CHAPITRE :
LES MÉTHODES DE SEGMENTATION D'IMAGES

Mr. HALLACI S.

Département d'informatique
Université Guelma

Segmentation

introduction

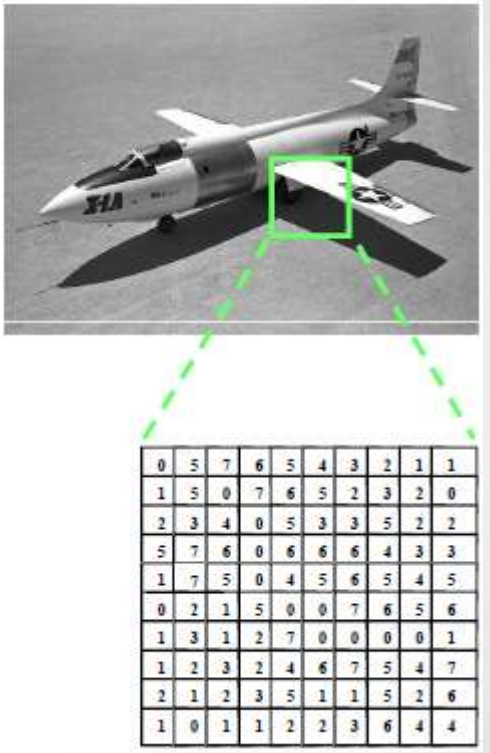
La machine ne peut pas, comme l'humain, séparer l'objet du fond.

Construire des systèmes qui peuvent séparer les objets dans une image acquise

?

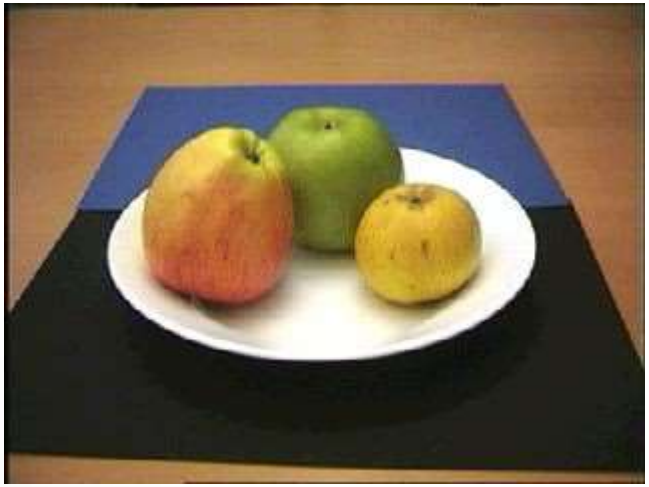
C'est une:

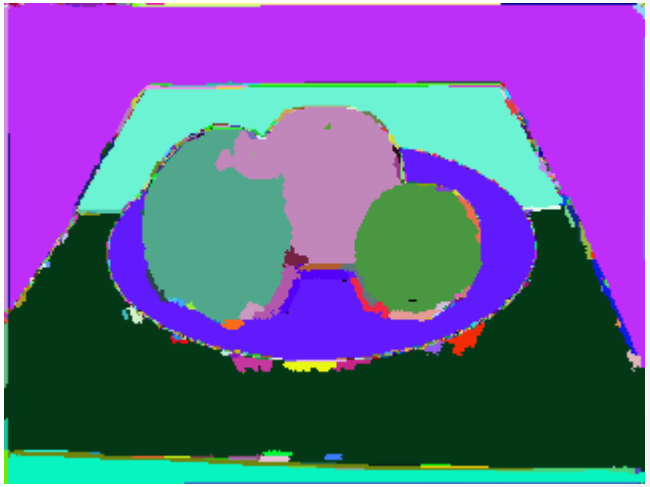
« SEGMENTATION »



Qu'est-ce que la segmentation ?

- La segmentation vise à diviser l'image en morceaux
  - Ces morceaux correspondent aux objets dans l'image
- La segmentation est liée à la reconnaissance
  - Quels objets voit-on dans l'image ?
  - Le paradoxe *reconnaissance/segmentation*





Qu'est-ce que la segmentation ?

- La segmentation est basée sur:
  - les discontinuités
    - les arrêtes, les changements abruptes, …
  - les similitudes (zones homogènes)
    - couleurs, textures, intensités, …
- La segmentation est le découpage d'une image en différentes ***régions connexes*** et/ou ***contours***.
- L'intérêt de ces régions est de pouvoir être manipulées ensuite via des traitements de haut niveau pour extraire des caractéristiques de forme, de position, de taille, etc.
- A chaque point d'un même groupe, on assigne la même valeur, qu'on appellera ***label*** ou ***étiquette***.

Choix de la segmentation ?

- Le problème est évidemment très mal posé, car on ne sait jamais dire quelle est la segmentation idéale.
- L'idée est bien sûr que la région se rapproche de la notion d'objet, au sens courant du terme. Néanmoins, on peut dégager des propriétés plus raisonnables qu'on cherche à obtenir dans un algorithme de segmentation, en particulier :
  - ➢ ***Stabilité*** : la segmentation obtenue ne doit pas varier beaucoup lorsque les conditions d'acquisition varie légèrement (bruit, illumination, point de vue,...)
  - ➢ ***Régularité*** : les régions obtenues doivent être simples à manipuler (taille suffisante, forme régulière,...)

DIFFÉRENTES SÉMANTIQUES

- La texture

– Zone homogène

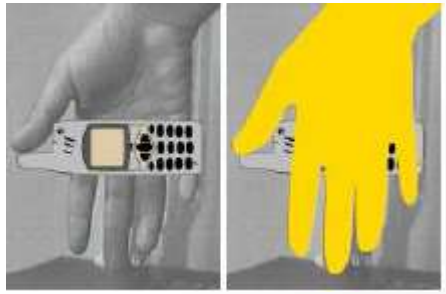
- L'intensité

– Distinction par rapport au fond

- Forme a priori

– Connaissance d'un modèle

– Pas de problème d'occultation



LES DIFFICULTÉS

Ambigüe

LES DIFFICULTÉS

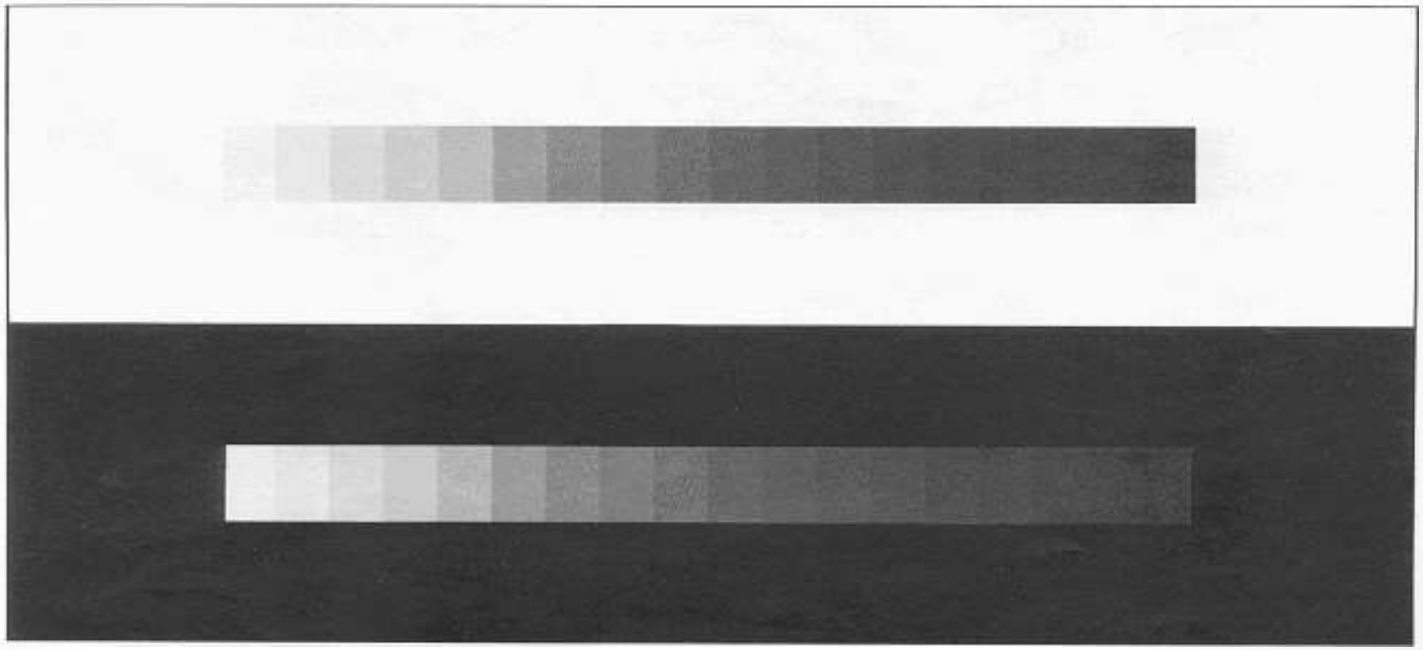
dégrader

LES DIFFICULTÉS

Complexe / beaucoup de détailles

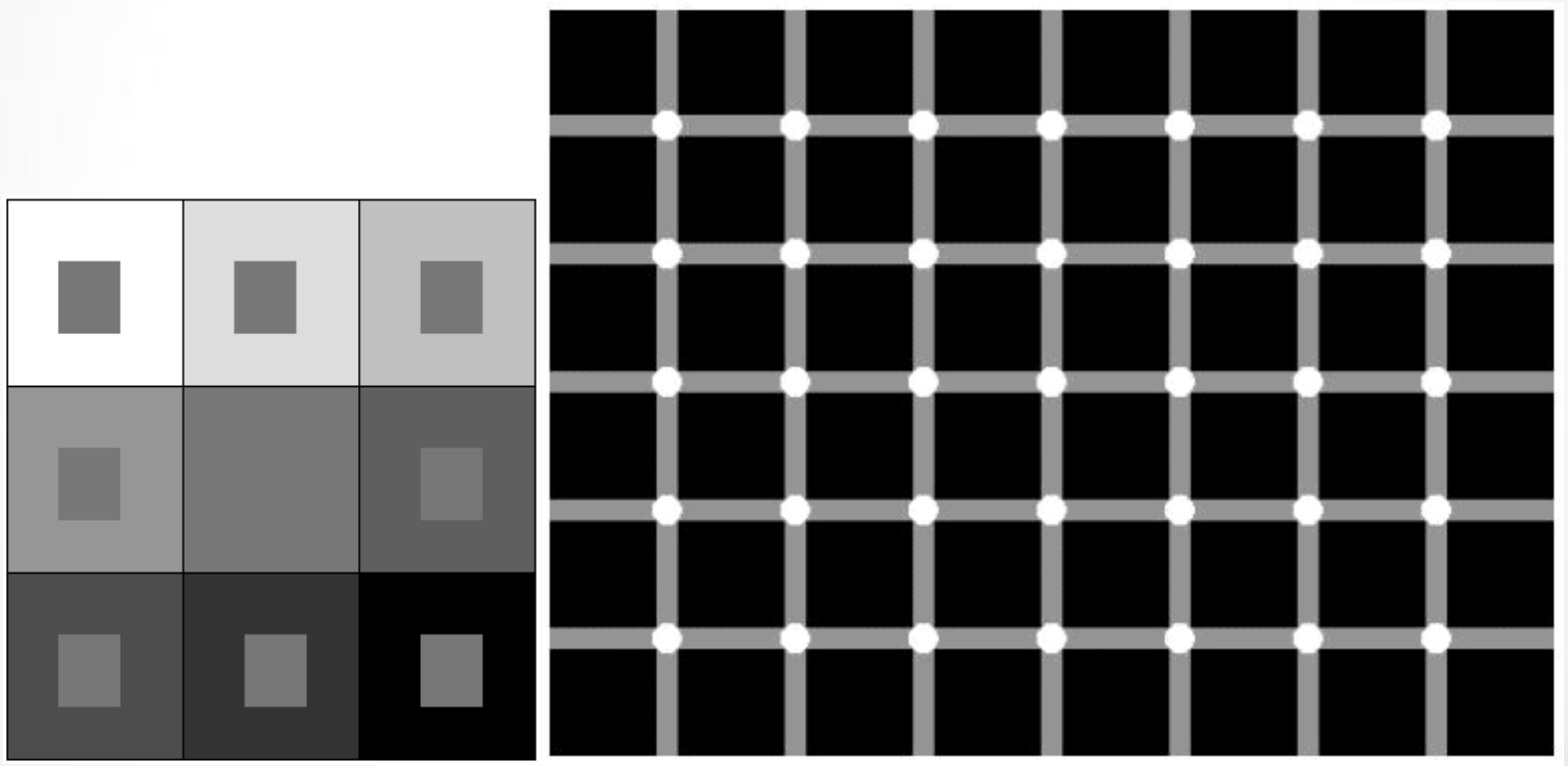
LES DIFFICULTÉS

Perception du contraste en fonction du fond



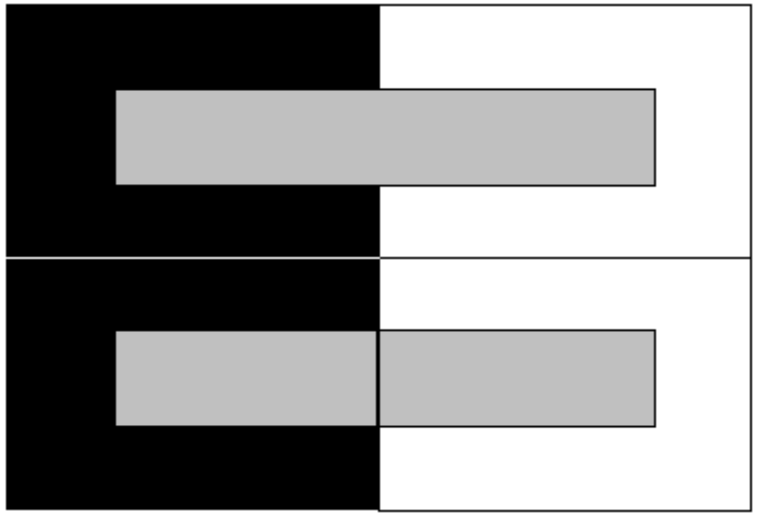
LES DIFFICULTÉS

Phénomènes dus aux interactions spatiales

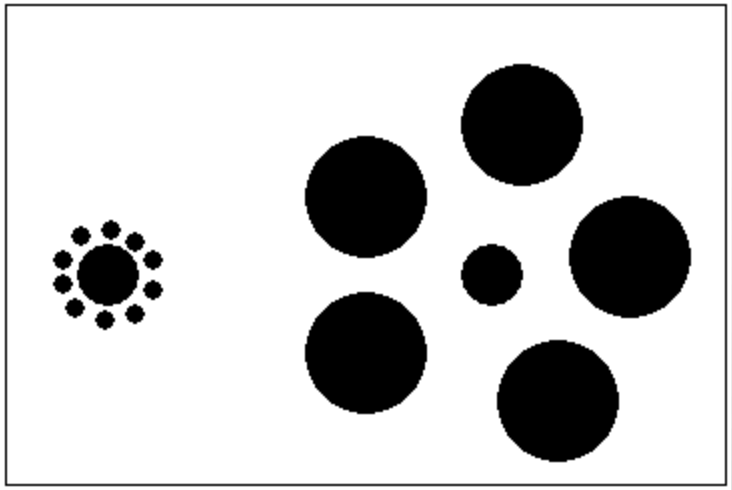


LES DIFFICULTÉS

Vrais/faux contours



proportions



Approches de segmentation

- Espace des couleurs
  - Segmentation par seuillage
  - Segmentation par classification
- Espace image (deux approches)
  - Approche par les régions
    - Segmentation par croissance de régions
    - Méthode Division-fusion (« split and merge »)
  - Approche par les contours
    - Segmentation par détection, fermeture de contours, et contours actifs.
    - Ligne de partage des eaux

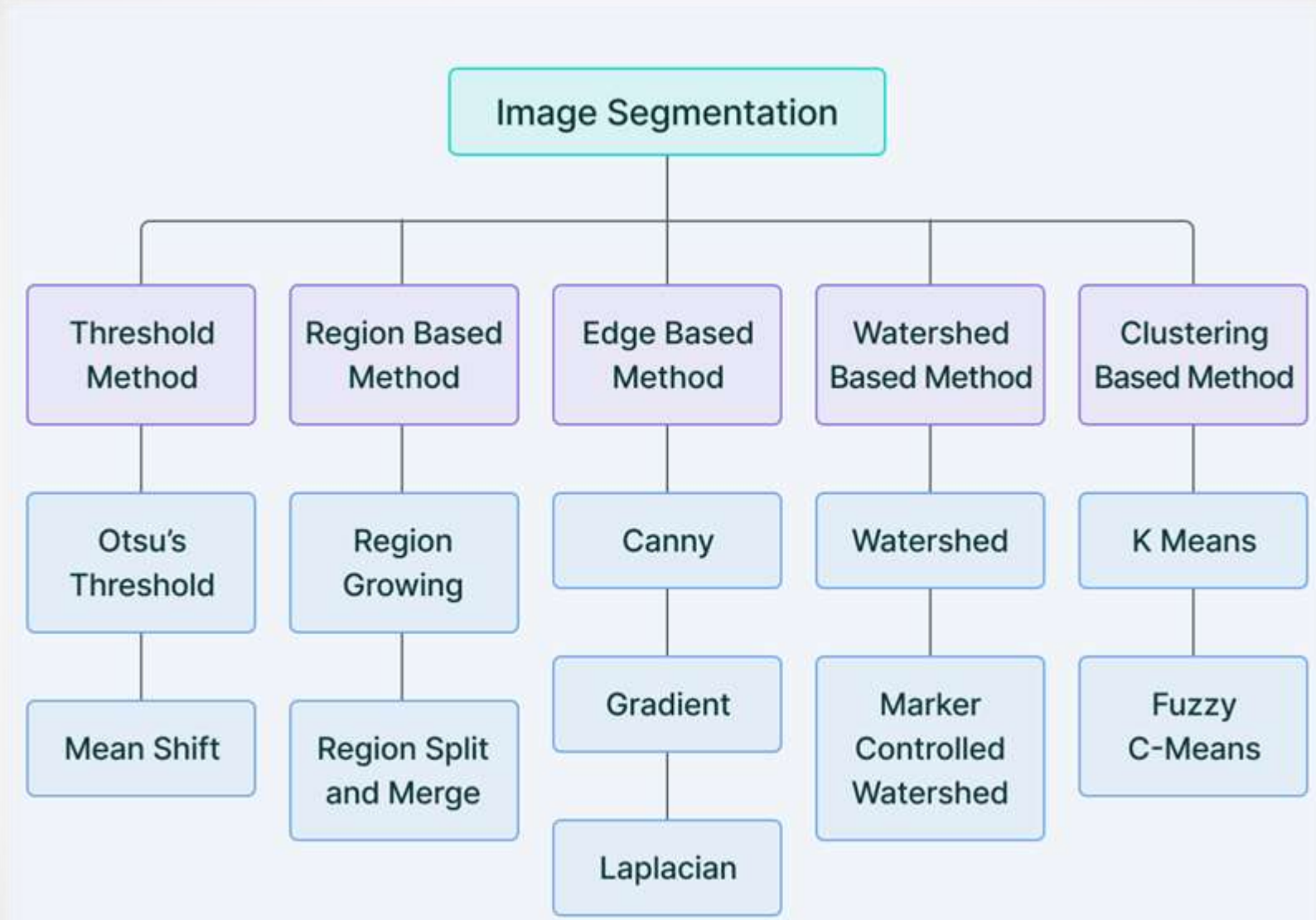


Image Segmentation
Threshold Method
Region Based Method
Edge Based Method
Watershed Based Method
Clustering Based Method
Otsu's Threshold
Region Growing
Canny
Watershed
K Means
Mean Shift
Region Split and Merge
Gradient
Marker Controlled Watershed
Fuzzy C-Means
Laplacian

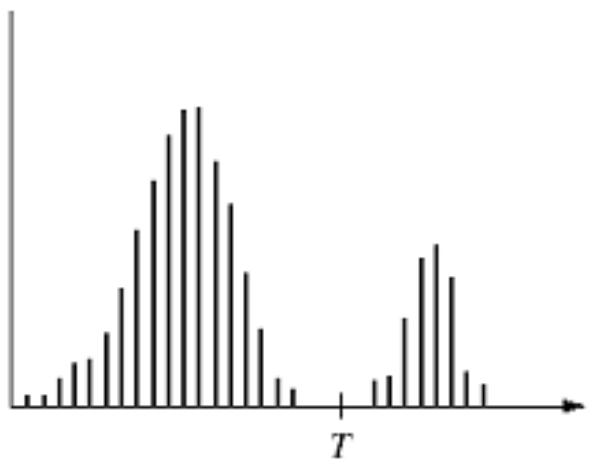
Segmentation par seuillage

- Le seuillage est une méthode simple et très populaire pour la segmentation d'objets dans les images numériques.
- Le seuillage peut être de nature
  - *Globale* : un seuil pour toute l'image
  - *Locale* : un seuil pour une portion de l'image
  - *Adaptative* : un seuil qui s'ajuste selon les images/parties de l'image.

Seuillage

- Seuillage global de base
  - Comment trouver le seuil S ?
    - Une valeur obtenue par tests
    - La valeur moyenne des tons de gris
    - La valeur médiane entre le ton maximum et le ton minimum
    - Une valeur qui balance les deux sections de l'histogramme

      - seuillage automatique



Segmentation par seuillage -global

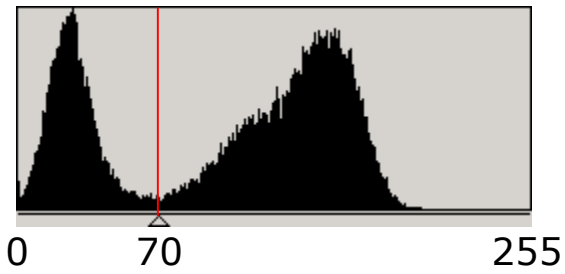
- Une image A(x,y) avec 256 niveaux de gris.
- Son histogramme est H(i) $i \in [0,255]$.
- Détermination d'un seuil S-> *« seuil »* qui permettra d'obtenir une image binaire B(x,y)
- Seuillage de base (2 classes) – principe :
  - *Si valeur(pixel) >= seuil alors valeur(pixel) = 1*
  - *Si valeur(pixel) < seuil alors valeur(pixel) = 0*
- Le résultat du seuillage est une image binaire.
- Il est aussi possible d'avoir ***n*** seuils pour séparer l'image en ***n-1*** classes.
- ***Problème*** : choix du seuil !!!

Segmentation par seuillage

Exemple :

A(x,y)

H(i)



Seuillage à 70

B(x,y)

Question : Comment trouver automatiquement le seuil ?

Choix de seuil

Image originale (256 niveaux de gris)

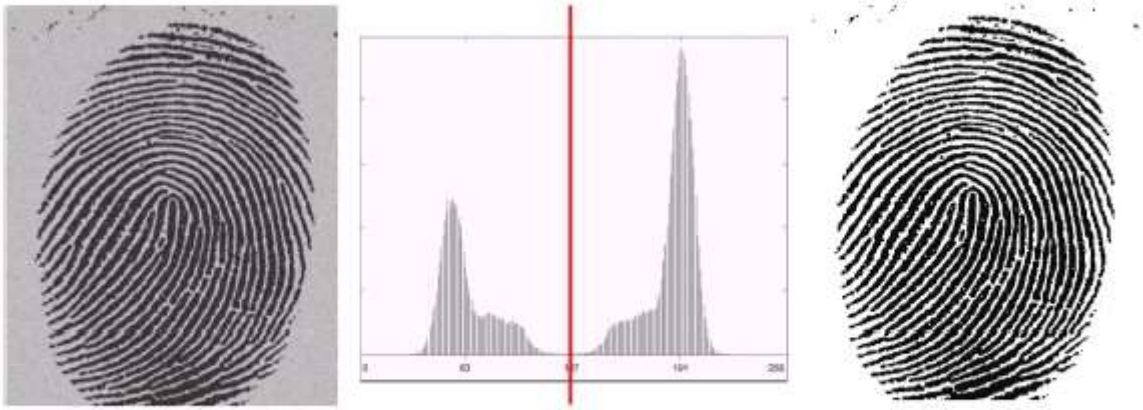
Seuil à 150

Seuil à 70

Seuil à 220

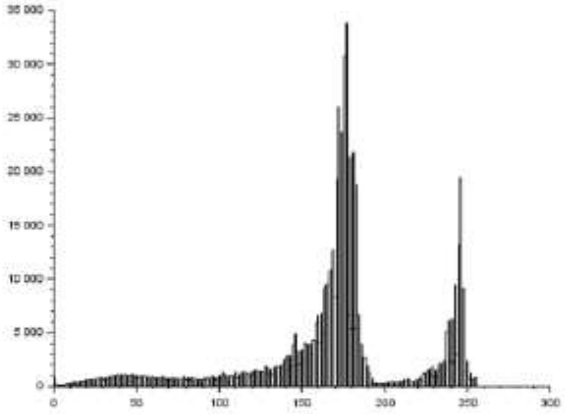
Choix de seuil

→ Dans certains cas, le choix du seuil est facile :

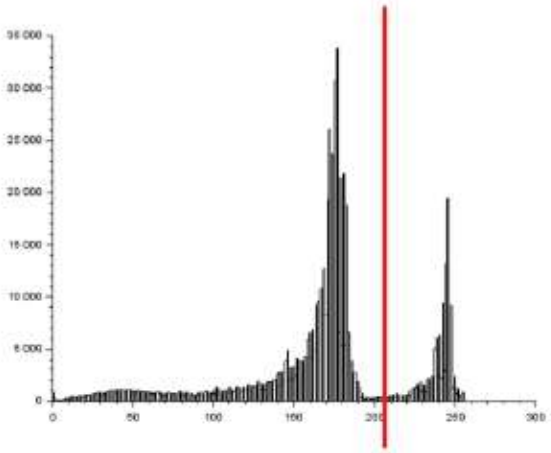


Choix de seuil

→ Dans d'autres cas, le choix du seuil est moins évident :

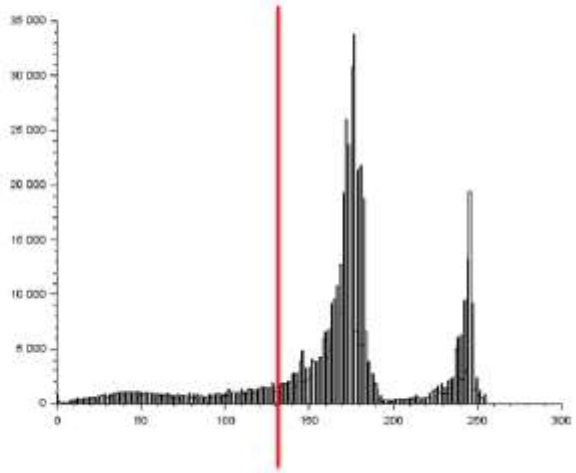


→ Dans d'autres cas, le choix du seuil est moins évident :



Choix de seuil

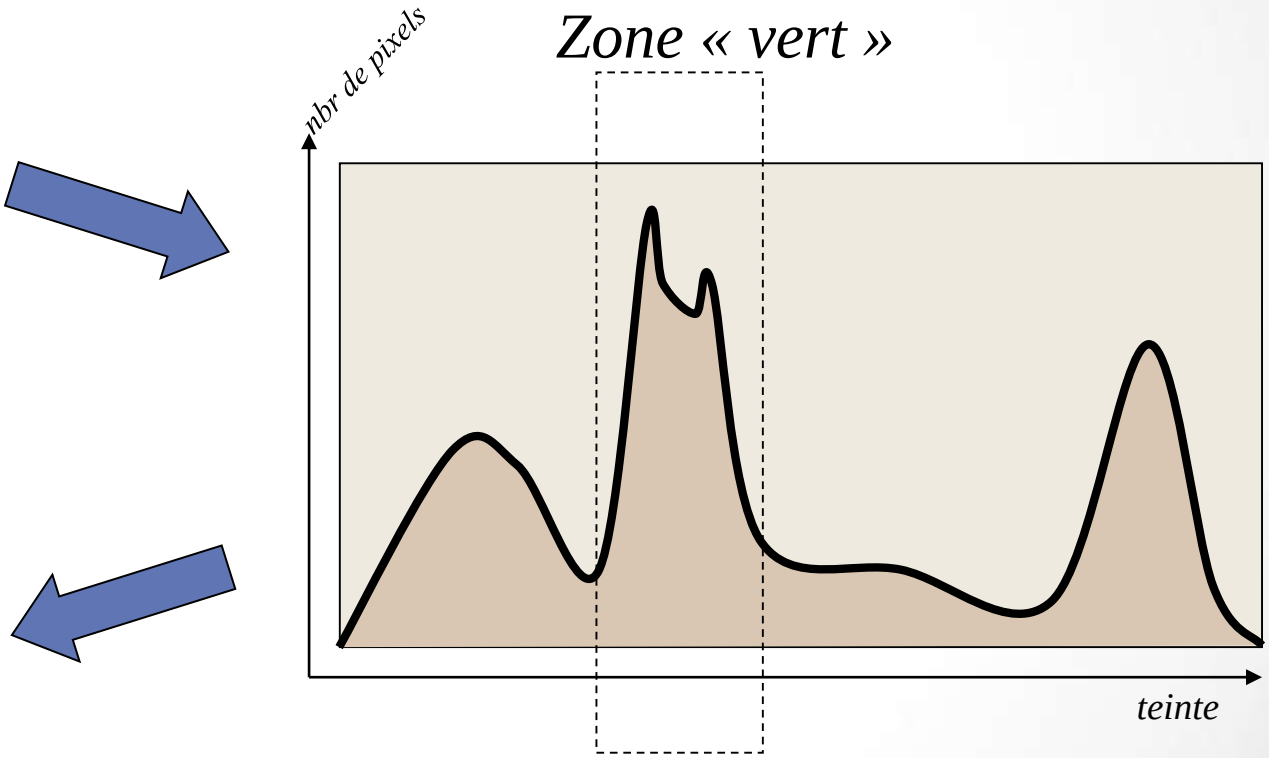
→ Dans d'autres cas, le choix du seuil est moins évident :



Segmentation par seuillage

Histogrammes

Idée: Si les objets présents dans l'image ont des couleurs bien distinctes et uniformes, ils vont apparaître comme des pics dans



=> Segmentation dans un espace dérivé de l'image

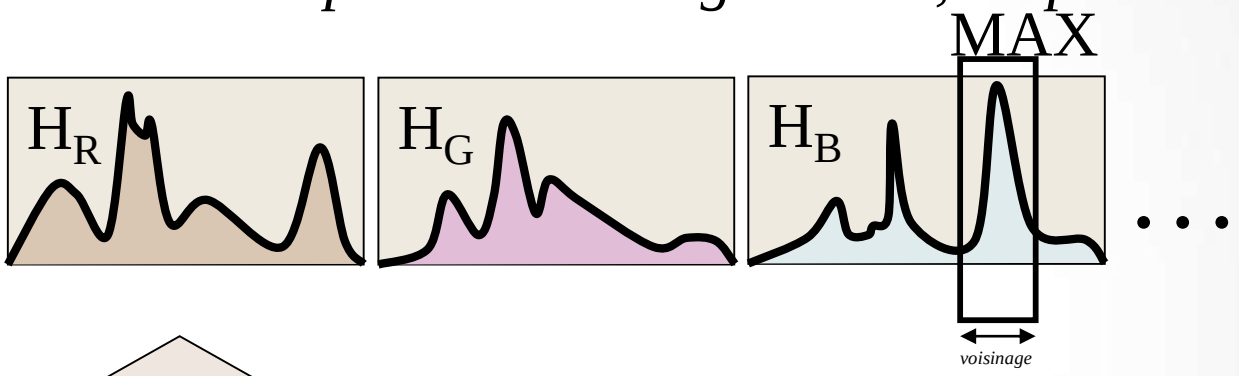
Segmentation par seuillage

Recursive Histogram Splitting (RHS)

Chaque pixel est décrit selon certains channels: R,G,B,H,S,V,…

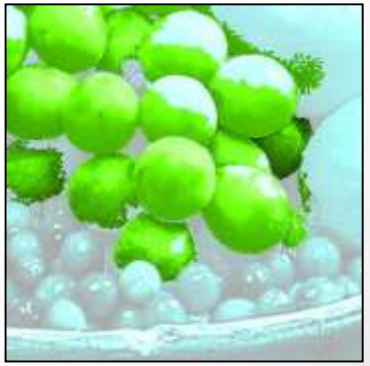
=> *L'algorithme travaille sur plusieurs histogrammes, un par channel*

Image initiale



Réinjection des régions de taille suffisante

Retroprojection de la fenêtre de l'histogramme



Suppression de la région extraite

Segmentation par seuillage

Recursive Histogram Splitting (RHS)

AVANTAGES

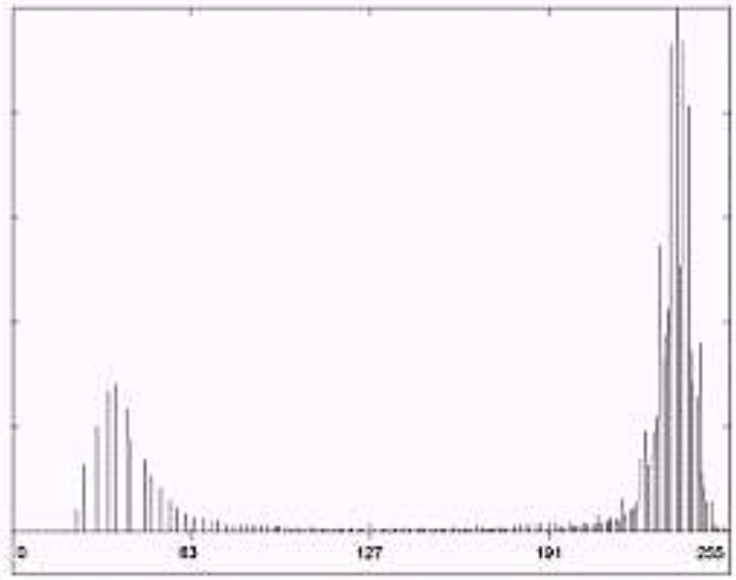
- Méthode très rapide, Universel, temps réel, simplicité
- Fonctionne bien sur des histogrammes multi-modaux
- Peu sensible au bruit

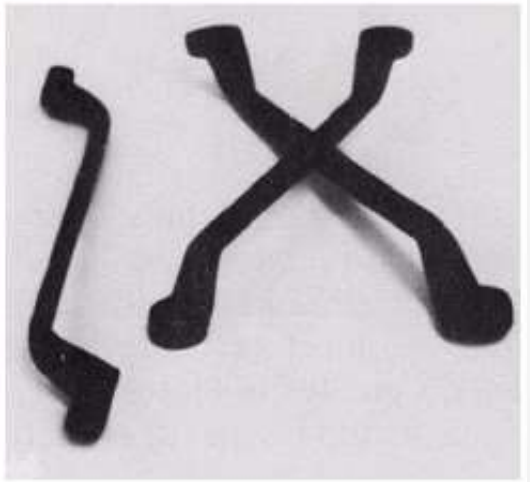
INCONVENIENTS

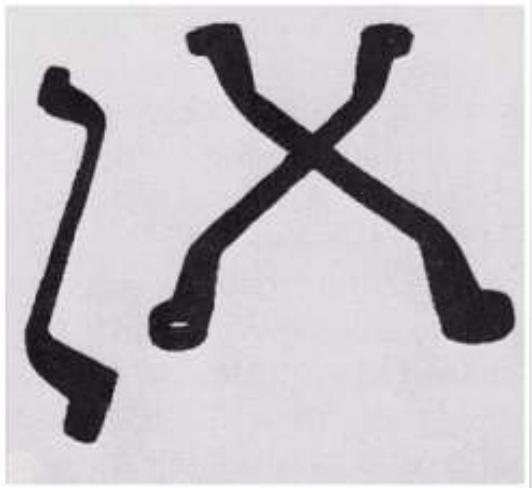
- Méthode globale: ignore les informations de proximité qui permettent d'utiliser des seuils variables locaux.
- Connaître le nombre de classes
- Apparition de faux éléments (aucune prise en compte de la composante spatiale)
- Nombre de modes souvent nombre de classes attendu
- Que se passe-t-il si deux objets ont la même couleur? => Nécessite en général un *Region Growing* pour détacher les composantes connexes.

Exemple de seuillage

- Seuillage global de base
  - Valeur médiane
    - Environnement contrôlé
    - Applications industrielles

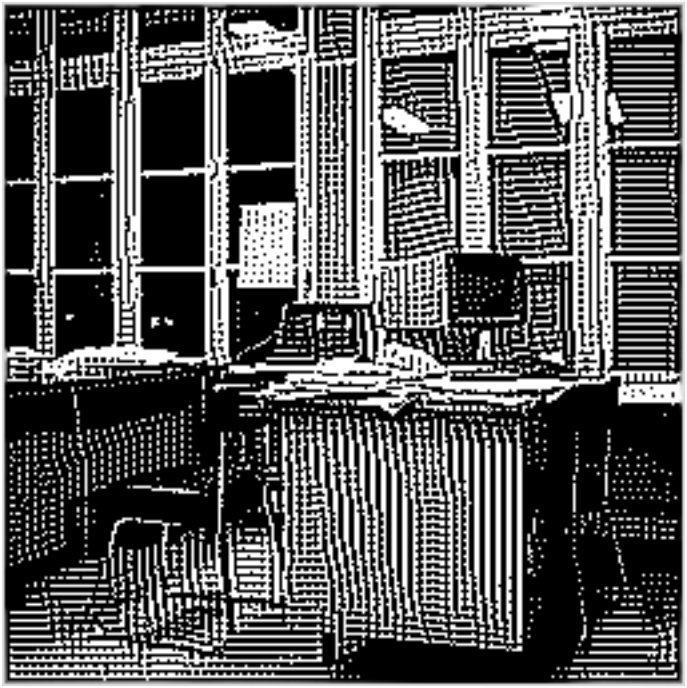






seuillage local

Seuillage local



Pour un pixel, le seuil est la valeur moyenne dans son voisinage 5 x 5

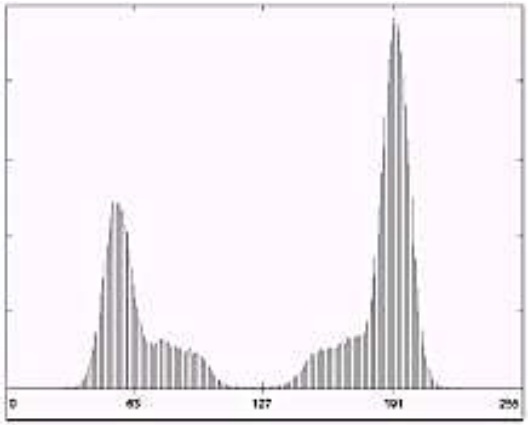
Seuillage global automatique

- Exemple d'algorithme :
  - Choisir un T initial (moyenne, médiane …)
  - On obtient 2 groupes de pixels
    G_1 si $f(x,y) > T$ et G_2 si $f(x,y) \leq T$
  - Calculer les moyennes de tons de gris pour G_1 et G_2 -> μ_1 et μ_2
  - Calculer une nouvelle valeur de T
    $T = 1/2\ (\mu_1 + \mu_2)$
  - Répéter jusqu'à ce que T soit ~ constant

- Il existe plusieurs autres méthodes globales automatiques
  - Otsu, Kittler, K-moyennes, …

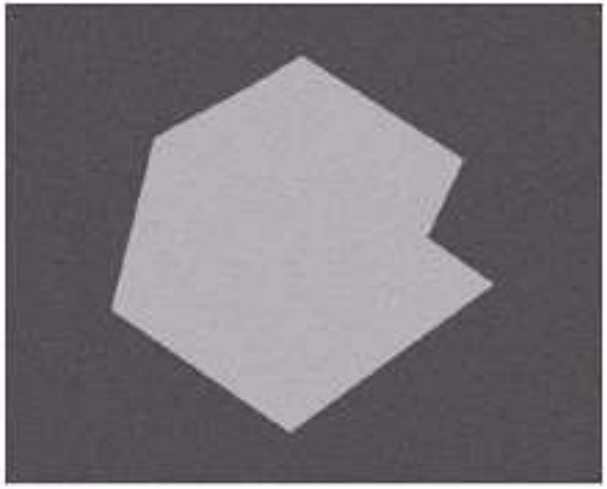
Seuillage global automatique –exemple

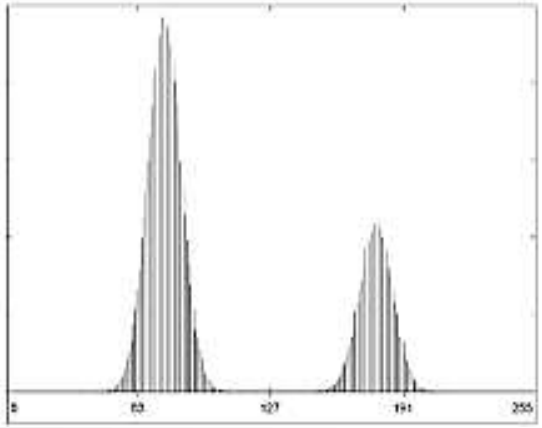
- Seuil trouvé par l'algorithme
  - $T = 125$

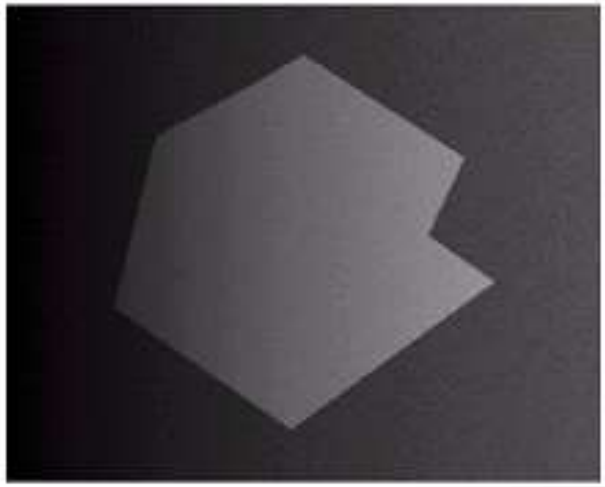


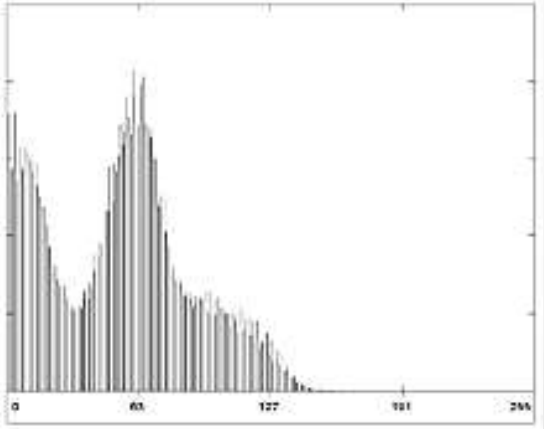
Seuillage global - problème

- Problème d'éclairage ?



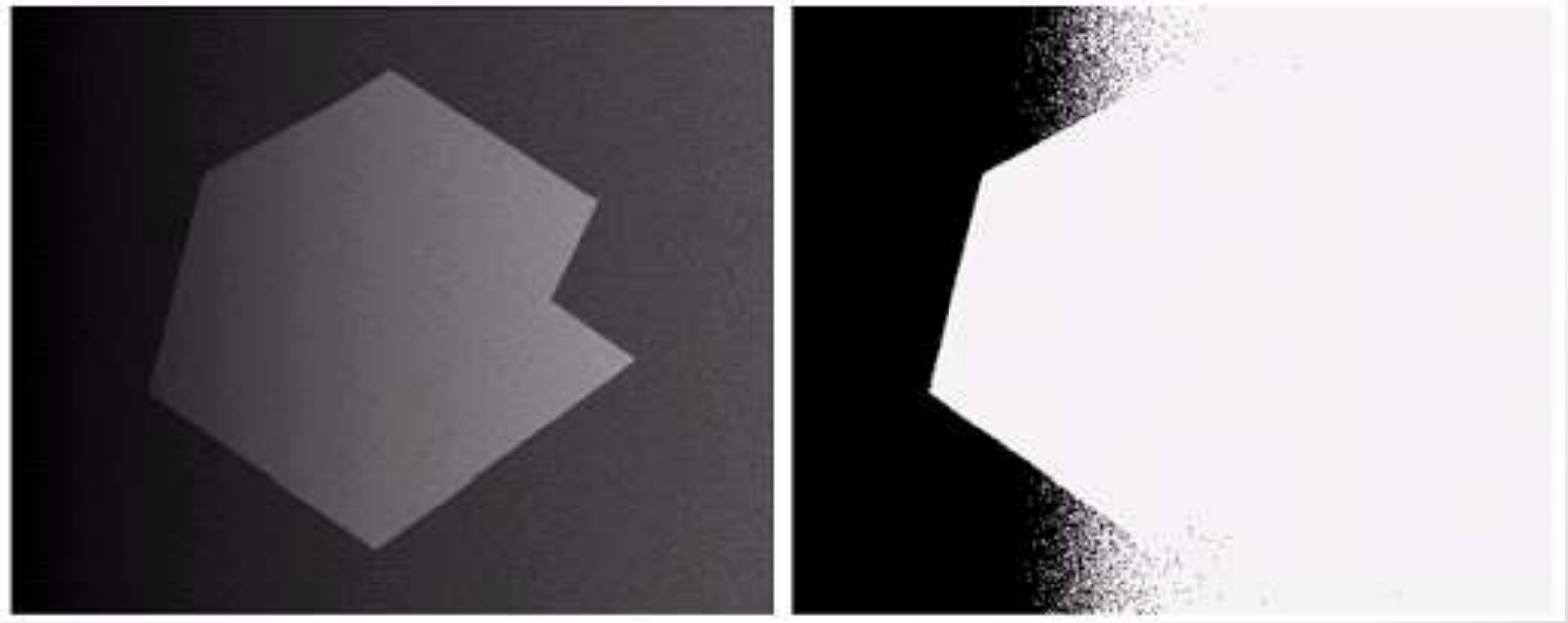






Seuillage global - problème

- Problème : Le seuillage global ne peut traiter ce cas
- Solution : seuillage local adaptatif

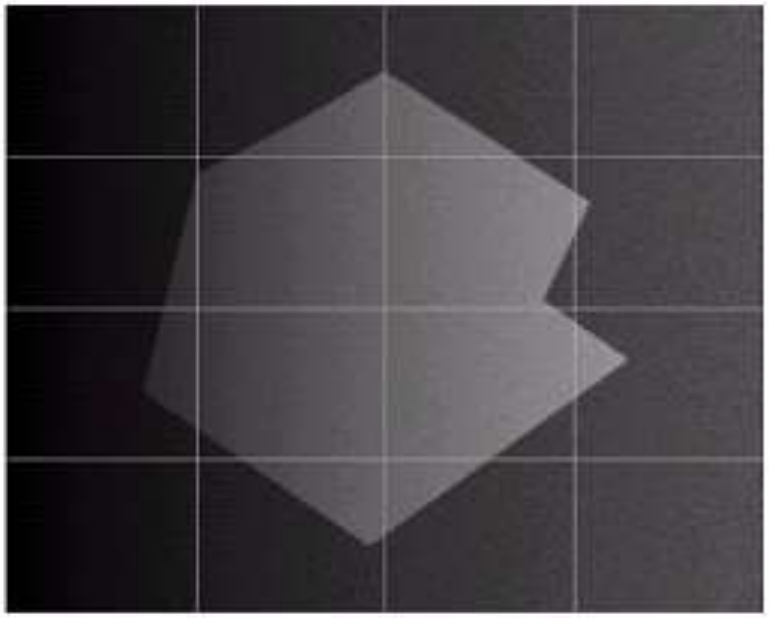


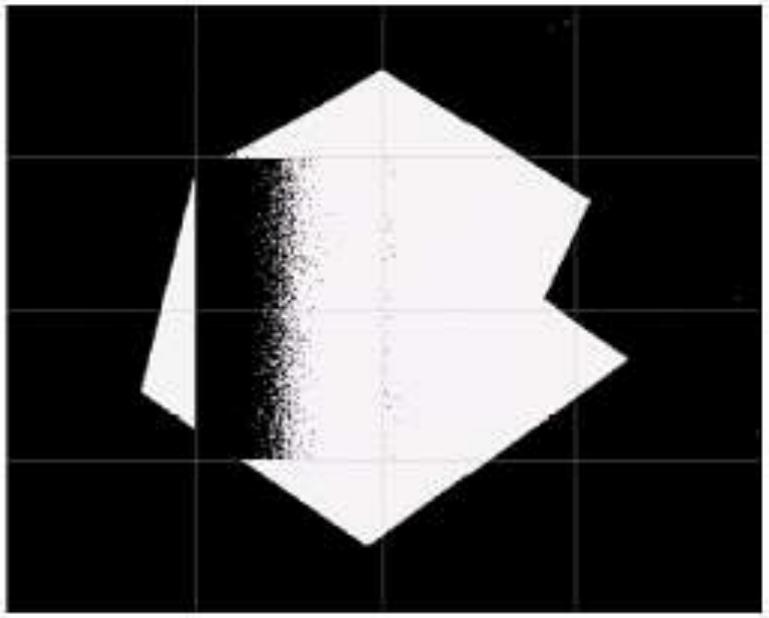
Exemple de seuillage adaptatif

- Nous avons besoin de séparer l'image en sous images, et de traiter chacune avec son propre seuil
- Le choix de la dimension des sous-images est critique
- Avant de traiter chaque sous-image, nous vérifions la variance des tons de gris pour décider s'il existe un besoin de segmentation
  - Exemple : pas besoin si variance<100

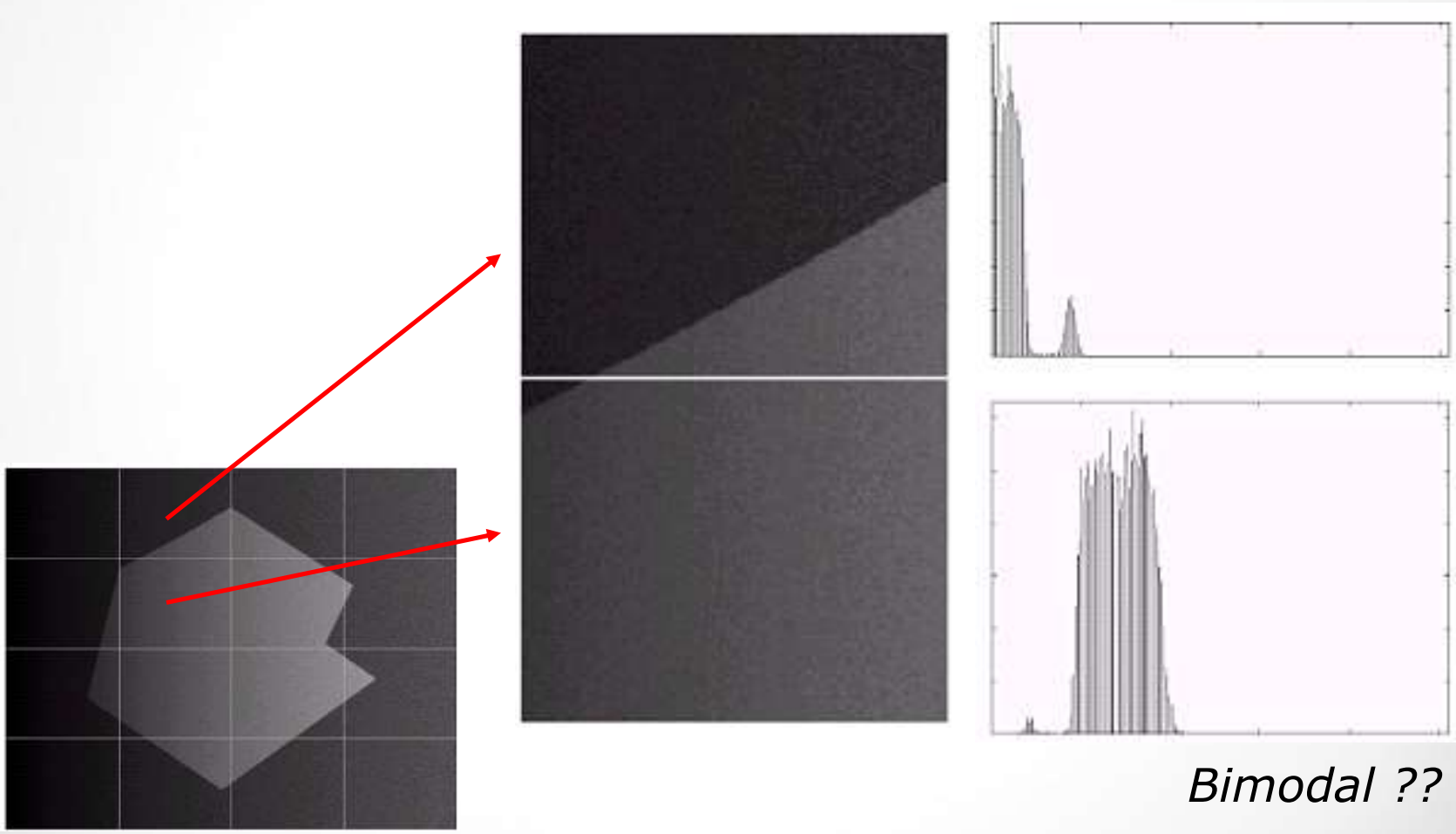
Exemple de seuillage adaptatif

- Les 4 sous images de coins ne sont pas traitées car *variance<100*



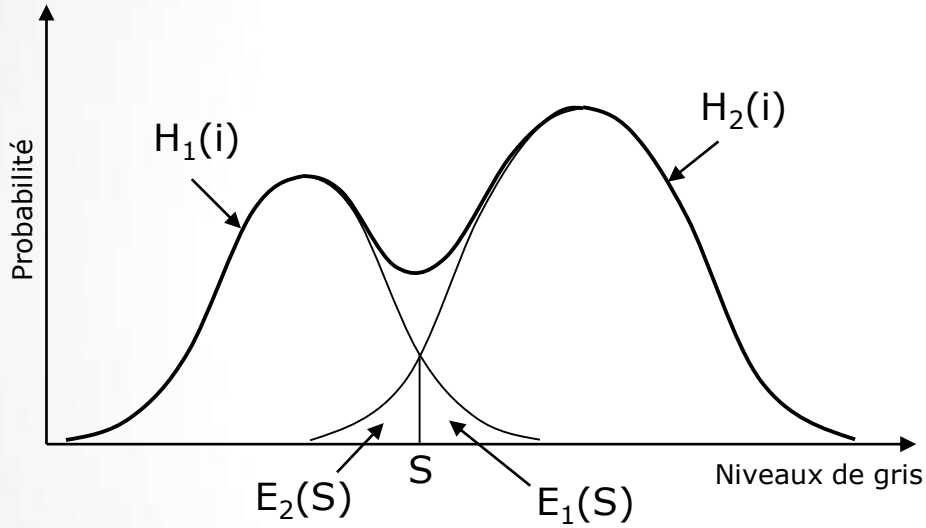


Exemple de seuillage adaptatif



Autres seuillage adaptatif

Seuillage optimal MEP (Minimum Error Probability) :



$$E_1(S) = P_1 \cdot \sum_{i=S+1}^{255} H_1(i)$$

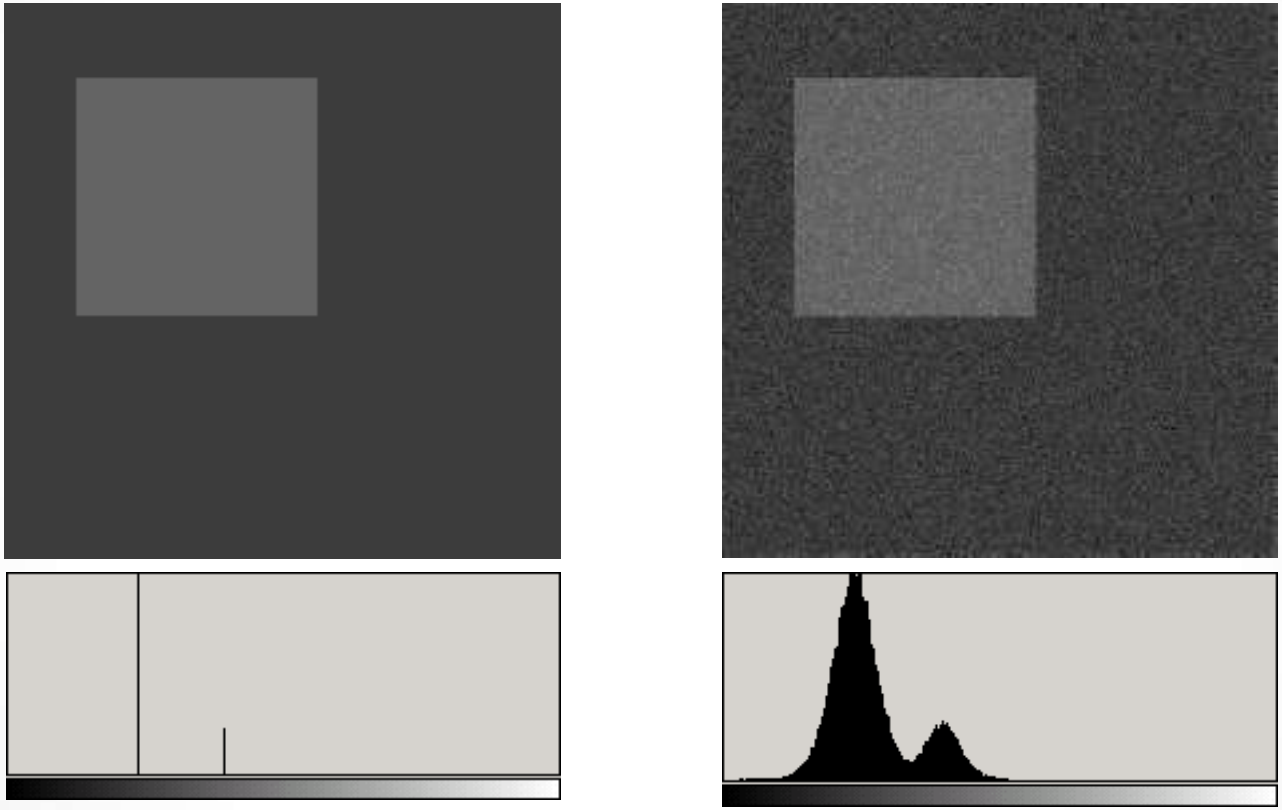
$$E_2(S) = P_2 \cdot \sum_{i=0}^{S} H_2(i)$$

Le seuil S doit être sélectionné pour minimiser :

$$E(S) = E_1(S) + E_2(S) = P_1 \cdot \sum_{i=S+1}^{255} H_1(i) + P_2 \cdot \sum_{i=0}^{S} H_2(i)$$

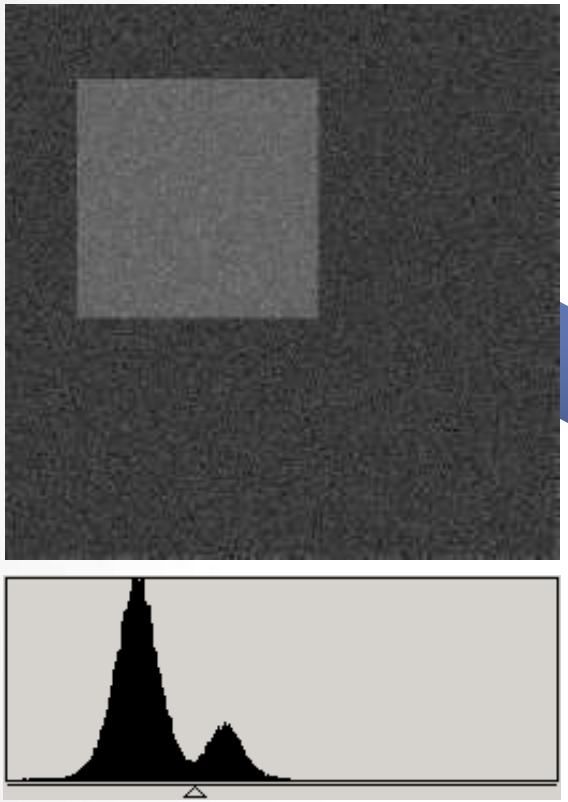
Autres seuillage adaptatif

Seuillage optimal sur une image de synthèse :



$m_1=60;\ m_2=100;\ P_1=0,8;\ P_2=0,2;\ \sigma=15$

Autres seuillage adaptatif



$$S_{opt} = \frac{m_1 + m_2}{2} + \frac{\sigma^2}{m_1 - m_2} \ln \frac{P_2}{P_1} = 87$$

Nombre de pixels mal classés = **38**

Autres seuillage adaptatif

Approche d'Otsu :

La distance entre 2 classes est définie par la variance inter-classe

$$\sigma^2_{\text{inter-classe}} = P_1(S)P_2(S)\left[m_1(S) - m_2(S)\right]^2$$

ou par la variance inter-classe / la variance intra-classe

$$\frac{\sigma^2_{\text{inter-classe}}}{\sigma^2_{\text{intra-classe}}} = \frac{P_1(S)P_2(S)\left[m_1(S) - m_2(S)\right]^2}{P_1(S)\sigma_1^2(S) + P_2(S)\sigma_2^2(S)}$$

Implantation :

- **Calculer la variance inter-classe ou la variance inter-classe / la variance intra-classe pour chaque valeur possible de seuil (0 - 254)**
- **Ensuite, rechercher le maximum auquel correspond le meilleur seuil.**

Autres seuillage adaptatif

Approche de Kapur :

Elle est basée sur la théorie de l'entropie qui se définie par

$$\text{Entropie totale} = -\sum_{i=0}^{S} \frac{H(i)}{P_1(S)} \ln\left(\frac{H(i)}{P_1(S)}\right) - \sum_{i=S+1}^{255} \frac{H(i)}{P_2(S)} \ln\left(\frac{H(i)}{P_2(S)}\right)$$

Implantation :

- **Calculer l'entropie totale pour chaque valeur possible de seuil (0 - 254).**
- **Ensuite, rechercher le maximum auquel correspond le meilleur seuil.**

Autres seuillage adaptatif

Image originale et son histogramme

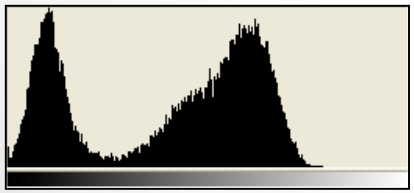
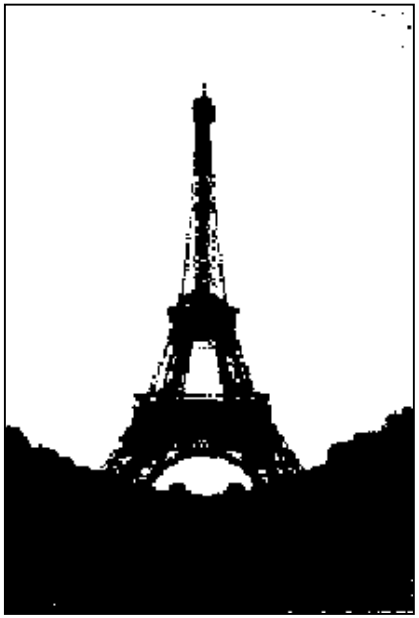
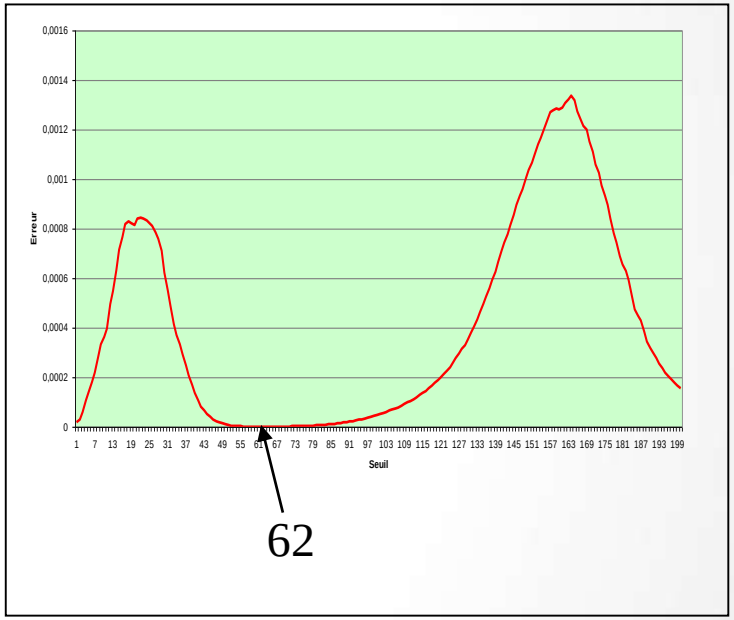


Image segmentée par le seuillage optimal MEP



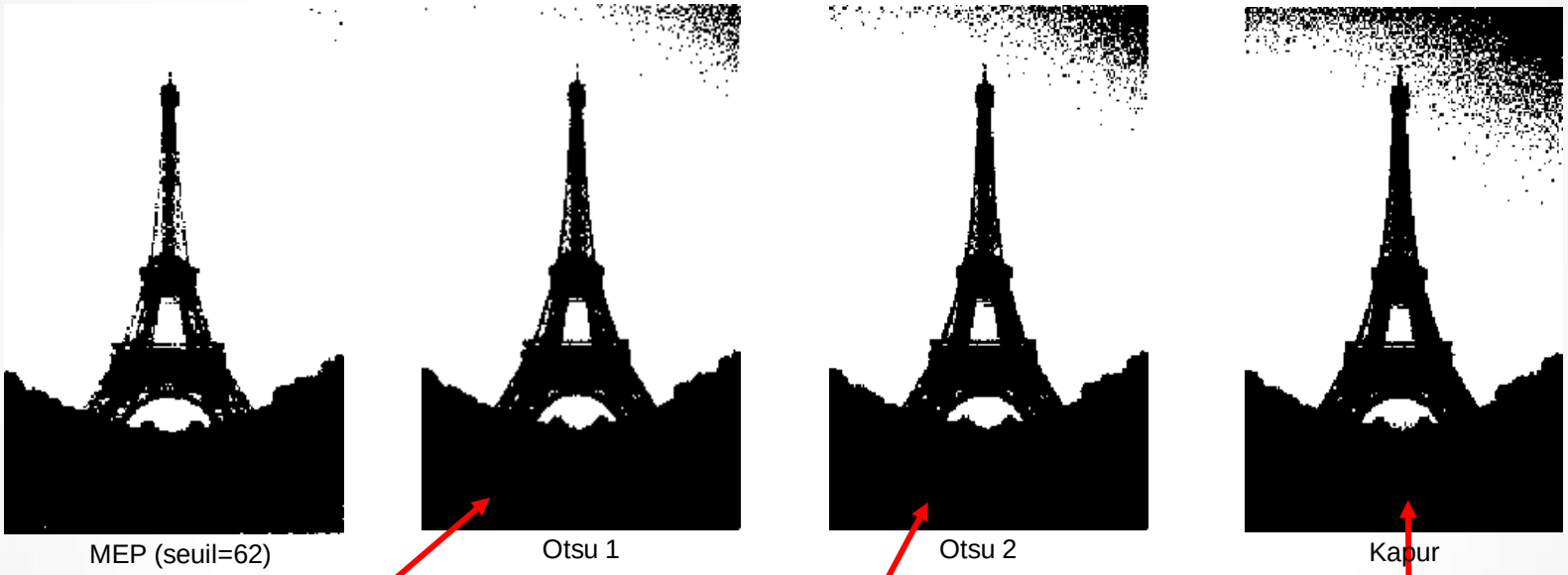
Seuil = 62



Chercher le seuil optimal

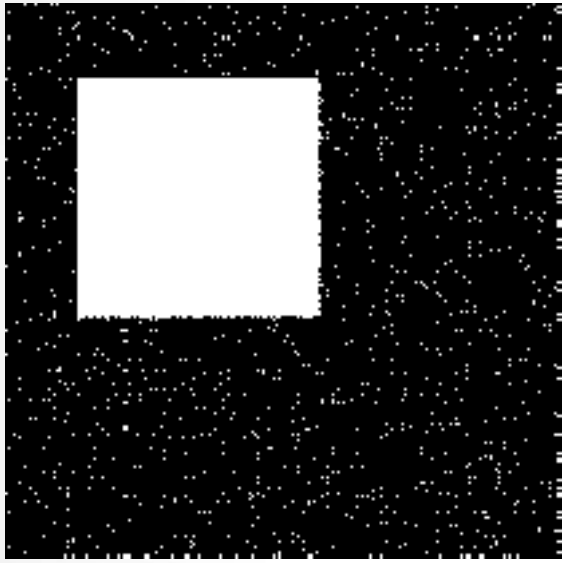
Autres seuillage adaptatif

Comparaison :

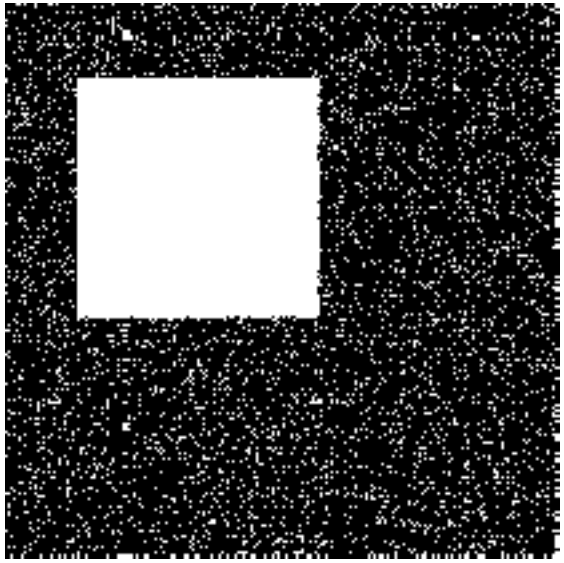


Autres seuillage adaptatif

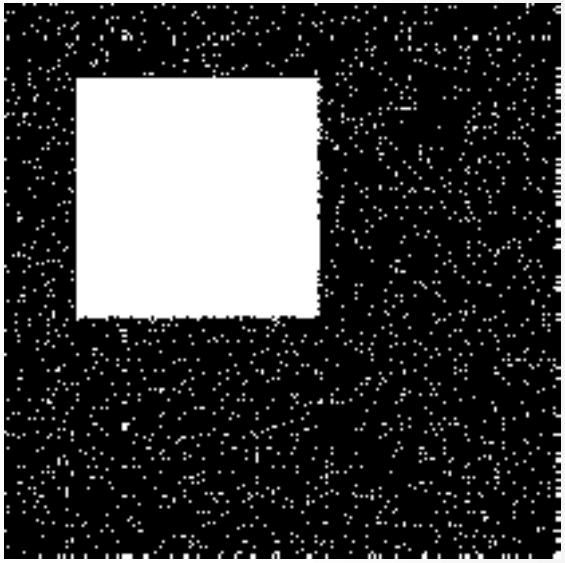
Comparaison :



Seuillage d'Otsu 1 (seuil=79)



Seuillage d'Otsu 2 (seuil=72)



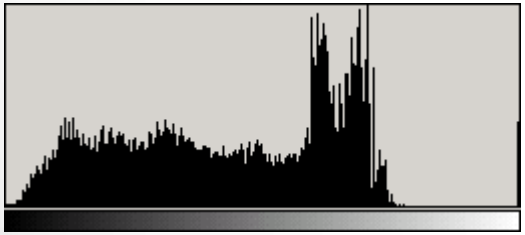
Seuillage de Kapur (seuil=76)

Méthode	Optimal	Otsu 1	Otsu 2	Kapur
Seuil	87	79	72	76
Nb de pixels mal classés	38	1163	3975	2020
	0,88%	2,69%	9,19%	4,67%

Autres seuillage adaptatif

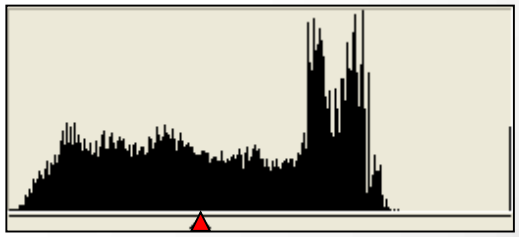
Exemple :

Image de départ



Otsu1 Seuil=108

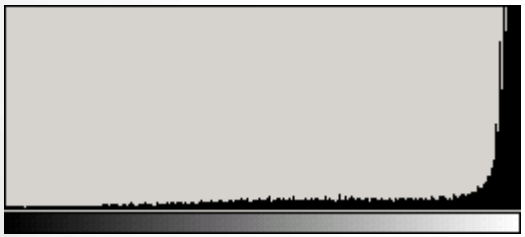
Kapur Seuil=97



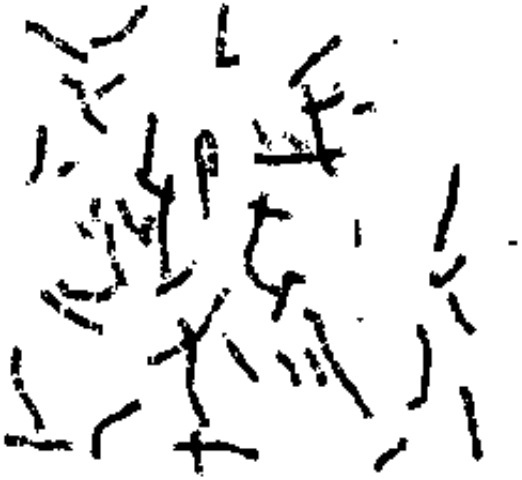
Autres seuillage adaptatif

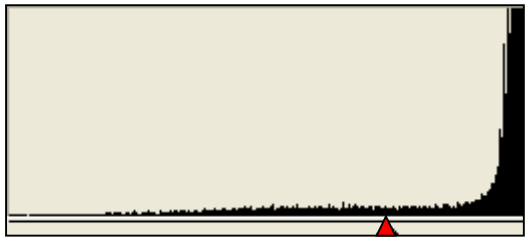
Exemple :

Image de départ

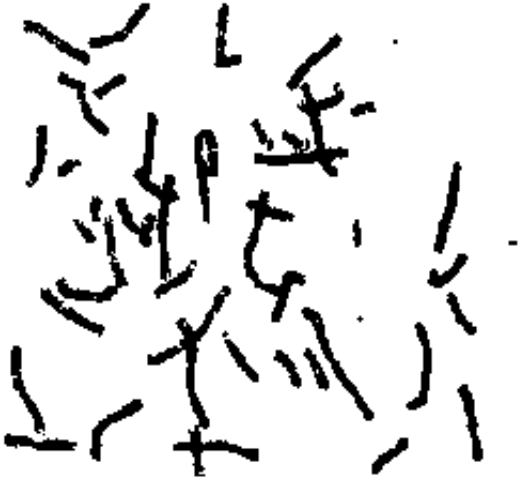


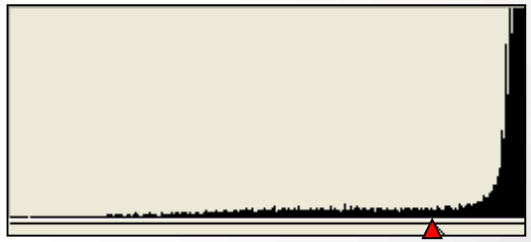
Otsu1 Seuil=188





Kapur Seuil=211

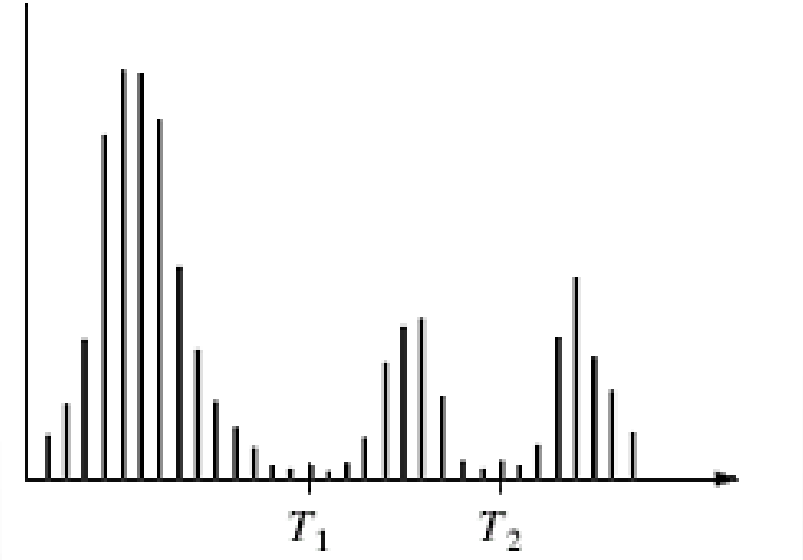




Seuillage d'histogramme multiple

- Seuillage multiple

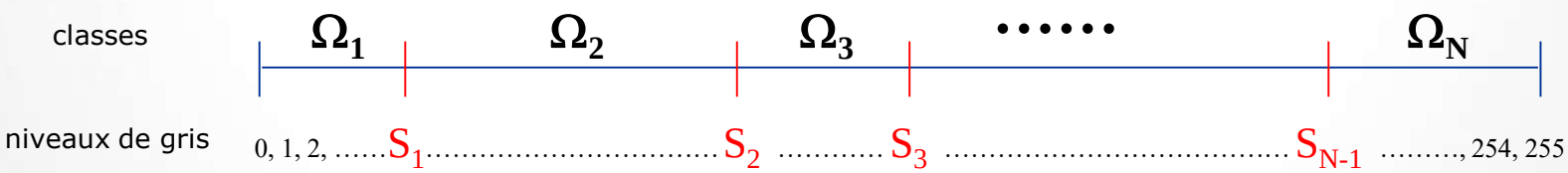
$$g(x,y)=\begin{cases}2 & si\ f(x,y)>T_2\\ 1 & si\ T_2\geq f(x,y)>T_1\\ 0 & si\ f(x,y)\leq T\end{cases}$$



Multi-seuillage

Multi-seuillage :

Soit G={0,1, 2, 3, … , 255} l'ensemble des niveaux de gris de l'image et P={Ω_1, Ω_2, Ω_3, …, Ω_N} une partition de G en N classes. Le problème consiste à optimiser la partition P en identifiant les séparateurs S_1, S_2, S_3, …, S_{N-1} (seuils) entre les classes.



Multi-seuillage

Exemple :

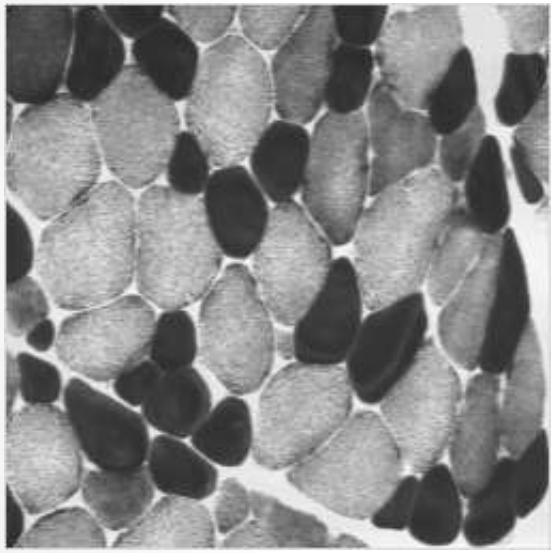
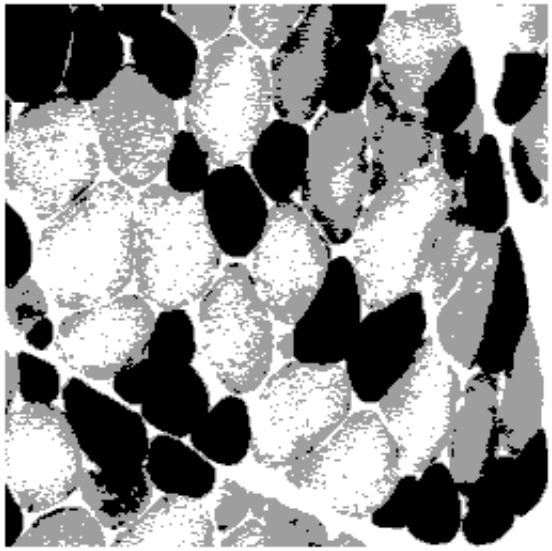
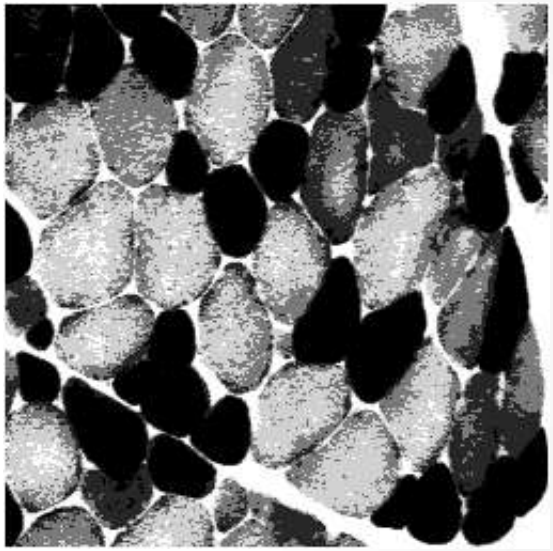


Image initiale: Muscle



3 classes avec seuils
102, 168



5 classes avec seuils
86, 132, 168, 205

Multi-seuillage

Détection des vallées :

On suppose que chaque classe correspond à une gamme distincte de niveaux de gris. L'histogramme est alors N-modal. La position des minima de l'histogramme H permet de fixer les (N-1) seuils nécessaires pour séparer les N classes.

$$H(s_i) = Min\left[H(k)\right] \qquad \text{pour } k \in \left]\, m_i, m_{i+1} \,\right[$$

où m_i **et** m_{i+1} **sont les valeurs moyennes (ou les modes) de niveau de gris dans les classes** C_i **et** C_{i+1}**.**

Multi-seuillage

Histogramme cumulé : $H_c(i)=\sum_{j=0}^{i} H(j)$

Fonction R : $R(i)=H_c(i)-\frac{1}{2N+1}\sum_{k=i-N}^{i+N} H_c(k)$

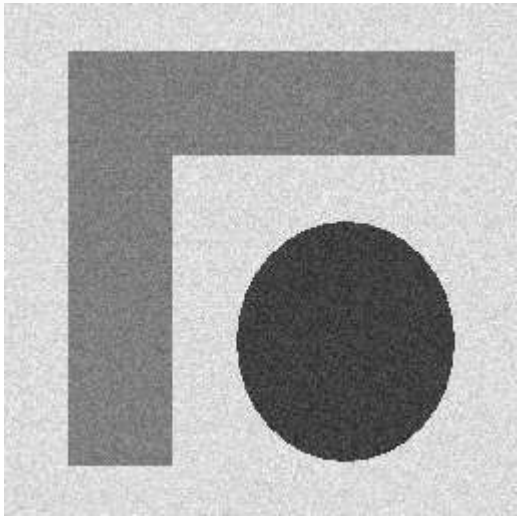
où $\frac{1}{2N+1}\sum_{k=i-N}^{i+N} H_c(k)$ **est un lissage de** $H_c(i)$

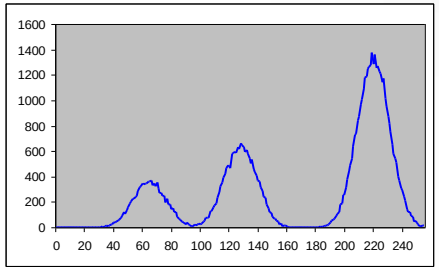
Passages par zéro sur R : $+ \rightarrow -$ **correspondent aux vallées**

$- \rightarrow +$ **correspondent aux pics**

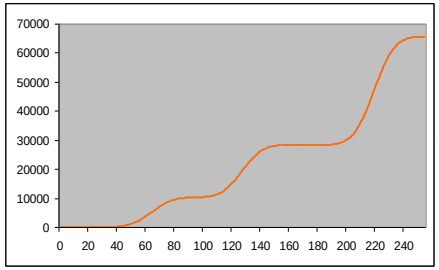
Multi-seuillage

Image initiale



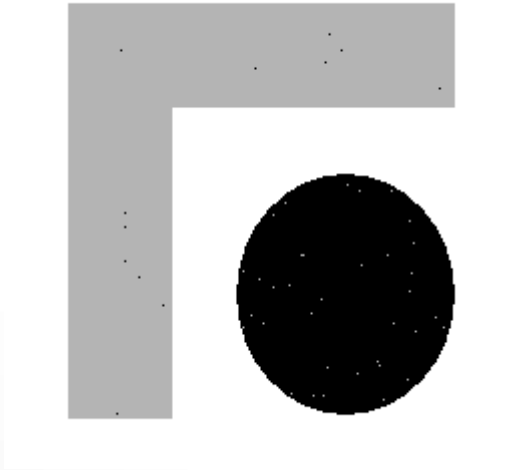


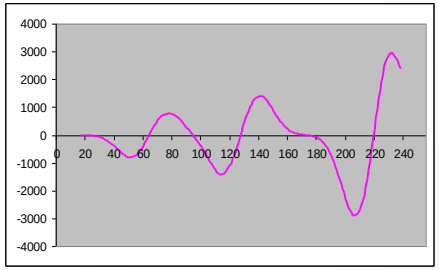
Histogramme H(i)



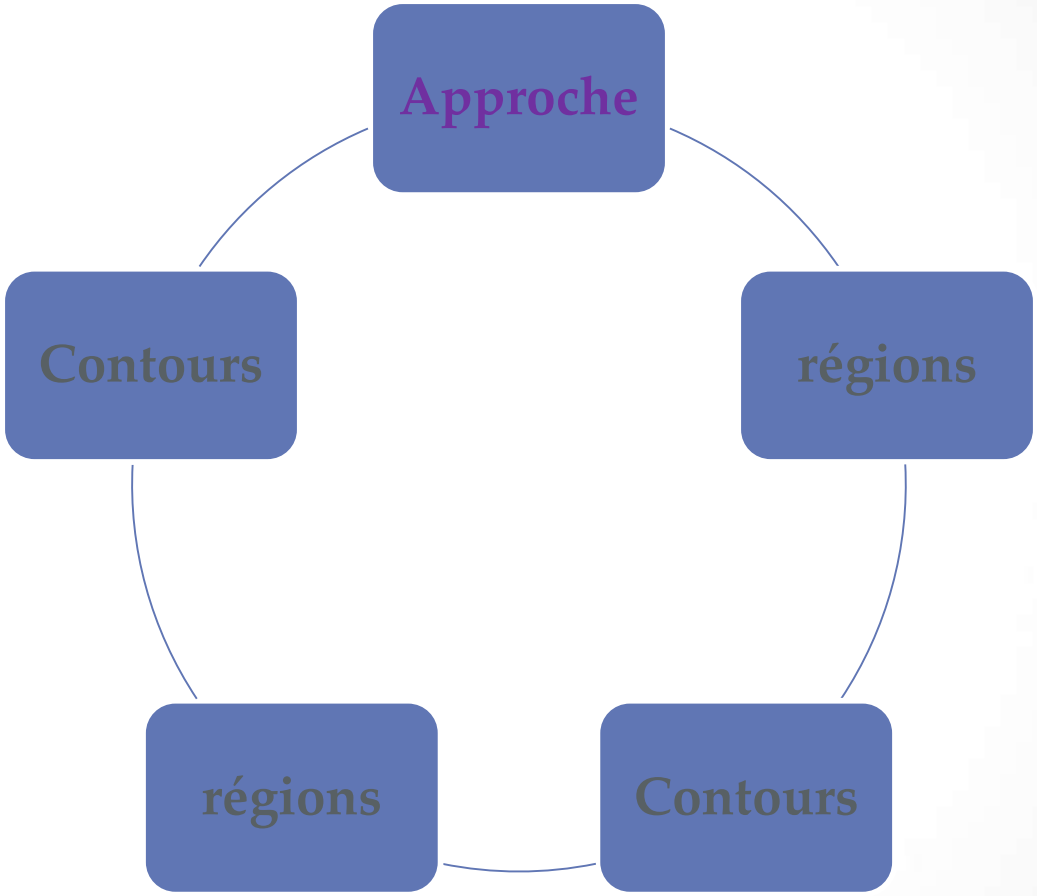
Histogramme cumulé $H_c(i)$

Image seuilée avec
s1=94 s2=171





R(i) avec N=17


Approche
régions
Contours
Contours
régions

Segmentation régions/contours

- Approches régions
  - Rechercher les zones dans l'image sur un critère d'homogénéité
- Approches contours
  - Rechercher les discontinuités entre régions
- Approches duales (régions et contours) seulement dans quelques rares cas

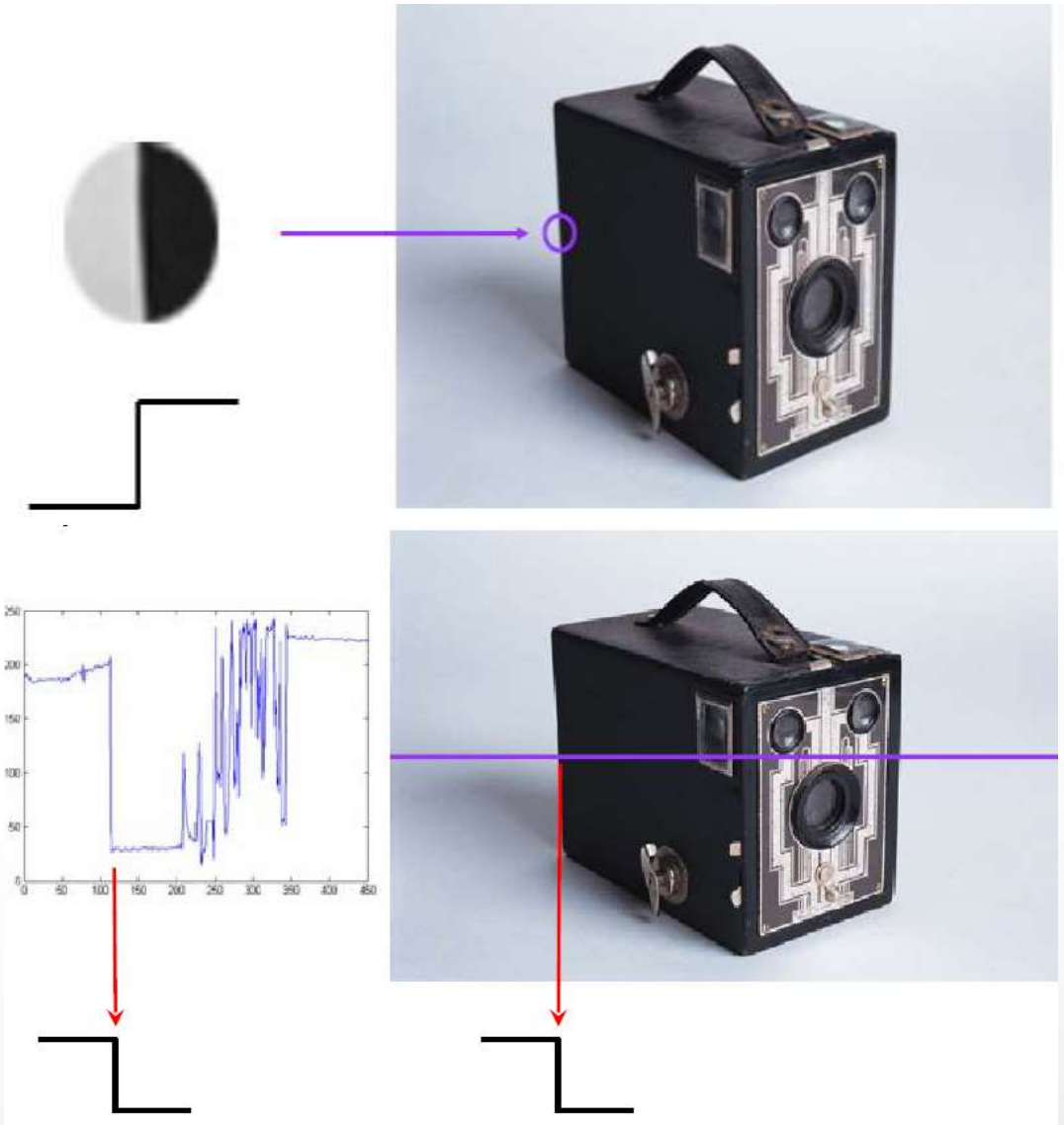
- Il existe une dualité entre régions et contours
  - Une région est délimitée par un contour
  - Un contour sépare deux régions

Segmentation contours

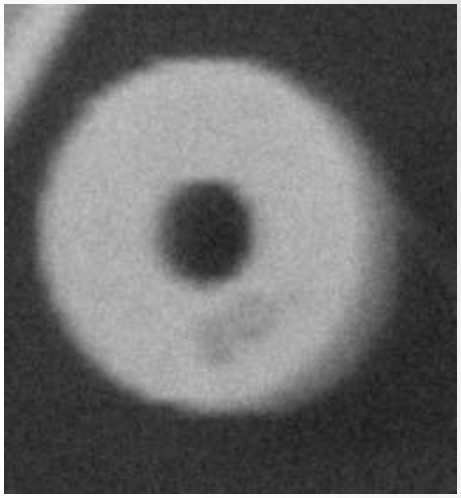
- La segmentation par contours exploite les transitions détectables entre deux régions voisines.
- La segmentation en contours d'une image se fait principalement en utilisant les détecteurs par convolution déjà vus.
- Cependant, ces détecteurs ne donne pas des contours fermés et sont sensibles aux bruits.
  - Une étape de seuillage du gradient, du Laplacien ou autre est nécessaire.
  - Ensuite, il faut fermer les contours et conserver les contours significatifs.

Qu'est-ce qu'un contour ?

Un contour est une variation brusque d'intensité

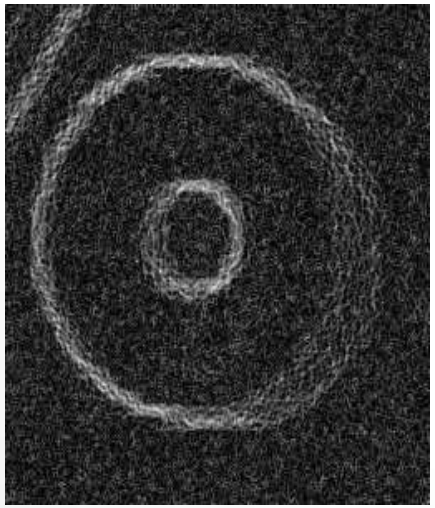


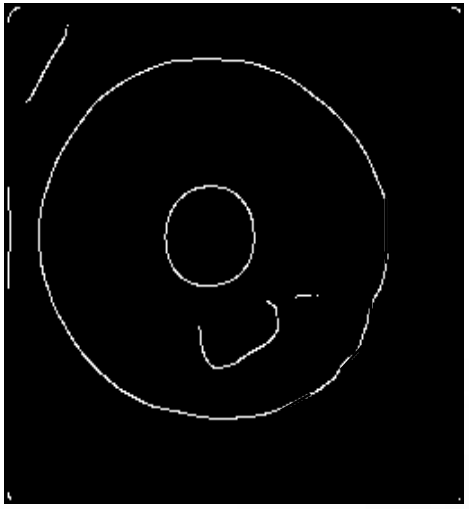
détection / fermeture de contours

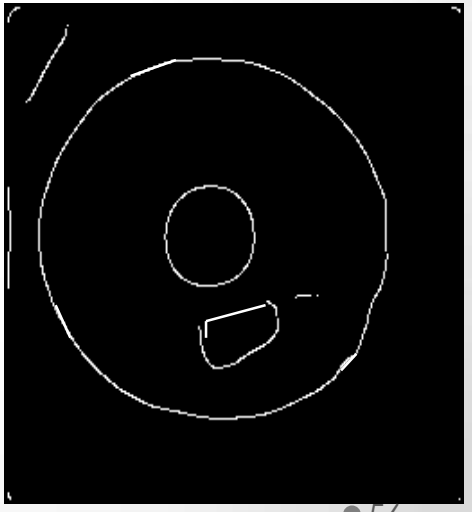


La détection de contour est suivie en générale d'une localisation de contour et de la recherche d'un ensemble connexe de points

Détection de contour → Extraction de contour → Fermeture de contour

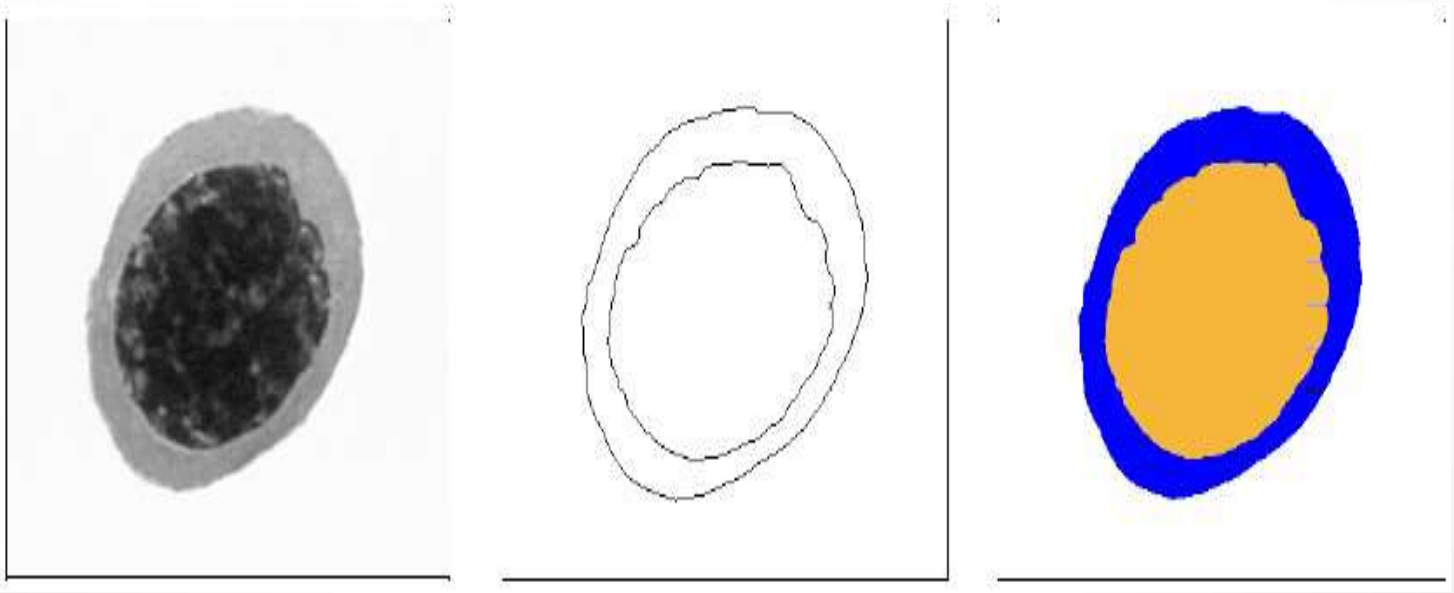






Segmentation Contour

Un contour fermé est équivalent à une région.



Dérivée discrète

On utilise la première dérivée de l'image pour les contours :

$$\frac{\Delta I}{\Delta x} = \frac{I(x+\Delta x) - I(x)}{\Delta x}$$

Approximation simple de la dérivée discrète :

-1	1

-1
1

ou encore :

-1	0	1

-1
0
1

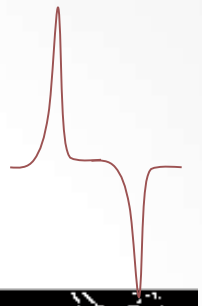
Comment les filtres sont calculer a la base?

- Plusieurs autres filtres existent pour la détection des contours dans l'image
- On fait lissage de l'image + dérivée de l'image (sauf Roberts, juste dérivée)

Extraction des points de contour

Méthodes dérivatives - Opérateurs de Robets

Détection de contours



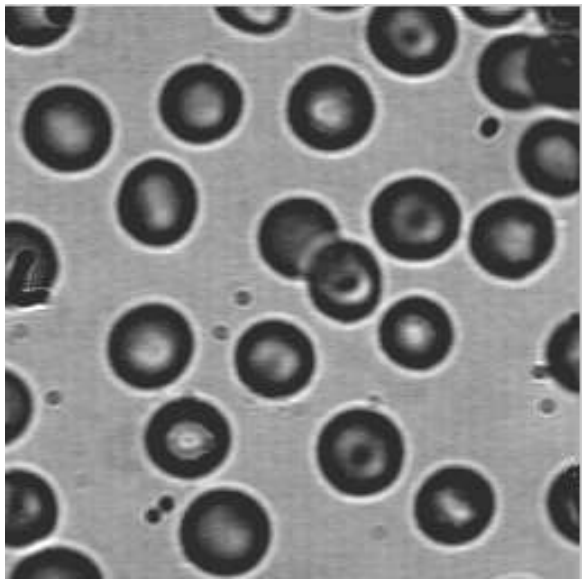


Image originale

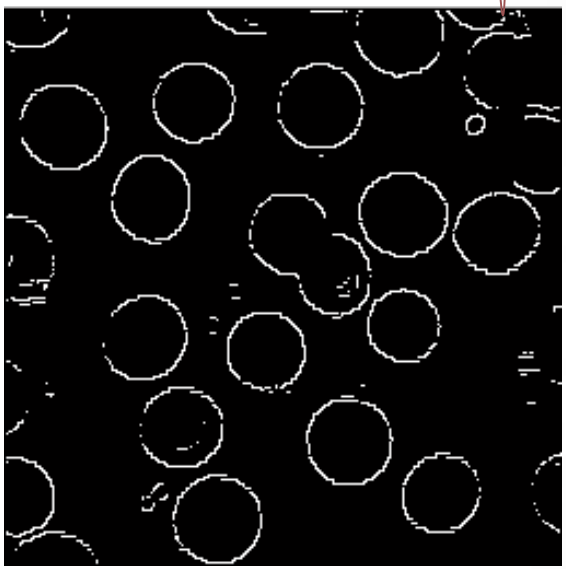


Image des contours
(opérateur Roberts)

1	0
0	-1

0	1
-1	0

Comment les autre filtres sont calculer?

- Il existe beaucoup d'autres filtres pour détecter les contours
- On fait lissage de l'image + dérivée de l'image

Filtre de Prewitt : Moyenneur + Dérivée

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} * \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} * \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

Filtre de Sobel : Gaussienne + Dérivée

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} * \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$$

⇒ ***Détection des contours moins sensible au bruit***

Extraction des points de contour

Méthodes dérivatives - Opérateurs de Prewitt

Détection de contours

-1	-1	-1
0	0	0
1	1	1

Dérivée selon y

-1	0	1
-1	0	1
-1	0	1

Dérivée selon x

$$R = \sum w_i f_i = (f_7 - f_1) + (f_8 - f_2) + (f_9 - f_3)$$

$$R = \sum w_i f_i = (f_3 - f_1) + (f_6 - f_4) + (f_9 - f_7)$$

Extraction des points de contour

Méthodes dérivatives - Opérateurs de Prewitt

Détection de contours

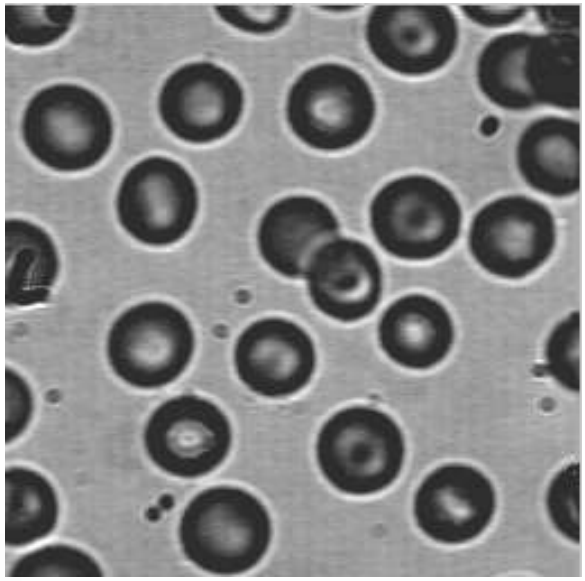


Image originale

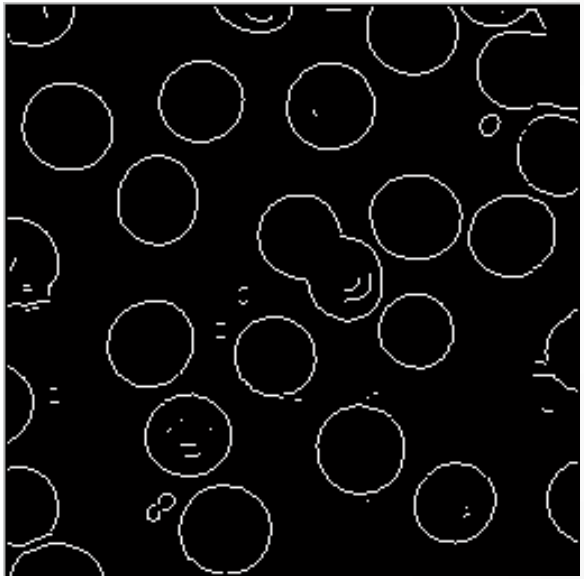


Image des contours
(opérateur Prewitt)

Extraction des points de contour

Méthodes dérivatives - Opérateurs de Sobel

Détection de contours

-1	-2	-1
0	0	0
1	2	1

Dérivée selon y

-1	0	1
-2	0	2
-1	0	1

Dérivée selon x

$$R = \sum w_i f_i = (f_7 - f_1) + 2(f_8 - f_2) + (f_9 - f_3)$$

Extraction des points de contour

Méthodes dérivatives - Opérateurs de Sobel

Détection de contours

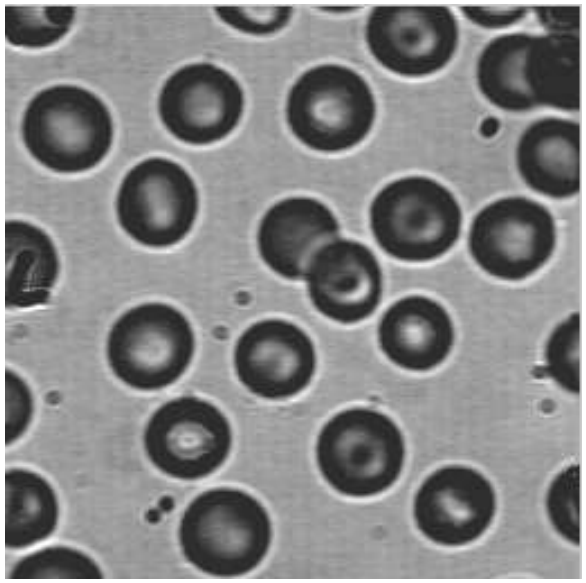


Image originale

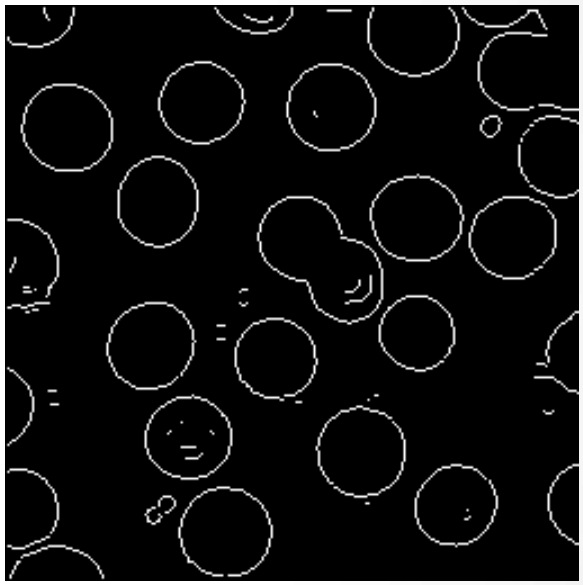
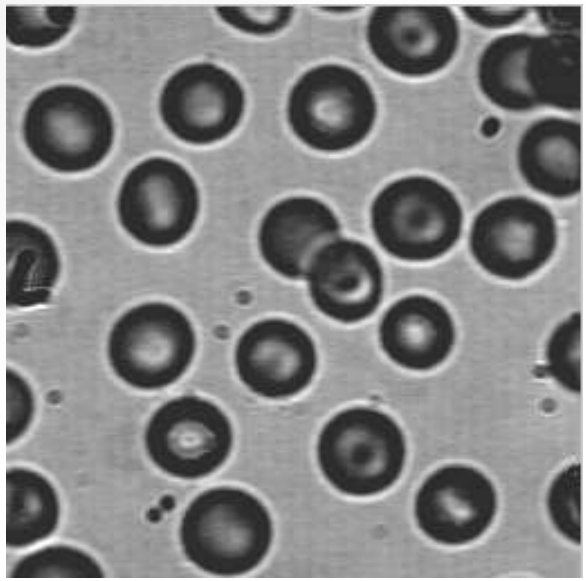
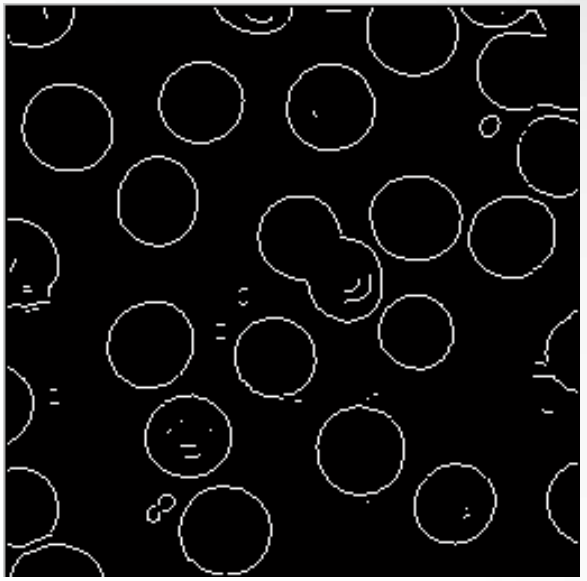
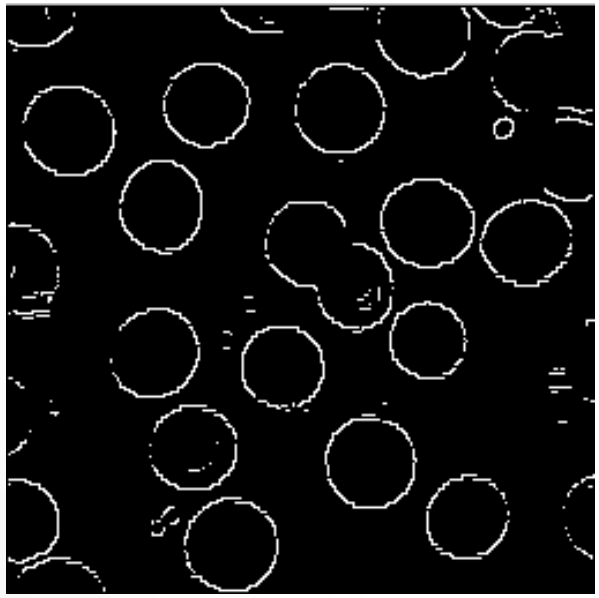


Image des contours
(opérateur Sobel)

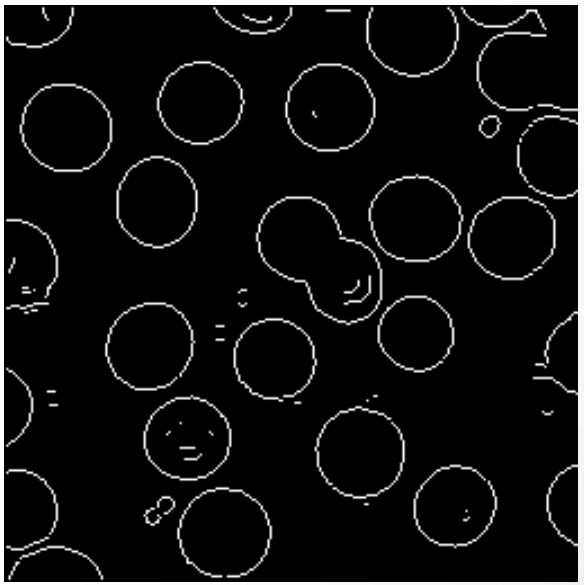




Prewit



Roberts



Sobel

Extraction des points de contour

Méthode Laplacien

Dérivée seconde directionnelle

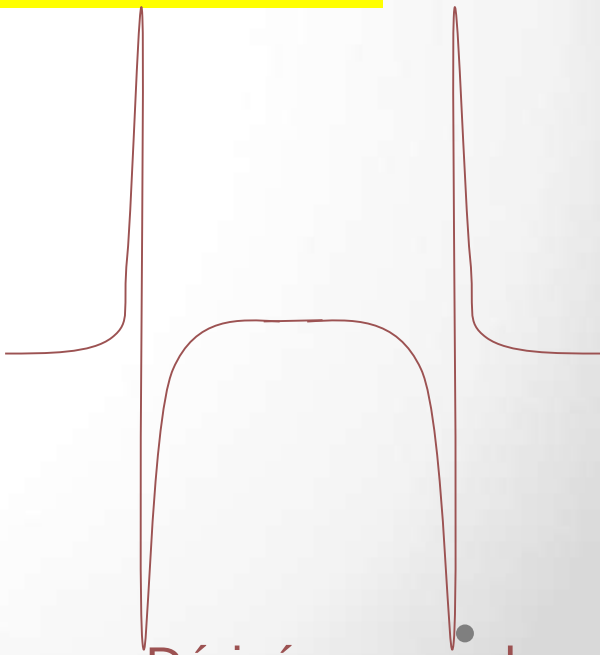
$$f'_\alpha = \frac{df}{d\alpha}$$

$$f''_\alpha = \lim_{h \to 0} \frac{f'_\alpha(x + h\cos\alpha, y + h\sin\alpha) - f'_\alpha(x, y)}{h}$$

$$\frac{d^2 f}{d\alpha^2} = \frac{\partial^2 f}{\partial x^2}.\cos^2\alpha + 2\frac{\partial^2 f}{\partial x \partial y}.\cos\alpha\sin\alpha + \frac{\partial^2 f}{\partial y^2}\sin^2\alpha$$

Laplacien

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



Dérivée seconde

Extraction des points de contour

Méthode Laplacien

Approximation des dérivées partielles

Ce qui conduit à:

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{p,q} \approx \frac{1}{4\lambda_x^2}\left[f(p+1,q)-2f(p,q)+f(p-1,q)\right]$$

$\frac{\partial^2 f}{\partial x^2}$: $\frac{1}{4}$

-1	2	-1

$$\left.\frac{\partial^2 f}{\partial y^2}\right|_{p,q} \approx \frac{1}{4\lambda_y^2}\left[f(p,q+1)-2f(p,q)+f(p,q-1)\right]$$

$\frac{\partial^2 f}{\partial y^2}$: $\frac{1}{4}$

-1
2
-1

Δ : $\frac{1}{4}$

	-1	
-1	4	-1
	-1	

Extraction des points de contour

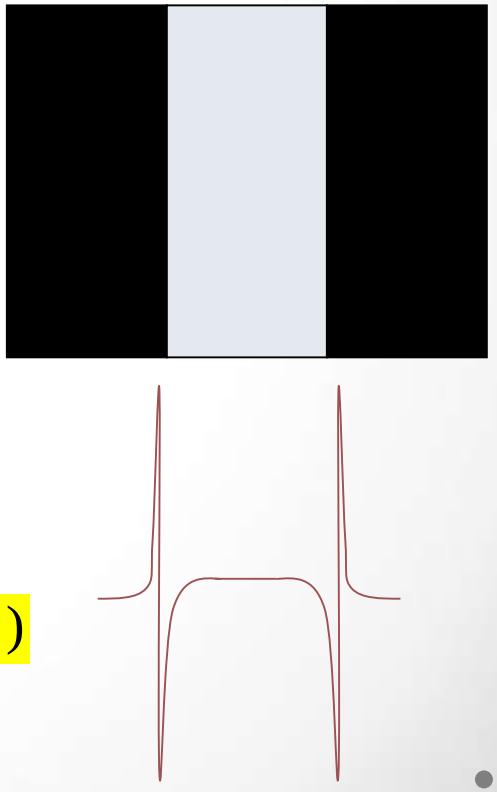
Méthode Laplacien

Détection de contours

- Sensible au bruit
- Il produit une ligne double
- Passage à 0 (*zero crossing*)

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

$$R = \sum w_i f_i = 4 f_5 - (f_2 + f_4 + f_6 + f_8)$$



Extraction des points de contour

Méthode Laplacien

Détection de contours

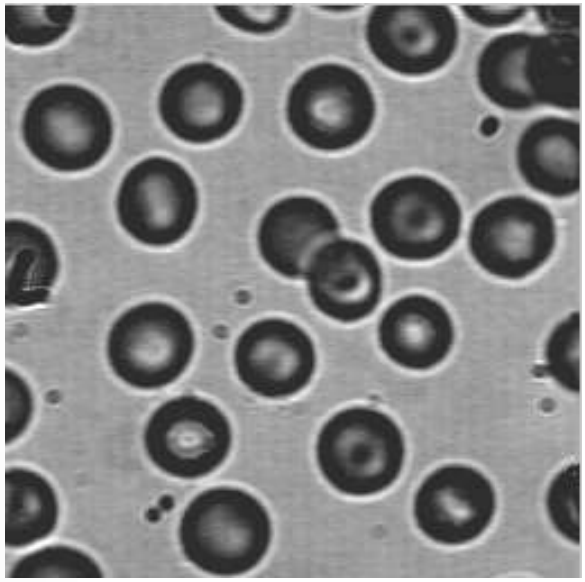


Image originale

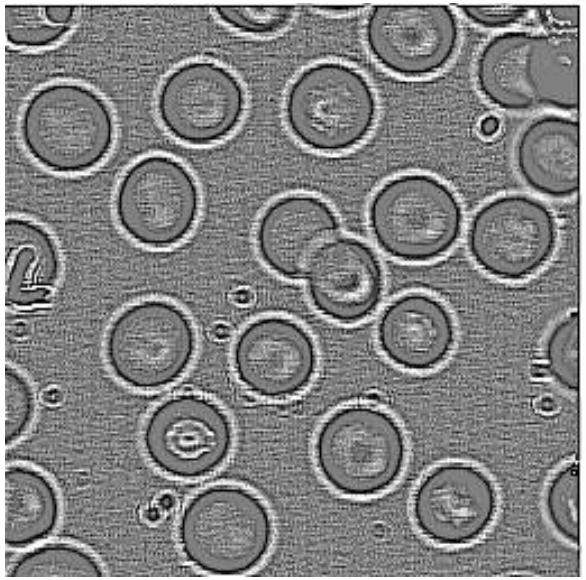


Image des contours
(zero crossing)

Extraction des points de contour

Détection de points isolés

$$R = \sum w_i f_i = w_1 f_1 + w_2 f_2 + \ldots + w_n f_n$$

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

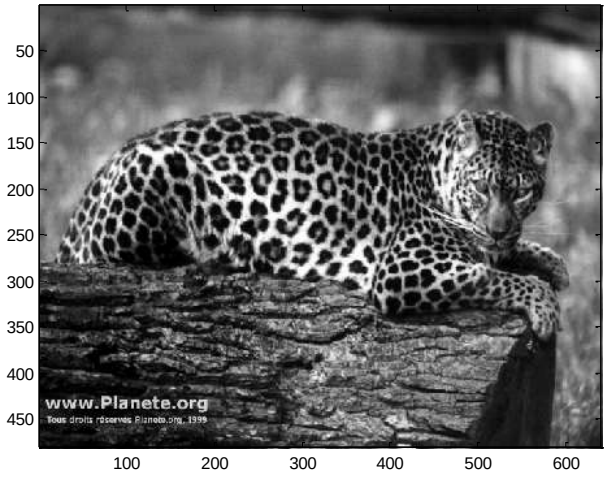
et test

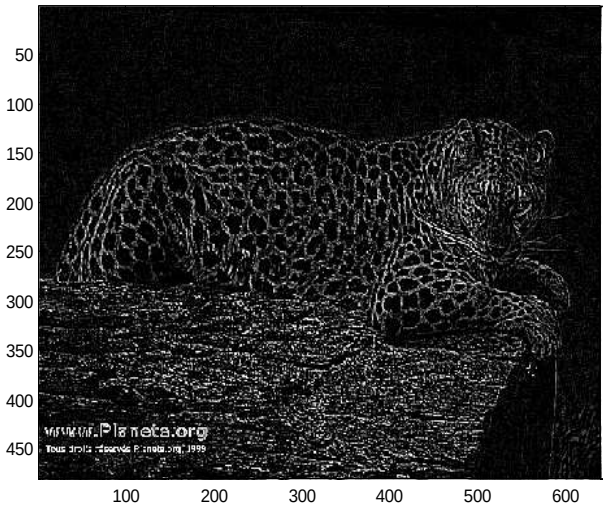
$$|R| > S$$

|R| : valeur absolue de R

S : Seuil

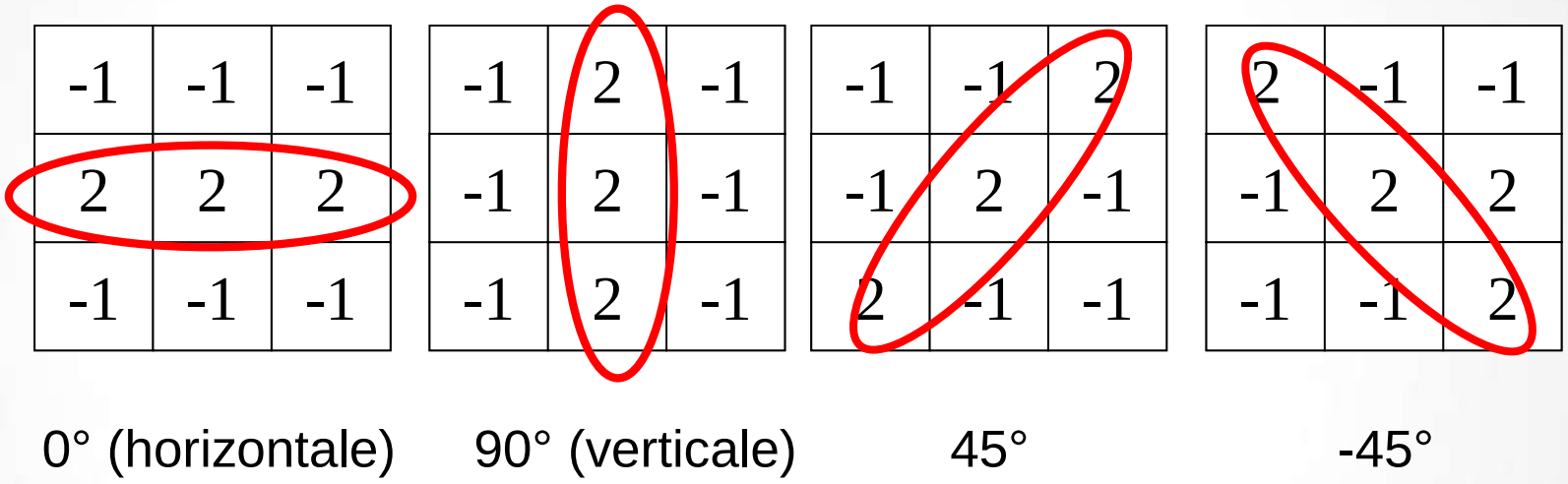
(Image,Masque) **Convolution** → Image





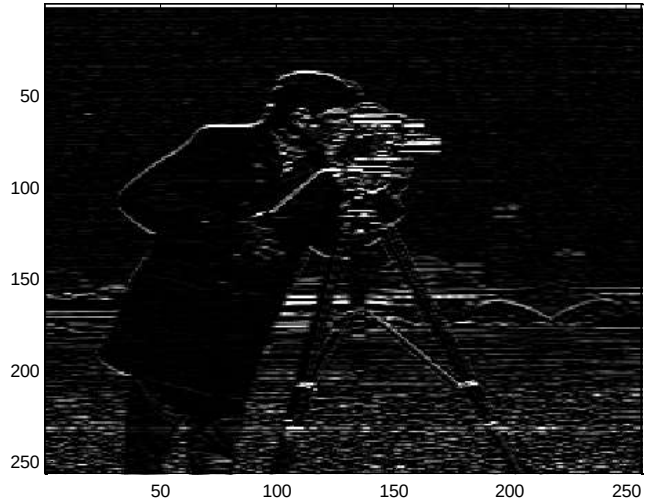
Extraction des points de contour

Détection de droites

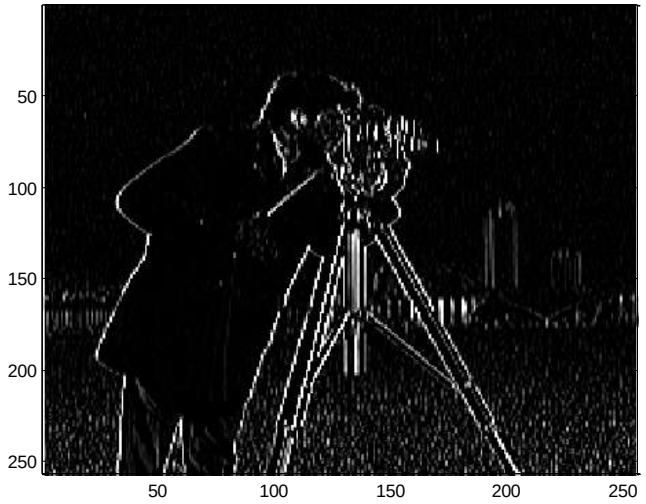


Extraction des points de contour

Détection de droites -Exemple-



0° (horizontale)



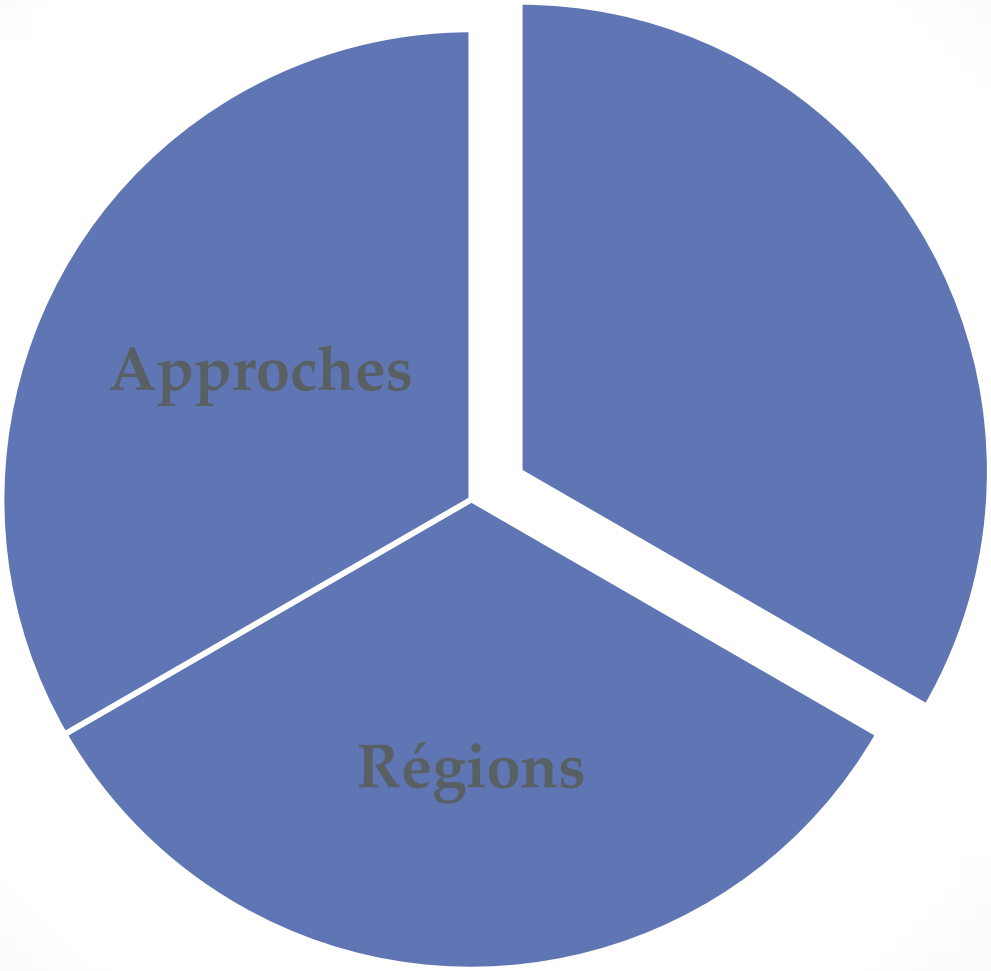
90° (verticale)

Éventail de méthodes de détection de contour

- Point d'interets:
  - Détecteur Harris-stephans, Détecteur SHIFT,
- Dérivation au premier ordre
  - Prewitt, Sobel, Roberts, Kirsh, Compass, dérivateurs...
- Dérivation au second ordre
  - Laplacien, Marr et Hildreth,...
- Filtrage optimal
  - Canny-Deriche, Shen
- Modélisation des contours
  - Hueckel, Haralick
- Morphologie mathématique
  - gradient morphologique, ligne de partage des eaux...
- Contours actifs
- Transformée de ***Hough***

Caractéristiques a prendre en considération :

Complexité, sensibilité géométrique, précision de localisation, sensibilité au bruit, création de faux contours

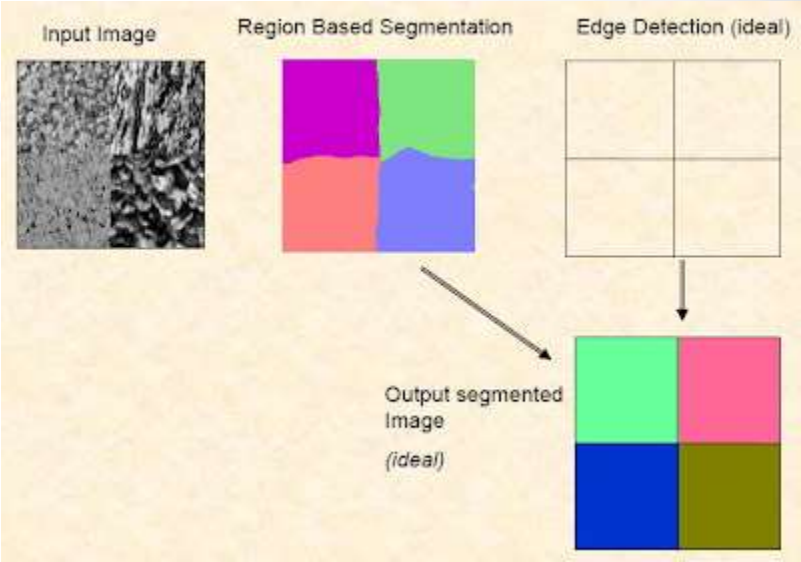

Approches
Régions

Segmentation en régions

- La segmentation par région consiste à regrouper les pixels similaires en régions, quant à la segmentation par contour, elle identifie les changements entre régions (contours)

Segmentation en régions

- Méthodes ascendantes (agrégation de pixels ou fusion de régions)
- Méthodes descendantes (division)
- Méthodes mixtes (division- fusion)



Contrairement à l'extraction des contours qui s'intéresse aux bords des régions, la segmentation en régions homogènes vise à segmenter l'image en se basant sur des propriétés intrinsèques des régions

Qu'est-ce qu'une Région ?

- **Définition de base :** un groupe de pixels connectés avec des propriétés similaires.
- Important dans l'interprétation d'une image car elles peuvent correspondre à des objets dans une scène.
- Pour une interprétation correcte, l'image doit être divisée en régions qui correspondent à des objets ou des parties d'un objet.
- s'appuient sur des modèles communs de valeurs d'intensité au sein d'un groupe de pixels voisins.
- Le cluster est appelé région et le but de l'algorithme de segmentation est de regrouper les régions en fonction de leurs rôles anatomiques ou fonctionnels.

Principes importants :

- Similitude des valeurs - Différences de valeurs de gris

  - Variation de la valeur de gris

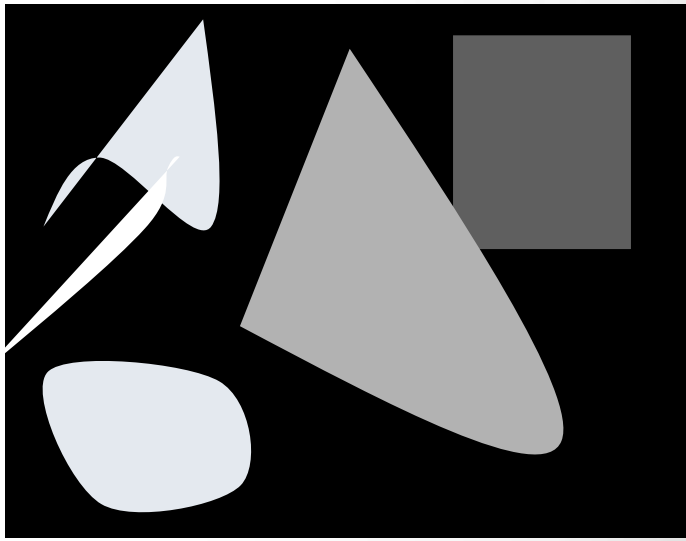
- Proximité spatiale - Distance euclidienne

  - Compacité d'une région

- Les méthodes de segmentation basées sur les régions tentent de partitionner ou de regrouper les régions en fonction de propriétés d'image communes. Ces propriétés d'image sont constituées de :
  - Valeurs d'intensité à partir d'images originales ou valeurs calculées basées sur un opérateur d'image
  - Textures ou motifs propres à chaque type de région

Segmentation en régions

La sortie d'un algorithme de segmentation est un ensemble de N régions {R_i, i = 1 …N} formant une partition du domaine de l'image, c'est-à-dire tel que :



1. $\bigcup_{i=1}^{N} R_i = \Omega,$ Ω : tout les regions
2. $R_i \cap R_j = 0, \forall i \neq j.$

Soit : Tout pixel doit appartenir à une et une seule région (classe) suivant un critère P appliqués aux Pixels

Les critère P de similarité sont: Valeurs de gris, Couleur, Texture, Taille Forme, Proximité spatiale et connecté

Mise en évidence des régions

La segmentation en régions ne doit pas être systématiquement associée à une représentation en image (informations sur la structure de l'image). Néanmoins, il est courant de visualiser des images de régions. Il faut pour cela :

– Affecter une couleur à chacune des régions
– Attribuer des niveaux très différents à deux régions adjacentes (pour pouvoir visualiser cette différence)
– Prendre en compte des nombres de régions supérieurs à 256

Croissance de région (Region Growing)

Idee: On part d'un point amorce (*seed) et l'on l'étend en ajoutant les points de la frontières* qui satisfont le critère d'homogénéité.

- On débute avec un pixel, et on « ajouter » les pixels voisins qui répondent à un critère d'appartenance :
  - *Variance faible*
  - *Niveau de gris répondant un seuil*
  - *…*
- Les pixels initiaux sont appelés « germes », « amorces » ou « semences »
  - *Choix des pixels initiaux automatiques ou semi-automatiques*
- La région « grandit » à partir de son germe
  - *Besoin d'une critère (ou prédicat) pour choisir les pixels à ajouter*

amorce *croissance* *région finale*

Croissance de région (Region Growing)

Méthode par amorce (complexité O(n))

On définit une zone R qui contient la région à extraire.
Initialement, ***R contient l'amorce***.
On utilise une file FIFO (First In, First Out) *S* qui contient les points frontière
Initialement, ***S contient le voisinage de l'amorce.***

Test local (pixels voisins) ou statistique global (calcul sur la couleur moyenne de R)

```
On retire p dans S
    si p est homogène avec R,
        on ajoute p à R et on ajoute à S les points du
        voisinage de p qui ne sont pas dans R et qui ne sont
        pas incompatibles.
    sinon,
        on marque p comme incompatible.
On recommence tant que S n'est pas vide.
```

Croissance de région (Region Growing)

AVANTAGES

- Méthode rapide
- Conceptuellement très simple

INCONVENIENTS

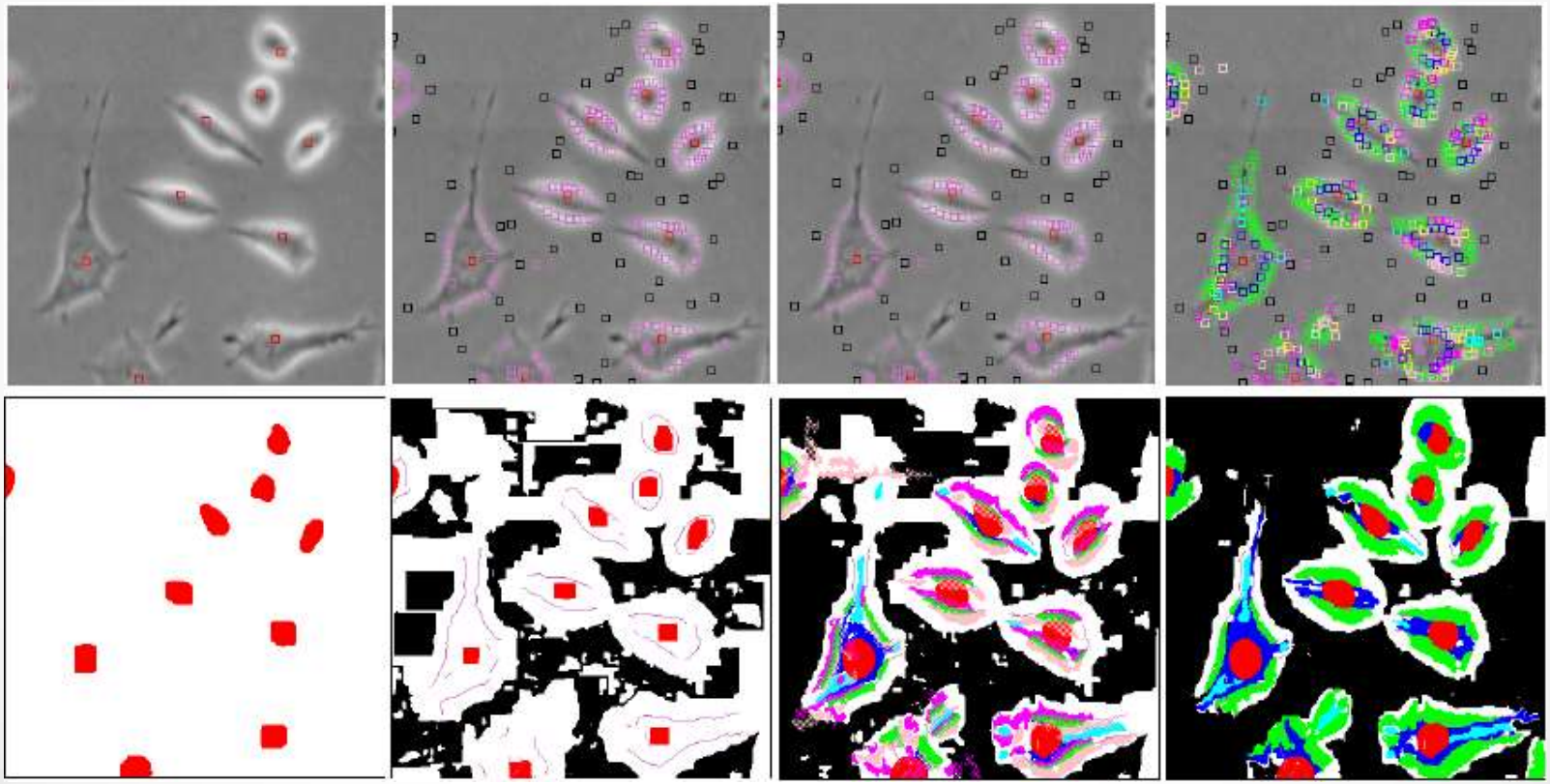
- Méthode locale: aucune vision globale du problème. En pratique, il y a presque toujours un chemin continu de points connexes de couleur proche qui relie deux points d'une image…

Problème du gradient:

- Tenir compte de l'homogénéité globale donne un algorithme sensible à l'ordre de parcours des points (méthode par amorce)
- Algorithme très sensible au bruit, peu stable.

Croissance de région (Region Growing)

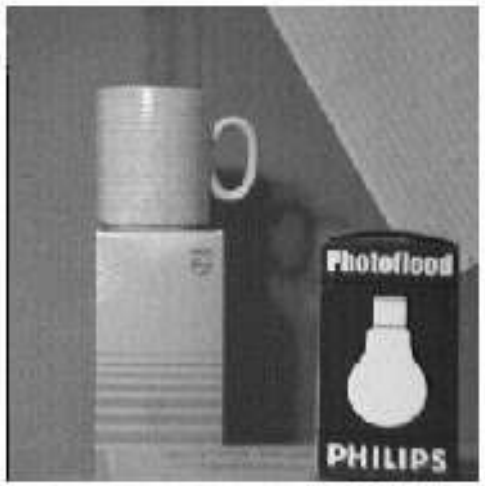
*** Le cas de plusieurs germes.**



Segmentation par division (Split)

Principe

– Définition d'un critère d'homogénéité
– Test de la validité du critère sur l'image
– Si le critère est valide, l'image est segmentée [arrêt de la méthode]
– Sinon, l'image est découpée en zones plus petites et la méthode est réappliquée sur chacune des zones

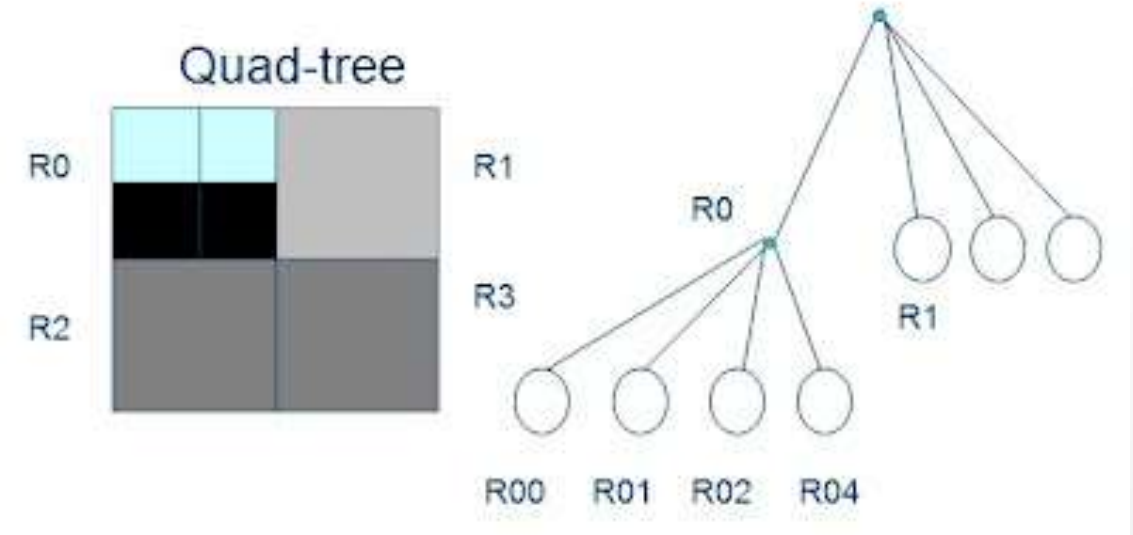


Segmentation par division (Split)

- L'approche opposée à la croissance des régions est la division des régions.
- Il s'agit d'une approche descendante qui part de l'hypothèse que l'image entière est homogène.
- Si ce n'est pas vrai, l'image est divisée en quatre sous-images
- Cette procédure de division est répétée de manière récursive jusqu'à ce que nous divisons l'image en régions homogènes.
- Si l'image originale est un carré N x N, dont les dimensions sont des puissances de 2 (N = 2n)
- Toutes les régions produites à l'exception de l'algorithme de division sont des carrés ayant des dimensions M x M, où M est également une puissance de 2.
- La procédure étant récursive, elle produit une représentation d'image qui peut être décrite par un arbre dont les nœuds ont chacun quatre fils.
- Un tel arbre est appelé arbre Quad.
- L'inconvénient est qu'ils créent des régions qui peuvent être adjacentes et homogènes, mais non fusionnées.

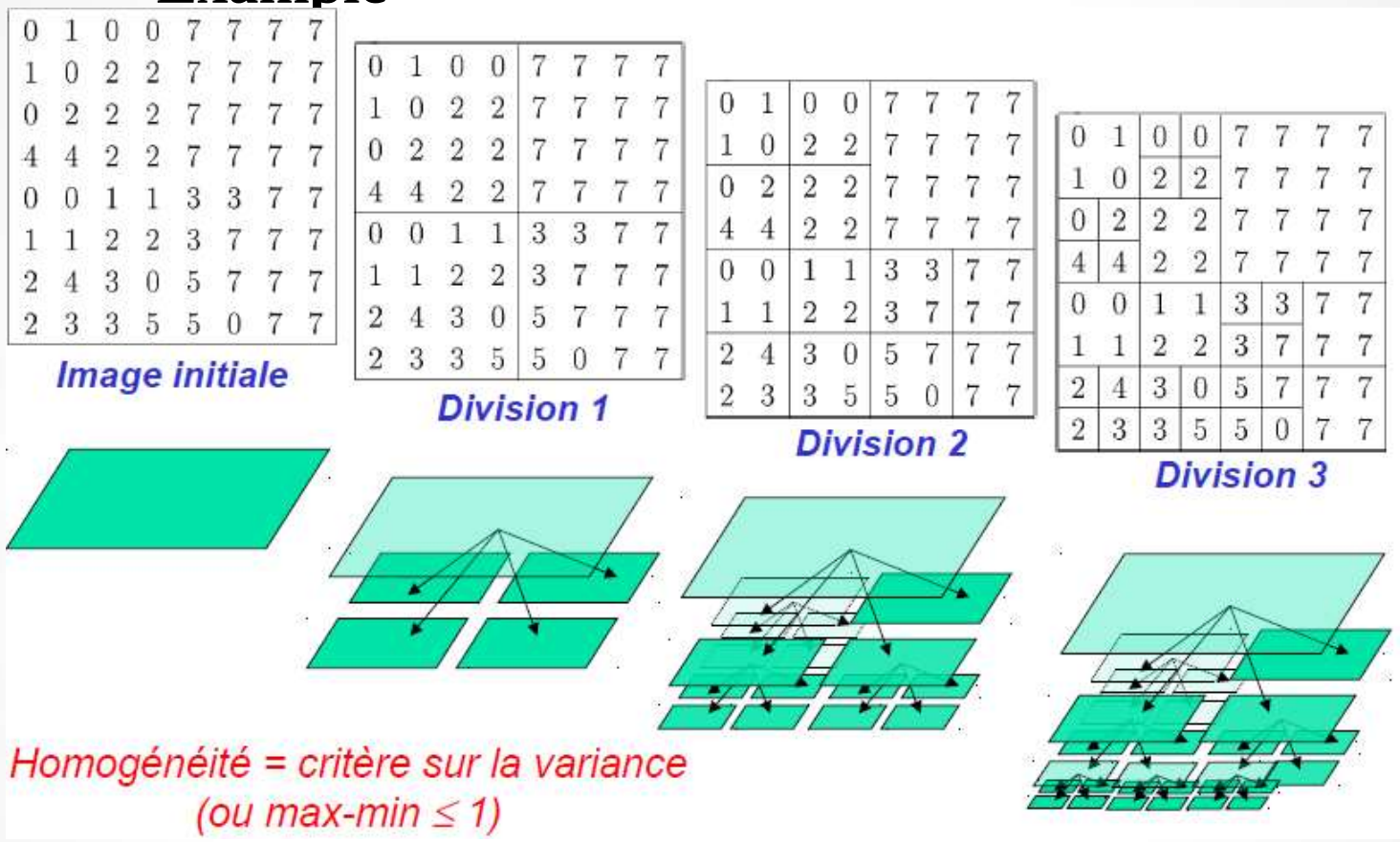
Segmentation par division (Split)

- L'image est stockée dans un arbre
  - Au début, arbre racine = image complète
  - Récursivement, chaque feuille F est
- subdivisée en 4 si elle n'est pas assez homogène
  - les 4 sous-images sont ajoutées en tant que feuilles de F
- L'algorithme poursuit tant qu'il reste des feuilles non homogènes à diviser.



Segmentation par division (Split)

Example



Limitation

– Division en 4 régions -> images initiales carrées et 2n

– Effets de blocs

– Sur segmentation -> fusions de régions

Segmentation par fusion (Merge)

Principe

Faire croître des régions en y ajoutant des pixels adjacents qui satisfont un critère d'homogénéité ou de regroupement.

– Sélection des germes des régions (1 pixel)

– Construction des régions, ajout de pixels connexes

– Tous les pixels traités = Fin de croissance

Différentes approchent

Algo Itératif

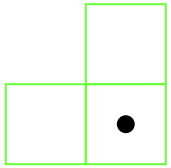
Algo Récursif

Algo Ligne de partage des eaux

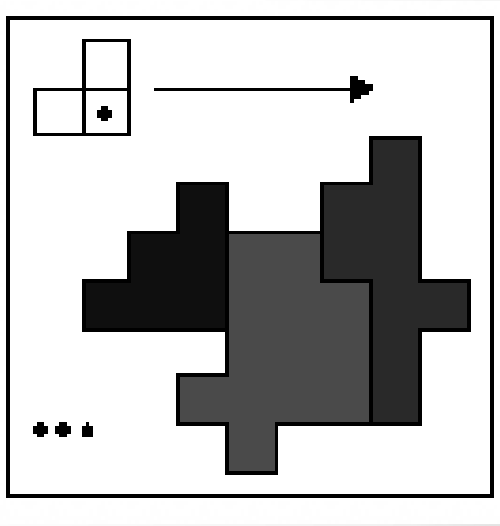
Segmentation par fusion (Merge)

Méthode locale itérative : coloration de taches [blob coloring]

Soit la fenêtre suivante :



Chaque couleur représente un index qui caractérise les régions, déterminées selon l'algorithme suivant « diapo suivant»



Inconvénients :

– Segmentation uniligne.

– Parcours déterminé *a priori*

Segmentation par fusion (Merge)

Algorithme de coloration de tâches

```
Pour chaque pixel I(i,j) Faire
Si Critère (i,j) = Critère (i-1,j)
Alors Couleur (i,j)← Couleur (i-1,j)
Sinon Si Critère (i,j) = Critère (i,j-1)
Alors Couleur (i,j) ← Couleur (i,j-1)
Sinon Couleur (i,j) ← NouvelleCouleur
Si (Critère (i,j) = Critère (i-1,j))
    ET (Critère (i,j) = Critère (i,j-1))
Alors Fusionner les régions en donnant la même
couleur aux
pixels I(i,j), I(i-1,j) et I(i,j-1)
FinPour
```

Segmentation par fusion (Merge)

Méthode locale récursive

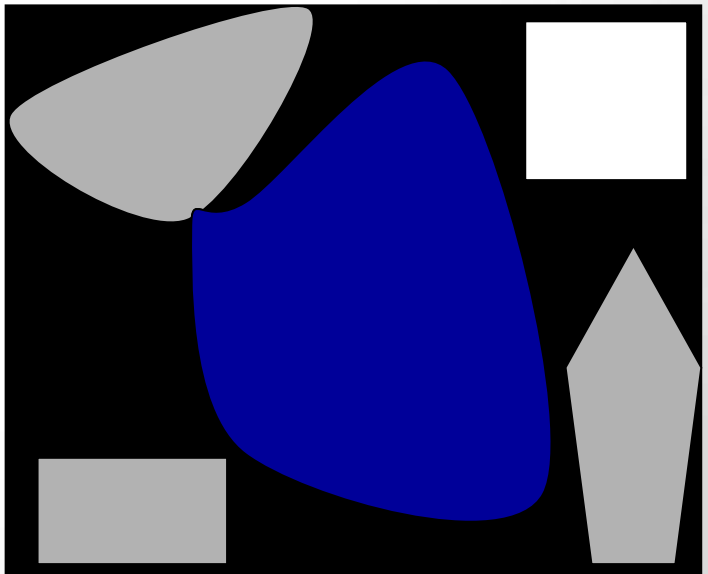
– Principe : on fait croître une région avant de passer à la suivante, sans parcours particulier déterminé a priori (méthode par agrégation libre de pixels)

- Germe [seed]
- Croissance suivant un critère de similarité
- Critère d'arrêt : convexité maximum, etc.

Inconvénients

– Méthode récursive risques de débordements (pile)

– Influence de la position initiale du germe



Segmentation par fusion (Merge)

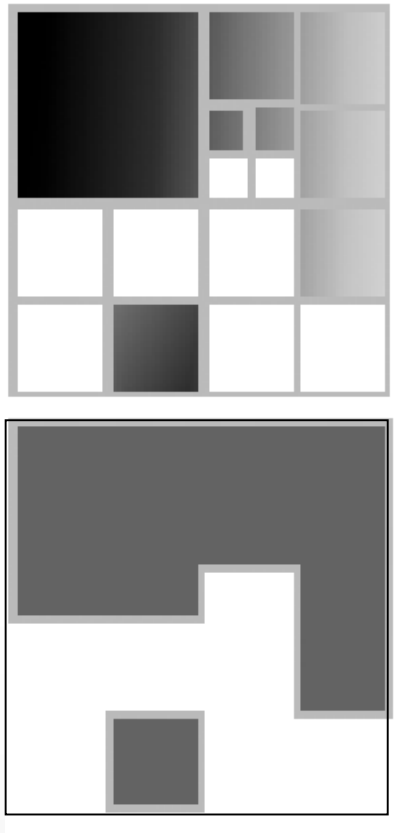
Algorithme de la méthode locale récursive

```
Pour chaque pixel I(i,j) Faire
Si I(i,j) n'a pas déjà été traité
Alors Sauvegarder (i,j), Croissance (i,j),
Incrémenter Région
FinPour
Croissance (i,j)
Pour tout Pixel(k,l) adjacent à I(i,j)
%Pour tous les 8-pixels
Si (Pixel(k,l) pas déjà traité)
   ET (Critère (Pixel(k,l)) = Critère (I(i,j))
Alors Croissance (k,l)
FinPour
```

Split & Merge

Idée: Plutôt que de regrouper des pixels dans le region growing, pourquoi ne pas regrouper des zones homogènes pré-calculées sur l'image?

- ***Initialisation***
  - l'image initiale entière forme un bloc
- ***Etape de division = split***
  - Diviser récursivement tout bloc non-homogène selon un prédicat défini (variance, max-min, …)
  - La division d'un bloc donne 4 sous-blocs
  - Les attributs de chaque sous-bloc sont recalculés
- ***Etape de fusion= merge***
  - Regrouper les blocs adjacents représentant des régions homogènes selon un prédicat défini
- ***Post-traitement***
  - Lissage pour éliminer la forme carré des régions



Split & Merge

- Si une région R est inhomogène (P(R)= False) alors elle est divisée en quatre sous-régions
- Si deux régions adjacentes Ri,Rj sont homogènes (P(Ri U Rj) = TRUE), elles sont fusionnées. L'algorithme s'arrête lorsqu'aucune division ou fusion supplémentaire n'est possible.
- L'algorithme de fractionnement et de fusion produit des régions plus compactes que l'algorithme de fractionnement pur.

Split & Merge

SPLIT

Homogénéité = critère sur la variance

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Image initiale

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

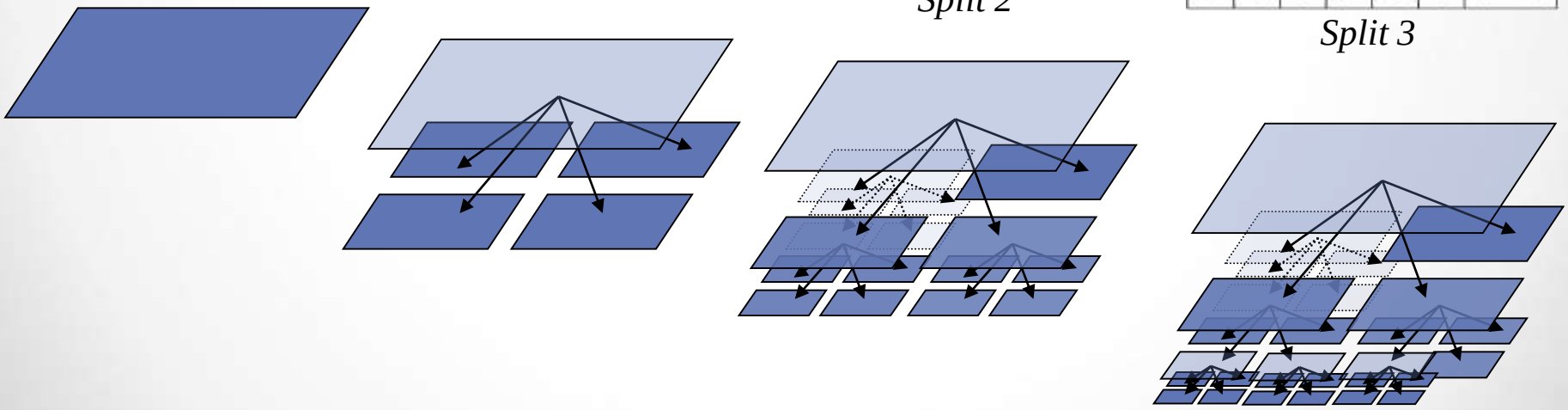
Split 1

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 2

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 3

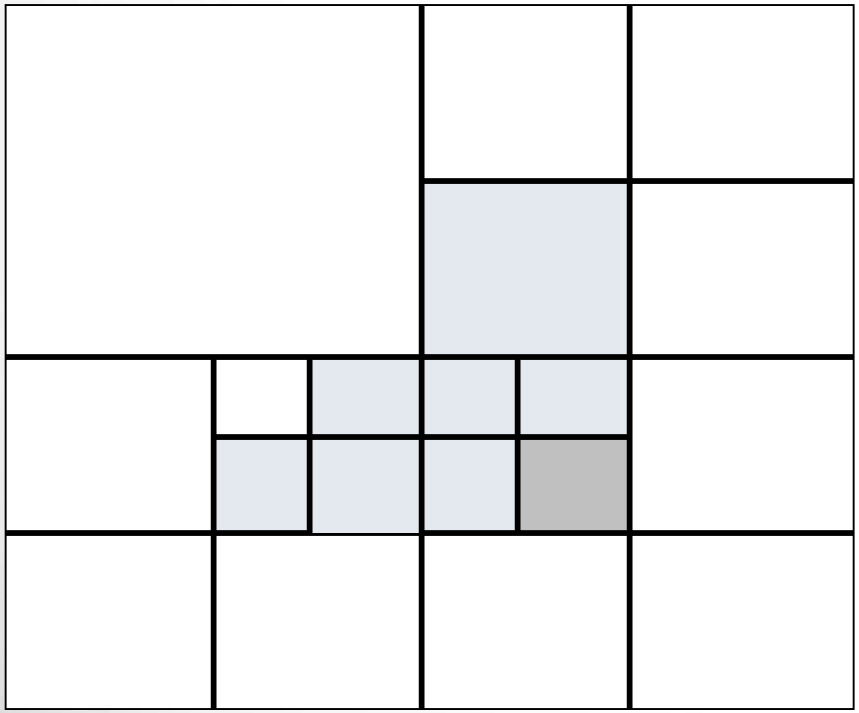


Split & Merge

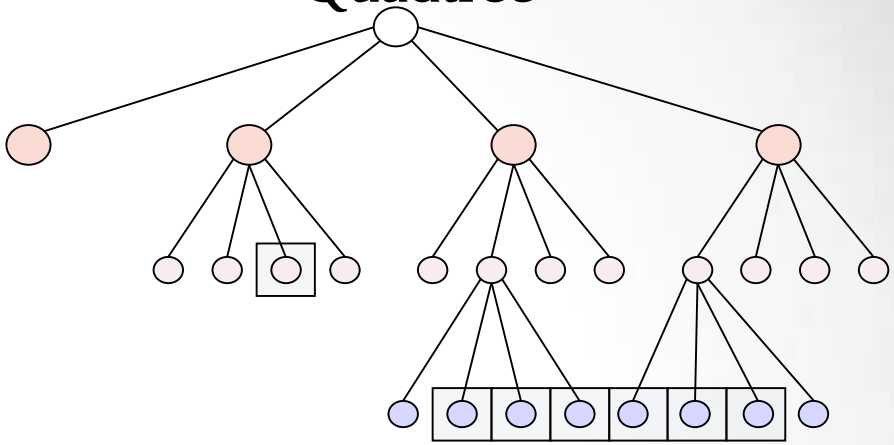
Construction du RAG

Region Adjacency Graph

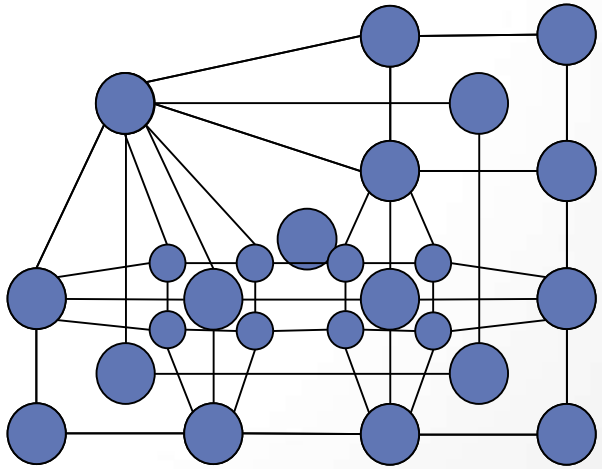
- Connecte les régions adjacentes
- Arrêtes = mesures de différence d'homogénéité



Quadtree



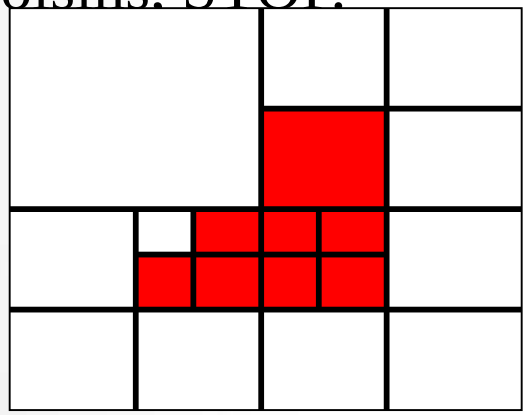
RAG

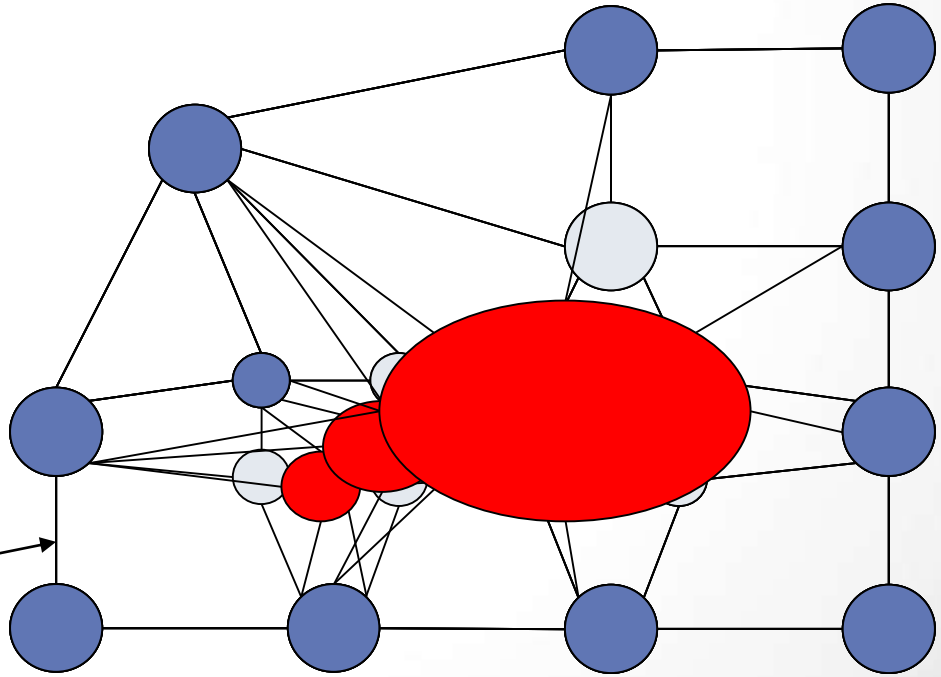


Split & Merge

MERGE:

- Chaque nœud du Region Adjacency Graph est examiné.
- Si un des voisins de ce nœud est à une distance inférieure à un seuil de regroupement, les deux nœuds fusionnent dans le RAG.
- Lorsque plus aucun nœud ne peut fusionner avec l'un de ses voisins. STOP.



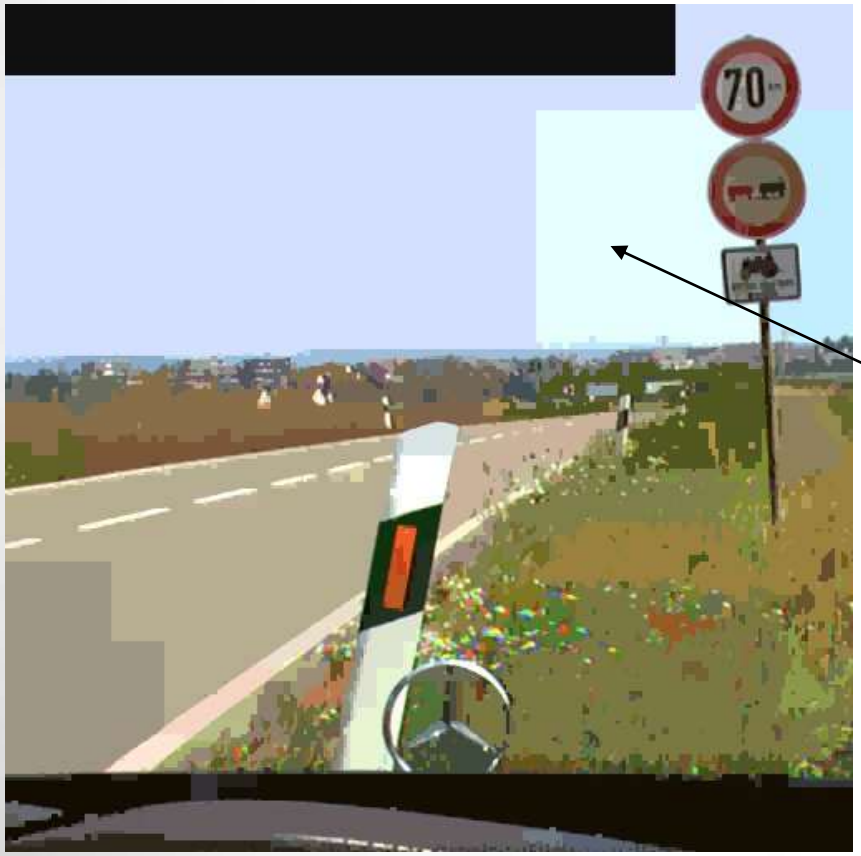


La distance en terme d'homogénéité de régions est portée par l'arrête valuée qui les relie dans le RAG

Split & Merge

Original

Split & Merge



AVANTAGES

- Méthode hybride locale/globale: permet de contrer le problème du gradient.

INCONVENIENTS

- Méthode assez complexe
- Découpage un peu « carré », dû à la topologie des quadtrees

Segmentation par partage des eaux

- On considère la visualisation de l'image en 3D, en utilisant le ton de gris comme troisième dimension

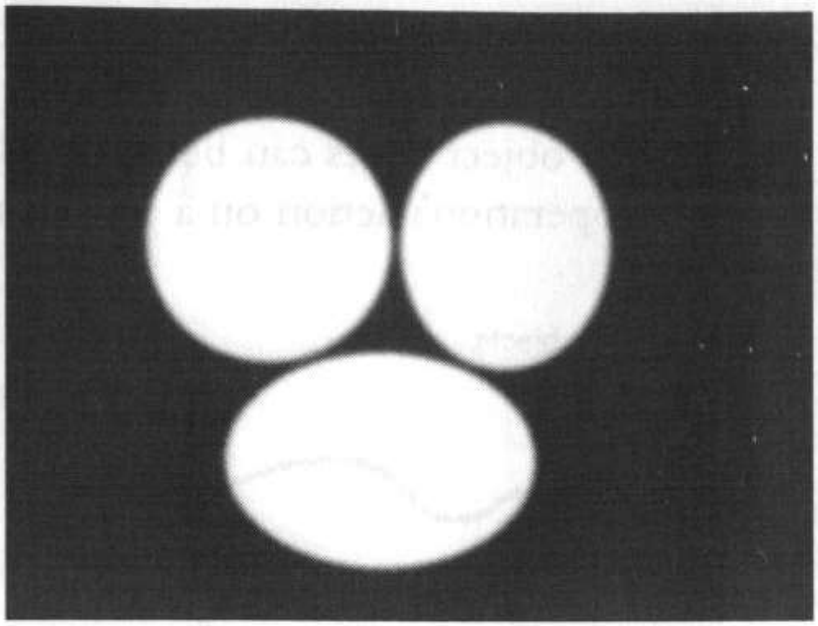
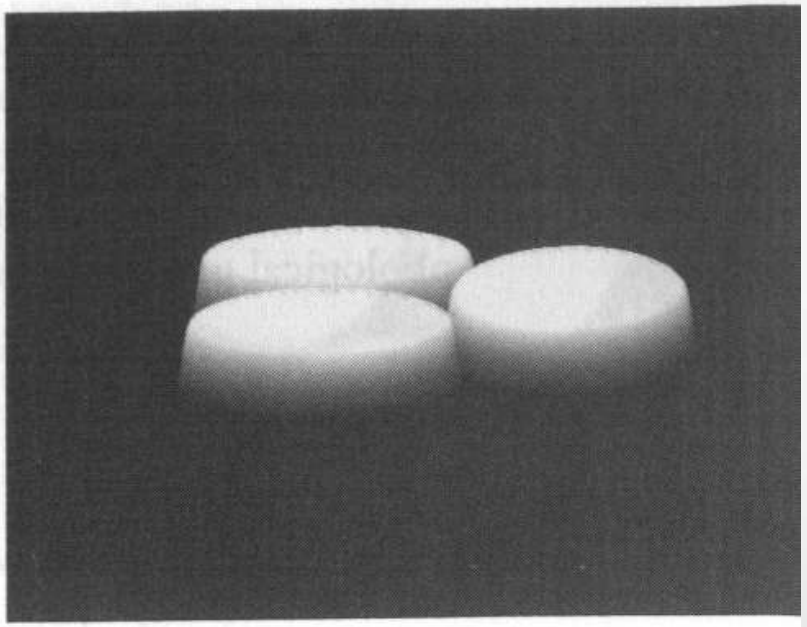


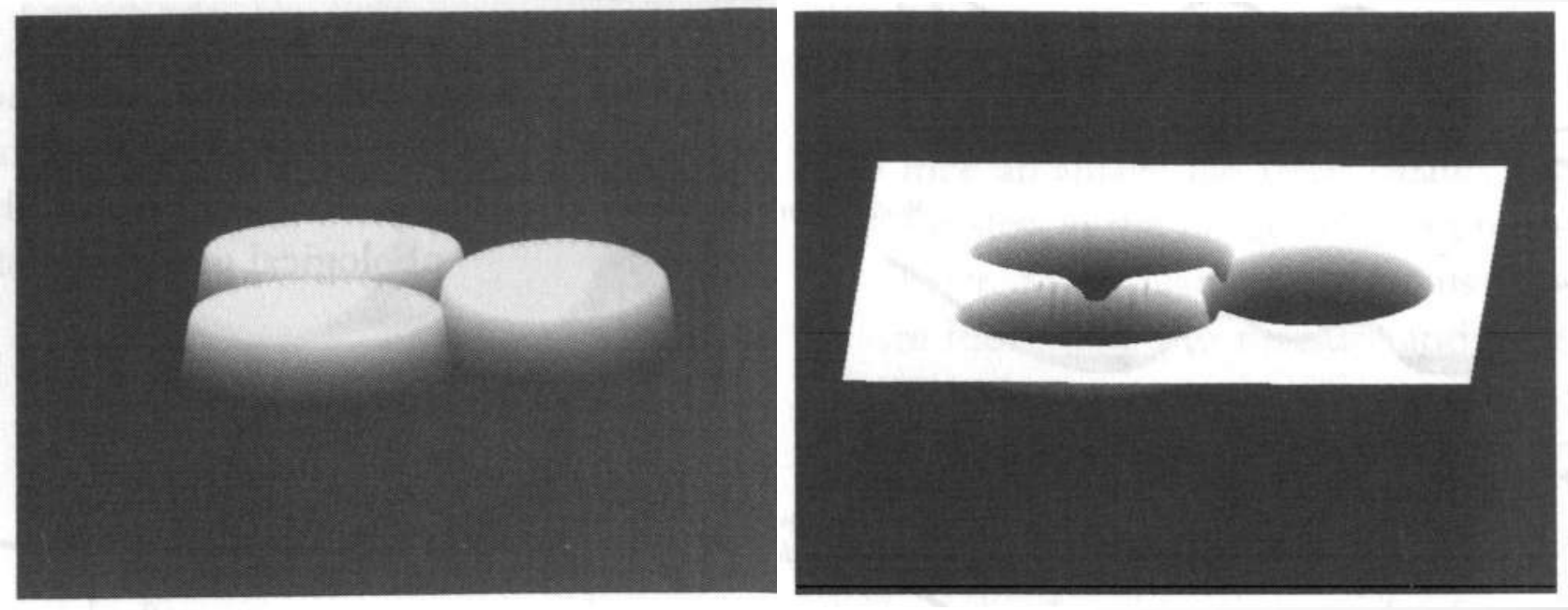
Image 2D



Visualisation en 3D

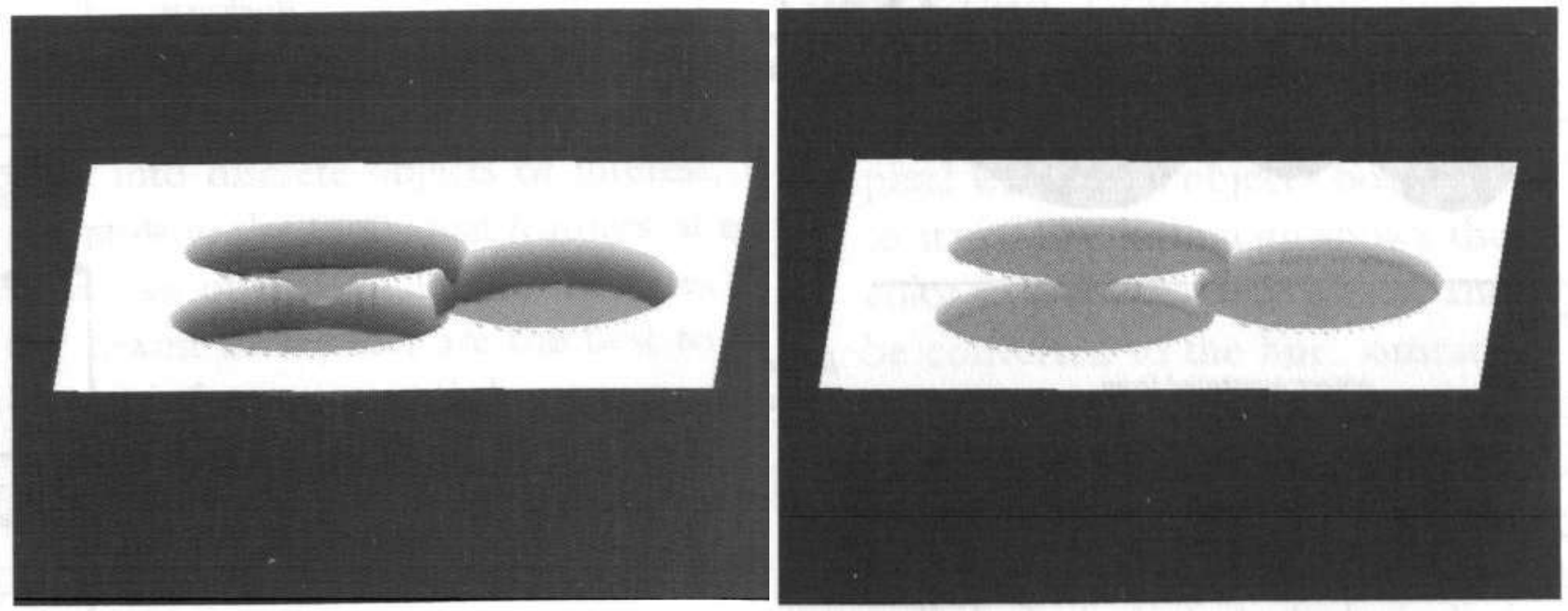
Segmentation par partage des eaux

- Ensuite on "complémente" les valeurs pour créer des zones inondables.

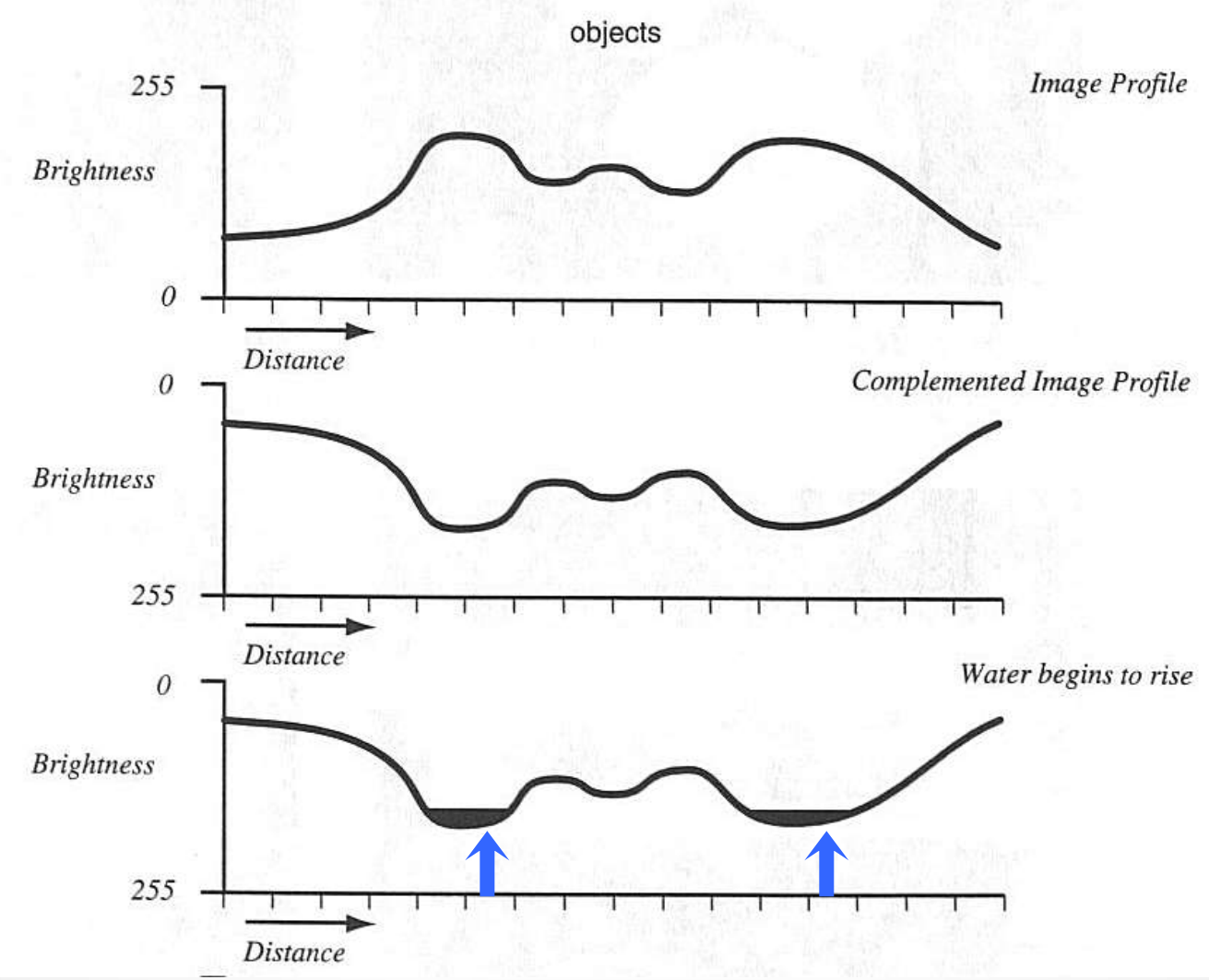


Segmentation par partage des eaux

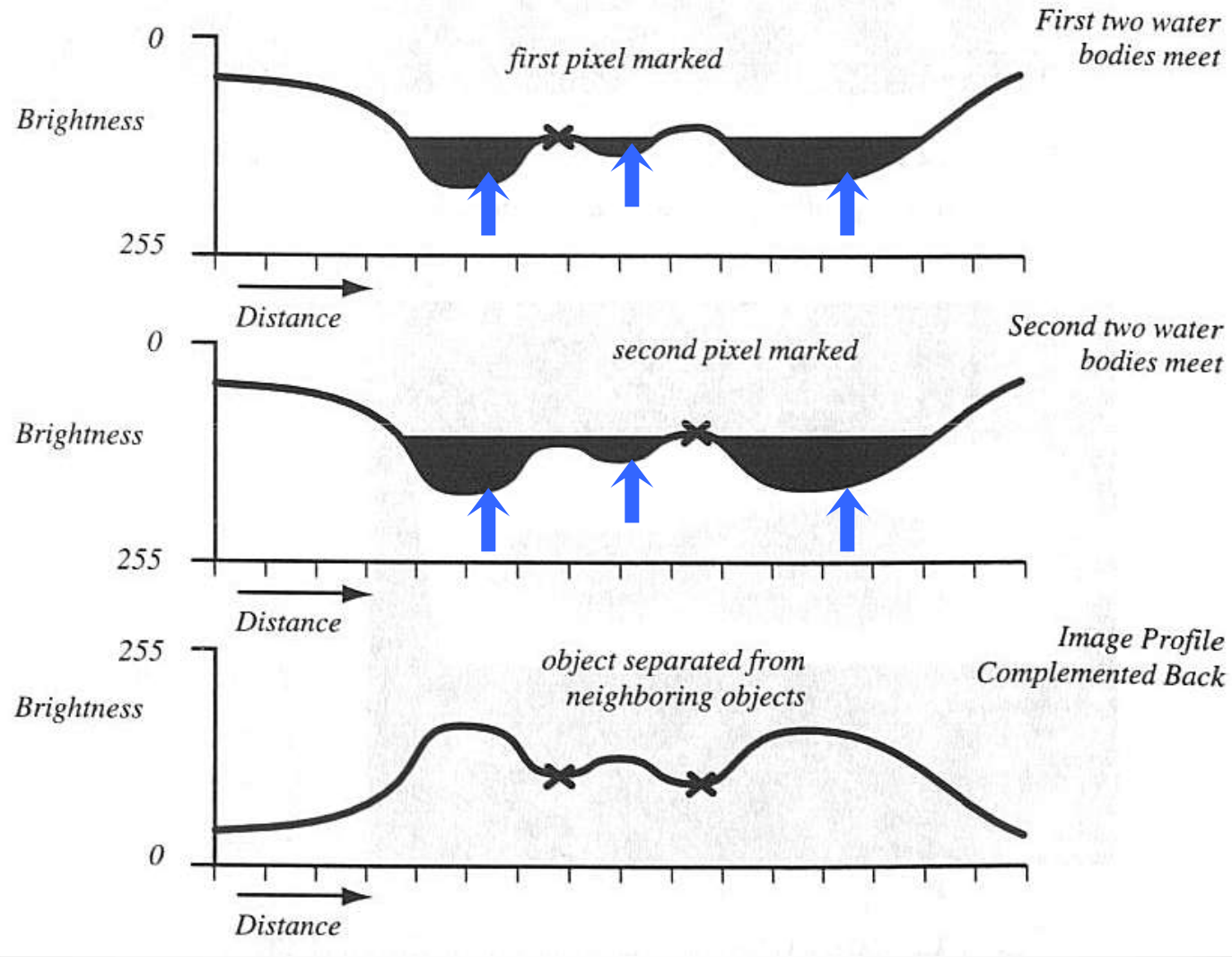
- Enfin, on "infiltre" les cavités des zones inondables



Segmentation par partage des eaux



Segmentation par partage des eaux



Ligne de partage des eaux

Principe

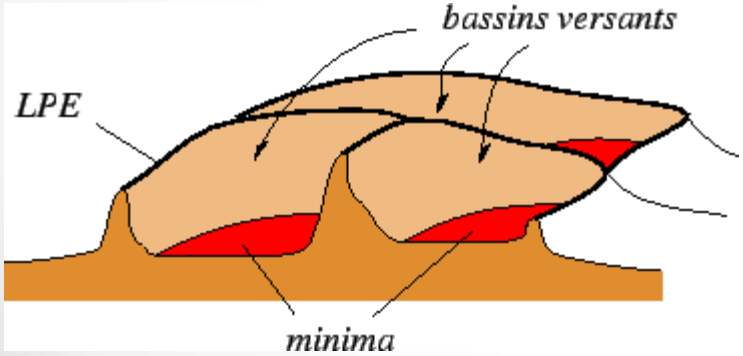
• Source d'eau (minima)

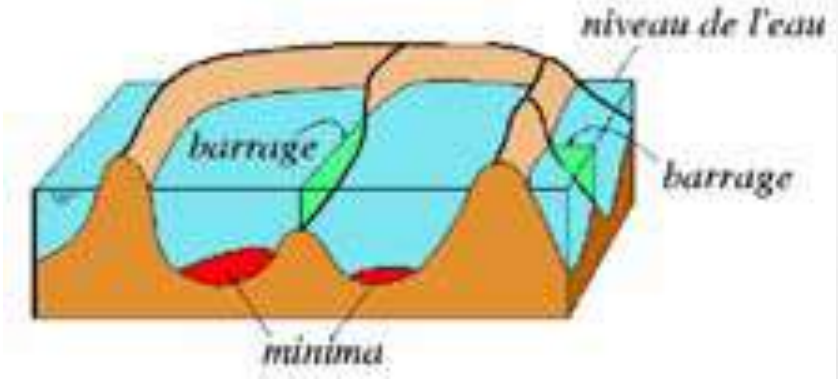
• Montée des eaux

• Barrage élémentaire à chaque point de rencontre des bassins

• Eau monte et on élève les barrages

• Altitude max => étendues d'eau cernées par des barrages (LPE)





Segmentation – conseils

- La segmentation d'une image cause encore aujourd'hui beaucoup de problèmes.
  - Aucune méthode ne fonctionne pour toutes les images.
  - Pas de garantie, pas de recette miracle!

- Pour certaines applications, on réalise qu'on peut éviter la segmentation complète de l'image. C'est souvent mieux.

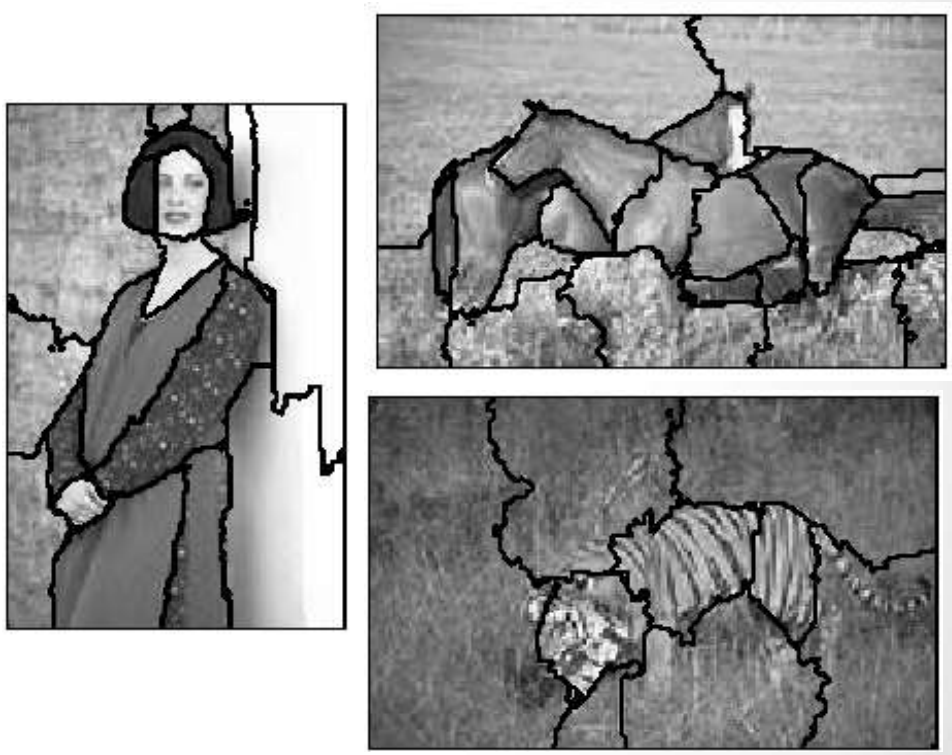
- Le pré-traitement des images, la sélection de capteurs et sources d'énergie appropriées, et la prise contrôlée des images rendent cette étape plus facile et plus efficace.

Segmentation – conseils

- Le problème d'évaluer le résultat d'une segmentation n'est pas évident. Bien souvent, il est subjectif et varie d'une personne à l'autre.
- Un des principaux problème est de définir le but de la segmentation : *Qu'est-ce qu'on recherche exactement dans l'image ?*
  - Eléments globaux de l'image ou détails fin de la composition ?
  - Présence d'un humain ou détails du visage ?
- Il est bon de se poser aussi la question de ce que l'on veut faire ensuite avec la segmentation. Cela permet de définir le degré de précision nécessaire.

Segmentation vs groupement

Aujourd'hui, on délaisse de plus en plus le terme de *segmentation*, qui sous-entend une séparation plus exacte de l'image, pour celui de *groupement de pixels (grouping)*, qui ne fait appel qu'à une notion de similarité de pixels sans aucune relation de contenu.



Extrait de [Malik 2001].

BONUS

Détection / Segmentation
Et deeplearning

tâches principales

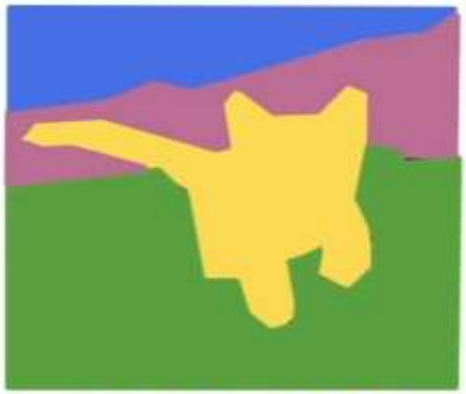
Classification

CAT

Détection

DOG, DOG, CAT

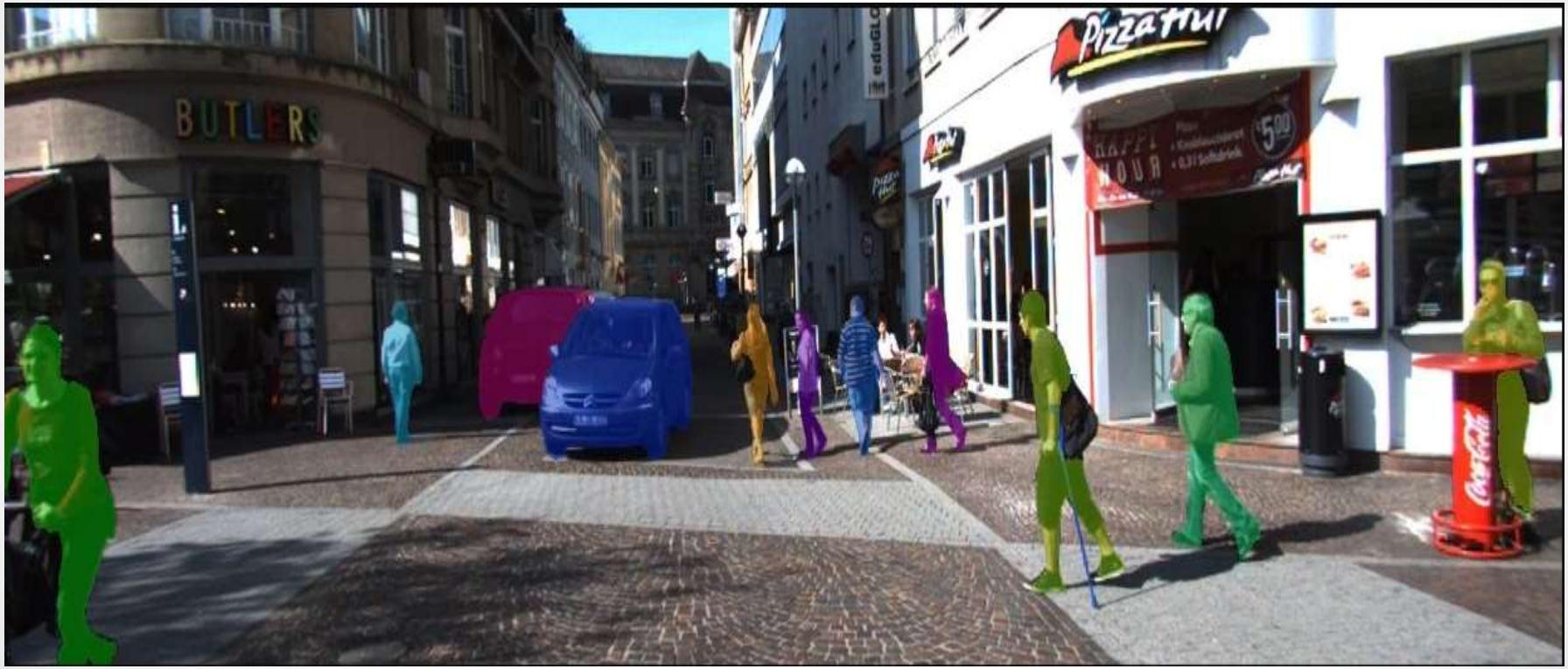
Segmentation



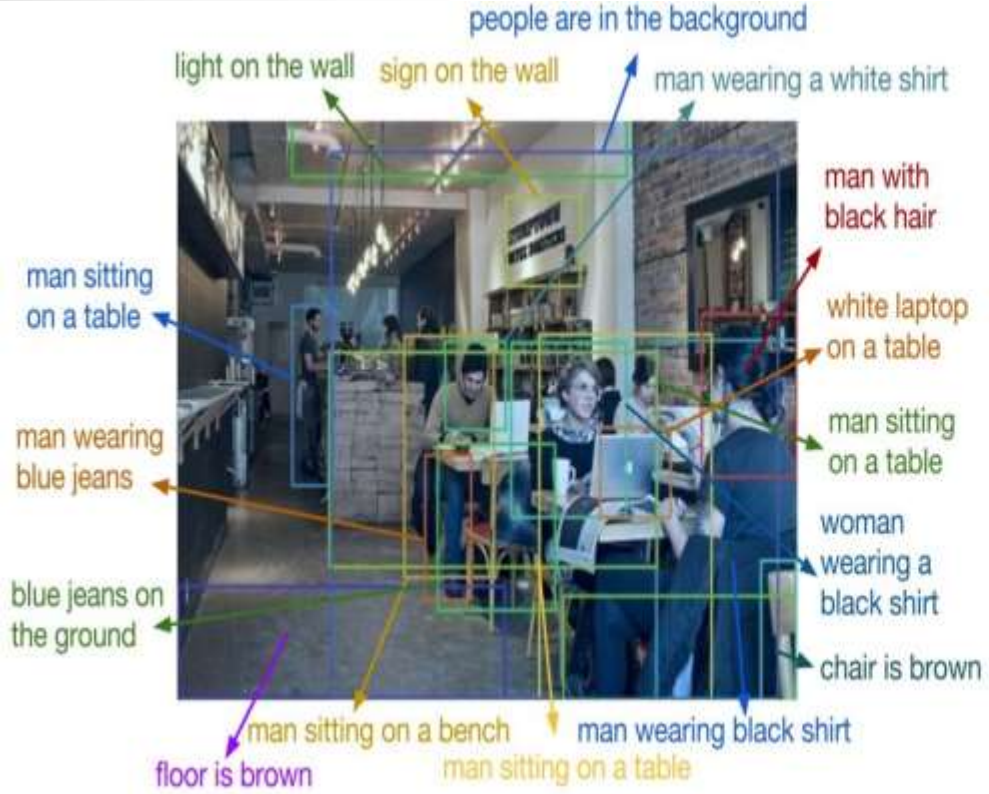
GRASS, CAT,
TREE, SKY

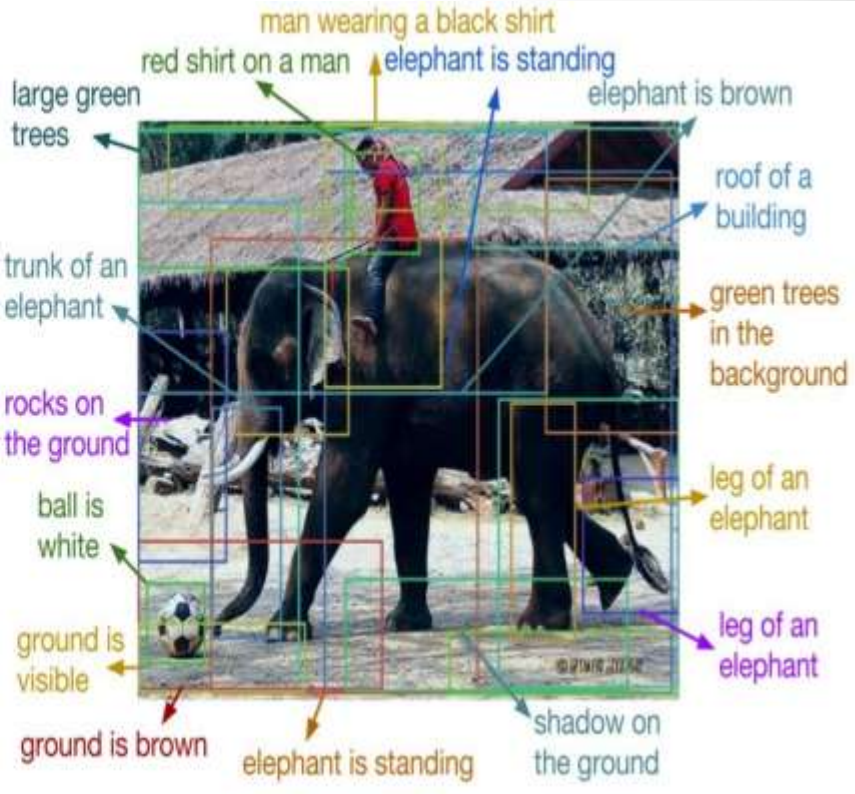
Segmentation
d'instances

DOG, DOG, CAT


BUTLERS
Pizza Hut

Object Detection + Captioning = Dense Captioning

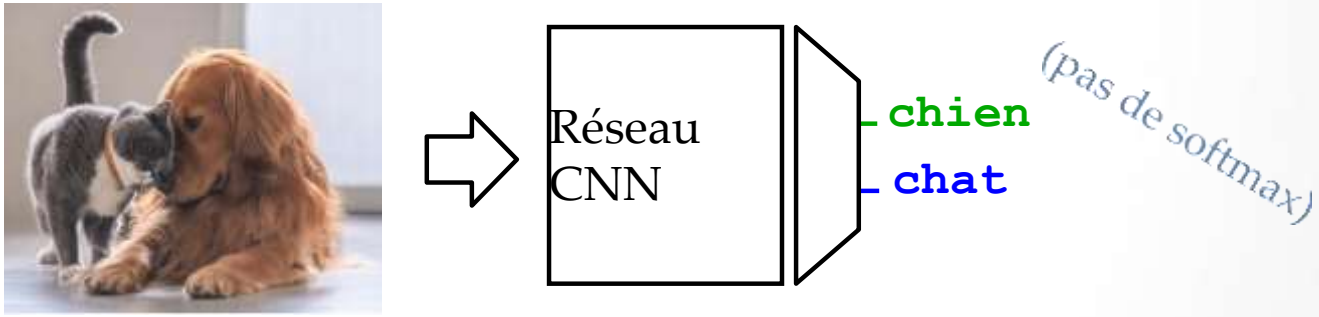




Détection

Détection

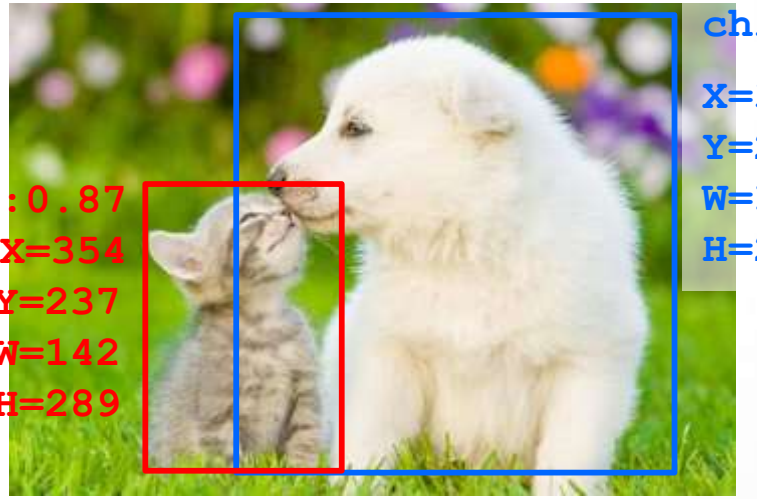
- Une des difficultés est que l'on ne connait pas le **nombre exact d'instances** dans l'image
  - si on savait d'avance que les images ne contiennent au maximum qu'un chat et un chien :



- Sujet de recherche très fertile en soit

Détection : définition

- Pour :
  - **une image d'entrée**
  - **une liste prédéterminée de classes**
- Trouver, pour tous les objets présents :
  - la position (bounding box)
  - la classe



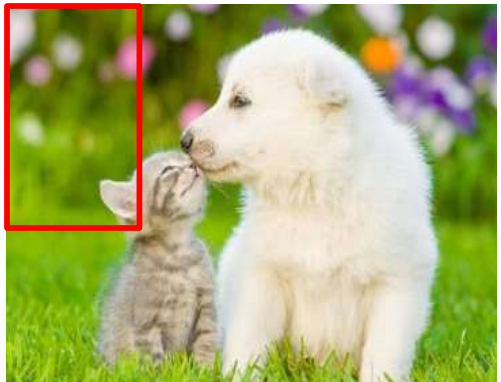
- Nombre de prédictions **va varier d'une image à l'autre**
  - Architecture doit en tenir compte

Détection via classification

- Possible de faire de la détection par un réseau de classification
- Approche par **fenêtre coulissante** (crops)
- Passe chaque crop dans un réseau classificateur
- Conserve *n* prédictions les plus confiantes
- Choix de la géométrie de la fenêtre :

– taille, aspect ratio

- **Fastidieux**, car des milliers de passes dans le réseau classificateur : dizaines de secondes



Catégories d'algorithmes

- **Basé sur des régions proposées** *(region proposal)*
  - R-CNN
  - Fast-RCNN
  - Faster RCNN

- **Grille fixe** *(grid-based)*
  - YOLO (v1, v2, v3)
  - SSD (single-shot detection)

segmentation

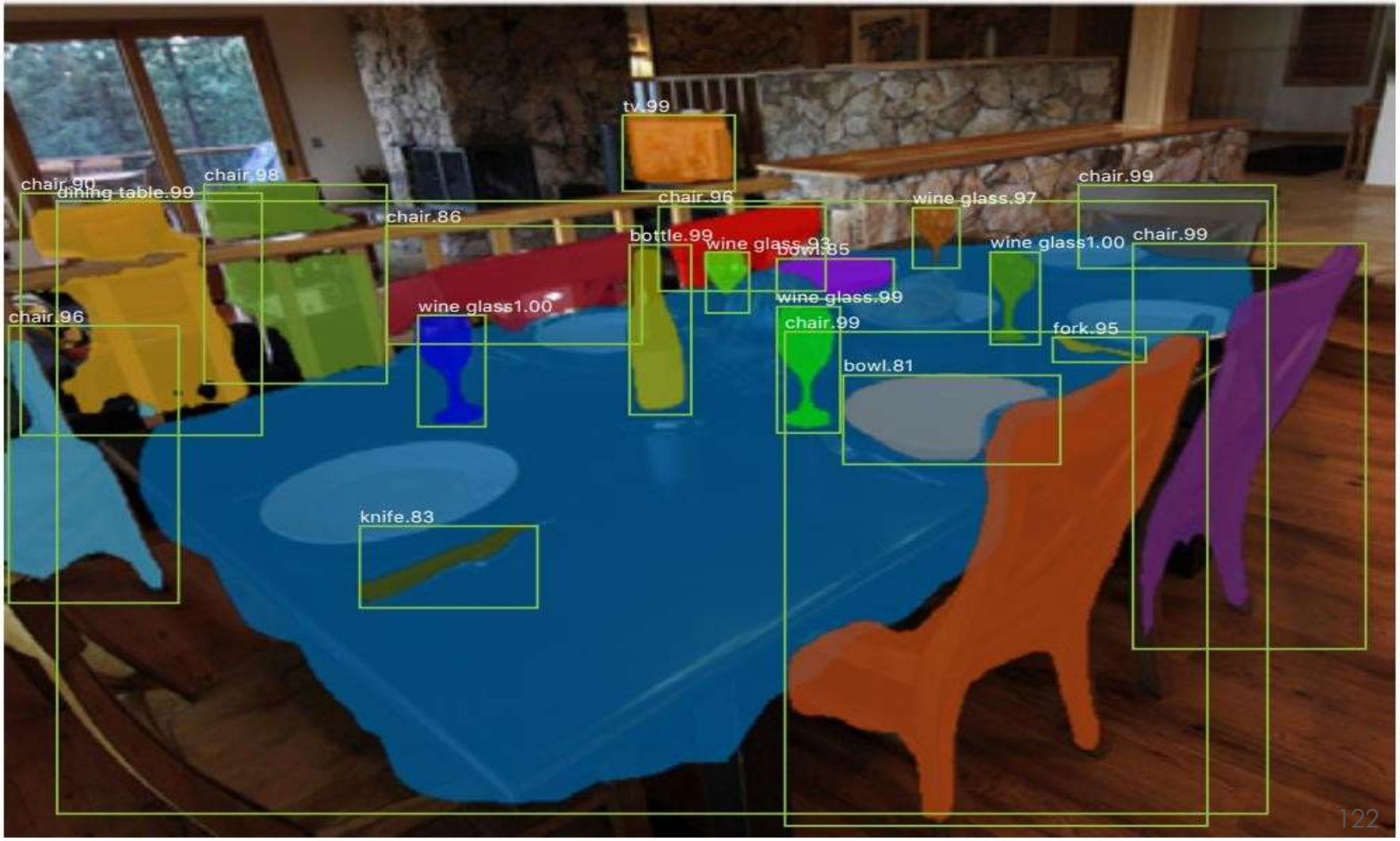
- Here are some of the most well-known deep learning-based algorithms used for segmentation:
  - **U-Net**: A convolutional neural network (CNN) architecture that is widely used for image segmentation tasks. It was originally designed for biomedical image segmentation, but has since been applied to other domains.
  - **Mask R-CNN**: A two-stage CNN-based architecture for object detection and segmentation. It builds on the popular Faster R-CNN framework by adding a segmentation branch.
  - **FCN**: Fully Convolutional Networks (FCN) is a type of neural network architecture that can perform end-to-end image segmentation. It involves replacing the fully connected layers of a CNN with convolutional layers.
  - **SegNet**: A deep encoder-decoder architecture that is designed for semantic segmentation. It uses a series of encoder and decoder layers to learn feature representations at multiple scales.
  - **DeepLab**: A CNN-based semantic image segmentation approach that uses dilated convolutions to capture multi-scale contextual information. It was originally developed for natural scene understanding and has been extended to medical image segmentation.
  - **PSPNet**: Pyramid Scene Parsing Network (PSPNet) is a deep learning-based segmentation model that uses a pyramid pooling module to capture contextual information at multiple scales. It was originally developed for scene parsing tasks, but has also been applied to medical image segmentation.

detection/segmentation

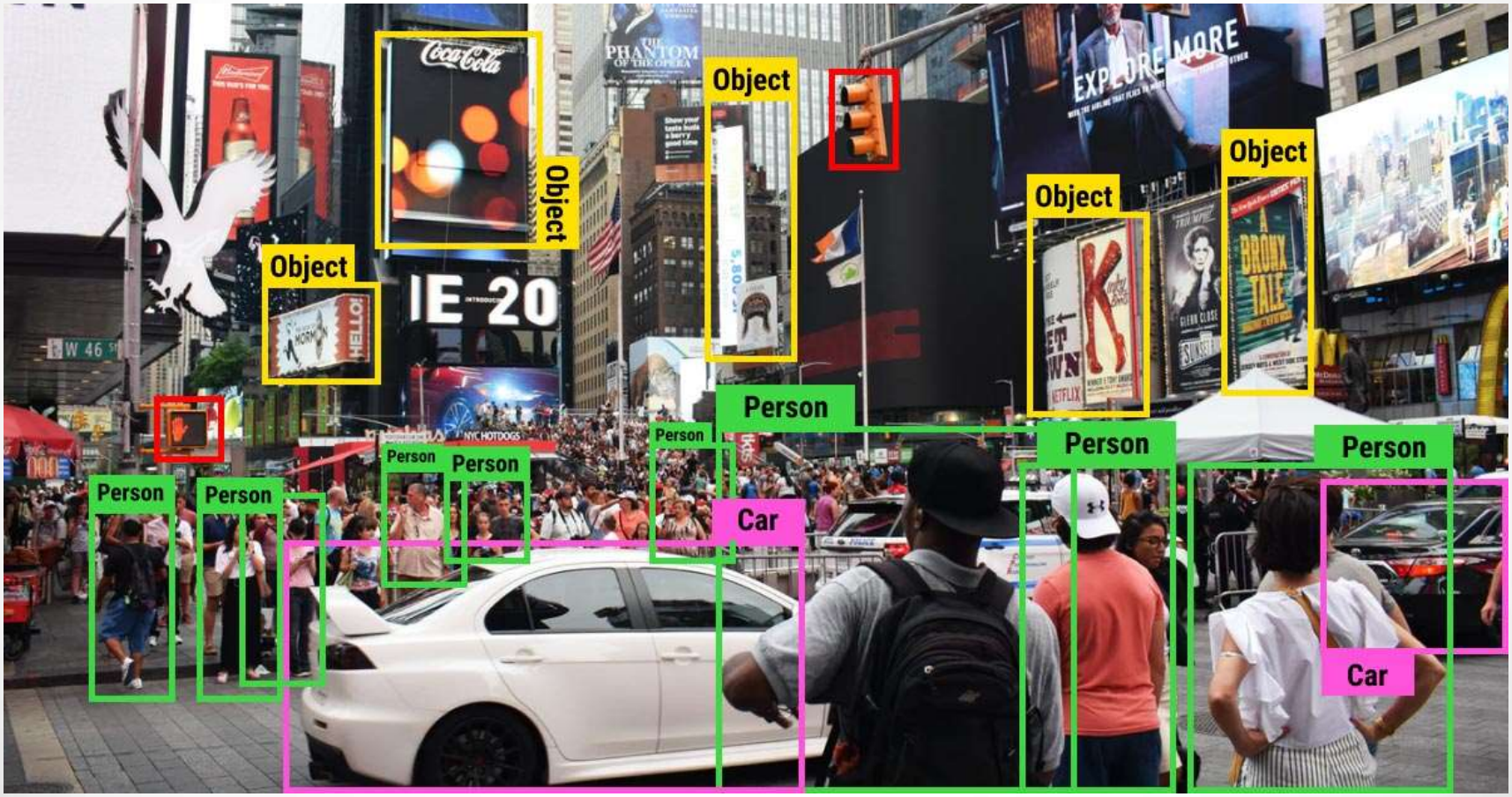
- Here are some of the most well-known deep learning-based algorithms used for detection/segmentation:

  - **RetinaNet**: This one-stage object detection algorithm is designed for high accuracy and efficiency. It uses a novel focal loss function that addresses the class imbalance problem in object detection.
  - **YOLO (You Only Look Once):** This real-time object detection algorithm uses a single CNN to predict both object bounding boxes and class probabilities in one pass.
  - **SSD (Single Shot MultiBox Detector):** This object detection algorithm is similar to YOLO in that it performs object detection in a single pass. It uses a set of default bounding boxes at different scales to detect objects.

Mask R-CNN : exemple



YOLOv8 : exemple



Exemple de segmentation

- Données de *Cityscape*