

Chapitre 07

Segmentation d'images

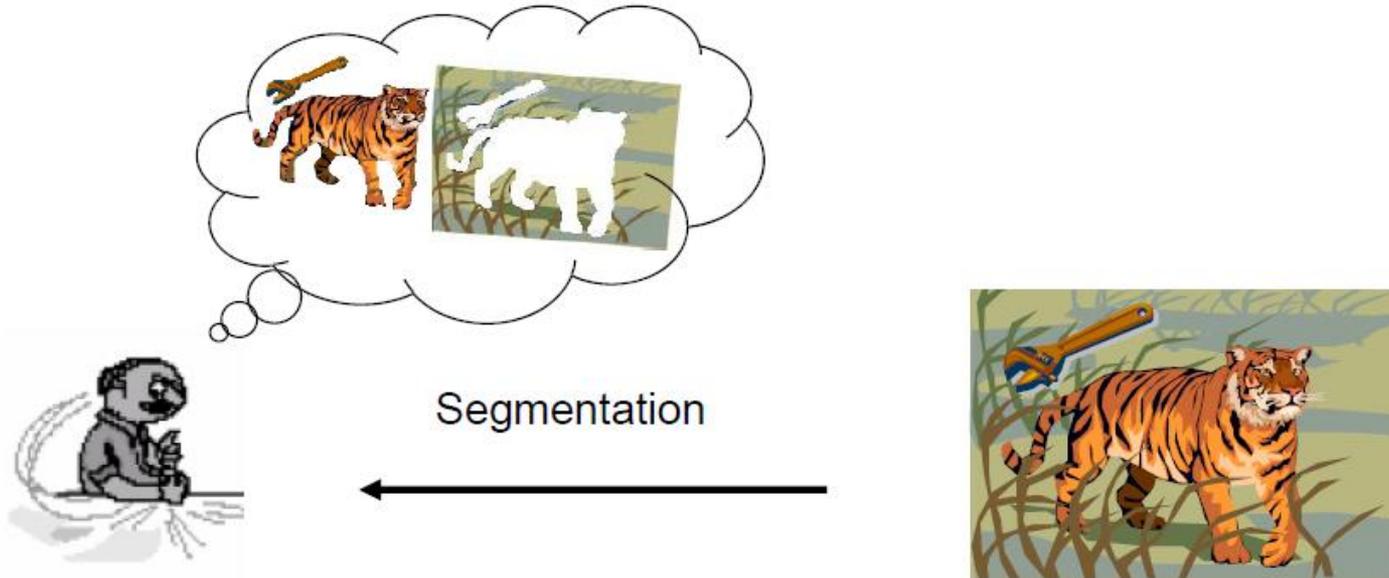
Aissa Boulmerka

2020-2021

1- PRINCIPES FONDAMENTAUX

Introduction

- La segmentation est un processus important dans la vision humaine.



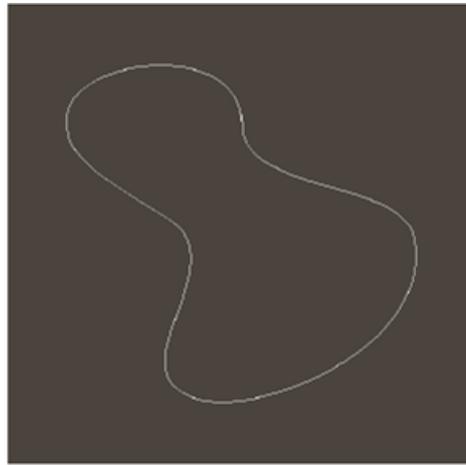
Introduction

- Plusieurs définitions pour la segmentation:

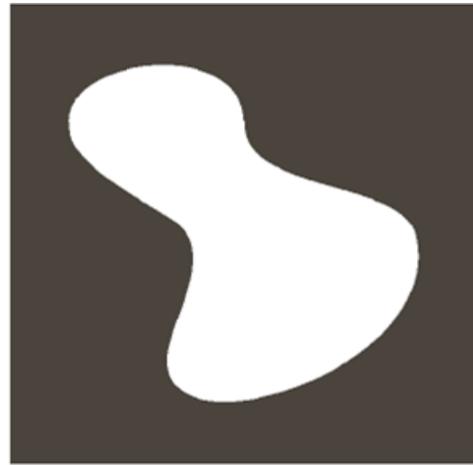
1-Segmentation par région versus contours ?



Image d'intensité



Contours



Régions

Introduction

2- Segmentation par couleur ou texture?



Image texturée



Segmentation par le niveau de gris



Image du gradient



Segmentation par le gradient

Introduction

3- Segmentation complète ou partielle?



Segmentation de
régions homogènes



Segmentation d'objets

Introduction

Définition:

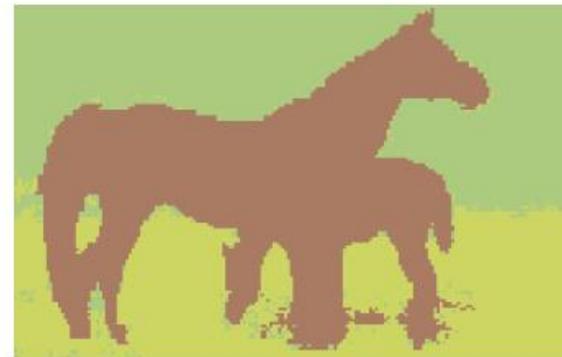
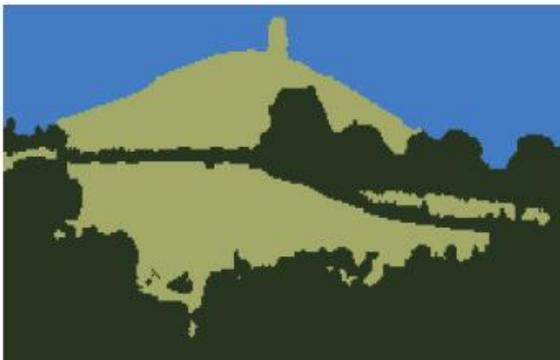
- C'est la division d'image en plusieurs régions homogènes (**segmentation partielle**), ou bien en objets ayant un sens sémantique (**segmentation complète**).
- Dans ce chapitre, nous étudierons quelques algorithmes célèbres qui font la **segmentation partielle**. De manière formelle:

Le problème revient à construire une partition $\{R_1, R_2, \dots, R_K\}$ de l'image I , qui respecte les conditions suivantes:

- R_i connexe et homogène.
- $R_i \cap R_j = \emptyset, \forall i \neq j$
- $\bigcup_{i=1}^K R_i = I$

Introduction

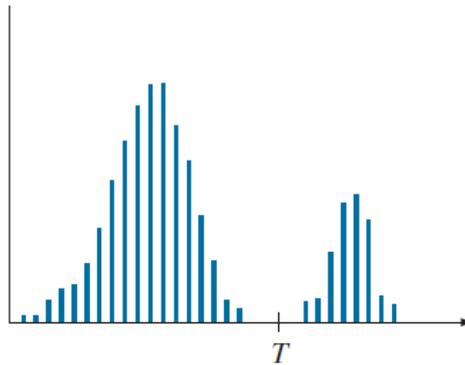
- **Exemples:**



2- MÉTHODE DU SEUILLAGE (THRESHOLDING)

Introduction

- Soit une image composée d'un **objet homogène** sur un **fond homogène**, et dont l'histogramme est donné par la figure suivante:

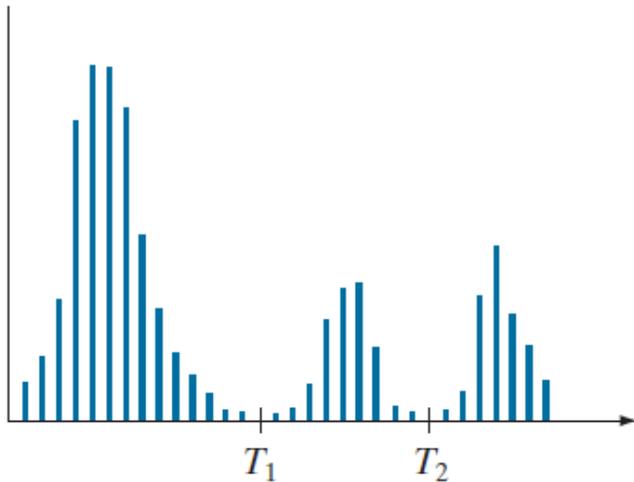


- Une façon de séparer l'objet du fond serait de trouver un seuil T tel que:
 - Si $I(x, y) > T$ alors le pixel (x, y) fait partie de l'objet
 - Si $I(x, y) \leq T$ alors le pixel (x, y) fait partie du fond.
- Soit alors $g(x, y)$ une fonction définie comme suit :

$$g(x, y) = \begin{cases} 1 & \text{si } I(x, y) > T. \\ 0 & \text{si } I(x, y) \leq T. \end{cases}$$

Méthode de seuillage

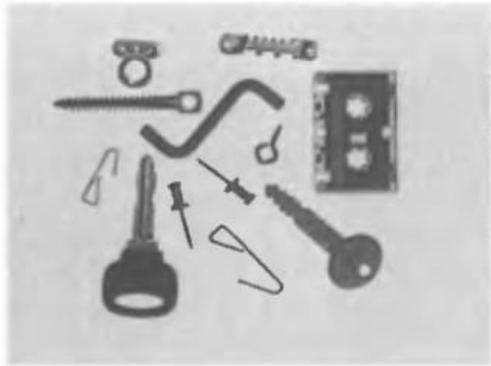
- Si le même seuil T utilisé est unique pour toute l'image, le seuillage est dit alors **global**, autrement il est dit **local**.
- On peut avoir aussi un seuillage **multiple** qui peut être appliqué pour un histogramme de plusieurs modes.
- Exemple:



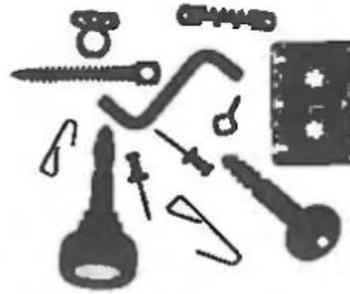
$$g(x, y) = \begin{cases} a & \text{Si } I(x, y) > T_1 \\ b & \text{Si } T_1 \leq I(x, y) \leq T_2 \\ c & \text{Si } I(x, y) \leq T_1 \end{cases}$$

Problème du choix du seuil

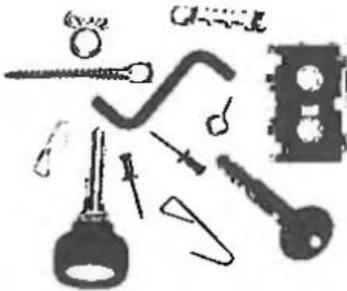
- Exemple avec un seuil:



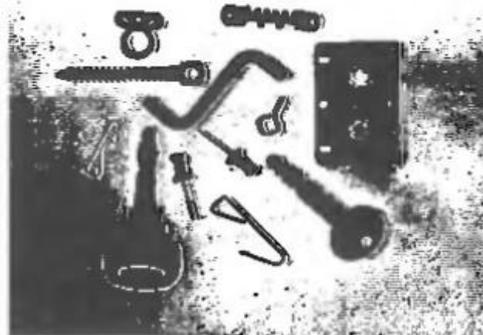
(a)



(b)



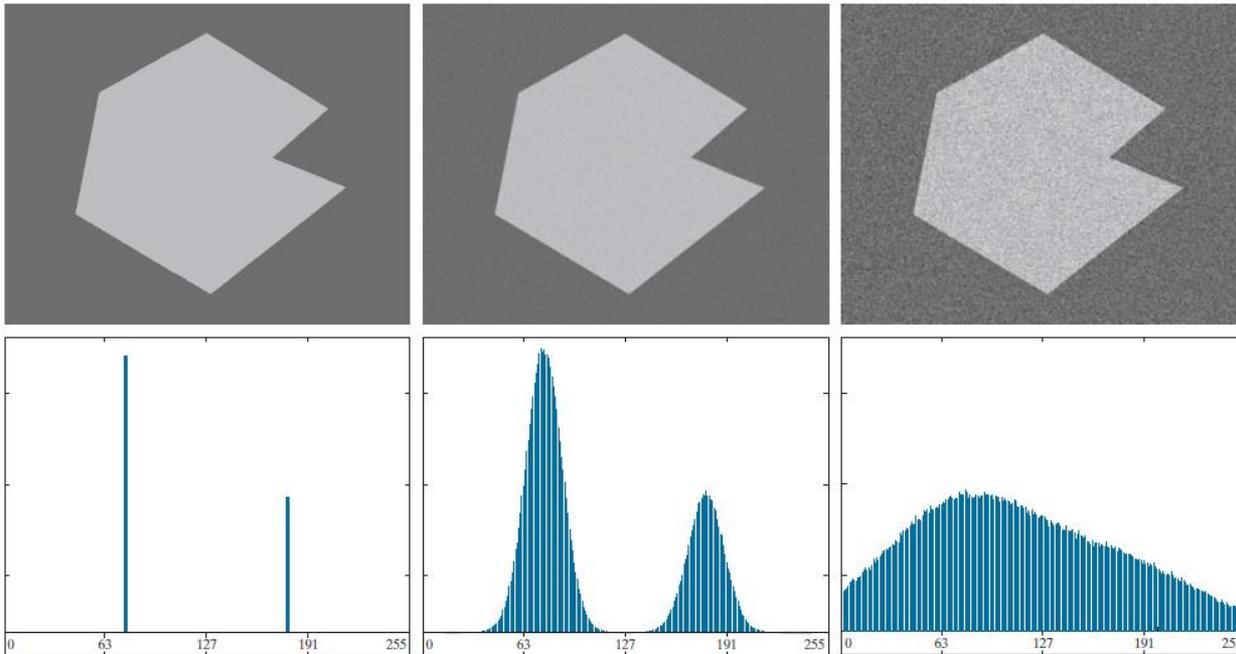
(c)



(d)

- (a) Image originale
- (b) Seuil optimal
- (c) Seuil trop grand
- (d) Seuil trop petit

Impact du bruit sur le seuillage



a	b	c
d	e	f

FIGURE 10.33 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d) through (f) Corresponding histograms.

- (a) Image sans bruit
- (b) Image avec bruit Gaussien $N(0,10)$
- (c) Image avec bruit Gaussien $N(0,50)$

Impact de l'illumination sur le seuillage

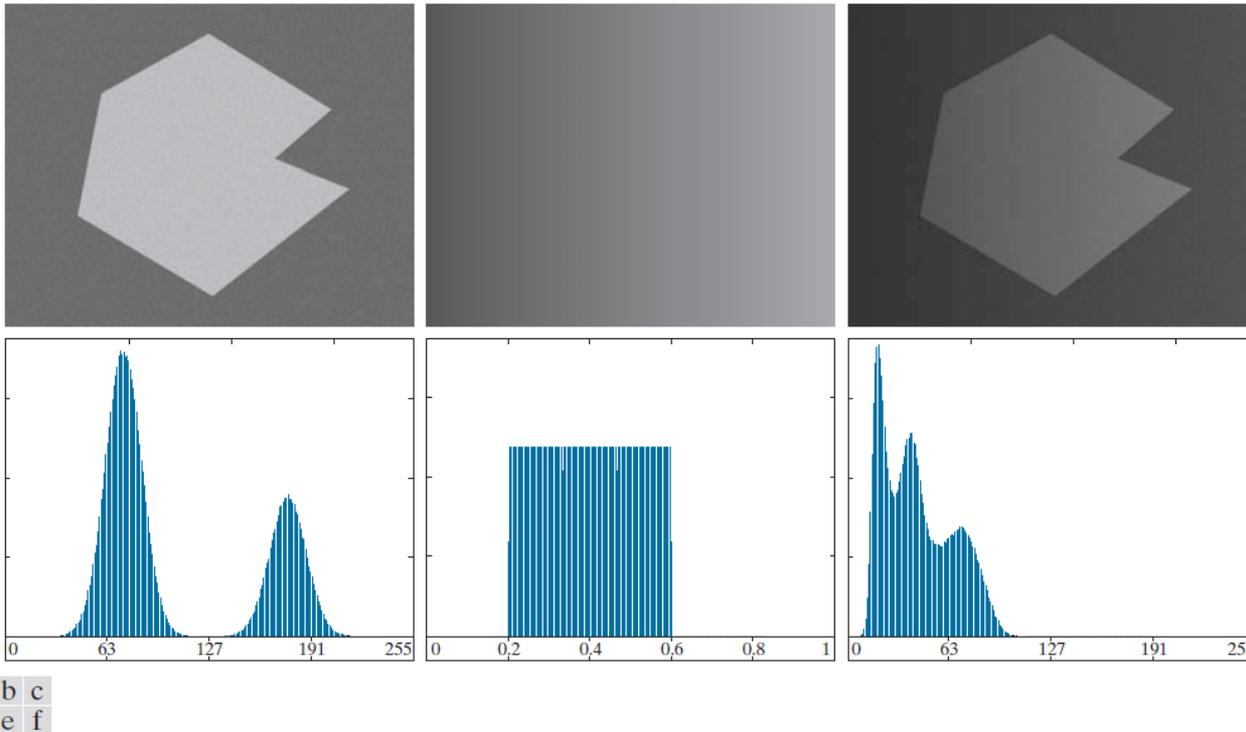


FIGURE 10.34 (a) Noisy image. (b) Intensity ramp in the range $[0.2, 0.6]$. (c) Product of (a) and (b). (d) through (f) Corresponding histograms.

- (a) Image originale
- (b) Image fonction « *ramp* »
- (c) Produit de l'image en (a) et l'image en (b)

Algorithme1: de seuillage

1. On ne suppose aucune connaissance a priori sur l'objet. Comme approximation, on peut utiliser au départ les 4 coins de l'image comme des pixels du fond.
2. À l'étape (t), calculer m_1 et m_2 comme moyennes du fond et de l'objet où les données utilisées seront celles définies dans l'étape ($t - 1$) par le seuil $T^{(t)}$:

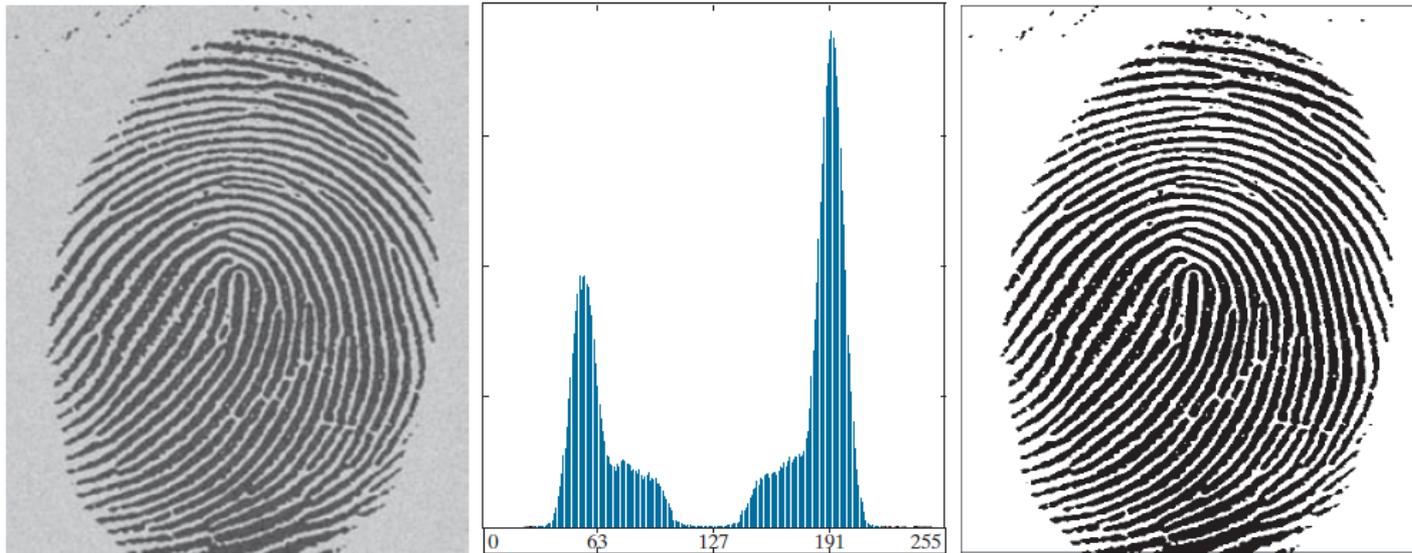
$$m_1 = \frac{\sum_{(x,y) \in \text{fond}} I(x,y)}{\# \text{ pixels du fond}}, \quad m_2 = \frac{\sum_{(x,y) \in \text{objet}} I(x,y)}{\# \text{ pixels de l'objet}}$$

3. Mettre à jour le seuil par :

$$T^{t+1} = \frac{1}{2} (m_1 + m_2)$$

4. Si $T^{t+1} = T^t$, sortir de l'algorithme, sinon, retourner à l'étape 2.

Algorithme1: de seuillage



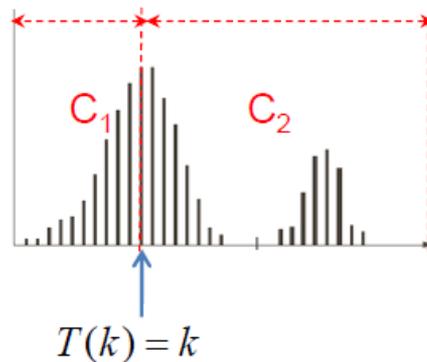
a b c

FIGURE 10.35 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (thin image border added for clarity). (Original image courtesy of the National Institute of Standards and Technology.)

Algorithme 2 de seuillage (Otsu)

Définitions:

- Soit $(0, 1, 2, \dots, L - 1)$ les niveaux de gris de l'image de taille $N \times M$.
- Soit le nombre d'occurrences du niveau de gris i , et $p_i = \frac{n_i}{NM}$ la probabilité du même niveau de gris.
- On a bien sur $\sum_{i=0}^{L-1} p_i = 1, p_i \geq 0$.
- Supposons qu'on sélectionne un seuil $T(k) = k, 0 < k < L - 1$, qui va diviser les niveaux de gris en deux classes: C_1 et C_2



$$[0, k] \in C_1.$$

$$[k+1, L-1] \in C_2.$$

Algorithme 2 de seuillage (Otsu)

- Basé sur le seuil k , les probabilités des classes C_1 et C_2 sont:

$$P_1(k) = \sum_{i=0}^k p_i \quad P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

- Les moyennes des classes C_1 et C_2 sont données par :

$$m_1(k) = \sum_{i=0}^k i \cdot p(i|C_1) \quad m_2(k) = \sum_{i=k+1}^{L-1} i \cdot p(i|C_2)$$

- Avec:

$p(i|C_1)$: la probabilité du niveau de gris i dans la classe C_1 .

$p(i|C_2)$: la probabilité du niveau de gris i dans la classe C_2 .

Algorithme 2 de seuillage (Otsu)

- Il est clair que:

$$p(i|C_1) = \frac{p_i}{p(C_1)} \quad p(i|C_2) = \frac{p_i}{p(C_2)}$$

- Il s'en suit que:

$$m_1(k) = \sum_{i=0}^k i \cdot p(i|C_1) = \sum_{i=0}^k i \cdot \frac{p_i}{p(C_1)} = \frac{1}{P_1(k)} \sum_{i=0}^k i \cdot p_i$$
$$m_2(k) = \sum_{i=k+1}^{L-1} i \cdot p(i|C_2) = \sum_{i=k+1}^{L-1} i \cdot \frac{p_i}{p(C_2)} = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} i \cdot p_i$$

Algorithme 2 de seuillage (Otsu)

- On définit la moyenne cumulative M_c au niveau de gris k par :

$$M_c(k) = \sum_{i=0}^k i \cdot p_i$$

- On définit aussi la moyenne globale de l'image par :

$$m_G = \sum_{i=0}^{L-1} i \cdot p_i$$

- On peut démontrer d'abord que :

$$m_G = P_1(k) \times m_1(k) + P_2(k) \times m_2(k)$$

Avec: $P_1(k) + P_2(k) = 1$

Algorithme 2 de seuillage (Otsu)

- Pour évaluer la qualité de la segmentation, on définit le critère suivant:

$$[\sigma_B(k)]^2 = P_1(k)[m_1(k) - m_G]^2 + P_2(k)[m_2(k) - m_G]^2$$

- On peut démontrer que:

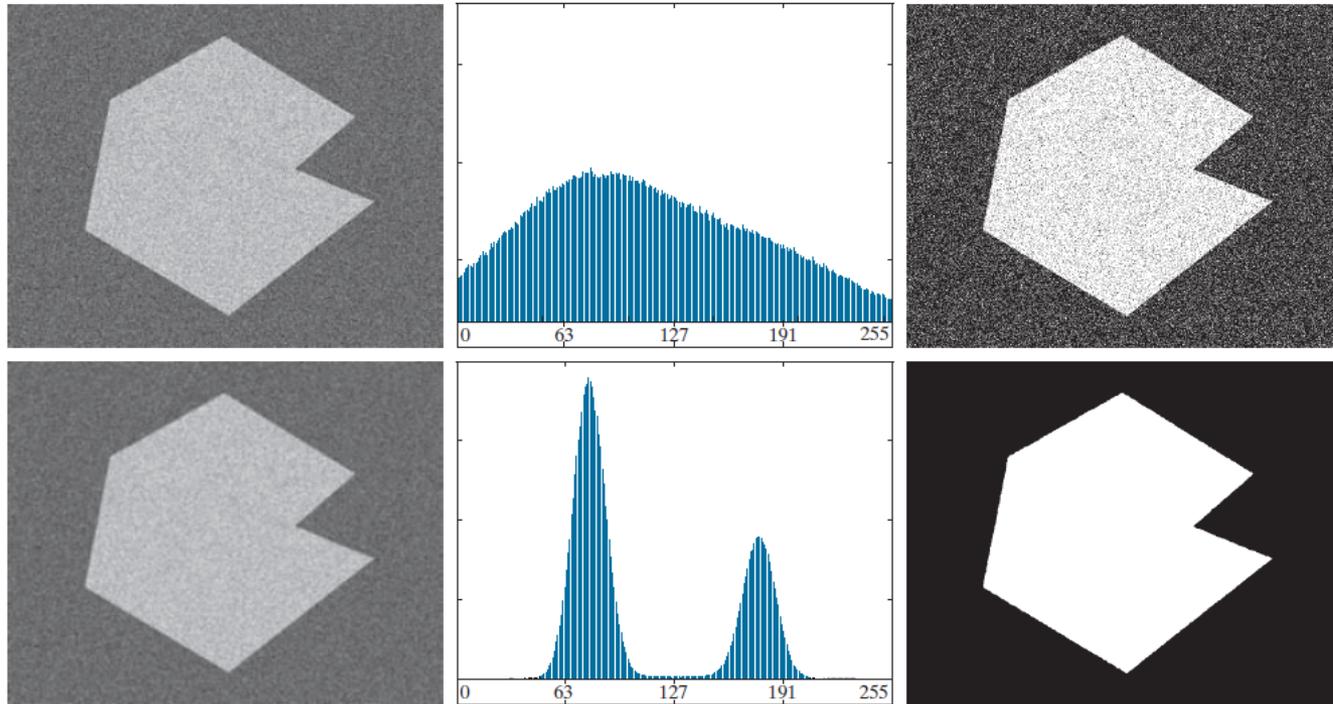
$$\begin{aligned} [\sigma_B(k)]^2 &= P_1(k)P_2(k)[m_1(k) - m_2(k)]^2 \\ &= \frac{m_G P_1(k) - M_c(k)}{P_1(k)(1 - P_1(k))} \end{aligned}$$

Algorithme 2 de seuillage (Otsu)

Résumé de l'algorithme

1. Calculer l'histogramme normalisé de l'image $(p_i, i = 0, \dots, L - 1)$.
2. Calculer les sommes cumulatives $P_1(k), k = 0, \dots, L - 1$.
3. Calculer les moyennes cumulatives $M_c(k), k = 0, \dots, L - 1$.
4. Calculer la moyenne globale m_G .
5. Calculer la variance inter – classes $[\sigma_B(k)]^2$
6. Obtenir le seuil k^* maximisant $[\sigma_B(k)]^2$. Si le maximum n'est pas unique, calculer la moyenne k^* des maximums.
7. Segmenter l'image à l'aide du seuil k^* .

Algorithme d'Otsu



a	b	c
d	e	f

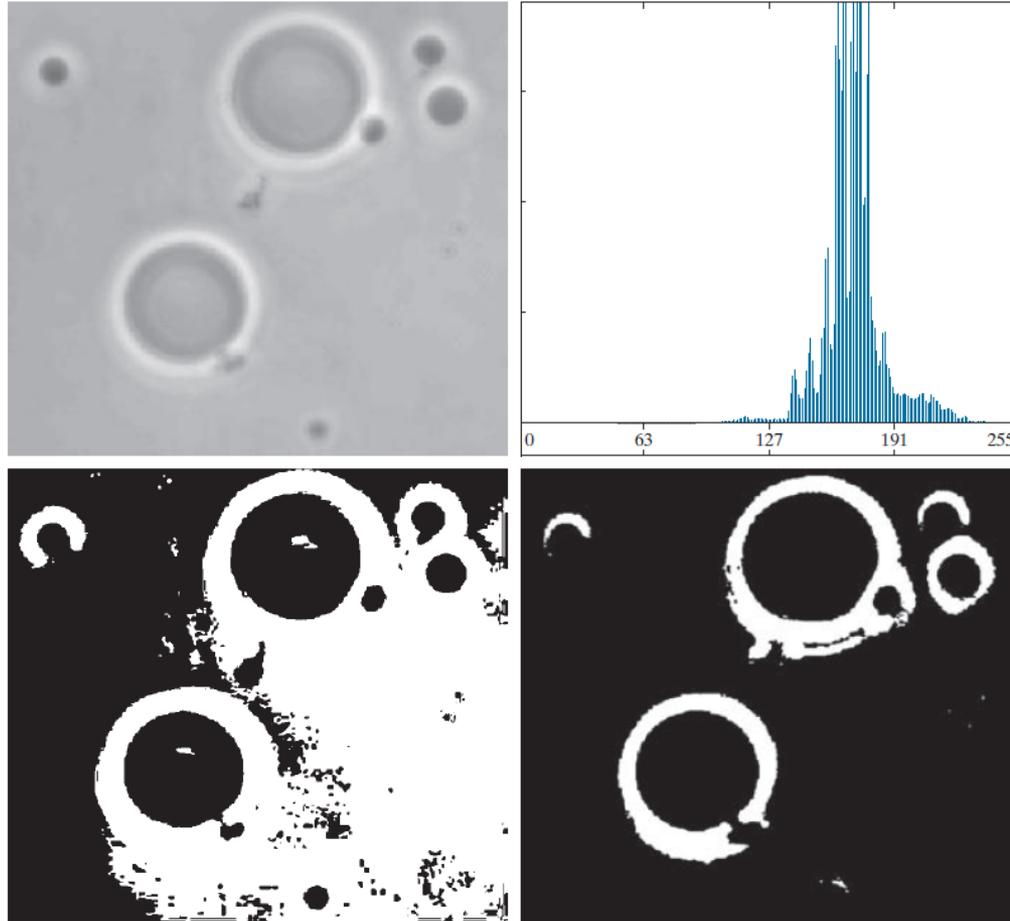
FIGURE 10.37 (a) Noisy image from Fig. 10.33(c) and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging kernel and (e) its histogram. (f) Result of thresholding using Otsu's method.

Exemple des algorithmes de seuillage

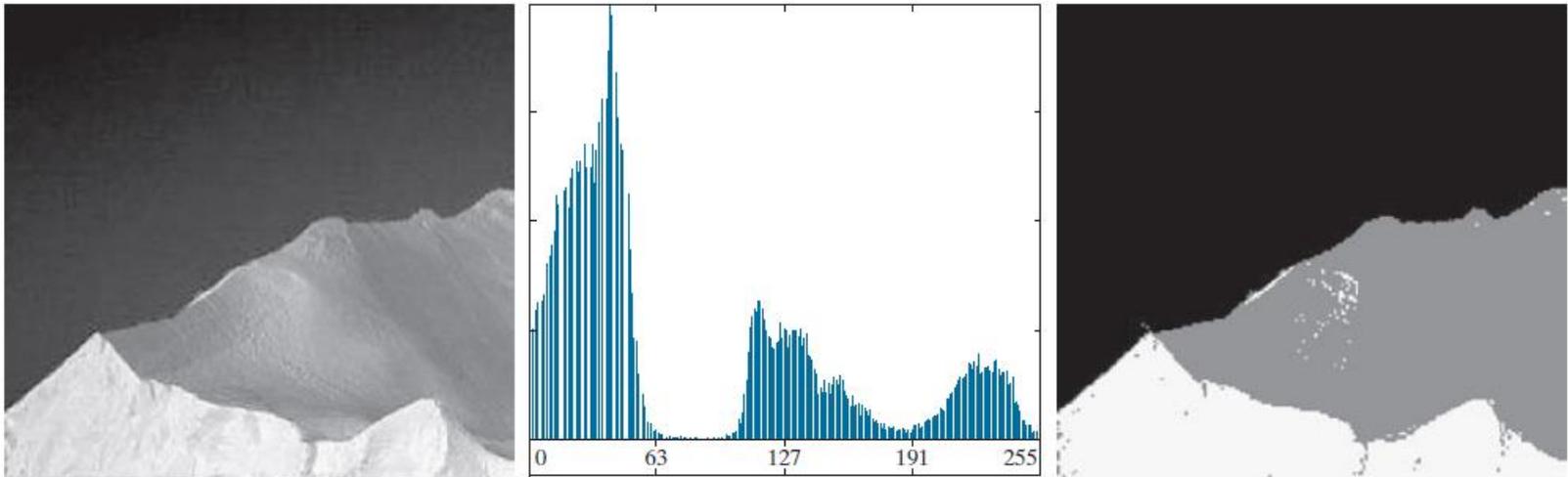
a b
c d

FIGURE 10.36

(a) Original image.
(b) Histogram (high peaks were clipped to highlight details in the lower values).
(c) Segmentation result using the basic global algorithm from Section 10.3.
(d) Result using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)



Exemple multi-seuillage



a b c

FIGURE 10.42 (a) Image of an iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

Remarques sur l'algorithme de seuillage

Avantages:

- 1) Simple et global.
- 2) Très efficace pour les images à deux régions (objet sur un fond).

Inconvénients:

- 1) Difficile à étendre pour la couleur.
- 2) Difficile à généraliser pour un nombre arbitraire de régions.
- 3) Ne tient pas compte de l'information spatiale de l'image.

3- MÉTHODE DE DIVISION/FUSION (SPLIT AND MERGE)

Algorithme de division/fusion

Définitions:

- Soit R l'image et Q un prédicat.
- L'approche consiste à subdiviser l'image en plusieurs régions R_i , telles que $Q(R_i) = TRUE$
- On commence par une région entière. Si $Q(R) = FALSE$, on divise la région en quatre sous-régions et on continue pour chaque sous-régions de manière récursive.
- La représentation par les **quadrees** est appropriée pour cette décomposition.
- Une fois la décomposition achevée, fusionner chaque deux régions adjacentes R_i et R_j respectant la condition: $Q(R_i \cup R_j) = TRUE$

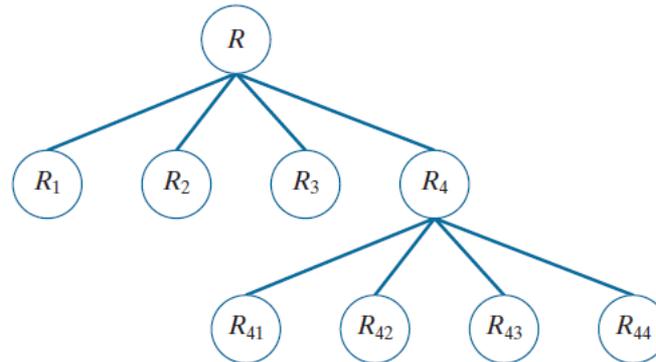
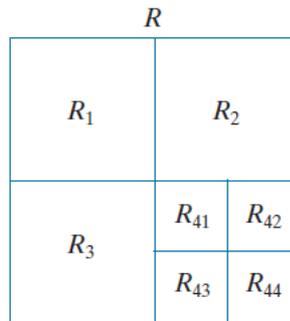
Algorithme de division/fusion

- Arrêter quand aucune fusion de régions n'est possible.

a b

FIGURE 10.47

(a) Partitioned image.
(b) Corresponding quadtree.
 R represents the entire image region.



Structure de représentation des décompositions de régions dans l'algorithme division/fusion.

Algorithme de division/fusion

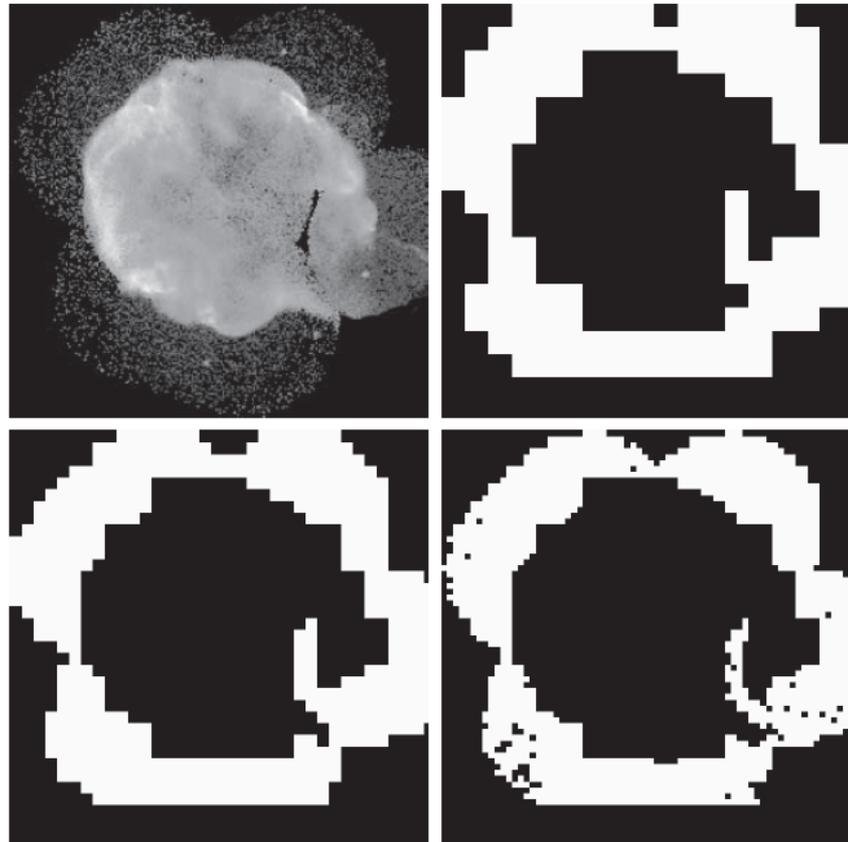
Exemple:

Ici , on a limité la taille des régions après décomposition à (b) 32x32, (c) 16x16 et (d) 8x8 respectivement.

a	b
c	d

FIGURE 10.48

(a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b) through (d) Results of limiting the smallest allowed quadregion to be of sizes of 32×32 , 16×16 , and 8×8 pixels, respectively. (Original image courtesy of NASA.)



Remarques sur l'algorithme de division/fusion

Avantages:

- 1) Tient compte de l'information spatiale.
- 2) Très simple à programmer.
- 3) S'applique à un nombre arbitraire de régions.

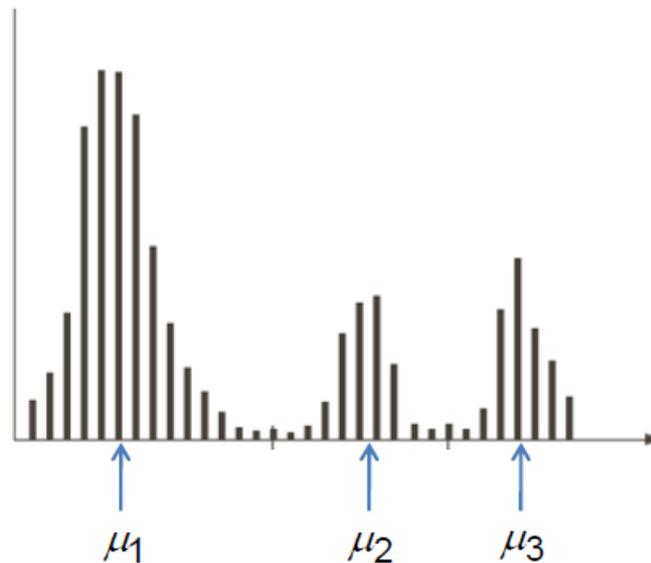
Inconvénients:

- 1) Difficulté de définir un prédicat approprié.
- 2) Produit des frontières de régions zigzagüées.

4- MÉTHODE DES K-MOYENNES (K-MEANS)

Principe de l'algorithme des K-Moyennes

- On suppose que chaque région est représentée par sa moyenne (de couleur ou de niveaux de gris). Ces moyennes représentent en général les centres des modes dans l'histogramme.
- Le problème est de rechercher ces moyennes. Ensuite, chaque niveau de gris sera assigné à la classe dont la moyenne est la plus proche.



Algorithme des K-Moyennes

- Soit K le nombre de régions dans l'image.
- Soit $\mu_1, \mu_2, \dots, \mu_K$ les moyennes (centres) de ces régions.
- Soit x_1, x_2, \dots, x_N les données de toute l'image.

1. Initialiser $\mu_1, \mu_2, \dots, \mu_K$.

2. Pour chaque donnée x_i : assigner x_i au centre k le plus proche.

$$r_{ik} = \begin{cases} 1 & \text{si } k = \operatorname{argmin}_j \|x_i - \mu_j\| \\ 0 & \text{sinon} \end{cases}$$

3. Mettre à jour chaque centre μ_k comme suit :

$$\mu_k = \frac{\sum_{i=1}^N r_{ik} x_i}{\sum_{i=1}^N r_{ik}}$$

4. Si les centres restent stables, sortir, sinon, retourner à l'étape 2.

Algorithme des K-Moyennes

Exemple:



Image originale



K=2



K=8



K=11



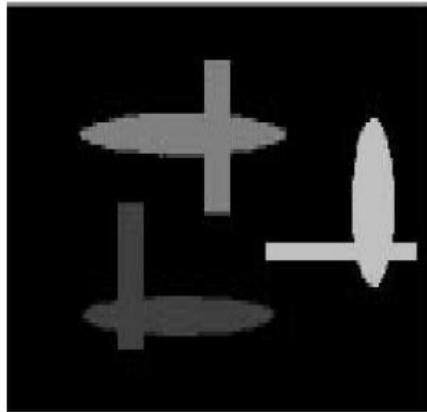
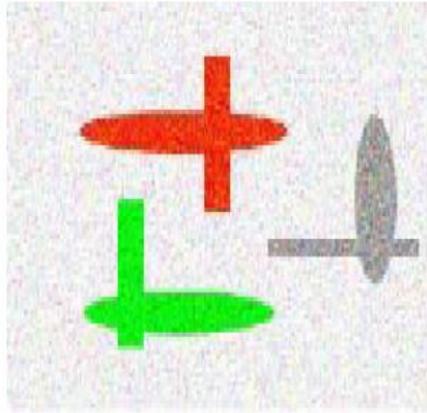
K=14



K=15

Algorithme des K-Moyennes

Autres exemples:



K=4



K=5

Algorithme des K-Moyennes

- On peut également utiliser **d'autres caractéristiques** pour segmenter une image. On peut par exemple utiliser les caractéristiques liées à la **texture**.

Exemple:

- il est impossible de segmenter cette image en 2 classes sur la base exclusive des niveaux de gris car l'objet central possède le même histogramme que le reste de l'image.



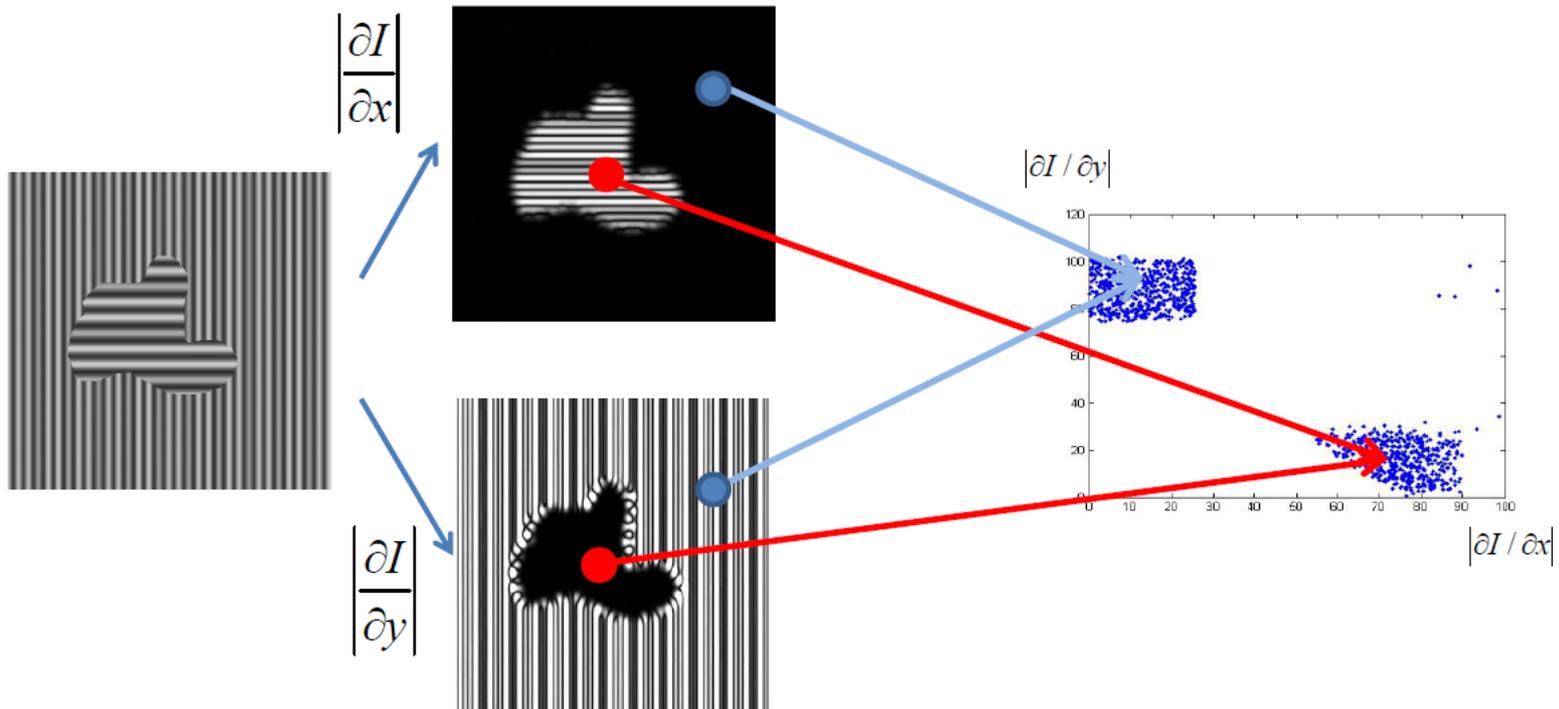
Que faire?

Il nous faut utiliser une autre caractéristique que le niveau de gris.

Algorithme des K-Moyennes

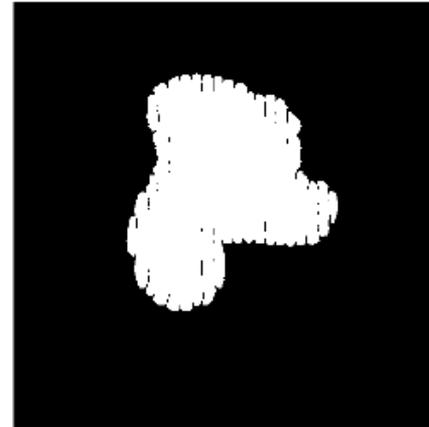
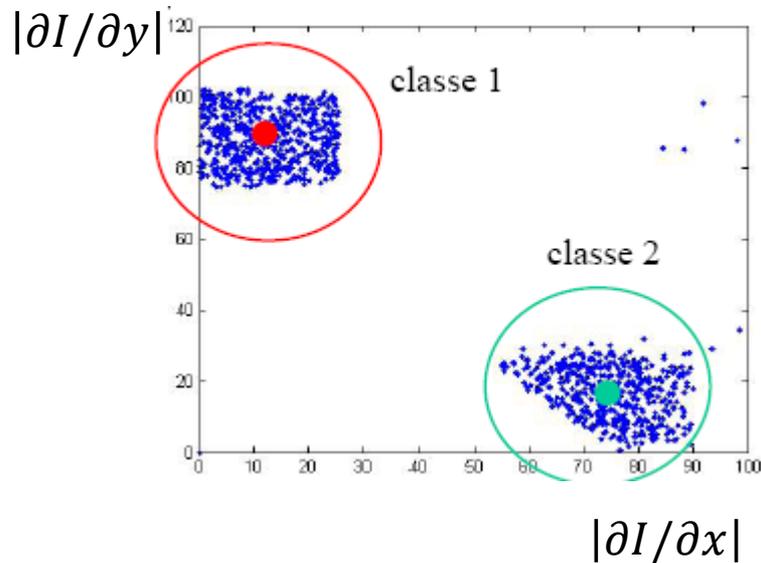
- On peut par exemple utiliser les dérivées en x et en y . De cette façon, chaque pixel (i, j) est associé à deux valeurs :

$$\left(\left| \frac{\partial I}{\partial x} \right|, \left| \frac{\partial I}{\partial y} \right| \right)$$



Algorithme des K-Moyennes

- Maintenant que chaque pixel est associé à un point 2D, il est facile de segmenter l'image à l'aide de l'algorithme des K-moyennes.



Remarques sur l'algorithme des K-moyennes

Avantages:

- 1) L'application aux images couleur n'ajoute pas une grande complexité.
- 2) Très simple à programmer.
- 3) S'applique à un nombre arbitraire de régions.

Inconvénients:

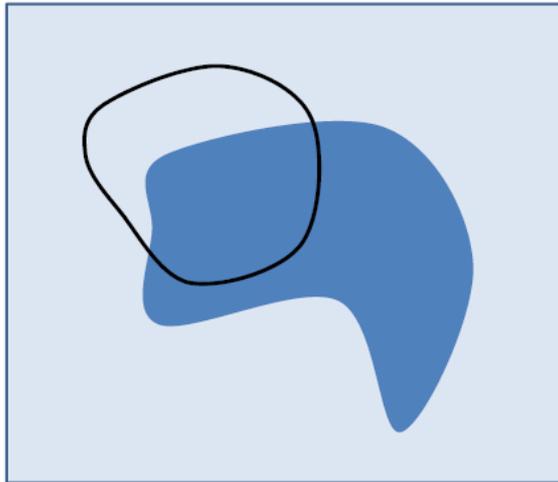
- 1) Ne contient pas l'information spatiale de l'image.
- 2) Nombre de régions optimal difficile à déterminer automatiquement.

5- MÉTHODE DES CONTOURS ACTIFS

Principe des contours actifs

Méthode des Snakes

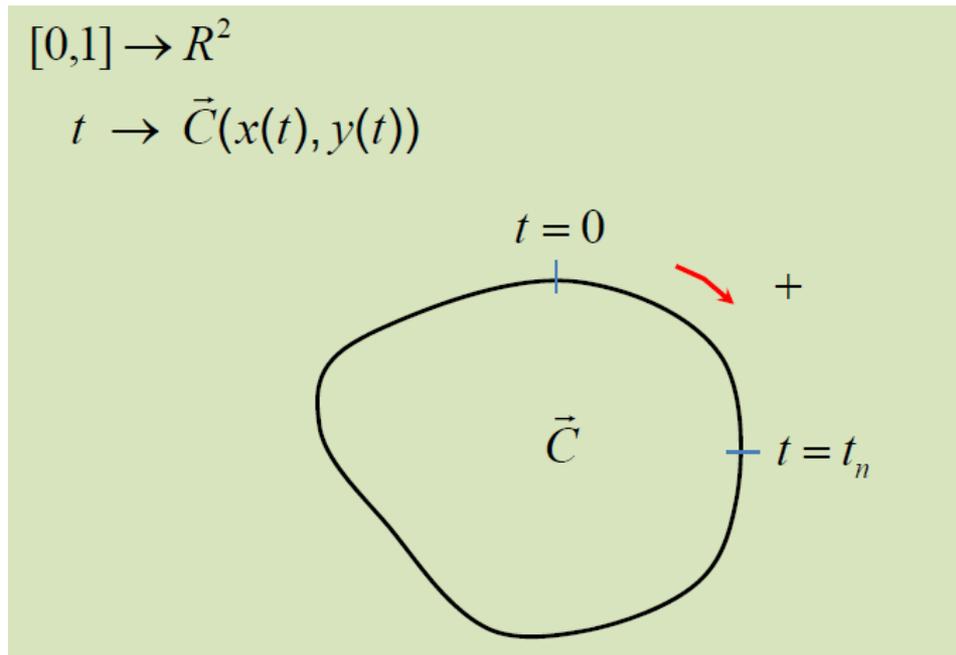
- Pour introduire l'information spatiale dans la segmentation, Kas et al. (1987) ont introduit la méthode des Snakes qui consiste à déformer une courbe initiale pour segmenter les frontières d'une région dans une image.



Principe des contours actifs

Méthode des Snakes (suite)

- Une courbe paramétrique est définie comme suit :



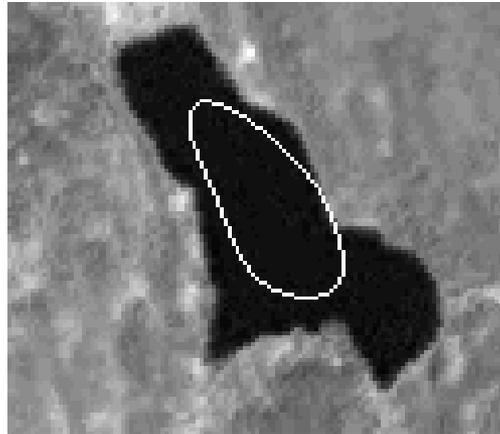
Méthode des Snakes (suite)

- La position finale de la courbe \vec{C} est obtenue en minimisant l'énergie suivante:

$$\begin{aligned} E(\vec{C}) &= E_{ext}(\vec{C}) + E_{int}(\vec{C}) \\ &= \alpha \int_0^1 g(\|\nabla I\|) dt + \beta \left[\int_0^1 \|\vec{C}'(t)\| + \|\vec{C}''(t)\| dt \right] \end{aligned}$$

- g est une fonction décroissante.
- L'énergie est minimisée par l'équation de Euler-Lagrange.

Méthode des Snakes (suite)



Limites:

- Sensibilité à l'initialisation.
- Changements topologiques (division/fusion) impossibles.

Méthode des level-sets

- Pour surmonter les limites des **Snakes**, la méthode des **level-sets** est proposée. Elle est formulée comme suit:
- Soit N l'ensemble des points de la courbe à déformer, et une fonction implicite définie comme suit :

$$\varphi(x, y) = \begin{cases} -d(x, y) & \text{si } (x, y) \text{ est à l'intérieur de } N_0 \\ 0 & \text{si } (x, y) \in N_0 \\ +d(x, y) & \text{si } (x, y) \text{ est à l'extérieur de } N_0 \end{cases}$$

Méthode des level-sets (suite)

- Chan et Vese (2001) on introduit la fonction d'énergie suivante à minimiser:

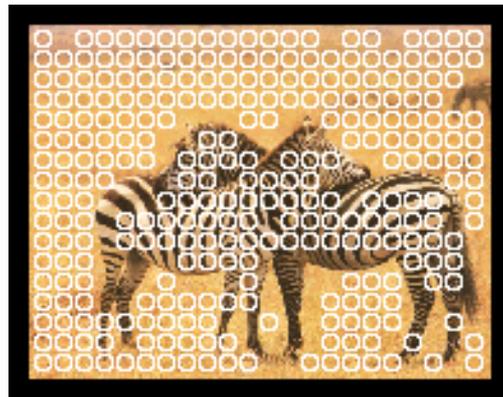
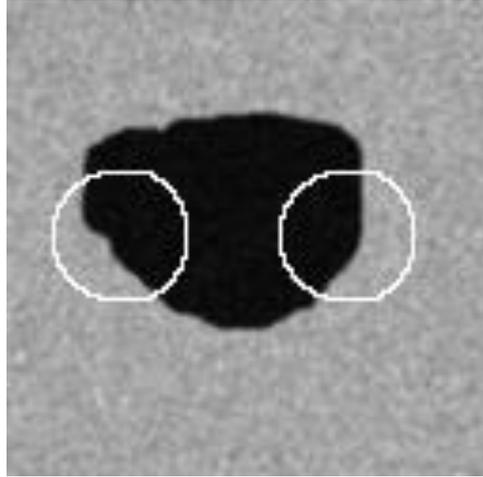
$$E(\varphi) = \alpha \int_0^L ds + \beta \left[\iint_{R_{in}} |I - u_{in}| dx dy + \iint_{R_{out}} |I - u_{out}| dx dy \right]$$

u_{in} : moyenne des niveaux de gris à R_{in} .

u_{out} : moyenne des niveaux de gris à R_{out} .

- L'énergie est minimisée par l'équation de Euler-Lagrange.

Méthode des level-sets (suite)



5- SEGMENTATION PAR CORRESPONDANCE (MATCHING)

Principe

- Le problème consiste à localiser un objet à l'intérieur d'une image. Pour cela, on doit avoir un motif de l'objet qui va être la référence.



Motif de
référence
de l'objet



Image où il faut chercher l'objet

Critères de recherche

On peut avoir plusieurs critères de recherche.

Soit I l'image (de taille $N \times M$) et h le motif de l'objet (de taille $H \times W$)

- $C_1: (x, y) \leftarrow \operatorname{argmax}_{(u,v)} \frac{\sum_{i=0}^{H-1} \sum_{j=0}^{W-1} I(u+i, v+j) \times h(i, j)}{\sum_{i=0}^{H-1} \sum_{j=0}^{W-1} I(u+i, v+j) \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} h(i, j)}$
- $C_2: C_2: (x, y) \leftarrow \operatorname{argmax}_{(u,v)} \frac{1}{1 + \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} [I(u+i, v+j) - h(i, j)]^2}$
- $(x, y) \leftarrow \operatorname{argmax}_{(u,v)} \frac{1}{1 + \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} |I(u+i, v+j) - h(i, j)|}$

Chapitre suivant

Chapitre 08 Représentation des images

Références:

1. M. S. Allili. *Eléments Avancés d'Analyse d'Images (Cours de 2e cycle)*. Université du Québec en Outaouais (UQO), Québec, Canada. Hivers 2014.
2. R. C. Gonzalez and R. E. Woods. *Digital image processing*. Pearson Education. 3rd Edition. 2008.
3. R. C. Gonzalez and R. E. Woods. *Digital image processing*. Pearson Education. 4th Edition. 2018.
4. R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital image processing using Matlab*. Gatesmark Publishing. 2nd Edition. 2009.