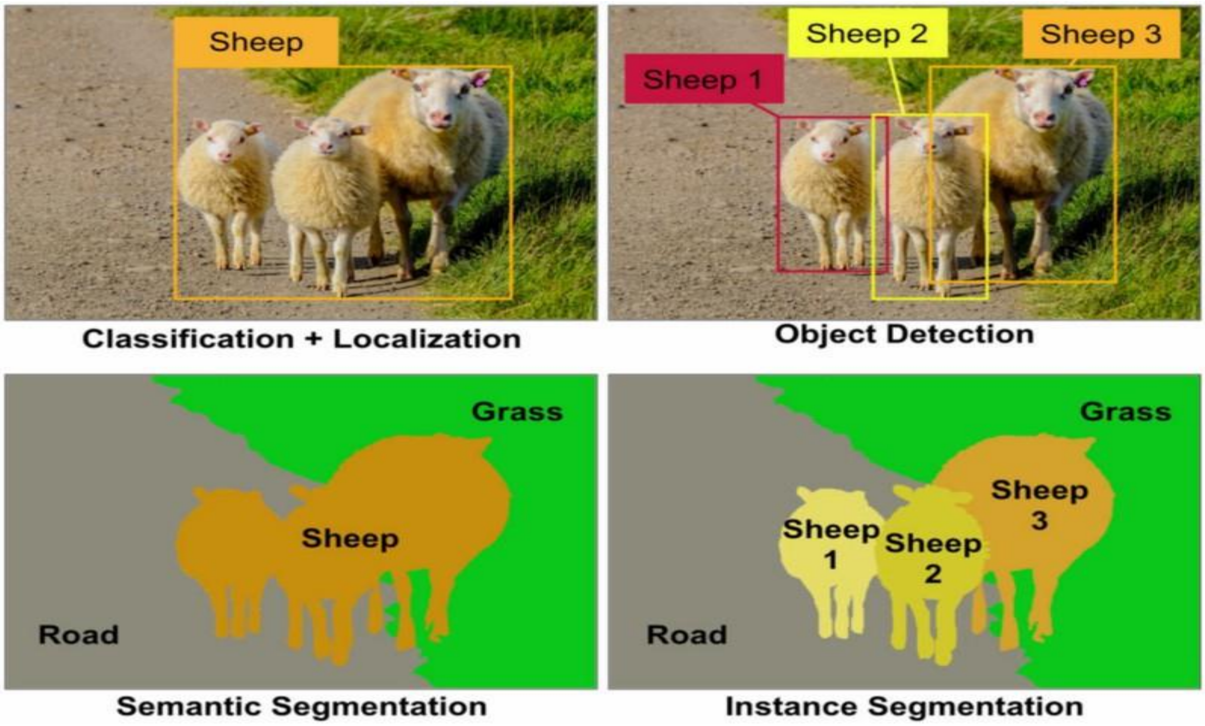
Chapitre 6

Segmentation et classification

6.1. Principe et Généralités

La **Segmentation** permet d'identifier et de définir des objets dans l'image qui est un ensemble de **régions** disjointes.



La Segmentation c'est la décomposition d'une image en régions qui ont un sens (les objets de l'image) :

- Segmentation = étiquetage des pixels de l'image.
- pixels de même étiquette = pixels de même région.

Il existe plusieurs approches :

– Région : regroupement des pixels présentant une caractéristique commune, dérivée par exemple de l'intensité des pixels. Grouper pixels semblables ce qui donne des régions homogènes.

– Contour : mise en place des frontières aux positions qui rendent localement maximale la variation d'un critère. Rechercher pixels dissemblables pour avoir des contours/surfaces entre zones hétérogènes.

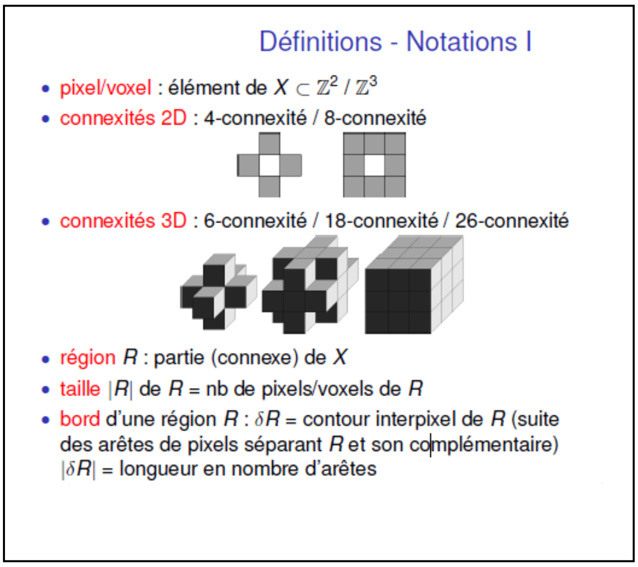
Ces deux conceptions sont duales : une région définit son contour, un contour définit une région.

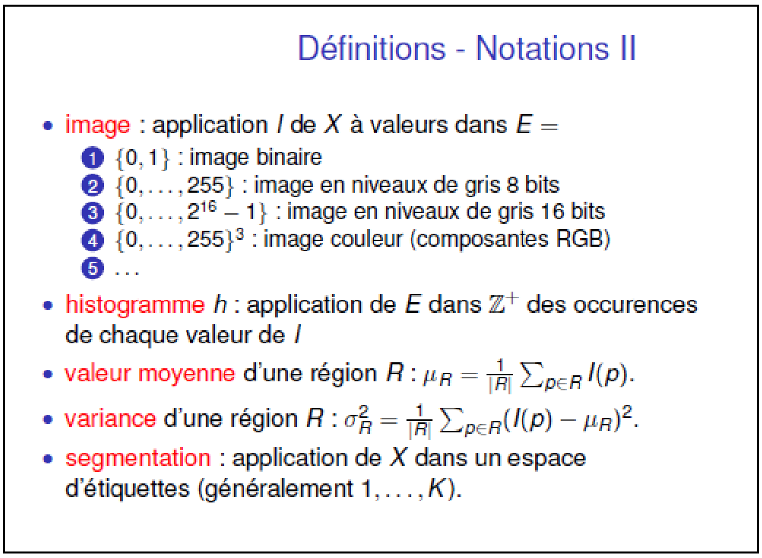
- Autres approches : watersheds, Mumford Shah, modèles déformables, level sets, champs de Markov

Dans la segmentation, on distingue trois techniques :

- Méthodes statistiques : Segmentation par sélection récursive sur histogrammes
- Méthodes géométriques : croissance de région (Region growing) et par fusion de région (Split & Merge)

- Méthodes par optimisation : Définition d'une fonctionnelle de coût.





6.2. Différentes approches de segmentation

6.2.1. Méthodes fondées sur l'histogramme

Ces méthodes consistent à trouver les différents modes de l'histogramme, représentatifs d'autant de classes d'objets dans l'image.

Les méthodes par histogrammes sont en général :

● Rapides à calculer

● Peu sensibles au bruit

Mais elles n'intègrent pas (ou peu) d'information géométrique ou topologique sur les régions. Ce sont des méthodes **globales**, au sens où la décision d'appartenance d'un pixel à une région dépend toujours de l'image entière. On détaillera ce type de segmentation dans la section seuillage d'une image.

6.2.2 Approches par régions

Ce sont des méthodes *locales*, pour lesquels l'étiquetage d'un pixel dépend uniquement, ou essentiellement, de son voisinage. L'exemple caractéristique est la segmentation par croissance de région (*region growing*). Ces algorithmes intègrent naturellement les propriétés topologiques, mais aussi parfois géométriques, des régions.

6.2.2.1. Segmentation par croissance de région(Region growing)

Tout d'abord, on initialise la région R à un groupe de pixels (*seed*). Cette région R possède certains moyenne μ_R et écart-type σ_R. Puis, on lui ajoute tous les pixels voisins qui lui sont suffisamment semblables, par exemple qui vérifient : $|\,I(x) - \mu_R| <$ seuil

On peut également ajouter des critères géométriques de régularité, comme par ex :

R ・V(x) est de cardinal au moins 3 et possède une seule composante connexe.

6.2.2.2. Segmentation par division et fusion de région (Split & Merge)

L'idée des algorithmes de type « Split & Merge » est de produire automatiquement une partition initiale en régions petites (Split), qui vont ensuite croître en se regroupant (Merge).

La partition initiale (Split) est réalisée en divisant récursivement l'image en régions de tailles identiques lorsqu'un certain critère d'homogéneité n'est pas satisfait (par ex: R est divisée si $\sigma_R > seuil$).

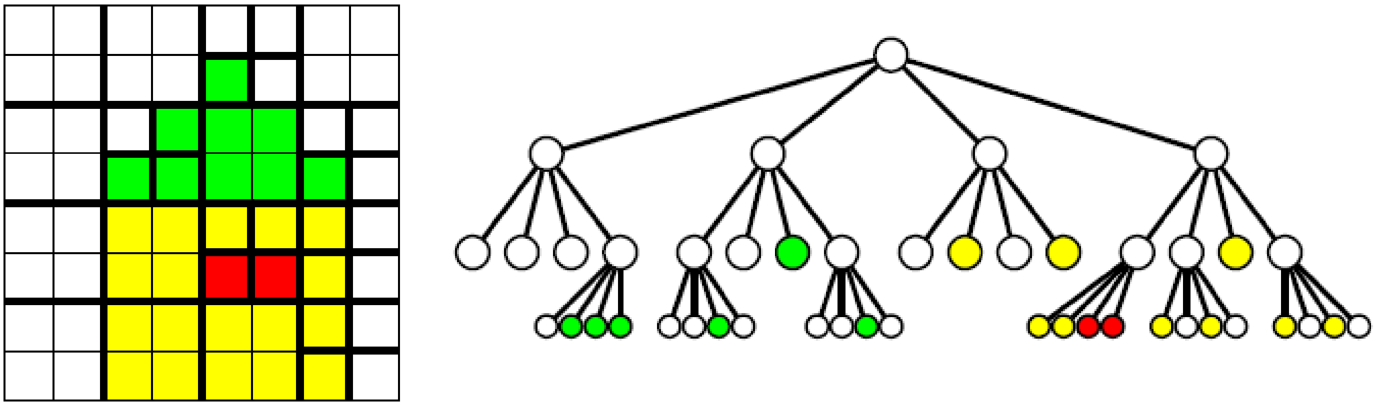
Lors de cette phase, le graphe d'adjacence, ou Region Adjacency Graph (RAG) est créé : à chaque région est associé un sommet du graphe, et des arêtes relient les sommets correspondants à deux régions qui se touchent.

La phase de regroupement (Merge) utilise le RAG pour modifier la partition initiale : pour chaque sommet R du RAG, on cherche s'il existe un sommet R' voisin dans le RAG et de valeur suffisamment proche, et si c'est le cas, on les fusionne (par ex: R et R' sont fusionnées si $|\mu_R - \mu_{R'}| <$ seuil).

Split & Merge: Quadtree split

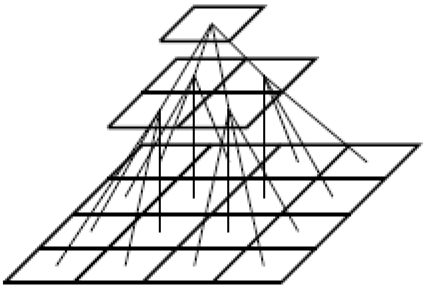
Split : la géométrie du découpage a une grande influence sur le résultat de la segmentation. Par exemple, le split en quadtree fait apparaître des régions carrées. D'autres types de découpage existent, éventuellement redondants (pyramides avec recouvrement).
Merge : l'ordre dans lequel est réalisé le regroupement des régions a aussi une influence sur le résultat.
Les méthodes de division et fusion partent de l'échelle la plus grossière vers les détails les plus fins.

Division par quadtree (Approche top-down)
L'image est codée par un arbre à définition récursive :
- racine : image entière
- découpage en 4 de la portion d'image correspondant à un nœud si elle n'est pas homogène et chaque nœud a 4 fils.



C'est une méthode de segmentation par division basée sur un critère d'homogénéité (et pas sur l'histogramme).

- le quadtree est un cas particulier de pyramide, une pyramide étant une suite de graphes représentant une image à différents niveaux de résolution.
- inconvénient : les découpes sont contraintes par le cadre rigide de la définition.
- utilisable comme partition initiale pour une méthode de fusion.



Pyramide rigide

Fusion (Approche bottom-up)

On part d'une partition de l'image en régions homogènes et on fusionner tout couple de régions adjacentes qui vérifie un critère d'homogénéité.

Soient Ri et Rj deux régions adjacentes dont on veut tester leur homogénéité pour la fusion alors on peut simplement mesurer l'homogénéité de la région comme suit :

Critère de [Beveridge89] pour la fusion

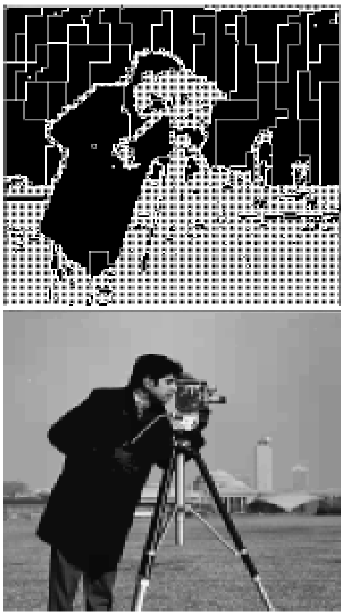
On peut simplement mesurer l'homogénéité de la région $R = R_i \cup R_j$:

- la variance des n. g. des pixels de R est inférieure à un seuil
- la proportion des pixels de R dont le n. g. est à l'extérieur de $[\mu_R - \sigma_R, \mu_R + \sigma_R]$ est inférieure à un seuil
- autres ?

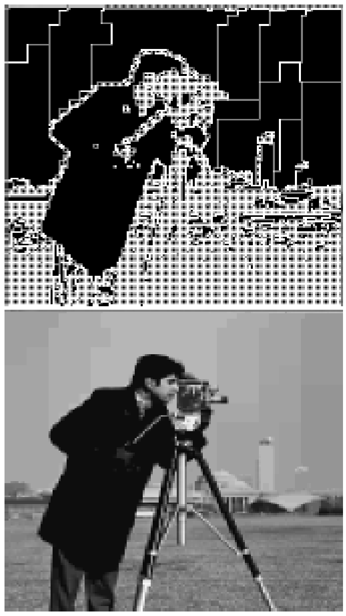
$$f(R_i, R_j) = f_{sim}(R_i, R_j)\sqrt{f_{taille}(R_i, R_j)}f_{cont}(R_i, R_j)$$

- Critère de similarité :
  $f_{sim}(R_i, R_j) = \frac{|\mu_i - \mu_j|}{max(1, \sigma_i + \sigma_j)}$
- Critère de taille :
  $f_{taille}(R_i, R_j) = min(2, \frac{min(|R_i|, |R_j|)}{T_{opt}})$, T_{opt} : fixé suivant la taille de l'image
- Critère de frontière commune :
  $$f_{cont}(R_i, R_j) = \begin{cases} C(R_i, R_j) & \text{si } \frac{1}{2} \leq C(R_i, R_j) \leq 2 \\ \frac{1}{2} & \text{si } C(R_i, R_j) < \frac{1}{2} \\ 2 & \text{sinon} \end{cases}$$
  $$C(R_i, R_j) = \frac{min(|\delta R_i|, |\delta R_j|)}{4|\delta R_i \cap \delta R_j|)}$$

Exemple :Division / fusion



après division



après fusion

Algorithmes du Split & Merge

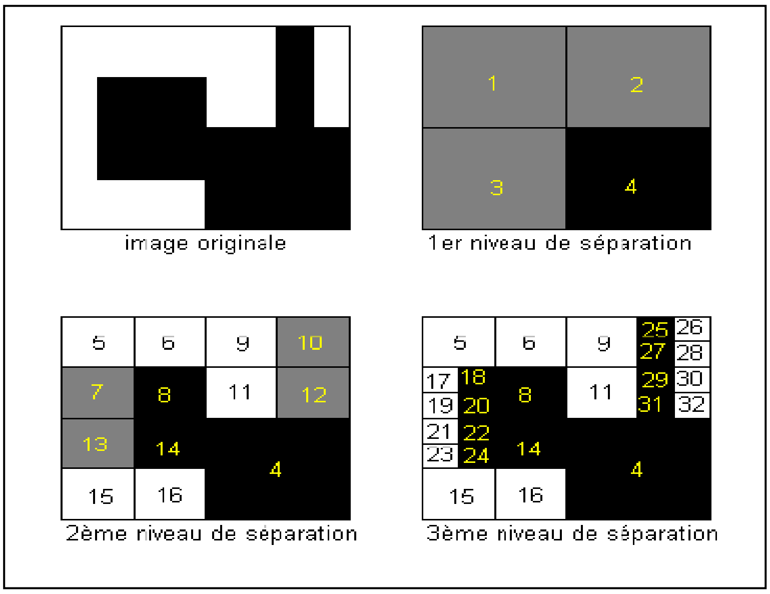
- Méthode par séparation(division)

Elle est basée sur la technique du Quad Tree : l'image est supposée carrée : on la divise en 4 quadrants, puis chaque quadrant en 4 sous-quadrants et ainsi de suite.

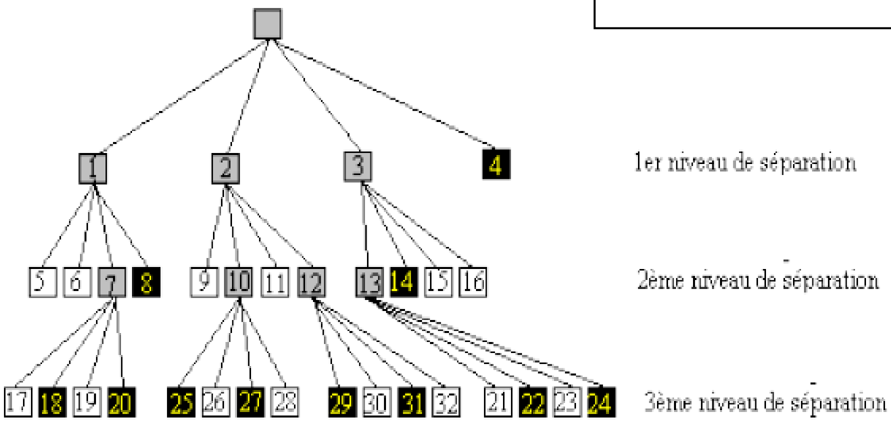
L'algorithme de division est simple et récursif :

Separation(zone)
Si critere(zone)=VRAI alors
classer
sinon
diviser zone en 4 sous-zones Z1,Z2,Z3,Z4
Pour i=1 à 4 faire :
Separation(Zi)

fin Pour
fin Si



et le quad tree est :

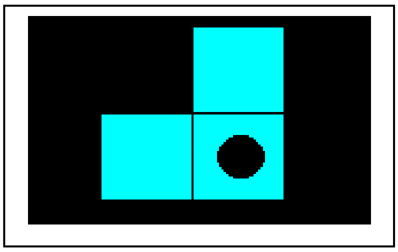


- **Méthodes par fusion**

Alors que la méthode précédente consistait à diviser l'image, les méthodes par fusion appliquent le principe inverse : on fait croître une région en y incorporant les régions voisines ayant les mêmes critères d'homogénéité (fusion de région). La règle de fusion est double : les deux régions correspondent au même critère et elles sont adjacentes.

La méthode de la coloration de tâches consiste à promener sur l'image une fenêtre de trois points. En chaque point on applique l'algorithme indiqué ci-dessous où Couleur est un niveau de couleur attribué aux pixels satisfaisant le même critère.

```
Pour chaque point (x,y) faire :
  Si critere(x,y)=critere(x-1,y) alors
       Couleur(x,y)=Couleur(x-1,y)
  sinon
     si critere(x,y)=critere(x,y-1) alors
         Couleur(x,y)=Couleurx,y-1)
     sinon
         Couleur(x,y)=Nouvelle_Couleur
     fin Si
   fin Si
fin Pour
```



La méthode locale récursive consiste à faire croître au maximum une région avant de passer à la suivante. L'algorithme est décrit ci-dessous :

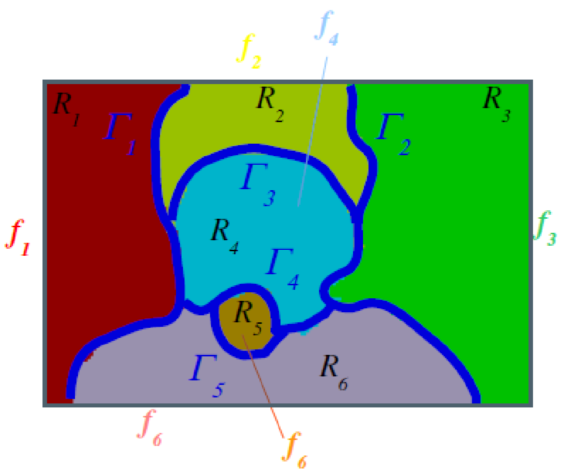
```
Pour chaque pixel (x,y) faire :
           Si I(x,y) <> 0 alors
               sauvegarder le point de départ x,y
              croissance(x,y)
              incrémenter NuméroSegment
         fin Si
 fin Pour
croissance(x,y)
       I(x,y)=0 // pour ne pas réexaminer(x,y)
       Pour tout pixel adjacent à (x,y) faire :
           Si I(x,y)<>0 et critere(pixel)=critere de départ alors
                croissance(pixel)
            fin Si
fin Pour
```

6.2.3.Méthodes par optimisation

Dans les méthodes par optimisation, le problème de la segmentation est formalisé par l'estimation d'une fonction *f* bidimensionnelle qui doit avoir certaines propriétés : régulière, constante par morceaux, à bords réguliers, etc, tout en étant « proche » de l'image analysée I. On recherche un compromis entre ces différentes propriétés antagonistes, en minimisant une fonctionnelle de coût *K* qui va dépendre de :

- I l'image analysée.
- $\{R_i\}_{i\in P}$ la partition (segmentation) calculée.
- $\{\Gamma_j\}_{j\in Q}$ les courbes frontières (contours) associées à la segmentation.
- f la fonction recherchée, représentant l'image I segmentée. La fonction f est représentée par ses restrictions f_i sur chaque région R_i, soit :

$$f \equiv \{f_i\}_{i\in P}$$



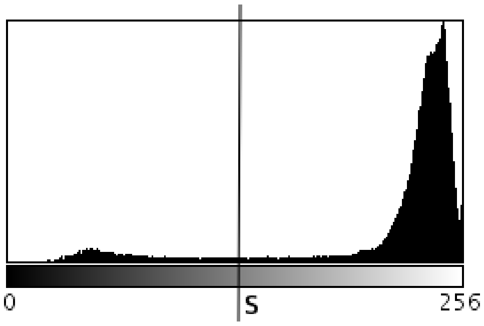
La segmentation en régions homogènes consiste à considérer une région comme un ensemble de points possédant la même propriété (homogénéité). Les critères d'homogénéité peuvent être la couleur ou le niveau de gris, la texture, ...

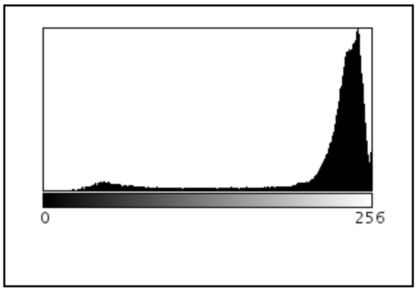
6.3. Seuillage et segmentation d'images

6.3.1. Seuillage et histogramme

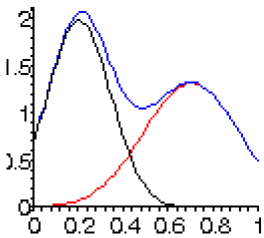
L'histogramme de I dans R € X représente la distribution des valeurs des niveaux de gris dans R. Par exemple l'histogramme d'un objet assez uniforme se rapproche d'une gaussienne de variance faible.

Un histogramme bimodal correspond à une partie de X avec deux objets de moyennes différentes

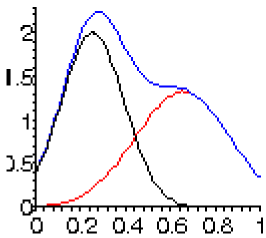




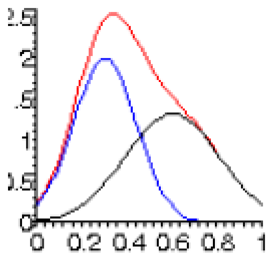
Le seuillage à trouver le(s) seuil(s) qui sépare(nt) au mieux les deux objets (ou plus) mai devient difficile lorsque les moyennes se rapprochent.



$(\mu_1 = 0.2, \sigma_1 = 0.2)$ $(\mu_2 = 0.7, \sigma_2 = 0.3)$



$(\mu_1 = 0.25, \sigma_1 = 0.2)$ $(\mu_2 = 0.65, \sigma_2 = 0.3)$



$(\mu_1 = 0.3, \sigma_1 = 0.2)$ $(\mu_2 = 0.6, \sigma_2 = 0.3)$

6.3.2.Segmentation par seuillage

Ce type de segmentation consiste à affecter chaque pixel d'une image en niveaux de gris à une classe (intervalles de niveaux de gris). Elle est basée sur l'extraction des seuils à partir de l'histogramme (image/région) et classification d'un pixel p par comparaison de I(p) aux seuils

Pour un pixel (x,y) de niveau de gris I(x, y) et de propriété locale P(x; y), le seuil utilisé pour le classer est : S(x, y). il existe 3 types de méthodes de seuillage :

- seuillage global : $S(x,y) \stackrel{\text{def}}{=} S(I(x,y))$
- seuillage local : $S(x,y) \stackrel{\text{def}}{=} S(I(x,y),P(x,y))$
- seuillage dynamique : $S(x,y) \stackrel{\text{def}}{=} S(I(x,y),P(x,y),x,y)$

Si on a 2 classes seulement, on parle de binarisation.

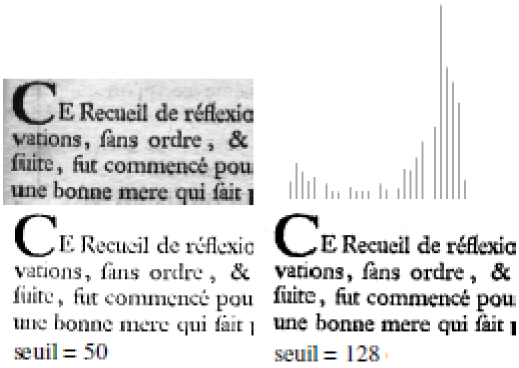
a) Exemple de seuillage global

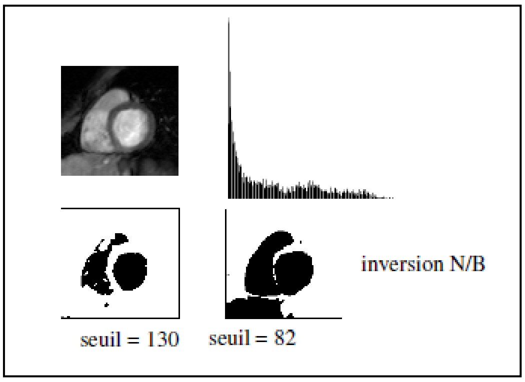
Binarisation [Otsu79] : On découpe de l'histogramme h de façon à minimiser l'erreur de partition. Son principe est basé sur la minimisation de la variance à l'intérieur de chaque classe (C1 et C2). Sachant que p(n) est la probabilité du niveau de gris n et **t** est le seuil on a :

$$p(C_1) = \sum_0^t p(n),\ \mu_{C_1} = \frac{\sum_0^t np(n)}{p(C_1)},\ \sigma^2_{C_1} = \frac{\sum_0^t (n-\mu_{C_1})^2 p(n)}{p(C_1)}$$
$$p(C_2) = \sum_{t+1}^N p(n),\ \mu_{C_2} = \frac{\sum_{t+1}^N np(n)}{p(C_2)},\ \sigma^2_{C_2} = \frac{\sum_{t+1}^N (n-\mu_{C_2})^2 p(n)}{p(C_2)}$$

$$\begin{aligned} \sigma^2_{intra} &= p(C_1)\sigma^2_{C_1} + p(C_2)\sigma^2_{C_2} \\ \sigma^2_{inter} &= p(C_1)p(C_2)(\mu_{C_1} - \mu_{C_2})^2 \end{aligned}$$

Minimiser σ^2_{intra} est équivalent à maximiser σ^2_{inter}





b) Exemple de seuillage local

Binarisation [Sauvola00]: le seuil est adapté au contraste local.

$$S(x,y) = \mu(x,y)\left(1 + k\left(\frac{\sigma(x,y)}{R} - 1\right)\right)$$

- $\mu(x,y)$ moyenne des ng dans la fenêtre de calcul.
- $\sigma(x,y)$ écart type des ng dans la fenêtre de calcul.
- valeurs standard des paramètres k = 0:5, R = 128

k=0.1, k=0.2, rayon de la fenêtre = 15

Binarisation [Bhanu01] : si un point appartient à Ci, la majorité de ses voisins (voisinage 8-connexe) appartient aussi à Ci

Algorithme itératif : μ seuil initial.

$$P^0_{C_1}(p) = \begin{cases} \frac{I(p)-\mu}{NgMax} + 0.5 & \text{si } I(p) > \mu \\ \eta\frac{I(p)-\mu}{NgMax} + 0.5 & \text{sinon } (\eta \text{ coef entre 0.5 et 1}) \end{cases}$$

$$P^{n+1}_{C_1}(p) = \begin{cases} (1-\alpha_1)P^n_{C_1}(p) + \alpha_1 & \text{si } Q^n_{C_1}(p) > 0.5 \\ (1-\alpha_2)P^n_{C_1}(p) & \text{sinon} \end{cases}$$

- α_1, α_2 coefs entre 0 et 1.

$\Rightarrow$ on renforce la probabilité pour p d'appartenir à C_1 si la probabilité d'appartenance à C_1 de son voisinage est forte.

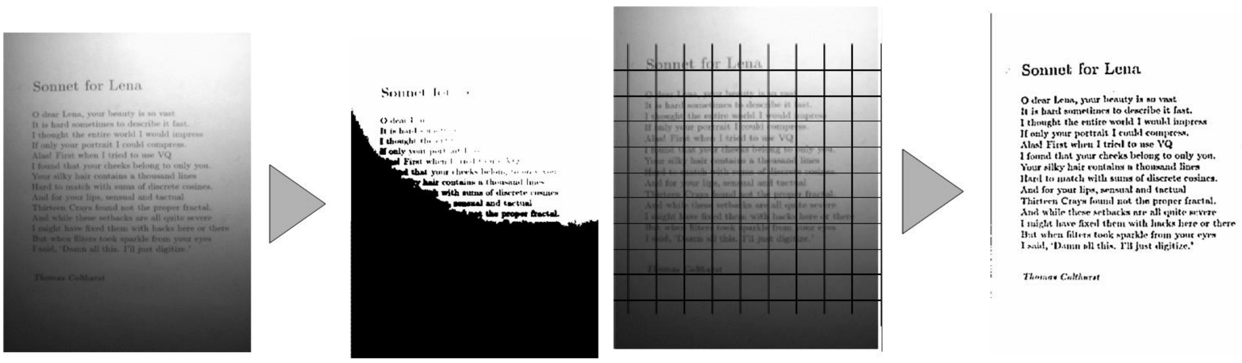
- Itérations successives jusqu'à majorité (90%) des pixels bien étiquetés ($P_{C_1}(p) > 0.9$ ou $P_{C_2}(p) > 0.9$)

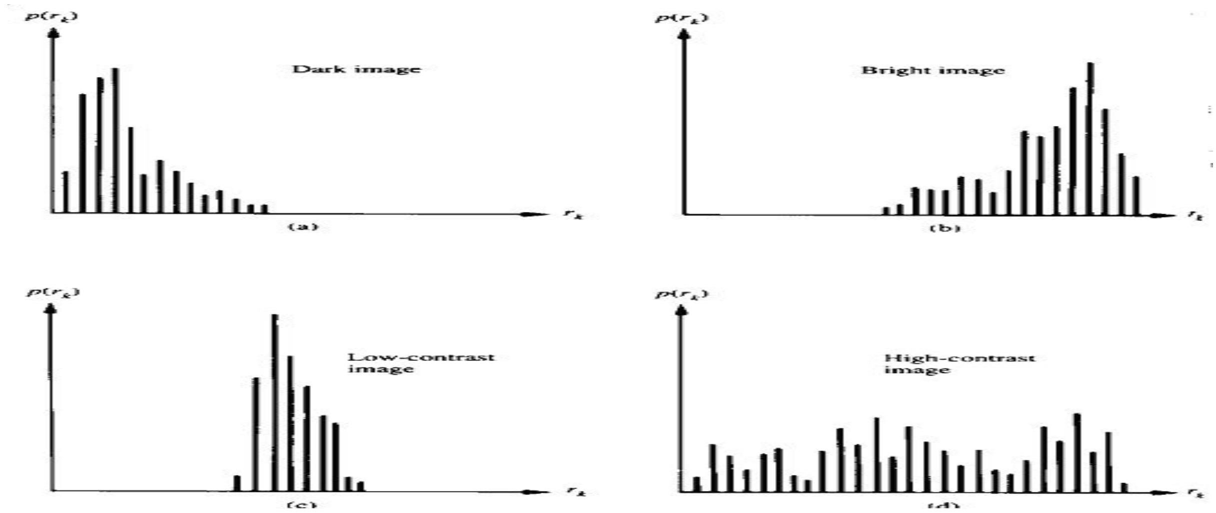
- $P(p) = (P_{C_1}(p), P_{C_2}(p))$ vecteur de probabilité associé à un pixel p
  $P_{C_i}(p)$: probabilité que le pixel p appartienne à C_i
  $P_{C_1}(p) + P_{C_2}(p) = 1$
- $Q(p) = (Q_{C_1}(p), Q_{C_2}(p))$ vecteur de compatibilité associé à un pixel p
  $Q_{C_i}(p) = \frac{1}{8}\sum_{q \in V_8(p)} P_{C_i}(q)$
- Critère à maximiser :
  $\sum_{image}(P_{C_1}(p)Q_{C_1}(p) + P_{C_2}(p)Q_{C_2}(p))$

c) Exemple de seuillage dynamique :Binarisation [Chow,Kaneko72]

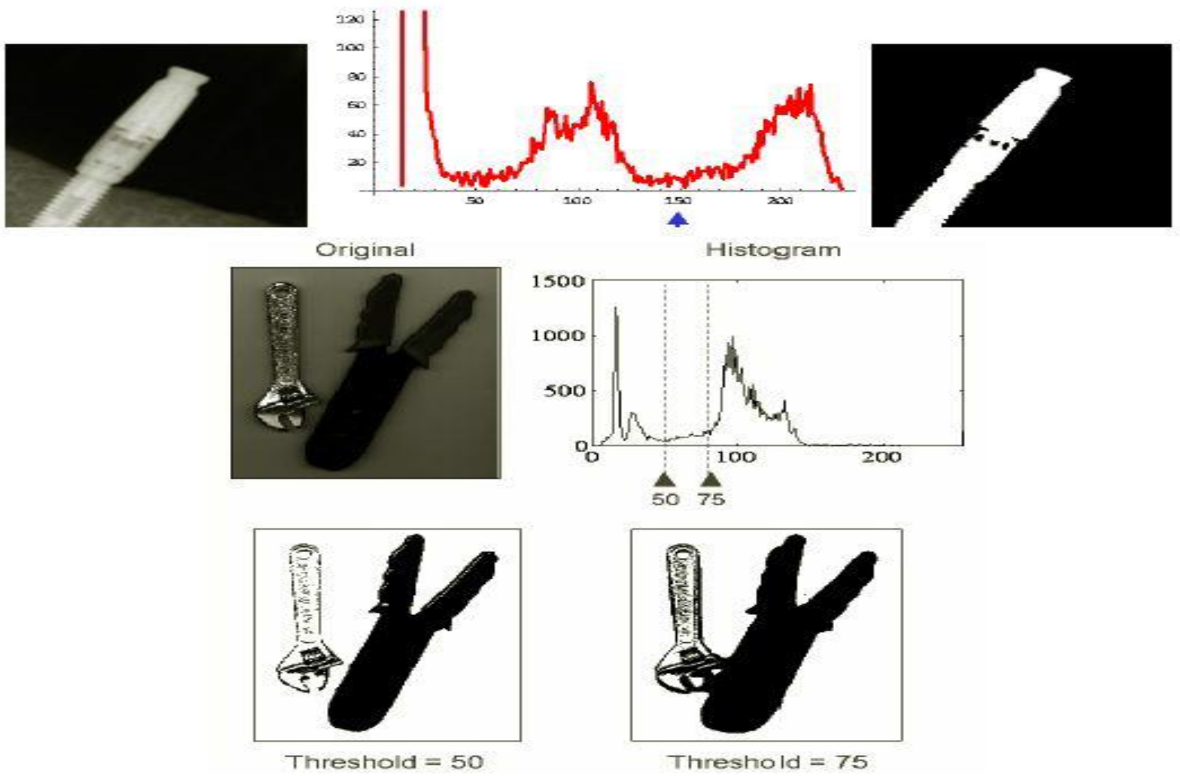
- Image découpée en blocs
- Calcul d'un seuil pour chaque bloc : l'histogramme du bloc est-il bimodal ?
- Si oui, le seuil trouvé est affecté au centre du bloc
- Si non, le seuil prend pour valeur la moyenne des seuils des blocs voisins.
- Le seuil de chaque pixel est calculé par interpolation bilinéaire à partir des seuils des centres des blocs voisins.

Problème lié au découpage en blocs : régions tronquées.





La figure ci-dessus montre la relation entre contraste et allure de l’histogramme.



Cette figure montre 3 situations qui sont :

- L’histogramme rouge où le seuil choisi est 150 a donné une image binaire comme résultat de séparation.
- L’histogramme noir :
  - Seuil=50 donne l’image en bas à gauche.
  - Seuil=75 donne l’image en bas à droite.

On voit clairement que le choix du seuil influence la qualité de séparation. Ce type de seuillage est appelé **seuillage global.**

c) Seuillage adaptatif : étude locale des critères

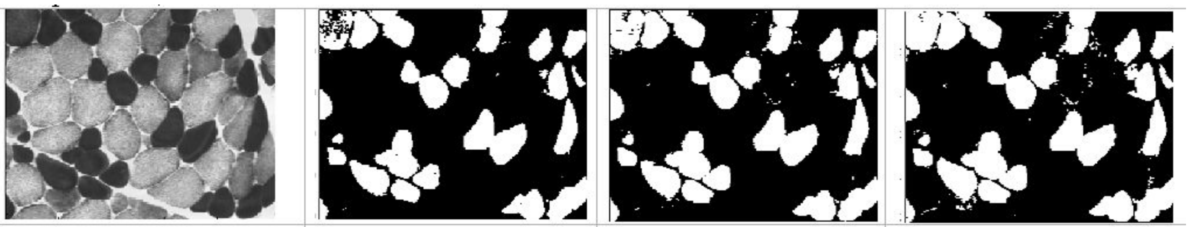


Image originale (cellules musculaires)	Seuillage 75	Seuillage 82	Seuillage 90 : seuil retenu

d) Seuillage optimal

On suppose que l'image contient deux régions de niveaux de gris différents c'est à dire l'histogramme est bimodal :

$$p(rk) \; = \; P_1.p_1(rk) \; + \; P_2.p_2(rk)$$

– p_i densité de probabilité des niveaux de gris dans la région i

– P_i constante proportionnelle à l'aire de la région i

Hypothèse : on connaît (ou on suppose connaître) p1 et p2, il est possible de trouver un seuil optimal de séparation ***des deux régions.***

e) Seuillage par hystérésis

S1-S2 : pixels rejetés *S2-S3 : pixels candidat* *S3-S4 : pixels acceptés*

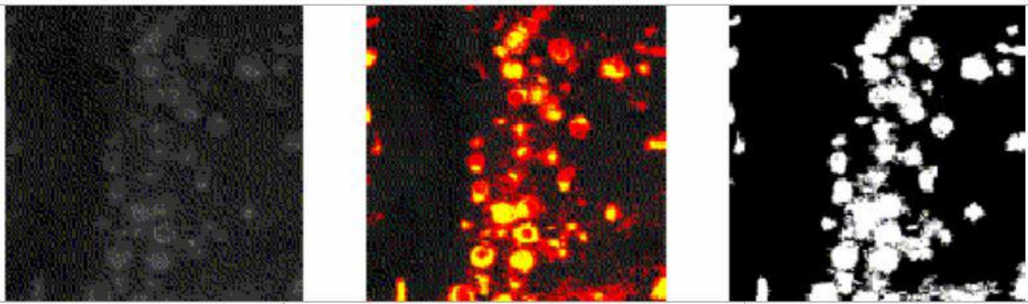


Image originale	En jaune : [S3-S4], en rouge [S2-S3]	Reconstruction

a) Seuillage avec apprentissage bayésien

Il est basé sur les hypothèses de connaissance sur les régions (cadre bayésien) et la stationnarité de l'image.

Pour deux populations X et Y, le seuil optimal de séparation est :

$$s/min\left(C_X\int_s^N P(X)P(n\,|X)dn + C_Y\int_0^S P(Y)P(n\,|Y)dn\right)$$

– $P(n|X)$, $P(n|Y)$: Probabilité conditionnelle qu'un pixel de X (Y) ait pour niveau de gris n.

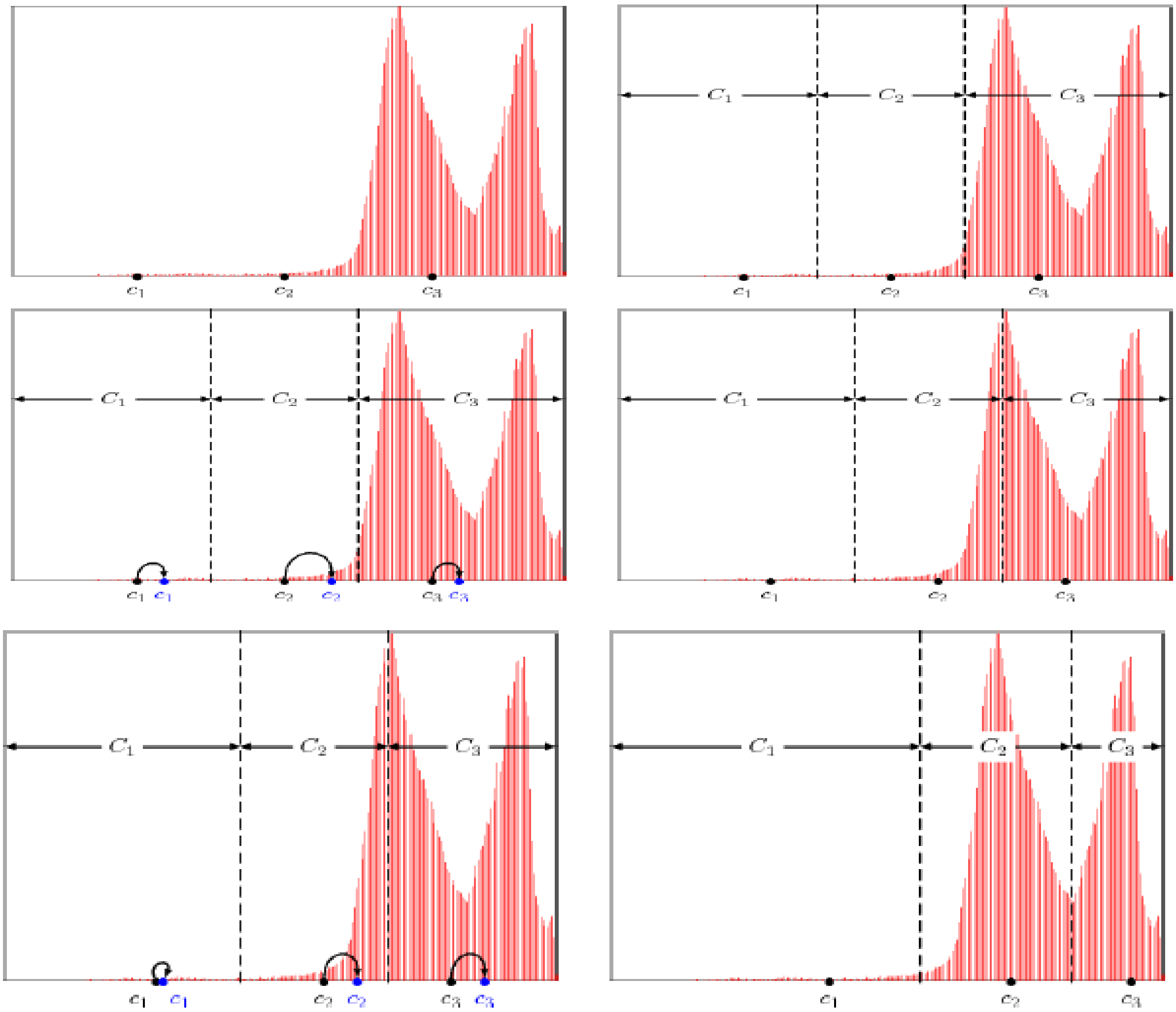
- C_X (C_Y) : coût des mauvaises classifications de X (Y).

6.3.3. Classification

Le pic de l’histogramme est la « composante image ». On découpe l’histogramme en k classes et chaque pixel sera étiqueter avec le numéro de sa classe.

La détection de “vallées” sur l’histogramme permet d’énumérer le nombre de classes qu’on approxime par un mélange de k Gaussiennes puis on peut appliquer par exemple l’algorithme des “k-means”. . .

Méthode des k-means : recherche de 3 classes



6.4. Opérations morphologiques

6.4.1. Introduction

Le but de la morphologie mathématique est d’étudier la forme, la granularité des objets à l’aide d’ensembles géométriques simples appelés éléments structurants. Ces éléments sont les entités de base de la morphologie mathématique. La morphologie mathématique a été développée à partir des années 70. Elle s'énonce et se comprend plus aisément sur des images binaires. Cette théorie peut être utilisée comme outil de :

- suppression des structures fines
- comblement des trous

Elément structurant :

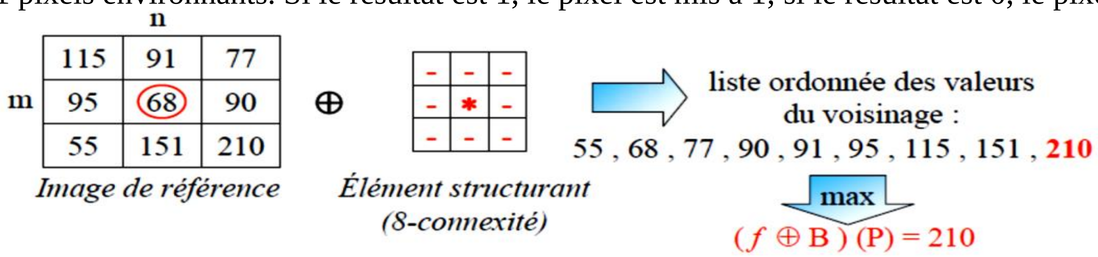
La morphologie mathématique repose sur l'utilisation d'un élément structurant « de forme : cercle , carré , hexagone » qui est composé :

- d'un pixel central (en noir)
- d'un ensemble de pixels (en gris)

Elle s’applique facilement à des images binaires, uniquement constituées de 0 et de 1 (noir et blanc respectivement). On peut néanmoins étendre les concepts morphologiques à des images en niveaux de gris.

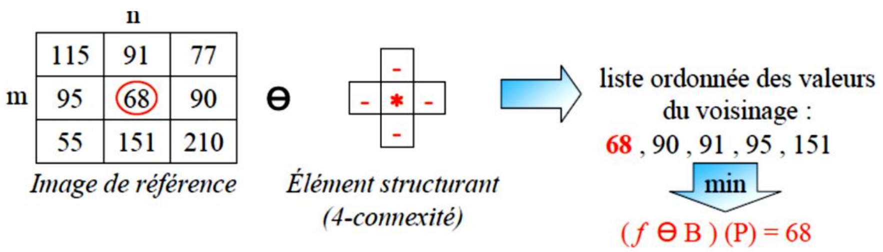
6.4.2.Dilatation

Lorsqu'une image est bruitée, sa binarisation fait apparaître des points noirs ou des points blancs parasites et isolés dont il faut se débarrasser par des techniques : la dilatation et l'érosion.
La dilatation consiste à éliminer les points "noirs" isolés : on dilate les parties "blanches" ce qui "mange" les points "noirs". La méthode est le balayage de l'image par un élément (2n+1)x(2n+1) avec utilisation du OU logique. On place le centre de la fenêtre sur le pixel courant et on effectue un OU logique sur les (2n +1)2 - 1 pixels environnants. Si le résultat est 1, le pixel est mis à 1; si le résultat est 0, le pixel est conservé.



6.4.3. Erosion

Il s'agit ici d'éliminer les points "blancs" isolés. La méthode est assez similaire à celle de la dilatation. On balaie l'image avec une fenêtre de taille (2n+1)x(2n+1). On place le centre de la fenêtre sur le pixel courant. On effectue un ET logique sur les pixels environnants. Si le résultat est 1, le pixel est conservé; si le résultat est 0, le pixel est mis à 0.



Notion de connexité