

Bases du traitement des images

► Segmentation des images ◀

Dominique Béréziat¹

Décembre 2016

1. D'après le cours de Mattieu Cord, merci à lui!

Introduction

► Définition :

- Partitionner l'image en zone homogènes selon un critère déterminé : couleur, texture, niveau de gris, ...
- 1 partition de l'image \mathcal{I} = ensemble de parties non vides R_i appelées régions, d'intersection vide, et d'union \mathcal{I}



Introduction

Contexte par rapport au cours de détection et caractérisation de primitives dans les images :

- ▶ Une région est une primitive
- ▶ Algorithme d'analyse basée régions :
 - ① Détection des régions \Rightarrow segmentation
 - ② Description des régions :
 - Caractéristiques de forme, de position, de taille
 - Caractéristiques radiométriques, ...

Introduction

► Angles d'attaques :

- Coté Optimisation (formalisme général) :
 - Minimisation globale sous contrainte entre l'image originale et l'image segmentée ⇒ **méthodes basées optimisation**
- Coté espace de description des pixels
 - Analyse d'histogrammes, quantification ⇒ **méthodes basées clustering**
- Coté géométrie de l'image : approches régions
 - Croissance, division ⇒ **méthodes basées régions**

► Difficultés :

- Problème de stabilité de la détection (répétabilité)
- Problème souvent mal posé, plein de solutions acceptables, très dépendant du contexte
- Souvent très calculatoire

Plan

Segmentation : un problème d'optimisation :

- ① **Méthodes variationnelles**
- ② Méthodes basées sur le *clustering*
- ③ Méthodes basées régions

Méthodes par optimisation

- ▶ Principe :
 - Minimiser une fonctionnelle mesurant l'écart entre l'image partitionnée et l'image originale sous contraintes
 - Contraintes liées à la forme de l'espace de solution (fonction régulière, constante par morceaux, ...)
- ▶ Il existe un très grand nombre de partitions d'une image
Comment choisir entre plusieurs partitions ?
- ▶ Le cardinal de la partition (à minimiser), la taille de la plus petite région (à maximiser), une distance entre régions (somme des distances entre zones adjacentes à maximiser)

Méthodes par optimisation : énergie

- ▶ Énergie pour chaque région R : somme pondérée de deux termes :

$$E(R) = E_I(R) + \lambda E_C(R)$$

- ▶ Énergie pour une image segmentée $\{R_i\}$:

$$E_\lambda(\{R_i\}) = \sum_{R \in R_i} E(R) + \lambda E_C(R)$$

- ▶ Énergie interne à la région (attache aux données) :
 - faible si image segmentée très proche de l'image initiale
- ▶ Énergie de complexité : faible si peu de régions, ...
- ▶ λ paramètre d'échelle

Méthodes par optimisation : énergies affines

$$E_{\lambda} = E_I + \lambda E_C$$

	Variationnel Mumford-Shah, 1989	Bayésien Geman-Geman, 1984
E_I	Attache aux données	Potentiel de vraisemblance
E_C	Énergie de régularisation	Potentiel a priori
E_{λ}	Énergie totale	Potentiel a posteriori (Gibbs)

Énergie interne : modèle de Mumford et Shah

- Modèle constant par morceaux : chaque région est représentée par une constante estimée comme une moyenne sur la région

- Notations :

- Soit une région R contenant les pixels x_1, \dots, x_N , chacun défini dans \mathbb{R}^3
- Soit m_1, m_2, m_3 la moyenne des trois canaux R,V,B sur R ,
 $m = (m_j)_{j=1,\dots,3}$
- $\| \cdot \|$ distance euclidienne
- Énergie interne = distance entre l'image et la moyenne sur la région :

$$E_I(R) = \sum_{i=1}^N \|X_i - m\|^2 = \sum_{i=1}^N \sum_{j=1}^3 (x_i^j - m^j)^2$$

- Calcul rapide :

- Soit V la matrice 3×3 de variance/covariance :
- $$v(i,j) = \sum_{k=1}^N (x_k^i - m^i)(x_k^j - m^j)$$
- λ_j valeurs propres de V :

$$E_I(R) = N \text{Tr}(V) = N \sum_{j=1}^3 \lambda_j$$

Autres énergies internes

- ▶ Autres énergies :
 - modèle gaussien :

$$E_I(R) = N \log(\det(V)) = N \sum_{j=1}^3 \log(\lambda_j)$$

$$E_I(R) = N \operatorname{Tr}(\sqrt{V}) = N \sum_{j=1}^3 \sqrt{\lambda_j}$$

- ▶ Énergie de complexité E_C
 - Exemple : longueur des contours de la région R

Méthodes par optimisation

Problème primal

$$\min D(I, x) \quad \text{s.c.} \quad C(x) \leq \epsilon$$

De deux modèles également complexes le plus fidèle est le meilleur

Problème dual

$$\min C(x) \quad \text{s.c.} \quad D(I, x) \leq \lambda$$

De deux modèles également fidèles le plus simple est le meilleur

Équivalence lagrangienne

$$E_\lambda(x) = \lambda C(x) + D(I, x) \iff E_\mu^*(x) = \mu D(I, x) + C(x)$$

Méthodes par optimisation

- Résultats de segmentation selon la valeur de λ :



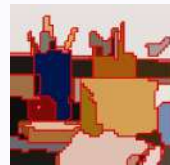
Image



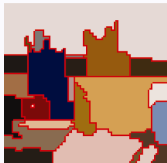
$\lambda = 0$



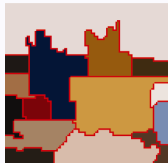
$\lambda = 0.1$



$\lambda = 0.2$



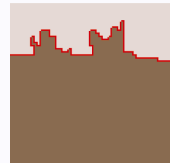
$\lambda = 0.5$



$\lambda = 0.8$



$\lambda = 0.9$



$\lambda = 1.2$

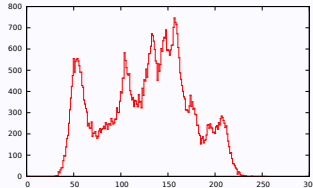
Plan

Segmentation : un problème d'optimisation :

- ① Méthodes variationnelles
- ② **Méthodes basées sur le *clustering***
- ③ Méthodes basées régions

Méthodes basées histogramme

- Exemple basique : seuillage binaire



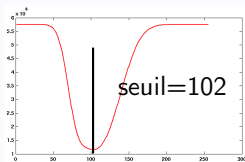
↑
seuil



$$f(x, y) = 0 \quad \text{si } I(x, y) < \text{seuil}$$
$$f(x, y) = 255 \quad \text{sinon}$$

Méthodes basées histogramme

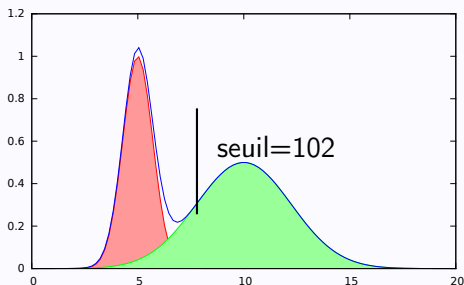
- ▶ Nombreuses méthodes basées sur l'analyse du(des) mode(s) de la distribution des pixels => plusieurs seuils
- ▶ Choix du(des) seuil(s) :
 - Minimisation de la somme des variances des deux classes



- ▶ Remarque : c'est une méthode par optimisation où l'on choisit la meilleure partition parmi 256 (au sens du critère de variance sur les partitions)

Méthodes basées histogramme

- Choix du seuil : seuillage avec apprentissage

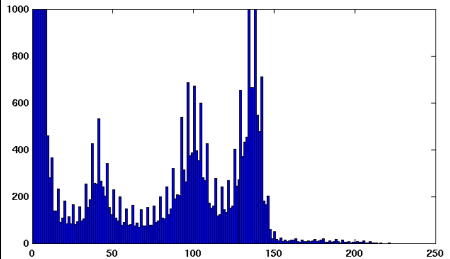
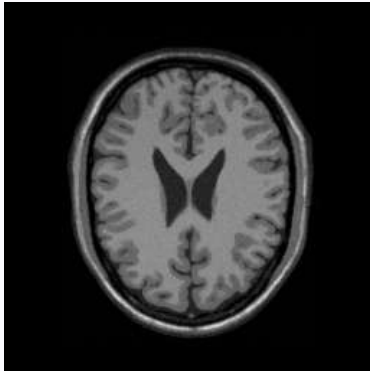


Méthodes basées histogramme

- ▶ Amélioration du seuillage avec post-traitement intégrant une analyse spatiale
- ▶ Principe :
 - ① Localisation sur un histogramme d'un mode isolé
 - ② Détection des zones correspondantes par seuillage
 - ③ Parmi les zones de l'image contribuant à ce mode, sélection de la région connexe la plus grande (par exemple par morphologie mathématique)
 - ④ Retour à 1

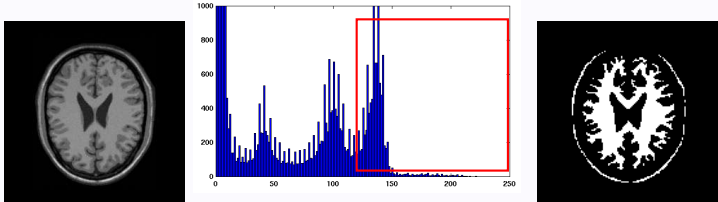
Méthodes basées histogramme

► Exemple

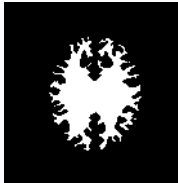


Méthodes basées histogramme

- Localisation sur l'histogramme d'un premier mode :

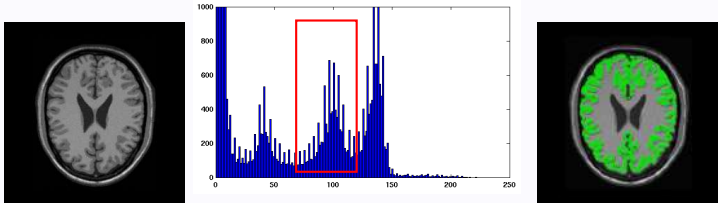


- Sélection de la région connexe la plus grande :



Méthodes basées histogramme

- Localisation sur l'histogramme d'un second mode :



- Sélection de la région connexe la plus grande

Méthodes basées histogramme

- Itération sur qu'au résultat final :

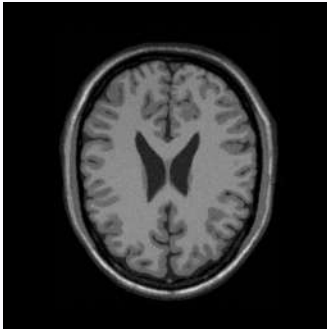


image originale

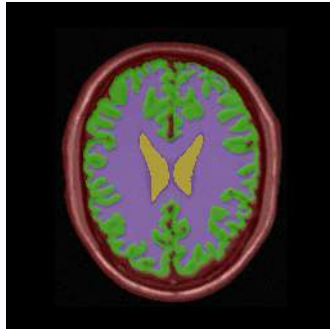


image segmentée

Méthodes basées histogramme

► Bilan :

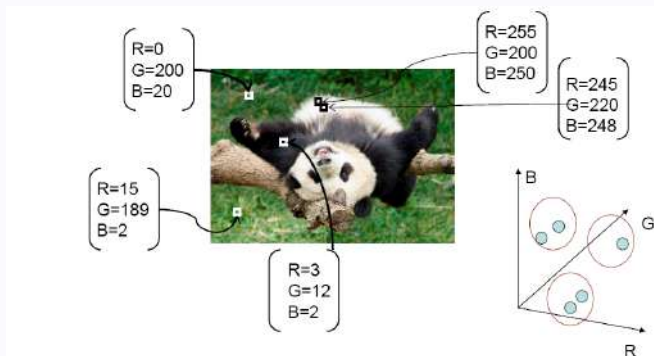
- Simples
- Performances limitées, efficaces si objets sur fond uniforme
- Pas (peu) d'utilisation de l'information spatiale

► Extension si le descripteur du pixel est un vecteur et non plus un scalaire (niveau de gris)

⇒ De l'histogramme au clustering

Méthodes basées clustering

- Clustering multidimensionnel :
 - Partition des vecteurs en groupes homogènes



La méthode des k -moyennes (k -Means)

- L'algorithme k -means est en 4 étapes :
 - 1 Choisir k objets formant ainsi k clusters
 - 2 (Ré)attribuer chaque objet O au cluster C_i de centre M_i tel que $\text{dist}(O, M_i)$ est minimal
 - 3 Recalculer M_i de chaque cluster (le barycentre)
 - 4 Aller à l'étape 2 si on vient de faire une affectation

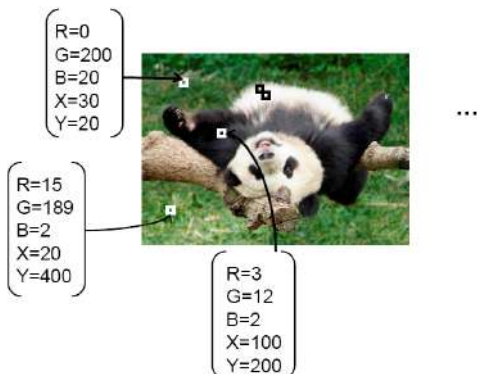
Segmentation basée clustering

- K-means clustering
 - Pas de cohérence spatiale



Segmentation basée clustering

- Regrouper (*to cluster*) des pixels ayants de caractéristiques (*features*) similaires



Segmentation basée clustering

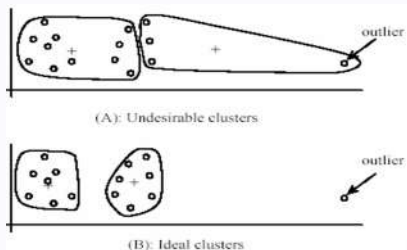
K-means :

► Avantages :

- Simplicité
- Convergence (local min)

► Inconvénients :

- Coûteuse en espace mémoire
- Choisir un K
- Sensible à l'initialisation
- Sensible aux artéfacts
- Limité aux clusters "sphériques"



Algorithme du *Mean shift*

- ▶ Le mean shift recherche des modes (maxima locaux) de la loi de distribution observée dans l'espace des caractéristiques (*feature space*)

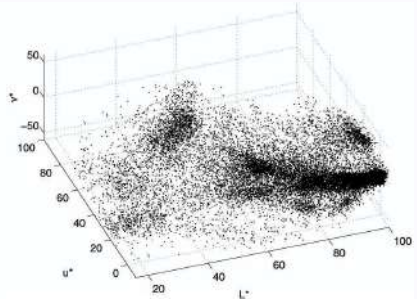


Image (gauche) et espace des caractéristiques (codage couleurs $L^*u^*v^*$, droite)

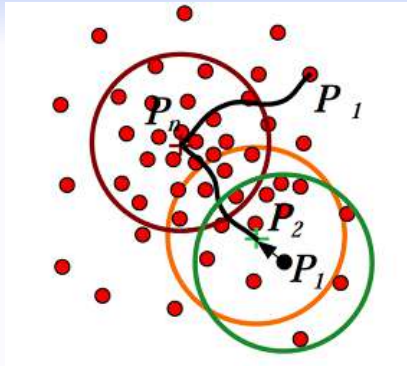
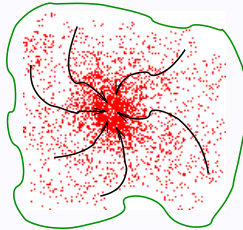


Figure – Principle of mean shift analysis : To find the cluster center for point P_1 , repeatedly find the centroid of points inside a sphere (initially at P_1) and recenter the sphere on the centroid, until the sphere is stationary. (For Gaussian-kernel mean shift analysis, points further from sphere centers are given exponentially decreasing weights in the centroid calculation.) This is an adaptive gradient ascent in the space of point densities.

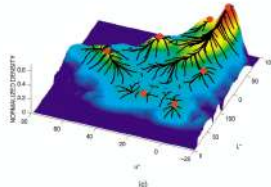
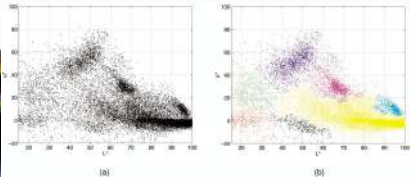
Mean shift clustering

- ▶ Cluster : all data points in the attraction basin of a mode
- ▶ Attraction basin : the region for which all trajectories lead to the same mode



Mean shift clustering/segmentation

- ▶ Find features (color, gradients, texture, *etc*)
- ▶ Initialize windows at individual feature points
- ▶ Perform mean shift for each window until convergence
- ▶ Merge windows that end up near the same “peak” or mode



Mean shift segmentation results



Mean shift segmentation results



Mean shift pros and cons

► Pros

- Does not assume spherical clusters
- Just a single parameter (window size)
- Finds variable number of modes
- Robust to outliers

► Cons

- Output depends on window size
- Computationally expensive
- Does not scale well with dimension of feature space

Plan

Segmentation : un problème d'optimisation :

- ① Méthodes variationnelles
- ② Méthodes basées sur le *clustering*
- ③ **Méthodes basées régions**

Segmentation basées régions

- ▶ Méthodes par transformation de régions
- ▶ 2 grandes approches :
 - ① Fusion, croissance de régions (du pixel vers l'image entière)
 - Agglomération de pixels voisins à une région selon un critère d'homogénéité donné
 - Exemple de critère courant : proximité colorimétrique du pixel candidat à la couleur du voisin ou de la couleur moyenne calculée sur la région
 - ② Division, partage de régions (de l'image entière vers le pixel)

Segmentation basées régions

► La croissance de régions – Principe

- ① Initialisation d'une région R_0 à un pixel ou un groupe de pixels (germes) :
 - peut être choisi manuellement,
 - ou de manière automatique en évitant les zones de fort contraste (gradient important)
- ② Ajout à R_0 de tous les pixels voisins satisfaisant le critère choisi d'homogénéité
- ③ Itération jusqu'à convergence

Segmentation basées régions

- Exemples de l'approche par croissance de régions



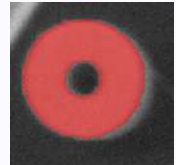
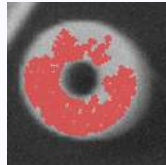
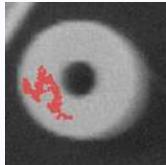
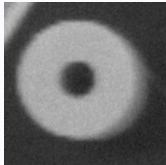
amorce



croissance

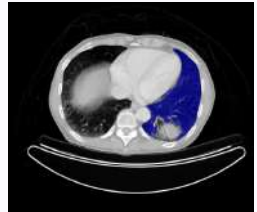
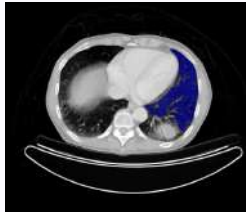
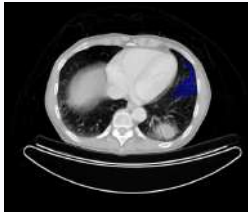
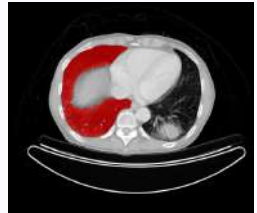
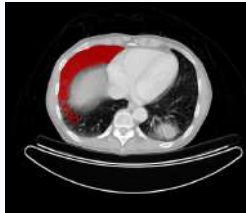
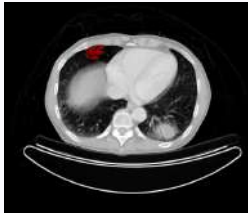


région finale



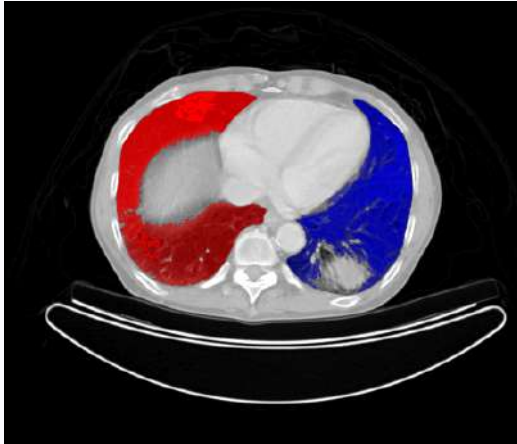
Segmentation basées régions

- Exemples de l'approche par croissance de régions



Segmentation basées régions

- Résultat (avec post-traitement)



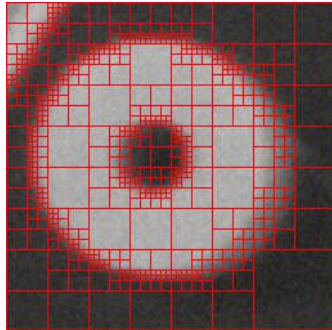
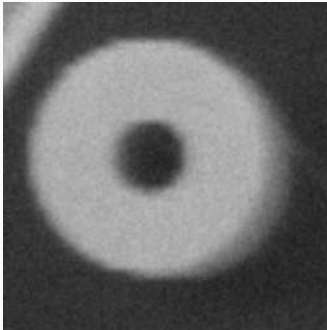
Segmentation basées régions

- ▶ Bilan sur la croissance de régions
 - Performance très dépendante de l'initialisation (germes)
 - Dépend souvent de l'ordre de traitement : L'ordre dans lequel sont ajoutés les pixels dans une région a une influence sur le résultat
 - Implémentation relativement simple
 - Méthodes rapides

Segmentation basées régions

- Le partage de régions

Le but de ces méthodes est de diviser, partager de façon automatique une image en un ensemble de régions homogènes selon le critère choisi (le même qu'avant)



Segmentation basées régions

► Le partage de régions – Principe

- ① Initialisation : $R = I$ l'image entière
- ② Application de plusieurs divisions δ produisant de nouvelles régions R_i^δ
- ③ Choix de la subdivision selon le processus suivant : pour chaque ensemble $\{R_i^\delta\}$ on compte le nombre de sous-régions homogènes, et on sélectionne celui qui en fournit le plus
- ④ Retour à 2 pour chaque sous-région non homogène

Segmentation basées régions

- Le partage de régions
- Exemple de partage de régions à deux partitions

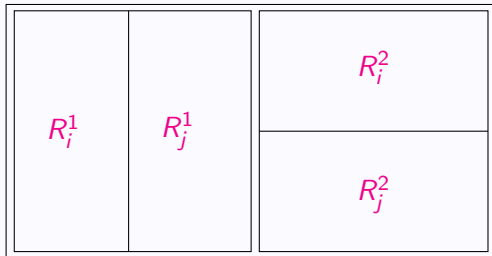


Image originale

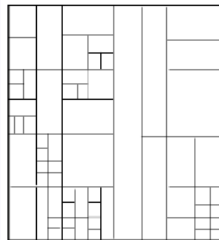


Image partitionnée

Segmentation basées régions

- ▶ Le partage de régions :
 - Exemple du Quad-tree
 - Une seule façon de diviser : en 4 régions carrées

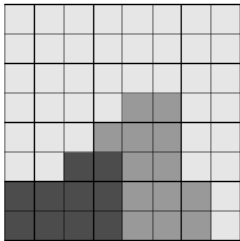
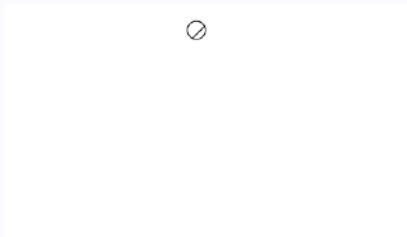


Image originale



Construction du Quad-tree

Segmentation basées régions

- ▶ Le partage de régions :
 - Exemple du Quad-tree
 - Une seule façon de diviser : en 4 régions carrées
 - Récursivement, chaque feuille F subdivisée en quatre si F pas assez homogène, et 4 sous images ajoutées en tant que feuilles de F

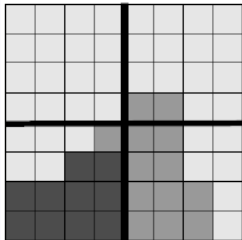
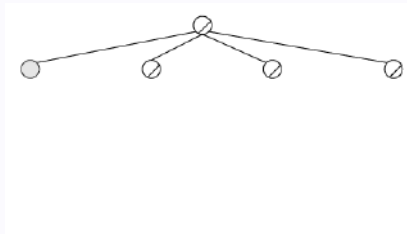


Image originale



Construction du Quad-tree

Segmentation basées régions

- ▶ Le partage de régions :
 - Exemple du Quad-tree
 - Une seule façon de diviser : en 4 régions carrées

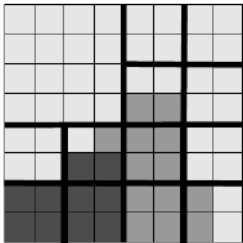
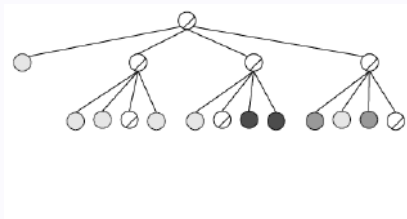


Image originale



Construction du Quad-tree

Segmentation basées régions

- ▶ Le partage de régions :
 - Exemple du Quad-tree
 - Une seule façon de diviser : en 4 régions carrées
 - L'algorithme se poursuit tant qu'il reste des feuilles non homogènes à diviser.

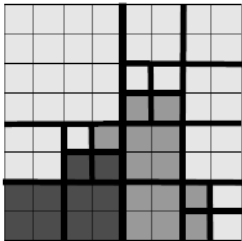
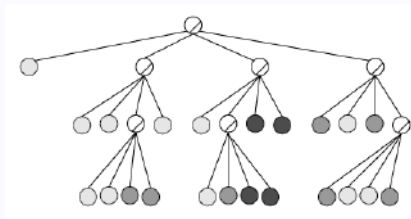


Image originale



Construction du Quad-tree

Segmentation basées régions

► Le partage de régions – Propriétés

- La géométrie du découpage a une influence directe sur le résultat de la segmentation
- Par exemple la méthode quad-tree fait apparaître des régions carrées
- Il existe d'autres types de partage (triangle, pyramide)
- Le choix du type de partage se fait en fonction des formes que l'on souhaite segmenter

Segmentation basées régions

- ▶ Méthodes hybrides
- ▶ Comment combiner la division/fusion en segmentation ?
- ▶ Idée : appliquer un algorithme de fusion non pas sur les pixels mais sur des régions homogènes détectées par un algorithme de division
- ▶ Exemple : algorithme SPLIT/MERGE

Segmentation basées régions

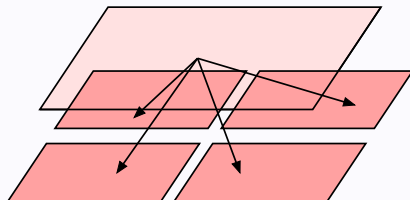
- ▶ Algorithme : à partir d'une segmentation initiale,
 - ① Définition d'un graphe d'adjacence où une région est un nœud et un arc une relation d'adjacence
 - ② Définition d'une fonction de similarité entre deux nœuds
 - ③ On trie tous les couples de nœuds adjacents dans une liste ordonnée
 - ④ On regroupe les deux meilleurs candidats
 - ⑤ On remet à jour la liste et on itère (retour en 3)

Segmentation basées régions (Split & merge)

- Split & Merge (Horowitz, Pavlidis (1976), Dubuisson, Jain (1993)) :
 - Phase 1 : Créer les zones homogènes = SPLIT
 - Phase 2 : Les regrouper = MERGE

Algorithme SPLIT : un
Quad-Tree

Itération de la subdivision
d'une feuille en 4



Segmentation basées régions (Split & merge)

Découpage sur homogénéité (critère sur la variance)

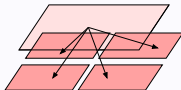
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

image initiale



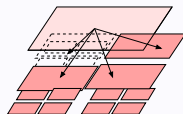
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 1



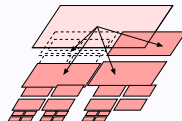
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 2



0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 3

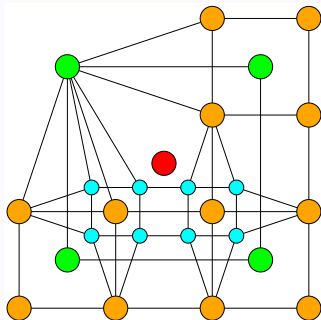
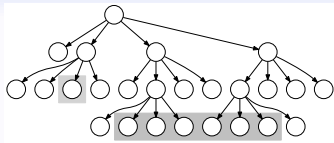
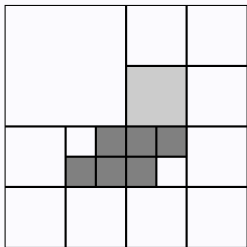


Segmentation basées régions (Split & merge)

- Construction du *Region Adjacency Graph*

Connecte les régions adjacentes

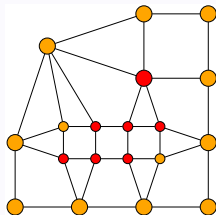
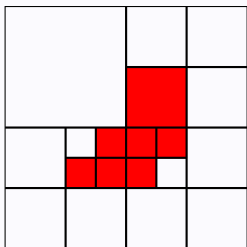
Arêtes = mesures de différence d'homogénéité



Segmentation basées régions (Split & merge)

► MERGE :

- Chaque nœud du *Region Adjacency Graph* est examiné.
- Si un des voisins de ce nœud est à une distance inférieure à un seuil de regroupement, les deux nœuds fusionnent dans le RAG.
- Lorsque plus aucun nœud ne peut fusionner avec l'un de ses voisins, STOP.



La distance en terme d'homogénéité de région est portée par l'arête évaluée qui les relie dans le RAG

Segmentation basées régions (Split & merge)



image originale

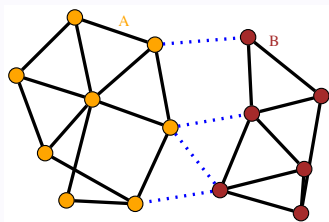


split & merge

Segmentation basées régions

- ▶ Bilan de la fusion du graphe
 - En général, il est préférable de regrouper les petites régions en premier
 - L'ordre dans lequel s'effectue le regroupement des régions a une influence sur le résultat
- ▶ Autre approche : partitionner le graphe récursivement

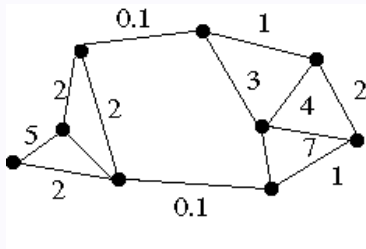
Algorithme du Graph cut



- ▶ Trouver un ensemble d'arêtes qui permettent de couper le graphe en deux
- ▶ Coût : somme des poids des arêtes coupées
- ▶ Une coupe fournit une segmentation
- ▶ Qu'est ce qu'une bonne coupe, comment la trouver ?

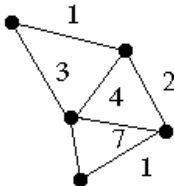
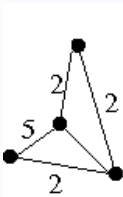
Minimum cut

- Segmentation = choisir le minimum cut dans le graphe
 - Il existe des algorithmes efficaces



Minimum cut

- Segmentation = choisir le minimum cut dans le graphe
 - Il existe des algorithmes efficaces
 - Itération du processus

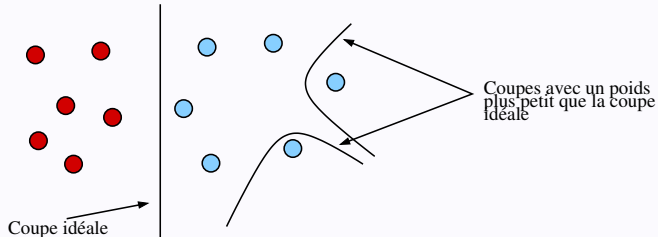


Fonction de coût

- ▶ $G = (V, E)$
- ▶ W matrice des poids entre les nœuds :

$$W(A, B) = \sum_{u \in A, v \in B} W(u, v)$$

- ▶ Problème : le *minimum cut* tend à sélectionner des coupes avec des éléments de petite taille :



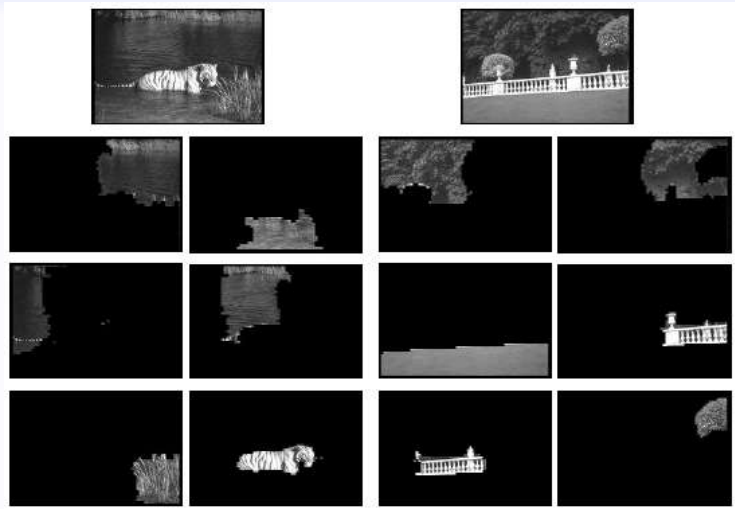
Normalisation du coût

- Pour corriger le problème précédent, on considère un coût normalisé N_W :

$$N_W(A, B) = \frac{W(A, B)}{W(A, V)} + \frac{W(A, B)}{W(B, V)}$$

- Solution numérique utilisant une diagonalisation matricielle.
- J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000

Exemples



Annexe

Complément sur les méthodes par optimisation

Segmentation Markovienne

Notations

- ▶ Approche discrète et probabiliste de la segmentation.
- ▶ Une image X est la fonction $X : s \in S \mapsto E$ où S et E sont deux ensembles discret (et finis).
- ▶ s un site (ou pixel), S l'ensemble des sites (domaine de l'image), on note X_s la valeur de X au site s . On a $X \in E^{|S|} = \Omega$.
- ▶ X est vu comme un processus stochastique. On note :
 - $P(X = x)$ la probabilité que l'image X soit la réalisation x ($|E|^{|S|}$ possibilités !)
Ex : soit l'ensemble des images binaires 2×2 : on a 2^4 images possibles.
 - $P_s(X_s = a)$ la probabilité que la valeur au site s de l'image X soit a . Cette loi dépend *a priori* du site s , c'est pourquoi P est indicée par s .

Champs de Markov

- ▶ Ce qui change par rapport aux approches de type *clustering* : la loi pour un pixel dépend du pixel lui-même.
- ▶ Précisément : la loi pour un pixel s dépend des valeurs des pixels situés dans le voisinage de s .
- ▶ Champs de Markov :
$$P_s(X_s = x_s | X_t = x_t, t \neq s) = P_s(X_s = x_s | X_t = x_t, t \in \mathcal{V}_s)$$
- ▶ Champs de Gibbs : $P(X = x) = \frac{1}{Z} e^{-U(x)}$ avec $Z = \sum_{x \in \Omega} e^{-U(x)}$
- ▶ Théorème (admis) : Hammersley, Clifford.
Il y a équivalence entre les champs de Markov et les champs de Gibbs. De plus, on a :

$$P_s(X_s = x_s | X_p = x_p, p \neq s) = \frac{e^{-U_s(x_s | x_p, p \in \mathcal{V}_s)}}{\sum_{a \in E} e^{-U(a | x_p \in \mathcal{V}_s)}}$$

Optimisation

- ▶ Le problème de segmentation est modélisé via l'énergie U :
 - la solution recherchée doit correspondre à la configuration de X la plus probable (elle maximise P)
 - et donc est elle correspond à l'énergie U la plus basse.
- ▶ On suppose U connu, comment calculer $\operatorname{argmax}_{x \in E|S|}(P(X = x))$?
force brute ? Ω trop grand !
- ▶ On dispose d'un procédé itératif qui va converger vers la configuration la plus probable : le recuit simulé.

Recuit simulé

- ▶ Théorème : soit P un champ de Gibbs. Il existe une suite $T_n > \log n$ décroissante telle que $X^n \xrightarrow[n \rightarrow \infty]{\text{Loi}} P(X|U = U_{\min})$ où X^n est une réalisation de $P^n(X) = \frac{1}{Z_n} e^{-\frac{U(X)}{T_n}}$.
- ▶ Algorithme (étape n) :
 - on tire un site s (loi uniforme)
 - on tire une valeur a selon la $P_s(X_s) = \frac{1}{Z_s} e^{-U_s(X_s|X_p, p \in \mathcal{V}_s)}$
 - si $U_s(a|\dots) < U_s(x_s|\dots)$ alors on accepte la valeur a
 - sinon : on accepte la valeur a avec probabilité
$$\Delta P_s = \frac{1}{Z_s} e^{\frac{-U_s(a|\dots) + U_s(x_s|\dots)}{T_n}}$$
 - on met à jour le site s avec la valeur a (si elle est acceptée) :
 $X_s^{n+1} = a, X_p^{n+1} = X_p^n, p \neq s$
- ▶ Il existe un algorithme plus rapide et déterministe (ICM) mais il ne calcule que des minima locaux :
Chaque site s est mis à jour avec la valeur la plus probable selon P_s .

Application à la segmentation

- ▶ Exemple 1 : une image binaire (issue d'un seuillage par exemple)
 - $E = \{-1, 1\}$
 - système de voisinage : 4 ou 8 plus proches voisins (sur la grille des pixels)
 - $U(X = x) = -\alpha \sum_{s \in S} x_s - \beta \sum_{\langle s, t \rangle} x_s x_t$
 - $\langle s, t \rangle$ notation pour désigner une paire de sites voisins.
 - $\alpha > 0$: favorise les valeurs 1 au détriment des valeurs -1.
 $\alpha < 0$: le contraire.
 - $\beta > 0$: favorise les régions connexes et régulières : le produit $x_s x_t$ est positif si les deux sites ont même valeur.
 - $U_s(X_s = x_s | X_p = x_p, p \neq s) = -\alpha x_s - \beta x_s \sum_{t \in \mathcal{V}_s} x_t$

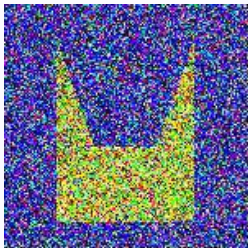
Application à la segmentation

- ▶ Exemple 2 : une image multi-label (issue d'une classification par exemple)
 - $E = \{a_1, \dots, a_k\}$
 - système de voisinage : 4 ou 8 plus proches voisins
 - $U(X = x) = \alpha \sum_{s \in S} (x_s - \mu_s)^2 + \beta \sum_{\langle s, t \rangle} (x_s - x_t)^2$
 - μ_s est la valeur initiale, celle attribuée par la classification.
 - α : attache aux données. On force x_s à rester proche de sa valeur initiale.
 - β : régularisation. On force les sites voisins à avoir une étiquette proche.
 - $U_s(X_s = x_s | X_p = x_p, p \neq s) = \alpha (x_s - \mu_s)^2 + \beta \sum_{t \in \mathcal{V}_s} (x_s - x_t)^2$

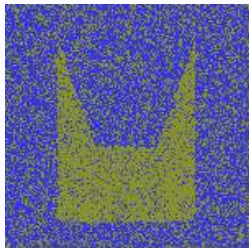
Application à la segmentation



Image originale

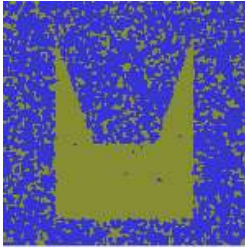


bruitée

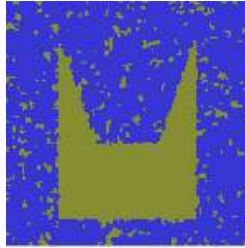


Kmean

Application à la segmentation



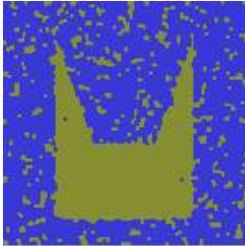
ICM



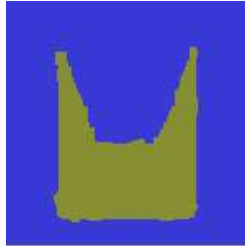
Recuit simulé

$$\beta = 0.3$$

Application à la segmentation



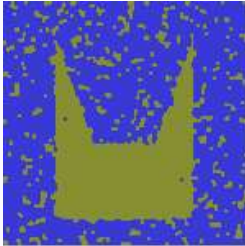
ICM



Recuit simulé

$$\beta = 1$$

Application à la segmentation



ICM



Recuit simulé

$$\beta = 1000$$

Approche variationnelle

- ▶ Formalisation continue : I est une fonction (intégrable, dérivable, ...) et on cherche une segmentation, elle-même une fonction.
- ▶ L'espace de recherche est celui des fonctions intégrables : il est de dimension infini.
- ▶ Fonctionnelle de Shah-Mumford, Λ : frontière entre les régions, Ω/Λ = domaine image sans Λ .

$$E(S) = \underbrace{\int_{\Omega/\Lambda} (S(x) - I(x))^2 dx}_{\text{attache aux données}} + \alpha \underbrace{\int_{\Omega/\Lambda} (\nabla S)^2 dx}_{\text{régul.}} + \beta \underbrace{\int_{\Lambda} ds}_{\text{MDL}}$$

- ▶ E est une **fonctionnelle** définie sur l'ensemble des fonctions intégrables, elle renvoie un scalaire positif.
- ▶ Pour trouver le minimum théorique de E : on doit calculer le gradient = dérivée directionnelle (infinité de direction), puis utiliser une méthode de descente de gradient.

Approche variationnelle

- Dérivée de Gâteau :

$$\frac{\partial E}{\partial S}.f = \lim_{\lambda \rightarrow 0} \frac{E(S + \lambda f) - E(S)}{\lambda}$$

$\frac{\partial E}{\partial S}$ est un opérateur différentiel, f est une direction (une fonction)

- En général, le calcul du gradient de E mène à une équation aux dérivées partielles plus ou moins complexes.
- Ces équations doivent être discrétisés : geste technique qui dépasse largement le cadre de ce cours.
- Fonctionnelle de Shah : pas de solution pour S général : on suppose S constant par morceaux et E devient :

$$E(\{S_i\}) = \sum_i \int (S_i - I(x))^2 dx + \beta |\Lambda|$$

- Dans ce cas : on se ramène naturellement à un problème discret.

Remerciements divers pour les illustrations empruntés dans ce cours :

Cord, Thome, Lovan, Valle, Leibe, Grauman, Lowe, Bernard, Baillie, Verbeek, Schmid, Wu, Guigue, Foliguet, étudiants de l'ENST.