

SEGMENTATION



TSI/I2SC

Marc Donias

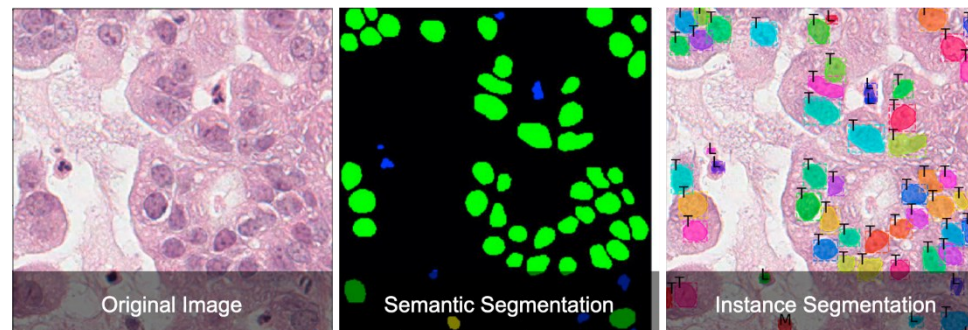
Problématique

► Notion de segmentation

- Littérature : « division en segments », fractionnement, fragmentation
- Economie : « classification d'individus en groupes homogènes »
- Embryologie, biologie, etc.

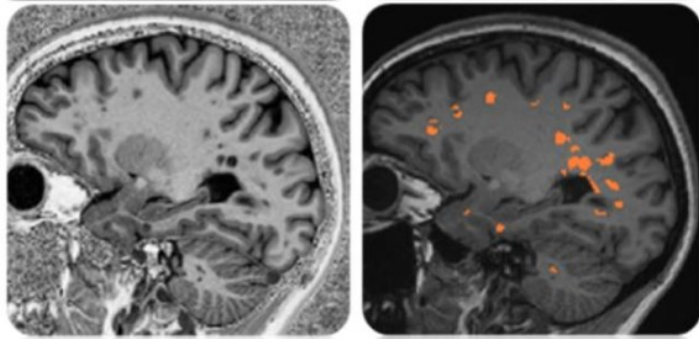
► Traitement des images numériques

- Partitionnement en groupes de pixels homogènes
 - Selon un critère (intensité, couleur, texture, etc.)
 - Sous la forme d'entités de différente dimension (régions, courbes, points)
 - De manière complète ou incomplète
- Sémantique vs Instance



Quelques applications

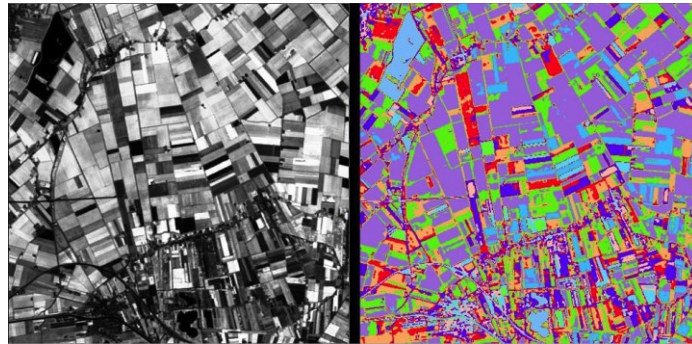
Souvent en prétraitement



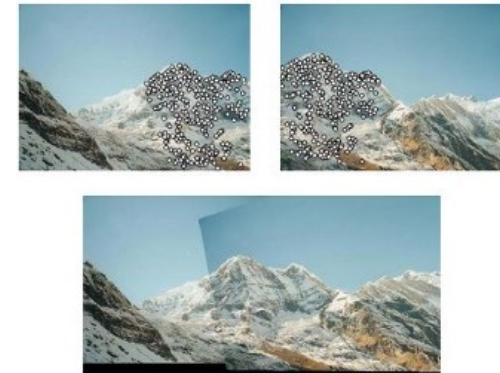
Instrumentation
Estimation de la taille de lésions



Composition d'images
Présentation météo

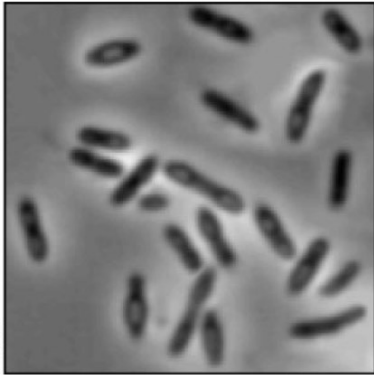


Classification/Reconnaissance de formes
(région agricole, Projet Ariana (INRIA))



Rectification/Fusion
Construction d'un panorama

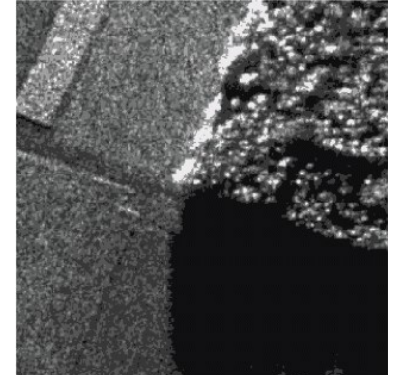
Homogénéité ?



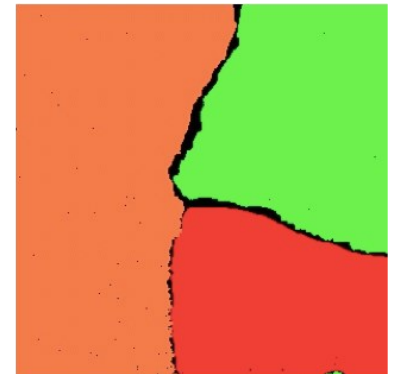
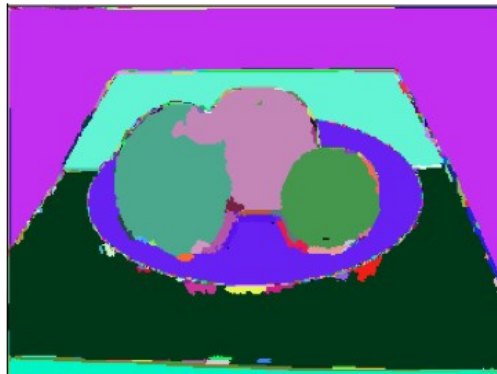
D'intensité scalaire



De couleurs

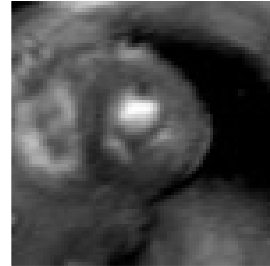
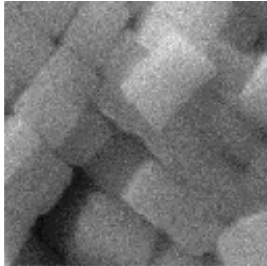


De texture



Un problème complexe

- Cas nécessitant des connaissances a priori de haut niveau

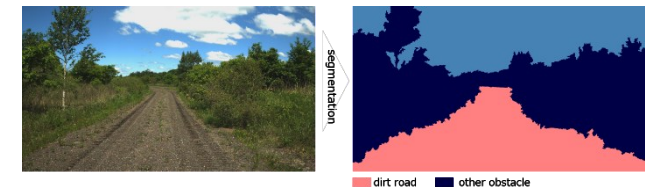


- ## ► Un domaine de « challenges »/ compétitions

- Calendrier, protocole de soumission, résultats, prix
- Base de données de grande dimensions, triées, avec annotations
- Générales ou thématiques



COCO 2020



Off road image segmentation challenge 2021

Un début de formalisation

► Partition selon Horowitz (1975)

$$1) \quad I = \bigcup_{i=1}^N S_i \quad \text{Partition}$$

$$2) \quad \forall i \in 1 \dots N, S_i \text{ est connexe et } P(S_i) = \text{Vrai}$$

$$3) \quad \forall i, j \in 1 \dots N, S_i \text{ adjacent à } S_j \text{ et } i \neq j \Rightarrow P(S_i \cup S_j) = \text{Faux}$$

► Exemples de prédicats P

$$P(R) = \text{Vrai} \Leftrightarrow \sigma_R < 5 \qquad P(R) = \text{Vrai} \Leftrightarrow \forall p \in R, |I(p) - \mu_R| < 10$$

► Unicité, stabilité, calculabilité ?

Approches

- ▶ Par Seuillage/Classification
- ▶ Basées « régions »
 - Division (Quad-tree, GraphCut, etc.)
 - Croissance (LPE, Fusion, etc.)
 - Migration (Champs de Markov, Superpixels, etc.)
- ▶ Basées « frontières »
 - Extraction de contours
 - Contours déformables
 - Surface de niveaux
- ▶ Basées « points remarquables » (SIFT, SURF, BRIEF, ORB)
- ▶ Par apprentissage profond (U-Net, Fast-FCN, Mask R-CNN)

+hybridation

Segmentation

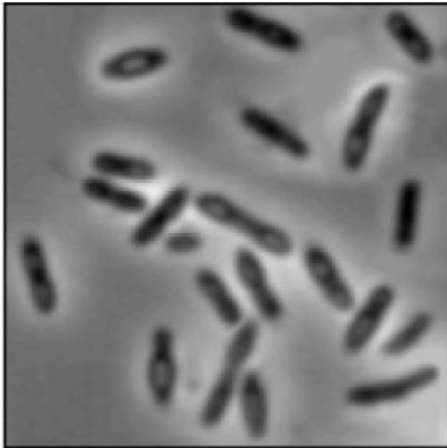


Approches « Seuillage/Classification »

Principe et définition

► Segmentation en deux classes/catégories

- Avant-plan (objets, structures, entités d'intérêt)
- Arrière-plan (fond)



Binarisation
(segmentation binaire)

► Test selon une valeur « seuil »

$$I_s = \begin{cases} 1 & \text{si } I_e \geq s \\ 0 & \text{sinon} \end{cases}$$

Choix du seuil

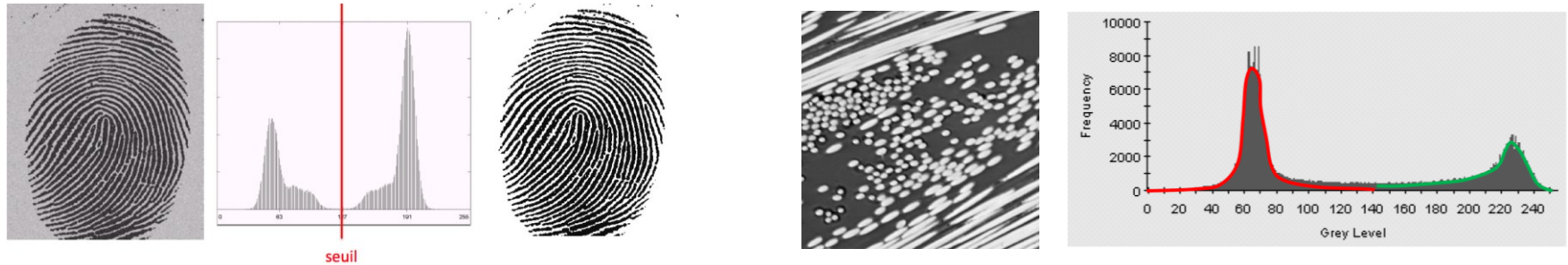
- ▶ Autant de segmentations que de seuils



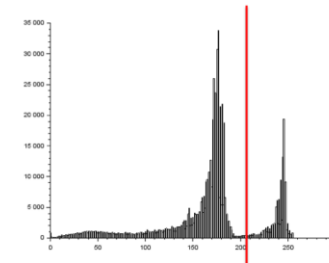
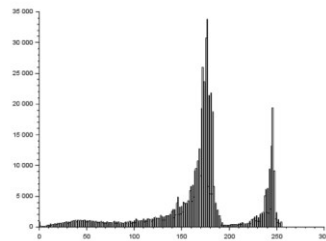
- ▶ Détermination par analyse (globale) de l'histogramme
 - Supervisée (observation et décision)
 - Automatique (algorithme de calcul)

Histogramme bi-modal

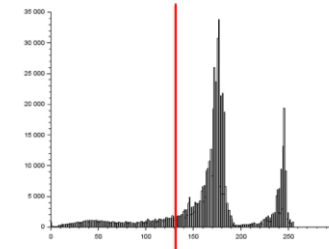
► Cas simple



► Cas peu évident



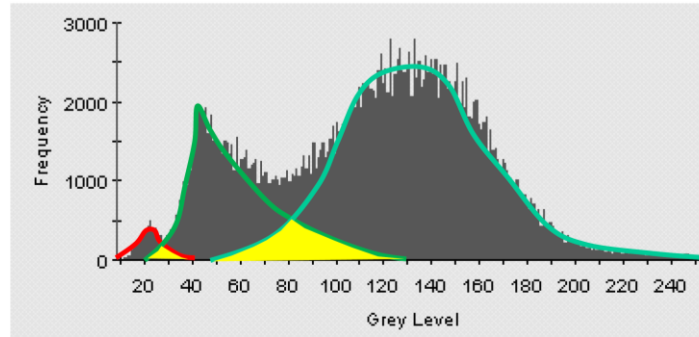
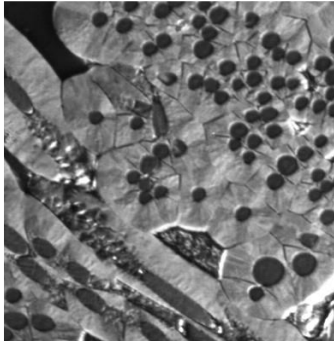
seuil
intermédiaire



seuil bas

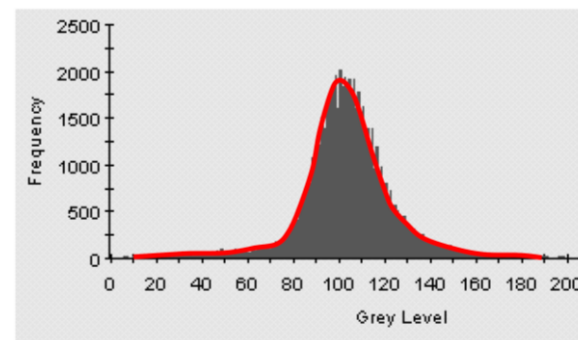
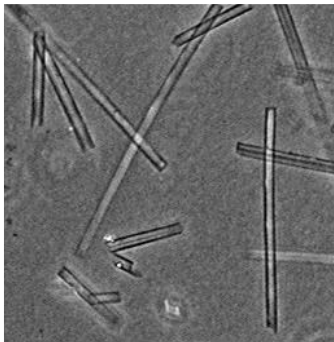
Autres histogrammes

► Cas difficile



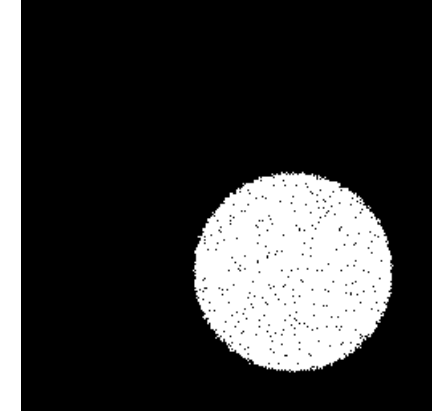
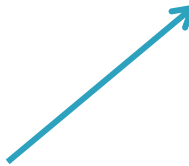
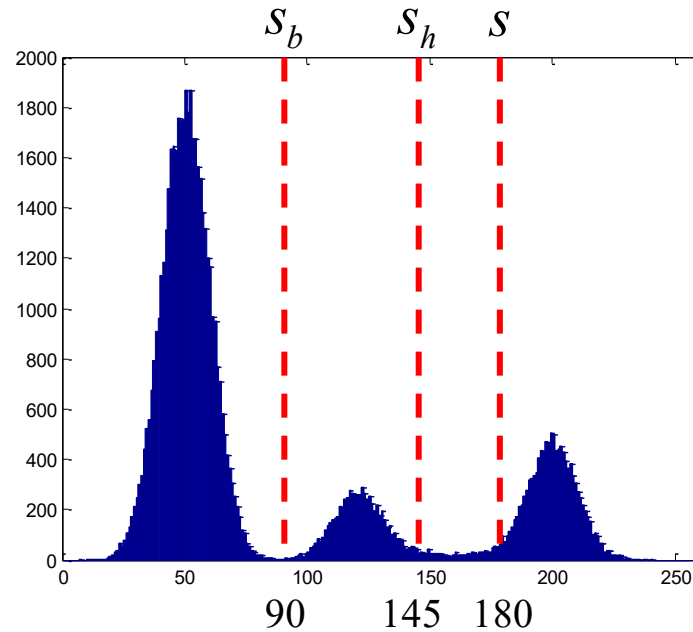
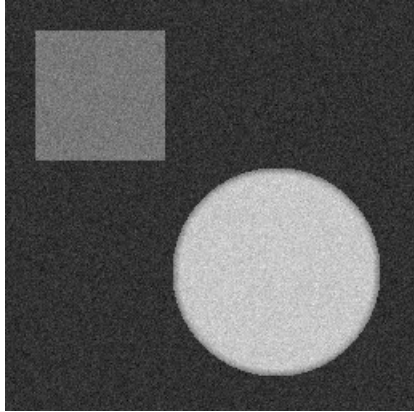
3 pics/modes
+
fort recouvrement

► Cas impossible

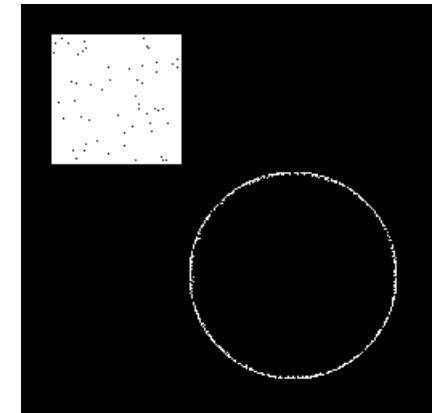
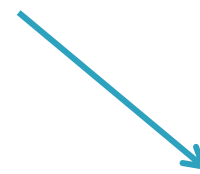


1 seul pic/mode

Seuils multiples



Seuillage à simple seuil

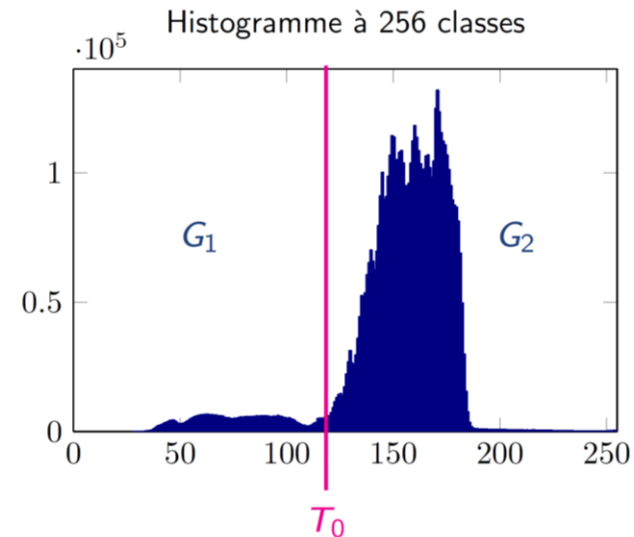


Seuillage à double seuil

$$I_s = \begin{cases} 1 & \text{si } s_b \leq I_e \leq s_h \\ 0 & \text{sinon} \end{cases}$$

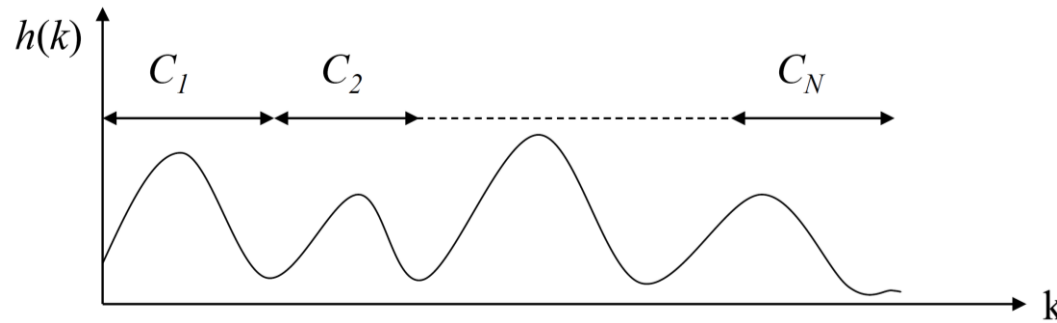
Un algorithme automatique « naïf »

1. Calcul de l'histogramme
2. Choix d'un seuil initial T_0
3. Calcul des intensités moyennes m_1 et m_2 des groupes G_1 et G_2
4. Calculer d'un nouveau seuil $T = (m_1 + m_2) / 2$
5. Itérer 3. jusqu'à ce que la variation de T soit inférieure à un seuil ε



Algorithme de Fisher

- Trouver la meilleure partition $P = \{C_1, C_2, \dots, C_N\}$



- Minimisant l'inertie

$$W(P) = \sum_{i=1}^N \sum_{k \in C_i} h(k) (k - G(C_i))^2 \quad \text{avec} \quad G(C_i) = \frac{\sum_{k \in C_i} k \times h(k)}{\sum_{k \in C_i} h(k)}$$

Version accélérée : algorithme d'OTSU

Bilan

► Avantages

- Simple
- Faible complexité calculatoire

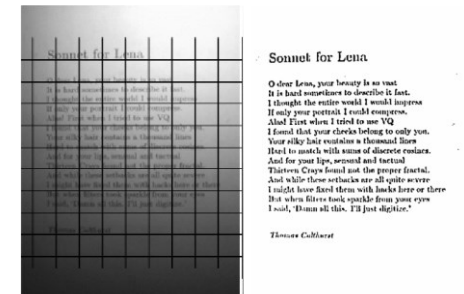
► Inconvénients

- Limité au cas scalaire ———> Extension au cas multidimensionnel par des approches de classification/clustering (K-means, Mean Shift)
- Démarche globale sans prise en compte spatiale

Approches locales



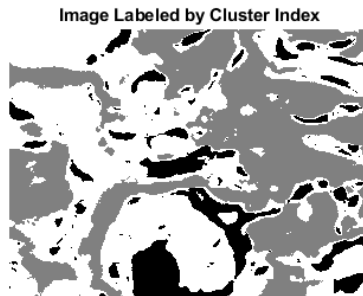
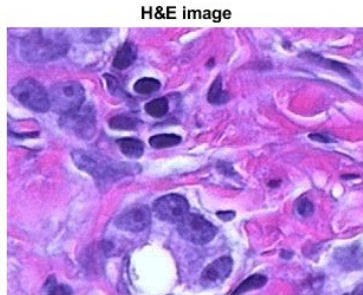
Approches dynamiques



K-means

Clustering scalaire ou multidimensionnel

1. Choisir K individus « centres »
2. (Re)Affecter chaque individu au centre le plus proche
3. Calculer les nouveaux centres
4. Itérer 2. jusqu'à convergence



- ▶ Avantages
 - Simple
 - Convergence (locale)
- ▶ Inconvénients
 - Nombre de classes a priori
 - Sensible à l'initialisation et aux artefacts
 - Limité au « clustering sphérique »

De la binarisation à l'entité

► Problématique



Binarisation



Etiquetage/labellisation

- Etiquetage en composantes connexes (ECC)
 - Approche classique séquentielle en 2 étapes, Rosenfeld (1966)
 - Approches optimale séquentielle en 3 étapes, Light Speed Labeling (LSL), Lacassagne (2011)
 - Approches parallèles

Approche ECC Rosenfeld (1 / 3)

- ▶ Initialisation par l'étiquette « nulle »
- ▶ Première attribution selon un parcours lexicographique
 - Extraction des étiquettes dans le voisinage « passé » d'un pixel « ON »

e_1	e_2	e_3
e_4	ON	

- Attribution d'une étiquette selon le nombre N d'étiquettes non nulles
 - Si $N = 0$, Nouvelle étiquette E
 - Si $N = 1$, Recopie de l'étiquette unique
 - si $N > 1$, Détermination d'une étiquette « fusionnée »
 - ✓ Détermination des étiquettes terminales

0	1	2	3	4	5	6	7	8	Etiquette
0	1	1	3	4	2	6	5	8	Etiquette équivalente

Etiquette terminale



Approche ECC Rosenfeld (2 / 3)

- ✓ Attribution de l'étiquette terminale minimale
 - ✓ Mise à jour des étiquettes terminales par l'étiquette terminale minimale
- Résolution de la table d'équivalence par un parcours dans le sens « croissant » puis suppression des étiquettes « inutiles »

0	1	2	3	4	5	6	7	8
0	1	1	3	4	2	6	5	8

0	1	2	3	4	5	6	7	8
0	1	1	3	4	1	6	1	8

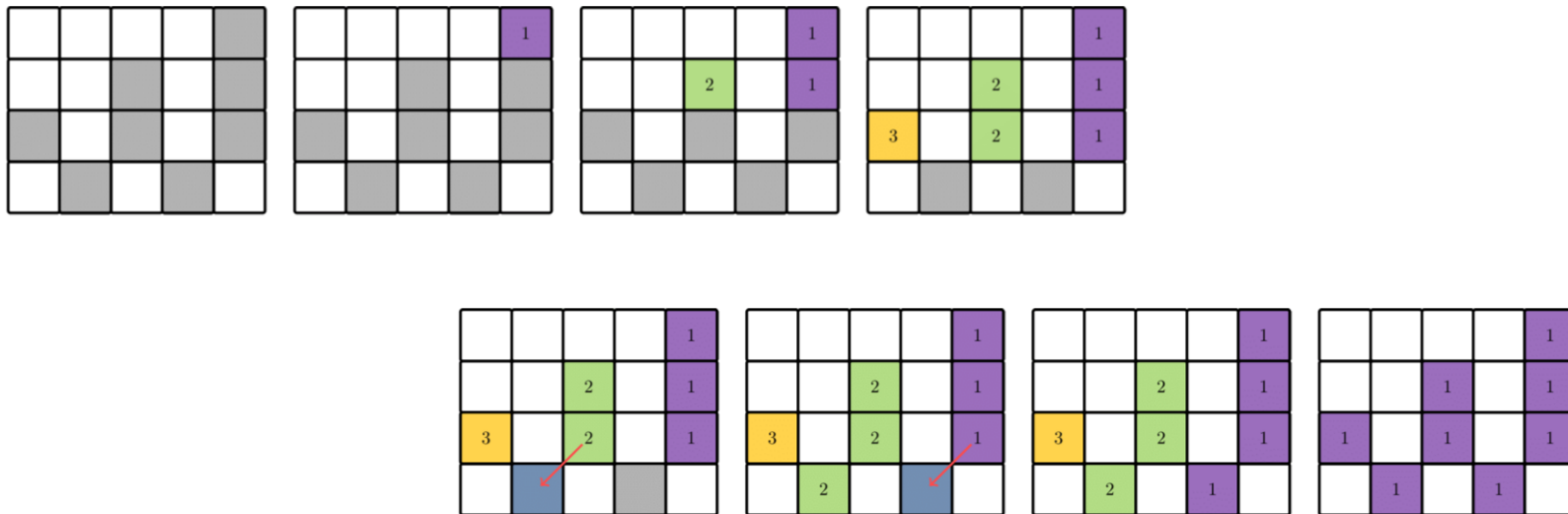
$$T(e) \leftarrow T(T(e)) \quad \text{si } e \neq T(e)$$

0	1	2	3	4	5	6	7	8
0	1	1	2	3	1	4	1	5

Approche ECC Rosenfeld (3 / 3)

- ▶ Deuxième attribution avec la table d'équivalence selon un parcours lexicographique

$$e \leftarrow T(e)$$



Segmentation



Approches « Régions »

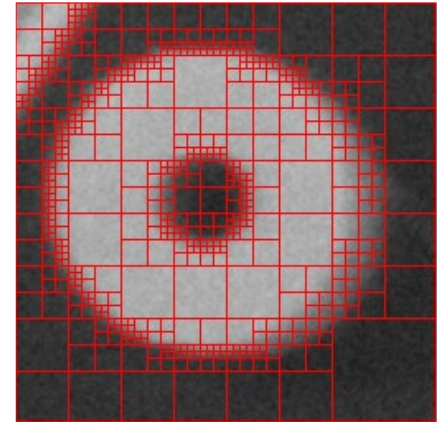
Segmentation par division

► Principe

- Segmentation par division de régions, « de l'image au pixel »
- Approche récursive sur la base d'un critère d'homogénéité
- Souvent utilisée en initialisation d'une autre approche

► Algorithme

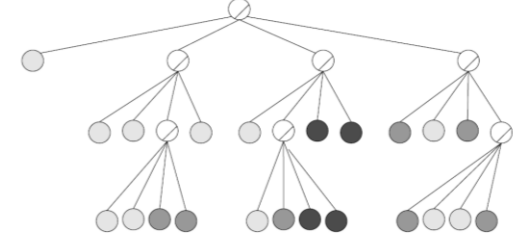
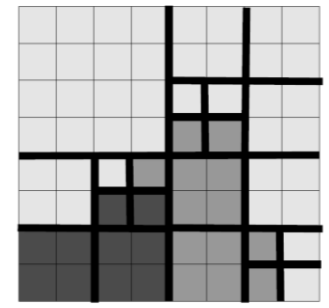
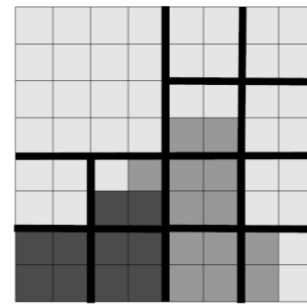
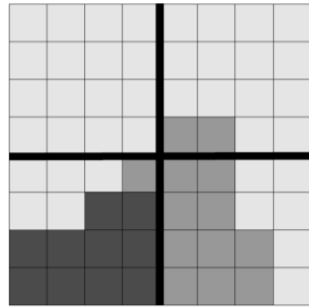
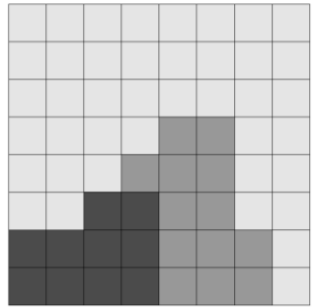
- Initialisation par image entière ou sous-partie
- Récursion
 - Si homogénéité inférieure à un seuil a priori
 - Test d'une ou de plusieurs divisions
 - Division retenue selon le nombre maximal de sous-régions homogènes
 - Arrêt
 - Homogénéité supérieure à un seuil a priori
 - Surface minimale de région atteinte



Influence de la géométrie (carrée, triangle, etc.) des divisions et non-réversibilité d'une division « erronée » !

Division par Quad-Tree

- ▶ Division unique en 4 régions
- ▶ Représentation par un arbre
 - Racine représentant l'image
 - 4 nouvelles feuilles à chaque division



Divisions rigides et peu naturelles

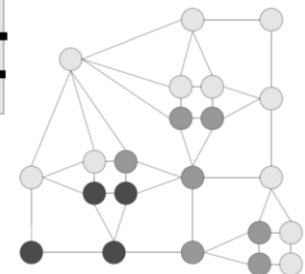
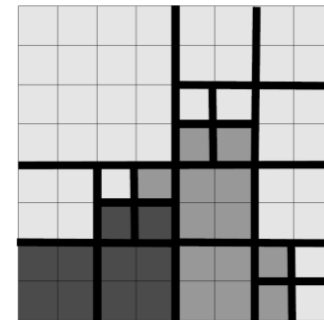
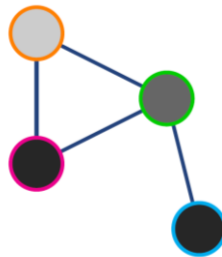
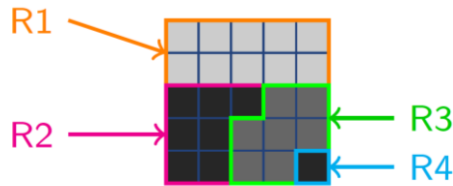
Segmentation par fusion (1 / 2)

► Principe

- Segmentation par fusion de régions, « du pixel à l'image »
- Construction d'un graphe d'adjacence
- Fusions de couples de régions adjacentes homogènes

► Graphe d'adjacence/Region Adjacency Graph (RAG)

- Chaque région est un nœud
- Chaque relation d'adjacence est un arc valué par la similarité des régions



Segmentation par fusion (2 / 2)

► Algorithme

- Initialisation par la grille des pixels ou une segmentation
- Construction d'un graphe d'adjacence initial
- Construction d'une liste ordonnée des couples de nœuds adjacents
- Itération/Récursion
 - Identification du meilleur couple selon un critère
 - Mise à jour du graphe
 - Contraction d'une arête
 - Suppression éventuelle d'arêtes multiples
 - Mise à jour de la liste

**Influence de l'ordre de regroupement
(les petites régions souvent en début de fusion)**

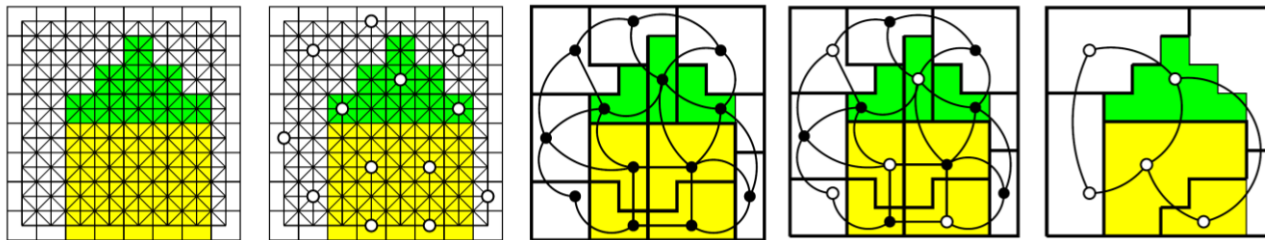
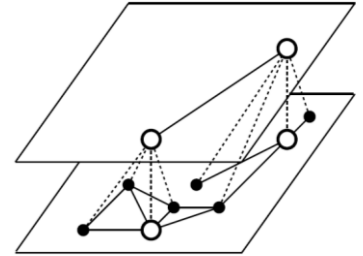
Similarité/Homogénéité de régions

- ▶ Union des régions
 - Variance globale
 - Proportion d'individus dont l'intensité est au-delà d'un écart-type à la moyenne
- ▶ Critère mixte entre régions, Beveridge (1989)
 - Différence de dispersion normalisée
 - Taille de la région la plus petite
 - Inverse de la longueur relative de la frontière à la longueur cumulée des frontières

$$f(R_i, R_j) = f_{sim}(R_i, R_j) f_{size}(R_i, R_j) f_{frontier}(R_i, R_j)$$

Fusion par pyramide adaptative

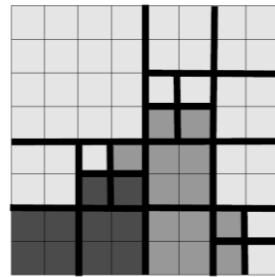
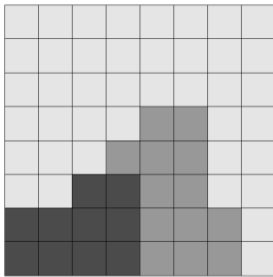
- ▶ Fusion de groupes de régions au lieu de couples
- ▶ Pyramide : structure de données hiérarchique reliant des graphes d'adjacence
- ▶ Algorithme
 - De la base (maille 8-connexe) au sommet (image)
 - Itération (remontée des niveaux)
 - Extraction des sommets survivants (selon un critère)
 - Deux survivants ne peuvent être adjacents
 - Tout non-survivant est adjacent à au moins un survivant
 - Fusion des non-survivants avec les survivants les plus similaires



**Réduction rapide du
graphe et indépendante
du sens de parcours**

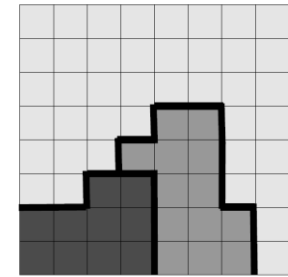
Division/Fusion

- ▶ Split&Merge, Horowitz/Pavlidis (1976)
- ▶ Principe générique et séquentiel
 - Segmentation par division de régions, « de l'image aux régions »
 - Segmentation par fusion de régions, « des régions à l'image »



Division par
Quad-Tree

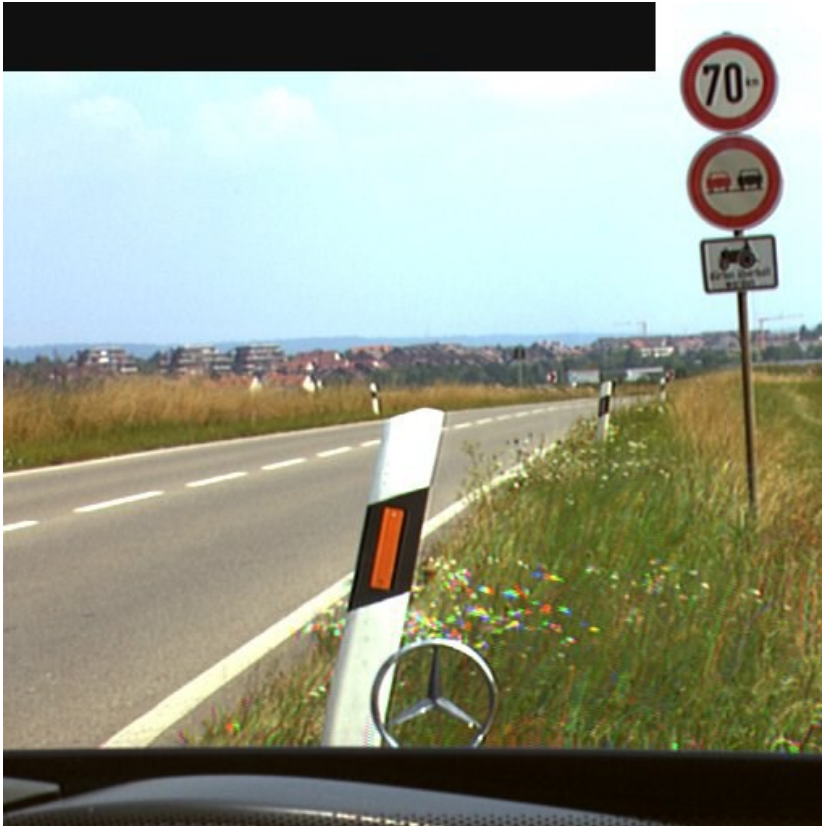
Construction du
graphe d'adjacence



Fusion

- ▶ Résultats parfois meilleurs en alternant les phases

Split & Merge



Géométrie de division visible, sur-décomposition, fusions erronées

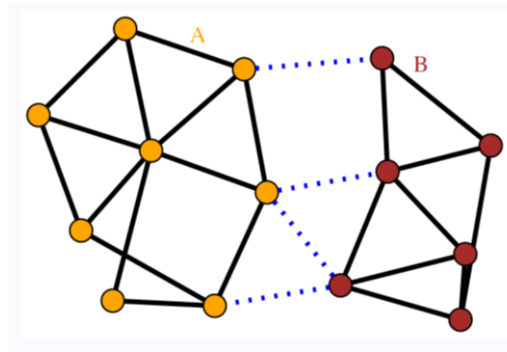
Division par Graph-Cut (1 / 2)

► Principe

- Segmentation obtenues par coupes successives d'un graphe
- Coupe : ensemble d'arêtes qui sépare un graphe en deux parties
- Coût de la coupe égal à la somme des poids des arêtes de séparation

► Minimum Cut

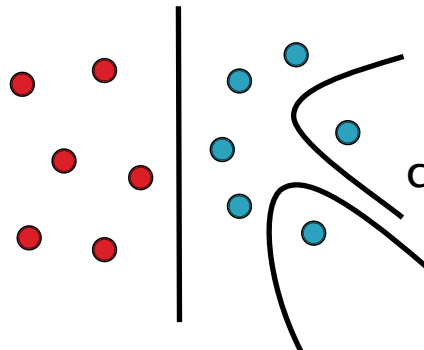
Graphe $G = (V, E)$



Coût de coupe

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

coupe optimale



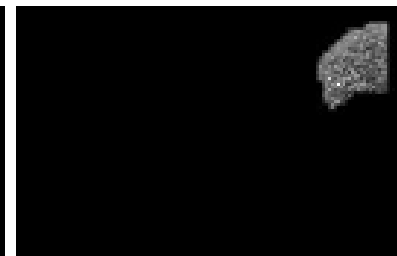
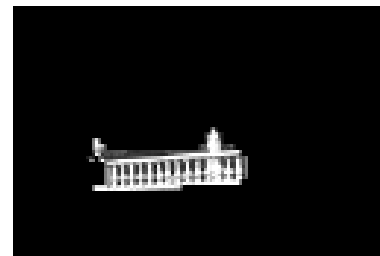
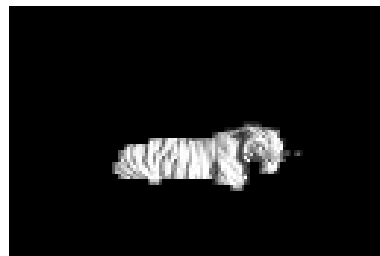
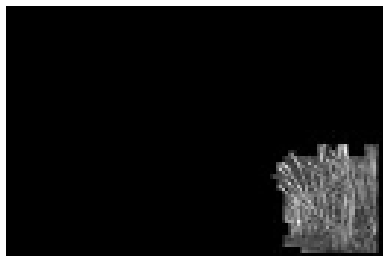
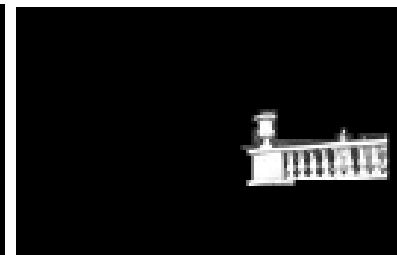
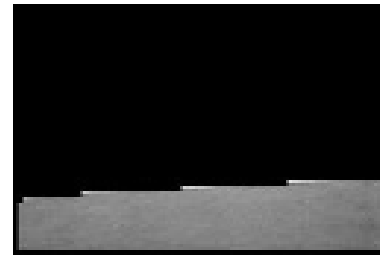
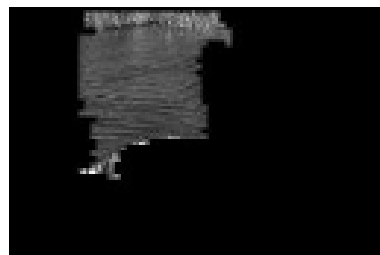
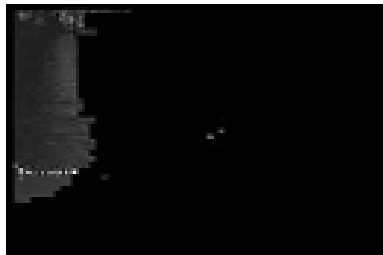
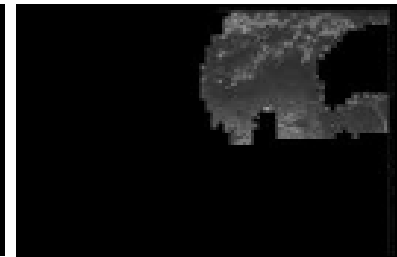
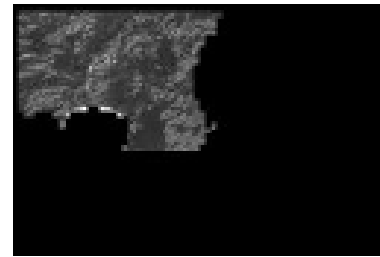
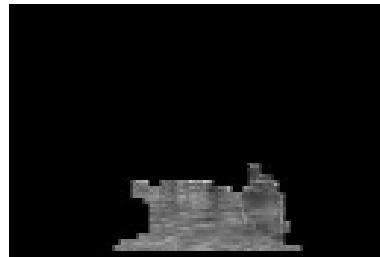
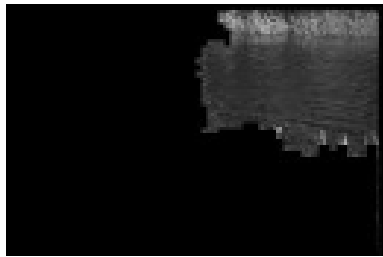
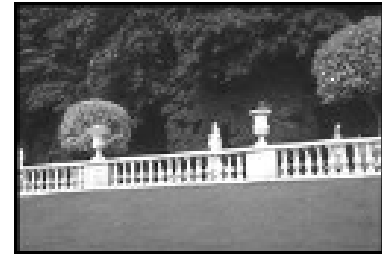
coupes de petite
taille et de
moindre coût



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

Coût normalisé

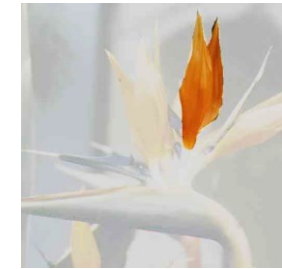
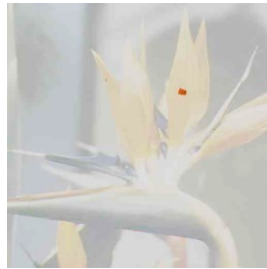
Division par Graph-Cut (2 / 2)



Croissance de régions

► Principe

- Point initial ou région initiale a priori (Germe)
- Croissance par ajouts successifs des points voisins de la région, selon un critère d'homogénéité point à région



- Convergence par seuil de critère (ou recouvrement total)
- Croissances multiples en compétition si plusieurs germes

► Propriétés

- Forte dépendance au choix de l'initialisation et à l'ordre des ajouts (caractéristique de région évolutive)
- Implémentation simple et temps de calcul relativement faible

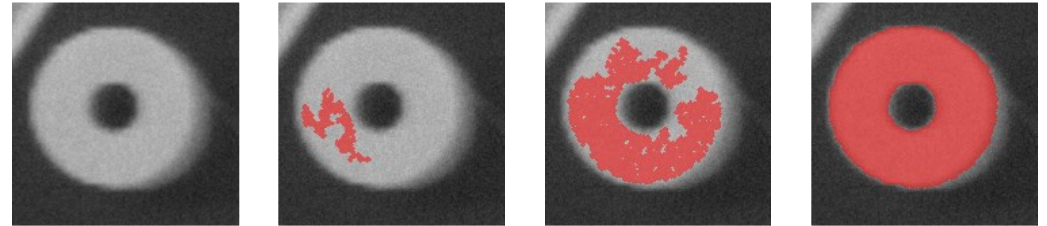
Algorithme

► Initialisation

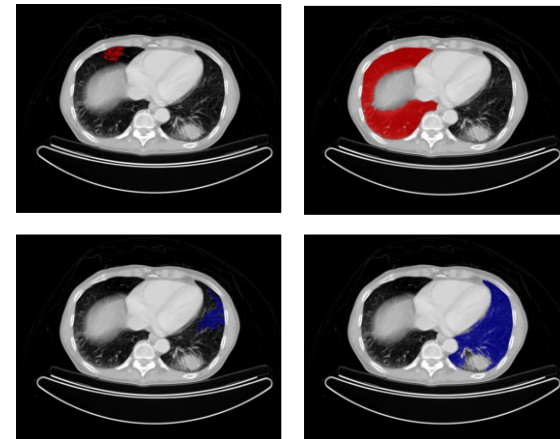
- Région initiale R
- File FIFO (First In First Out) S qui contient les points frontière de R

► Itération

- Extraction de p de S
- Si p est homogène avec R
 - Ajout de p à R
 - Ajout à S des points du voisinage de p non inclus dans R



► Arrêt si S est vide

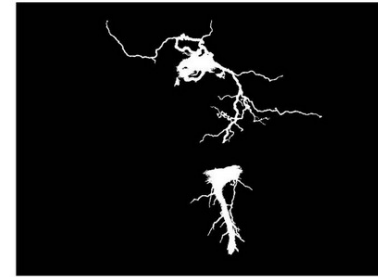
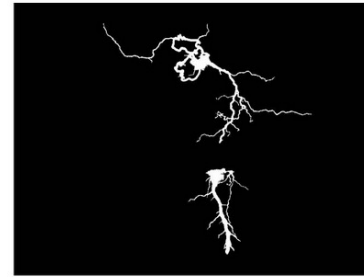


Résultats

► Influence du seuil

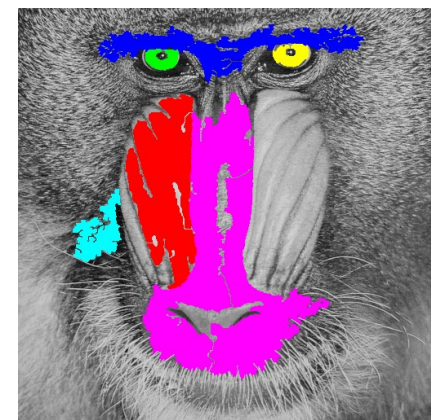
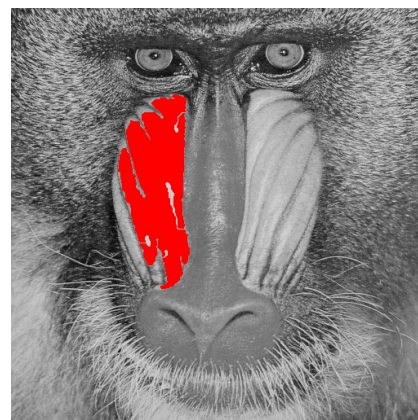
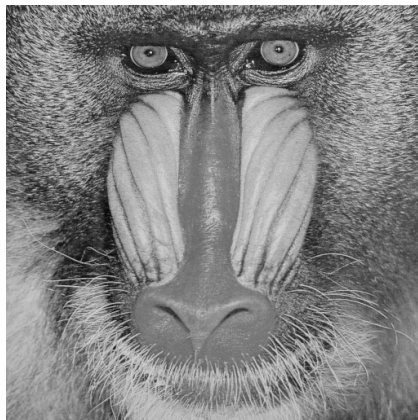


2 germes



seuil croissant

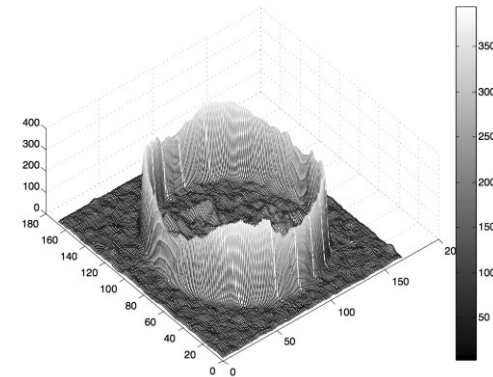
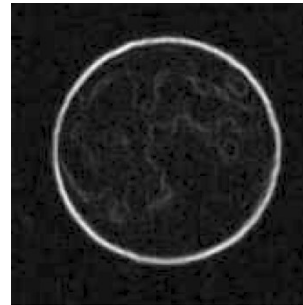
► Croissances multiples



Ligne de partage des eaux (1 / 2)

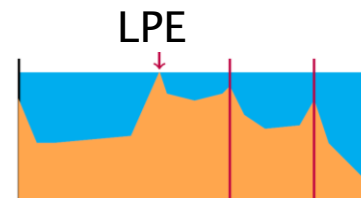
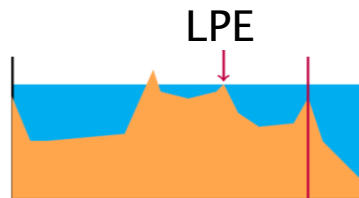
► Principe

- Norme du gradient considérée en tant que relief (carte d'élévation)



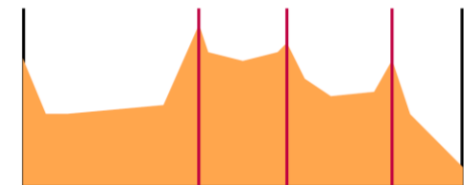
- Détermination des séparations de bassins versants (lignes de crête) lors d'une simulation d'immersion progressive
 - ❑ Extraction de minimas locaux considérés en tant que fonds de bassins
 - ❑ Montée progressive de l'eau (traitement selon des élévations croissantes) et extension des bassins immergés
 - ❑ Marquage de la « ligne de partage des eaux » (LPE) dès la jonction de deux bassins

Ligne de partage des eaux (2/2)



Montée de l'eau
(immersion)

segmentation finale



Algorithme LPE (1 / 4)

- ▶ Calcul de l'image G de la norme du gradient de l'image I
- ▶ Calcul de l'image binaire B des minimas locaux de G
- ▶ Labellisation E de B (étiquettes de bassins, segmentation initiale)
- ▶ Pour chaque intensité possible g de G selon un parcours croissant
 - Pour chaque site s de G d'intensité égale à g et non affecté dans E
 - ▢ Construction du vecteur E_s des étiquettes de E dans le voisinage de s
 - ▢ Affectation d'une nouvelle étiquette $E(s)$ au site s
 - ▢ Etiquette 'S' dans le cas d'au moins une étiquette de région ou d'une étiquette 'LPE' présente dans E_s
 - ▢ Etiquette 'M' dans le cas contraire
 - Création d'une liste (FIFO) L composée des sites de E d'étiquette 'S'
 - Tant que la liste L n'est pas vide
 - ▢ Extraction du premier élément s de L

Algorithme LPE (2 / 4)

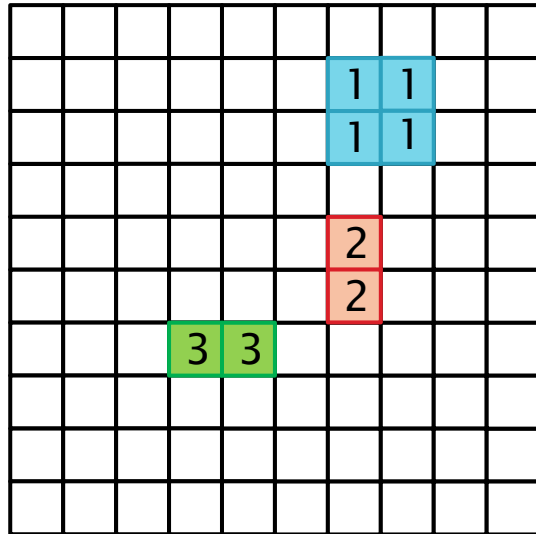
- ❑ Construction du vecteur E_s des étiquettes de E dans le voisinage de s
- ❑ Affectation d'une étiquette $E(s)$ au site s
 - ❑ Etiquette (de région ou 'LPE') présente dans E_s dans le cas d'une étiquette unique présente dans E_s
 - ❑ Etiquette 'LPE' dans le cas contraire
 - ❑ Ajout à la liste L des sites d'étiquette 'M' inclus dans E_s
- Création de l'image binaire B des sites s de E d'étiquette 'M'
- Labellisation E_M de B (étiquettes de nouveaux bassins)
- « Ajout » de E_M à E avec décalage de $\max(E)$ des étiquettes de E_M

- ▶ Création de l'image binaire B des sites de E d'étiquette non égale à 'LPE'

- ▶ Labellisation E de B (étiquettes de bassins, segmentation finale)

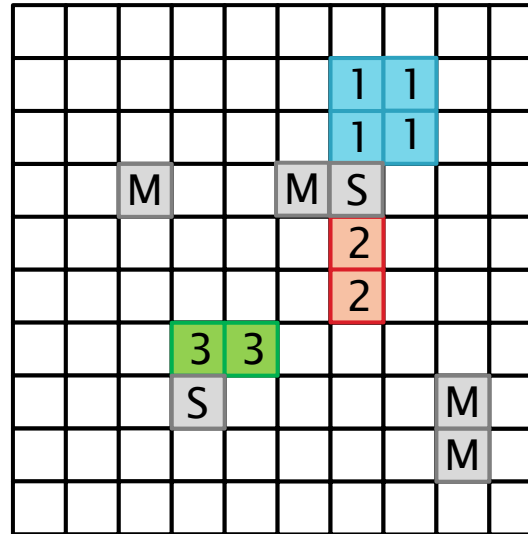
Algorithme LPE (3 / 4)

Bassins labellisés du niveau d'immersion précédent

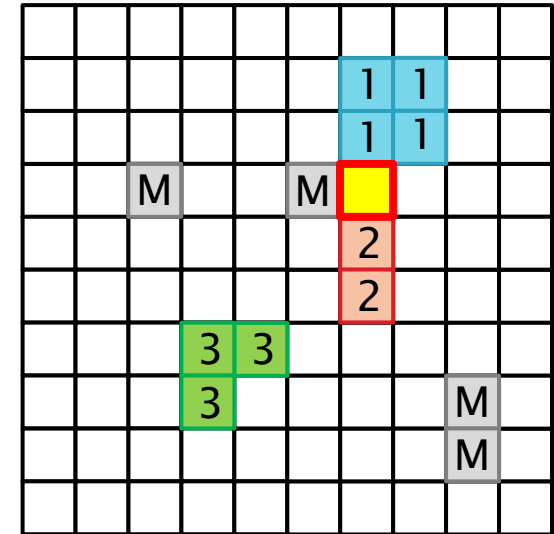


Sites du niveau d'immersion courant

adjacent/non-adjacent



adjacent → label bassin ou LPE



S adjacent à un bassin

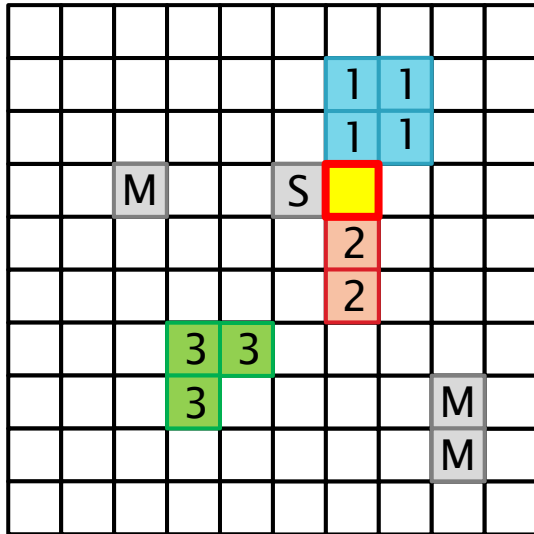
M non-adjacent à un bassin

point de LPE

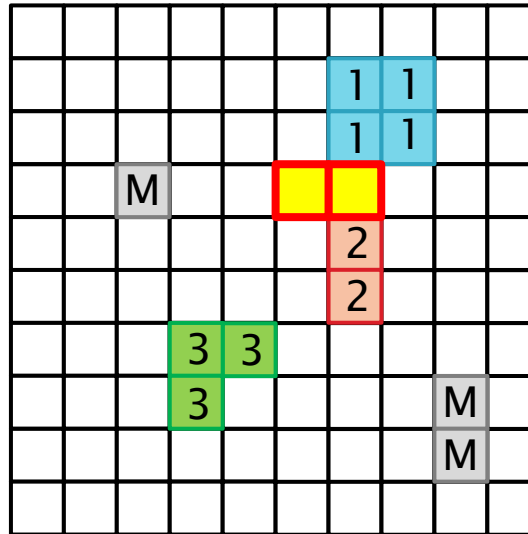
Algorithme LPE (4/4)

Sites du niveau d'immersion courant

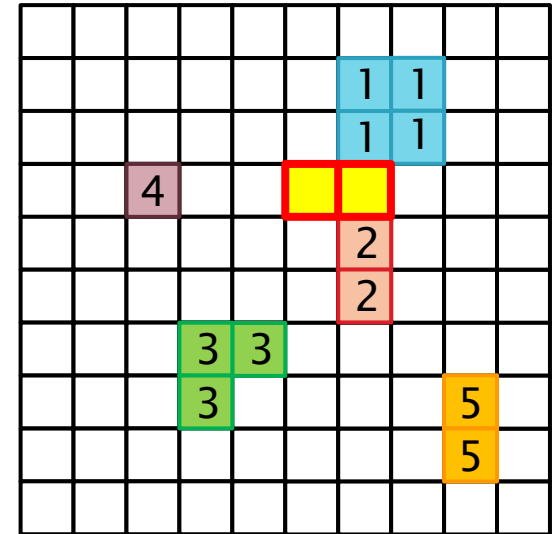
non-adjacent→adjacent



adjacent→label bassin ou LPE

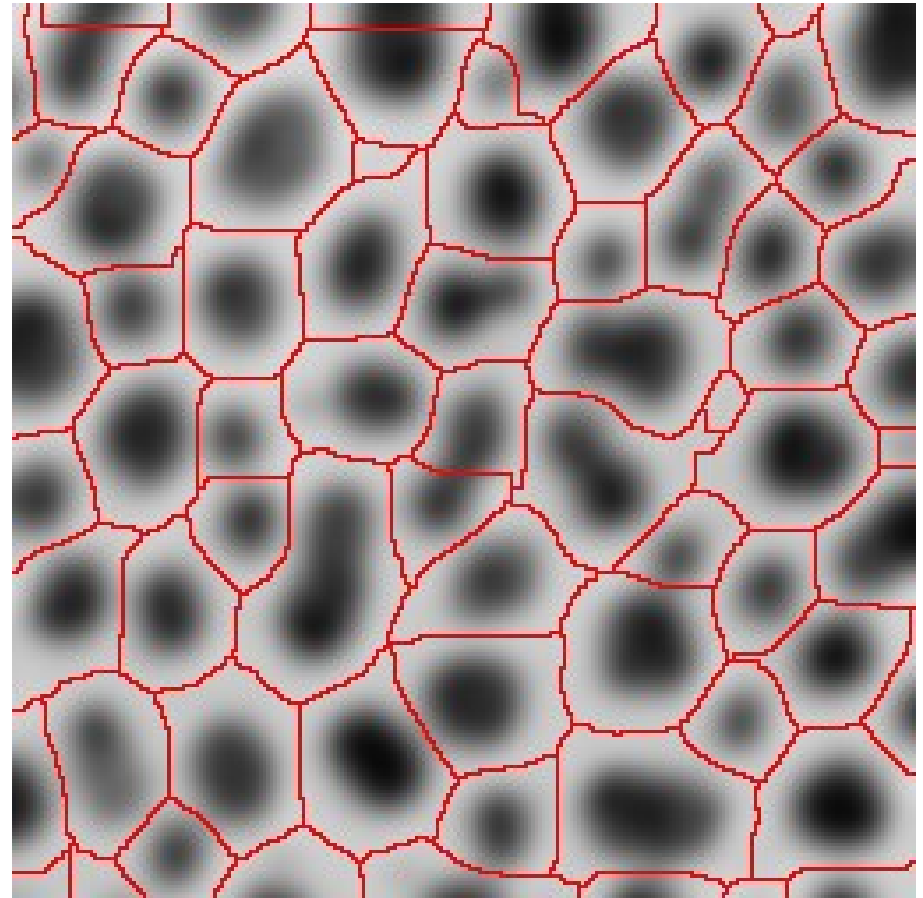


non-adjacent→label bassin



Bassins labellisés du niveau d'immersion courant (+LPE)

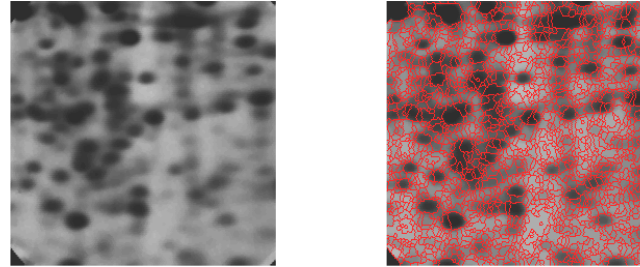
Résultats



LPE contrainte

► Limite et amélioration

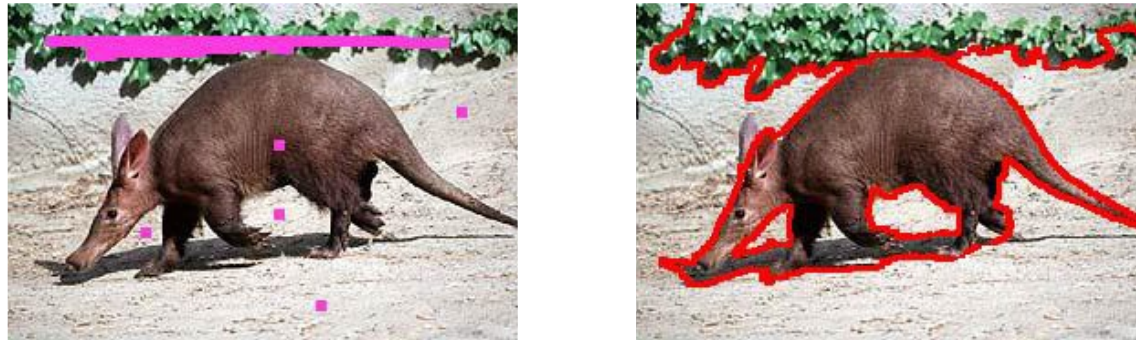
- Sur-segmentation due à la multiplicité des minima locaux du gradient



- Réduction du nombre de régions par lissage amont du gradient

► Initialisation par marqueurs

- Bassins de minima remplacés par des régions a priori
- Marqueurs jouant le rôle de contraintes



Segmentation Markovienne

► Principes

- L'image A est considérée en tant qu'observation constituée par la juxtaposition de réalisations de processus aléatoires gaussiens distincts
- La segmentation Λ est considérée en tant que processus « caché » de type champ de Markov (la loi de probabilité dépend localement du voisinage)

► Notations

$S = \{s_1, s_2 \dots s_N\}$ sites de cardinal N

$A = \{a_s, s \in S\}$ image d'intensité

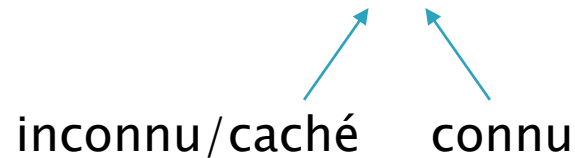
$\Lambda = \{\lambda_s, s \in S\}$ image d'étiquettes

**Cadre applicatif : une image
composée de régions caractérisables
par des statistiques d'ordre 1 et 2
(moyenne et écart-type)**

Problème posé (1 / 2)

► Formalisation

Maximisation de $P(\Lambda|A)$



► Hypothèse bayésienne

$$P(\Lambda|A) = \frac{P(A|\Lambda)P(\Lambda)}{P(A)} \rightarrow \text{maximisation de } P(A|\Lambda)P(\Lambda)$$

► Premier terme

$$P(A|\Lambda) = \prod_{s \in S} P(a_s|\lambda_s) \quad \text{indépendance des intensités entre sites}$$

$$\text{avec } P(a_s|\lambda_s) = \frac{1}{\sigma_{\lambda_s} \sqrt{2\pi}} \exp\left(-\frac{(a_s - \mu_{\lambda_s})^2}{2\sigma_{\lambda_s}^2}\right)$$

Problème posé (2 / 2)

► Second terme

$$P(\Lambda) = \prod_{s \in S} P(\lambda_s | \lambda_{V(s)}) \quad \text{processus Markovien}$$

voisinage

► Equivalence avec les champs de Gibbs (Physique)

$$P(\Lambda) = \exp(-U(\Lambda)) \quad \text{avec} \quad U(\Lambda) = \sum_{c \in C} W_c(\lambda_c)$$

énergie d'étiquetage

ensemble des cliques

clique (ensemble de sites mutuellement voisins)

potentiel autologistique

$$W_c = \begin{cases} -\beta & \text{si } \lambda_{s_1} = \lambda_{s_2} \\ +\beta & \text{si } \lambda_{s_1} \neq \lambda_{s_2} \end{cases}$$

régularité attendue

Reformulation du problème

Minimisation d'une énergie

$$E(A|\Lambda) = -\log(P(A|\Lambda)P(\Lambda))$$

Expression globale

$$E(A, \Lambda) = \sum_{s \in S} \left(\log(\sigma_{\lambda_s} \sqrt{2\pi}) + \frac{(a_s - \mu_{\lambda_s})^2}{2\sigma_{\lambda_s}^2} \right) + \sum_{c \in C} W_c(\lambda_c)$$

Expression locale

$$e_s = \log(\sigma_{\lambda_s} \sqrt{2\pi}) + \frac{(a_s - \mu_{\lambda_s})^2}{2\sigma_{\lambda_s}^2} + \sum_{Cs} W_{Cs}(\lambda_{Cs})$$

Iterated Conditional Mode (ICM)

- ▶ Initialisation par un étiquetage « de qualité »
- ▶ Tant que la convergence n'est pas atteinte
 - Pour chaque site s
 - Test exhaustif des étiquettes possibles
 - Affectation de l'étiquette λ_s minimisant l'énergie locale e_s

Méthode déterministe qui converge vers un minimum local
Algorithme pertinent si initialisation proche du minimum global

Metropolis

- ▶ Initialisation par un étiquetage aléatoire et choix d'une « température » T
- ▶ Tant que la convergence n'est pas atteinte
 - Pour chaque site s
 - ▣ Tirage aléatoire d'une étiquette candidate x
 - ▣ Calcul de l'énergie locale $e_s(x)$
 - ▣ Affectation conditionnelle de l'étiquette x
 - ▣ Si $e_s(x) < e_s(\lambda_s)$
 - ▣ Si $e_s(x) \geq e_s(\lambda_s)$ avec une probabilité $e^{-\Delta e/T}$

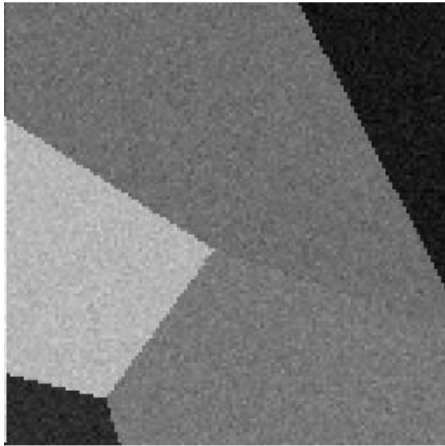
Méthode stochastique : l'acceptation d'un « mauvais » état permet d'explorer un espace de solutions plus grand et non limité à un minimum local
Algorithme optimal mais au temps d'exécution « déraisonnablement » long

Algorithme de recuit simulé

- ▶ Initialisation par un étiquetage aléatoire et choix d'une « température » T
- ▶ Tant que la convergence n'est pas atteinte
 - Pour chaque site s
 - ▣ Tirage aléatoire d'une étiquette candidate x
 - ▣ Calcul de l'énergie locale $e_s(x)$
 - ▣ Affectation conditionnelle de l'étiquette x
 - ▣ Si $e_s(x) < e_s(\lambda_s)$
 - ▣ Si $e_s(x) \geq e_s(\lambda_s)$ avec une probabilité $e^{-\Delta e/T}$
 - Décroissance de T (selon $T \leftarrow kT$ avec $k \in [0,8 \ 0,999 \dots]$)

Variante sous-optimale mais au temps d'exécution « raisonnable » (quelques centaines d'itérations)

ICM vs Récuit simulé



ICM



Récuit simulé

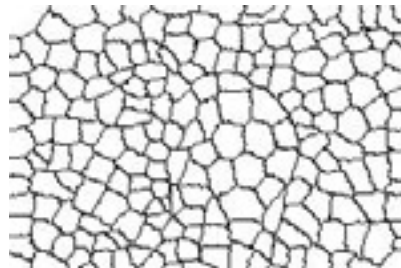
	k = 0.9	k = 0.95	k = 0.99	k = 0.999
T = 2				
T = 10				
T = 30				

Superpixels (1 / 3)

- ▶ Regroupement de pixels (Ren et Malik, 2003)
 - Similaires en couleurs ou en propriétés de bas-niveau
 - Offrant de nouvelles entités
 - Naturelles, pas simplement issues d'une discrétisation régulière
 - Structurelles, représentant les constituants de l'image
 - En nombre réduit, compatible avec des traitements ultérieurs



segmentation
humaine



superpixels



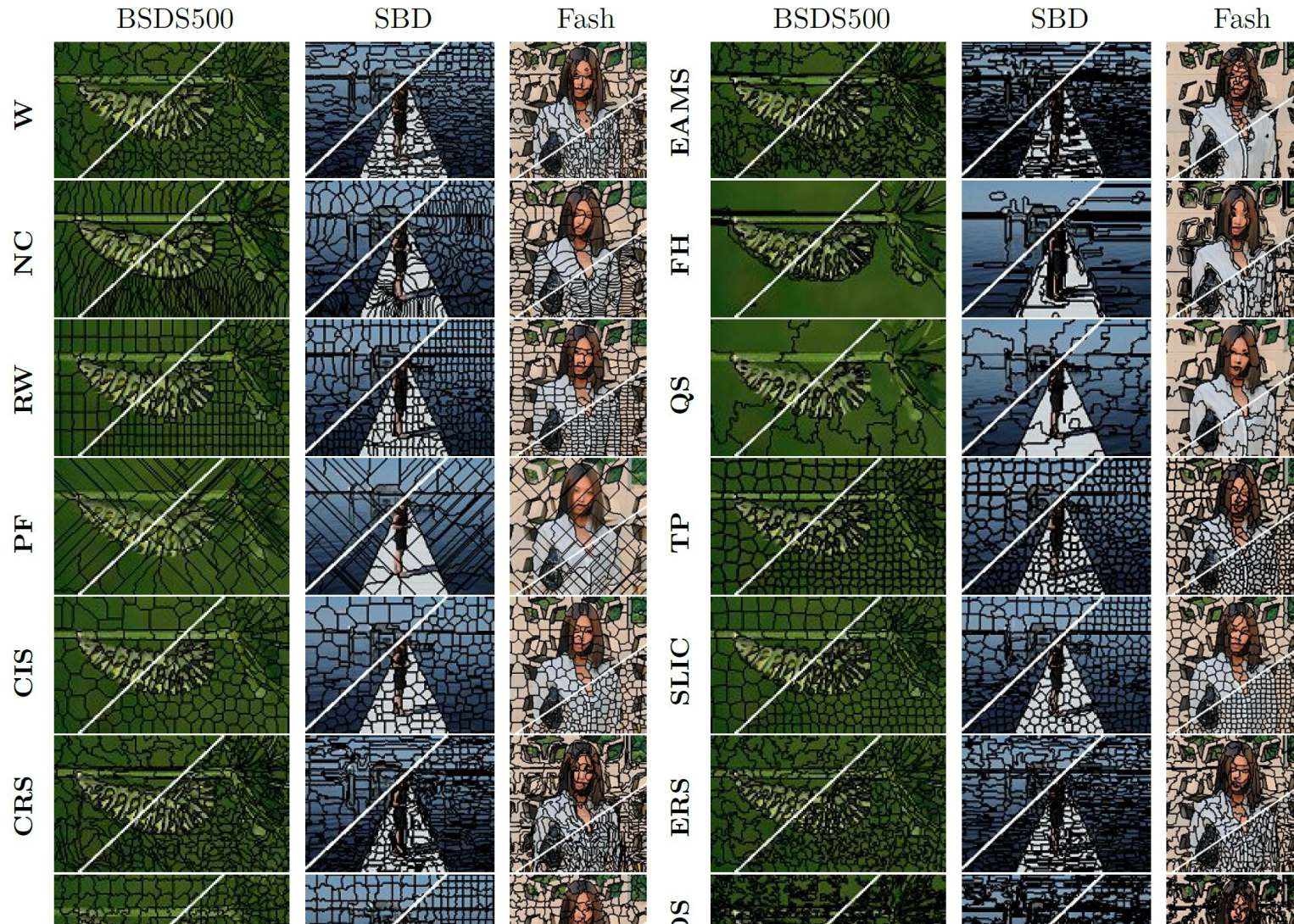
segmentation
humaine à
partir des
superpixels

Superpixels (2 / 3)

- ▶ **Nombreux algorithmes ≥ 28 (Stutz, 2017)**
 - Watershed (Compact Watershed, Water Pixel, etc.)
 - Densité (Edge-Augmented Mean Shift, Quick Shift, etc.)
 - Graphes (Random Walks, Constant Intensity Superpixels, etc.)
 - Evolution de contours (Turbo Pixels, Eikonal Region Growing Clustering, etc.)
 - Chemin (Path Finder, Topology Preserving Superpixels, etc.)
 - Clustering (Simple Linear Iterative Clustering, Vcells, etc.)
 - Optimisation d'énergie (Contour Relaxed Superpixels, etc.)
 - Ondelettes (Superpixels from Edge-Avoiding Wavelets, etc.)

- ▶ **Caractéristiques différentiantes**
 - Contrôle du nombre de superpixels
 - Compacité des superpixels
 - Nombre d'itérations

Superpixels (3 / 3)



Simple Linear Iterative Clustering (1 / 3)

- ▶ Principe : k-means étendu et adapté

SLIC (Achanta, 2011)

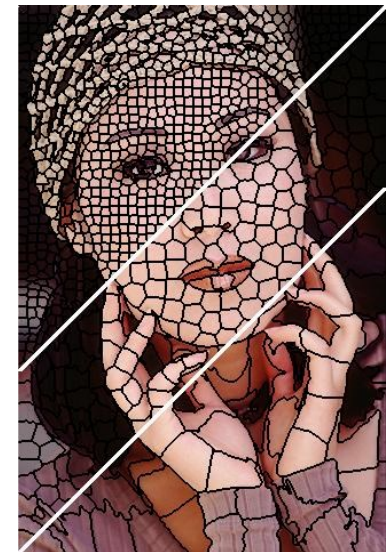
- ▶ Géométrie de la segmentation

- Nombre fixe de superpixels (grille régulière) → contrôle de taille moyenne
- Critère mixte et pondéré mêlant attache à l'image et compacité → contrôle d'un compromis entre représentativité et régularité

$$D = d_{lab} + \frac{m}{S} d_{xy}$$

Diagram illustrating the SLIC distance formula components:

- d_{lab} : attaché à l'image (labeled as "pondérateur" in the diagram)
- $\frac{m}{S}$: « taille » d'un superpixel (labeled as "compacité" in the diagram)
- d_{xy} : $\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$ (labeled as "compacité" in the diagram)
- $\sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$: attaché à l'image



- ▶ Complexité calculatoire maîtrisée

- Linéaire selon le nombre de pixels (fenêtre de recherche fixe)
- Linéaire au nombre d'itérations

Simple Linear Iterative Clustering (2 / 3)

► Lab vs Intensité

► S vs (S_x, S_y)

► Seuil vs nombre d'itérations

Algorithm 1 SLIC superpixel segmentation

/ Initialization */*

Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S .

Move cluster centers to the lowest gradient position in a 3×3 neighborhood.

Set label $l(i) = -1$ for each pixel i .

Set distance $d(i) = \infty$ for each pixel i .

repeat

/ Assignment */*

for each cluster center C_k **do**

for each pixel i in a $2S \times 2S$ region around C_k **do**

 Compute the distance D between C_k and i .

if $D < d(i)$ **then**

 set $d(i) = D$

 set $l(i) = k$

end if

end for

end for

/ Update */*

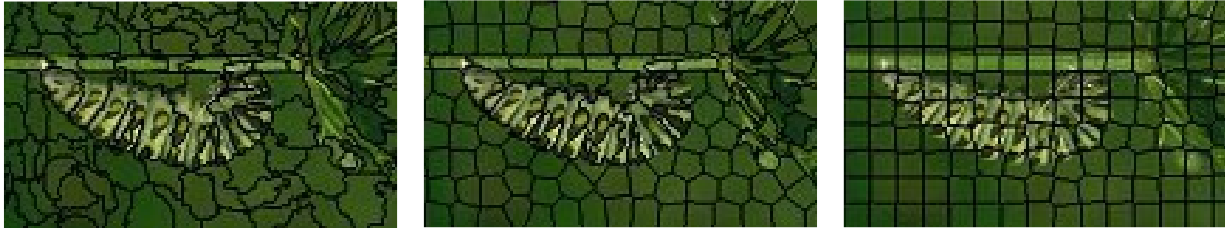
Compute new cluster centers.

Compute residual error E .

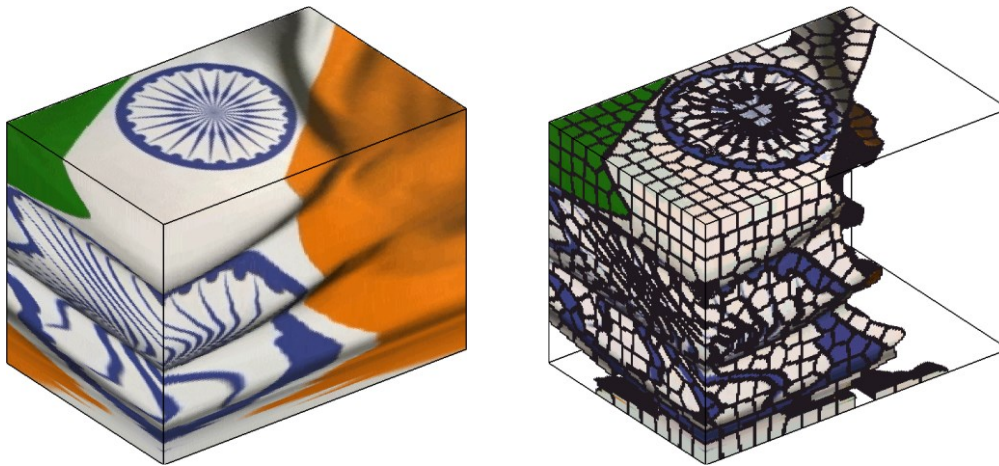
until $E \leq \text{threshold}$

Simple Linear Iterative Clustering (3/3)

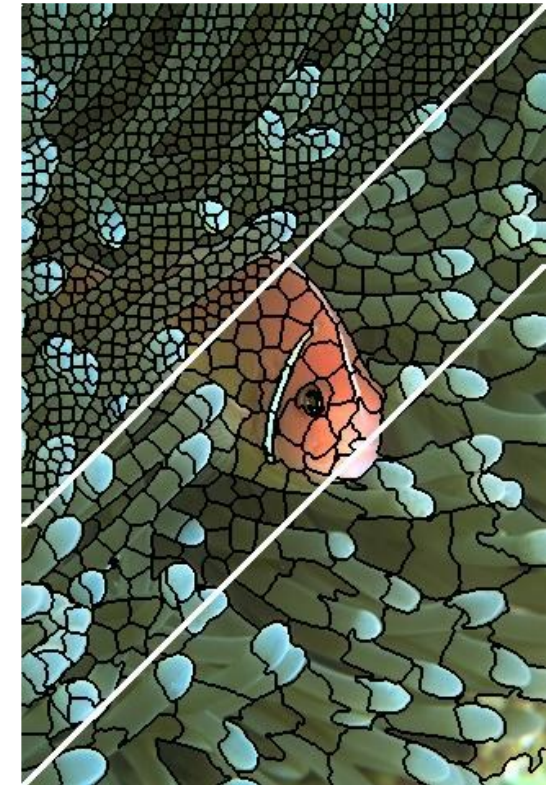
Influence de la compacité



Extension à une séquence vidéo



Influence de la taille



Approche par optimisation globale

- ▶ Minimisation d'une fonctionnelle mesurant l'écart entre l'image partitionnée et l'image originale sous contraintes liées à la forme de l'espace des solutions (régularité, constante par morceaux, etc.)
- ▶ Modèle de Mumford et Shah
 - Chaque région est considérée uniforme
 - Energie globale composite (énergie interne + énergie de complexité)

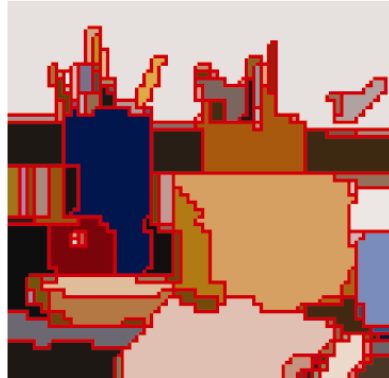
$$E_{\lambda}(\{R_i\}) = \sum_{R_i} E_i(R_i) + \lambda E_c(R_i)$$

longueur des contours, etc.

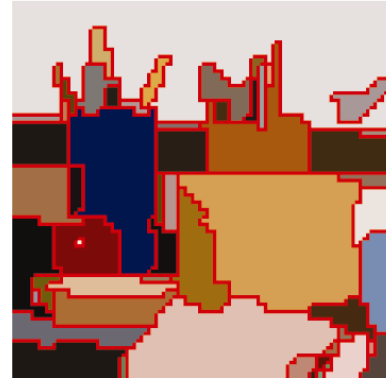
$$E_i(R_i) = \begin{cases} NT_r(V) \\ NT_r(\sqrt{V}) \\ N \log(\det(V)) \end{cases}$$

matrice de covariance de R, V et B

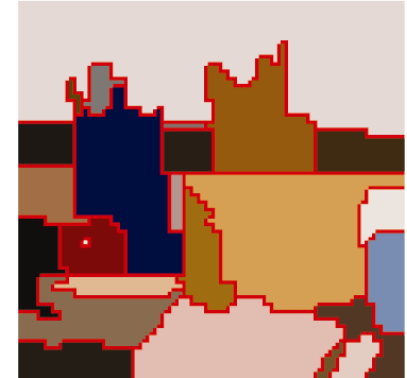
Approche par optimisation globale



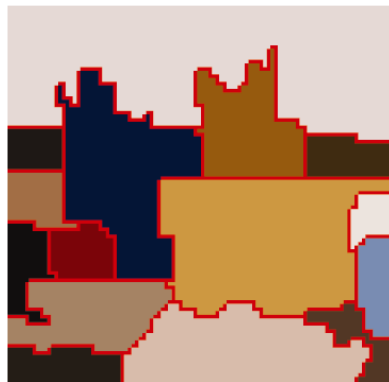
$\lambda=0,1$



$\lambda=0,2$



$\lambda=0,5$



$\lambda=0,8$



$\lambda=0,9$



$\lambda=1,2$

Segmentation



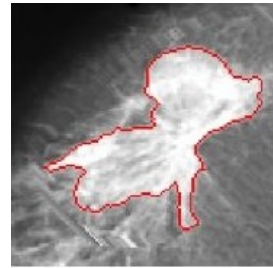
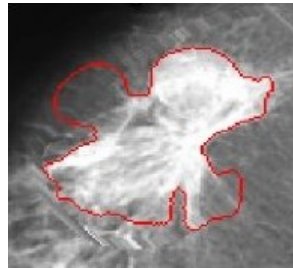
Approches « Frontières »

Modèles déformables (1 / 2)

► Principe

Contours actifs Snakes

- Contour initial fermé proche de la région à segmenter
- Evolution progressive de sorte à adhérer aux bords



► Minimisation d'une fonction d'énergie (Kass, 1987)

$$C = \{v(s) = (x(s), y(s)); s \in [0, 1]\}$$

représentation paramétrique en
abscisse curviligne

$$E(C) = \int_0^1 \left(\alpha \left| \frac{\partial v(s)}{\partial s} \right|^2 + \beta \left| \frac{\partial^2 v(s)}{\partial s^2} \right|^2 \right) ds + \lambda \int_0^1 -|\nabla I(v(s))|^2 ds$$

↑
longueur

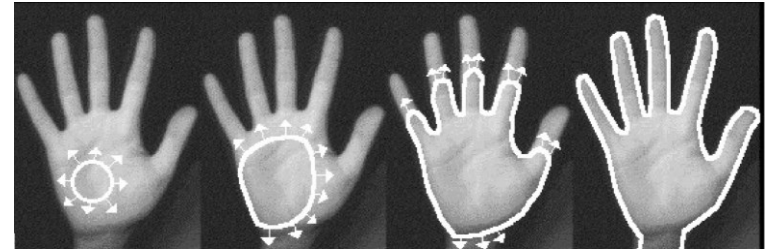
↑
torsion

↑
adhérence aux contours

Modèles déformables (2 / 2)

► Contour initial discrétisé

$$X_0 = \begin{bmatrix} x_0^0 \\ \vdots \\ x_0^i \\ \vdots \\ x_0^{N-1} \end{bmatrix} \quad Y_0 = \begin{bmatrix} y_0^0 \\ \vdots \\ y_0^i \\ \vdots \\ y_0^{N-1} \end{bmatrix}$$



► Itération

$$\begin{cases} X_t = (A + \gamma I)^{-1}(\gamma X_{t-1} + f_X(X_{t-1}, Y_{t-1})) \\ Y_t = (A + \gamma I)^{-1}(\gamma Y_{t-1} + f_Y(X_{t-1}, Y_{t-1})) \end{cases}$$

► Convergence vers un minimum local

Implémentations

- ▶ K-means
- ▶ Etiquetage en composantes connexes (Rosenfeld)
- ▶ Division par Quad-Tree
- ▶ Croissance de régions
- ▶ Segmentation markovienne
- ▶ Ligne de partage des eaux (LPE)
- ▶ Simple Linear Iterative Clustering (SLIC)

SEGMENTATION



TSI/I2SC

Marc Donias