

Modélisation markovienne pour le groupement de contours avec une coopération contours-régions

Sabine Urago, Josiane Zerubia, Marc Berthod

► **To cite this version:**

Sabine Urago, Josiane Zerubia, Marc Berthod. Modélisation markovienne pour le groupement de contours avec une coopération contours-régions. RR-2875, INRIA. 1996. <inria-00073816>

HAL Id: inria-00073816

<https://hal.inria.fr/inria-00073816>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Modélisation markovienne
pour le groupement de contours
avec une coopération contours-régions*

Sabine Urago, Josiane Zerubia et Marc Berthod

N° 2875

Mai 1996

PROGRAMME 4



*Rapport
de recherche*

Modélisation markovienne pour le groupement de contours avec une coopération contours-régions

Sabine Urago, Josiane Zerubia* et Marc Berthod

Programme 4 — Robotique, image et vision
Projet PASTIS

Rapport de recherche n° 2875 — Mai 1996 — 57 pages

Résumé : Dans ce rapport, nous présentons un nouveau modèle markovien pour la restauration de contours incomplets en utilisant une coopération contours-régions. Cet algorithme de prolongation des lignes de contours a pour contrainte de ne pas prolonger un contour dans une zone homogène. Pour ce faire, nous définissons des critères à minimiser afin de compléter les parties manquantes, et nous les insérons dans une nouvelle fonction d'énergie. De plus, nous déterminons des contraintes sur les constantes incluses dans la fonction d'énergie. C'est à l'aide de la méthode des "boîtes qualitatives" que nous justifions le choix de ces constantes. Pour optimiser cette fonction, nous mettons en oeuvre des méthodes stochastique ou déterministe, qui permettent d'aboutir à une configuration stable dans laquelle les contours sont complétés. Cette méthode a été testée sur différentes images et permet d'obtenir des résultats de bonne qualité.

Mots-clé : Détection et groupement de contours, champs de Markov, distributions de Gibbs, boîtes qualitatives, algorithmes d'optimisation.

(Abstract: pto)

*. Email : zerubia@sophia.inria.fr

Contour grouping with contours-regions cooperation using a markovian model

Abstract: In this report, we present a new markovian model which restores incomplete contour images. This technique yields a cooperation between detected contours and homogeneous regions obtained by a “split and merge” method.

For this purpose, we define an energy function containing several criteria which must be optimized. One of these criteria prevents contour extension inside homogeneous regions detected by the “split and merge” algorithm. The parameters involved in the energy function are estimated using “qualitative boxes”.

A stochastic or a deterministic relaxation algorithm is used in order to generate an optimal configuration in which the contours are reconstructed. A parallel implementation on a Connection Machine (CM200) has been tested on several images (indoors, outdoors, satellite (SPOT), industrial and medical images) and resulted in better contour grouping.

Key-words: Detection and grouping of contours, Markov Random Fields, Gibbs distribution, parameter estimation by qualitative boxes, optimization algorithms.

Table des matières

1	Introduction	4
2	Description du champ de Markov	5
3	Les images d'entrée	6
3.1	Image initiale de contours incomplets	6
3.2	Image initiale des régions non-homogènes	6
4	Fonction d'énergie	8
4.1	Notations	11
4.2	Définition de la fonction d'énergie	13
5	Utilisations des boîtes qualitatives pour le modèle de coopération Contours- Régions proposé	17
6	Résultats obtenus sur différentes images	46
7	Conclusion	54

1 Introduction

Pour traiter le problème de la détection des contours d'une image, les modèles de traitement informatique aussi bien que les systèmes visuels biologiques utilisent comme information initiale les intensités lumineuses réfléchies par les objets d'une scène. Les contours intéressants sont les limites des objets de la scène, qui ont un rapport direct avec les discontinuités de l'intensité lumineuse perçue.

Nous avons mis au point un algorithme de restauration d'images de contours incomplets utilisant une modélisation markovienne. Cette méthode présente une coopération entre les contours, obtenus à l'aide du détecteur de Canny-Deriche et des régions définies à partir d'un algorithme "fusion-éclatement".

Nous définissons certains critères à minimiser afin de compléter les parties manquantes et nous les insérons dans une fonction d'énergie. Cet algorithme de prolongation des lignes de contours a pour contrainte de ne pas prolonger un contour dans une zone homogène.

De plus, pour justifier le choix de ces constantes, nous déterminons des contraintes sur les constantes incluses dans la fonction d'énergie, à l'aide de la méthode des "boîtes qualitatives".

Pour optimiser la fonction d'énergie, nous mettons en oeuvre des méthodes d'optimisation stochastique (l'échantillonneur de Gibbs) ou déterministe (l'ICM: "Iterated Conditional Mode"), qui permettent d'aboutir à une configuration stable dans laquelle les contours sont complétés.

Cette méthode a été testée sur différents types d'images (images de scènes réelles d'intérieur, d'extérieur, satellitaires (SPOT), médicales). Les résultats obtenus sont de bonne qualité.

2 Description du champ de Markov

Nous utilisons, comme système de voisinage, la 8-connexité pour déterminer le champ de Markov. De plus, chaque site correspond à un pixel de l'image.

Le champ de Markov est défini sur l'ensemble \mathbf{S} des sites et a pour espace d'états l'ensemble Λ , dont les éléments sont des vecteurs (notés x_s) à trois composantes.

Deux états possibles sont définis : contour (1) ou pas de contour (0).

Soit \mathbf{X} le champ de Markov, et x la configuration définie sur \mathbf{S} et d'espace d'états Λ , l'application x est définie par :

$$\begin{aligned} x : (S) &\longrightarrow (\Lambda, \xi, p) \\ s &\longrightarrow x(s) = x_s = (ch_s, D_s, st_s) \end{aligned}$$

avec :

- s : le site considéré.
- ch_s : le numéro de la chaîne à laquelle le site s appartient.
- D_s : la direction de cette chaîne de contour.
- st_s : l'état du pixel s avec les deux états possibles : 0 - pas de contour, 1 - si le site s appartient à un contour.

Nous n'allons pas détailler ici les trois composantes du vecteur x_s , mais seulement en donner une brève description (pour plus de détails, se reporter à [6]).

Remarque :

Le fait d'utiliser une modélisation markovienne nous permet d'avoir un modèle mathématique simple, ne faisant intervenir dans le calcul de la fonction de coût que des interactions locales. Par contre, le modèle utilisé possède une caractéristique importante, qui est la suivante : une information globale est mémorisée en chaque pixel et permet d'avoir une connaissance sur les chaînes de contour, sa direction moyenne.

3 Les images d'entrée

A partir de l'image initiale des niveaux de gris, nous conservons deux informations importantes : l'image des contours à compléter et l'image des régions non-homogènes. Ce sont donc ces deux images, dérivées de l'image originale des niveaux de gris, qui seront prises pour référence dans la fonction de potentiels permettant le rappel aux données (voir fig. 1).

3.1 Image initiale de contours incomplets

La première image initiale \mathbf{I} , qui constitue une des informations d'entrée fournie à l'algorithme, est l'image des contours obtenue par le détecteur de Canny-Deriche [2], [4] suivi de la suppression des non-maxima locaux et d'un seuillage par hystérésis. En chaque point de l'image \mathbf{I} , la norme du gradient est déterminée.

$$I(s) = \begin{cases} 0 & \text{si le site } s \text{ n'appartient pas à un contour} \\ \|\vec{\nabla}(s)\| & \text{sinon} \end{cases}$$

3.2 Image initiale des régions non-homogènes

L'algorithme attend comme deuxième donnée initiale, l'image des régions non-homogènes. Pour la déterminer, on définit tout d'abord des régions homogènes par l'algorithme "fusion-éclatement", qui s'effectue en deux étapes :

– **Première étape : Diviser toutes les régions non homogènes.**

On considère, au début de cet algorithme, l'image des niveaux de gris. Puis on définit un critère permettant de déterminer si une région (tout d'abord constituée par l'image entière des niveaux de gris) est homogène ou non. Le critère que nous avons choisi est fonction de la variance de la région. Si la variance est inférieure à un seuil Se_1 , la région est dite homogène. Dans le cas contraire, la région non homogène est divisée en quatre régions de même taille. Puis, on recommence cet algorithme de manière récursive, jusqu'à ce qu'il ne reste plus de régions hétérogènes.

– **Deuxième étape : Regrouper les régions homogènes adjacentes.**

La deuxième étape consiste à regrouper récursivement des régions adja-

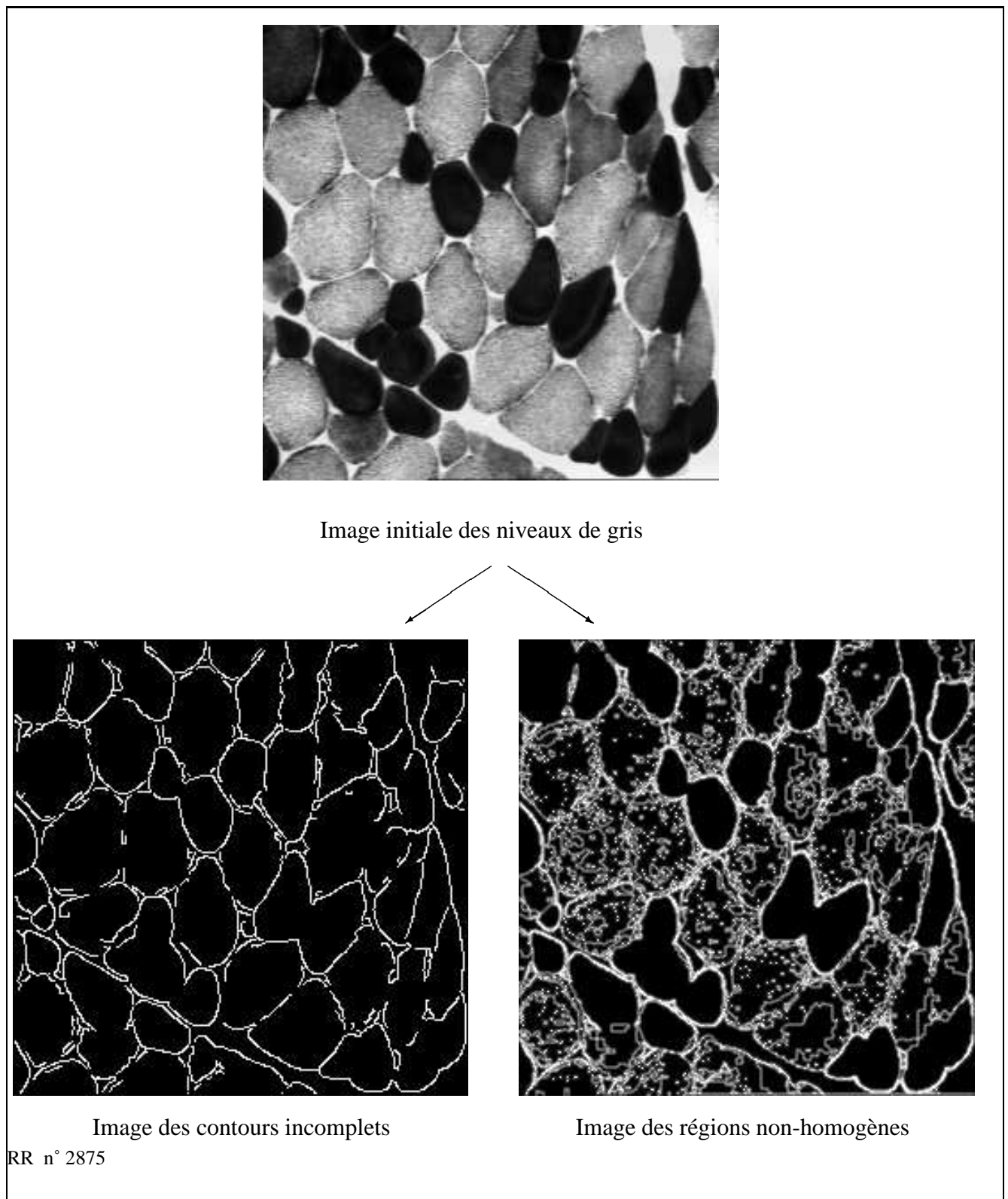


FIG. 1 - Informations initiales retenues à partir de l'image originale des niveaux de gris.

centes qui sont homogènes. Le critère pour définir si deux régions voisines sont homogènes est différent du critère utilisé dans la première partie de l'algorithme. En effet, il est défini de la manière suivante :

Soit deux régions voisines R_1 et R_2 , ces deux régions sont dites homogènes si la différence entre les moyennes des niveaux de gris des deux régions est inférieure à un certain seuil Se_2 .

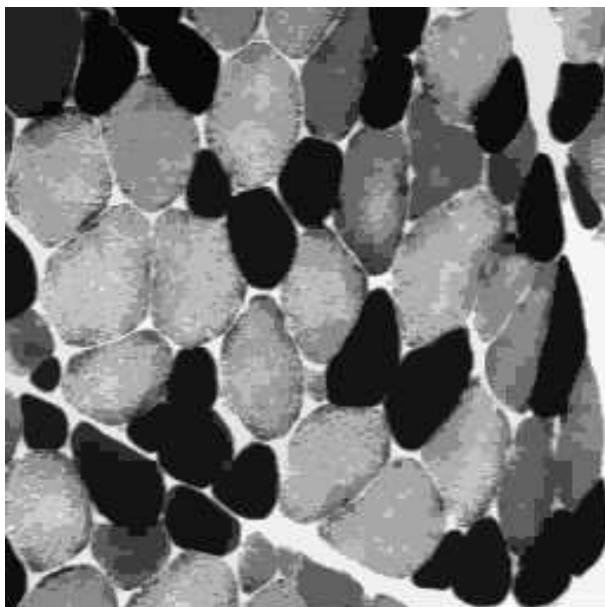
Voici la méthode utilisée pour obtenir l'image initiale des régions non homogènes à partir de cette segmentation en régions :

On comptabilise pour chaque pixel s , le nombre de régions voisines différentes de la région contenant le pixel s considéré. On obtient ainsi une image où, en chaque site, est défini un poids appartenant à l'intervalle $[0, \dots, 8]$. Si s et tous ses voisins appartiennent à la même région, alors le poids associé au site s est nul, ce qui signifie que s appartient à une région homogène. Dans le cas contraire, plus le poids (noté **pds**) est élevé, plus la région est "instable". Nous aurons donc tendance à prolonger les contours dans ces zones "instables" qui sont hétérogènes.

Un exemple d'exécution de l'algorithme de "fusion-éclatement", pour différentes valeurs des seuils utilisés dans les première et deuxième étapes, est donné dans les pages suivantes.

4 Fonction d'énergie

Pour définir la fonction d'énergie, des contraintes sont choisies de manière à prolonger les lignes de contours dans les directions les plus adaptées, et à créer des angles et des "jonctions en T", ceci dans le but d'obtenir des contours fermés. Une contrainte supplémentaire consiste à ne pas prolonger un contour dans une région homogène. En effet, la probabilité de ne pas fermer un contour est élevée si, pour le compléter dans une direction adaptée, le contour doit pénétrer dans une région homogène (voir [6], pour plus de détails).



Segmentation en régions obtenue par l'algorithme de "fusion-éclatement"
Valeurs des seuils : $Se_1=110$, $Se_2=20$

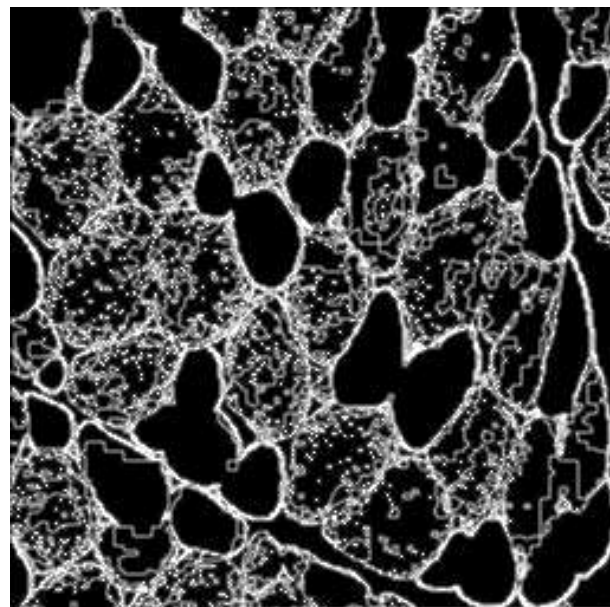
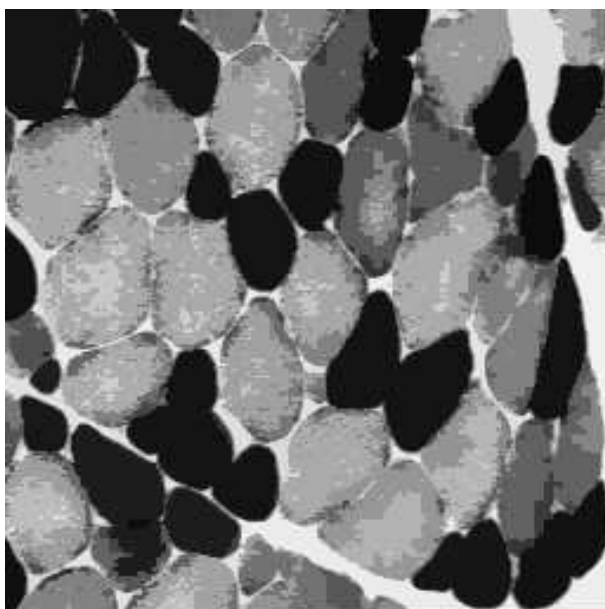


Image des régions non homogènes
Valeurs des seuils : $Se_1=110$, $Se_2=20$



Segmentation en régions obtenue par l'algorithme de "fusion-éclatement"
Valeurs des seuils : $Se_1=110$, $Se_2=25$

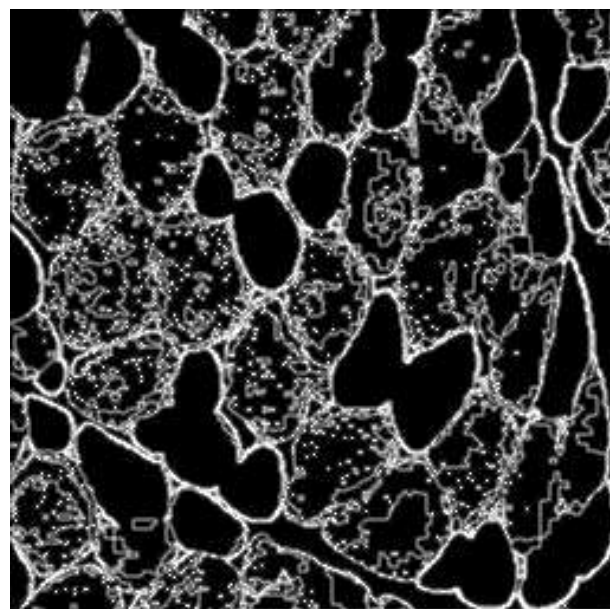
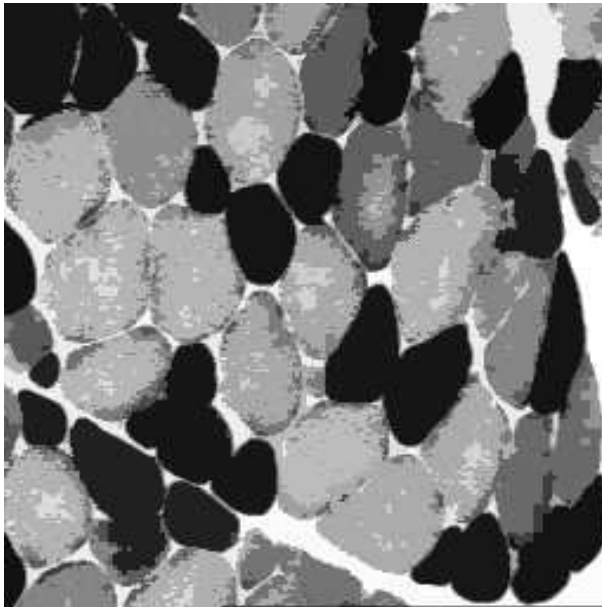


Image des régions non homogènes
Valeurs des seuils : $Se_1=110$, $Se_2=25$

RR n° 2875

FIG. 2 - Algorithme de "fusion-éclatement"



Segmentation en régions obtenue par l'algorithme de "fusion-éclatement"
Valeurs des seuils : $Se_1=110$, $Se_2=30$

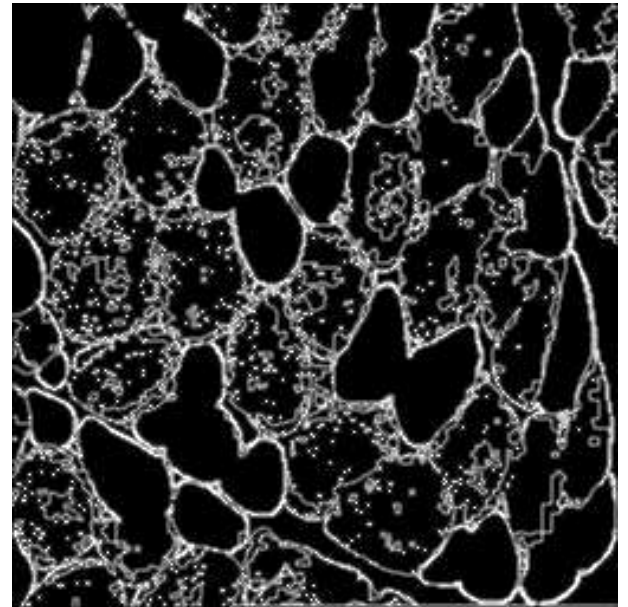
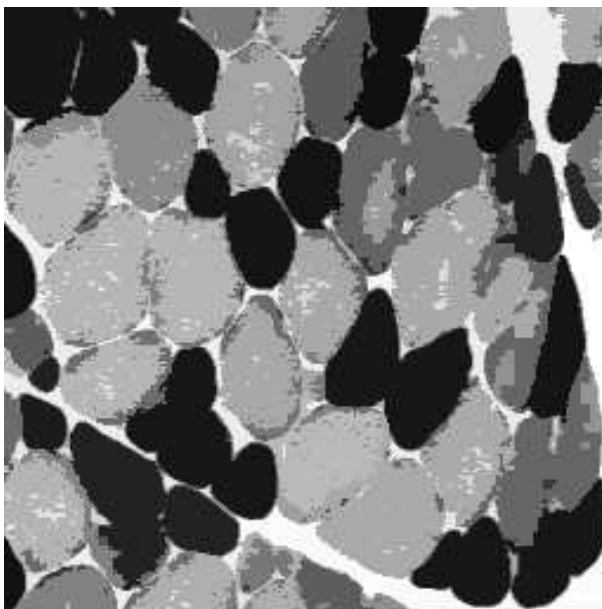


Image des régions non homogènes
Valeurs des seuils : $Se_1=110$, $Se_2=30$



Segmentation en régions obtenue par l'algorithme de "fusion-éclatement"
Valeurs des seuils : $Se_1=110$, $Se_2=36$

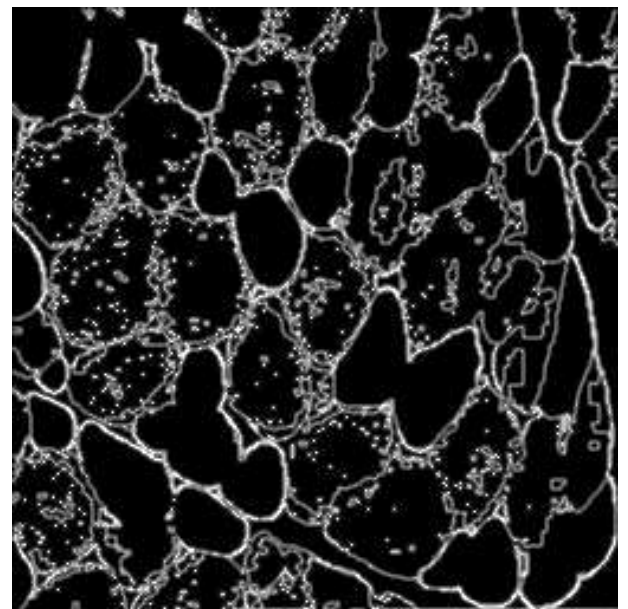


Image des régions non homogènes
Valeurs des seuils : $Se_1=110$, $Se_2=36$

INRIA

FIG. 3 - Algorithme de "fusion-éclatement"

L'énergie totale U est la somme, sur toutes les cliques, de fonctions de potentiels.

$$U = \sum_{c \in \mathcal{C}} V_c$$

où :

- c représente une clique, et \mathbf{C} l'ensemble de toutes les cliques.
- V_c est la fonction de potentiels associée à la clique c .

Avant de définir ces fonctions de potentiels, nous devons introduire quelques notations.

4.1 Notations

- \mathbf{Im} est un booléen relatif aux données initiales

$$Im(s) = \begin{cases} 1 & \text{Si } I(s) \neq 0 \\ 0 & \text{Sinon} \end{cases}$$

- \mathbf{pds} est une variable, comprise dans l'intervalle $[0, \dots, 8]$, déterminée à partir de la segmentation en régions. Si la valeur prise par \mathbf{pds} au site s est nulle, alors s appartient à une région homogène et inversement.
- E_p indique l'épaisseur du contour :

$$E_p(s) = \begin{cases} 1 & \text{Si l'épaisseur du contour est supérieur à 1.} \\ 0 & \text{Sinon} \end{cases}$$

Convention: Représentation d'un contour épais ($E_p(s) = 1$):

$$\begin{array}{ccc} \bullet & \bullet & \mathbf{X} \ \mathbf{X} \\ & & \mathbf{X} \ \mathbf{X} \ \bullet \ \bullet \end{array} \quad \begin{array}{l} \mathbf{X} : \text{contour épais} \\ \bullet : \text{contour fin} \end{array}$$

- \mathbf{Br} est aussi un booléen qui est pris en compte dans le calcul de la fonction d'énergie. Cette variable permet d'avoir une énergie élevée pour une configuration qui prolonge les contours, dans le cas des coins et des "jonctions en T".

- **Bout** est un booléen correspondant à la notion de “terminaison”, prenant la valeur **un** au site **s**, si ce site est situé en fin de chaîne, et zéro sinon.
Un bout est donc un pixel ayant un ou deux voisins adjacents. Si **s** a deux voisins adjacents **s1** et **s2**, la condition pour que **s** soit un bout est la suivante : $(\mathbf{s1}, \widehat{\mathbf{s}}, \mathbf{s2})$ doit être un angle aigu.
- Pour chaque chaîne **ch**, la valeur de $\mathbf{D}_k(\mathbf{s}, \mathbf{ch})$ (notée $\mathbf{D}_k(\mathbf{s})$ s’il n’y a aucune ambiguïté) **détermine une mesure de vraisemblance de la direction k**. Cette valeur est calculée en faisant une moyenne des mesures de vraisemblance locales, sur l’ensemble des sites de la chaîne.

$$D_s = D_{(s, ch_s)} = D(s, ch_s) = (D_0(s, ch_s), D_1(s, ch_s), D_2(s, ch_s), D_3(s, ch_s))$$

- Nous définissons quatre constantes positives C_0, C_1, C_2 et C_3 , qui conservent leurs valeurs durant l’algorithme de restauration, et pour tout type d’images. Nous les choisirons par la suite d’après des contraintes obtenues par la méthode des boites qualitatives (nous verrons plus loin que l’on peut choisir $C_3 = 2 * C_1$).
- $K_1 = C_1 * (E_p + B_r) + C_3 * Bout$

La variable K_1 est donc relativement grande dans plusieurs situations :

- Si le contour est épais.
- Si les coins ou les “jonctions en T” sont instables.
- Si le site **s** étudié est une terminaison de chaîne.

- Le champ de Markov défini sur **S** a pour espace d’états Λ , dont les éléments sont des vecteurs x_s de la forme :

$$x_s = (ch_s, D_s, st_s) :$$

On a : $x_s = (-1, -1, 0)$ lorsqu’aucun contour ne passe par le site **s**.

4.2 Définition de la fonction d'énergie

La fonction d'énergie U est la somme sur toutes les cliques de trois fonctions de potentiels V_1 , V_2 et V_9 , définies respectivement sur des cliques d'ordre 1, 2 et 9.

Une définition des trois fonctions de potentiels est présentée dans les pages suivantes.

– Fonction de potentiels V_1 :

En chaque site s , la fonction V_1 dépend uniquement des caractéristiques de s . Plus précisément, V_1 permet d'effectuer un rappel aux données initiales, c'est-à-dire de préserver les contours détectés initialement par le filtrage de Canny-Deriche ([2], [4]), et de prendre en compte le type de la région comprenant le pixel s (région homogène ou hétérogène).

$$V_1(x_s) = \begin{cases} C_0 * Im(s) + C_1 & Si (pds > 0) et si x_s = (-1, -1, 0). \\ C_0 * Im(s) & Si (pds = 0) et si x_s = (-1, -1, 0). \\ C_1 & Si (pds = 0) et si x_s \neq (-1, -1, 0). \\ 0 & Sinon \end{cases}$$

– Fonction de potentiels V_2 :

La fonction de potentiels V_2 prend en compte la direction des chaînes de contour, définie par le vecteur à quatre composantes réelles $\mathbf{D}(s, \mathbf{ch})$:

$$D_s = D(s, ch) = (D_0(s, ch), D_1(s, ch), D_2(s, ch), D_3(s, ch))$$

où \mathbf{ch} représente la chaîne de contour passant par le site s .

V_2 utilise en fait uniquement la composante $\mathbf{D}_{\mathbf{k}}(s, \mathbf{ch})$ du vecteur $\mathbf{D}(s, \mathbf{ch})$, où $\mathbf{D}_{\mathbf{k}}(s, \mathbf{ch})$ mesure la vraisemblance de la direction \mathbf{k} (\mathbf{k} appartenant à l'intervalle $[0..3]$), où \mathbf{k} est la direction de la droite $(s, s1)$ passant par s et un site voisin $s1$.

On a donc :

$$D_k(s, ch) = D_{(s,s1)}(s, ch)$$

Les contours sont donc prolongés dans une direction compatible avec la direction des contours des sites voisins.

$$V_2(x_s, x_{s1}) = \begin{cases} C_3 & Si \left((x_s \neq (-1, -1, 0)), (x_{s1} \neq (-1, -1, 0)) \right. \\ & \left. (ch_s = ch_{s1}) \text{ et } (D_{(s,s1)}(s, ch) < seuil) \right) \\ C_1 + C_2 & Si \left((x_s \neq (-1, -1, 0)), (x_{s1} \neq (-1, -1, 0)) \right. \\ & \left. (ch_s \neq ch_{s1}) \text{ et } (D_{(s,s1)}(s, ch) < seuil) \right) \\ 0 & Sinon \end{cases}$$

Pour un site donné, plus la valeur de la composante D_k est grande (k appartenant à l'intervalle $[0..3]$), plus la probabilité de prolonger le contour dans la direction k est élevée, et inversement.

– **Fonction de potentiels V_9 :**

La fonction de potentiels V_9 en un site s est déterminée en fonction du voisinage de s . En effet, V_9 dépend du nombre de contours voisins (“**nb-vois**”).

$$V_9(x_s) = \begin{cases} C_1 + C_2 & Si \left((x_s \neq (-1, -1, 0)) \text{ et } (nb - vois = 0) \right) \\ K_1 & Si \left((x_s \neq (-1, -1, 0)) \text{ et } (nb - vois \neq 0) \right) \\ \infty & Si \left(ch_s \neq ch_{si} \right) \quad \forall i \in [1..8] \\ 0 & Sinon \end{cases}$$

Quelques remarques sur ces fonctions de potentiels :

– **Remarques concernant la fonction V_1 :**

La probabilité d’avoir un état “pas de contour”, lorsqu’un contour a été initialement détecté est faible. Ce qui signifie que l’énergie doit être

assez élevée dans ce cas de figure. La conséquence immédiate est de conserver les données initiales. C'est pour cette raison que l'on affecte à V_1 la valeur $C_0 * Im(s)$, si l'état de s est non nul.

On pose la contrainte suivante: C_0 doit être une constante "suffisamment grande" (cf. §5).

La probabilité d'avoir un état "pas de contour" dans une zone non homogène est faible (donc l'énergie élevée), car la prolongation des contours s'effectue dans des zones hétérogènes.

C'est pourquoi, on introduit la constante C_1 dans le premier cas considéré dans la définition de la fonction V_1 .

Inversement, dans une zone homogène, la probabilité d'avoir un état "contour" est faible, ceci pour ne pas prolonger les contours dans des régions homogènes (troisième cas de la fonction V_1).

La fonction V_1 ainsi définie, nous permet d'effectuer un rappel aux données (à partir des contours initialement détectés et de la segmentation en régions).

Le fait d'avoir un contour dans une zone non homogène ne doit pas être pénalisé (quatrième cas).

Pour les autres fonctions de potentiels, nous prenons en compte d'autres informations comme la direction des chaînes voisines, l'épaisseur du contour, etc.

– **Remarques sur la définition de la fonction V_2 :**

Cette énergie prend en compte le site s et son voisin $s1$. Si l'état associé à l'un des deux sites est nul, alors V_2 prend la valeur nulle.

V_2 est égale à zéro lorsque la direction de la droite $(s,s1)$ est compatible avec la direction moyenne de la chaîne ch passant par s (donc si $D_{(s,s1)}(s, ch) \geq \text{seuil}$).