

Chapitre 5 : Segmentation (Partie 2)

Joël Lefebvre (UQÀM)

INF600F – Traitement d'images

Automne 2024

Annonces

- **Atelier** : Segmentation
- **À venir**
 - **Présentation Mercredis-Recherche** (URL à venir)
 - **TP3** (S9 à S11)
 - **TP4** (S11 à S13)
 - **Examen final** (S15, 11 décembre 2024)
 - **Évaluation des enseignements**

Survol du cours

- Introduction
- Détection de contours
- Détection de courbes simples
- **Segmentation de régions**

Références

- (Chityala, 2020) Chapitre 8 : *Segmentation*
- (Gonzalez, 2018) Chapitre 10 : *Image Segmentation I*
- (Szeliski, 2021) Chapitre 7.5 : *Segmentation*
- (Burger, 2009) Vol3, Ch2 : *Automatic Thresholding*

Segmentation de régions

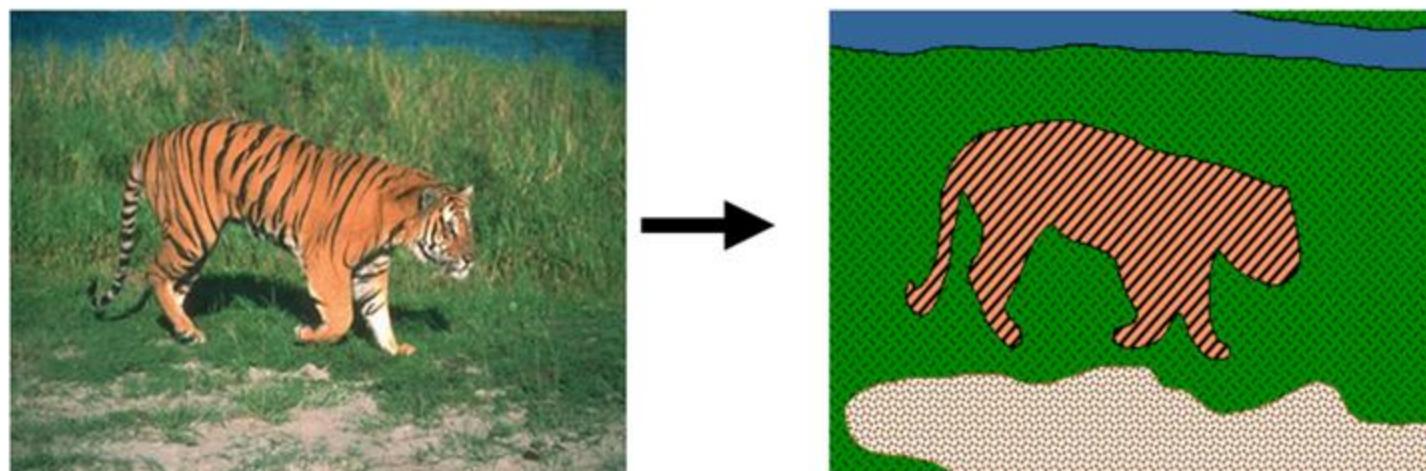
Chapitre 5 : Segmentation (Partie 2)

Joël Lefebvre (UQÀM)

INF600F – Traitement d'images

Rappel : Qu'est-ce que la segmentation ?

- Regroupement de pixels en objets
- « Organisation perceptuelle »
- Chaque région doit satisfaire une propriété commune
- Deux régions voisines ne doivent pas satisfaire la même propriété



Plusieurs méthodes de segmentation

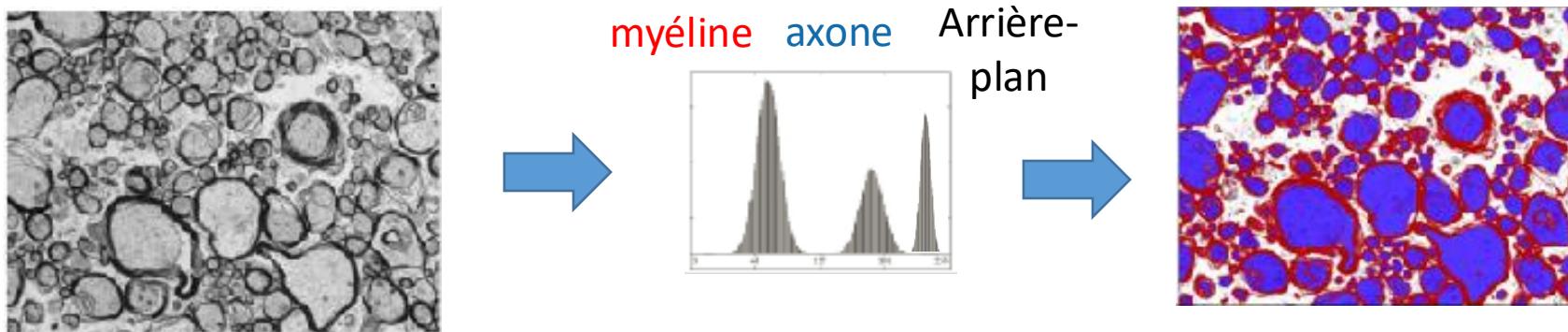
- P1 : Segmentation des **bords (edge)**
- P1 : Segmentation basée sur les équations différentielles
- P1 : Segmentation des **contours**
- P2 : Segmentation basée sur l'**histogramme**
- P2 : Segmentation basée sur les **régions**
- Segmentation par **partitionnement de graphe**
- Segmentation basée sur des **modèles**
- Méthodes de **partitionnement**

Segmentation de régions

- **Objectif** : Partitionnement de l'image
- **Comment faire ?**
- Chaque région est caractérisée par :
 - Son intensité moyenne
 - Petites variations dans chaque région (ex. bruit)
- **Idée** : Utiliser **l'histogramme** de l'image pour trouver et appliquer un **seuillage global**

Approche empirique basée sur l'histogramme

- Choix des seuils basé sur **l'inspection de l'histogramme**
- **Exemple** : Segmentation de neurones à partir d'image histologique



- **Problème** : difficile de généraliser à des images légèrement différentes (autres systèmes d'acquisition, autres paramètres, autres préparations du tissu, ...)

Région d'image binarisée : Seuillage

- Une technique **existant depuis longtemps**
- Encore très utilisée, pour la **simplicité** et l'efficacité d'implémentation
- **Étape critique** : Conversion d'une image en niveaux de gris (ou en couleur) en une image binaire
- La plupart du temps, ceci est réalisé par une opération de **seuillage** de l'intensité

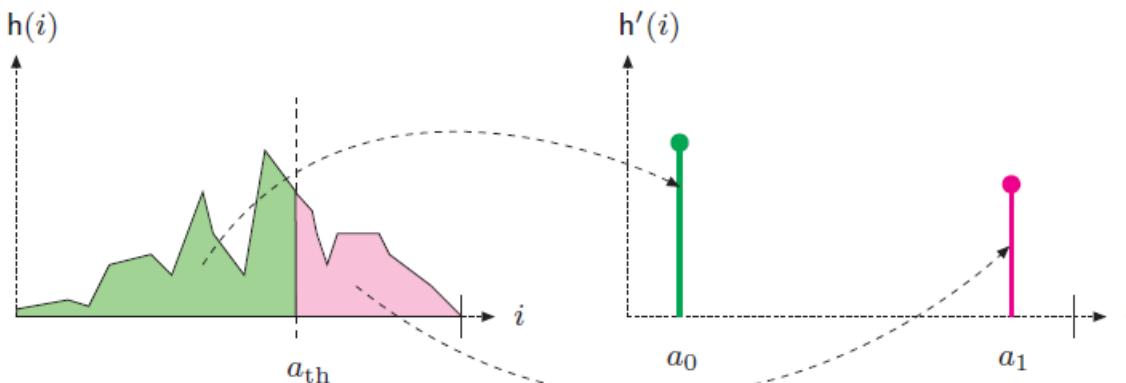
Rappel : Seuillage (*Thresholding*)

- Type de **quantification** de l'intensité
- Appliqué **pixel par pixel** (Opération ponctuelle)

$$f_{seuil}(a) = \begin{cases} a_0 & \text{pour } a < a_{th} \\ a_1 & \text{pour } a \geq a_{th} \end{cases}$$

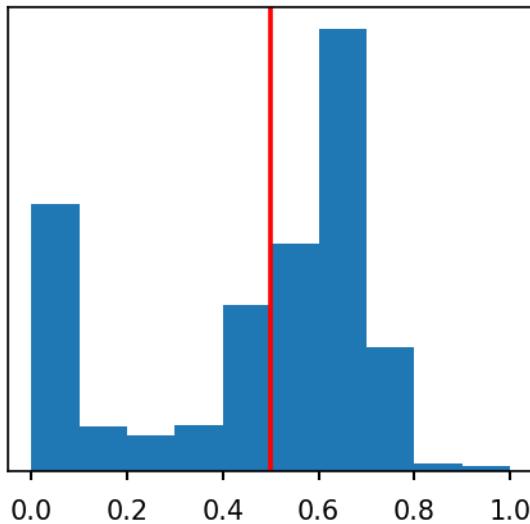
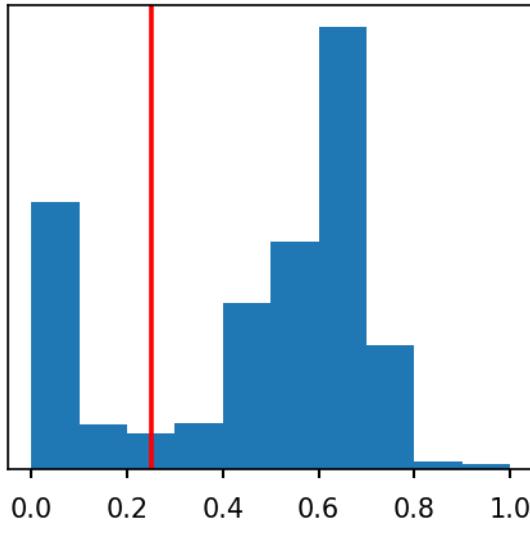
pour $0 < a_{th} < a_{\max}$ et a_{th} est un seuil d'intensité (*threshold*)

- **Exemple :** *Binarisation* d'une image en utilisant $a_0 = 0$ et $a_1 = 1$

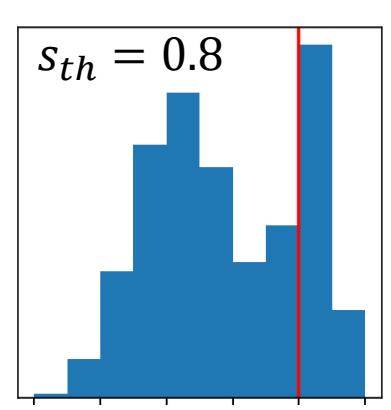
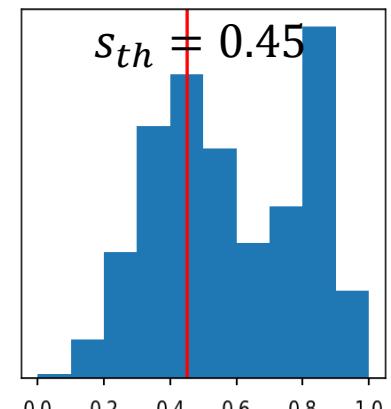
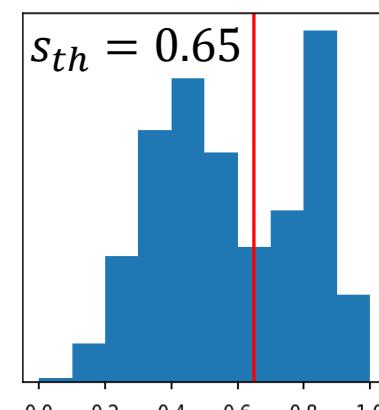
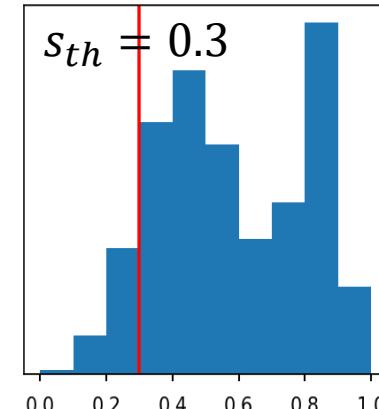
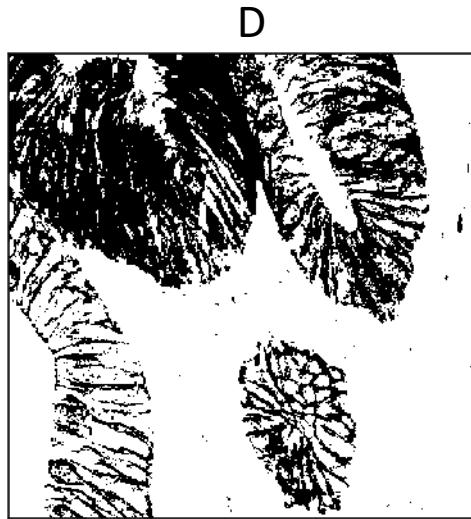
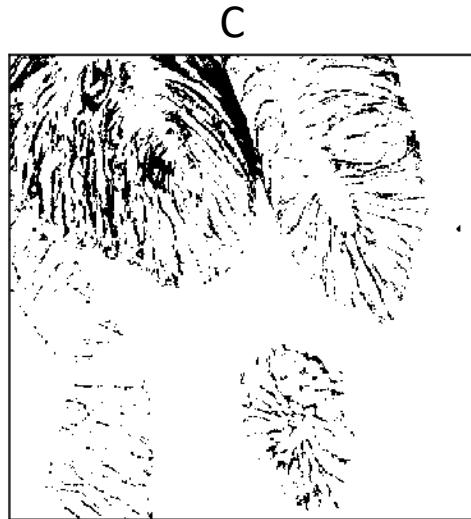
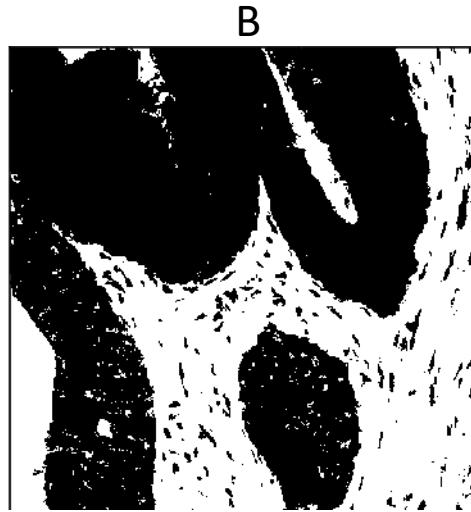


(Burger, Vol1, Fig4.2)

Exemple : Seuillage manuel

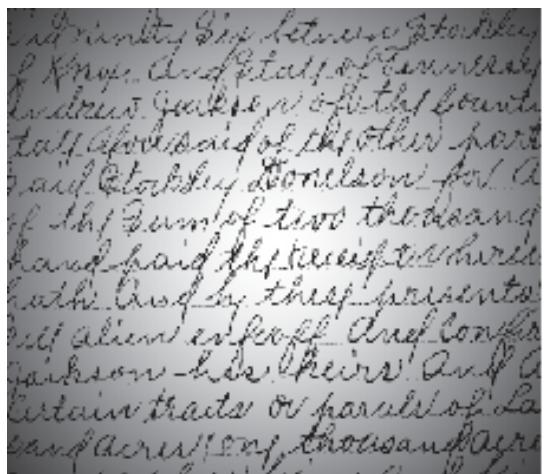


Exemple : Associer seuil et segmentation

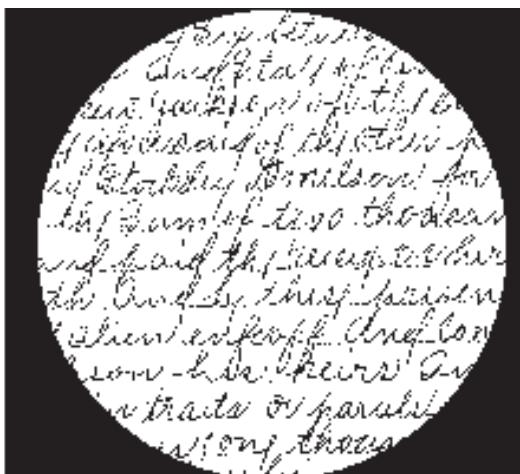


Seuillage automatique : Contexte

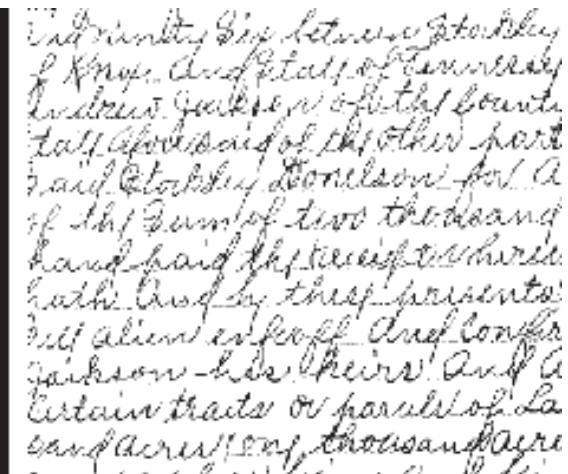
- **Exemple :** Conversion d'un texte numérisé en image binaire pouvant être lue
- La qualité du résultat est très sensible au choix de seuil
- **Comment choisir automatiquement ce seuil ?**
- **2 approches :** globale et adaptative



Texte à segmenter



Seuillage global



Seuillage adaptatif

(Gonzalez & Woods, Fig10.44)

Seuillage automatique : Historique

- Domaine de recherche traditionnel (*et encore actif*) en traitement d'images
- Plusieurs techniques développées au cours des années 1980 et 1990
- Otsu, N. (1979). A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9, 62-66. (>32000 citations)
- C'est une technique de segmentation parmi d'autres



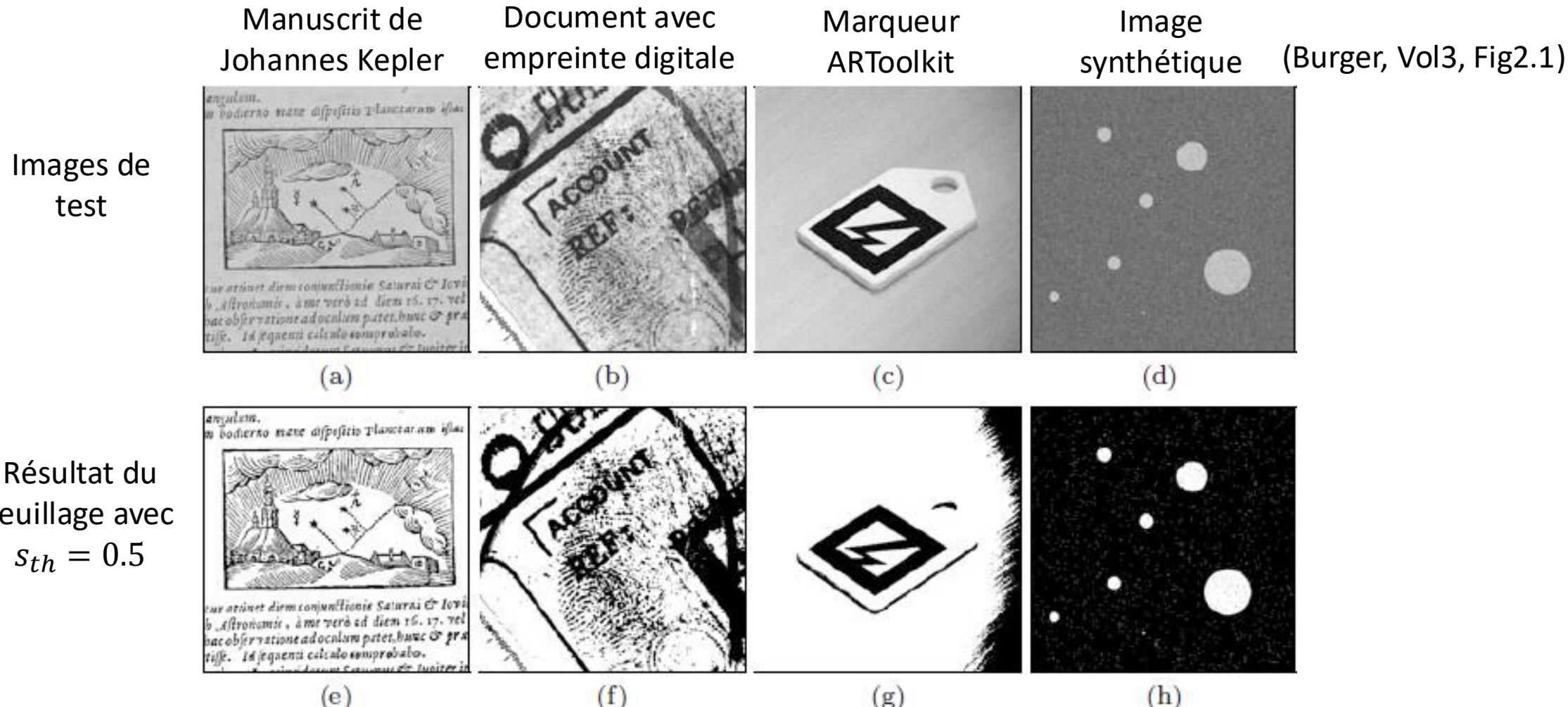
Seuillage global basé sur l'histogramme

- Soit une **image I** pour laquelle un **seuil d'intensité « *optimal* »** doit être trouvé pour la **binariser**
- Appliquer un seuil d'intensité q est équivalent à **classifier** chaque pixel en 2 catégories : ***arrière-plan* (*background*) et *avant-plan* (*foreground*)**

$$(u, v) \in \begin{cases} \mathcal{C}_0 & \text{si } I(u, v) \leq q \text{ (arrière-plan)} \\ \mathcal{C}_1 & \text{si } I(u, v) > q \text{ (avant-plan)} \end{cases}$$

- La **signification** de l'avant-plan et de l'arrière-plan dépend de l'application donnée

Exemple d'images à segmenter



Le BG/FG dépend de l'application.

Approche globale basée sur l'histogramme (1)

(Burger, Vol3, Fig2.2)



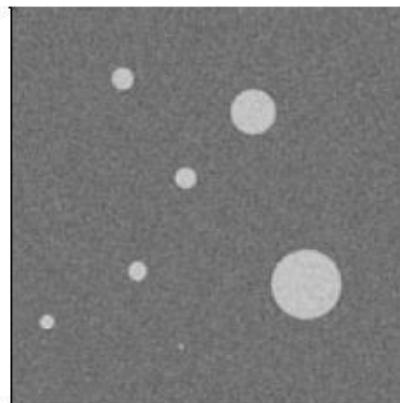
(a)



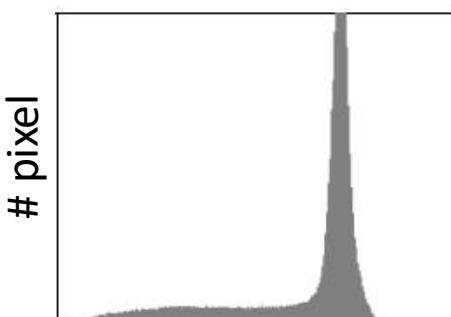
(b)



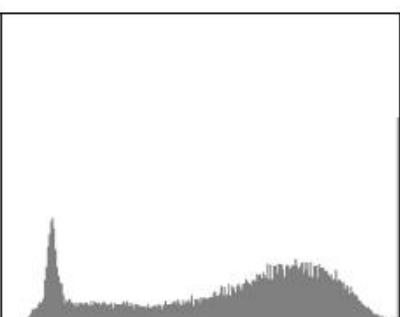
(c)



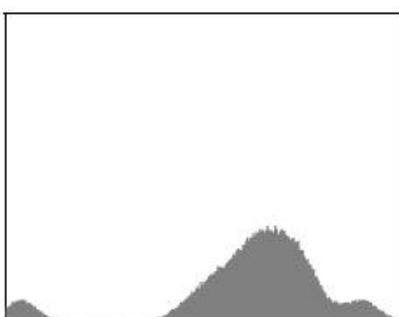
(d)



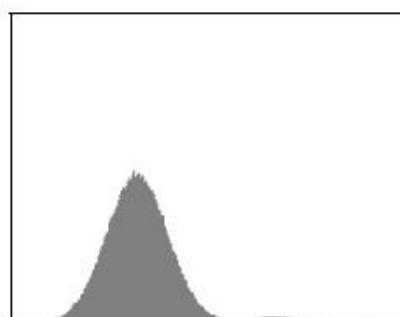
(e)



(f)



(g)



(h)

Intensité

Intensité

Intensité

Intensité

Approche globale basée sur l'histogramme (2)

- Utilise principalement l'information contenue dans l'histogramme, **sans observer l'image pixel par pixel**
- Méthode **simple et efficace**
 - Petit ensemble de données
 - **Exemple** : 256 valeurs seulement pour une image 8bit
- 2 groupes de techniques
 - Basé sur la **forme** de l'histogramme
 - Basé sur les **statistiques** de l'histogramme

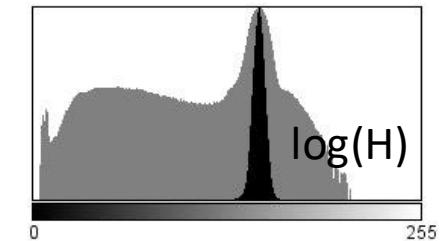
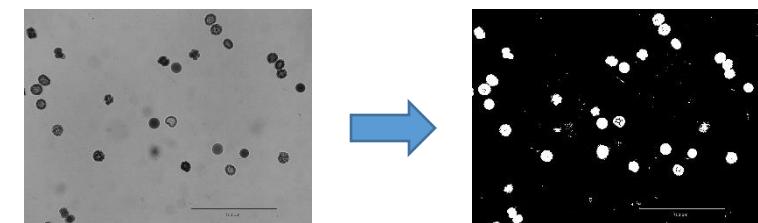
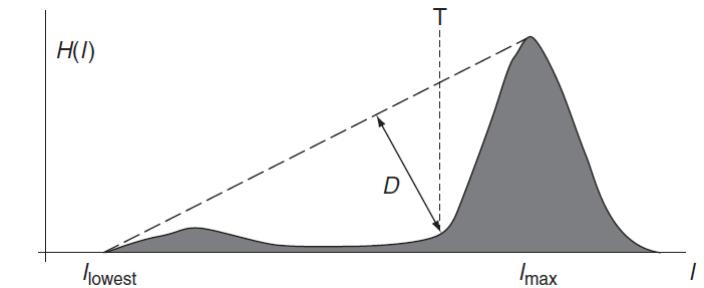
Méthodes basées sur la forme de l'histogramme

- Analyse la **structure** de la distribution de l'histogramme
- Repère des **pics**, des **vallées**, et autres **motifs**
- Généralement, l'histogramme est d'abord **lissé** pour retirer les pics étroits et les trous
- Premières méthodes utilisées, elles sont **moins robustes** que les méthodes statistiques
- **Exemple classique : Méthode du triangle**

Méthode du triangle (*Seuillage global*)

Efficace lorsque les pixels de l'objet produisent un **faible pic dans l'histogramme**.

1. Détection du pic maximum de l'histogramme
2. Traçage d'une ligne liant le pic maximum et le pic minimum.
3. Calcul de la distance entre les éléments de l'histogramme et la ligne
4. Le seuil s_{th} est choisi comme étant l'intensité associée à la plus grande distance



Statistiques de l'histogramme (1)

- Soit l'**histogramme** $h(g)$ d'une image I avec un total de N pixels et K valeurs possibles $0 \leq g < K$
- La **moyenne** μ_I de l'intensité de tous les pixels de I

$$\mu_I = \frac{1}{N} \cdot \sum_{x,y} I(x,y) = \frac{1}{N} \cdot \sum_{g=0}^{K-1} g \cdot h(g)$$

- La **variance** σ_I^2 de l'intensité de tous les pixels de I

$$\sigma_I^2 = \frac{1}{N} \cdot \sum_{x,y} [I(x,y) - \mu_I]^2 = \frac{1}{N} \cdot \sum_{g=0}^{K-1} (g - \mu_I)^2 \cdot h(g)$$

Statistiques de l'histogramme (2)

- Si on applique un **seuil** q à l'image, on obtient 2 sous-ensembles de pixels \mathcal{C}_0 et \mathcal{C}_1 correspondant à l'arrière-plan et l'avant-plan.
- **Nombre de pixels** assignés à chaque sous-ensemble

$$n_0(q) = |\mathcal{C}_0| = \sum_{g=0}^q h(g)$$

$$n_1(q) = |\mathcal{C}_1| = \sum_{g=q+1}^{K-1} h(g)$$

avec $n_0(q) + n_1(q) = |\mathcal{C}_0 \cup \mathcal{C}_1| = N$

Statistiques de l'histogramme (3)

- **Intensité moyenne par sous-ensemble**

$$\mu_0(q) = \frac{1}{n_0(q)} \cdot \sum_{g=0}^q g \cdot h(g)$$

$$\mu_1(q) = \frac{1}{n_1(q)} \cdot \sum_{g=q+1}^{K-1} g \cdot h(g)$$

Statistiques de l'histogramme (4)

- **Variance de l'intensité par sous-ensemble**

$$\sigma_0^2(q) = \frac{1}{n_0(q)} \cdot \sum_{g=0}^q (g - \mu_0(q))^2 \cdot h(g)$$

$$\sigma_1^2(q) = \frac{1}{n_1(q)} \cdot \sum_{g=q+1}^{K-1} (g - \mu_1(q))^2 \cdot h(g)$$

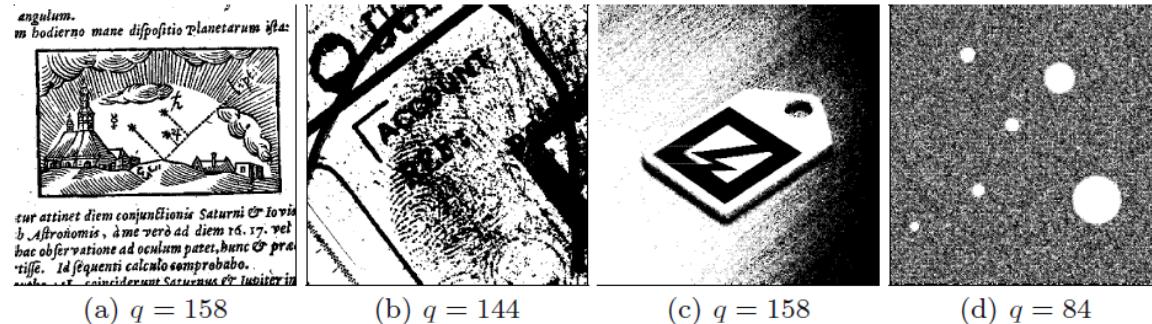
Méthodes de seuillage automatique simple basé sur les statistiques de l'histogramme

- **Moyenne** : $q = \text{moyenne}(I) = \mu_I$
- **Médiane** : Même quantité de pixels avant/arrière-plan. $q = \text{médiane}(I)$
- **Mi-étendue** : $q = \text{round} \left(\frac{\max(I) + \min(I)}{2} \right)$
- **Quantile** : On connaît à priori la proportion b de pixels avant/arrière-plan

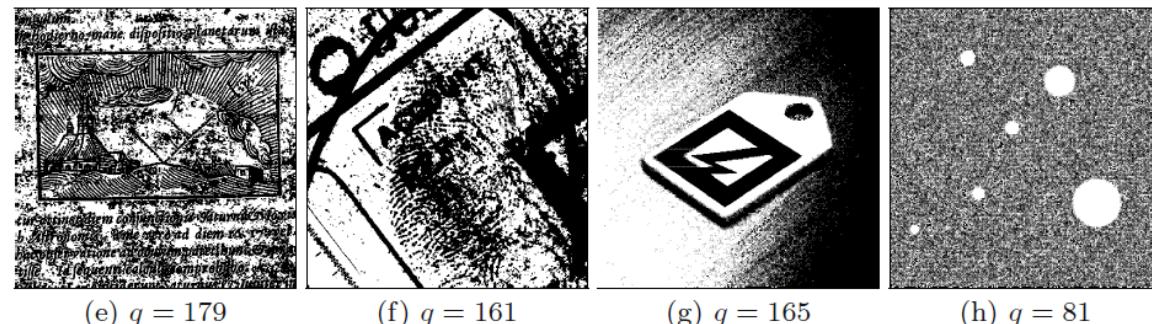
$$q = \operatorname{argmin}_i \sum_{g=0}^i h(g) \geq N \cdot b$$

Exemple : Seuillage automatique simple

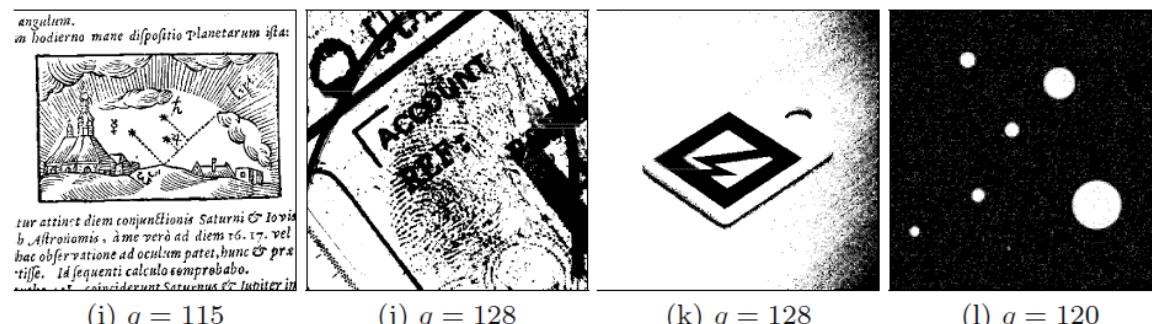
q=moyenne



q=médiane



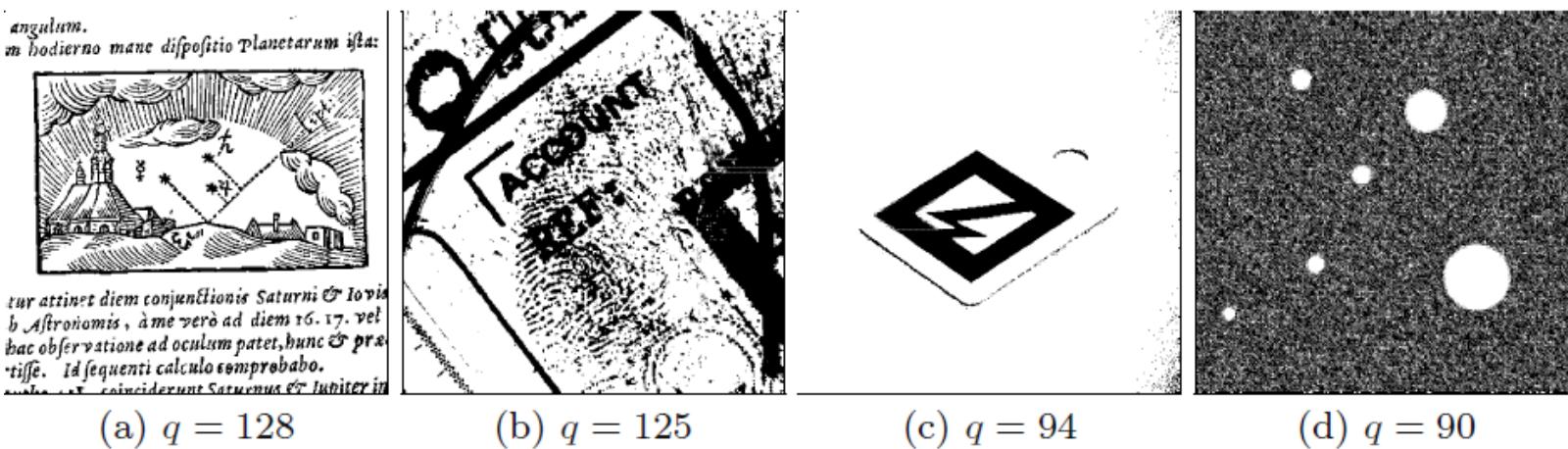
q=mi-étendue



Algorithme ISODATA : Sélection itérative du seuil

- **Hypothèse 1** : L'image est formée de **2 distributions distinctes d'intensité** (arrière/avant-plan)
- **Hypothèse 2** : Distribution **gaussienne** de moyennes différentes ($\mu_0 \neq \mu_1$) et de **même variance** ($\sigma_0^2 = \sigma_1^2$)
- **Résumé** : Ajustement **itératif** du seuil en utilisant l'intensité moyenne pour chaque groupe

(Burger, Vol3, Fig2.4)



Exemple : Algorithme ISODATA

1: ISODATATHRESHOLD(h)

 Input: $h : [0, K-1] \mapsto \mathbb{N}$, a grayscale histogram.

 Returns the optimal threshold value or -1 if no threshold is found.

2: $K \leftarrow \text{Size}(h)$

 ▷ number of intensity levels

3: $q \leftarrow \text{Mean}(h, 0, K-1)$

 ▷ set initial threshold to overall mean

4: **repeat**

5: $n_0 \leftarrow \text{Count}(h, 0, q)$

 ▷ background population

6: $n_1 \leftarrow \text{Count}(h, q+1, K-1)$

 ▷ foreground population

7: **if** $(n_0 = 0) \vee (n_1 = 0)$ **then**

 ▷ backgrd. or foregrd. is empty

8: **return** -1

9: $\mu_0 \leftarrow \text{Mean}(h, 0, q)$

 ▷ background mean

10: $\mu_1 \leftarrow \text{Mean}(h, q+1, K-1)$

 ▷ foreground mean

11: $q' \leftarrow q$

 ▷ keep previous threshold

12: $q \leftarrow \left\lfloor \frac{\mu_0 + \mu_1}{2} \right\rfloor$

 ▷ calculate the new threshold

13: **until** $q = q'$

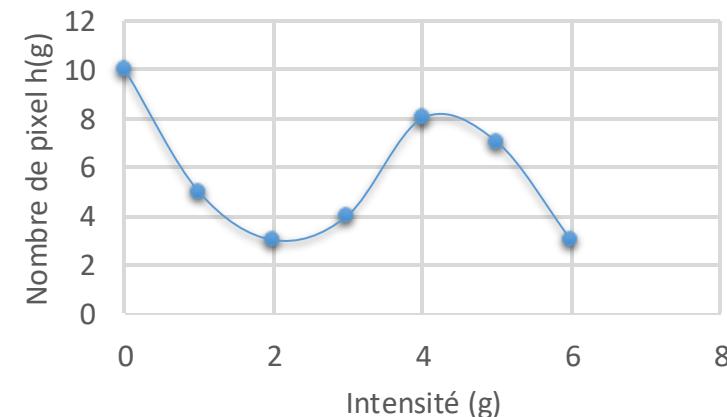
 ▷ terminate if no change

14: **return** q

15: $\text{Count}(h, a, b) := \sum_{g=a}^b h(g)$

16: $\text{Mean}(h, a, b) := \left[\sum_{g=a}^b g \cdot h(g) \right] / \left[\sum_{g=a}^b h(g) \right]$

Intensité g	# pixel $h(g)$
0	10
1	5
2	3
3	4
4	8
5	7
6	3



Méthode d'Otsu

- **Hypothèse** : 2 distributions distinctes d'intensité
- **Distributions inconnues**
- **But** : Trouver un seuil q permettant de séparer les distributions maximalement, c.-à-d.
 - Chaque distribution est mince (*variances minimales*)
 - Les moyennes de chaque distribution sont éloignées
- Otsu, N. (1979). A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9, 62-66. (>32000 citations)

Méthode d’Otsu : Variance intraclasse

Variance *intraclasse*

$$\sigma_w^2(q) = P_0(q) \cdot \sigma_0^2(q) + P_1(q) \cdot \sigma_1^2(q)$$

où

$$P_0(q) = \frac{1}{N} \sum_{i=0}^q h(i) = \frac{n_0(q)}{N}$$

$$P_1(q) = \frac{1}{N} \sum_{i=q+1}^{K-1} h(i) = \frac{n_1(q)}{N}$$

sont la probabilité de chaque classe \mathcal{C}_0 et \mathcal{C}_1

Méthode d’Otsu – Variance interclasse

Variance *interclasse*

$$\sigma_b^2(q) = P_0(q) \cdot (\mu_0 - \mu_I)^2 + P_1(q) \cdot (\mu_{1(q)} - \mu_I)^2$$

- Mesure la distance entre la moyenne des partitions μ_0 , μ_1 et la moyenne de l’image μ_I

Variance totale de l’image

$$\sigma_I^2 = \sigma_w^2 + \sigma_b^2$$

On obtient le seuil optimal q en **minimisant** la variance **intraclasse** σ_w^2 ou en **maximisant** la variance **interclasse** σ_b^2

Méthode d’Otsu : Maximisation

- Choix naturel : **Maximiser** σ_b^2 , car dépend uniquement de statistiques de 1^{er} ordre (μ_0 et μ_1)
- Puisque $\mu_I = \mu_0 + \mu_1$, on peut reformuler :

$$\sigma_b^2(q) = P_0(q) \cdot P_1(q) \cdot [\mu_0(q) - \mu_1(q)]^2$$

$$= \frac{n_0(q) \cdot n_1(q)}{N^2} \cdot [\mu_0(q) - \mu_1(q)]^2$$

La méthode d’Otsu consiste à maximiser cette expression de la variance interclasse selon le seuil q

Méthode d’Otsu - Algorithme

```

1: OTSUTHRESHOLD( $h$ )
Input:  $h : [0, K-1] \mapsto \mathbb{N}$ , a grayscale histogram.
Returns the optimal threshold value or  $-1$  if no threshold is found.

2:  $K \leftarrow \text{Size}(h)$                                  $\triangleright$  number of intensity levels
3:  $(\mu_0, \mu_1, N) \leftarrow \text{MAKEMEANTABLES}(h, K)$      $\triangleright$  see Alg. 2.3
4:  $\sigma_{b\max}^2 \leftarrow 0$ 
5:  $q_{\max} \leftarrow -1$ 
6:  $n_0 \leftarrow 0$ 
7: for  $q \leftarrow 0, \dots, K-2$  do     $\triangleright$  examine all possible threshold values  $q$ 
8:    $n_0 \leftarrow n_0 + h(q)$ 
9:    $n_1 \leftarrow N - n_0$ 
10:  if  $(n_0 > 0) \wedge (n_1 > 0)$  then
11:     $\sigma_b^2 \leftarrow \frac{1}{N^2} \cdot n_0 \cdot n_1 \cdot [\mu_0(q) - \mu_1(q)]^2$      $\triangleright$  see Eqn. (2.25)
12:    if  $\sigma_b^2 > \sigma_{b\max}^2$  then                                 $\triangleright$  maximize  $\sigma_b^2$ 
13:       $\sigma_{b\max}^2 \leftarrow \sigma_b^2$ 
14:       $q_{\max} \leftarrow q$ 
15:  return  $q_{\max}$ .

```

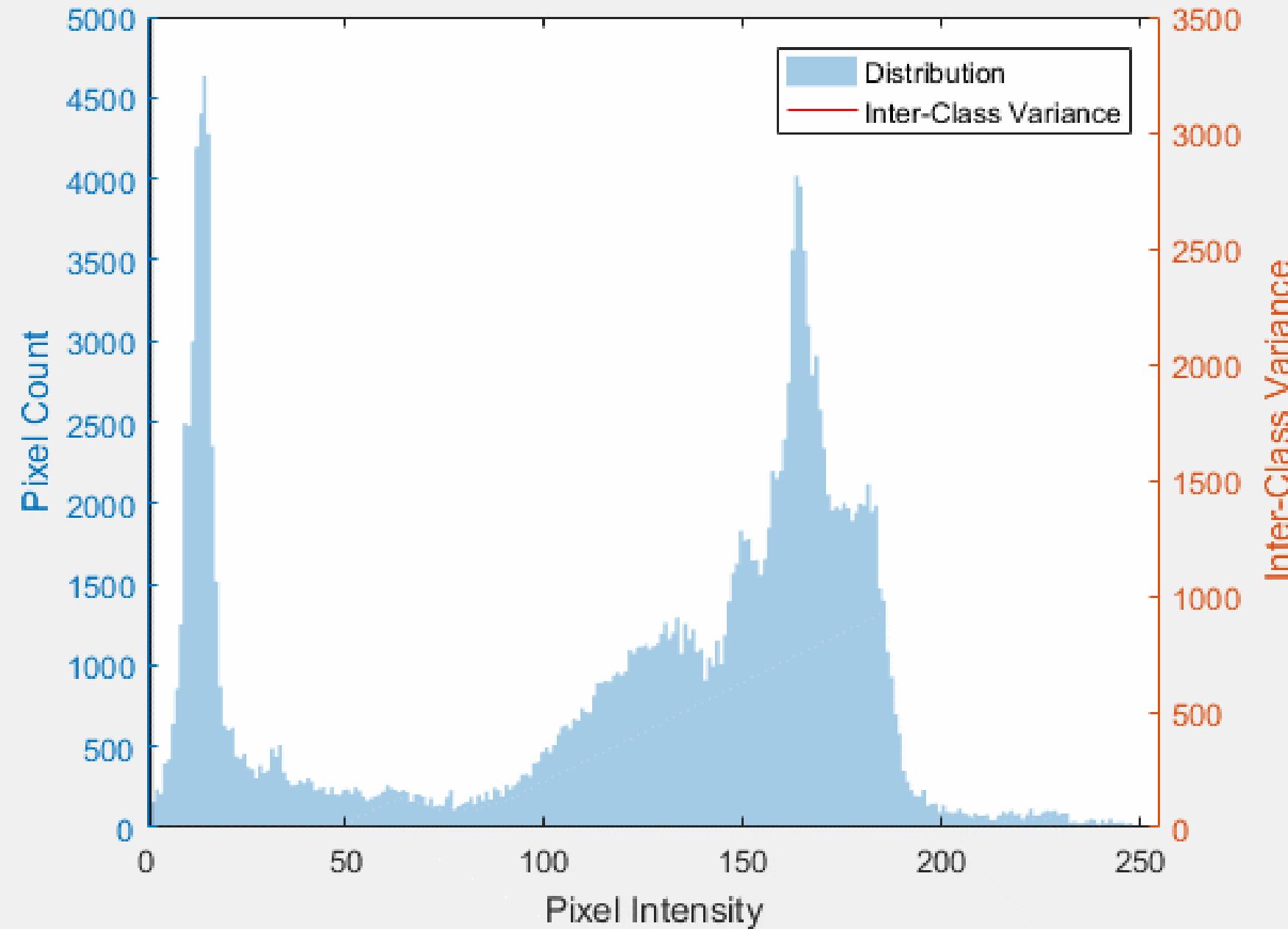
```

MAKEMEANTABLES( $h, K$ )
Create maps  $\mu_0, \mu_1 : [0, K-1] \mapsto \mathbb{R}$ 
 $n_0 \leftarrow 0, s_0 \leftarrow 0$ 
for  $q \leftarrow 0, \dots, K-1$  do           $\triangleright$  tabulate background means  $\mu_0(q)$ 
   $n_0 \leftarrow n_0 + h(q)$ 
   $s_0 \leftarrow s_0 + q \cdot h(q)$ 
   $\mu_0(q) \leftarrow \begin{cases} s_0/n_0 & \text{if } n_0 > 0 \\ -1 & \text{otherwise} \end{cases}$ 
 $N \leftarrow n_0$ 
 $n_1 \leftarrow 0, s_1 \leftarrow 0$ 
 $\mu_1(K-1) \leftarrow 0$ 
for  $q \leftarrow K-2, \dots, 0$  do           $\triangleright$  tabulate foreground means  $\mu_1(q)$ 
   $n_1 \leftarrow n_1 + h(q+1)$ 
   $s_1 \leftarrow s_1 + (q+1) \cdot h(q+1)$ 
   $\mu_1(q) \leftarrow \begin{cases} s_1/n_1 & \text{if } n_1 > 0 \\ -1 & \text{otherwise} \end{cases}$ 
return  $\langle \mu_0, \mu_1, N \rangle$ 

```

« Qualité » du seuillage

$$\eta = \frac{\sigma_b^2(q_{\max})}{\sigma_I^2} \in [0, 1]$$



Exemple : Méthode d'Otsu

*angulum.
in hodierno mane dispositio Planetarum ista:*



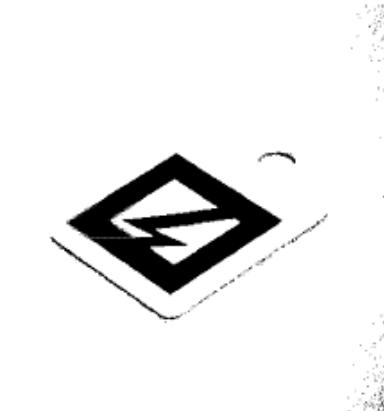
*tur attinet diem coniunctionis Saturni & Iovis
b Astronomis, à me vero ad diem 16. 17. vel
hic observatione ad oculum patet, hunc & pra
tisse. Id sequenti calculo comprobabo.*

... coincidunt Saturnus & Jupiter in

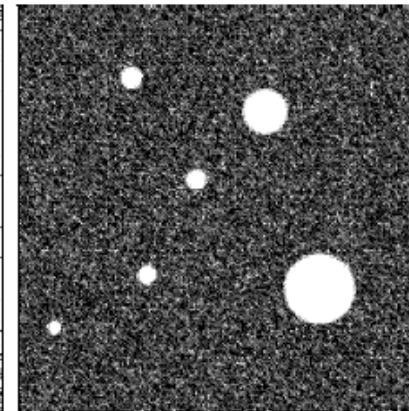
(a) $q_{\max} = 128$



(b) $q_{\max} = 124$

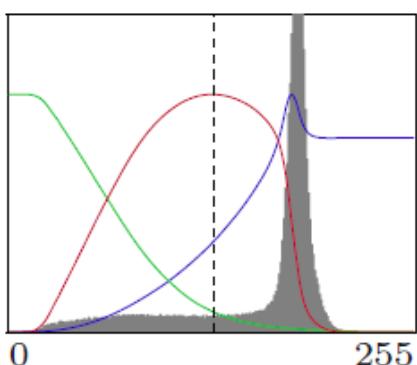


(c) $q_{\max} = 94$

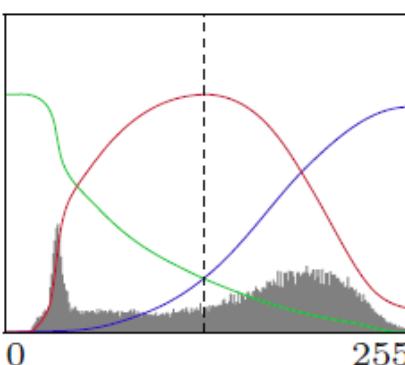


(d) $q_{\max} = 92$

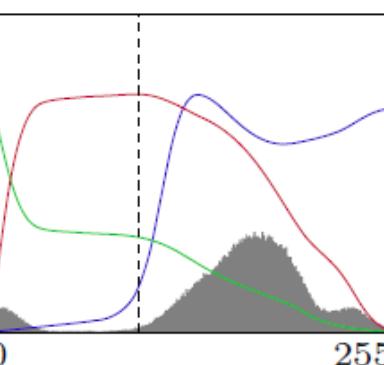
σ_0^2
 σ_1^2
 σ_b^2



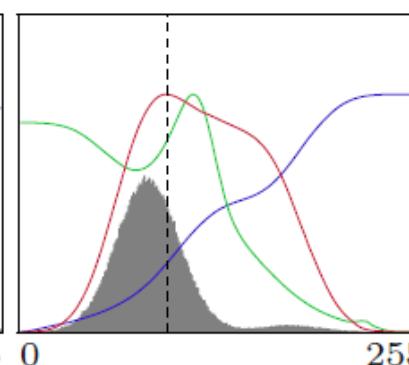
(e) $\eta = 0.84$



(f) $\eta = 0.77$



(g) $\eta = 0.62$

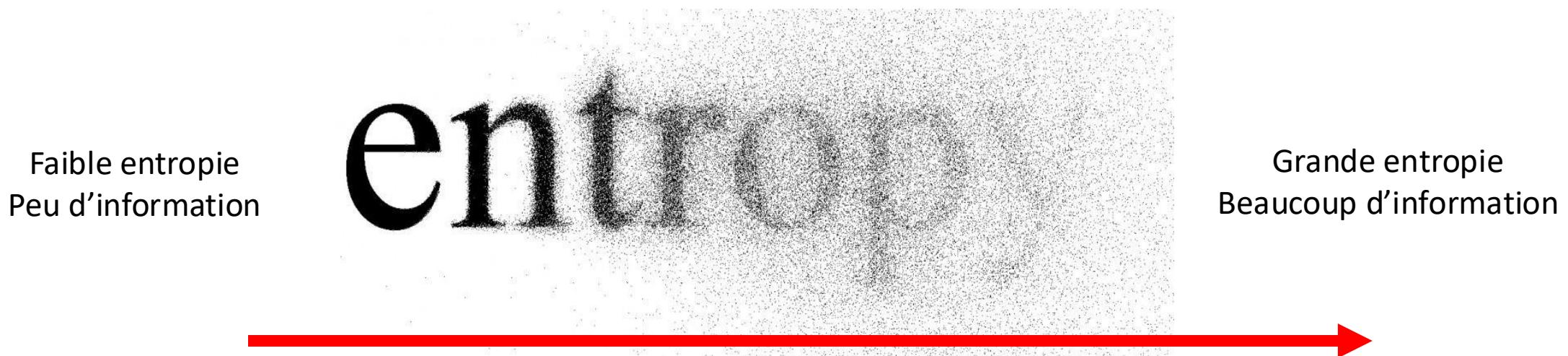


(h) $\eta = 0.53$

(Burger, Vol3, Fig2.5)

Seuillage automatique global : Maximisation de l'entropie (1)

- En théorie de l'information, **l'entropie** mesure la quantité moyenne d'information contenue dans un message
 - **Faible entropie** : peu d'information
 - **Grande entropie** : beaucoup d'information
- Indique la **probabilité d'obtenir un message** donné (*surprise*)



Seuillage automatique global : Maximisation de l'entropie (2)

- Distribution de probabilité des intensités de l'image

$$p(g) = \frac{h(g)}{N}$$

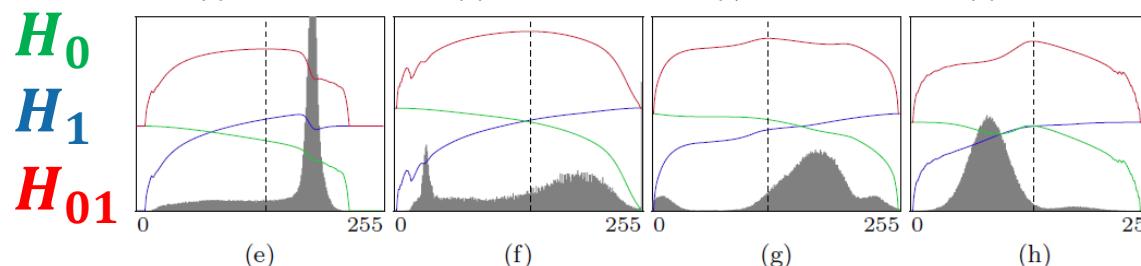
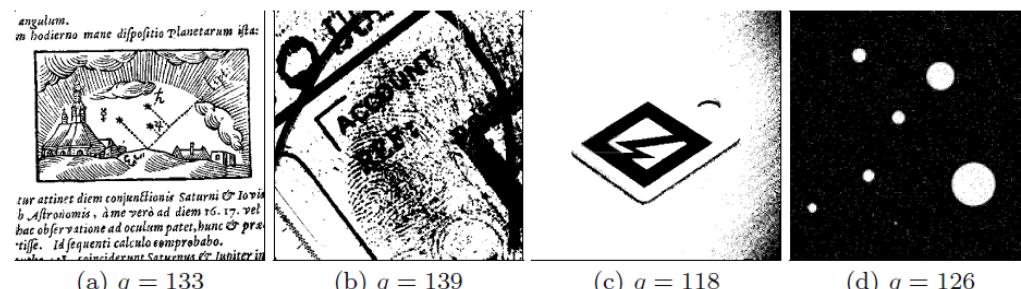
- **Entropie de l'image**

$$H(I) = \sum_{x,y} p(g) \cdot \log_b(p(g))$$

- Où $b=2$ est l'entropie (contenu en information) mesurée en bit

Seuillage automatique global : Maximisation de l'entropie (3)

- **Entropie nulle** : Image uniforme (ex. : tous les pixels = 0)
- **Entropie maximale** : Toutes les intensités ont la même probabilité (*distribution uniforme*)
- Approche similaire à ISODATA et Otsu pour déterminer le seuil *q* maximisant l'entropie.



*angulum.
m hodierno mane dispositio Planetarum ista:*



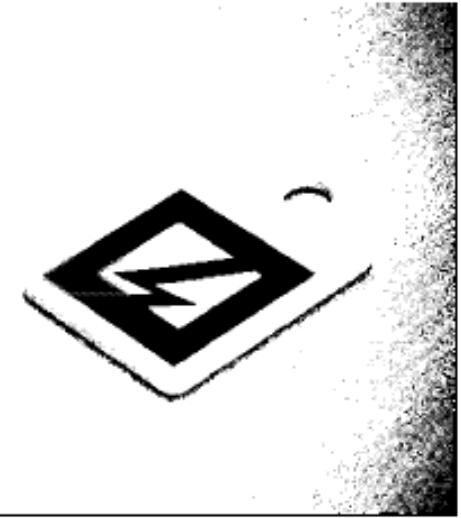
*tur atinet diem conjunctionis Saturni & Iovis
b Astronomis, à me verò ad diem 16. 17. vel
bac observatione ad oculum patet, hunc & pra-
tissē. Id sequenti calculo comprobabo.*

coinciderunt Saturnus & Iupiter in

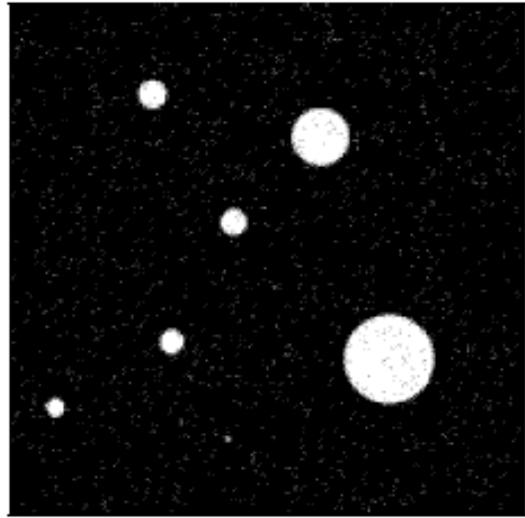
(a) $q = 133$



(b) $q = 139$

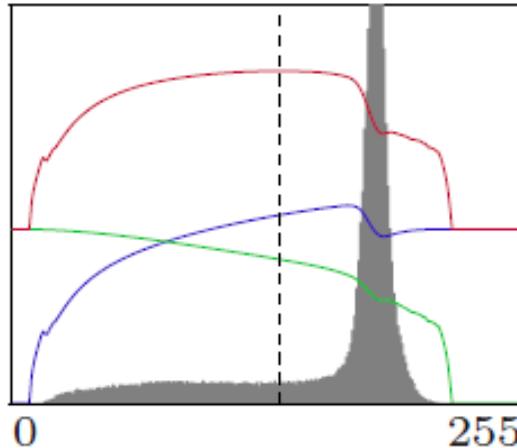


(c) $q = 118$

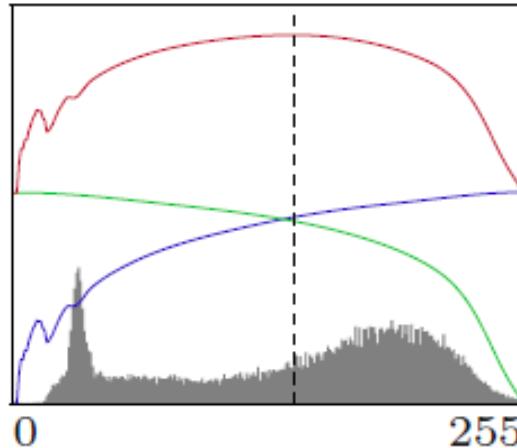


(d) $q = 126$

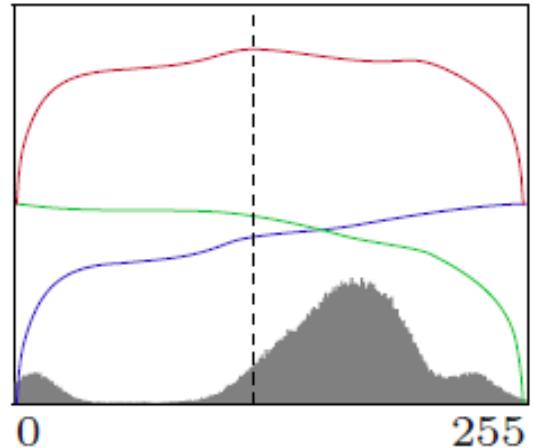
H_0
 H_1
 H_{01}



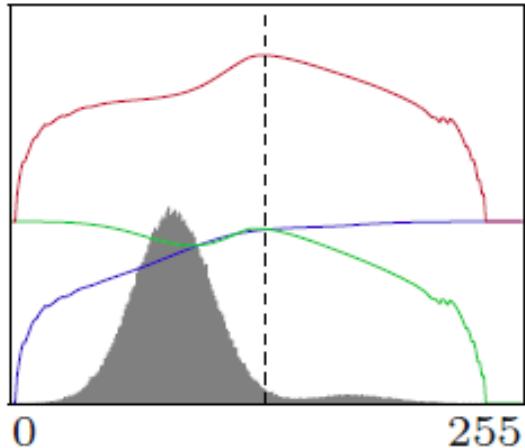
(e)



(f)



(g)



(h)

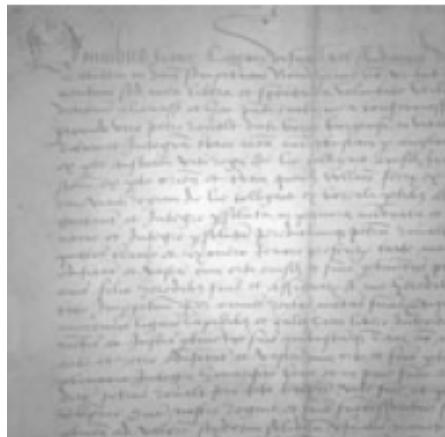
Seuillage global : scikit-image

Méthode	Description
skimage.filters.threshold_isodata	Return threshold value(s) based on ISODATA method.
skimage.filters.threshold_li	Compute threshold value by Li's iterative Minimum Cross Entropy method.
skimage.filters.threshold_mean	Return threshold value based on the mean of grayscale values.
skimage.filters.threshold_minimum	Return threshold value based on minimum method.
skimage.filters.threshold_multiotsu	Generate classes-1 threshold values to divide gray levels in image.
skimage.filters.threshold_otsu	Return threshold value based on Otsu's method.
skimage.filters.threshold_triangle	Return threshold value based on the triangle algorithm.
skimage.filters.threshold_yen	Return threshold value based on Yen's method.
skimage.filters.try_all_threshold	Returns a figure comparing the outputs of different thresholding methods.

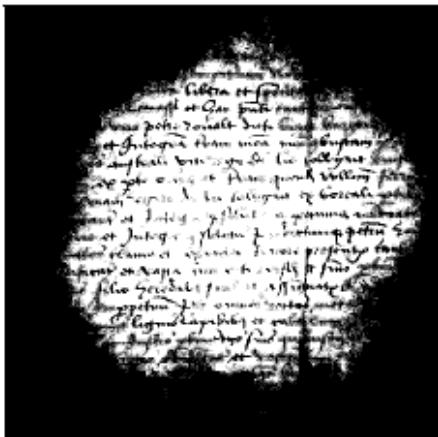
Comment appliquer le seuil ?

Problème avec le seuillage global

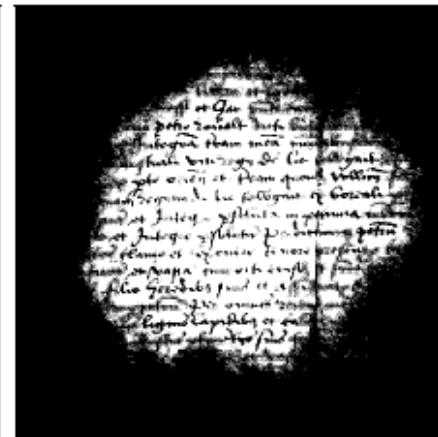
- Un seuillage global n'est **pas approprié** lorsqu'il y a des **variations locales de l'intensité**
- **Exemple** : Illumination non homogène
- **Exemple** : Présence de bruit
- **Exemple** : Taille des objets à segmenter



(a) Original



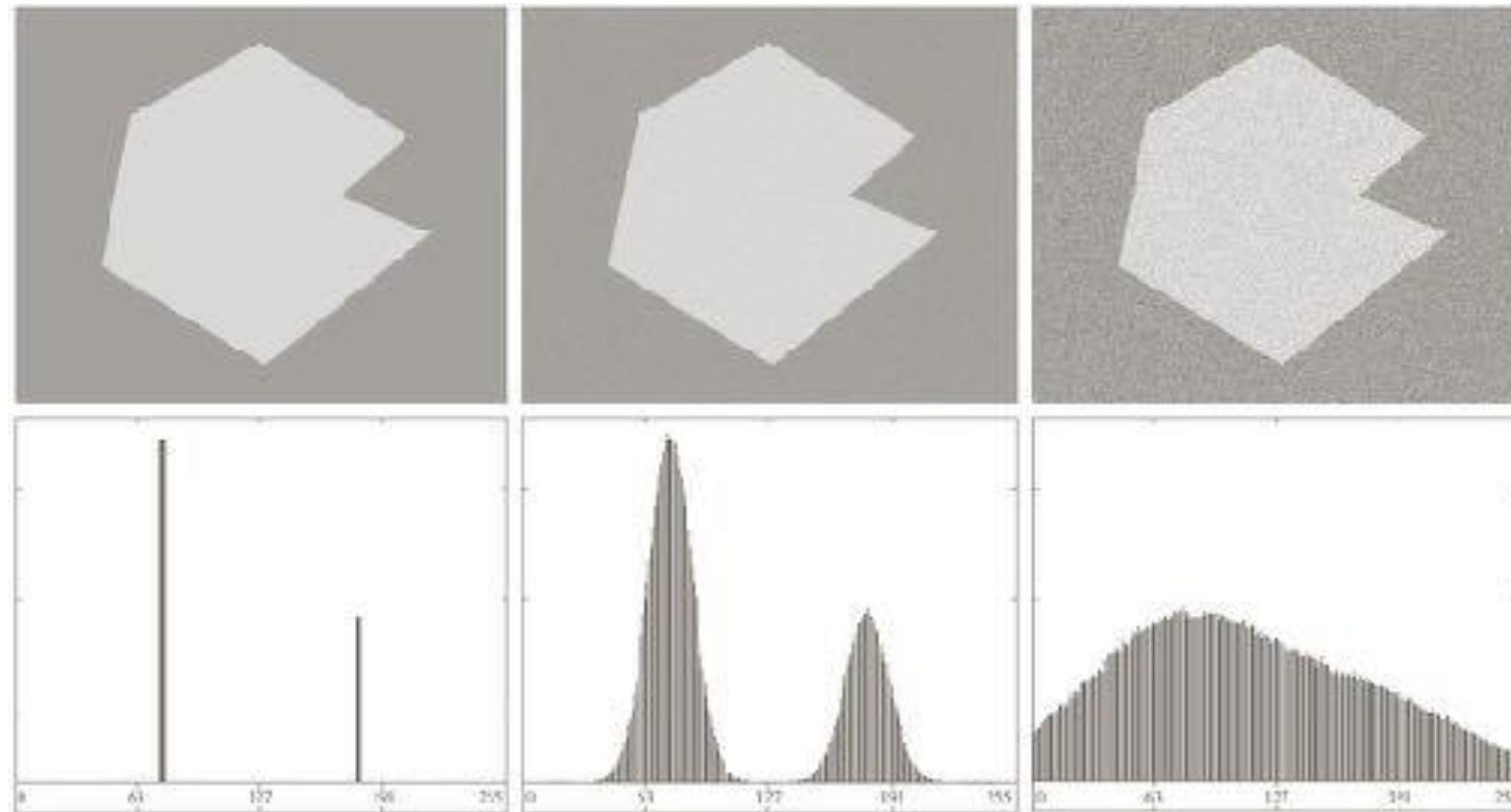
(b) Otsu



(c) Max. entropy

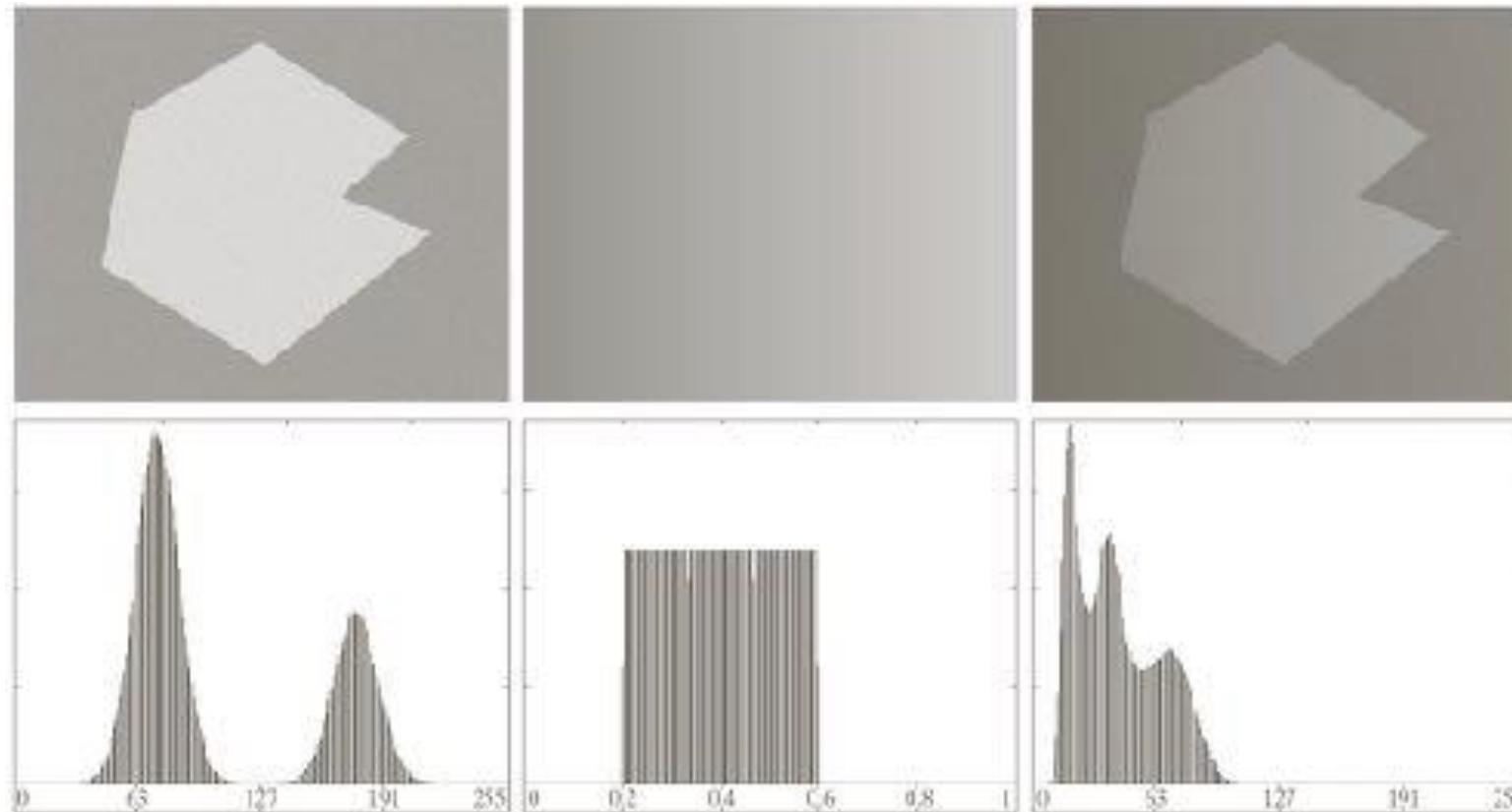
Problème #1 : Effet du bruit

- Difficile de séparer l'histogramme s'il y a beaucoup de bruit



Problème #2 : Biais d'intensité

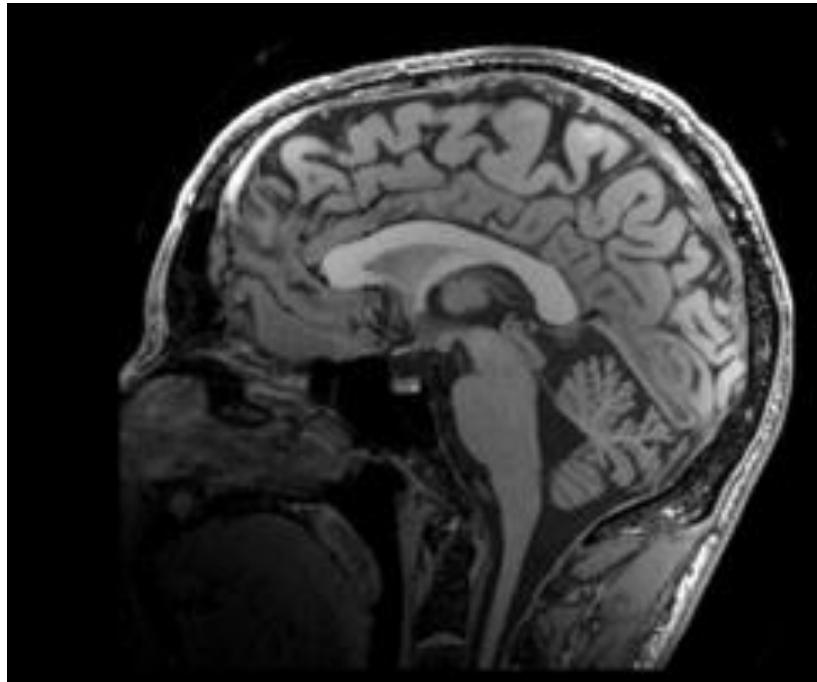
- Difficile de séparer l'histogramme en présence d'un biais d'intensité



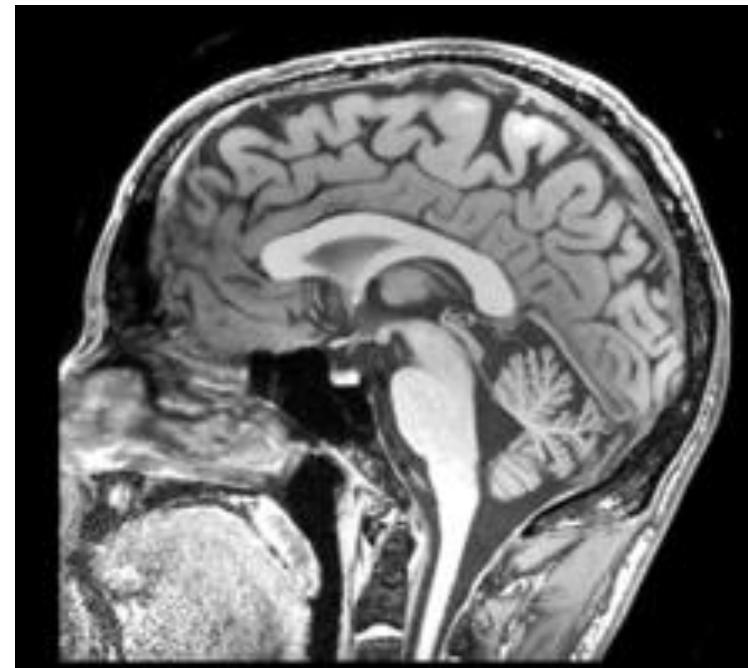
Correction pour le problème #2 : Biais d'intensité

- Les biais d'intensité peuvent être corrigés avec différentes méthodes (p. ex. l'algorithme N4 en IRM)

Avant correction



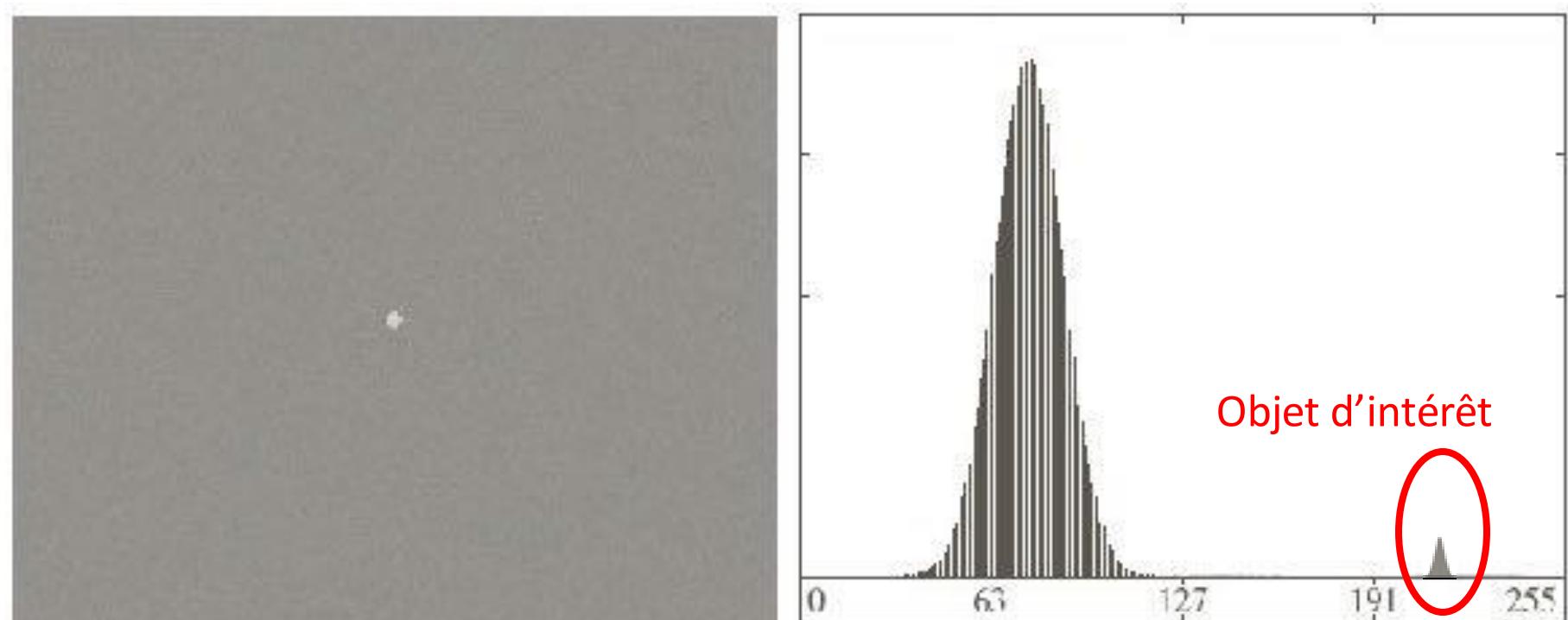
Après correction



[Source : www.slicer.org](http://www.slicer.org)

Problème #3 : Taille des régions

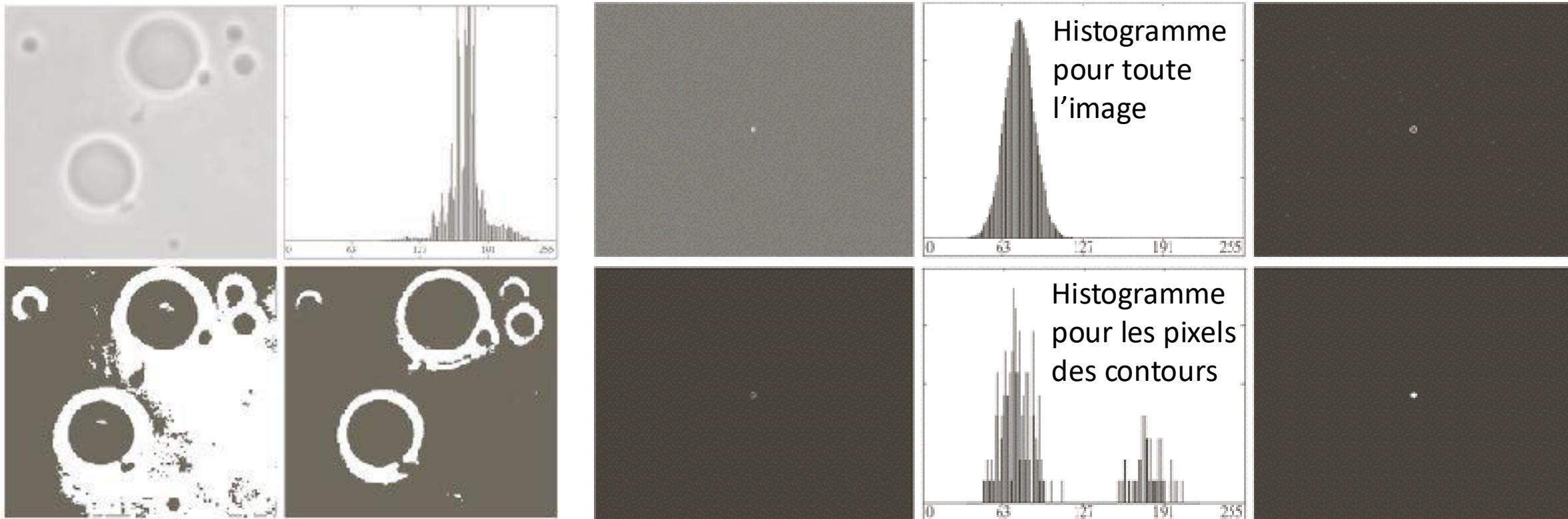
- Exemple de limitation de l'approche histogramme. Dans le cas d'un objet très petit, la segmentation est délicate.



Comment pallier le bruit et la taille des régions ?

- Calcul de l'histogramme dans une partie de l'image rétablissant l'équilibre entre tailles des régions
- **Étapes du traitement**
 - Détection des **contours** et seuillage par une méthode vue précédemment
 - Calcul de **l'histogramme sur les pixels des contours** ou un voisinage de ceux-ci
 - **Méthode de Otsu** sur l'histogramme ainsi obtenu

Exemple : Méthode d’Otsu en utilisant les contours



Problème avec le seuillage global

Solution possible : Seuillage adaptatif

- **Seuillage adaptatif** : Le seuil $Q(x, y)$ varie spatialement pour s'adapter localement à l'image
- Quelques techniques
 - **Méthode de Bernsen** : skimage.filters.threshold_local
 - **Méthode de Niblack** : skimage.filters.threshold_niblack
 - **Méthode de Sauvola*** : skimage.filters.threshold_sauvola

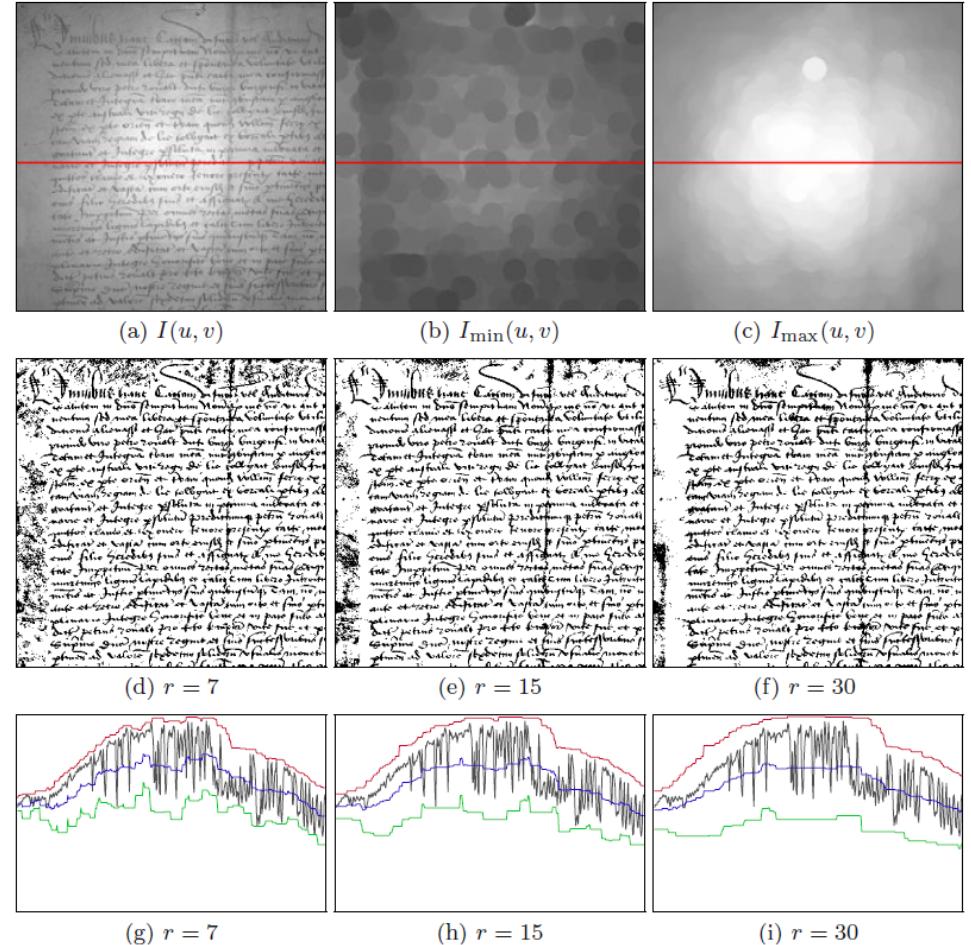
* La méthode de Sauvola est une modification de la méthode de Niblack

Seuillage adaptatif : Méthode de Bernsen

- Le seuil en tout point est calculé à partir de l'intensité minimale $I_{\min}(x, y)$ et maximale $I_{\max}(x, y)$ dans un voisinage N du pixel.

$$Q(x, y) = \frac{I_{\min}(x, y) + I_{\max}(x, y)}{2}$$

- D'autres types de seuils peuvent être utilisés localement
 - Moyenne
 - Médiane
 - Otsu local
 - Etc.



(Burger, Vol3, Fig2.9)

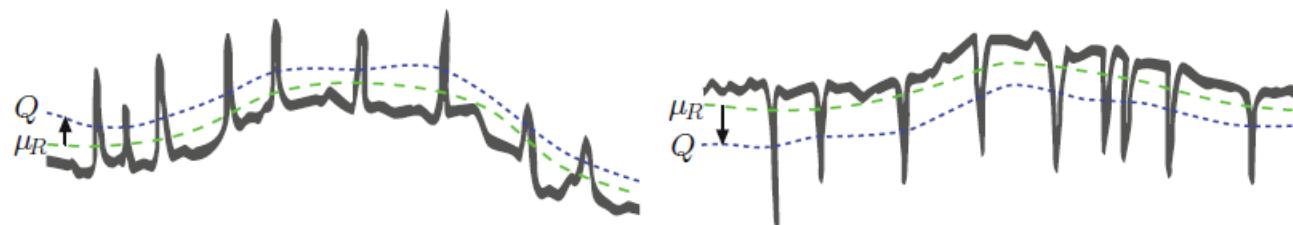
Seuillage adaptatif : Méthode de Niblack

- Le seuil local est défini en fonction de la **moyenne** $\mu_N(x, y)$ et de l'**écart-type** $\sigma_N(x, y)$ **local** sur un voisinage N

$$Q(x, y) = \begin{cases} \mu_N(x, y) + (\kappa \cdot \sigma_N(x, y) + d) & \text{pour un fond sombre} \\ \mu_N(x, y) - (\kappa \cdot \sigma_N(x, y) + d) & \text{pour un fond clair} \end{cases}$$

$$\kappa \geq 0, d \geq 0$$

- Le **rayon** R du voisinage N devrait être **le plus grand possible**, au moins plus grand que la structure à détecter (ex. : taille d'une lettre)

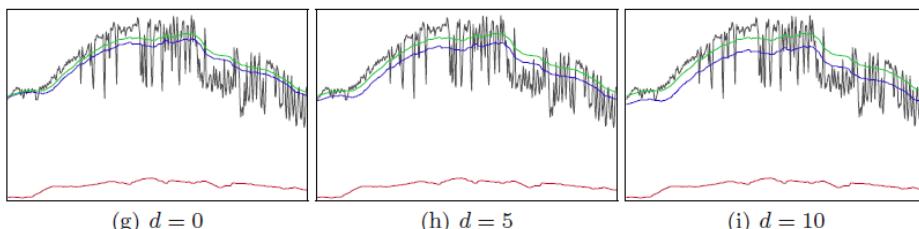
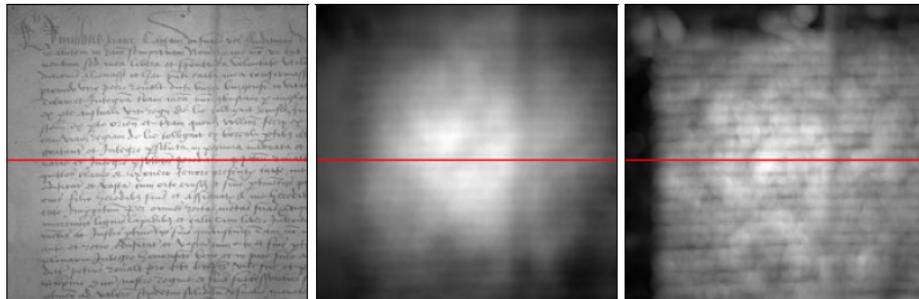


(a) dark background

(b) bright background

(Burger, Vol3, Fig2.12)

Exemple : Seuillage adaptatif avec la méthode de Niblack



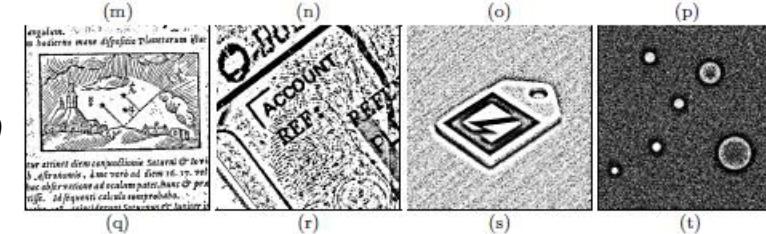
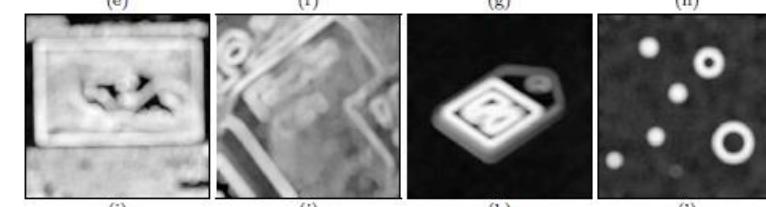
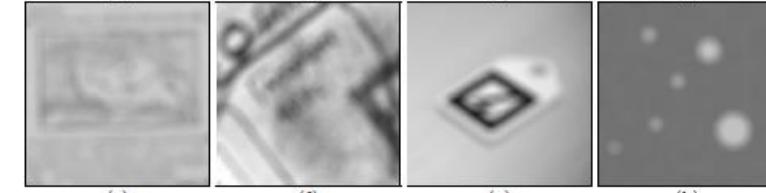
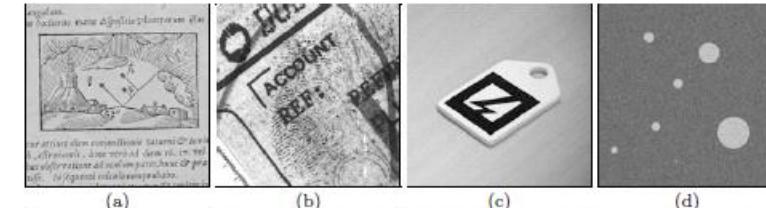
$|x, y|$

μ_N

σ_N

Q

$I > Q$



(Burger, Vol3, Fig2.13-2.14)

Résumé : Techniques de seuillage local ou adaptatif

Seuillage global

- Simple
- Forte composante empirique
- Caractéristiques propres aux images non prises en compte (p. ex. taille des objets)

Seuillage local ou adaptatif

- Division en sous-images (vignette)
- Traitement indépendant des vignettes avec les techniques précédentes
- Efficace pour des variations lentes de l'intensité
- Meilleure prise en compte des caractéristiques propres aux images
- Paramètres supplémentaires : Taille des voisinages ? Notion de voisinage ?

Méthodes avancées

Chapitre 5 : Segmentation (Partie 2)

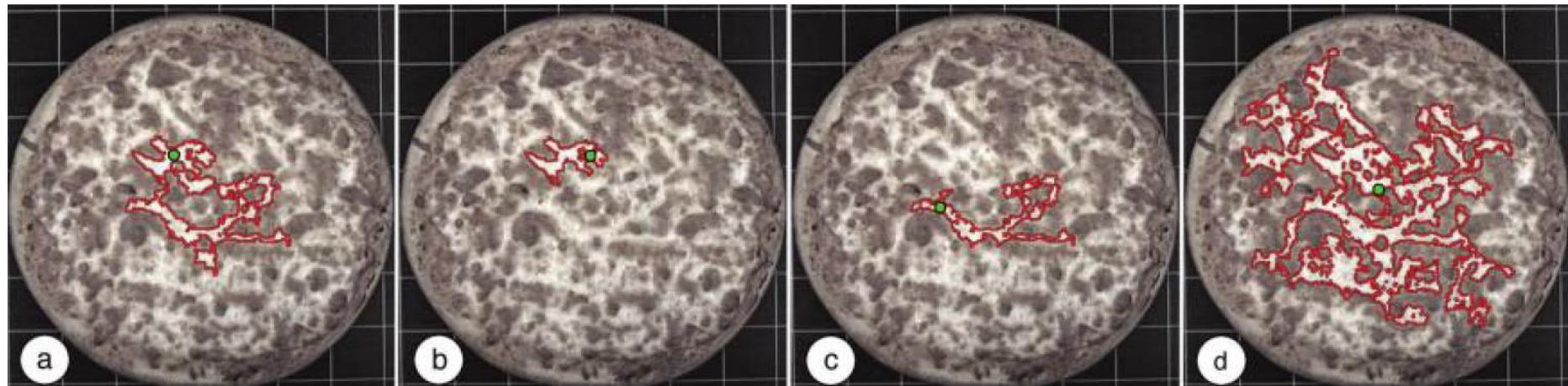
Joël Lefebvre (UQÀM)

INF600F – Traitement d'images

Approche par croissance de régions

- Méthode de **segmentation directe des régions**
- On débute avec des **pixels sources** (seeds)
- On itère ensuite sur le voisinage de chaque pixel
- Les pixels sont **ajoutés progressivement à la région** selon leurs propriétés (ex. : similarité d'intensité)

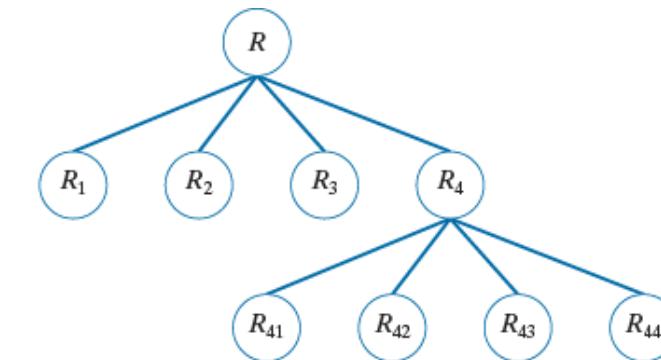
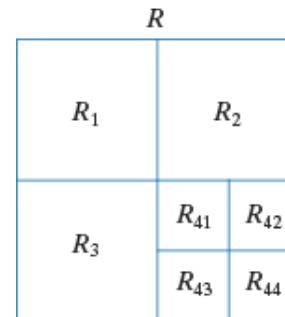
(Russ, Fig 7.60)



Approche Fractionnement / Fusion (*Split and merge*)

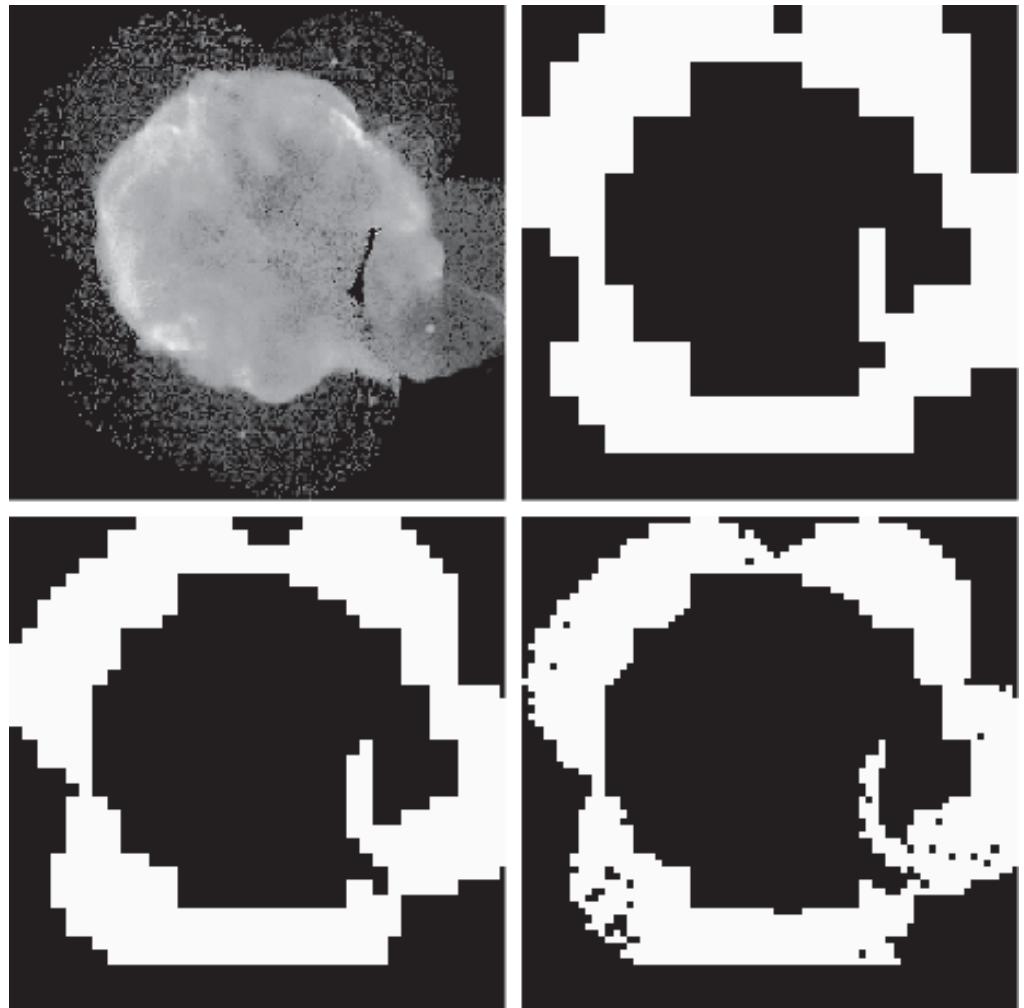
1. Choisir un critère Q que chaque région R doit respecter
2. Séparer une région R_i en 4 quadrants disjoints si $Q(R_i) = FAUX$
3. Lorsqu'il n'est plus possible de fractionner, fusionner des régions adjacentes si $Q(R_j \cup R_k) = VRAI$
4. Arrêter lorsqu'il n'y a plus de possibilité de fusion.

Représentation de l'image en *quadtree*
(Gonzalez Woods, Fig 10.47)



Exemple : Approche *Split / Merge*

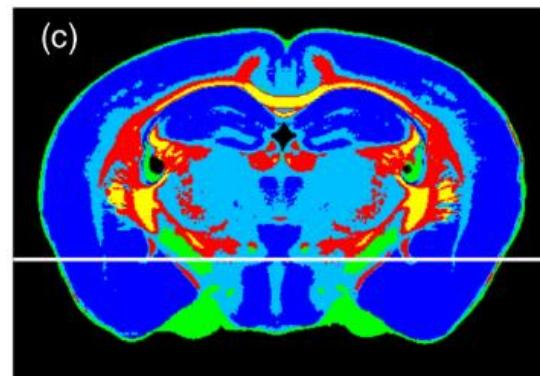
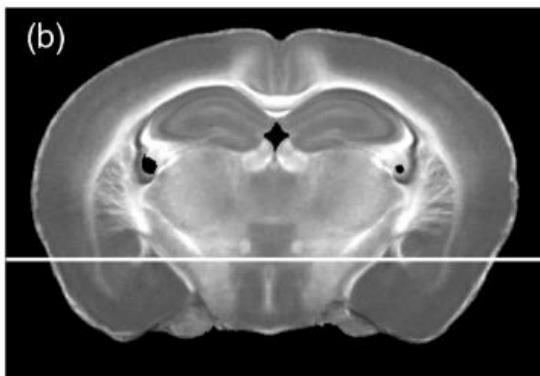
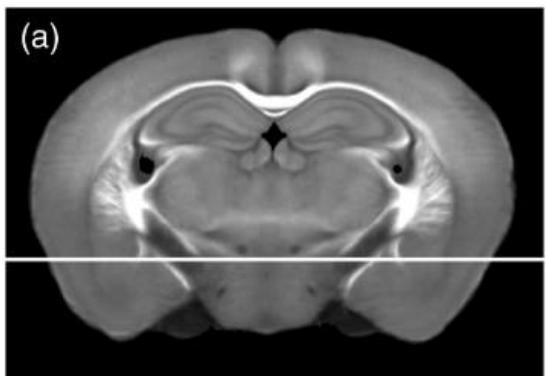
- Image (rayon X) de la supernova « Dentelles du Cygne » (Nasa)
- Segmentation *Split and Merge*
- Tailles minimales des régions : 32, 16, et 8 px.
- Critère de similarité
$$\sigma_R > a \text{ AND } 0 < m_R < b$$
- Source : (Gonzalez, Woods, Fig10.48)



Segmentation de région par partitionnement (*clustering*)

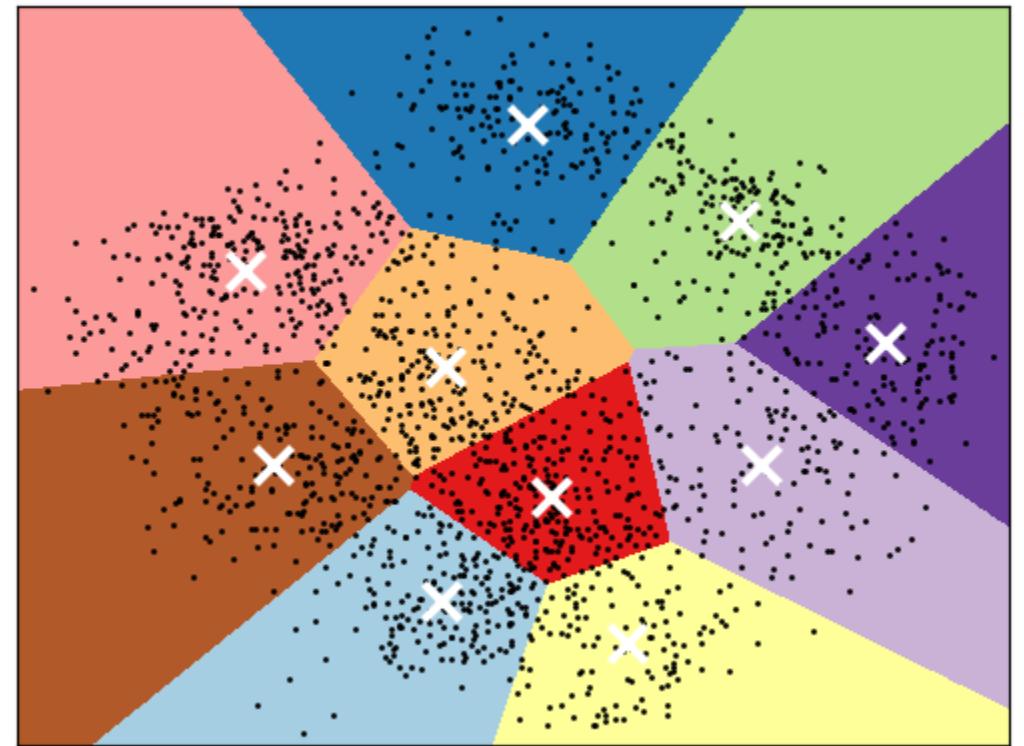
- **Apprentissage machine non supervisé** pour séparer les données en plusieurs groupes
- **Exemple : la méthode *K-moyennes*** sépare les données en K groupes en raffinant de façon itérative la moyenne de chaque groupe jusqu'à convergence.
- Peut utiliser plusieurs propriétés pour chaque pixel (ex. : réflectivité et atténuation pour la microscopie)

Exemple de segmentation par partitionnement K-Means pour identifier les types de tissus en microscopie 3D (Cerveau de souris, [Lefebvre2017](#))



Partitionnement (*Clustering*)

- **Clustering** : Regrouper ensemble des points similaires et les représenter par un seul point
- Qu'est-ce qui rend 2 points / images / régions similaires ?
- Comment calculer un regroupement global à partir de similarités entre chaque paire de points ?



Source : [scikit-learn](#)

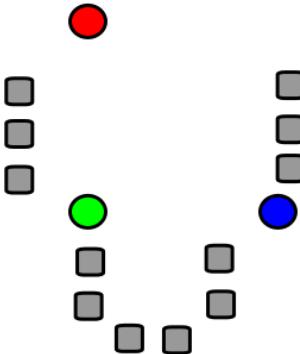
Pourquoi utiliser le partitionnement ?

- **Résumer les données**
 - Observer de grandes quantités de données
 - Compression & Débruitage de sous-régions
 - Représentation d'un grand vecteur par son numéro de partition
- **Dénombrement** : Histogramme des textures, des couleurs, des propriétés
- **Segmentation** : Séparation d'une image en différentes régions
- **Prédiction** : Des images dans une même partition ont peut-être la même étiquette sémantique

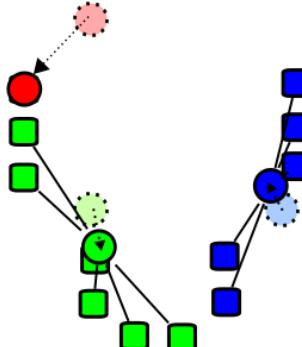
Comment partitionner des données ?

- **K-moyennes** (*K-means clustering*)
 - Réassigner les points de façon itérative à la partition dont le centre est le plus près
- **Partitionnement hiérarchique** (*Agglomerative Clust.*)
 - Commencer en considérant chaque point comme étant sa propre partition, et ensuite fusionner les partitions les plus près itérativement
- **Partitionnement décalage moyen** (*Mean-shift clustering*)
 - Estimer les modes de la densité de probabilité

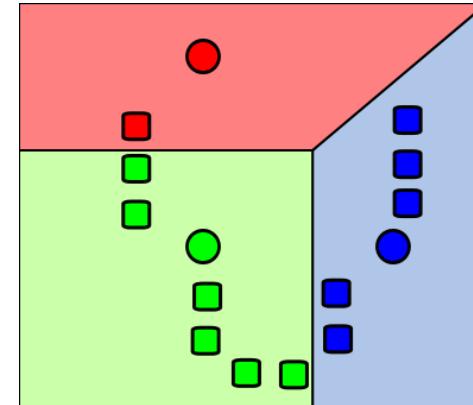
Partitionnement K-Moyennes



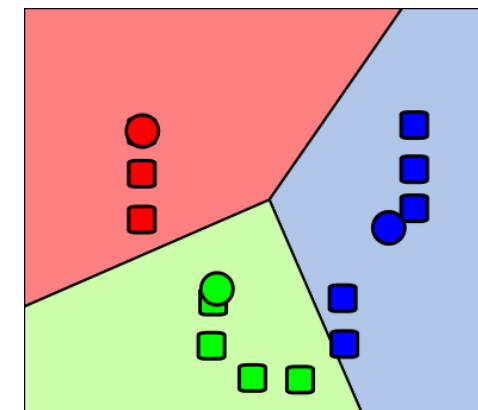
1. Sélectionner les centroïdes initiaux au hasard



3. Calculer chaque centroïde comme étant la moyenne des objets dans sa partition (voir 2)



2. Assigner chaque objet à la partition dont le centroïde est le plus près.



2. Assigner chaque objet à la partition dont le centroïde est le plus près.

Répéter les 2 étapes précédentes jusqu'à ce qu'il n'y ait plus de changement.

Algorithme du partitionnement *K-means*

Étant donné K:

1. Sélectionner les centroïdes initiaux au hasard
2. Assigner chaque objet à la partition dont le centroïde est le plus près
3. Calculer chaque centroïde comme étant la moyenne de la position des objets dans sa partition (voir 2)
4. Répéter les 2 étapes précédentes jusqu'à ce qu'il n'y ait plus de changement.

K-means : Choix de conception

- **Initialisation**
 - Sélectionner aléatoirement K points en tant que centres initiaux de partitions
 - Choisir les K points de façon en minimisant un résidu de façon gloutonne
- **Mesures de distance**
 - Traditionnellement Euclidienne, mais on peut utiliser d'autres distances
- **Optimisation**
 - Convergera vers un minimum local
 - On peut redémarrer l'algorithme plusieurs fois

Exemple : Partitionnement K-moyennes utilisant l'intensité ou la couleur

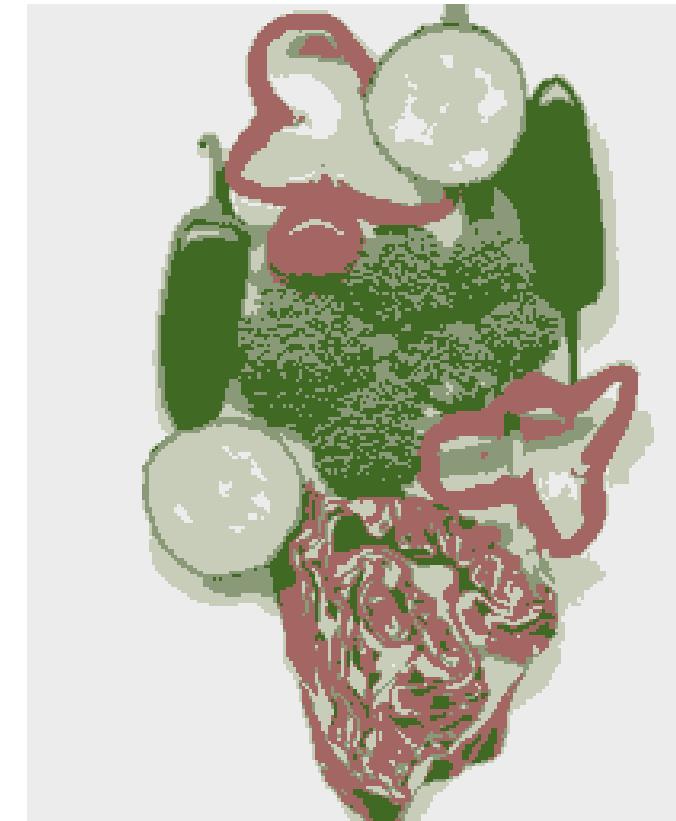
Image



Partitions (Intensité)



Partitions (Couleur)



Comment choisir le nombre de partitions ?

- Principe de longueur de description minimum (**Rasoir d'Occam**) pour la comparaison de modèle
- Minimisation du critère de Schwarz
 - Aussi nommé le **critère d'information de Bayes (BIC)**

$$\text{Distortion} + \lambda (\#\text{parameters}) \log R$$

$$= \text{Distortion} + \lambda m k \log R$$

$m = \#\text{dimensions}$

$k = \#\text{Centers}$

$R = \#\text{Records}$

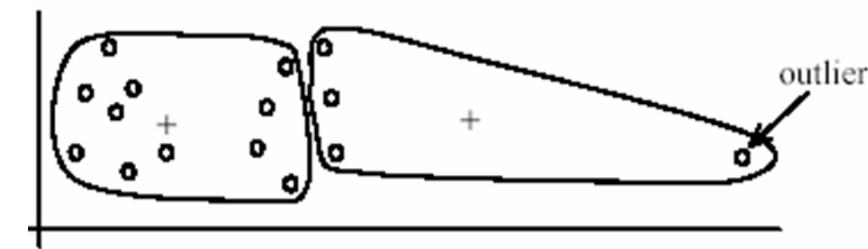
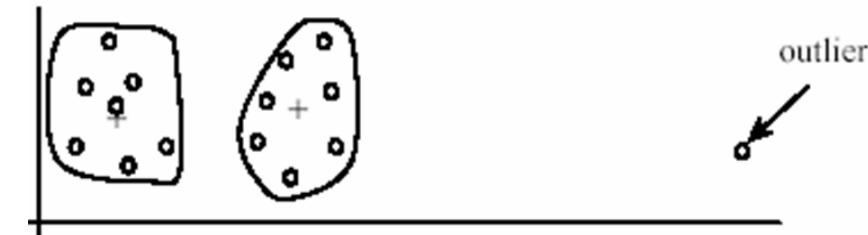
Avantages et inconvénients de la méthode K-moyennes

- **Avantages**

- Déetecte des centres de partitions qui minimisent la variance conditionnelle (bonne représentation des données)
- Simple et rapide
- Facile à implémenter

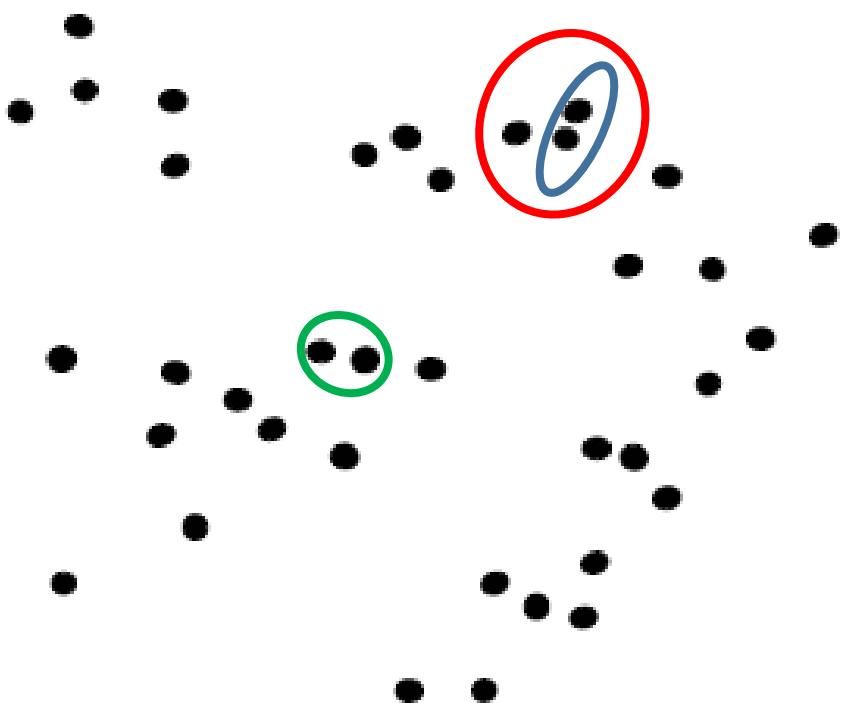
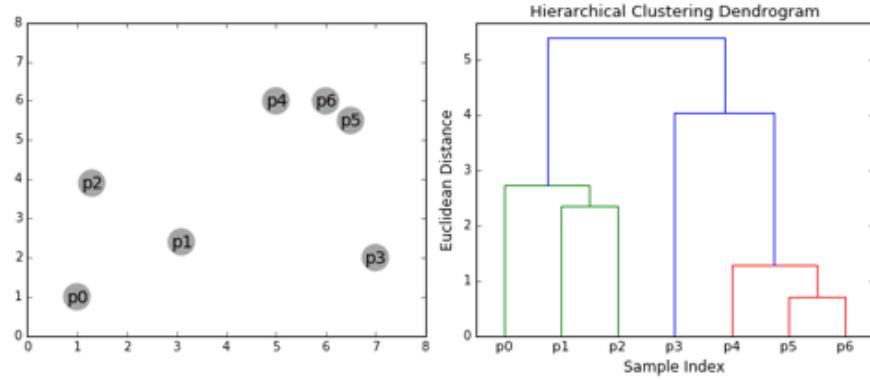
- **Inconvénients**

- On doit choisir K
- Sensible aux données aberrantes
- Tend à trouver les minimums locaux
- Toutes les partitions ont les mêmes paramètres



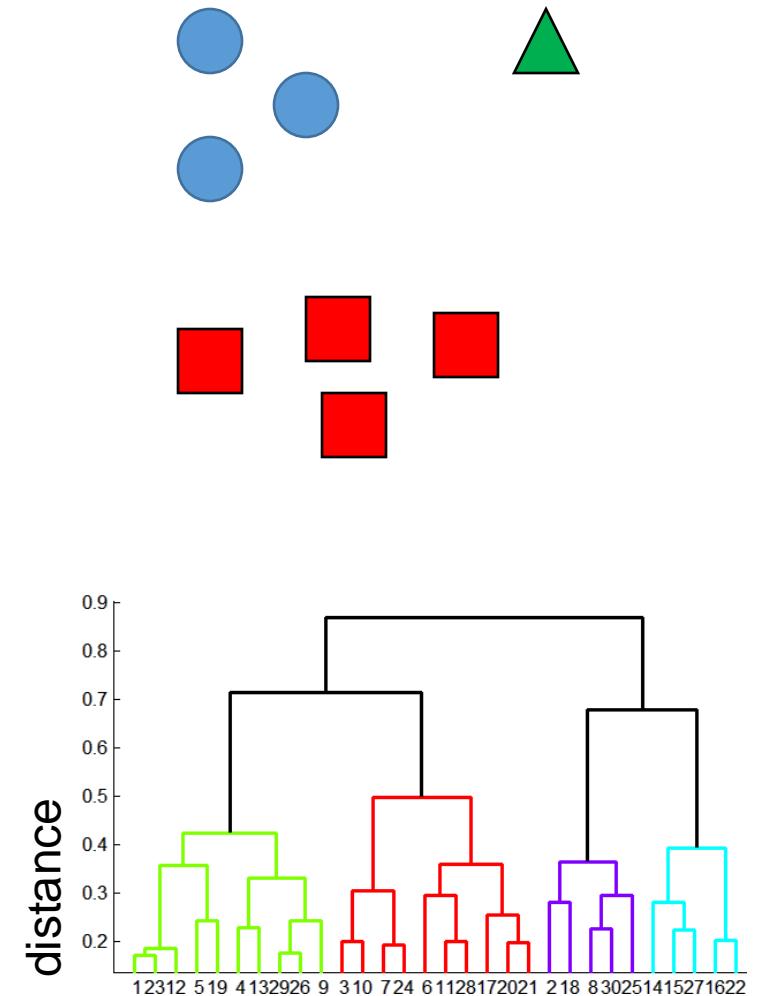
Partitionnement hiérarchique

1. On commence en assignant une partition à chaque point
2. On trouve ensuite la paire de partitions la « plus similaire »
3. On la fusionne en une partition parente
4. Répéter



Partitionnement hiérarchique (1)

- **Comment définir la similarité entre partitions ?**
 - Distance moyenne entre les points, distance maximum, distance minimum
 - Distance entre les moyennes
- **Combien de partitions ?**
- Le partitionnement crée un **dendrogramme** (un arbre)
- Un seuil basé sur le nombre maximum de partitions ou sur la distance entre les fusions



Partitionnement hiérarchique (2)

Avantage

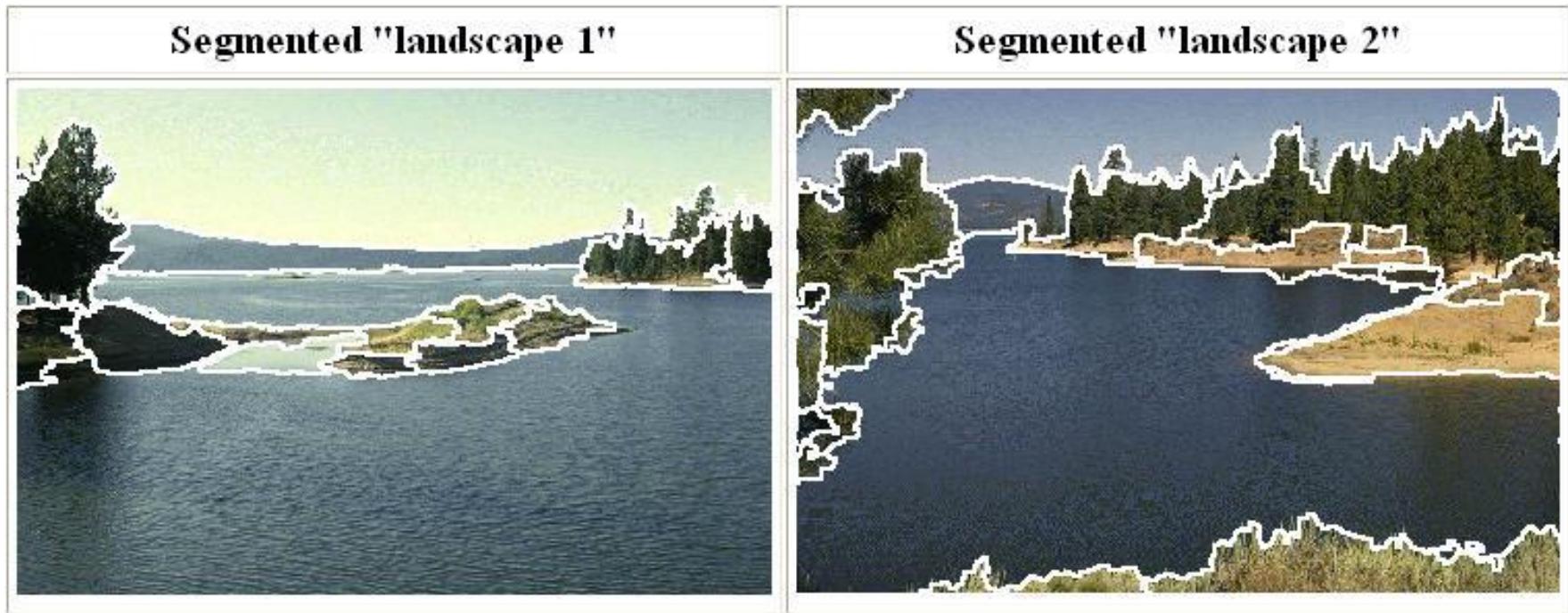
- Simple à implémenter, applications très répandues
- Les partitions ont une forme adaptative
- Fournit une hiérarchie de partitions

Inconvénient

- Peut former des partitions déséquilibrées
- On doit choisir un nombre de partitions ou un seuil
- On a besoin d'utiliser une métrique supplémentaire pour obtenir une hiérarchie utile

Segmentation par *Mean shift*

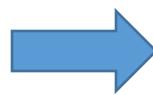
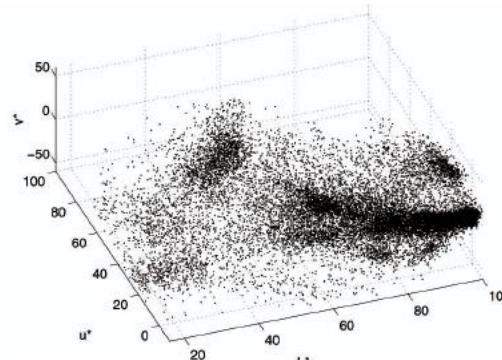
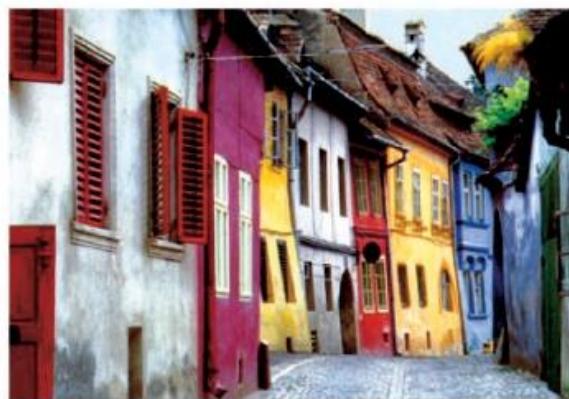
- **Technique versatile** pour la **segmentation** basée sur le partitionnement



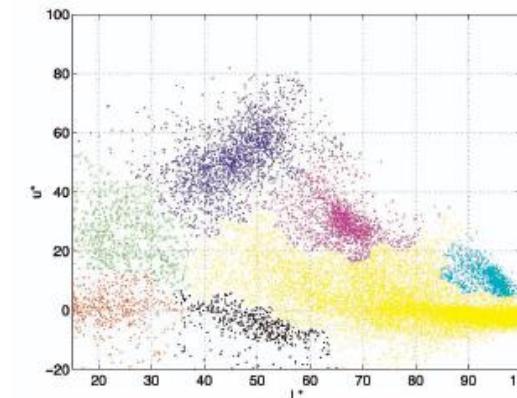
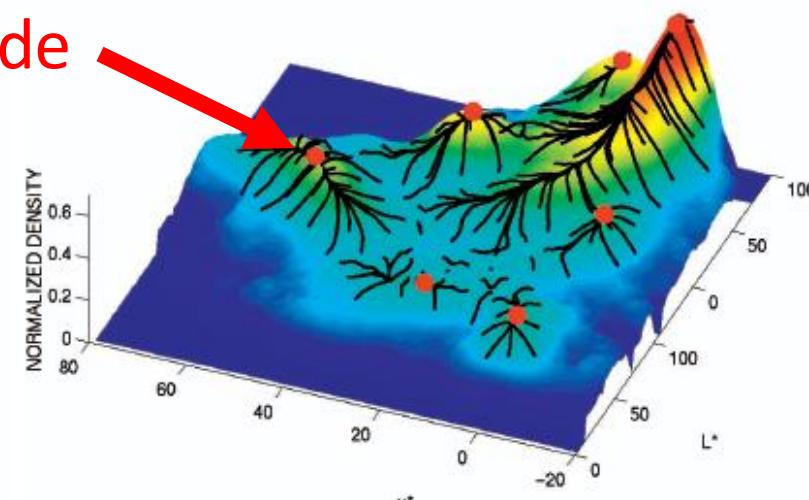
D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Algorithme *Mean shift*

- Tente de trouver les **modes** d'une **densité non paramétrique**



Mode

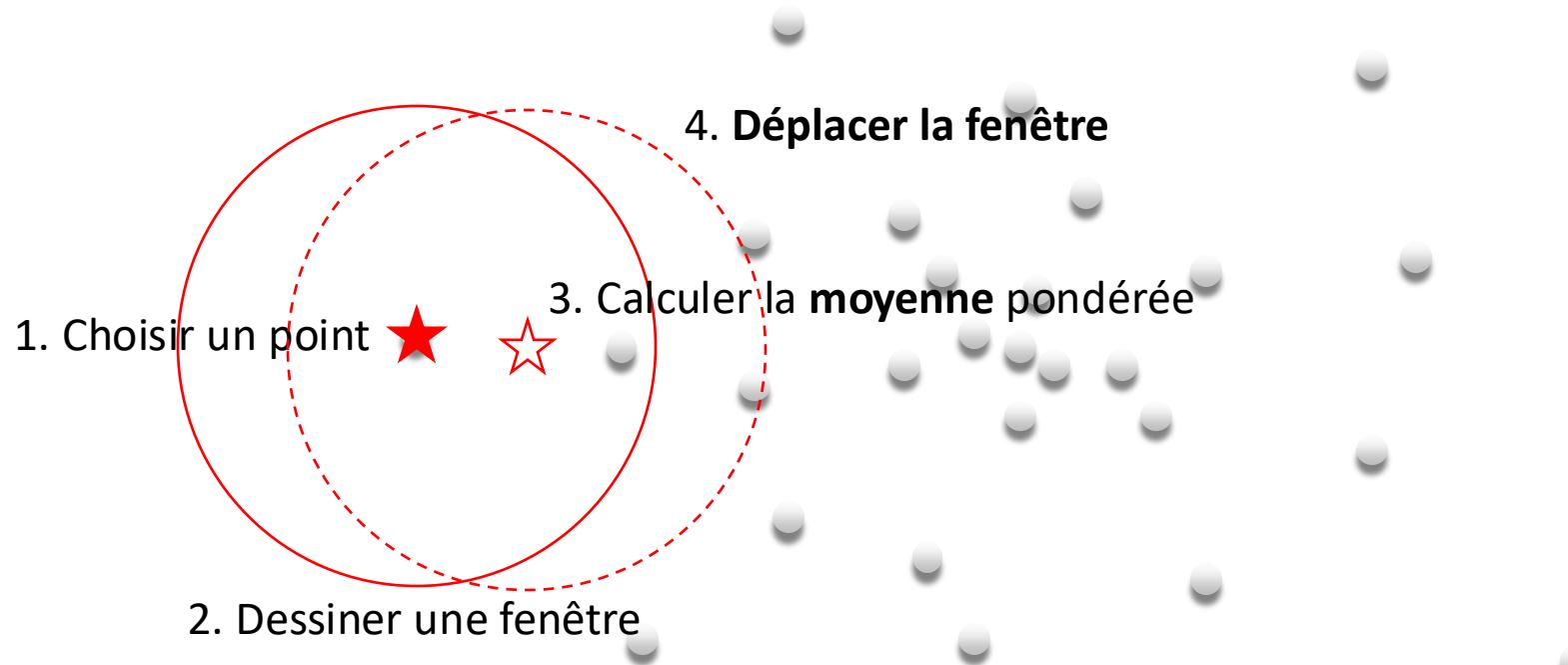


Source : CMU2019

Exemple : Algorithme *Mean shift*

Fukunaga & Hostetler (1975), A « mode seeking » algoritmm

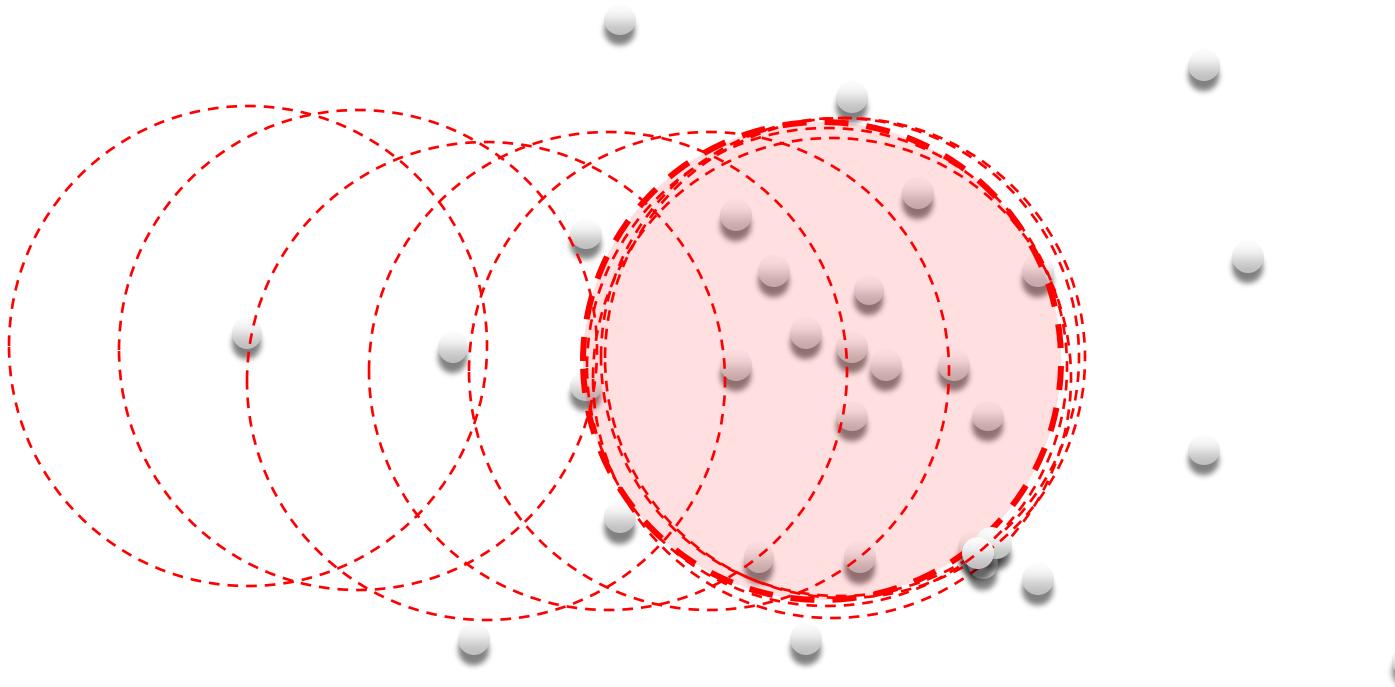
- Objectif : Trouver la région de plus haute densité



Exemple : Algorithme *Mean shift*

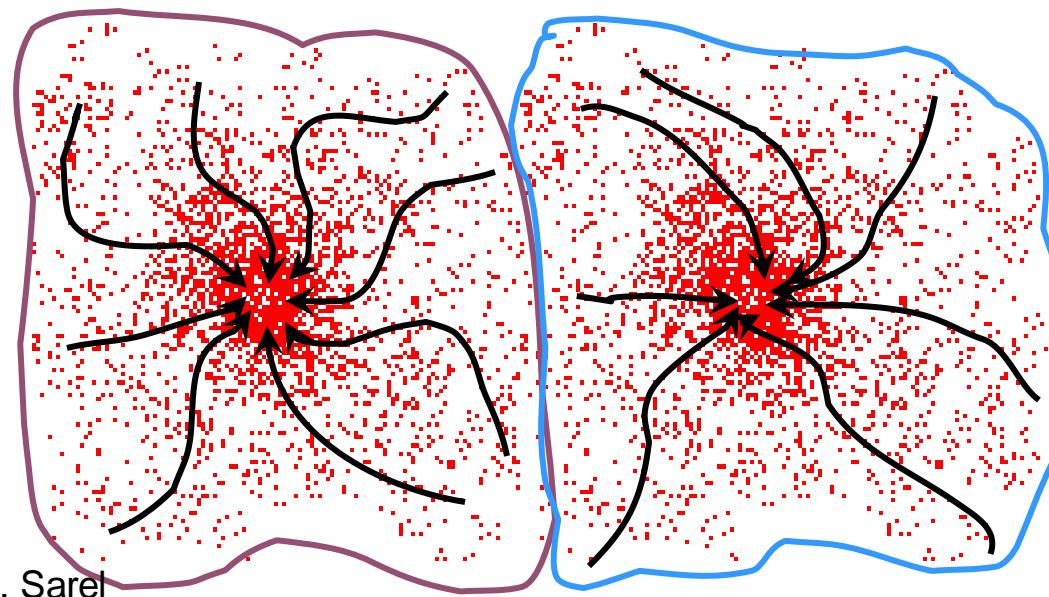
Fukunaga & Hostetler (1975), A « mode seeking » algoritmm

- **Objectif : Trouver la région de plus haute densité**



Bassin d'attraction

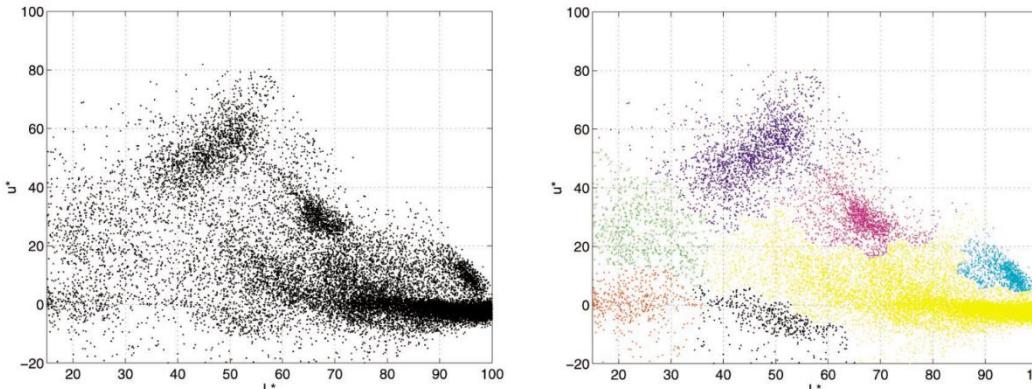
- **Bassin d'attraction** : la région pour laquelle toutes les trajectoires aboutissent au même mode
- **Partition** : tous les points inclus dans le bassin d'attraction d'un mode



Slide by Y. Ukrainitz & B. Sarel

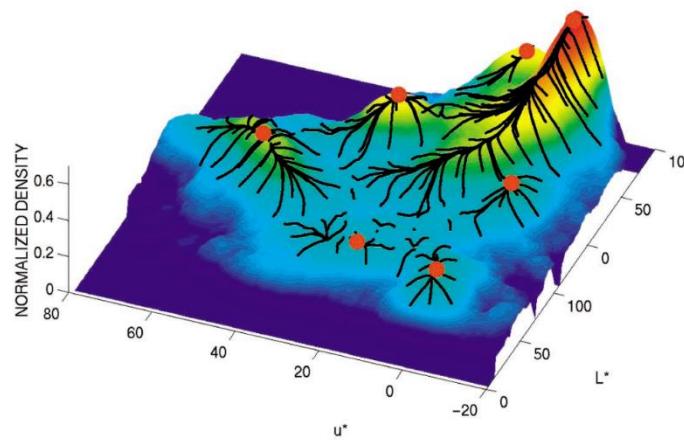
Source : CMU2019

Bassins d'attraction (image couleur)



(a)

(b)



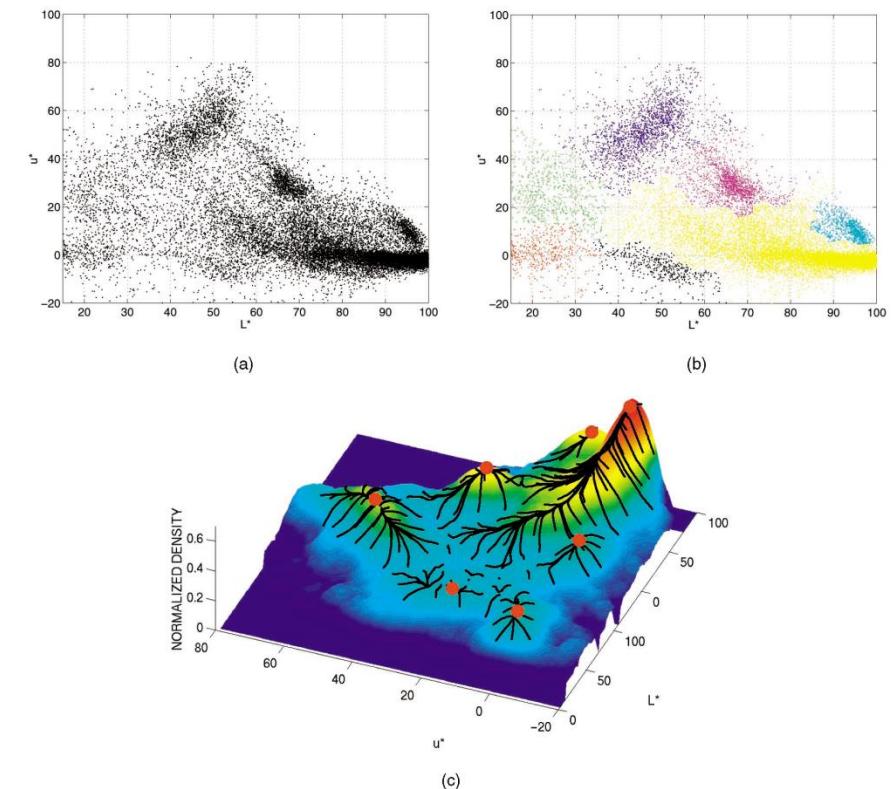
Algorithme : Partitionnement *Mean shift*

L'algorithme *Mean shift* cherche les modes pour un ensemble de points donnés

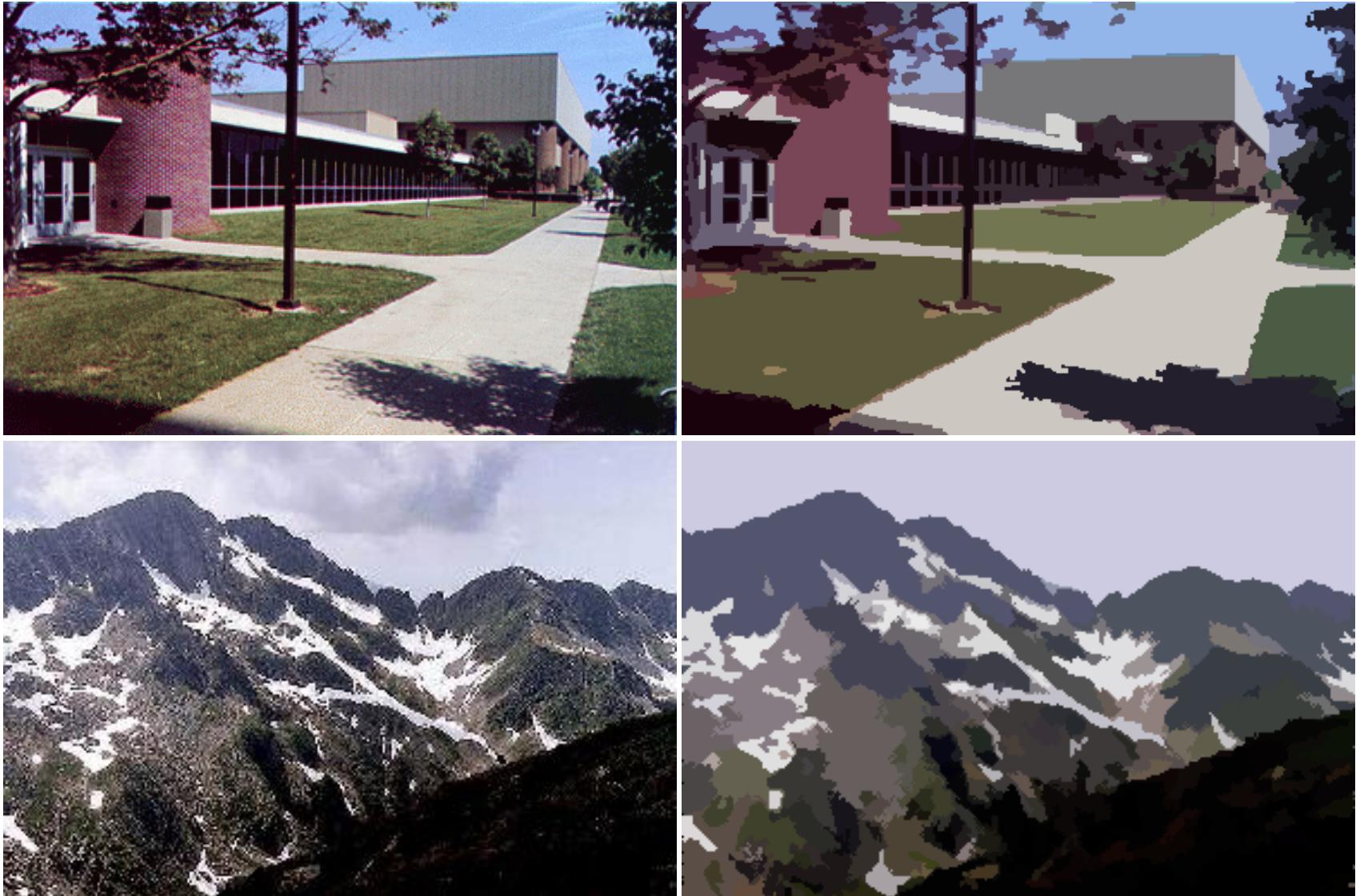
1. Choisir un noyau (*kernel*) et ses paramètres
2. Pour chaque point :
 - a) **Centrer** la fenêtre sur ce point
 - b) Calculer la **moyenne** des données dans la fenêtre
 - c) **Centrer** la fenêtre à la nouvelle position de la moyenne
 - d) **Répéter** (b,c) jusqu'à convergence
3. Assigner les points qui ont abouti aux mêmes modes et aux mêmes partitions

Segmentation par *Mean shift*

- Calculer des **propriétés** pour chaque pixel (couleur, gradient, texture, etc)
- Définir une **taille** de noyau pour une propriété Kf et une position Ks
- Initialiser les fenêtres à chaque position de pixels individuels
- Appliquer *mean shift* pour chaque fenêtre jusqu'à convergence
- Fusionner les fenêtres qui sont en dessous d'une distance de Kf et Ks.



Mean shift : Résultat de segmentation (1)



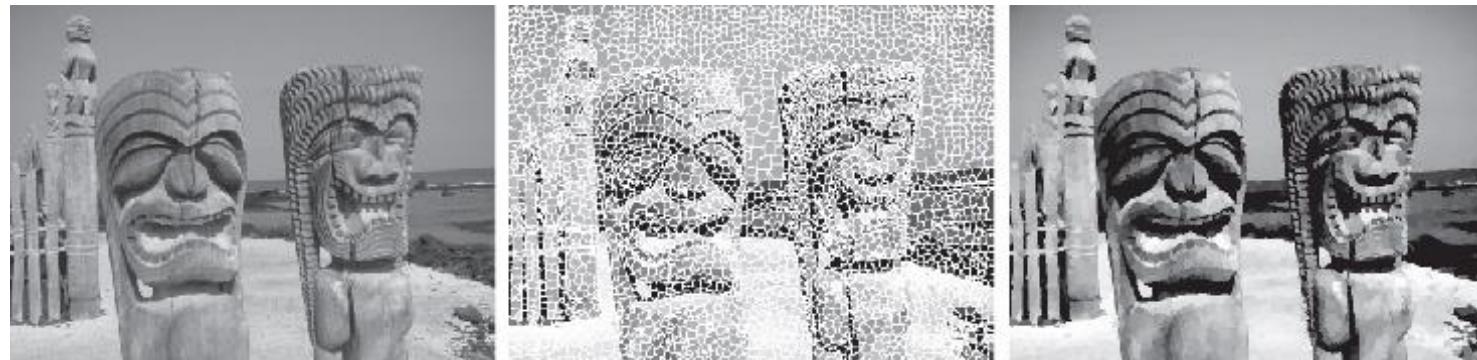
[Source](#)

Mean shift : Résultat de segmentation (2)



Segmentation par superpixel

- Remplacer les pixels par des groupes de pixels ayant une propriété commune : ***superpixel***
- Les superpixels doivent **se coller aux frontières**
- Plusieurs méthodes existent pour créer les superpixels (ex.: *Simple linear iterative clustering SLIC*)
- C'est une tâche de **partitionnement utilisant à la fois l'intensité** des pixels et leurs **positions**
- Disponible : ***skimage.segmentation.slic***



Exemple : Superpixel

N=1000



N=500



N=250



Segmentation par ligne de partage des eaux (Watershed)

- **Principe** : On considère une image à niveaux de gris comme un **relief topographique**. Le but est de trouver la ligne de partage des eaux de ce relief.

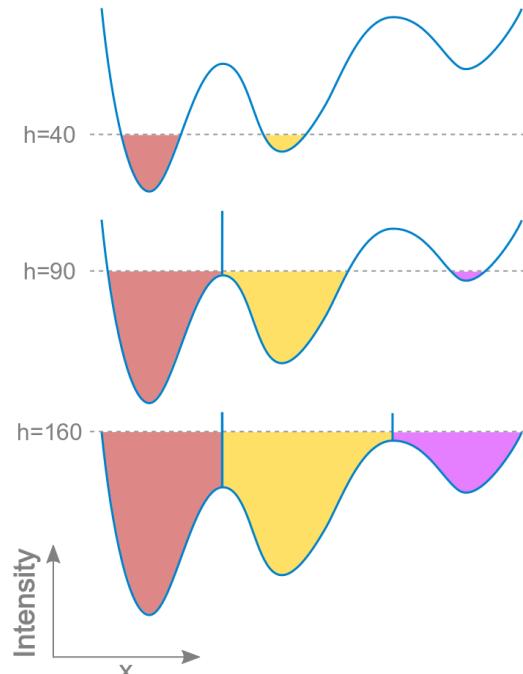
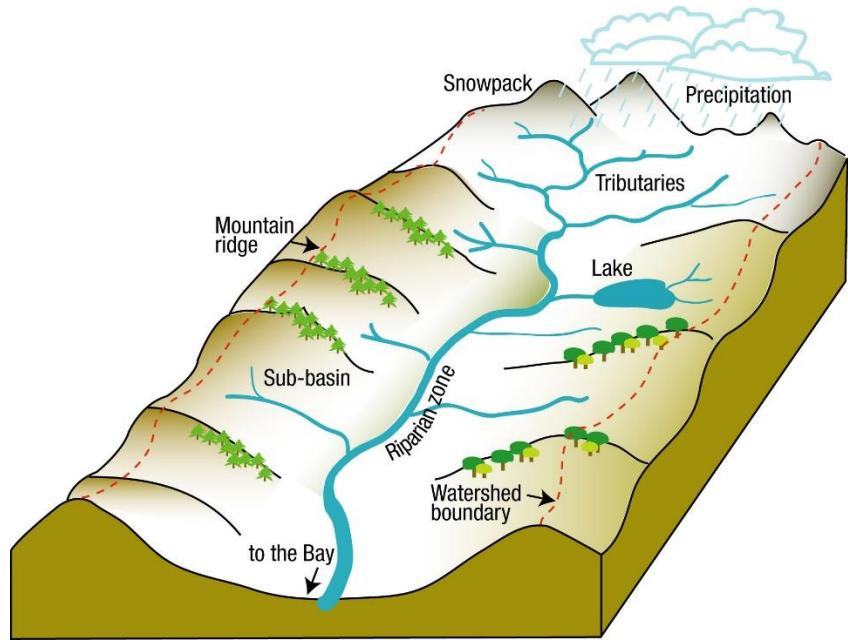
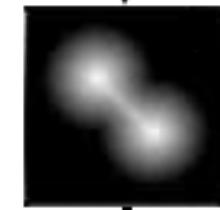
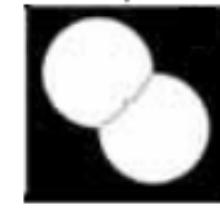


Image à segmenter



On calcule la distance entre chaque pixel du masque et l'arrière-plan



On trouve les « pics » de la carte de distance, puis on sépare les objets

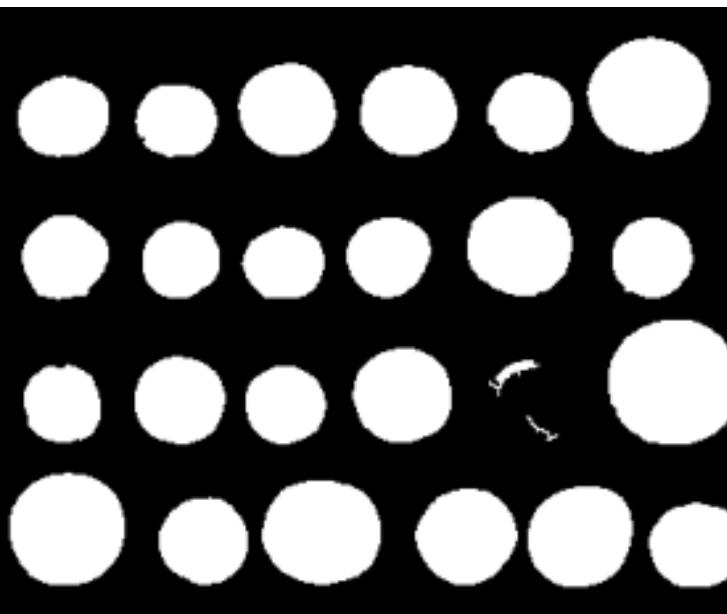
Exemple : *Watershed* v.s. *Canny*

- Démo : [scikit-image](#)

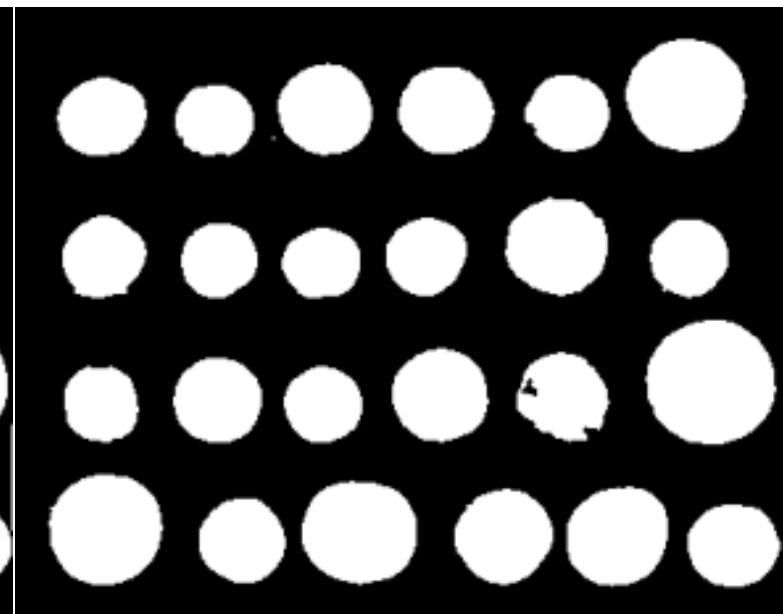
Image originale



Segmentation basée sur
les frontières (Canny)

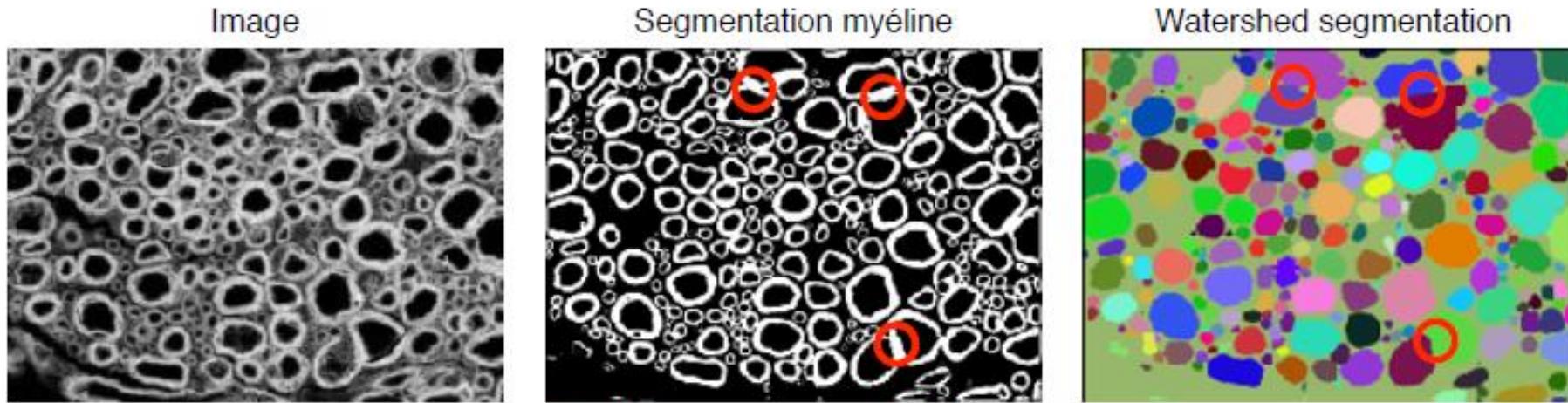


Segmentation basée sur
les régions (Watershed)



Exemple : Segmentation par ligne de partage des eaux (*Watershed*)

- *Segmentation et séparation des axones au sein d'images de microscopie électronique.*



Problème : axones adjacents se touchent. Il faut les séparer



AxonDeepSeg <https://github.com/neuropoly/axondeepseg>

Méthode des contours actifs (*snakes*)

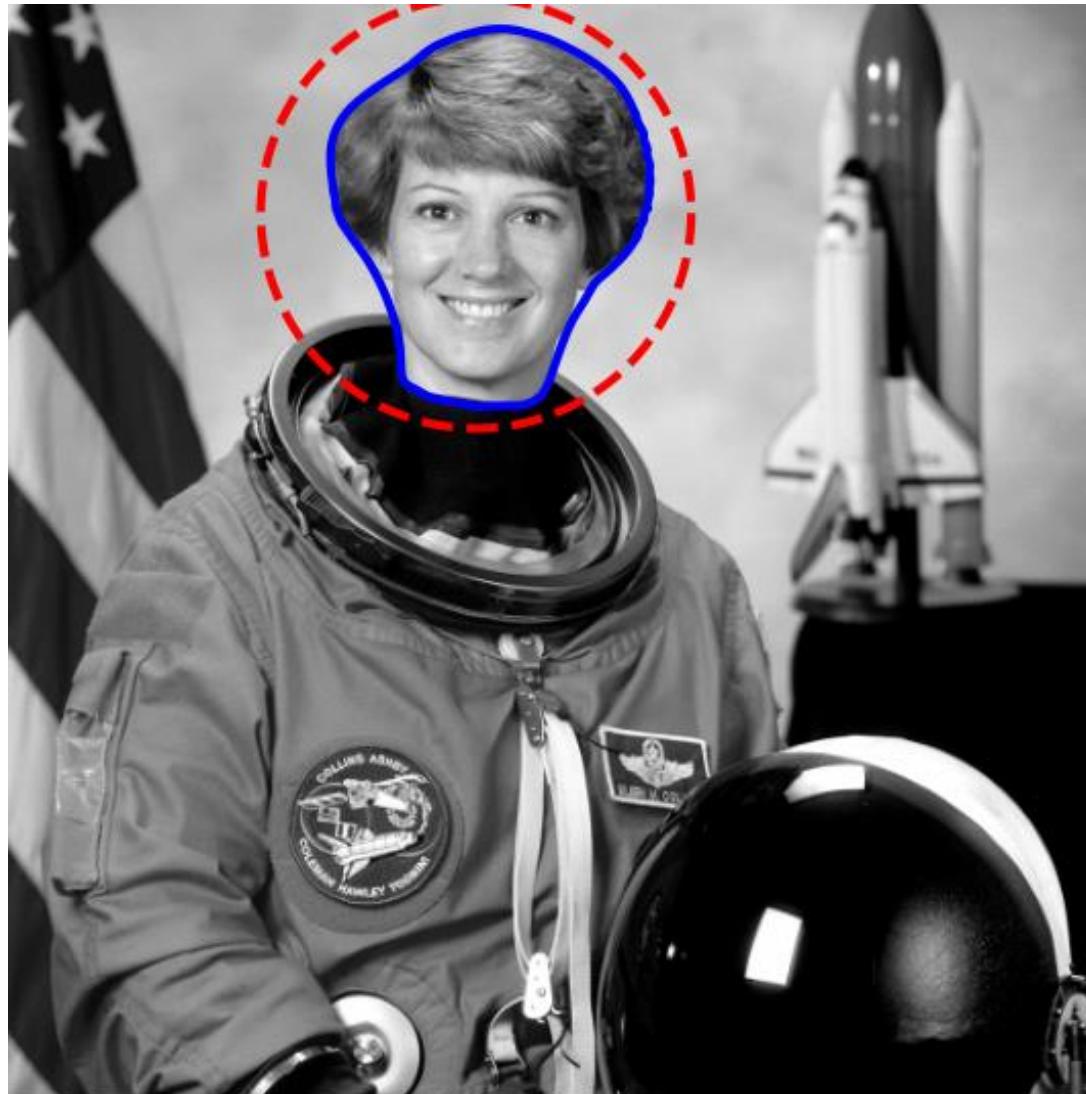
- **Contour actif** : On part d'un état initial, puis le contour évolue selon des contraintes (ex. gradients de l'image)
- **Méthode des snakes**: Utilise des splines pour décrire les contours.
- L'évolution des snakes est **contrainte** par : l'image (gradient) et les propriétés des snakes (longueur, courbure, etc.)
- Souvent utilisé en médecine pour trouver la **forme 3D** des tumeurs, os, etc. avant l'intervention.



<https://medpure.blogspot.com/2012/03/active-contour-in-c.html>

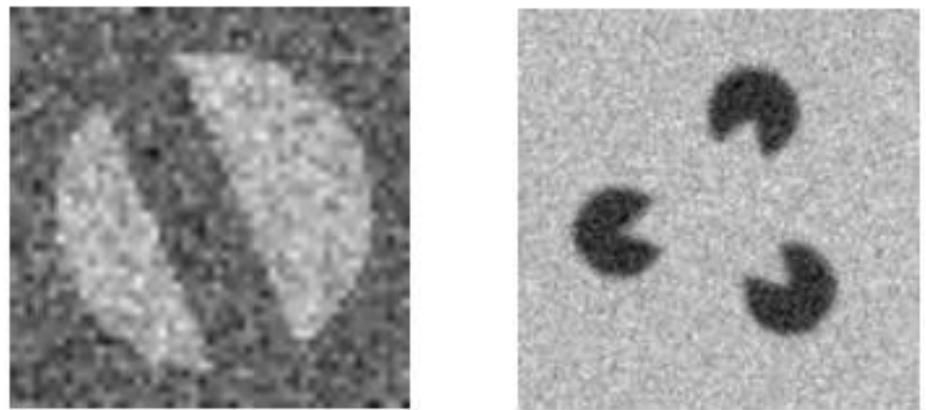
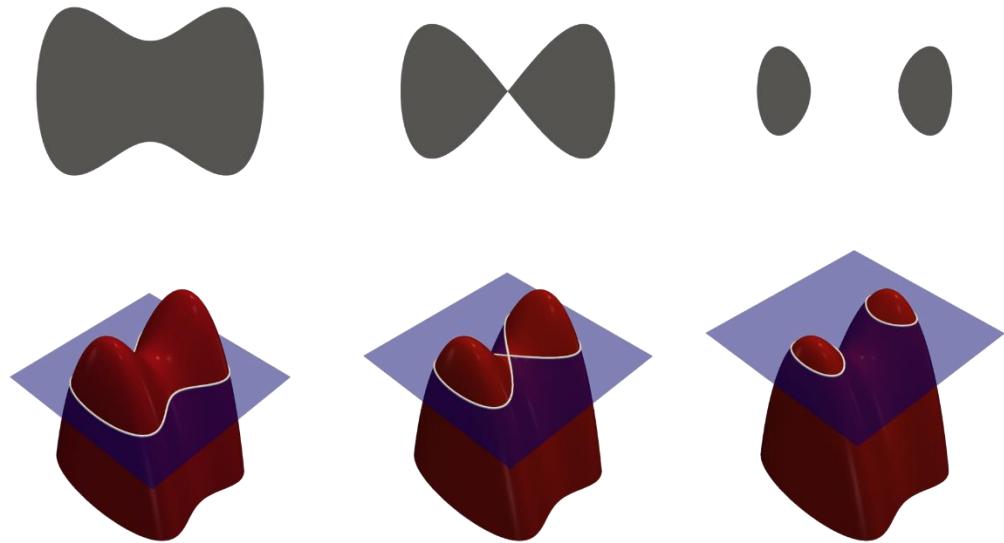
Exemple : Méthode des contours actifs

- Démo : [Scikit-image](#)



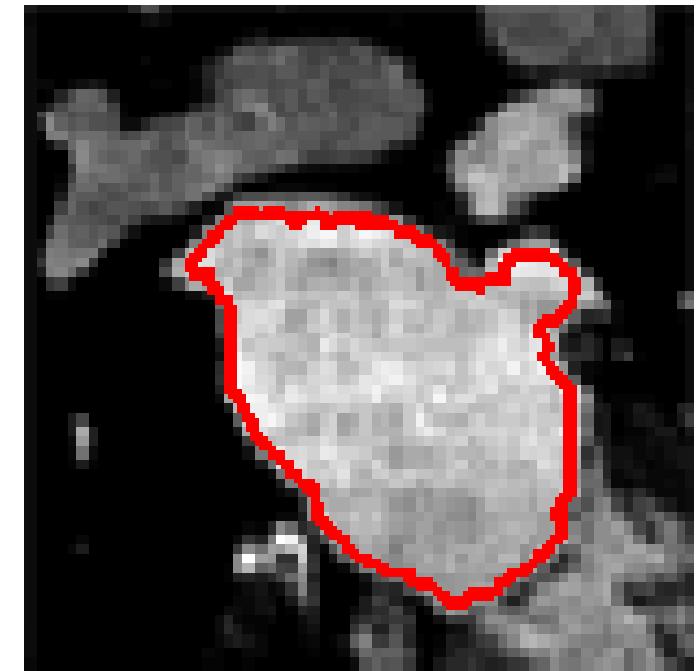
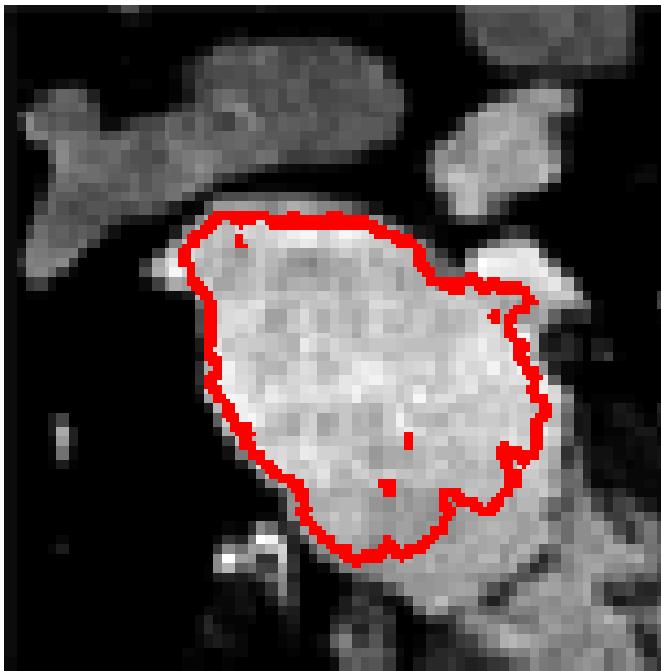
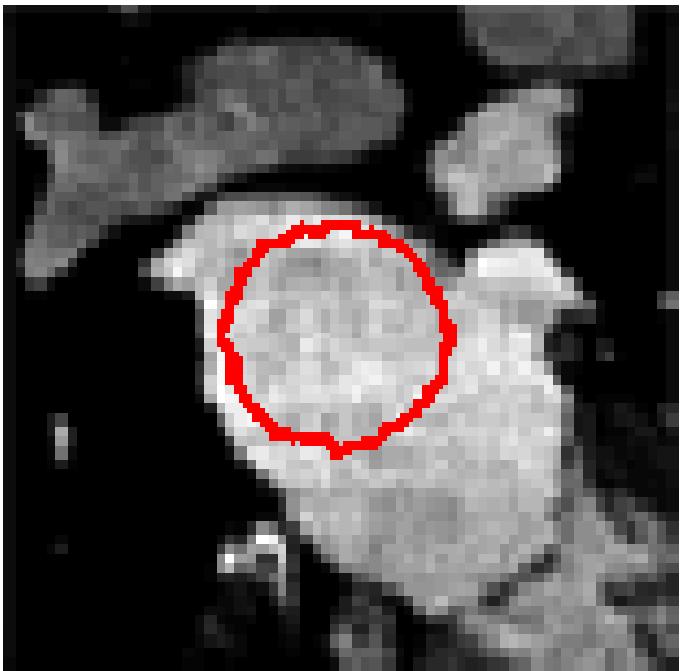
Méthode des surfaces de niveau (*level set*)

- **Principe** : Décrire l'évolution d'une région 2D en étudiant l'intersection entre un plan et un objet 3D.
- Avantage du *level set* par rapport au *snake* : Ne nécessite pas de paramétrisation de la courbe



<https://github.com/notmatthancock/level-set-machine-learning>

Exemple : Méthode des surfaces de niveau (*Segmentation de l'IRM cardiaque*)



<https://profs.etsmtl.ca/hlombaert/levelset/>

Résumé : Méthodes de segmentation

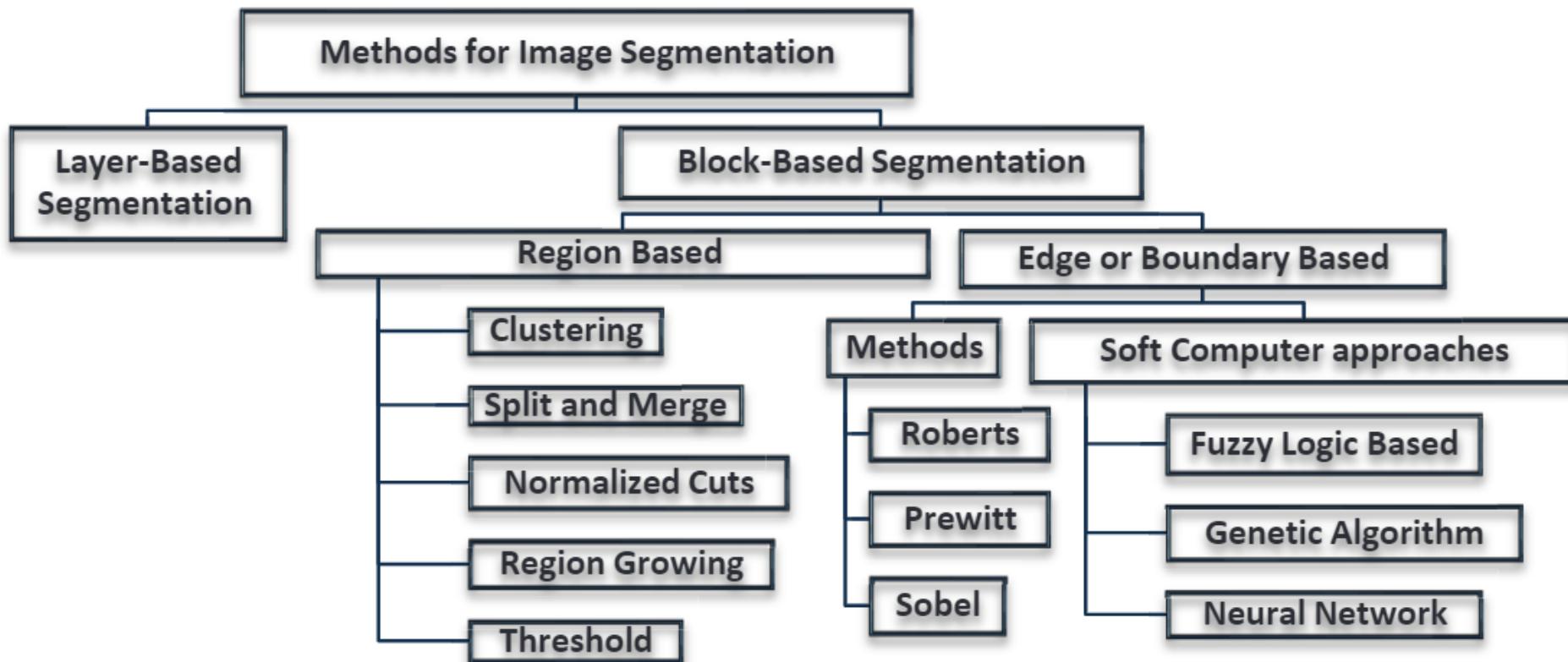


Fig. 2 Methods of Image Segmentation

Zaitoun, N., Aqel, M. (2015). Survey on Image Segmentation Techniques Procedia Computer Science 65(), 797-806. <https://dx.doi.org/10.1016/j.procs.2015.09.027>