Thèse de Doctorat

Specialité Informatique

Une approche incrémentale à base de processus coopératifs et adaptatifs pour la segmentation des images en niveaux de gris

Fabrice Bellet

<Fabrice.Bellet@imag.fr>

1 – Plan

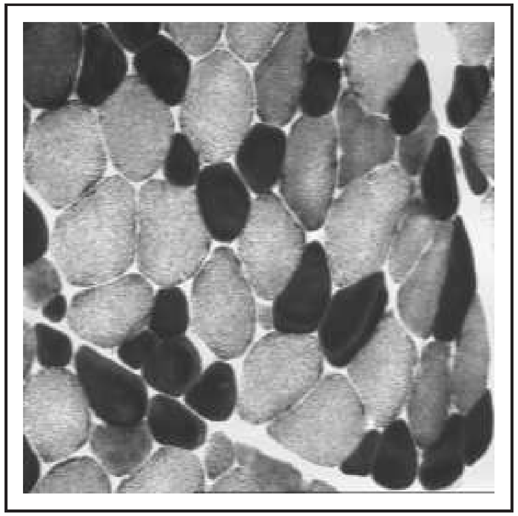
- Introduction
- Les méthodes de segmentation en vision
- Principes de coopération en vision
- Principes des systèmes d'exploitation pour la vision
- Mise en application
- Intérêt et limites

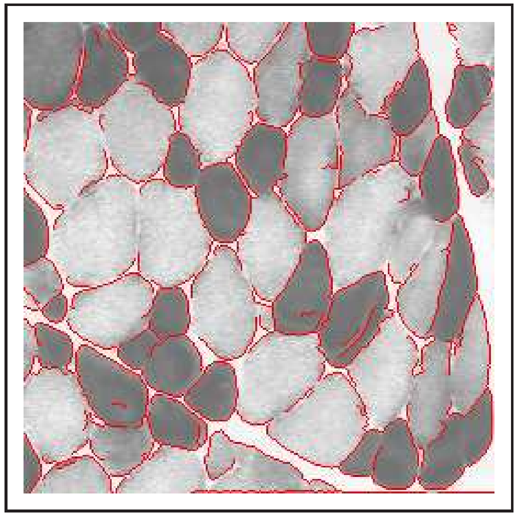
2 – Introduction

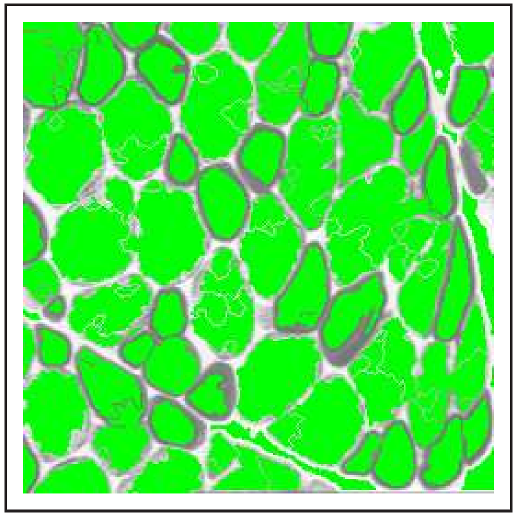
2.1 – Domaine de la vision de bas niveau

- Images en niveaux de gris : caractéristiques photométriques
  (Gros volume d'information, non structurée, bruitée, dégradée);
- La vision de bas niveau
  (Pas de connaissances sur le domaine et le contenu des images);
- Extraction des caractéristiques de l'image : exploration d'un environnement
  (Contours : les changements *visibles* d'intensité [gradient])
  (Régions : regroupement de pixels homogènes [écart-type local]);
- Domaine difficile. (Difficulté de modélisation, problèmes de prise de décision, gestion de l'information).

Exemple : image de cellules musculaires en coupe.







2.2 – Approche coopérative

- Construire les contours **ET** les régions en même temps;
- dualité : amélioration de la qualité individuelle;

2.3 – Approche incrémentale

- Construction pixel par pixel;
- Fonction de choix du pixel suivant (multi-critères, contrôle puissant).

2.4 – Approche multi-processus

- Distribuer les traitements dans l'image;
- Individualiser les tâches (processus contour, processus région);
- Localiser les tâches (pixel de démarrage);
- Paralléliser les tâches (séquenceur de tâches anonymes, type OS).

3 – Méthodes de segmentation en vision

3.1 – Boîte noire vs. points de contrôle

Les méthodes incrémentales ont beaucoup de points de contrôle. Elles sont en opposition aux méthodes de type filtrage, caractérisées par :

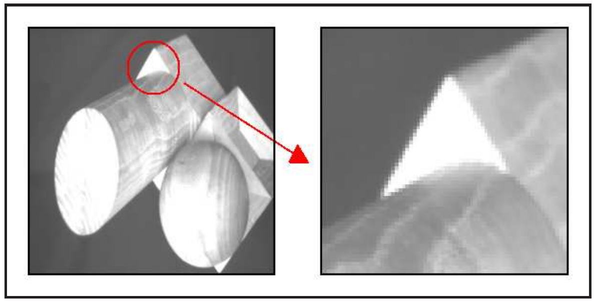
- ✔ peu de paramètres à ajuster
- ✔ leur rapidité
- ✔ les possibilités de mise en pipeline
- ✔ leur aspect rassurant pour l'utilisateur
- ✘ peu/pas de contrôle local
- ✘ un résultat obtenu par essai/erreur

3.2 – Intérêt des méthodes incrémentales

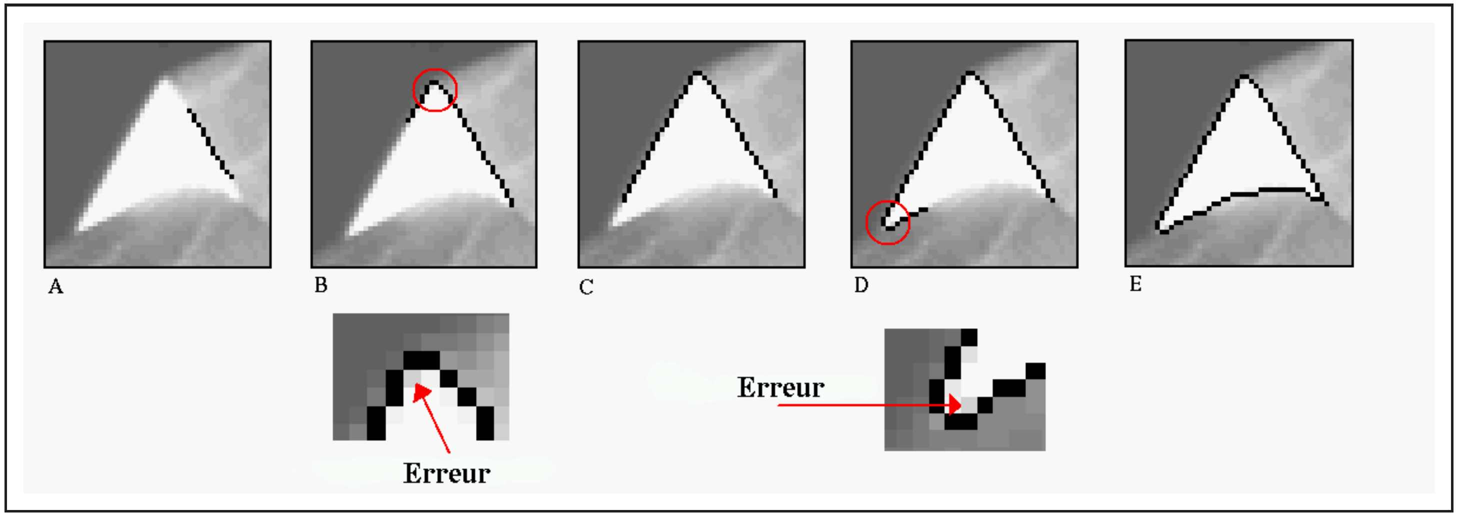
- La solution est construite progressivement (ici pixel par pixel);
- Le contrôle : le choix du pixel suivant (multi-critères);
- Gestion efficace de l'information (prendre des décisions motivées par l'information existante, utiliser le maximum d'information disponible, créer l'information manquante au besoin);
- Forte adaptation (fonctions de choix du pixel paramétrables, décisions par rapport à l'image, et aux primitives environnantes paramétrables).

3.3 – Exemple de contrôle incrémental

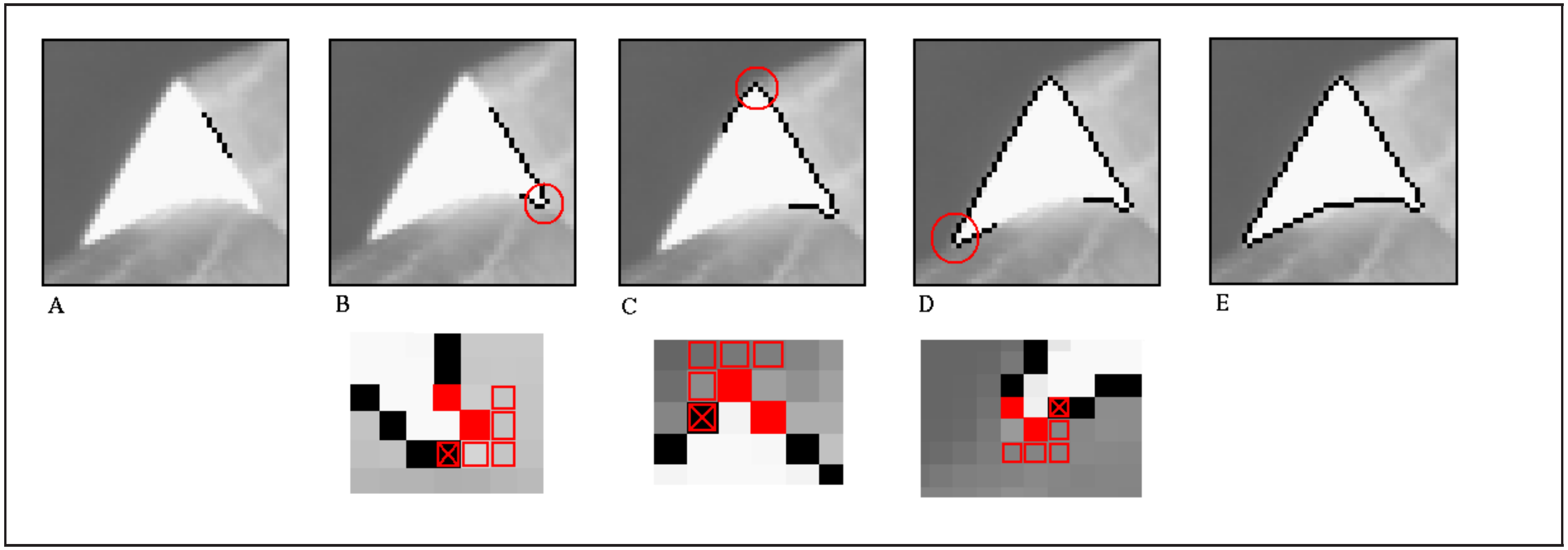
Le nombre de pixels examinés aux extrémités fait partie des points de contrôle.



Exemple avec 3 pixels testés.



Exemple avec 5 pixels testés.

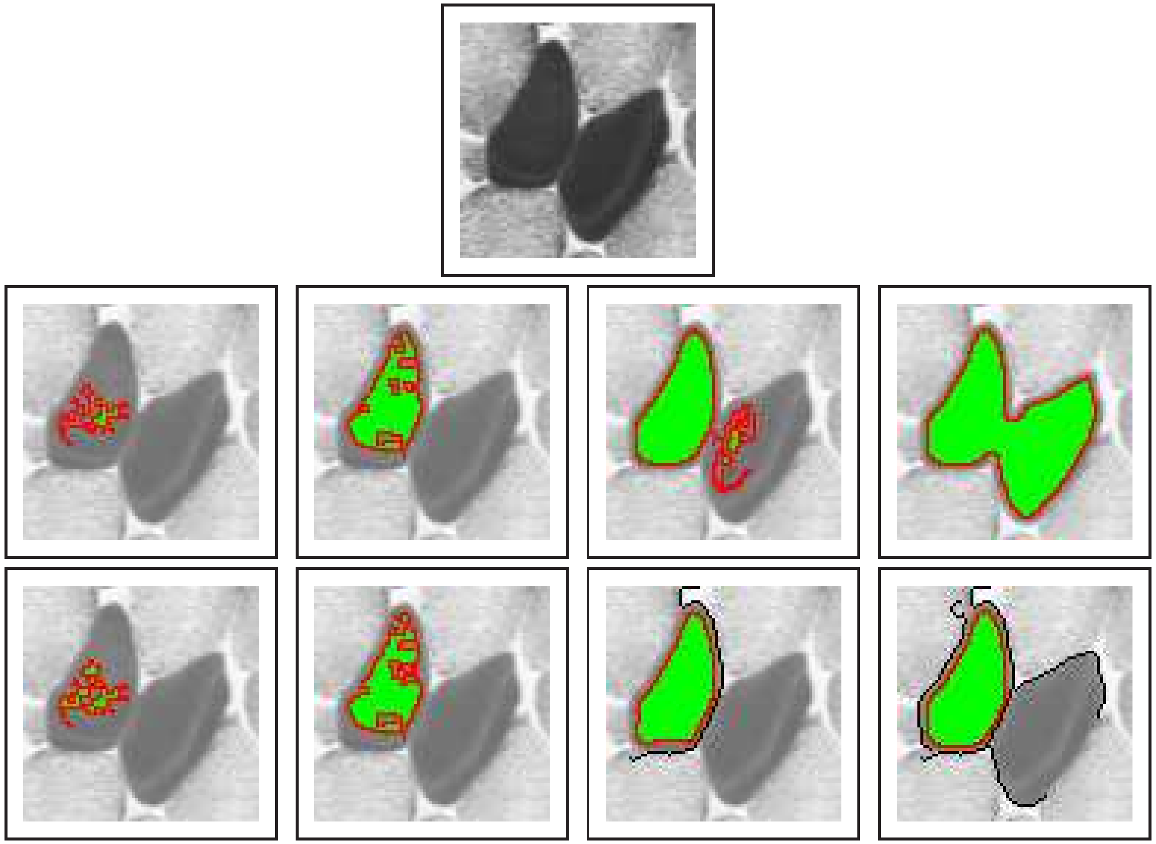


4 – Principes de coopération

4.1 – Objectifs

- Diviser le problème en sous-tâches (déléguer, *diviser pour régner*);
- Entrelacer les tâches coopérantes
  (renoncer à l'enchaînement séquentiel des méthodes);
- Définir un protocole de coopération
  (quelles interactions, entre quelles tâches, à quel moment, dans quel but);
- Rendre réactif (à l'environnement, aux autres tâches);
- Ne pas craindre les effets de bord
  (effets émergents, non explicitement programmés);
- Lien étroit avec la parallélisation
  (problèmes liés : accès aux ressources, exclusion mutuelle, synchronisation ...).

4.2 – Exemple



5 – Un système d'exploitation en vision

Certaines méthodes utilisées dans les systèmes d'exploitation permettent d'implémenter ces principes de coopération.

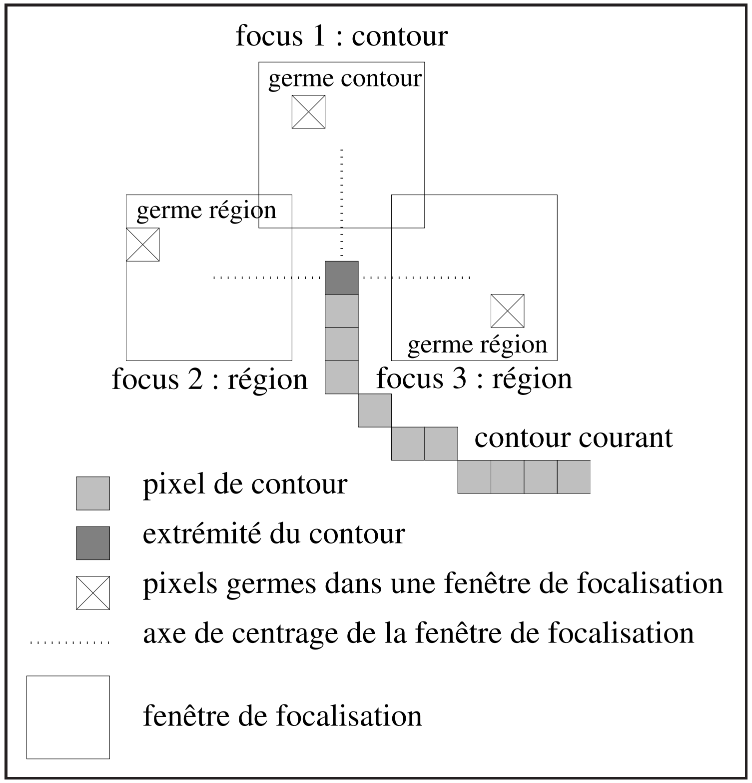
- Un séquenceur de tâches;
- Des tâches *dynamiques* (création, destruction, interaction);
- Des tâches utilisant des ressources communes (l'image : l'étiquetage des pixels);
- Des tâches qui doivent se synchroniser (échange de messages, de signaux, des liens de filiation).

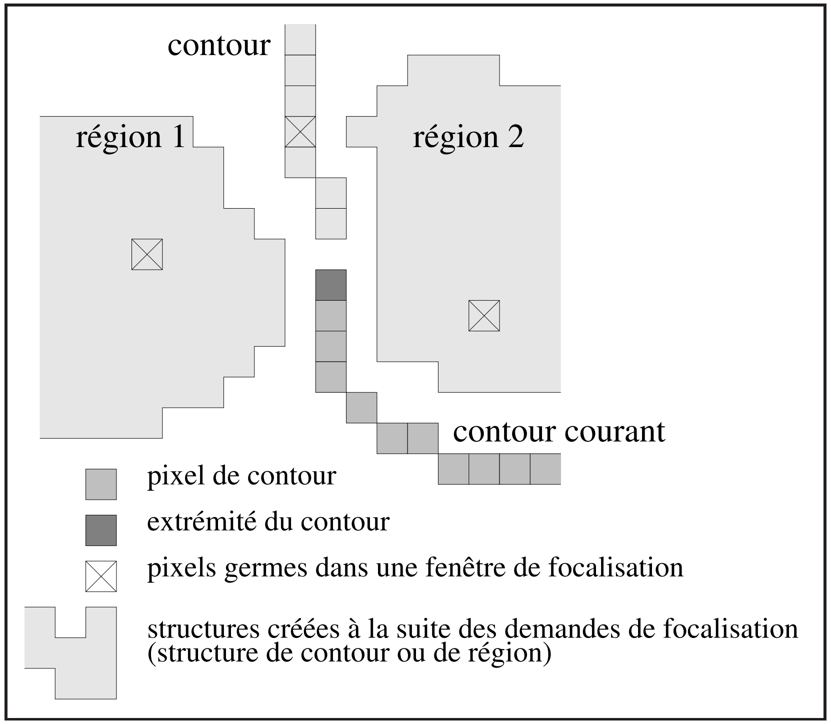
6 – Mise en application

Description du système de segmentation réalisé.

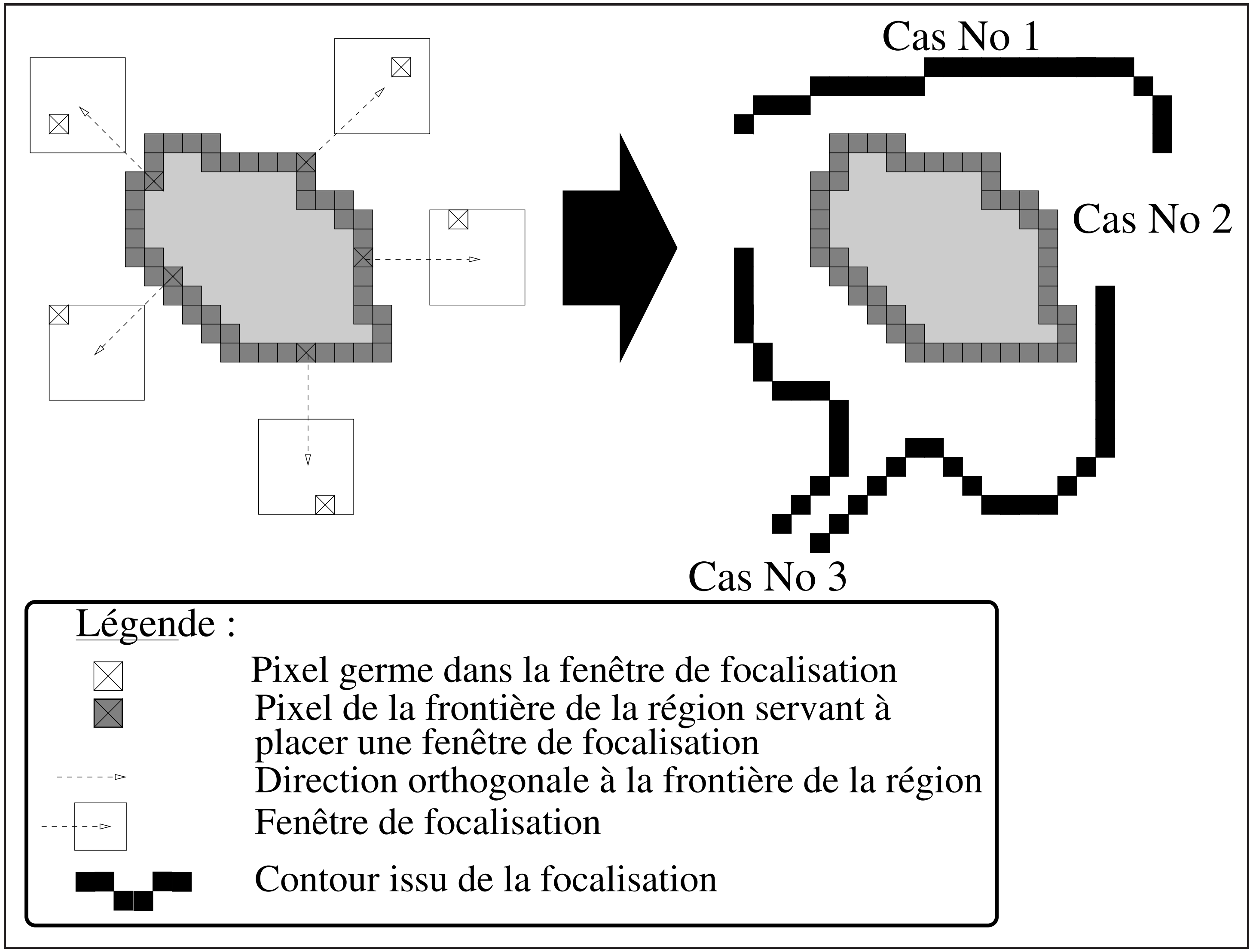
- Un environnement (l'image, des pixels étiquetés : contour, région);
- Des tâches :
  - de détection de contours/régions;
  - incrémentales;
  - localisées dans l'image (pixel germe = pixel initial);
  - éphémères.

- Des interactions entre les tâches :
  - Les contours utilisent les régions;

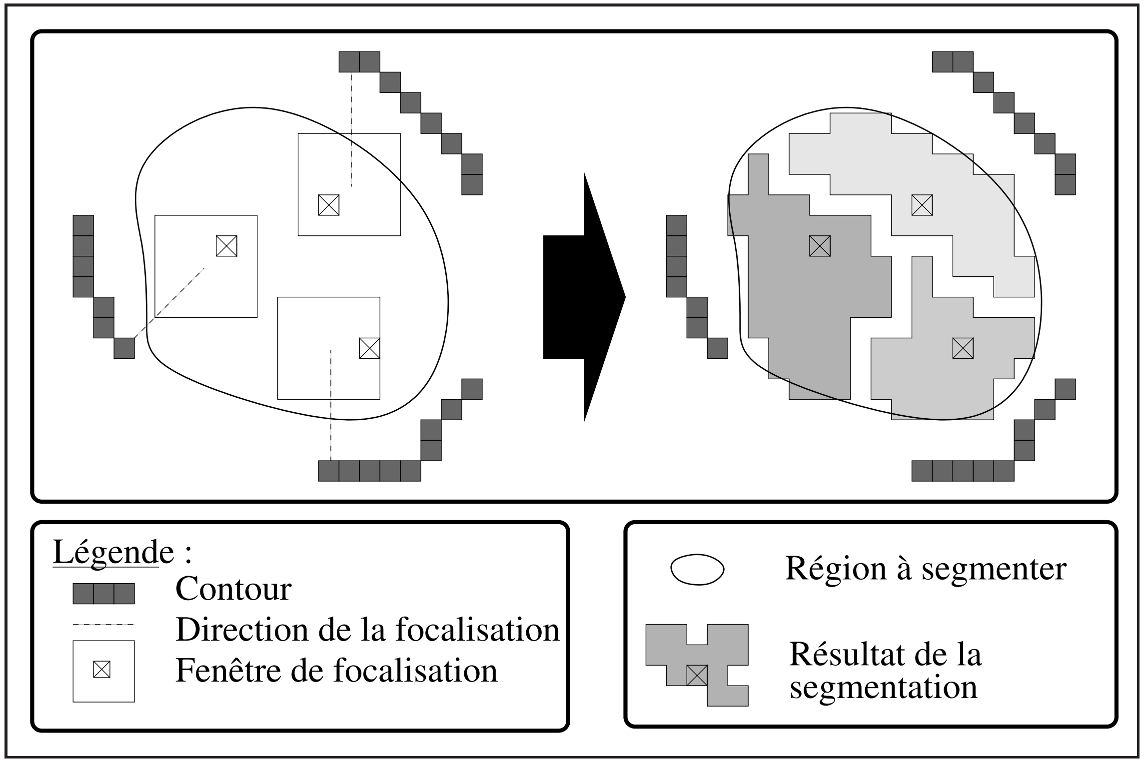


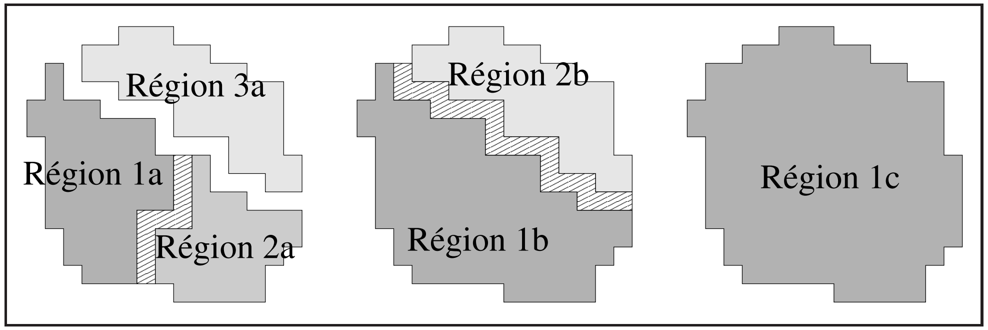


➪ Les régions utilisent les contours;

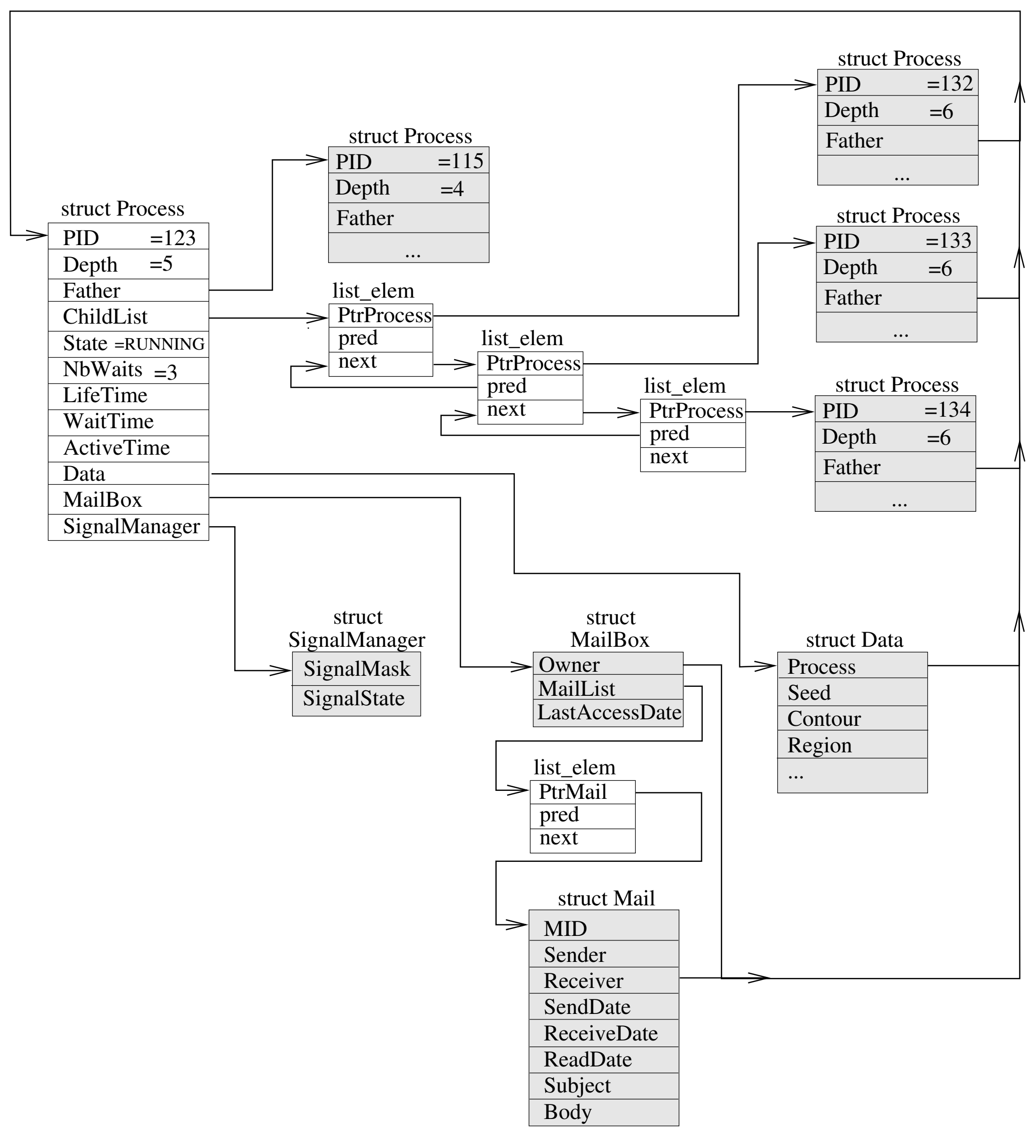


➪ Les fusions de primitives.

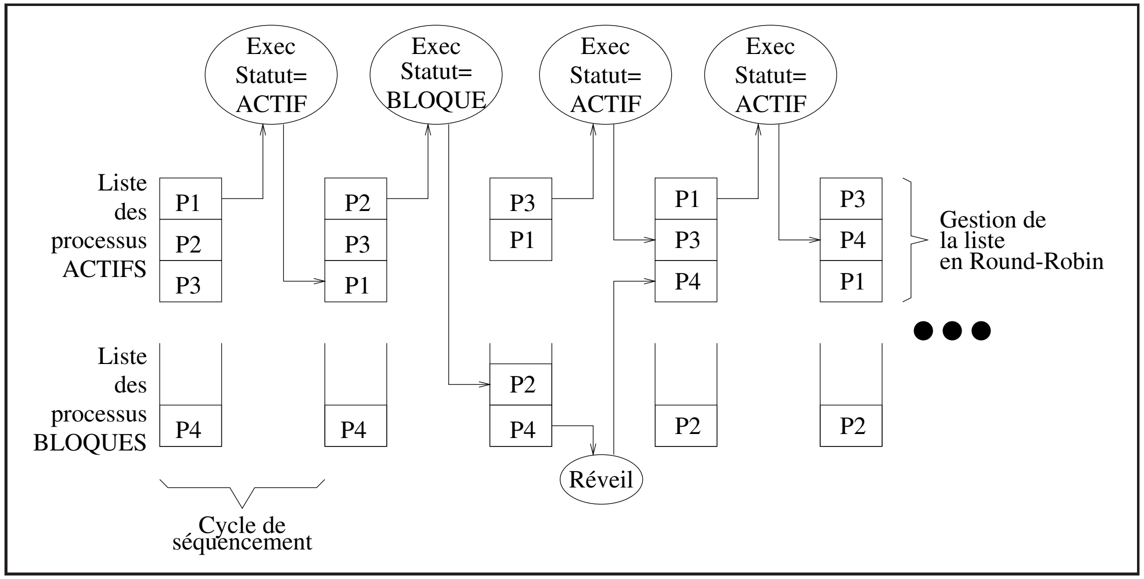




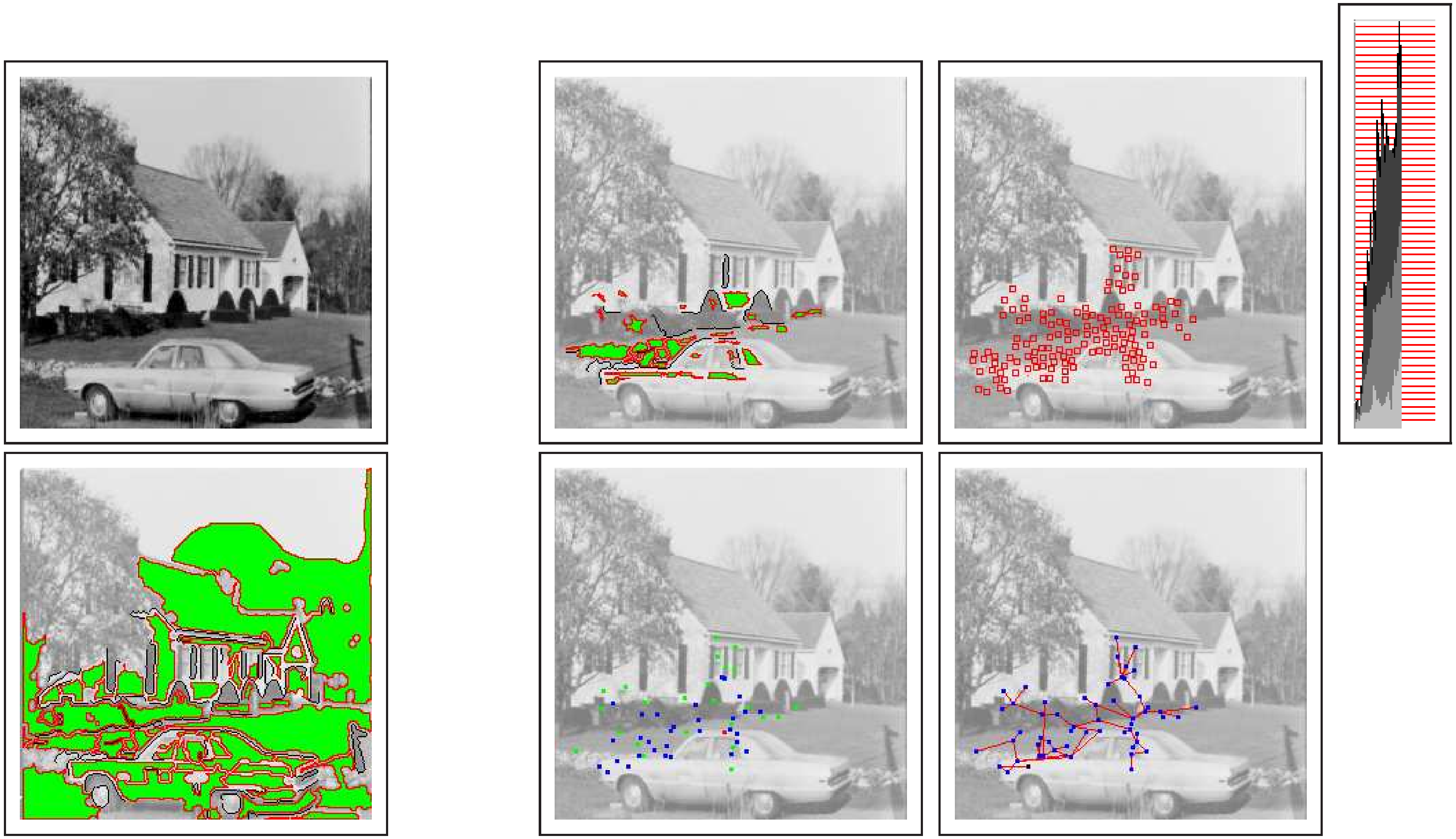
- Un séquenceur de tâches banalisées :
  - Séparation des espaces de mémoire des processus/tâches;


struct Process
PID =123
Depth =5
Father
ChildList
State =RUNNING
NbWaits =3
LifeTime
WaitTime
ActiveTime
Data
MailBox
SignalManager
struct Process
PID =115
Depth =4
Father
...
list_elem
PtrProcess
pred
next
list_elem
PtrProcess
pred
next
list_elem
PtrProcess
pred
next
struct Process
PID =132
Depth =6
Father
...
struct Process
PID =133
Depth =6
Father
...
struct Process
PID =134
Depth =6
Father
...
struct SignalManager
SignalMask
SignalState
struct MailBox
Owner
MailList
LastAccessDate
struct Data
Process
Seed
Contour
Region
...
list_elem
PtrMail
pred
next
struct Mail
MID
Sender
Receiver
SendDate
ReceiveDate
ReadDate
Subject
Body

- Deux listes de processus (processus *actifs* et *en attente*);



- Une politique de l'élection du processus actif;
- Des priorités entre les processus.



7 – Intérêt et limites

7.1 – Protocole expérimental

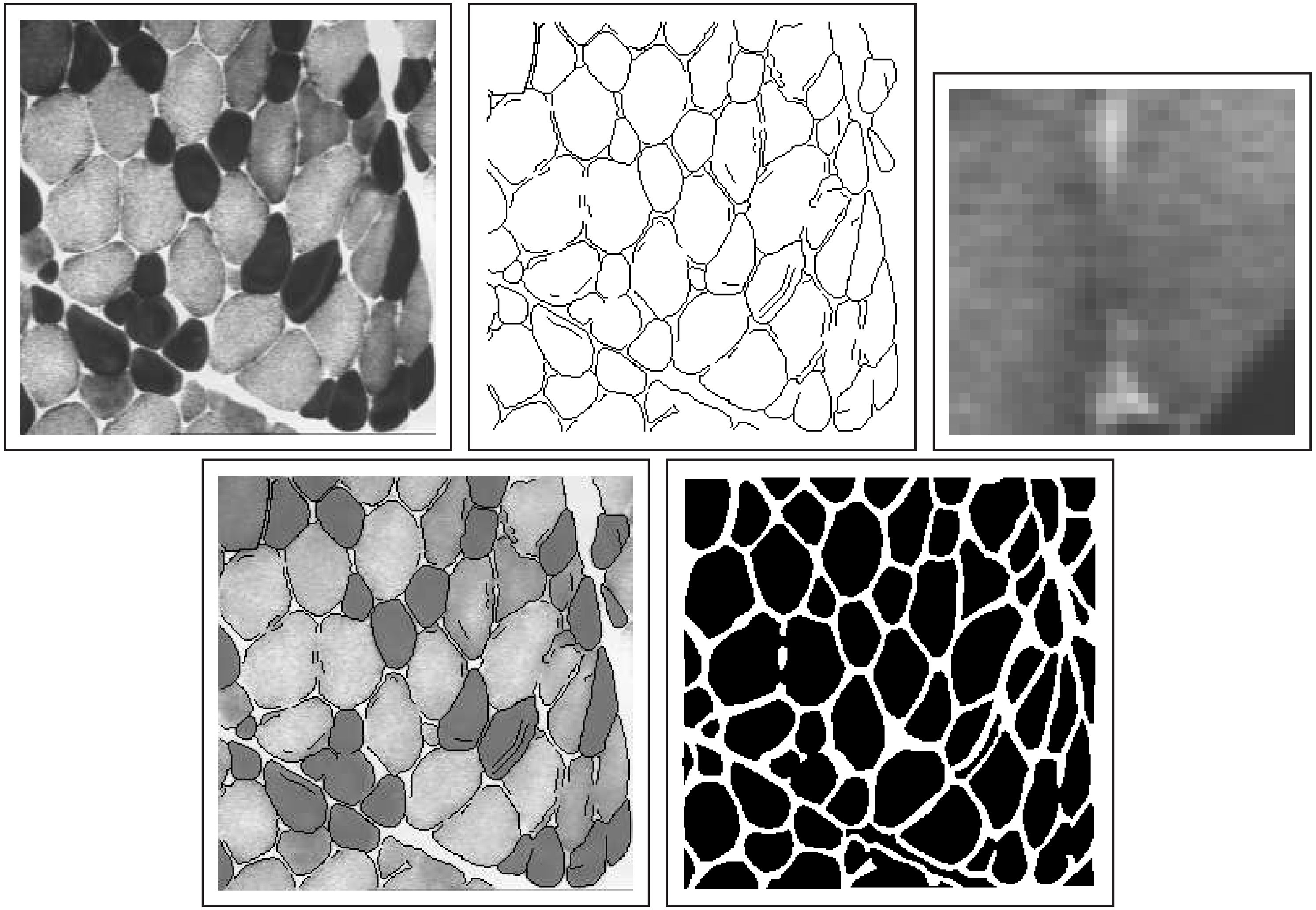
Des questions ...

- ☞ Un protocole expérimental générique au bas niveau ? (nécessité d'uniformiser les entrées et les sorties);
- ☞ Des mesures qualitatives ou quantitatives ?
- ☞ Validation sur des images naturelles ou de synthèse ?

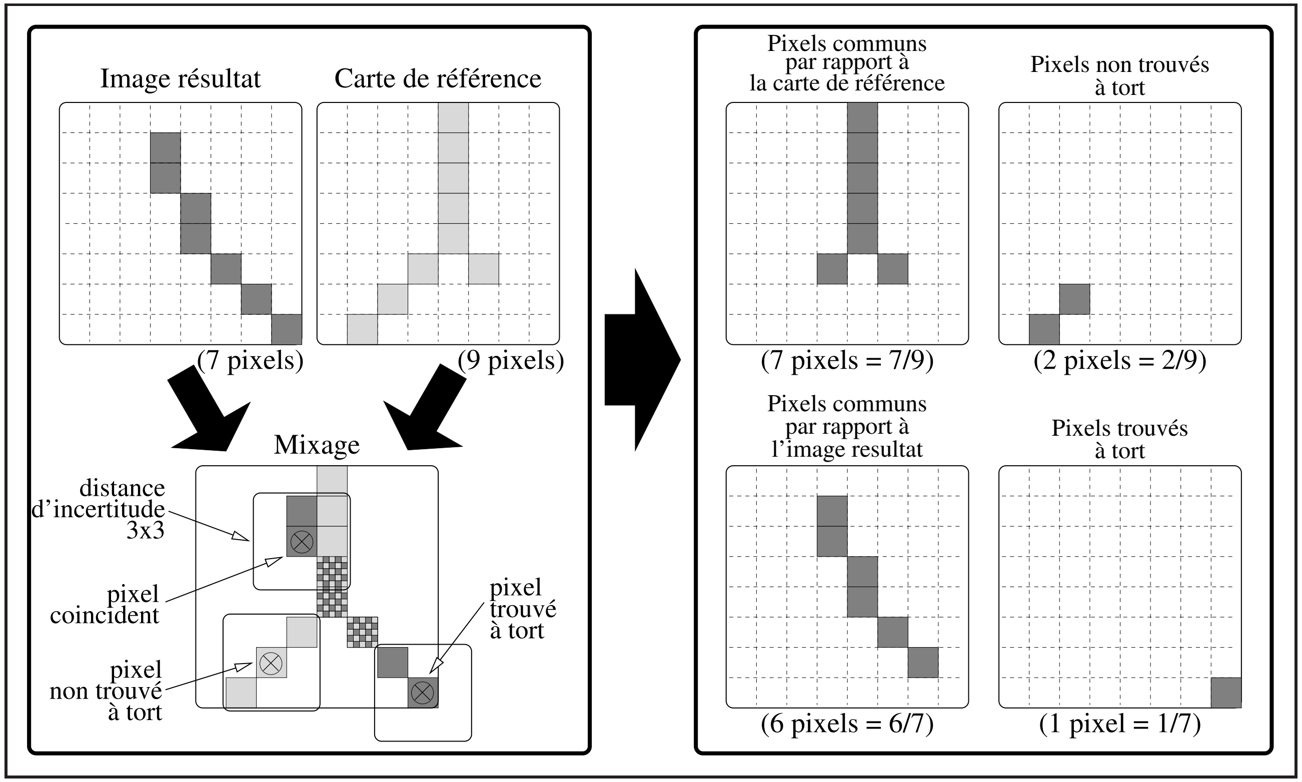
Des idées de réponse ...

- ☞ Des contours/régions de référence sur des images naturelles (pas uniquement sur des images de synthèse);

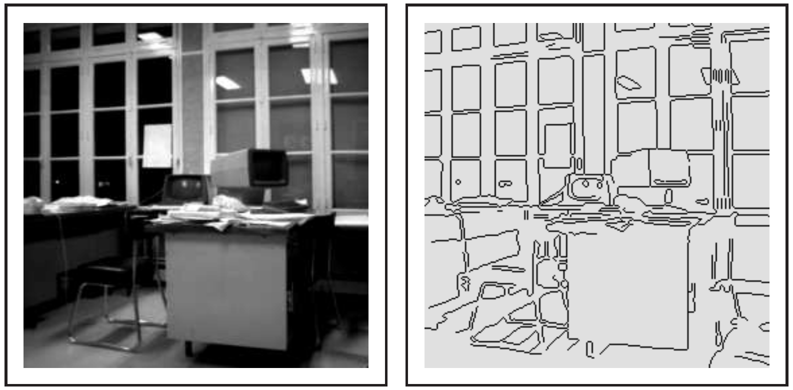
7 – Intérêt et limites



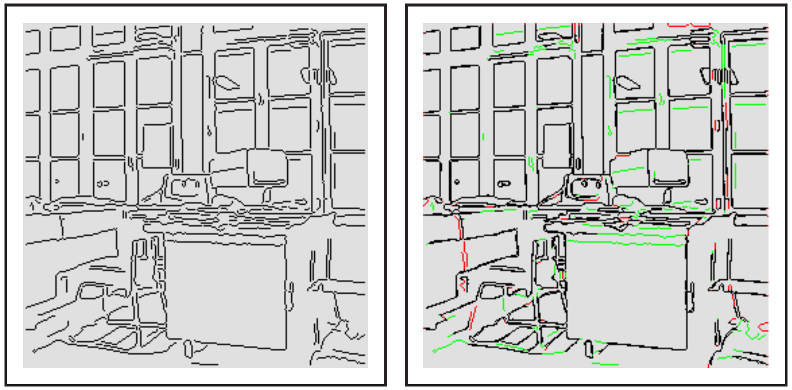
- Une évaluation quantitative à plusieurs niveaux (pixel, contour, région);
- Distances d'incertitude.



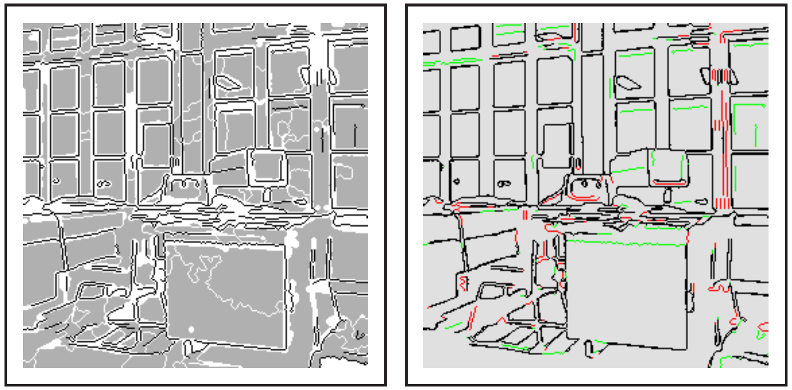
7.2 – Comparaison avec la méthode de Deriche



	Non trouvés	En trop
Deriche	5.7 %	15.1 %
Coopératif	11.0 %	12.1 %

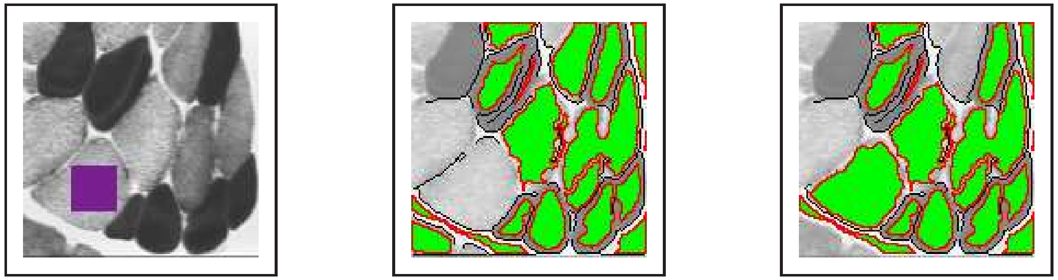


Deriche

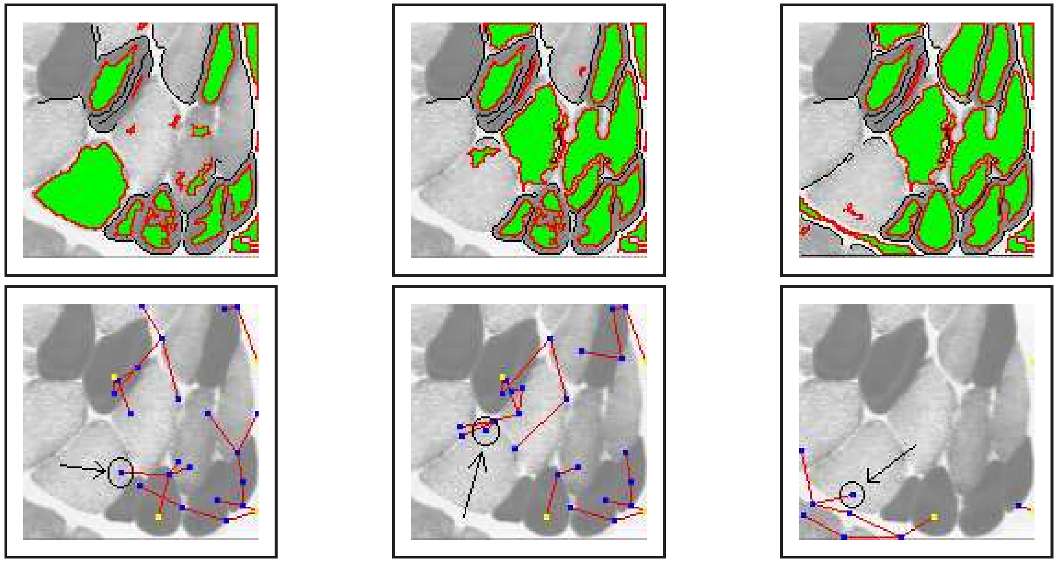


Coopératif

7.3 – Comportement dynamique

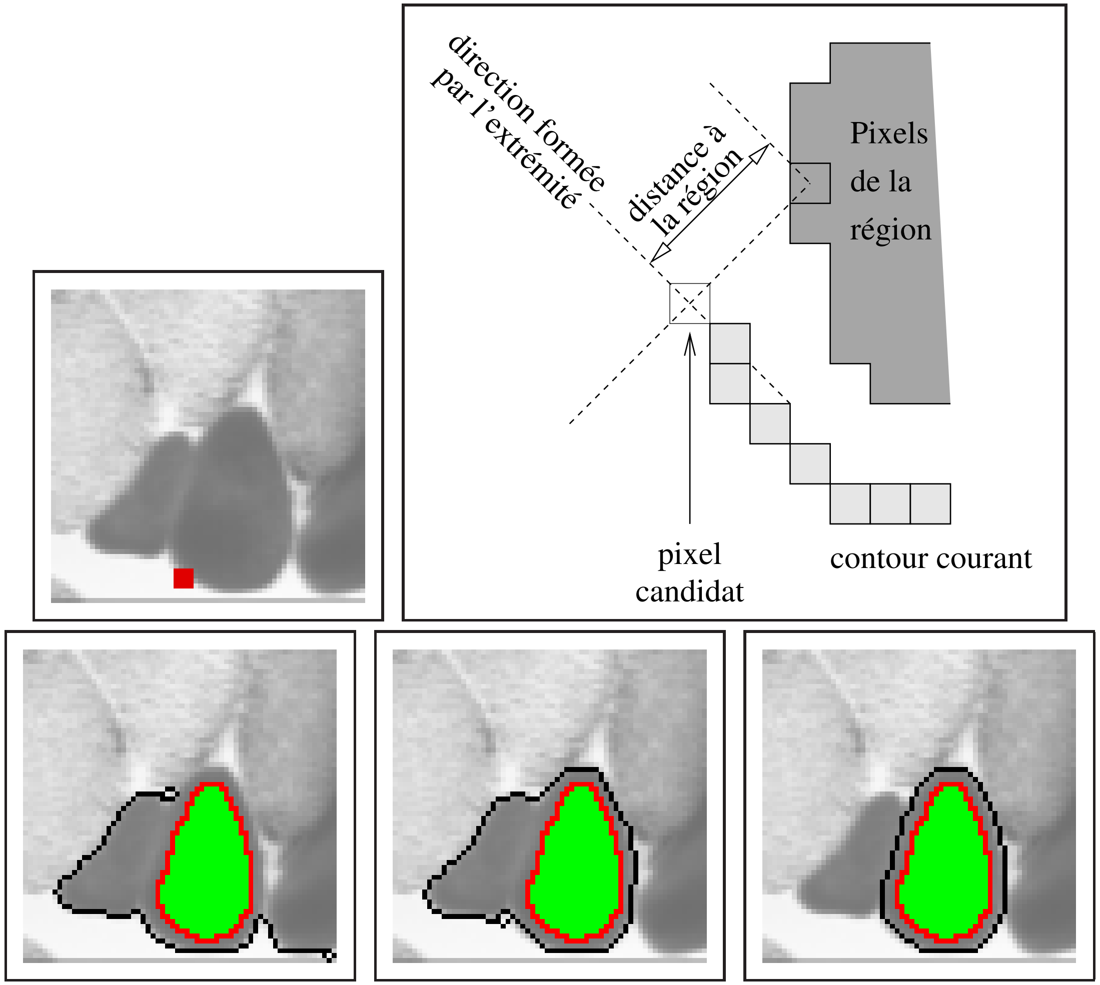


zone bloquée final à t=1529s final à t=1034s



t=346 s t=665 s t=894 s

7.4 – Comportement adaptatif



8 – Conclusions

Perspectives ?

- Véritable parallélisation;
- Extraire de nouveaux types d'information de l'image;
- Spécialiser (introduire de la connaissance de haut-niveau);
- Formaliser la coopération (base de règles);

L'abondance de points de contrôle nuit-il à l'intérêt d'une méthode de segmentation ?

- Problématique en général, **MAIS**
- Richesse des choix et des possibilités de décision;
- Des valeurs par défaut pour les paramétrages;
- Trier les points de contrôle par ordre d'importance.
- Apprentissage pour diminuer les points de contrôle.

Liste des transparents