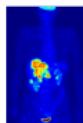


Imagerie Médicale

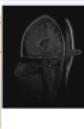
Partie 3: Segmentation

Pierre Maurel

pierre.maurel@irisa.fr



TEP



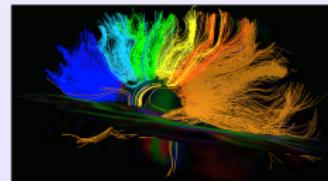
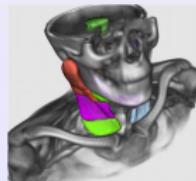
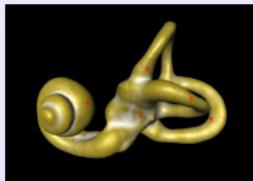
IRM



EEG



MEG



Analyse d'images, vision par ordinateur

- Segmentation : partitionner l'image en ses différentes parties.



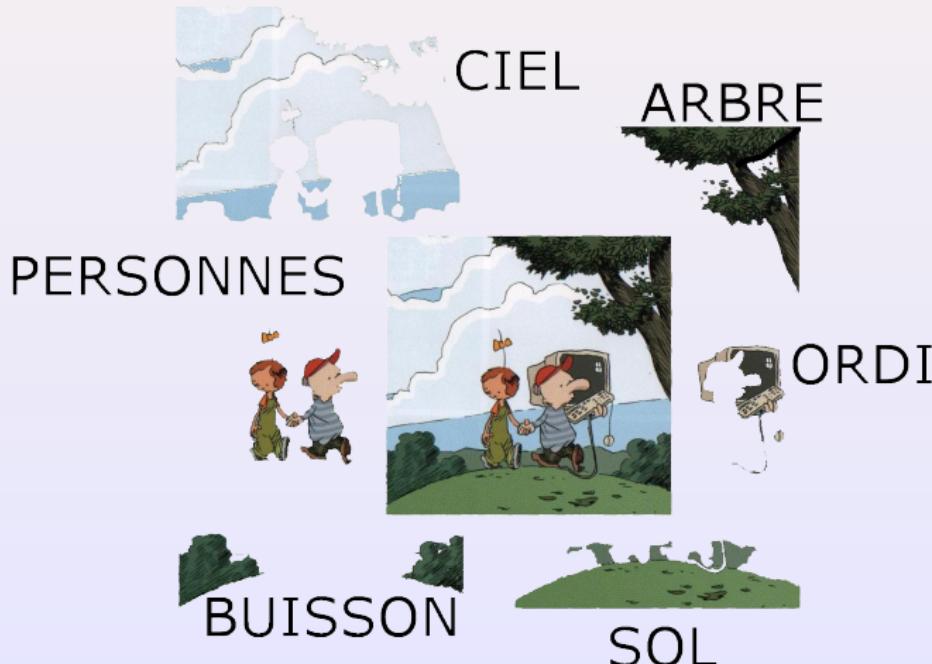
Analyse d'images, vision par ordinateur

- Segmentation : partitionner l'image en ses différentes parties.



Analyse d'images, vision par ordinateur

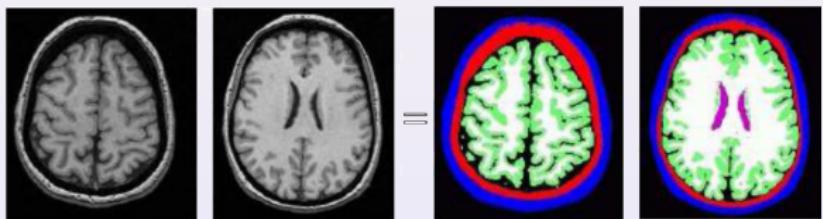
- Segmentation : partitionner l'image en ses différentes parties.
- Reconnaissance : étiqueter les différentes parties



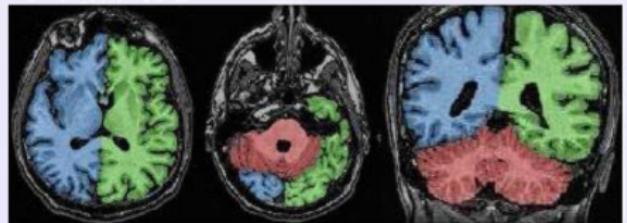
Segmentation ?

- La segmentation vise à découper une image en régions connexes présentant une homogénéité selon un certain critère.
- différentes possibilités → fonction de ce qu'on veut en faire
- Exemples

Peau, os, LCR, matière grise, matière blanche, ventricules



Hémisphère gauche, hémisphère droit, cervelet



Images issues de l'HDR de J.F. Mangin

Segmentation ?

À quoi ça sert ?

Important prérequis pour les étapes de mesure, de compréhension de la scène :

- reconnaissance d'objets
- indexation : rechercher dans une base d'images, les images "ressemblantes" à une image initiale
- compression
- recalage d'images, mises en correspondance
- ...

Segmentation ?

Exemple d'applications : Segmentation de visages

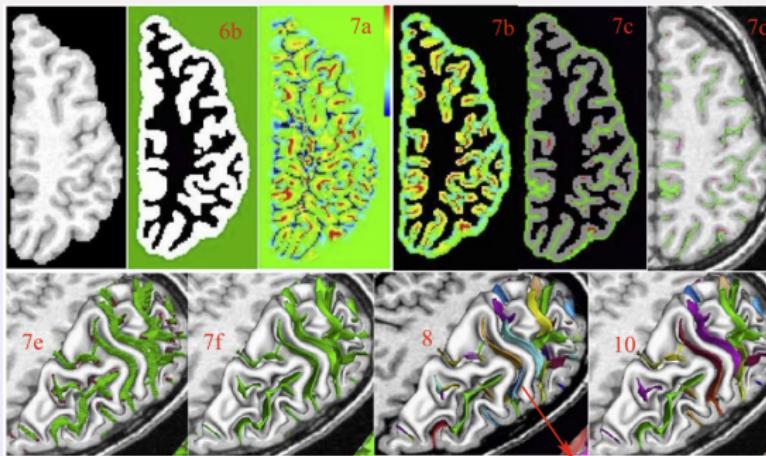


Face Segmentation Using Skin-Color Map in Videophone Applications, Douglas Chai, and King N. Ngan, 1999

- reconnaissance
- compression
- mise au point automatique

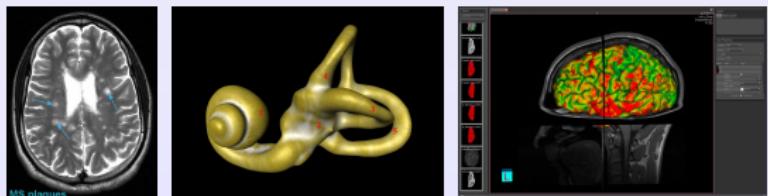
Segmentation ?

Exemple d'applications : Imagerie Médicale



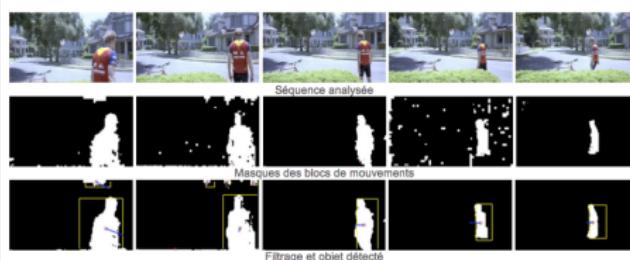
A framework to study the cortical folding patterns, Mangin et al., NeuroImage 2004

- Quantification des volumes des tissus, des organes
- Localisation d'une pathologie
- Étude d'une structure anatomique
- Planification d'un traitement
- Chirurgie assistée par ordinateur



Segmentation ?

Exemple d'applications : segmentation de vidéos

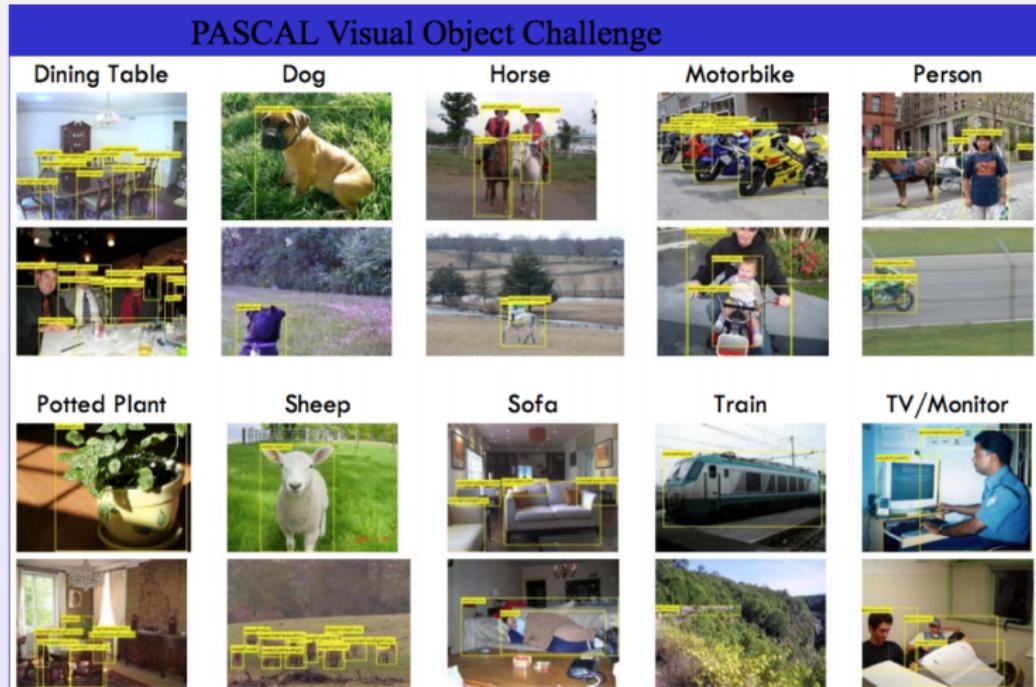


- Suivi d'objets/personnes
- Compression
- Reconnaissance

<http://www.labri.fr/projet/AIV/segmentationindexation.php>

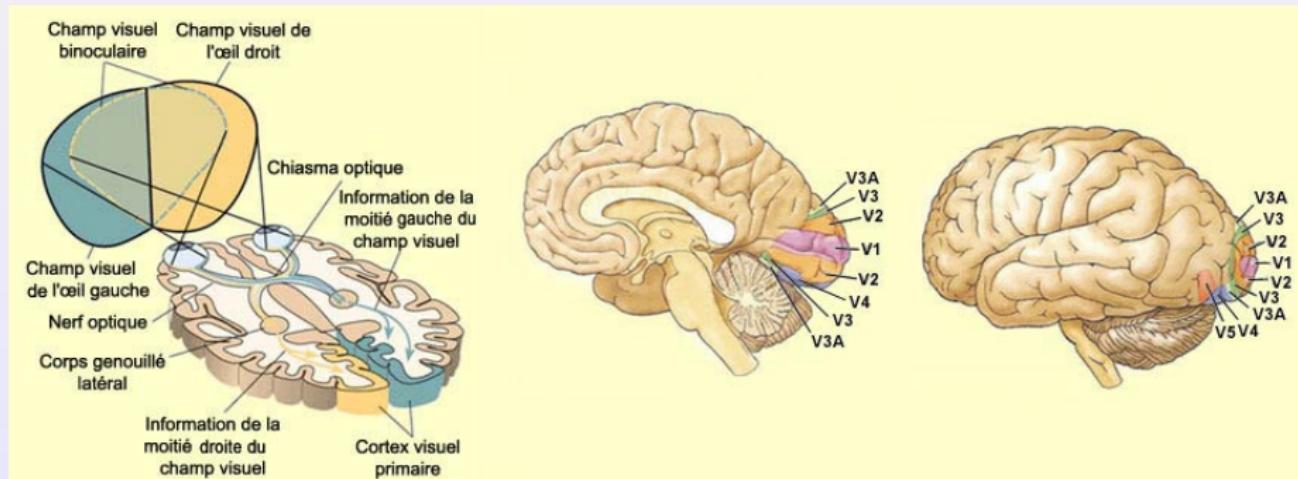
Segmentation ?

Exemple d'applications : reconnaissance d'objets



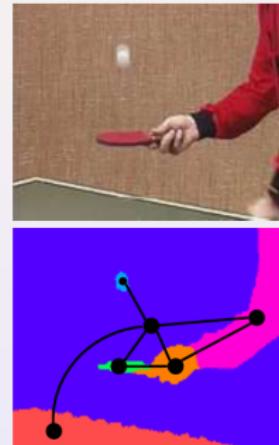
Segmentation ?

- Dans le système visuel, on a montré que les aires V1 et V2 sont sensibles à l'orientation du stimulus et que V3 et V4 extraient des contours.



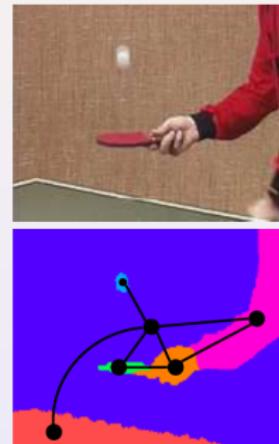
Segmentation ?

- Description "haut-niveau" d'une image
- Représentation sous forme de graphe d'adjacence
- La segmentation peut être basée sur
 - les discontinuités de l'image (contours)
 - les similitudes entre région (couleur, intensité, texture ...)



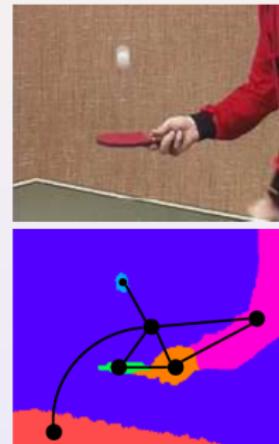
Segmentation ?

- Description "haut-niveau" d'une image
- Représentation sous forme de graphe d'adjacence
- La segmentation peut être basée sur
 - les discontinuités de l'image (contours)
 - les similitudes entre région (couleur, intensité, texture ...)
- Pas de solution universelle : en général, algo limité à un type d'application et/ou d'image



Segmentation ?

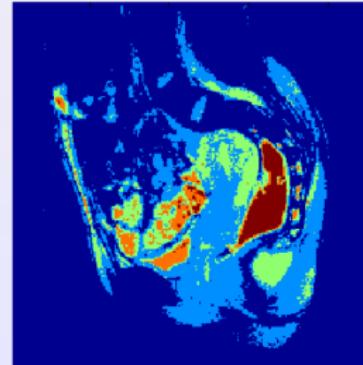
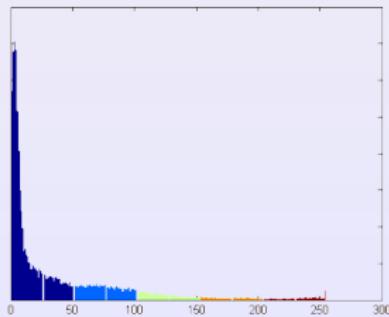
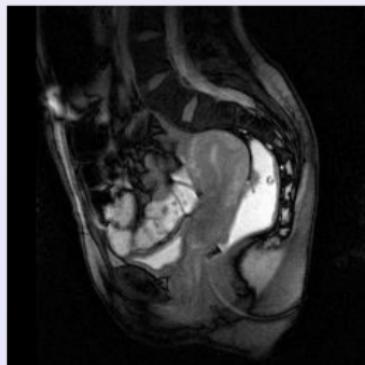
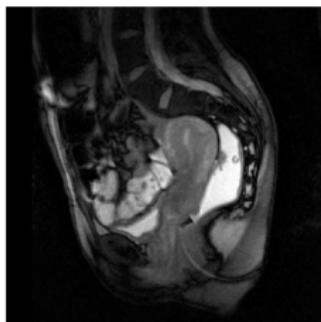
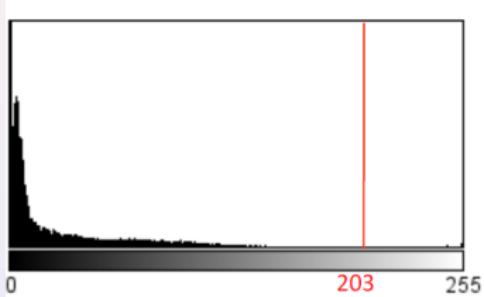
- Description "haut-niveau" d'une image
- Représentation sous forme de graphe d'adjacence
- La segmentation peut être basée sur
 - les discontinuités de l'image (contours)
 - les similitudes entre région (couleur, intensité, texture ...)
- Pas de solution universelle : en général, algo limité à un type d'application et/ou d'image
- Différentes approches :
 - approches globales
 - approches régions
 - approches contours



Approches Globales

Approches Globales

exemple le plus simple : seuillage d'histogramme



Seuillage d'histogramme

- Détermination du (ou des) seuil(s)

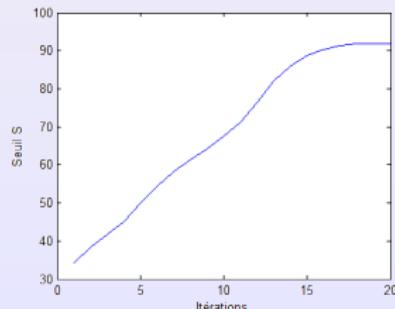
- valeur obtenue par test
- valeur moyenne
- valeur médiane
- choix automatique

Seuillage d'histogramme

- Détermination du (ou des) seuil(s)
 - valeur obtenue par test
 - valeur moyenne
 - valeur médiane
 - choix automatique
- Un algorithme simple
 - Choisir un seuil S initial (moyenne, médiane, ...)
 - On seuille → 2 groupes de pixels de moyenne μ_1 et μ_2
 - On calcule $S = \frac{\mu_1 + \mu_2}{2}$
 - On itère jusqu'à ce que S soit constant

Seuillage d'histogramme

- Détermination du (ou des) seuil(s)
 - valeur obtenue par test
 - valeur moyenne
 - valeur médiane
 - choix automatique
- Un algorithme simple
 - Choisir un seuil S initial (moyenne, médiane, ...)
 - On seuille \rightarrow 2 groupes de pixels de moyenne μ_1 et μ_2
 - On calcule $S = \frac{\mu_1 + \mu_2}{2}$
 - On itère jusqu'à ce que S soit constant



DEMO MATLAB

Seuillage d'histogramme

Méthode d'Otsu (1979)

- Un seuil t définit deux groupes de pixel : C_1 et C_2
- On cherche alors le seuil qui minimise la variance intra-classe :

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

- Les poids $\omega_i(t)$ représentent la probabilité d'être dans la ième classe
- les σ_i^2 sont les variances de ces classes

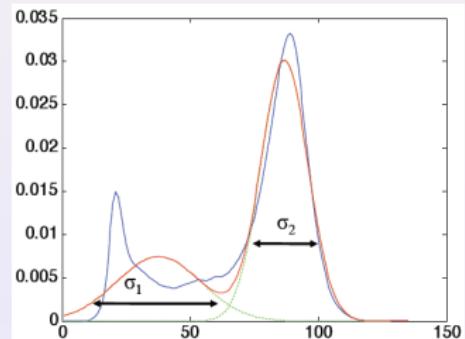


Seuillage d'histogramme

Seuillage par classification bayésienne

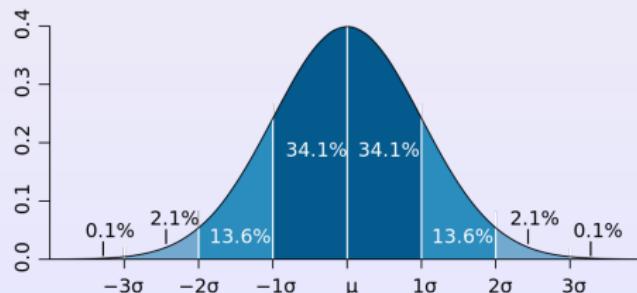
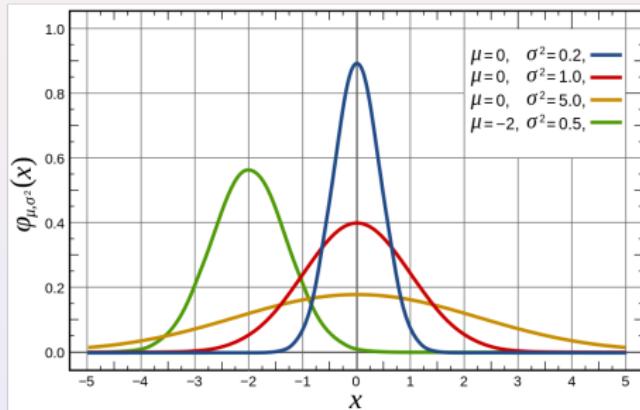
- Approximation de l'histogramme par un mélange de gaussiennes

$$\begin{cases} p_1(x) = \frac{P_1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \\ p_2(x) = \frac{P_2}{\sigma_2 \sqrt{2\pi}} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \end{cases} \quad \text{et} \quad P_1 + P_2 = 1$$

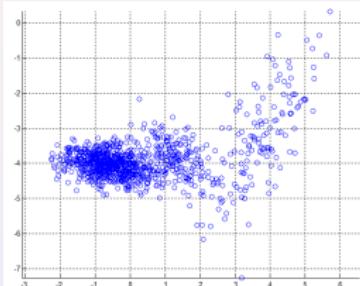


- Estimation de 5 paramètres libres (EM, gradient)

Aparté : Loi gaussienne



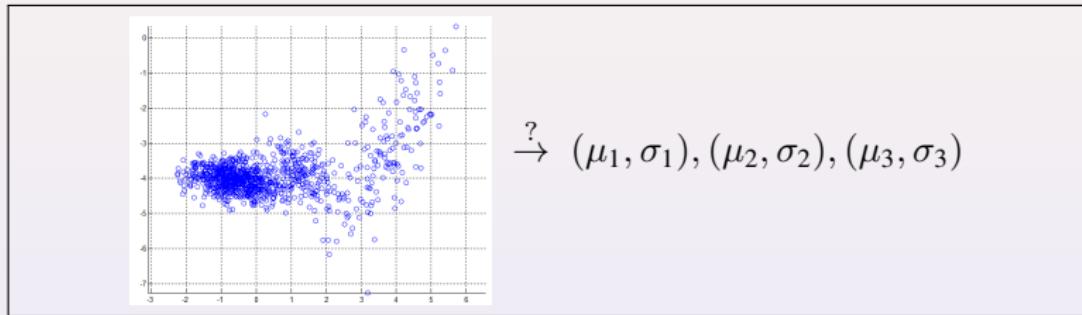
Algorithme EM, Modèle de mélanges gaussiens



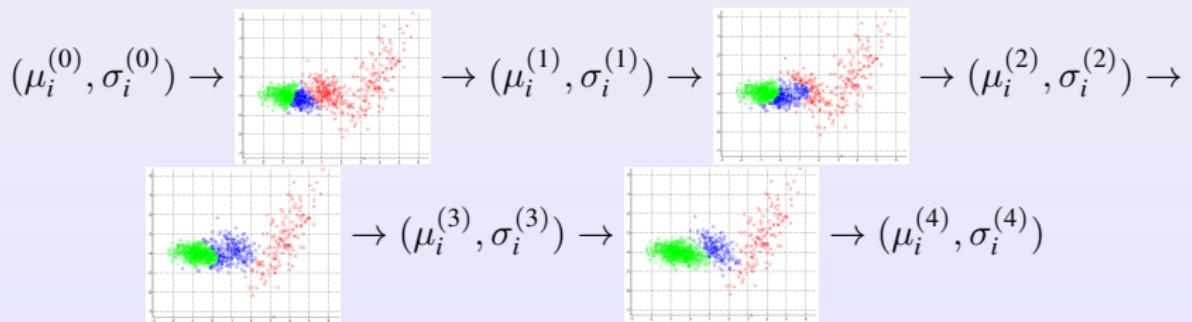
? $\rightarrow (\mu_1, \sigma_1), (\mu_2, \sigma_2), (\mu_3, \sigma_3)$

DEMO MATLAB

Algorithme EM, Modèle de mélanges gaussiens



DEMO MATLAB



Seuillage d'histogramme

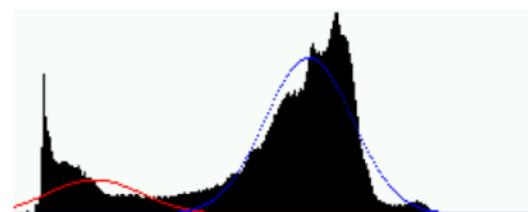
Seuillage par classification bayésienne



Image originale

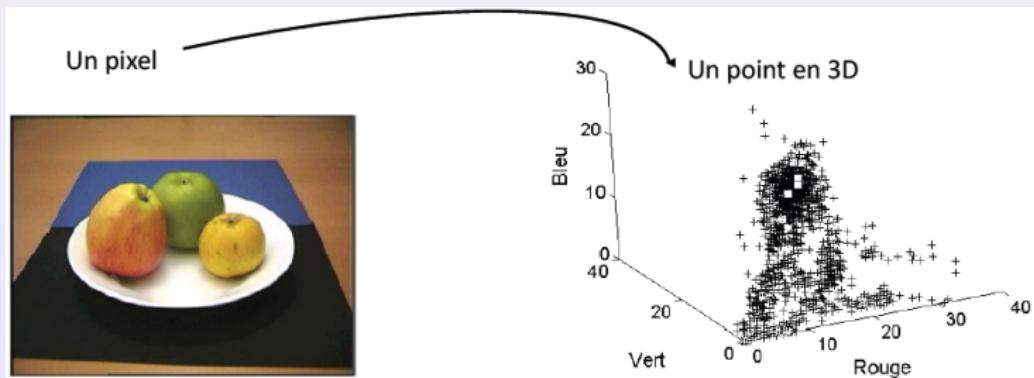


Image segmentée



Algorithme des K-moyennes (K-means)

- Algorithme de classification dans un espace n -dimensionnel
- ici : $n = 1$ (image en niveaux de gris) ou $n = 3$ (image en couleurs) ou plus

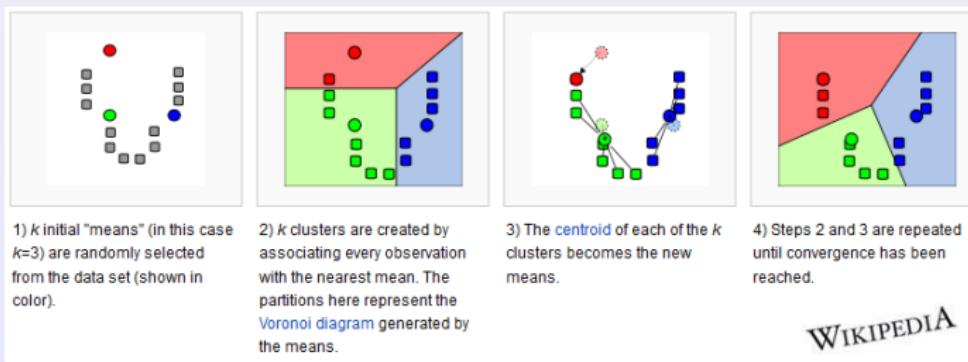


Algorithme des K-moyennes (K-means)

- IRM multimodal : n modalités en niveau de gris → k-means dans \mathbb{R}^n

Algorithme des K-moyennes (K-means)

- On initialise K graines (aléatoires par ex.) étiquetées de 1 à K
- On répète, jusqu'à convergence :
 - ➊ Pour chaque pixel, on trouve la graine i la plus proche au sens de la distance euclidienne
 - ➋ On donne à ce pixel l'étiquette de la graine i
 - ➌ On calcule le barycentre de chaque classe → les barycentres deviennent les nouvelles graines



Algorithme des K-moyennes (K-means)

- Algorithme des K-moyennes en 1D

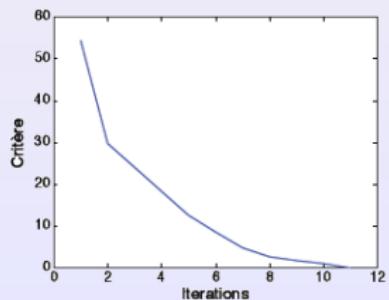
Initialisation (103,239,234)



Segmentation finale (19,219,114)



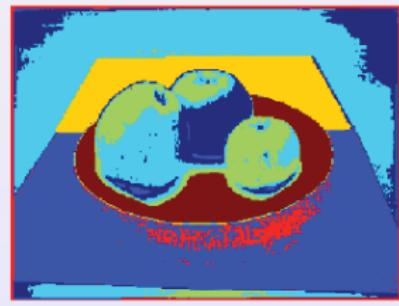
Convergence du critère



Algorithme des K-moyennes (K-means)

Inconvénients

- Sensibilité à l'initialisation
- Choix du nombre de classe k



variante : Algorithme des Fuzzy c-means

- Fuzzy C-Means : chaque point a un degré "flou" d'appartenance à chaque classe
- On donne maintenant un poids d'appartenance d'un pixel s à une classe k : u_{sk} tel que $\sum_k u_{sk} = 1$, $\forall s$

variante : Algorithme des Fuzzy c-means

- Fuzzy C-Means : chaque point a un degré "flou" d'appartenance à chaque classe
- On donne maintenant un poids d'appartenance d'un pixel s à une classe k : u_{sk} tel que $\sum_k u_{sk} = 1$, $\forall s$
- ainsi, la moyenne de la classe k devient :

$$\mu_k = \frac{\sum_s u_{sk} I(s)}{\sum_s u_{sk}}$$

variante : Algorithme des Fuzzy c-means

- Fuzzy C-Means : chaque point a un degré "flou" d'appartenance à chaque classe
- On donne maintenant un poids d'appartenance d'un pixel s à une classe k : u_{sk} tel que $\sum_k u_{sk} = 1$, $\forall s$
- ainsi, la moyenne de la classe k devient :

$$\mu_k = \frac{\sum_s u_{sk} I(s)}{\sum_s u_{sk}}$$

- le problème devient donc maintenant : trouver μ_1, \dots, μ_K et $U = (u_{sk})$ tels que

$$\sum_k \sum_s u_{sk}^m |I(s) - \mu_k|^2 \text{ soit minimal.}$$

$m > 1$ est un paramètre constant (degré de flou / fuzziness)

variante : Algorithme des Fuzzy c-means

- Ici encore : algorithme itératif

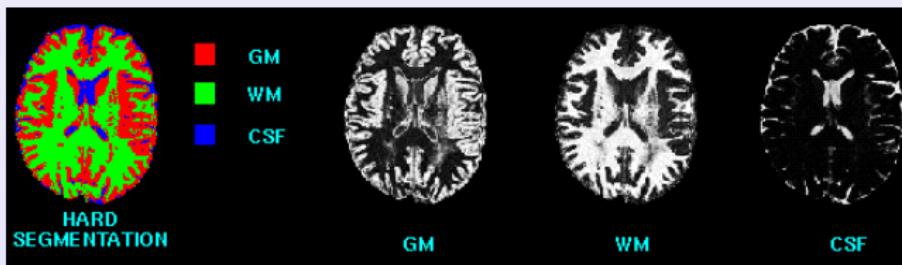
1 Répéter

$$1 \quad u_{sk} = \sum_{l=1}^K \left(\frac{|I(s) - \mu_k|}{|I(s) - \mu_l|} \right)^{-\frac{2}{m-1}}$$

$$\sum u_{sk} I(s)$$

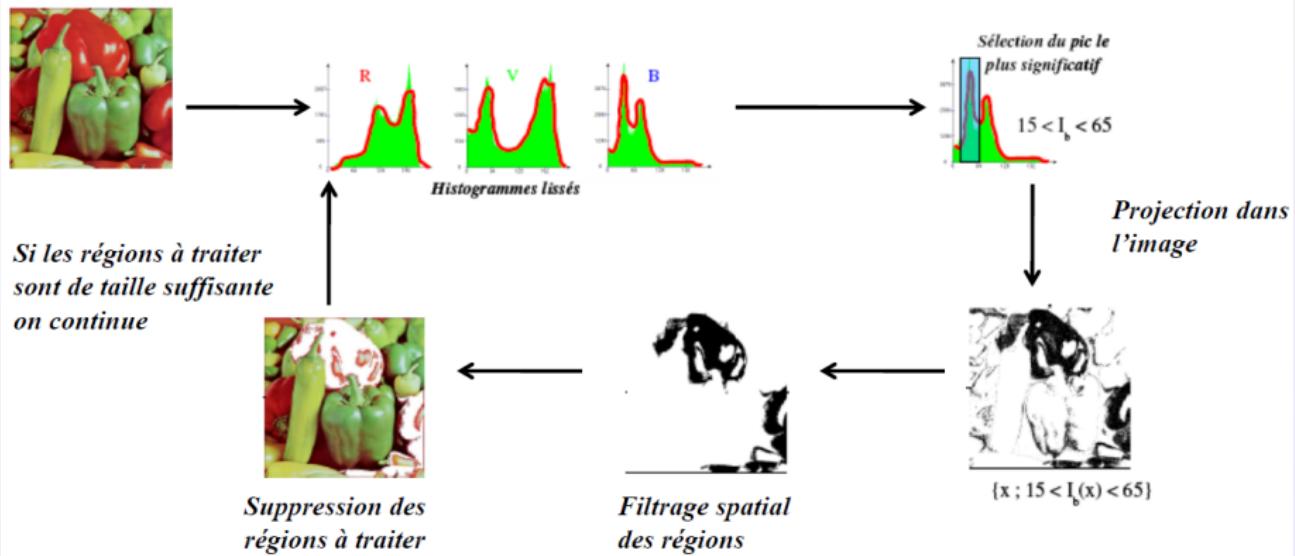
$$2 \quad \mu_k = \frac{\sum_s u_{sk}}{\sum_s}$$

2 jusqu'à ce que $\max(|u_{sk}^n - u_{sk}^{n-1}|) < \epsilon$

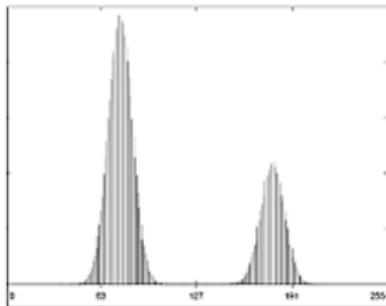
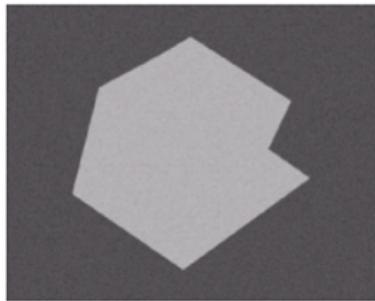


Sélection recursive d'histogramme

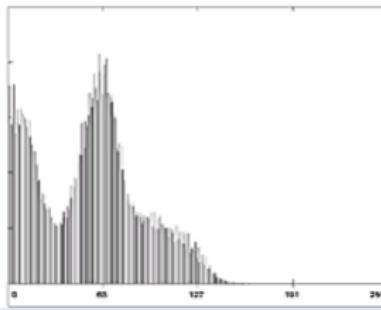
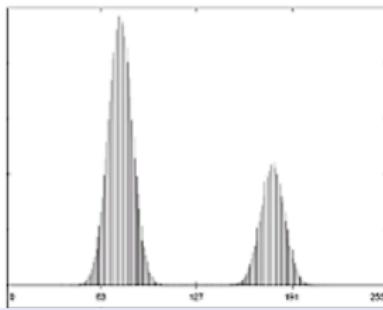
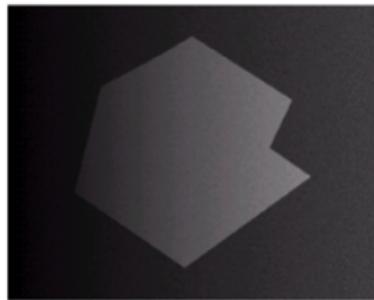
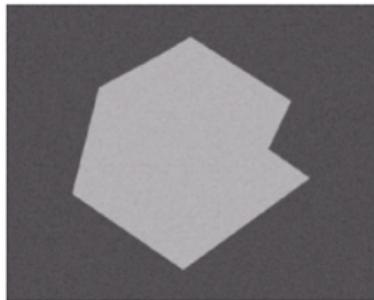
Ohlander, Price et Reddy (1978)



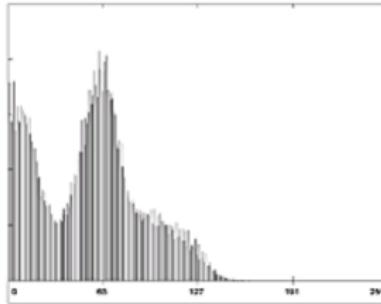
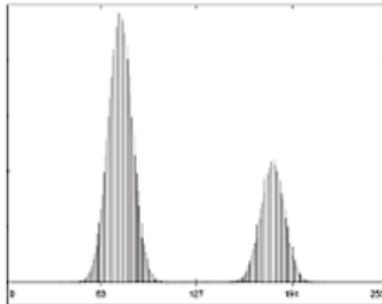
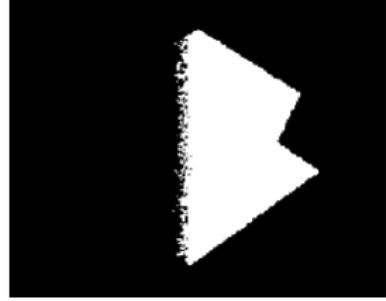
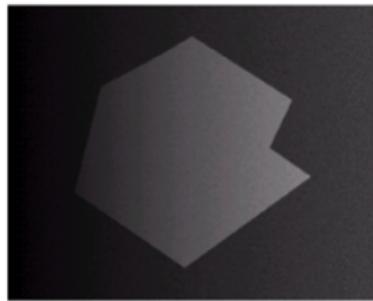
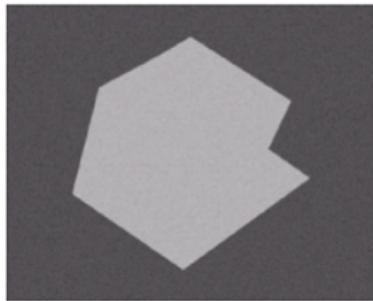
Limite des approches globales



Limite des approches globales



Limite des approches globales



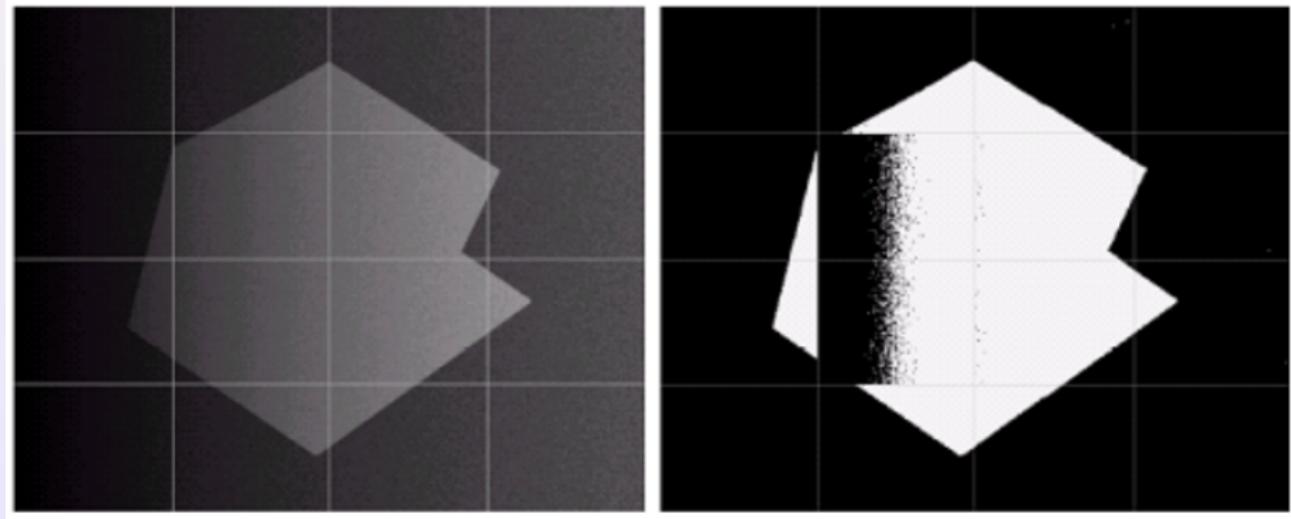
Seuillage par la
méthode d'Otsu

Seuillage adaptatif

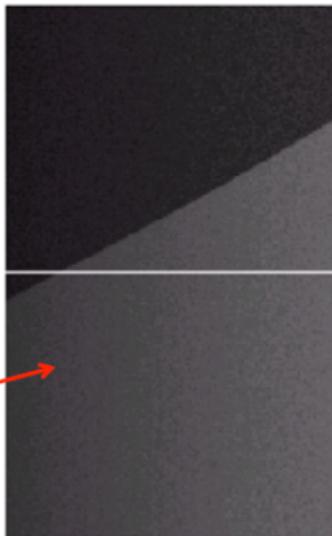
- On divise l'image en un certain nombre de sous-régions
- → seuillage sur chaque région

Seuillage adaptatif

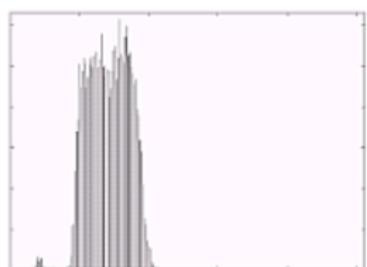
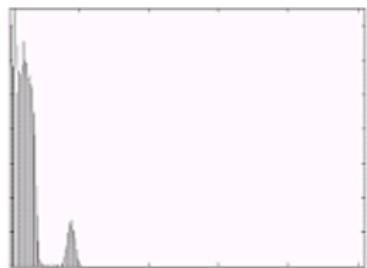
- On divise l'image en un certain nombre de sous-régions
- → seuillage sur chaque région



Seuillage adaptatif



Histogramme bimodal



Histogramme monomodal

Approches globales : bilan

- **Avantages :**

- simple, rapide
- bien adapté aux histogrammes multimodaux

Approches globales : bilan

- **Avantages :**

- simple, rapide
- bien adapté aux histogrammes multimodaux

- **Inconvénients :**

- il faut connaître le nombre de classes
- choisir les seuils
- Pas d'information de connexité

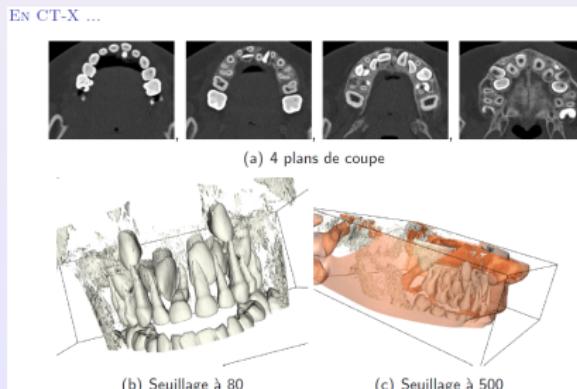
Approches globales : bilan

- **Avantages :**

- simple, rapide
- bien adapté aux histogrammes multimodaux

- **Inconvénients :**

- il faut connaître le nombre de classes
- choisir les seuils
- Pas d'information de connexité



Parfois suffisant :

Approches "Régions"

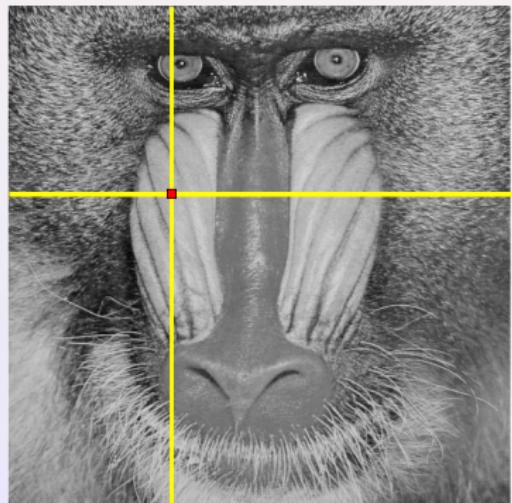
Croissance de régions - Region growing

- on choisit une (ou plusieurs) "graine(s)"
- La région R (pour l'instant réduite à un point) possède une moyenne μ_R et un écart-type σ_R
- On ajoute alors à R tous les pixels voisins de R qui sont suffisamment semblables à R , exple

$$|I(x) - \mu_R| < \text{seuil}$$

ou bien

$$\left\{ \begin{array}{l} \min\{|I(x) - I(y)|; y \in R \cap V(x)\} < \text{seuil} \\ |I(x) - \mu_R| < \sigma_R \end{array} \right.$$



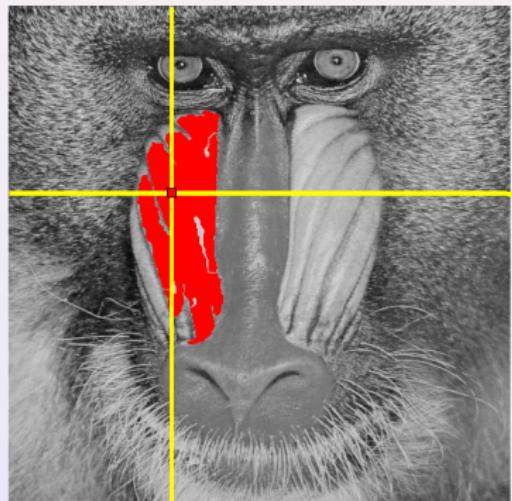
Croissance de régions - Region growing

- on choisit une (ou plusieurs) "graine(s)"
- La région R (pour l'instant réduite à un point) possède une moyenne μ_R et un écart-type σ_R
- On ajoute alors à R tous les pixels voisins de R qui sont suffisamment semblables à R , exple

$$|I(x) - \mu_R| < \text{seuil}$$

ou bien

$$\left\{ \begin{array}{l} \min\{|I(x) - I(y)|; y \in R \cap V(x)\} < \text{seuil} \\ |I(x) - \mu_R| < \sigma_R \end{array} \right.$$



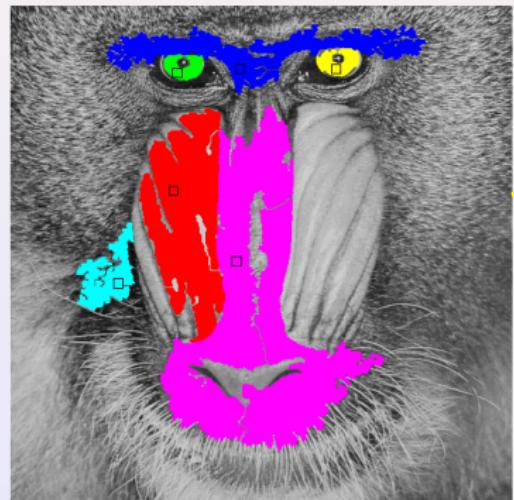
Croissance de régions - Region growing

- on choisit une (ou plusieurs) "graine(s)"
- La région R (pour l'instant réduite à un point) possède une moyenne μ_R et un écart-type σ_R
- On ajoute alors à R tous les pixels voisins de R qui sont suffisamment semblables à R , exple

$$|I(x) - \mu_R| < \text{seuil}$$

ou bien

$$\left\{ \begin{array}{l} \min\{|I(x) - I(y)|; y \in R \cap V(x)\} < \text{seuil} \\ |I(x) - \mu_R| < \sigma_R \end{array} \right.$$



Croissance de régions - Region growing

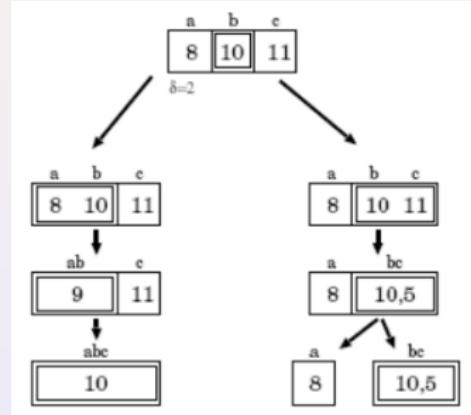
Limitations :

- Influence du choix des graines

Croissance de régions - Region growing

Limitations :

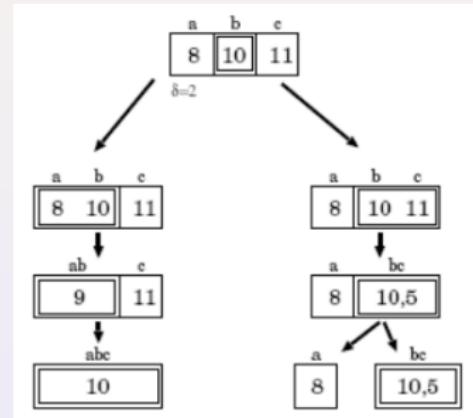
- Influence du choix des graines
- Influence de l'ordre de parcours des points de la frontière



Croissance de régions - Region growing

Limitations :

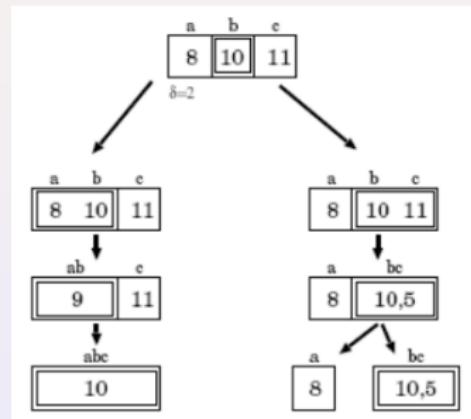
- Influence du choix des graines
- Influence de l'ordre de parcours des points de la frontière
- choix du seuil



Croissance de régions - Region growing

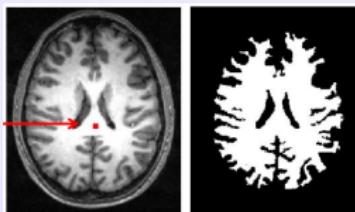
Limitations :

- Influence du choix des graines
- Influence de l'ordre de parcours des points de la frontière
- choix du seuil



Avantage :

- Implémentation : très rapide, si l'on utilise une structure de données adaptée (files d'attente).

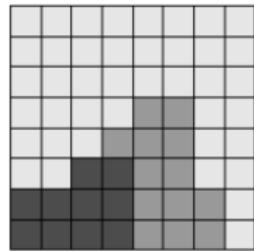


Split and Merge

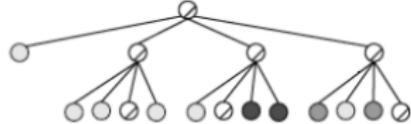
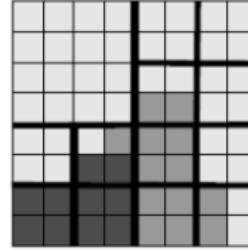
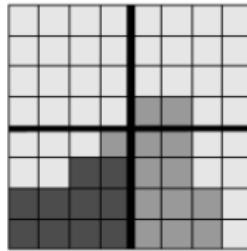
- Algorithme “split and merge” [Pavlidiset Horowitz, 1974]
 - Le principe consiste à (sur-)diviser l’image en régions homogènes (**split**) que l’on va ensuite regrouper (**merge**)
 - étape **split** : on crée une partition de l’image par division récursive en régions de taille identique lorsqu’un critère d’homogénéité n’est pas satisfait.
 - étape **merge** : on utilise le graphe d’adjacence créé pendant le **split** pour regrouper des régions voisines et suffisamment homogènes.

Split and Merge

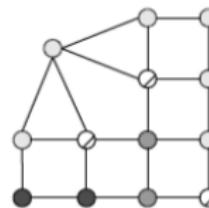
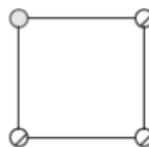
- Illustration de l'algorithme : **SPLIT**



Quadtree
 \emptyset

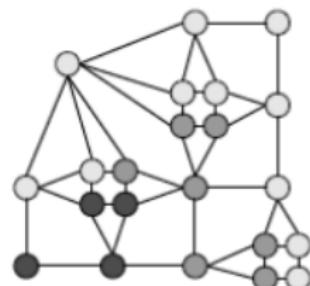
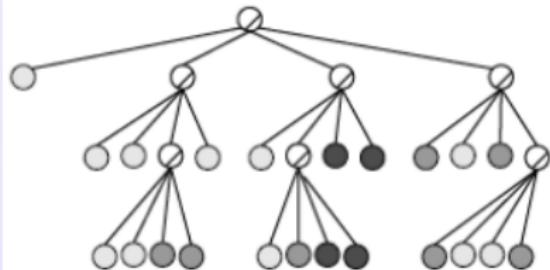
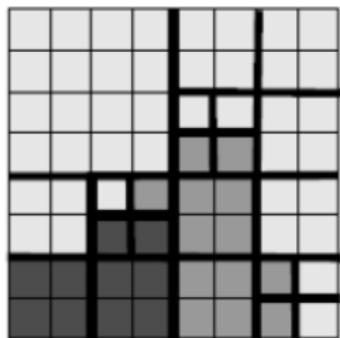


Graphe d'adjacence



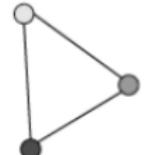
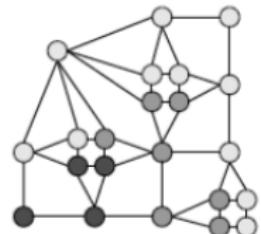
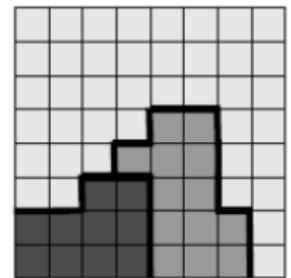
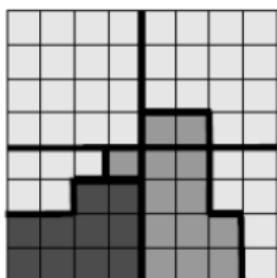
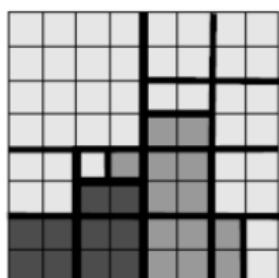
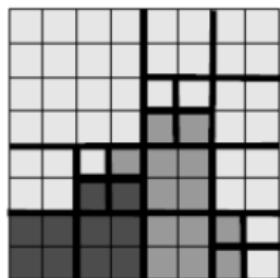
Split and Merge

- Illustration de l'algorithme : **SPLIT**
 - Résultat final de cette étape : sur-segmentation



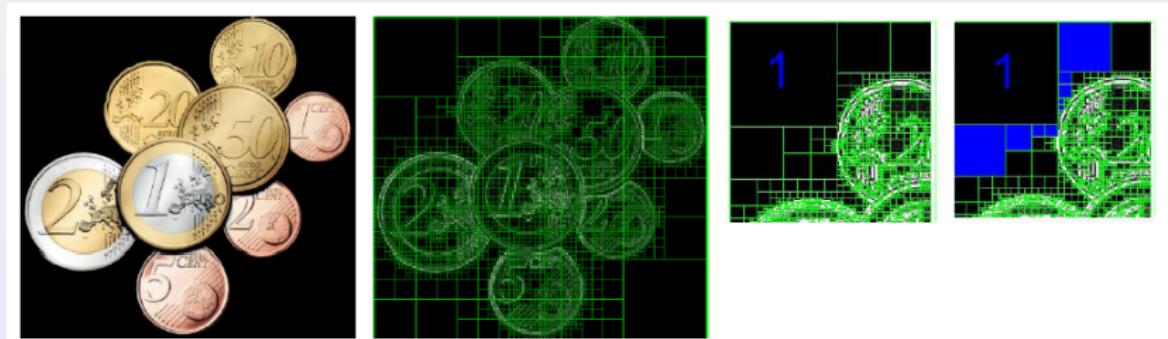
Split and Merge

- Illustration de l'algorithme : **MERGE**



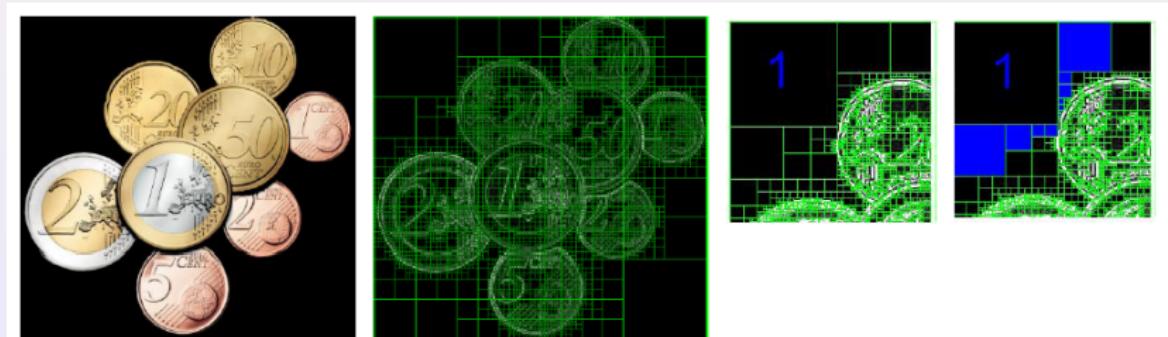
Split and Merge

- La phase SPLIT crée une sur-segmentation de l'image que la phase MERGE vient corriger
- La phase MERGE → croissance de régions



Split and Merge

- La phase SPLIT crée une sur-segmentation de l'image que la phase MERGE vient corriger
- La phase MERGE → croissance de régions



- critère d'homogénéité :
 - sur les extrema des régions $\max_R(I(x,y)) - \min_R(I(x,y)) < \epsilon$
 - sur la variance au sein de la région $\sum_R (I(x,y) - \mu_R)^2 < \epsilon$

Split and Merge

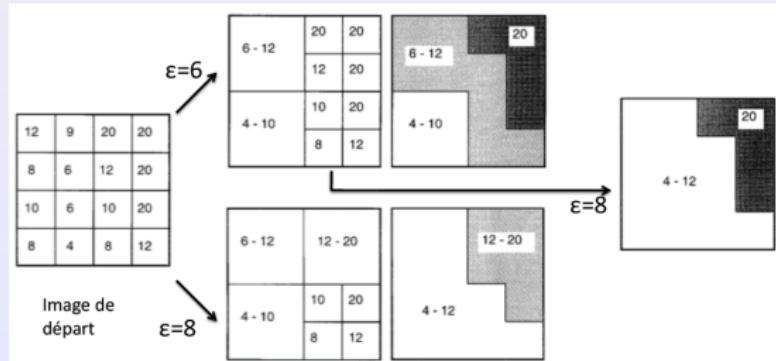
Amélioration

- une sorte de recuit simulé (Manousakas et al. 1998)
- constat :
 - ϵ trop grand \rightarrow splits pas assez nombreux
 - ϵ trop petit \rightarrow pas assez de regroupements
- On utilise $\epsilon/2$ pour la phase de split puis on regroupe en passant progressivement de $\epsilon/2$ à ϵ

Split and Merge

Amélioration

- une sorte de recuit simulé (Manousakas et al. 1998)
- constat :
 - ϵ trop grand \rightarrow splits pas assez nombreux
 - ϵ trop petit \rightarrow pas assez de regroupements
- On utilise $\epsilon/2$ pour la phase de split puis on regroupe en passant progressivement de $\epsilon/2$ à ϵ



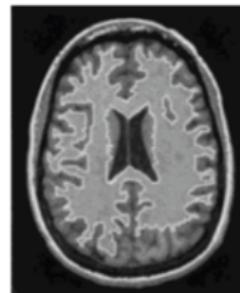
Split and Merge

Amélioration

Image originelle



SM classique

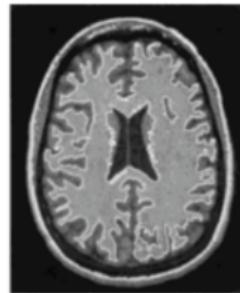


SM classique



SM avec recuit simulé

SM avec recuit simulé



Split and Merge

Avantage :

- méthode hybride (locale/globale)

Inconvénients :

- crée des structures carrées dans la segmentation de l'image
- comme pour la croissance de régions : sensibilité à l'ordre de parcours des régions

Il existe encore d'autres méthodes

- CSC (Color Structure Code), Hartmann 1987, Rehrmann et Priese 1993
- MDL (Minimum Length Description), Thomas C. M. Lee 1999

Méthodes markoviennes

- En restauration d'images (méthodes variationnelles), on cherche à minimiser une énergie de la forme : $E_{\text{données}}(u) + E_{\text{régularité}}(u)$
- Dans le cadre bayésien, on donne une interprétation statistique
- Formule de Bayes :



$$\mathbf{P}(A|B) = \frac{\mathbf{P}(B|A) \mathbf{P}(A)}{\mathbf{P}(B)}$$

- On considère donc nos images comme des réalisations de variables aléatoires : $\mathbf{P}(u|f) = \frac{\mathbf{P}(f|u) \mathbf{P}(u)}{\mathbf{P}(f)}$ où f est l'image observée (à traiter), et u est l'inconnue.

Méthodes markoviennes

- On définit/modélise donc $\mathbf{P}(f|u)$ et $\mathbf{P}(u)$
- Puis on cherche le u qui **maximise** la probabilité $\mathbf{P}(u|f)$. Donc, pour un f fixé, qui maximise $\mathbf{P}(f|u) \mathbf{P}(u)$
- Lien avec méthodes variationnelles : revient à **minimiser**

$$E(u) = -\log \mathbf{P}(u) - \log \mathbf{P}(f|u)$$

- probabilité **a priori** $\mathbf{P}(u)$ ← critère de régularité

- en général



moins probable que



(si images "naturelles")

- débruitage : image régulière plus probable que du bruit

- segmentation :



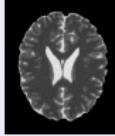
moins probable que



Méthodes markoviennes

- **vraisemblance $P(f | u)$** : modélise le processus de formation de l'image

- débruitage : si $u =$  alors  plus probable que 

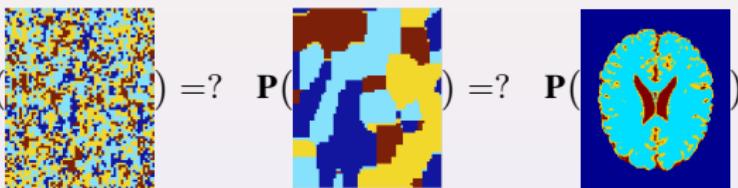
- segmentation : si $u =$  alors  plus probable que 

- Hypothèse de base : $P(f | u)$, $P(u)$ plus facile à modéliser que $P(u | f)$ (alors que c'est ce dernier qui nous intéresse) → utilisation de la formule de Bayes

Méthodes markoviennes en segmentation

- probabilité **a priori** $P(u)$

- $P(\text{Image}) = ? \quad P(\text{Segmentation}) = ? \quad P(\text{Brain}) = ?$



The image consists of three side-by-side square panels. The first panel on the left shows a noisy, multi-colored image with no discernible patterns. The second panel in the middle shows a segmented image where the same noisy pattern has been divided into several distinct regions, each colored blue, yellow, or red. The third panel on the right shows a grayscale image of a human brain, with specific regions highlighted in yellow and red.

Méthodes markoviennes en segmentation

- probabilité **a priori** $\mathbf{P}(u)$

- $\mathbf{P}(\text{image bruitée}) = ?$ $\mathbf{P}(\text{image segmentée}) = ?$ $\mathbf{P}(\text{cerveau segmenté}) = ?$

- critère de régularité pour des images à m classes \longrightarrow modèle de Potts

$$\mathbf{P}(u) = \frac{1}{Z} \exp \left(-\beta \sum_{\langle s,t \rangle} \varphi(u_s, u_t) \right)$$

où $\langle s, t \rangle$ = "les pixel s et t sont voisins" et $\varphi(a, b) = \begin{cases} -1, & \text{si } a = b \\ 1, & \text{si } a \neq b \end{cases}$

Méthodes markoviennes en segmentation

- probabilité **a priori** $\mathbf{P}(u)$

- $\mathbf{P}(\text{image bruitée}) = ?$ $\mathbf{P}(\text{image segmentée}) = ?$ $\mathbf{P}(\text{image cerveau}) = ?$

- critère de régularité pour des images à m classes \rightarrow modèle de Potts

$$\mathbf{P}(u) = \frac{1}{Z} \exp \left(-\beta \sum_{\langle s,t \rangle} \varphi(u_s, u_t) \right)$$

où $\langle s, t \rangle$ = "les pixel s et t sont voisins" et $\varphi(a, b) = \begin{cases} -1, & \text{si } a = b \\ 1, & \text{si } a \neq b \end{cases}$

si $\beta > 0 \quad \rightarrow \quad \mathbf{P} \left(\text{image bruitée} \right) < \mathbf{P} \left(\text{image segmentée} \right) < \mathbf{P} \left(\text{image cerveau} \right)$

Méthodes markoviennes en segmentation

Modélisation Bayésienne Markovienne :

- vraisemblance des observations $P(f | u)$

$$P(\text{[gris]} \mid \text{[cerveau]}) = ? \quad P(\text{[bateau]} \mid \text{[cerveau]}) = ? \quad P(\text{[cerveau]} \mid \text{[cerveau]}) = ?$$

Méthodes markoviennes en segmentation

Modélisation Bayésienne Markovienne :

- vraisemblance des observations $P(f | u)$

$$P(\text{[Image Grise]} \mid \text{[Image cerveau]}) = ? \quad P(\text{[Image bateau]} \mid \text{[Image cerveau]}) = ? \quad P(\text{[Image noir]} \mid \text{[Image cerveau]}) = ?$$

- on connaît u (répartition et localisation des différentes zones texturées)

Méthodes markoviennes en segmentation

Modélisation Bayésienne Markovienne :

- vraisemblance des observations $\mathbf{P}(f | u)$

$$\mathbf{P}(\text{[Image grisée]} \mid \text{[Image cerveau]}) = ? \quad \mathbf{P}(\text{[Image bateau]} \mid \text{[Image cerveau]}) = ? \quad \mathbf{P}(\text{[Image cerveau]} \mid \text{[Image cerveau]}) = ?$$

- on connaît u (répartition et localisation des différentes zones texturées)
- Pour un u donné, la probabilité $\mathbf{P}(f|u)$ sera grande (par définition) si f est "cohérent" avec u , i.e. si les textures présentes dans f sont dans les zones indiquées par u

Méthodes markoviennes en segmentation

Modélisation Bayésienne Markovienne :

- vraisemblance des observations $\mathbf{P}(f | u)$

$$\mathbf{P}(\text{[grey square]} \mid \text{[brain texture]}) = ? \quad \mathbf{P}(\text{[ship image]} \mid \text{[brain texture]}) = ? \quad \mathbf{P}(\text{[brain grayscale]} \mid \text{[brain texture]}) = ?$$

- on connaît u (répartition et localisation des différentes zones texturées)
- Pour un u donné, la probabilité $\mathbf{P}(f|u)$ sera grande (par définition) si f est "cohérent" avec u , i.e. si les textures présentes dans f sont dans les zones indiquées par u
- on choisit de représenter une "texture" par : cste + bruit blanc

Méthodes markoviennes en segmentation

Modélisation Bayésienne Markovienne :

- vraisemblance des observations $\mathbf{P}(f | u)$

$$\mathbf{P}(\text{[grey texture]} \mid \text{[brain image]}) =? \quad \mathbf{P}(\text{[ship image]} \mid \text{[brain image]}) =? \quad \mathbf{P}(\text{[brain image]} \mid \text{[brain image]}) =?$$

- on connaît u (répartition et localisation des différentes zones texturées)
- Pour un u donné, la probabilité $\mathbf{P}(f|u)$ sera grande (par définition) si f est "cohérent" avec u , i.e. si les textures présentes dans f sont dans les zones indiquées par u
- on choisit de représenter une "texture" par : cste + bruit blanc
- Pour un u donné (à valeurs dans $\{0, \dots, M-1\}$), on choisit donc M textures, c'est à dire M moyennes μ_i et M écart-types σ_i

Méthodes markoviennes en segmentation

Modélisation Bayésienne Markovienne :

- vraisemblance des observations $\mathbf{P}(f | u)$

$$\mathbf{P}(\text{[grey square]} \mid \text{[brain texture]}) = ? \quad \mathbf{P}(\text{[ship image]} \mid \text{[brain texture]}) = ? \quad \mathbf{P}(\text{[brain grayscale]} \mid \text{[brain texture]}) = ?$$

- on connaît u (répartition et localisation des différentes zones texturées)
- Pour un u donné, la probabilité $\mathbf{P}(f|u)$ sera grande (par définition) si f est "cohérent" avec u , i.e. si les textures présentes dans f sont dans les zones indiquées par u
- on choisit de représenter une "texture" par : cste + bruit blanc
- Pour un u donné (à valeurs dans $\{0, \dots, M-1\}$), on choisit donc M textures, c'est à dire M moyennes μ_i et M écart-types σ_i
- On choisit donc de dire que la valeur de f au point s , qui fait partie de la classe $i = u_s$, va suivre une loi gaussienne : $\frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(f_s - \mu_i)^2}{2\sigma_i^2}\right)$

Méthodes markoviennes en segmentation

Maximum A Posteriori

estimateur MAP (Maximum A Posteriori) :

- u tel que $\mathbf{P}(U = u \mid F = f)$ maximum (pour f donné)
- i.e. u tel que $H(u,f)$ minimum

Méthodes markoviennes en segmentation

Maximum A Posteriori

estimateur MAP (Maximum A Posteriori) :

- u tel que $\mathbf{P}(U = u \mid F = f)$ maximum (pour f donné)
 - i.e. u tel que $H(u,f)$ minimum
-
- Comment minimiser ? énergie trop irrégulière pour descente de gradient ou similaire
 - Algorithme stochastique → **recuit simulé**
 - Besoin d'un outil permettant d'échantillonner une distribution de probabilité → **Algorithme de Metropolis-Hastings**

Méthodes markoviennes en segmentation

Algorithme de Metropolis

Principe

- On veut échantillonner la loi

$$\pi(x) = \frac{1}{Z} \exp\left(-H(x)\right) = \frac{1}{Z} \exp\left(-\sum_C V_C(x_C)\right)$$

- Principe : Transition pour un site courant s
 - Tirer une valeur λ uniformément dans Λ (ens. des valeurs possibles)
 - **Proposer** λ comme nouvelle valeur pour le site s (définissant une nouvelle image y)
 - **L'accepter** si elle fait diminuer H
 - **L'accepter** avec probabilité $\exp(H(x) - H(y))$ si elle fait augmenter H
 - Si λ est rejetée alors x_s ne change pas

Méthodes markoviennes en segmentation

Algorithme de Metropolis

Algorithme final

- choisir une image initiale x^0 aléatoirement
- à l'étape m , remettre à jour le site $s_{m[n]}$ selon la procédure précédente
- au bout d'un certain temps les images x^{r_n} obtenues suivent la loi $\pi(x)$

Application aux modèles d'Ising et Potts ($B = 0$)

DEMO MATLAB



$\beta = 0$



$\beta = 0.5$



$\beta = 1$



$\beta = 10$

Méthodes markoviennes en segmentation

Recuit simulé

- Étant donné une distribution de Gibbs $\pi(x) = \frac{1}{Z} \exp\left(-H(x)\right)$, on cherche le x tel que $H(x)$ soit minimal
- On échantillonne la distribution de Gibbs **avec température** associée :

$$\pi_T(x) = \frac{1}{Z_T} \exp\left(-\frac{H(x)}{T}\right)$$

avec $H(x) = \sum_{C \in \mathcal{C}} V_C(x_C)$ et $Z_T = \sum_{y \in \Omega} \exp\left(-\frac{H(y)}{T}\right)$ en faisant tendre T vers 0

- si décroissance logarithmique : convergence assurée
- en pratique : décroissance arithmétique

Méthodes markoviennes en segmentation

méthode ICM (Iterated Conditional Modes)

- Sorte de recuit simulé à température nulle (proposé par Besag en 1983)
- algorithme déterministe : au site courant **maximiser la descente d'énergie**

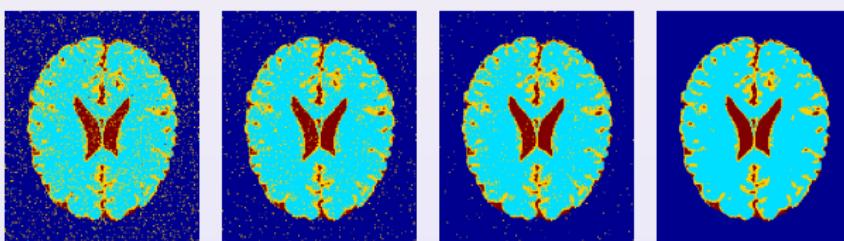
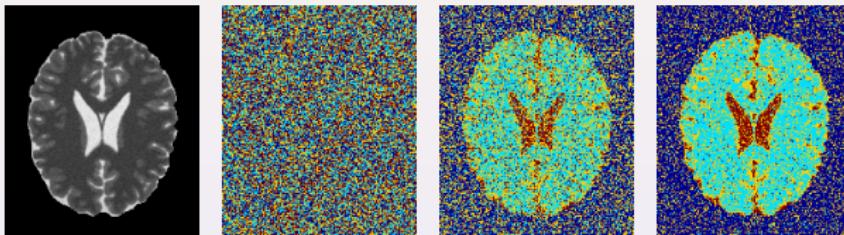
- ▷ Initialisation x^0 proche de la solution
- ▷ Suite d'images x^n
- ▷ à l'étape n , on balaye tous les sites s et

$$x_s^n \leftarrow \arg \max_{\lambda \in \Lambda} \mathbf{P}(X_s = \lambda | y, x_{\partial(s)}^{n-1})$$

Propriétés

- Algorithme **déterministe**, résultat dépendant de l'initialisation
- Convergence rapide (quelques balayages)
- Risque de converger vers un minimum local de H

Exemple : segmentation texturale (recuit simulé)



■ matière blanche

■ matière grise

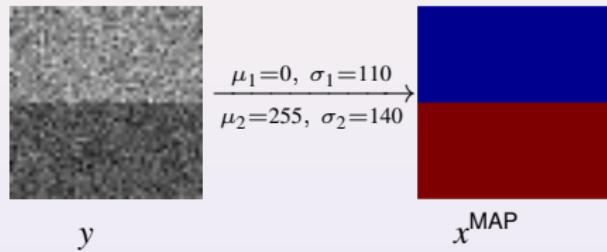
■ liquide céphalo-rachidien

DEMO MATLAB

DEMO MATLAB

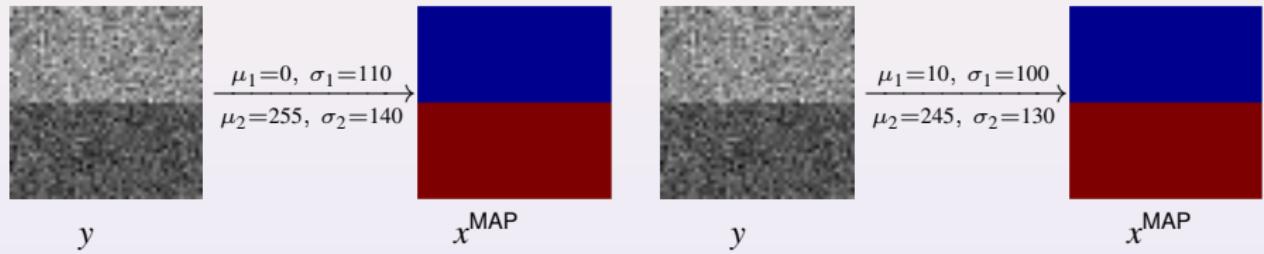
Méthodes markoviennes en segmentation

- Problème : dépend beaucoup des choix qu'on a fait pour μ_i et σ_i



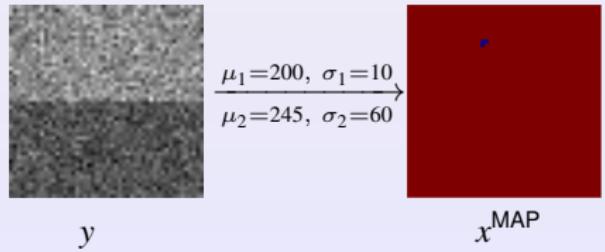
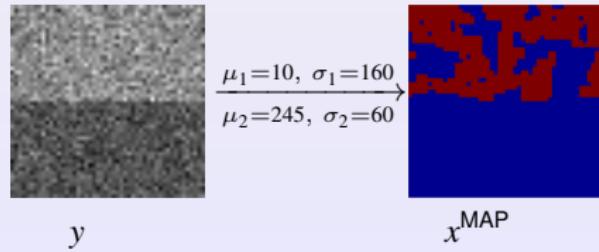
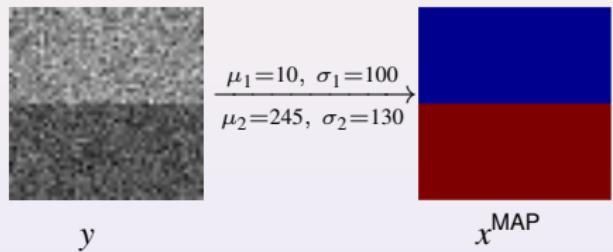
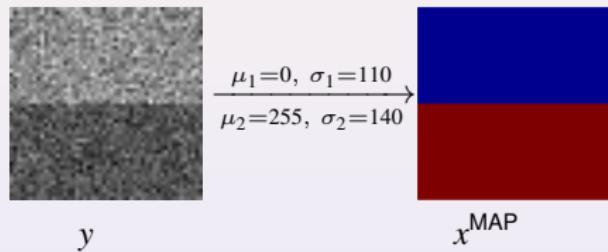
Méthodes markoviennes en segmentation

- Problème : dépend beaucoup des choix qu'on a fait pour μ_i et σ_i



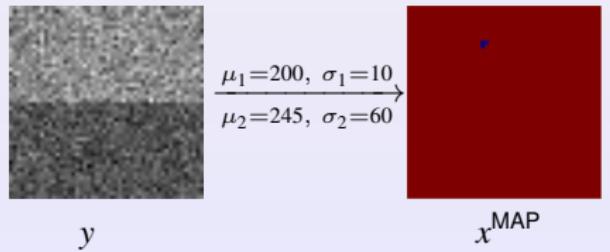
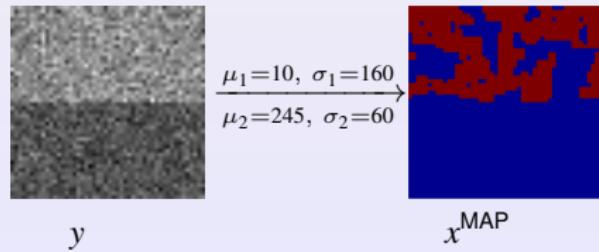
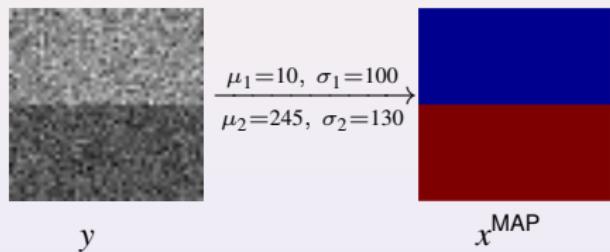
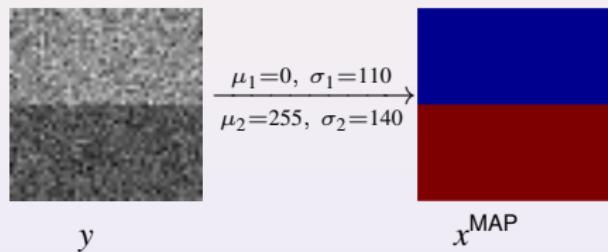
Méthodes markoviennes en segmentation

- Problème : dépend beaucoup des choix qu'on a fait pour μ_i et σ_i



Méthodes markoviennes en segmentation

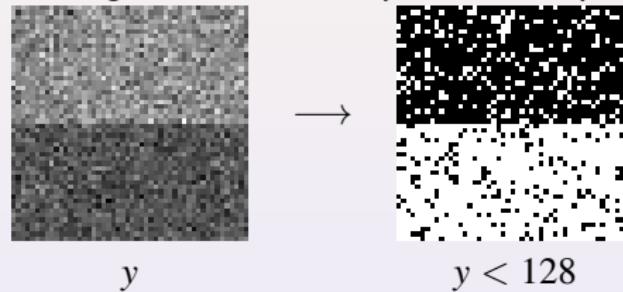
- Problème : dépend beaucoup des choix qu'on a fait pour μ_i et σ_i



- Comment estimer/choisir ces paramètres *avant* de lancer l'optimisation ?

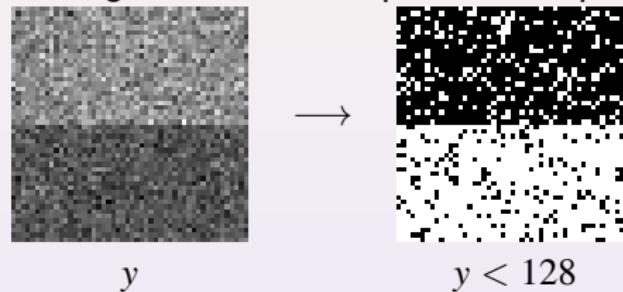
Méthodes markoviennes en segmentation

- utiliser une première segmentation basique → exemple : seuillage



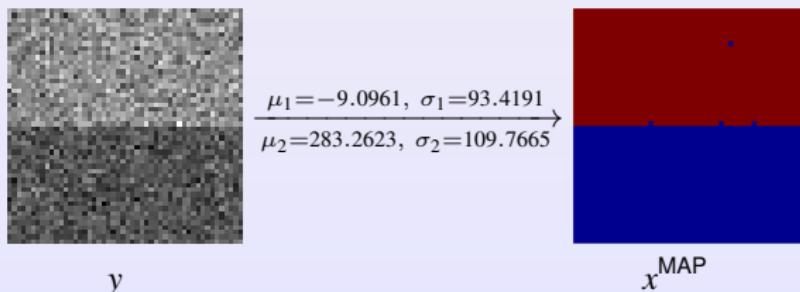
Méthodes markoviennes en segmentation

- utiliser une première segmentation basique → exemple : seuillage



- $\text{mean}(\text{Y}(\text{Y}<128)) \rightarrow -9.0961$
- $\text{std}(\text{Y}(\text{Y}<128)) \rightarrow 93.4191$

$$\begin{aligned}\text{mean}(\text{Y}(\text{Y}>=128)) &\rightarrow 283.2623 \\ \text{std}(\text{Y}(\text{Y}>=128)) &\rightarrow 109.7665\end{aligned}$$



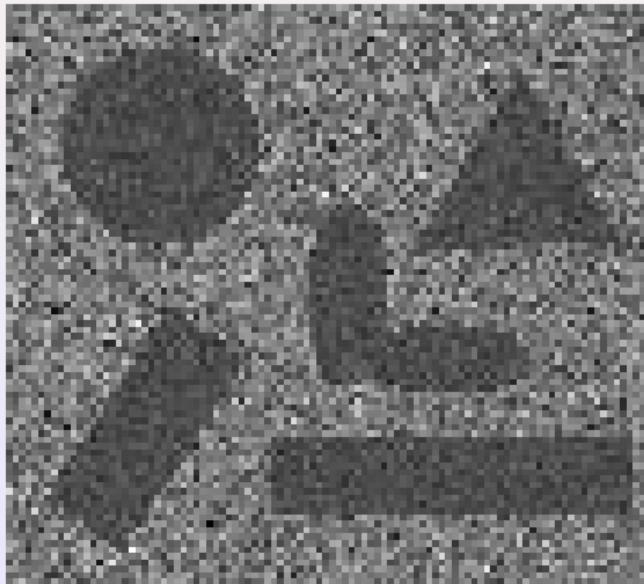
Méthodes markoviennes en segmentation

- Mieux (mais plus long) → utiliser une méthode d'espérance-maximisation
- Algorithme EM¹
 - On initialise les paramètres μ_i et σ_i , puis on alterne entre
 - 1 Segmentation en utilisant les μ_i et σ_i actuels
 - 2 Remise à jour des μ_i et σ_i , en calculant moyenne et écart-type de chaque classe obtenue à l'étape précédente
 - On s'arrête lorsque les paramètres ne changent plus suffisamment (et donc la segmentation associée non plus)

1. http://fr.wikipedia.org/wiki/Algorithme_espérance-maximisation

Méthodes markoviennes en segmentation

Algorithme EM. exemple : segmentation basée texture



$$\begin{array}{ll} \mu_1 = 128 & \sigma_1 = 30 \\ \mu_2 = 255 & \sigma_2 = 90 \end{array}$$

DEMO MATLAB

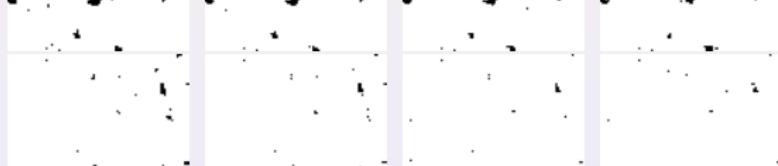
Méthodes markoviennes en segmentation

Algorithme EM. exemple : segmentation basée texture

$$\begin{array}{ll} \mu_1 = 50 & \sigma_1 = 155 \\ \mu_2 = 100 & \sigma_2 = 60 \end{array}$$



$$\begin{array}{ll} \mu_1 = 283 & \sigma_1 = 260 \\ \mu_2 = 212 & \sigma_2 = 111 \end{array}$$



$$\begin{array}{ll} \mu_1 = 275 & \sigma_1 = 112 \\ \mu_2 = 215 & \sigma_2 = 96 \end{array}$$



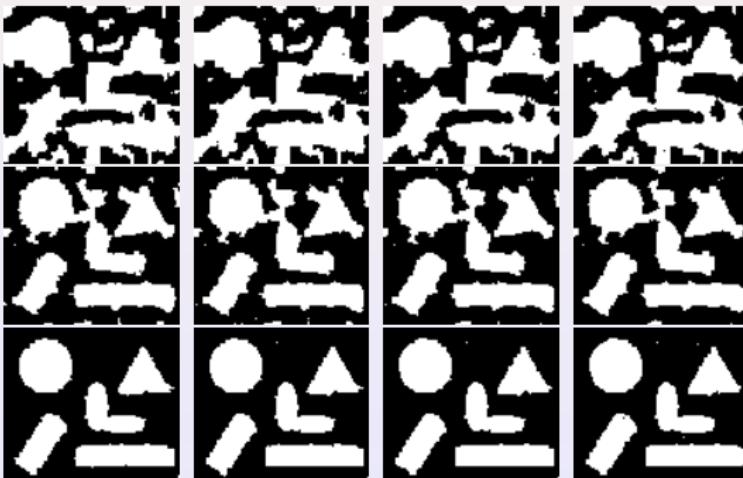
$$\begin{array}{ll} \mu_1 = 264 & \sigma_1 = 94 \\ \mu_2 = 197 & \sigma_2 = 93 \end{array}$$



Méthodes markoviennes en segmentation

Algorithme EM. exemple : segmentation basée texture

$$\begin{array}{ll} \mu_1 = 261 & \sigma_1 = 90 \\ \mu_2 = 181 & \sigma_2 = 89 \end{array}$$



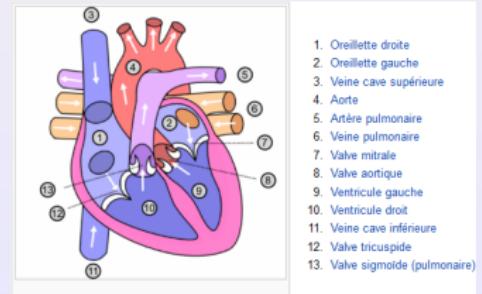
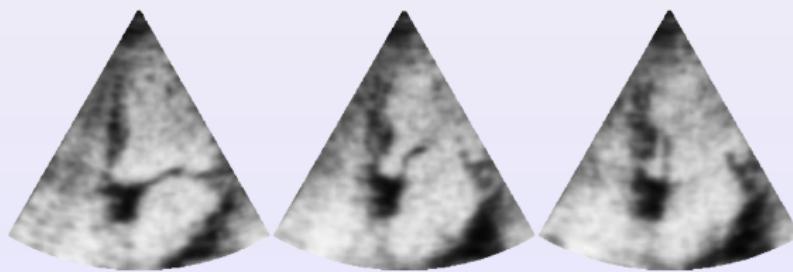
$$\begin{array}{ll} \mu_1 = 262 & \sigma_1 = 89 \\ \mu_2 = 167 & \sigma_2 = 81 \end{array}$$

$$\begin{array}{ll} \mu_1 = 257 & \sigma_1 = 90 \\ \mu_2 = 130 & \sigma_2 = 39 \end{array}$$

$$\begin{array}{ll} \mu_1 = 256 & \sigma_1 = 90 \\ \mu_2 = 127 & \sigma_2 = 30 \end{array}$$

Segmentation Markovienne en imagerie US cardiaque

- Segmentation du ventricule gauche du coeur
- difficultés : le bruit, la valve mitrale qui peut être ouverte
- propriétés : une certaine uniformité des n.d.g. dans la cavité, des contours assez marqués près du myocarde



Segmentation Markovienne en imagerie US cardiaque

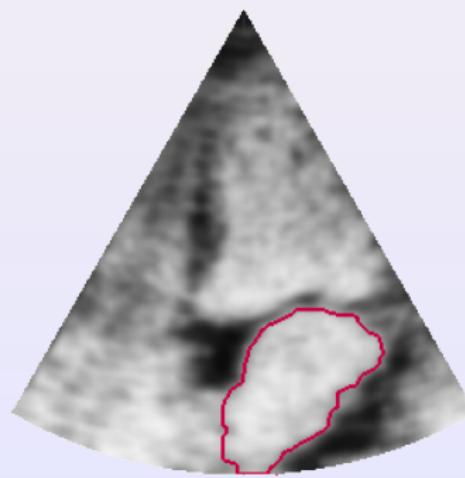
Modèle

- y_s niveau de gris du pixel s , $x_s = -1, x_s = 1$ (segmentation)
- Énergie :
$$H(x, y) = \sum_{s|x_s=1} \left(\frac{(y_s - \mu)^2}{\sigma^2} \right) - \alpha \sum_{\langle s, t \rangle} x_s x_t$$
- $\langle s, t \rangle$ système de cliques aux 4 plus proches voisins
- μ et σ peuvent être estimés : maximum de vraisemblance (sur l'ensemble des pixels tels que $x_s = 1$)
- On peut complexifier un peu pour ajouter prise en compte des discontinuités (cf MATI)

Segmentation Markovienne en imagerie US cardiaque

paramètres/résultat

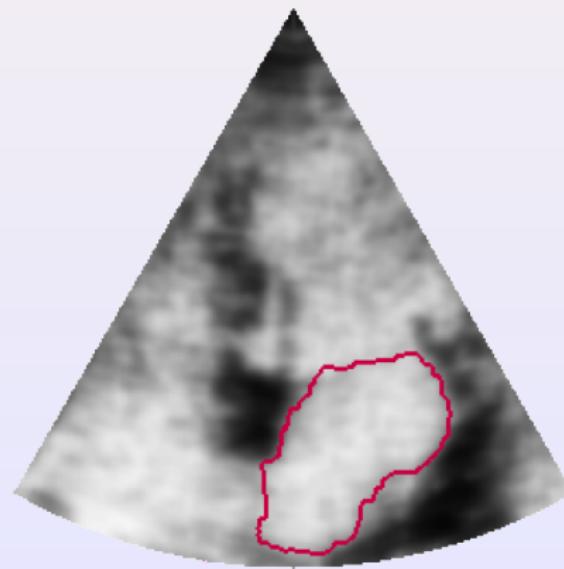
- α est choisi empiriquement, peut venir d'a priori sur l'image
- il faut initialiser
- il faut optimiser : recuit simulé / ICM



Segmentation Markovienne en imagerie US cardiaque

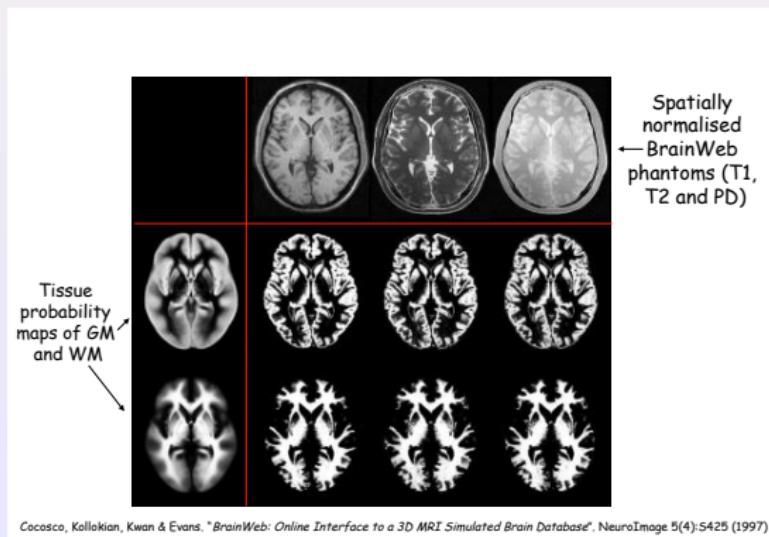
cohérence temporelle

- Ces modèles ont l'avantage d'être assez souple : on modifie l'énergie pour prendre en compte la cohérence temporelle de notre segmentation



Segmentation Markovienne : imagerie cérébrale

- "Unified segmentation" (Ashburner, Friston 2005) : modèle probabiliste/bayésien prenant en compte **recalage, correction biais et a priori cérébral (atlas)**



Méthodes markoviennes

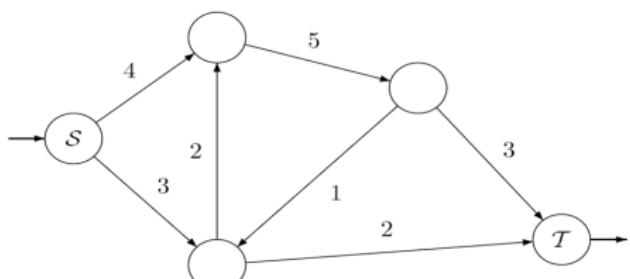
Bilan

- par rapport aux approches continues/variationnelles : modélisation du bruit, ajout d'**a priori**
- Très à la vogue dans les années 85-95 : elles ont été moins utilisées lorsque les images sont devenues trop grandes (temps de calcul)
- Regain d'intérêt depuis 5/10 ans grâce à l'apparition des techniques de coupure de graphe (Graph Cut) qui permettent une optimisation plus rapide

Approches d'optimisation par graphe : Méthode des "Graph-Cuts"²

Notations

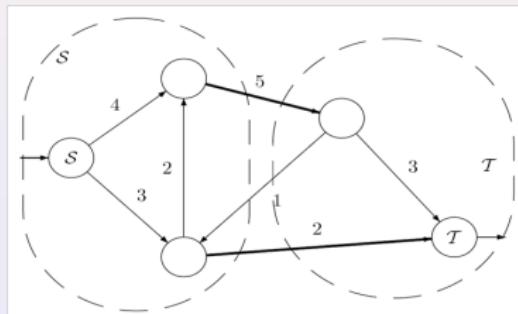
- $G = (V, E)$: Graphe Orienté
- V : sommets, E : arêtes
- pour chaque $(v_1, v_2) \in E$, une capacité w_{v_1, v_2} (poids des arêtes)
- S : source, T : puits



2. adapté de <http://mickaelpechaud.free.fr/graphcuts.pdf>

Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

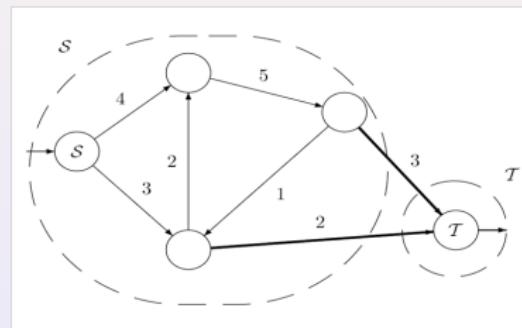
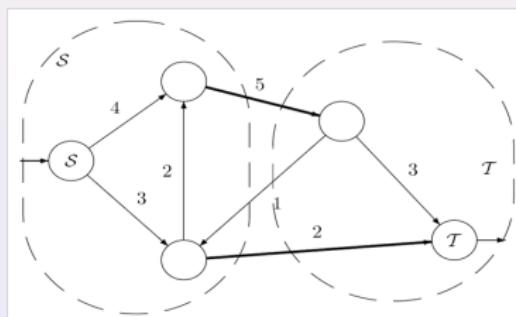
Coupe



- coupe = partition du graphe en deux régions \mathcal{S} et \mathcal{T} ($S \in \mathcal{S}$ et $T \in \mathcal{T}$)
- poids d'une coupe = $\sum_{\substack{(v_1, v_2) \in E \\ v_1 \in \mathcal{S}, v_2 \in \mathcal{T}}} w_{v_1, v_2}$
- coupe minimale = coupe de poids minimal

Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

Coupe



- coupe = partition du graphe en deux régions S et T ($S \in \mathcal{S}$ et $T \in \mathcal{T}$)
- poids d'une coupe = $\sum_{\substack{(v_1, v_2) \in E \\ v_1 \in S, v_2 \in T}} w_{v_1, v_2}$
- coupe minimale = coupe de poids minimal

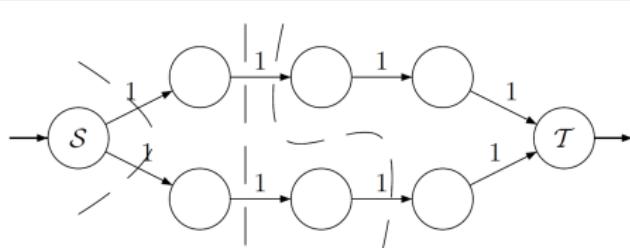
Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

Théorème

Rechercher une coupe minimale dans un graphe est un problème P.

Remarques :

- beaucoup de problèmes sur les graphes sont NP (exemple : recherche de la coupe maximale)
- il peut y avoir plusieurs solutions



Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

Reformulation en terme de flot maximal

On appelle flot, une fonction $f : E \mapsto \mathbb{R}$ vérifiant

- pour tout sommet $p \in V$ autre que S ou T , on a :

$$\sum_{e=(p,\cdot) \in E} f(e) = 0 \quad (1)$$

- pour toute arête $e \in E$, on a

$$f(e) \leq w_e \quad (2)$$

- (1) \rightarrow conservation du flot en chaque sommet
- (2) \rightarrow une arête ne peut contenir un flot dépassant sa capacité
- on définit la "valeur du flot"

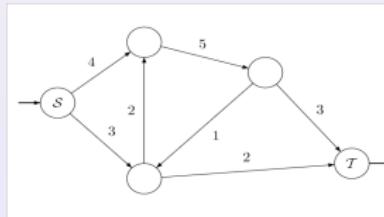
$$\sum_{e=(S,\cdot) \in E} f(e) = \sum_{e=(T,\cdot) \in E} f(e)$$

Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

Reformulation en terme de flot maximal

Théorème

Pour un graphe G vérifiant nos hypothèses, la valeur d'une coupe minimale est égale à la valeur d'un flot maximal.

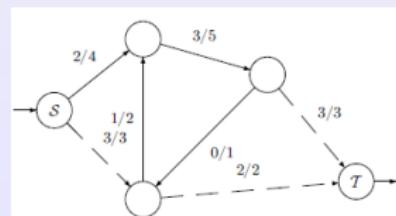
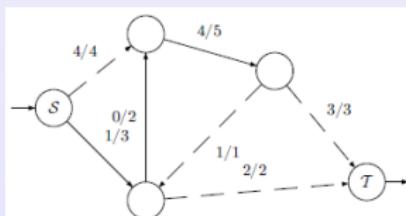
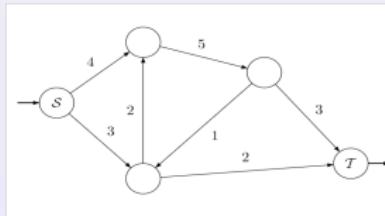


Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

Reformulation en terme de flot maximal

Théorème

Pour un graphe G vérifiant nos hypothèses, la valeur d'une coupe minimale est égale à la valeur d'un flot maximal.



Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

- De plus toute arête intervenant dans une coupe minimale est une arête saturée par un flot maximal
- Si on a trouvé un flot maximal, on regarde les arêtes saturées et on supprime itérativement les arêtes inutiles

Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

- De plus toute arête intervenant dans une coupe minimale est une arête saturée par un flot maximal
- Si on a trouvé un flot maximal, on regarde les arêtes saturées et on supprime itérativement les arêtes inutiles
- Pour trouver flot maximal, plusieurs méthodes :
 - **saturation de chemins** : à partir du flot nul, trouver itérativement un chemin de S à T sur lequel il n'y a pas d'arête saturée. On rajoute alors autant de flot que possible à ce chemin
 - **poussage de flot** : on envoie autant de flot que l'on peut à partir de S . On se retrouve alors avec des noeuds dits actifs, c'est à dire qui reçoivent un excès de flot. On pousse alors ce flot excessif vers d'autres noeuds disponibles.

Méthode des "Graph-Cuts" : utilisation en image

- Les "graph-cut" vont nous servir pour minimiser une énergie de la forme :

$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)$$

Méthode des "Graph-Cuts" : utilisation en image

- Les "graph-cut" vont nous servir pour minimiser une énergie de la forme :

$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)$$

- typiquement :

- $D \rightarrow$ attache aux données :
- $V \rightarrow$ critère de régularité ($\langle i,j \rangle \rightarrow i$ et j sont voisins)

Méthode des "Graph-Cuts" : utilisation en image

- Les "graph-cut" vont nous servir pour minimiser une énergie de la forme :

$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)$$

- typiquement :
 - D → attache aux données :
 - V → critère de régularité ($\langle i,j \rangle \rightarrow i$ et j sont voisins)
- Exemple : segmentation binaire d'une image bruitée I ($0 \leq I_i \leq 1$)
 - pour chaque pixel i , on cherche donc une étiquette $x_i \in \{0, 1\}$

Méthode des "Graph-Cuts" : utilisation en image

- Les "graph-cut" vont nous servir pour minimiser une énergie de la forme :

$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)$$

- typiquement :
 - D → attache aux données :
 - V → critère de régularité ($\langle i,j \rangle \rightarrow i$ et j sont voisins)
- Exemple : segmentation binaire d'une image bruitée I ($0 \leq I_i \leq 1$)
 - pour chaque pixel i , on cherche donc une étiquette $x_i \in \{0, 1\}$
 - on prend

$$\begin{cases} D_i(0) = I_i \\ D_i(1) = 1 - I_i \end{cases} \quad \text{et} \quad V_{ij}(x_i, x_j) = \begin{cases} 0 \text{ si } x_i = x_j \\ 1 \text{ sinon} \end{cases}$$

Méthode des "Graph-Cuts" : utilisation en image



Méthode des "Graph-Cuts" : utilisation en image



Seuillage
→



Méthode des "Graph-Cuts" : utilisation en image



Seuillage
→



Seuillage
→



Méthode des "Graph-Cuts" : utilisation en image

Minimisation (par Graph-Cuts) de l'énergie précédente



$$\alpha = 0.2$$



$$\alpha = 0.3$$



$$\alpha = 0.4$$



$$\alpha = 0.5$$

Méthode des "Graph-Cuts" : utilisation en image

- Les "graph-cut" vont nous servir pour minimiser une énergie de la forme :

$$E(x) = \underbrace{\sum_i D_i(x_i)}_{\text{terme de fidélité, attache aux données}} + \alpha \underbrace{\sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)}_{\text{régularisation}}$$

Méthode des "Graph-Cuts" : utilisation en image

- Les "graph-cut" vont nous servir pour minimiser une énergie de la forme :

$$E(x) = \underbrace{\sum_i D_i(x_i)}_{\text{terme de fidélité, attache aux données}} + \alpha \underbrace{\sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)}_{\text{régularisation}}$$

- Lien avec le cadre Bayésien :

$$\begin{aligned}\mathbf{P}(X|I) &= \mathbf{P}(I|X) \mathbf{P}(X) \\ \Rightarrow \log(\mathbf{P}(X|I)) &= \log(\mathbf{P}(I|X)) + \log(\mathbf{P}(X))\end{aligned}$$

Méthode des "Graph-Cuts" : utilisation en image

- Les "graph-cut" vont nous servir pour minimiser une énergie de la forme :

$$E(x) = \underbrace{\sum_i D_i(x_i)}_{\text{terme de fidélité, attache aux données}} + \alpha \underbrace{\sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)}_{\text{régularisation}}$$

- Lien avec le cadre Bayésien :

$$\mathbf{P}(X|I) = \mathbf{P}(I|X) \mathbf{P}(X)$$
$$\Rightarrow \log(\mathbf{P}(X|I)) = \underbrace{\log(\mathbf{P}(I|X))}_{\text{terme de fidélité, attache aux données}} + \underbrace{\log(\mathbf{P}(X))}_{\text{régularisation}}$$

Méthode des "Graph-Cuts" : utilisation en image

- Les "graph-cut" vont nous servir pour minimiser une énergie de la forme :

$$E(x) = \underbrace{\sum_i D_i(x_i)}_{\text{terme de fidélité, attache aux données}} + \alpha \underbrace{\sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)}_{\text{régularisation}}$$

- Lien avec le cadre Bayésien :

$$\mathbf{P}(X|I) = \mathbf{P}(I|X) \mathbf{P}(X)$$

$$\Rightarrow \log(\mathbf{P}(X|I)) = \underbrace{\log(\mathbf{P}(I|X))}_{\text{terme de fidélité, attache aux données}} + \underbrace{\log(\mathbf{P}(X))}_{\text{régularisation}}$$

- Dans le cas général, minimiser

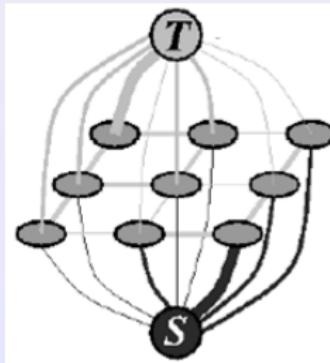
$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)$$

est un problème NP-complet ("arg")

Méthode des "Graph-Cuts" : utilisation en image

Principe général

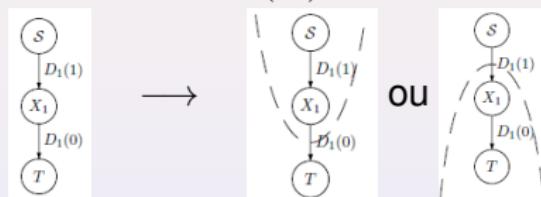
- On transforme le problème de minimisation en problème de coupe minimal dans un graphe
- On construit un graphe constitué d'un sommet par variable de l'énergie (pour chaque pixel)
- On ajoute une source et un puits, qui vont représenter en quelque sorte les valeurs possibles de chaque variable (0 pour la source et 1 pour le puits)



Méthode des "Graph-Cuts" : utilisation en image

exemples simples

- Un seul pixel : on veut minimiser $D_1(x_1)$.

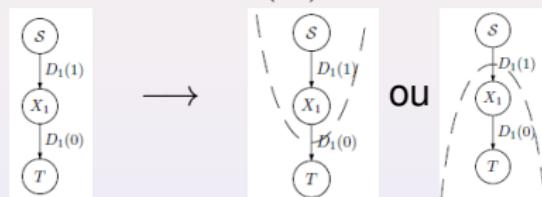


- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.

Méthode des "Graph-Cuts" : utilisation en image

exemples simples

- Un seul pixel : on veut minimiser $D_1(x_1)$.

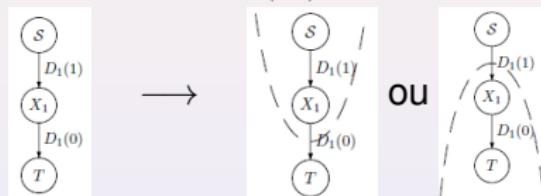


- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.
- avec $D_1(0) = I_1$, $D_1(1) = 1 - I_1 \rightarrow$

Méthode des "Graph-Cuts" : utilisation en image

exemples simples

- Un seul pixel : on veut minimiser $D_1(x_1)$.

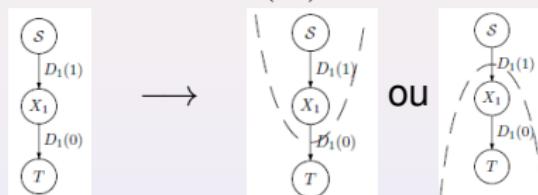


- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.
- avec $D_1(0) = I_1$, $D_1(1) = 1 - I_1 \rightarrow$ seuillage à $\frac{1}{2}$

Méthode des "Graph-Cuts" : utilisation en image

exemples simples

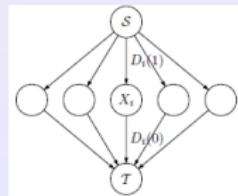
- Un seul pixel : on veut minimiser $D_1(x_1)$.



- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.

- avec $D_1(0) = I_1$, $D_1(1) = 1 - I_1 \rightarrow$ seuillage à $\frac{1}{2}$

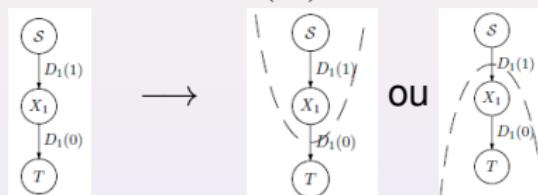
- n pixels sans interaction ($\alpha = 0$) : on veut minimiser $\sum_i D(x_i)$.



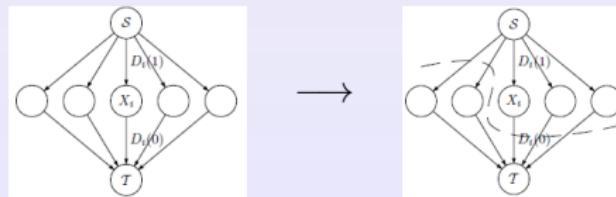
Méthode des "Graph-Cuts" : utilisation en image

exemples simples

- Un seul pixel : on veut minimiser $D_1(x_1)$.



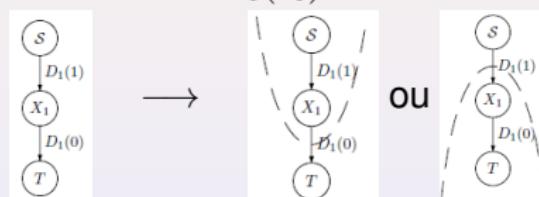
- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.
- avec $D_1(0) = I_1$, $D_1(1) = 1 - I_1 \rightarrow$ seuillage à $\frac{1}{2}$
- n pixels sans interaction ($\alpha = 0$) : on veut minimiser $\sum_i D(x_i)$.



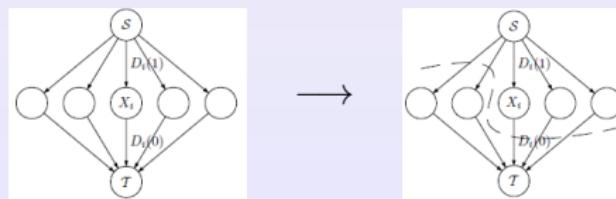
Méthode des "Graph-Cuts" : utilisation en image

exemples simples

- Un seul pixel : on veut minimiser $D_1(x_1)$.



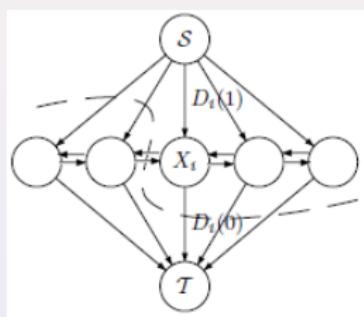
- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.
- avec $D_1(0) = I_1$, $D_1(1) = 1 - I_1 \rightarrow$ seuillage à $\frac{1}{2}$
- n pixels sans interaction ($\alpha = 0$) : on veut minimiser $\sum_i D(x_i)$.



Indep. des variables donc solution : $x_i = 0$ si $D_i(0) < D_i(1)$ (seuillage à $\frac{1}{2}$)

Méthode des "Graph-Cuts" : utilisation en image

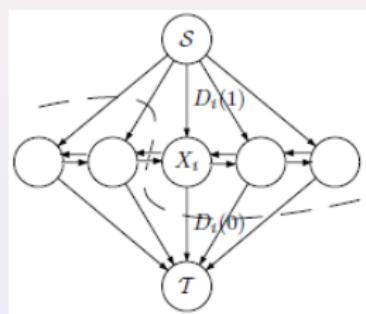
- n pixels avec interactions ($\alpha > 0$)



- il faut que la valeur d'une coupe, corresponde à l'énergie de la configuration associée Ainsi :
 Coupe minimale \leftrightarrow énergie minimale
- Les poids des arêtes sont plus compliqués à définir (en fonction de l'énergie, i.e. de D et V)
- Principe : deux voisins seront reliés par une arête de poids fort s'ils sont de valeurs proches

Méthode des "Graph-Cuts" : utilisation en image

- n pixels avec interactions ($\alpha > 0$)



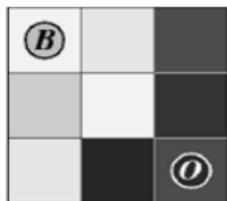
- il faut que la valeur d'une coupe, corresponde à l'énergie de la configuration associée Ainsi :
Coupe minimale \leftrightarrow énergie minimale
- Les poids des arêtes sont plus compliqués à définir (en fonction de l'énergie, i.e. de D et V)
- Principe : deux voisins seront reliés par une arête de poids fort s'ils sont de valeurs proches

Attention !

Toutes les énergies de la forme précédente ne sont pas minimisables par Graph-Cut

- Mais si l'énergie à minimiser vérifie certaines propriétés, alors on trouve le **minimum global** et la minimisation est **rapide**

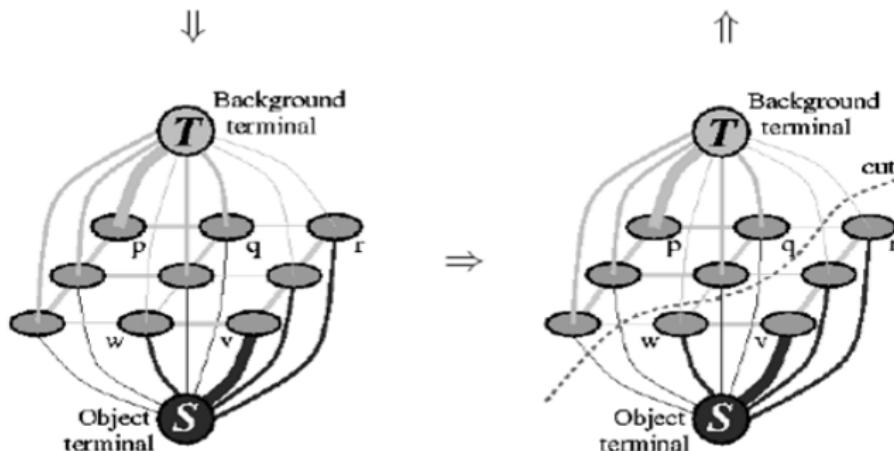
Méthode des "Graph-Cuts" :segmentation



(a) Image with seeds.

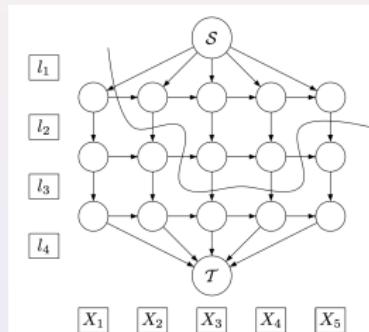


(d) Segmentation results.



Méthode des "Graph-Cuts" :segmentation

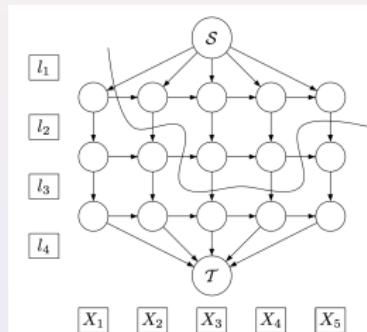
- Le cas multi-étiquette



$$\rightarrow (X_1 = l_1, X_2 = l_2, X_3 = l_3, X_4 = l_3, X_5 = l_2)$$

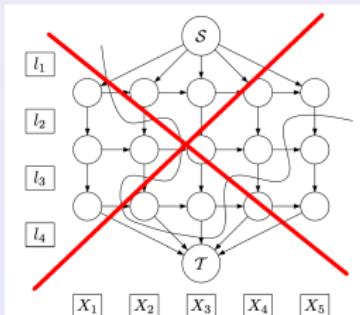
Méthode des "Graph-Cuts" :segmentation

- Le cas multi-étiquette



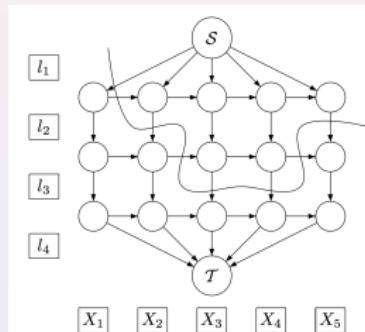
$$\rightarrow (X_1 = l_1, X_2 = l_2, X_3 = l_3, X_4 = l_3, X_5 = l_2)$$

- Attention, des précautions à prendre :



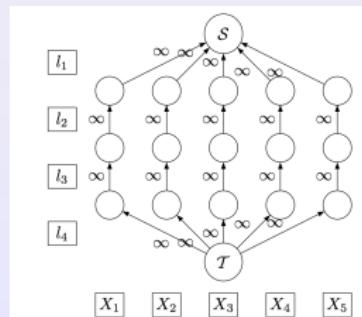
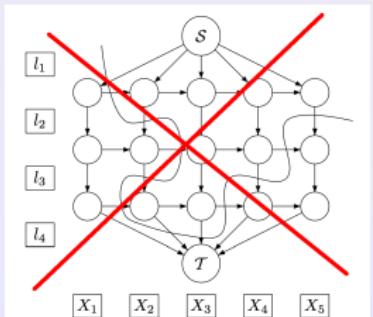
Méthode des "Graph-Cuts" :segmentation

- Le cas multi-étiquette



$$\rightarrow (X_1 = l_1, X_2 = l_2, X_3 = l_3, X_4 = l_3, X_5 = l_2)$$

- Attention, des précautions à prendre :



Méthode des "Graph-Cuts" :segmentation

- Le cas multi-étiquette



4 niveaux de gris

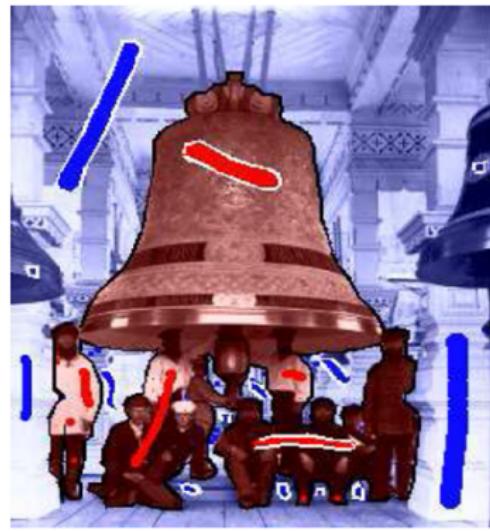


16 n.d.g



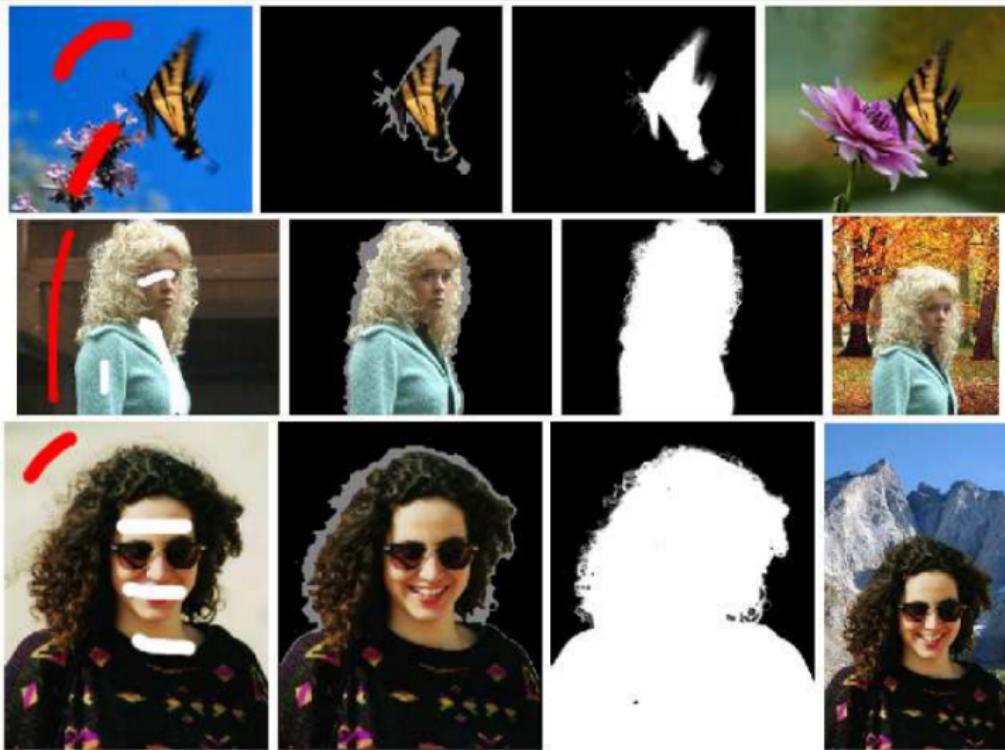
64 n.v.g

Méthode des "Graph-Cuts" : segmentation

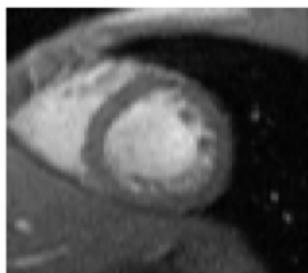


Boykov and Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. 2001. On peut initialiser avec des "graines" (liens infini entre le puit et l'objet et la source et le fond par exemple)

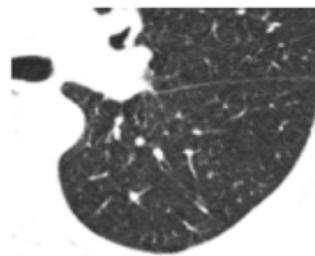
Méthode des "Graph-Cuts" : segmentation



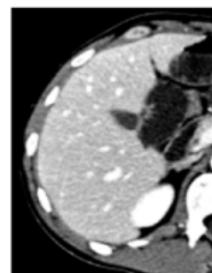
Méthode des "Graph-Cuts" :segmentation



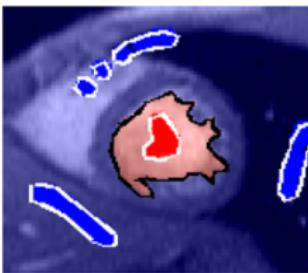
(c) Cardiac MR



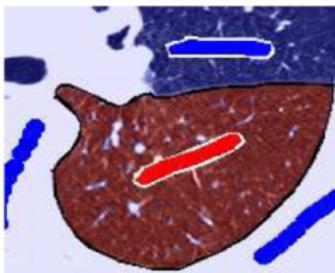
(e) Lung CT



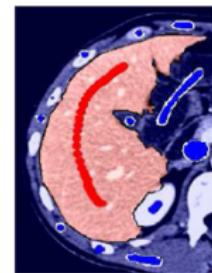
(g) Liver MR



(d) LV Segment

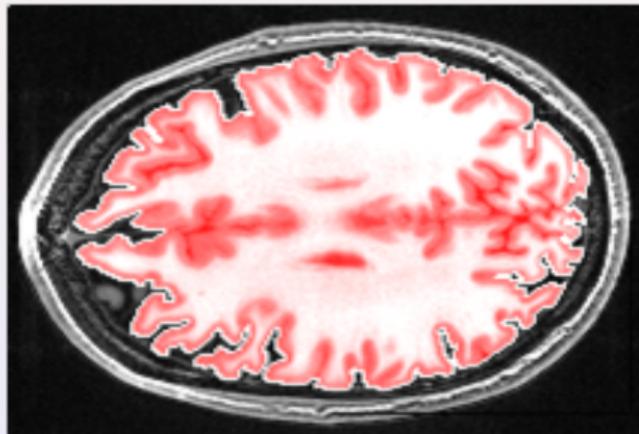


(f) Lobe Segment



(h) Liver Segment

Méthode des "Graph-Cuts" :segmentation



- Temps moyen en 2D (438×522) : 250ms
- Temps moyen en 3D ($256 \times 256 \times 124$) : 9.5s
- Pour aller plus loin
 - <http://mickaelpechaud.free.fr/graphcuts.pdf>
 - et références associées

Approches "contours"

Approches "contours"

- contour = ensemble des pixels séparant deux régions
- correspond à une importante variation locale d'intensité

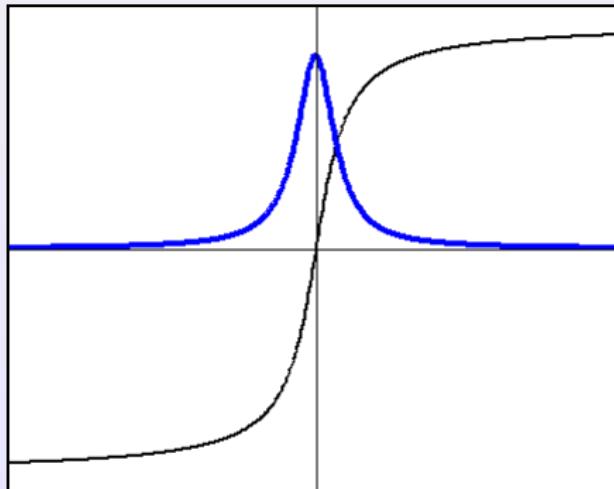
Approches "contours"

- contour = ensemble des pixels séparant deux régions
- correspond à une importante variation locale d'intensité
 - on regarde la variation d'**intensité** autour d'un contour



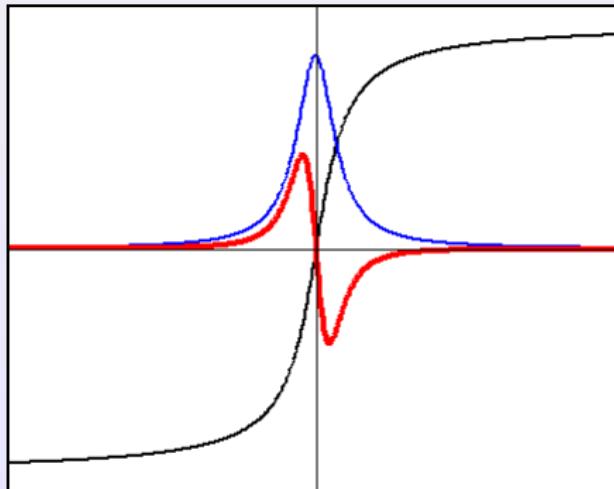
Approches "contours"

- contour = ensemble des pixels séparant deux régions
- correspond à une importante variation locale d'intensité
 - on regarde la variation d'**intensité** autour d'un contour
 - la **dérivée première** passe par un maximum



Approches "contours"

- contour = ensemble des pixels séparant deux régions
- correspond à une importante variation locale d'intensité
 - on regarde la variation d'**intensité** autour d'un contour
 - la **dérivée première** passe par un maximum
 - la **dérivée seconde** passe par un zéro



Gradient de l'image

$$G = \overrightarrow{\text{grad}} f(x, y) \quad \begin{cases} G_x = \frac{\partial f}{\partial x}(x, y) \\ G_y = \frac{\partial f}{\partial y}(x, y) \end{cases}$$



G_x



G_y



$$\sqrt{G_x^2 + G_y^2}$$

Seuillage de la norme du gradient de l'image lissée

$$\sigma = 1$$



seuil = 5

seuil = 10

seuil = 15

Seuillage de la norme du gradient de l'image lissée

$$\sigma = 2$$



seuil = 5



seuil = 10



seuil = 15

Seuillage de la norme du gradient de l'image lissée

$$\sigma = 3$$



seuil = 5



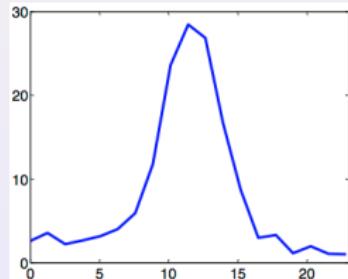
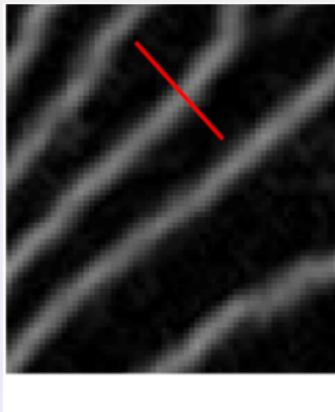
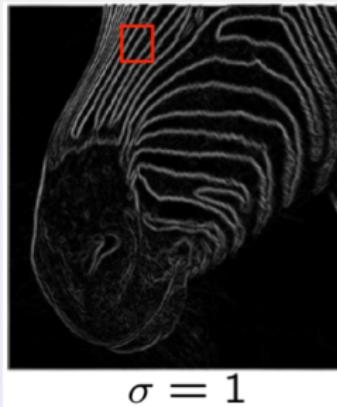
seuil = 10



seuil = 15

Détecteur de Canny

- Profil dans la direction du gradient



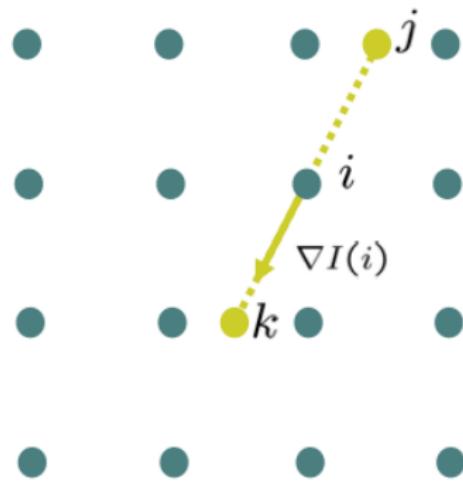
- Principe : ne garder que les maxima locaux (dans la direction du gradient) et supérieurs à un seuil

Détecteur de Canny

- Principe : ne garder que les maxima locaux (dans la direction du gradient) et supérieurs à un seuil

contour en i si

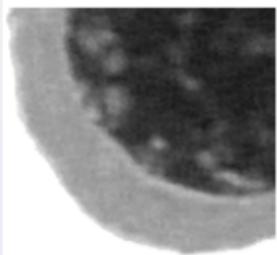
- $|\nabla I(i)| > |\nabla I(j)|$
- $|\nabla I(i)| > |\nabla I(k)|$
- $|\nabla I(i)| >$ seuil



Détecteur de Canny : Seuillage par hystérésis

- compromis dans le choix du seuil sur la norme du gradient pour sélectionner les contours
 - fermeture des contours (peu de faux négatif)
 - immunité au bruit (peu de faux positifs)
- Choix de deux seuils, un haut (spécificité) et un bas (sensibilité)
 - ① On sélectionne les points au-dessus du seuil haut
 - ② pour tout point entre le seuil haut et le seuil bas, on regarde s'il existe un point de son voisinage $>$ seuil haut
- Principe utilisé dans le détecteur de Canny.

Détecteur de Canny : Seuillage par hystérésis



Seuillage simple
 $S=50$

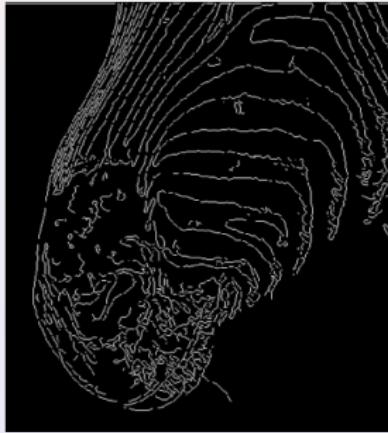


Seuillage simple
 $S=100$



Seuillage par
hystérésis
 $Sh=100$, $Sb=50$

Détecteur de Canny, résultats



Détecteur de Canny, résultats



Détecteur de Canny, résultats



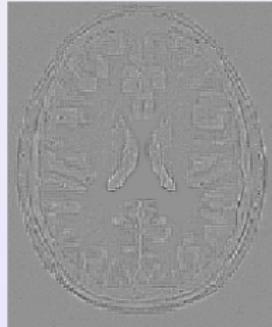
Détecteur de Canny, résultats



Laplacien de l'image

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

"équivalent de la dérivée seconde" en 2D



Détection des passages par 0
du laplacien (Marr et Hildreth,
1976).

Passages par zéros du laplacien

- Lignes de niveau zéro (courbes fermées) du laplacien lissé (LoG) et interpolé spatialement



- Mais, soit trop de lignes (lissage faible), soit peu de lignes mais en partie décalées (lissage fort)

Passages par zéros du laplacien

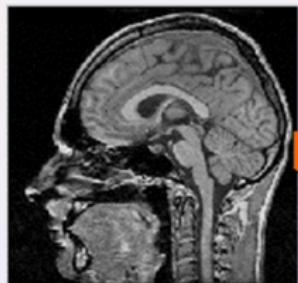


Image originale

L.o.G.



« Zero crossing »

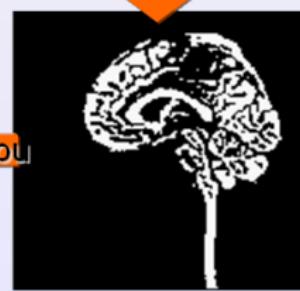


Composantes connexes



Image finale

Masque flou



Extraction de chaînes de contours

- Problème : extraction d'une chaîne de contours entre deux points donnés
- Applications :
 - analyse automatique de certains types d'images (extraction de routes dans images satellite, d'artères en angiographie, etc.)
 - découpe interactive en édition d'image
- Approche classique
 - définir une fonction de coût pour une chaîne candidate
 - trouver la chaîne de coût minimum entre les deux extrémités par programmation dynamique : problème combinatoire de plus court chemin dans un graphe résolu par l'algorithme de Dijkstra

Coût d'une chaînes de pixels

- Chaîne : succession de pixels voisins au sens de la 4 ou 8 connexité



- Minimiser une somme de coûts élémentaires positifs

$$E_n(x_0, \dots, x_n; I) = \sum_{i=1}^n \phi(x_{i-1}, x_i; I)$$

- Exemple avec f décroissante et g croissante

$$\phi(x_{i-1}, x_i; I) = \alpha \|x_i - x_{i-1}\| + \beta f(\|\nabla I(x_i)\|) + \gamma g(|\nabla I(x_i)| \cdot (x_i - x_{i-1}))$$

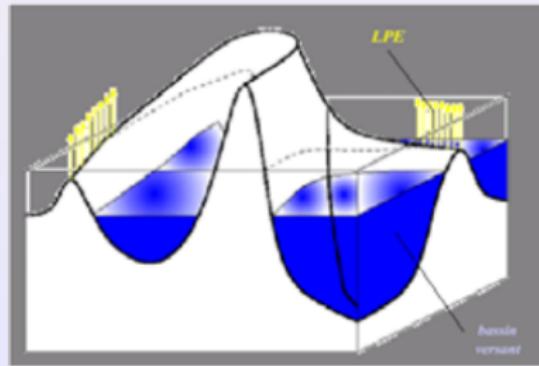
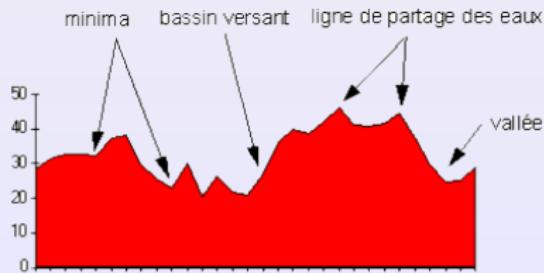
Lignes de partage des eaux (watershed)

- Digabel et Lantuéjou, 1972
- l'image est représenté par un relief que l'on inonde progressivement
- pour la segmentation, on travaille sur l'image de la norme du gradient

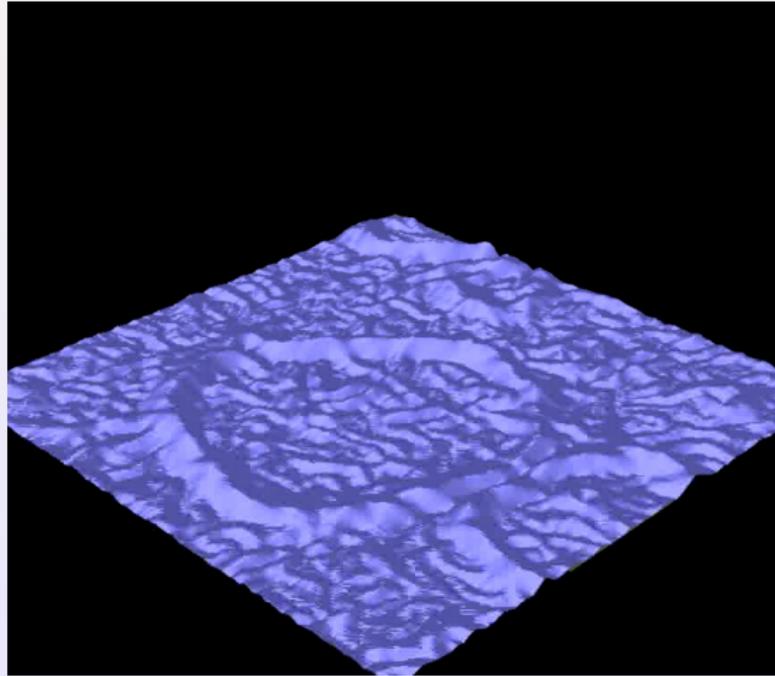


Lignes de partage des eaux (watershed)

- Points de départs : minima locaux
- montée des eaux
- barrage élémentaire à chaque rencontre de bassins : ligne de partage des eaux

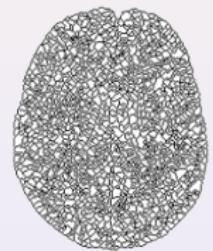
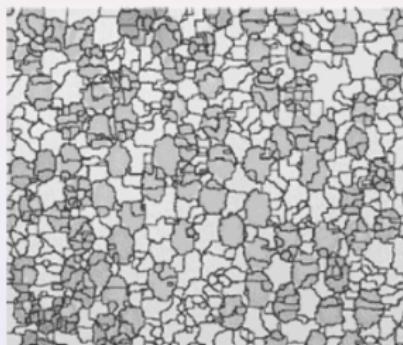
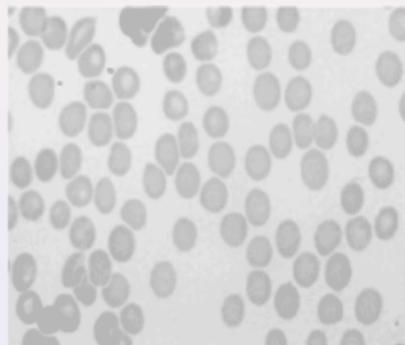


Lignes de partage des eaux (watershed)



<http://pmc.polytechnique.fr/~vta/water.mpeg>

Lignes de partage des eaux (watershed)



- algorithme non local
- problème de sur-segmentation
- sensibilité à tout minimum local, au bruit (\rightarrow lissage préventif)

Contours actifs

- Idée générale : on cherche la forme (courbe/surface) \mathcal{C} qui minimise une énergie

$$E(\mathcal{C}) = E_{\text{image}}(\mathcal{C}) + E_{\text{interne}}(\mathcal{C})$$

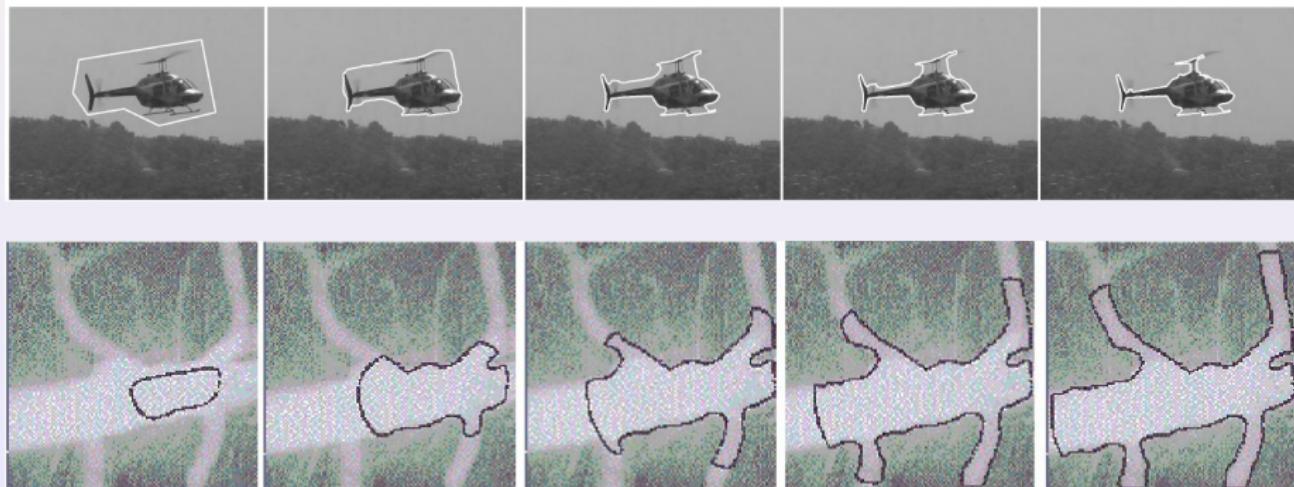
- $E_{\text{image}}(\mathcal{C})$ va essayer d'attirer la courbe sur les contours de l'image
- $E_{\text{interne}}(\mathcal{C})$ permet d'ajouter des a priori sur la forme de la courbe que l'on cherche (régularité)
- Contours actifs géodésiques :

$$E(\mathcal{C}) = \int_0^1 |\mathcal{C}'(p)|^2 dp + \beta \int_0^1 g_{\mathcal{I}}(\mathcal{C}(p)) dp$$

- la fonction de potentiel $g_{\mathcal{I}}$ vise à attirer la courbe vers les contours de l'objet à segmenter. Elle dépend donc de **l'image \mathcal{I} à segmenter**.
- Minimisation par "descente de gradient" : Euler-Lagrange

Approches "contours"

Contours actifs / Level-set



Représentation par courbes de niveaux : méthode "Level Set"

- Réprésentation/caractérisation d'une courbe
 - paramétrisation
 - liste de points
 - implicite \leftarrow Level Set

Level Set : Osher-Sethian, 1988.

Idée principale :

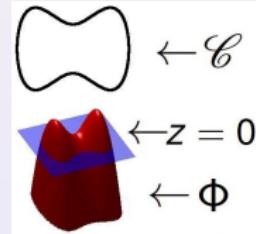
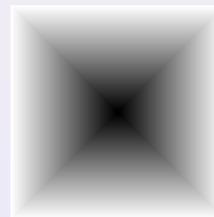
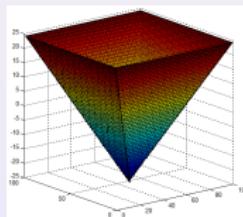
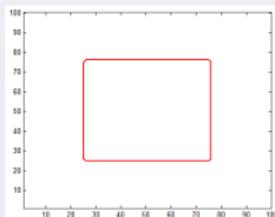
- la courbe \mathcal{C} est représentée de manière **implicite**.
- \mathcal{C} est représenté par une surface de dimension supérieure Φ dont elle est le **niveau zéro**

$$\mathcal{C} = \{ (x, y) \mid \Phi(x, y) = 0 \}$$

- Pour une courbe (dans \mathbb{R}^2) $\rightarrow \phi$ est une image plane
- Pour une surface (dans \mathbb{R}^3) $\rightarrow \phi$ est une image volumique

"Level Set"

- une courbe plane (dans \mathbb{R}^2) : représentée par une fonction $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ (image, surface)



- un bon choix : Φ fonction distance signée à la courbe C (négative à l'intérieur, positive à l'extérieur).

Exemple : mouvement par courbure moyenne

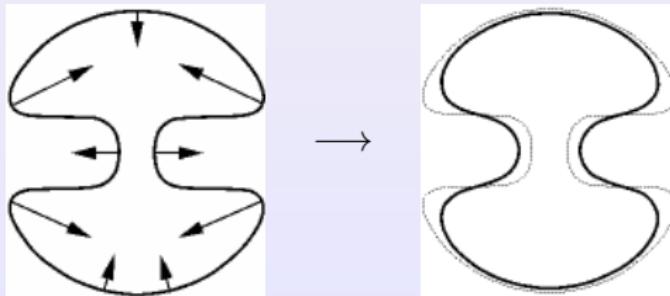
équation d'évolution sur \mathcal{C}

$$\begin{cases} \frac{\partial \mathcal{C}}{\partial t}(t, p) = \kappa(p) \mathbf{n}(p), \\ \mathcal{C}(0, p) = \mathcal{C}_0(p) \end{cases}$$

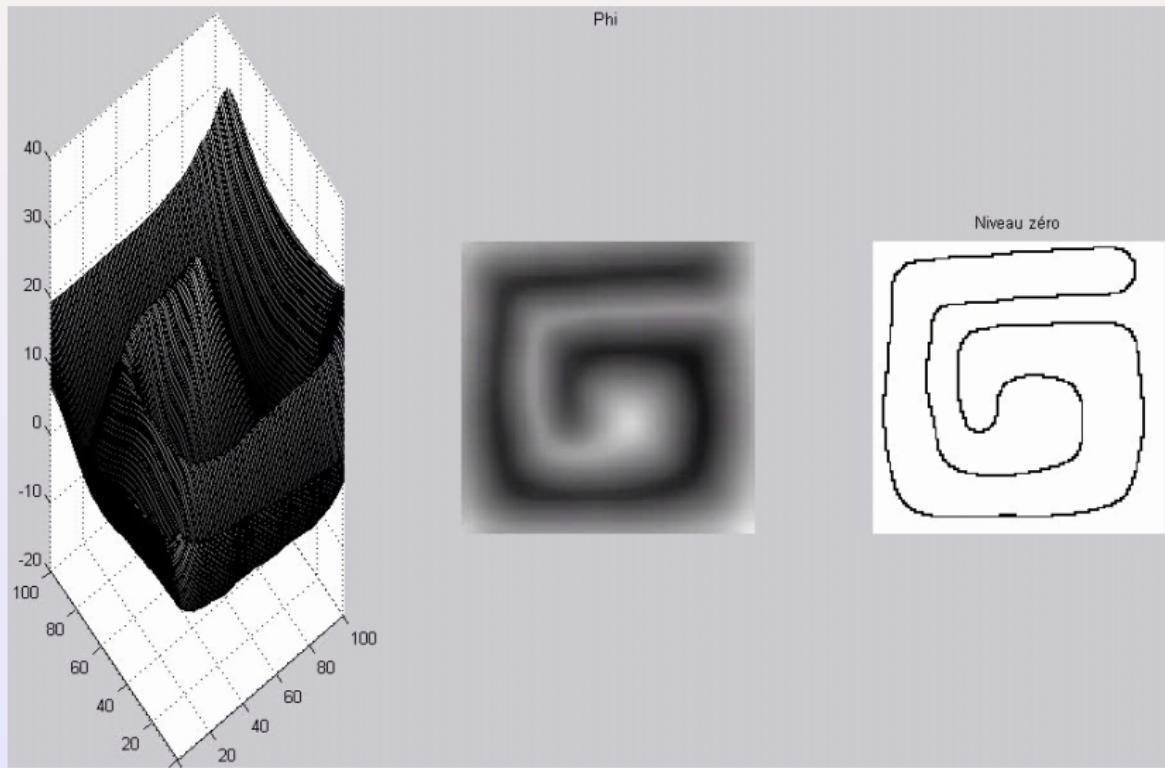
\Updownarrow

équation d'évolution sur Φ

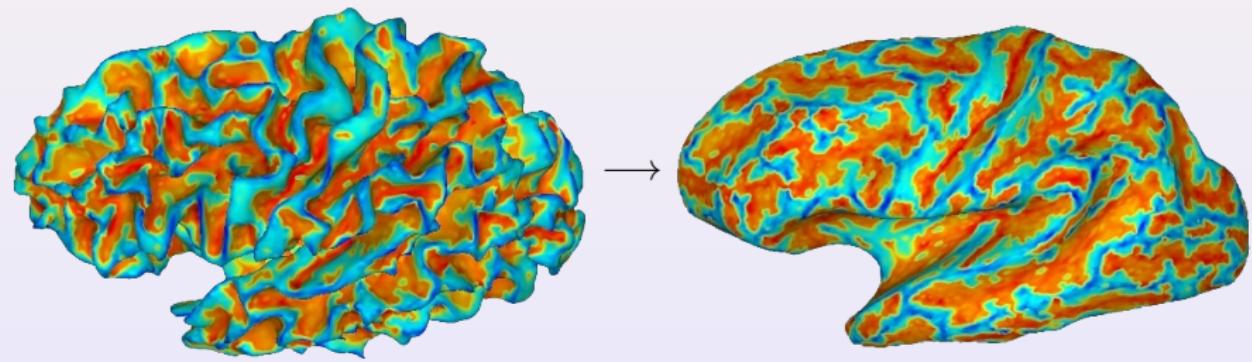
$$\begin{cases} \frac{\partial \Phi}{\partial t} = \operatorname{div}\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right) |\nabla \Phi| \\ \Phi(0, x, y) = \Phi_0(x, y) \end{cases}$$



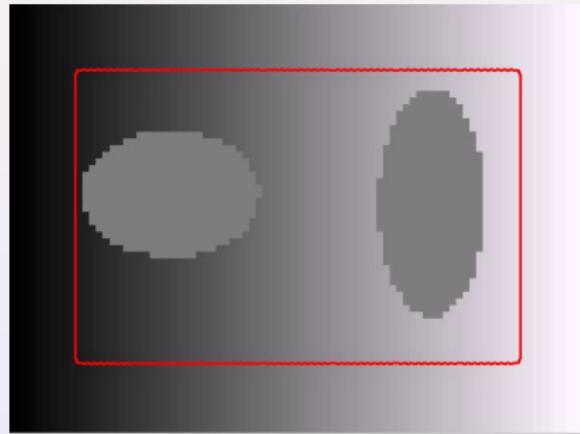
Exemple : mouvement par courbure moyenne



Exemple : mouvement par courbure moyenne

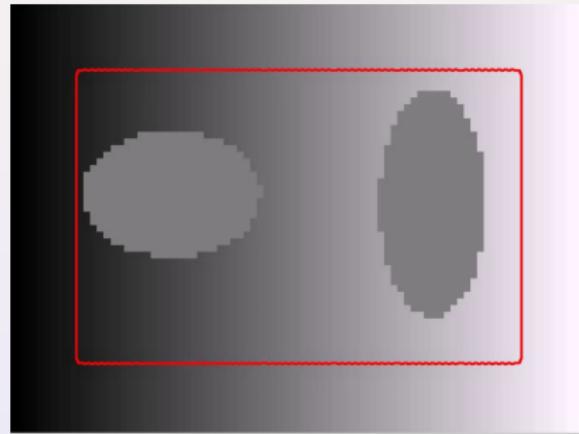


Segmentation par contours actifs : exemples

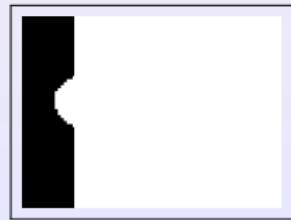


Seuillage ?

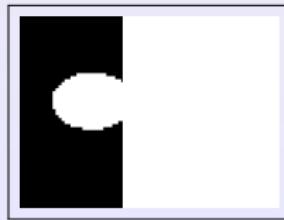
Segmentation par contours actifs : exemples



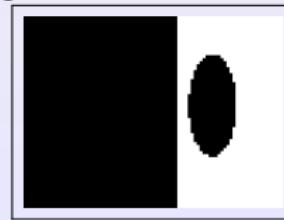
Seuillage ?



$s = 50$



$s = 100$

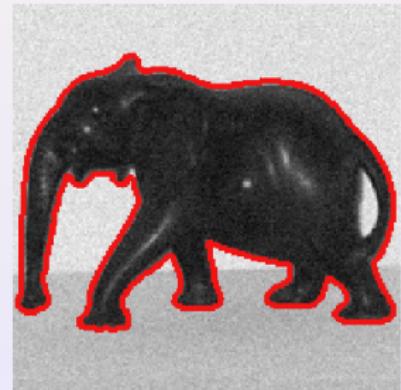
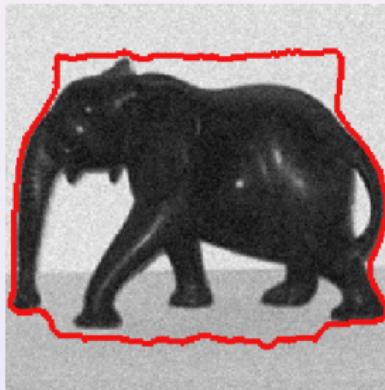
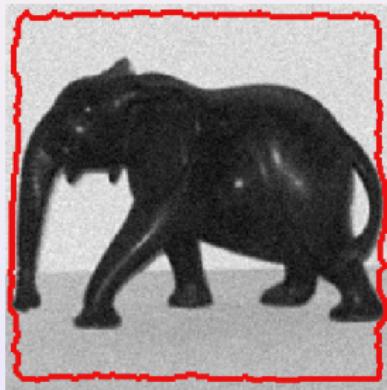


$s = 150$

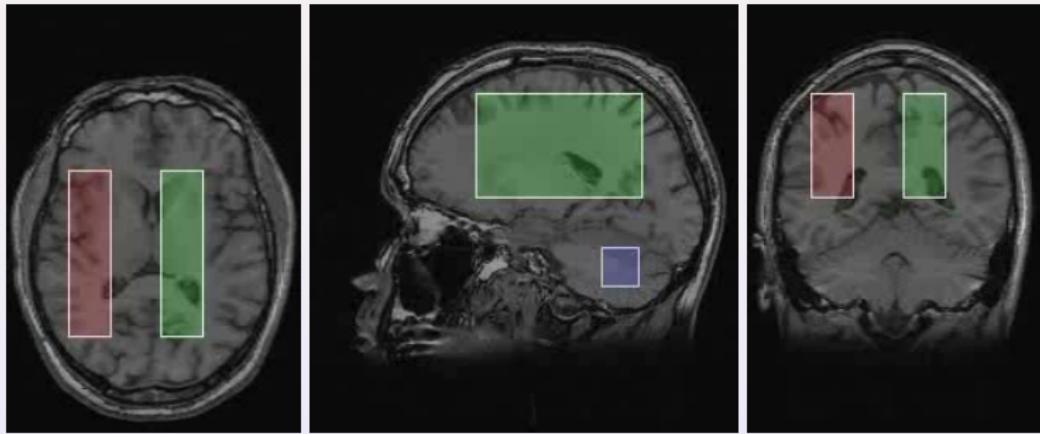


$s = 200$

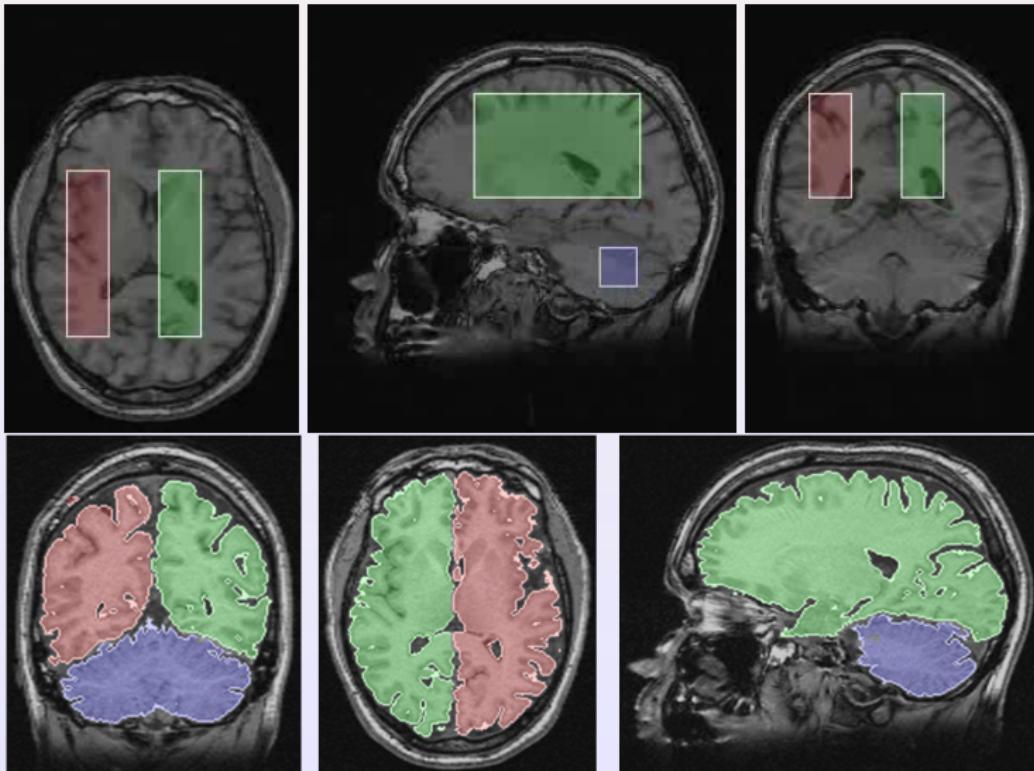
Segmentation par contours actifs : exemples



Segmentation par contours actifs : exemples



Segmentation par contours actifs : exemples



Représentation implicite : récapitulatif

Représentation Implicite

On représente une courbe \mathcal{C} , de manière implicite, par une surface de dimension supérieure Φ dont elle est le **niveau zéro**

$$\mathcal{C} = \left\{ (x, y) \mid \Phi(x, y) = 0 \right\}$$

- une courbe est représentée par une image (2D) $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}$
- une surface est représentée par une image volumique (3D) $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}$

EDP sur $\mathcal{C} \leftrightarrow$ EDP sur Φ

$$\boxed{\begin{cases} \frac{\partial \mathcal{C}}{\partial t}(t, p) = v(p) \mathbf{n}(p), \\ \mathcal{C}(0, p) = \mathcal{C}_0(p) \end{cases}} \Leftrightarrow \boxed{\begin{cases} \frac{\partial \Phi}{\partial t} = v |\nabla \Phi| \\ \Phi(0, x, y) = \Phi_0(x, y) \end{cases}}$$

Représentation implicite : récapitulatif

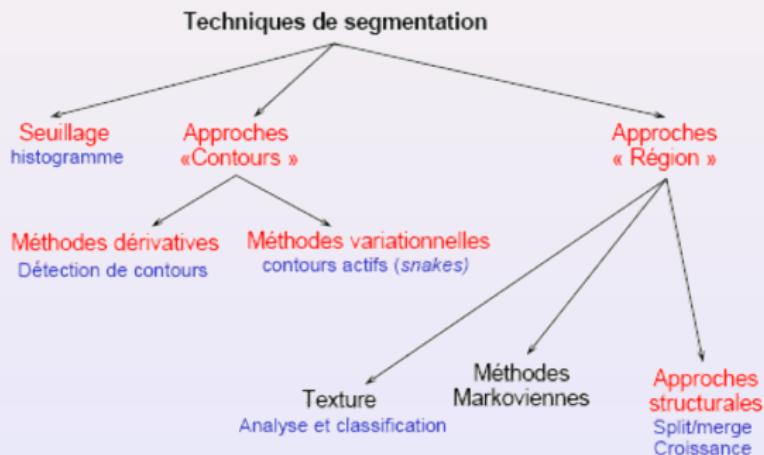
Avantages :

- changements de topologie "gratuits"
- discréétisation classique (grille discrète fixée → schéma aux différences finies)
- des éléments géométriques **intrinsèques** à la courbe \mathcal{C} sont facilement exprimable à partir de Φ
- même formulation (par Level Set) pour n'importe quelle **dimension** (courbe, surface) : implémentation aussi simple

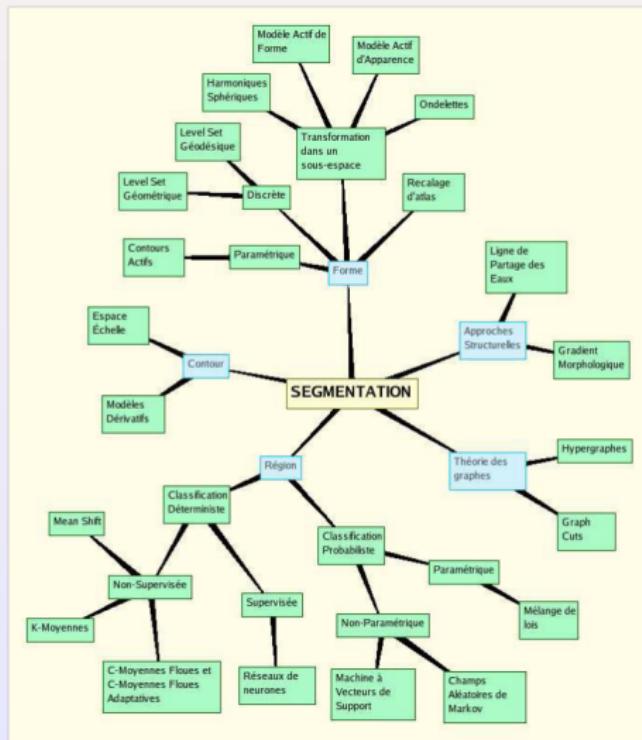
Inconvénients :

- temps de calculs
- impossible de représenter des courbes, ou surfaces, qui s'auto-intersectent
- difficile de représenter des courbes, ou surfaces, ouvertes

Récapitulatif



Récapitulatif (bis)



Imagerie médicale : segmentation basées sur un atlas

Atlas anatomique

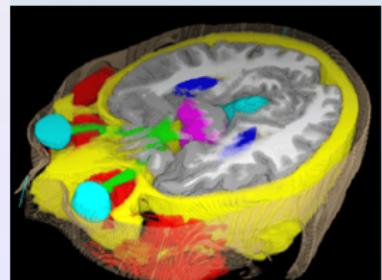
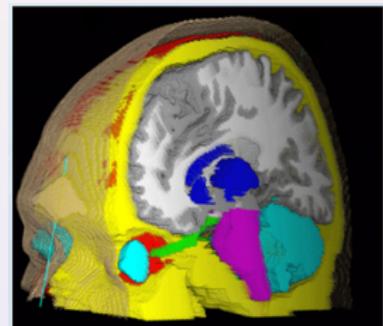
- image d'une anatomie moyenne
- segmentation associée effectuée par un expert

Objectif

- Utilisation de l'atlas pour segmenter une nouvelle image (patient)

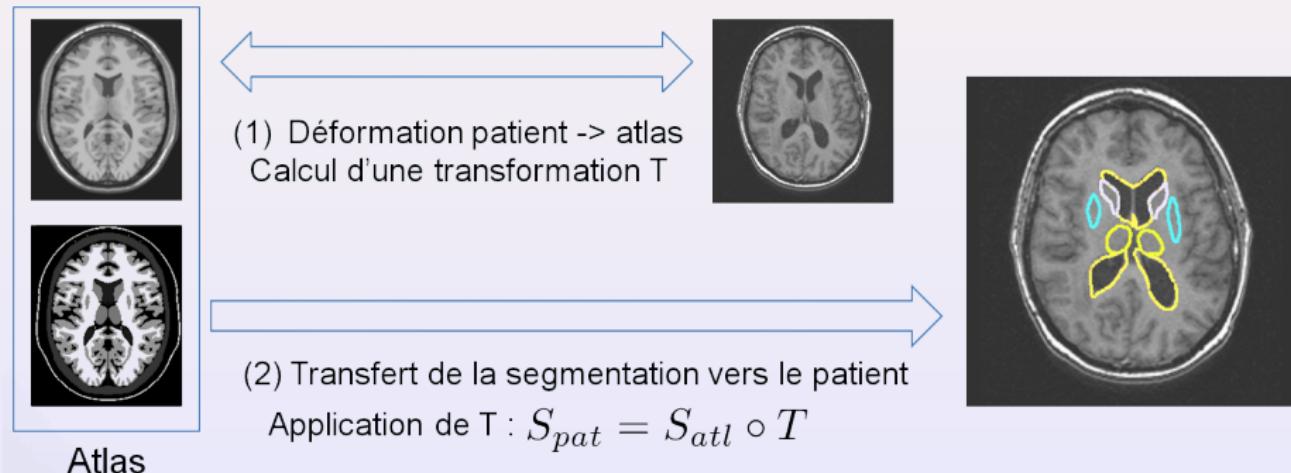
Avantage

- prise en compte d'a priori spécifiques au résultat recherché
- Segmentation de multiples structures en une fois
- possibilité d'estimer des structures peu visible



Segmentation par Atlas : Méthode

- On transforme le problème de segmentation en un problème de **recalage**



On a donc besoin d'être capable de recaler une image sur une autre.