

République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE d'ORAN Mohamed Boudiaf



Faculté des Sciences
Département d'informatique

Spécialité : Informatique

**Option : Imagerie, Vision Artificielle
et Robotique Médicale**

MEMOIRE
Présenté par

Mlle MEDDEBER Hayat

Pour l'obtention du diplôme de Magister en Informatique

Thème

Segmentation d'images Par Les Contours Actifs

Devant la commission d'examen composée de :

Président	Mr Abdelkader BENYETTOU	Professeur	USTO-MB
Rapporteur	Mme Nacéra BENAMRANE	M.conf. A	USTO-MB
Examinateur	Mme Fatima BENDELLA	M.conf. A	USTO-MB
Examinateur	Mr Rachid NOURINE	M.conf. A	Université d'Oran
Invité	Mr Abdelkade FEKIR	M.A.A	Université de Mascara

Année Universitaire : 2012/2013

Remerciements

Je tiens à adresser mes remerciements à Mm Nacéra Benamrane, Maître de Conférences à l'université d'USTO-Oran-, d'avoir accepté d'être le rapporteur de ce mémoire et de m'avoir dirigé et soutenu tout au long de ce travail.

Je tien à remercier tout particulièrement Mr Abdelkader Fekir, Maître Assistant à l'Université de Mascara, pour sa disponibilité et ses nombreux conseils. Il m'a initié au domaine de traitement d'images, et m'a encouragé à poursuivre dans cette voie. Je tiens à lui exprimer ma vive gratitude pour sa patience et son endurance pendant ces mois de recherche.

Mes remerciements s'adressent à tous les membres du jury pour l'honneur qu'ils m'ont fait en acceptant de juger ce travail.

Un grand Merci également à ma sœur Meriem, Docteur en informatique à l'Université de Mascara, parce que j'ai toujours su que je pouvais compter sur elle, d'ailleurs presque à chaque fois que je craquais, c'est sur elle que ça tombait et puis elle n'a jamais cessé de me proposer son aide et de m'encourager. J'en ai appris beaucoup de choses grâce à toi Meriem, « MERCI Sœur ».

*A ceux qui m'ont encouragé et soutenu
Mon père, ma mère
Mes sœurs Meriem, Wahiba, Chaimaa et l'adorable Ikrām
A toute ma famille...*

Table des matières

Introduction	1
---------------------	----------

Chapitre 1 Multirésolution et Pyramide	3
---	----------

1.1 Introduction.....	4
1.2 Analyse Multirésolution.....	4
1.2.1 Multirésolution par réduction.....	4
1.2.2 Multirésolution par simplification.....	5
1.3 Décomposition Pyramidale : les différentes approches.....	6
1.3.1 Pyramides par itération d'un filtrage et d'un échantillonnage.....	6
1.3.1.1 Convolution et sous-échantillonnage.....	6
1.3.1.2 Pyramides des Moyennes.....	7
1.3.1.3 Pyramides Gaussienne.....	7
1.3.1.4 Pyramides Laplacienne.....	9
1.3.1.5 Pyramides des contrastes.....	10
1.3.1.6 Pyramides des gradients.....	11
1.3.2 Pyramides géométriques.....	12
1.3.2.1 Pyramides quaternaires.....	12
1.3.2.2 Pyramides liées.....	13
1.3.3 Pyramides par ondelette.....	14
1.4 Conclusion.....	16

Chapitre 2 Les contours actifs	17
---------------------------------------	-----------

2.1 Introduction.....	18
2.2 La segmentation d'images.....	18
2.2.1 Définition.....	18
2.2.2 Les approches de la segmentation.....	18
2.2.2.1 Approches basées région.....	19
2.2.2.2 Approches basées contour.....	19
2.3 Contours actifs.....	20
2.3.1 Principe de fonctionnement des contours actifs.....	20
2.3.2 Energies	22
2.3.2.1 Energie interne.....	22
2.3.2.2 Energie externe.....	22
2.3.2.3 Energie de contexte.....	23
2.3.3 Minimisation de la fonctionnelle d'énergies	23

2.3.3.1	Approches variationnelles.....	23
2.3.3.2	Programmation dynamique.....	24
2.3.3.3	Algorithme de Greedy	25
2.3.3.4	Algorithme de Fast Greedy	27
2.4	Intérêts et limites des contours actifs	28
2.5	Les descendants du snake.....	29
2.6	Travaux Antérieurs sur la détection et le suivi d'objet.....	30
2.7	Conclusion	33

Chapitre 3 Agent et Système Multi-agents 34

3.1	Introduction.....	35
3.2	De l'intelligence artificielle distribuée (IAD) aux systèmes multi-agents.....	35
3.3	Le concept d'agent.....	36
3.3.1	Définitions.....	36
3.3.2	Principales caractéristiques d'un agent.....	36
3.3.3	Typologies des agents.....	37
3.3.3.1	Agents cognitifs.....	38
3.3.3.2	Agents réactifs.....	38
3.3.3.3	Agent hybride.....	40
3.4	Les systèmes multi-agents.....	40
3.4.1	Définition.....	40
3.4.2	Caractéristiques d'un SMA.....	41
3.4.3	L'interaction dans les SMA.....	42
3.4.3.1	Coordination.....	42
3.4.3.2	Coopération.....	42
3.4.3.3	Communication.....	43
3.4.4	Les avantages des agents et les systèmes multi-agents.....	45
3.5	Systèmes multi-agents et segmentation.....	46
3.6	Conclusion.....	48

Chapitre 4 Conception et Implémentation 49

4.1	Introduction.....	50
4.2	Approche proposée.....	50
4.2.1	Prétraitement.....	51
4.2.1.1	Filtrage.....	51
4.2.1.2	Calcul du gradient.....	52
4.2.2	Agents.....	52
4.2.2.1	Agent Pyramide.....	53
4.2.2.2	Agent Snake.....	53
4.2.2.2.1	Initialisation du snake dans la première image de la séquence...54	
4.2.2.2.2	Evolution du Snake.....	55
4.2.2.2.3	Critère d'arrêt du contour actif.....	58
4.2.2.2.4	Calculer le centre de l'objet détecté.....	58

4.2.2.3 Agent Projection.....	59
4.3. Expérimentations.....	60
4.3.1 Séquence d'images.....	60
4.3.2 Environnement matériel et logiciel.....	60
4.3.3 Résultats.....	60
4.3.3.1 Agent Pyramide.....	60
4.3.3.2 Agent Snake.....	62
4.3.3.3 Détection et Suivi d'objet.....	64
4.3.3.4 Le Temps de Suivi.....	72
4.4 Conclusion.....	74
Conclusion et perspectives	75
Bibliographie	76

Table des figures

Chapitre 1

1.1	Représentation pyramidale à quatre niveaux d'une coupe IRM.....	5
1.2	Une image à différentes résolutions de détails.....	5
1.3	Pyramide des moyennes	7
1.4	Exemple de Pyramide Gaussienne.....	8
1.5	Exemple de Pyramide Laplacienne.....	10
1.6	Une image ainsi que le quadtree et l'arbre quaternaire correspondant.....	12
1.7	Pyramide quaternaire. chaque père possède quatre fils, chaque fils possède un père....	12
1.8	Structure d'une pyramide liée.....	13
1.9	Principe de décomposition d'une image par ondelette.....	15
1.10	Décomposition par ondelette de Haar.....	15

Chapitre 2

2.1	Principe des contours actifs.....	20
2.2	Les trois formes du snake.....	21
2.3	Voisinage. (a) voisinage 3×3 (b) voisinage cardinaux (c) voisinage diagonaux.....	27
2.4	Les points noir représentent les pixels examinés.....	28
2.5	Le snake ne peut se scinder pour délimiter correctement deux objets dans l'image.....	29

Chapitre 3

3.1	Agent Cognitif.....	38
3.2	Agent Réactif	39
3.3	Représentation schématique d'un SMA, du point de vue d'un observateur extérieur....	40
3.4	Les différents modes de coopération.....	43
3.5	Communication par partage d'informations.....	44
3.6	Communication par envoi de messages.....	45

Chapitre 4

4.1	Etapes de l'approche proposée.....	51
4.2	Élimination de bruit par le filtre médian.....	52
4.3	Fonctionnement des agents sur des pyramides de trois niveaux.....	52
4.4	Décomposition par pyramide Gaussienne.....	53
4.5	Organigramme qui représente le travail de l'agent Snake	54

4.6	Initialisation du snake	55
4.7	Algorithme de Greedy.....	57
4.8	Algorithme de Fast Greedy.....	57
4.9	Evolution du Snake.....	58
4.10	Changement du centre d'initialisation	59
4.11	Pyramide Gaussienne avec $a=0.68$, filtre (5×5)	61
4.12	Pyramide Gaussienne avec $a=0.6$, filtre (5×5).....	61
4.13	Pyramide Gaussienne avec $a=0.68$, filtre (3×3).....	61
4.14	Pyramide Moyenne.....	61
4.15	Continuité 0.7.....	62
4.16	Courbure 0.5.....	62
4.17	Gradient 0.4.....	62
4.18	Ballon 0.2	62
4.19	Continuité 1, courbure 0.13, Gradient 0.9, Ballon 0.1.....	63
4.20	Influence de centre : avec la même initialisation de départ	63
4.21	Changement du centre d'initialisation.....	64
4.22	Résultat de la Séquence Cercle (80 images)	65
4.23	Résultat de la Séquence Triangle (80 images).....	66
4.24	Résultat de la Séquence Biologique « Microbe » (64 images).....	68
4.25	Résultat de la Séquence Biologique « Cellule » (72 images).....	69
4.26	Résultat de la Séquence échocardiographique (64 images).....	71
4.27	Comparaison entre les trois approches de suivi.....	72
4.28	Comparaison entre Greedy et Fast greedy pour chaque séquence.....	73

Introduction

Le suivi d'objet dans une séquence d'images occupe une place importante dans de nombreux domaines, comme la robotique, la vidéosurveillance, mais aussi l'imagerie médicale qui est un secteur en expansion et fait toujours appel à de nouvelles méthodes de suivi.

Pour développer une approche de suivi robuste, deux contraintes se posent :

- La première est la qualité de suivi,
- La deuxième est l'aspect du temps de suivi qui est très important surtout dans le domaine médical.

Pour satisfaire la première contrainte, nous nous intéressons aux contours actifs ou "*Snakes*" qui se présentent comme un modèle pour l'extraction de caractéristiques visuelles dans une image tel que les contours d'objet. Grâce à leurs souplesses et leurs capacités à intégrer les processus de détection des contours en un seul processus de minimisation d'énergie, les contours actifs sont largement utilisés pour la détection et la localisation des frontières afin de faciliter le problème de segmentation d'images et du suivi d'objet.

Pour avoir un suivi rapide d'un objet dans une séquence d'images, c'est-à-dire la satisfaction de la deuxième contrainte, nous allons proposer dans ce Mémoire une approche parallèle de suivi d'objet qui combine la méthode des contours actifs paramétriques avec une analyse multirésolution. Cette combinaison permet d'accélérer la convergence du Snake vers les frontières de l'objet et de diminuer la sensibilité des contours actifs aux bruits. Deux techniques de représentation multirésolution seront utilisées : les pyramides gaussiennes et les pyramides moyennes. L'approche que nous proposons comporte trois Agents travaillant en parallèle : Agent pyramide, Agent Snake et Agent Projection.

Le présent Mémoire est structuré comme suit :

Le premier chapitre est consacré à une introduction à la multirésolution ainsi qu'à un certain nombre de modèles pyramidaux pour la représentation d'image en multirésolution.

Dans le deuxième chapitre nous détaillons la méthode des contours actifs et nous citons quelques travaux récents qui utilisent cette méthode pour la détection et le suivi d'objet. Ces travaux montrent bien que cet outil se développe et intéresse de plus en plus de chercheurs.

Le troisième chapitre présente les systèmes multi-agents et leur utilisation en analyse d'images.

Le quatrième chapitre décrit l'étude et la mise en œuvre de l'approche de suivi proposée. Nous détaillons les différentes étapes que nous avons parcourues. À la fin, nous exposons les résultats obtenues sur des images synthétiques et des images réelles.

Une conclusion et des perspectives finalisent notre travail.

1

Multirésolution et Pyramide

Sommaire

1.1 Introduction.....	4
1.2 Analyse Multirésolution.....	4
1.2.1 Multirésolution par réduction.....	4
1.2.2 Multirésolution par simplification.....	5
1.3 Décomposition Pyramidale : les différentes approches.....	6
1.3.1 Pyramides par itération d'un filtrage et d'un échantillonnage.....	6
1.3.2 Pyramides géométriques.....	12
1.3.3 Pyramides par ondelette.....	14
1.4 Conclusion.....	16

1.1 Introduction

Les techniques d'analyse multirésolution, sont des outils de plus en plus répandus pour le traitement d'image qui trouvent de nombreuses applications comme la segmentation, le codage, la compression ou le filtrage d'image. La raison principale de leurs succès est leurs similitudes avec le système visuel humain qui utilise des décompositions multirésolution pour analyser l'information visuelle.

Dans ce premier chapitre, on va présenter trois techniques de décomposition d'une image en multirésolution. Ces techniques permettent de construire des pyramides multirésolution.

1.2 Analyse multirésolution

L'analyse multirésolution (AMR) décompose le signal ou l'image en versions d'approximation et de détails aux échelles de plus en plus petites. Les décompositions les plus connues sont les pyramides [Pop08].

Intuitivement, la multirésolution peut apparaître à notre système visuel de différentes manières :

- réduction de la taille de la scène observée (éloignement) ;
- diminution de la netteté ;
- simplification du contenu.

Sous l'aspect structural, la représentation multirésolution d'une image se présente de la manière suivante : en utilisant un traitement approprié sur une image, on peut obtenir une série d'images « dérivées » qui contiennent de moins en moins d'information. Considérons les éléments de la multirésolution comme un empilement de ces images de résolution décroissante. Cette structure est une pyramide, composée d'une succession de niveaux, où le niveau de la base est l'image originale et les niveaux successifs sont de résolution de plus en plus réduite. Dans la grande majorité des méthodes multirésolution, chaque résolution (ou niveau) est obtenue à partir de la résolution précédente et non pas à partir de l'image originale [Ber95].

1.2.1 Multirésolution par réduction

La multirésolution peut être vue sous son aspect spatial : dans ce cas, la taille de l'image diminue et provoque une diminution de la résolution.

Donc l'image est décomposée en des résolutions multiples, à différentes échelles tout en conservant les données initiales. L'empilement de cette série d'images permet de construire une pyramide.

Un schématique est présenté par la figure 1.1, le niveau le plus bas de la pyramide a la même échelle que l'image originale et contient des informations de plus haute résolution. Une image d'un niveau N de la pyramide I_n a des dimensions moitié en largeur et en hauteur par rapport aux dimensions de l'image I_{n-1} du niveau précédent (N-1).

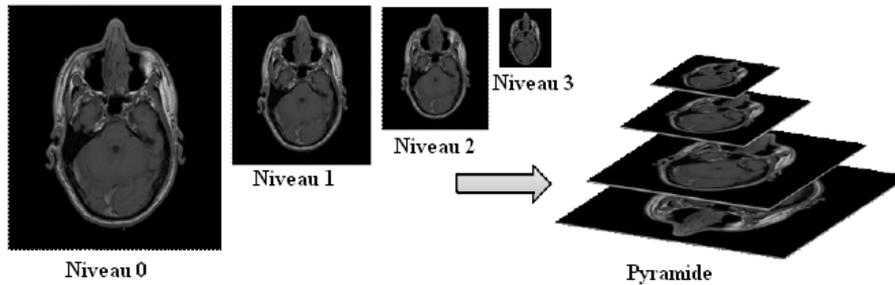


Figure 1.1 : Représentation pyramidale à quatre niveaux d'une coupe IRM

Les méthodes qui mettent en jeu la multirésolution spatiale ne sont pas des techniques de segmentation à part entière mais permettent généralement de segmenter des images à une dimension réduite. Ensuite le résultat obtenu est converti à la taille de l'image originale.

1.2.2 Multirésolution par simplification

Sous cet aspect, la résolution est liée au détail : les différents niveaux ont tous la taille de l'image initiale. Seul le degré de détail décroît avec la résolution.

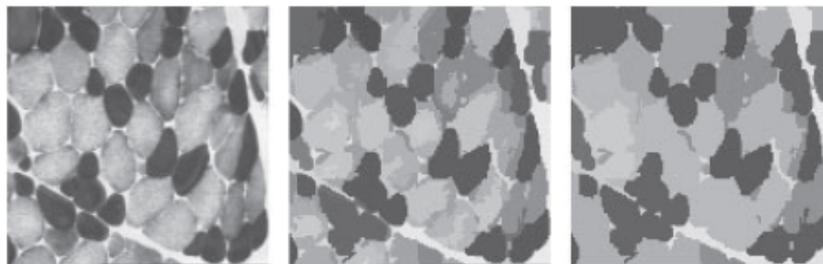


Figure 1.2 - Une image à différentes résolutions de détails

Certaines des méthodes qui procèdent par simplification sont des techniques de segmentation multirésolution car chacune des images qu'elles génèrent est une segmentation à part entière (voir figure 1.2). Un niveau de la pyramide est alors un ensemble de régions (cellules) dont la forme est souvent liée à la méthode, et la taille est liée au niveau de résolution [Ber95].

De façon générale, de nombreux algorithmes pyramidaux se dégagent, parmi lesquels:

- Les pyramides basées sur l’itération d’un filtre et d’un échantillonnage,
- Les pyramides géométriques,
- Les pyramides basées sur l’analyse en ondelettes.

1.3 Décomposition Pyramidale : les différentes approches

1.3.1 Pyramides par itération d'un filtrage et d'un échantillonnage

Une pyramide basée sur l’itération d’un filtre et d’un échantillonnage est une séquence d’images dans laquelle chaque niveau est une copie filtrée et sous-échantillonnée du prédécesseur [Ari09]. Une image est décomposée en des résolutions multiples, à différentes échelles tout en conservant les données initiales.

1.3.1.1 Convolution et sous-échantillonnage

Le sous-échantillonnage est un élément important de la décomposition. Il définit le rapport de résolution entre deux niveaux successifs de la pyramide. C’est donc lui qui caractérise la taille des images à un niveau donné.

Il est toujours possible de réduire successivement la taille d’une image par un facteur quelconque en affectant à chaque élément père une valeur calculée à l’aide de ses fils : médiane, etc... Cette méthode, mise au point par [kel71] en 71, a été la toute première utilisée pour représenter une image entière à différentes résolutions. Dans ce cas, la perte d’information entre deux niveaux successifs est quasi aléatoire et non maîtrisée.

Burt [Bur81a] montre que la perte minimale d’information se fait en accompagnant le sous-échantillonnage par un filtrage passe-bas. Ce filtrage est effectué par un produit de convolution entre les pixels de l’image et un masque de convolution idoine (également appelé noyau).

Donc la génération des images à multiples résolutions requiert l’utilisation d’un filtre passe bas. A pleine résolution, l’image comporte beaucoup trop de détails, beaucoup de primitives détectées à haute résolution correspondent à du bruit de numérisation et ne correspondent donc pas à des primitives physiques de la scène.

La convolution de l’image par un filtre passe-bas nous fait passer à un niveau de résolution grossière qui ne contient que les primitives de grande échelle spatiale contenues dans la scène observée. Les primitives détectées sont moins nombreuses, et correspondent à des primitives physiques.

Toute la théorie de l’analyse multi résolution repose sur la prémisse qu’aucune primitive n’est générée lorsqu’on se dirige vers les plus basses résolutions. Il faut donc que le filtre passe bas ne génère pas de nouvelles primitives lorsque la résolution est baissée [Rid06].

[Wit84] avait trouvé que le filtre gaussien répond à cette exigence sévère pour les signaux à une dimension.

1.3.1.2 Pyramides des Moyennes

Une pyramide M des moyennes est un ensemble d'images de résolution décroissante, la base étant l'image d'origine et le haut l'image de plus basse résolution (voir figure 1.3).

Une image M_k de niveau k est obtenue en sous-échantillonnant l'image M_{k-1} et en moyennant le voisinage par exemple 2×2 de chaque pixel de M_{k-1} . Donc chaque pixel de M_{k-i} correspond au moyennage de la zone $2^i \times 2^i$ de M_k .

Chaque valeur de pixel est calculée en prenant la moyenne des quatre pixels juste en dessous: le filtre passe-bas appliqué est un filtre moyeneur et les coefficients du filtre sont simplement égaux à 0,25 [Lep05].

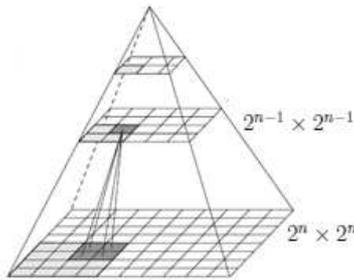


Figure 1.3 : Pyramide des moyennes

L'information contenue dans chaque pixel des niveaux supérieurs de la pyramide est une combinaison linéaire des valeurs du groupe de pixels localisés en dessous, aux niveaux plus bas de la pyramide.

1.3.1.3 Pyramides Gaussienne

Dans une pyramide Gaussienne [Bur83], le processus de réduction est effectué à la fois par un filtrage et par un sous-échantillonnage. Le filtrage de type passe-bas est obtenu par un noyau de convolution dont les valeurs des coefficients discrets approximent les valeurs d'une courbe Gaussienne.

Considérons une image de résolution K . Une représentation multirésolution de cette image, peut se décrire par le procédé itératif suivant :

$$G_k = \Delta(G_{k-1}) \quad k = 1, \dots, K$$

$$G_k(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) \cdot G_{k-1}(2i + m, 2j + n)$$

$$i = 0, \dots, 2^k - 1 \quad j = 0, \dots, 2^k - 1 \quad \dots (1)$$

Où : W est un noyau gaussien de taille 5×5 ,
 Δ symbolise l'opérateur de réduction.

La collection d'images G_0, G_1, \dots, G_K est appelée pyramide gaussienne où G_0 est l'image initiale (voir figure 1.4). Dans l'équation (1), nous remarquons que chaque niveau de la pyramide gaussienne est le résultat d'une convolution du niveau inférieur par un noyau gaussien combinée à une opération de sous-échantillonnage de facteur 2.



Figure 1.4 : Exemple de Pyramide Gaussienne

Burt et Adelson ont imposé certaines contraintes au noyau W qui s'expriment de la manière suivante:

- *La séparabilité:* pour réduire le coût de calcul en dimension 2, on prend un noyau W sous forme de produit tensoriel de deux noyaux unidimensionnels identiques w .

$$W(m, n) = w(m) \cdot w(n)$$

$$m = -2, \dots, 2 \quad n = -2, \dots, 2 \quad \dots(2)$$

- *La normalisation:* cette condition porte sur le noyau monodimensionnel w et permet de préserver la moyenne des niveaux de gris.

$$\sum_{n=-2}^2 w(n) = 1 \quad \dots(3)$$

- *La symétrie:* le noyau w doit être symétrique pour garantir une réponse implusionnelle réelle.

$$w(-i) = w(i) \quad \dots(4)$$

- *L'équicontribution* : chaque point de l'image à un niveau donné doit contribuer de la même façon aux nœuds du niveau successeur de la pyramide.

$$\sum_{n=\text{pair}} w(n) = \sum_{m=\text{impair}} w(m) \quad \dots(5)$$

- *l'unimodalité*: la réponse impulsionnelle doit être unimodale afin de mieux approcher un filtre gaussien, ce qui pour une détection de contours permet d'éviter les faux contours. En prenant la valeur de $w(0) = a$ comme paramètre, on peut exprimer les coefficients du filtre monodimensionnel sous les conditions précitées comme suit:

$$\frac{1}{4} \leq w(0) = a \leq \frac{1}{2}$$

$$w(1) = w(-1) = \frac{1}{4} \quad w(2) = w(-2) = \frac{1}{4} - \frac{a}{2} \quad \dots(6)$$

1.3.1.4 Pyramides Laplacienne

La différence entre un niveau de la pyramide gaussienne et sa version estimée par expansion du niveau supérieur, conduit à un niveau de la pyramide laplacienne [Cal97]. Chaque niveau k de cette pyramide est construit en faisant la différence pixel à pixel de deux niveaux successifs k et $k + 1$ de la pyramide Gaussienne. Cette différence pixel à pixel est rendue possible par une expansion du niveau $k + 1$ Gaussien à la taille du niveau k . Le niveau Laplacien k ainsi construit comporte toutes les hautes fréquences (détails) perdues par le passage du niveau k au niveau $k + 1$ Gaussien.

Les différences pouvant être positives ou négatives, la valeur zéro est habituellement représentée par le niveau de gris 128, ceci explique la teinte 'grise' d'une image Laplacienne (voir figure 1.5). Notons que chaque niveau $L[k]$ de la pyramide Laplacienne dépend du même niveau $G[k]$ de la pyramide Gaussienne mais également du niveau supérieur $G[k + 1]$. De ce fait, la pyramide Laplacienne possède un niveau de moins que la pyramide Gaussienne [Ber95].

Ce principe se décrit par :

$$L_k = G_k - E(G_{k+1}) = G_k - \hat{G}_k \quad \dots(7)$$

Où : L_k représente un niveau k de la pyramide laplacienne.
 E représente l'opération d'expansion.

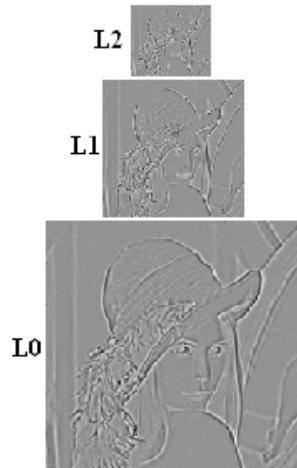


Figure 1.5 : Exemple de Pyramide Laplacienne

1.3.1.5 Pyramides des contrastes

La pyramide C des contrastes d'une image I est définie par [Jol94]:

$$C_k(P) = \frac{G_k(P) + L_k(P)}{G_k(P) - L_k(P)} \quad \dots(8)$$

Où P est un pixel, k le niveau dans la pyramide, G_k le niveau k de la pyramide gaussienne construite sur I, N la hauteur de la pyramide.

En tenant compte de (7), on obtient :

$$C_k(P) = 2 \frac{G_k(P)}{E[G_{k+1}](P)} - 1$$

Pour $K = 0, \dots, N-1$... (9)

Où $E[G_{k+1}]$ l'expansion du niveau k + 1 de la pyramide G sur le niveau k.

Le comportement de C_k est lié directement à celui du $G_k / E[G_{k+1}]$, C'est pourquoi, la mesure de contraste multirésolution est choisie comme :

$$C_k(P) = \frac{G_k(P)}{E[\tilde{G}_{k+1}](P)} \quad \dots(10)$$

Puisque C_N n'est pas définie, les auteurs [Jol94] ont proposé que $E[G_{N+1}] = G_N$ d'où il résulte $C_N = 1$.

1.3.1.6 Pyramides des gradients

En plus de la pyramide laplacienne, Burt et Kolczynski [BuK93] ont proposé une autre pyramide basée sur la détection de contours : la pyramide de gradient.

La première étape consiste à construire la pyramide gaussienne avec un noyau w et un sous échantillonnage avec un facteur 2 sur l'horizontale ainsi que sur la verticale : A l'exception de la première étape de la pyramide gaussienne, Burt et Kolczynski construisent une pyramide d'orientation du gradient, qui consiste à chaque étape à construire 4 images représentant l'image selon l'orientation à 0° , 45° , 90° et 135° .

Une autre pyramide non linéaire a été proposée à partir d'opérateurs de morphologie mathématique : érosion, dilatation, ouverture et fermeture. Cette pyramide est construite en utilisant des filtres morphologiques successifs [Pop08].

Pour Haralick *et al.* [Har89], l'opérateur de réduction est composé d'une ouverture, d'une dilatation puis d'un sous-échantillonnage afin de générer l'étape suivant.

Une structure pyramidale basée sur l'itération d'un filtrage et d'un échantillonnage offre les avantages suivants :

- *Représentation compacte* : La structure nécessite moins d'espace en mémoire, cette grande compacité provient du sous échantillonnage à chaque niveau de la pyramide qui divise les dimensions de l'image résultante par rapport aux dimensions de l'image au niveau précédent.
- *Filtrage simple et rapide* : Chaque niveau de la pyramide est généré à partir du niveau précédent en appliquant toujours le même filtre passe-bas. Le même filtre peut être utilisé car il est appliqué sur une image qui est déjà sous-échantillonnée, ce qui déplace la fréquence de coupure effective du filtre. Le même filtre, de petite taille, est donc appliqué de couche en couche, résultant en une génération très rapide des images aux diverses résolutions spatiales puisque la somme pondérée utilisée pour calculer la valeur de chaque pixel est limitée en nombre de termes.
- *Information globale disponible* : L'information contenue dans chaque pixel des niveaux supérieurs de la pyramide est une combinaison linéaire des valeurs du groupe de pixels localisés en dessous, aux niveaux plus bas de la pyramide. Donc, à mesure que l'on monte le long de la pyramide, les pixels ont un champ récepteur de plus en plus grand sur la couche de base [Lep05].

1.3.2 Pyramides géométriques

1.3.2.1 Pyramides quaternaires

Quadtree et octree

La structure de *quadtree* (figure 1.6) ou arbre quaternaire [Hor74] (étendue en 3D par la structure d'*octree*) utilise la propriété de récursivité du maillage carré. L'idée intuitive consiste à partir de l'image toute entière formant le bloc initial, puis, à diviser récursivement tout bloc non homogène selon un critère en quatre sous-blocs.

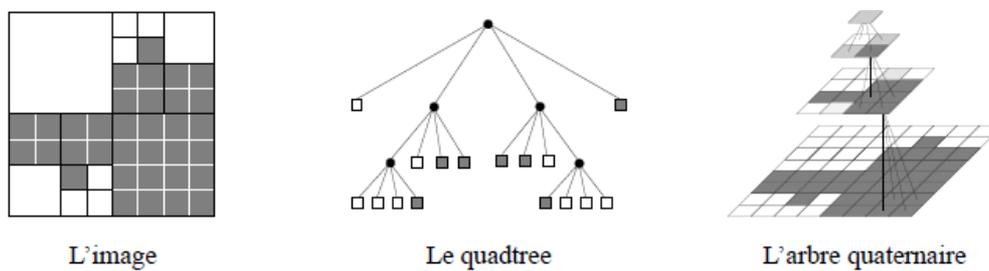


Figure 1.6 : Une image ainsi que le quadtree et l'arbre quaternaire correspondant

La structure de *quadtree* est proche de celle de pyramide quaternaire [Ros84]. Cette dernière est construite de bas en haut en fusionnant chaque bloc 2×2 de cellules en une cellule de plus faible résolution.

Cette structure, contrairement aux *quadtrees*, permet de conserver les liens d'adjacence entre les cellules [Gra06].

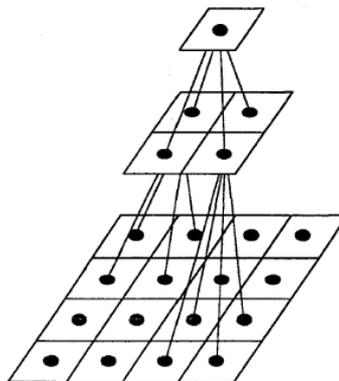


Figure 1.7 : Pyramide quaternaire.
chaque père possède quatre fils,
chaque fils possède un père

La structure d'une pyramide quaternaire est obtenue en représentant de manière récursive 4 pixels voisins appelés fils du niveau k par un pixel père du niveau $k + 1$ (figure 1.7). L'image doit nécessairement être de taille $N \times N$ (avec $N = 2^n$) et le sommet est atteint en $\log_2(N)$ étapes depuis la base de la pyramide.

1.3.2.2 Pyramides liées

Les pyramides liées ou pyramides à recouvrements ont été introduites pour la première fois par Burt et al [Bur81b]. La structure de la pyramide liée est caractérisée par les points suivants (voir figure 1.8) :

- Chaque cellule du niveau k est liée à un bloc de 4×4 cellules du niveau $k-1$;
- Au niveau $k-1$, les blocs se recouvrent : chaque bloc de 4×4 cellules recouvre à 25% (c'est-à-dire avec 2×2 cellules) chacun de ses 4 voisins [Ber95] ;

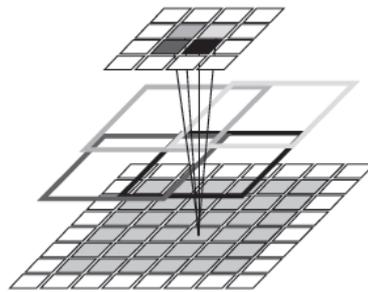


Figure 1.8 : Structure d'une pyramide liée. Les 4 pères (colorés) se partagent les fils (grisés) selon 4 groupes (qui se recourent). Pour plus de clarté, la projection de chaque groupe de fils reprend la couleur de son père. Les 4 fils du centre appartiennent chacun aux 4 pères

Le niveau 0 de la pyramide est initialisé à l'image originale. Les résolutions correspondant aux niveaux supérieurs peuvent être calculées en effectuant un moyennage 4×4 dans les fenêtres enfants. L'initialisation de la pyramide liée constitue la première étape d'une approche de segmentation d'image proposée par [Bur81b].

Une variante de ces pyramides sont les pyramides liées pondérées. Introduites par Hong *et al* [Hon82], elles permettent d'éviter à un fils de ne choisir qu'un unique père. Dans ces pyramides, chaque fils conserve tous ses pères. Afin de rendre compte de la préférence du fils envers l'un de ses pères, un poids est associé à chaque arête père-fils indiquant la similarité de ces deux nœuds. Ces pyramides sont construites sur le même principe que précédemment [Gra06].

1.3.3 Pyramides par ondelette

La transformation par ondelettes est une technique de traitement de signal qui décompose une image (c'est-à-dire d'un signal à deux dimensions) en une série d'images de résolution inférieure.

L'analyse multirésolution par ondelettes d'un signal, d'une image ou plus généralement d'une fonction $i(x)$ consiste en sa projection sur des bases de fonctions $\Psi_{a,b}(x)$ appelées fonctions analysantes, donnant des approximations de moins en moins fines de la fonction originale.

$$W(a, b) = \int_{-\infty}^{+\infty} i(x) \Psi_{a,b}(x) dx \quad \dots(11)$$

Ces fonctions analysantes sont obtenues à partir de la déformation d'une fonction mère $\Psi(x)$, telle que :

$$\Psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \Psi\left(\frac{x-a}{b}\right) \quad \dots(12)$$

Le paramètre a permet d'appliquer la fonction analysante en différents points de la fonction i et le paramètre b permet d'appliquer la fonction analysante à différentes échelles [Zla06].

La décomposition par ondelette d'une image, consiste à séparer chacune des composantes horizontale (x), verticale (y) et diagonale (x et y), sous-échantillonnées d'un facteur 2 par rapport à l'image originale. Il en résulte un schéma à quatre cadrans, comme présenté dans la figure 1.9. Par exemple, lorsque la composante y est traitée, on obtient une image sous-échantillonnée d'un facteur 2, qui contient les détails verticaux (hautes fréquences, coin inférieur gauche).

L'image originale peut alors être reformée, à partir des composantes basses-fréquences sur x et y (approximation grossière de l'image, coin supérieur gauche) auxquelles on ajoute les détails hautes fréquences des trois autres cadrans. De même, le traitement de la composante x donne une image sous-échantillonnée qui contient les détails horizontaux. L'opération ainsi répétée est appelée analyse multi-résolution (AMR) par ondelettes.

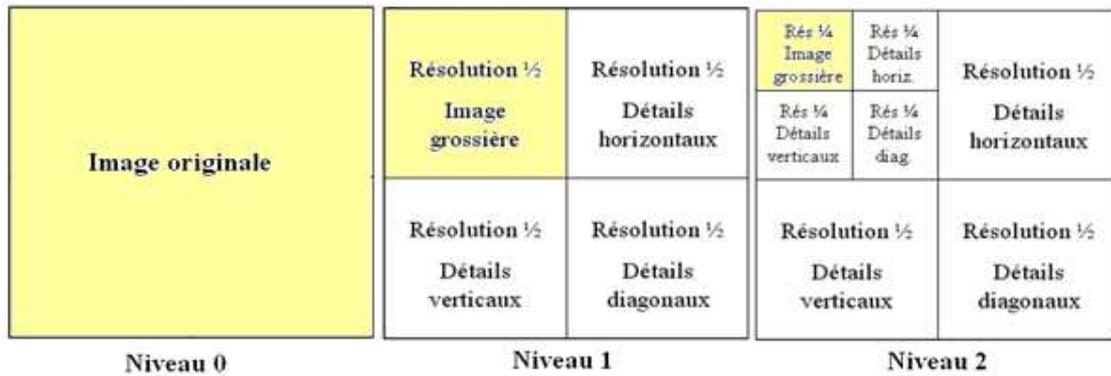


Figure 1.9 : Principe de décomposition d'une image par ondelette

Décomposition par ondelette de Haar

Pour obtenir un niveau de résolution inférieur, on moyenne les pixels de l'image originale deux à deux suivant l'axe horizontal (par exemple : $H(X) = (X_n + X_{n+1})/2$). Puis on calcule l'erreur entre l'image originale et l'image sous-échantillonnée dans le sens horizontal (par exemple : $G(X) = (X_n - X_{n+1})/2$).

Pour chacune des deux images intermédiaires, on moyenne les pixels deux à deux suivant l'axe vertical (par exemple : $H(Y) = (Y_n + Y_{n+1})/2$). Enfin pour chacune des deux images intermédiaires, calculer l'erreur suivant l'axe vertical (par exemple : $G(Y) = (Y_n - Y_{n+1})/2$) [Pop08].

Le résultat est une image d'approximation qui a une résolution divisée par 2 et trois images de détails qui donnent les erreurs entre l'image originale et l'image d'approximation (voir figure 1.10 [Zla06]).

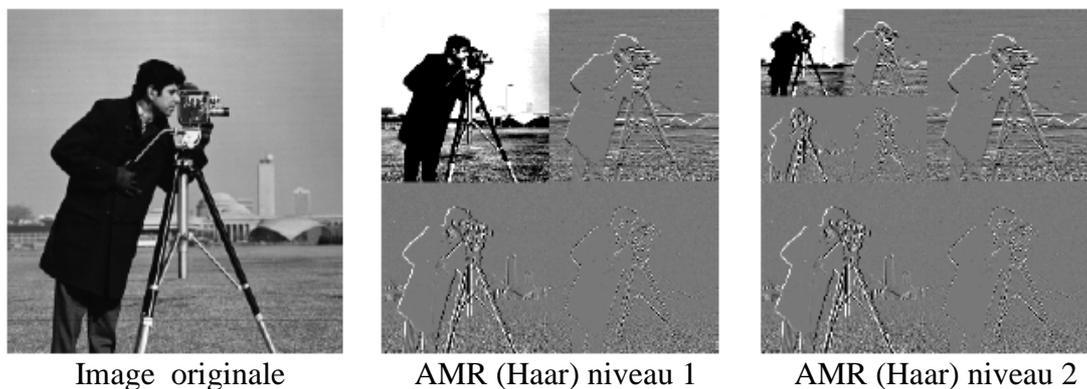


Figure 1.10 : Décomposition par ondelette de Haar

1.4 Conclusion

Nous avons présenté dans ce chapitre des techniques d'analyse multirésolution, on a commencé par les modèles pyramidales basés sur l'itération d'un filtre et d'un échantillonnage. Ensuite on a parlé des pyramides géométriques, et on a terminé par la décomposition par ondelettes.

La segmentation d'image constitue la pierre de base de tout système de vision, elle a pour objectif de fournir une description des objets contenus dans l'image. Nous aborderons dans le chapitre suivant une méthode de segmentation basée sur les contours, appelée les contours actifs paramétriques ou Snakes.

2

Les contours actifs

Sommaire

2.1 Introduction.....	18
2.2 La segmentation d'images.....	18
2.2.1 Définition.....	18
2.2.2 Les approches de la segmentation.....	18
2.3 Contours actifs.....	20
2.3.1 Principe de fonctionnement des contours actifs.....	20
2.3.2 Energies.....	22
2.3.3 Minimisation de la fonctionnelle d'énergies.....	23
2.4 Intérêts et limites des contours actifs.....	28
2.5 Les descendants du snake.....	29
2.6 Travaux Antérieurs sur la détection et le suivi d'objet.....	30
2.7 Conclusion.....	33

2.1 Introduction

La segmentation d'image peut être considérée comme un traitement de bas niveau. C'est une phase importante dans la plupart des systèmes de vision par ordinateur, c'est après la segmentation que viennent les traitements de haut niveau, tels que la description de l'image, et les décisions qui pourront être prises à partir des résultats fournis par celle-ci.

Dans ce chapitre nous présenterons de manière précise la technique de segmentation par les contours actifs paramétriques ensuite nous exposerons des travaux récents de détection et de suivi d'objets à l'aide des contours actifs.

2.2 La segmentation d'images

2.2.1 Définition

La segmentation d'images consiste à diviser les pixels d'une image en différentes régions R_1, R_2, \dots, R_n tels que les pixels de la même région sont similaires au maximum en terme de couleur, intensité...et les pixels des différentes régions sont dissimilaires, cette similarité est évaluée en prenant considération un certain critère d'homogénéité.

Les régions de l'image segmentée doivent satisfaire chacune des règles suivantes [Neb06] :

$\forall i, R_i$ est connexe

$\forall i, j, i \neq j \quad R_i \cap R_j = \emptyset$

$\cup R_i = I \quad (I : \text{image})$

$\forall i, p(R_i) = \text{vrai} \quad p : \text{prédicat d'homogénéité}$

$\forall i, j, i \neq j, R_i \text{ et } R_j \text{ étant deux régions voisines, et } P(R_i \cup R_j) = \text{faux} \quad \dots(1)$

Ce qui se traduit par :

- Les pixels appartenant à une même région sont connexes.
- Deux régions différentes sont disjointes.
- L'union de toutes les régions nous donne l'image initiale.
- Dans une région donnée, chaque pixel satisfait le critère d'homogénéité.
- Deux régions différentes sont hétérogènes.

2.2.2 Les approches de la segmentation

Pour subdiviser une image en régions distincts homogènes, deux grandes approches sont reconnues, l'approche basée région et l'approche basée contour. La segmentation par régions caractérise l'image à partir de l'homogénéité des régions qui la composent, tandis que la segmentation par contours délimite les différentes régions par leurs frontières.

Ces deux approches sont duales car une région définit une ligne sur son contour et une ligne fermée définit une région intérieure [Rou03].

2.2.2.1 Approches basées région

L'approche région fusionne directement les pixels et les régions voisines ayant une certaine propriété commune; l'ensemble final des régions fusionnées de pixels constitue l'image segmentée. Les méthodes basées segmentation en région les plus connues sont celles qui procèdent par croissance de régions ou par division/fusion de régions [Neb06].

Techniques de fusion et de partage

Les techniques de partage et de fusion de régions divisent l'image en petites régions et fusionnent itérativement les régions voisines qui satisfont un certain critère d'uniformité. La fusion la plus simple vérifie essentiellement les contraintes de ressemblances d'intensités moyennes. Des méthodes plus complexes sont les méthodes basées modèle, où les régions sont seulement fusionnées si elles sont du même modèle, et si les paramètres du modèle sont convenables. Les résultats du processus de fusion donnent de plus en plus grandes régions qui sont homogènes en tenant en considération un certain critère d'uniformité.

Segmentation d'images par croissance de région

La segmentation par croissance de région vise à fusionner les pixels les plus proches de l'image qui sont uniformes en terme de leurs attributs (la couleur, l'intensité...) en prenant en considération une certaine mesure d'uniformité. Cette approche par croissance de région doit choisir un ensemble de pixels initiaux considérés comme des germes à partir desquels la croissance des régions commencera. Le critère d'uniformité utilisé peut par exemple être la variance d'une région ou la somme des erreurs quadratiques... qui doivent ne pas dépasser un seuil prédéterminé. La croissance des régions s'effectue par agrégation de pixels aux germes initiaux, elle consiste à fusionner successivement aux régions les pixels voisins qui sont similaires, en ce sens que les régions continuent à vérifier le critère d'uniformité après chaque agrégation d'un pixel. Les régions adjacentes qui ensemble vérifient le critère d'uniformité peuvent être fusionnées et de nouveaux germes peuvent être insérés dans les régions ne pouvant pas être fusionnées avec les régions existantes [Neb06].

2.2.2.2 Approches basées contour

Cette approche s'intéresse essentiellement aux contours (frontière) des régions, ces contours sont les lieux de variation significative du niveau de gris par rapport aux pixels adjacents. On peut procéder pour cela de différentes manières [Bol04]:

- On construit directement l'image résultat contenant la frontière entre le fond et l'objet en appliquant des opérateurs de segmentation directement sur l'image (filtres de seuillage, lissage, opérateurs de dérivée, ...).

- En utilisant des méthodes qui manipulent une forme englobant l'objet à extraire et qui la font évoluer d'une position initiale vers une position finale correspondant à l'objet ciblé. Cette approche est celle des contours actifs.

La segmentation d'image par les méthodes dérivatives présente comme avantages de ne pas faire d'hypothèse sur la topologie de l'objet et ainsi isoler plusieurs zones de l'image sans augmenter la complexité. L'approche contours actifs permet d'avoir une modélisation dès le départ alors que la première méthode nécessite une construction du contour en post-traitement. Les *snakes* ont été largement étudiés dans le traitement d'images. En effet, ils constituent une méthode simple et rapide d'extraction d'objets.

2.3 Contours actifs

Les contours actifs tirent leur origine des modèles élastiques, mais la communauté s'accorde à les attribuer à l'équipe Kass, Witkin et Terzopoulos [Kas87], qui introduisirent les Snakes ou courbes minimisantes. Les snakes tiennent leur nom de leur aptitude à se déformer comme des serpents. L'approche est variationnelle, c'est-à-dire que l'équation d'évolution du contour actif se déduit de la minimisation d'une énergie modélisant l'objet d'intérêt [Gas05]. Depuis la publication de cette équipe, les modèles déformables sont devenus un sujet très important pour la communauté du traitement d'images.

2.3.1 Principe de fonctionnement des contours actifs

Un contour actif est une courbe définie dans l'image et qui est capable de se mouvoir sous l'influence de plusieurs forces qui vont la tirer ou la pousser vers la forme. Un snake consiste à placer aux alentours de l'objet à détecter une ligne initiale de contour, l'évolution se fait à travers un processus itératif qui déforme la courbe à chaque itération jusqu'à sa position finale (voir figure 2.1).

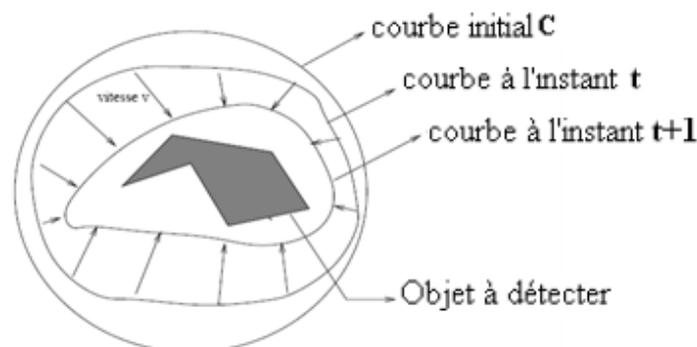


Figure 2.1 : Principe des contours actifs

Les forces impliquées vont dériver trois énergies associées au snake [Rou03] :

- Une énergie propre, due uniquement à la forme du contour, dite énergie *interne* : $E_{interne}$,
- Une énergie potentielle imposée par l'image dite énergie *externe* : $E_{externe}$. C'est elle qui va attirer la ligne du snake vers les contours réels présents sur l'image.
- Une énergie de *contexte* : $E_{contexte}$ qui exprime certaines contraintes supplémentaires qui peuvent être imposées par l'utilisateur vu le snake qu'il veut obtenir.

Ces différentes énergies seront étudiées en détail par la suite.

Mathématiquement, un contour actif est une courbe paramétrée C représenté par la notion suivante :

$$C: [a, b] \rightarrow \mathbb{R}^2$$

$$s \rightarrow v(s) = \begin{bmatrix} x(s) \\ y(s) \end{bmatrix}$$

$$C = \{v(s, t) = (x(s, t), y(s, t)) / s \in [a, b] \text{ et } t \in [0, T]\} \quad \dots(2)$$

Où s désigne l'abscisse curviligne le long du contour, a et b sont les extrémités du contour et $v(s)$ le point courant de coordonnées $x(s)$, $y(s)$ [All03].

Un snake peut prendre l'une des trois formes (voir figure 2.2) [Cou05]:

- Les contours actifs fermés ; le point de départ est celui d'arrivée,
- Les contours actifs à extrémités libre ; tous les points sont susceptibles de bouger,
- Les contours actifs à extrémités fixes ; les points de départ et d'arrivée restent fixes.



Figure 2.2 : les trois formes du snake

La fonctionnelle d'énergie attachée au contour actif est la suivante:

$$E(C) = E_{interne} + E_{externe} + E_{contexte} \quad \dots(3)$$

2.3.2 Energies

2.3.2.1 Energie interne

Appelée aussi énergie de régularisation, elle ne dépend pas de l'image ni de la forme à détourner, elle ne dépend que des points du contour, alors elle est intrinsèque au Snake. L'énergie interne gère la cohérence de la courbe, en empêchant les points individuels sur le contour de trop s'éloigner de leurs voisins. Elle se compose en général de deux termes [All03] :

$$E_{interne} = \alpha \int_a^b \left| \frac{\partial v(s,t)}{\partial s} \right|^2 ds + \beta \int_a^b \left| \frac{\partial^2 v(s,t)}{\partial s^2} \right|^2 ds \quad \dots(4)$$

Les deux termes de l'Equation (3) sont multipliés par deux coefficients de pondération, respectivement de lissage ($\alpha \geq 0$) et de rigidité ($\beta \geq 0$) qui sont utilisés pour donner plus ou moins d'influence à chaque énergie.

Le premier terme est la force de *continuité* (avec $\partial v(s, t) / \partial s$ première dérivée de la courbe par rapport à s), qui influe sur le rayon de courbure du contour en conduisant les points du contour à se positionner de manière à être équidistants. Lorsque $\alpha=0$, la courbe peut présenter des discontinuités. Dans le cas contraire, c'est-à-dire lorsque le coefficient de continuité est très élevé, le Snake sera rigide.

Le deuxième terme correspond à la force de *courbure* (avec $\partial^2 v(s, t) / \partial s^2$ dérivée seconde de la courbe par rapport à s). Son but est d'éviter que le contour contienne des points isolés qui ne seraient pas cohérents avec la forme. Lorsque $\beta=0$, la courbe peut prendre une forte convexité.

2.3.2.2 Energie externe

Les énergies externes prennent en compte les caractéristiques des images traitées. Elles attirent la ligne du snake vers les contours réels présents sur l'image.

Parmi les énergies externes existantes, on trouve : *l'énergie d'intensité* qui représente simplement la valeur du pixel de l'image transformée en niveaux de gris.

$$E_{inte} = \text{Niveaux De Gris}[x][y] \quad \dots(5)$$

Et *l'énergie de Gradient* (la dérivée première de l'image) qui est d'une importance première pour la détection du contour. Un contour est généralement caractérisé par une forte différence entre les valeurs de plusieurs pixels. Ainsi, si la dérivée d'une

fonction représente les pentes d'une courbe, le gradient montrera les fortes différences, les contours de l'image.

$$E_{grad} = -\gamma \int_a^b |\nabla I(v(s,t))|^2 ds \quad \dots(6)$$

Où ∇I est l'application de l'opérateur gradient sur l'image I , γ est un coefficient de pondération [All03].

2.3.2.3 Energie de contexte

L'énergie de contexte exprime certaines contraintes supplémentaires qui peuvent être imposées par l'utilisateur selon le Snake qu'il veut obtenir [Rou03]. On peut par exemple imposer une distance minimale ou maximale entre deux points consécutifs du contour actif.

Parmi les énergies de contexte on trouve, l'énergie de ballon introduite par [Coh91]. L'énergie de Ballon est l'énergie qui décide du sens de propagation du contour actif, Un coefficient d'énergie de ballon positif va concentrer le snake, alors qu'un coefficient négatif va rendre le snake expansif.

Alors, le contour actif est formé d'une série de points mobiles et répartis sur une courbe en deux dimensions. La courbe est placée dans la zone d'intérêt de l'image ou autour d'un objet. La courbe se déplace et épouse lentement les contours des objets en fonction de divers paramètres comme l'élasticité, ..., etc. Cette dynamique est basée sur la notion d'énergies interne, externe et de contexte, le but étant de minimiser l'énergie totale présente le long de la courbe. Des contraintes permettent de conserver une courbe lisse avec des points équidistants tout en laissant un certain champ libre pour les déformations.

2.3.3 Minimisation de la fonctionnelle d'énergies

Plusieurs méthodes ont été proposées pour résoudre le problème de minimisation de la fonctionnelle d'énergies $E(C)$. On en compte trois principales, la famille des approches variationnelles [Kas87], la famille des approches reposant sur la programmation dynamique [Ami88], l'algorithme de *Greedy* [Will92] et l'algorithme de « Fast Greedy » proposé par Lam et Yan [Lam94] qui est reconnu pour son temps de traitement rapide. Dans ce mémoire nous nous intéressons aux algorithmes de *Greedy et Fast Greedy*.

2.3.3.1 Approches variationnelles

En ne retenant que les deux premiers types d'énergie de l'équation $E(c)$ (3), c'est-à-dire la régularisation (lissage + rigidité) et l'énergie du gradient, on aboutit à la formulation des contours actifs introduite par Kass *et al* [Kas87].

$$E = \underbrace{\alpha \int_a^b \left| \frac{\partial v(s,t)}{\partial s} \right|^2 ds}_{\text{lissage}} + \underbrace{\beta \int_a^b \left| \frac{\partial^2 v(s,t)}{\partial s^2} \right|^2 ds}_{\text{rigidité}} - \underbrace{\gamma \int_a^b |\nabla I(v(s,t))|^2 ds}_{\text{énergie image}} \dots(7)$$

énergie interne

La minimisation de l'énergie totale E est effectuée en utilisant l'équation de Euler-Lagrange ; elle permet d'aboutir à l'équation différentielle dynamique régissant l'évolution du contour :

$$\kappa \frac{\partial v(s,t)}{\partial t} - \alpha \frac{\partial}{\partial s} \left(\left| \frac{\partial v(s,t)}{\partial s} \right|^2 \right) + \beta \frac{\partial^2}{\partial s^2} \left(\left| \frac{\partial^2 v(s,t)}{\partial s^2} \right|^2 \right) - \gamma |\nabla I(v(s,t))|^2 = 0 \dots(8)$$

Le snake se déplace alors suivant des forces dues à l'image (terme en ∇I), moyennant les contraintes d'élasticité et de rigidité (termes en α et β), mais son évolution est freinée par des forces de frottement visqueux (terme en κ) i.e. tout se passe comme si le contour actif était plongé dans une solution de viscosité κ . En discrétisant l'équation (8) grâce aux différences finies, et en appliquant une nouvelle fois Euler-Lagrange, on aboutit à une représentation matricielle déterminant la position des pixels du snake discrétisé à implémenter :

$$\begin{cases} X(t) = (\kappa Id + K)^{-1} \left[\kappa X(t-1) + \gamma |\nabla I(v)|^2 (t-1) \right] \\ Y(t) = (\kappa Id + K)^{-1} \left[\kappa Y(t-1) + \gamma |\nabla I(v)|^2 (t-1) \right] \end{cases} \dots(9)$$

Dans cette expression, Id représente la matrice identité, et K la matrice de déformation, qui ne dépend que des pondérations des énergies de lissage et de rigidité α et β , et qui représente le poids de l'énergie interne (ou poids de déformation) par rapport aux autres énergies [All03].

2.3.3.2 Programmation dynamique

La programmation dynamique est une méthode classique de résolution de problème d'optimisation. Son application aux contours actifs peut être une alternative intéressante au calcul variationnel. Dans [Ami88], l'auteur considère l'équation :

$$E_{\text{tot}} = \int_0^1 E_{\text{ext}}(v(s)) + \frac{1}{2} \left(\alpha(s) |v_s(s)|^2 + \beta(s) |v_{ss}(s)|^2 \right) ds = \int_0^1 E_{\text{ext}} + E_{\text{int}} \dots(10)$$

Pour un contour actif constitué de n points, on aura :

$$E_{\text{tot}} = \sum_{i=0}^{n-1} E_{\text{int}}(v_i) + E_{\text{ext}}(v_i) \dots(11)$$

Le terme d'énergie interne est composé du terme du premier degré et du terme du second degré. Après la discrétisation, cette énergie interne met en jeu un élément du contour, son prédécesseur et son successeur :

$$E_{int}(v_i) = \frac{1}{2}(\alpha_i |v_i - v_{i-1}|^2 + \beta_i |v_{i+1} - 2v_i + v_{i-1}|^2) \quad \dots(12)$$

Il est possible d'exprimer E_{tot} :

$$E_{total}(v_1, v_2, \dots, v_n) = E_1(v_1, v_2, v_3) + E_2(v_2, v_3, v_4) + \dots + E_{n-2}(v_{n-2}, v_{n-1}, v_n) \quad (13)$$

On se ramène donc à un problème d'optimisation d'une fonction numérique à plusieurs variables. Les variables seront ici les positions des différents points du snake. La formulation standard sous forme récursive de la programmation dynamique peut s'écrire :

$$S_i(v_{i+1}, v_i) = \min_{v_{i-1}} \left\{ S_{i-1}(v_i, v_{i-1}) + \alpha |v_i - v_{i-1}|^2 + \beta |v_{i+1} - 2v_i + v_{i-1}|^2 + E_{ext}(v_i) \right\} \quad \dots(14)$$

Chaque itération donne un contour optimal. La convergence de la minimisation de l'énergie est garantie mais la complexité est élevée. Cette implémentation, d'autre part, autorise l'introduction de différentes contraintes comme l'énergie ballon ou d'autres énergies.

2.3.3.3 Algorithme de Greedy

L'algorithme de Greedy consiste à faire évoluer le Snake en minimisant la fonctionnelle d'énergies $E(C)$ définie par [Rou04]:

$$E(C) = \sum_{i=1}^N (\alpha E_{continuité}(P_i) + \beta E_{courbure}(P_i) + \gamma E_{gradient}(P_i) + k E_{intensité}(p_i) + \lambda E_{ballon}(P_i))$$

$P_i : i=1..N$ les points du snake
 $\alpha, \beta, \gamma, k, \lambda$ sont des coefficients attribués à chaque énergie.

L'initialisation du contour conduit au choix des N premiers points. Le processus permet ainsi de construire une suite de points P_i , dont N points consécutifs définissent une approximation du contour.

Principe

Pour chaque point P_i du contour actif, la fonctionnelle d'énergie est calculée pour tous les points v_i appartenant au voisinage de P_i (par exemple 8 voisins (voir Figure. 8a)). Le point caractérisé par l'énergie minimale est alors choisi pour remplacer P_i si

$E(v_i) < E(P_i)$. Dans le cas contraire, le point de contour n'est pas modifié. On cherche donc l'ensemble des points pour lesquelles l'énergie est minimale, ce mécanisme est répété jusqu'à convergence.

L'algorithme de Greedy est donné comme suit :

Algorithme Greedy

- 1 : **Début**
- 2 : **Faire**
- 3 : **Pour** tous les points du Snake **faire**
- 4 : **Pour** tous les points du voisinage **faire**
- 5 : Calculer les énergies
- 6 : **Fin Pour**
- 7 : **Pour** tous les points du voisinage **faire**
- 8 : Normalisation
- 9 : **Fin Pour**
- 10 : Minimiser pour obtenir le nouveau point
- 11 : **Fin Pour**
- 12 : **Jusqu'au** Critère d'arrêt
- 13 : **Fin**

Afin d'éviter une trop grande dispersion des valeurs des énergies (certaines énergies varient entre 0 et 500 alors que d'autres entre -1 et 0), il est absolument nécessaire de normaliser, c'est-à-dire de diviser la valeur calculée par la plus grande valeur du voisinage. Cela nous permet ainsi, d'obtenir des variations comprises entre -1 et 1, pour toutes les énergies [Rou04].

Donc l'algorithme de Greedy est un algorithme itératif qui déplace un point unique dans son propre voisinage, pour constituer un nouveau contour actif. Tous les points sont traités successivement lors de chaque itération.

Les critères d'arrêts

Les critères d'arrêt sont très importants, notamment lorsque l'on travaille avec des images réelles. En effet, il faut parfois préciser au snake quand s'arrêter car, dans certains cas, le contour actif s'arrête pendant un moment sur les contours de l'image puis dépasse ceux-ci, attirée par des voisins dont l'énergie est plus faible. Parmi les critères d'arrêt du snake :

- Moins de x % des points du snake se sont déplacés entre deux itérations.
- La moyenne des déplacements sur une itération est inférieure à x pixel(s).
- La variation de l'énergie est inférieure à x % sur les six dernières itérations [Rou04].

2.3.3.4 Algorithme de Fast Greedy

De nombreuses améliorations ont été apportées à l'algorithme de Greedy, toutes étant basées sur l'optimisation de la fenêtre de voisinage. Le but est de réduire l'espace de recherche, en calculant l'énergie à des endroits sélectionnés [Mil07].

Dans [Lam94], les auteurs ont proposé un algorithme appelé « Fast Greedy » qui utilise des fenêtres dont les directions sont alternées, une avec les voisins cardinaux et une avec les voisins diagonaux. La différence entre l'algorithme de « Greedy » et l'algorithme de « Fast Greedy », se trouve donc dans le nombre de voisins utilisés pour faire évoluer le contour actif.

Pour chaque point du Snake, et pour un voisinage 3×3 par exemple, l'algorithme de « Fast Greedy » peut être résumé par les étapes suivantes :

1. Calculer l'énergie $E(c)$ pour le point du Snake et ses quatre voisins cardinaux (voir Figure 2.3 b)
2. Normalisation des énergies calculées en 1.
3. Si l'un des quatre voisins cardinaux possède une énergie inférieure ou égale à celle du point du Snake, alors il n'est pas nécessaire d'examiner les quatre voisins qui restent et allez à 6.
4. Calculer l'énergie $E(c)$ pour les voisins diagonaux (voir Figure 2.3c).
5. Normalisation des énergies calculées en 4.
6. Déplacer le point du snake vers le voisin qui minimise l'énergie.

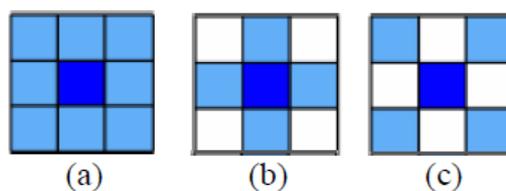


Figure 2.3 : Voisinage. (a) voisinage 3×3 (b) voisinage cardinaux
(c) voisinage diagonaux

Le Fast Greedy examine donc que quatre voisins. Si l'un de ces quatre améliore l'énergie totale, alors il n'est pas nécessaire d'aller plus loin. Sinon, les quatre qui restent sont examinés. Cela décroît le temps de calcul de chaque itération. Pour cela il est apprécié pour son temps de traitement rapide par rapport à l'algorithme de Greedy.

Un autre algorithme est proposé par Sakalli et al [Sly06], il offre des améliorations de temps par rapport aux approches précédente. L'algorithme suit les mêmes étapes que [Lam94] en utilisant des pas différents (figure 2.4c).

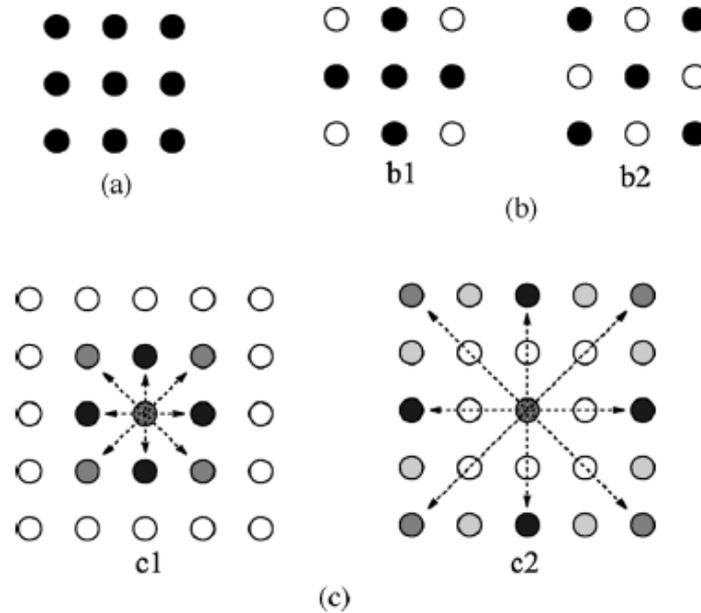


Figure 2.4 : Les points noir représentent les pixels examinés. (a) algorithme de Greedy. (b) algorithme Fast Greedy [Lam94]. (c) Fenêtres alternées de l'approche de Sakalli et al

2.4 Intérêts et limites des contours actifs

Les contours actifs que nous venons d'introduire, permettent de segmenter efficacement des images de différentes natures sans connaissance a priori sur la géométrie des objets à extraire.

Pour résumer, on peut dire qu'un contour actif a simplement besoin d'être initialisé, puis qu'il est autonome dans sa recherche de l'état d'énergie minimale. Cette propriété est très utile, notamment dans des applications particulières comme la le suivi d'objets en mouvement dans les séquences vidéo.

Malgré tous les avantages qu'il propose, le modèle du Snake soulève un certain nombre de questions :

- *Le choix crucial de la position initial* : si le contour actif est placé trop loin des frontières de l'objet, il sera pas attiré par celles-ci (la portée du gradient dans l'image n'est pas assez étendue) ;
- *Le choix des poids à attribuer à chaque énergie* (c'est-à-dire les valeurs des coefficients α , β , γ , k , λ .) ; ceci est d'autant plus critique que leur nombre est important [All03].
- *Le problème de la topologie* : un snake ne permet pas de gérer les changements de topologie au cours du temps (voir figure 2.5). Il restera une unique courbe fermée et sera incapable de détecter plusieurs objets sur une image (Surtout si on veut détecter deux objets qui sont trop proches).

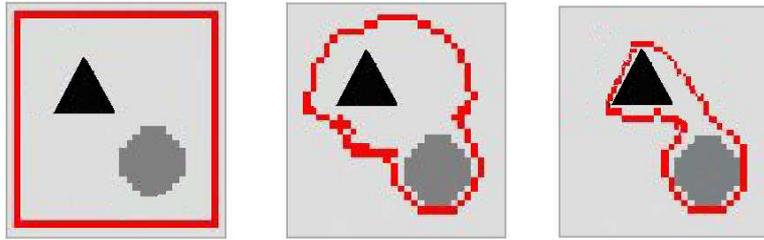


Figure 2.5 : Le snake ne peut se scinder en deux pour délimiter correctement deux objets dans l'image

2.5 Les descendants du snake

Afin de palier aux problèmes rencontrés avec la méthode des snakes [Kas87], Stanley Osher et James Sethian en 1988 [Osh88] ont développé une méthode alternative aux contours actifs appelée « Les Level Set » ou « ensembles de niveau ». Les auteurs ont débuté par une étude de la propagation de front séparant deux milieux dans le domaine de la physique. Contrairement aux contours actifs, ces méthodes ne se focalisent pas sur une collection de point qui se déplacent, mais elles prennent en compte l'évolution globale du milieu.

L'idée est de construire une hyper-surface dont l'intersection avec le plan contenant l'image donne la courbe à différents pas de temps. Pour segmenter une image avec les level-sets, il faut définir une vitesse de propagation du level-set en utilisant des caractéristiques de l'image. La fonction de vitesse doit être élevée lorsque la courbe est loin du contour et elle doit tendre vers zéro lorsque la courbe se rapproche de la frontière à segmenter. De cette façon, la courbe cessera d'évoluer une fois arrivée sur le contour et mettra fin à la segmentation de l'image

En 1999 sethian [Set 99] a proposé une autre approche qui est « le Fast Marching ». La différence entre les méthodes dites Level Set et Fast Marching vient du sens de la propagation.

Dans l'approche Fast Marching, le front n'évolue que dans un sens, et donc il ne croise chaque point de la grille qu'une seule fois, contrairement aux Level Set, où le sens de propagation peut changer.

Pour les deux approches, les problèmes liés aux changements de topologie sont naturellement gérés par l'utilisation de l'hypersurface.

Caselles et al. [Cas93, Cas95, Cas97] ont proposés les contours actifs « géodésiques ». Les auteurs ont tenté de résoudre le problème d'initialisation et de paramétrisation du contour, leurs approches intègrent au calcul d'énergie l'information de distance à l'objet. L'énergie est la longueur du contour dans une métrique non euclidienne dépendant de l'image. Les contours actifs géodésiques se rapprochent de la détection de contours par snakes, mais ils ont pour avantages de pouvoir détecter plusieurs objets en même temps lorsqu'ils sont combinés avec les

ensembles de niveau. L'inconvénient majeur des contours actifs géodésique est leurs grandes sensibilités aux bruits [All03].

[Dav95] Ont montré que les Snakes ont des difficultés à progresser dans les frontières concaves. Pour résoudre ce problème, Xu et Prince proposent dans [Xu97] un nouveau champ de forces : le flux de vecteurs gradients (ou gradient vector flow - GVF, en anglais). Un «GVF» est un champ de vecteurs invariant dans le temps et indépendant de la position du snake dans l'image, il s'agit d'une grandeur à calculer une fois pour toutes en début de traitement, c'est-à-dire avant de lancer le processus de convergence. L'approche contour par la méthode de GVF se divise en deux étapes : (i) Le calcul de la carte de contours (Edge Map) qui permet de caractériser les zones d'intérêt dans l'image ; (ii) Le calcul de flux du vecteur du gradient qui minimise l'énergie.

Un GVF permet à la fois de conserver les propriétés du gradient près des frontières, de diffuser cette information dans les régions homogènes de l'image et de gérer les zones concaves. Il ne nécessite pas un contour initial proche de l'objet à détecter. Grâce à la largeur de la gamme de capture du champ GVF. Le principal inconvénient de l'approche GVF est son coût en temps de calcul.

Parallèlement, d'autres équipes de recherche ont proposé d'autres solutions aux problèmes posés par les snakes. En 1990 Menet et al [Men90], ont introduit les « B-snakes » en étendant le modèle des snakes, à une structure de contour utilisant les fonctions de B-spline. Les propriétés de régularité de telles courbes permettaient de s'affranchir des problèmes liés à l'évaluation des grandeurs géométriques du contour actif. En effet, la formulation paramétrique de la courbe fournit, de façon immédiate, les expressions analytiques de ces grandeurs et en permet un calcul exact pour une complexité inférieure.

Plusieurs travaux, dont ceux de Leitner et al. En 1993 [Lei93], ont par la suite étendu les fonctionnalités des contours B-snakes en définissant des contours déformables qui puissent changer de topologie [Pre04].

2.6 Travaux Antérieurs sur la détection et le suivi d'objet

La principale motivation de l'utilisation des contours actifs est la localisation et le suivi d'objets. Un des domaines qui a fortement contribué au développement des contours actifs est l'imagerie médicale.

Parmi les approches qui ont été proposées pour *la détection d'objet* : [Kha05] ont développé une approche de segmentation basée sur les contours actifs pour la localisation du ventricule gauche dans une séquence d'images scintigraphiques du coeur. L'approche consiste à initialiser un contour par une suite de nœuds soumis à des forces externes qui représentent les caractéristiques de l'image et à des forces internes d'élasticité et de rigidité qui assurent une cohérence pendant l'évolution du contour initial. La minimisation d'énergies a été réalisée par l'algorithme de programmation dynamique qui a apporté une nette amélioration des résultats obtenus par programmation classique. Ensuite les auteurs ont implémenté la méthode du GVF-

snake (vu dans la section précédente) qui a marquée une nette amélioration pour la segmentation des objets dans les zones à fortes concavités.

Les auteurs de [Dan05] ont décrit une méthode permettant de détecter automatiquement les contours du foie en IRM qui est intégrée dans une étude de la quantification de la graisse hépatique.

L'initialisation du contour est réalisée de manière automatique par l'étude d'un l'histogramme qui est estimé par une courbe de Rayleigh et des gaussiennes. Le premier pic de l'histogramme est estimé par la courbe de Rayleigh et représente le fond bruité de l'image.

A partir de la courbe de Rayleigh, les auteurs ont défini un seuil qui permet de masquer le fond de l'image. Le seuillage et différents filtrages sont appliqués successivement pour obtenir un contour initial. Ce contour sert d'initialisation au modèle déformable qui est ensuite attiré vers les frontières de la forme à détecter suivant différentes forces internes et externes dont la force du flux de vecteurs de gradient et une force ballon. L'initialisation proposée fournit un contour proche du contour réel ce qui a augmenté la convergence du modèle.

En 2005 [Mou05] ont appliqué la méthode des contours actifs pour la détection des contours des thrombus veineux dans des images ultrasonores acquises à l'aide d'une sonde échographique. Initialement l'opérateur choisit un niveau de seuil pour binariser l'image, applique une ouverture morphologique et désigne un centre d'initialisation du modèle déformable pour déterminer le contour du thrombus.

La détection de contour dépend de la détermination d'un certain nombre de points de contrôle à l'intérieur de l'objet qui est délimité par deux cercles (par l'opérateur). Le contour est filtré par un filtre médian pour éliminer les points non significatifs. Enfin une interpolation par spline cubique a été réalisée pour donner le contour désiré.

[Bou09] ont développé une méthode de segmentation d'images par contour actif (B-spline Snake) pour la détection des cavités cardiaque dans les images échographiques cardiaques. L'initialisation du snake est réalisée de deux manières, automatique et manuelle. Pour l'évolution du snake les auteurs ont utilisé quatre énergies (continuité, courbure, gradient et carte de distance). [Ash11] ont proposé une nouvelle approche de segmentation basée sur les contours actifs et l'alignement topologique. L'approche comporte deux étapes : la première est la réalisation d'une segmentation d'image par la méthode d'alignements topologique pour améliorer la performance du suivi de la cellule ; la deuxième étape utilise les résultats de la première segmentation, qui seront considérés comme entrée du Snake. La méthode a été testée sur des images biologiques « des cellules » et a donné une bonne détection d'objet.

Le suivi temps-réel d'un objet dans une séquence d'images reste encore un problème sensible quand il s'agit d'obtenir des résultats précis tout en tenant compte des

différents mouvements possibles. La résolution du problème du suivi d'un objet dans une séquence d'vidéo revient à détecter l'objet dans chaque image constituant la vidéo, et à déterminer le mouvement de l'objet d'une image I_n à l'image I_{n+1} . Dans ce contexte différents approches ont été développées parmi lesquelles: [His03] ont proposé une méthode de suivi des déformations intra-myocardiques en IRM cardiaque marquée. L'approche utilise les contours actifs dont l'énergie image se construit grâce à une diffusion anisotrope non linéaire paramétrée. Cette méthode à permet d'obtenir des résultats satisfaisants à la fois en terme de précision et de reproductibilité. [Ray02] ont appliqué les contours actifs pour suivre le mouvement des leucocytes (cellules sanguines, globules blanc). La nouveauté de la proposition réside dans la fonctionnelle d'énergie qui impose des contraintes de forme et de taille sur le contour actif. Une nouvelle énergie est proposée basée sur l'échantillonnage implicite des points sur le contour actif. Les auteurs ont pu suivre des leucocytes roulant à des vitesses élevées. Toujours dans l'imagerie biologique, les auteurs de [Zim02] ont développé une méthode pour suivre des cellules biologique, sur une vidéomicroscopie par les contours actifs paramétrique. Pour détecter les objets à faible contrast, les auteurs ont utilisé une carte de contours des dérivations d'intensité locale. La carte de contours comporte les différences d'intensité entre le fond et les cellules. Cette méthode a donnée de bonne résultat de suivi même lorsque les objets sont très proche.

[Fek09a] ont proposé une méthode de détection et de suivi d'un objet dans une séquence d'images. Une fonctionnelle d'énergies est attachée au contour actif, elle comporte quatre énergies : continuité, courbure, gradient et ballon. Après une initialisation du contour actif dans la première image de la séquence, une minimisation d'énergies attachées est effectuée afin de détecter le contour. Puis une initialisation automatique du contour actif dans les images suivantes est proposée pour le suivi dans les autres images de la séquence. Elle consiste à utiliser le barycentre de l'objet détecté. La méthode à donnée de bonne résultat pour trois types de séquences d'images : synthétiques, biologiques et échocardiographique. Les mêmes auteurs [Fek09b], ont proposé une autre méthode qui consiste à modifié la fonctionnelle d'énergies attaché au snake dont le but de réduire le temps de suivi. Ils ont remplacé l'énergie de ballon par une nouvelle énergie qui est l'énergie directionnelle. Elle permet de suivre la direction du mouvement des points du contour.

Les auteurs de [Li11] présentent une méthode optimale de contour actif paramétrique basée sur les descripteurs de Fourier qui permet de détecter et de suivre en temps réel un contour dans une séquence d'images. La combinaison entre Snake et descripteur de Fourier permet de suivre à la fois les contours convexes et concaves. Cette méthode permet également de réinitialiser automatiquement le contour actif lorsque des changements de topologie sont détectés au cours du suivi. Le principe général est d'utiliser l'algorithme d'extraction de région pour initialiser automatiquement le contour actif. Tout d'abord, le médecin indique dans l'image initiale l'élément anatomique à suivre en cliquant avec sa souris simplement sur un

point appartenant à la section qui l'intéresse. Ce point définit le germe de l'algorithme d'extraction de région qui fournit l'ensemble des points de contours permettant d'initialiser le contour actif proche du contour réel de l'objet à suivre. La méthode a été testée sur une séquence d'images échographiques et a donné de bons résultats de suivi.

2.7 Conclusion

Dans ce chapitre nous nous sommes intéressés aux contours actifs paramétriques qui sont des courbes capables de détecter les frontières des objets en réponse à trois types d'énergies; énergies internes, énergies externes et énergies de contexte. Plusieurs travaux ont été présentés montrant l'importance de cet outil pour la détection et le suivi des objets dans des séquences d'images médicales.

Dans ce mémoire nous proposons d'apporter des améliorations en terme de temps de suivi à l'approche [Fek09a] décrite précédemment dans la section 2.6. La méthode proposée est fondée sur un environnement multi-agents, pour cela dans le chapitre suivant nous présenterons les systèmes multi-agents.

3

Agent et Système multi-Agents

Sommaire

3.1 Introduction.....	35
3.2 De l'intelligence artificielle distribuée (IAD) aux systèmes multi-agents.....	35
3.3 Le concept d'agent	36
3.3.1 Définitions.....	36
3.3.2 Principales caractéristiques d'un agent.....	36
3.3.3 Typologies des agents.....	37
3.4 Les systèmes multi-agents.....	40
3.4.1 Définition.....	40
3.4.2 Caractéristiques d'un SMA.....	41
3.4.3 L'interaction dans les SMA.....	42
3.4.4 Les avantages des agents et les systèmes multi-agents.....	45
3.5 Systèmes multi-agents et segmentation.....	46
3.6 Conclusion	48

3.1 Introduction

Actuellement, la technologie Agent retient de plus en plus l'attention de la communauté informatique. Ce phénomène est dû principalement à l'incessante montée des applications qui nécessite des traitements parallèles de plus en plus complexes.

Dans ce chapitre, nous présentons un aperçu de l'intelligence artificielle distribuée, ensuite nous verrons les concepts fondamentaux de la notion d'agent. Puis nous mettons l'accent sur l'évolution de l'aspect individuel (le comportement d'un agent seul) vers l'aspect collectif (son comportement dans une société d'agents). À la fin, nous donnons un aperçu de quelques approches qui utilisent les systèmes multi-agents dans le domaine de segmentation d'images.

3.2 De l'intelligence artificielle distribuée (IAD) aux systèmes multi-agents

L'Intelligence Artificielle Distribuée est née, au début des années 80 [Fer95], son but est de concevoir des approches de développement de communautés d'agents intelligents qui puissent interagir de façon coopérative mais aussi conflictuelle ou concurrente afin de résoudre des problèmes complexes.

Les systèmes de vision sont composés d'un grand nombre d'entités traduisant des connaissances hétérogènes, tant descriptives qu'opératoires. Dans le but de fournir un cadre favorisant une intégration harmonieuse de ces composants, ils utilisent l'une des approches de l'Intelligence Artificielle Distribuée comme architecture de contrôle. Les justifications d'un tel choix sont nombreuses :

1. *Préserver la modularité et l'ouverture* : garantir la capacité d'améliorer et d'ajouter de nouveaux agents, sans dégrader les performances du reste du système ou provoquer la réécriture massive du code.
2. *Flexibilité* : le contrôle doit être suffisamment flexible pour permettre, d'une part, une évolution en fonction des agents présents et, d'autre part, installer dynamiquement des boucles de contrôle entre eux.
3. *Calcul distribué* : l'IAD permet de traiter les problèmes qui sont par nature physiquement et/ou fonctionnellement distribués.
4. *Robustesse* : la fusion de l'analyse de plusieurs agents permet d'améliorer la robustesse.
5. *Focalisation et adaptation des connaissances* : la focalisation et l'adaptation des connaissances peuvent offrir une adaptation des paramètres en fonction d'un contexte local de l'image; ou la focalisation des agents sur des régions riches en information.

L'intelligence artificielle distribuée a conduit à la réalisation de systèmes multi-agents qui permettent de modéliser le comportement d'un ensemble d'entités plus ou moins expertes, plus ou moins organisées selon des lois de type social. Ces entités, ou agents, disposent d'une certaine autonomie et sont émergés dans un environnement dans lequel ,et avec lequel elles interagissent. D'où leurs structures autour de trois fonctions principales : celles de percevoir, de décider et d'agir [Lag10].

Avant d'aborder les systèmes multi-agents, il est utile d'évoquer quelques notions de base sur les agents, c'est l'objet de la section suivante.

3.3 Le concept d'agent

3.3.1 Définitions

Selon Wooldridge [Woo02], « Un agent est un système informatique capable d'agir de manière autonome et flexible dans un environnement changeant ».

Pour Ferber [Fer95], Un agent est une entité située, réelle ou virtuelle, agissant dans un environnement, capable de le percevoir, d'agir sur celui-ci et d'interagir avec les différents composants l'entourant. Une entité est un agent si elle est capable d'exercer un contrôle local sur ses processus de perception, de communication, d'acquisition de connaissances, de raisonnement, de prise de décision et d'exécution.

A partir de ces définitions, nous pouvons définir un agent comme :

- une entité autonome qui peut offrir des services,
- une entité dont le comportement est la conséquence de ses objectifs, de sa perception, de ses représentations, de ses compétences et des communications qu'elle peut avoir les autres agents,
- une entité qui possède des ressources,
- une entité qui est apte à agir sur l'environnement du système auquel il appartient,
- une entité qui peut communiquer avec les autres agents,
- une entité qui est capable de se reproduire.

3.3.2 Principales caractéristiques d'un agent

Un agent doit posséder les caractéristiques suivantes :

Autonomie

- Un agent a un certain degré d'autonomie.
- Il possède certains états (non-accessibles aux autres agents et composants du système).
- Il peut prendre certaines décisions par rapport à ses états (sans intervention externe directe).

Situé

- Un agent est situé dans son environnement (physique ou virtuel).
- Il a une représentation de son environnement.

Réactif

- Un agent peut percevoir son environnement via des senseurs.
- Il peut agir sur son environnement via des effecteurs.

Social

- Un agent est capable d'interagir et de communiquer avec les autres agents (par des langages de communication).
- Il est capable de coopérer pour résoudre des problèmes ou effectuer des tâches.

Proactif

- Un agent est capable de « prendre de l'initiative » pour atteindre son but ou effectuer des tâches (et d'adopter les comportements appropriés).

Actif

- Un agent est toujours actif. Il s'exécute donc nécessairement dans un *thread* ou un *process* indépendants.

Apprentissage

- Un agent est capable d'apprendre et d'évoluer en fonction de cet apprentissage.
- Il est capable de changer de comportement (en fonction des expériences passées).

3.3.3 Typologies des agents

Il existe deux grandes écoles de pensée dans la communauté des agents : l'école cognitive qui conçoit les agents comme des entités intelligentes et l'école réactive qui conçoit les agents comme des entités très simples réagissant directement aux modifications de l'environnement. D'autres types d'agents sont ensuite apparus ; les agents hybrides qui utilisent ces deux types de comportements [Rou09].

3.3.3.1 Agents cognitifs

Les agents cognitifs sont généralement « intentionnels » c'est-à-dire qu'ils possèdent des objectifs et des plans explicites leur permettant d'accomplir leurs buts. Ils agissent suivant un cycle (perception/décision/action). La Figure 3.1 représente un agent cognitif [Lag10].

Ces agents disposent d'une base de connaissances comprenant les diverses informations liées à leurs domaines d'expertise et à la gestion des interactions avec les autres agents et leur environnement. Cette base de connaissance regroupe ce qu'on appelle les états mentaux de l'agent. Un état mental ou cognition est une structure cognitive élémentaire, il concerne les croyances, les désirs et les intentions.

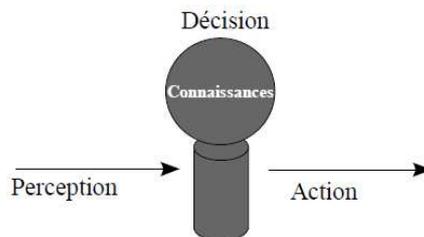


Figure 3.1 : Agent Cognitif

On peut définir ces trois notions comme suit :

- *Les croyances* : ce sont les informations que l'agent possède à propos du monde qui l'entoure.
- *L'intention* : c'est une attitude orientée vers l'action : elle va conduire l'agent à chercher les moyens et à mener des actions pour satisfaire cette intention. Si l'agent a pour but, entre autres, de satisfaire d'autres agents, il pourra prendre si on le lui demande, des engagements vis-à-vis des demandeurs sur l'exécution d'une tâche.
- *Le désir* : c'est une attitude voisine de l'intention. Cependant, sa durée de validité est plus courte. Le désir par opposition à l'intention est une attitude changeante à court terme. De plus, le désir n'engage pas l'agent à l'action : mais sous certaines conditions, le désir peut devenir intention.

Un système cognitif est composé d'un petit nombre d'agents "intelligents". Dans ce genre de système, les problèmes vont être résolus grâce aux compétences de chaque agent indépendamment des autres et par leur aptitude à coordonner leurs actions ainsi qu'à leur coopération [Zar07].

3.3.3.2 Agents réactifs

Comme son nom l'indique, un agent réactif ne fait que réagir aux changements qui surviennent dans l'environnement. Autrement dit, un tel agent se contente simplement

d'acquérir des perceptions et de réagir à celles-ci en appliquant certaines règles prédéfinies (voir. figure 3.2).

Un agent réactif n'est donc pas possédant de représentations symboliques de son environnement, ni de lui-même. Un système d'agents réactifs peut présenter un comportement intelligent ou satisfaisant à un but, c'est le phénomène d'émergence. En ce sens, le contrôle du comportement d'un agent est donc en partie effectué par l'environnement. L'intérêt d'un agent réactif est uniquement l'interaction avec les autres agents.

Étant donné qu'il n'y a pratiquement pas de raisonnement, ces agents peuvent agir et réagir très rapidement.

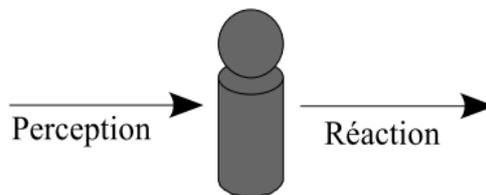


Figure 3.2 : Agent Réactif

Le tableau 1 résume les différentes propriétés des deux approches [Rou09] :

Systèmes d'agents cognitifs	Systèmes d'agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
L'agent peut tenir compte de son passé	Pas de mémoire de son historique
Agents complexes	Fonctionnement stimulus/réponse
Petit nombre d'agents	Grand nombre d'agents

Tableau 3. 1 : Les agents cognitifs et réactifs

Pour conclure, les agents cognitifs, du fait de leur autonomie et de leur intelligence, peuvent résoudre des problèmes beaucoup plus complexes que les agents réactifs.

En outre, les agents cognitifs sont capables d'anticiper sur des actions et de prévoir des événements, contrairement aux agents réactifs qui ne possèdent pas de représentation explicite de leur environnement [Zar07].

3.3.3.3 Agent hybride

C'est une architecture composée d'un ensemble de modules organisés dans une hiérarchie chaque module étant soit une composante cognitive, soit une réactive. Les agents hybrides sont conçus pour combiner des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de l'environnement [Mou11].

De cette manière le comportement proactif de l'agent dirigé par les buts, et combiné avec un comportement réactif afin d'obtenir simultanément les avantages des architectures cognitives et réactives, tout en éliminant leurs limitations.

3.4 Les systèmes multi-agents

Les systèmes multi-agents s'appuient sur le principe suivant : au lieu d'avoir un seul agent en charge de l'intégralité d'un problème, on considère plusieurs agents qui n'ont chacun en charge qu'une partie de ce problème. La solution au problème initial est alors obtenue au travers de l'ensemble des comportements individuels et des interactions, c'est à dire par une résolution collective.

3.4.1 Définition

Un système multi-agents (avec son acronyme SMA, et MAS pour *Multi-Agent System* en anglais) est constitué d'un ensemble de processus informatiques se déroulant en même temps, donc d'un ensemble de plusieurs agents vivant au même moment, partageant des ressources communes et communiquant entre eux. Le point clé des systèmes multi-agents réside dans la formalisation de la coordination entre les agents (voir figure 3.3) [Lag10].

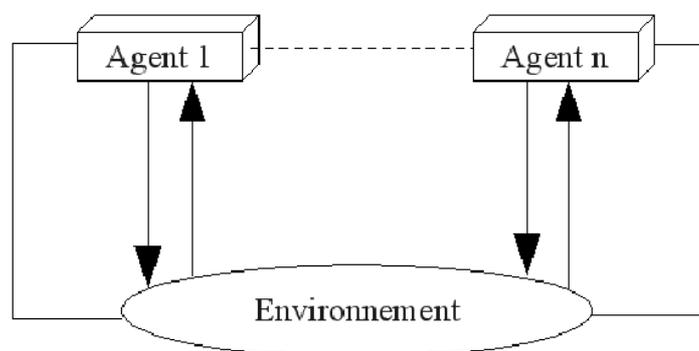


Figure 3.3: Représentation schématique d'un SMA, du point de vue d'un observateur extérieur

Les agents sont capables de le percevoir et d'agir sur l'environnement commun qu'ils partagent. Les perceptions permettent aux agents d'acquérir des informations sur l'évolution de leur environnement, et leurs actions leur permettent entre autres de le modifier.

Un système multi-agents est composé des éléments suivants:

1. Un environnement **E**, c'est-à-dire un espace disposant généralement d'une métrique.
2. Un ensemble d'objets **O** situé dans **E**. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
3. Un ensemble **A** d'agents, qui représentent les entités actives du système.
4. Un ensemble de relations **R** qui unissent des objets (et donc des agents) entre eux.
5. Un ensemble d'opérations **Op** permettant aux agents de **A** de percevoir, produire, consommer, transformer et manipuler des objets de **O**.
6. Des opérateurs chargés de représenter l'application de ces opérations et la réaction de l'environnement envers les tentatives de modification.

3.4.2 Caractéristiques d'un SMA

Généralement, un SMA possède les caractéristiques suivantes [Rou09]:

- Les agents agissent et travaillent indépendamment les uns des autres.
- Chaque agent est une partie du système.
- Chaque agent travaille dans le but d'accomplir ses tâches.
- Chaque agent est capable de communiquer et d'interagir avec d'autres agents.
- Un agent coopère avec les autres agents lorsque nécessaire.
- Un agent est capable de coordonner ses activités avec les autres agents pour accéder à des ressources et à des services partagés dont il a besoin (pour réaliser ses buts).
- Les agents ont un but commun.
- Chaque agent a une vue partielle du SMA.

3.4.3 L'interaction dans les SMA

Les interactions entre agents sont cruciales dans l'élaboration d'un système multi-agents. Selon Ferber [Ferb95], une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'actions réciproques.

Principalement, un agent interagit avec les membres de son environnement pour trois raisons principales :

- Partager de l'information ;
- Atteindre ses buts ;
- Eviter, autant que possible, les conflits.

Il existe différents types d'interactions que les agents peuvent utiliser comme la coordination, la coopération et la communication [Rou09].

3.4.3.1 Coordination

La coordination détermine en quelque sorte quelles sont les règles de bon fonctionnement du système auquel les agents appartiennent. Selon la nature des agents et des interactions, plusieurs formes de coordination sont possibles.

La coordination consiste à organiser la solution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées.

La coordination peut être imposée à un agent par une sorte de contrat, comme le respect d'un protocole d'interaction donné. Elle peut aussi être apprise, lorsque l'agent trouve un intérêt à participer à la collectivité [Rou09].

3.4.3.2 Coopération

La coopération consiste à établir qui fait quoi, avec quel moyen, de quelle manière et avec qui. C'est simplement le fait que le comportement du groupe permet d'atteindre un certain objectif. Du point de vue des agents, la coopération consiste à se mettre d'accord sur les actions que chacun doit effectuer pour que la combinaison de ces actions permette d'atteindre un but commun [Zar07].

On distingue trois types de coopération [voir. figure 3.4] :

1. *La coopération confrontative* : plusieurs agents de différentes spécialités exécutent une même tâche (sur le même ensemble de données), de manière concurrente, ainsi le résultat est obtenu par fusion.

2. *La coopération augmentative* : Ici, on a une collection d'agents de compétences similaires, et qui travaillent en parallèle sur des sous-ensembles disjoints de données. La solution est obtenue sous la forme d'un ensemble de solutions locales.

3. *La coopération intégrative* selon laquelle une tâche est décomposée en sous-tâches accomplies par des agents de spécialités différentes et œuvrant de manière coordonnée. La solution est obtenue au terme de leur exécution [Lag10].

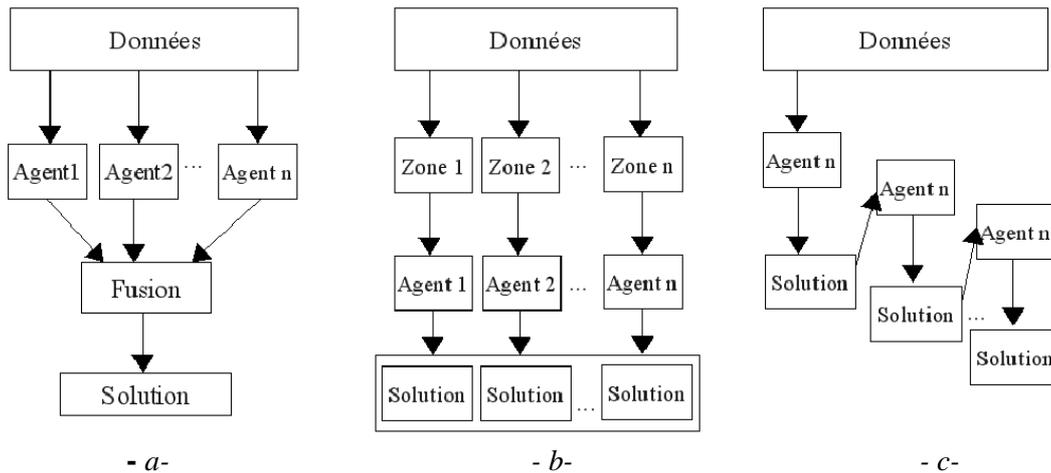


Figure 3.4: Les différents modes de coopération.

a- coopération confrontative, b- coopération augmentative, c- coopération intégrative.

3.4.3.3 Communication

La communication désigne l'ensemble des processus physiques et psychologiques par lesquels s'effectue l'opération de mise en relation d'un émetteur avec un ou plusieurs récepteurs, dans l'intention d'atteindre certains objectifs.

Les communications dans les SMA comme chez les humains sont à la base des interactions et de l'organisation. Sans communication, l'agent n'est qu'un individu isolé. C'est parce que les agents communiquent qu'ils peuvent coopérer, coordonner leurs actions et réaliser des tâches en commun [Bou11].

- *Pourquoi communiquer ?* les agents communiquent et interagissent pour synchroniser leurs actions et pour résoudre des conflits, qui sont des conflits de ressources, de buts ou d'intérêts. Ils communiquent également pour s'aider mutuellement.

- *Quand et avec qui communiquer ?* Pour répondre à cette question, il faut identifier les situations qui vont nécessiter la communication des agents. En général, les agents communiquent lorsqu'ils sont face à un problème qu'ils ne savent pas résoudre (soit par manque de compétences ou de ressources), lorsqu'il est nécessaire de coordonner leurs actions, ou encore lorsqu'il y a un conflit entre plusieurs agents et que le conflit ne peut pas être résolu de façon déterministe.

Les communications peuvent être diffusées à l'ensemble des agents ou à des agents particuliers (des agents susceptibles d'être intéressés par le message).

- *Comment communiquer ?* Les procédures de communication pour véhiculer les messages (qui sont porteurs d'informations ou d'actions) entre agents sont la communication par envoi de messages, la communication par partage d'informations [Kab99].

a. Communication par partage d'informations

C'est historiquement, le premier modèle de communication qui est apparu au début des années 60. Les agents peuvent communiquer via une structure de données partagée (tableau noir (blackboard)), représentant l'état courant du problème. Un Agent agit sur cette structure afin de résoudre le problème [Noo98].

Dans ce mode de communication, toute l'information est centralisée dans une structure de données globale. Les agents viennent lire et écrire dans cette base pour faire évoluer le système qui contient initialement les données du problème.

Le fonctionnement global du système n'est pas purement multi-agents puisque le comportement d'un des agents est dépendant du contenu d'une base de connaissances commune.

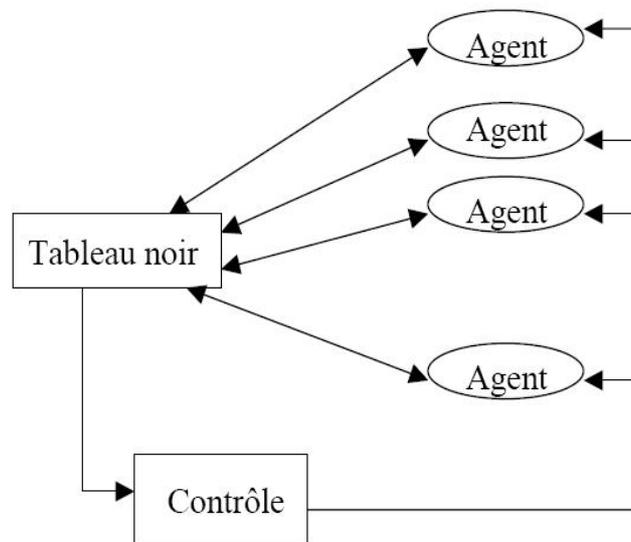


Figure 3.5: Communication par partage d'informations.

b. Communication par envoi de messages

Ce type de communication permet aux agents d'envoyer leurs messages directement aux destinataires. La seule contrainte est la connaissance de l'agent destinataire:

Si un agent A connaît l'agent B, alors il peut entrer en communication avec lui.

Les systèmes fondés sur la communication par envoi de messages relèvent d'une distribution totale à la fois de la connaissance, des résultats et des méthodes utilisées pour la résolution du problème.

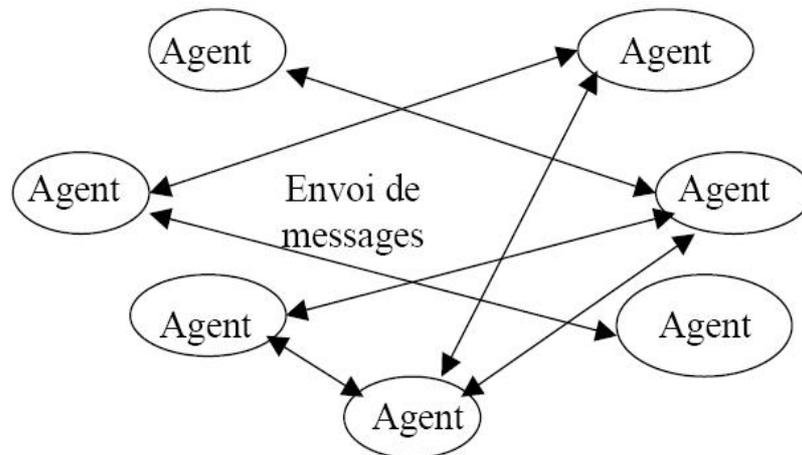


Figure 3.6: Communication par envoi de messages.

3.4.4 Les avantages des agents et les systèmes multi-agents

Ils possèdent les avantages traditionnels de la résolution distribuée et Concurrente de problèmes :

– **La modularité** permet de rendre la programmation plus simple. Elle permet, de plus, aux systèmes multi agents d'être facilement extensibles, parce qu'il est plus facile d'ajouter de nouveaux agents à un système multi agent que d'ajouter de nouvelles capacités à un système monolithique.

– **La vitesse** est principalement due au parallélisme, car plusieurs agents peuvent travailler en même temps pour la résolution d'un problème.

– **La fiabilité** peut être également atteinte, dans la mesure où le contrôle et les responsabilités étant partagés entre les différents agents, le système peut tolérer la

défaillance d'un ou de plusieurs agents. Si une seule entité contrôle tout, alors une seule défaillance de cette entité fera en sorte que tout le système tombera en panne.

- **Et d'autres avantages** : Distribué ; Parallèle ; Flexible ; Adaptatif ; Récursif et facile [Gue11].

3.5 Systèmes multi-agents et segmentation

Les systèmes multi-agents sont utilisés dans un grand nombre de méthodes de segmentation d'images. Dans ce qui suit, nous allons exposer brièvement quelques travaux utilisant les SMA en segmentation d'images.

L'approche proposée par Y. Jacquelet et al [Jac01] est une méthode de segmentation d'image, basée sur une architecture pyramidale et sur les concepts agent. Les interactions locales d'agents, assurent une segmentation progressive par fusions pertinentes de régions. L'aspect novateur de ce travail réside, dans la transposition du graphe d'adjacence issu de la pyramide irrégulière adaptative en un réseau d'agents accointants, favorisant ainsi au mieux les choix de fusion. Le traitement pyramidal agent a recours à deux sortes de coopération, la première met en œuvre une interaction entre agents régions tandis que la seconde une collaboration entre les agents régions et les agents contours. L'approche est appliquée sur des images de mammographie.

Les travaux de [Ric03] sont destinés à la segmentation des tissus cérébraux en matière blanche, matière grise et liquide céphalo-rachidien. Des agents situés dans l'image travaillent en coopération. Il existe plusieurs types d'agents :

- un agent de contrôle global,
- plusieurs agents de contrôle local,
- et des agents de segmentation qui effectuent une croissance de régions, chacun de ces agents concerne l'un des trois tissus.

Les auteurs de [Idi05] ont proposé une méthode hybride de segmentation d'image par une approche multi-agent basée sur une pyramide irrégulière duale. Une approche incrémentale est utilisée pour pré-segmenter l'image et dont le résultat représente le niveau zéro de la pyramide. Les agents régions-contours vont coopérer d'un niveau à un autre de la pyramide afin d'assurer la convergence de la segmentation. Différents types d'agents sont proposés pour ce système:

- Des agents permettant le contrôle et le bon fonctionnement du système multi-agents ;
- Des agents régions et des agents contours: Chaque agent est lié à une primitive, région ou contour, d'une image. Il dispose de toutes les informations pertinentes à son sujet (exemples: moyenne et compacité de la région, taille et continuité du contour).

Les travaux de Haroun et al. dans [Har05] portent sur la segmentation d'image IRM du cerveau. Les auteurs y présentent une approche coopérative mettant en œuvre deux algorithmes de segmentation d'image, que sont la FCM « Fuzzy C-Means » (pour gérer l'incertitude et l'imprécision) et la croissance de région (pour agir localement sur l'image). Un système multi-agents est introduit dans la phase de croissance de région afin d'améliorer la qualité de la segmentation. Il existe deux types d'agents, un agent de type "contrôleur" et plusieurs agents de type "Croissance" déployés sur l'image. Noriko Matsumoto et al [Mat07] ont proposé une approche Multi-Snakes, qui réalise la détection des frontières des objets avec une collaboration entre des couches micro agents et macro agents. Où Chaque point de snake représente un micro agent. La coopération dans un ensemble de micro agents construit le comportement d'un macro agent. Ce mécanisme rend l'opération de détection plus souple et dynamique. L'approche proposée a été testée sur des images réelles et a donné des résultats de détection robuste.

Les auteurs de [Fleu09] ont développé une méthode de segmentation permettant une détection multi-objets, appliquée à l'extraction de structures cardiaques en imagerie scanner multi-barettes. L'approche proposée repose sur l'élaboration d'un schéma multi-agents combiné à une méthode de classification supervisée qui permet l'introduction d'a priori dans le processus de segmentation ainsi que des temps de calcul rapides. Le système multi-agents proposé produit une segmentation d'images 3D basée sur une étape initiale de sélection, durant laquelle l'utilisateur pointe interactivement un ou plusieurs points germes à l'intérieur de chacun des objets d'intérêt. Le système multi-agents proposé est composé :

- d'un ensemble d'agents purement situés appelés agents « Workers », qui coopèrent et entrent en compétition de manière à, par croissance de région, maximiser leur « territoire image » (selon la vérification d'un critère de texture et d'intensité) ;
- d'un agent purement communicant (appelé agent « Contrôleur ») qui coordonne les comportements de tous les agents « Workers ».

L'approche proposée par [Mou11] est une méthode de segmentation multi-agents appliquée aux IRM 3D. Dans ce travail, l'auteur s'est focalisé sur des modèles Multi-Agents basés sur les comportements biologiques des araignées et des fourmis pour effectuer la tâche de segmentation. Pour les araignées, l'auteur a proposé une approche semi-automatique utilisant l'histogramme de l'image pour déterminer le nombre d'objets à détecter. Tandis que pour les fourmis, l'auteur a proposé deux approches : la première dite classique qui utilise le gradient de l'image et la deuxième, plus originale, qui utilise une partition intervoxel de l'image. L'auteur a également proposé un moyen pour accélérer le processus de segmentation grâce à l'utilisation des GPU (Graphics Processing Unit). Ces deux méthodes ont été évaluées sur des images d'IRM de cerveau et elles ont été comparées aux méthodes classiques de segmentation : croissance de régions et Otsu pour le modèle des araignées et le gradient de Sobel pour les fourmis.

3.6 Conclusion

Dans ce chapitre, nous avons présenté une vue générale sur le domaine des Systèmes multi-agents. Le concept de base de ce domaine est la notion d'agent, qui représente une entité autonome capable de percevoir, de se représenter et d'agir sur son environnement.

Les avantages des SMA déjà cités dans ce chapitre, permettent de les utiliser dans la segmentation d'image.

Concernant notre travail, l'idée est d'utiliser le concept Agent pour avoir un suivi rapide d'un objet dans une séquence d'images médicales tout en assurant une bonne détection d'objet par les contours actifs.

Le prochain chapitre sera consacré à la présentation de l'approche que nous avons développée.

4

Conception et Implémentation

Sommaire

4.1 Introduction.....	50
4.2 Approche proposée.....	50
4.2.1 Prétraitement.....	51
4.2.1.1 Filtrage.....	51
4.2.1.2 Calcul du gradient.....	52
4.2.2 Agents.....	52
4.2.2.1 Agent Pyramide	53
4.2.2.2 Agent Snake... ..	53
4.2.2.3 Agent Projection.....	59
4.3 Expérimentations.....	60
4.3.1 Séquence d'images.....	60
4.3.2 Environnement matériel et logiciel.....	60
4.3.3 Résultats.....	60
4.4 Conclusion.....	74

4.1 Introduction

On a vu durant les trois premiers chapitres l'essentiel de notions et de technique auxquelles nous aurons recours durant le reste de ce mémoire. Dans le deuxième chapitre nous avons présenté les contours actifs, leurs caractéristiques et leurs importances dans le domaine du suivi d'objets en mouvements qui a connu une activité de recherche considérable.

Dans le présent chapitre nous commencerons par une présentation de l'approche proposée qui a comme but d'avoir un suivi robuste et rapide d'un objet dans une séquence d'images médicales. L'approche proposée définit trois agents qui s'exécutent en parallèles et qui utilisent la multirésolution pyramidale afin de décomposer les images de la séquence en des résolutions multiples, et les contours actifs paramétriques pour détecter l'objet à suivre dans toute la séquence d'images.

Une fois la méthode de suivi définis, la phase la plus importante réside dans sa mise en oeuvre. Cette mise en oeuvre permettra d'étudier le comportement de la méthode proposée à travers une série d'expérimentations. Donc la deuxième partie de ce chapitre est consacrée aux résultats expérimentaux de notre travail.

4.2 Approche proposée

L'approche proposée [Med11] [Med12] commence par l'application d'un certain nombre de prétraitements aux images de la séquence pour réduire le bruit tout en préservant la forme des structures présentes dans les images et pour extraire l'information du gradient contenue dans celles-ci.

Ensuite débute notre méthode de détection et de suivi d'objet qui comprend trois étapes. Ces étapes peuvent être résumées comme suit :

- La première étape consiste à réduire les images originaux par un processus de multirésolution pyramidale jusqu'à un certain niveau de résolution.
- La deuxième étape concerne la détection d'objet dans les images à faible résolution (les sommets des pyramides) par les contours actifs ou Snake.
- La troisième étape correspond à la projection des contours détectés dans les images à faible résolution vers les images à haute résolution (les images originales).

Ce procédé de descente en résolution a pour effet *d'atténuer le bruit* dans les images originales qui peuvent gêner l'évolution de la courbe pour détecter le contour de l'objet et il *accélère la convergence* du snake vers les frontières de l'objet (dans les images à faible résolution).

Afin d'avoir un suivi rapide d'un objet dans toute la séquence d'images, nous avons défini trois Agents. Chacun d'eux exécute une des trois étapes en parallèle avec les autres.

Le premier Agent (Agent Pyramide) réalise la décomposition des images de la séquence en multirésolution.

Dès qu'il termine la décomposition de la première image de la séquence, le deuxième agent (Agent Snake) démarre l'opération de détection d'objet dans les images à faible résolution par les contours actifs. Le troisième agent (Agent Projection) réalise l'opération de projection des points des snakes vers les niveaux inférieurs des pyramides en parallèle avec les deux autres agents.

Les étapes de l'approche proposée peuvent être schématisées comme suit:

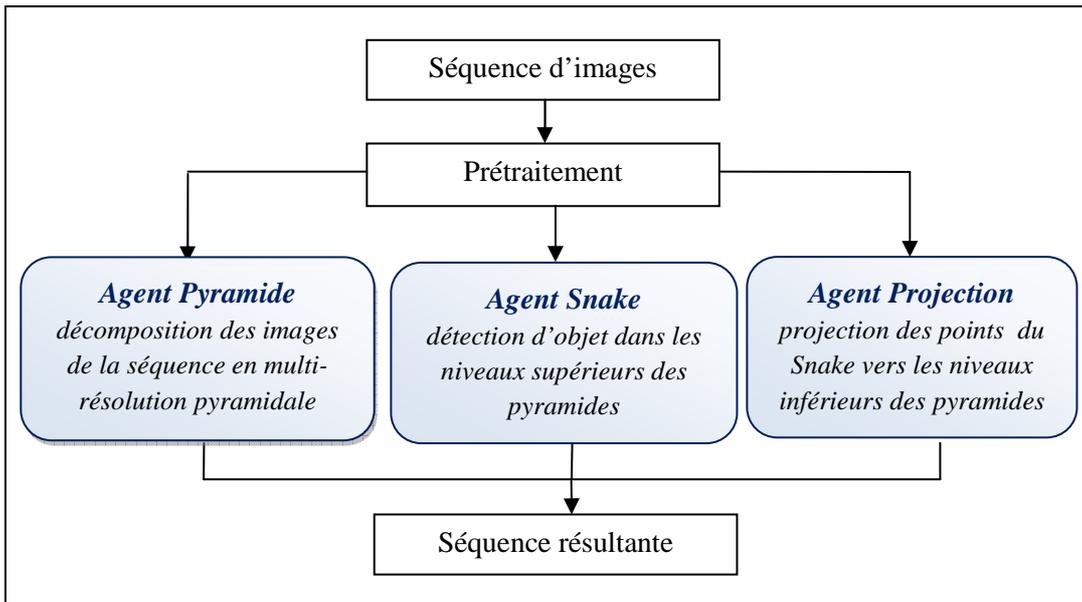


Figure 4.1 : Etapes de l'approche proposée

4.2.1 Prétraitement

4.2.1.1 Filtrage

Le filtrage est une opération qui consiste à réduire le bruit contenu dans une image. Nous avons utilisé le filtre Médian qui peut se résumer de la façon suivante :

1: Début

2: pour toute l'image $I(x, y)$ **faire**

3: -On range les pixels voisins et le pixel courant dans un tableau en ordre croissant.

- Dans la nouvelle image, on remplace le pixel courant par la valeur située au milieu du tableau.

4: Fin pour

5: Fin

Algorithme 4.1 : Algorithme filtre médian.

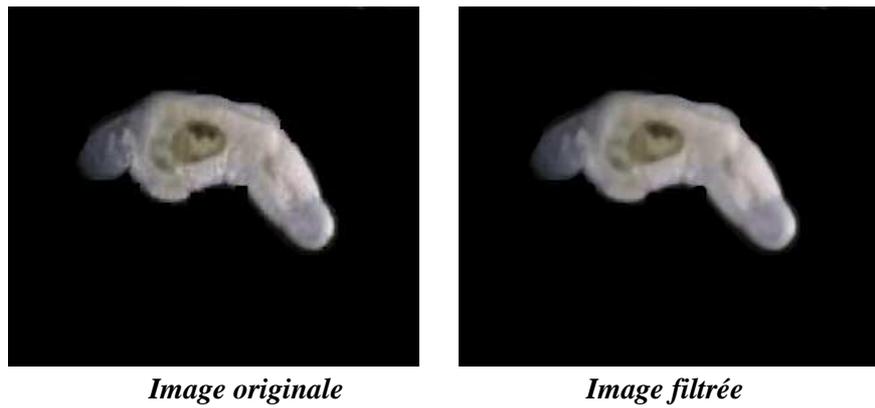


Figure 4.2 : Elimination de bruit par le filtre médian.

4.2.1.2 Calcul du gradient

L'extraction de contours est une fonction qui permet d'alléger les processus de reconnaissance d'objet. Il existe plusieurs opérateurs de convolution pour calculer le gradient d'une image, nous avons utilisé l'opérateur de Sobel.

4.2.2 Agents

Comme nous avons vu dans la section 4.2, l'approche proposée comporte trois agents qui s'exécutent en parallèle (voir figure 4.3) :

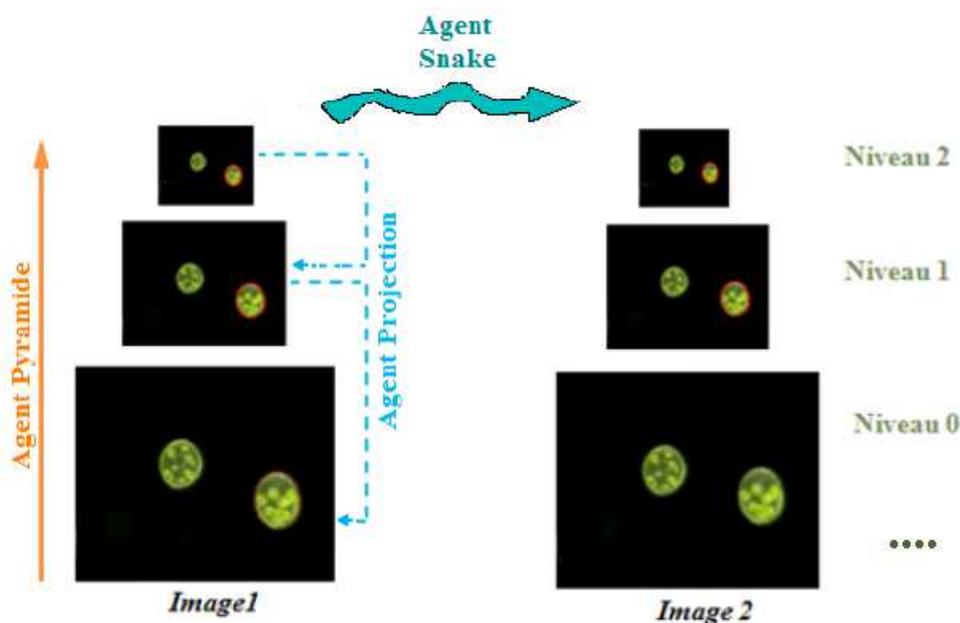


Figure 4.3 : Fonctionnement des agents sur des pyramides de trois niveaux

4.2.2.1 Agent Pyramide

L'Agent pyramide décompose les images de la séquence une après l'autre en multirésolution. Nous avons le choix entre deux techniques de décomposition pyramidale : la pyramide gaussienne et la pyramide moyenne. Ces deux techniques ont été présentées dans le premier chapitre.

Le nombre de pyramides construites est égale au nombre d'images dans la séquence (voir figure 4.4).

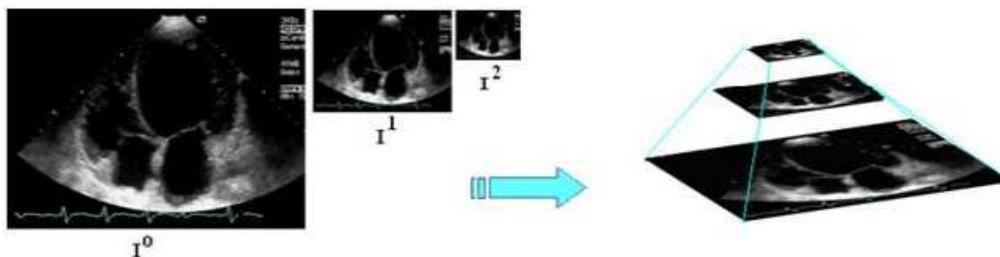


Figure 4.4 : Décomposition par pyramide Gaussienne

Lorsque l'Agent pyramide termine la décomposition en multirésolution de la première image de la séquence, il envoie l'image du sommet de la pyramide au deuxième agent (Agent Snake) pour qu'il démarre l'opération de détection d'objet.

4.2.2.2 Agent Snake

Le deuxième Agent détecte l'objet à suivre dans les images à basse résolution par les contours actifs paramétriques (présentés dans le deuxième chapitre).

L'agent Snake débute par une initialisation d'un contour dans la première image réduite, ensuite il commence à évoluer pour converger vers les frontières de la forme à localiser.

Après avoir détecté l'objet, l'agent Snake passe à la deuxième image qui se trouve au sommet de la deuxième pyramide. Pour l'initialisation il garde la même valeur du rayon et il calcule un autre centre d'initialisation puisque l'objet est en mouvement. Cette initialisation est effectuée de manière automatique.

Et ainsi de suite jusqu'à la détection de l'objet dans tous les sommets des pyramides.

Le travail de cet agent est organisé comme suit :

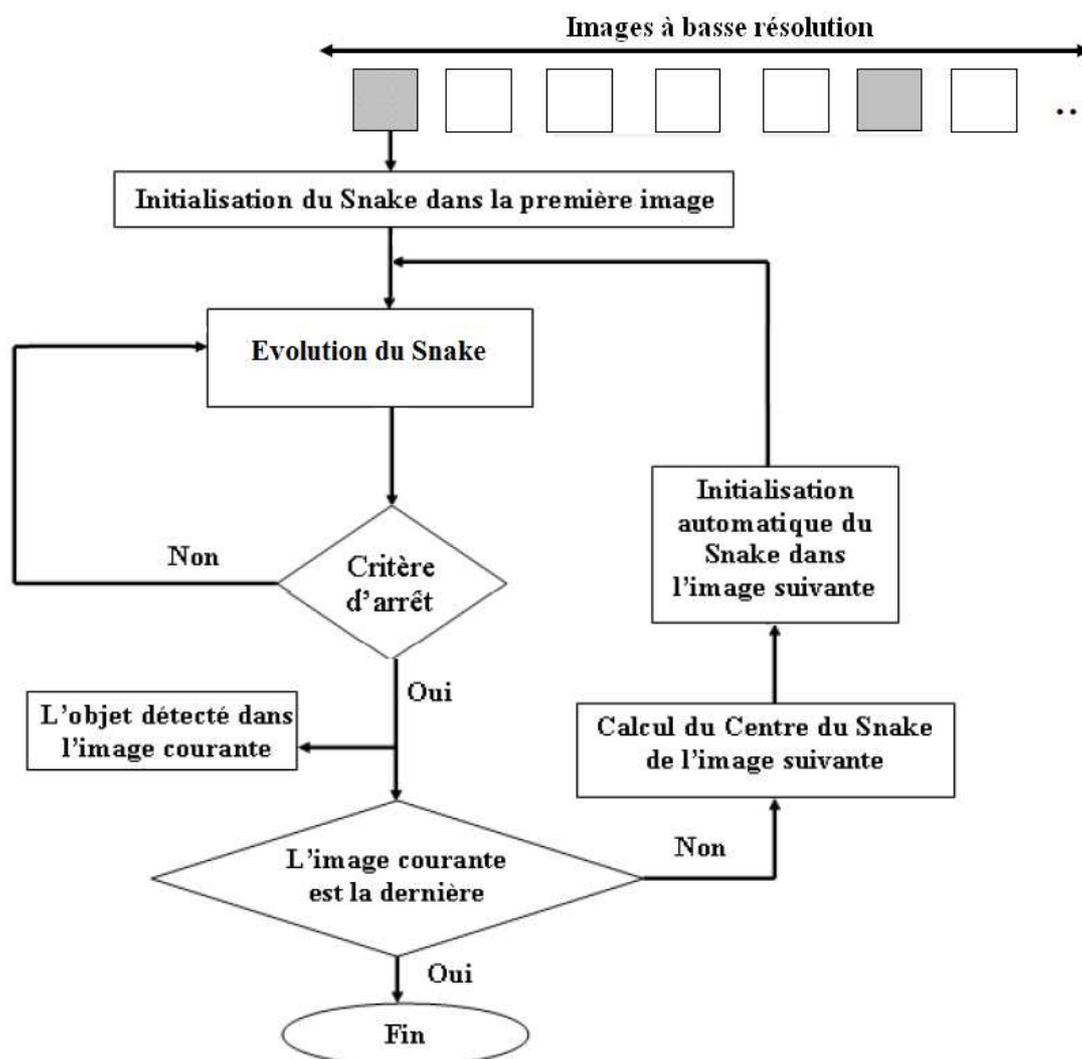


Figure 4.5 : Organigramme qui représente le travail de l'agent Snake

4.2.2.2.1 Initialisation du snake dans la première image de la séquence

L'Agent Snake cherche à détecter l'objet qui se trouve au sommet de la première pyramide. Pour cela, une initialisation d'un Snake est effectuée manuellement sous forme d'un cercle, cette initialisation est réalisée par la détermination d'un rayon et d'un point du centre de coordonnées (x_c, y_c) .

Le cercle est divisé en certain nombre de points N afin d'obtenir des angles similaires a . Cette division garantie l'équidistance entre les points tel que :

$$\begin{aligned}
 X_i &= x_c - \sin(i \times a) \times \text{rayon} \\
 Y_i &= y_c - \cos(i \times a) \times \text{rayon} \\
 &\text{avec } i=0 \dots N
 \end{aligned}$$

La position du snake et sa taille sont déterminées de telle sorte qu'il englobe l'objet qu'on veut segmenter ou bien qu'il se positionne à l'intérieur de cet objet (voir figure 4.6).

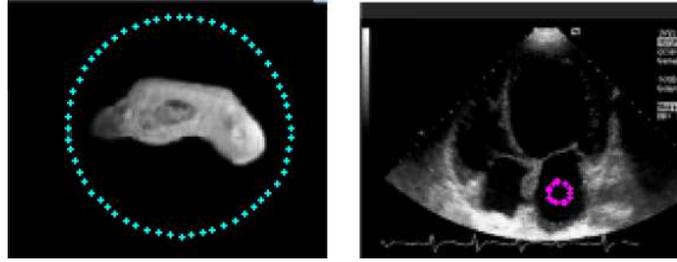


Figure 4.6 : Initialisation du snake

4.2.2.2 Evolution du Snake

a- Calcul des énergies

Nous rappelons que le snake consiste à placer aux alentours de la forme à détecter une ligne initiale de contour. Cette ligne va se déformer progressivement selon l'action de plusieurs forces qui vont la tirer ou la pousser vers la forme.

Les forces impliquées vont dériver trois énergies associées au snake. La fonctionnelle d'énergie attachée au snake est :

$$E(C) = \alpha E_{continuit} + \beta E_{courbure} + \gamma E_{gradient} + \lambda E_{ballon}$$

$P_i : i=1..N$ les points du snake

$\alpha, \beta, \gamma, \lambda$ sont des coefficients attribués à chaque énergie.

Chacun des termes correspond à un type de force particulier, nous avons introduit quatre terme dans l'énergie qui doit être minimisée le long du contour, donc l'évolution du contour s'obtient par minimisation de cette fonctionnelle.

Energie continuité

L'énergie $E_{con}(n_i)$ est définie pour tout point n_i dans le Snake comme la valeur absolue de la différence entre deux distances, la distance moyenne entre deux points successifs du contour (calculée à chaque itération) et la distance du point précédent n_{i-1} de contour au point n_i . Telle que :

$$E_{con}(n_i) = 0.5 \text{ abs (Distance Moyenne) - } \frac{\sqrt{((X_{point}(n_{i-1}) - X_{point}(n_i))^2 + (Y_{point}(n_{i-1}) - Y_{point}(n_i))^2)}}{2}$$

Nous remarquons que pour minimiser cette expression, le point n_{i-1} doit se positionner à une distance égale à la distance moyenne (Distance Moyenne) du point n_i .

Energie courbure

L'énergie de courbure est donnée par la relation suivante :

$$E_{cou}(n_i) = 0.5 ((X_{point}(n_{i-1}) - 2X_{point}(n_i) + X_{point}(n_{i+1})))^2 + (Y_{point}(n_{i-1}) - 2Y_{point}(n_i) + Y_{point}(n_{i+1})))^2$$

Energie gradient

Pour obtenir le gradient d'une image, donc d'une matrice de pixels, nous avons utilisé l'opérateur de Sobel cité précédemment. Chaque pixel ou chaque case de la matrice image sera calculé en fonction de ses voisins pondérés par une matrice de coefficients déterminés.

Energie ballon

Pour calculer l'énergie de Ballon, nous étudions la distance entre un point du snake et le barycentre des différents points composant le Snake.

$$E_{ballon} = \text{abs}(\sqrt{(PointX - MoyenneX)^2 + (PointY - MoyenneY)^2} \times \sqrt{(PointX - Point_{avant}X)^2 + (PointY - Point_{avant}Y)^2})$$

Nous remarquons que pour minimiser cette énergie, le point (n_i), aura tendance à vouloir atteindre le barycentre du snake.

b-Minimisation d'énergies

Pour résoudre le problème de minimisation de la fonctionnelle d'énergies $E(C)$, nous utilisons deux algorithmes (décrit dans le deuxième chapitre).

Le premier est l'algorithme de Greedy, et le deuxième est l'algorithme de « Fast Greedy » qui est une amélioration de l'algorithme de « Greedy ».

Les deux algorithmes sont itératifs, ils déplacent un point unique dans un certain voisinage pour constituer un nouveau contour actif à chaque itération.

Pour mieux comprendre le fonctionnement des deux algorithmes (pour un voisinage 3x3), voici ci-dessous leurs organigrammes :

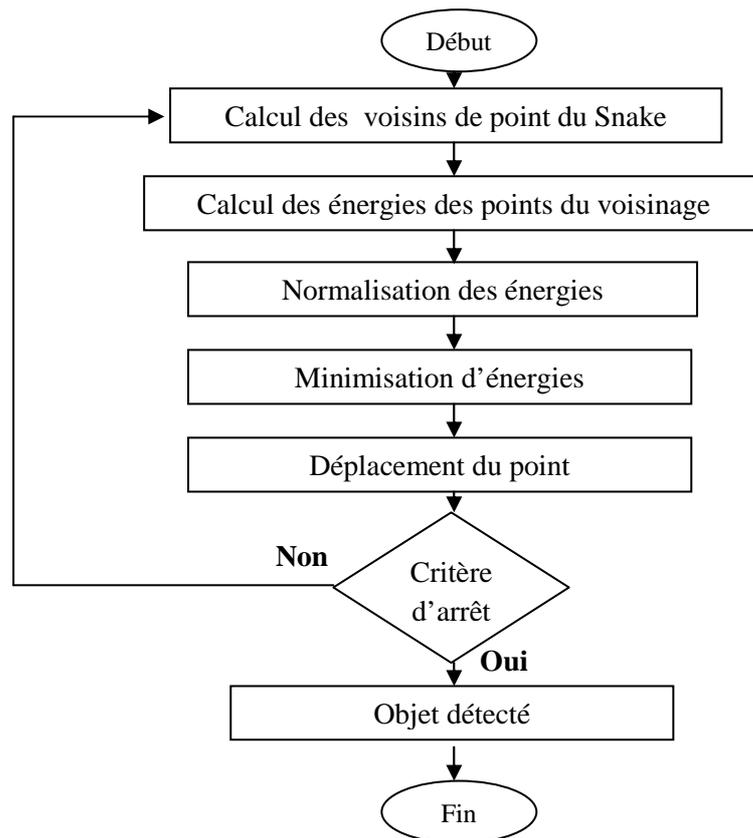


Figure 4.7 : Algorithme de Greedy

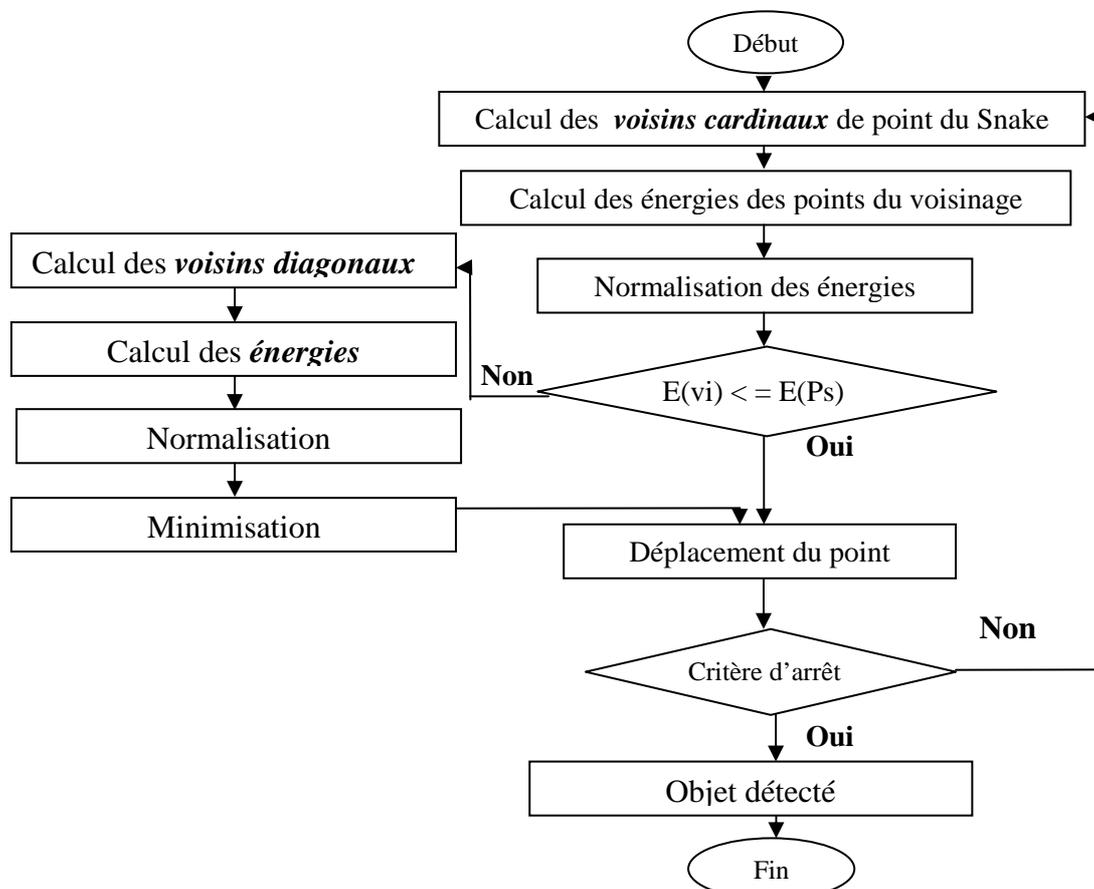


Figure 4.8 : Algorithme de Fast Greedy

Le Fast Greedy minimise la fenêtre du voisinage par rapport à l'algorithme de Greedy. Par exemple pour un voisinage (3×3), si l'un des voisins cardinaux améliore l'énergie totale, alors on n'examine pas les voisins diagonaux. Sinon les quatre voisins qui restent sont traités.

Cela augmente le nombre d'itérations pour atteindre la convergence mais décroît le temps de calcul de chaque itération.

4.2.2.3 Critère d'arrêt du contour actif

Comme nous l'avions vu dans le deuxième chapitre, le snake s'arrête en vérifiant un critère d'arrêt, que nous représentons par le pourcentage du nombre de point qui ne bouge pas pendant un certain nombre d'itération.

Si cette condition d'arrêt n'est pas vérifiée, le snake continue alors à se déformer jusqu'à atteindre un certain nombre d'itération.

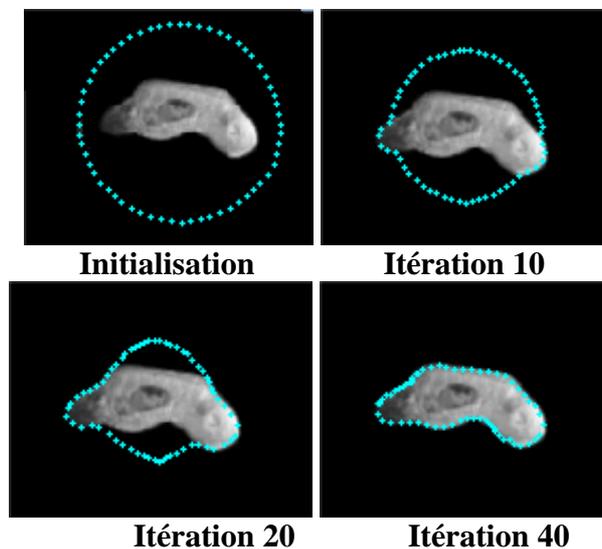


Figure 4.9 : Evolution du Snake

4.2.2.4 Calculer le centre de l'objet détecté

Après avoir détecté l'objet dans la première image, nous voulons le localiser sur les autres images de la séquence. Pour cela on calcule le centre de l'objet détecté dans la première image par l'algorithme suivant :

1: Début

2: pour tout point du snake faire

3: - Calculer la somme des X_i .

- Calculer la somme des Y_i .

4: Fin pour

5: x_c = la somme des X_i / nombre de points du snake.

6: y_c = la somme des Y_i / nombre de points du snake.

7: Fin

Algorithme calcul du centre d'initialisation

Les coordonnées obtenues par cet algorithme (x_c, y_c) seront le centre d'une initialisation automatique dans l'image suivante, voir la figure ci dessous.

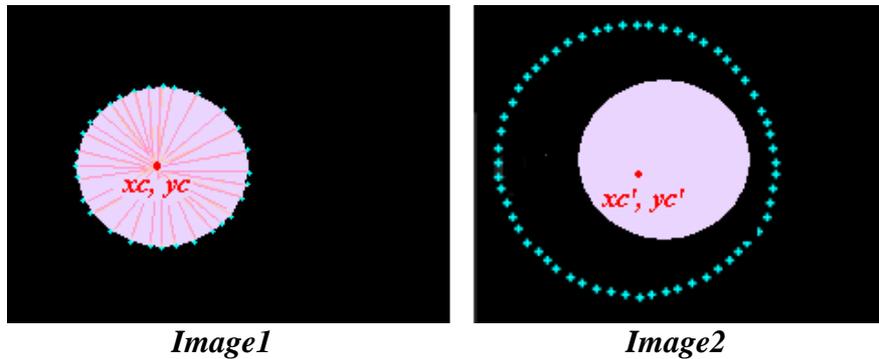


Figure 4.10 : *Changement du centre d'initialisation,*
 $(x_c, y_c)=(67, 116) \rightarrow (x_c', y_c')=(62, 113)$

L'agent Snake envoie un message au troisième agent, à chaque fois qu'il détecte un objet. Ce message contient les coordonnées des points du Snake trouvées aux niveaux supérieurs (les sommets des pyramides) qui vont être utilisées par l'Agent Projection.

4.2.2.3 Agent Projection

Cette étape est initialisée par le contour détecté à basse résolution, et on se propose de le ramener à la résolution supérieure et ceci de manière itérative jusqu'à accéder à la résolution de l'image originale.

L'*Agent Projection* permet donc de déplacer les points des Snakes des images à basse résolution (niveaux supérieurs) vers les images à haute Résolution (niveaux inférieurs).

Pour afficher chaque point du Snake de l'image du niveau « i » sur l'image de niveau « i-1 », nous avons suivi l'algorithme ci-dessous:

-
- 1 : **Début**
 - 2 : **Pour** tous les points du Snake de l'image de niveau « i » **faire**
 - 3 : Calculer l'emplacement du point « p » à l'image de niveau « i-1 »
 - 4 : **Pour** tous les points du voisinage du point « p » **faire**
 - 5 : Calculer les énergies
 - 6 : **Fin Pour**
 - 7 : **Pour** tous les points du voisinage du point « p » **faire**
 - 8 : Normalisation
 - 9 : **Fin Pour**
 - 10: Minimiser pour obtenir le point du snake au niveau « i-1 »
 - 11: Afficher le nouveau point du snake sur l'image du niveau « i-1 »
 - 12: **Fin Pour**
 - 13: **Fin**

Algorithme de Projection des points du Snake

4.3. Expérimentations

4.3.1 Séquence d'images

Afin d'extraire une séquence d'images à partir d'une vidéo nous avons utilisé le logiciel VirtualDub, (téléchargeable depuis le site <http://www.virtualdub.com>). Les images de la séquence obtenue sont stockées dans un fichier BMP.

4.3.2 Environnement matériel et logiciel

Pour expérimenter l'approche proposée, nous avons utilisé une machine de type Pentium 4 dotée d'un processeur *Core 2 Duo* de fréquence 1.86 GHZ, d'une RAM de 1GO et d'un disque dur de 160 GO. La machine fonctionne sous l'environnement Windows, qui offre plusieurs avantages notamment la facilité de manipulation des images et des fichiers.

Nous avons opté pour le langage de programmation Borland C++ Builder 6, pour les nombreux outils qu'il offre pour un développement rapide et efficace des applications sous Windows.

4.3.3 Résultats

Pour étudier les performances de l'approche proposée, nous avons utilisé trois types de séquences d'images :

- Deux séquences d'images synthétiques ; la première contient un cercle en mouvement, et la deuxième présente un triangle en mouvement.
- Deux séquences d'images biologiques ; la première contient un microbe en mouvement qui change sa forme, et la deuxième présente une cellule en mouvement.
- Et une séquence d'images échocardiographiques.

Chaque séquence comporte plus de 60 images.

4.3.3.1 Agent Pyramide

Les figures suivantes présentent quelques résultats obtenues en décomposant une image biologique « microbe » par les techniques de représentation pyramidale : pyramide gaussienne et pyramide moyenne.

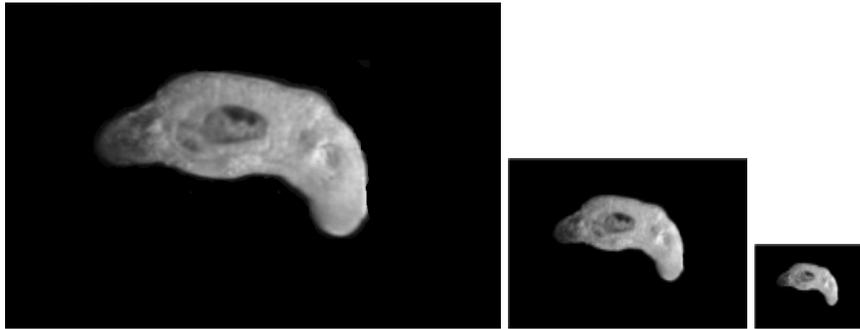


Figure 4.11 : Pyramide Gaussienne avec $a=0.68$, filtre (5×5)

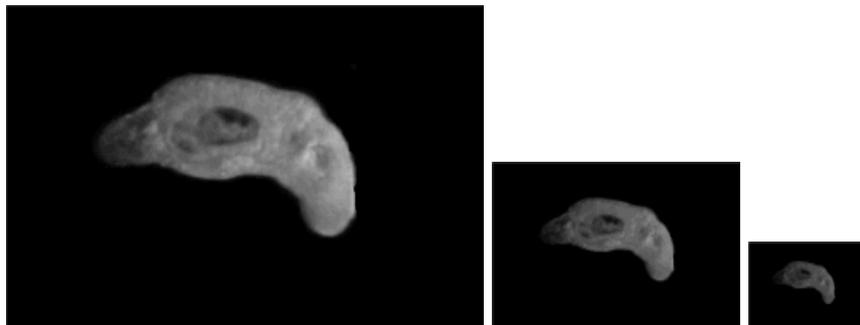


Figure 4.12 : Pyramide Gaussienne avec $a=0.6$, filtre (5×5)

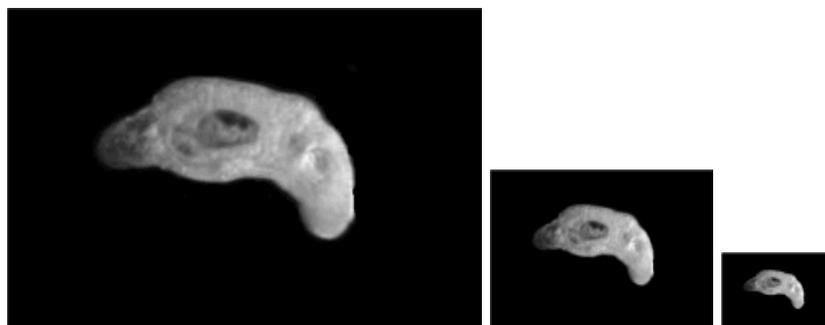


Figure 4.13 : Pyramide Gaussienne avec $a=0.68$, filtre (3×3)



Figure 4.14 : Pyramide Moyenne

Dans nos expérimentations qui suivent, nous avons utilisé des filtres (5×5) et nous avons fixé la valeur du paramètre a à 0.68 pour toute décomposition par pyramide Gaussienne.

4.3.3.2 Agent Snake

Sensibilité des paramètres des énergies

Dans l'étape de détection d'objet par les Snakes, nous nous plaçons par rapport à une démarche classique qui consiste à régler par essai-erreur les pondérations de la combinaison linéaire d'énergies. L'instabilité liée aux valeurs de ces coefficients montre la difficulté qu'il y a à déterminer les valeurs optimales.

Un jeu de paramètres correct a été obtenu après de multiples essais. Nous avons alors fait varier la valeur de chacun des paramètres par rapport à la valeur idéale. Nous montrons, sur quelques exemples, les points que l'on peut considérer comme des points inexacts sur le contour final.

Sur une image biologique « microbe » de résolution réduite (qui se trouve au sommet de la première pyramide construite par l'agent Pyramide), nous allons voir la sensibilité des résultats à ces paramètres.

Paramètre de Continuité

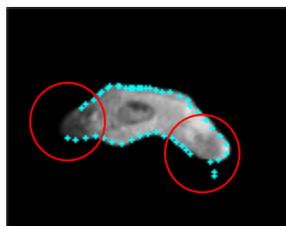


Figure 4.15 : Continuité 0.7

Paramètre de Courbure

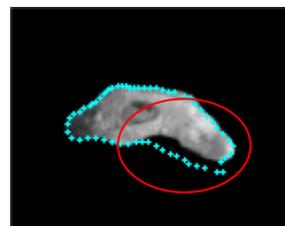


Figure 4.16 : Courbure 0.5

Paramètre de Gradient

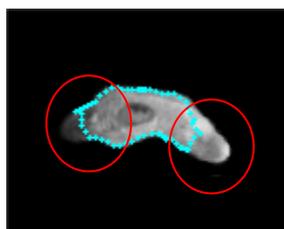


Figure 4.17 : Gradient 0.4

Paramètre de Ballon

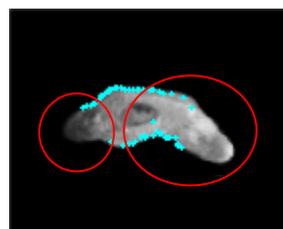
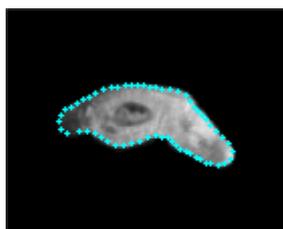


Figure 4.18 : Ballon 0.2

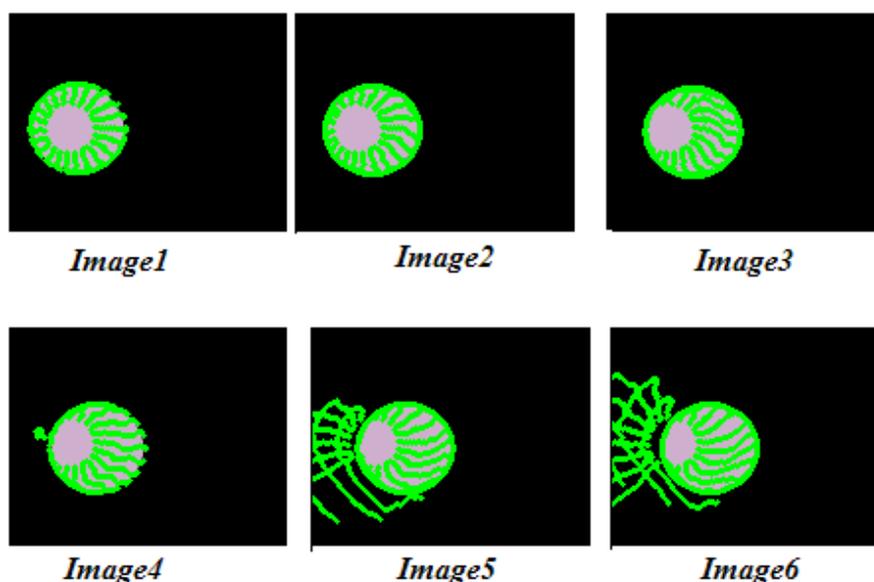
On appliquant les paramètres originaux on obtient :



*Figure 4.19 : Continuité 1, courbure 0.13,
Gradient 0.9, Ballon 0.1*

Influence du centre d'initialisation sur la détection d'objet

Après avoir localisé les frontières de l'objet dans la première image réduite, on veut le détecter sur les autres images de la séquence à faible résolution (les sommets des pyramides). Avec la même initialisation et les mêmes paramètres de départ nous obtenons les résultats suivants sur une image synthétique « cercle » :



*Figure 4.20 : Influence de centre : avec
la même initialisation de départ*

Dans cette figure (4.20), on remarque que l'objet n'est pas bien détecté à partir de l'image 4, cela revient du fait que l'objet est en mouvement. Pour cela on calcul un nouveau centre d'initialisation à chaque fois qu'on détecte un objet. Cette initialisation est effectuée de manière automatique.

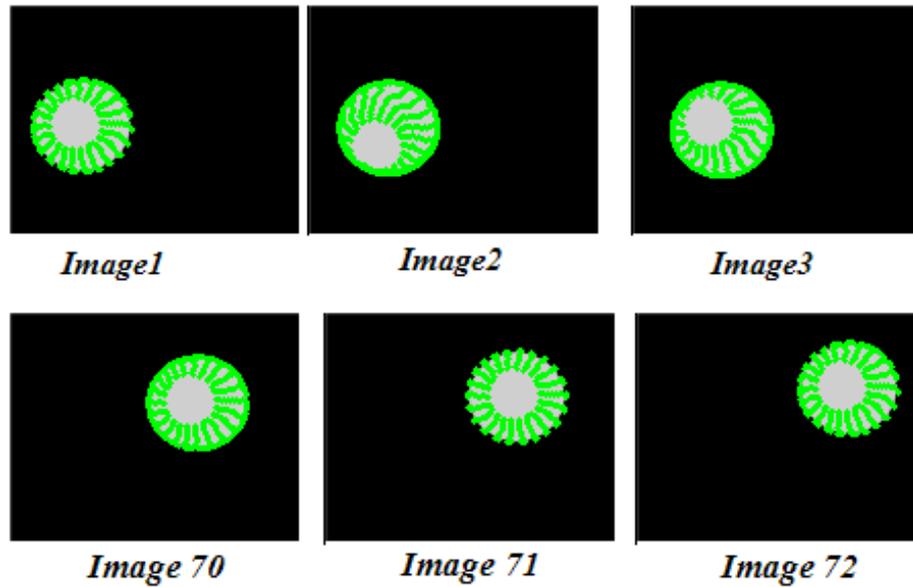


Figure 4.21 : Changement du centre d'initialisation

L'approche proposée comporte donc trois agents qui s'exécutent en parallèle, son but est d'avoir un suivi robuste et rapide d'un seul objet dans une séquence d'images.

En premier lieu, on va présenter les résultats trouvés concernant la détection et le suivi de l'objet sur toute la séquence d'images, ensuite on expose les améliorations trouvées en termes de temps de suivi.

4.3.3.3 Détection et Suivi d'objet

Puisqu'on a un nombre assez grand d'images dans une séquence, nous allons sélectionner et présenter les résultats de seulement quatre images de la séquence.

Les résultats de détection et de suivi d'objet pour toutes les séquences, sont les mêmes en appliquant l'algorithme de Greedy ou de Fast Greedy.

Séquence Synthétique « Cercle »

La figure 4.22 montre des pyramides de trois niveaux (celles des image 1, image 30, image 60 et image 80). La décomposition est faite par pyramide Gaussienne sur des images de résolution 337×225 , le niveau 1 contient des images de résolution 167×113 et la résolution du plus haut niveau (niveau 2) est de 84×57 .

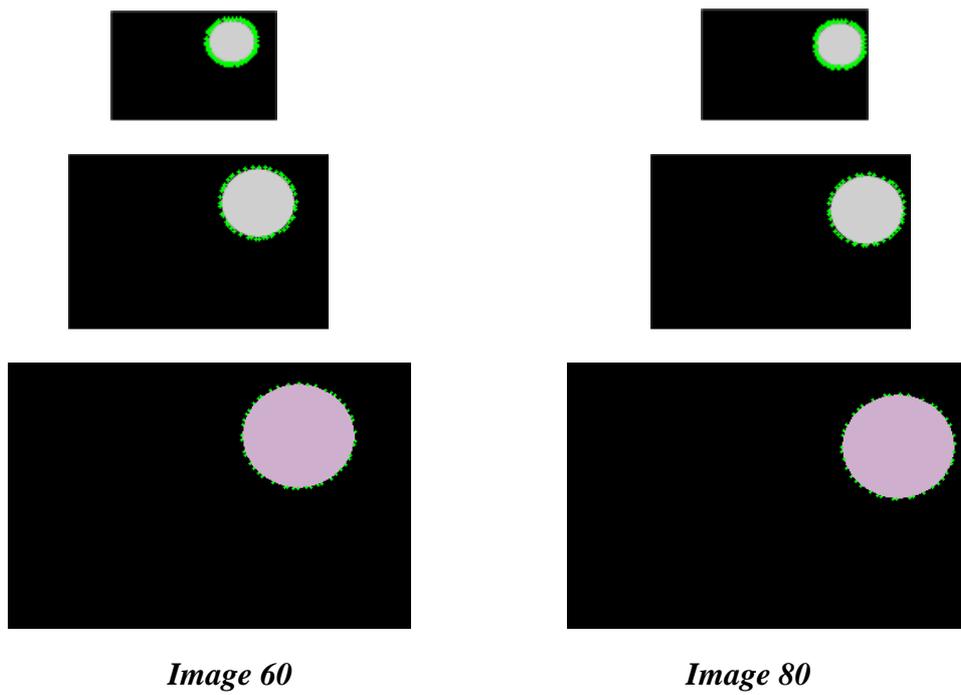
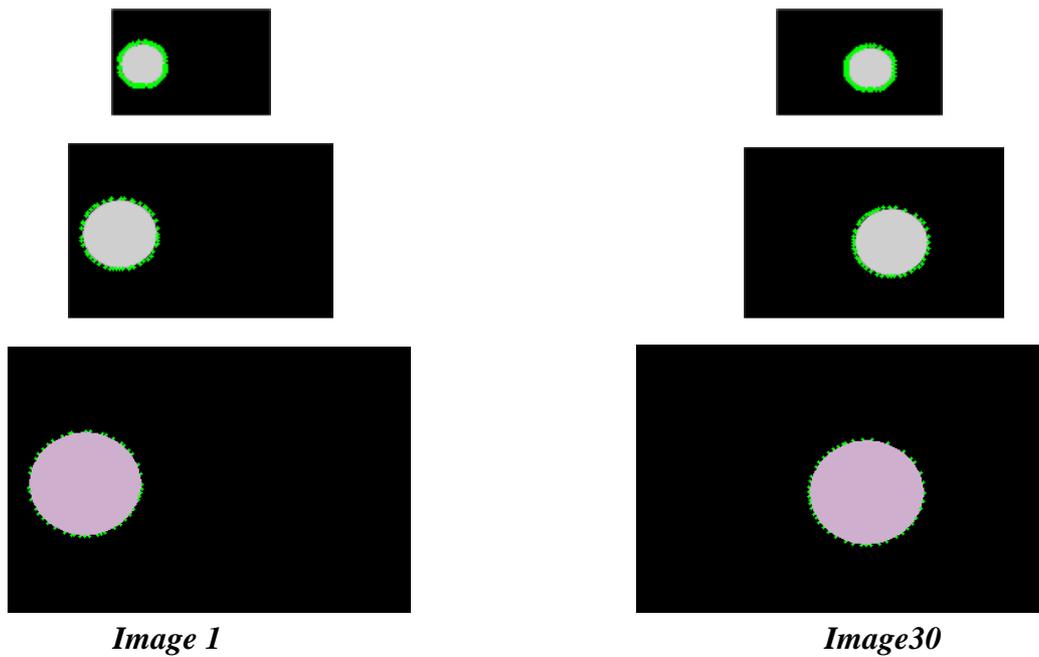


Figure 4.22 : Résultat de la Séquence Cercle (80 images)

Les paramètres utilisés pour la détection de l'objet sont :

Paramètres du Snake :

<i>X centre</i>	<i>Ycentre</i>	<i>Rayon</i>	<i>Voisinage</i>	<i>Nombre de points</i>	<i>Nombre d'itérations</i>
16	28	6	1	60	100

Coefficients des énergies :

<i>Continuité</i>	<i>Courbure</i>	<i>Ballon</i>	<i>Gradient</i>
0.5	0.1	-0.1	1

Séquence Synthétique « Triangle »

La figure 4.23 montre des pyramides de trois niveaux (celles des image1, image 30, image60 et image80). La décomposition est faite par pyramide Moyenne sur des images de résolution 337×225 , le niveau 1 contient des images de résolution 167×113 et la résolution du plus haut niveau (niveau 2) est de 84×57 .

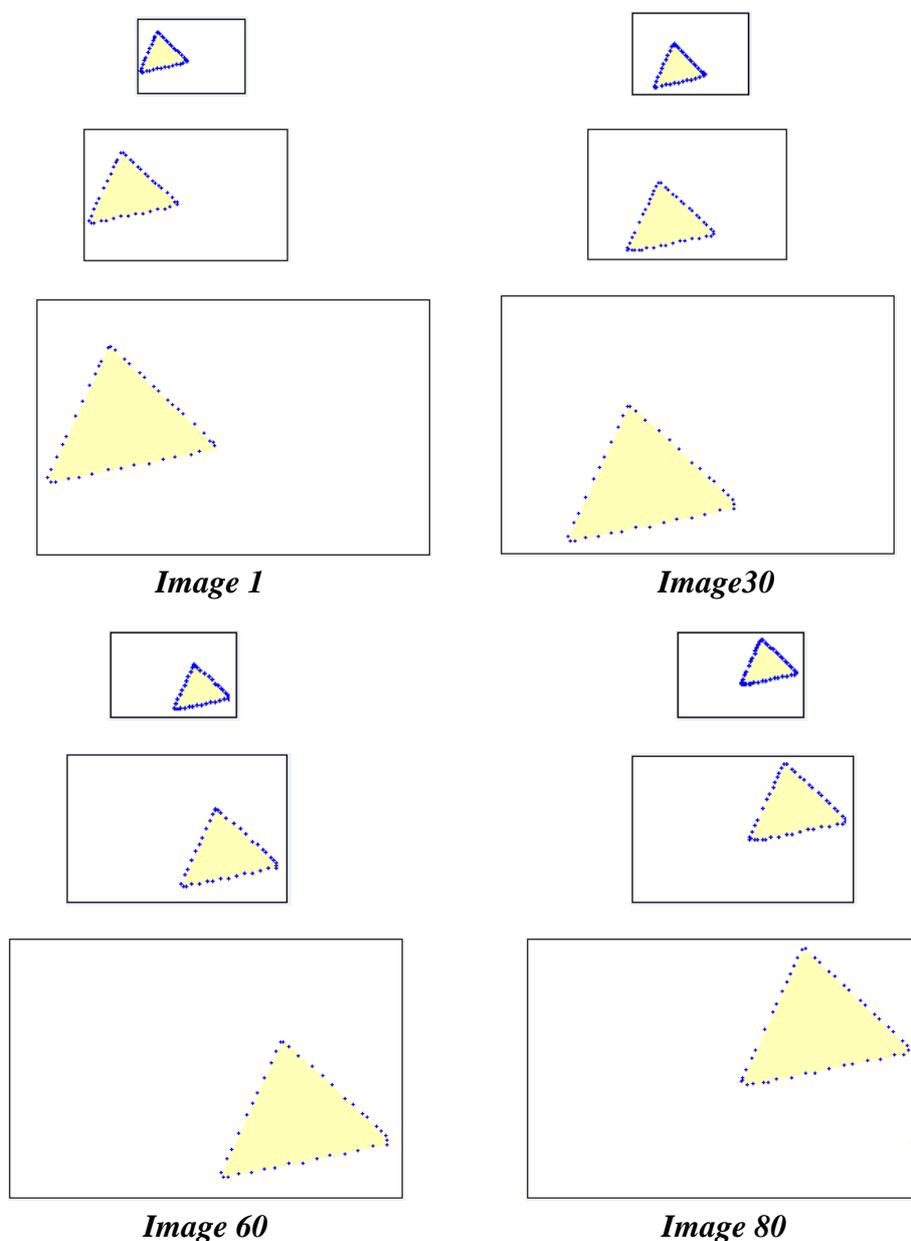


Figure 4.23 : Résultat de la Séquence Triangle (80 images)

Paramètres du Snake :

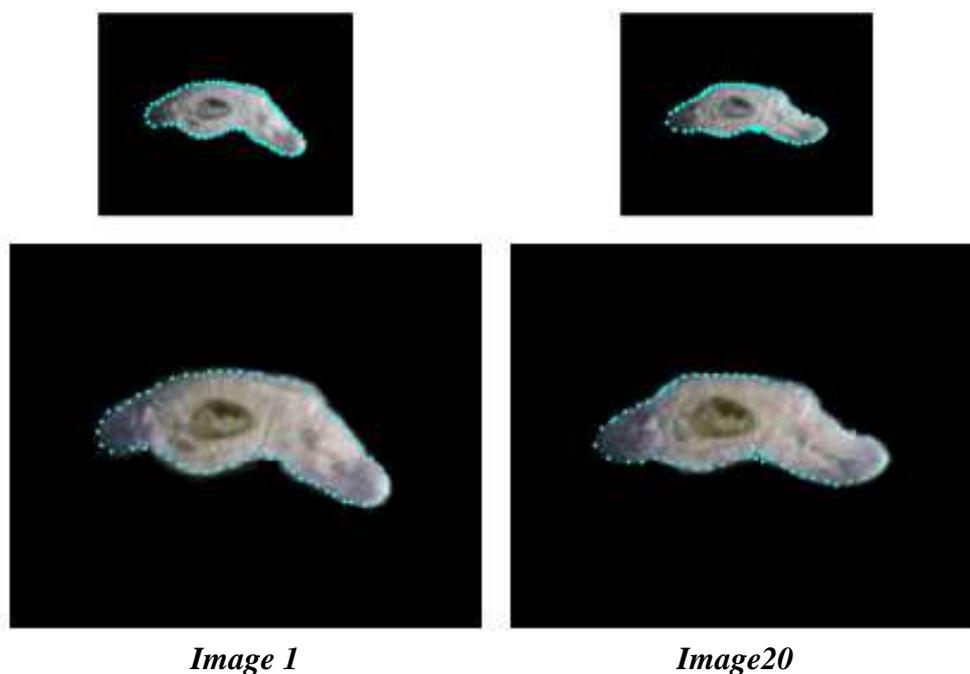
<i>X centre</i>	<i>Ycentre</i>	<i>Rayon</i>	<i>Voisinage</i>	<i>Nombre de points</i>	<i>Nombre d'itérations</i>
15	27	6	1	90	60

Coefficients des énergies :

<i>Continuité</i>	<i>Courbure</i>	<i>Ballon</i>	<i>Gradient</i>
0.8	0.1	-0.1	1

Séquence biologique « Microbe »

La figure 4.24 montre des pyramides de deux niveaux (celles des image1, image 20, image 60 et image 64). La décomposition est faite par pyramide Gaussienne sur des images de résolution 280×230 , le niveau 1 contient des images de résolution 140×115 .



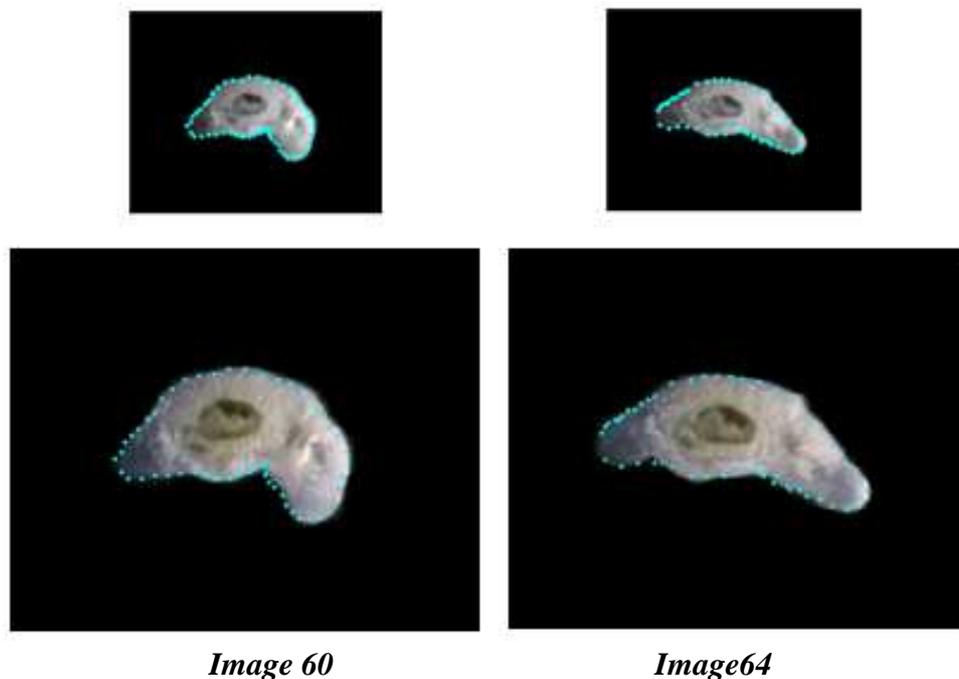


Figure 4.24 : Résultat de la Séquence Biologique « Microbe » (64 images)

Paramètres du Snake :

<i>X centre</i>	<i>Ycentre</i>	<i>Rayon</i>	<i>Voisinage</i>	<i>Nombre de points</i>	<i>Nombre d'itérations</i>
75	54	48	1	60	100

Coefficients des énergies :

<i>Continuité</i>	<i>Courbure</i>	<i>Ballon</i>	<i>Gradient</i>
1	0.13	0.1	0.9

Séquence biologique « Cellule »

La figure 4.25 montre des pyramides de deux niveaux (celles des image1, image 30, image 50 et image72). La décomposition est faite par pyramide Moyenne sur des images de résolution 512×512 , le niveau 1 contient des images de résolution 256×256 et la résolution du plus haut niveau (niveau 2) est de 128×128

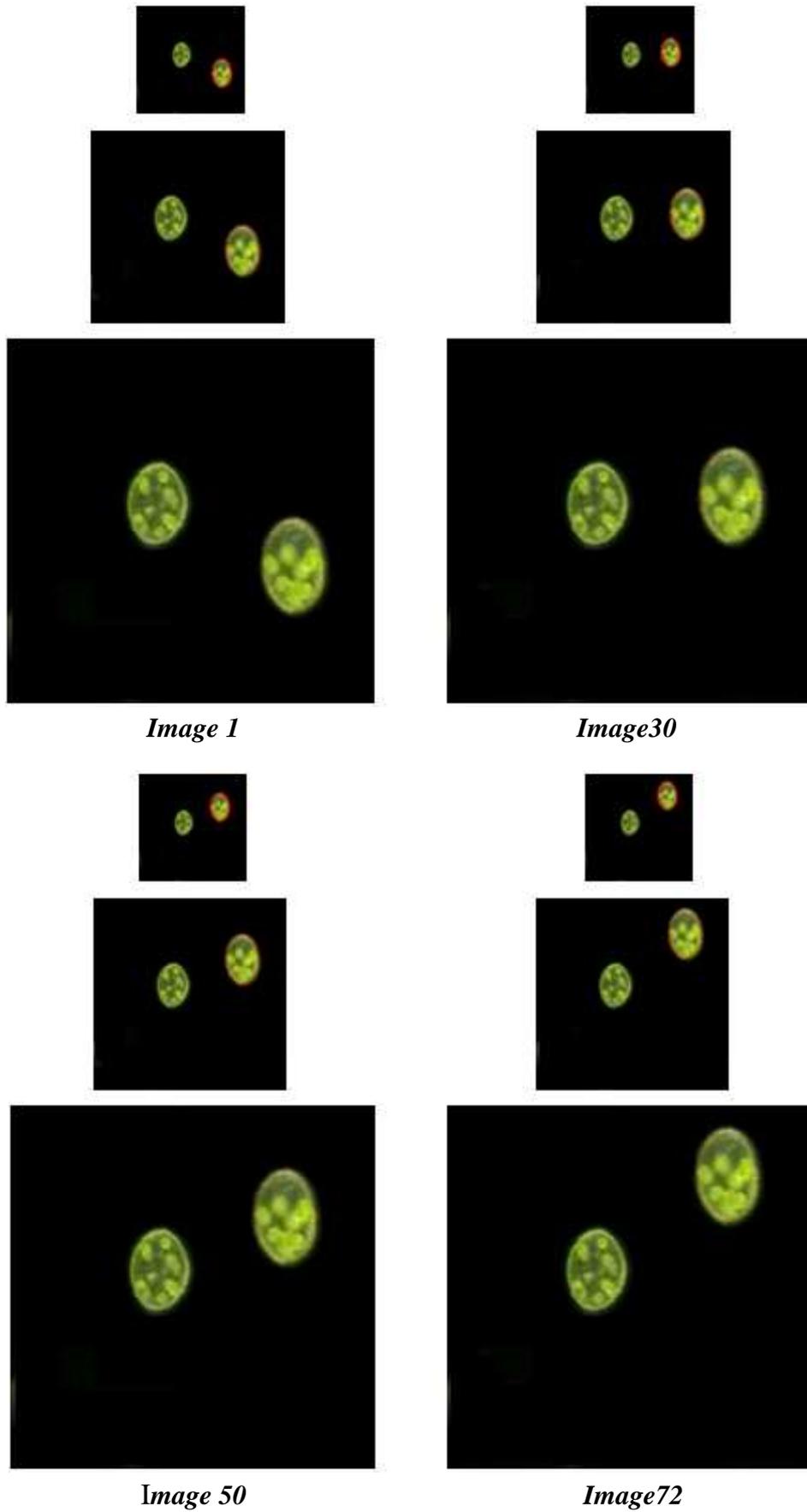


Figure 4.25 : Résultat de la Séquence Biologique « Cellule » (72 images)

Paramètres du Snake :

<i>X centre</i>	<i>Ycentre</i>	<i>Rayon</i>	<i>Voisinage</i>	<i>Nombre de points</i>	<i>Nombre d'itérations</i>
100	81	22	1	50	100

Coefficients des énergies :

<i>Continuité</i>	<i>Courbure</i>	<i>Ballon</i>	<i>Gradient</i>
0.5	0.1	0.1	1

Séquence échocardiographique

La figure 4.26 montre des pyramides de deux niveaux (celles des image1, image 44 et image 64). La décomposition est faite par pyramide Gaussienne sur des images de résolution 320×240 , les images réduites (de niveaux 1) sont de résolution 160×120 .

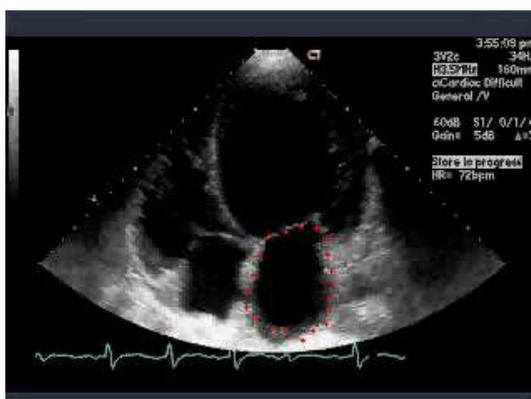
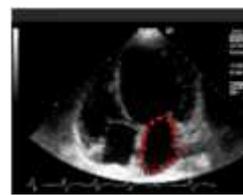
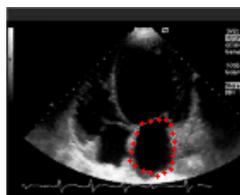


Image 1

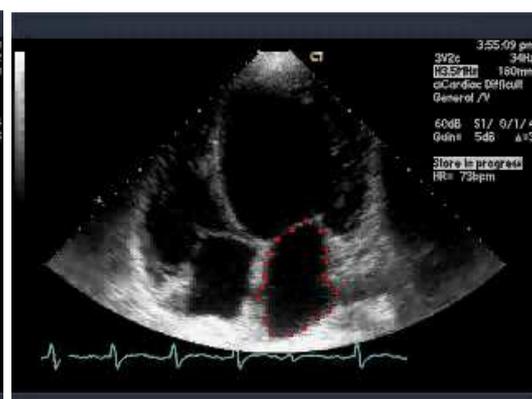


Image44

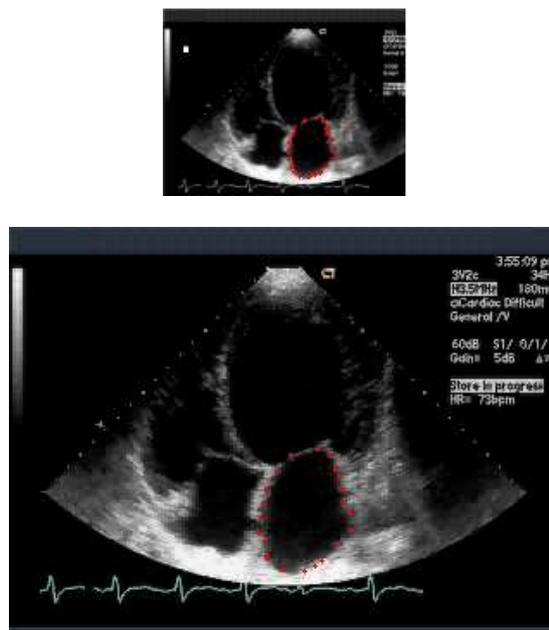


Image64

Figure 4.26 : Résultat de la Séquence échocardiographique (64 images)

Paramètres du Snake :

<i>X centre</i>	<i>Ycentre</i>	<i>Rayon</i>	<i>Voisinage</i>	<i>Nombre de points</i>	<i>Nombre d'itérations</i>
86	84	3	1	20	100

Coefficients des énergies :

<i>Continuité</i>	<i>Courbure</i>	<i>Ballon</i>	<i>Gradient</i>
0.8	0.14	-0.1	1

Nous remarquons que le contour d'objet est bien détecté dans tous les niveaux des séquences synthétiques, cela revient à la simplicité de leurs images qui ne représentent aucun bruit. De même, le contour d'objet est bien détecté dans tous les niveaux des séquences biologiques malgré que leurs images présentent du bruit. Cela revient à l'étape du prétraitement et même à celle de la décomposition en multirésolution qui est basée sur l'application de filtres passe bas.

Dans la séquence échocardiographique l'activité cardiaque est cyclique, pour cela on n'a pris que 64 images. Puisque les images échocardiographiques sont beaucoup plus bruitées par rapport aux biologiques, on a trouvé quelques difficultés à placer

correctement un à deux points du snake sur le contour de l'objet. Malgré ça, on peut dire que la détection et le suivi obtenus sont assez satisfaisants pour ce type d'images.

4.3.3.4 Le Temps de Suivi

Une comparaison temporelle a été faite entre notre approche et une méthode proposée dans [Fek11] qui est basée sur :

- les techniques de multirésolution,
- Les contours actifs paramétriques dont l'évolution se fait par l'itération de l'algorithme « Greedy ».

Cette méthode n'introduit pas le concept d'agents (traitement séquentiel). La comparaison est résumée par le tableau 1 et la figure 4.27 :

	Approche séquentielle	Approche proposée Greedy	Approche proposée Fast Greedy
Séquence synthétique « Cercle »	450.181	314.141	192.853
Séquence synthétique « Triangle »	521.220	356.038	208.129
Séquence « échocardiographique »	124.711	84.032	63.721
Séquence biologique « cellule »	597.039	402.632	344.780
Séquence biologique « microbe »	499.610	392.981	291.224

Tableau 1 : Les améliorations du temps de suivi (en seconde)

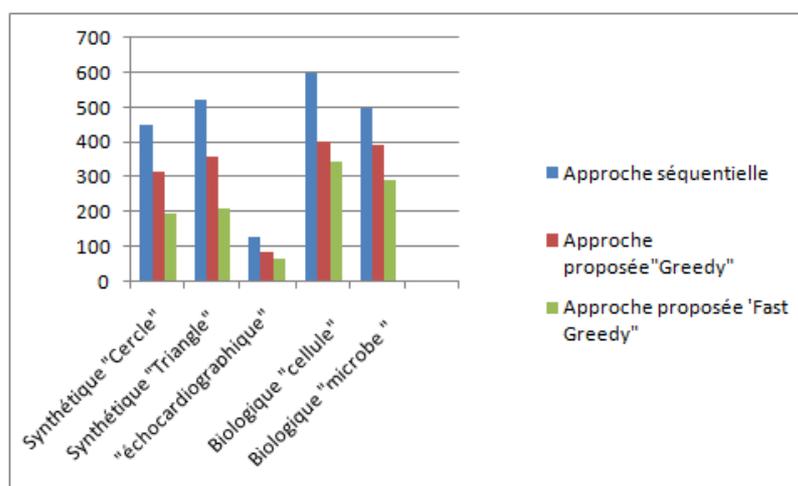


Figure 4.27 : Comparaison entre les trois approches de suivi

À travers ces résultats, nous constatons que la moyenne des améliorations obtenues est de 49%. Cela revient à l'exécution parallèle des agents et même à l'utilisation de l'algorithme «Fast Greedy » qui optimise la fenêtre du voisinage par rapport à l'algorithme de « Greedy ».

L'avantage majeur de « Fast Greedy » est la minimisation du temps de détection d'un objet dans une seule image. Cela est montré par les tableaux et l'histogramme suivants :

Séquence synthétique « Triangle »

Images	Greedy(s)	Fast Greedy(s)	Différence
1	02.80	01.96	00.84
30	02.77	01.97	00.80
60	02.78	01.80	00.98
80	02.85	01.93	00.92

Tableau2 : Comparaison entre Greedy et fast greedy sur quatre images de la séquence Synthétique « Triangle »

Séquence biologique « Microbe »

Images	Greedy(s)	Fast Greedy(s)	Différence
1	03.50	02.45	01.05
20	03.65	02.66	00.99
60	03.49	02.61	00.88
64	03.77	02.71	01.06

Tableau3: Comparaison entre Greedy et fast greedy sur sur quatre images de la séquence biologique « Microbe »

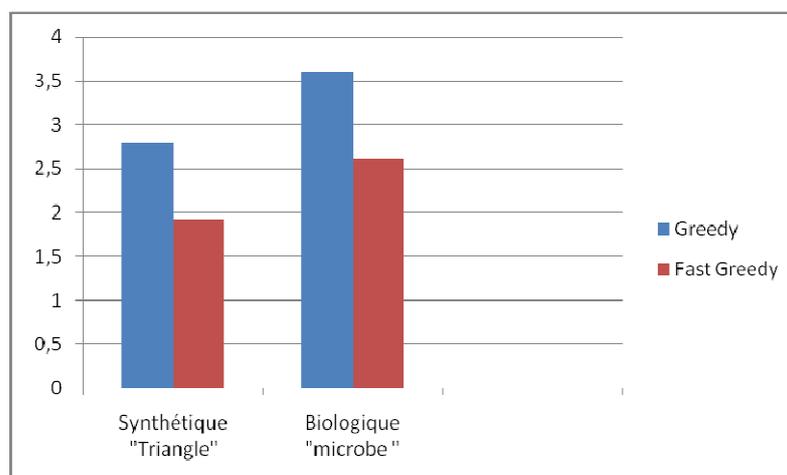


Figure 4.28 : Comparaison entre Greedy et Fast greedy pour chaque séquence

4.4 Conclusion

Dans ce chapitre nous avons présenté notre méthode de suivi d'objet dans une séquence d'images médicales. C'est une approche parallèle qui introduit trois agents : le premier décompose les images de la séquence en utilisant des techniques de représentation multirésolution (pyramide moyenne, pyramide gaussienne), le deuxième détecte l'objet à suivre dans les niveaux supérieurs des pyramides par les contours actifs paramétriques et le troisième agent réalise la projection des points du snake vers les images des niveaux inférieurs.

Les expérimentations ont été menées sur des images synthétiques, biologiques et échocardiographiques. Les résultats obtenus montrent que la coopération entre les trois agents permet un suivi d'objet robuste et rapide et étant donné la nature des traitements effectués, il nous semble que les améliorations du temps acquises sont importantes et pourront, éventuellement, être améliorées.

Conclusion et perspectives

Dans ce Mémoire nous avons présenté une approche parallèle à base d'agents qui permet d'avoir un suivi rapide d'un objet en mouvement dans une séquence d'images.

Dans cette approche nous avons développé une méthode de segmentation par contours actifs paramétriques intégrant une analyse multirésolution pyramidale dans un environnement multi-agents, qui vise à accélérer le temps de suivi de l'objet par un traitement parallèle de trois étapes de l'approche. Ce système comporte trois agents, chacun exécute une des trois étapes en parallèle avec les autres.

Afin de présenter notre travail, premièrement nous avons étudiés les différentes techniques de décomposition d'images en multirésolution. L'intégration de cet outil a permis l'accélération de la convergence du Snake vers les frontières de l'objet et la diminution de la sensibilité des contours actifs aux bruits.

Ensuite nous avons présenté les contours actifs. L'intérêt principal de cette méthode est de détecter un objet dans une image en utilisant les techniques d'évolution de courbes. L'idée est de partir d'une courbe initiale (cercle), et de la déformer jusqu'à l'obtention du contour de l'objet. Son principal avantage est de fournir un contour consistant en une chaîne de points. En outre, un Snake a de nombreux inconvénients : sensibilité à l'initialisation, au bruit et il peut être piégé par des objets non significatifs.

L'approche a été validée sur trois types de séquences d'images : synthétiques, échocardiographiques et biologiques. Les résultats obtenus ont montré l'efficacité de l'approche proposée, où le temps de suivi a été amélioré pour toutes les séquences d'images.

Néanmoins, le travail est loin d'être achevé. Comme perspectives, nous visons à remplacer l'agent Snake par un système d'agents qui permet d'accélérer l'implémentation du contour actif (par une coopération) et d'améliorer ses résultats (par une concurrence). Nous projetons aussi de suivre plusieurs objets dans la même séquence d'images.

Bibliographie

- [All03] B. Allier. *Contribution à la numérisation des collections : Apport des contours actifs*. Thèse de doctorat, Institut National des sciences appliquées de Lyon, 2003.
- [Ami88] A.A. Amini, S. Tehrani, T.E. Weymouth. *Using Dynamic Programming for Minimizing the Energy of Active Contours in the Presence of Hard Constraints*. 2nd Int. Conf. Comput. Vision, décembre 1988, Tampa, FL, p. 95-99.
- [Ari09] W. Aribi, A. Kalfallah, N. Elkadri, L. Farhat, W. Siala, J. Daoud, et M. Bouhlel. (2009). *Évaluation de techniques pyramidales de fusion multimodale (IRM/TEP) d'images cérébrales*. 5th International Conference : Sciences of Electronic, Technologies of Information and Telecommunications, TUNISIA.
- [Ash11] A. Ashraf, B. Safaai , Z. Nazar. *A novel image segmentation enhancement technique based on active contour and topological alignments*. Advanced Computing: An International Journal (ACIJ), Vol.2, No.3, May 2011.
- [Ber95] P. Bertolino. *Contribution des pyramides irregulieres en segmentation d'images multiresolution*. Thèse de Doctorat. Institut national polytechnique de Grenoble. Soutenue le 30 Novembre 1995.
- [Bol04] S. Boltz. *Segmentation spatio-temporelle d'une séquence d'images par compétition de mouvements*. Rapport de Stage - Image Vision. Ecole Doctorale STIC.2004
- [Bou09] R. Boudermine. *Détection des cavités cardiaques en utilisant les contours actifs*. Mémoire d'ingénieur d'état en informatique, Ecole National Supérieure d'Informatique (E.S.I) Oued-Smar, Alger. Soutenu Septembre 2009.
- [Bou11] I. Boussebough. *Les systèmes multi-agents dynamiquement adaptables*. Thèse de Doctorat. Université mentouri constantine. 2011.
- [BuK93] P.J. Burt, R.J. Kolczynski. *Enhanced image capture through fusion*. In Fourth International Conference on Computer Vision, pp. 173-182, 1993.
- [Bur81a] .J. Burt. *Fast filter transforms for image processing*. Computer Graphics and Image Processing, 16:20-51, 1981.
- [Bur81b] P.J. Burt, T.H. Hong, and A. Rosenfeld. *Image segmentation and region property computation by cooperative hierarchical computation*. IEEE Transactions on Systems, Man, and Cybernetics, SMC-11:802-809, 1981.
- [Bur83] J. Burt and E.H. Adelson. *The laplacian pyramid as a compact image code*. IEEE Transactions on Communications, COM-31:532-540, 1983.

- [**Cal97**] Calle, D. et A. Montanvert. *Pyramide résiduelle pour la compression d'images*. Seizième Colloque Gretsi, *GRENOBLE*. 1997
- [**Cas93**] V. Caselles, F. Catta, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1-31, 1993
- [**Cas95**] V.Caselles, Kimmel Ron, Sapiro Guillermo. *Geodesic active Contour*. International Conference on computer vision '95. June 1995 pp.694-699.
- [**Cas97**] V.Caselles, Kimmel Ron, Sapiro Guillermo. *Geodesic active Contour*. International Journal of Computer Vision 22(1), 61-79 1997
- [**Coh91**] L. Cohen. *On active contour models and balloons*. Computer vision, graphic, and image processing : Image Understanding, Vol. 53, N° 2, pp 211-218, Mars 1991.
- [**Cou05**] A.Coutant. *La méthode des contours actifs en traitement des images*. Mémoire pour l'examen probatoire en calcul scientifique. Conservatoire national des arts et métiers, Paris. Soutenue en février 2005.
- [**Dan05**] P. Y. Daniau-clavreul, V. Roullier, C. Cavaro-menard . *Segmentation automatique du foie sur des IRM abdominales*. Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC. MajeSTIC 2005.
- [**Dav95**] C. Davatzikos and J. L. Prince. *An active contour model for mapping the cortex*. IEEE Trans. On Medical Imaging, 14(1):65–80, March 1995.
- [**Fek09a**] A. Fekir, N. Benamrane, A. Taleb-Ahmed. *Détection et suivi d'objets dans une séquence d'images par contours actifs*. Proceedings of the 2nd Conférence Internationale sur l'Informatique et ses Applications (CIIA'09), Saida, Algeria, 2009.
- [**Fek09b**] A. Fekir, N. Benamrane, A. Taleb-Ahmed. *Segmentation d'une séquence d'images médicales par les contours actifs*. International Conference on Systems and Information Processing (ICSIP'09) Guelma , 2009.
- [**Fek11**] A. Fekir, N. Benamrane. *Fast object tracking in medical image sequence using active contours*. ICCA 2011, Riyadh, Saudi Arabia.
- [**Fer95**] J. Ferber, Les Systèmes Multi-Agents. Vers une intelligence collective. Interéditions, 1995
- [**Fleu09**] J. Fleureau, M. Garreau, D. Boulmier, C. Leclercq et A. Hernandez. *Segmentation 3D multi-objets d'images scanner cardiaques : une approche multi-agents*. vol.30 no.3. *IRBM*, 2009.
- [**Gas05**] M. Gastaud. *Modèles de contours actifs pour la segmentation d'images et de vidéos*. Thèse de doctorat. Université de Nice-Sophia Antipolis, École doctorale de STIC. 2005
- [**Gra06**] C. Grasset-Simon. *Définition et étude des pyramides généralisées n D : application pour la segmentation multi-échelle d'images 3D*. Thèse de Doctorat. Université de Poitiers. 2006.

- [Gue11]** M. Guermoudi et A. Benamar. *Conception et Implémentation d'un Système multi-agent Pour le test de primalité de nombre premier*. Mémoire de licence. Université Abou Bakr Belkaid– Tlemcen.2011.
- [Har89]** R. Haralick, Zhuang Xinhua, Lin Charlotte et Lee James. *The digital morphological sampling theorem*. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37, n° 12, d'écembre 1989, pp. 2067–2090.
- [Har05]** R. Haroun, F. Boumghar, S. Hassas, and L. Hamami. *A Massive Multi-agent System for Brain MRI Segmentation*, pages 174–186. Springer-Verlag, 2005.
- [His03]** A. Histace, V. Courboulay, C. Cavaro-Ménard et M. Ménard. *Détection et suivi des déformations intra-myocardiques en IRM cardiaque marquée*. In actes du congrès GRETSI03, volume3, pages 316-320, 2003
- [Hon82]** T.H. Hong, K.A. Narayanan, S. Peleg, A. Rosenfeld et T. Silberberg. *Image smoothing and segmentation by multiresolution pixel linking : further experiments and extensions*. IEEE Trans. Systems, Man Cybernetics, 12(5):611–623, 1982.
- [Hor74]** J.L. Horowitz et Pavlidis. *Picture segmentation by a direct split-and-merge procedure*. In Proc. of the 2nd ICPR, pages 424–433, 1974.
- [Idi05]** K. Idir, H. Merouani et Y. Tlili. *Proposition d'une pyramide duale d'agents pour la segmentation d'image*. SETIT, 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunication. TUNISIA. 2005.
- [Jac01]** Y. Jacquélet, Montois JJ, Duchesnay E, Garreau M, Kinié A. *Architecture pyramidale agent pour la segmentation d'image : application à l'extraction d'une zone lobulaire issue d'une mammographie*. 18ème Colloque GRETSI; 2001; Toulouse, France; 2001. p. 469-472.
- [Jol94]** M. Jolion. *Analyse multirésolution du contraste dans les images numériques*. Traitement du Signal, 11(3) :245–255, 1994.
- [Kab99]** N. Kabachi. *Modélisation et apprentissage de la prise de décision dans les organisations productives : approche multi-agents*. Thèse de doctorat. Informatique. Université Jean Monnet et Ecole nationale Supérieure des Mines de Saint Etienne, janvier 1999.
- [Kas87]** M. Kass, A. Witkin, D. Terzopoulos. *Snakes : Active contour models*. In proceedings of the first International Conference on computer vision, Juin 1987, p.259-268.
- [Kel71]** M.D. Kelly. *Edge detection in pictures by computer using planning*. In B. Meltzer and D. Michie, editors, Machine Intelligence, volume 6, pages 397-409, Edinburgh, Scotland, 16-20 septembre 1971. Edinburgh Univ. Press.
- [Kha05]** N. Khalifa, A. Malek, K. Hamrouni. *Segmentation d'images par contours actifs : Application à la détection du ventricule gauche dans les images de scintigraphie cardiaque*. 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, Tunisia, 2005.

- [Lag10]** H. Laguel. *Déploiement sur une plateforme de visualisation, d'un algorithme coopératif pour la segmentation d'images IRM basé sur les systèmes multi-agents*. Mémoire d'ingénieur. Université des Sciences et de la Technologie Houari Boumediene. 2010.
- [Lam94]** K. Lam et H. Yan. *Fast greedy algorithm for active contours*. Electronic Letters, 30(1) :21–23, 1994.
- [Lei93]** F. Leitner, P. Cinquin. *From snake and splines to snake-spline, Workshop on Geometric Reasoning for Perception and Action*, Grenoble, France, 1993, vol. 708 of LNCS, pp. 264–281.
- [Lep05]** R. Lepage. *La vision par ordinateur*. Presses de l'École de technologie supérieure. 2005
- [Li11]** T. Li, A. Krupa, C. Collewet. *Un contour actif robuste basé sur les descripteurs de Fourier*. In ORASIS'11, journées francophones des jeunes chercheurs en vision par ordinateur, France, Juin 2011.
- [Mat07]** Noriko Matsumoto, Norihiko Yoshida, Shuji Narazaki. *Layered Cooperation of Macro Agents and Micro Agents in Cooperative Active Contour Model*. PRIMA 2007: 86-97
- [Med11]** H. Meddeber, A. Fekir, N. Benamrane. *Suivi d'objet dans une séquence d'images médicales par les contours actifs*. EGCM 2011, Maroc.
- [Med12]** H. Meddeber, A. Fekir, N. Benamrane. *Suivi parallèle d'objet dans une séquence d'images médicales par les contours actifs*. Conférence Internationale sur le traitement de l'information multimédia, CITIM'2012 Mascara, Algérie.
- [Men90]** S. Menet, P. Saint-Marc, G. Medioni. *B-snakes : Implementation and application to stereo*. Image Understanding Workshop, September 1990, pp. 720–726.
- [Mil07]** J. Mille. *Modèles déformables pour la segmentation et le suivi en imagerie 2D et 3D*. Thèse de Doctorat. Université François Rabelais de Tours. 2007.
- [Mou05]** D. Mounir, J. Puentes, P. Clémence, L. Bressolette, B. Guias, B. Solaiman. *Détection des contours des thrombus veineux dans les images échographiques*. 3^{ème} Conférence internationale: Sciences Electroniques, Technologies de l'Information et des Télécommunications. SETIT 2005.
- [Mou11]** R. Moussa. *Segmentation multi-agents en imagerie biologique et médicale : application aux IRM 3d*. Thèse de Doctorat . Université BORDEAUX I. 2011
- [Neb06]** S. Nebti. *Optimisation par écosystèmes artificiels : application à la segmentation d'images*. Mémoire de Magister, Université Mentouri de Constantine. 2006
- [Noo98]** H. Noomen. *Outil de communication pour les systèmes multiagents*. Mémoire de maîtrise des sciences appliquées. Université de Sherbrooke Canada. 1998.

- [Osh88]** S. Osher, J.A. Sethian. *Fronts propagation with curvature dependent speed : Algorithms based on Hamilton-Jacobi formulations*. Journal of Computational Physics, 79,1988.
- [Pop08]** S. POP. *Modèles de fusion et diffusion par équations aux dérivées partielles : application à la sismique azimutale*. Thèse de Doctorat, Spécialité : automatique, productique, signal et images . Université Bordeaux I. Soutenue le 11 novembre 2008.
- [Pre04]** F. Precioso. *Contours actifs paramétriques pour la segmentation d'images et vidéos*. Thèse de Doctorat. Université de Nice-Sophia Antipolis, École doctorale de STIC. Soutenue en septembre 2004
- [Ray02]** Ray, S. T. Acton et K. Ley. *Tracking leukocytes in vivo with shape and size constrained active contours*. IEEE Transactions on Medical Imaging, vol. 21, no. 10, 2002.
- [Rid06]** T. Ridene. *Mécanismes d'Attention Visuelle sur Rétine Artificielle*. Mémoire de stage de 2ème année Master IPCC spécialité Image. Université Paris V, René Descartes.2006
- [Ric03]** N. Richard, M. Dojat, and C. Garbay. *Multi-agent approach for image processing for mri human brain scans interpretation*. In 9th Conference on Artificial Intelligence in Medicine Europe, 2003.
- [Rou03]** J. Rousselle, *Les contours actifs, une méthode de segmentation. Application à l'imagerie médicale*. Thèse de doctorat, Université de François Rabelais de Tours, Soutenue le 9 juillet 2003.
- [Rou04]** J. Rousselle, F. Nebel. *Amélioration des paramètres des contours actifs par une technique issue de la théorie de l'apprentissage*. Rapport de stage de l'Université François Rabelais, 2004.
- [Rou09]** L. Rouane. *L'apport des systèmes multi-agents à la recherche d'information dans les bases de données réparties*. Mémoire de Magister, Institut National de formation en Informatique (I.N.I) Oued-Smar Alger. 2009
- [Ros84]** A. Rosenfeld : *Multiresolution Image Processing and Analysis*. Springer, Berlin, 1984.
- [Set 99]** J. A. Sethian. Fast marching methods. SIAM Review, 41(2) :199{235, June 1999.
- [Sly06]** M. Sakalli, K-M. Lam et H. Yan. *A faster converging snake algorithm to locate object boundaries*. IEEE Transactions on Image Processing, 15(5) :1182–1191, 2006.
- [Will92]** D. Williams et M. Sham. *A fast algorithm for active contour and curvature estimation*. Computer vision graphic image process : Image understanding, vol. 55, n°1, pp 14-26, 1992.
- [Wit84]** A.Witkin. *Scale space filtering : a new approach to multi-scale description*. Image Understanding. 1984

[Woo02] M. Wooldridge. *An Introduction to Multiagent Systems*. 340 p, John Wiley & Sons Publishers, Chichester, England, 2002.

[Xu97] C. Xu and J.L. Prince. *Snakes, Shapes, and Gradient Vector Flow*. IEEE Trans. on Image Processing, 1997.

[Zar07] K. Zarour. *Vers un système d'information coopératif pour la prise en charge des soins à domicile : une architecture basée agent*. Mémoire de Magister. Université Mentouri de Constantine. 2007

[Zim02] C. Zimmer, E. Labruyère, V. Meas-Yedid, N. Guillén and J.C Olivo-Marin. *Improving Active Contours for Segmentation and Tracking of Motile Cells in Videomicroscopy*. 16th International Conference on Pattern Recognition, August 2002 , pp. 286 -289.

[Zla06] N. Zlatoff. *Indexation d'images 2d vers une reconnaissance d'objets multicritères*. Thèse de Doctorat, Institut national des sciences appliquées de Lyon. Soutenue le 12/07/2006.

Résumé

La segmentation d'images par contours actifs est une méthode permettant de localiser les frontières d'un objet, en utilisant une courbe qui se déforme en fonction de différentes énergies. Dans ce Mémoire nous proposons une approche parallèle permettant un suivi rapide d'un objet dans une séquence d'images médicales. L'approche proposée définit trois agents : Le premier utilise une analyse multirésolution pyramidale afin de décomposer les images de la séquence en des résolutions multiples. Le deuxième agent détecte l'objet à suivre dans les images à basse résolution par les contours actifs paramétriques. Le troisième agent réalise la projection des contours détectés dans les images à basse résolution vers les images à haute résolution.

Les expérimentations ont été menées sur des images synthétiques, biologiques et échocardiographiques, les résultats obtenus sont encourageants.

Mots Clés : suivi d'objet; séquence d'images; imagerie médicale; agents; multirésolution; contours actifs.

Abstract

Image segmentation by active contours is a method to locate the boundaries of object, using a curve that deforms by minimization of energies. In this memory, we propose a parallel approach for fast object tracking in medical images sequence. The proposed approach defines three Agents: The first uses a pyramidal multiresolution analysis to decompose the images of the sequence in multiple resolutions. The second agent detects the object in low-resolution images by parametric active contours. The third agent realizes the projection of the contours detected in low-resolution images to high-resolution images. Our approach was tested using three sequence types: synthetic, biological and echocardiographic, the obtained results are encouraging.

Key words: Tracking objects, image sequences, medical image, Agent, multirésolution, Active contour.