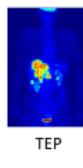


Imagerie Médicale

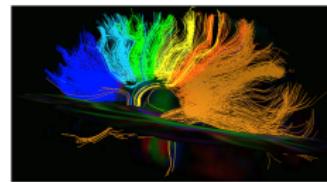
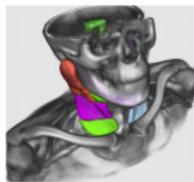
Partie 3: Segmentation

Pierre Maurel

pierre.maurel@irisa.fr



MEG



Analyse d'images, vision par ordinateur

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



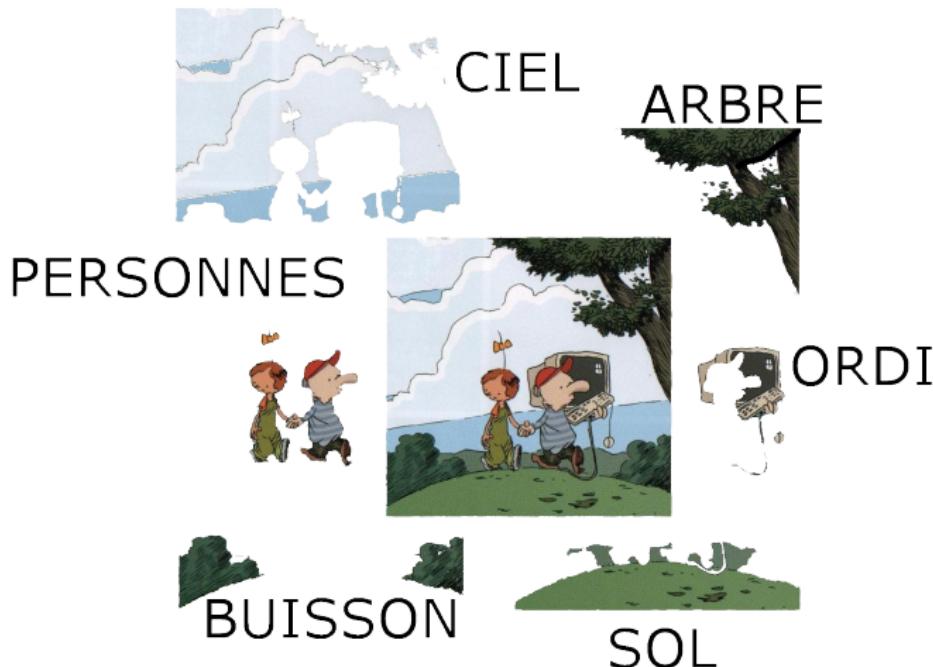
Analyse d'images, vision par ordinateur

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



Analyse d'images, vision par ordinateur

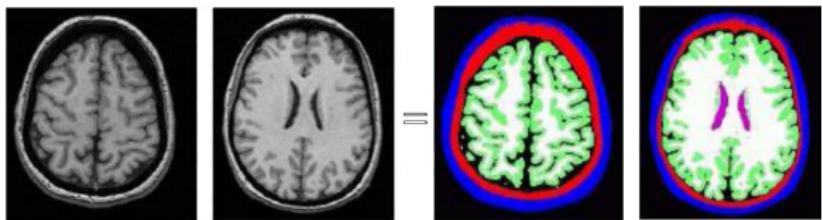
- Segmentation : partitionner l'image en ses différentes parties.
- Reconnaissance : étiqueter les différentes parties



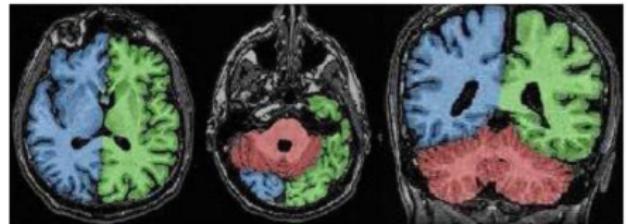
Segmentation ?

- 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

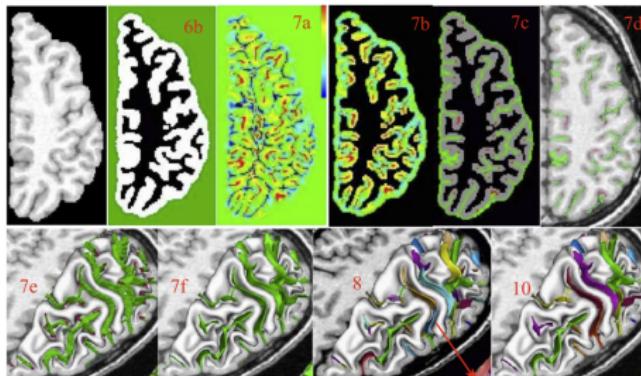
Segmentation ?

À quoi ça sert ?

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

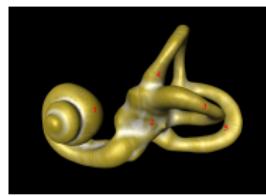
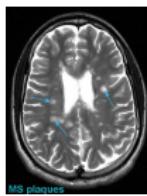
- reconnaissance d'objets
- indexation : rechercher dans une base d'images, les images "ressemblantes" à une image initiale
- compression
- recalage d'images, mises en correspondance
- ...

Segmentation ?



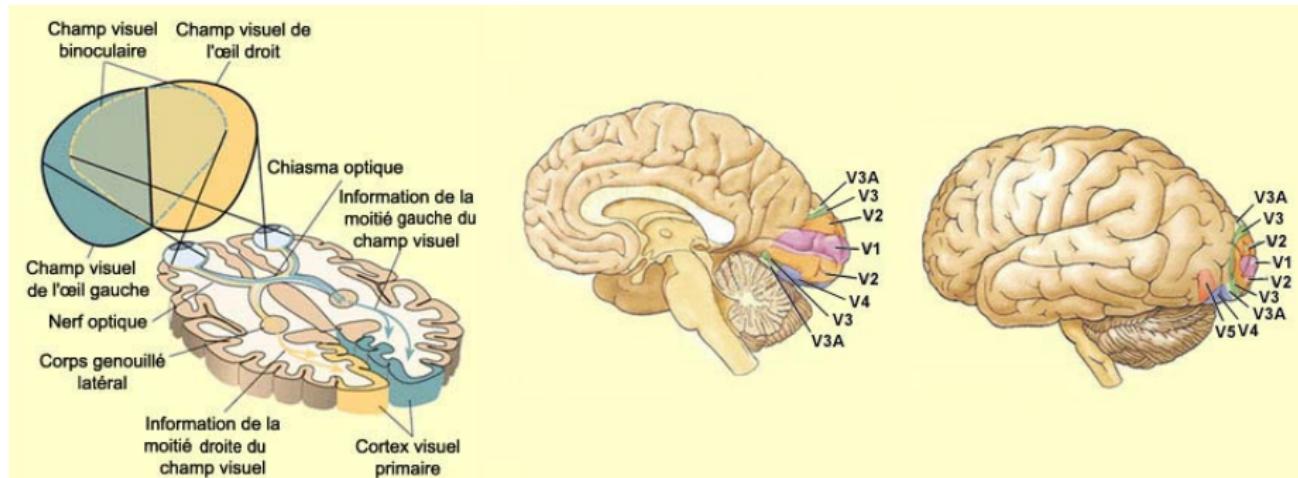
A framework to study the cortical folding patterns, Mangin et al., NeuroImage 2004

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

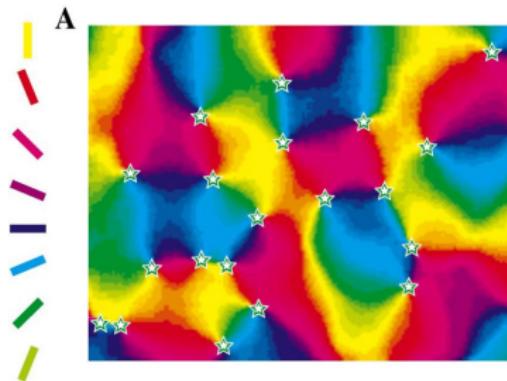
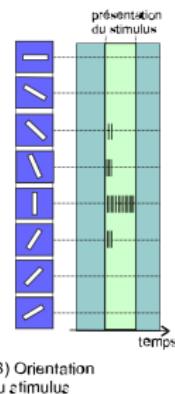
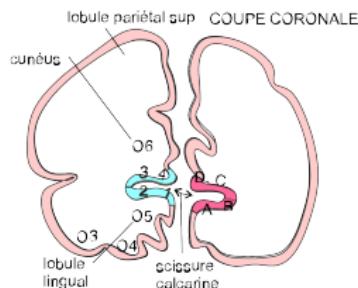


Segmentation ?

- 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 ?



Segmentation ?

- **Pas de solution universelle** : en général, algo limité à un type d'application et/ou d'image
 - Différentes approches :
 - Approches traditionnelles, orientées "modèles"
 - Approches orientées "données"
 - Approches basées atlas/recalage
- 
- approches globales
 - approches régions
 - approches contours

Approches orientées "modèles"

Approches orientées "données"

Approches basée atlas/recalage

Approches Globales → seuillage, k-means, seuillage adaptatif

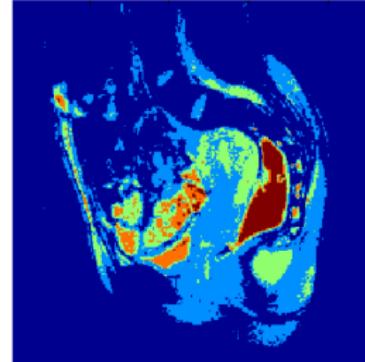
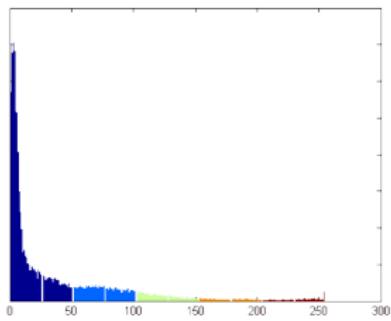
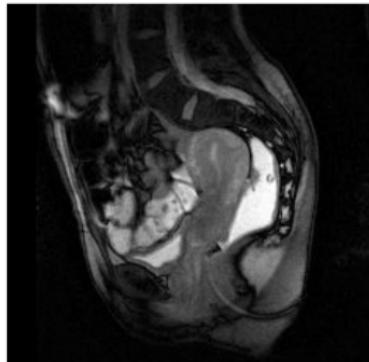
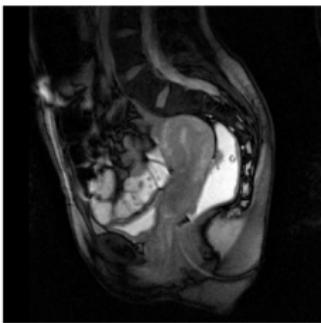
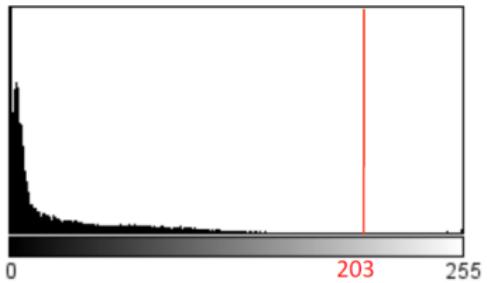
Approches "Régions" → croissance, markoviennes, graph-cut

Approches "contours" → gradient, Canny, Watershed, contours actifs

Approches Globales

Approches Globales

exemple le plus simple : seuillage d'histogramme



Seuillage d'histogramme

- Détermination du (ou des) seuil(s)

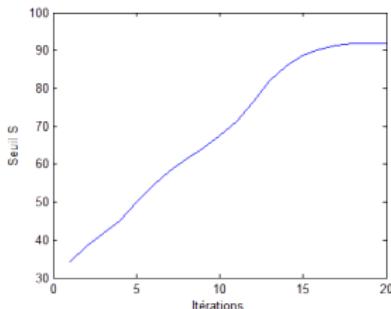
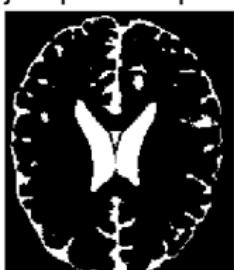
- valeur obtenue par test
- valeur moyenne
- valeur médiane
- choix automatique

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 → 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 i ème classe
- les σ_i^2 sont les variances de ces classes



<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>

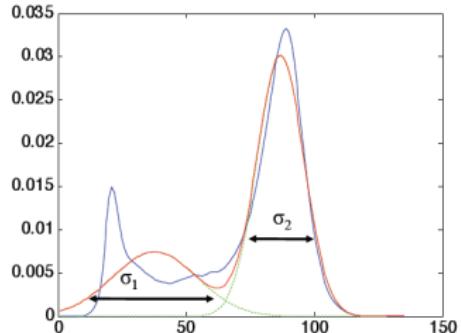
Matlab (image processing toolbox) : `graythresh(I)`

Seuillage d'histogramme

Seuillage par classification bayésienne

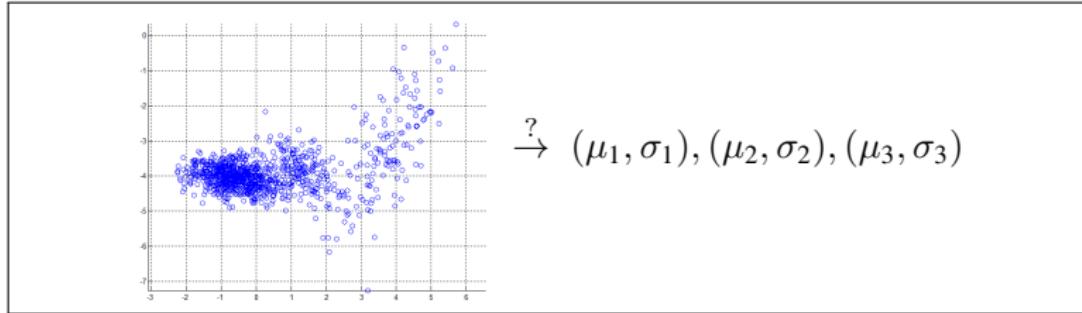
- Approximation de l'histogramme par un mélange de gaussiennes

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

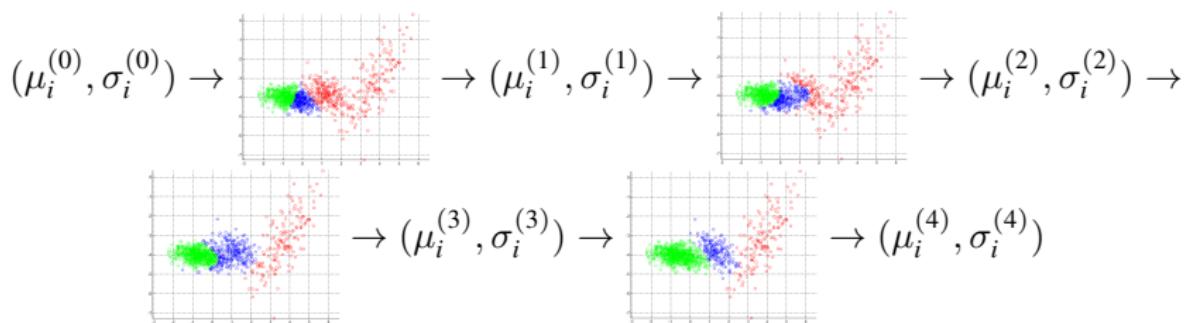


- Estimation de 5 paramètres libres (EM, gradient)

Algorithme EM, Modèle de mélanges gaussiens



DEMO MATLAB



Seuillage d'histogramme

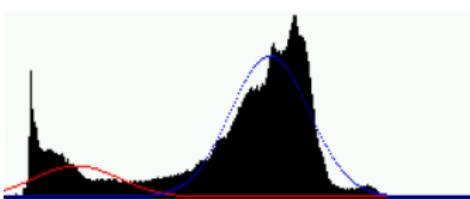
Seuillage par classification bayésienne



Image originale



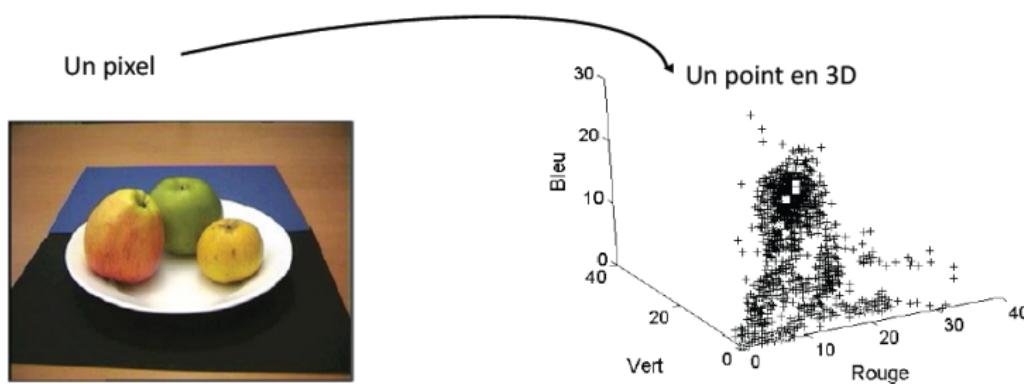
Image segmentée



un tuto : <http://drdik.me/expectation-maximisation.html>

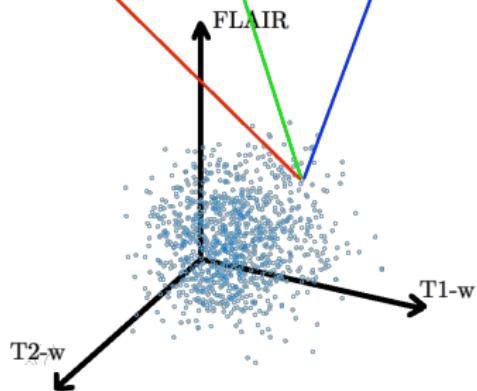
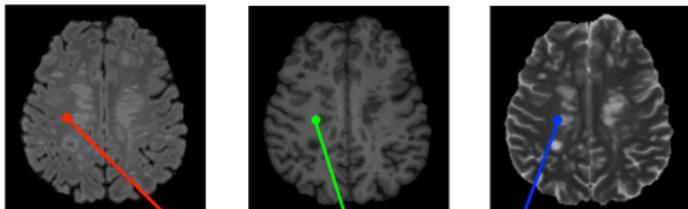
Algorithme des K-moyennes (K-means)

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



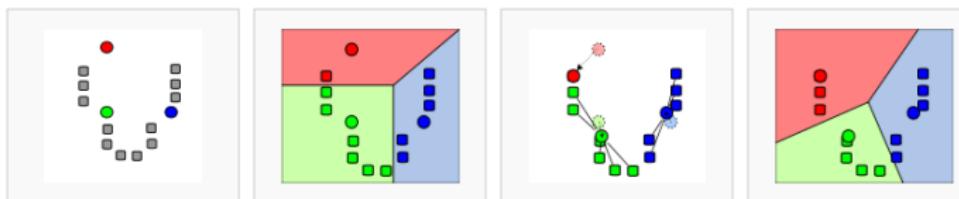
Algorithme des K-moyennes (K-means)

- 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



1) k initial "means" (in this case $K=3$) are randomly selected from the data set (shown in color).

2) k clusters are created by associating every observation with the nearest mean. The partitions here represent the [Voronoi diagram](#) generated by the means.

3) The [centroid](#) of each of the k clusters becomes the new means.

4) Steps 2 and 3 are repeated until convergence has been reached.

WIKIPEDIA

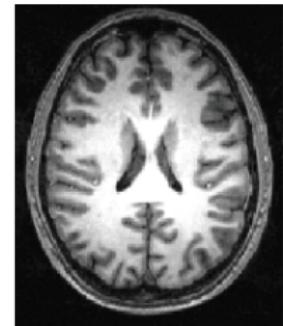
Algorithme des K-moyennes (K-means)

- Algorithme des K-moyennes en 1D

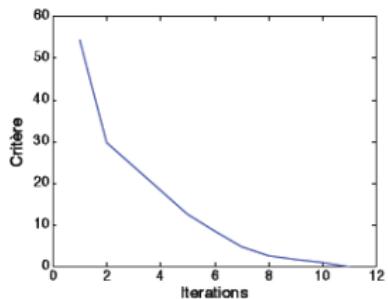
Initialisation (103,239,234)



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



Algorithme des K-moyennes (K-means)

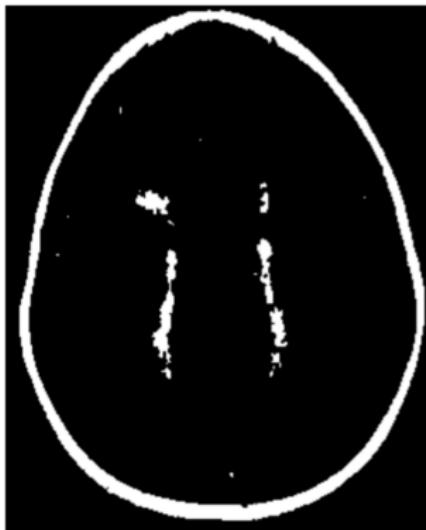


FIGURE: Classification sur 2 classes.

Algorithme des K-moyennes (K-means)

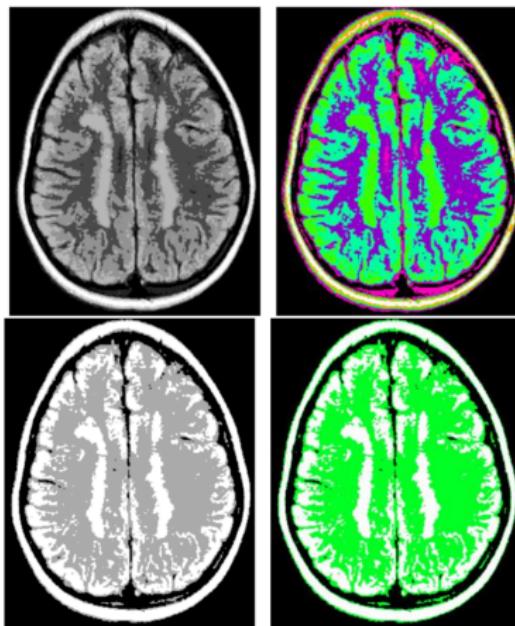


FIGURE: Classification sur 8 classes. sur 4 classes

Algorithme des K-moyennes (K-means)

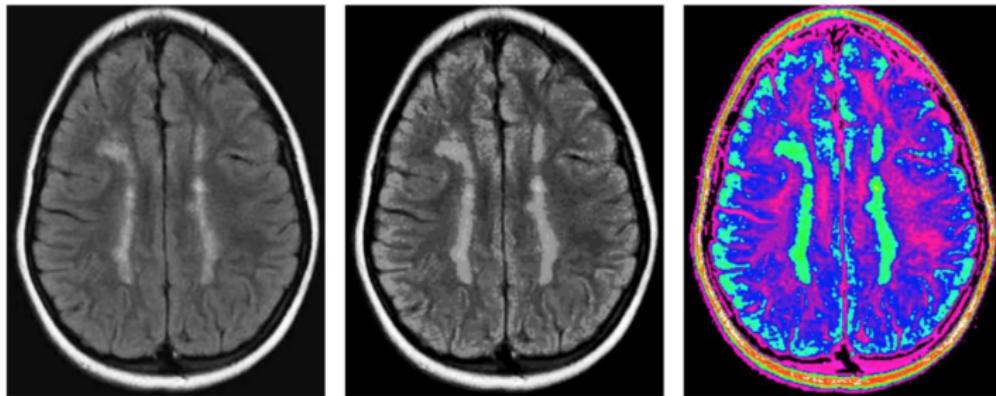


FIGURE: Image originale. Classification sur 16 classes

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

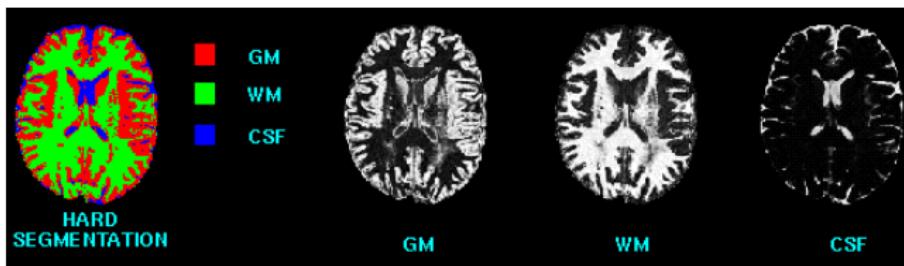
1 Répéter

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

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

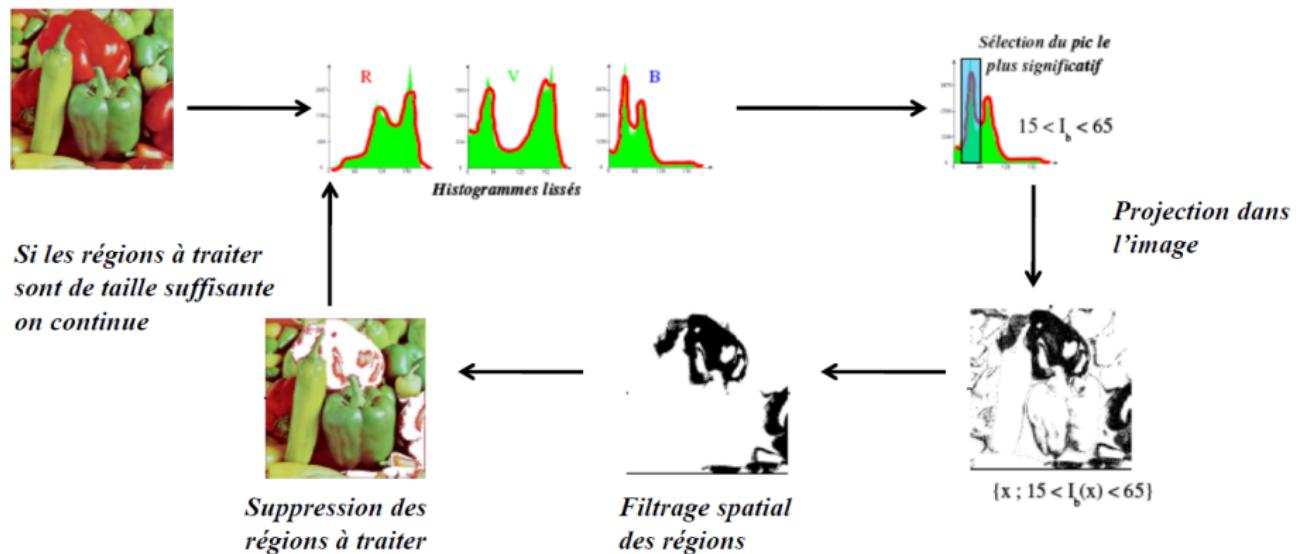
$$2 \quad \mu_k = \frac{\sum_s u_{sk}}{\sum_s}$$

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

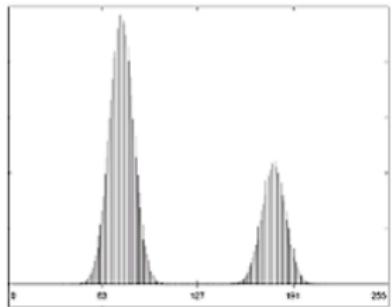
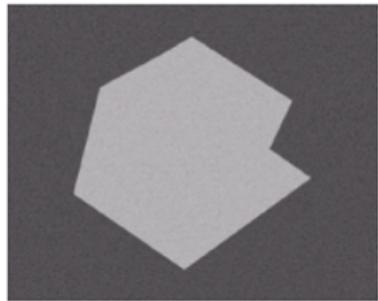


Sélection récursive d'histogramme

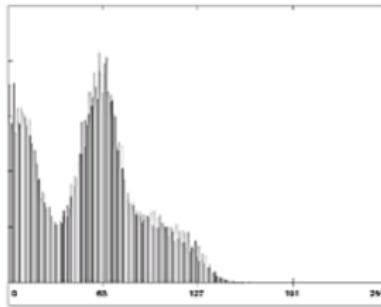
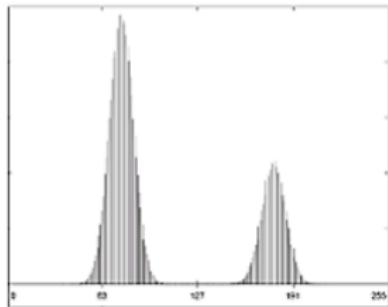
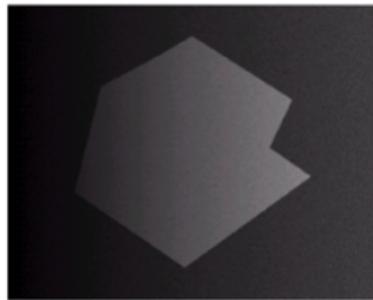
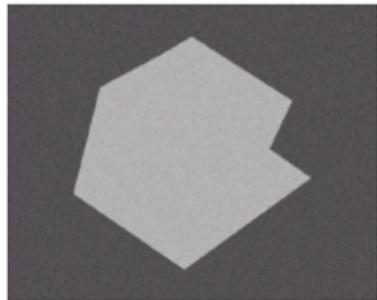
Ohlander, Price et Reddy (1978)



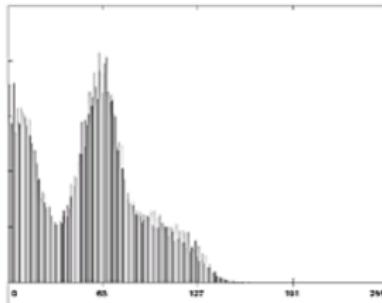
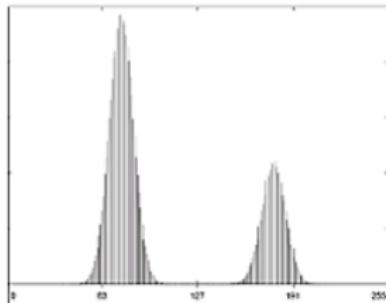
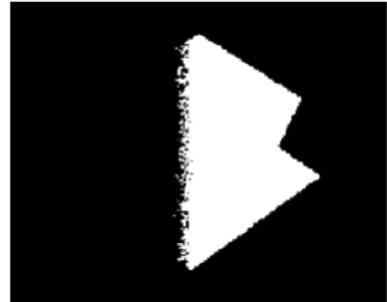
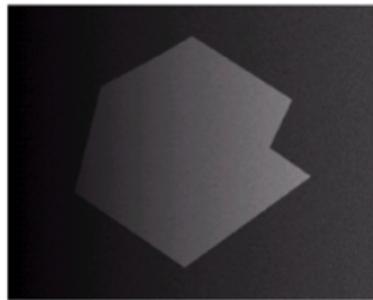
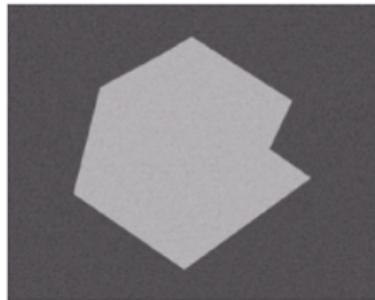
Limite des approches globales



Limite des approches globales



Limite des approches globales



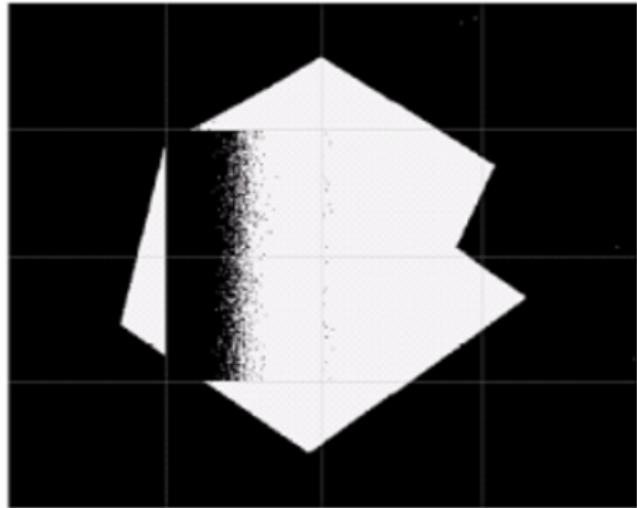
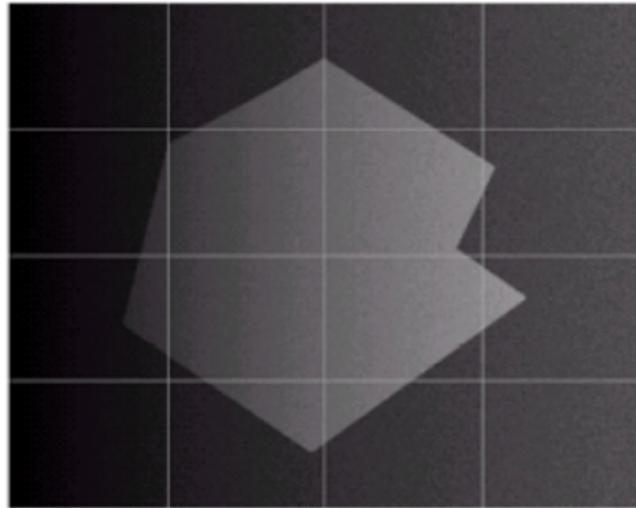
Seuillage par la
méthode d'Otsu

Seuillage adaptatif

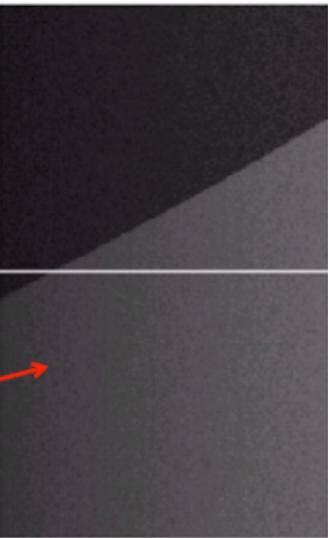
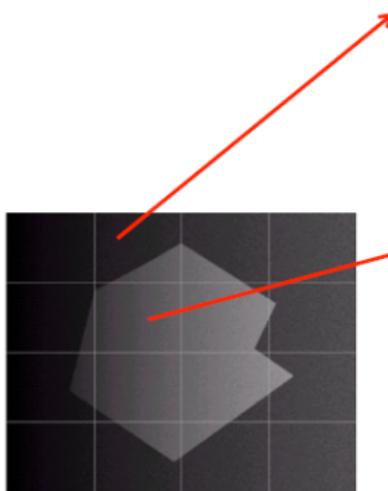
- On divise l'image en un certain nombre de sous-régions
- → seuillage sur chaque région

Seuillage adaptatif

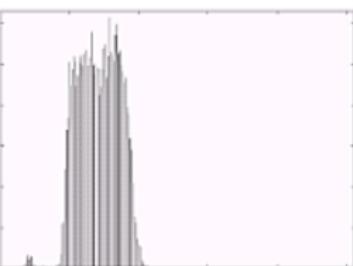
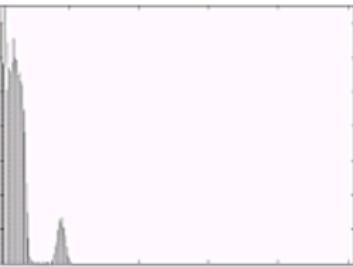
- On divise l'image en un certain nombre de sous-régions
- → seuillage sur chaque région



Seuillage adaptatif



Histogramme bimodal



Histogramme monomodal

Seuillage adaptatif

Matlab (image processing toolbox) : `imbinarize(I, 'adaptive')`¹

What Is Image Filtering in the Spatial Domain?

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel. (See Neighborhood or Block Processing: An Overview for a general discussion of neighborhood operations.) Linear filtering is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood.

Convolution

Linear filtering of an image is accomplished through an operation called convolution. Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is called the convolution kernel, also known as the filter. A convolution kernel is a correlation kernel that has been rotated 180 degrees.

For example, suppose the image is

$$A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 18 & 12 & 19 & 21 & 3 \end{bmatrix}$$

What Is Image Filtering in the Spatial Domain?

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

Filtering is a neighborhood operation, in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel. (See Neighborhood or Block Processing: An Overview for a general discussion of neighborhood operations.) Linear filtering is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the input pixel's neighborhood.

Convolution

Linear filtering of an image is accomplished through an operation called convolution. Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is called the convolution kernel, also known as the filter. A convolution kernel is a correlation kernel that has been rotated 180 degrees.

For example, suppose the image is

$$A = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 18 & 12 & 19 & 21 & 3 \end{bmatrix}$$

1. Bradley, D., G. Roth, "Adapting Thresholding Using the Integral Image," Journal of Graphics Tools, 2007

Approches globales : bilan

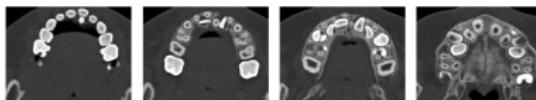
- **Avantages :**

- simple, rapide
- bien adapté aux histogrammes multimodaux

- **Inconvénients :**

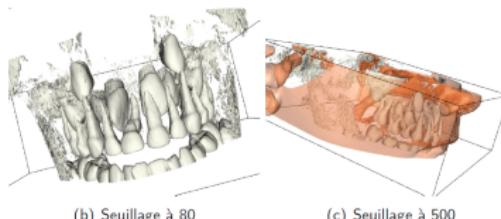
- il faut connaître le nombre de classes
- choisir les seuils
- **Pas d'information de connexité**

EN CT-X ...



(a) 4 plans de coupe

Parfois suffisant :



(b) Seuillage à 80

(c) Seuillage à 500

Approches orientées "modèles"

Approches orientées "données"

Approches basée atlas/recalage

Approches Globales → seuillage, k-means, seuillage adaptatif

Approches "Régions" → **croissance, markoviennes, graph-cut**

Approches "contours" → gradient, Canny, Watershed, contours actifs

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

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

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

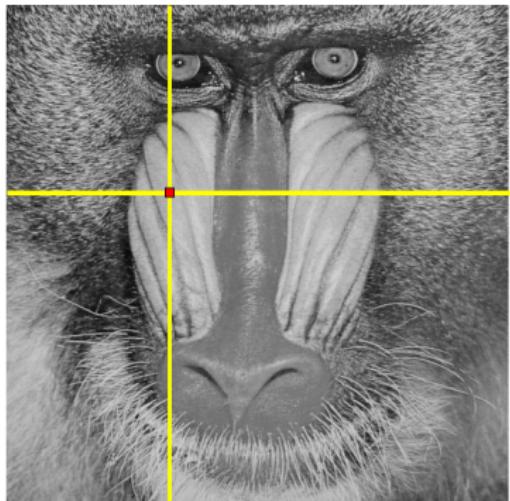
Croissance de régions - Region growing

- 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

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



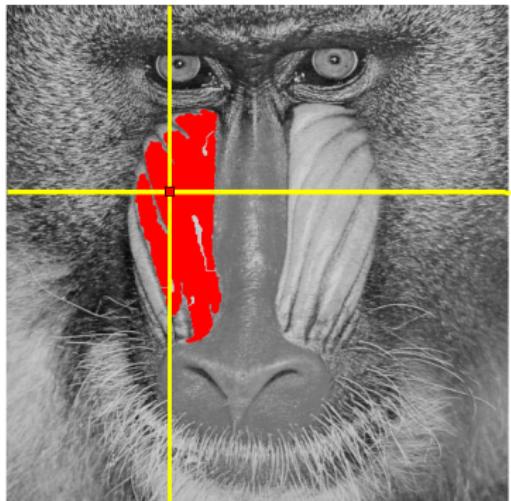
Croissance de régions - Region growing

- 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

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



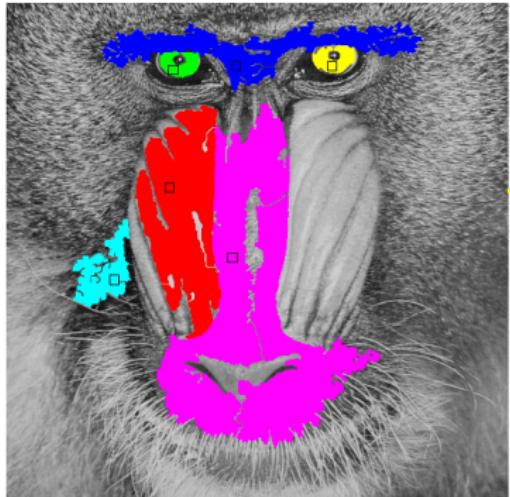
Croissance de régions - Region growing

- 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

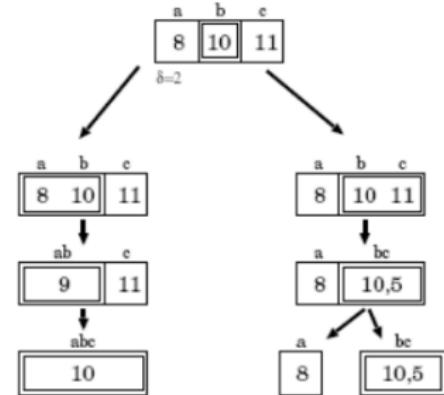
$$\left\{ \begin{array}{l} \min\{|I(x) - I(y)|; y \in R \cap V(x)\} < \text{seuil} \\ |I(x) - \mu_R| < \sigma_R \end{array} \right.$$



Croissance de régions - Region growing

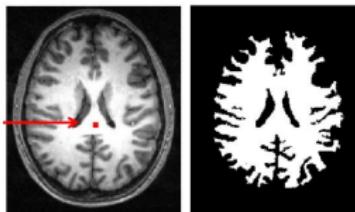
Limitations :

- Influence du choix des graines
- 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).

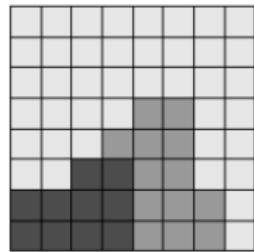


Split and Merge

- 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.

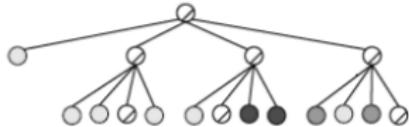
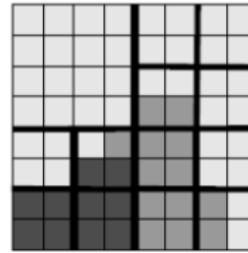
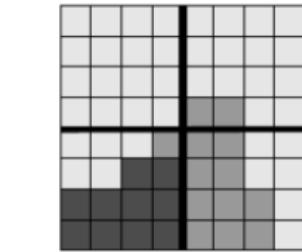
Split and Merge

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

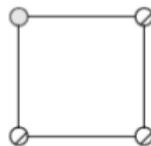


∅

Quadtree

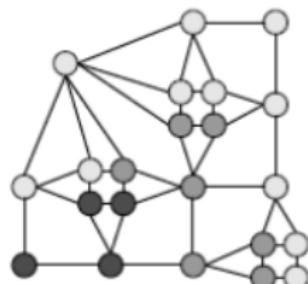
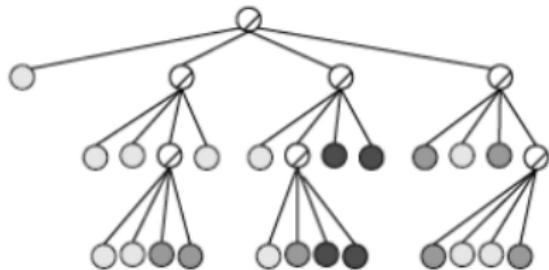
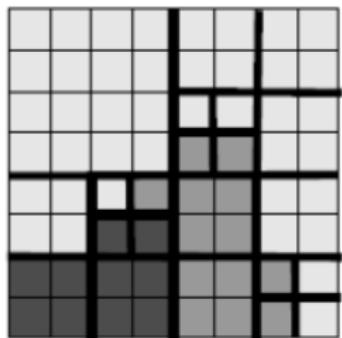


Graphe d'adjacence



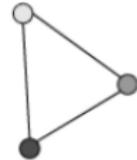
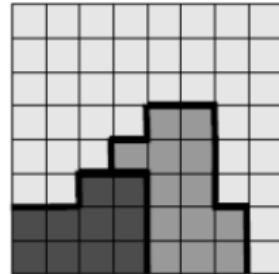
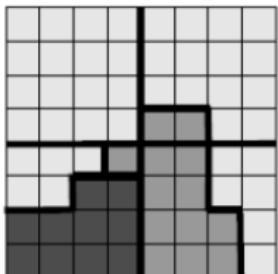
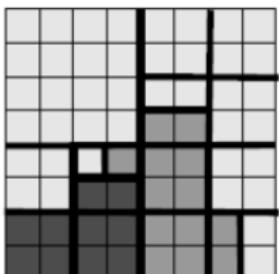
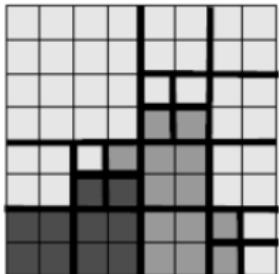
Split and Merge

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



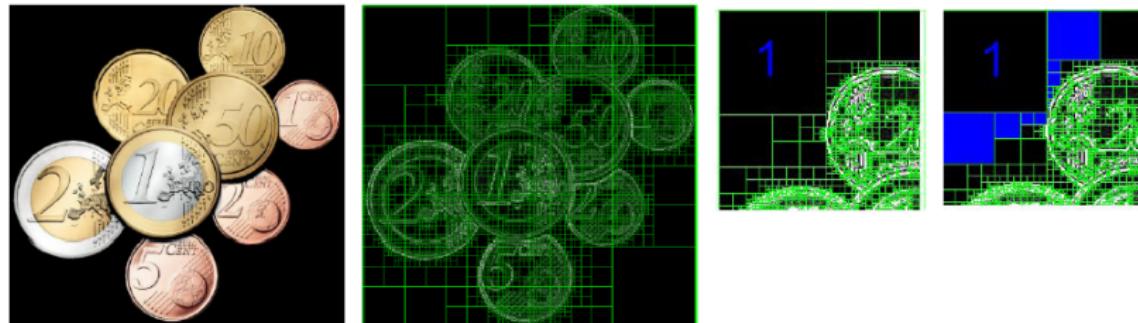
Split and Merge

- 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

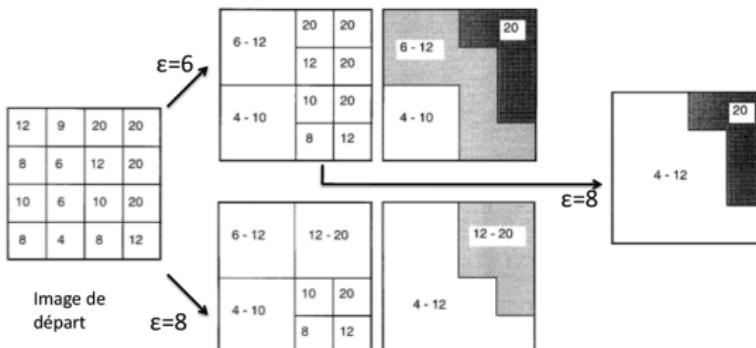


- critère d'homogénéité :
 - sur les extrema des régions $\max_R(I(x, y)) - \min_R(I(x, y)) < \epsilon$
 - sur la variance au sein de la région $\sum_R (I(x, y) - \mu_R)^2 < \epsilon$

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



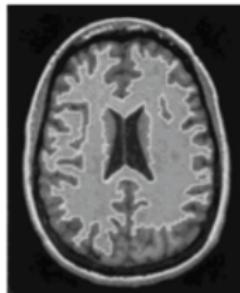
Split and Merge

Amélioration

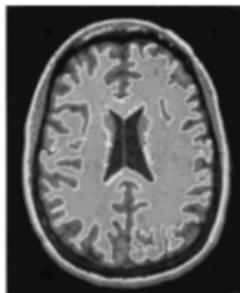
Image originelle



SM classique



SM avec recuit simulé



SM classique

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.

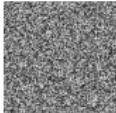
Méthodes markoviennes

- 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



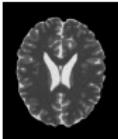
plus probable que



- 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

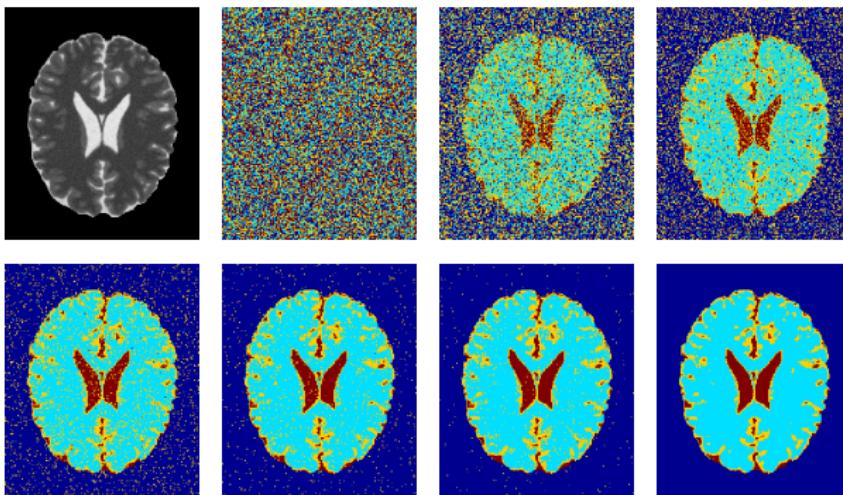
Méthodes markoviennes en segmentation

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**

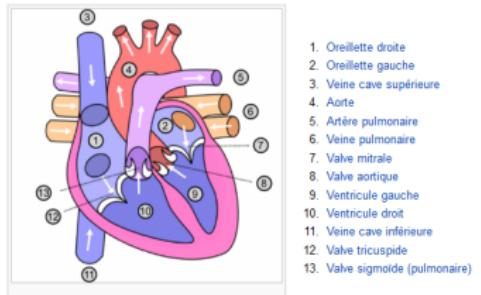
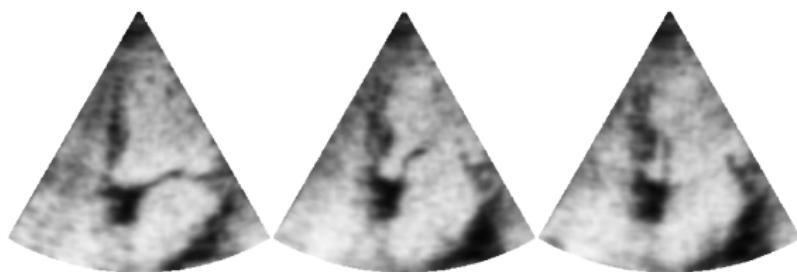
Exemple : segmentation texturale (recuit simulé)



■ matière blanche ■ matière grise ■ liquide céphalo-rachidien

Segmentation Markovienne en imagerie US cardiaque

- Segmentation du ventricule gauche du cœur
- 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

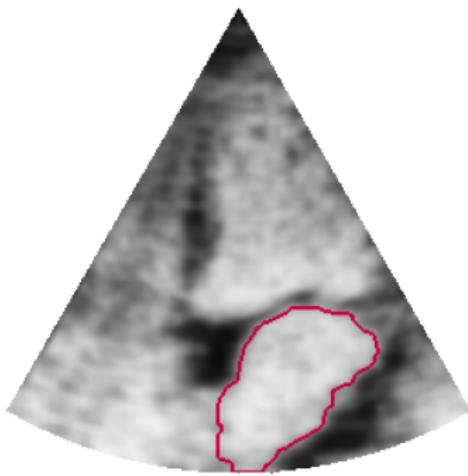
Modèle

- 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

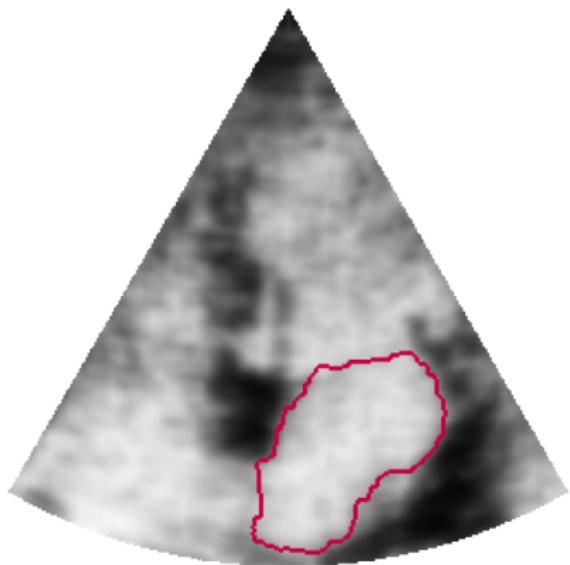
- α est choisi empiriquement, peut venir d'a priori sur l'image
- il faut initialiser
- il faut optimiser : recuit simulé / ICM



Segmentation Markovienne en imagerie US cardiaque

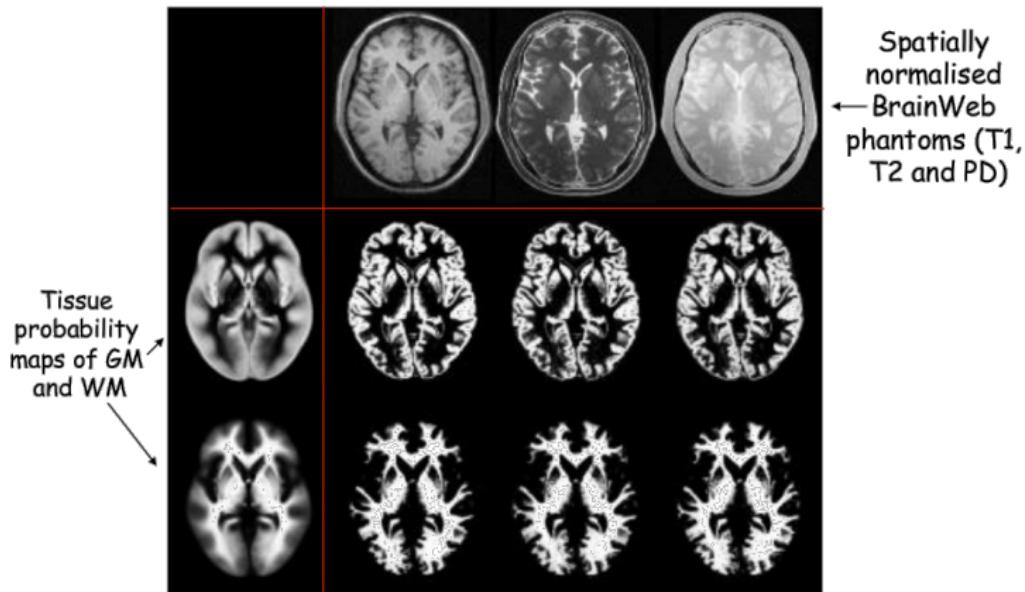
cohérence temporelle

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

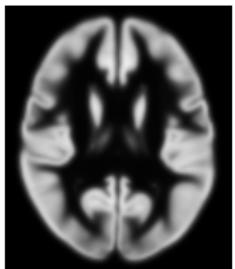


Segmentation Markovienne : imagerie cérébrale

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



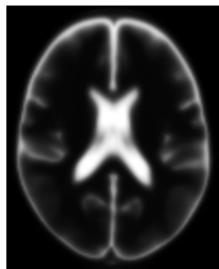
Imagerie cérébrale : cartes de proba, a priori



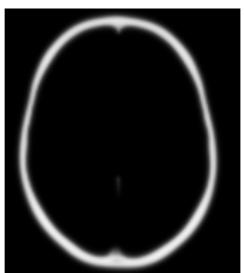
TPM,1 : GM



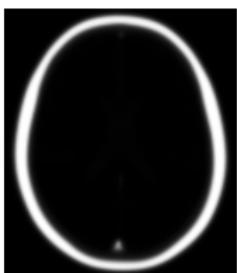
TPM,2 : WM



TPM,3 : CSF



TPM,4 : Bone



TPM,5 : Soft



TPM,6 : other

Méthodes markoviennes

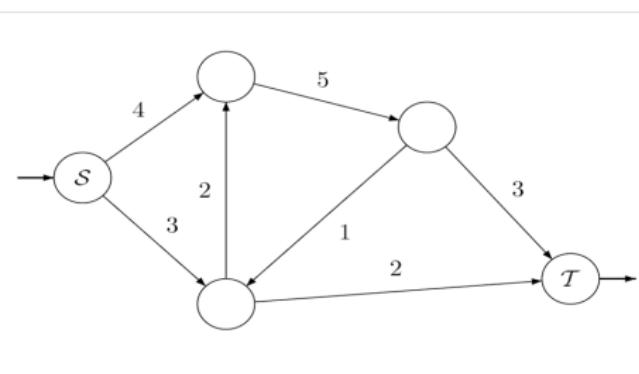
Bilan

- 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)
- 2005 : Regain d'intérêt 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

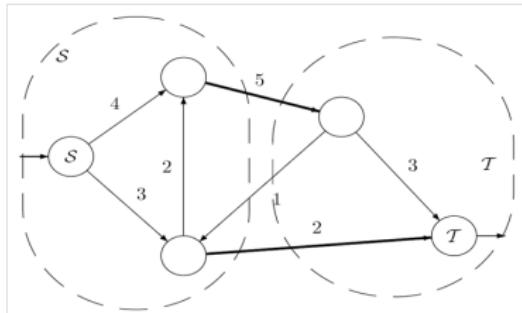
- $G = (V, E)$: Graphe Orienté
- V : sommets, E : arêtes
- pour chaque $(v_1, v_2) \in E$, une capacité w_{v_1, v_2} (poids des arêtes)
- S : source, T : puits



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

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

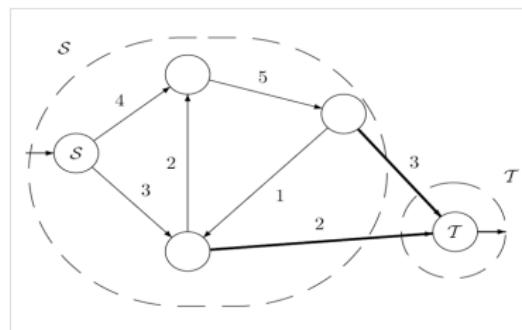
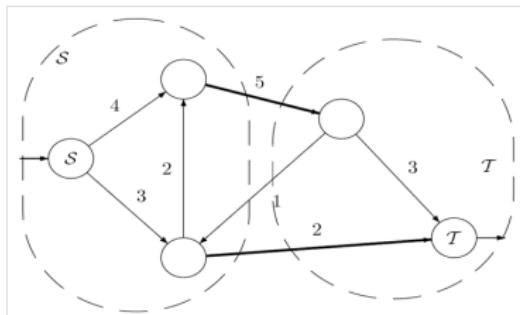
Coupe



- coupe = partition du graphe en deux régions 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"

Coupe



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

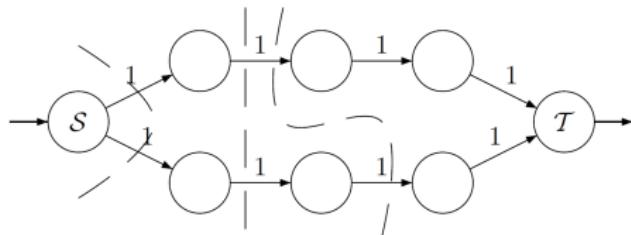
Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

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

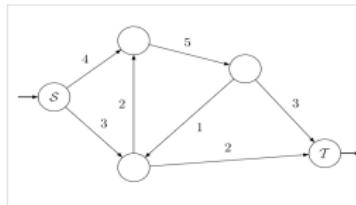
$$\sum_{e=(S,\cdot) \in E} f(e) = \sum_{e=(T,\cdot) \in E} f(e)$$

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.

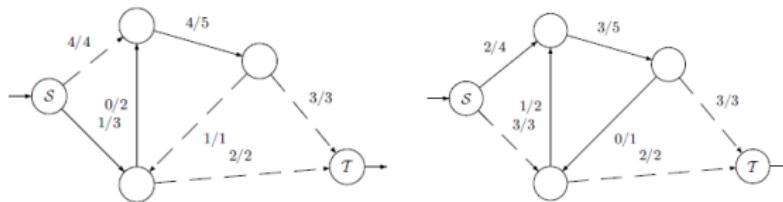
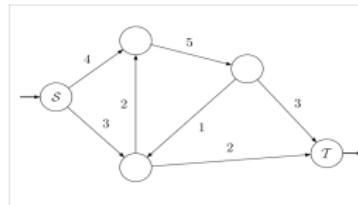


Approches d'optimisation par graphe : Méthode des "Graph-Cuts"

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 :

$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j) \quad (x : \text{image}, x_j : \text{valeur de } x \text{ au pixel } j)$$

- typiquement :

- D → attache aux données :
- V → 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 :

$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j) \quad (x : \text{image}, x_j : \text{valeur de } x \text{ au pixel } j)$$

- typiquement :
 - $D \rightarrow$ attache aux données :
 - $V \rightarrow$ 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 :

$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j) \quad (x : \text{image}, x_j : \text{valeur de } x \text{ au pixel } j)$$

- 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

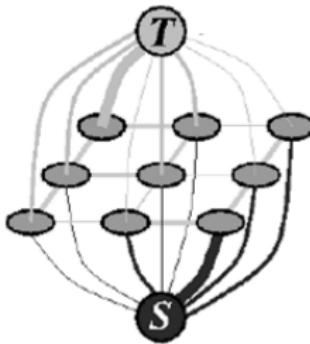
$$E(x) = \sum_i D_i(x_i) + \alpha \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j)$$

est un problème NP-complet ("arg")

Méthode des "Graph-Cuts" : utilisation en image

Principe général

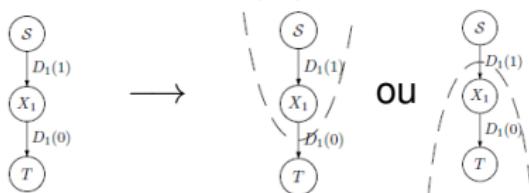
- On transforme le problème de minimisation en problème de coupe minimal dans un graphe
- On construit un graphe constitué d'un sommet par variable de l'énergie (pour chaque pixel)
- On ajoute une source et un puits, qui vont représenter en quelque sorte les valeurs possibles de chaque variable (0 pour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



Méthode des "Graph-Cuts" : utilisation en image

exemples simples

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

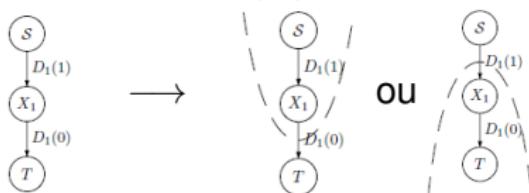


- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.

Méthode des "Graph-Cuts" : utilisation en image

exemples simples

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



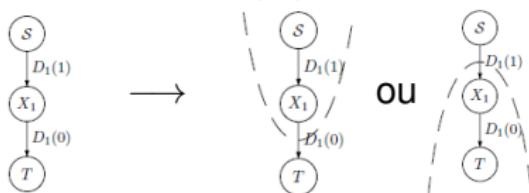
- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.

- 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 $D_1(x_1)$.



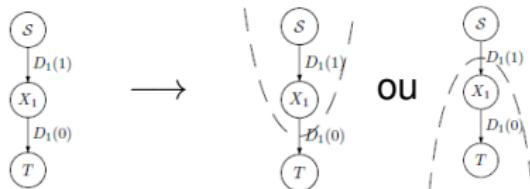
- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.

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

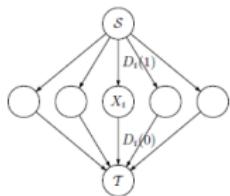
Méthode des "Graph-Cuts" : utilisation en image

exemples simples

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



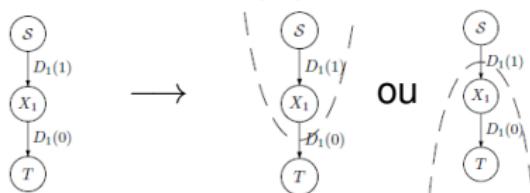
- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.
- avec $D_1(0) = I_1, \quad D_1(1) = 1 - I_1$ → seuillage à $\frac{1}{2}$
- n pixels sans interaction ($\alpha = 0$) : on veut minimiser $\sum_i D(x_i)$.



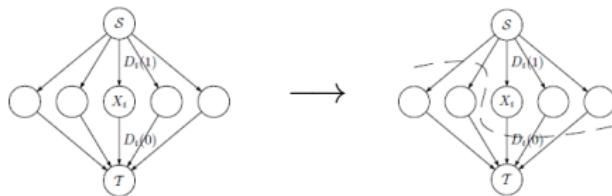
Méthode des "Graph-Cuts" : utilisation en image

exemples simples

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



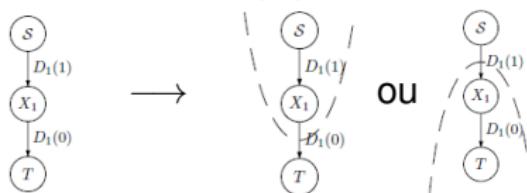
- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.
- avec $D_1(0) = I_1, \quad D_1(1) = 1 - I_1$ → seuillage à $\frac{1}{2}$
- n pixels sans interaction ($\alpha = 0$) : on veut minimiser $\sum_i D(x_i)$.



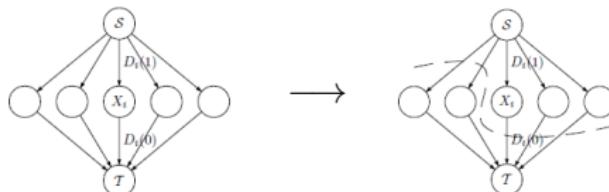
Méthode des "Graph-Cuts" : utilisation en image

exemples simples

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



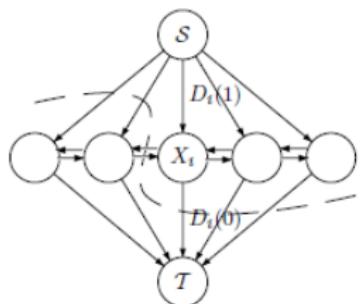
- Si $D_1(0) < D_1(1)$ la solution est donc $x_1 = 0$ et récip.
- avec $D_1(0) = I_1, D_1(1) = 1 - I_1 \rightarrow$ seuillage à $\frac{1}{2}$
- n pixels sans interaction ($\alpha = 0$) : on veut minimiser $\sum_i D(x_i)$.



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

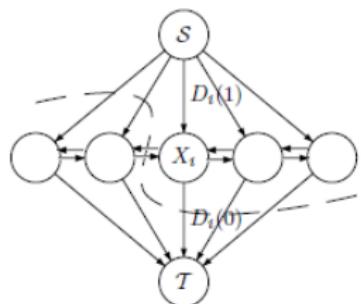
- n pixels avec interactions ($\alpha > 0$)



- 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 ($\alpha > 0$)



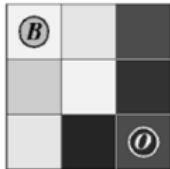
- il faut que la valeur d'une coupe, corresponde à l'énergie de la configuration associée Ainsi :
Coupe minimale \leftrightarrow énergie minimale
- Les poids des arêtes sont plus compliqués à définir (en fonction de l'énergie, i.e. de D et V)
- Principe : deux voisins seront reliés par une arête de poids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



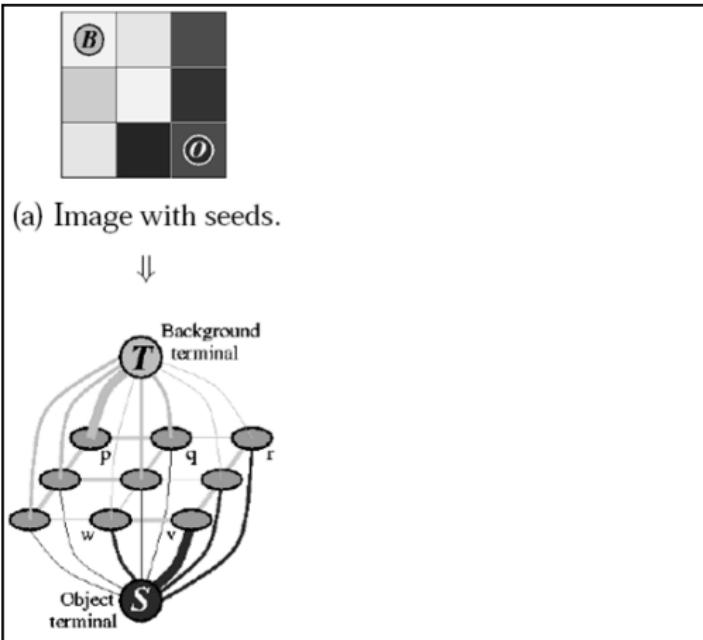
(a) Image with seeds.

- B et O contraintes dures spécifiées par l'utilisateur (B=Background, O=Object)

Boykov, Funka-Lea "Graph Cuts and Efficient N-D Image Segmentation" IJCV 2006

Méthode des "Graph-Cuts" : segmentation interactive

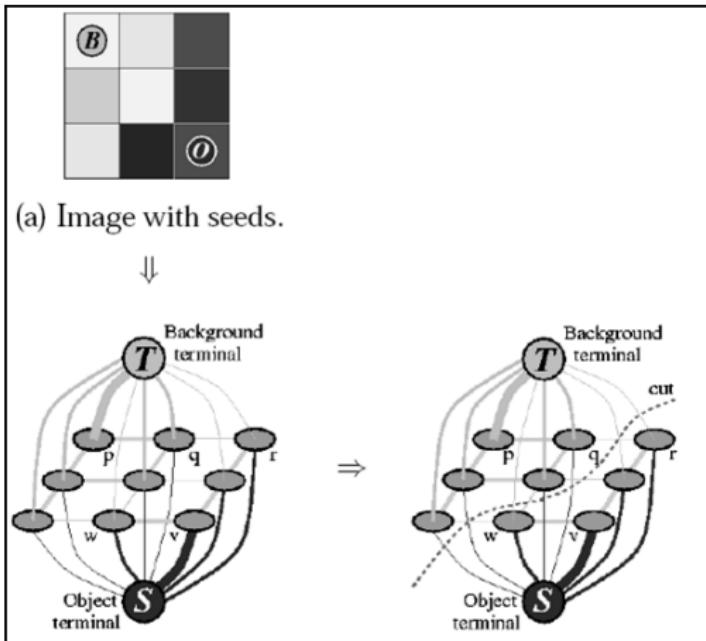
- B et O contraintes dures spécifiées par l'usager (B=Background, O=Object)
- $w_{pt} = \infty, w_{sp} = 0, w_{vt} = 0, w_{sv} = \infty$



Boykov, Funka-Lea "Graph Cuts and Efficient N-D Image Segmentation" IJCV 2006

Méthode des "Graph-Cuts" : segmentation interactive

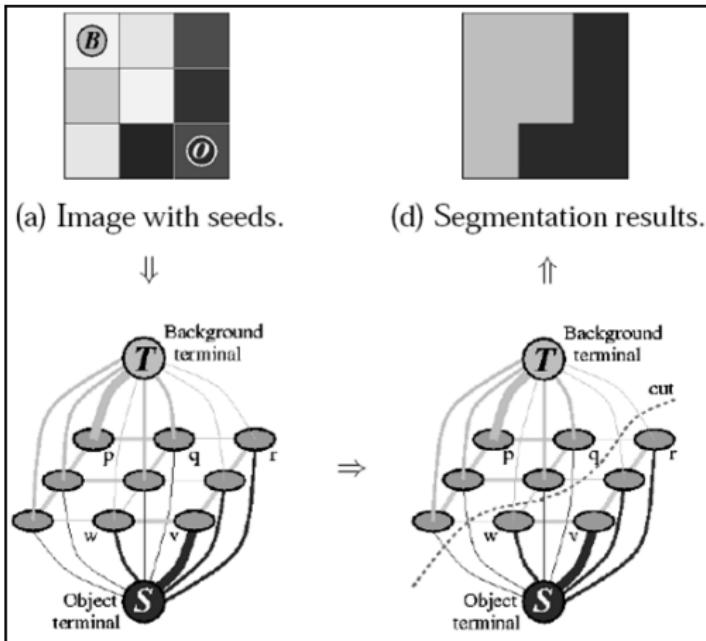
- B et O contraintes dures spécifiées par l'usager (B=Background, O=Object)
- $w_{pt} = \infty, w_{sp} = 0, w_{vt} = 0, w_{sv} = \infty$



Boykov, Funka-Lea "Graph Cuts and Efficient N-D Image Segmentation" IJCV 2006

Méthode des "Graph-Cuts" : segmentation interactive

- B et O contraintes dures spécifiées par l'usager (B=Background, O=Object)
- $w_{pt} = \infty, w_{sp} = 0, w_{vt} = 0, w_{sv} = \infty$



Boykov, Funka-Lea "Graph Cuts and Efficient N-D Image Segmentation" IJCV 2006

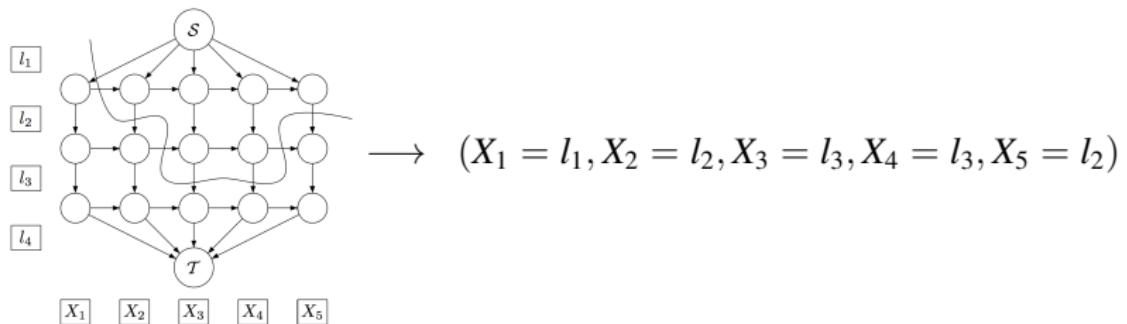
Méthode des "Graph-Cuts" : segmentation



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)

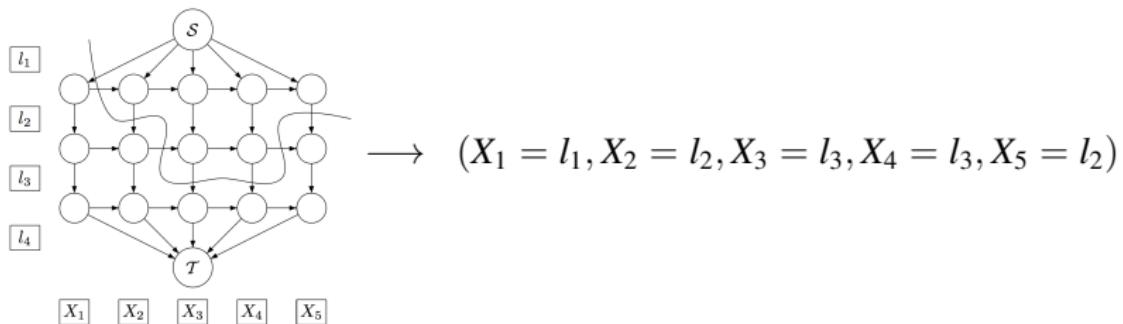
Méthode des "Graph-Cuts" :segmentation

- Le cas multi-étiquette

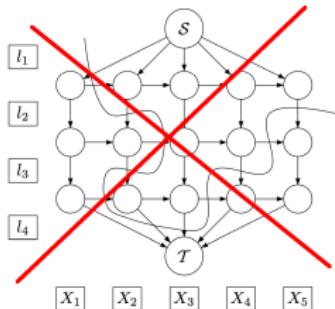


Méthode des "Graph-Cuts" :segmentation

- Le cas multi-étiquette

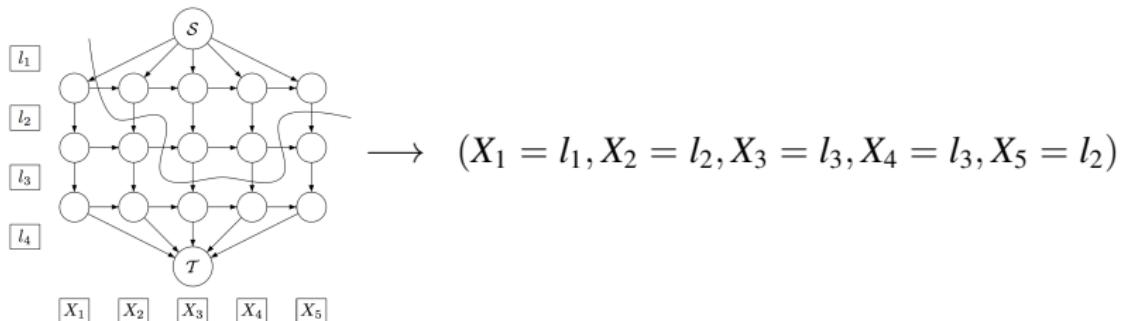


- Attention, des précautions à prendre :

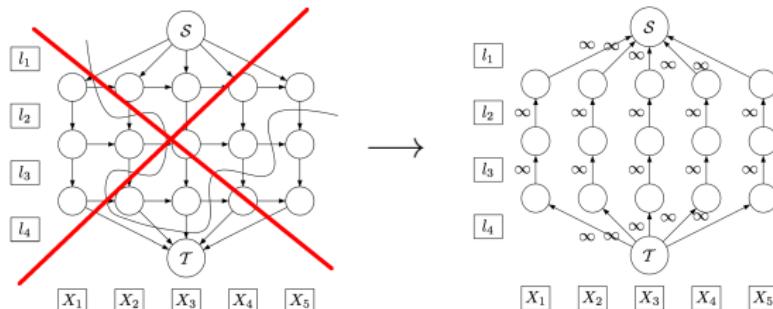


Méthode des "Graph-Cuts" :segmentation

- Le cas multi-étiquette



- 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

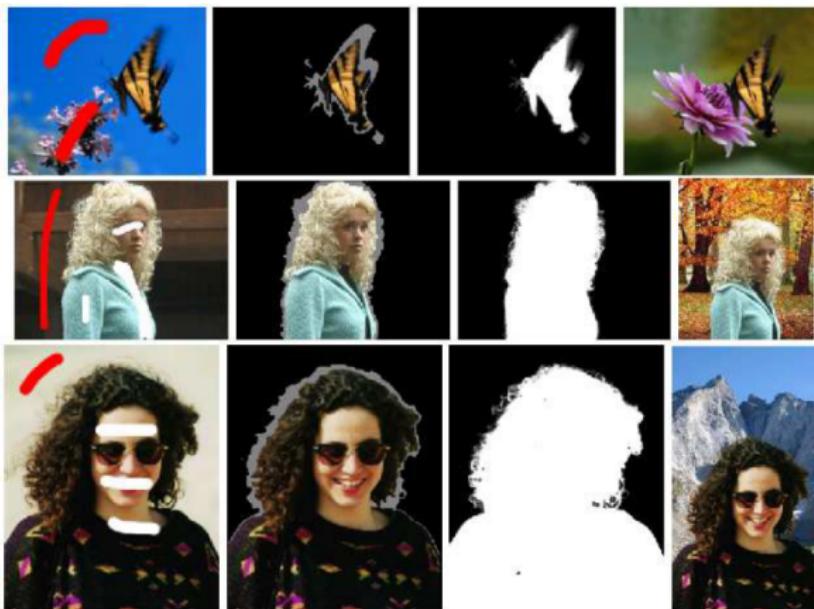
Méthode des "Graph-Cuts" : segmentation



segmentation avec modèle paramétrique du fond et du premier plan

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. grabcut : interactive foreground extraction using iterated graph cuts. ACM Trans. Graph.

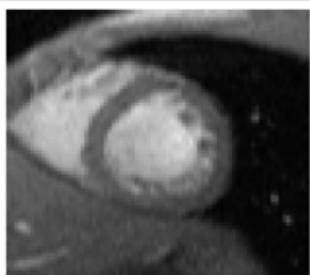
Méthode des "Graph-Cuts" : segmentation



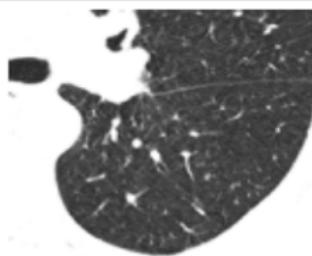
idem avec du α -matting

Juan.O. and Keriven.R. Trimap segmentation for fast and user-friendly alpha matting. 2005

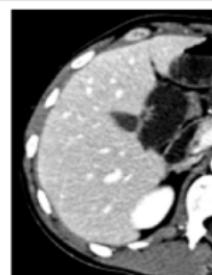
Méthode des "Graph-Cuts" : segmentation



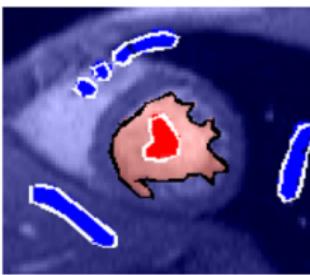
(c) Cardiac MR



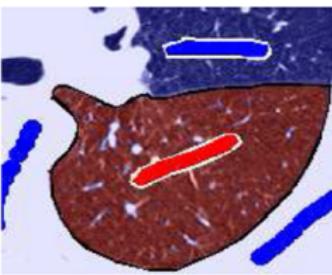
(e) Lung CT



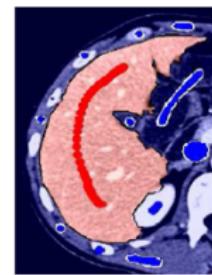
(g) Liver MR



(d) LV Segment

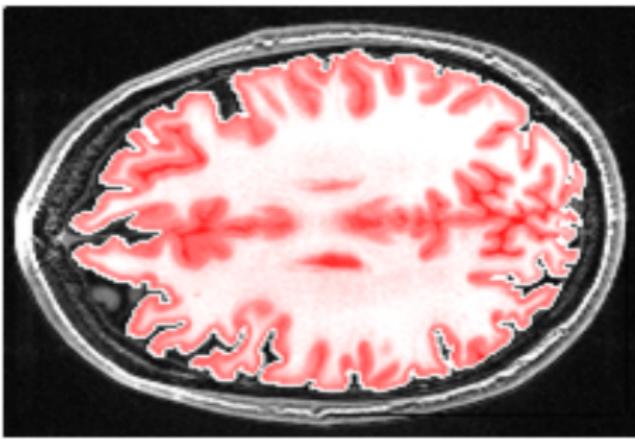


(f) Lobe Segment



(h) Liver Segment

Méthode des "Graph-Cuts" : segmentation



- Pour aller plus loin

<http://mickaelpechaud.free.fr/graphcuts.pdf>
et références associées

Approches orientées "modèles"

Approches orientées "données"

Approches basée atlas/recalage

Approches Globales → seuillage, k-means, seuillage adaptatif

Approches "Régions" → croissance, markoviennes, graph-cut

Approches "contours" → gradient, Canny, Watershed, contours actifs

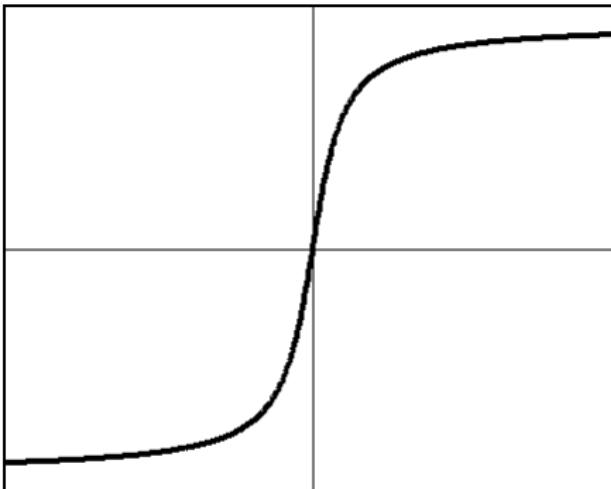
Approches "contours"

Approches "contours"

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

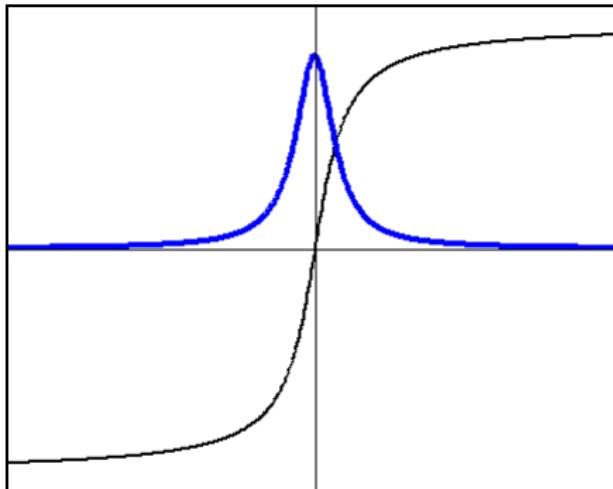
Approches "contours"

- 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



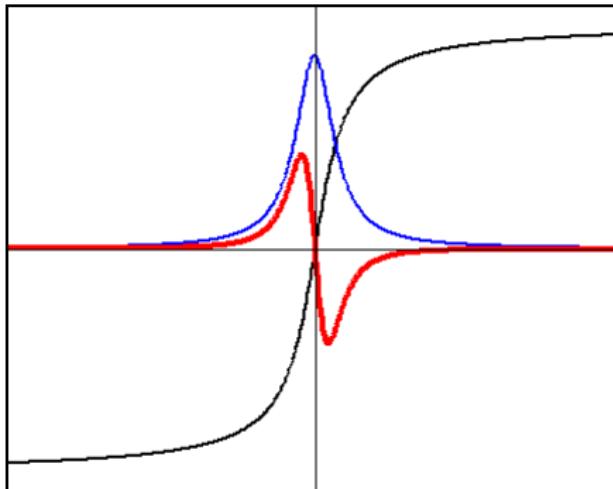
Approches "contours"

- 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

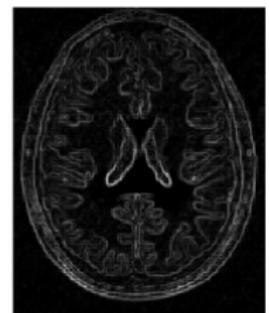
$$G = \overrightarrow{\text{grad}} f(x, y) \quad \begin{cases} G_x = \frac{\partial f}{\partial x}(x, y) \\ G_y = \frac{\partial f}{\partial y}(x, y) \end{cases}$$



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

$$\sigma = 3$$



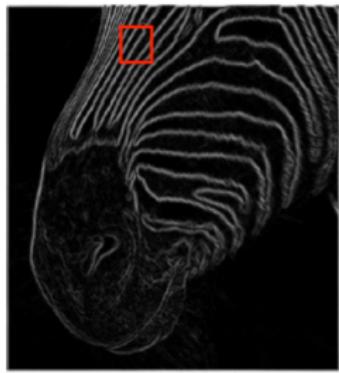
seuil = 5

seuil = 10

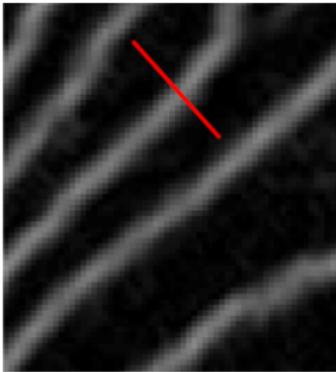
seuil = 15

Détecteur de Canny

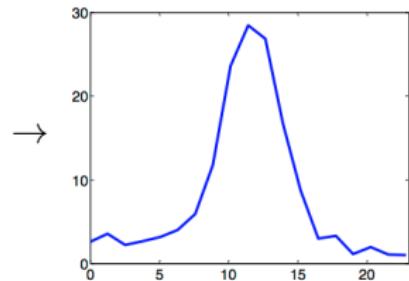
- Profil dans la direction du gradient



→



$\sigma = 1$



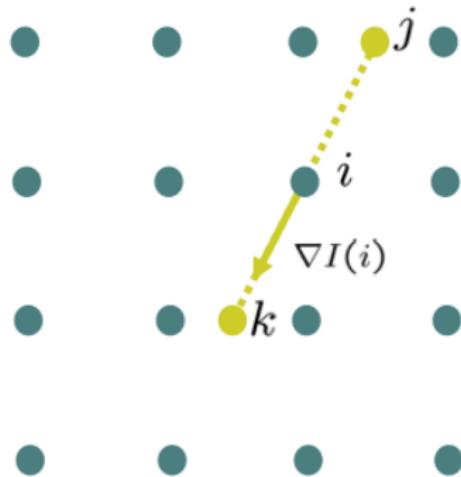
- Principe : ne garder que les maxima locaux (dans la direction du gradient) et supérieurs à un seuil

Détecteur de Canny

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

contour en i si

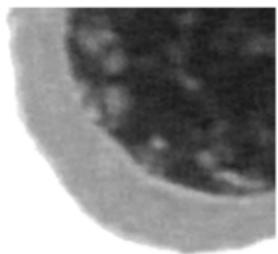
- $|\nabla I(i)| > |\nabla I(j)|$
- $|\nabla I(i)| > |\nabla I(k)|$
- $|\nabla I(i)| > \text{seuil}$



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
 $S=100$

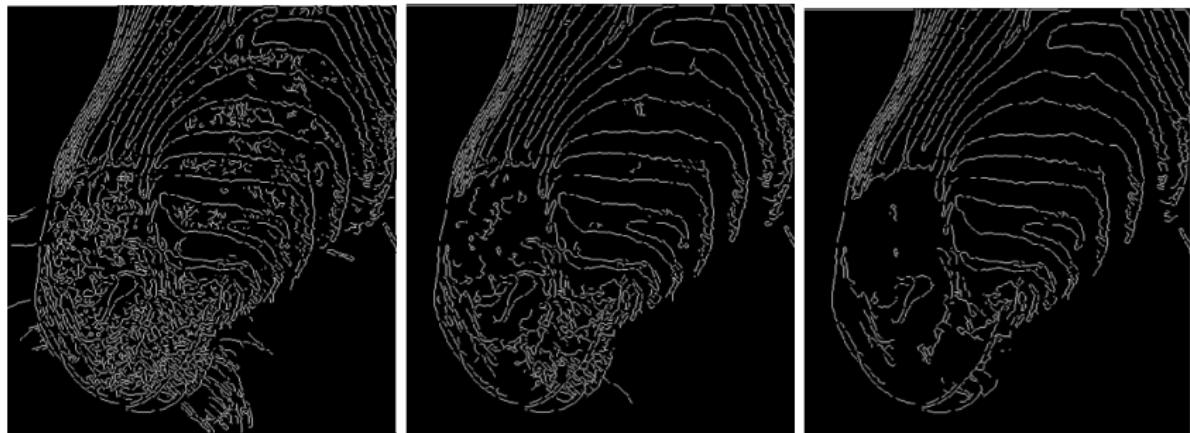


Seuillage par
hystérésis
 $Sh=100$, $Sb=50$

Approches orientées "modèles"
Approches orientées "données"
Approches basée atlas/recalage

Approches Globales → seuillage, k-means, seuillage adaptatif
Approches "Régions" → croissance, markoviennes, graph-cut
Approches "contours" → gradient, Canny, Watershed, contours actifs

Détecteur de Canny, résultats



Détecteur de Canny, résultats



Détecteur de Canny, résultats



Approches orientées "modèles"
Approches orientées "données"
Approches basée atlas/recalage

Approches Globales → seuillage, k-means, seuillage adaptatif
Approches "Régions" → croissance, markoviennes, graph-cut
Approches "contours" → gradient, Canny, Watershed, contours actifs

Détecteur de Canny, résultats



Laplacien de l'image

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

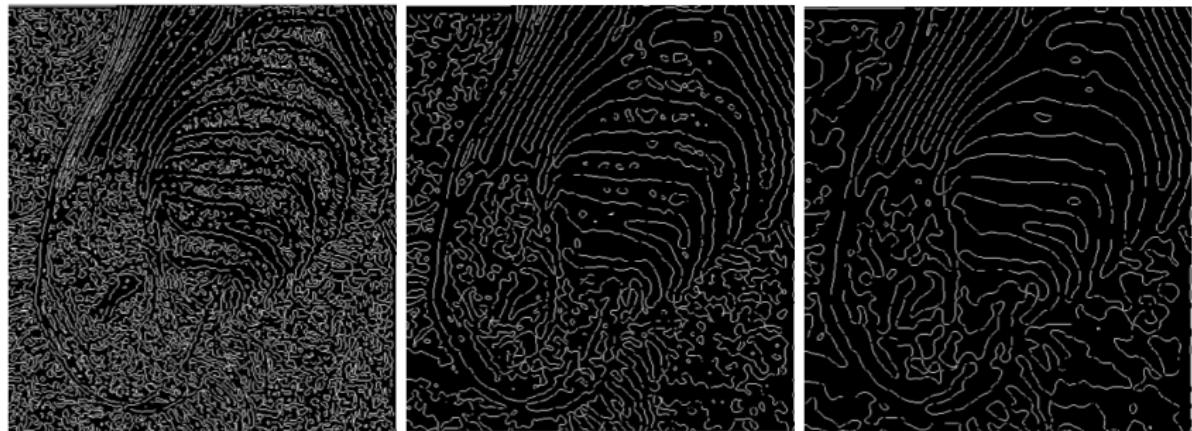
"équivalent de la dérivée seconde" en 2D



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

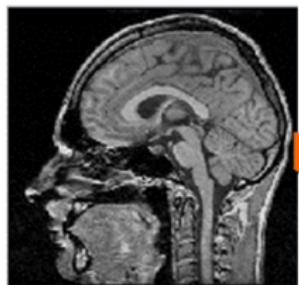


Image originale

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

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

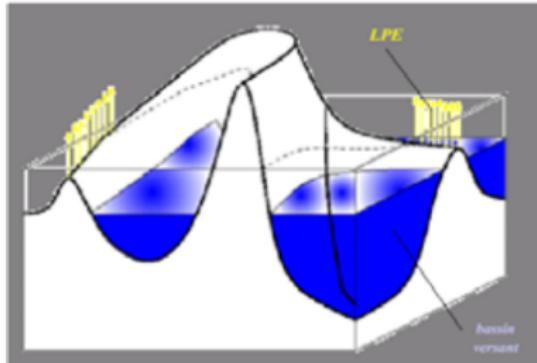
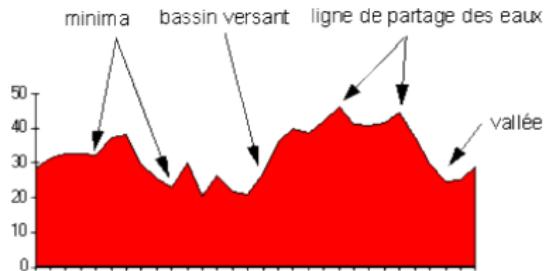
Lignes de partage des eaux (watershed)

- 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

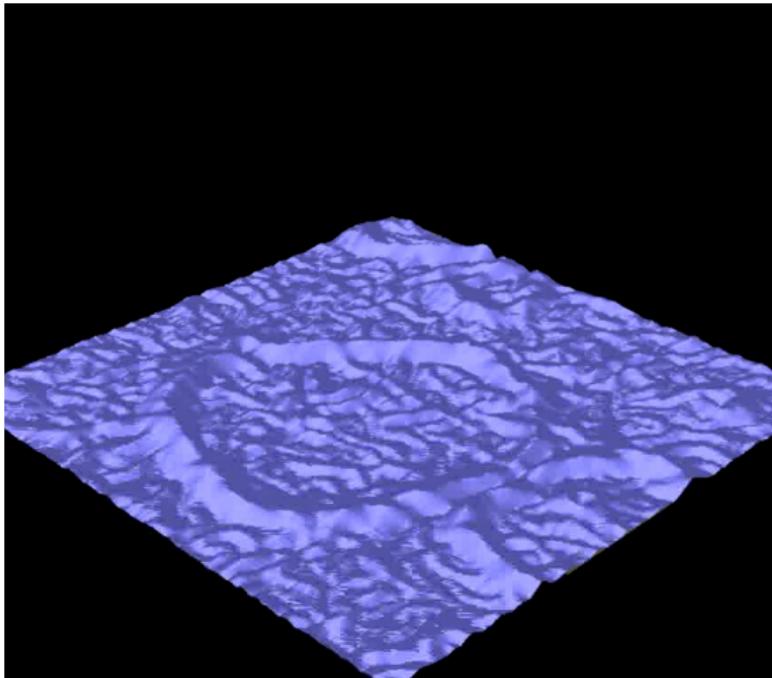


Lignes de partage des eaux (watershed)

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

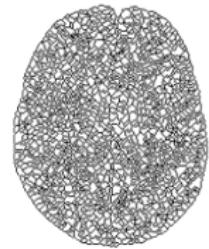
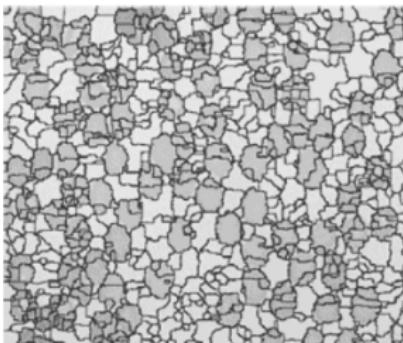
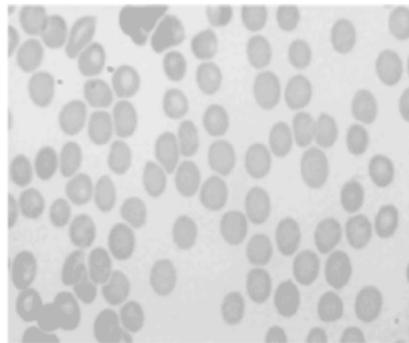


Lignes de partage des eaux (watershed)



<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

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

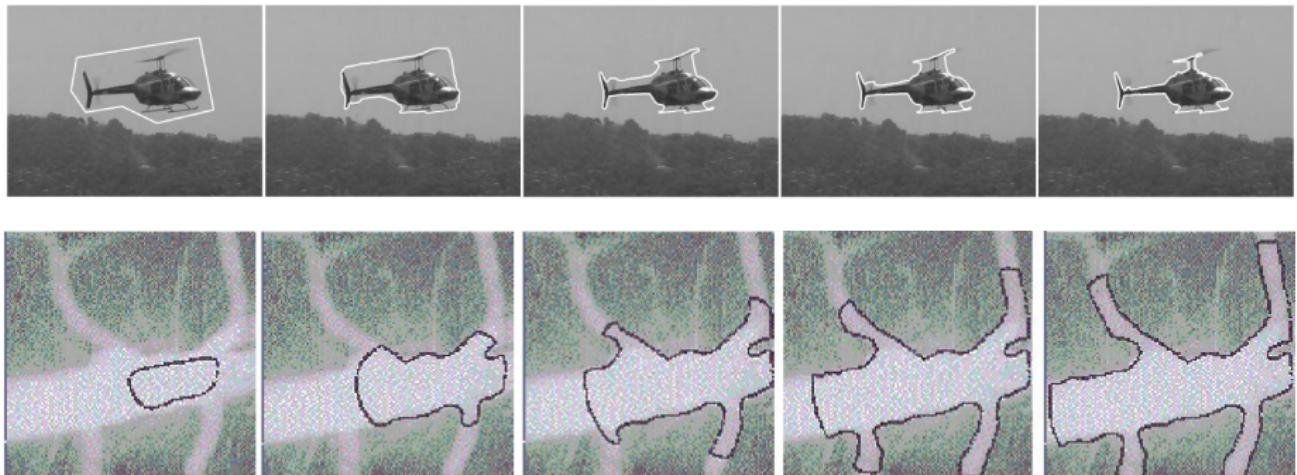
- $E_{\text{image}}(\mathcal{C})$ va essayer d'attirer la courbe sur les contours de l'image
- $E_{\text{interne}}(\mathcal{C})$ permet d'ajouter des a priori sur la forme de la courbe que l'on cherche (régularité)
- Contours actifs géodésiques :

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

- 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 ← **Level Set**

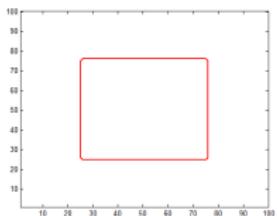
Level Set : Osher-Sethian, 1988.

Idée principale :

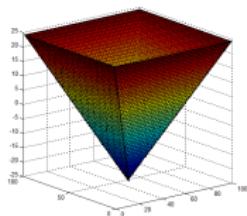
- la courbe \mathcal{C} est représentée de manière **implicite**.
- \mathcal{C} est représenté par une surface de dimension supérieure Φ dont elle est le **niveau zéro**
$$\mathcal{C} = \left\{ (x, y) \mid \Phi(x, y) = 0 \right\}$$
- Pour une courbe (dans \mathbb{R}^2) → ϕ est une image plane
- Pour une surface (dans \mathbb{R}^3) → ϕ est une image volumique

"Level Set"

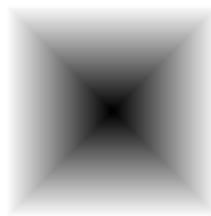
- une courbe plane (dans \mathbb{R}^2) : représentée par une fonction $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ (image, surface)



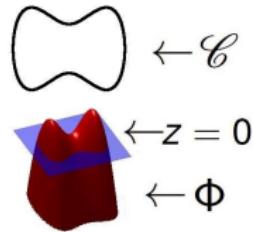
\mathcal{C}



Φ



Φ



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

Exemple : mouvement par courbure moyenne

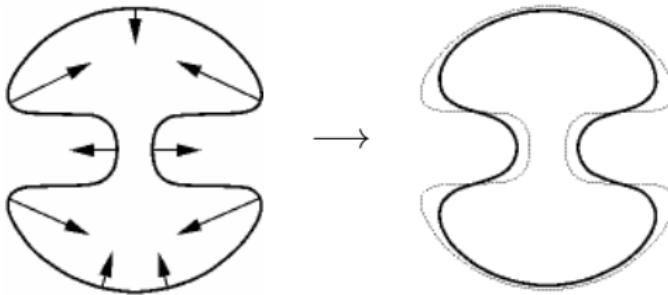
équation d'évolution sur \mathcal{C}

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

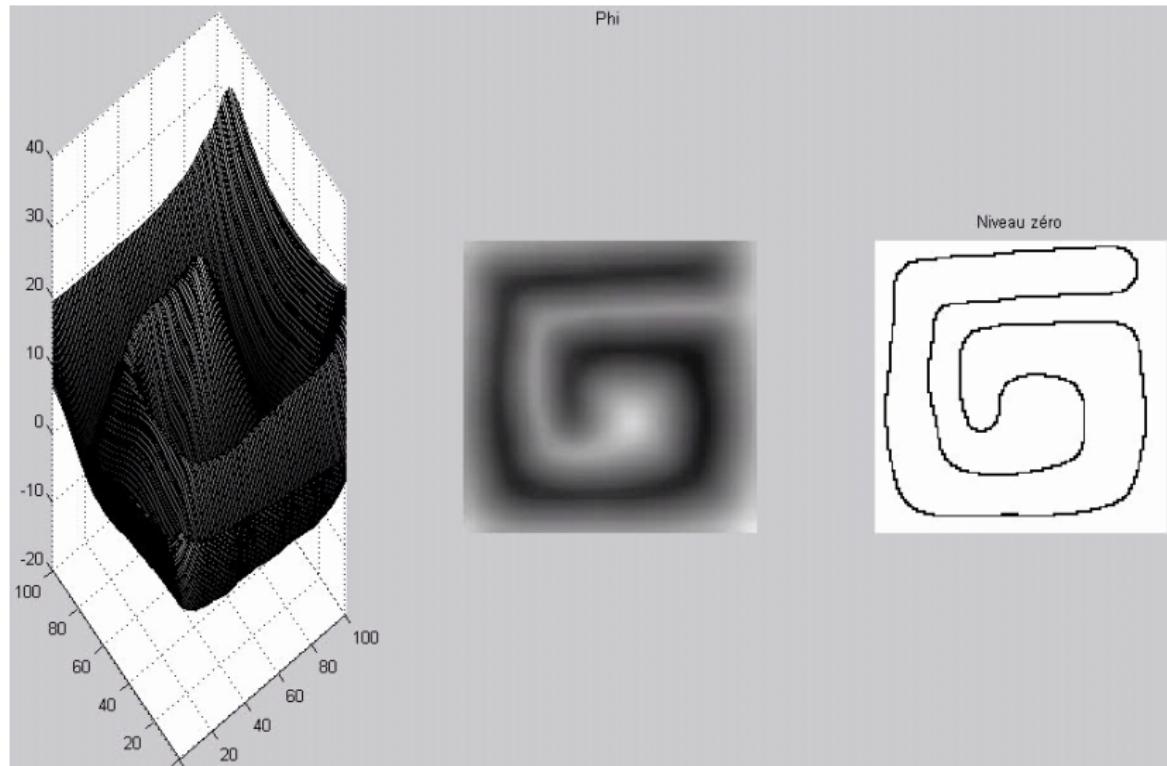
↔

équation d'évolution sur Φ

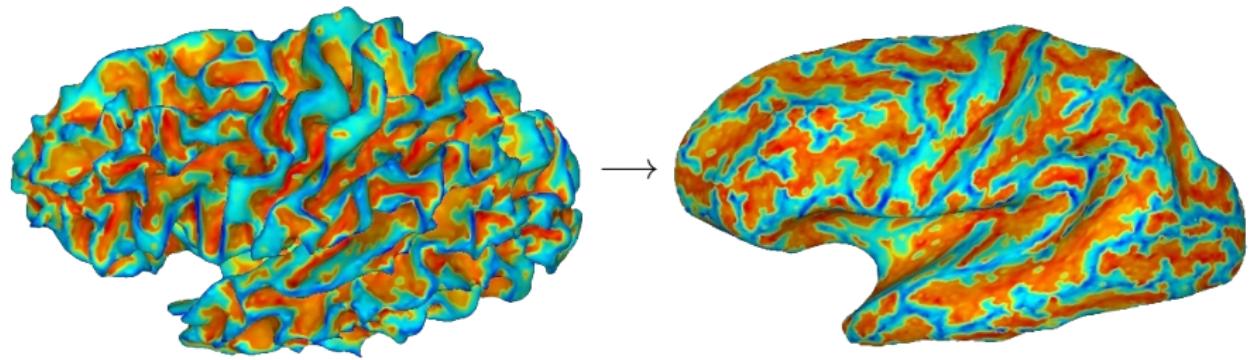
$$\begin{cases} \frac{\partial \Phi}{\partial t} = \operatorname{div}\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right) |\nabla \Phi| \\ \Phi(0, x, y) = \Phi_0(x, y) \end{cases}$$



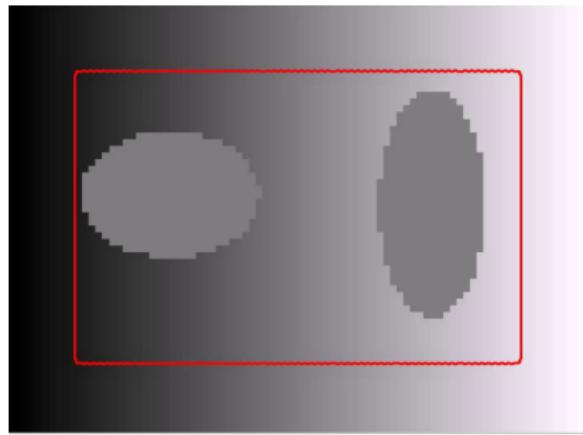
Exemple : mouvement par courbure moyenne



Exemple : mouvement par courbure moyenne

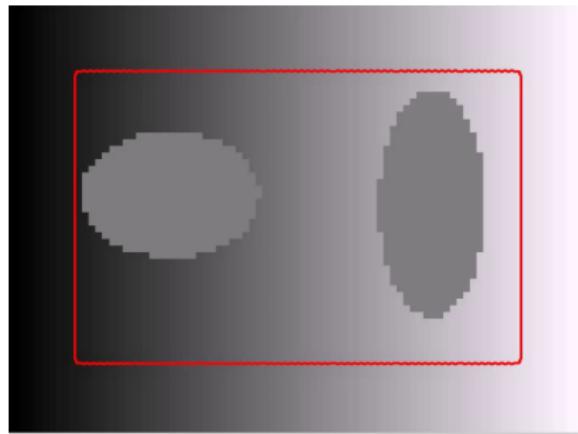


Segmentation par contours actifs : exemples

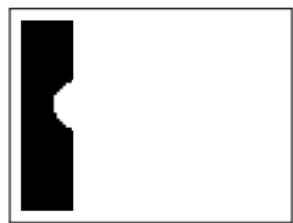


Seuillage ?

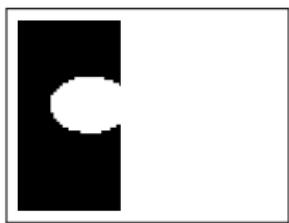
Segmentation par contours actifs : exemples



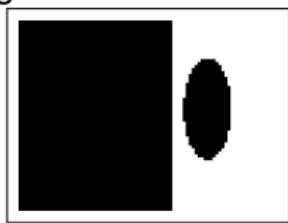
Seuillage ?



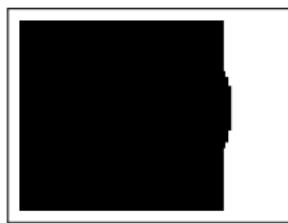
$s = 50$



$s = 100$

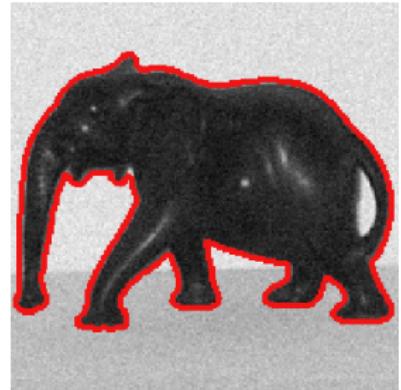
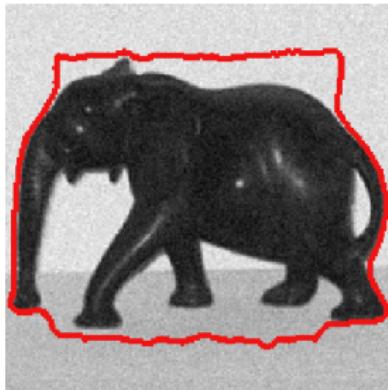
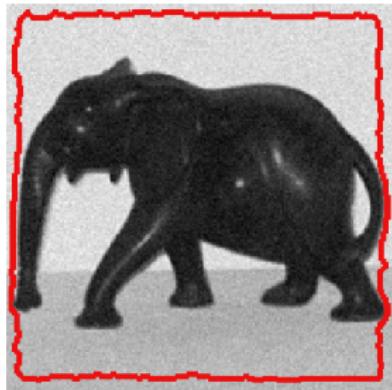


$s = 150$

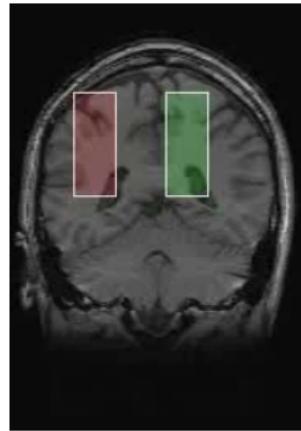
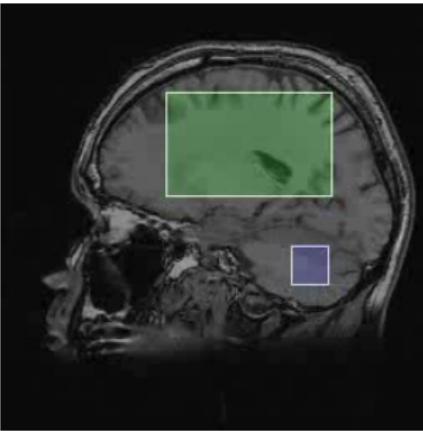
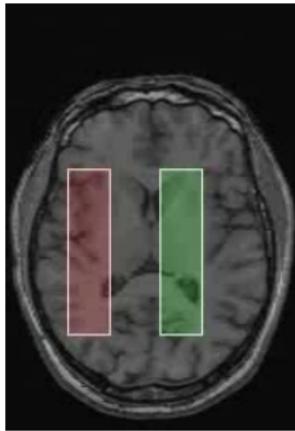


$s = 200$

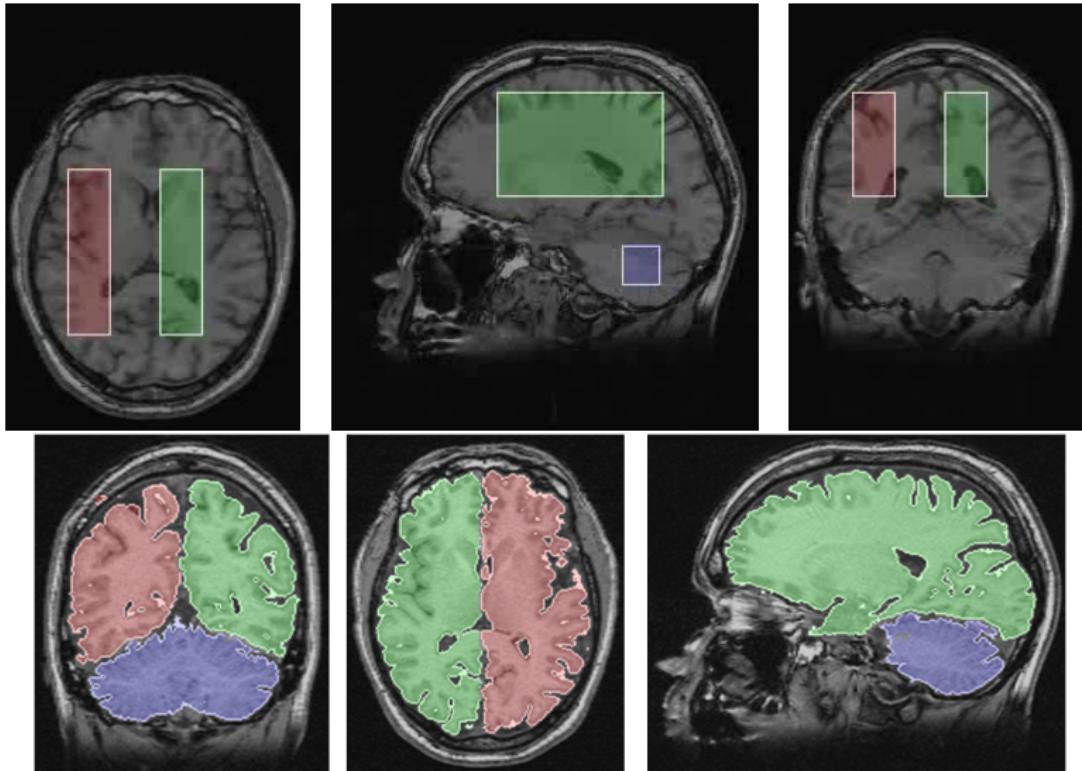
Segmentation par contours actifs : exemples



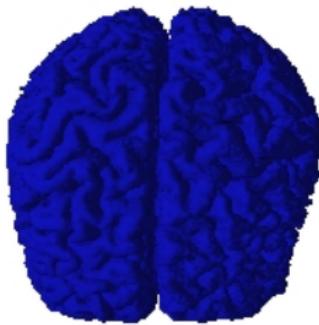
Segmentation par contours actifs : exemples



Segmentation par contours actifs : exemples



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

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

Représentation implicite : récapitulatif

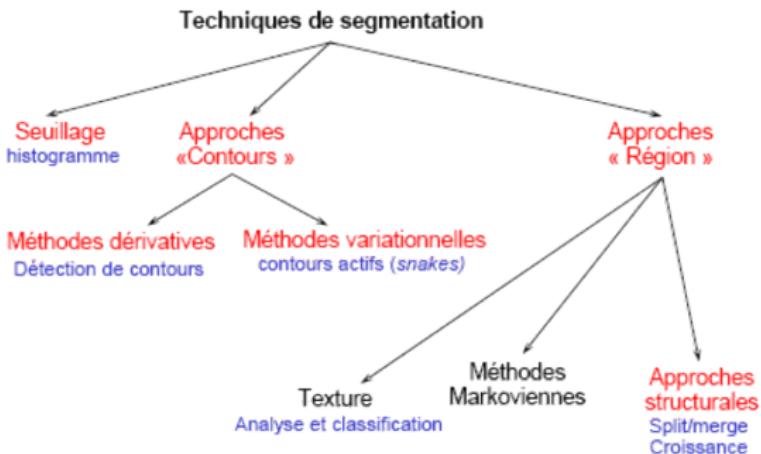
Avantages :

- 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

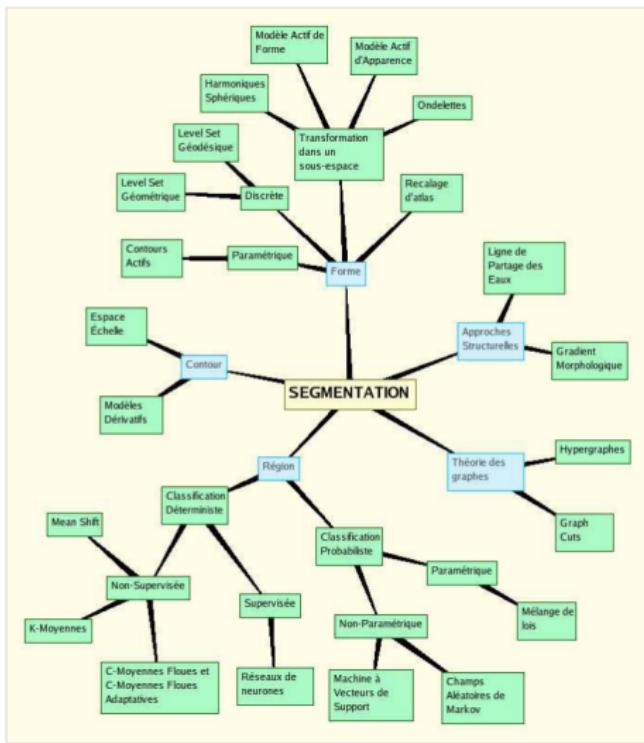
Inconvénients :

- 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 basée données → machine learning

Apprentissage Automatique, Machine Learning

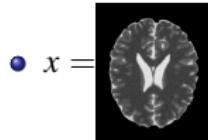
- On a des entrées $x_i \in E$ et des sorties $y_i \in F$ connues
ensemble d'apprentissage
- On veut apprendre une fonction $f : E \rightarrow F$ telle que
 - $f(x_i) \approx y_i$
 - si x est nouveau, $f(x)$ fournit une sortie "satisfaisante"

Apprentissage Automatique, Machine Learning

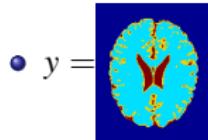
- On a des entrées $x_i \in E$ et des sorties $y_i \in F$ connues
ensemble d'apprentissage
- On veut apprendre une fonction $f : E \rightarrow F$ telle que
 - $f(x_i) \approx y_i$
 - si x est nouveau, $f(x)$ fournit une sortie "satisfaisante"
- Pour segmentation :
 - $x =$
 - $y =$

Apprentissage Automatique, Machine Learning

- On a des entrées $x_i \in E$ et des sorties $y_i \in F$ connues
ensemble d'apprentissage
- On veut apprendre une fonction $f : E \rightarrow F$ telle que
 - $f(x_i) \approx y_i$
 - si x est nouveau, $f(x)$ fournit une sortie "satisfaisante"
- Pour segmentation :



→ une image en ndg, souvent 3D



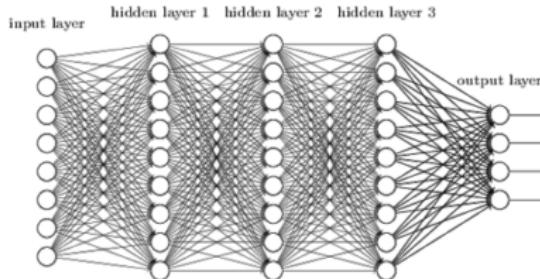
→ une image d'étiquettes de même taille

Apprentissage Automatique, Machine Learning

- Choix de la méthode d'apprentissage (SVM, Random Forest, Neural Network) = choix de la famille de fonctions

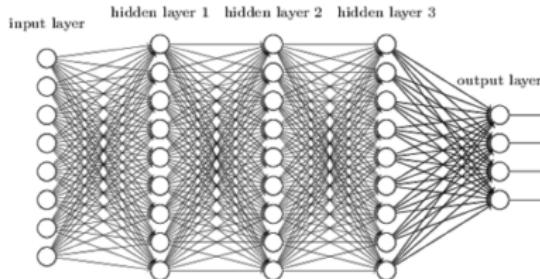
Apprentissage Automatique, Machine Learning

- Choix de la méthode d'apprentissage (SVM, Random Forest, Neural Network) = choix de la famille de fonctions
- En ce moment : réseaux de neurones profonds / deep learning



Apprentissage Automatique, Machine Learning

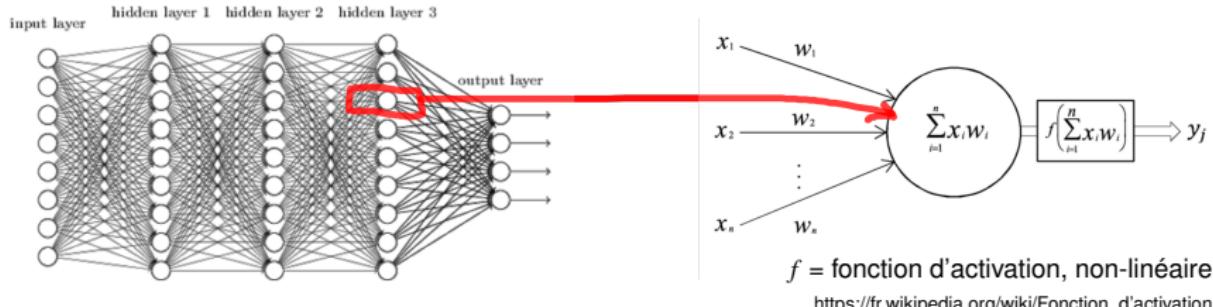
- Choix de la méthode d'apprentissage (SVM, Random Forest, Neural Network) = choix de la famille de fonctions
- En ce moment : réseaux de neurones profonds / deep learning



- ce réseau représente une fonction (non linéaire) de \mathbb{R}^8 dans \mathbb{R}^4

Apprentissage Automatique, Machine Learning

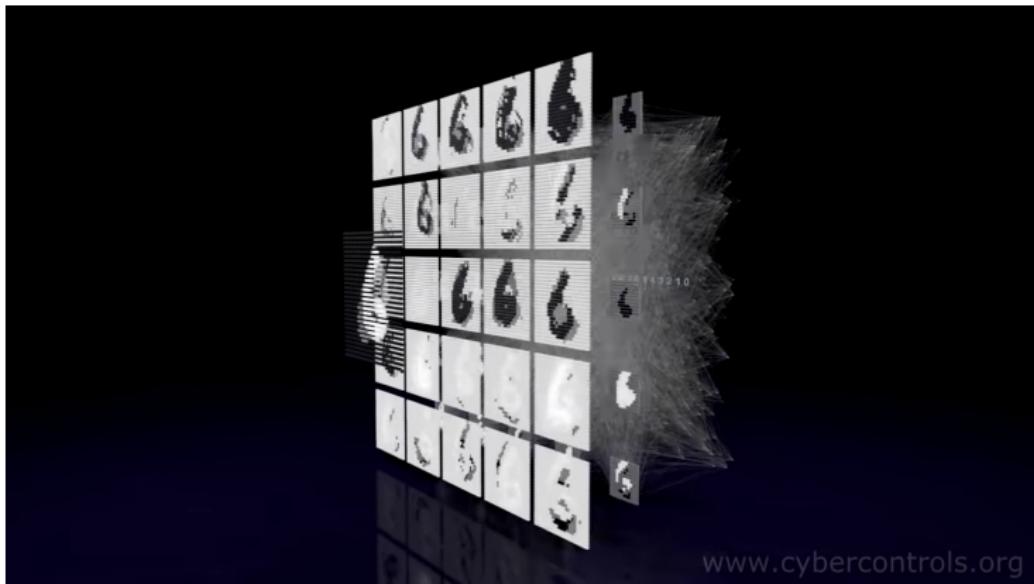
- Choix de la méthode d'apprentissage (SVM, Random Forest, Neural Network) = choix de la famille de fonctions
- En ce moment : réseaux de neurones profonds / deep learning



- ce réseau représente une fonction (non linéaire) de \mathbb{R}^8 dans \mathbb{R}^4
- un paramètre $w_{i,j} \in \mathbb{R}$ pour chaque connexion
- des millions de paramètres pour un réseau de grande taille mais ça ne reste "qu'une fonction paramétrée"

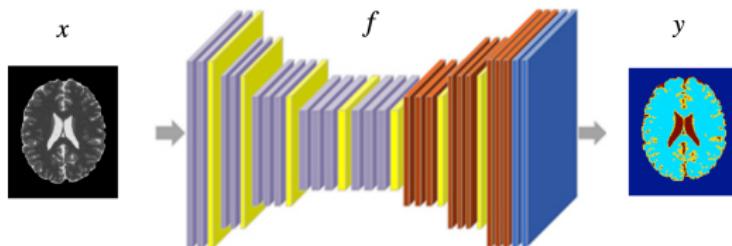
Deep Learning

- La phase d'apprentissage consiste donc à **calibrer les paramètres du réseau** de manière à avoir les y_i (exples du *learning set*) proche des $f(x_i)$



Deep Learning & Segmentation

- Segmentation :



- Pourquoi un réseau de neurones ?

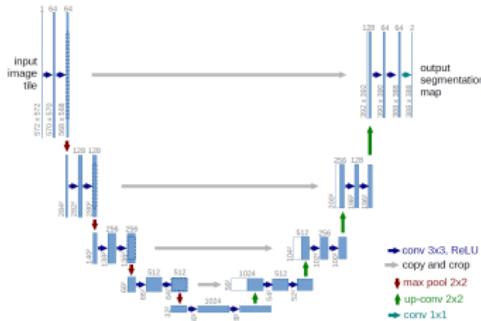
- permet d'approcher n'importe quelle fonction (nbr quelconque de couches/neurones)
- tout en permettant un apprentissage susceptible de fournir une bonne généralisation : de bonnes chances de produire des solutions satisfaisantes en ayant appris à partir de données légèrement différentes

Deep Learning & Segmentation

- **Apprentissage** : on apprend la fonction f (c'est à dire les paramètres du réseau neuronal) qui fait correspondre au mieux les y_i (connus) aux x_i (observations)
- **Prédiction** : pour une nouvelle observation x , on lui applique f (avec les param. appris plus haut) et on obtient $f(x)$ une estimée de la segmentation

Deep Learning & Segmentation

- **Apprentissage** : on apprend la fonction f (c'est à dire les paramètres du réseau neuronal) qui fait correspondre au mieux les y_i (connus) aux x_i (observations)
- **Prédiction** : pour une nouvelle observation x , on lui applique f (avec les param. appris plus haut) et on obtient $f(x)$ une estimée de la segmentation
- Exemple : U-net³ : un réseau convolutionnel performant pour la segmentation d'images biomédicales



1. U-Net : Convolutional Networks for Biomedical Image Segmentation. O.Ronneberger, P.Fischer, T.Brox. 2015

Deep Learning & Segmentation

- **Apprentissage** : on cherche les param. w_1, w_2, \dots , qui minimise la différence entre les y_i et les $f(x_i)$

$$\arg \min_{w_1, w_2, \dots} \sum_{i=1}^N \text{dist} \left(f(x_i, w_1, w_2, \dots), y_i \right)$$

- En segmentation d'images, cette partie se fait (une fois pour toutes) sur des supercalculateurs (bcp de mémoire, GPU puissants)

Deep Learning & Segmentation

- **Apprentissage** : on cherche les param. w_1, w_2, \dots , qui minimise la différence entre les y_i et les $f(x_i)$

$$\arg \min_{w_1, w_2, \dots} \sum_{i=1}^N \text{dist} (f(x_i, w_1, w_2, \dots), y_i)$$

- En segmentation d'images, cette partie se fait (une fois pour toutes) sur des supercalculateurs (bcp de mémoire, GPU puissants)
- **Les données** d'apprentissage (x_i, y_i) : élément essentiel
 - le modèle va essayer de s'adapter au mieux à ces données
 - doivent être une source fiable d'information
 - les données d'apprentissage doivent être bien représentatives de celles sur lesquelles on voudra faire la prédiction

Deep Learning & Segmentation

- En imagerie médicale, on dispose en général d'un nombre relativement restreint de données pour réaliser l'apprentissage
 - Segmentations manuelles très longues et "qualifiées"
 - Difficultés à partager les données pour des raisons légales

Deep Learning & Segmentation

- En imagerie médicale, on dispose en général d'un nombre relativement restreint de données pour réaliser l'apprentissage
 - Segmentations manuelles très longues et "qualifiées"
 - Difficultés à partager les données pour des raisons légales
- Data Augmentation** : augmenter artificiellement la taille de l'ensemble d'apprentissage / aider le modèle à apprendre des principes d'invariance

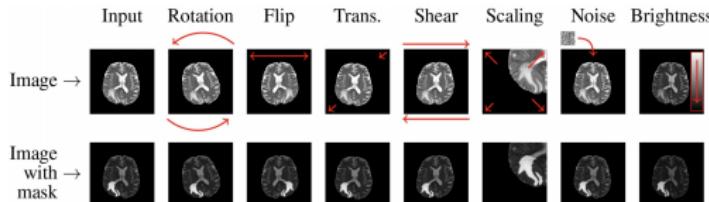


FIGURE 3 | Applying affine and pixel-level (discussed in more detail in section 2.3) transformations can help significantly increase the size (and potentially representativeness) of training sets. In this example, we generate seven new images based on the original MRI (coupled with its ground truth in the bottom row).

Data Augmentation for Brain-Tumor Segmentation : A Review. J.Nalepa, M.Marcinkiewicz and M.Kawulok. Front. Comput. Neurosci., 2019

- Utilisée à bon escient, l'augmentation de données peut nettement améliorer les performances du modèle appris

Approches orientées données

- Avantages

- Performant
- Pas de modèle à formuler
- Peut, en théorie, si on a suffisamment de données, approcher n'importe quelle fonction aussi complexe soit-elle

- Inconvénients

- extrêmement coûteux en temps de calcul (en particulier l'apprentissage)
- nécessite des données pour l'apprentissage
- modèle peu intuitif, résultats parfois difficilement interprétables
- Adversarial machine learning ?

https://www.youtube.com/embed/piYnd_wY1T8?rel=0

<https://www.youtube.com/embed/5nQ7loiPmdA?rel=0>

Imagerie Médicale : segmentation basée atlas/recalage

Imagerie médicale : segmentation basées sur un atlas

Atlas anatomique

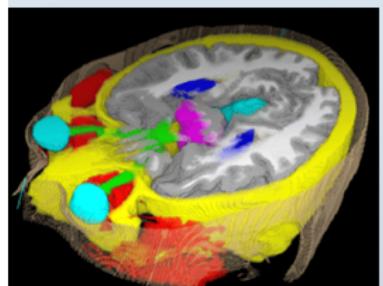
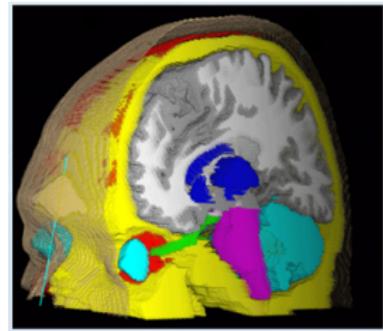
- image d'une anatomie moyenne
- segmentation associée effectuée par un expert

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 visibles



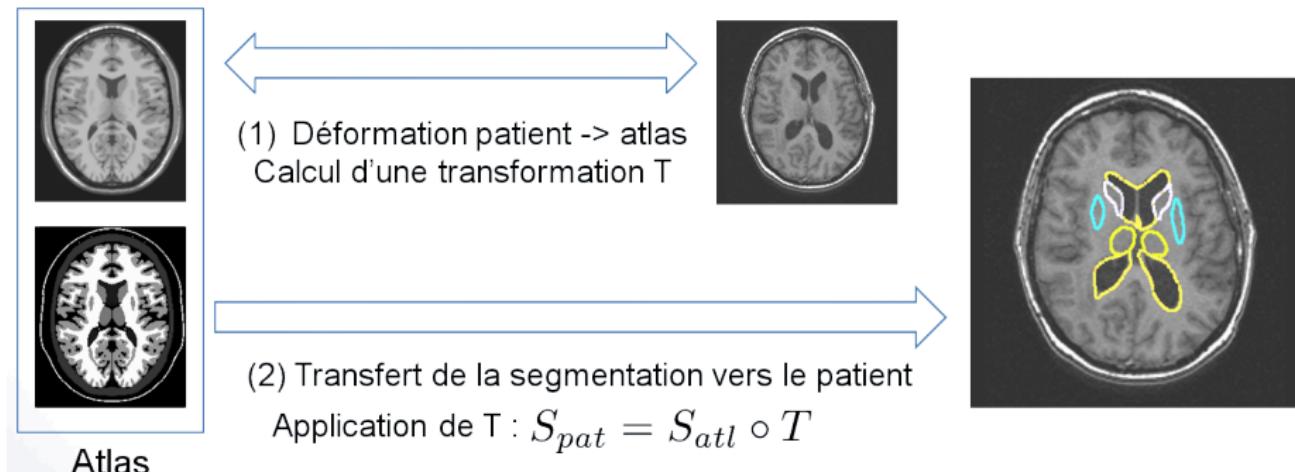
0. [Bondiau et al., 2005] Atlas-based automatic segmentation of MR images : Validation study on the brainstem in radiotherapy context. *Int J Radiat Oncol Biol Phys*

Construction d'un atlas

Images : Antoine Legouhy

Segmentation par Atlas : Méthode

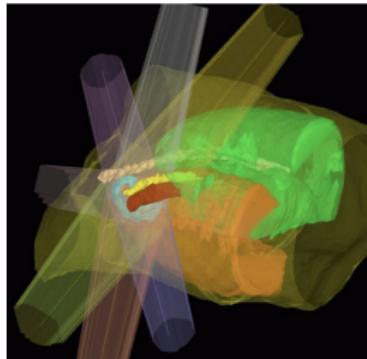
- On transforme le problème de segmentation en un problème de **recalage**



On a donc besoin d'être capable de **recaler** une image sur une autre.
→ **prochaine partie**

Segmentation par Atlas pour la radiothérapie

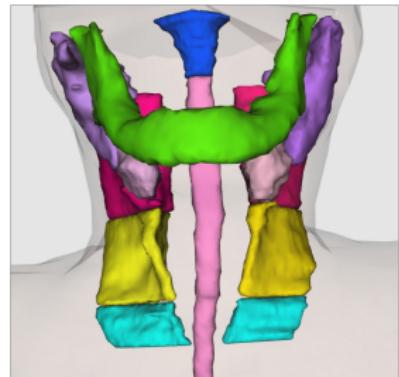
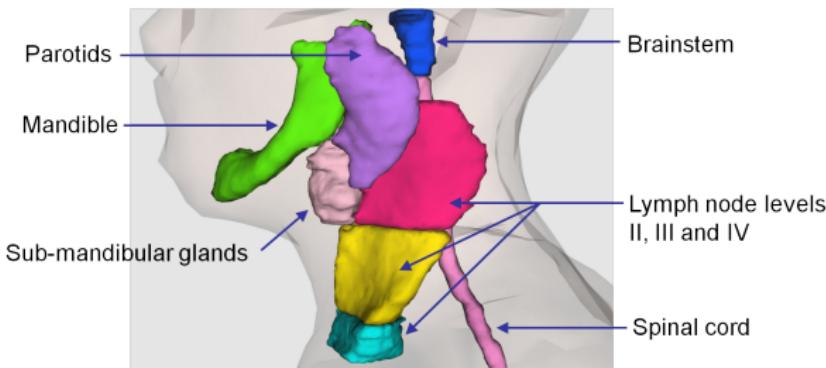
- méthode de traitement des cancers →, radiations pour détruire les cellules cancéreuses tout en épargnant les tissus sains périphériques.



- Besoin d'un planning précis
 - calcul des doses sur chaque organe
 - comparaison avec les niveaux attendus
 - requiert la segmentation des structures

Segmentation par Atlas pour la radiothérapie

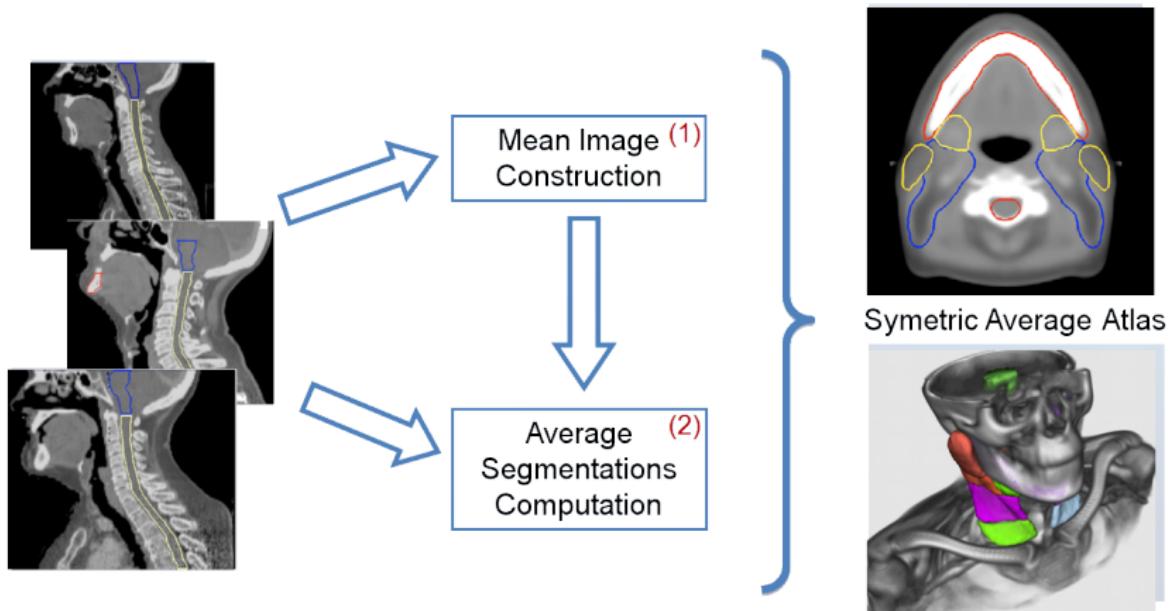
- Segmentation par atlas, exemple : Tête et nuque



0. [Grégoire et al., 2003] CT-based delineation of lymph node levels and related CTVs in the node-negative neck. 2003.

Segmentation par atlas, exemple : Tête et nuque

- Construction de l'atlas à partir de n images manuellement segmentées



0. Atlas-Based Delineation of Lymph Node Levels in Head and Neck Computed Tomography
Images O. Commowick et al. 2008

Segmentation par atlas, exemple : Tête et nuque

- Construction de l'atlas à partir de n images manuellement segmentées

Segmentation par atlas, exemple : Tête et nuque

- Construction de l'atlas à partir de n images manuellement segmentées
- des problèmes
 - comment faire la moyenne des segmentations ? → cf TP STAPLE

Segmentation par atlas, exemple : Tête et nuque

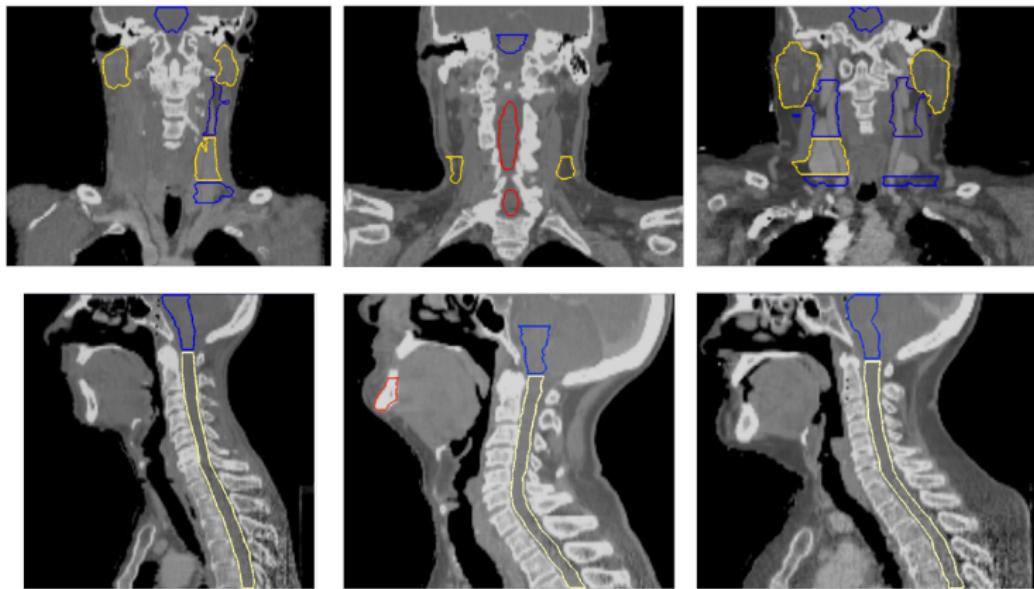
- Construction de l'atlas à partir de n images manuellement segmentées
- des problèmes
 - comment faire la moyenne des segmentations ? → cf TP STAPLE
 - dans quel "référentiel" faire la moyenne ? → une solution : faire aussi la moyenne des transformations

Segmentation par atlas, exemple : Tête et nuque

- Construction de l'atlas à partir de n images manuellement segmentées
- des problèmes
 - comment faire la moyenne des segmentations ? → cf TP STAPLE
 - dans quel "référentiel" faire la moyenne ? → une solution : faire aussi la moyenne des transformations
 - amélioration par symétrisation du résultat

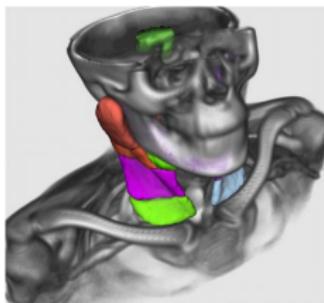
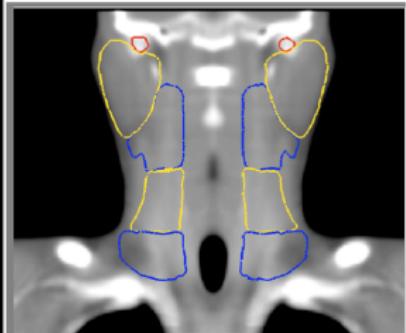
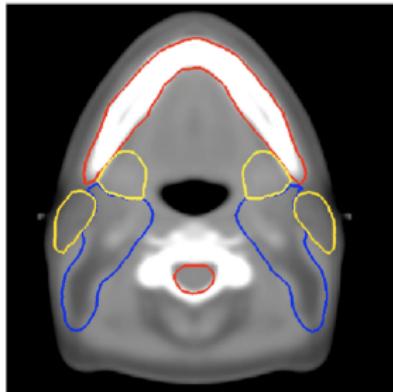
Segmentation par atlas, exemple : Tête et nuque

- Exemples de données



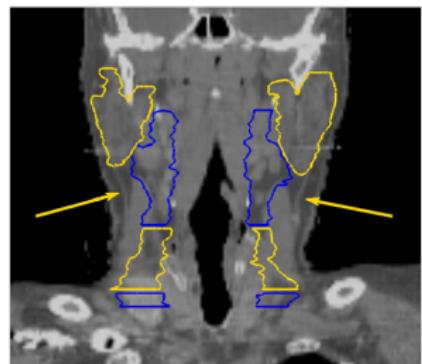
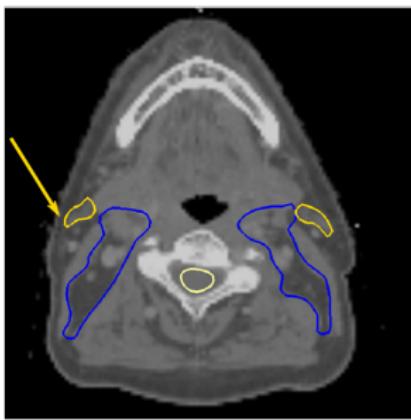
Segmentation par atlas, exemple : Tête et nuque

- Atlas obtenu



Segmentation par atlas, exemple : Tête et nuque

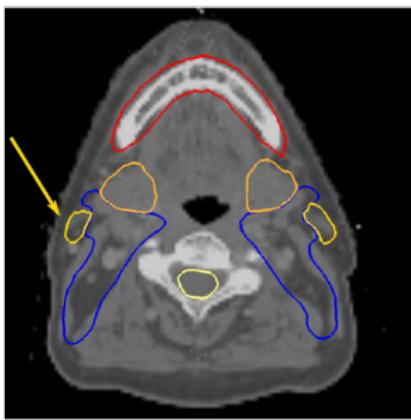
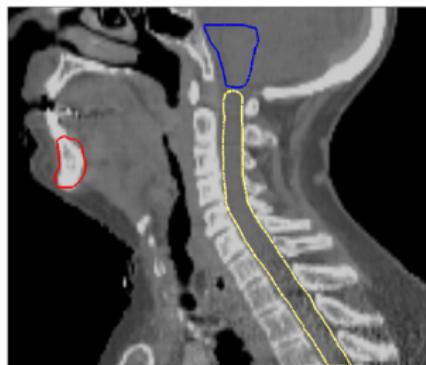
- Résultats (qualitatifs)



Segmentation manuelle

Segmentation par atlas, exemple : Tête et nuque

- Résultats (qualitatifs)



Segmentation par atlas