

Une approche opportuniste et coopérative pour la vision de bas niveau

An Opportunist and Cooperative Approach for Low Level Vision

par Fabrice BELLET *, Marc SALOTTI **, Catherine GARBAY *

* Lab. TIMC / IMAG, Institut Bonniot, Fac de médecine, Domaine de la Merci, 38706 La Tronche Cedex, France

Tel. : (33) 76 54 94 90, Fax : (33) 76 54 95 49

(email : Fabrice.Bellet@imag.fr)

** Groupe VISIA, CMCS

Université de Corse, BP 52, 20250 Corte, France

(email : Marc.Salotti@univ-corse.fr)

Résumé

Nous présentons une nouvelle approche de la coopération entre plusieurs méthodes de segmentation. L'idée est d'introduire la coopération dans le processus de décision. Chaque méthode de segmentation peut alors prendre des décisions adaptées, différer les décisions difficiles et résoudre les problèmes délicats en demandant et en accumulant les informations fournies par d'autres méthodes. Dans ce but, chaque méthode comporte une structure de contrôle incrémentale, interrompible à tout moment pour demander l'aide opportune d'autres méthodes : des processus « fils » sont créés aux endroits critiques dès qu'une situation complexe est rencontrée, puis des informations sont retournées au processus père qui est alors en mesure de prendre des décisions plus fiables. Les processus sont contrôlés par un séquenceur, dans l'esprit d'un système d'exploitation multi-tâches. Ce séquenceur initialise des processus à différents endroits de l'image. Ensuite, la stratégie de parcours en largeur d'abord ou en profondeur d'abord permet de traiter immédiatement ou non les demandes de coopération. Le parcours en largeur d'abord implique l'analyse des problèmes les plus simples avant ceux qui sont plus complexes. Nous illustrons les différents choix sur des exemples variés.

Mots clés : Vision de bas niveau, coopération opportuniste, segmentation incrémentale, détection des contours et des régions.

Abstract

This paper presents a new approach for the design of cooperative segmentation systems. The idea is to introduce cooperation as an integrant part of the decision process. Each segmentation method is then given the ability to make adaptive decisions, to postpone difficult decisions and to solve pending problems by requesting and accumulating information. For this purpose, each method is implemented as an incremental process that can interrupt itself at any time to ask for a cooperation with other processes, in an opportunist way : "child" processes are then created at certain locations in the image. Child processes are thus created each time a "complex" situation is encountered, to gather more information which, collected at the "parent" level, is used to take "better" decisions, in a more secure way. The processes are controlled by a scheduler, like in a multi-task operating system. Depth-first or breadth-first control strategies may be implemented, where information queries are treated immediately or deferred. The potential interest of the approach is illustrated on a variety of examples.

Key words : Low level vision, opportunist cooperation, incremental segmentation, edge and region detection.

1. Introduction

Nous définissons la segmentation par une description en primitives simples de l'image, par exemple les régions, correspondant aux zones de même homogénéité, et les contours, marquant les transitions entre des zones d'homogénéité différente. L'utilisation de connaissances *a priori* sur le contenu de l'image est ici exclue, ce qui élimine toutes les techniques fondées sur une approche descendante du problème, ou centrées sur des modèles. La segmentation proposée consiste en un étiquetage des pixels de l'image en trois classes : *pixel région*, *pixel contour*, ou *pixel non*

affecté. L'image résultat ne prétend pas fournir une segmentation complète [28], qui force souvent à commettre des erreurs d'étiquetage, mais tente plutôt d'extraire des primitives significatives et correctement localisées. L'expertise humaine valide actuellement les résultats de notre algorithme.

Les approches coopératives de la segmentation peuvent être classées en deux catégories : les méthodes *orientées modèle*, et les méthodes *orientées contrôle*. Parmi les techniques de la première classe, Haddon et Boyce [10] effectuent une segmentation de l'image en régions par une méthode de partitionnement statistique de la matrice de cooccurrence, les pics diagonaux correspondant aux régions, et les autres pics de la matrice se rappor-

tant aux frontières. L'information de type contour est utilisée dans une approche de relaxation. Cette approche diffère de la nôtre par l'aspect de classification, n'autorisant pas que des pixels restent non étiquetés. De plus, ce mécanisme est suivi d'une étape de raffinement liée à la configuration locale dans le voisinage des pixels, laissant à penser que l'étape antérieure est un peu trop globale. Les stratégies *orientées contrôle* constituent la majorité des techniques proposées, l'idée de base étant que la détection des contours peut aider à contrôler l'expansion ou la fusion des régions, tandis que la détection des régions contribue à la localisation et à la fermeture des contours ([1], [3], [4], [5], [7], [8], [15], [17], [20], [26], [27]). Pour une première sous-classe de ces méthodes, le but est d'assister une méthode de croissance de région par des informations de type contour. La méthode de croissance de région est de type *Split And Merge*, ou de type croissance incrémentale pixel par pixel. Par exemple, Wrobel et Monga proposent d'appliquer d'abord un détecteur de contours, puis de contraindre une méthode de fusion des régions en prenant en compte la présence des contours à leurs frontières [26]. Benois et Barba [4] utilisent une carte précalculée des contours robustes de l'image pour contraindre une méthode de segmentation en régions de type *Split And Merge*. Le problème de l'ordre de fusion des régions est résolu à l'aide de la carte des contours : la fusion privilégie les zones du *quad-tree* les plus éloignées de tout pixel contour (présument les plus homogènes). Kara Falah *et al.* [8] proposent une coopération région-région, ayant pour but de créer des « noyaux » de régions robustes par superposition de plusieurs cartes régions différentes. L'expansion de ces noyaux de régions se fait ensuite sous le contrôle de l'information gradient, selon une coopération région-contour. Suivant le même principe, Xiaohan *et al.* [27] intègrent directement l'information sur la maximalité du gradient comme un critère de sélection des pixels dans une croissance de région classique. Une deuxième sous-classe de méthodes repose sur la dualité entre les informations de type région et de type contour. Pavlidis propose d'appliquer d'abord une méthode de croissance de régions, afin d'obtenir des germes de contours fermés dont la position définitive est déterminée par une technique de déformation de contours (snakes) [20]. Chu et Aggarwal [7] intègrent des informations de type région et contour dans une structure résultat. Les cartes de régions sont converties en cartes contour par simple extraction des frontières, et l'image résultat provient d'un compromis judicieux entre différentes sources d'information de type contour. La dernière sous-classe réunit plus dynamiquement les deux approches. Anderson, Bajcsy et Mintz [1] intègrent effectivement une croissance de région et une détection de contour dans un même processus itératif de raffinement, dont l'objectif est le meilleur compromis entre les deux segmentations relativement à des contraintes utilisateur en nombre de régions par exemple. Cela se traduit dans l'implantation par un mécanisme d'ajustement mutuel des paramètres des deux approches.

La plupart de ces approches présupposent en fait l'existence d'informations bas niveau, par exemple sous la forme d'une carte des contours, et sont focalisées sur la meilleure façon de combiner les résultats de chaque méthode. La coopération est alors

considérée comme une sorte de règle de raffinement final, au lieu d'être partie intégrante de l'analyse des primitives à détecter. En conséquence, elle est fondée sur des informations déjà calculées qui peuvent être entachées d'erreur.

La coopération devrait être considérée au contraire comme un moyen d'accumuler les informations, à l'aide d'une stratégie opportuniste, et incrémentale, dans le sens où la segmentation solution est construite « pixel par pixel ». Le paragraphe 2 introduit les principes de bases sur lesquels repose le système de coopération proposé. Le paragraphe 3 donne une description détaillée des structures de données et des stratégies de contrôle des processus de bas niveau. Le paragraphe 4 décrit le niveau système avec les techniques de focalisation et de séquençement des tâches. Nous illustrons finalement les principes proposés et les différentes stratégies possibles à travers un ensemble d'exemples.

2. Comment obtenir une coopération efficace

Les faiblesses des techniques de détection de contours et de régions sont d'abord discutées, et les avantages théoriques de la coopération sont soulignés. Les principes de base de notre approche sont ensuite présentés.

2.1. DÉTECTION DES CONTOURS

Selon De Micheli *et al.* [16] et Fleck [9], les différences constatées entre deux cartes de contours obtenues à partir de deux détecteurs différents dépendent peu de la façon de calculer le gradient. Le seuil sur le gradient revêt en revanche une plus grande importance [29]. La plupart des modèles sont trop locaux [6], [21] et de nombreux détecteurs ne sont pas capables de trouver des contours avec faibles gradients sans présenter de fausses réponses en zone texturée ou bruitée [22]. Un pixel contour ne devrait pas être défini uniquement par la force du gradient, mais aussi par la présence et l'arrangement des autres pixels-contours voisins ainsi que par les propriétés des régions voisines [24]. Pour tenir compte des informations sur la continuité de la transition, une solution consiste à effectuer un suivi de celle-ci [14], [24]. Par ce moyen, le seuil sur le gradient peut dépendre du nombre de pixels déjà agrégés à la chaîne. Pour obtenir des informations sur les régions voisines, une coopération avec des opérateurs de segmentation de type région, qui opèrent avant que le processus contour ne prenne sa décision, peut s'avérer judicieuse. Par exemple, si la région voisine est très bruitée, le seuil doit être relativement haut pour que ne soient validés que les contours avec forts gradients, comme le suggère par exemple [24]. A titre d'illustration, la table 1 donne le seuil sur le gradient utilisé dans notre système en fonction de la moyenne des écarts types locaux des régions qui bordent le contour courant. De plus,

dans l'article de synthèse de Pal et Pal [18], une référence à Pavlidis [19] indique : « (visual) image segmentation is a problem of psycho-physical perception, and therefore, not susceptible of purely analytical solution. Any mathematical algorithm usually should be supplemented by heuristics which involve semantic information about the class of images under consideration ». Notre approche est similaire dans le sens où le comportement de notre algorithme repose sur des heuristiques de prise de décision. La différence réside dans le fait que ces heuristiques contiennent une sémantique des contraintes de bas niveau permettant une segmentation pertinente de l'image.

Tableau 1. — Seuil adaptatif du gradient.

$[\sigma]$	1	2	3	4	5
Seuil appliqué	5	5	10	10	30

2.2. DÉTECTION DES RÉGIONS

Les techniques de croissance de régions sont des méthodes efficaces pour obtenir des régions d'une façon progressive [5], [23]. La difficulté majeure est de déterminer les critères pour fusionner et diviser les régions, ou pour agréger les pixels [23]. Deux types d'erreur sont habituellement rencontrés tous les deux étant liés à la difficulté d'adapter la décision au problème traité.

Le premier type apparaît lors de l'analyse des pixels situés près de la frontière d'une région. Il est en effet très difficile de déterminer à quel endroit la croissance de la région doit se terminer, les propriétés locales près des frontières étant nettement différentes des propriétés moyennes de la région, et pouvant biaiser la suite des traitements. Un détecteur de contours peut déterminer la position exacte de la frontière [20]. Dans ce cas, il n'est plus nécessaire de faire croître la région pour qu'elle épouse sa frontière.

Le second type d'erreur est rencontré lorsque deux régions voisines de même nature sont séparées par une frontière faiblement marquée. La plupart des méthodes de croissance de région qui vont effectuer la fusion de ces deux régions en une seule (généralement car les seuils utilisés sont globaux à toute l'image) fourniront au niveau supérieur un objet pratiquement inexploitable par la suite, puisque ses propriétés géométriques et photométriques ne sont plus représentatives d'aucune des deux parties qui le composent. Une gestion rigoureuse des informations et des prises de décision [23], et une croissance s'interrompant régulièrement pour rechercher d'éventuels faibles contours est souhaitable pour éviter ce type d'erreur.

2.3. PROPOSITIONS POUR UNE GESTION EFFICACE DE LA COOPÉRATION

La recherche d'informations complémentaires de type région et de type contour peut faire partie du mécanisme de décision, et être codée directement dans chaque processus de segmentation.

Le problème le plus important est en fait de fournir plus d'informations à une méthode, à bon escient : c'est le *principe de focalisation*, en référence à la nécessité de concentrer l'activité du système sur une zone de l'image particulièrement délicate. Selon ce principe, une méthode doit pouvoir demander localement l'aide d'une autre méthode, attendre les résultats de celle-ci, puis continuer en exploitant les informations nouvellement acquises. La méthode appelée devra fournir des informations *complémentaires et utilisables* par le processus appelant. L'approche présente le double avantage :

- de guider la progression d'un processus particulier;
- de participer à l'enrichissement du résultat global et d'aider par *effet de bord* d'autres processus qui trouveront par la suite cette information déjà disponible.

Les décisions difficiles sont ainsi différées, jusqu'à ce que davantage d'informations soient localement disponibles, pour permettre une décision plus robuste du processus en difficulté. Les principes suivants doivent finalement être respectés :

- différer les décisions problématiques;
- demander la focalisation de plusieurs méthodes en cas de situations difficiles et accumuler les informations locales;
- essayer de prendre une décision adaptée au contexte local.

Chacun des problèmes est ainsi résolu de façon incrémentale, et non planifiée. Les informations sont progressivement obtenues et accumulées, sans conflit, dès lors que celles-ci ont été validées durant leur construction.

2.4. STRUCTURE DE CONTRÔLE DES TRAITEMENTS DE BASE

Chaque traitement bas niveau doit posséder une structure de contrôle incrémentale, qui permet une construction « pixel par pixel » de la solution. Ce traitement doit être capable de démarrer sur un endroit précis de l'image, à partir d'un germe demandé par un processus voisin, de fournir des résultats intermédiaires, de prendre en compte des informations fournies par d'autres processus de segmentation, et de s'arrêter à tout moment, s'il est nécessaire de demander un supplément d'information.

La figure 1 présente de manière synthétique la structure de contrôle des processus et leur modèle de coopération. Après chaque *étape élémentaire de croissance*, une *évaluation locale* est effectuée pour décider d'agréger ou non le pixel courant à la primitive en cours de construction. Si la décision à prendre est simple (c'est-à-dire si l'algorithme est capable de trouver au moins

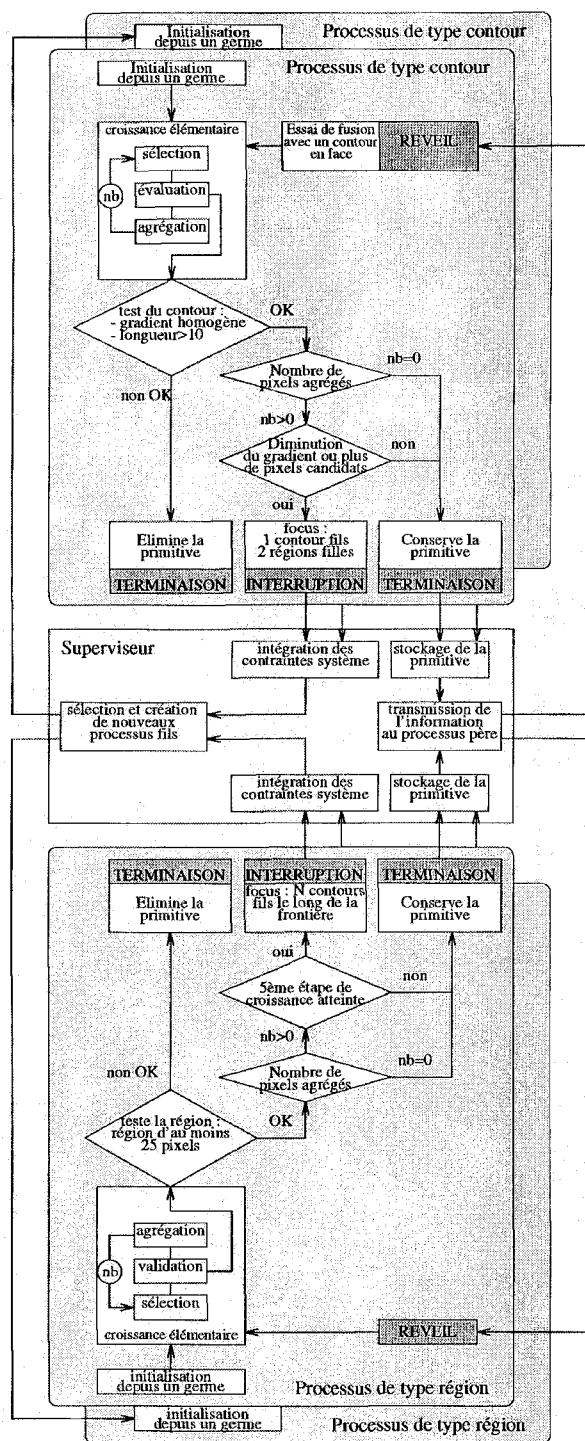


Figure 1. – Structure de contrôle de base.

un pixel à agréger, qui possède une évaluation *satisfaisante* – ce terme sera explicité par la suite), alors la croissance se poursuit. Sinon, la primitive peut être abandonnée, il peut être nécessaire de faire émerger de l'information supplémentaire en émettant des requêtes de focalisations, ou enfin le processus peut terminer son activité en conservant ce qui a été construit. Le processus qui

demande de l'information se met en attente pendant l'exécution des processus fils dédiés à cette tâche. Lorsque le processus père redevient actif, il peut alors utiliser à bon escient l'information acquise dans une étape d'intégration.

Deux méthodes spécifiques au caractère incrémental vont être décrites par la suite, pour la détection de contours et de régions.

2.5. STRUCTURE DE CONTRÔLE AU NIVEAU SYSTÈME

Le rôle d'un séquenceur est d'interpréter les demandes de focalisation en termes de création de processus. Une étape préliminaire du système lance les premiers processus sur les zones les plus prometteuses de l'image. Ces processus contours initiaux sont sélectionnés par défaut sur les pixels ayant le plus fort gradient maximum local, sur l'ensemble de l'image, mais ils peuvent aussi être placés arbitrairement par l'utilisateur. Une carte des gradients est précalculée en utilisant des masques directionnels inspirés des travaux de Kirsch [12]. Le gradient est *maximum local*, si sa norme est supérieure à celle de ses deux 8-voisins, qui sont définis par la direction du gradient. Les processus régions sont sélectionnés par défaut sur les pixels ayant le plus faible écart-type local calculé sur des sous-fenêtres 3x3.

Comme les demandes d'informations peuvent être récursives (voir figure 1), la gestion de la coopération est complexe : il faut gérer la liste des processus interrompus, archiver globalement les informations acquises lorsqu'ils se terminent, sélectionner le processus à appliquer, et créer, sauvegarder ou restaurer le contexte de travail d'un processus.

Le contrôle des processus peut ainsi utiliser des techniques spécifiques aux systèmes d'exploitation ([13], [2]), dont le rôle est d'assurer une gestion efficace de ces objets et de leur contexte de travail. L'implantation proposée se contente de « simuler un pseudo-parallélisme » à l'intérieur d'un unique processus Unix. Cette programmation présente l'avantage d'une représentation unique et non-distribuée des données du système, et donc d'un accès non-concurrent à ces données. Un seul processus travaille à un instant donné, ce qui évite les problèmes classiques du parallélisme réel. Les demandes d'informations ne peuvent pas boucler sur elle-mêmes, car un processus fils ne travaille pas sur une zone de travail d'un processus encore actif. Il retourne simplement l'information disponible à cet endroit, à ce moment, même si elle est incomplète.

3. Les processus élémentaires

L'implantation actuelle présente deux types de techniques de segmentation de l'image, un processus de suivi de contour, et un processus de croissance de région.

3.1. LE PROCESSUS DE SUIVI DE CONTOUR

Le but de ce processus est de créer un contour, en utilisant une technique de suivi des crêtes de plus fort gradient selon une méthode de base inspirée des travaux de Martelli [14], [22], [24]

3.1.1. Les variables

La structure de données utilisée pour représenter le contour est assez simple et regroupe un grand nombre d'informations numériques sur cette primitive image :

- une liste doublement chaînée des pixels composant ce contour, ordonnée d'une extrémité à l'autre;
- des informations photométriques (la valeur moyenne et extrême du gradient sur la chaîne de pixels considérée);
- des informations géométriques (le rectangle délimitant ce contour, la longueur du contour).

3.1.2. La structure de contrôle

Mécanisme de croissance élémentaire : le mécanisme de base du processus de suivi de contour est une boucle de *sélection, évaluation, et agrégation* de pixels au contour (figure 1). Le choix du pixel suivant à incorporer au contour se fait en examinant trois pixels potentiels, situés aux deux extrémités. Ils sont recherchés dans la direction orthogonale au gradient sur ce pixel extrémité (voir la figure 2, [22]). Chacun de ces pixels est soumis à une fonction d'évaluation, qui sera fournie par la suite, dont le rôle est de déterminer le meilleur pixel à agréger. Une étape de validation permet ensuite d'accepter ou de refuser ce meilleur pixel. Cette fonction d'évaluation prend notamment en compte la norme du gradient, qui est un critère décisif, mais d'autres influences plus globales sont également incorporées. Il est important de noter que la sélection du pixel candidat repose sur une évaluation relative aux autres pixels plutôt que sur une évaluation absolue.

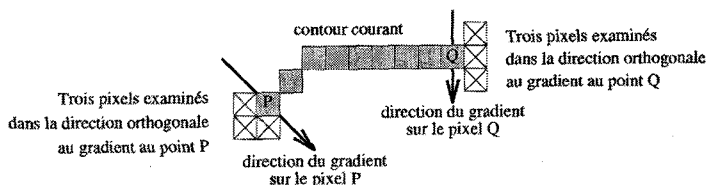


Figure 2. - Le choix du prochain pixel contour.

La fonction d'évaluation des pixels contours : Cette fonction mesure la qualité d'un pixel par rapport au contour courant (table 2). Les contributions sont normalisées entre 0 et 1, et pondérées dans le calcul de l'évaluation globale, suivant l'importance que l'on apporte à chacune d'elles. Expérimentalement, les deux contributions E_1 et E_2 sont prépondérantes dans la prise de décision, mais les deux autres permettent d'influencer le choix en faveur ou en défaveur de pixels présentant des caractéristiques de gradient voisines. Cette évaluation porte sur le couple pixel/contour,

puisque des informations à la fois locales sur le pixel et globales sur le contour servent à la prise de décision.

Validation des pixels candidats : cette étape fait aussi partie du mécanisme de croissance élémentaire. Il convient au cours de cette phase de s'assurer que :

- le pixel candidat n'est pas déjà rattaché à une autre primitive image (contour ou région), il est dans ce cas rejeté.
- l'évaluation de ce pixel est supérieure au seuil associé au processus. Ce seuil est par ailleurs réajusté dynamiquement (voir table 3). Le seuil est élevé lorsque deux régions texturées existent de part et d'autre du contour courant. Le seuil est plus faible lorsque ces deux régions sont très homogènes. Le terme de texture se rapporte pour l'instant à une simple mesure locale de dispersion des valeurs de niveau de gris sur des pixels voisins d'une fenêtre 3×3 . L'adaptation des seuils est virtuellement induite par l'émergence de nouvelles régions dans le voisinage proche du processus contour courant, sans que celui-ci en soit nécessairement l'initiateur. Les seuils numériques fournis dans cette table ont été déterminés empiriquement pour être adaptés à la majorité des discontinuités que l'on souhaite détecter dans des images raisonnablement altérées. Ces constantes numériques répondent à des problèmes de sensibilité et de visibilité des transitions des niveaux de gris au plus bas niveau du codage de l'image et sont donc relativement indépendantes du type d'image à traiter. La décision s'adapte donc dynamiquement au contexte, et à la disponibilité de l'information dans l'image.

Mécanisme de focalisation : lorsque le pixel candidat n'a pas pu être validé au cours de la phase de croissance, le processus a besoin d'informations supplémentaires. Par exemple, un contour ne trouvant plus de pixels avec un gradient suffisant pour poursuivre la croissance va pouvoir (figure 3) :

- demander une recherche de régions près de l'extrémité posant un problème, afin d'ajuster par la suite ses seuils de décision;
- demander la recherche d'un contour avec un seuil adapté en face de l'extrémité posant problème.

Le processus exprime ses besoins en fournissant trois sous-fenêtres (de dimension 9×9), dans lesquelles il indique le nombre et le type des processus fils qu'il désire générer (dans l'exemple en figure 3, un seul processus région est souhaité dans les fenêtres 2 et 3, et un seul processus contour dans la fenêtre 1).

La sélection des germes est abordée dans le paragraphe 4.1.

Intégration des connaissances : le processus contour peut éventuellement fusionner avec un contour détecté dans le voisinage de l'extrémité :

- la distance entre les deux extrémités à raccorder ne doit pas excéder 10 pixels;
- l'écart entre les directions du gradient aux deux extrémités à raccorder (discretisées selon Freeman) ne doit pas dépasser 2. Cela définit approximativement un cône de tolérance de 90 degrés, et évite des erreurs de raccordement au voisinage des intersections. La nouvelle primitive obtenue est réintroduite dans le cycle de croissance élémentaire.

Tableau 2. – Les composantes de la fonction d'évaluation mesurant la qualité d'un pixel contour candidat.

Type d'information	Valeurs	Pondération
Norme du gradient	p_0 = le pixel candidat $G_{Max} = \max_{p(x,y) \in image} \text{Grad}\vec{p} $ $g_0 = \text{Grad}\vec{p}_0 $ $E_1 = \min\left(\frac{g_0}{G_{Max}}, 1\right)$	$P_1 = 0.4$
Gradient maximum local	$E_2 = \begin{cases} 1 & \text{si maximum local} \\ 0 & \text{sinon} \end{cases}$	$P_2 = 0,3$
Continuité de la direction du gradient	$d1$ = direction gradient à l'extrémité $d2$ = direction gradient du pixel candidat (code de Freeman : $0 \leq d1, d2 \leq 7$) $diff = d1 - d2 + 8 \pmod{8}$ $E_3 = \begin{cases} 1 & \text{si } diff < 3 \\ 0 & \text{sinon} \end{cases}$	$P_3 = 0.1$
La taille du contour	l = longueur du contour $E_4 = \begin{cases} 0.0 & \text{si } 0 \leq l \leq 4 \\ 0.3 & \text{si } 5 \leq l \leq 9 \\ 0.6 & \text{si } 9 \leq l \leq 15 \\ 1.0 & \text{sinon} \end{cases}$	$P_4 = 0.2$
Évaluation totale	$F = \sum_{i=1}^4 P_i E_i$	$\sum_{i=1}^4 P_i = 1.0$

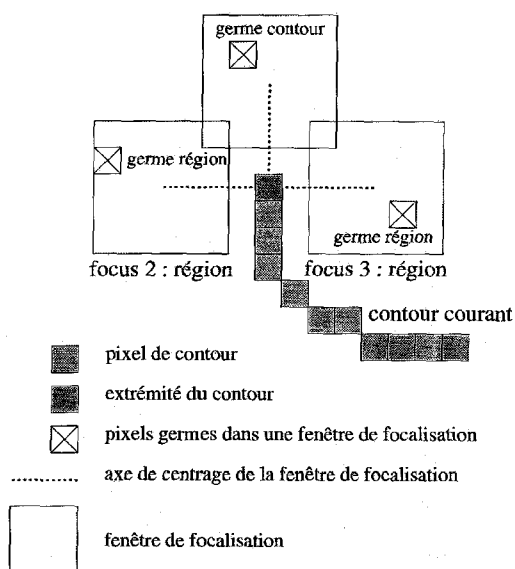


Figure 3. – Focalisation du processus contour.

Validation : Trois tests sont effectués dans cet ordre lorsque le cycle de croissance élémentaire se termine (figure 1) :

– Un contour de trop mauvaise qualité conduit à supprimer la primitive contour et à terminer le processus. Cela se produit si :

- la norme du gradient est fluctuante le long du contour construit : l'écart-type de la norme du gradient est supérieur à 3.0;
- la direction du gradient est fluctuante le long du contour : la direction change d'au moins 90 degrés entre des pixels voisins plus d'une fois tous les 5 pixels;
- le contour possède moins de 10 pixels.

La destruction de la primitive construite ne remet pas en cause la cohérence du système vis-à-vis des autres processus, qui pourraient avoir connaissance de l'existence de ce contour. En effet, chaque processus vérifie, avant de manipuler une primitive image, que celle-ci existe effectivement dans les données globales du système;

- la primitive est conservée et le processus se termine lorsque l'étape antérieure de croissance n'a pas apporté de nouveaux pixels ($nb = 0$ sur le schéma de la figure 1);
- la primitive est conservée et le processus émet des requêtes de focalisation lorsque le contour perd localement sa qualité :
 - aucun pixel candidat n'a une évaluation supérieure au seuil courant du processus;
 - la norme du gradient sur le pixel extrémité a diminué de plus de 50% par rapport à la moyenne sur les trois derniers pixels du contour à cette extrémité.
- Sinon la primitive est conservée et le processus se termine.

Tableau 3. – Les ajustements du seuillage de la fonction d'évaluation dans le cas contour.

		$G_{Max} = \max_{p(x,y) \in image} \text{Grad}(p) $
Germes initiaux	Seuil <i>a priori</i>	$(S_1, S_2, S_3, S_4) = (\min(\frac{35}{G_{Max}}, 1), 1, 1, 0)$
Autres germes	Seuil <i>hérité</i>	$\overline{\text{Grad}}$ = gradient moyen sur la fenêtre de focalisation $(S_1, S_2, S_3, S_4) = (\frac{1, 5\overline{\text{Grad}}}{G_{Max}}, 1, 1, 1, 0)$
Autres germes	Seuil <i>ajusté</i> par les régions	σ_1 = moyenne des écarts-types locaux des pixels de la région à gauche σ_2 = moyenne des écarts-types locaux des pixels de la région à droite $i = \min(\lfloor \sigma_1 \rfloor, 4)$ $j = \min(\lfloor \sigma_2 \rfloor, 4)$ $M = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 10 & 10 & 10 \\ 5 & 5 & 10 & 10 & 10 \\ 5 & 5 & 10 & 10 & 30 \end{bmatrix}$ = matrice des seuils du gradient $(S_1, S_2, S_3, S_4) = (\min(\frac{M_{i,j}}{G_{Max}}, 1), 1, 1, 0)$
		$S_{total} = P_1 S_1 + P_2 S_2 + P_3 S_3 + P_4 S_4$

3.2. LE PROCESSUS DE CROISSANCE DE RÉGION

Le but du processus de croissance de région est de produire une région homogène, en utilisant comme technique de base une méthode d'agrégation de pixels [23]. La croissance présente un nombre prédéfini de phases entre lesquelles le seuil de la fonction d'évaluation du mécanisme d'agrégation est relâché.

3.2.1. Les variables

La structure créée pour représenter une région est plus complexe que la précédente. Elle contient des informations sur les pixels intérieurs à la région, sur les pixels voisins, et sur les pixels situés à la frontière de la région :

- trois listes doublement chaînées mémorisent les pixels intérieurs, les pixels frontières et les pixels directement voisins. Les pixels frontières sont calculés en 8-connexité, et les pixels voisins (et susceptibles d'être agrégés) considérés en 4-connexité. L'incorporation des pixels dans le 8-voisinage de la région fournirait en effet une croissance trop désordonnée;

- des informations photométriques sont prises en compte : les niveaux de gris moyens et extrêmes, les valeurs moyennes et extrémales des écarts-types locaux, calculés pour chaque pixel à partir d'une fenêtre 5×5. La moyenne des écarts type locaux sur tous les pixels de la région sera appelée par abus de langage *homogénéité* de la région;

- des informations géométriques sur la région sont également stockées : position du centre de gravité, pixel frontière le plus proche/éloigné du centre de gravité, boîte englobante de la région.

3.2.2. La structure de contrôle

Mécanisme de croissance élémentaire : le mécanisme central du processus de croissance de région est constitué d'une boucle de *sélection*, *évaluation*, et *agrégation* de pixels à la région. Les pixels incorporables à la région sont choisis parmi les voisins de la région (au sens du 4-voisinage). Chaque pixel candidat est soumis à une évaluation, permettant de trier les pixels par ordre de qualité décroissante. Le *meilleur* pixel à agréger à la région est retenu.

La fonction d'évaluation des pixels régions : cette fonction est calculée afin de mesurer de manière heuristique la qualité d'un

Tableau 4. – Les composantes de la fonction d'évaluation mesurant la qualité d'un pixel région candidat.

Type d'information	Valeurs	Pondération
La différence d'homogénéité	ne = numéro de l'étape de croissance $ne \in \{0, 1, 2, 3, 4\}$ (5 étapes) ld = Ecart-type local 3×3 du pixel candidat gd = Moyenne des écarts types locaux 3×3 des pixels de la région $\Delta = gd - ld $ $M = \begin{bmatrix} 0.7 & 0.8 & 1.0 & 1.4 & 1.8 \\ 0.8 & 0.9 & 1.2 & 1.6 & 2.0 \\ 0.9 & 1.1 & 1.3 & 1.8 & 2.2 \\ 1.0 & 1.2 & 1.5 & 2.0 & 2.6 \\ 1.0 & 2.0 & 3.0 & 3.5 & 4.5 \end{bmatrix}$ $i = \begin{cases} 0 & \text{Si } gd < 2.0 \\ 1 & \text{Si } gd < 5.0 \\ 2 & \text{Si } gd < 8.0 \\ 3 & \text{Si } gd < 10.0 \\ 4 & \text{Sinon} \end{cases}$ $\Delta_0 = M_{i, ne}$ $E_1 = \frac{1000\Delta_0}{\Delta_0 + 99\Delta}$ (évaluation hyperbolique, voir figure 4) $E_1 = \begin{cases} 10 & \text{Si } \Delta < \Delta_0 \\ 0 & \text{Sinon} \end{cases}$ (évaluation en échelon, voir figure 4)	$P_1 = 0.9$
Distance au centre de gravité de la région	p = pixel candidat p_g = centre de gravité de la région p_1 = pixel région le plus proche de p_g p_2 = pixel région le plus loin de p_g $d(x, y)$ = distance entre les pixels x et y $\Delta = 10 \frac{d(p, p_g) - d(p_1, p_g)}{d(p_2, p_g) - d(p_1, p_g)}$ $\Delta_0 = 4$ $\Delta_0 \in [0 \dots 10]$ $E_2 = \frac{1000\Delta_0}{\Delta_0 + 99\Delta}$ (évaluation hyperbolique, figure 4) $E_2 = \begin{cases} 10 & \text{Si } \Delta < \Delta_0 \\ 0 & \text{Sinon} \end{cases}$ (évaluation en échelon, figure 4)	$P_2 = 0.1$
Évaluation totale	$F = P_1 E_1 + P_2 E_2$	$P_1 + P_2 = 1$

pixel candidat par rapport à la région courante. Elle prend en compte les informations fournies en table 4. De façon analogue aux contours, les constantes numériques de ces tables ont été définies empiriquement. Le choix des pondérations a pour effet de rendre prépondérante la contribution E_1 relative à la différence d'homogénéité entre le pixel candidat et la région. Le rôle de E_2 est de favoriser les pixels proches du centre de gravité de la région pour effectuer une croissance la plus compacte possible.

Validation des pixels candidats : cette étape fait aussi partie du mécanisme de croissance élémentaire. Elle permet en particulier de s'assurer que :

- Le pixel candidat n'appartient pas déjà à une autre primitive image (contour ou région);
- L'évaluation du pixel est supérieure au seuil courant du processus, qui est fixé à 10 pour un processus région. L'adaptation se

fait, non pas par variation du seuil, mais à l'intérieur même de la fonction, puisque la valeur Δ_0 servant de pivot au calcul de E_1 est tabulée en fonction de l'homogénéité courante de la région (voir figure 4). Ces valeurs pivots sont stockées dans la matrice M , qui fournit la variation acceptable entre l'écart-type d'un pixel et la moyenne des écarts-types locaux de la région, compte tenu d'une part de l'homogénéité de la région (gd) et d'autre part du niveau de relâchement atteint (ne). L'évaluation sera ainsi plus stricte pour de petites valeurs de ne correspondant aux premières étapes de croissance, et permet une expansion progressive de la région. Le calcul de la contribution E_1 et E_2 par une des deux méthodes fournies (hyperbolique ou échelon) permet de noter le pixel (figure 4) : une note inférieure à 10 correspondra à une *mauvaise évaluation*, et une note supérieure ou égale à 10 correspondra à une *bonne évaluation*. La méthode implantée est le calcul de la fonction hyperbolique, avec une classification plus fine de la qualité des pixels.

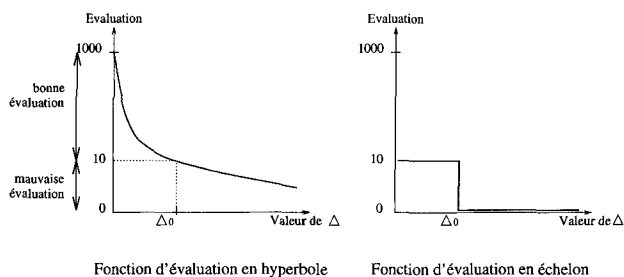


Figure 4. – Le calcul des contributions.

Mécanisme de focalisation : le processus région recherche des contours situés dans son proche voisinage, et permet ainsi de cerner avec précision sa frontière réelle avant de prendre le risque de déborder sur une autre région. Des fenêtres de taille 9×9 sont créées à partir d'un échantillonnage de la frontière de la région, par exemple un pixel sur dix (voir figure 5). Les processus contours fils héritent d'un seuil pour leur fonction d'évaluation qui dépend linéairement de la valeur moyenne de la norme du gradient sur la fenêtre de focalisation dont ils sont issus (voir table 3).

Intégration des connaissances : cette intégration est ici implicite, puisque la présence des contours n'est pas directement utilisée par la région. Cependant, leur création contribue à éviter que la région, lorsqu'elle redevient active et reprend son agrégation de pixels, ne déborde hors de sa frontière effective.

Validation : trois tests sont effectués dans cet ordre à la fin du cycle de croissance élémentaire (figure 1) :

- Une région de trop mauvaise qualité est supprimée, et le processus correspondant se termine. C'est ce qui se produit lorsque la première étape de croissance n'a pas pu fournir une région d'au moins 25 pixels. C'est le seul test de qualité effectué à ce niveau du traitement, mais on pourrait également juger l'homogénéité de la région obtenue, ou encore des critères reposant sur des considérations plus géométriques;

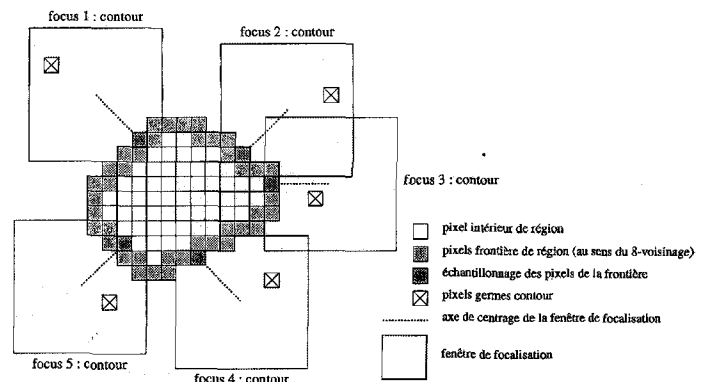


Figure 5. – Focalisation du processus de région.

- De même que pour le processus de type contour, la primitive région est conservée et le processus se termine lorsque l'étape antérieure de croissance n'a pas apporté de nouveaux pixels, ce qui correspond au cas $nb = 0$ sur le schéma de contrôle de la figure 1;

- La primitive est conservée et le processus lance des demandes de focalisation lorsque la cinquième étape de croissance n'a pas été atteinte;

- Sinon la primitive est conservée et le processus se termine.

4. Implantation du système

L'activité des processus de bas niveau est régulée au niveau supérieur par un mécanisme de focalisation qui crée effectivement les processus à partir des sous-fenêtres fournies par le processus demandeur, et par un séquenceur qui active successivement chaque processus actif du système, et qui gère leurs mises en attente et leurs réactivations.

4.1. MÉCANISME DE FOCALISATION

Ce module reçoit en entrée une sous-fenêtre de recherche, et le nombre et le type des processus à générer. Il doit définir les pixels de cette sous-fenêtre qui servent de pixels de départ (*germes*) aux processus de type contour et région (figure 6). Les pixels de la sous-fenêtre sont examinés et classés en fonction de :

- la norme du gradient pour les germes de type contour. Le pixel ayant la plus forte norme, et qui réalise le maximum de la norme du gradient dans la direction du gradient, est choisi. D'autres germes peuvent être extraits sur ce principe, en respectant un critère d'éloignement des autres germes retenus;

- l'écart-type local pour les germes de type région. Le même principe de sélection et d'espacement est utilisé.

L'initialisation du système est en général manuelle, en terme de position et de nature des germes initiaux. Le système peut

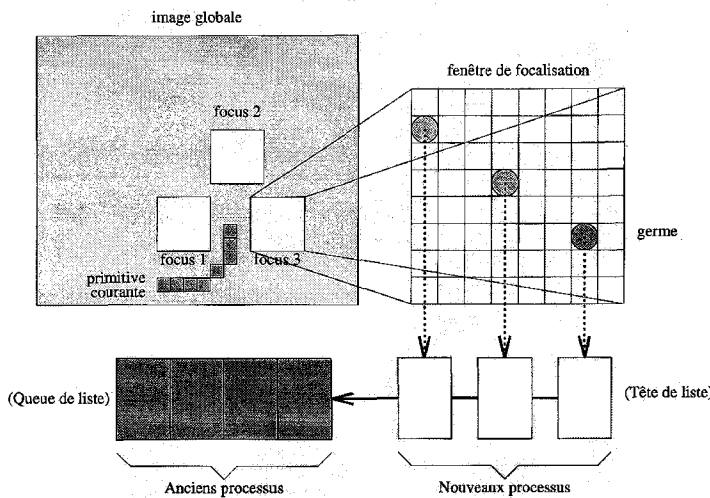


Figure 6. – Le principe de la focalisation.

cependant générer automatiquement les germes initiaux avec une sous-fenêtre de focalisation correspondant à l'image entière.

4.2. LE SÉQUENCEUR

Le séquenceur est chargé de créer, de tuer, de mettre en attente, de réveiller et d'activer les processus de bas niveau. Le séquenceur travaille indépendamment du rôle précis des processus, en les considérant à ce niveau comme des tâches anonymes. Les processus sont placés dans deux listes suivant qu'ils sont *actifs* ou en *attente*. Chaque processus *actif* est extrait de la liste, peut s'exécuter pendant une période de temps, puis est replacé dans la liste.

– Si la liste des *actifs* est gérée en FIFO (*First In First Out*, ou file d'attente), l'effet sur le système est le traitement des germes initiaux dans un premier parcours de toute la liste des processus. Les processus fils de première génération s'exécutent ensuite lors du deuxième parcours de la liste, et ainsi de suite. Les problèmes sont abordés en largeur d'abord;

- Si la liste des *actifs* est gérée en LIFO (*Last In First Out*, ou pile), l'effet sur le système est le traitement immédiat des requêtes de focalisation émises par les processus fils. Les problèmes sont abordés en profondeur d'abord.

L'autre tâche du séquenceur est de faire remonter les primitives images segmentées par un processus fils vers son père, ce qui permet d'accumuler les informations à mesure qu'un processus est «*généalogiquement*» proche des processus initiaux.

5. Expérimentations

Les principes de notre approche sont illustrés sur une image biomédicale de fibres musculaires (figure 7). L'analyse de cette

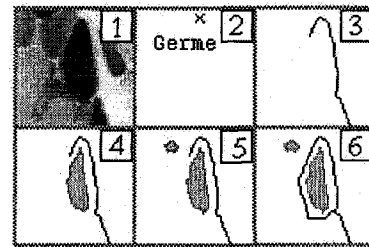


Figure 7. – Un unique germe initial de détection de contour.

image présente de sévères difficultés liées en particulier à un regroupement très dense de cellules texturées claires, et de cellules homogènes noires, dont la frontière apparaît souvent faiblement marquée. La majeure partie des détecteurs de contours fondés sur un seuillage global par hystérésis de la norme du gradient sont trop sensibles à la texture des cellules claires, et en même temps ne parviennent pas à mettre en évidence les frontières faiblement contrastées entre les cellules noires. De même, il est difficile d'éviter la sur-segmentation des cellules texturées en un grand nombre de régions de petite taille aux frontières non significatives, et dans le même temps d'éviter des fusions abusives entre des régions correspondant à des cellules homogènes. Le comportement de l'algorithme, en accord avec les principes proposés, va être discuté au travers d'exemples issus de cette image. Des expérimentations sur d'autres images sont présentées dans le dernier paragraphe.

5.1. UN UNIQUE GERME INITIAL DE DÉTECTION DE CONTOURS

Cet exemple présente l'exécution isolée d'un unique processus de détection de contours initialisé manuellement (figure 7), et créant des processus fils (contour et région) uniquement au premier niveau de parenté. Les images 2 à 5 présentent quelques résultats intermédiaires, l'image 6 est le résultat final. Les contours sont tracés en noir, une région est représentée avec ses pixels intérieurs en gris clair, et avec ses pixels frontière en gris un peu plus foncé.

L'agrégation des pixels du contour se fait sans problème en suivant les forts gradients délimitant la cellule noire du fond blanc de l'image (image 3); les gradients deviennent alors plus faibles lorsque le processus tente de faire le tour de la cellule, et à partir de cet instant, de l'information supplémentaire devient nécessaire au processus contour. Il demande donc au système la création de deux processus fils de croissance de région (images 4 et 5), ainsi qu'un contour en face de la zone litigieuse (image 6). Ces nouveaux processus sont appliqués à cette zone afin d'accumuler de l'information locale. Le processus fils de suivi de contour a reçu un seuil adapté de son père. Il est apte à segmenter correctement le reste du contour, et ainsi à terminer la détection du contour entourant la cellule, ceci malgré la frontière «*floue*» qui existe entre la cellule noire centrale et les cellules texturées environnantes, sur la gauche.

5.2. UN UNIQUE GERME INITIAL DE CROISSANCE DE RÉGION

L'exécution d'un unique processus de croissance de région en zone texturée a été isolée (figure 8). Les processus fils contour sont limités au premier niveau de parenté. Le germe de la région a été fourni manuellement à l'algorithme. Cet exemple montre la façon dont l'expansion d'une région est contrainte par la mise en évidence de contours environnants. La séquence d'images de la figure 8 présente certaines étapes caractéristiques de l'exécution du système. Sur l'image 3, le critère initial d'agrégation est très sévère, et la région correspondante est petite. Le seuil est alors relâché afin de permettre une nouvelle phase d'expansion. Cette opération se renouvelle cinq fois consécutives, afin d'étendre progressivement les frontières de la région, par l'emploi de seuils de plus en plus tolérants. Les résultats correspondants – cinq expansions consécutives de la région – sont proposés dans les images 3, 4 à 6, 7 à 10, 11 à 12, et 13 à 15.

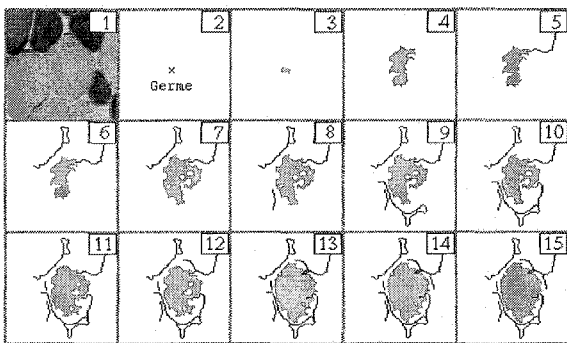


Figure 8. – Un unique germe initial de croissance de région.

Plusieurs requêtes de focalisation de type contour sont émises à la fin de chaque étape de croissance. Le système peut à ce moment découvrir des contours aux faibles valeurs irrégulières de gradient, grâce aux seuils adaptatifs déterminés pour ces processus. Les informations apportées par les contours s'avèrent ici suffisantes pour contraindre correctement l'expansion de la région.

5.3. COMPORTEMENT AU NIVEAU SYSTÈME

Le comportement au niveau système est illustré par les figures 9 et 10. Ces deux exemples diffèrent des précédents par l'ampleur du test qui a été réalisé. Le système est à présent testé dans une configuration normale, ce qui induit l'analyse de plusieurs germes initiaux de type région ou contour, et également la possibilité d'émettre des requêtes de focalisation à tous les niveaux du système, et pour tous les types de processus.

Les figures 9 et 10 illustrent la gestion en LIFO et en FIFO des files de processus. L'étape d'initialisation est la même pour les deux expériences : les mêmes germes ont été sélectionnés (10 germes de

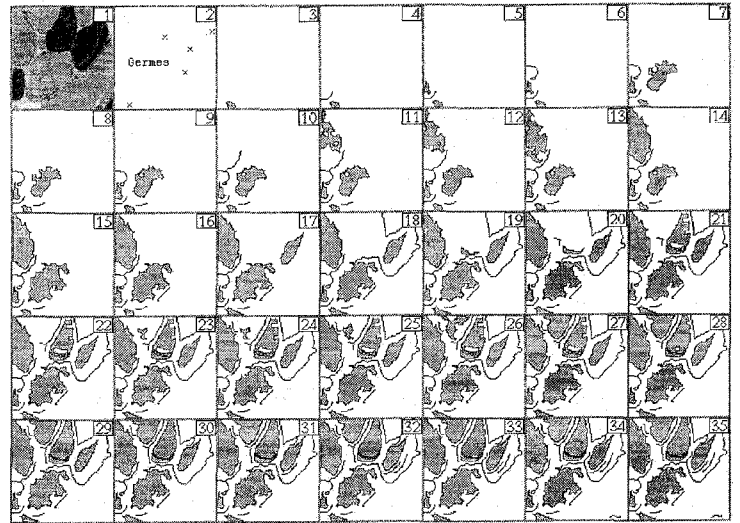


Figure 9. – Stratégie de séquençement en LIFO.

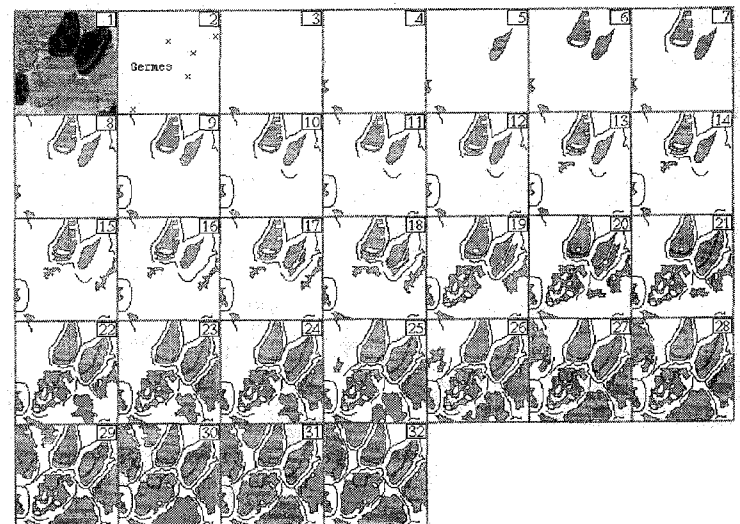


Figure 10. – Stratégie de séquençement en FIFO.

type contour et 10 germes de type région), de façon automatique cette fois, en nombre et en quantité. Le système s'exécute sur une sous-fenêtre 100×100 de l'image des cellules.

La stratégie de gestion des processus en pile (LIFO) est illustrée figure 9. Le système commence en se focalisant sur un certain germe de région, et poursuit sa tâche en empilant les processus pères et en exécutant les demandes de focalisation à un niveau de plus en plus profond dans l'arbre des processus. Les effets sur la carte de la segmentation sont faciles à comprendre : l'analyse se propage à partir du coin en bas à gauche, vers la droite de l'image. Les problèmes complexes sont traités au fur et à mesure qu'ils sont rencontrés.

La stratégie de gestion des processus en file d'attente (FIFO) est illustrée figure 10. Les germes initiaux, correspondant à des situations faciles à résoudre, sont donc traités en premier, rela-

tivement à leur ordre initial dans la liste : les processus initiaux de type région (images 2 à 6) sont exécutés avant les processus initiaux de type contour (images 7 à 11). Les requêtes de focalisation – que l'on assimile aux situations plus complexes de l'image (par exemple les régions texturées, les contours aux transitions faiblement marquées) – issues de ces processus initiaux sont traitées seulement ensuite, dans un second temps, comme cela est illustré à partir de l'image 12. Il est intéressant de comparer les résultats obtenus par les deux stratégies de séquençement. Dans les deux cas, l'exécution s'est poursuivie jusqu'à son terme. Les deux approches donnent des résultats satisfaisants, la détection des régions est correcte, malgré une légère sursegmentation avec la stratégie en file d'attente. En effet, comme cette deuxième approche consiste à traiter tous les germes initiaux en pseudo-parallèle, il est difficile d'éviter l'expansion simultanée de plusieurs germes de type région, sur une zone qui correspond à la même région réelle dans l'image. L'adjonction d'une méthode permettant de fusionner des régions résoudrait ce petit handicap. Il apparaît que le système atteint différents états de complétion, selon la stratégie de séquençement. Le résultat est plus complet dans la stratégie en largeur d'abord, en accumulant dès les premières étapes de l'algorithme une bonne quantité d'informations robustes. Ces informations permettent d'aider efficacement les autres processus dans des situations plus complexes.

5.4. AUTRES RÉSULTATS

5.4.1. Influence des conditions initiales

Nous présentons ici une application de notre coopération sur l'image des fibres musculaires, avec un unique germe initial de croissance de région. Le germe est placé dans deux endroits différents de l'image afin d'étudier l'influence des conditions initiales sur le résultat global fourni par l'algorithme. Dans les deux cas, les processus sont gérés en FIFO, par une stratégie de parcours de l'image en largeur d'abord. La figure 11(a) montre l'emplacement du germe pour les deux exécutions. La figure 11(b) présente le résultat final obtenu à partir du germe 1, dans laquelle les contours sont représentés en noir, et les régions sont représentées en gris. La carte des contours est presque complète. Les discontinuités faiblement marquées (les contours entre deux cellules noires) ont pour la plupart été trouvées. Peu de contours non significatifs apparaissent dans les cellules texturées. L'unique germe placé volontairement dans un coin de l'image n'a pas empêché le système d'évoluer et de traiter l'ensemble de l'image. Seules deux ou trois régions n'ont pas été trouvées. Elles correspondent à des zones trop petites, ou bien leur homogénéité n'est pas assez forte. La figure 11(c) donne des résultats de qualité comparable, se distinguant de la précédente essentiellement par quelques régions manquantes. Les temps d'exécution sont du même ordre (respectivement 4 min 46 et de 5 min 23). Le système est implanté sur une Silicon Graphics Indigo (MIPS R4000 à

50 Mhz), compilé avec les options d'optimisation, en inhibant toute visualisation graphique par souci de rapidité.

5.4.2. Expérimentation sur d'autres images

Le fonctionnement du système est illustré sur d'autres images, provenant de la base de données du GDR TDSI, afin de valider notre approche sur diverses applications présentant des difficultés de nature différente. Pour les deux images suivantes, le système sélectionne automatiquement 10 germes de type contour et 10 germes de type région, par la technique décrite dans le paragraphe 4.1. Les processus sont stockés en file d'attente. L'image du *bureau* est faiblement bruitée (voir les figures 12 et 13), elle propose des transitions fortement marquées au niveau des fenêtres, et des faibles discontinuités en particulier avec les pieds des sièges, et avec le fil du téléphone. Les régions homogènes correspondant aux vitres et aux meubles sont correctement localisées. La carte des contours est de bonne qualité, avec en particulier la détection des ombres des sièges et du fil du téléphone. Cependant, la carte des régions est incomplète, car la zone entre les vitres n'a pas ou peu été détectée, étant sans doute considérée comme trop étroite. L'image d'une *vue aérienne* présente d'autres difficultés (voir les figures 14 et 15). Les transitions ne sont plus aussi géométriques, des dégradés et des zones d'ombre existent, et la texture est forte sur toute l'image. L'algorithme a correctement segmenté toute la partie de l'image correspondant à l'ombre de la montagne. Un contour a correctement suivi toute la crête, les routes ou chemins qui correspondaient à des transitions plus faibles sont également présents. Les seules régions manquantes correspondent à des endroits trop texturés de l'image (rocaïlle, ou zone d'habitation) que la résolution de l'image ne permet pas de discerner correctement. Comme sur l'image des cellules, les temps d'exécution varient autour de 5 minutes. Les bons résultats obtenus par notre approche, tant du point de vue de la quantité d'information (les segmentations obtenues sont relativement complètes), que de la qualité des résultats (les primitives images extraites sont pertinentes) permettent de conclure que notre système dispose de qualités d'adaptations lui permettant d'être pertinent sur des images issues de domaines très différents.

6. Discussion

Les performances de notre système sont pour l'instant jugées visuellement. Actuellement, des études sont menées pour évaluer la qualité de segmentation d'une image naturelle, à partir d'une carte de référence établie séparément par l'expert humain ([11]).

Il convient de noter, néanmoins, que l'intérêt de l'approche ne réside pas tant dans la qualité absolue des résultats, que dans le cheminement qui a permis de les obtenir. La sensibilité des résultats aux conditions initiales constitue donc à nos yeux non pas un problème qu'il convient à tout prix de résoudre, mais au

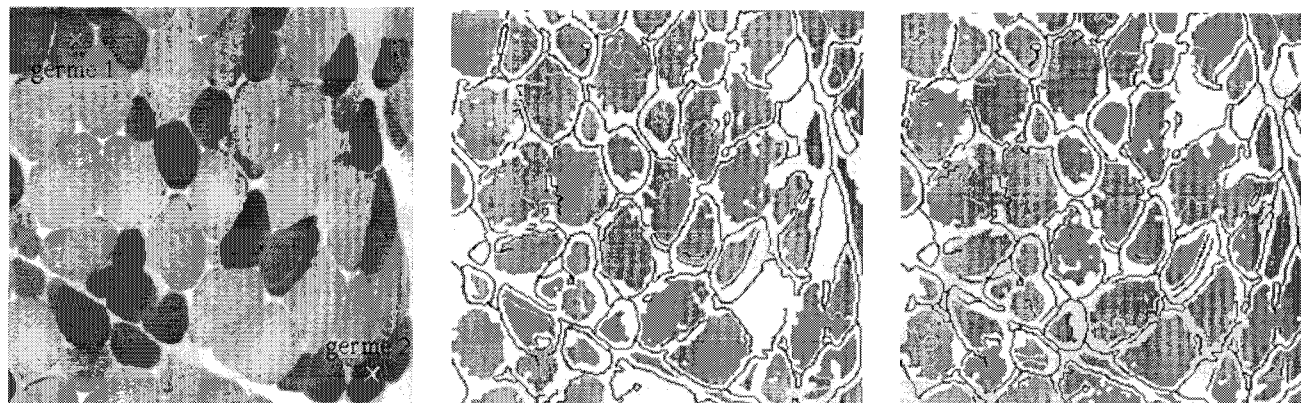


Figure 11. – Influence aux conditions initiales.

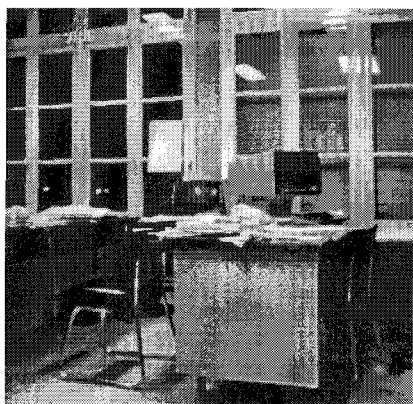


Figure 12. – Image de scène d'intérieur.

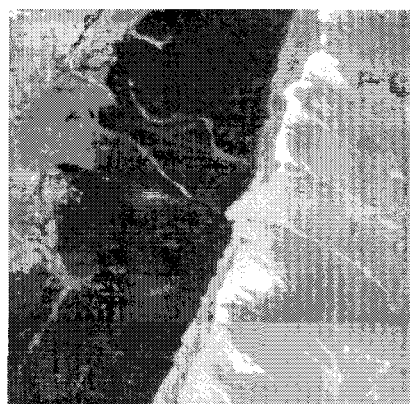


Figure 14. – Image de scène aérienne.



Figure 13. – Segmentation de l'image de scène d'intérieur.

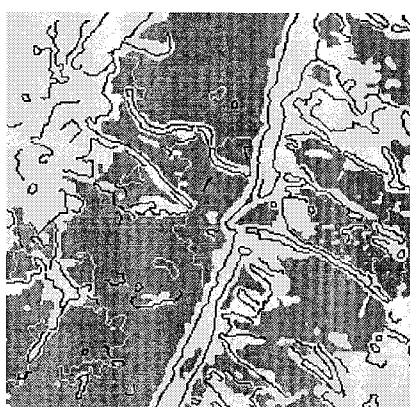


Figure 15. – Segmentation de l'image de scène aérienne.

contraire une ouverture sur une nouvelle voie de recherche tentant de lier les conditions initiales d'exécution de notre algorithme à telle ou telle configuration finale du résultat.

L'originalité de notre approche est de tirer parti des points forts de plusieurs des techniques de l'IA, sans se restreindre à n'en utiliser qu'une seule, afin de résoudre un problème de bas niveau, qui possède par définition une sémantique diffuse et peu explicite. Nous posons en effet le problème de la segmentation en termes de

recherche d'information et de prise de décision. Un système expert est dans ce cadre difficilement envisageable, car il nécessiterait de définir un contour sous forme de règles et de prédicats booléens. La sémantique d'un *contour* est bien trop riche et complexe pour être englobée dans une structure de ce type. Sa définition reste d'ailleurs souvent problématique si l'on en juge par les introductions de nombreux articles en vision. La seule connaissance fiable et utilisable au bas niveau réside dans les niveaux de gris, et elle

n'est pas suffisamment structurée pour pouvoir être utilisée avec succès par les techniques de l'IA. L'IA en effet excelle lorsqu'une modélisation du monde peut être donnée (monde des blocs par exemple), c'est-à-dire lorsque les objets que l'on va trouver dans la scène sont prévus, et leur caractéristiques parfaitement cernées.

Ainsi, plutôt qu'une approche fondée sur la représentation, nous proposons une approche de la segmentation fondée sur la spécification de processus développant des comportements adaptés de recherche, d'exploitation et de construction de l'information. Ces processus communiquent par le partage d'un environnement commun – l'image et les différentes informations qui lui sont associées – qu'ils participent à construire. Notre méthode s'apparente aussi au *tableau noir* car chaque processus contribue à son niveau à l'enrichissement d'une base de connaissance globale, qui présente l'état d'avancement de la segmentation à un instant donné. Chaque processus peut de plus être considéré comme un expert spécialisé dans un domaine. Néanmoins, le problème se pose alors de la granularité de l'expertise utilisée. En effet, il semblerait logique, selon notre point de vue, de concevoir deux experts (région et contour) autour d'un *tableau noir*. Or, la dynamique d'un *tableau noir*, liée à la diversité des experts, risque d'être restreinte dans ce cas. Ceci implique que toute la performance du système devra reposer sur la compétence des experts, ce qui soulève à nouveau le problème de l'explicitation des connaissances sémantiques des experts. Ainsi, le système proposé diffère aussi de cette approche, dans le sens où la représentation des connaissances, explicite et sous forme de règles (condition \Rightarrow action) dans beaucoup d'implantations de mécanismes de type *tableaux noirs*, apparaît de manière diffuse et moins formalisée dans notre système : toute la stratégie de création des processus fils constitue une représentation *explicite* des connaissances de nos processus/experts, tandis que toute les techniques d'adaptation à l'environnement, par des mécanismes d'évaluation numérique des situations constituent une connaissance représentée de façon plus implicite. L'autre différence notable entre notre approche et les *tableaux noirs* réside dans le contrôle de l'activation des processus/experts qui s'apparente davantage aux systèmes acteurs ou multi-agents.

En effet, notre système présente des aspects relatifs aux *acteurs* par la possibilité de création et de destruction des processus qui le composent, ce qui en fait toute l'originalité. Un processus est un mécanisme global, mais il constitue une version « personnalisée » du mécanisme global du fait des seuils qu'il hérite de son processus père.

Une métaphore plus adaptée pour analyser notre système nous paraît en fait celle des systèmes « multi-robots » ([25]), dans le sens où la résolution d'un problème global est physiquement distribuée sur plusieurs entités, réparties dans notre cas en différents points de l'image. Chacune est conçue comme spécialiste dans un domaine particulier (pour l'instant uniquement la segmentation en régions ou bien en contours). De manière identique aux robots autonomes de Luc Steels ([25]), chaque processus/agent travaille en interaction étroite avec son environnement. Le résultat

du travail d'un processus est par la suite disponible pour tous les autres processus.

Ce rapprochement entre la vision de bas niveau et les techniques de résolution de type « multi-robots » nous paraît approprié car les deux approches satisfont aux critères suivants :

- la segmentation peut se définir comme l'exploration d'un environnement sur lequel on ne dispose pas de connaissances *a priori*, et qui n'est pas modélisable de façon simple. Les approches « multi-robots » répondent à des problèmes de cette nature;
- le but global (disposer en fin de traitement d'une segmentation complète) n'est spécifié à aucun moment dans les entités élémentaires du système, mais pourtant constitue une fonction émergente d'un travail de groupe;
- les entités élémentaires (robots ou processus) doivent être spécialisées dans un domaine précis. Leur atout réside dans leur possibilité de s'adapter aux situations rencontrées;
- le modèle de coopération sous-jacent permet davantage d'opportunisme. Si une région n'est pas découverte par un processus, elle peut éventuellement l'être par un autre processus à un autre moment;
- le modèle de résolution adopté permet la construction progressive de la solution, chaque entité apportant une parcelle d'information, et contribuant ainsi à l'émergence de la solution;
- la communication est efficace lorsqu'elle est ciblée et justifiée. Ainsi la communication indirecte par l'environnement (se traduisant par des effets de bord en informatique) remplit ces deux conditions;

7. Conclusion

Une nouvelle approche de l'implantation des systèmes de segmentation coopérative a été présentée, dans laquelle la coopération est partie intégrante du processus de décision. Les principales caractéristiques de cette approche sont les possibilités de prendre des décisions adaptatives, de différer les décisions difficiles, et de résoudre des problèmes en suspens par un mécanisme de requêtes et d'accumulation d'informations. Les efforts futurs devront porter sur une meilleure formalisation de cette approche, et sur sa généralisation qui implique d'autres types de processus, et de stratégies de contrôle. Une coopération avec des méthodes spécialisées serait souhaitable. Il semble nécessaire de rendre davantage accessible à l'expert humain, mais également plus flexibles toutes les connaissances explicites et implicites de notre système, afin d'en accroître la souplesse. D'un autre point de vue, une implantation sur une architecture parallèle permettrait l'analyse des comportements sociaux émergents de ces processus, que nous simulons actuellement de manière séquentielle.

BIBLIOGRAPHIE

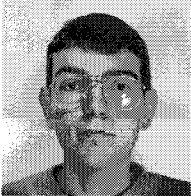
- [1] H.L. Anderson, R. Bajcsy, and M. Mintz, A modular feedback system for image segmentation. Technical report, University of Pennsylvania, GRASP Lab., 1987.
- [2] M.J. Bach, *The desing of the Unir operating system*. Prentice-Hall, 1986.
- [3] O. Baujard, S. Pesty, and C. Garbay, A programming environment for distributed vision system design. In *Proceedings of the 6th International Conference on Image Analysis and Processings*, pages 380-384, 1991. Como (Italy).
- [4] J. Benois and D. Barba, Image segmentation by region-contour cooperation for image coding. In *Proceedings of the 11th International Conferenee on Pattern Recognition*, volume 3, pages 331-334, 1992.
- [5] P. Bonnin, *Méthode Systématique de conception et de réalisation d'applications en vision par ordinateur*. Thèse de doctorat, Université de Paris VII, France, 1991.
- [6] J. Canny, A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6) : 679-698, 1986.
- [7] C.C. Chu and J.K. Aggarwal, The integration of image segmentation maps using region and edge information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12) :1241-1252, 1993.
- [8] R.K. Falah, P. Bolon, and J.P. Cocquerez, A region-region and region-edge cooperative approach of image segmentation. In *Proceedings of the IEEE ICIP Conference*, volume 3, pages 470-474, Nov 1994.
- [9] M.M. Fleck, Some defects in finite-difference edge finders. *IEEE Transaetions on Pattern Analysis and Machine Intelligence*, 14(3) :337-345,1992.
- [10] J.F. Haddon and J.F. Boyce, Image segmentation by unifying region and boundary information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10) :929-948,1990.
- [11] T. Kanungo, M.Y. Jaisimha, J. Palmer, and R.M. Haralick, A quantitative methodology for analysing the performance of detection algorithms. In *Proceedings of the 4th International Conference on Computer Vision*, pages 247-252, May 1993.
- [12] R. Kirsh, Computer determination of the constituent structures of biomedical images. *Computer and Biomedical Research*, 4(3) :315-328, 1971.
- [13] S. Krakowiak, *Principes des systèmes d'exploitation des ordinateurs*. Dunod Informatique, 1987.
- [14] A. Martelli, An application of heuristic search methods to edge and contour detection. *Communications of the ACM*, 19(2) :73-83, 1976.
- [15] M. Melkemi and J.M. Chassery. Eldge-region segmentation process based on generalized voronoï diagram representation. In *Proceedings of the 11th International Conference on Pattern Recognition*, volume 3, pages 323-326, 1992. The Hague.
- [16] E. De Micheli, B. Caprile, P. Ottonello, and P. Torre, Localization and noise in edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 :1106-1117, 1989.
- [17] A.M. Nazif and M.D. Levine, Low level image segmentation : An expert system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5) :555-577, March 1984.
- [18] N.R. Pal and S.K. Pal, A review of image segmentation techniques. *Pattern Recognition*, 26(9) :1277-1294,1993.
- [19] T. Pavlidis, *Structural Pattern Recognition*. Springer, 1977.
- [20] T. Pavlidis and Y.T. Liow, Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3) :225-233,1990.
- [21] P. Perona and J. Malik, Detecting and localisings edges composed of steps, peaks, and roofs. In *Proceedings of the 3rd International Conference on Computer vision*, pages 52-87, 1990. Osaka (Japan).
- [22] M. Salotti, Progress in edge detection. Technical Report RR 914-I, TIMC-IMAG, Informatique et Mathématiques Appliqués de Grenoble, April 1993.
- [23] M. Salotti and C. Garbay, Cooperation between edge detection and region growing : the problem of control. In Vernazza, Venetsanopoulos, and Braccini, editors, *Image Processing, Theory and Applications*, pages 95-98. Elsevier, 1993.
- [24] M. Salotti and C. Garbay, Détection de contours : Les heuristiques remplacent avantageusement les modèles. In *Proceedings du 9^e Congrès Reconnaissance des Formes et Intelligence Artificielle (RFIA 94)*, volume 1, pages 721-726, 1994.
- [25] L. Steels, Cooperation between distributed agents through self-organisation. In Werner and Demazeau, editors, *Decentralized Artificial Intelligence*, volume 1. Elsevier Science, 1990.
- [26] B. Wrobel and O. Monga. Segmentation d'images naturelles : coopération entre un détecteur-contours et un détecteur-région. In *11^e Colloque « Traitement du Signal et des images » (GRETSI)*, pages 539-542, 1987. Nice (France).
- [27] Y. Xiaohan, J. Ylä-Jääski, O. Huttunen, T. Vehkomaki, O. Sipila, and T. Katila, Image segmentation combining region growing and edge detection. In *Proceedings of the 11 th International Conference on Pattern Recognition*, volume 3, pages 481-484, 1992.
- [28] S.W. Zucker, Region growing : childhood and adolescence. *Computer Graphics and Images Processing*, 5 :382-399,1976.
- [29] O.A. Zuniga and R.M. Haralick, Gradient thresholding selection using the facet model. *Pattern Recognition*, 21(5) :493-503, 1988.

Manuscrit reçu le 21 Avril 1994.

Version acceptée le 14 Décembre 1995.

LES AUTEURS

Fabrice BELLET



Diplômé de l'ENSIMAG, titulaire d'un DEA en Robotique, Vision et Image, et F. Bellet prépare actuellement une thèse de Doctorat portant sur la coopération et l'adaptation de méthodes de segmentation en vision par ordinateur de bas niveau.

Catherine GARBAY



Docteur-Ingénieur INPG, et Docteur es Sciences de l'Université Joseph Fourier, les domaines d'intérêt de C. Garbay sont l'Intelligence Artificielle Distribuée, la Vision et les Sciences Cognitives. Le champ privilégié de ses recherches est l'ingénierie biomédicale.

Marc SALOTTI



Marc Salotti a obtenu le diplôme de Docteur Es Sciences spécialité informatique à l'Institut National Polytechnique de Grenoble en 1994. Il est actuellement Maître de Conférences en Informatique à la Faculté des Sciences de Corte, où il est responsable du D.E.S.S. Informatique Double Compétence. Ses principaux centres d'intérêts sont les premières étapes de la vision par ordinateur, la segmentation et la reconnaissance des formes.