mesures géométriques 4

- Directions principales d'inertie
 - Vecteur propre de la matrice d'inertie

Matrice d'inertie : $\begin{pmatrix} a & c \\ c & b \end{pmatrix}$

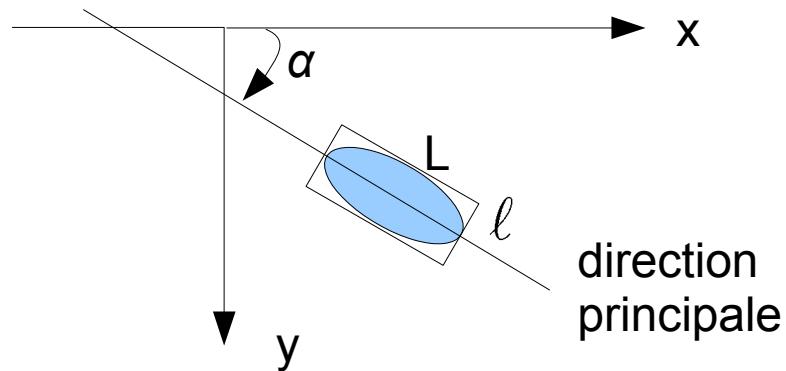
$$a = \frac{1}{K} \sum_{k=1}^K (x_k - x_G)^2$$
$$c = \frac{1}{K} \sum_{k=1}^K (x_k - x_G)(y_k - y_G)$$
$$b = \frac{1}{K} \sum_{k=1}^K (y_k - y_G)^2$$

Angle de la direction principale avec l'axe x :

$$\alpha = \tan^{-1} \frac{b-a+\sqrt{(b-a)^2+4c^2}}{2c}$$

Mesures géométriques 5

- Rectangle d'encadrement
 - Diamètres suivant les directions principales
 - Longueur et largeur



- Elongation

$$\text{el}(R) = \frac{L}{\ell}$$

Mesures sous Matlab

- **Exemple**

```
%% Mesures  
% fe : image labelisée ou image binaire  
mes=regionprops(fe,'basic'); % Ensemble de mesures de base  
a=[mes.Area];  
c = cat(1, s.Centroid);  
...
```

- **Propriétés disponibles (extrait)**

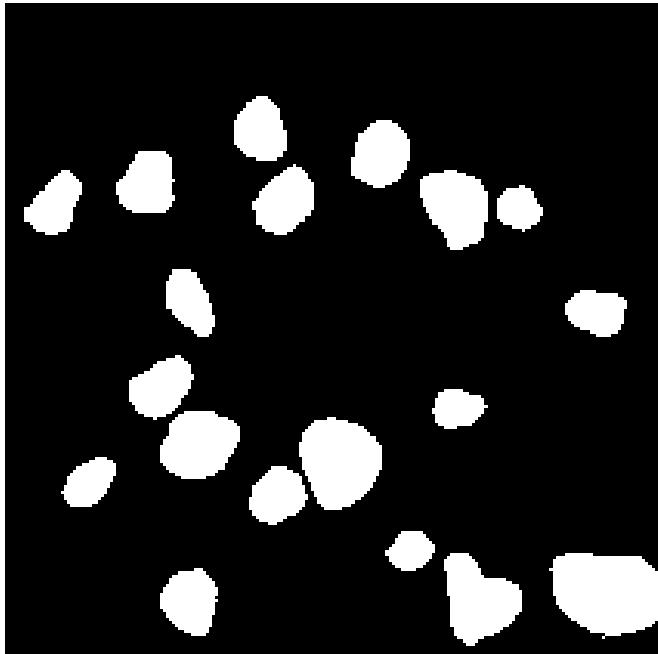
Area	Orientation	Image	MaxIntensity
BoundingBox	Eccentricity	PixelList	MinIntensity
Centroid	MajorAxisLength		MeanIntensity
Perimeter	MinorAxisLength		WeightedCentroid
EquivDiameter	Extent		

Mesures de régions



Num	Ox	Oy	Label	X0	Y0	TX	TY	Aire	Xg	Yg	Perimetre	Long	Larg	Angle	Dmax	Dequ	Elong	Compa	Excent	I1_AxePr	I2_AxePr
0	74	30	1	74	30	17	21	278	82,17	40,79	60,87	21,75	17,47	80,02	21,92	18,81	1,25	0,94	0,05	28,06	17,99
1	112	38	2	112	38	19	22	322	121,07	48,55	65,7	22,4	19,14	-67,63	22,62	20,25	1,17	0,94	0,04	31,2	21,17
2	36	48	3	36	48	19	20	303	45,64	58,03	64,87	22,37	19,68	-69,39	22,84	19,64	1,14	0,9	0,03	29,8	20,49
3	80	52	4	80	52	20	23	315	90,08	63,76	66,77	24,06	18,22	-52,64	23,94	20,03	1,32	0,89	0,09	34,47	18,55
4	6	54	5	6	54	19	21	250	15,62	64,8	61,94	22,5	16,11	-54,51	22,84	17,84	1,4	0,82	0,19	33,17	12,97

Filtrage suivant des caractéristiques



Aire >400

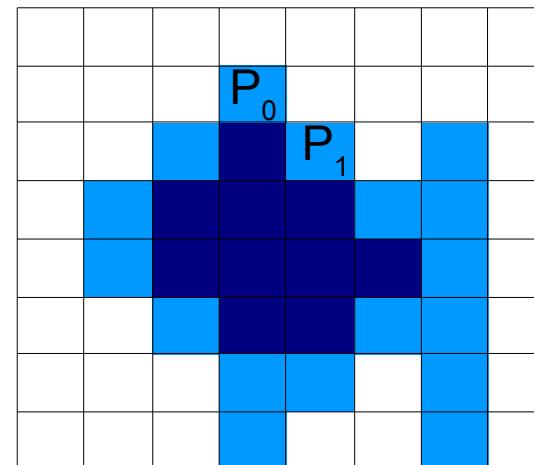
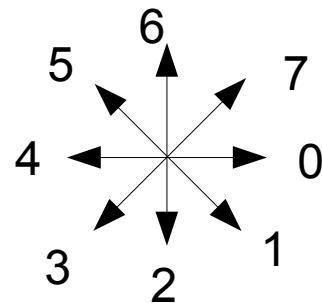


Elongation >1.3

```
Matalb  
%% Sélection d'objets  
idx = find(a > 400);  
fe2 = ismember(fe,idx);  
figure;imshow(fe2);
```

Codage de Freeman

- Contour de la région = suite de codes



117222226533655677

Codage de freeman

- **Algorithme**

Recherche du premier point P_0 de contours (balayage électronique)

Si tous les voisins de P_0 sont dans le fond, alors P_0 est isolé => fin

Soient

s : point courant, t : point suivant

dir : direction désignant le point suivant à partir de s

dir_{init} : direction à partir de laquelle on est sûr de trouver le fond

$\text{fini}=\text{faux};$

$t=\text{point suivant à partir de } P_0 \text{ dans la direction } 0;$

Tant que non fini

Tant que $f[t]==0$

dir=(dir+1)%8;

t=s+vect[dir];

dir_{init}=(dir+5)%8;

Si t est le deuxième point alors $P_1=t$;

Sinon, si ($s==P_0$) && ($t==P_1$) alors fini=vrai;

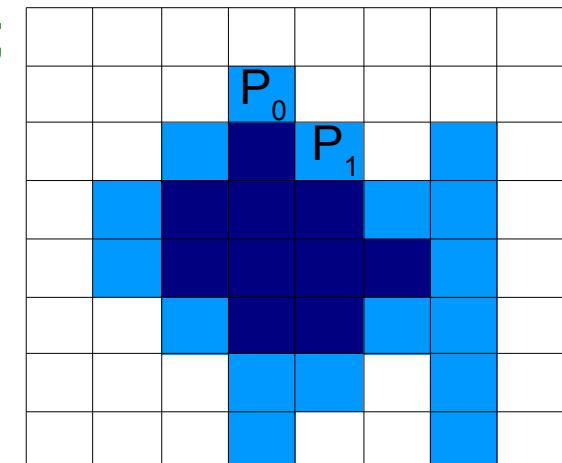
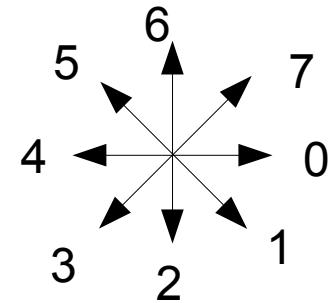
s=t;

t=t+vect[dir_{init}];

ch[k]=dir;

k=k+1;

dir=dir_{init};



117222226533655677

Exercice 5

- Segmenter l'image "Cailloux"
 - Inclure la correction du fond et les post-traitements
 - Ajuster les paramètres pour s'approcher au plus près de la vérité terrain
- Calculer la distribution de taille
 - Calculer le diamètre équivalent des objets
 - Tracer la distribution de taille des cailloux (histogramme des diamètres)

Séance 6

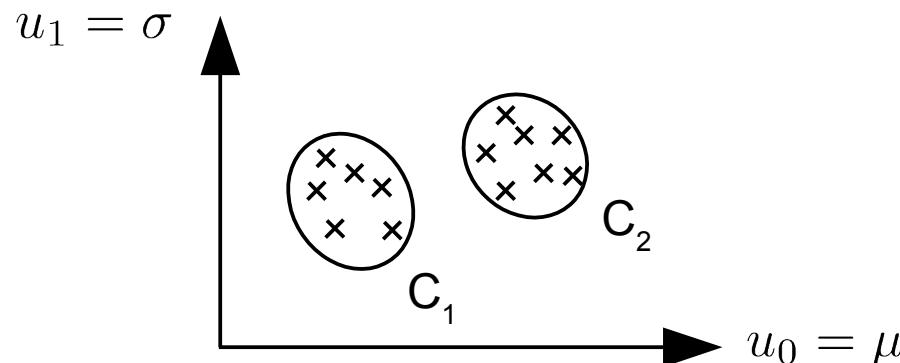
Plan

1. Introduction
2. Pré-traitements
3. Seuillage automatique
4. Mesure de paramètres
5. Segmentation par classification
6. Détection de contours
7. Points d'intérêts et mise en correspondance

Segmentation par classification

- **Principe**

- On dispose de plusieurs attributs par pixels
 - exemple : R, V, B, μ , σ local, attributs de texture
- Chaque pixel = individu avec attributs
= vecteur u_0, u_1, \dots, u_{q-1}
- On réalise une classification dans l'espace des attributs
 - exemple pour 2 attributs : $u_0 = \mu$ et $u_1 = \sigma$



Exemple avec une image couleur

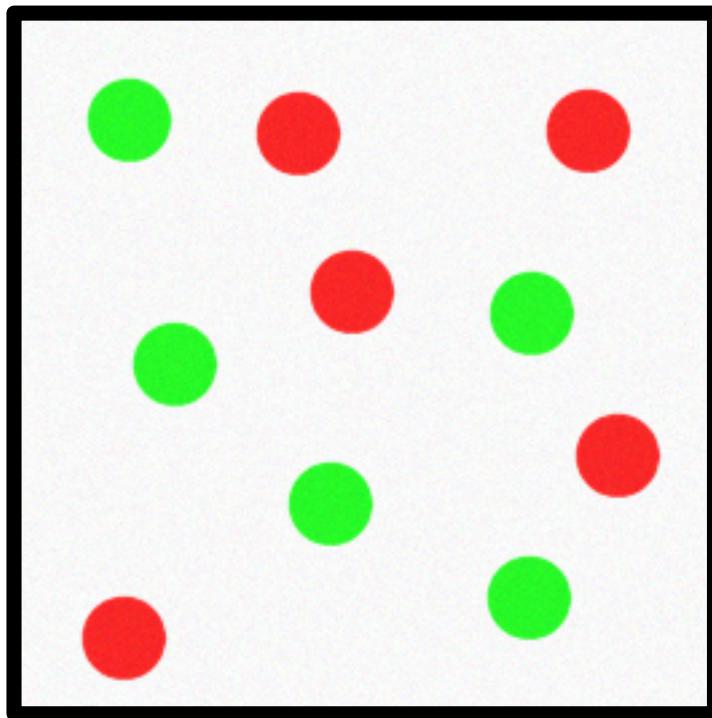
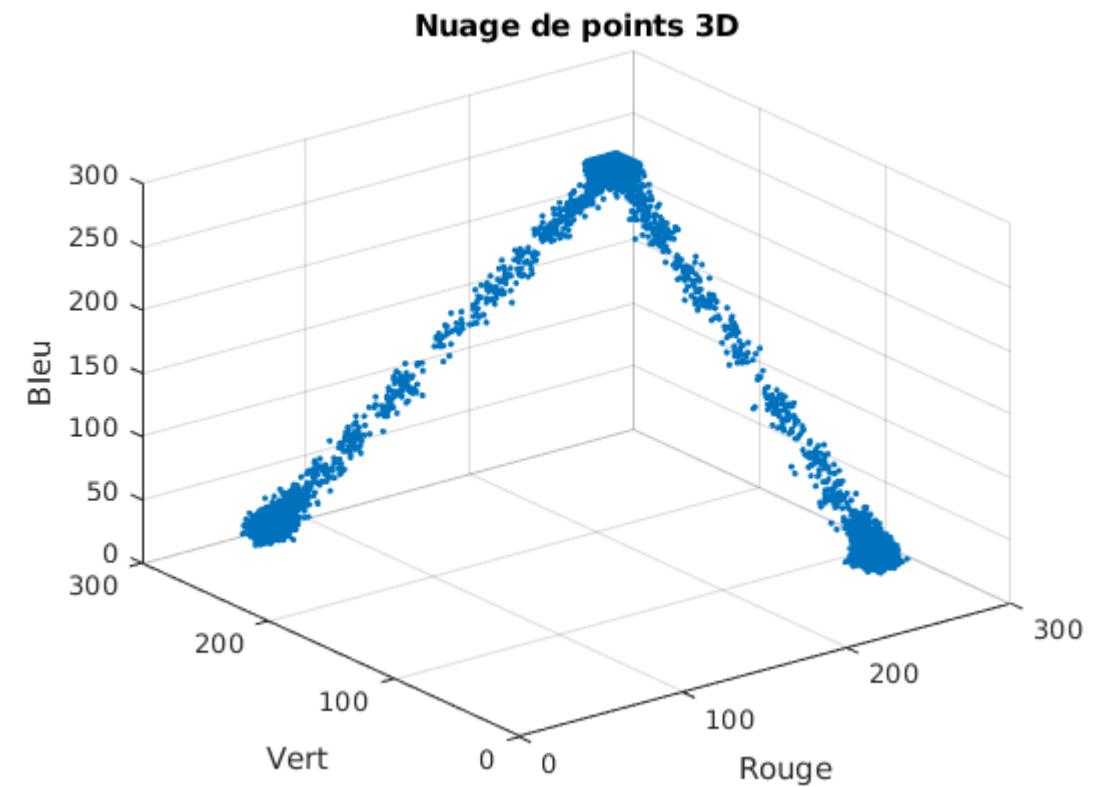


Image couleur originale



Approche « marginale »

- On pourrait être tenté de traiter les canaux séparément :

Si $f(m,n).B > 128$ alors $f_e(m,n) = \text{fond}$
sinon si $f(m,n).V > f(m,n).R$ alors $f_e(m,n) = c_{\text{vert}}$
sinon $f_e(m,n) = c_{\text{rouge}}$

- Inconvénients
 - Difficilement généralisable à d'autres couleurs
 - Arbitraire pour le choix des conditions
 - Difficilement applicable pour plus de 3 attributs
- Il vaut mieux utiliser les méthodes de classification automatique

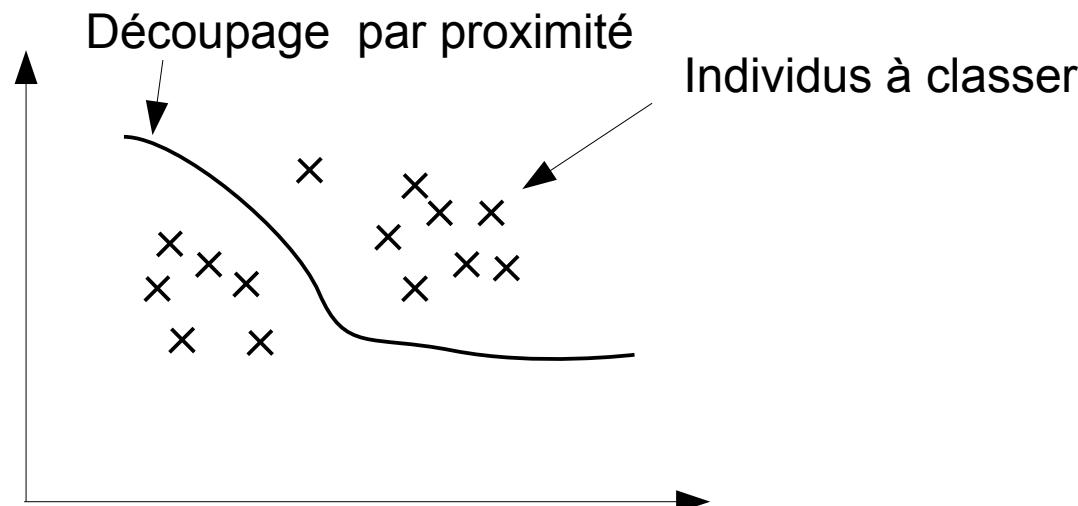
Deux types de classification

- Classification non supervisée
- Classification supervisée

Classification non supervisée

- **Principe**

- On n'utilise pas d'exemples d'individus de chaque classe
- On essaye de séparer les individus d'après la forme des nuages de points (cf. seuillage automatique)

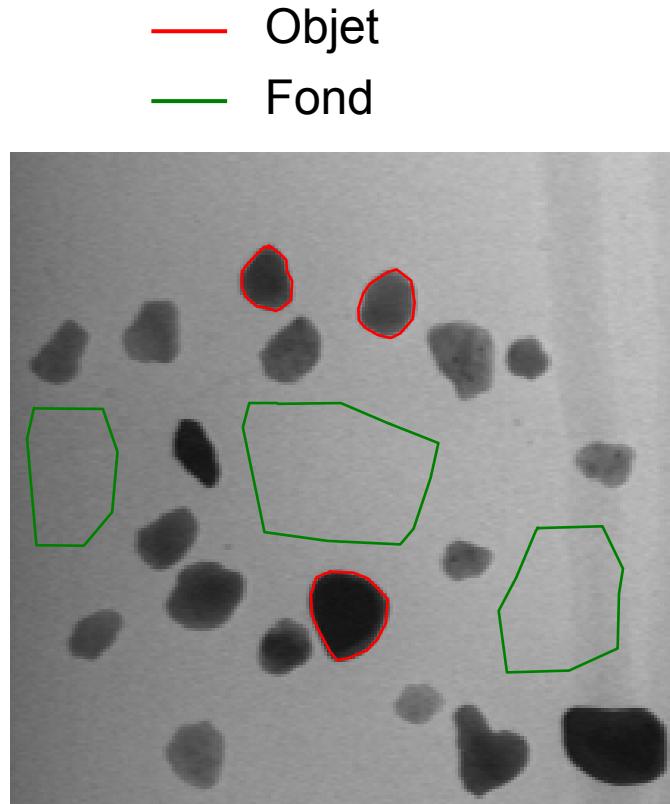


Classification supervisée

- **Principe**
 - On a des exemples d'individus de chaque classe
→ ensemble d'apprentissage
 - On veut classer des individus "inconnus"
→ ensemble de test

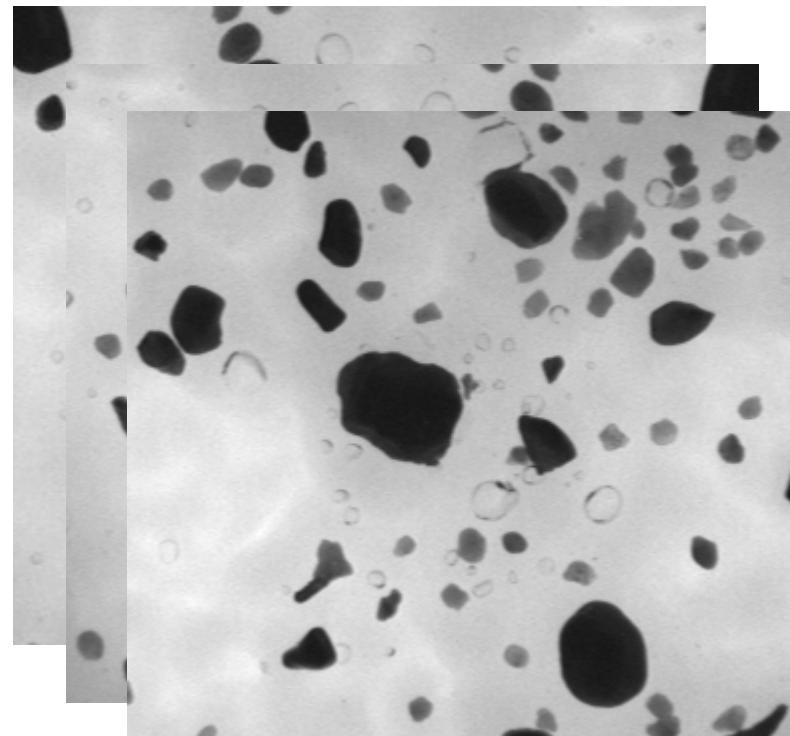
Exemple

Ensemble d'apprentissage



Segmentation de
référence fournie par
un expert

Ensemble de test



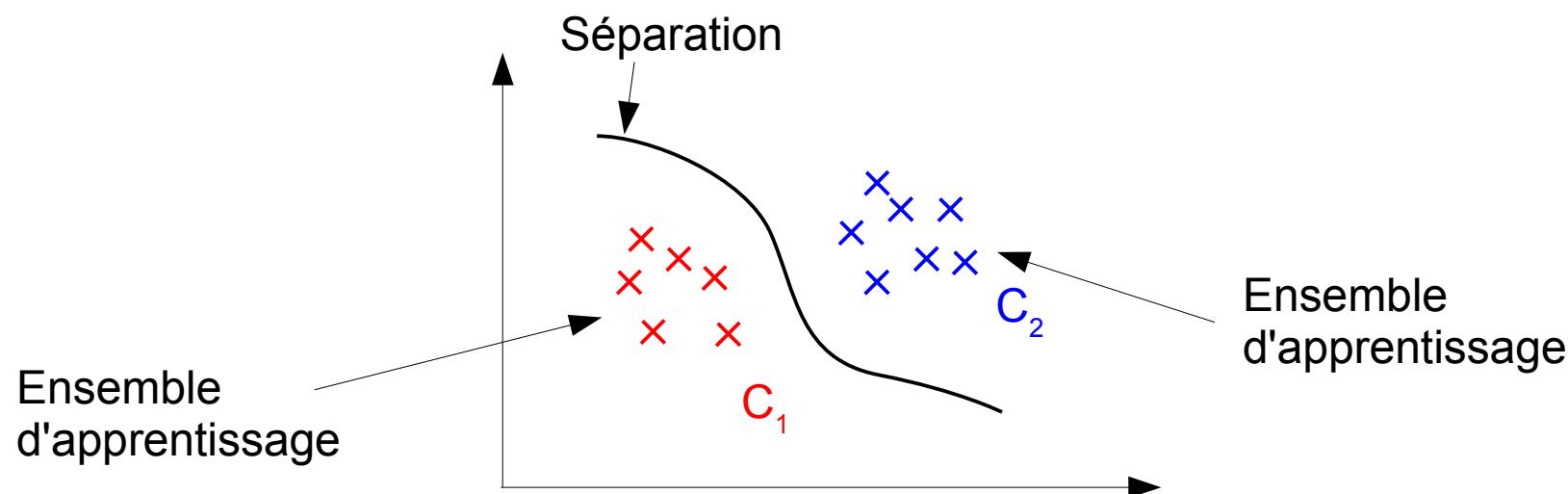
Série d'image à
segmenter

Deux étapes

- Etape d'apprentissage
- Etape de classification

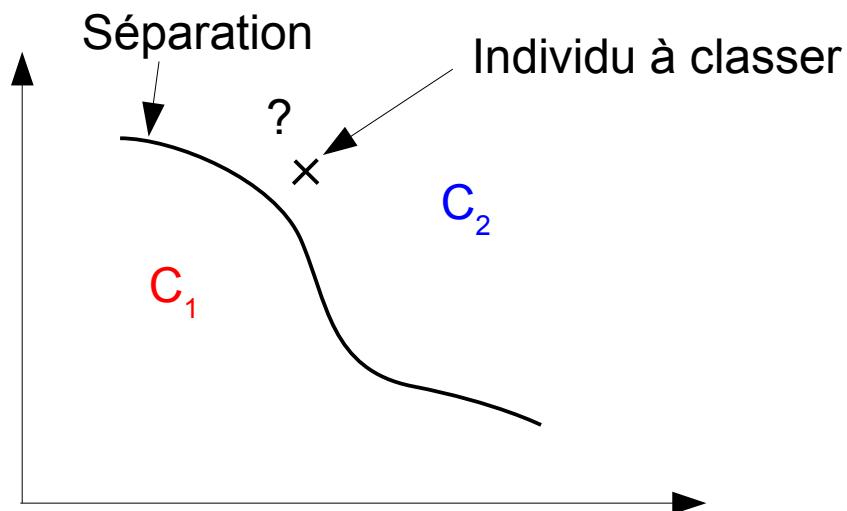
Étape d'apprentissage

- On calcule les attributs pour les individus d'apprentissage
- Recherche d'une séparation dans l'espace des attributs



Étape de classification

- Individus inconnu
- Affectation d'une des classes



Choix des attributs

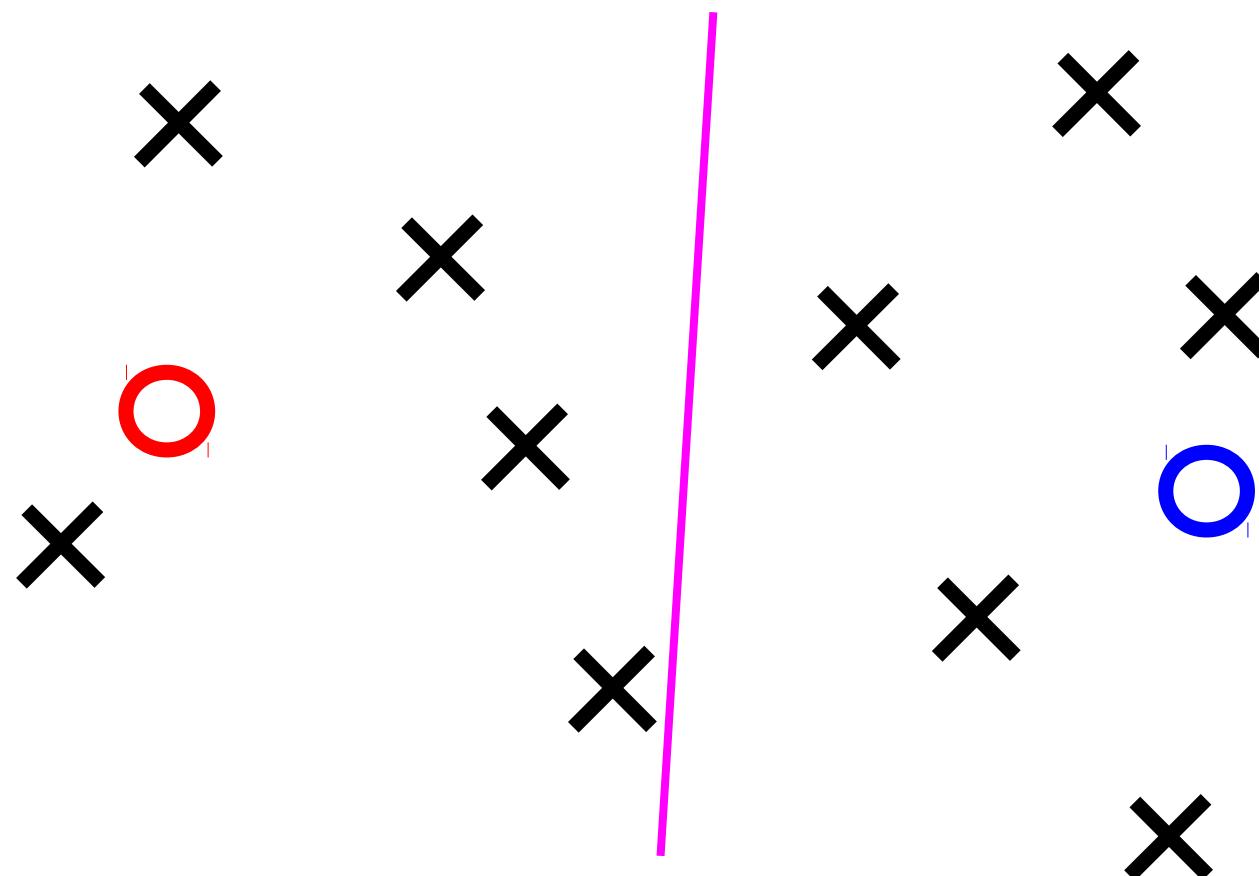
- **Point Délicat**
 - Trouver des attributs discriminants vis à vis des classes
 - Éviter les attributs qui donnent la même information (redondance)
 - Rechercher de nouveaux attributs en cas d'échec
- **Sélection d'attributs**
 - Méthodes de sélection d'attributs
 - Méthodes de réduction de dimension

5.1 Nuées dynamiques

- K-means ou K-moyennes
 - Très utilisé
 - Efficace en grandes dimensions
- Non supervisée
 - 1 paramètre : nombre de classes
- Principe
 - On modélise chaque classe par un centre dans l'espace des attributs
 - La classe attribuée à un point est celle correspondant au centre le plus proche

Principe

- Espace des attributs

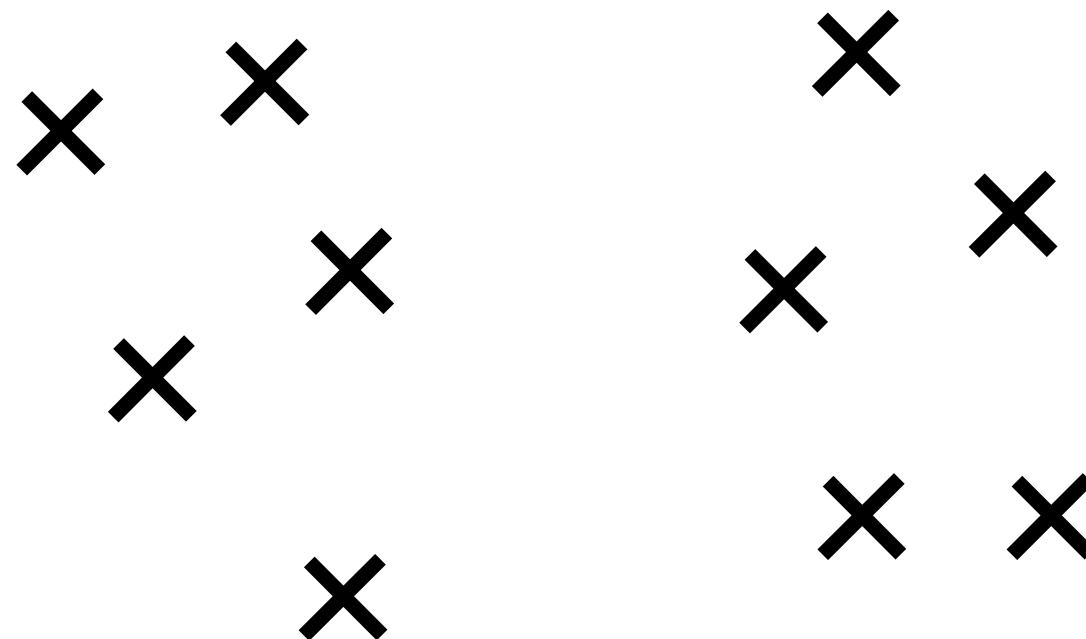


Algorithme

- Fixer *a priori* le centre des classes (tirage aléatoire)
- Classer les individus (centre le plus proche)
- Recalculer les nouveaux centres (barycentre)
- S'arrêter lorsqu'il n'y a plus d'évolution

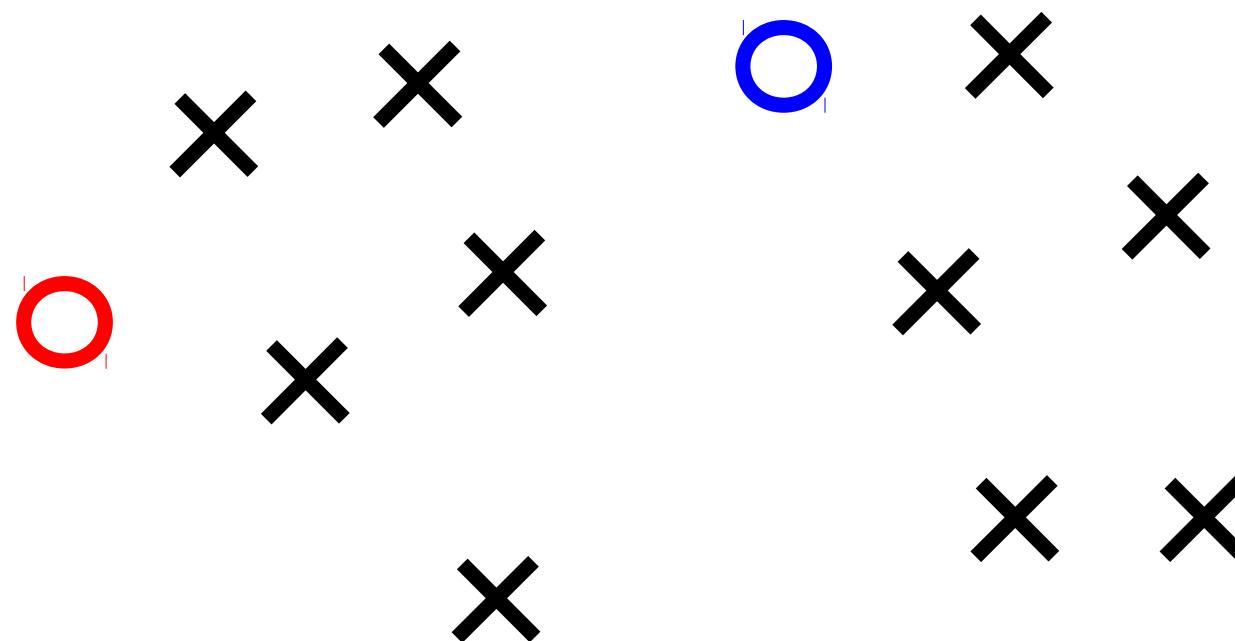
Illustration

- Etat initial



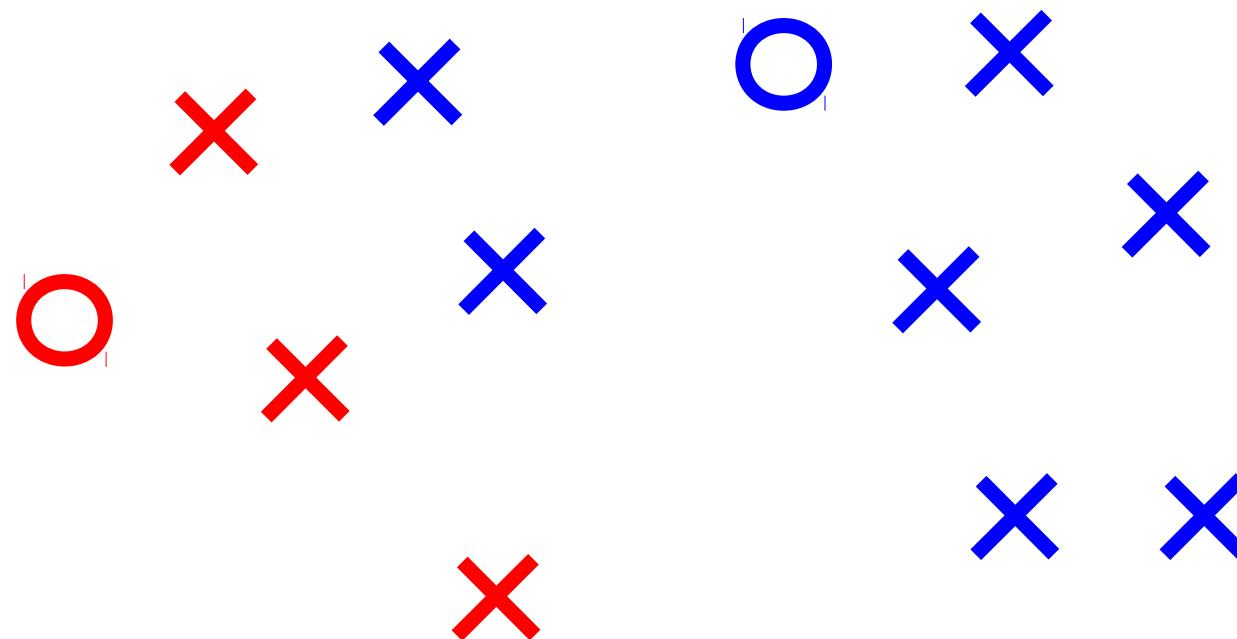
Illustration

- Tirage aléatoire des centres



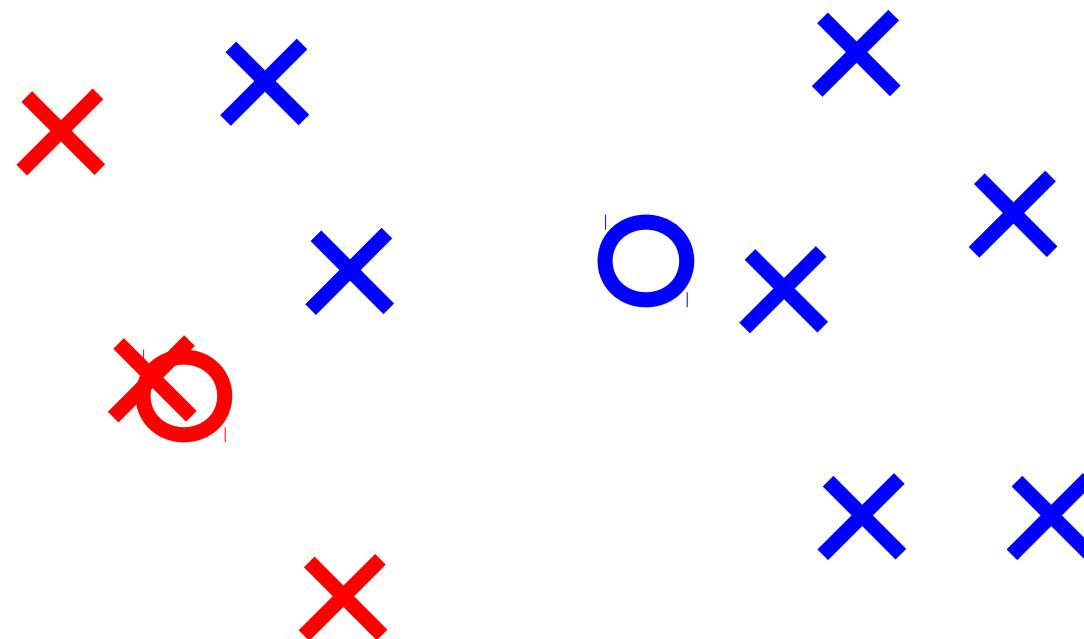
Illustration

- Affectation au centre le plus proche



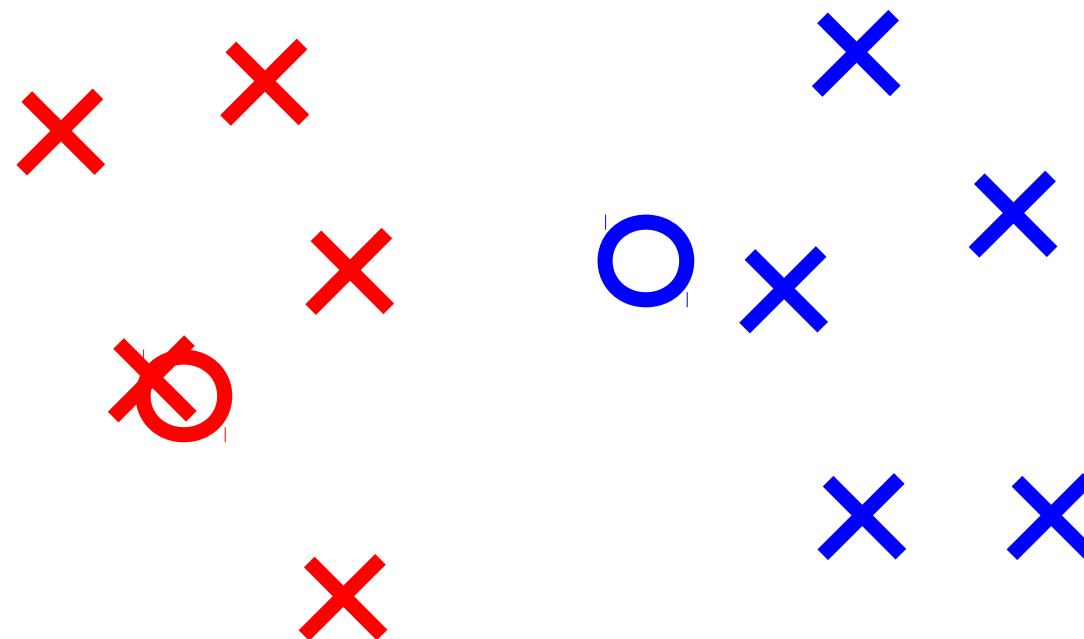
Illustration

- Calcul des nouveaux centres



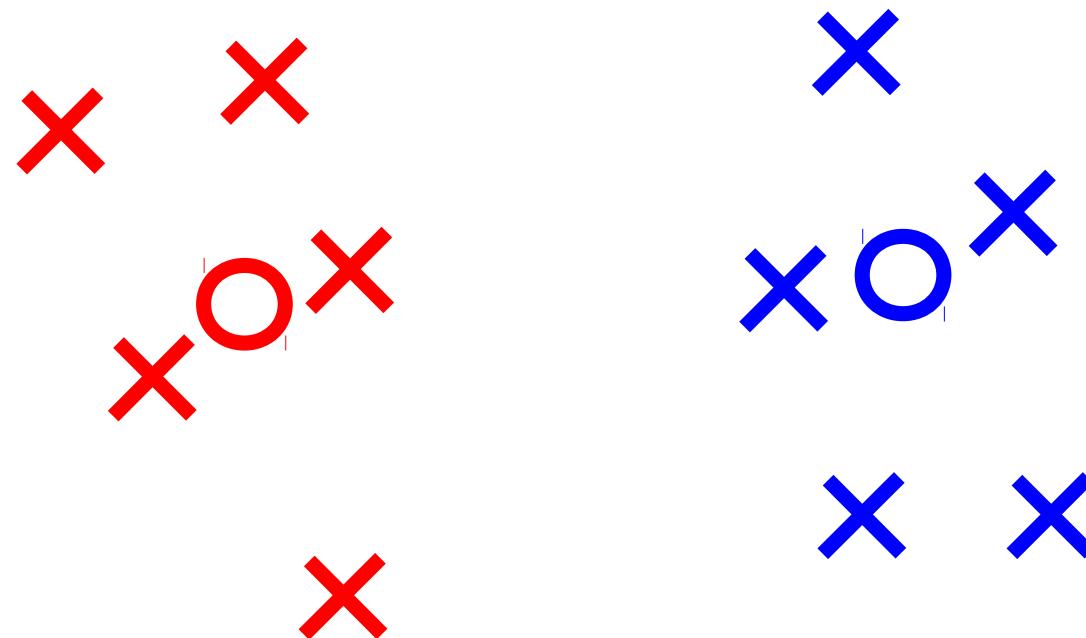
Illustration

- Affectation au centre le plus proche



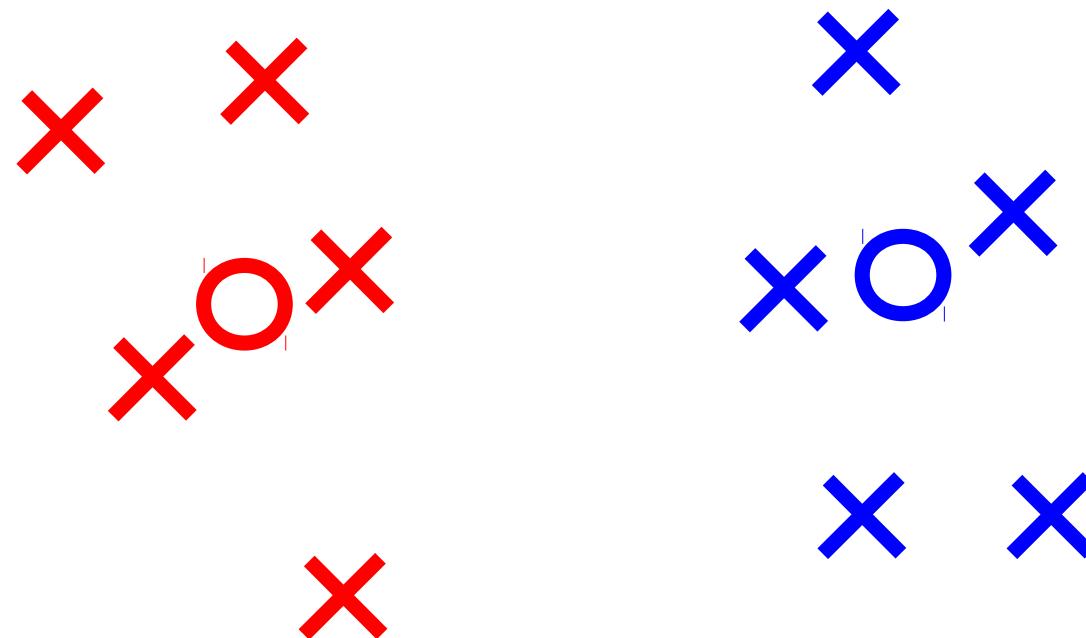
Illustration

- Calcul des nouveaux centres



Illustration

- Plus d'évolution : fin de l'algorithme



Remarques

- Il faut connaître le nombre de classes
 - On peut surévaluer ce nombre
- Le résultat dépend du tirage des centres
 - On peut avoir un résultat différent d'une exécution à l'autre
 - Autres versions de l'algorithme
 - faire plusieurs essais
 - retenir les centres qui maximisent la variance interclasse
- Maximisation de la variance inter-classe
 - En 1D donne le même résultat que la méthode de seuillage

Exemple 1 : image couleur

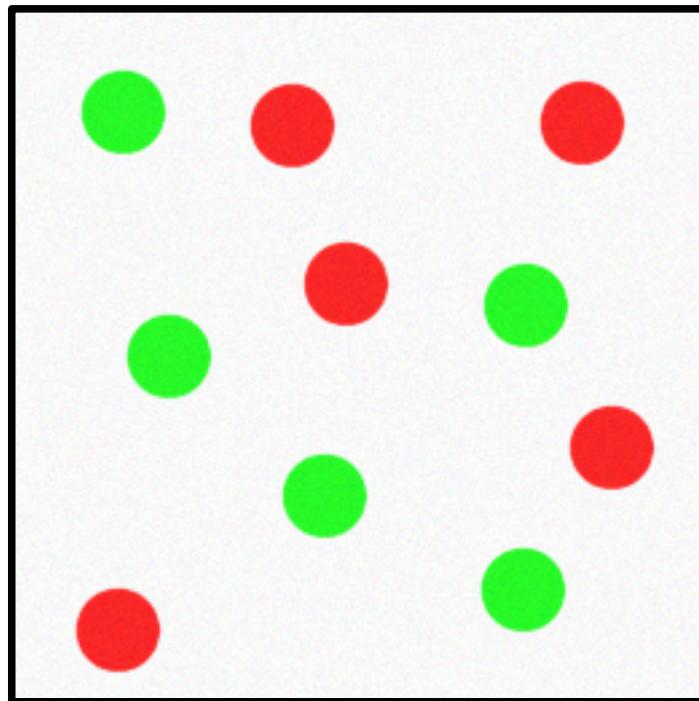
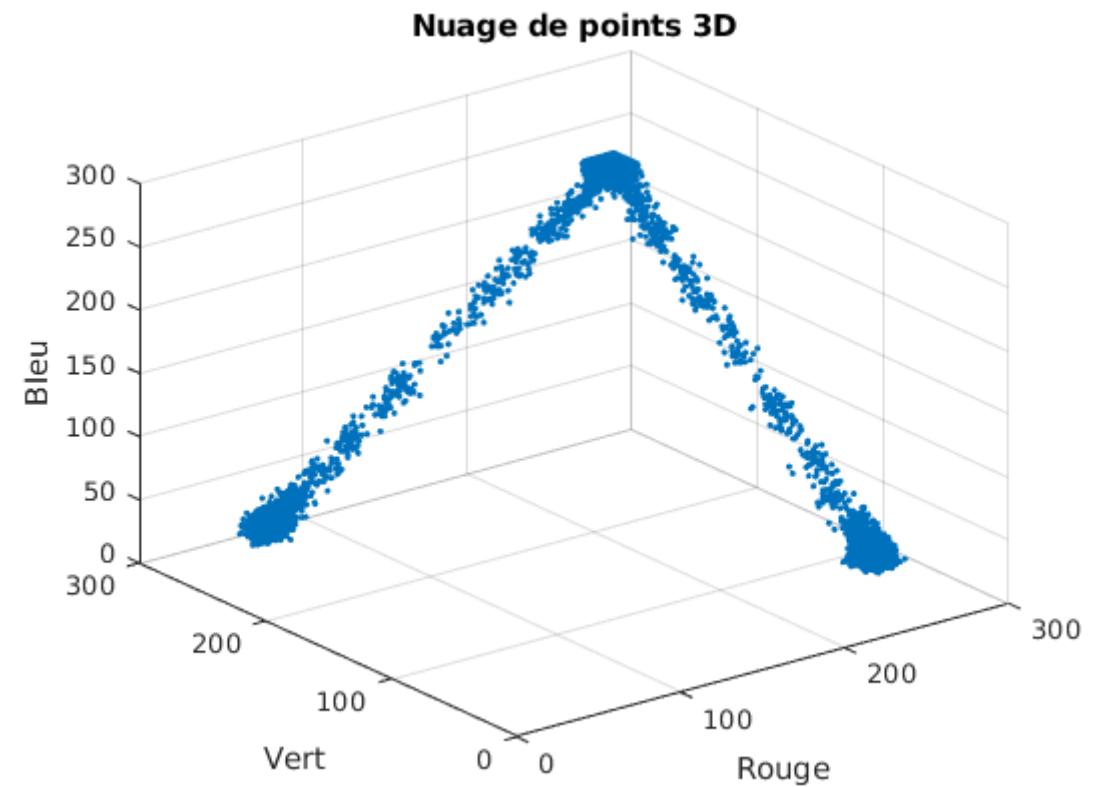


Image couleur originale



Segmentation par nuées dynamiques

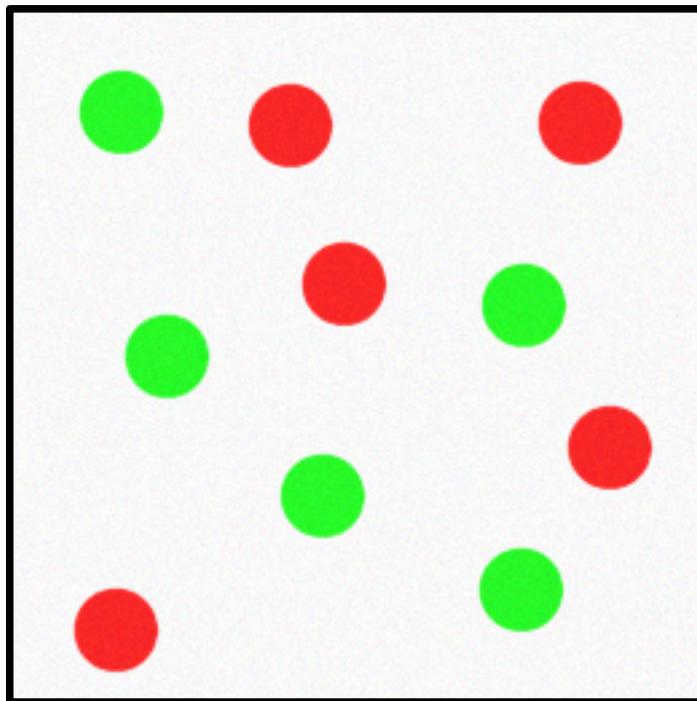


Image couleur originale

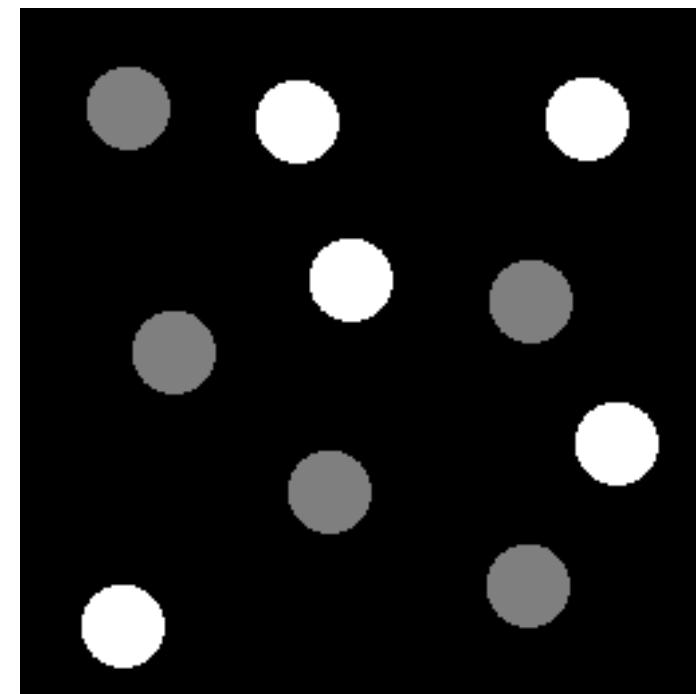
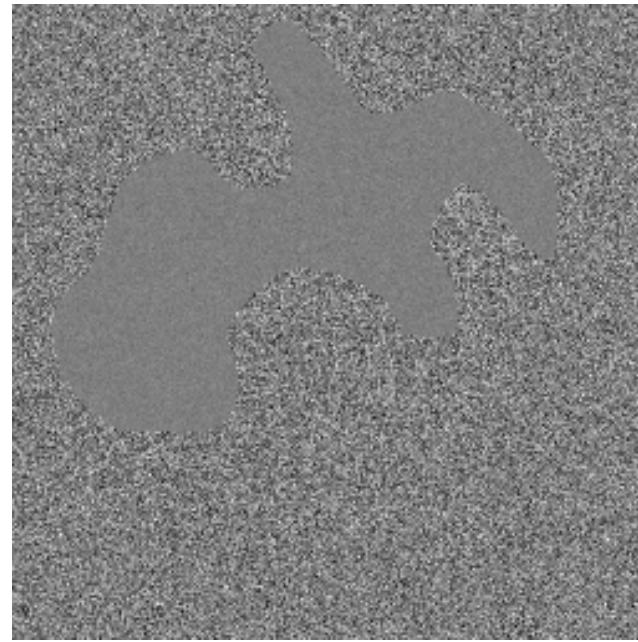
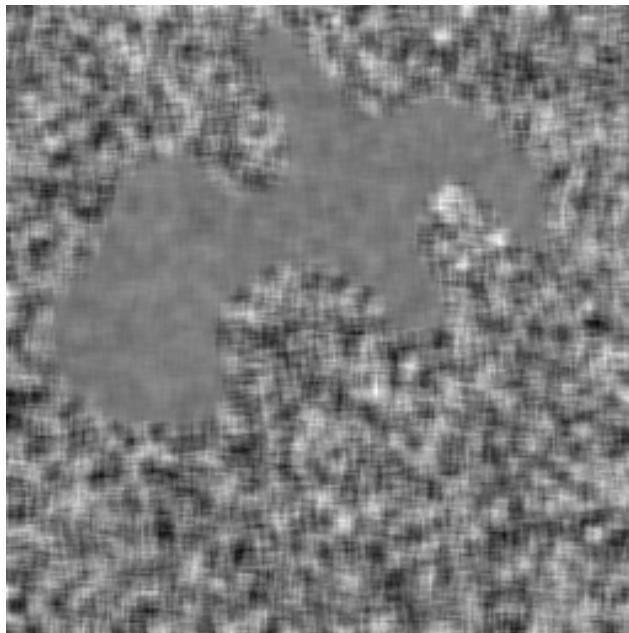
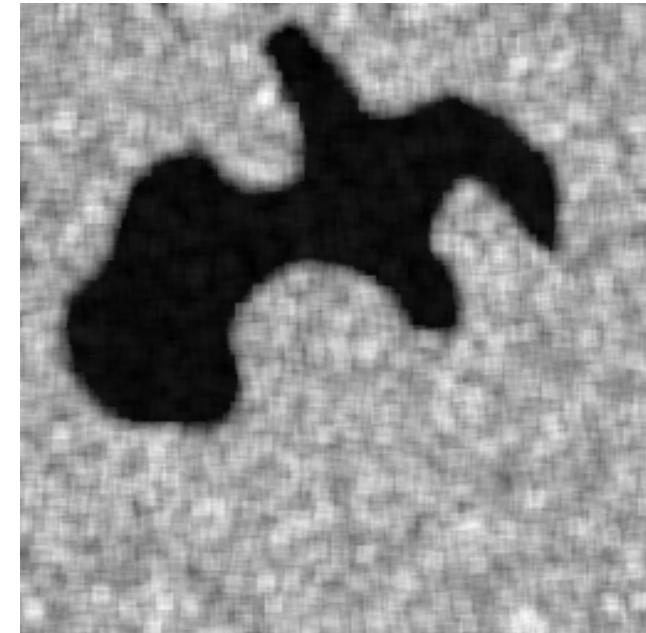


Image segmentée en 3 classes

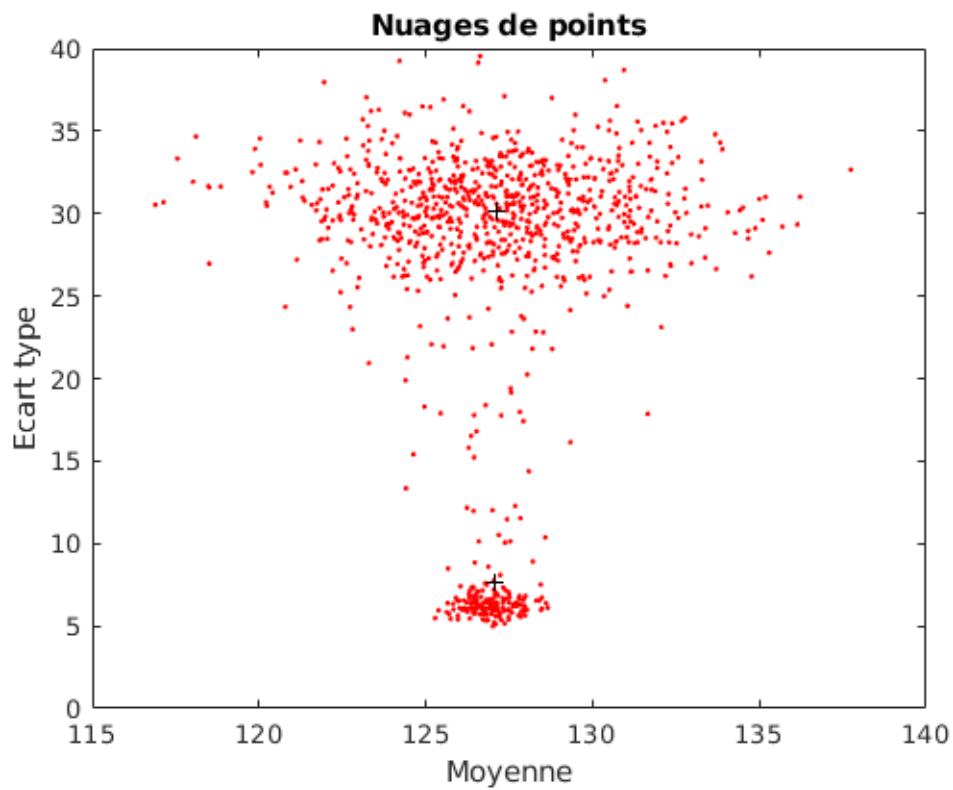
Exemple 2 : texture



Ecart type local



Exemple 2 : texture



Segmentation par k-means

5.2 k plus proches voisins

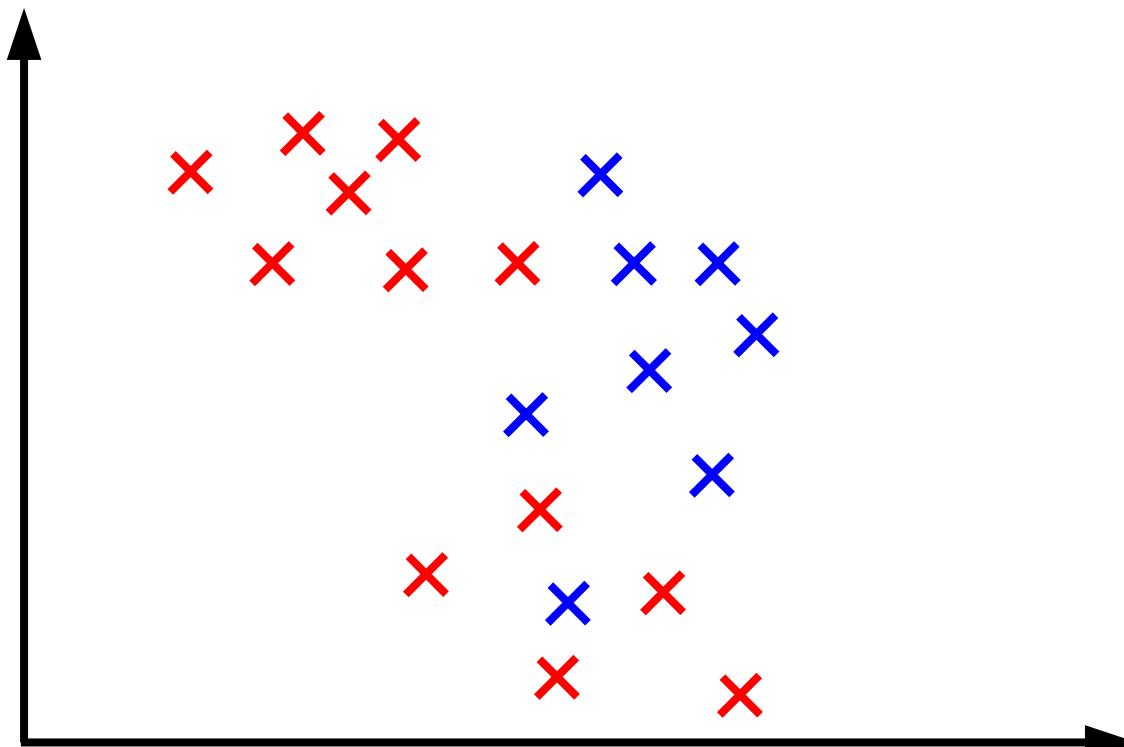
- Méthode supervisée
 - Ensemble d'apprentissage avec des classes connues
 - k est une valeur fixée à l'avance (1, 3,...)
- Principe
 - Examen des k voisins de l'ensemble d'apprentissage
 - Affectation de la classe la plus représentée

Étape d'apprentissage (1/2)

- Calcul des attributs pour tous les points de l'ensemble d'apprentissage
- Chaque classe est modélisée par l'ensemble des échantillons d'apprentissage
- Pas de calcul explicite de la séparation entre les classes

Étape d'apprentissage (2/2)

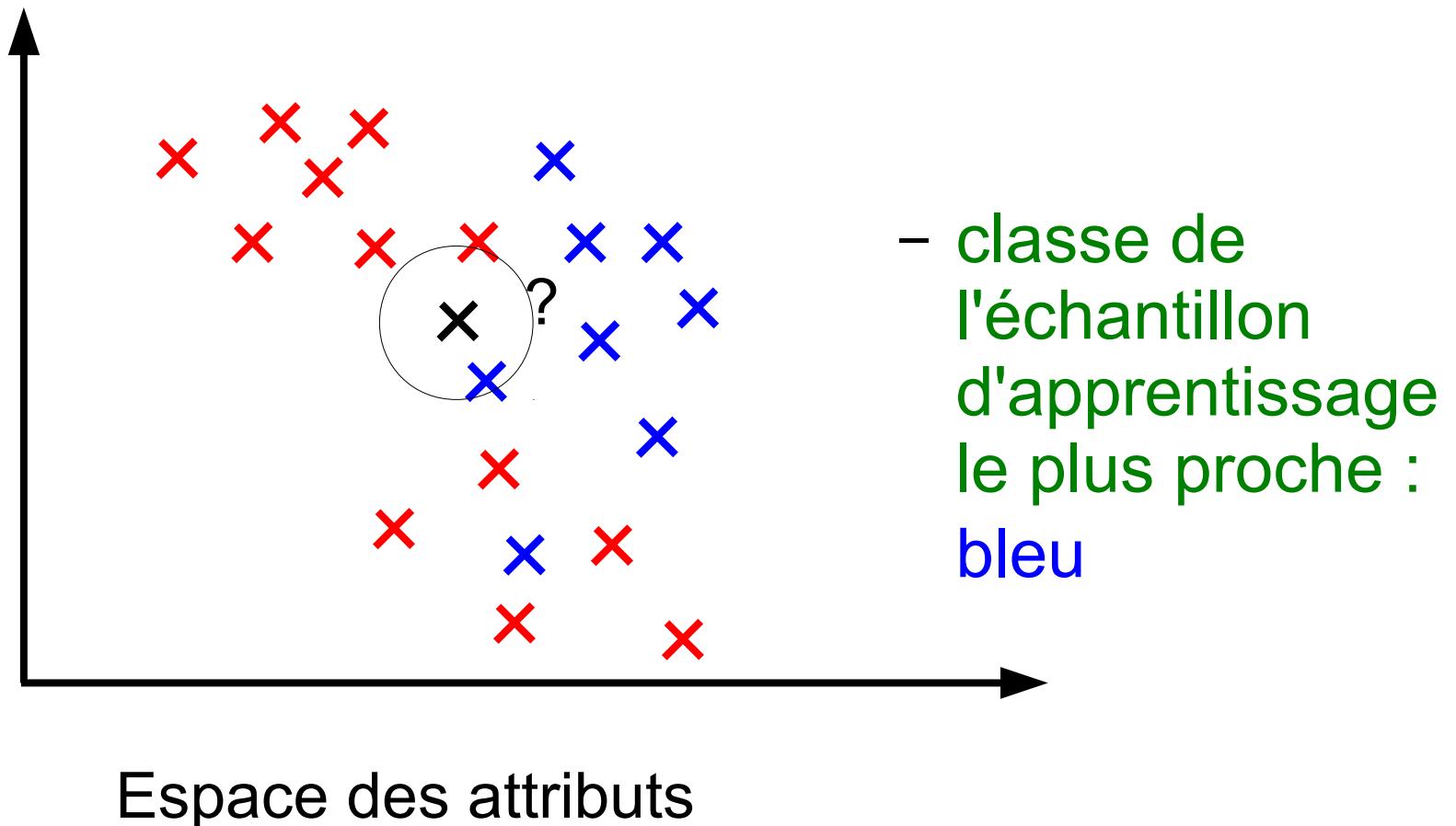
- Projection des individus d'apprentissage



Espace des attributs

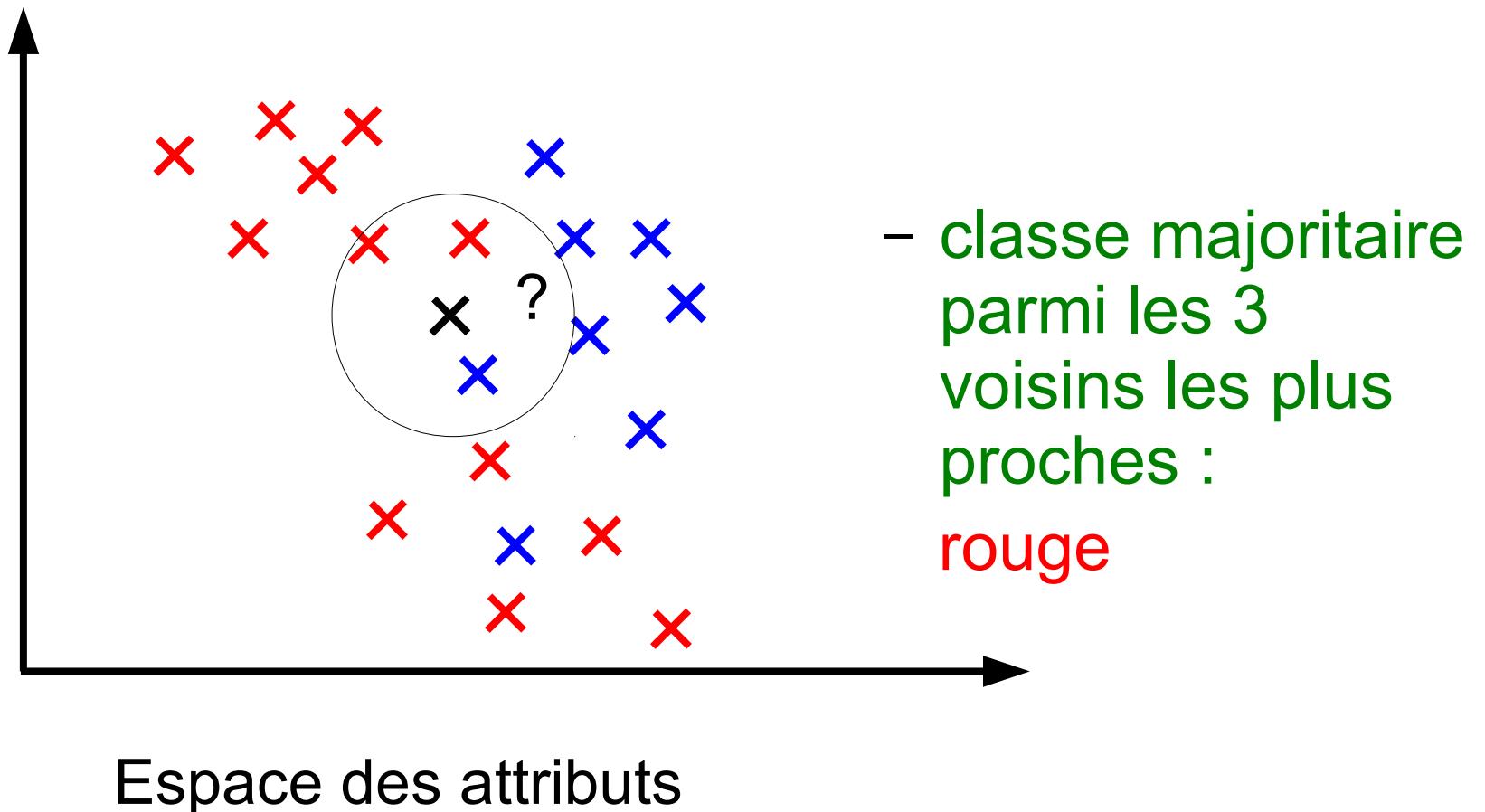
Étape de classification (1/2)

- Classification d'un individu inconnu ($k=1$)



Étape de classification (2/2)

- Classification d'un individu inconnu ($k=3$)



Remarques sur les kppv

- Pas de modèle explicite des classes
 - On utilise tous les exemples de chaque classe
- Avantage
 - Représentation d'une forme de nuage quelconque
- Inconvénients
 - Sensibilité aux points aberrants
 - Nécessité de mémoriser tous les points
 - Perte de généricité possible

Exercice 6

- On donne l'image "BillesColoreesBruit.bmp", le script "exercice6.m" et la fonction `tse_imkmeans.m` (cf. Mootse)
 - Executer le script et interpréter les figures qui sont tracées.
 - Examiner la fonction `tse_imkmeans` et segmenter l'image en choisissant le bon nombre de classes.
 - Vérifier les coordonnées des centres rentrés
 - Tester ce qu'il se passe si l'on surestime le nombre de classes dans la fonction `tse_imkmeans`.

Séance 7

5.3 Classifieur par distance minimale

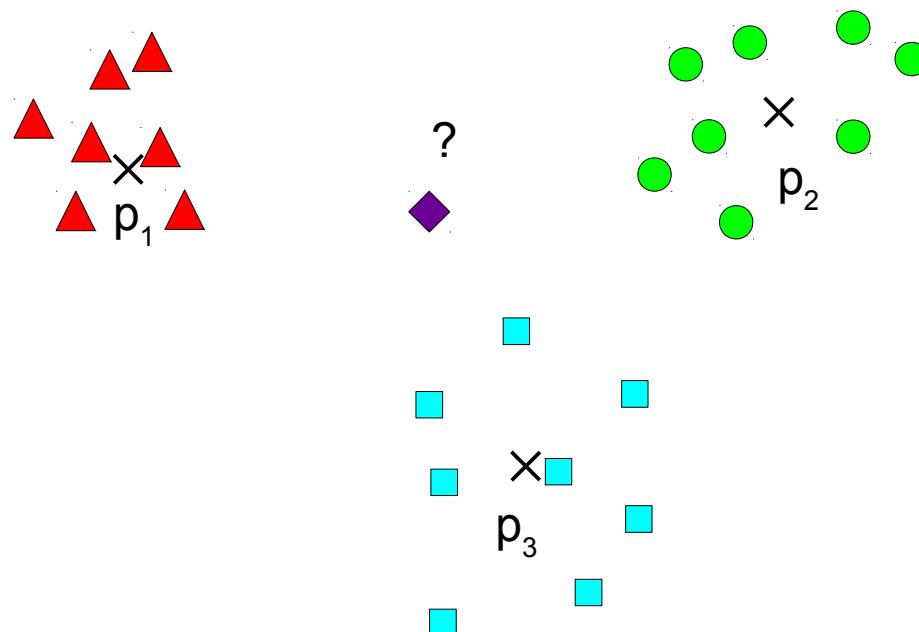
- Principe
 - Construire des prototypes (représentants de la classe)
 - Classifier en recherchant le prototype le plus proche

Construction de prototypes

- Modéliser chaque classe par un ou plusieurs représentants = prototypes
- Construction d'un prototype
 - Moyenne des attributs des échantillons d'apprentissage de la classe :
 - Le prototype p_k associé à la classe C_k est :
 - $$p_k = \frac{1}{N_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}$$

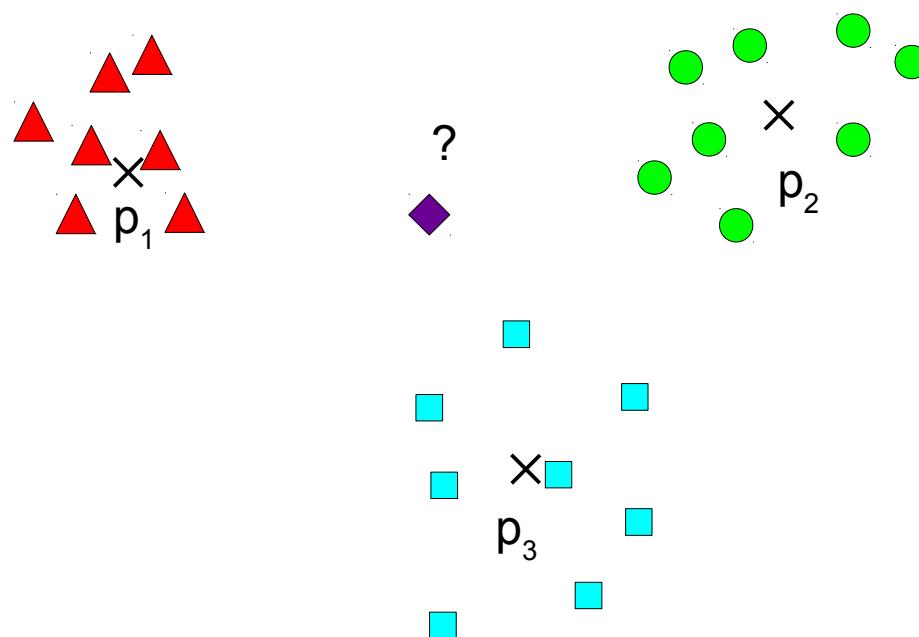
Principe de classification

- Classification par distance minimale = « 1 plus proche voisin »
- Attribuer la classe dont le prototype est le plus proche du point inconnu



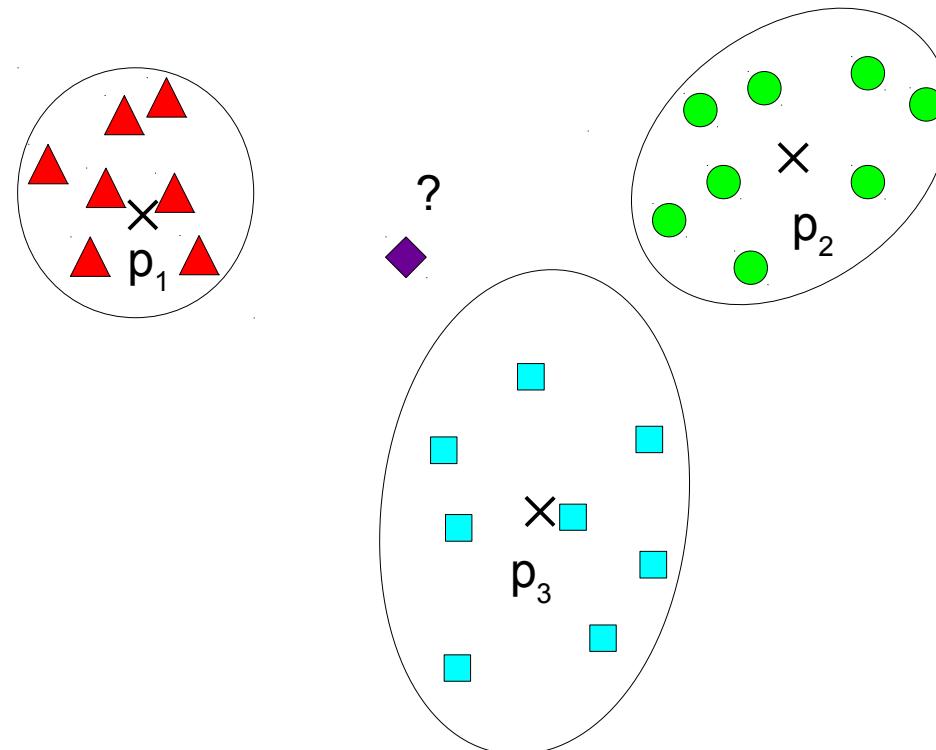
Complexité

- Calculer la distance à tous les prototypes
- Rechercher la distance minimale obtenue



Analyse critique

- Problème des prototypes
 - Moyen simpliste pour représenter la classe
 - Ne tient pas compte de l'extension du nuage
- Illustration



Améliorations possibles

- Adapter la distance à la distribution des points
 - Distance de Mahalanobis

$$D(\mathbf{x}, \mathbf{y}) = \left((\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y}) \right)^{1/2}$$

$$\Sigma = E \left[(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{y} - \mu_{\mathbf{y}})^T \right]$$

- Approche bayésienne
 - Modèle probabiliste de la distribution
 - Attribuer la classe la plus probable

5.4 Exemple de problème de segmentation supervisée

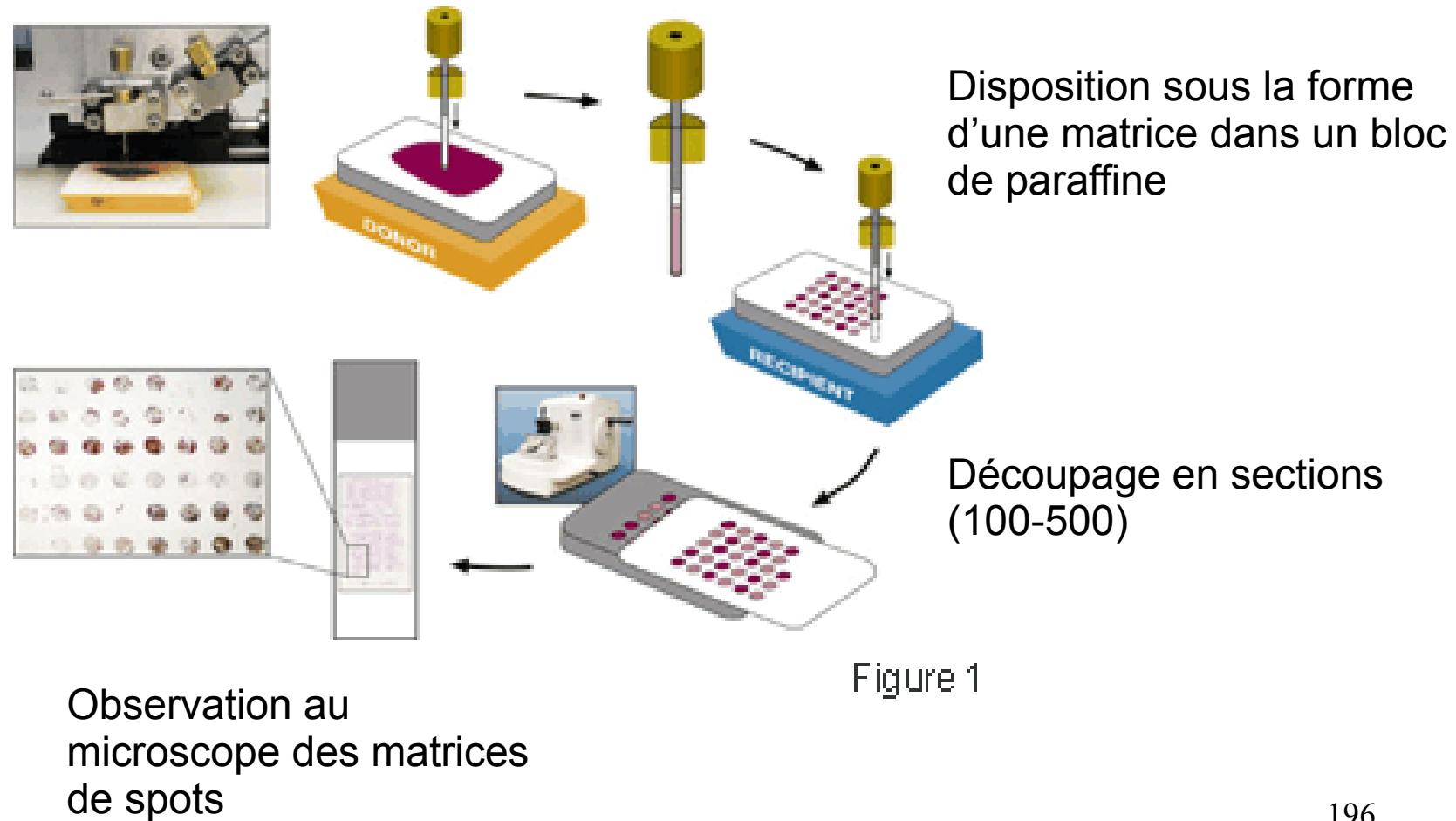
- Segmentation d'images
 - Images couleur de TMA
 - Segmenter les différentes zones
 - Associer une classe de tissu à ces zones
- Interaction avec un expert
 - L'expert marque des zones sur des images d'apprentissage
 - Il faut ensuite segmenter automatiquement les autres images

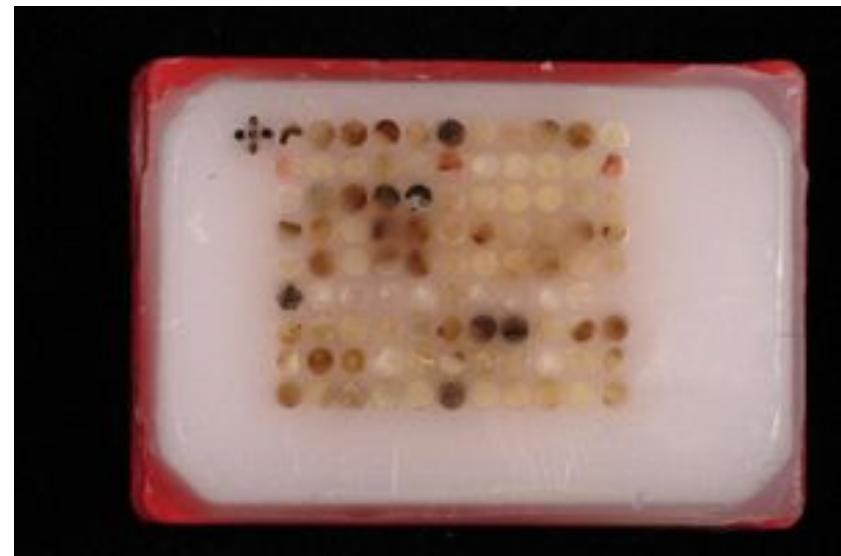
Principe des « puces à tissus »

Tissue Micro Array (TMA)

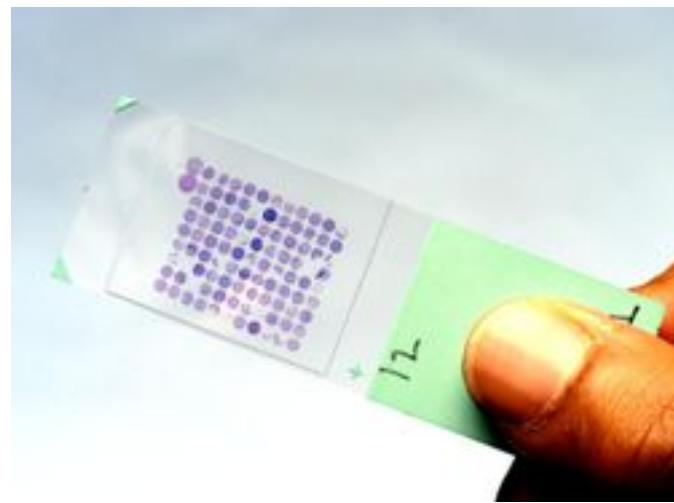
Tissu biologique
intégré dans de la paraffine

Prélèvement de carottes
dans des zones d'intérêt ($d=0.6$ mm)



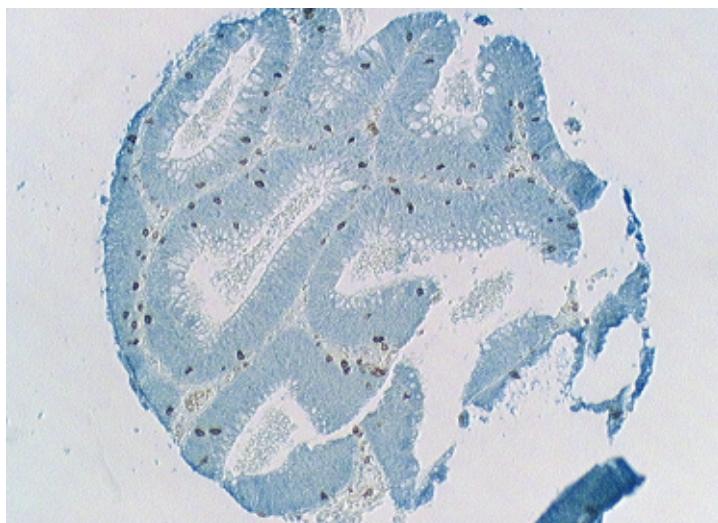
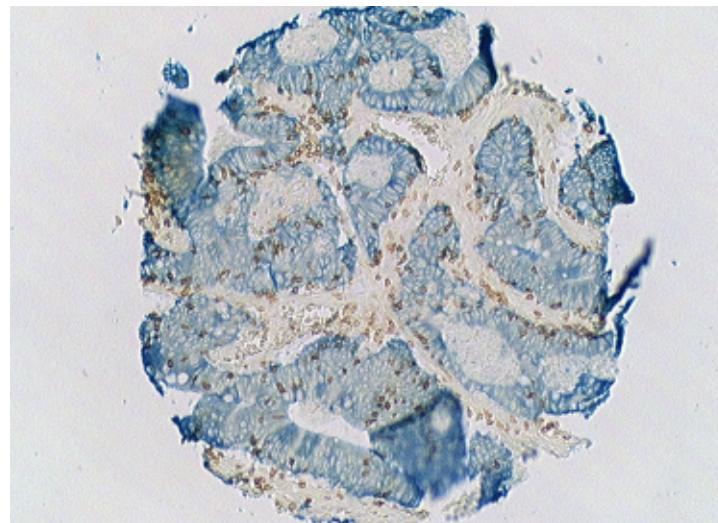
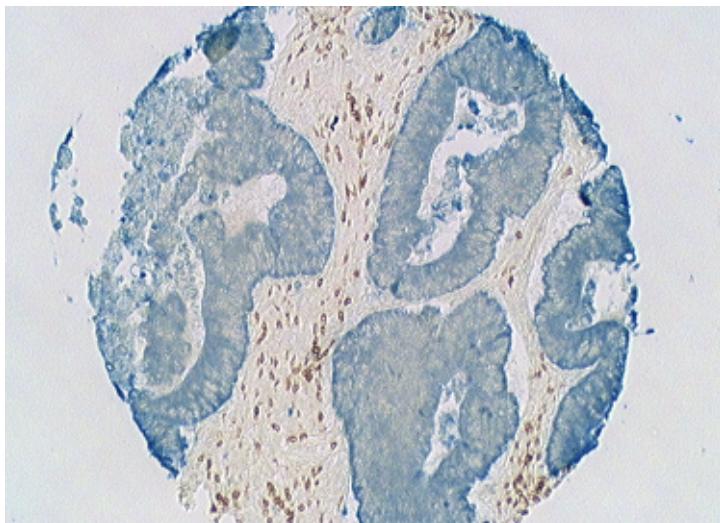


Un bloc de TMA



Une coupe

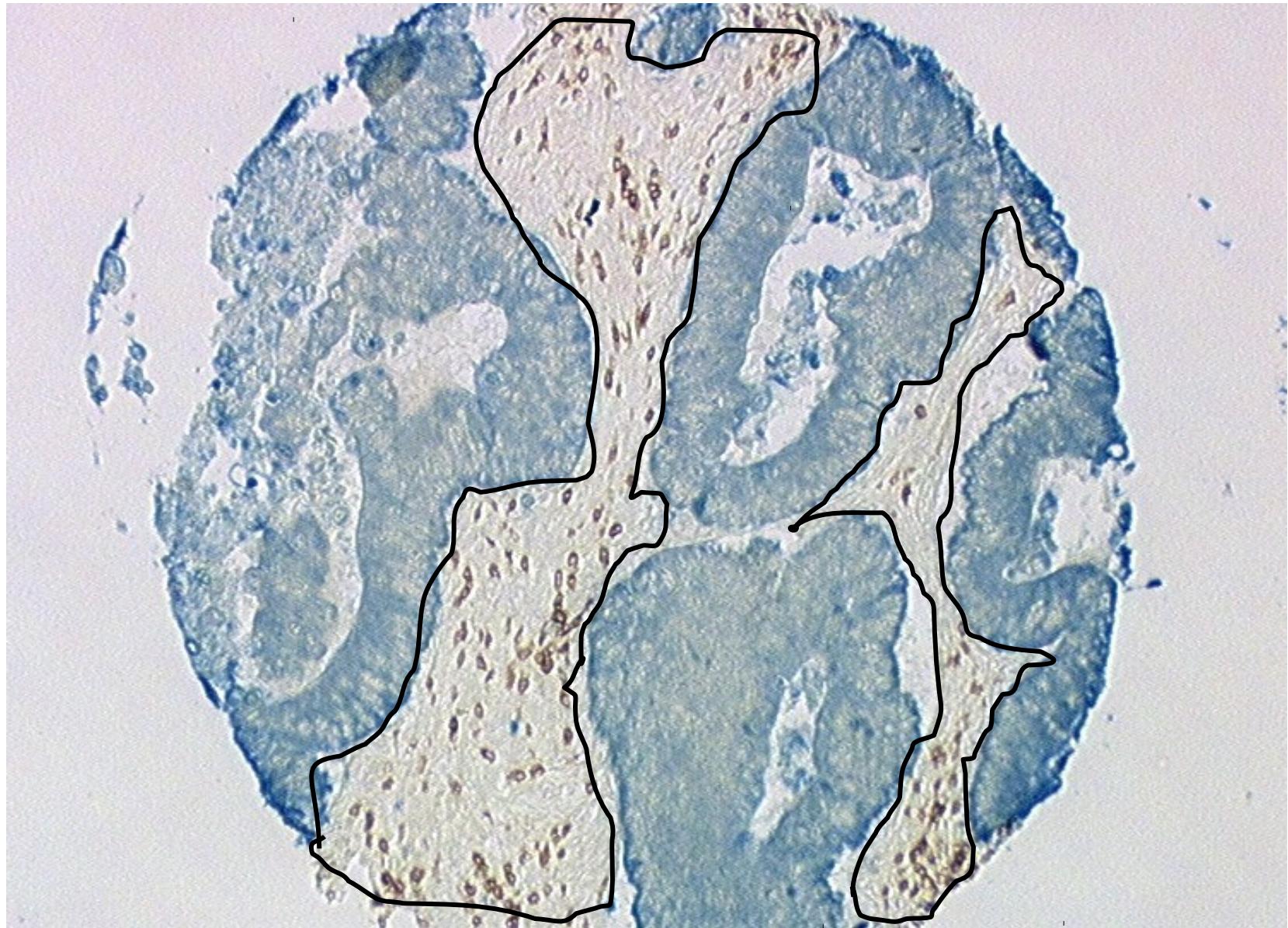
Exemples d'images



Prélèvement issus du colon

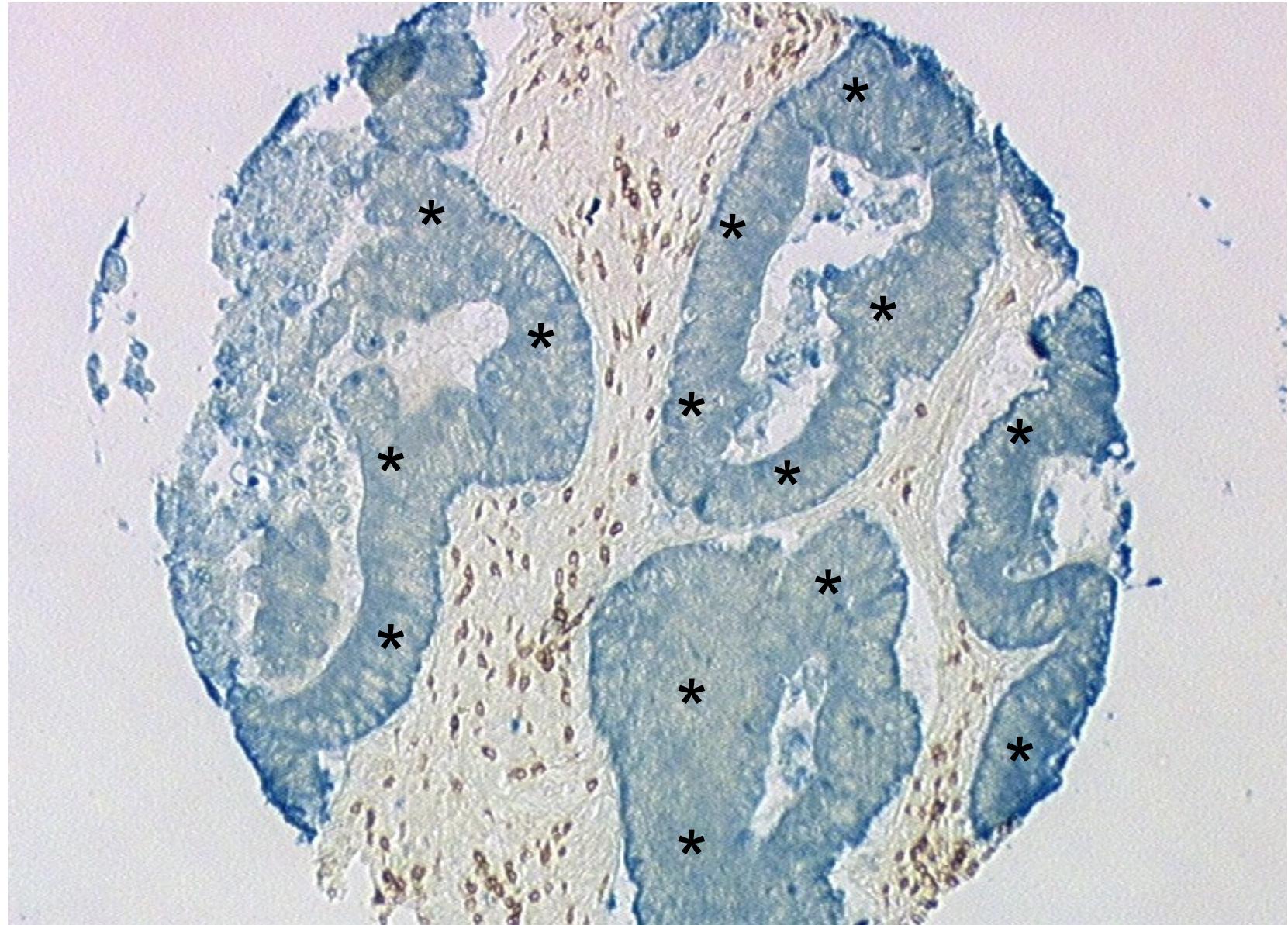
Données d'expertise disponibles
sur les images

Le stroma



glande tumorale immunomarquée (AE1-AE3)

*



Lymphocytes immunomarqués (DAB en marron)
Localisés dans le stroma et dans la glande tumorale

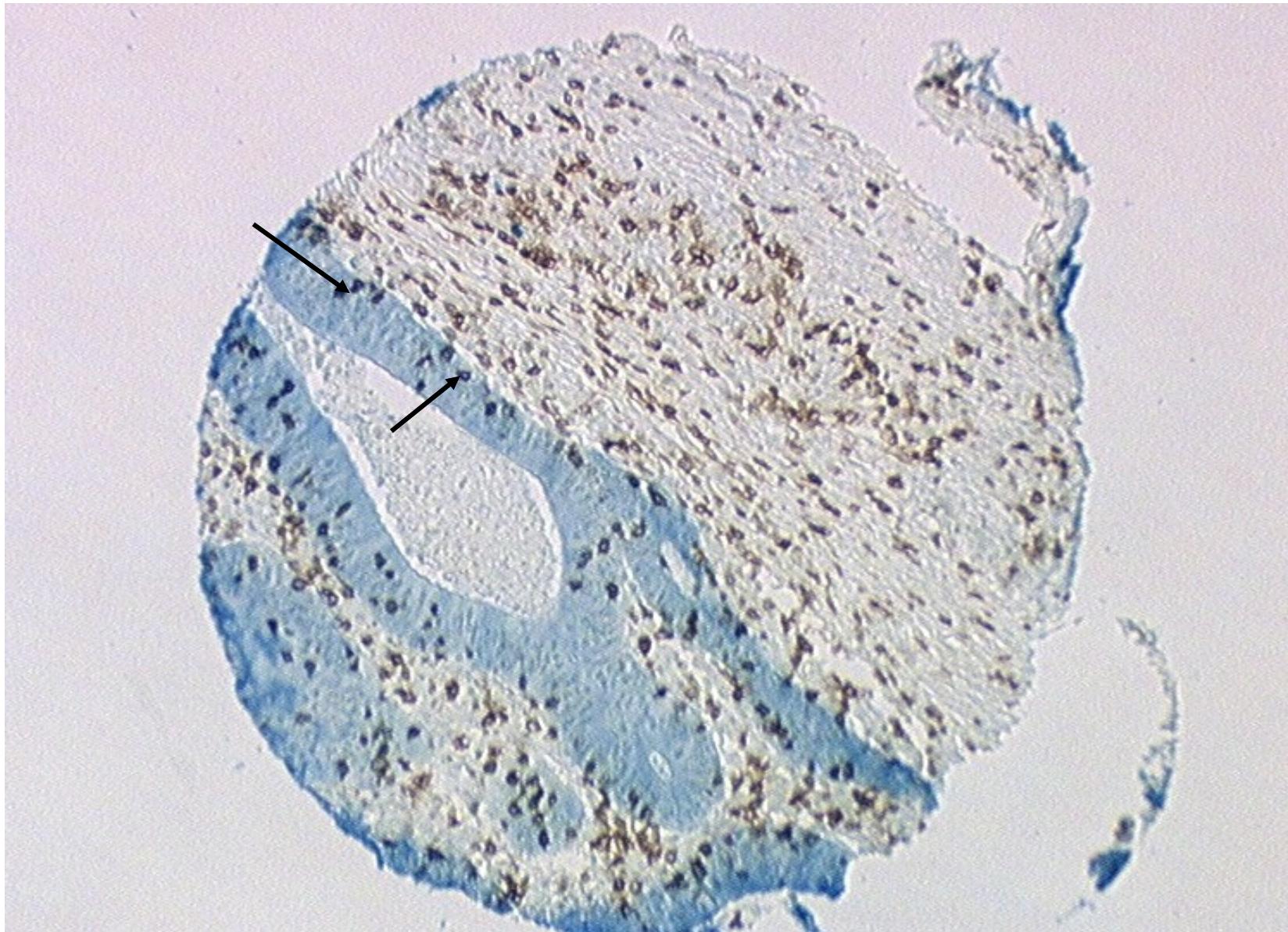
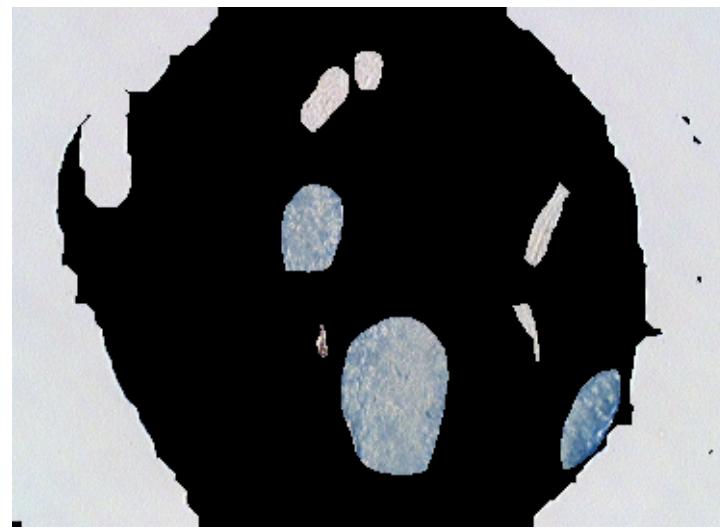
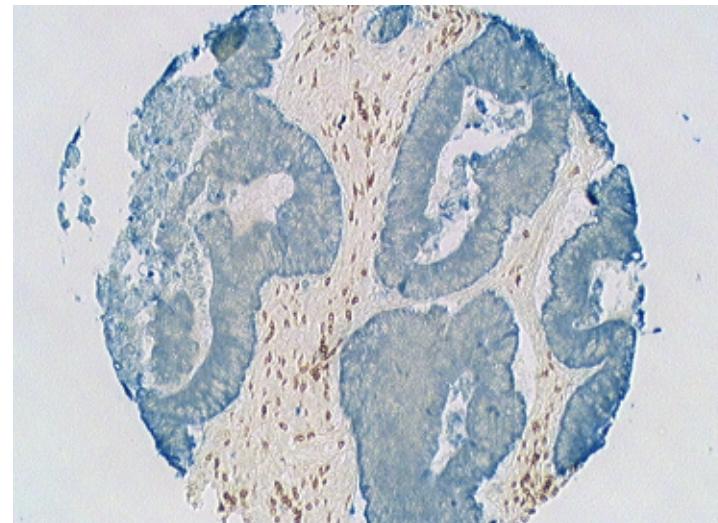


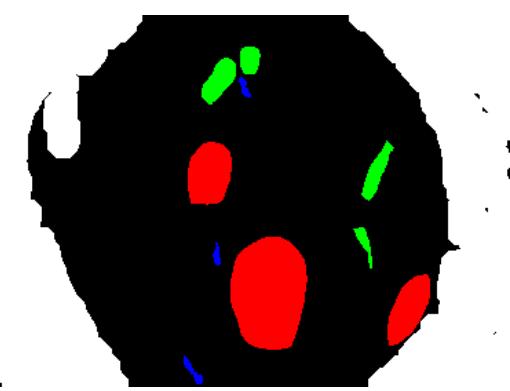
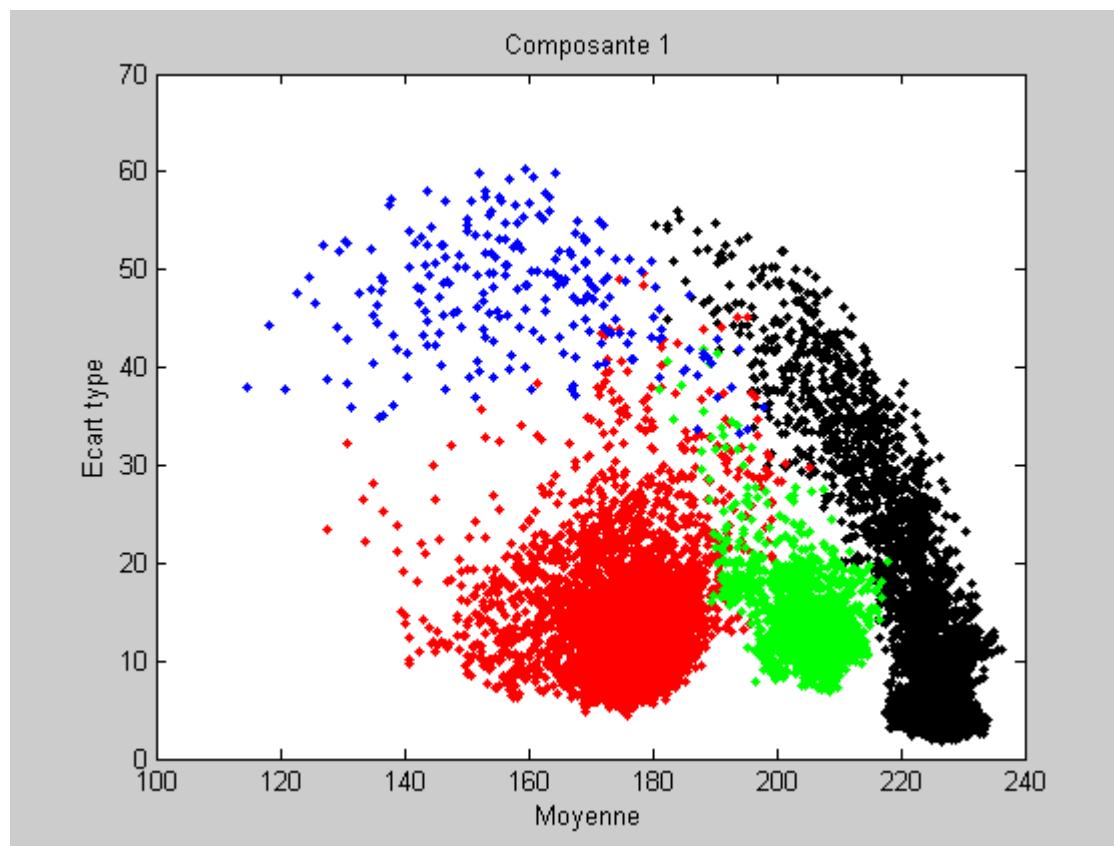
Image d'apprentissage



3 classes sélectionnées

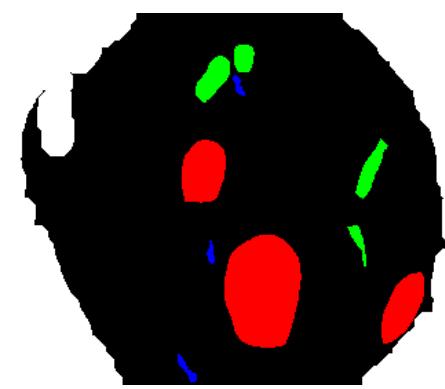
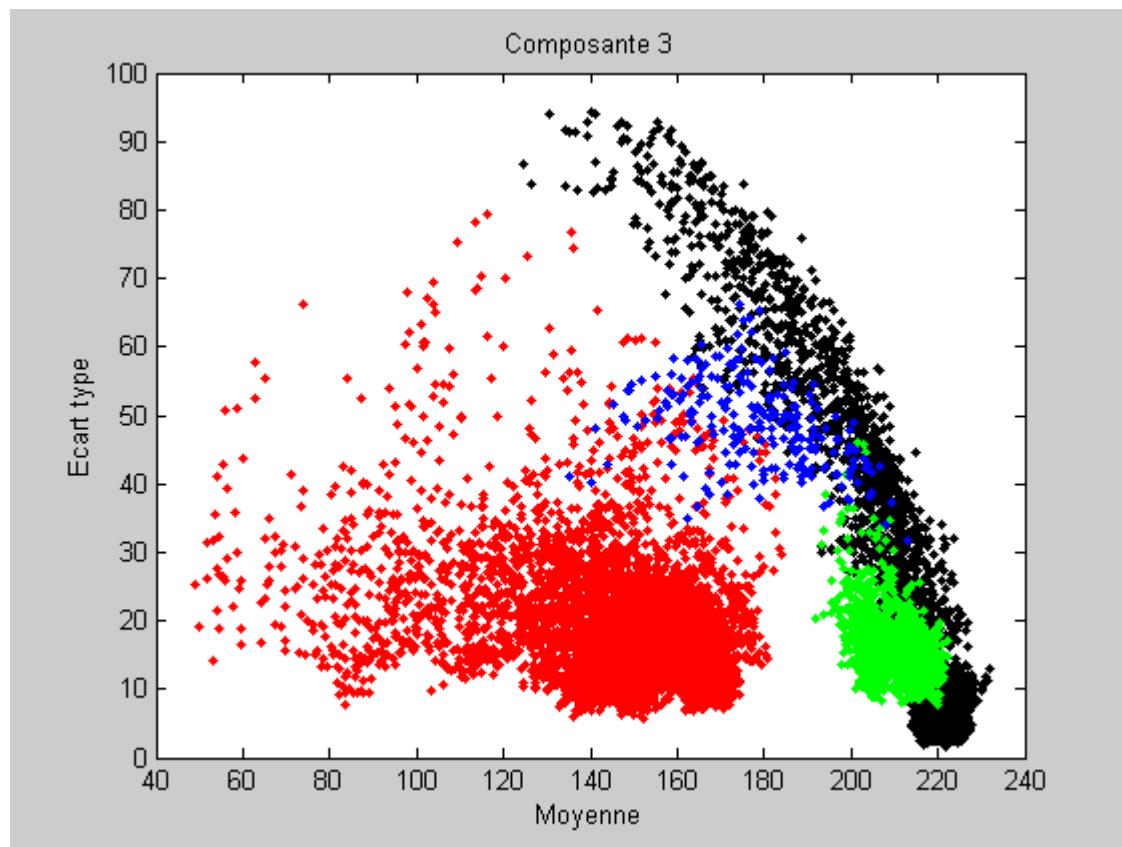
Calcul des attributs

Moyenne et écart type sur la composante 1



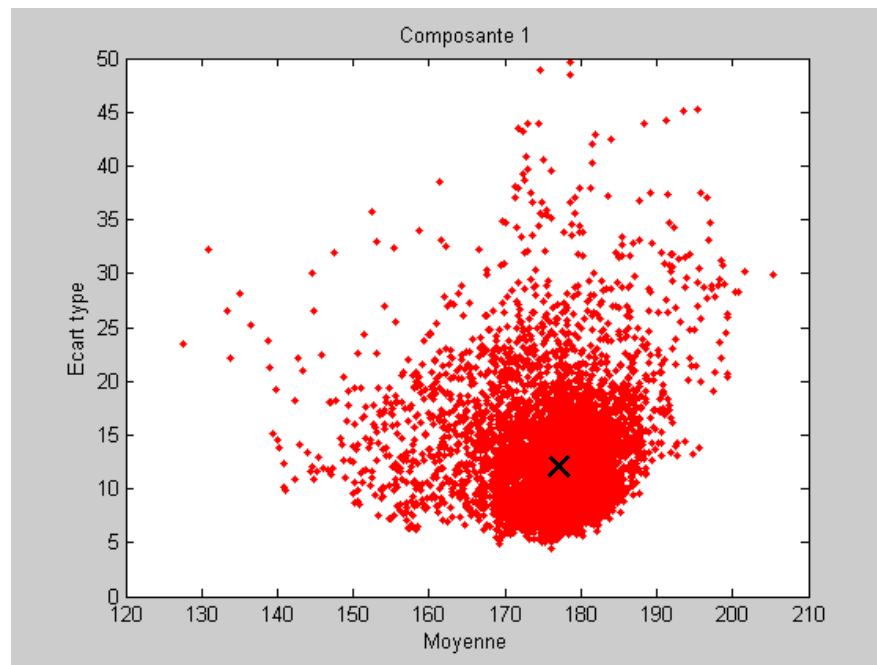
Calcul des attributs

Moyenne et écart type sur la composante 3



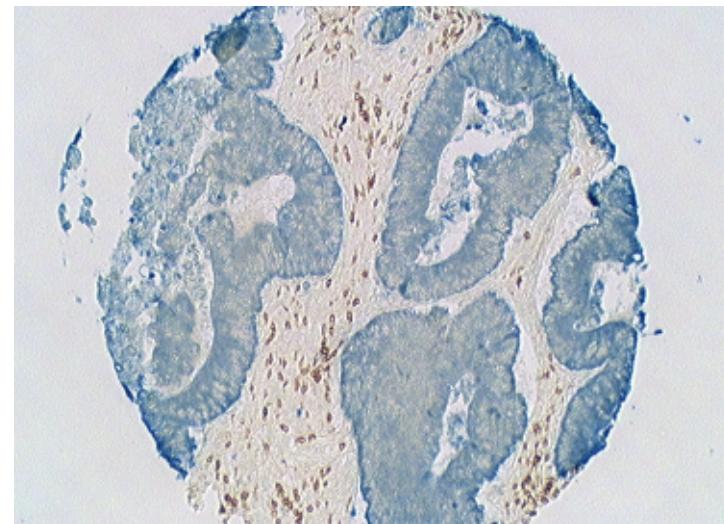
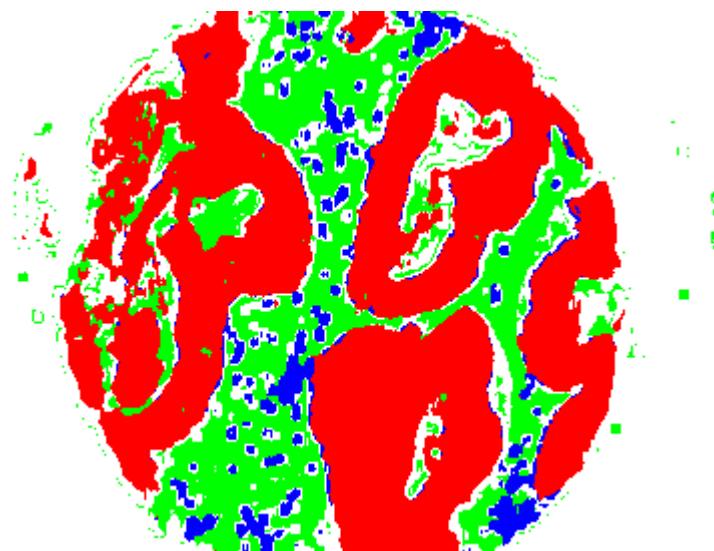
Calcul des prototypes

- Pour chaque classe
 - Calcul d'un ou plusieurs centres sur le sous ensemble de points de cette classe



Classification

- Centre du prototype le plus proche



Conclusion

- **Avantages**
 - Nombre d'attributs quelconque
 - Méthode générale
 - Supervisée ou non-supervisée
- **Inconvénients**
 - Choix des attributs
 - Sensibilité à des dérives des attributs

Séance 8

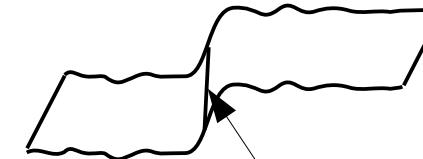
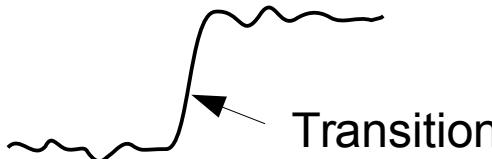
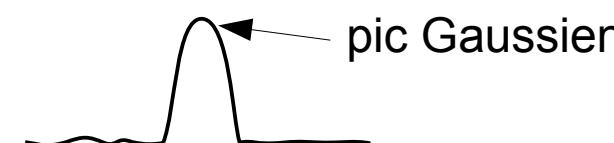
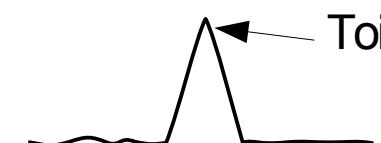
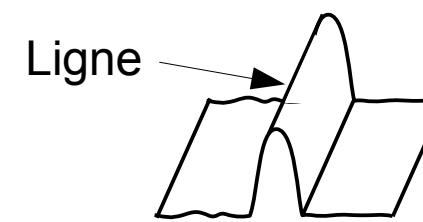
Plan

1. Introduction
2. Pré-traitements
3. Seuillage automatique
4. Mesure de paramètres
5. Segmentation par classification
6. Détection de contours
7. Points d'intérêts et mise en correspondance

6.1 Généralités

- **Introduction**
 - **Approches précédentes**
 - Approches région
 - > regroupement de pixels de caractéristiques voisines
 - **Approche contours**
 - Détection de changement de caractéristiques
 - > détection de variations rapides
 - Modèle d'image fonctionnel (dérivées)
 - Il existe des méthodes morphologiques (non traitées)

Définition d'un contour

- Plusieurs types de contours
 - Transition
 - variation rapide entre 2 zones de moyenne constante
 - Pic ou toit
 - maximum ou minimum local
 - Ligne (2D)
 - maximum dans une direction

Détection de contours

- Transition 1D
 - Détection d'un point d'inflexion

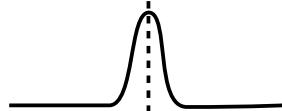
- maximum local de $\frac{df(x)}{dx}$

- passage par 0 avec changement de signe de $\frac{d^2 f(x)}{dx^2}$

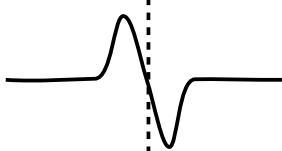
image : $f(x)$



dérivée 1ère : $\frac{df(x)}{dx}$



dérivée 2nde : $\frac{d^2 f(x)}{dx^2}$



Détection de contour 2D

- Dérivée directionnelle suivant u
 - Dérivées de la restriction à la direction u

Soit $\vec{u} = \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix}$

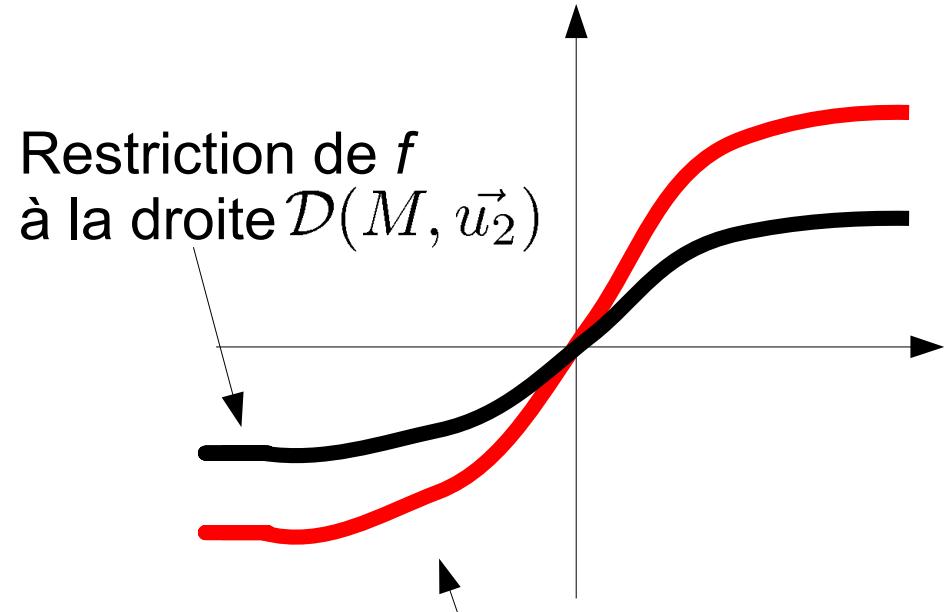
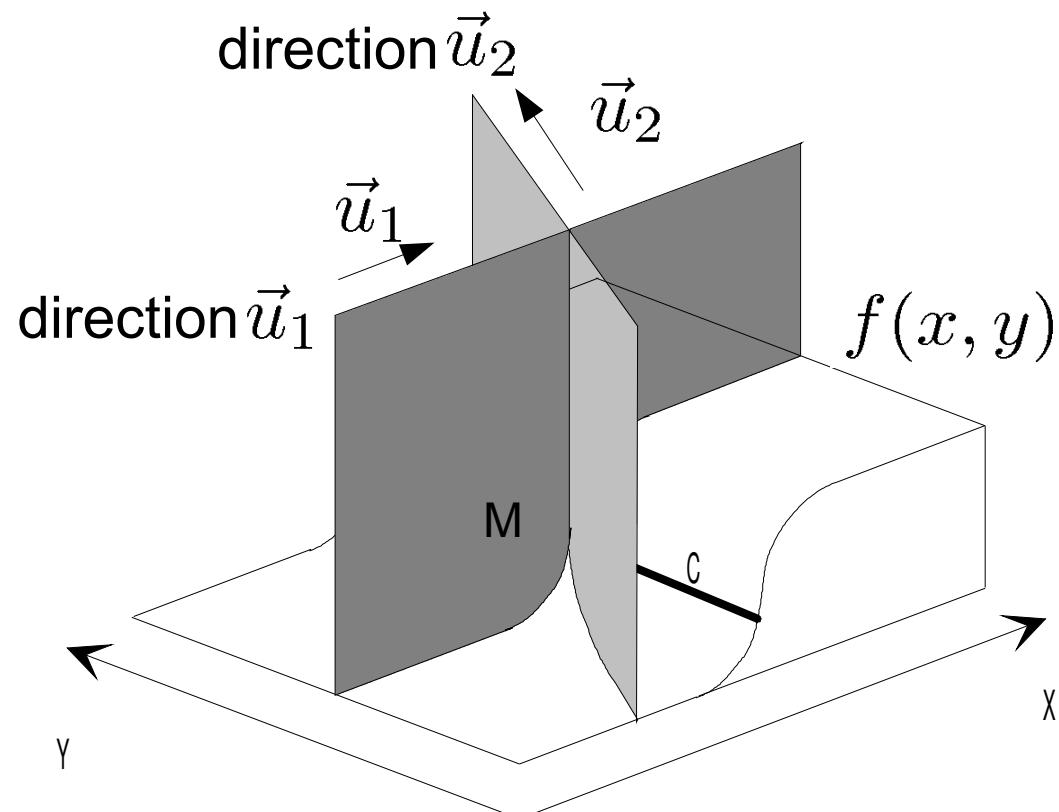
$$\frac{\partial f}{\partial u} = \frac{\partial f}{\partial x} \cos \phi + \frac{\partial f}{\partial y} \sin \phi = \vec{\nabla} f \cdot \vec{u}$$

$$\frac{\partial^2 f}{\partial u^2} = \frac{\partial^2 f}{\partial x^2} \cos^2 \phi + \frac{\partial^2 f}{\partial x \partial y} \cos \phi \sin \phi + \frac{\partial^2 f}{\partial y^2} \sin^2 \phi$$

- Transition
 - maximum local de la dérivée directionnelle
 - passage par 0 de la dérivée seconde directionnelle

Détection de contour 2D

- Dérivée directionnelle



Restriction de f
à la droite $D(M, \vec{u}_2)$

Restriction de f à
la droite $D(M, \vec{u}_1)$

Détection de contour 2D

- Remarques sur l'équation

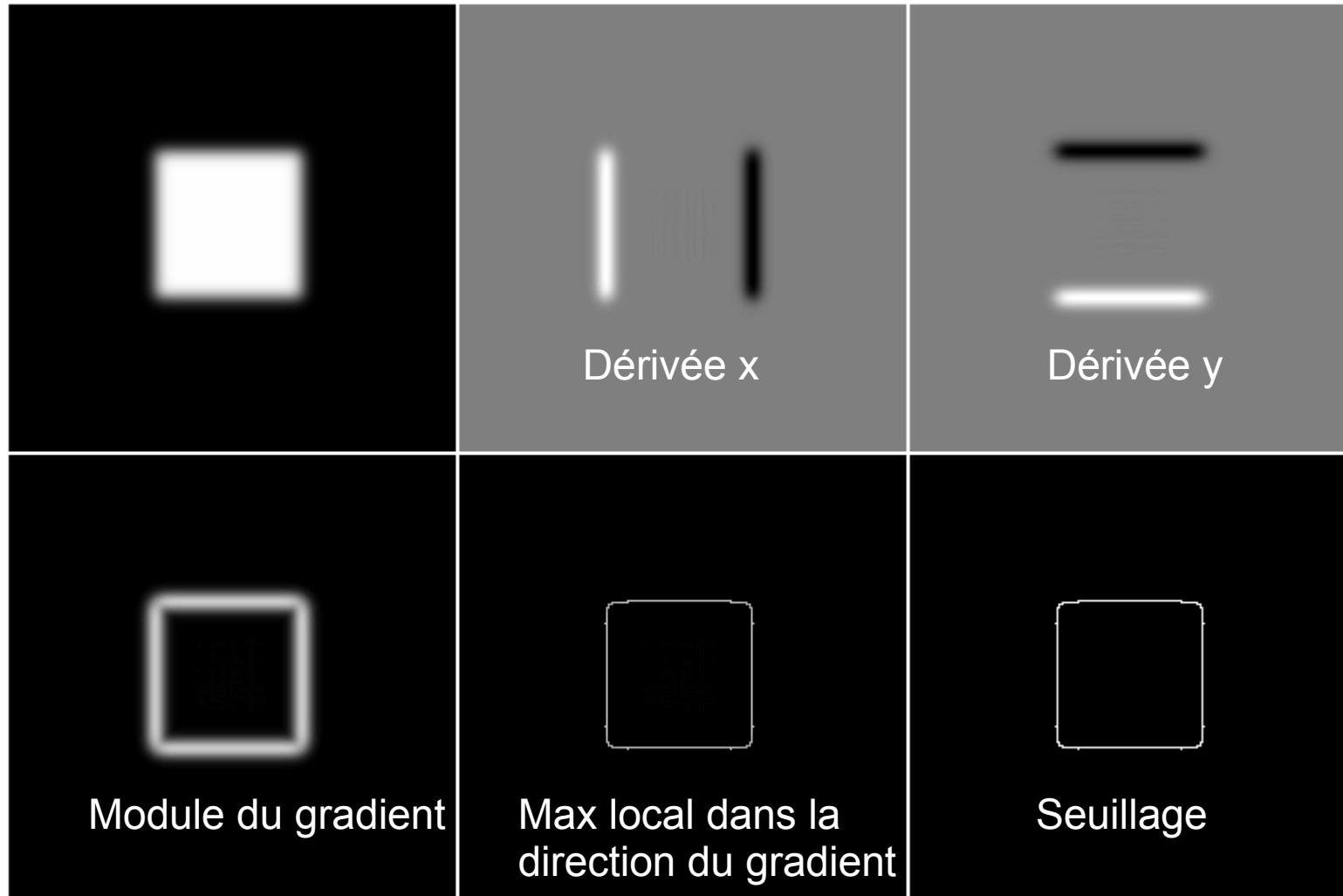
$$\frac{\partial f}{\partial u} = \frac{\partial f}{\partial x} \cos \phi + \frac{\partial f}{\partial y} \sin \phi = \vec{\nabla} f \cdot \vec{u}$$

- Direction du gradient = normale au contour
- La dérivée est maximale si la direction u est la direction du gradient
- La dérivée est alors égale au module du gradient
 - Module du gradient = dérivée suivant la plus grande pente

Détection à partir du gradient

1. Calcul des 2 composantes du gradient pour chaque pixel
2. Détecter les points de contour tel que
 - le module du gradient est maximal dans la direction donnée par le vecteur gradient
3. Seuillage sur le module du gradient
4. Fermeture des contours

Exemples



Détection à partir du laplacien

- A partir de l'équation

$$\frac{\partial^2 f}{\partial u^2} = \frac{\partial^2 f}{\partial x^2} \cos^2 \phi + \frac{\partial^2 f}{\partial x \partial y} \cos \phi \sin \phi + \frac{\partial^2 f}{\partial y^2} \sin^2 \phi$$

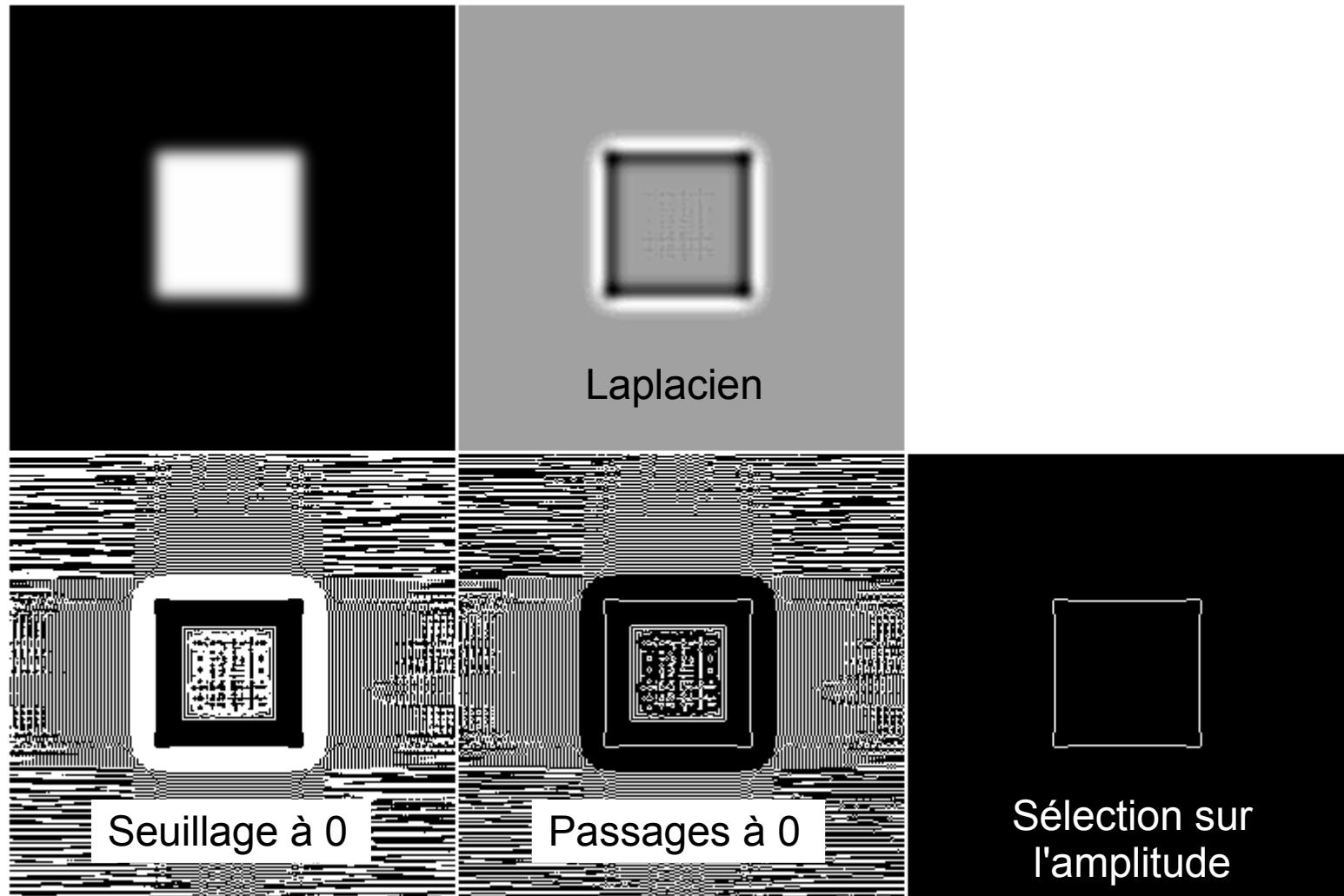
- Approximation : contour localement rectiligne
 - on supprime les termes de dérivée croisée
 - on supprime la direction de dérivation
 - on obtient le laplacien :

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Détection à partir du laplacien

1. Calcul du laplacien en chaque pixel
2. Déetecter les points de contour tel que :
 - Le laplacien s'annule en changeant de signe
 - Seuillage à 0
 - Détection des frontières d'objet : $f - \epsilon_B(f)$
3. Sélection des passages par 0
 - Amplitude du changement de signe
 - Dilaté - Erodé

Exemple



6. Détection de contours

6.1 Généralités

6.2 Opérateurs de gradient simple

6.3 Opérateur dérivée de gaussienne

6.4 Détection et fermeture des contours

6.5 Influence du lissage

6.6 Laplacien

6.7 Ligne de partage des eaux

5.2 Opérateurs gradient simples

- **Principe**

- $f(x, y) \rightarrow f[i, j]$

$$\frac{\partial f}{\partial x}(x, y) \approx \frac{\Delta f}{\Delta j}[i, j] = f[i, j + 1] - f[i, j] = f_j[i, j]$$

$$\frac{\partial f}{\partial y}(x, y) \approx \frac{\Delta f}{\Delta i}[i, j] = f[i + 1, j] - f[i, j] = f_i[i, j]$$

Gradients de Prewit et Sobel

- **Principe**

- Combinaison d'un lissage et d'une dérivée
- Définition

$$\begin{aligned} f_j[i, j] &= h_j * f[i, j] \\ f_i[i, j] &= h_i * f[i, j] \end{aligned}$$

Prewit : $c = 1$
Sobel : $c = 2$

$$h_j = \begin{pmatrix} 1 \\ c \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{pmatrix}$$

$$h_i = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & c & 1 \end{pmatrix} = \begin{pmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{pmatrix}$$

Gradients de Prewit et Sobel

- Calcul du module et de la direction du gradient

$$|\nabla f| = \sqrt{f_j^2 + f_i^2}$$

$$\phi = \arctan \left(\frac{f_i}{f_j} \right)$$

```
%Matlab  
gh = imfilter(f,fspecial('sobel') /8,'replicate');  
gv = imfilter(f,fspecial('sobel')/8,'replicate');  
g=sqrt(gh.*gh+gv.*gv) ;
```

6. Détection de contours

6.1 Généralités

6.2 Opérateurs de gradient simple

6.3 Opérateur dérivée de gaussienne

6.4 Détection et fermeture des contours

6.5 Influence du lissage

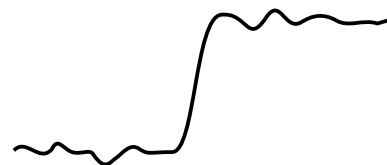
6.6 Laplacien

6.7 Ligne de partage des eaux

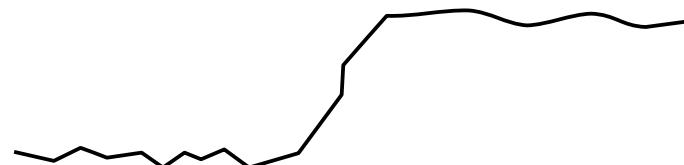
Nécessité d'un lissage

- Le gradient est sensible au bruit

Contour bruité

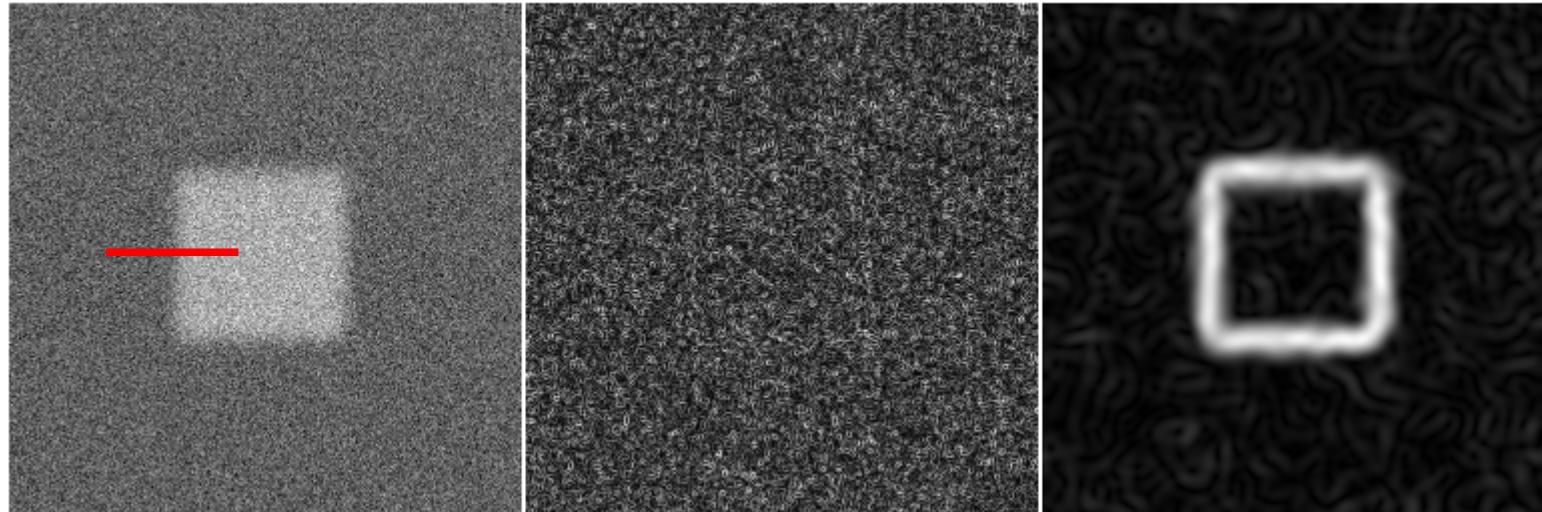


Contour flou et bruité



- Il faut appliquer un lissage avant de dériver
 - La taille du filtre est fonction de
 - l'amplitude du bruit
 - le degré de flou
 - Le lissage produit une délocalisation
 - Bon compromis : lissage gaussien

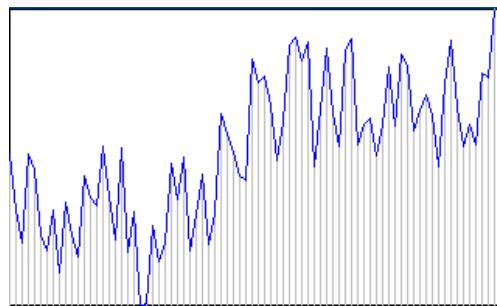
Nécessité d'un lissage



Bruit 4DB

Sobel

Lissage (gauss 4) +
Sobel



Profil 1D

Filtre dérivée de gaussienne

- Lissage + dérivée = filtre dérivée de gaussienne

Gaussienne : $g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$

Lissage : $f_l(x) = g_\sigma(x) * f(x)$

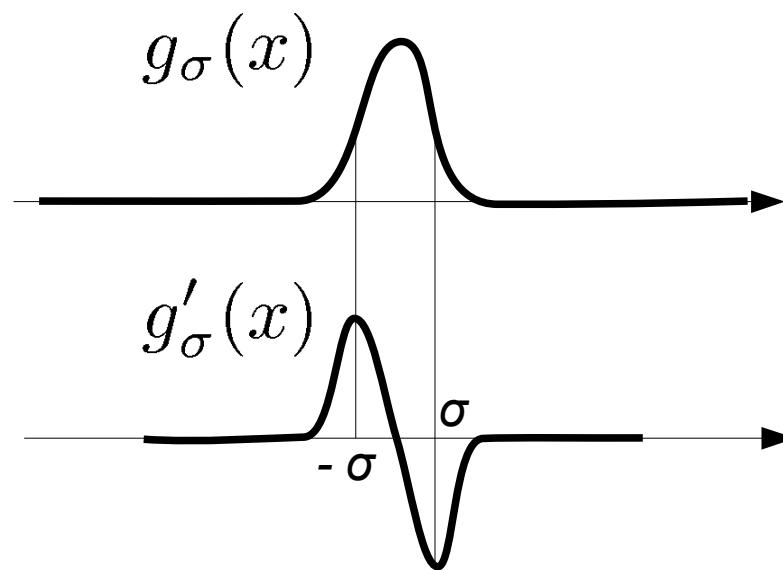
Lissage + dérivée : $f'_l(x) = \frac{d}{dx}(g_\sigma * f)(x) = (f * \frac{d}{dx}g_\sigma)(x) = f * g'_\sigma(x)$

Filtre dérivée de gaussienne : $g'_\sigma(x) = -\frac{x}{\sigma^2} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^3\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$

Filtre dérivée de gaussienne

- Forme du filtre

$$g'_\sigma(x) = -\frac{x}{\sigma^3 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$



En 2D : filtre gradient de gaussienne

- Gradient de gaussienne

$$\vec{\nabla} g_\sigma(x, y) = \begin{pmatrix} -\frac{x}{\sigma^4 2\pi} e^{-\frac{x^2+y^2}{2\sigma^2}} \\ -\frac{y}{\sigma^4 2\pi} e^{-\frac{x^2+y^2}{2\sigma^2}} \end{pmatrix}$$

- Filtre complexe associé

$$\nabla g_\sigma(x, y) = -\frac{1}{\sigma^4 2\pi} (x + iy) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

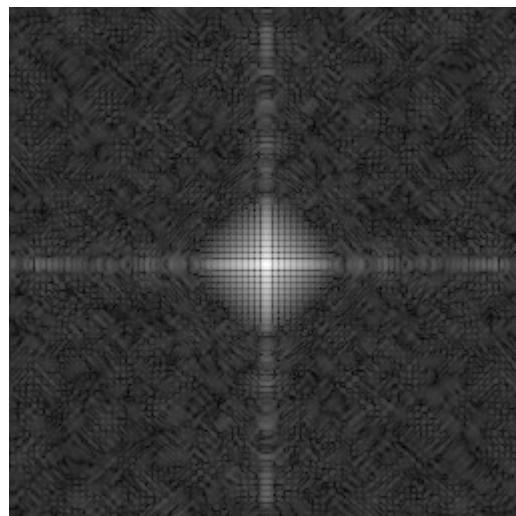
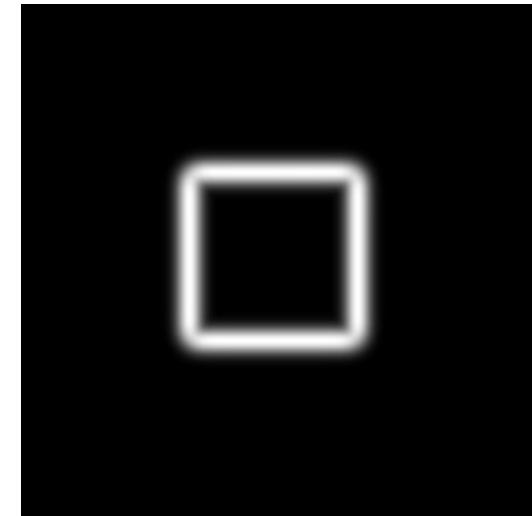
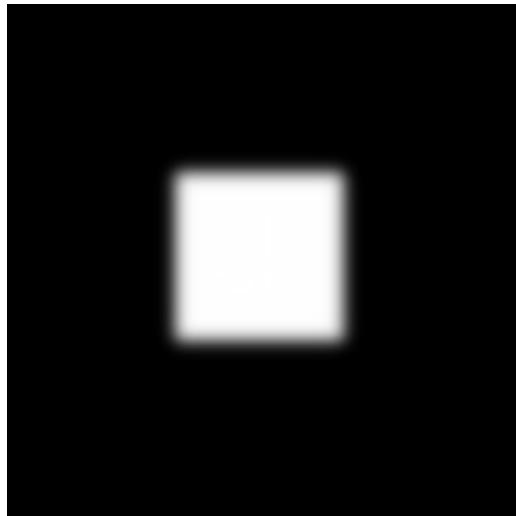
- Réponse fréquentielle du filtre

$$\nabla \hat{g}_\sigma(\omega_x, \omega_y) = \sigma(i\omega_x - \omega_y) e^{-\frac{\omega_x^2 + \omega_y^2}{2\sigma^2}}$$

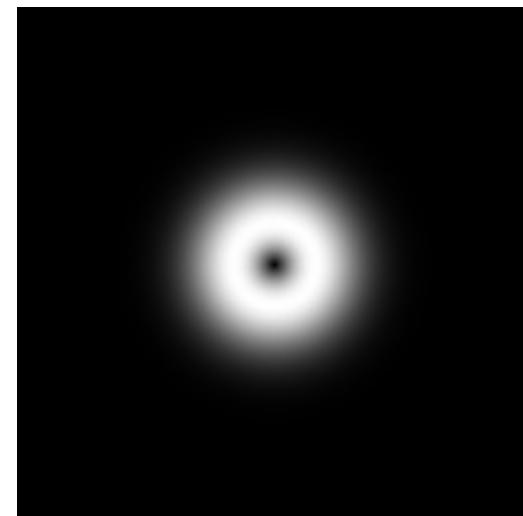
En 2D : filtre gradient de gaussienne

- Calcul de la convolution
 - 1. Calcul de la TF de l'image (FFT)
 - 2. X complexe par $\nabla \hat{g}_\sigma$
 - 3. TF⁻¹ (FFT inverse)
- Exploitation du résultat (image complexe)
 - Partie réelle : dérivée en x
 - Partie imaginaire : dérivée en y
 - Module : module du gradient
 - Phase : direction du gradient

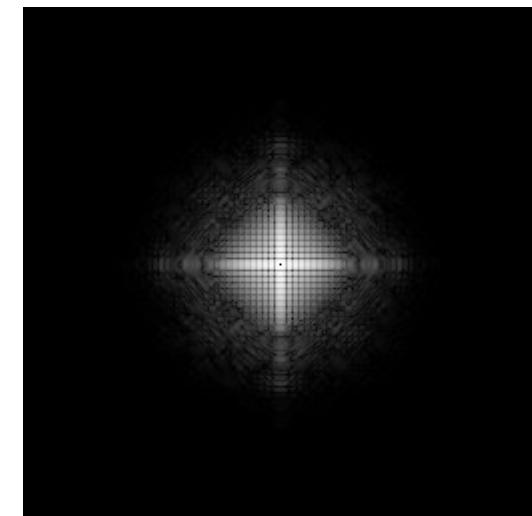
Illustration



$$\hat{f}(\omega_x, \omega_y)$$



$$\nabla \hat{g}_\sigma(\omega_x, \omega_y)$$



$$\hat{f} \cdot \nabla \hat{g}_\sigma(\omega_x, \omega_y)$$

En 2D : filtre gradient de gaussienne

- Autre moyen de calcul : utiliser la séparabilité

$$\frac{\partial g_\sigma}{\partial x}(x, y) = -\frac{x}{\sigma^4 2\pi} e^{-\frac{x^2+y^2}{2\sigma^2}} = -\frac{x}{\sigma^3 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}$$

- Convoyer selon les colonnes par une gaussienne
- Convoyer selon les lignes par la dérivée d'une gaussienne

```
%Matlab
width=ceil(4*sigma);
[x,y]=meshgrid(-width:width,-width:width);
gog = -x.*exp(-(x.*x+y.*y)/(2*sigma^2))/(2*pi*sigma^4)
gh=imfilter(double(f),gog,'conv','replicate');
gv=imfilter(double(f),gog','conv','replicate');
g=sqrt(gh.*gh+gv.*gv) ;
```

Séance 9

6. Détection de contours

6.1 Généralités

6.2 Opérateurs de gradient simple

6.3 Opérateur dérivée de gaussienne

6.4 Détection et fermeture des contours

6.5 Influence du lissage

6.6 Laplacien

6.7 Ligne de partage des eaux

Après le calcul du gradient

6.4.1 Détection des points de contours

6.4.2 Seuillage

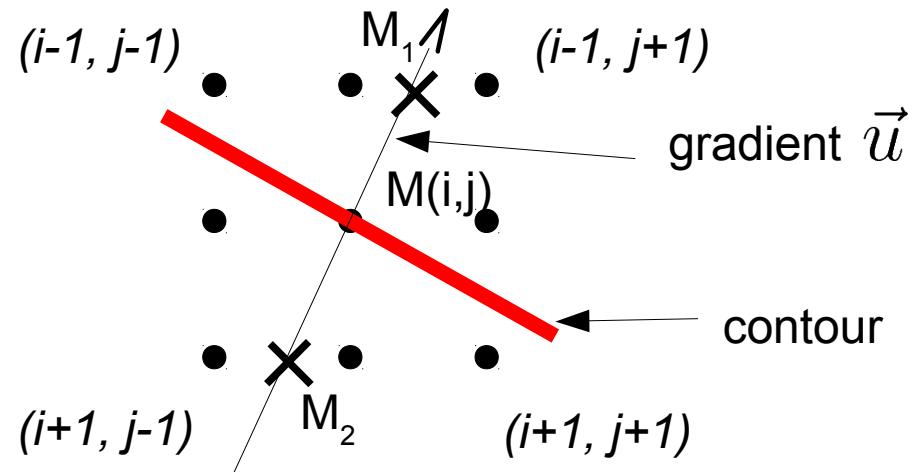
6.4.3 Fermeture des contours

6.4.1 Détection des points de contour

- Extraire les max locaux du gradient dans la direction du gradient
 - Soit $M[i,j]$ de norme de gradient Gr
 - Soient le points voisins dans la direction du gradient
 - M_1 de norme de gradient Gr_1
 - M_2 de norme de gradient Gr_2
 - M est un contour si

$$\begin{cases} Gr > Gr_1 \\ Gr \geq Gr_2 \end{cases}$$

Détection des points de contour



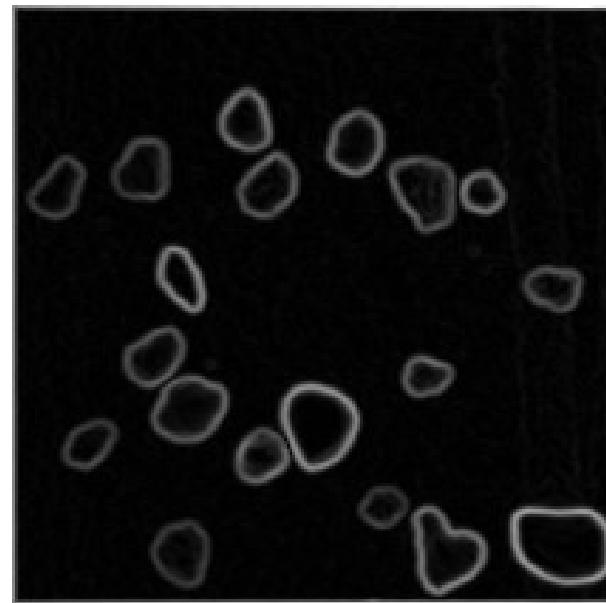
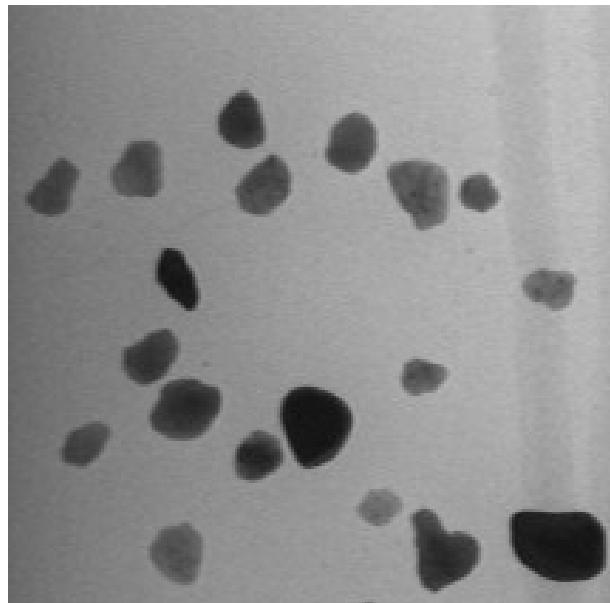
- Calcul de Gr1 et Gr2 par interpolation

– si $u_i > u_j$:

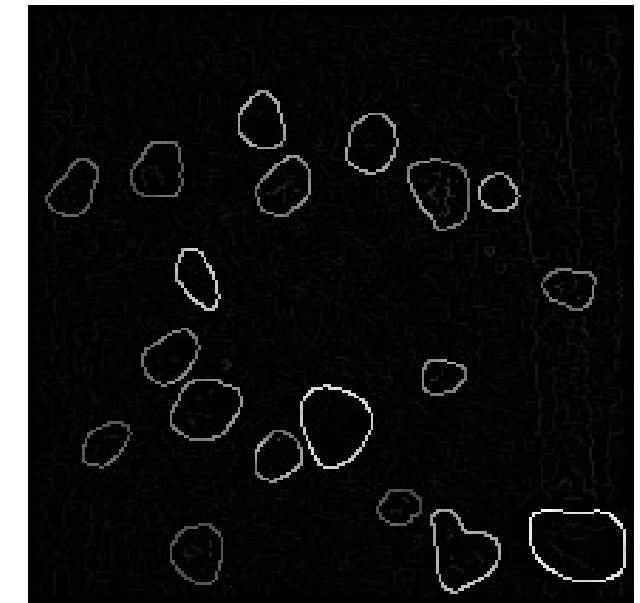
$$Gr_1 = \frac{u_j}{u_i} Gr(i - 1, j + 1) + \frac{u_i - u_j}{u_i} Gr(i - 1, j)$$

$$Gr_2 = \frac{u_j}{u_i} Gr(i + 1, j - 1) + \frac{u_i - u_j}{u_i} Gr(i + 1, j)$$

Illustration



Module du gradient



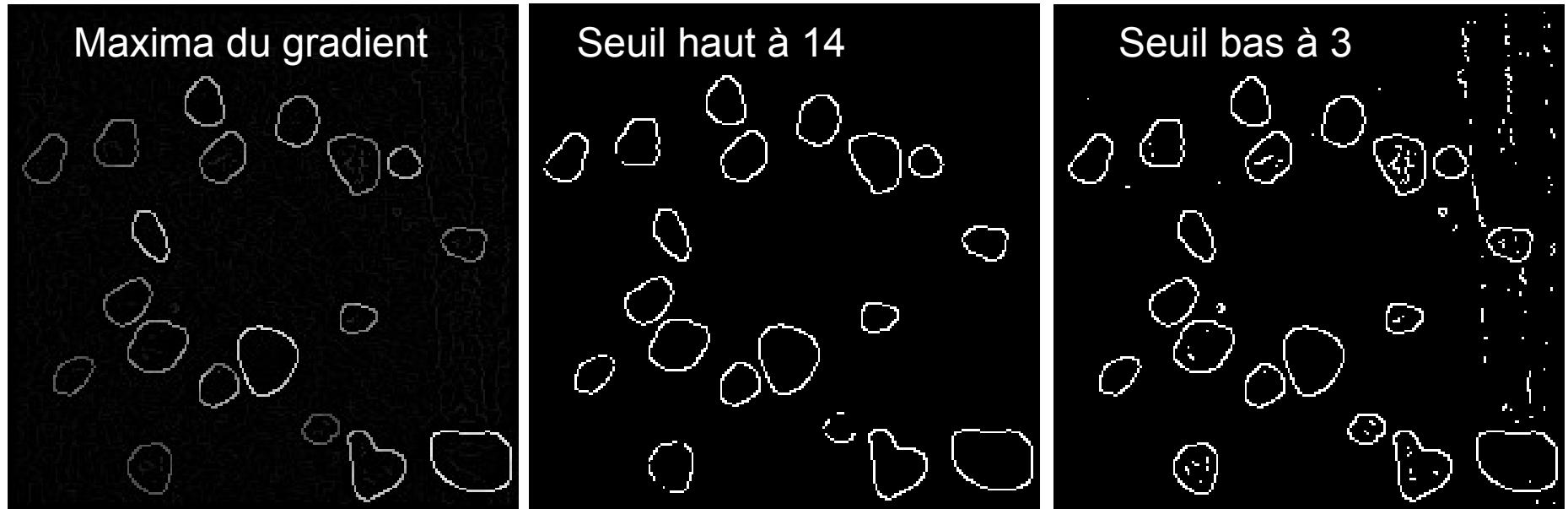
Max locaux dans la
direction du gradient

6.4.2 Seuillage

- Utilisation de 2 seuils : hystérésis
 - Seuillage 1 avec un seuil haut $s_h \rightarrow f_{s_h}$
 - Seuillage 2 avec un seuil bas $s_b \rightarrow f_{s_b}$
 - Reconstruction de f_{s_b} à partir des marqueurs f_{s_h}

```
%Matlab  
fs=imreconstruct(f>high,f>low);
```

Illustration

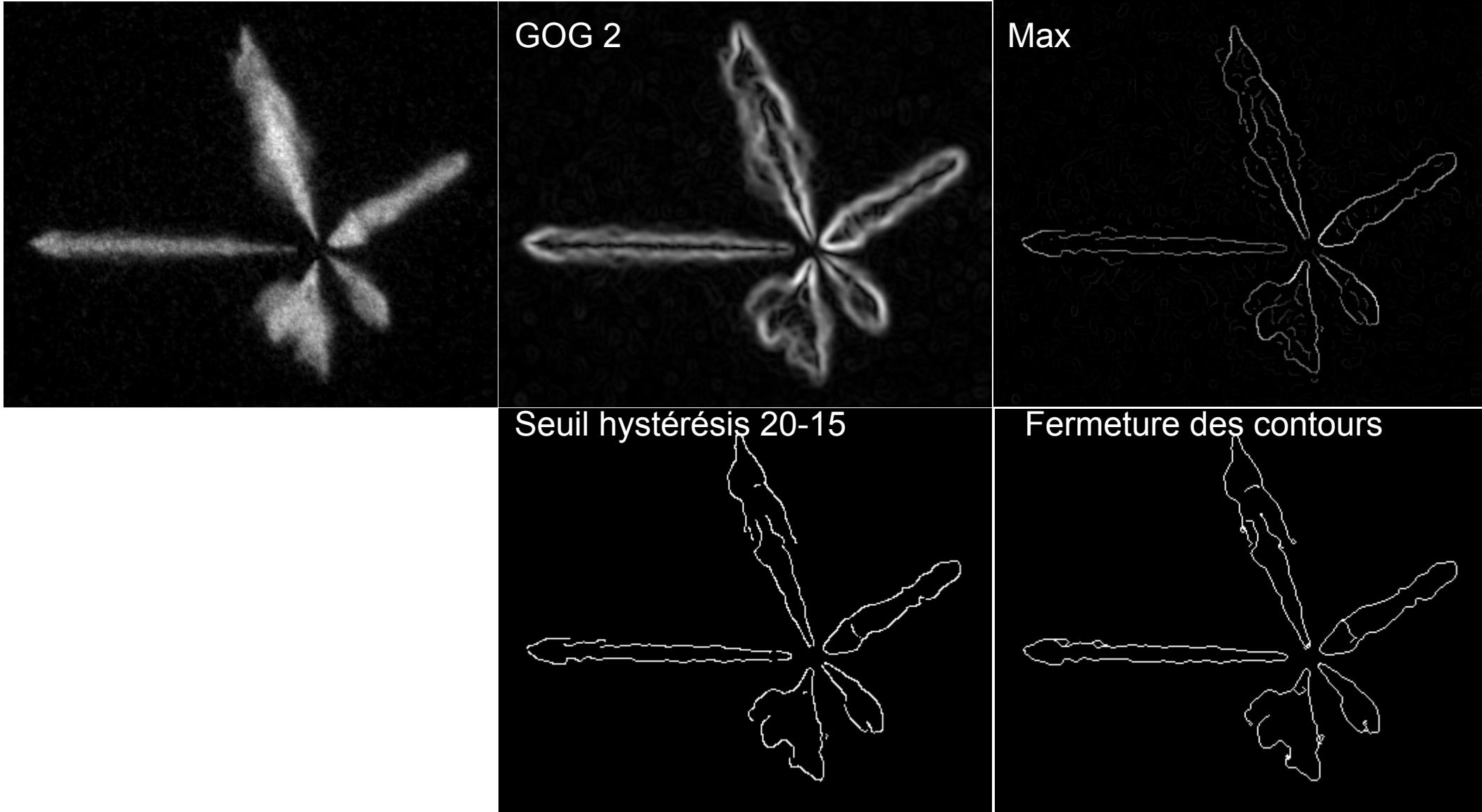


```
% Matlab  
% GOG + Max detection + hysteresis threshold  
fe=edge(f,'canny');
```

6.4.3 Fermeture des contours

- Nombreux algorithmes
 - certains efficaces mais complexes
- Principe d'un algorithme simple
 - Recherche des extrémités (amincir les contours)
 - Prolongation en cherchant à minimiser la somme des gradient le long du contour
 - Itération jusqu'à une longueur max ou jusqu'à la rencontre d'un point de contour
- La ligne de partage des eaux permet de trouver une solution élégante

Illustration



6. Détection de contours

6.1 Généralités

6.2 Opérateurs de gradient simple

6.3 Opérateur dérivée de gaussienne

6.4 Détection et fermeture des contours

6.5 Influence du lissage

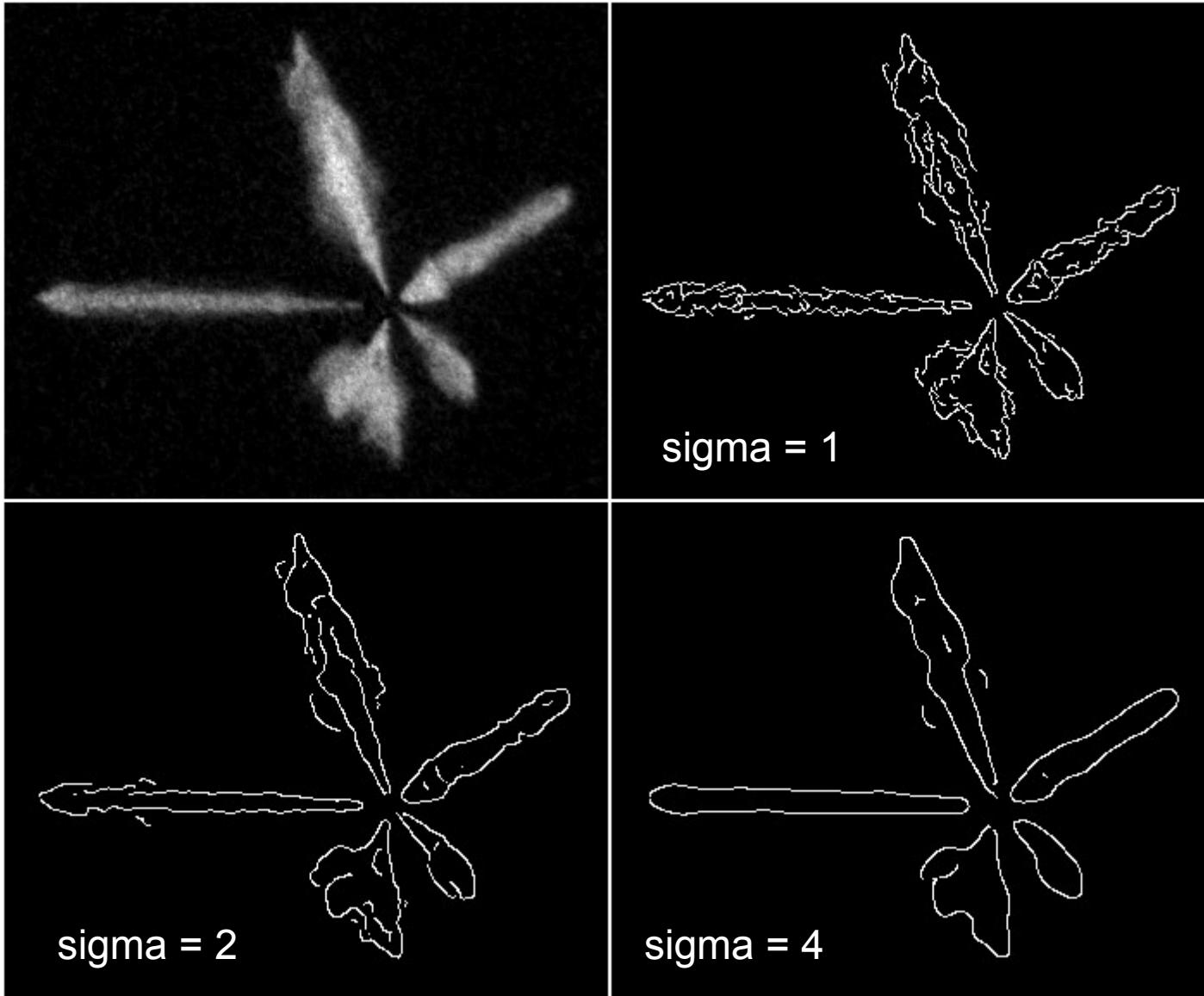
6.6 Laplacien

6.7 Ligne de partage des eaux

6.5 Influence du lissage

- Contours flous et bruités
 - Grand σ
 - Délocalisation
- Contours nets
 - Petit σ

Illustration



6. Détection de contours

6.1 Généralités

6.2 Opérateurs de gradient simple

6.3 Opérateur dérivée de gaussienne

6.4 Détection et fermeture des contours

6.5 Influence du lissage

6.6 Laplacien

6.7 Ligne de partage des eaux

Laplacien simple

- Version continue

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Version discrète
 - Masque Laplacien 3x3

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

Opérateur Laplacien de Gaussienne

- Même démarche que pour le gradient

Gaussienne 2D :
$$g_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Laplacien de gaussienne :
$$\Delta g_\sigma(x, y) = \frac{1}{2\pi\sigma^4} \left(\frac{x^2 + y^2}{\sigma^2} - 2 \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Calcul à partir de TF

Réponse fréquentielle :
$$\Delta \hat{g}_\sigma(\omega_x, \omega_y) = \sigma(\omega_x^2 + \omega_y^2) e^{-\frac{\omega_x^2 + \omega_y^2}{2\sigma^2}}$$

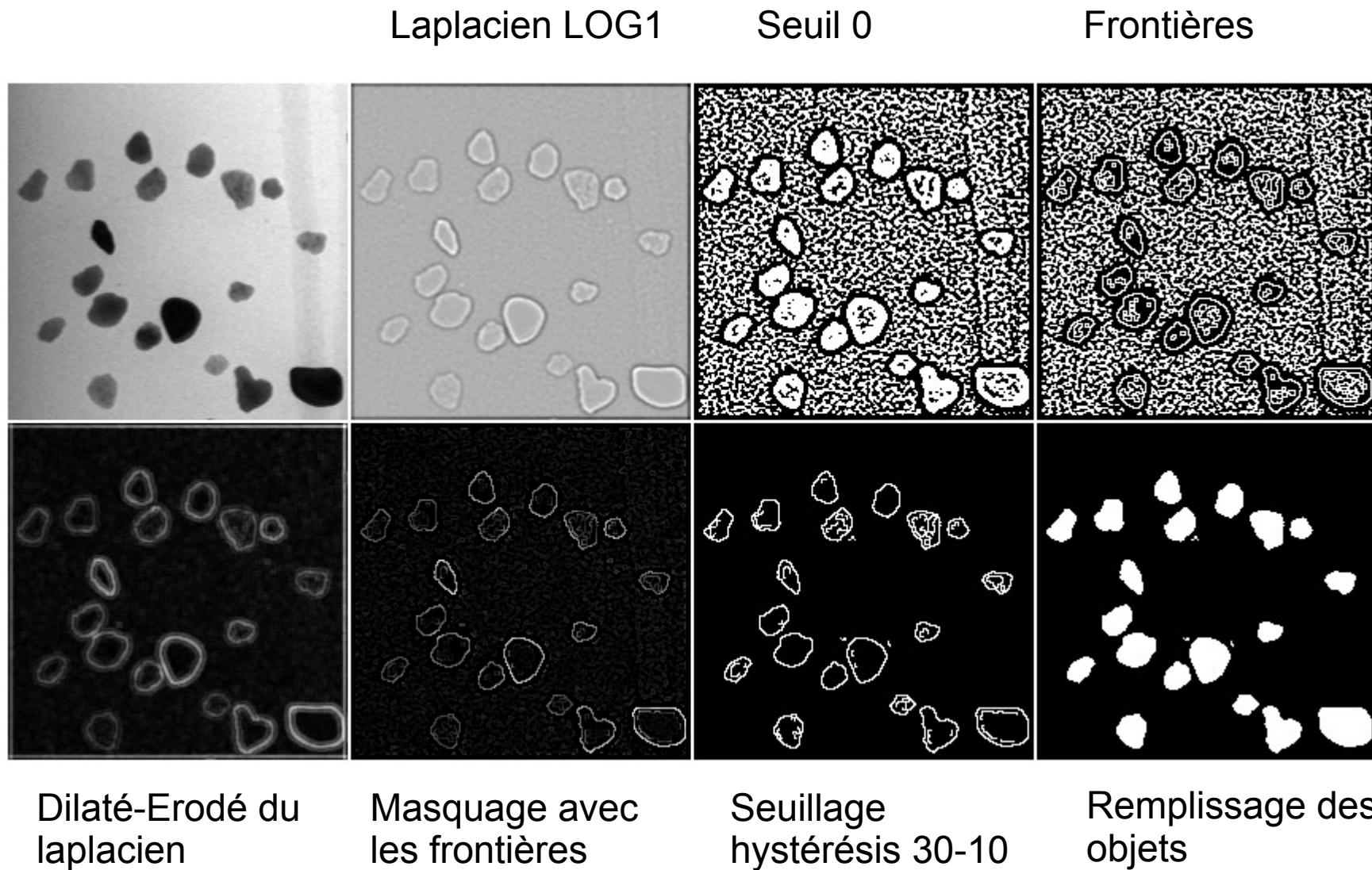
```
% Matlab  
fI=imfilter(double(f),fspecial('log',[13 13],2),'replicate');
```

Détection des contours

- **Principe**
 - Passage par 0 du laplacien
 - seuil à 0 de l'image
 - détection des frontières des régions
 - Problème des passage à 0 non significatifs
 - soit prendre un seuil >0
 - soit mesurer l'amplitude du passage à 0 puis faire un seuillage hystérésis
 - gradient morphologique : dilaté-érodé du laplacien
 - masquage par les frontières des régions

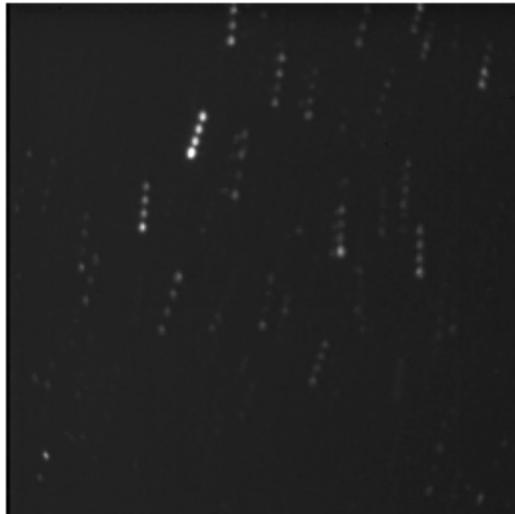
```
% Matlab  
fs=edge(f,'log');
```

Exemple 1 : mesure de l'amplitude des passages à 0

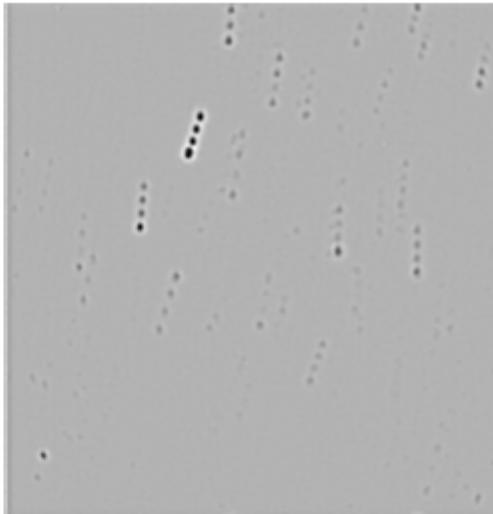


Exemple 2 : détection de petits objets

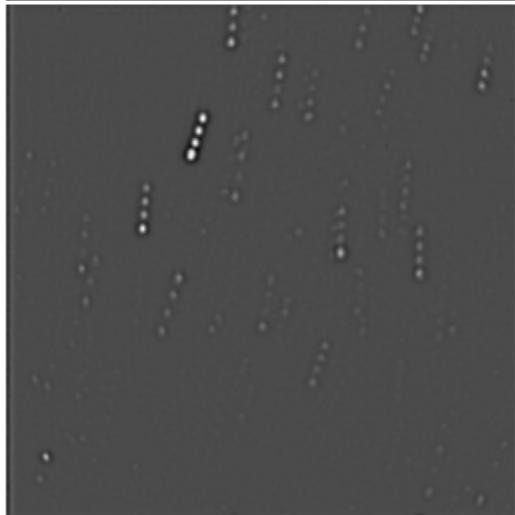
Tomographie
de particules



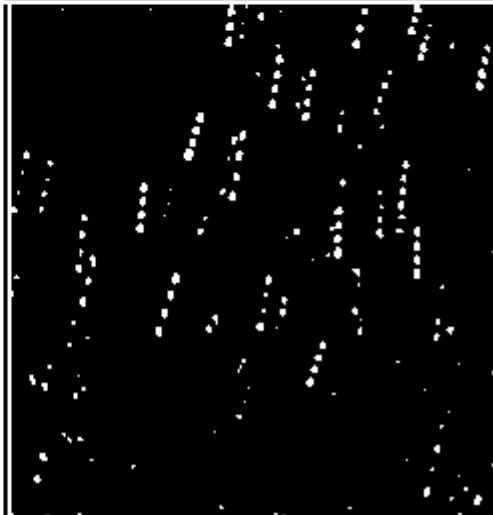
Laplacien
LOG1



$\times (-1)$



Binarisation
(Seuil 4)



Séance 10

6. Détection de contours

6.1 Généralités

6.2 Opérateurs de gradient simple

6.3 Opérateur dérivée de gaussienne

6.4 Détection et fermeture des contours

6.5 Influence du lissage

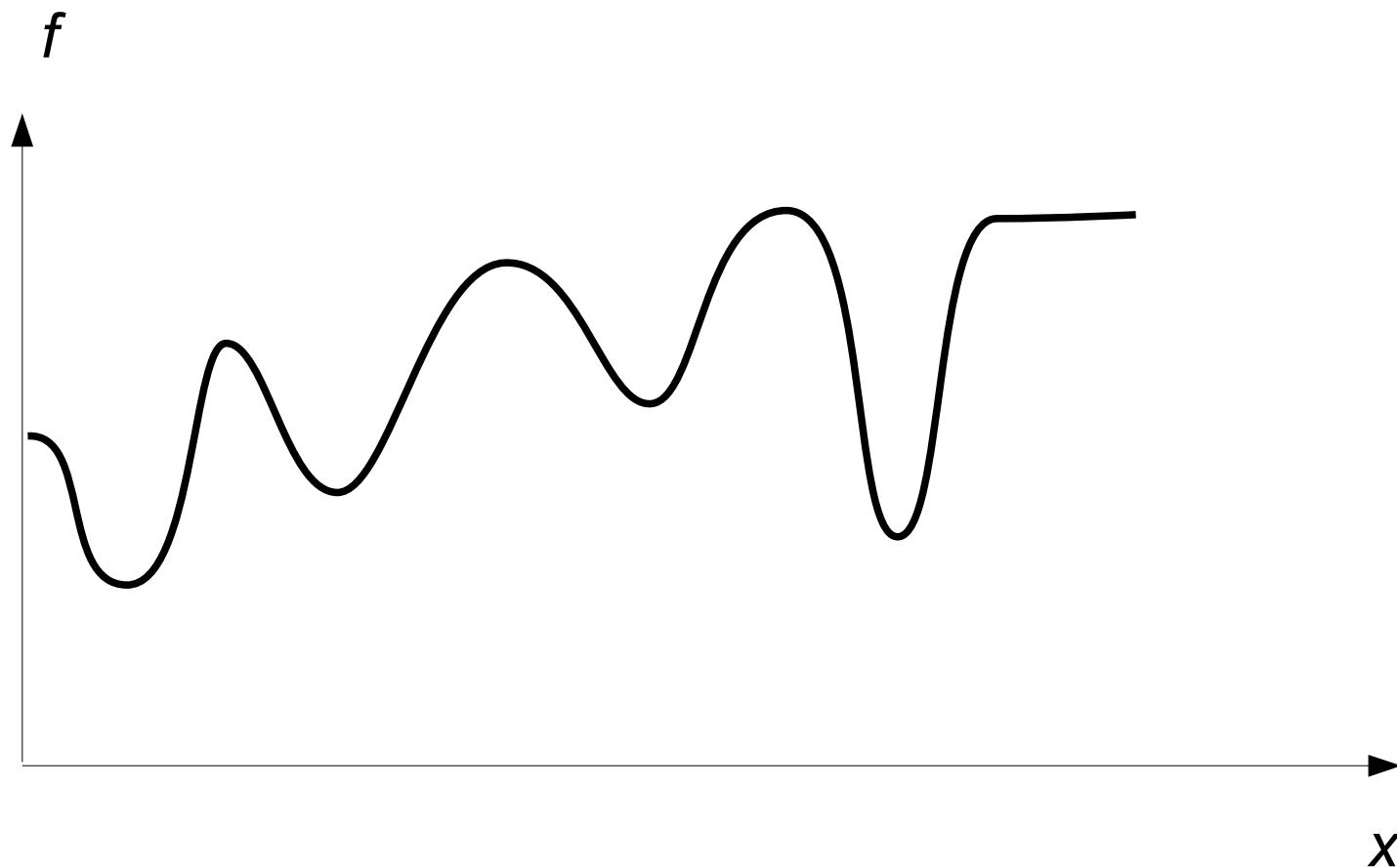
6.6 Laplacien

6.7 Ligne de partage des eaux

Ligne de partage des eaux

- Principe
 - $f = \text{surface}$
 - On perce les minima locaux (trous dans la surface)
 - On fait monter le niveau de l'eau
 - Les bassins se remplissent (inondation)
 - On construit des barrages pour éviter que les bassins se mélangent
 - A la fin, les barrages = ligne de partage des eaux

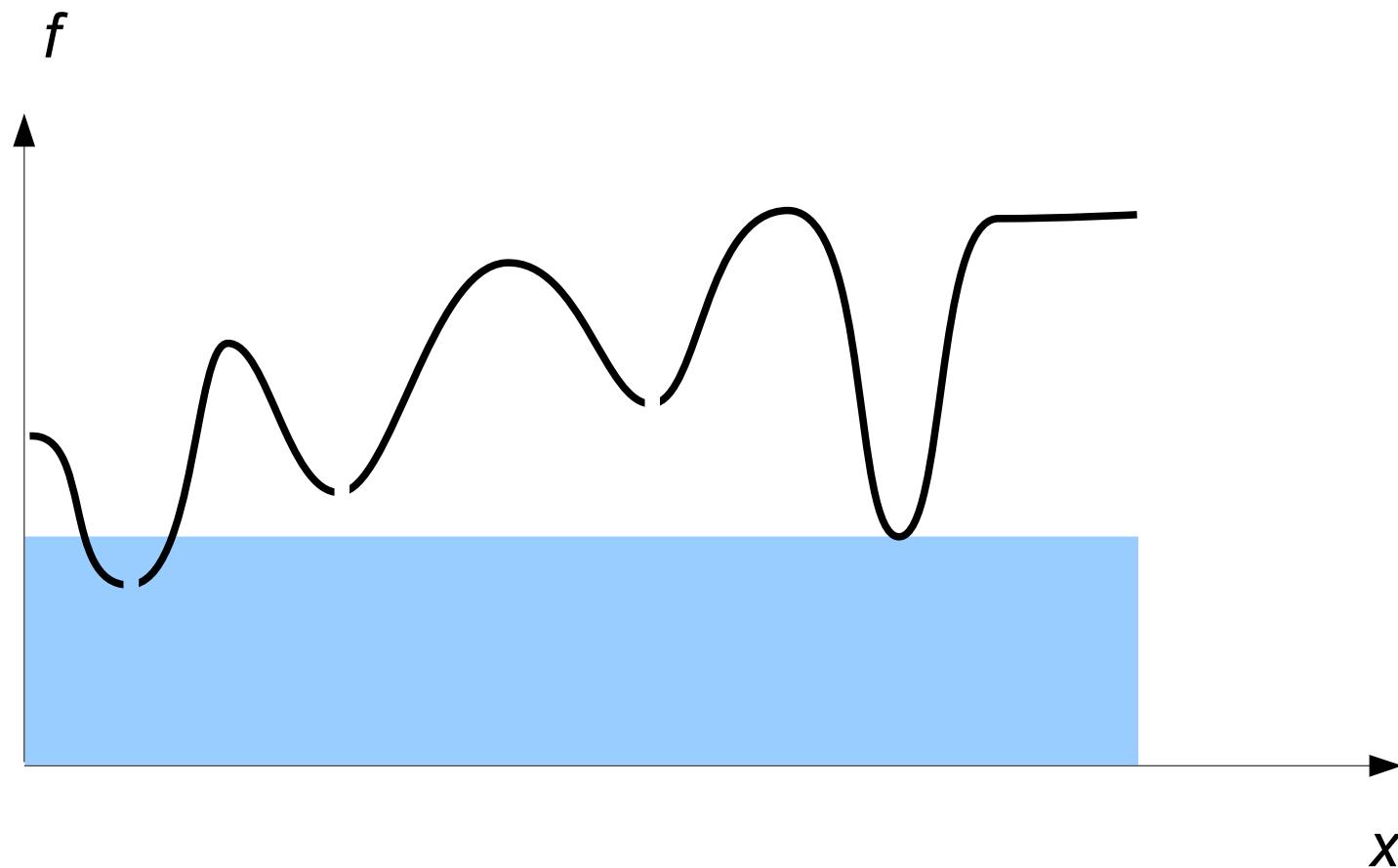
Ligne de partage des eaux



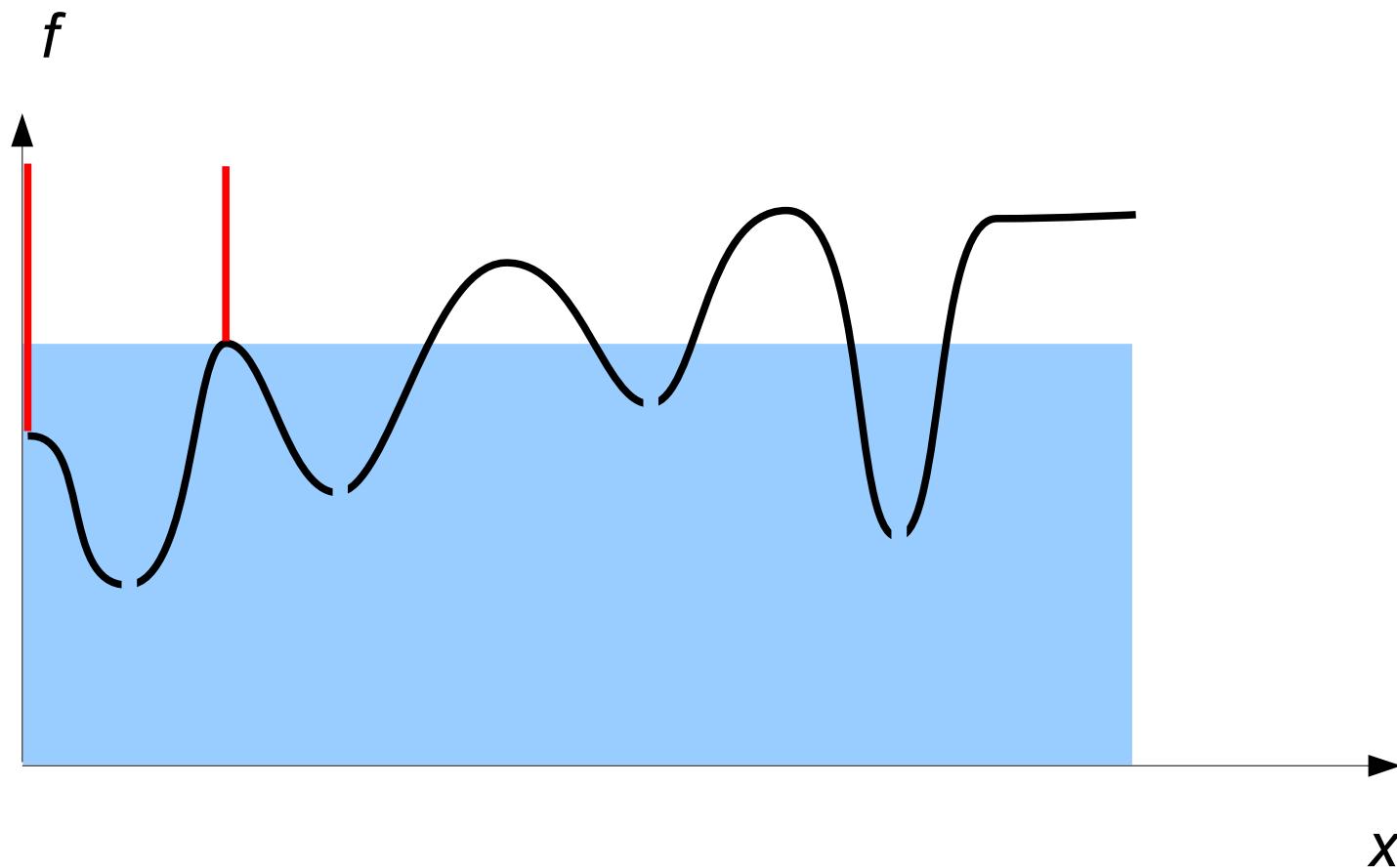
Ligne de partage des eaux



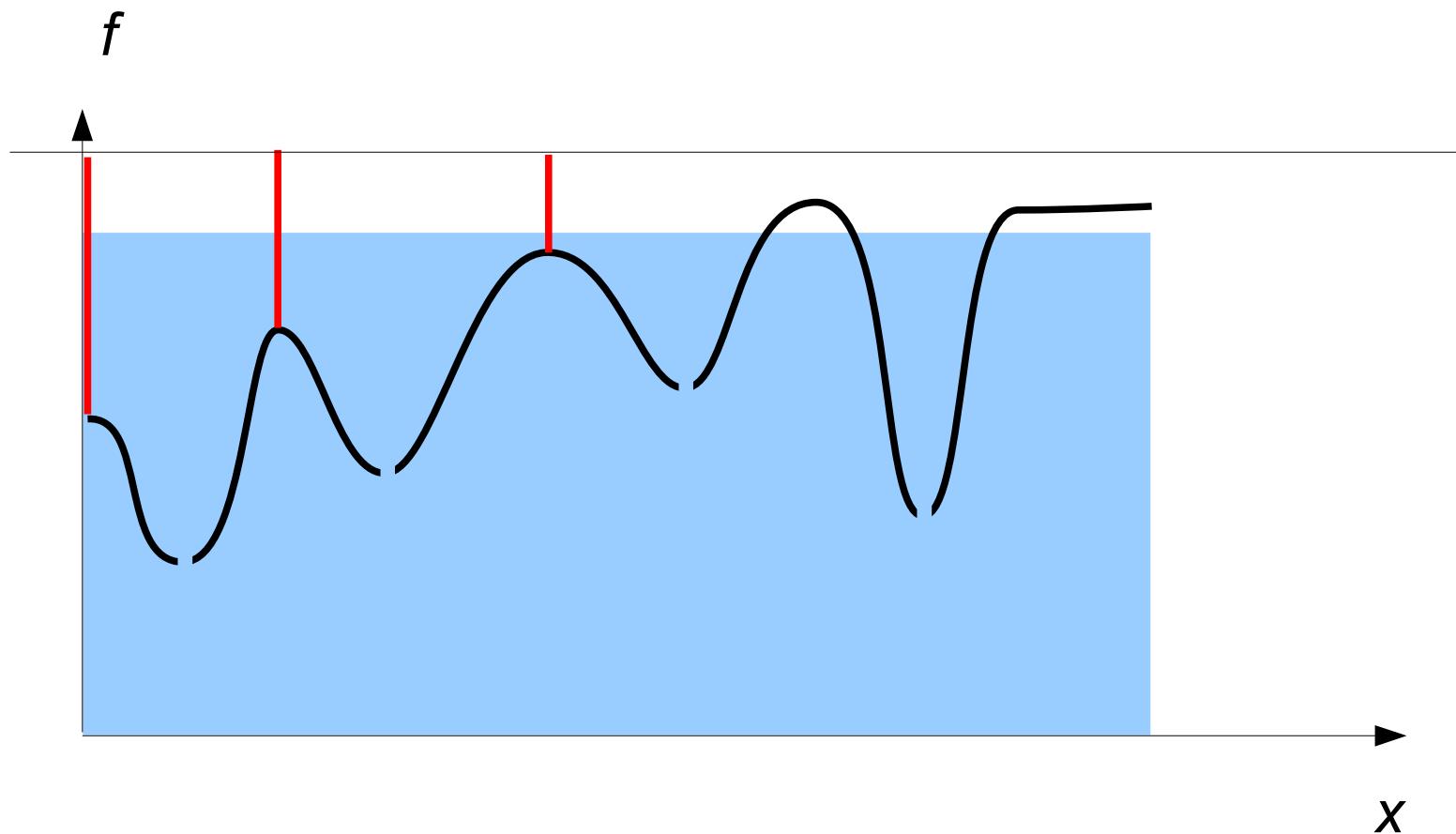
Ligne de partage des eaux



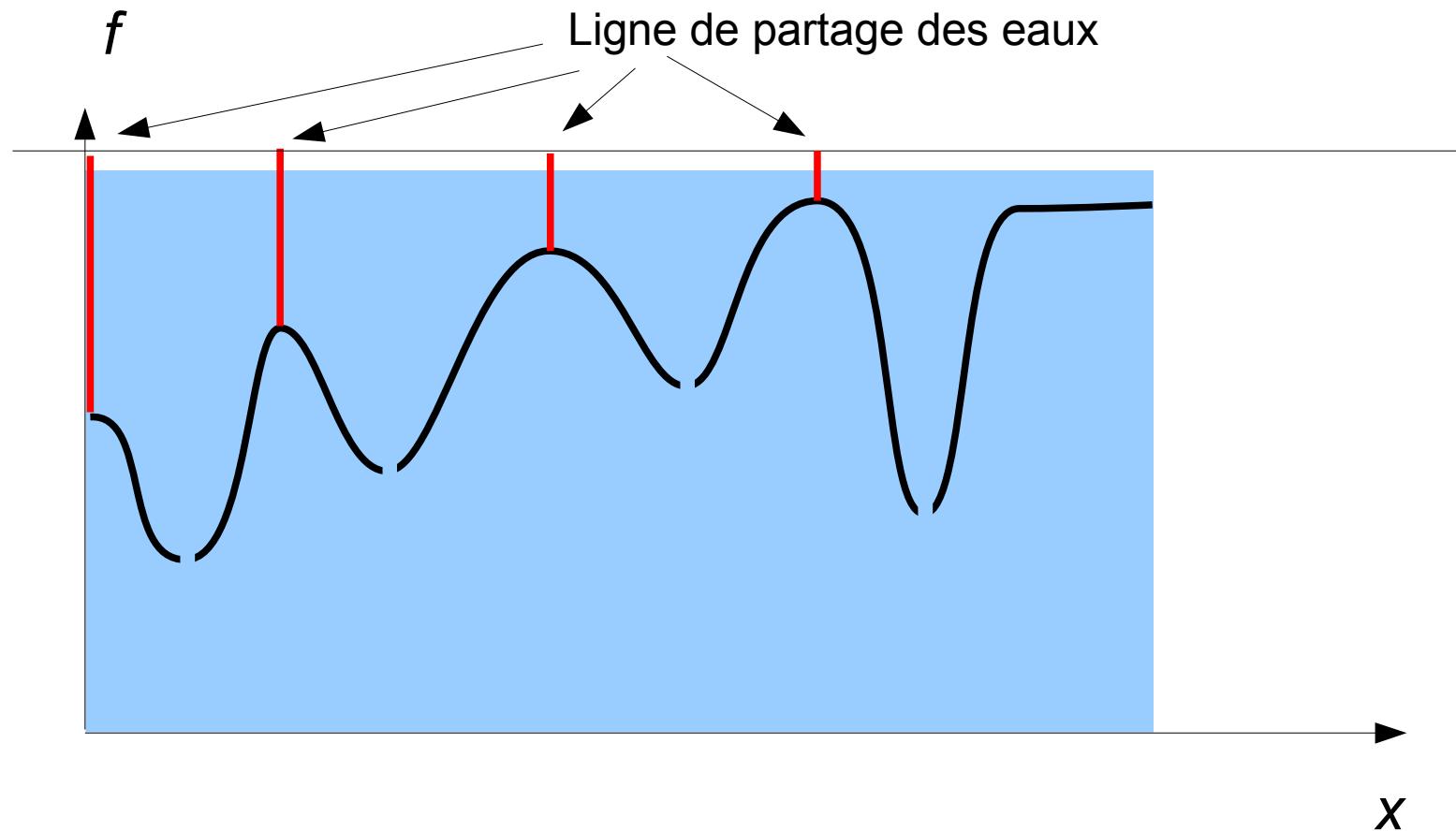
Ligne de partage des eaux



Ligne de partage des eaux



Ligne de partage des eaux



LPE sur une image

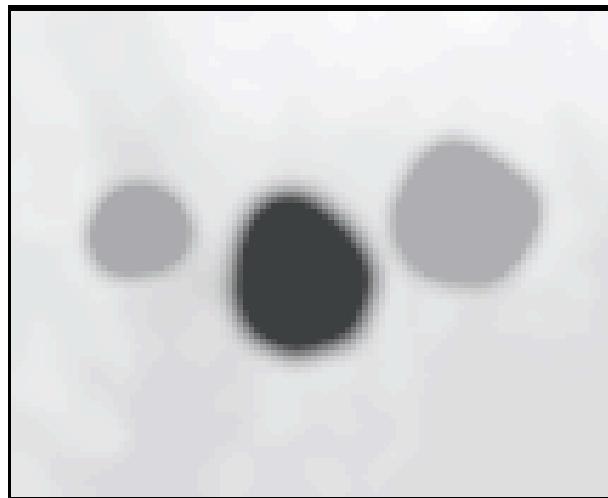


Image originale

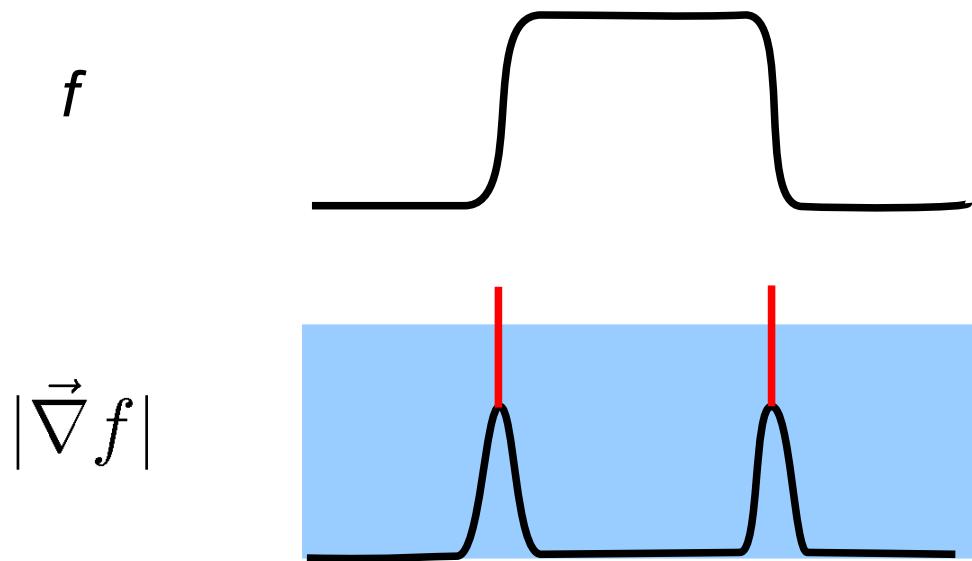


Lpe image

```
lpe=watershed(f);  
figure,imshow(lpe==0);
```

Segmentation par LPE

- Comment faire pour que les bassins soient les objets à segmenter ?
 - Calculer le module du gradient
- LPE du gradient = segmentation



Segmentation par LPE : exemple 2D

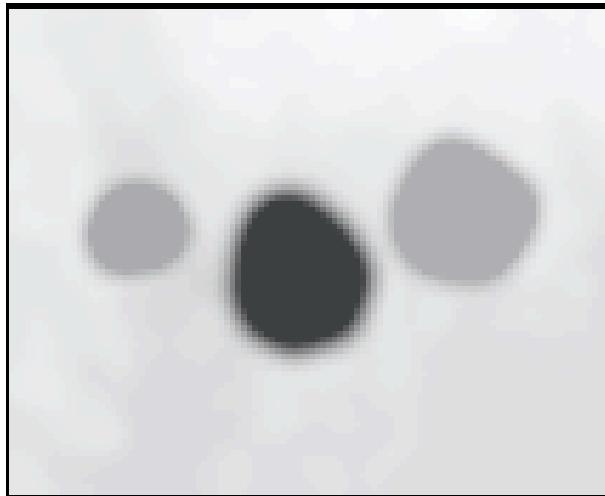
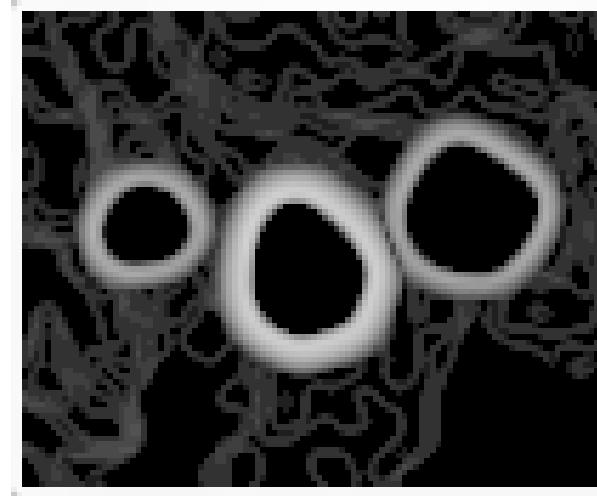


Image originale



Module du gradient



LPE

Avantages de la segmentation par LPE

- Unicité de la LPE
- Peut être calculée sur des grandeurs diverses :
 - Niveau de gris,
 - Gradient,
 - ...
- Détection de la ligne de crête du gradient
 - Meilleurs résultats que les maxima locaux du gradient
- Algorithmes efficaces
 - Cf. cours de morphologie

Problème de sélection des minima locaux

- En pratique : sur-segmentation

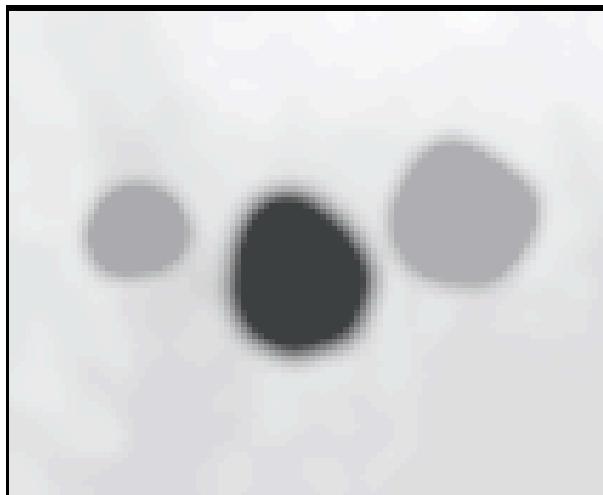
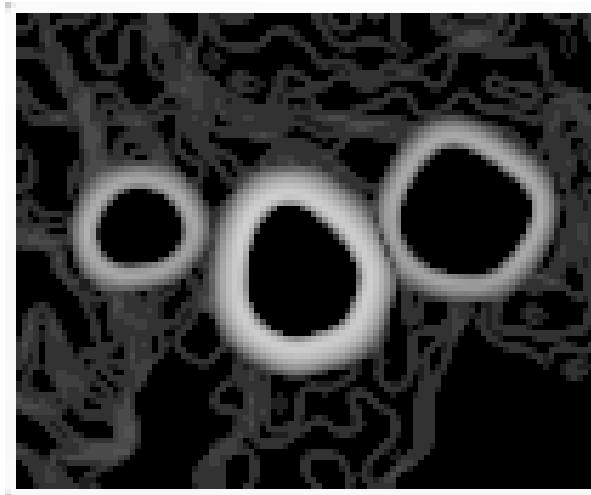


Image originale



Module du gradient



LPE

- Trop de bassins !

Problème de sélection des minima locaux

- Sur-segmentation
 - Chaque minimum local produit un bassin
 - Chaque bassin produit une région différent
 - Présence de minima de faible amplitude
 - Nombre important
- Solutions
 1. Supprimer les minima de faible amplitude : h_{min}
 2. Sélectionner les minima à garder : marqueurs

Utilisation d'opérateurs géodésiques

- Suppression des minima de hauteur inférieure à h :

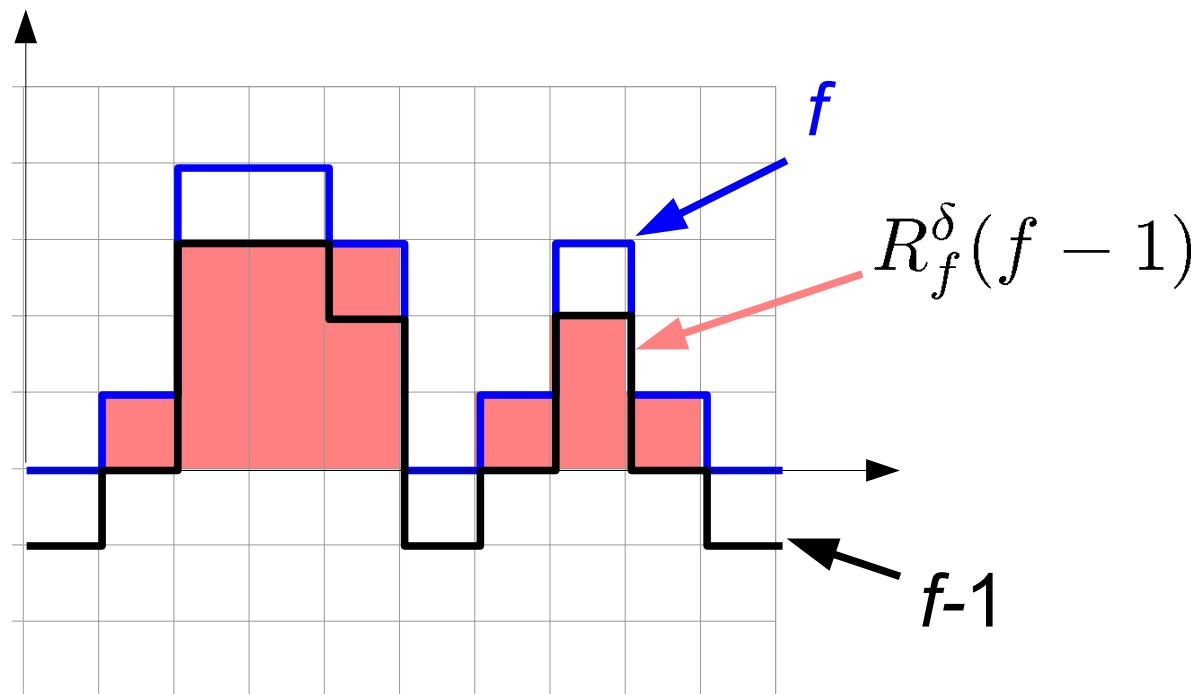
$$\text{HMin}_h(f) = R_f^\epsilon(f + h)$$

- Détection des maxima régionaux ou régionaux étendus :

$$\text{RMax}_h(f) = f - R_f^\delta(f - 1)$$

$$\text{EMax}_h(f) = \text{RMax}(\text{HMax}_h(f))$$

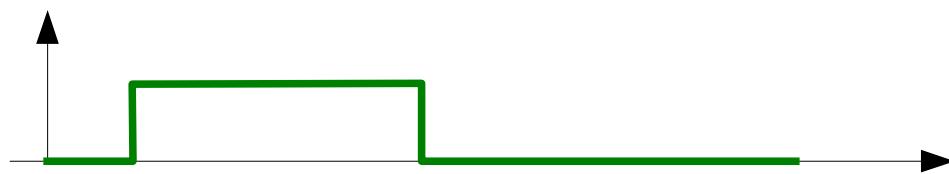
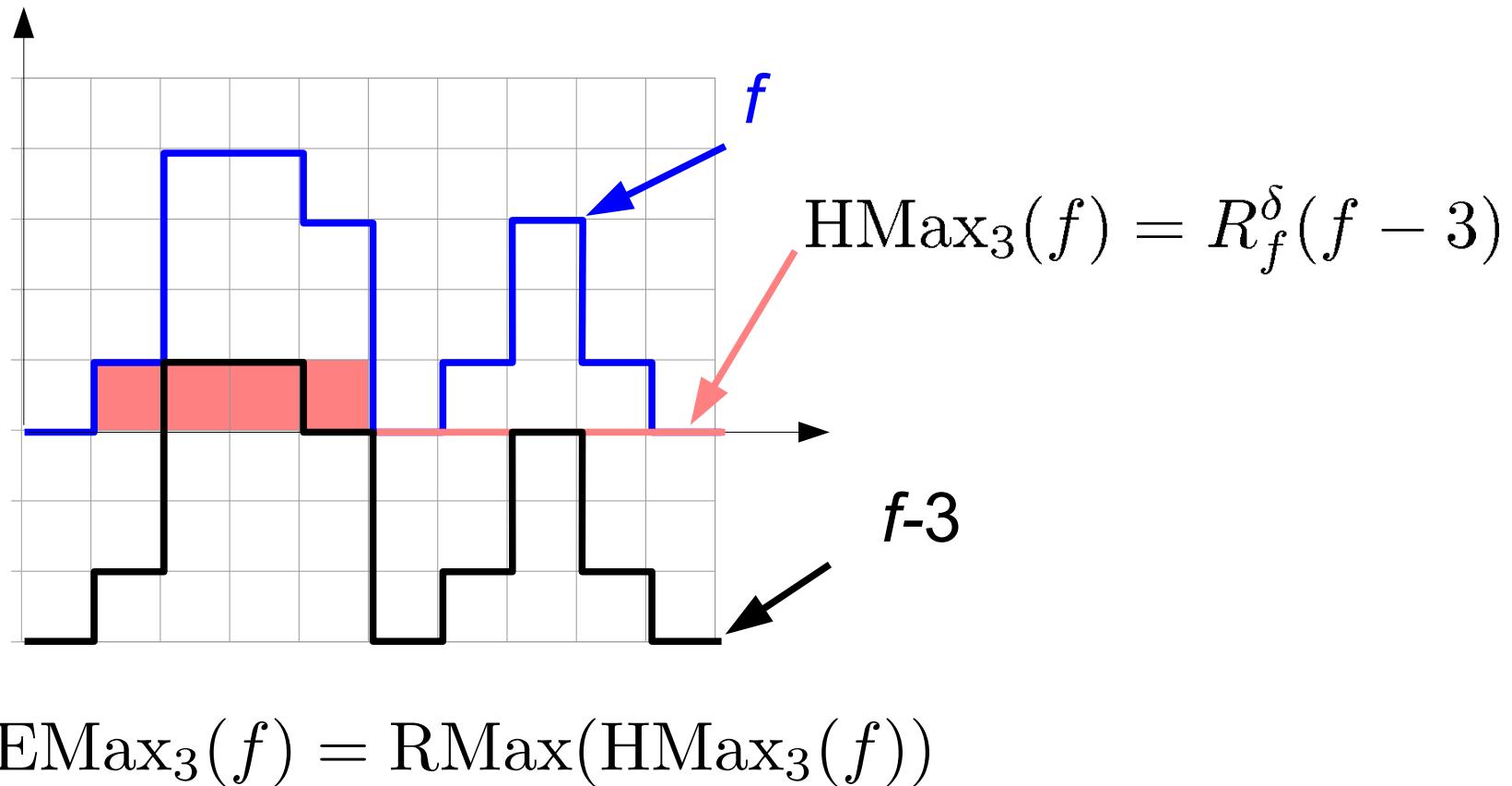
Maxima régionaux : exemple



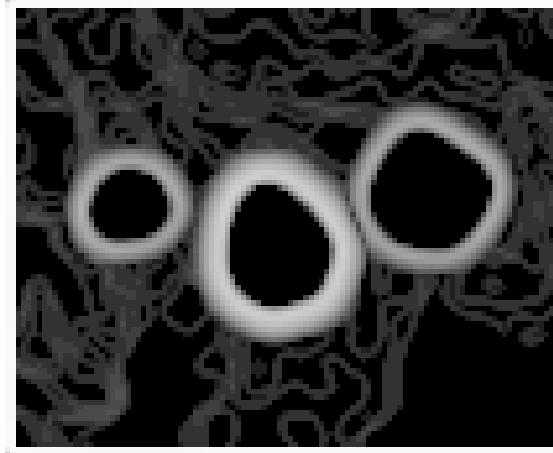
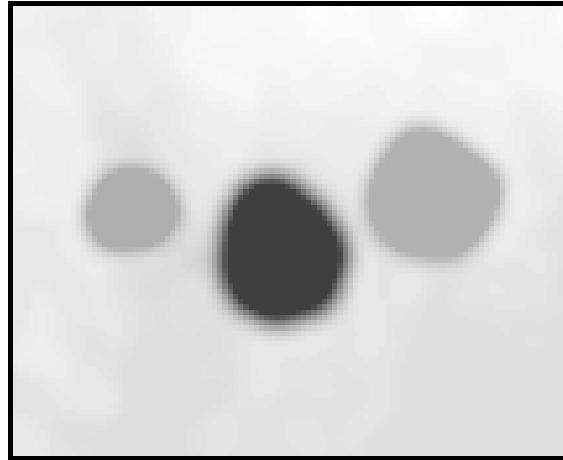
$$\text{RMax}(f) = f - R_f^\delta(f - 1)$$



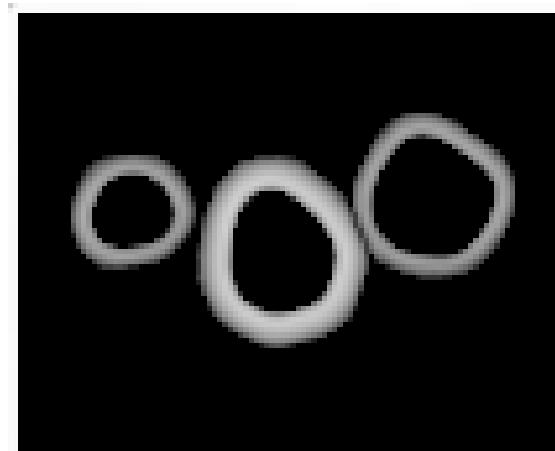
Maxima régionaux étendus : exemple



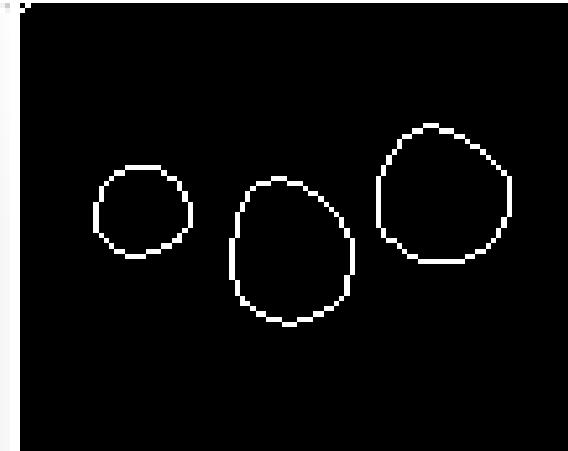
Solution 1 : suppression des minima non significatifs



Module du gradient (GOG1)

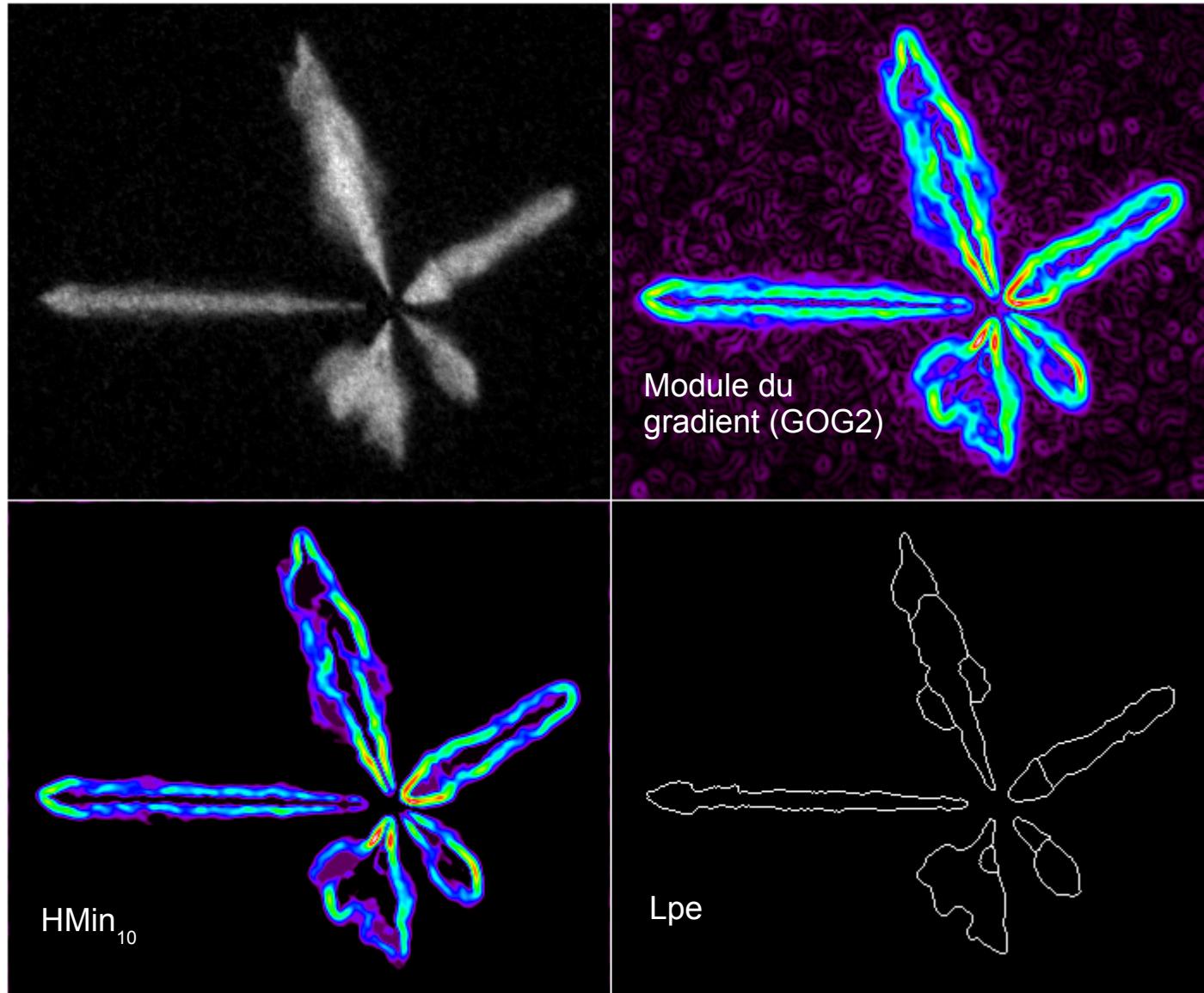


$HMin_{10}$



Lpe

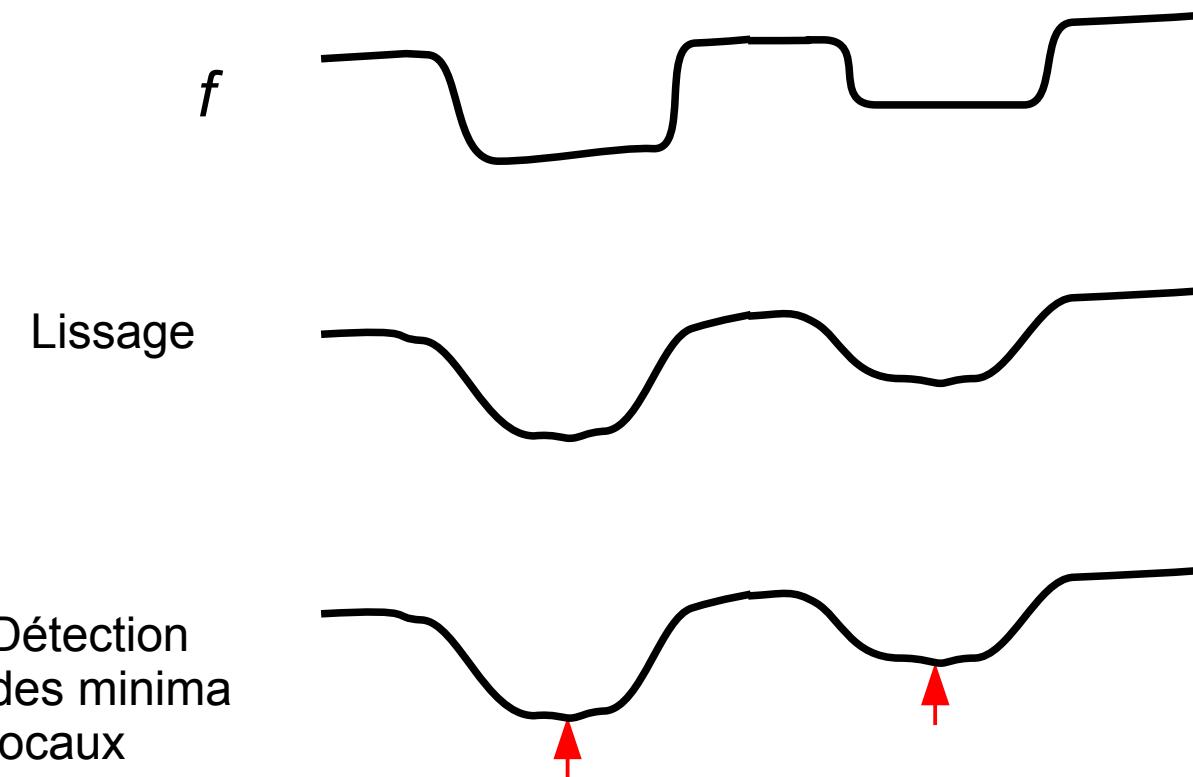
Solution 1 : autre exemple



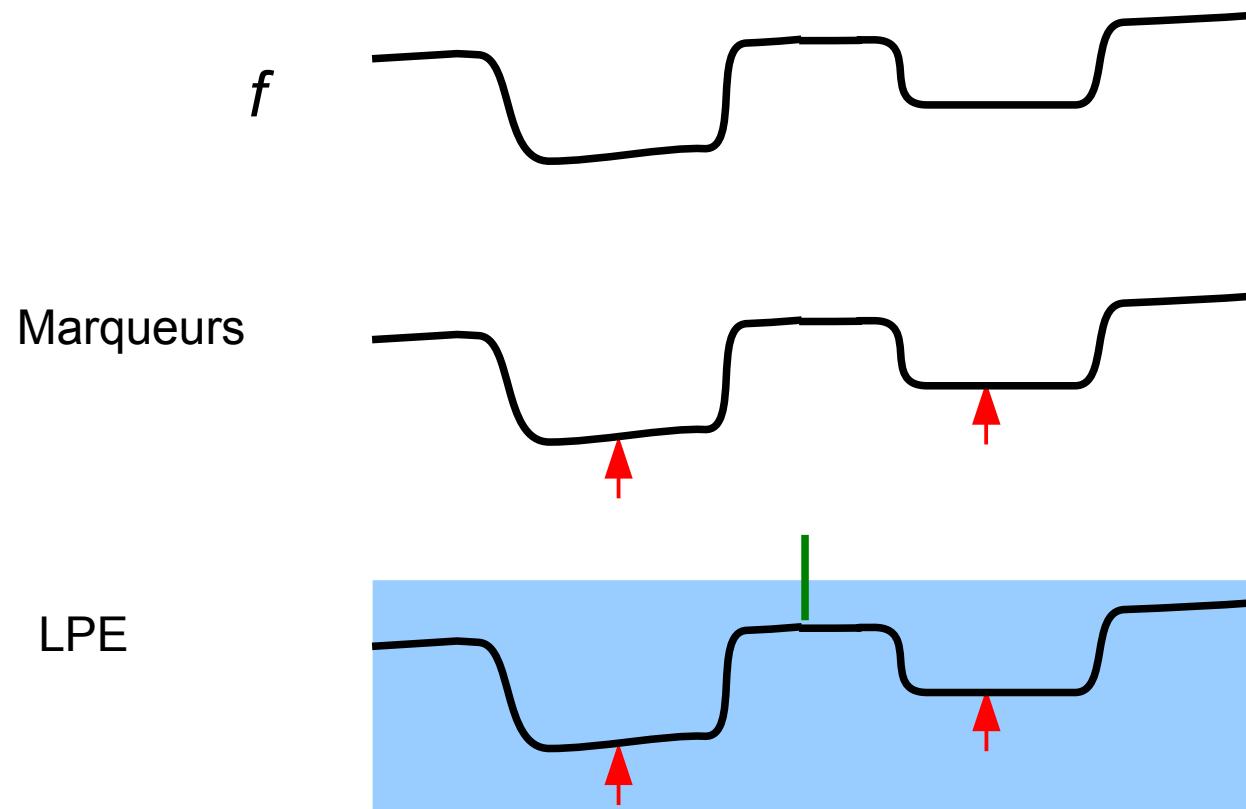
Solution 2 : utilisation de marqueurs

- Sélection des marqueurs internes
 - A partir de l'image originale
 - Min ou max régionaux étendus
 - Pré-segmentation
- Sélection des marqueurs externes
 - Déduits à partir des marqueurs internes
 - LPE sur l'image originale
 - Avec marqueurs internes
- LPE sur le gradient
 - Avec marqueurs internes+externes

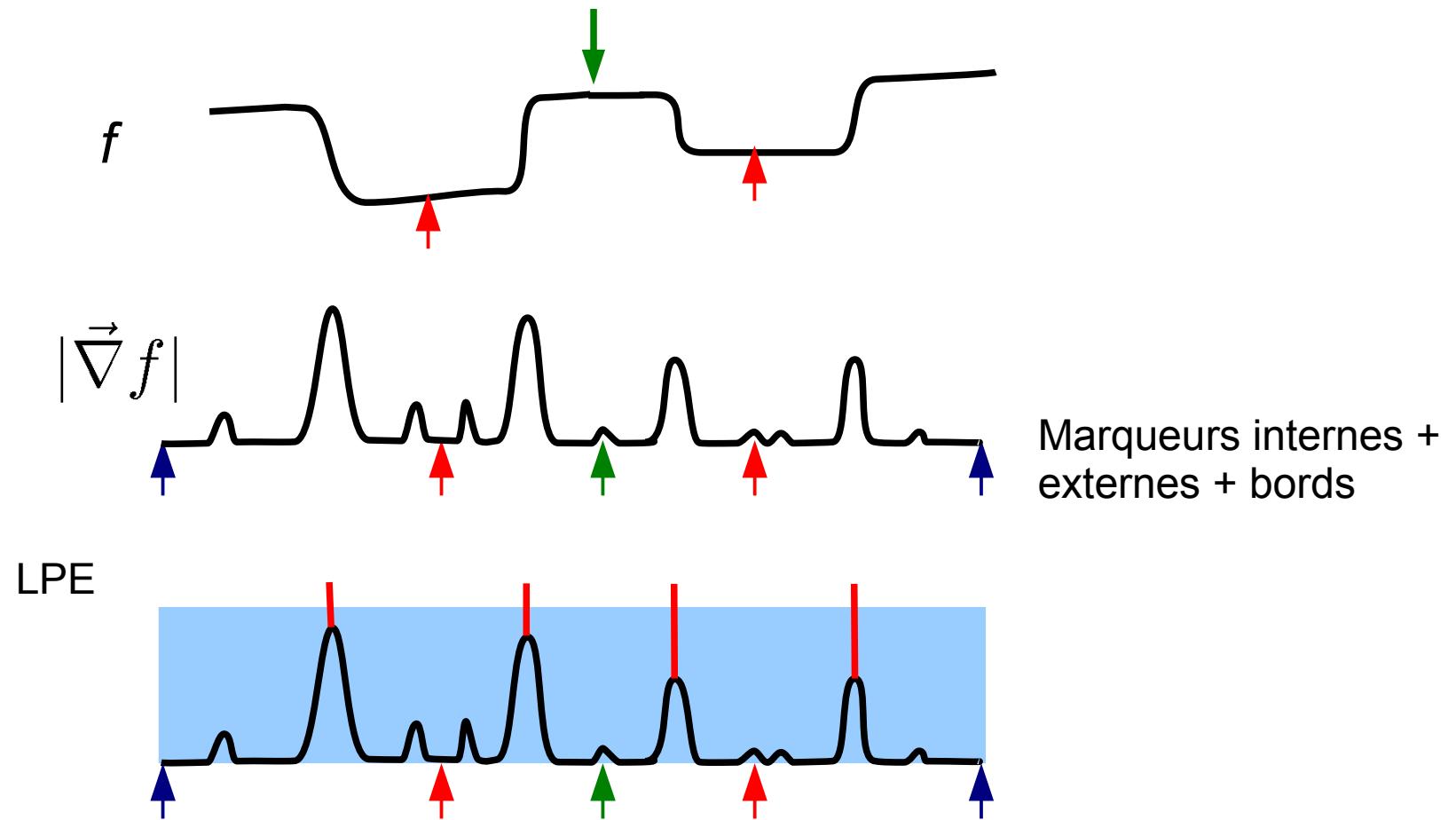
Sélection des marqueurs internes



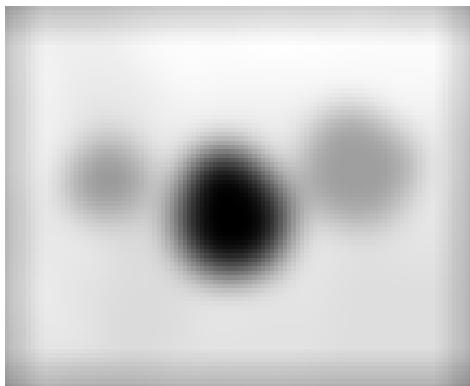
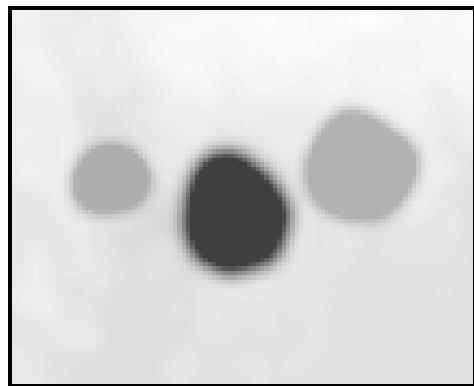
Sélection des marqueurs externes



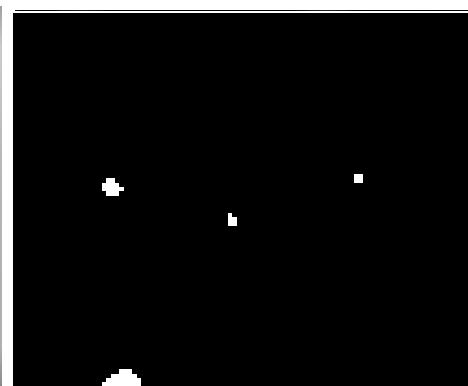
Principe de LPE avec marqueurs



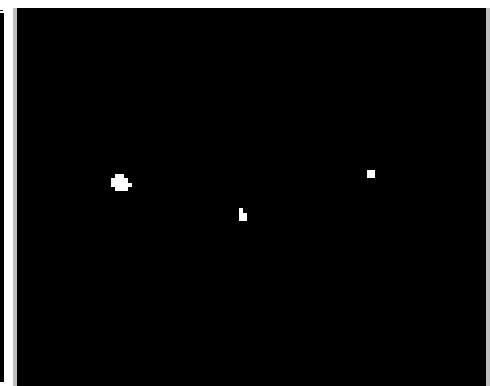
Exemple 1



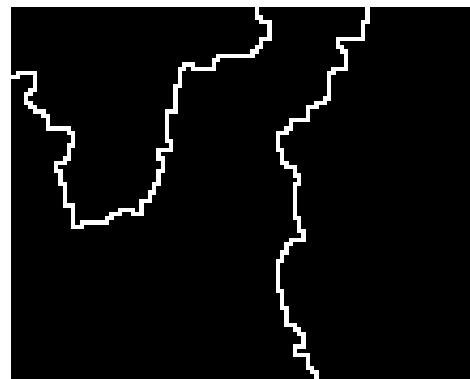
Lissage



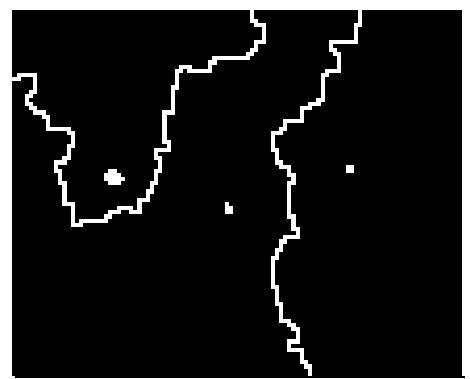
EMin



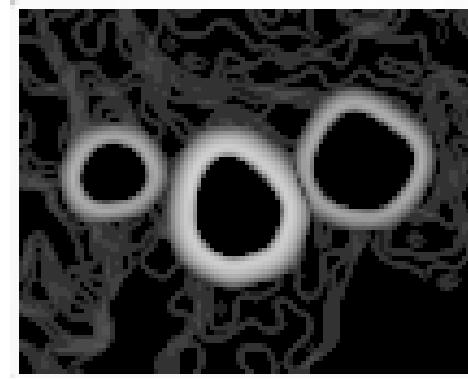
Suppression des objets
touchant le bord



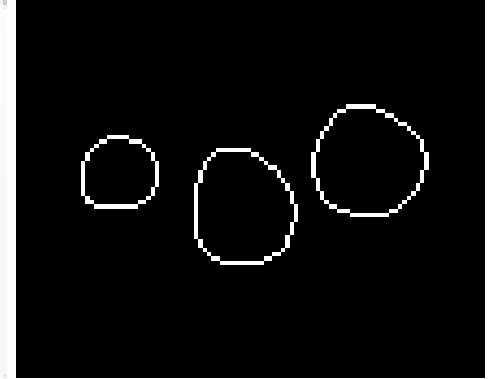
Marqueurs externes
obtenus par LPE sur
l'image initiale



Fusion des
marqueurs +
bord

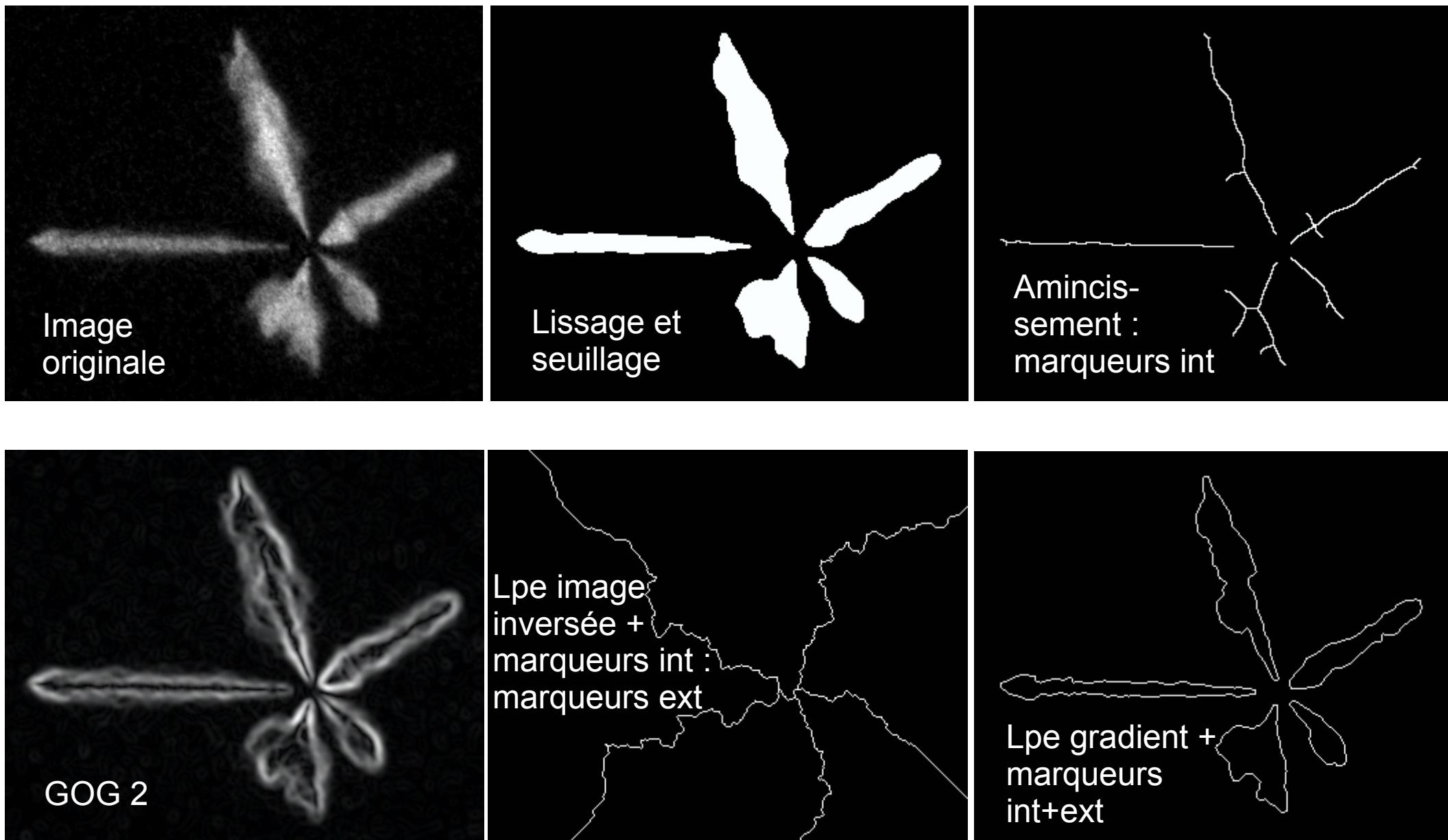


Gradient



LPE sur le gradient
avec marqueurs
int+ext

Exemple 2



Séance 11

Diapositives empruntées à
Damien Muselet

Plan

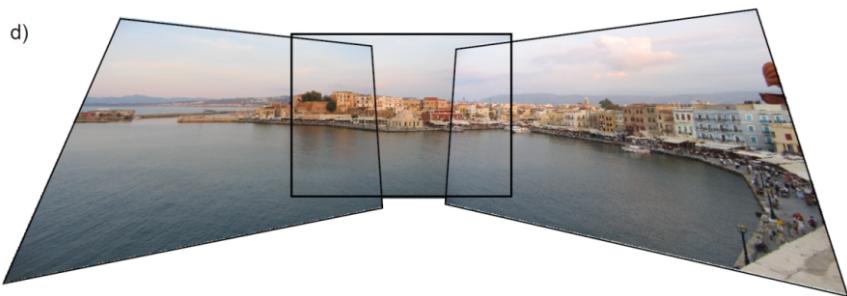
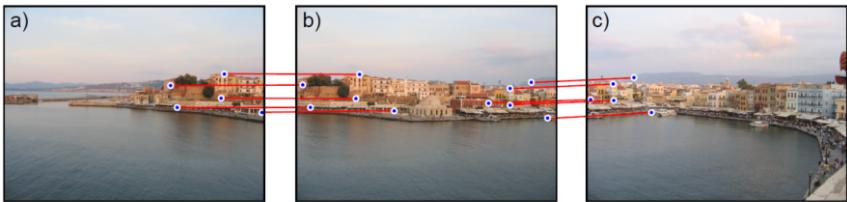
1. Introduction
2. Pré-traitements
3. Seuillage automatique
4. Mesure de paramètres
5. Segmentation par classification
6. Détection de contours
7. Points d'intérêts et mise en correspondance

Points d'intérêt

- Points « caractéristiques » invariants
 - Changement de point de vue
 - Transformations géométriques
 - Changement d'illumination
- Utilisés en vision par ordinateur pour :
 - Mise en correspondance d'images (stéréoscopie,...)
 - Détection de mouvement
 - Assemblage d'images
 - Reconnaissance d'objets
 - ...

Applications des points d'intérêts

Détection et mise en correspondance



Reconstruction 3D



La classification d'images par sac de mots



Trois problèmes étudiés

1. Détection de points caractéristiques

- DéTECTeur de Harris
- Différence de gaussiennes
- Harris-Laplace

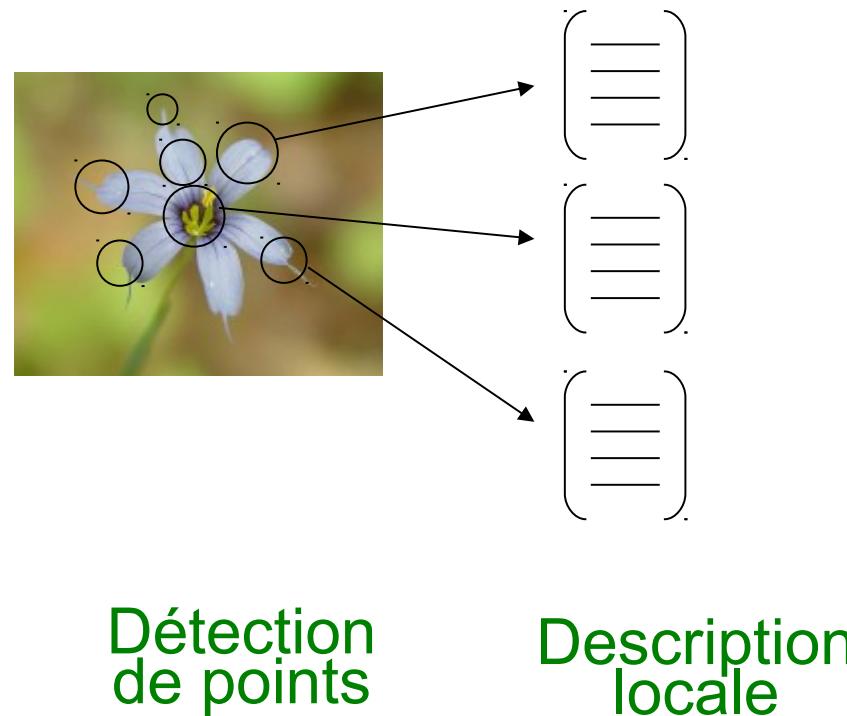
2. Description locale

- Descripteur SIFT

3. Mise en correspondance

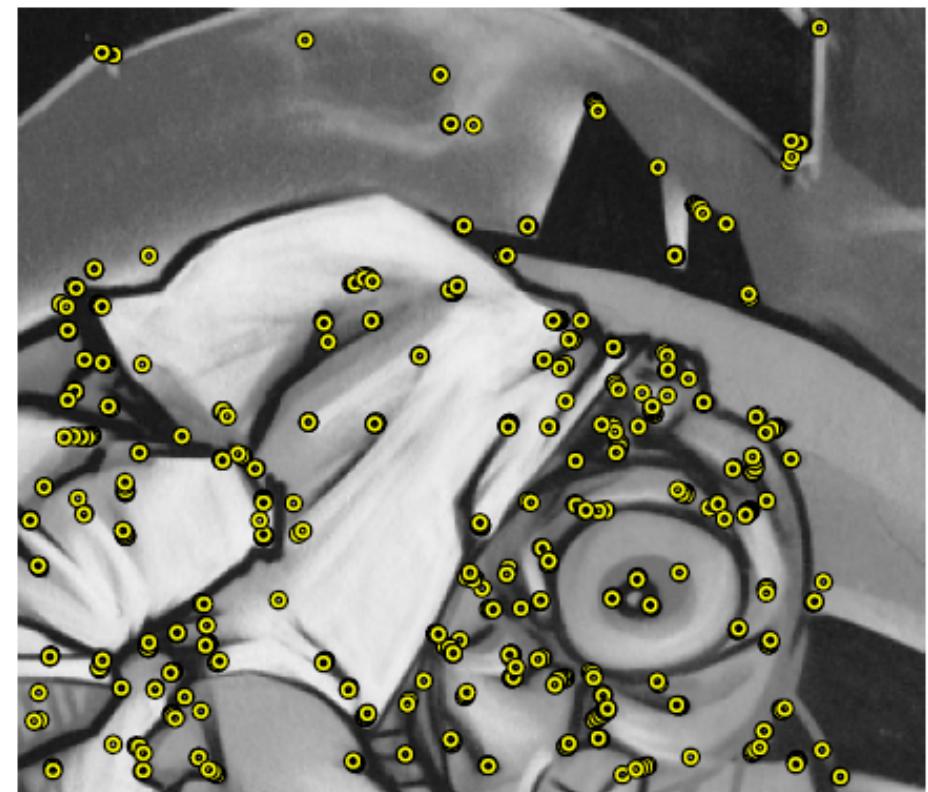
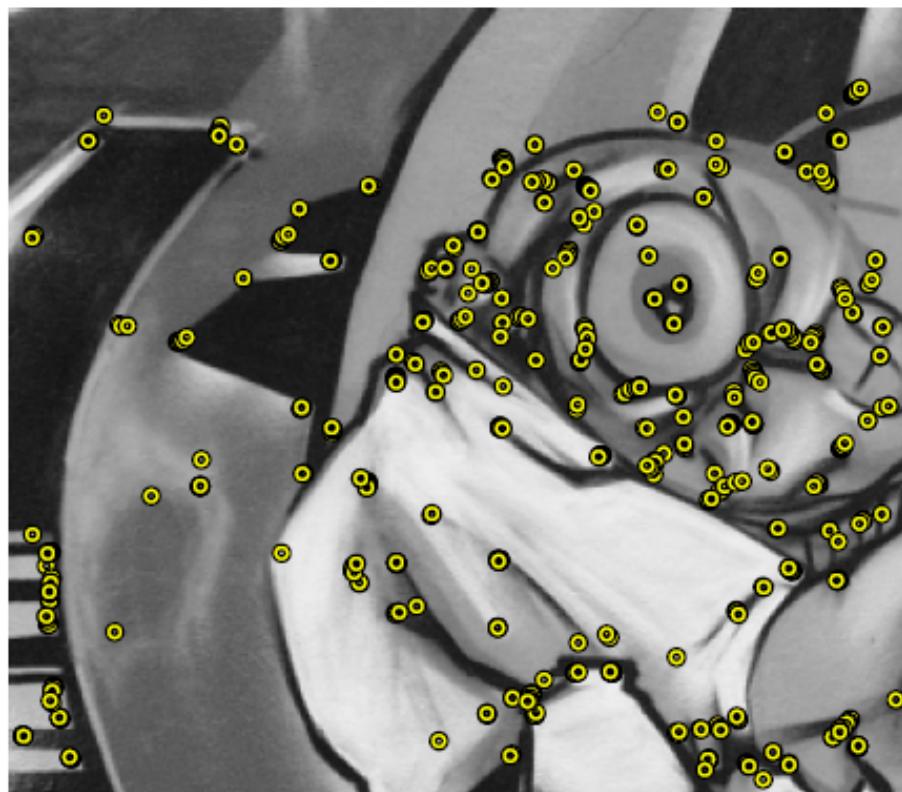
- Matrice et géométrie
- Mise en correspondance robuste

Description locale

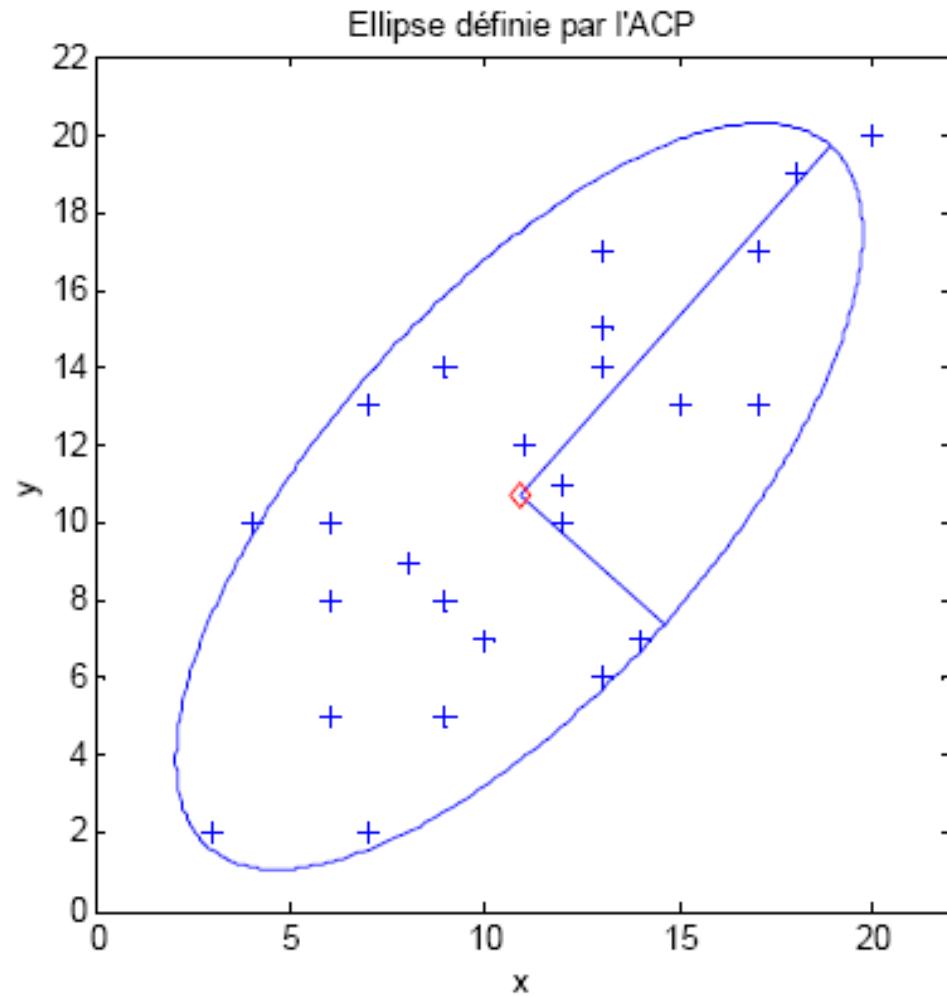


Détection de points

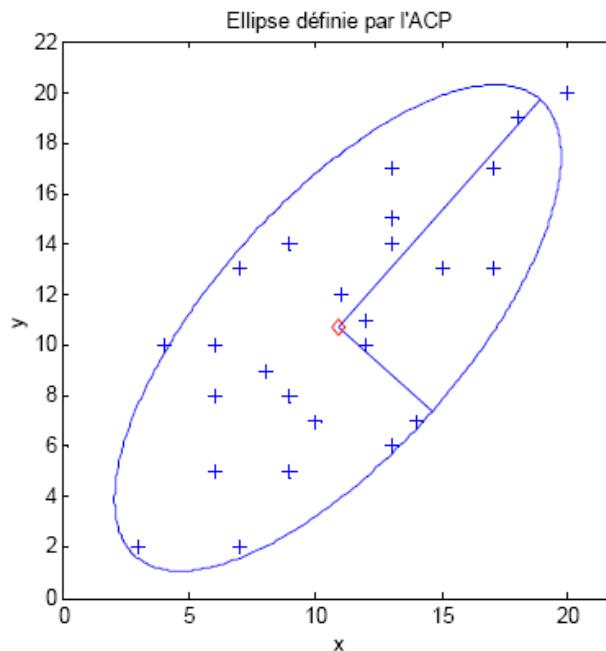
Le détecteur de Harris



Analyse en Composantes Principales



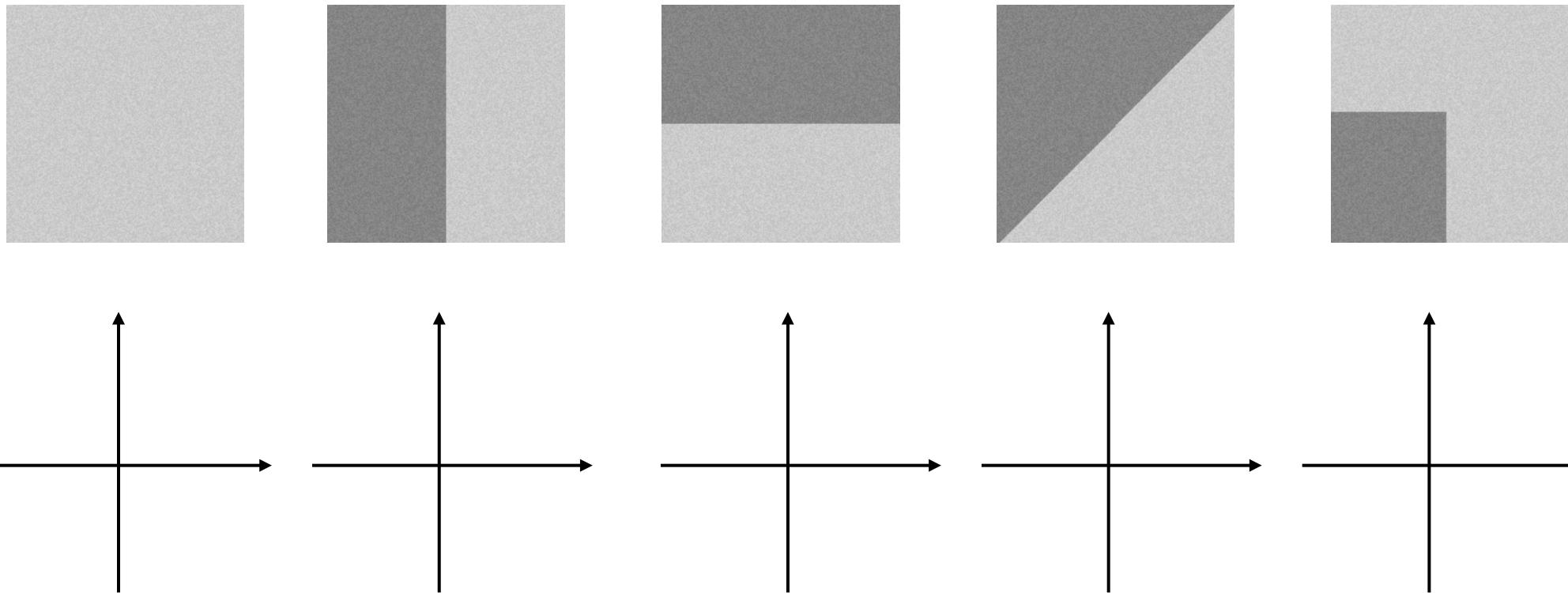
Analyse en Composantes Principales



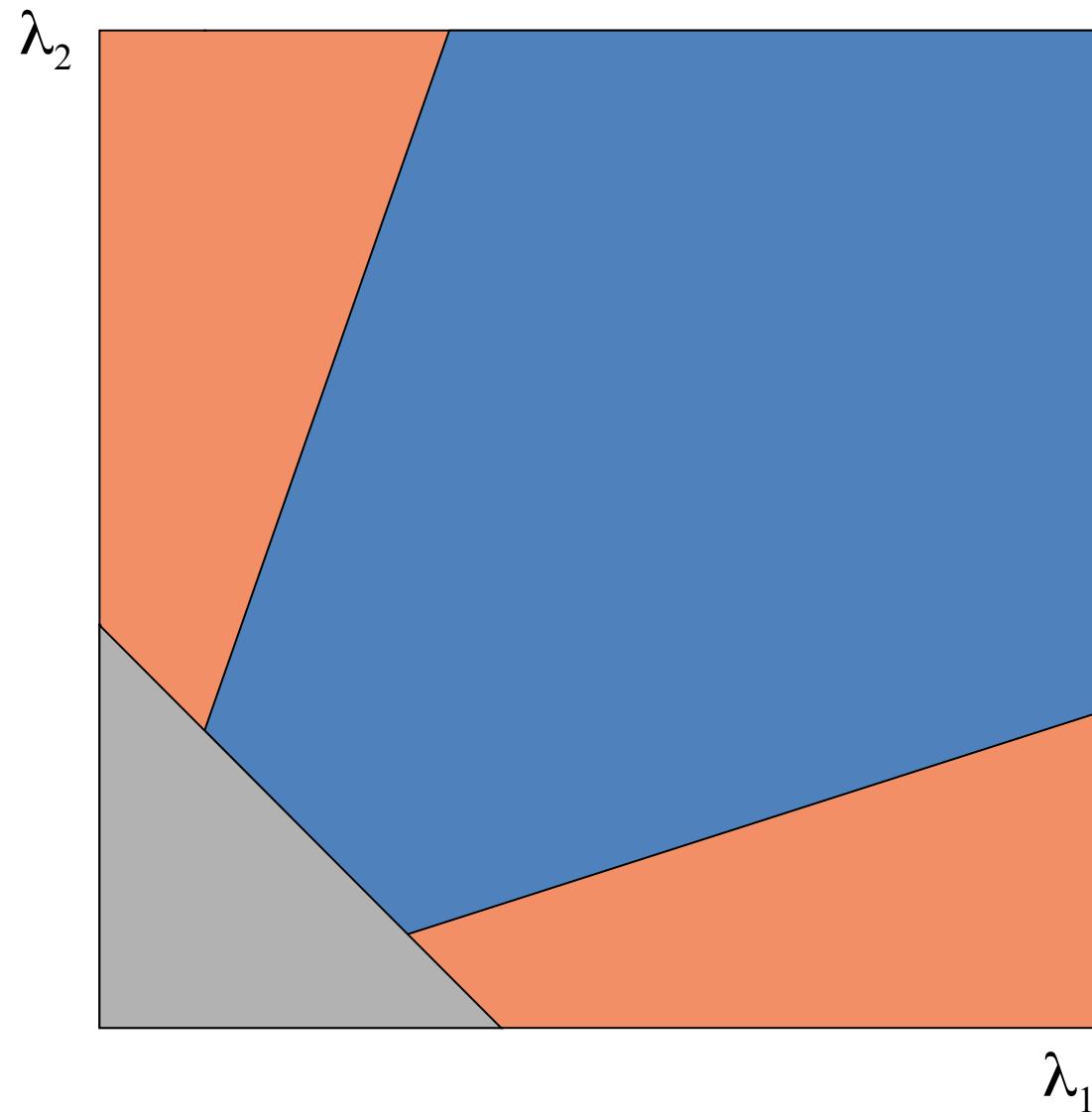
$$M_{vc} = \begin{bmatrix} \sum_i^N (x_i - m_x)^2 & \sum_i^N (x_i - m_x)(y_i - m_y) \\ \sum_i^N (x_i - m_x)(y_i - m_y) & \sum_i^N (y_i - m_y)^2 \end{bmatrix}.$$

⇒ Vecteurs propres et valeurs propres de la matrice M_{vc}

Le détecteur de Harris

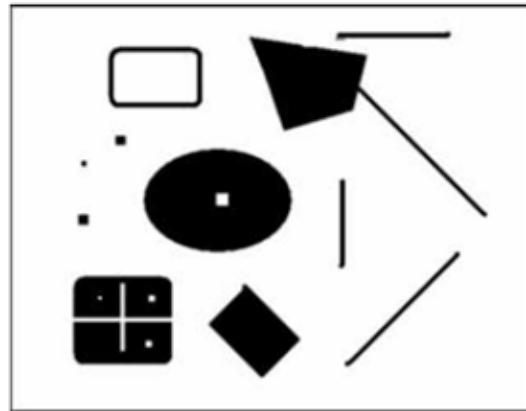


Le détecteur de Harris



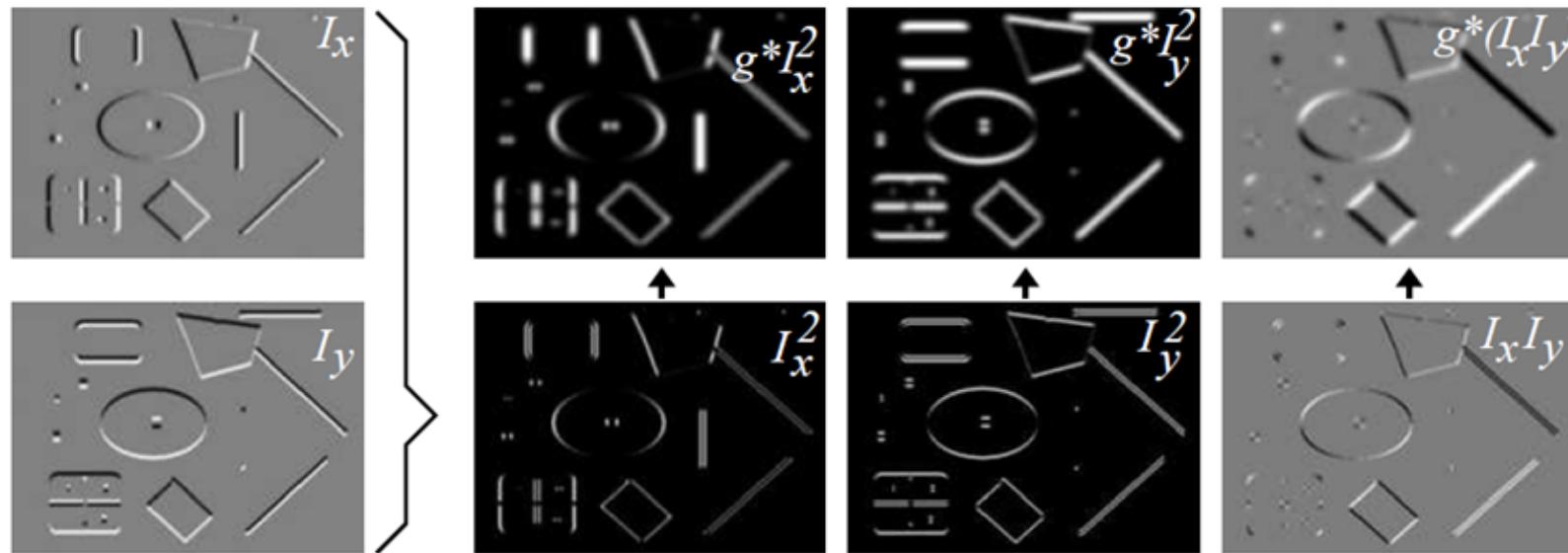
Détection de points

Le détecteur de Harris



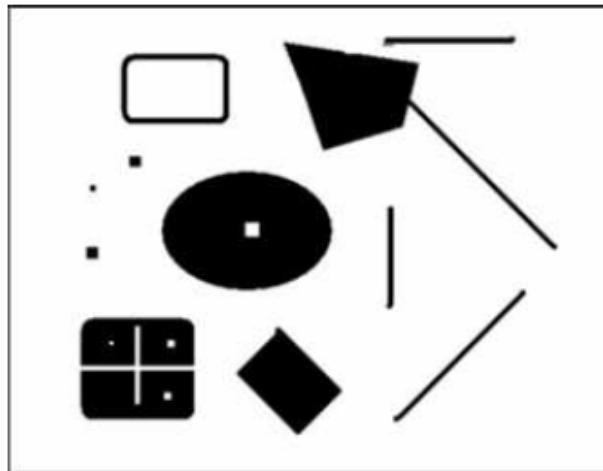
Matrice des moments d'ordre 2

$$M = \begin{bmatrix} g^*I_x^2 & g^*(I_x I_y) \\ g^*(I_x I_y) & g^*I_y^2 \end{bmatrix}$$



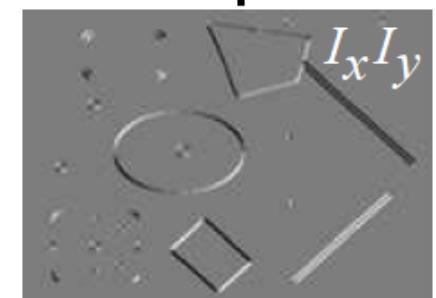
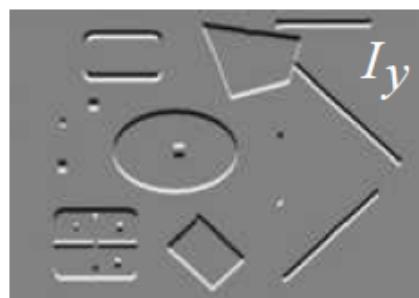
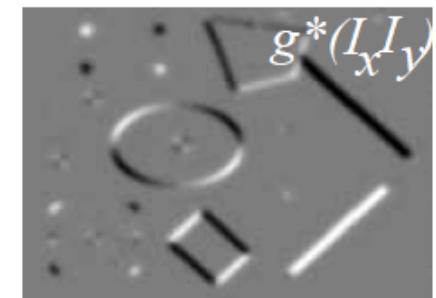
Détection de points

Le détecteur de Harris



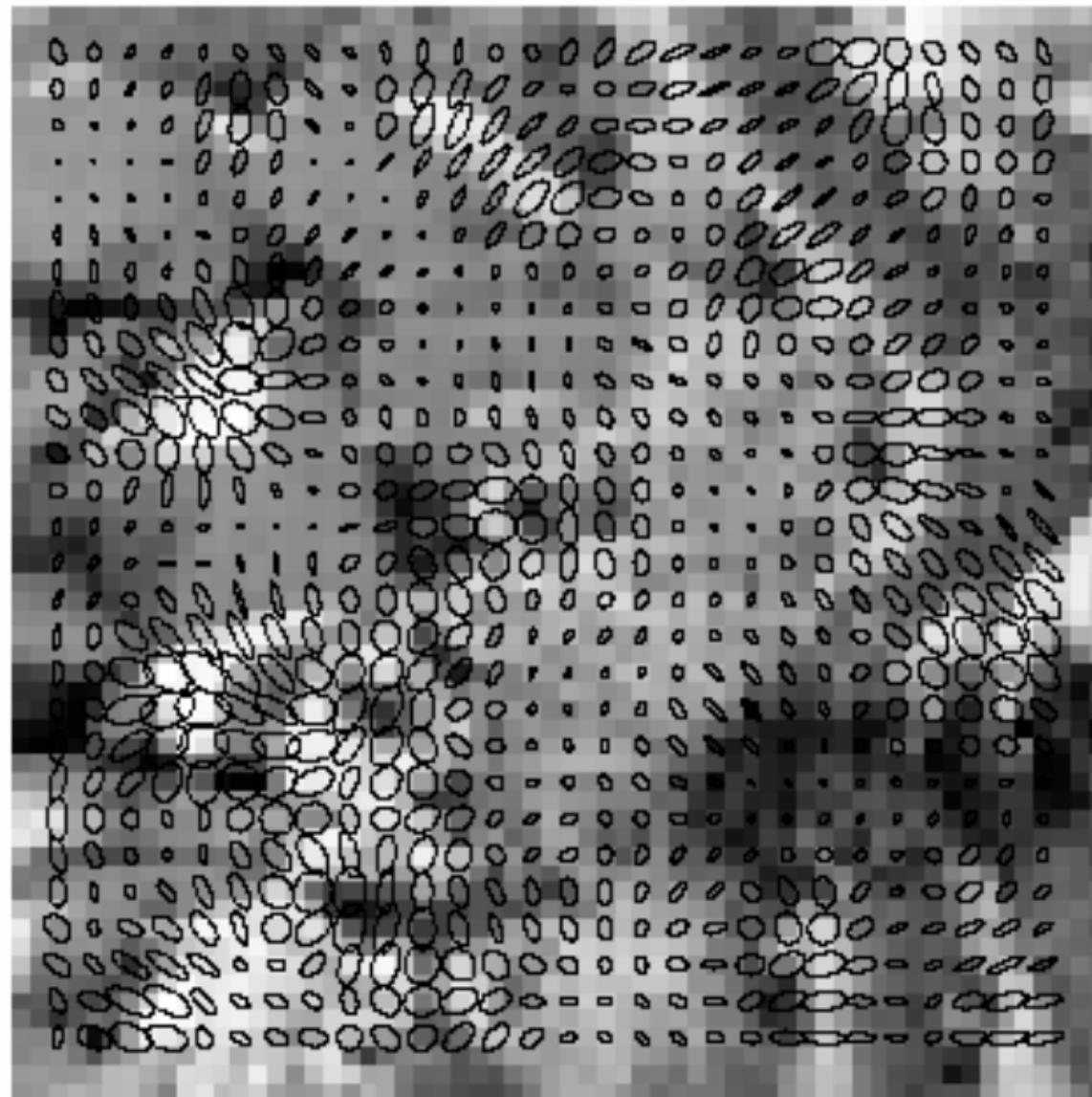
$$\det(M) - 0.04 \text{ } \text{trace}^2(M) =$$

$$M = \begin{bmatrix} g^*I_x^2 & g^*(I_x I_y) \\ g^*(I_x I_y) & g^*I_y^2 \end{bmatrix}$$



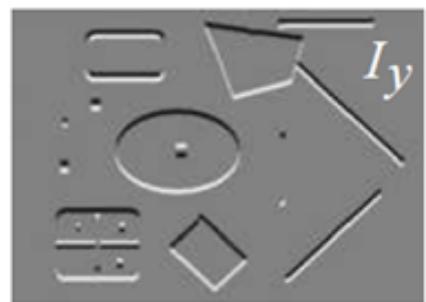
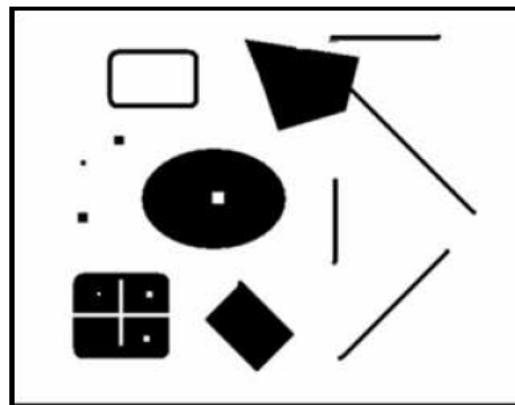
Détection de points

Le détecteur de Harris



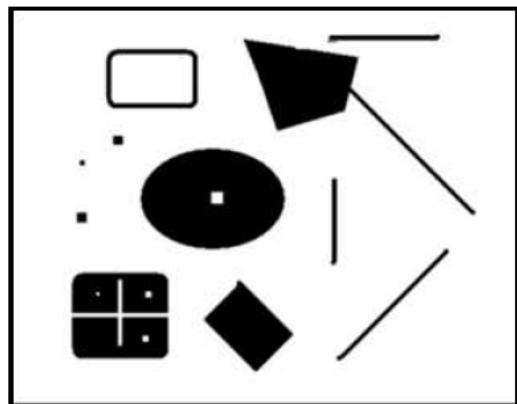
La description locale

Le détecteur Hessian

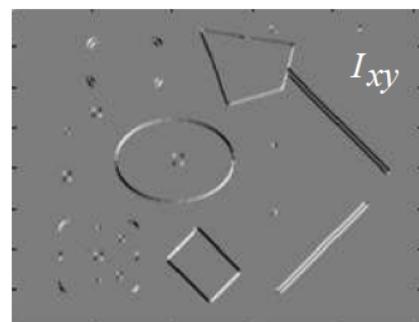


La description locale

Le détecteur Hessien

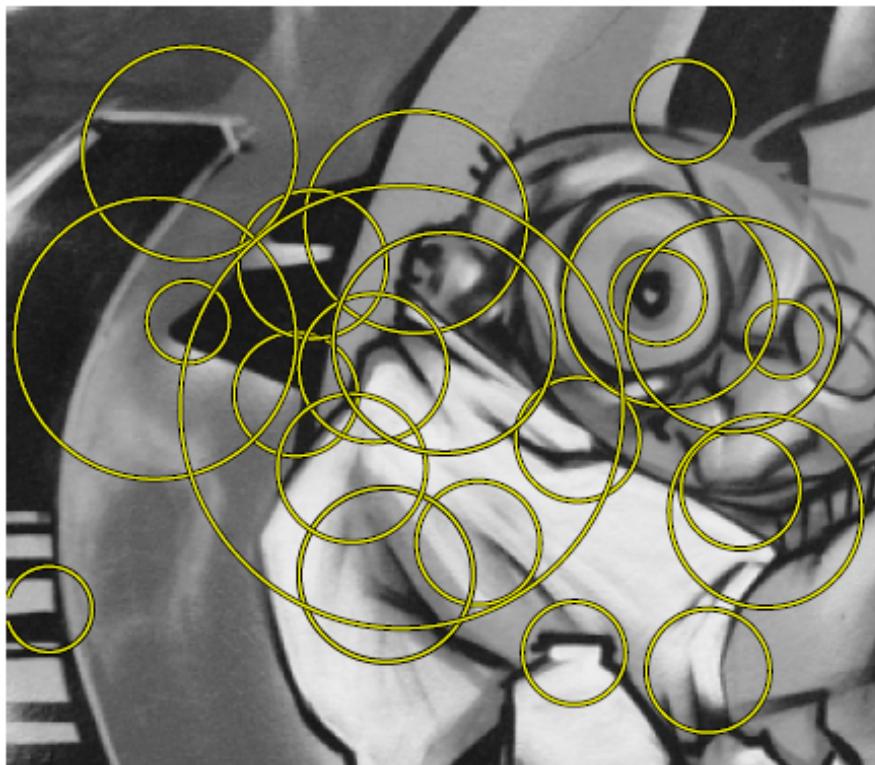


$$I_{xx} \ I_{yy} - I_{xy}^2 =$$



Détection de points

Les détecteurs : problème de l'échelle



Détection de points

La différence de gaussiennes

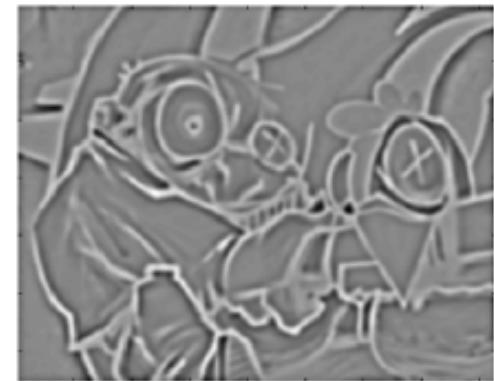
$$I(k\sigma)$$



$$I(\sigma)$$



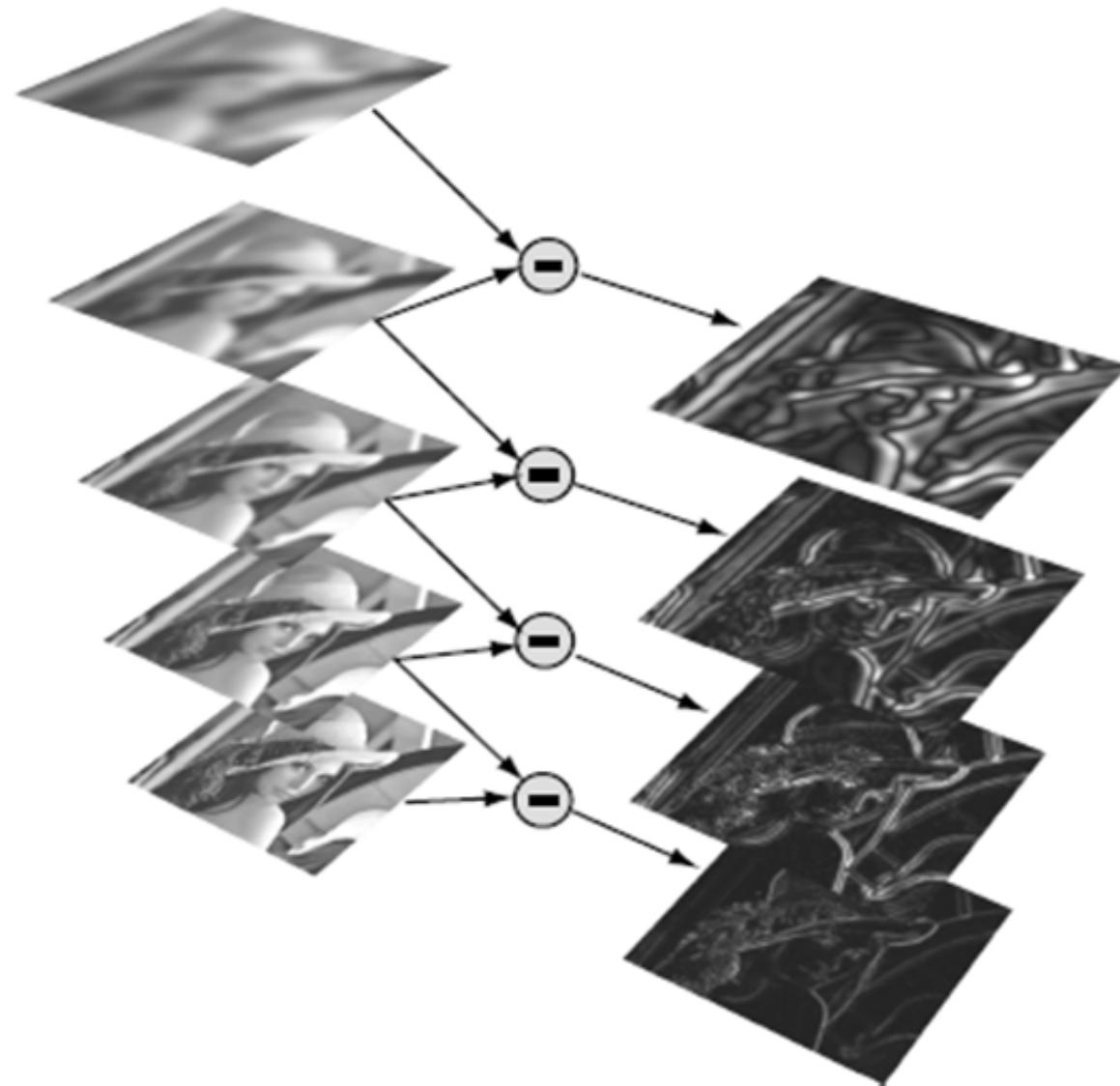
$$I(k\sigma) - I(\sigma)$$



-

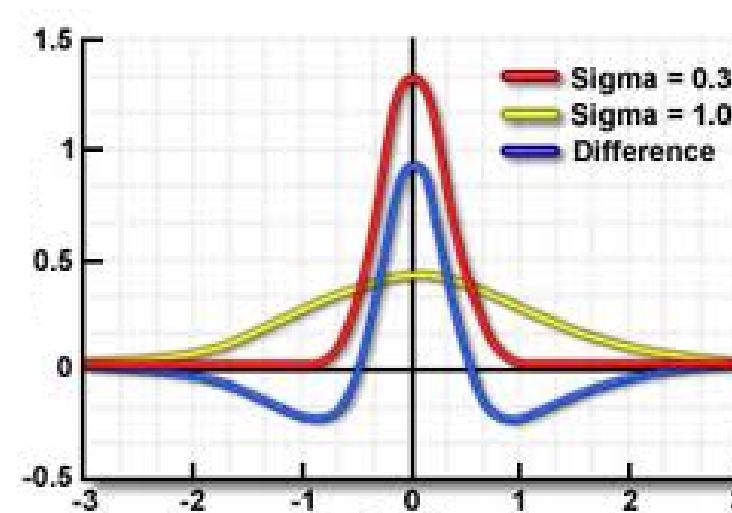
=

La différence de gaussiennes



Détection de points

La différence de gaussiennes : proche du laplacien



Exemples de Laplacien (dérivées secondes) :

0	-1	0
-1	4	-1
0	-1	0

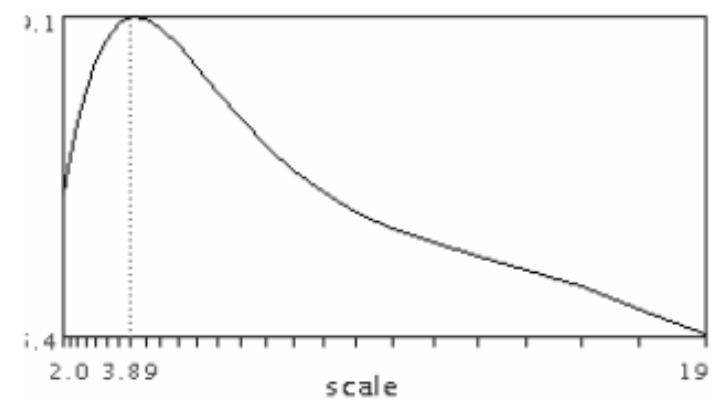
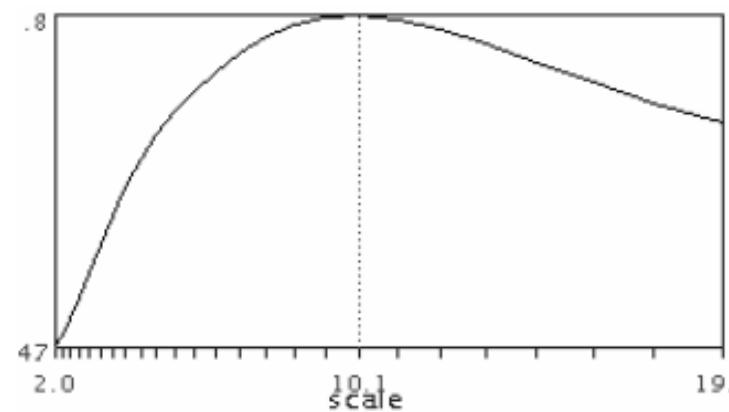
-1	-1	-1
-1	-8	-1
-1	-1	-1

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

Le laplacien



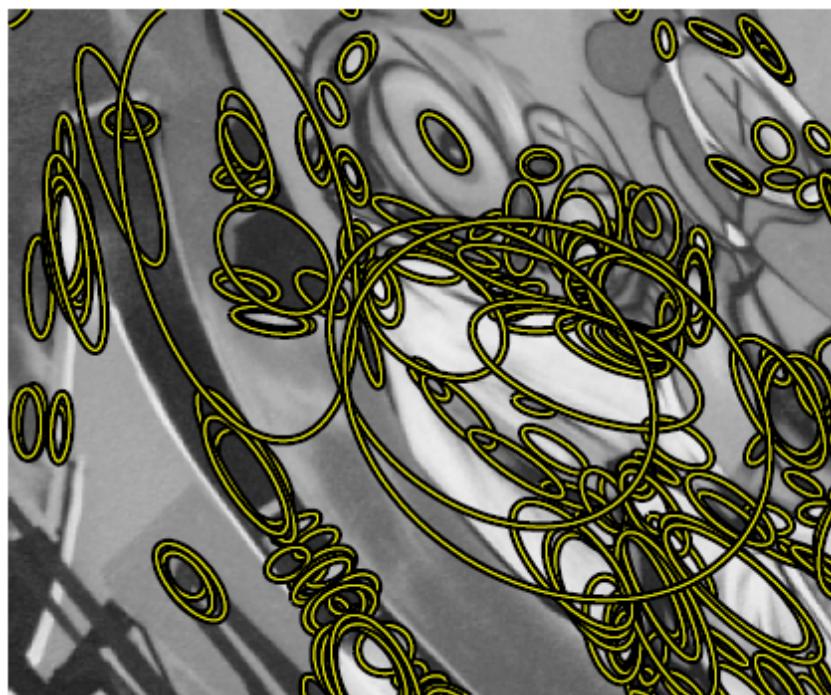
Trace(H)=
Laplacien



→ Harris Laplace, Hessien Laplace.

Les transformations affines

→ Utilisation de la matrice des moments d'ordre 2 (Harris affine, Hessien affine)



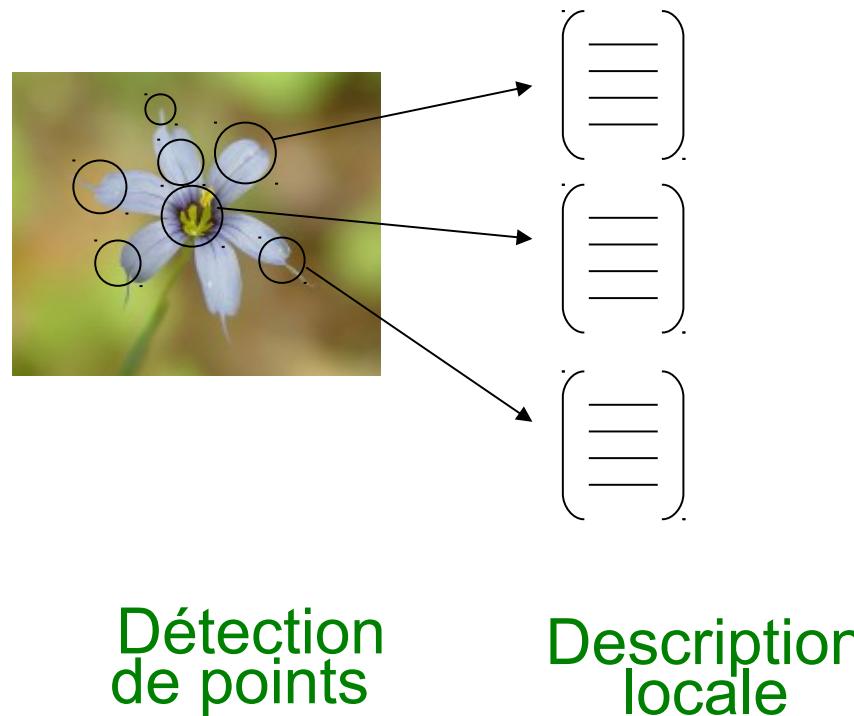
Exemples de points



302

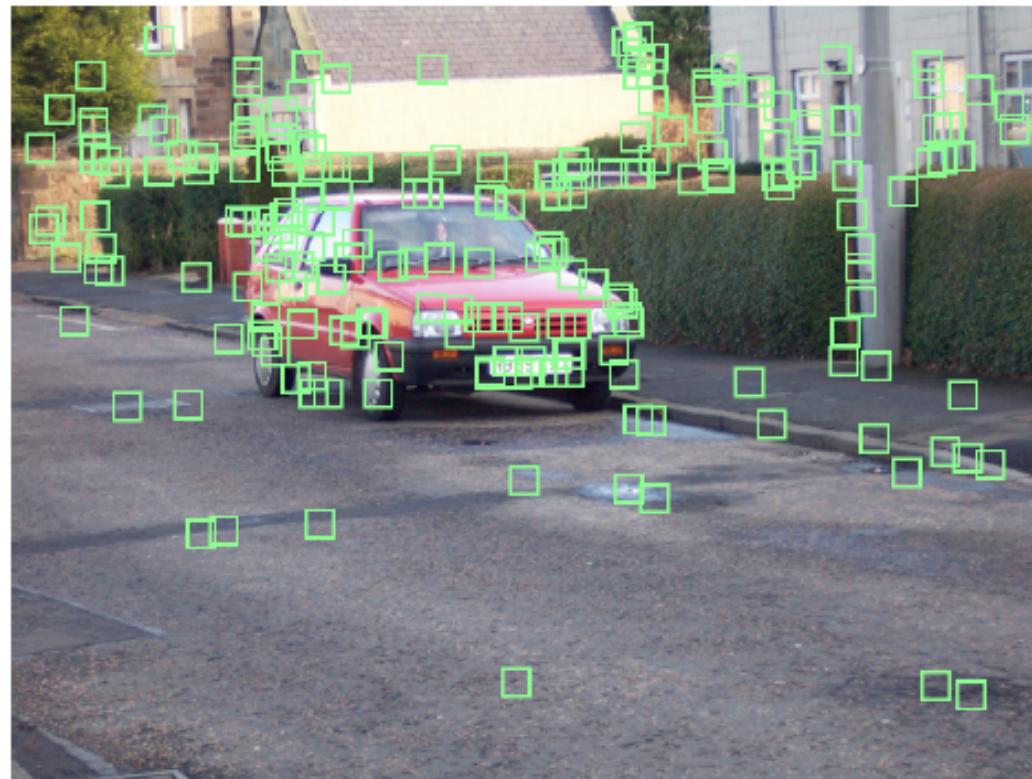
Source : Frolova & Simakov, Invariant Feature Detectors and Descriptors, Weizmann Ins. (Israel).

Description locale



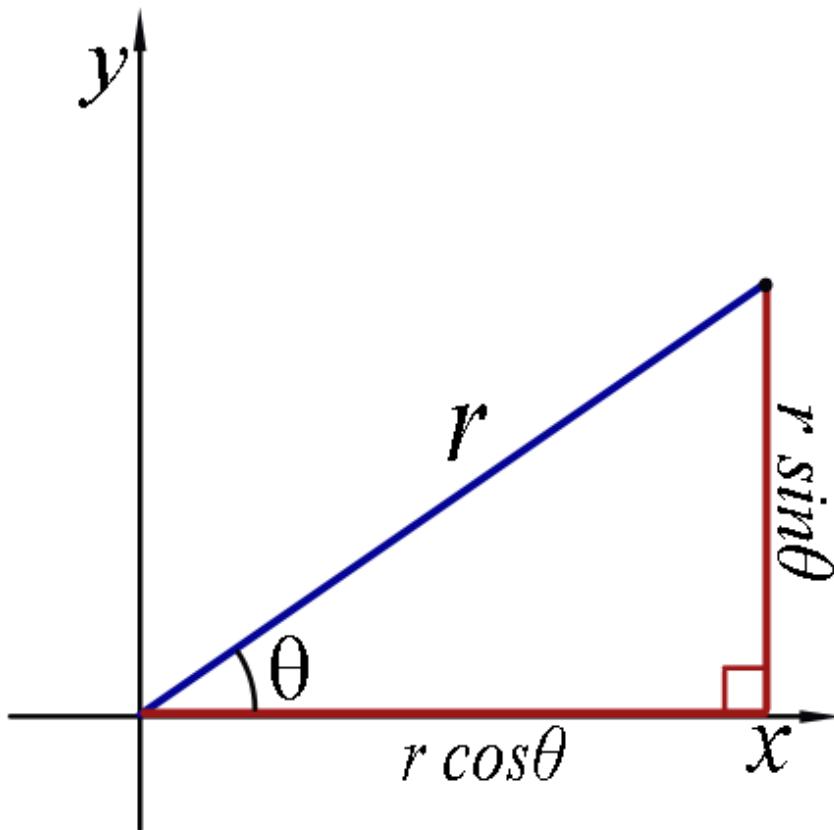
La description locale

Les descripteurs locaux



Le descripteur SIFT : histogrammes d'orientation de gradients

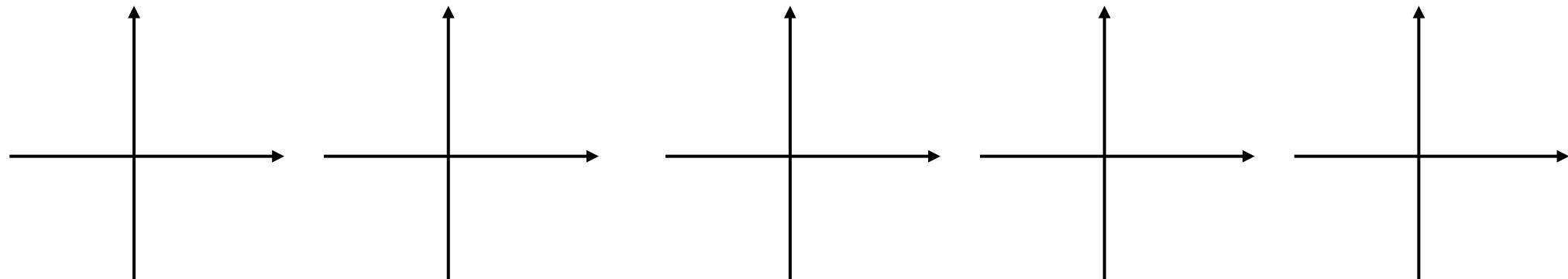
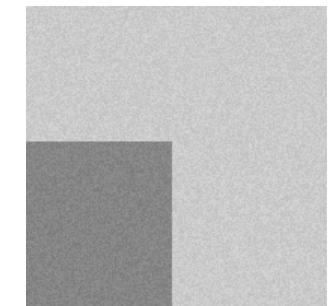
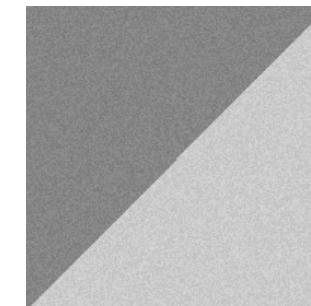
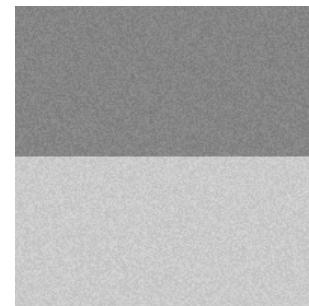
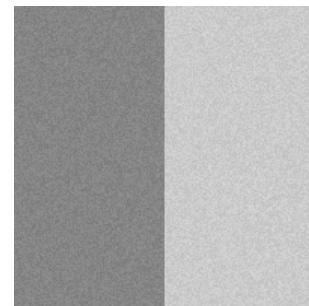
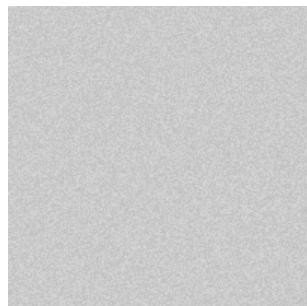
Les coordonnées polaires et cartésiennes



$$r = \sqrt{x^2 + y^2}$$

$$\theta = \arctan\left(\frac{y}{x}\right)$$

Le descripteur SIFT : histogrammes d'orientation de gradients

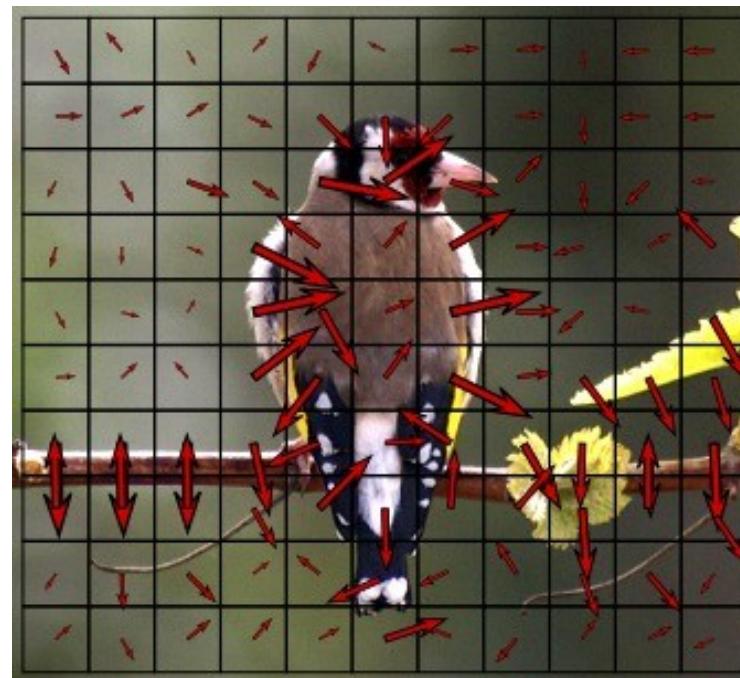


La description locale

Le descripteur SIFT : histogrammes d'orientation de gradients

Le gradient :

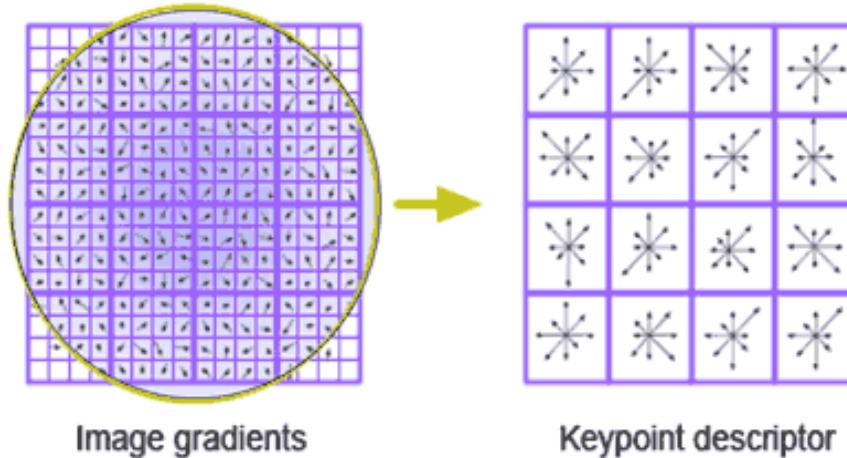
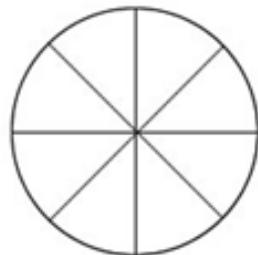
$$\vec{\nabla} f \begin{cases} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{cases}$$
$$|\vec{\nabla} f| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$
$$\theta = \text{Arg}(\vec{\nabla} f) = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$



La description locale

Le descripteur SIFT : histogrammes d'orientation de gradients

8 orientations :



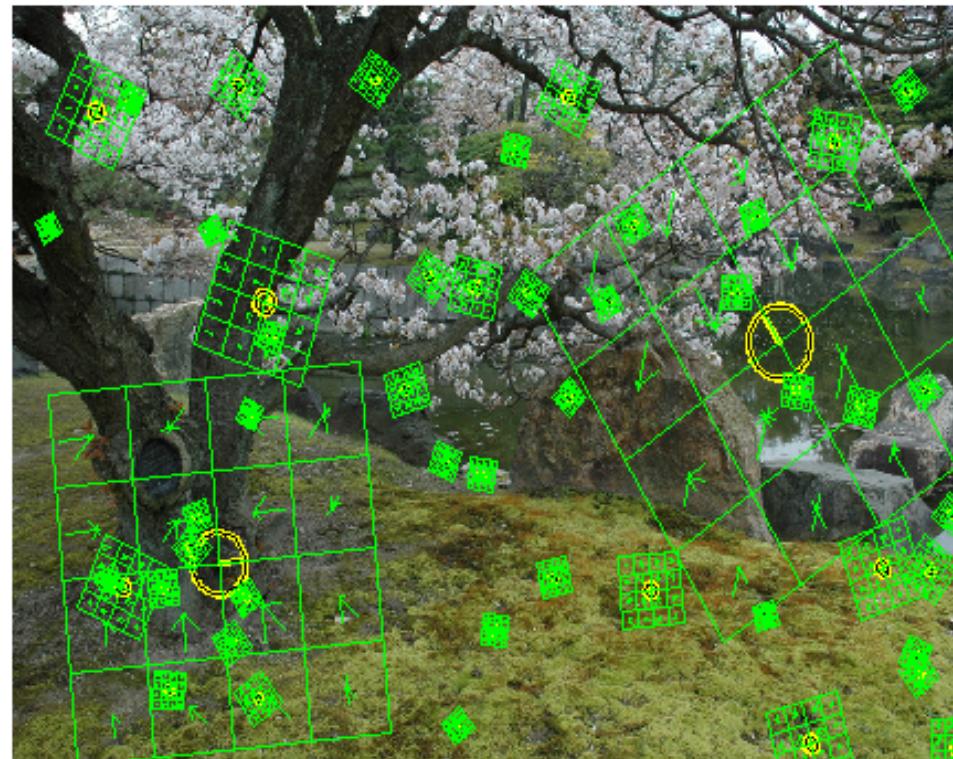
$$4 \times 4 \times 8 = 128\text{-D}$$

NB : SIFT couleur

La description locale

SIFT :

- détection par différences de gaussiennes
- description par histogrammes d'orientation de gradients



➔ Invariance à la rotation

Trois problèmes étudiés

1. Détection de points caractéristiques

- DéTECTeur de Harris
- Différence de gaussiennes
- Harris-Laplace

2. Description locale

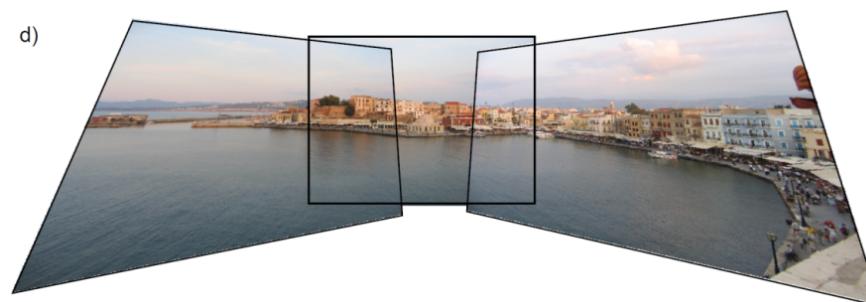
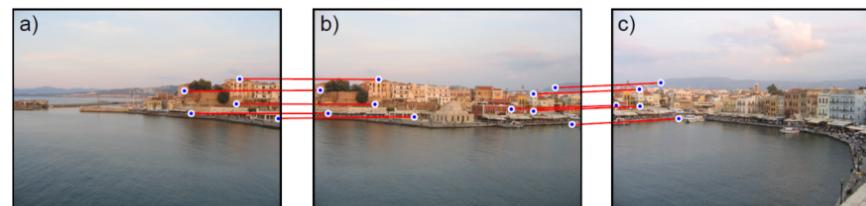
- Descripteur SIFT

3. Mise en correspondance

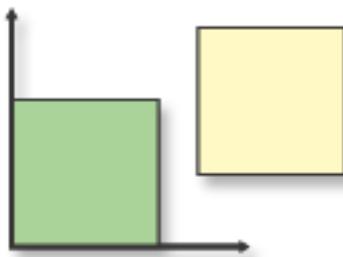
- Matrice et géométrie
- Mise en correspondance robuste

Mise en correspondance

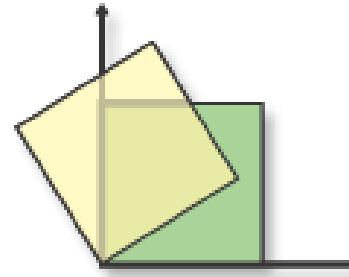
- Déetecter les points (déTECTeur)
- Apparier entre les images (descripteur)
- Estimer la transformation géométrique



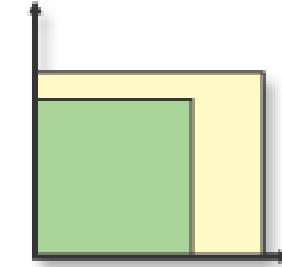
La translation :



La rotation :



Le changement d'échelle :



Transfor-
mations
2D

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

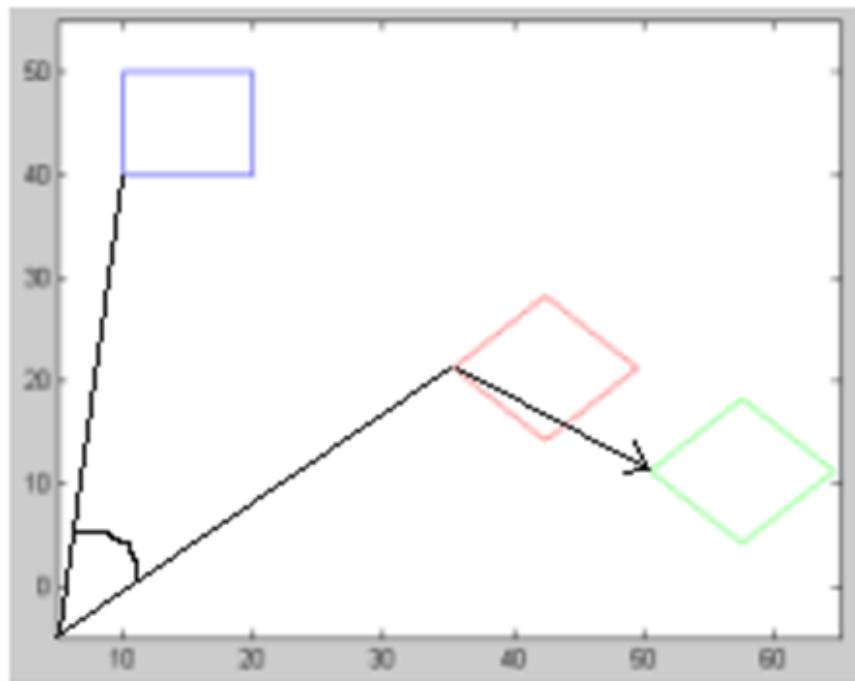
Transfor-
mations
inverses

$$\begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix}$$

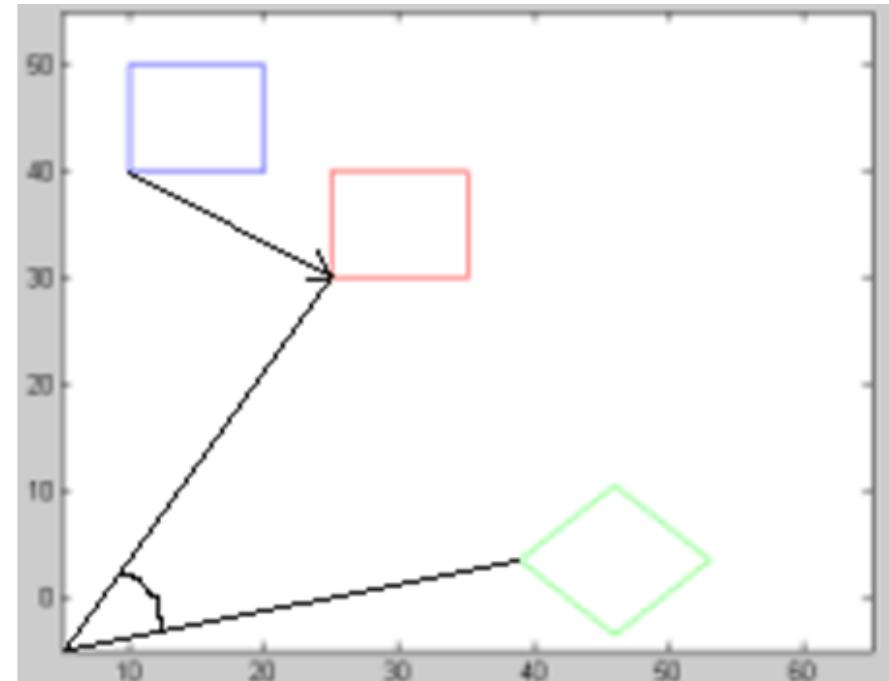
$$\begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Combinaison :



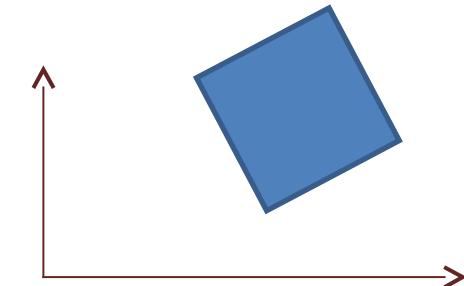
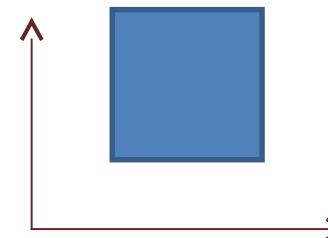
Rotation puis translation



Translation puis rotation

Transformation euclidienne :

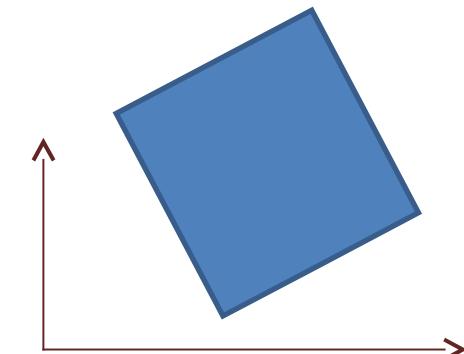
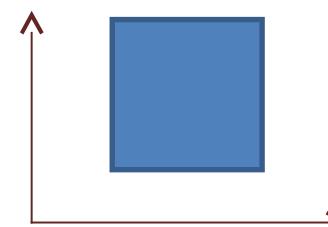
$$\begin{pmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$



→ Conservation des distances, des angles, du parallélisme

Similarité :

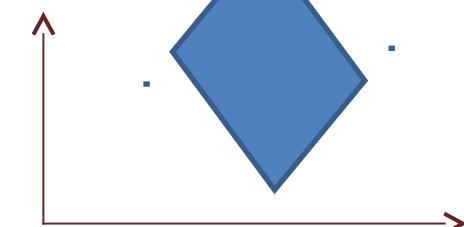
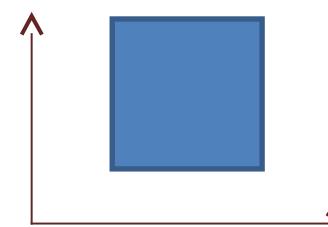
$$\begin{pmatrix} sR & t \\ 0 & 1 \end{pmatrix}$$



→ Conservation des angles, du parallélisme

Transformation affine :

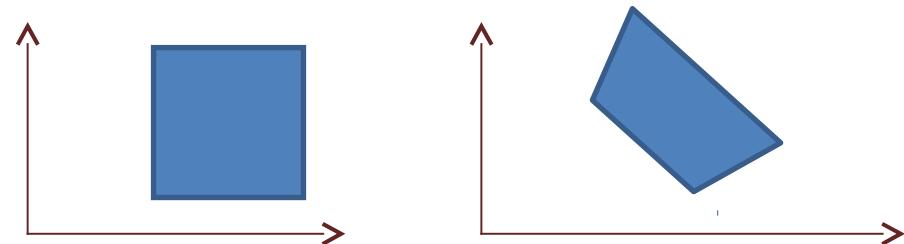
$$\begin{pmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$



→ Conservation du parallélisme

Transformation projective ou homographie :

$$\begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$



→ Conservation des alignements

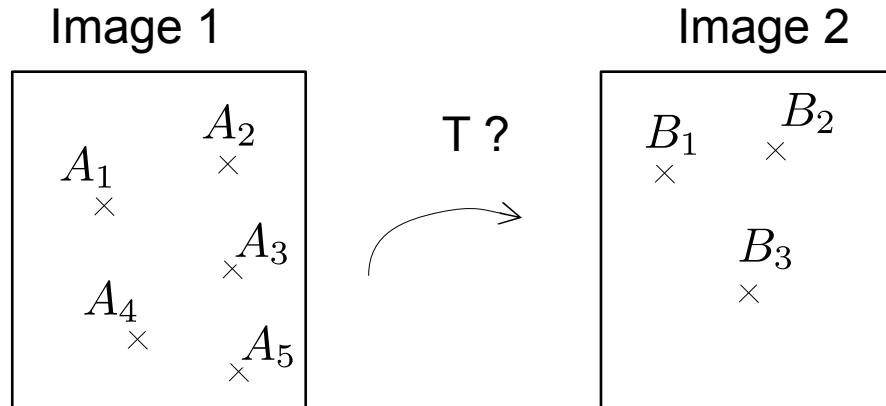
Transformation projective ou homographie :



Lors d'un changement de point de vue, on a en général une **homographie**

S'il y a peu d'inclinaison (ou profondeur de champs faible),
on peut utiliser une **transformation affine**

Estimation d'une transformation entre 2 images (1/2)



1. Recherche des points ayant le descripteur le plus proche :

$$\begin{aligned} B_1 &\longrightarrow A_2 \\ B_2 &\longrightarrow A_5 \\ B_3 &\longrightarrow A_4 \end{aligned}$$

2. Recherche de la matrice vérifiant :

$$X_2 \approx TX_1$$

avec :

$$X_1 = \begin{pmatrix} x_{A_2} & x_{A_5} & x_{A_4} \\ y_{A_2} & y_{A_5} & y_{A_4} \\ 1 & 1 & 1 \end{pmatrix}$$

$$X_2 = \begin{pmatrix} x_{B_1} & x_{B_2} & x_{B_3} \\ y_{B_1} & y_{B_2} & y_{B_3} \\ 1 & 1 & 1 \end{pmatrix}$$

Estimation d'une transformation entre 2 images (1/2)

- Transformation affine

Matrice recherchée :

$$T = \begin{pmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

Solution minimisant l'écart (moindres carrés) :

$$X_2 \approx TX_1$$

$$T = X_2 X_1^+ \leftarrow$$

si $A = USV'$ (svd)
 $A^+ = VS^+U'$

Pseudo-inverse

- Homographie

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$

$$HX_1 = \alpha X_2$$

l'homographie est définie à une constante multiplicative près

Écriture sous la forme :

$$A h = 0$$

Solution telle que $h \neq 0$

Décomposition en SVD : $A = USV'$

$$h = \alpha V(:, \text{end})$$

Estimation robuste

Estimation robuste : RANSAC (RANdom SAMpling by Consensus)

- 1 → sélection aléatoire d'un sous ensemble minimal de correspondances
- 2 → estimation de la transformation sur ce sous-ensemble
- 3 → évaluation du nombre de correspondances validant cette transformation
- 4 → répétition des étapes 1 à 3 un certain nombre de fois
- 5 → sélection de la meilleure transformation T_{opt}
- 6 → ré-estimation de la transformation en considérant toutes les correspondances qui validaient T_{opt}

Calcul de l'image transformée

$$\begin{pmatrix} 1.3 & 0.5 & 0 \\ 0.2 & 3.0 & 30 \\ 0 & 0 & 1 \end{pmatrix}$$

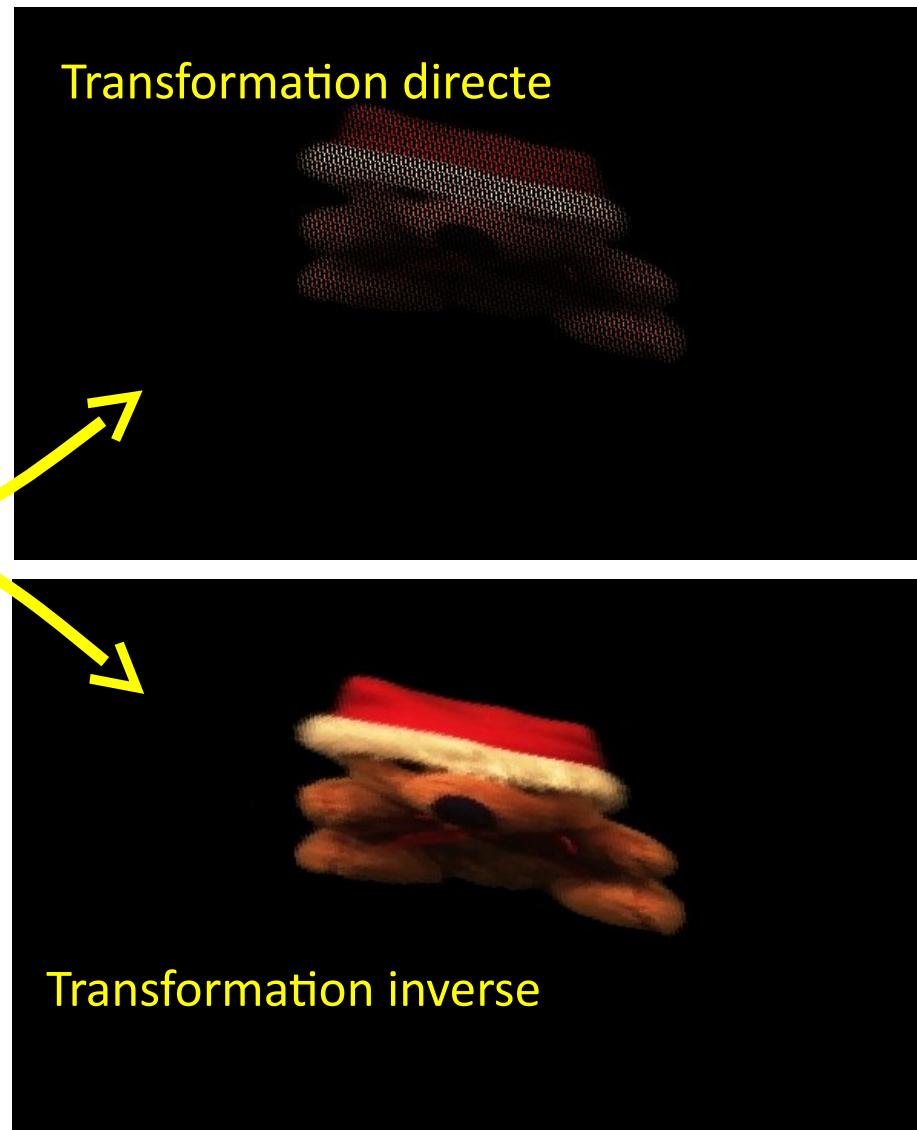


Image mosaïque finale

