

N° d'ordre / FS / UMBB / 2013

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté des Sciences

Résumé du Mémoire de Magister

Présenté par :

BENNAI Mohamed Tahar

En vue de l'obtention du diplôme de Magister en :

Filière : Systèmes Informatiques et Génie Logiciel

Option : Spécification Logiciels et Traitement de l'Information

Réalisation d'un système multi-agents adaptatifs pour l'imagerie médicale

Devant le jury

Mr BOUFAIDA	Mahmoud	Prof.	UMC	Président
Mme GUESSOUM	Zahia	MC/HdR	UPMC	Promotrice
Mr MEZGHICHE	Mohamed	Prof.	UMBB	Examineur
Mr DIAF	Moussa	Prof.	UMMTO	Examineur
Mr MAZOUZI	Smaine	MCA	Université de Skikda	Examineur

Année Universitaire 2012/2013

Remerciements

Je tiens tout d'abord à remercier dieu tout puissant pour m'avoir donné la force et la santé me permettant de réaliser ce modeste travail.

Je tiens aussi à exprimer ma profonde gratitude envers ma directrice de thèse, madame Zahia Guessoum non seulement pour m'avoir proposé ce sujet et m'avoir encadré lors de la réalisation de ce travail, mais surtout pour le soutien et la confiance qu'elle a eue à mon égard.

Je tiens également à remercier les membres du jury, M. Boufaïda Mahmoud, M. Mezghiche Mohamed, M. Diaf Moussa et M. Mazouzi Smaïne pour avoir accepté d'évaluer ce modeste travail et d'assister à sa présentation.

Mes remerciements s'adressent tous particulièrement à monsieur Mazouzi Smaïne pour son aide et pour les conseils qu'il m'a prodigués lors de l'élaboration de cette thèse.

Je remercie aussi ma mère, ma soeur Amel, son mari Yazid, mon frère Liès et sa femme Ferièle, mon frère Reda ainsi que sa femme Radia, ma soeur Naouel et son mari Abdeslam, mes nièces Angelina, Sarah, Iness, Chahinez et mon neveu pour tout le soutien qu'ils m'ont apporté dans les moments difficiles. Je remercie aussi mes oncles et mes tantes, tout particulièrement ma tante Rachida, mon oncle Slimane et monsieur Khaddouci M'hamed qui n'ont cessé de venir me voir et ce malgré toutes leurs responsabilités.

Je tiens à adresser de vifs remerciements à Dr Bendaoued, Dr Louhadj, Dr Bahloul, Dr Chikhi, Dr Benamira, Dr Asla et à toute l'équipe médicale de l'hôpital de Bologhine.

Je remercie aussi tous mes amis qui ont été jours après jours, parmi eux Mokrane, Djallel, Fateh, Khalife, Hocine, Yacine, Kamel, Ibrahime, Nouradine et à toute personne qui de près ou de loin, m'a soutenu lors de la réalisation de cette thèse.

Enfin, je tiens également à remercier tous mes professeurs du département informatique de l'université Amhamed Bougara de Boumerdes pour m'avoir formé tout au long de mon cursus universitaire.

Abstract

This thesis combines image processing and multi-agent systems to propose a segmentation system image that provides an efficient segmentation of the various types of frames used in the field of medical imaging. Multi-agent systems allow to cope with the inconvenience of image segmentation that are the prohibitive time of segmentation and the big size of data of medical images.

The approach that we propose in this thesis is based on a system which contains a set of situated reactive agents. The latter use the image as an environment and the method of region growing to identify the various homogeneous areas of the image. The system uses the properties of cooperation and competition among agents to achieve a collective segmentation of the image into a set of regions and contours. The implementation of our approach is performed in Microsoft Windows environment using the technology Dot Net. Microsoft Dot Net is a platform of tools for development, deployment and execution of application of various types. Tests made during our experiments produce satisfying results for different types of images, including images of CT and MRI images.

Key words

multi-agent systems, medical imaging, image segmentation, distributed system

Résumé

Dans cette thèse, nous nous intéressons à deux domaines de l'informatique qui sont le traitement d'image et les systèmes multi-agent, en vue de la réalisation d'un système de segmentation d'image.

Notre travail aura pour principal objectif d'aboutir à un système de segmentation d'image, offrant une segmentation efficace des différents types d'images utilisées dans le domaine de l'imagerie médicale. C'est pour résoudre les inconvénients que sont le temps de segmentation et les grands volumes de données des images médicales, que nous nous sommes tournés vers les systèmes multi-agents.

L'approche que nous proposons dans notre thèse se base sur un système constitué d'un ensemble d'agents réactifs situés, ayant pour environnement l'image à segmenter et utilisant l'accroissement de régions pour déterminer les différentes zones homogènes composant l'image. Le système exploitera les propriétés de coopération et de concurrence entre agents pour aboutir à une segmentation collective de l'image en un ensemble de régions et de contours. L'implémentation de notre approche fut réalisée sous environnement Microsoft Windows à l'aide de la technologie .Net. Microsoft .Net est une plateforme d'outils pour le développement, le déploiement et l'exécution d'application de divers types. Les tests réalisés lors de nos expérimentations montrent des résultats satisfaisants pour différents types d'images, parmi lesquelles des images de scanner et des images IRM.

Mots clés

Systemes multi-agents, images médicales, segmentation image, systèmes distribués.

ملخص

لقد اهتمنا في هذه الاطروحة بتخصصين من تخصصات الاعلام ألاي اللذان هما التخصص في تحليل الصور و التخصص في الانظمة متعددة الأعوان ،هذا بهدف تصميم نظام لتجزئة الصور.

هدفنا الرئيسي في عملنا هذا ،هو التوصل الى نظام لتجزئة الصور قادر على توفير تجزئة فعالة لمختلف أنواع الصور المستخدمة في المجال الطبي ؛و لقد توجهنا نحو اختصاص الانظمة متعددة الأعوان لحل العوائق المتمثلة في الوقت الذي تستغرقه تجزئة الصور بالإضافة الى الحجم الكبير الذي هو من خصائص الصور الطبية.

النموذج الذي نقترحه في هذه الاطروحة يعتمد اساساً على نظام يتكون من مجموعة من الأعوان التفاعلية والتي تتخذ الصورة الخاضعة للتجزئة كوسط لها. تستعمل هذه الاعوان تقنية نمو المناطق لتحديد مختلف المساحات المتجانسة التي تتكون منها الصورة. سيقوم النظام باستخدام خصائص التعاون والتنافس ما بين الاعوان للوصول الى تجزئة جماعية للصورة مؤلفة من مناطق و ملامح. لقد تمّ تطبيق منهجنا تحت نظام التشغيل مايكروسوفت ويندوز و باستخدام تكنولوجيا دوت نت و التي هي منصة من أدوات لتطوير ،نشر و تطبيق برامج متنوعة. الاختبارات التي أنجزت أثناء تجاربنا تبين نتائج مرضية لتجزئة أنواع مختلفة من الصور، بما في ذلك صور الماسح الضوئي و صور ناتجة عن الرنين المغناطيسي.

الكلمات الرئيسية

الانظمة متعددة الأعوان ، تجزئة الصور ، الصور الطبية ، الأنظمة الموزعة

Table des matières

1	Introduction générale	7
2	Les IRM et les différentes approches de segmentation d'images	11
2.1	Introduction	11
2.2	L'imagerie par Résonance Magnétique (IRM)	12
2.2.1	Définition de l'IRM	12
2.2.2	Principes de l'IRM	12
2.2.3	Codage des images IRM	16
2.2.4	Images volumiques	17
2.3	Les différentes approches de segmentation d'images	17
2.3.1	Qu'est ce que la segmentation d'images	17
2.3.2	L'approche contour	18
2.3.3	L'approche région	21
2.3.4	L'approche Structurelles	27
2.3.5	L'approche Forme	31
2.3.6	L'approche théorie des graphes	41
2.4	Conclusion	44
3	Les systèmes multi-agents et la segmentation d'images	46
3.1	Introduction	46

TABLE DES MATIÈRES

3.2	Les Systèmes Multi-Agents	47
3.2.1	Définition des systèmes multi-agents	47
3.2.2	Architectures des systèmes multi-agents	48
3.3	Les Systèmes Multi-Agents pour la segmentation d'images médicales . . .	50
3.3.1	Une approche biomimétique pour la segmentation d'images	52
3.3.2	Une architecture pyramidale agent pour la segmentation d'images : application à l'extraction d'une zone lobulaire issue d'une mam- mographie	56
3.3.3	Segmentation 3d multi-objets d'images de scanner cardiaques : une approche multi-agents	59
3.3.4	Application de l'apprentissage par renforcement à la segmentation d'images	64
3.3.5	Une approche distribuée basée sur la logique flou et les algorithmes génétiques pour la segmentation d'images	67
3.3.6	Un nouveau mécanisme d'essaim basé sur les colonies d'araignées sociales pour la segmentation d'image	71
3.4	Conclusion	75
4	Un SMA pour la segmentation d'images médicales	77
4.1	Introduction	77
4.2	Spécification de l'approche	78
4.3	Comportement des agents	80
4.3.1	Accroissement d'une région	82
4.3.2	Fusion de région	84
4.3.3	Concurrence entre régions	86
4.3.4	Épilogue de l'accroissement	87
4.4	Conclusion	88

TABLE DES MATIÈRES

5	Implémentation et résultats expérimentaux	89
5.1	Introduction	89
5.2	Le Framework .Net	90
5.3	Implémentation de l'approche	91
5.3.1	Scheduler	92
5.3.2	Pixel	93
5.3.3	ImageSlice	93
5.3.4	Region	94
5.3.5	Agent	94
5.4	Résultats expérimentaux	97
5.5	Conclusion	101
6	Conclusion générale	103
A	Format Analyse 7.5	113
A.1	Description du format Analyse 7.5	113
A.1.1	Description du fichier d'entête	113
A.1.2	Description du fichier .img	115
A.2	Ordre des voxels et terminologie d'orientation utilisée	119

Table des figures

2.1	Mouvement de l'hydrogène soumis au champ B_0 [1]	13
2.2	Les composantes de l'aimantation lors de l'excitation [1]	14
2.3	Les axes se coupent (de droite à gauche axiale, sagittale et coronaire) [2]	15
2.4	Les pondération des images IRM (de gauche à droite pondération T1, T2 et ρ) [1]	16
2.5	Classification des différentes approches de segmentation d'images [3]	19
2.6	Structure d'un neurone artificiel [3]	21
2.7	Hyperplan optimal d'une machine à vecteur de support	27
2.8	Érosion de plusieurs objets d'une scène par un élément structurant	28
2.9	Dilatation de plusieurs objets d'une scène par un élément structurant	29
2.10	Principe de la Ligne de Partage des Eaux	30
2.11	Processus de segmentation par l'approche d'atlas utilisé dans [4]	32
2.12	Étiquetage manuel des formes par N points	34
2.13	Étiquetage manuel des formes Sur une IRM cérébrale [5]	35
2.14	Isolation d'une tumeur dans une IRM cérébrale à l'aide de la segmentation AAM [6]	36
2.15	Isolation d'une tumeur dans une IRM cérébrale à l'aide d'un modèle de contours actifs basé sur le GVF [7]	38
2.16	Changement de topologie dans les surfaces [8]	40
2.17	Exemple d'un Hypergraphe	42

TABLE DES FIGURES

2.18	Exemple d'une coupe minimale d'un graphe	43
3.1	Agent cognitif	48
3.2	Agent reactif	49
3.3	Origine des agents (régions et contours) [9]	57
3.4	Comportement des agents dans le système [10]	63
3.5	Comportement de l'agent d'apprentissage dans le système [11]	66
3.6	Communication des agents dans l'architecture maitre-esclave [12]	68
3.7	Communication des agents dans l'architecture Island model [12]	69
3.8	Modèle de l'environnement des agents [13]	72
3.9	Exemple de possibilité de déplacement d'un agent à un instant t [13]	74
4.1	Diagramme d'activité d'un agent	80
4.2	Le voisinage d'ordre "1" d'un pixel	81
5.1	Principe d'une simulation à pas de temps constant [14]	92
5.2	Diagramme de Classes	96
5.3	segmentation d'une image synthétique hautement bruitée	97
5.4	segmentation d'une image photographique de grains de riz	98
5.5	segmentation d'une coupe de scanner thoracique	99
5.6	segmentation d'une coupe axiale d'une IRM cérébrale en pondération T1	100
5.7	segmentation d'une autre coupe axiale d'IRM cérébrale	101
A.1	Axes utilisés dans les images IRM	119

Liste des tableaux

A.1	structure du Header key	114
A.2	structure de Image dimension	116
A.3	Les différents types de codage d'un voxel	117
A.4	Les différentes valeurs de Hist.orient	118

Chapitre 1

Introduction générale

La segmentation peut être définie simplement comme le partitionnement d'une image en un ensemble de régions (ou sous-volumes) dont les membres (les pixels ou voxels) ont en commun des propriétés de cohésion.

Le partitionnement de l'image est essentiel pour extraire les objets d'intérêt qui pourront par la suite être classés et étiquetés en structures anatomiques avec des propriétés physiologiques correspondantes. Cette classification peut ainsi permettre de mettre en évidence la présence d'anomalies et de ce fait, la détection de certaines pathologies.

Tout comme il existe de nombreuses façons d'interpréter une image, il existe de nombreuses approches possibles pour résoudre le problème de la segmentation. Ces différentes approches ont en commun qu'elles sont généralement spécifiques à un type bien défini d'image et elles éprouvent donc une difficulté à segmenter des images de types différents. On obtiendra, ainsi, dans certains cas, une sous segmentation ou au contraire une sur segmentation de l'image requérant ainsi l'intervention d'un expert humain pour procéder à une correction.

L'adaptabilité des systèmes de segmentation est donc le principal défi, auquel sont confrontés les chercheurs dans le domaine de l'imagerie médicale. La difficulté réside dans la complexité des images médicales. Ce sont en effet dans la majorité des cas des images

non photographiques comme c'est le cas pour les images IRM, qui sont générées à partir d'ondes électromagnétiques. Ces ondes sont émises par les noyaux d'hydrogènes lors de leurs relaxations. Ce type d'image est caractérisé par la présence de formes complexes dont l'apparence varie d'un sujet à un autre. La nature même des images qui sont en général à niveau de gris, engendre des problèmes d'interprétation. En effet, deux tissus différents peuvent avoir la même représentation, et un tissu peut parfois avoir plusieurs représentations à cause d'interférences.

« Réalisation d'un Systèmes Multi-Agents Pour l'imagerie médicale » est un projet visant à faire cohabiter deux disciplines de l'informatique, qui sont le traitement d'image et les systèmes multi-agents. Il a pour principal objectif le développement d'un outil plus rapide et plus précis pour l'interprétation des images médicales. Pour le traitement des images, l'outil aura recours à la segmentation, qui est un procédé de partitionnement d'une image pour en extraire les objets d'intérêts.

La segmentation d'images médicales a fait l'objet d'un grand intérêt des chercheurs ces dernières années. On a donc pu voir émerger différentes techniques offrant des résultats plus ou moins satisfaisants. Ces différentes techniques ont pour principal obstacle le grand volume de données à traiter (caractéristique des images médicales) et pour principal inconvénient, le temps de calcul nécessaire au traitement de ces données. Pour permettre le traitement d'un grand volume de données dans un temps relativement acceptable, les chercheurs ont commencé à explorer l'éventualité de l'utilisation d'une nouvelle discipline qui n'est autre que les systèmes multi-agents. Les systèmes multi-agents (appelés aussi systèmes d'intelligence artificielle distribuée) est une discipline relativement jeune dans le domaine de l'informatique, et qui trouve ses racines dans l'observation de la nature et d'autres disciplines (biologie, physique...). On peut en effet, observer chez certaines sociétés d'animaux et d'insectes des comportements complexes émergeant de la coopération d'individus exécutant des opérations simple. Les systèmes multi-agents sont donc des systèmes informatiques (distribués en général) composés d'un

certain nombre d'entités appelées agents, qui exécutent des opérations simples et dont la coopération permettra l'émergence d'un comportement complexe, qui se trouve être dans le cas présent la segmentation d'une image.

La puissance de calcul qu'offrent les systèmes multi-agents (grâce à leur distributivité) permet le traitement de grands volumes de données dans des temps relativement courts, c'est la raison pour laquelle ils ont suscité l'intérêt des chercheurs en segmentation d'images. On a de ce fait, pu observer ces dernières années la publication de plusieurs travaux sur des systèmes multi-agents pour la segmentation d'images, parmi ces travaux figure l'article de S. Mazouzi, Z. Guessoum et F. Michel intitulé « Une approche multi-agent pour la segmentation d'images de profondeur » dans lequel est exposée une approche très prometteuse permettant le partitionnement d'une image de profondeur en ses différentes zones d'intérêt. Cependant, cette approche n'est pas adaptée aux images médicales.

Le projet « Réalisation d'un Systèmes Multi-Agents Pour l'imagerie médicale » sera donc dans une optique de poursuite des travaux effectués par Mazouzi et al., et aura pour but l'amélioration de l'approche existante de manière à qu'elle puisse fournir des résultats satisfaisants sur les différents types d'images médicales. L'un des objectifs du projet est donc la réalisation d'un système qui sera auto adaptatif et qui puisse acquérir de l'expérience au fil du traitement des images pour augmenter ses performances ainsi que la qualité de ses résultats.

la présente thèse est donc organisée de la manière suivante :

- Le chapitre 2 est dédié au domaine de la segmentation d'image. On y présente d'abord une des techniques les plus répandues en imagerie médicale et qui n'est autre que l'imagerie par résonance magnétique. Nous y présenterons ensuite, les différentes techniques de segmentation d'image présentes dans la littérature.
- Le chapitre 3 traite quant à lui, le domaine des systèmes multi-agent. Il définit ainsi ce qu'est un système multi-agent, ainsi que les différentes architectures SMA

réalisées à ce jour. Par ailleurs, dans ce chapitre, nous passerons en revue les différentes approches SMA publiées dans la littérature et qui traitent du domaine de la segmentation d'image.

- Le chapitre 4 est consacré à la présentation de l'approche proposée pour la segmentation multi-agent d'images médicales. Nous présentons donc dans ce chapitre les composantes du système ainsi que le comportement des agents qui y évoluent.
- Le chapitre 5 décrit l'implémentation de notre système réalisée à l'aide de la plateforme .NET. On y retrouve les différentes classes réalisées dans notre implémentation ainsi que les résultats obtenus de l'expérimentation de cette dernière sur différents types d'images.
- Enfin, nous terminerons par une conclusion générale contenant un résumé de l'apport de notre travail dans le domaine de la segmentation d'image multi-agent ainsi que les améliorations pouvant être apportés à notre approche dans de futurs travaux.

Chapitre 2

Les IRM et les différentes approches de segmentation d'images

2.1 Introduction

La segmentation d'images est un domaine qui vise à séparer les différentes zones présentes au sein d'une image. Cette discipline a vu ces dernières décennies l'apparition d'un grand nombre d'algorithmes différents et utilisant des principes distincts allant du calcul de probabilité, au calcul différentiel.

La principale raison de la multiplicité de ces algorithmes est la diversification des types d'images que l'on peut rencontrer de nos jours. Images photographique, image infrarouge, image radar, IRM et Scanner, sont différents types d'images fournissant des formes distinctes d'informations. Cette hétérogénéité fait croître la difficulté qu'ont les chercheurs pour aboutir à une technique de segmentation efficace et performante. Cela est particulièrement vrai pour les images médicales (IRM, Scanner, etc.) vu leur nature et leur complexité.

Nous allons donc dans ce chapitre donner en premier lieu quelques notions sur l'IRM, expliquant le principe d'acquisition de ce type d'images, ainsi que leurs spécificités. Nous

entamerons ensuite une présentation des différents procédés de segmentation d'images parmi les plus présents dans la littérature et ceci en les classant en fonction de leur principe de base.

2.2 L'imagerie par Résonance Magnétique (IRM)

2.2.1 Définition de l'IRM

L'IRM constitue l'une des avancées les plus significatives de l'imagerie médicale après la découverte de l'imagerie par rayon X. L'IRM est donc une technique de visualisation non invasive des tissus et des organes du corps humain, permettant la détection de malformations, pathologies et tumeurs chez les patients sans avoir recours à l'exploration chirurgicale.

2.2.2 Principes de l'IRM

La technique de l'IRM se base sur les propriétés magnétiques des atomes composant les molécules du corps humain. Elle se focalise plus précisément sur les molécules d'hydrogène contenues dans l'eau qui compose notre corps à plus de 70 pour-cent.

Le noyau d'hydrogène se comporte comme une charge en rotation autour de son axe : c'est le mouvement de spin. Les protons peuvent alors être assimilés à des dipôles magnétiques. En l'absence de tout champ magnétique, ceux-ci vont s'orienter dans l'espace de façon aléatoire. Ce mouvement confère au noyau un moment cinétique qui dépend de sa masse et un moment magnétique qui dépend de sa charge.

Dans un champ magnétique B_0 , les protons s'orientent alors par rapport à B_0 et décrivent autour de ce champ un mouvement de précession (en forme de cône), de vitesse angulaire constante. L'ensemble des protons s'orientant dans le sens de B_0 forment alors une sous-population de spins $+1/2$; l'ensemble s'orientant dans le sens inverse

forment la sous-population de spins $-1/2$. La sous-population de spins $+1/2$ étant la plus importante, il existe une aimantation résultante M_0 proportionnelle au nombre de protons présents. M_0 ne possède qu'une composante longitudinale, dans le sens de B_0 .

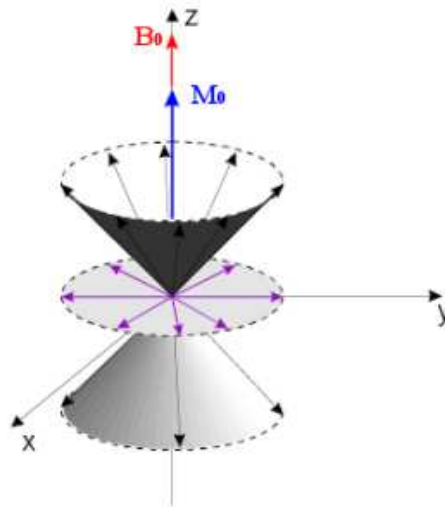


FIGURE 2.1 – Mouvement de l'hydrogène soumis au champ B_0 [1]

2.2.2.1 La résonance

À l'aide des antennes émettrices de l'IRM, on excite les noyaux d'hydrogène par l'émission d'ondes électromagnétiques de fréquence égale à la fréquence de résonance de ces derniers, perturbant ainsi, l'état d'équilibre du système (M_0 et B_0 ne sont plus alignés) : c'est le phénomène de résonance.

L'aimantation résultante de l'excitation M possède dès lors, deux composantes (une composante longitudinale et une composante transversale).

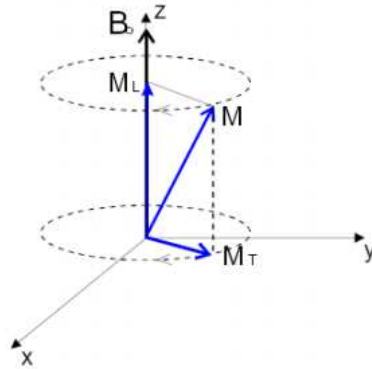


FIGURE 2.2 – Les composantes de l'aimantation lors de l'excitation [1]

2.2.2.2 La relaxation

Le retour à l'état d'équilibre des atomes d'hydrogène après la résonance, appelé aussi relaxation, où M_0 et B_0 sont de nouveau alignés requière un laps de temps qui peut être divisé en deux composantes T_1 et T_2 (une composante longitudinale et une composante transversale) relatives aux composantes du champ magnétique induit par l'excitation. Les valeurs T_1 et T_2 dépendent ainsi du tissu rencontré.

2.2.2.3 Formation de l'IRM

Lors de son passage dans l'appareil IRM, le patient est soumis à un champ magnétique initial B_0 , puis à une onde électromagnétique d'excitation B_1 . Les atomes d'hydrogène entrent alors en résonance avec l'onde de cette dernière. Au terme de la phase d'excitation, les atomes d'hydrogène commencent leur relaxation. Lors de se retour à l'équilibre ils continuent à émettre un champ appelé « décroissance d'induction libre ».

Ce signal est réceptionné par les capteurs de la machine et transformé en signal électrique qui sera analysé et codé pour constituer une image numérique. La difficulté de constitution de l'image est donc la capacité de distinction des signaux relatifs aux zones

examinées. Cette constitution est réalisée en couplant trois gradients :

- le gradient de coupe sélective sélectionne le plan de coupe, celui-ci pouvant être quelconque dans l'espace,
- le gradient de codage de phase sélectionne les lignes dans le plan de coupe sélectionné,
- le gradient de fréquence sélectionne les colonnes dans le plan de coupe sélectionné.

L'image résultant sera donc composée d'un certain nombre de coupes dont l'orientation sera en fonction de la pathologie recherchée, les principales orientations étant l'orientation axiale, l'orientation coronaire et l'orientation sagittale.



FIGURE 2.3 – Les axes se coupent (de droite à gauche axiale, sagittale et coronaire) [2]

2.2.2.4 Les pondérations des images IRM

Lors de l'acquisition d'une image IRM, les principaux paramètres influant sur le contraste de l'image sont :

- l'intensité et la constance du champ magnétique B_0 ,
- le temps de répétition TR séparant deux impulsions de l'onde B_1 ,
- le temps d'écho TE séparant l'impulsion de la lecture du signal.

Ces différents paramètres peuvent être modulés par configuration du matériel d'acquisition. On peut distinguer dans le domaine médical trois principales configurations

(appelées aussi pondérations) citées ci-dessous :

- **Image pondérée en T1** : le contraste entre les tissus dépend essentiellement de leur vitesse d'aimantation, donc de T1 et indépendant de T2.
- **Image pondérée en T2** : le contraste entre les tissus dépend de T2 et donc indépendant de T1.
- **Image pondérée en densité de protons ρ** : On obtient une séquence qui reflète le contraste entre les tissus et dépend de localisation et la concentration des noyaux d'hydrogène des différentes tissus.

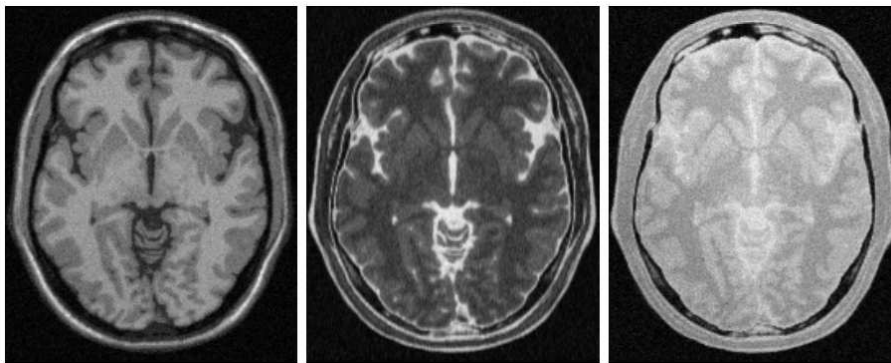


FIGURE 2.4 – Les pondération des images IRM (de gauche à droite pondération T1, T2 et ρ) [1]

2.2.3 Codage des images IRM

L'IRM fournit des images monochromes à différents niveaux de gris. Elles n'ont donc pas à gérer les niveaux RGB comme cela est le cas dans les images photographiques. La restitution d'une image à niveau de gris se fait donc par une égalité des trois composantes de l'image photographique ($R=G=B$). Cette restitution offre donc un total de 256 niveaux de gris pouvant être affichés, ce qui est largement suffisant pour les capacités de distinction de l'œil humain. Cette dynamique rencontre cependant un problème de

taille qui est la dynamique des appareils d'acquisition qui peuvent coder sur jusqu'à 12 bits. Des techniques de réduction de dynamique et/ou de fenêtrage peuvent être alors introduites.

2.2.4 Images volumiques

Par analogie à l'image numérique bidimensionnelle où le processus d'échantillonnage s'articule autour des composantes élémentaires nommées pixels, l'échantillonnage volumique comme cela est le cas pour les images IRM, ajoute une troisième dimension. Les pixels acquièrent donc une épaisseur et deviennent des voxels (contraction de «volumetric element »).

Les images IRM de par leur nature volumique doivent suivre un procédé adapté pour permettre leur visualisation, la plus répandue étant la visualisation MPR pour « Multi-Planar Reconstruction ». Cette visualisation est la résultante de la reconstitution d'images 2D sous forme de coupes du volume, selon un axe choisi par l'utilisateur. Les plans les plus utilisés étant le plan axial, le plan sagittal et le plan coronaire.

2.3 Les différentes approches de segmentation d'images

2.3.1 Qu'est ce que la segmentation d'images

La segmentation d'images est le processus qui consiste à partitionner une image en sous ensembles pour pouvoir en extraire les objets qui la composent (régions, contours.etc.). Il existe plusieurs définitions formelles de ce processus, parmi ces définitions se trouve celle de J. P. Cocquerez et S. Philipp [15] :

Définition : Soit une image A , la segmentation de A est la création d'une partition R_I telle que :

1. $\forall i, R_i \neq \emptyset$

$$2. \forall i, j; i \neq j \quad R_i \cap R_j = \emptyset$$

$$3. A = \bigcup_i R_i$$

Chaque partition R_i appelée aussi région, est un sous ensemble de pixels de l'image A partagent un ou plusieurs critères d'homogénéité qui les différencient des pixels appartenant aux régions connexes. L'appartenance d'un pixel X à une région est déterminée en utilisant un prédicat P se basant sur le ou les critères d'homogénéité tel que : *Si $P(X) = \text{vrais alors } X \in R_i$.*

Dans le contexte médical, chaque région de l'image correspond à une structure anatomique donnée comme un tissu ou un organe humain. La segmentation d'images médicales est de ce fait une opération délicate étant donnée la nature hautement complexe de l'anatomie humaine.

La segmentation est un vaste sujet d'étude et fait partie des grands thèmes de l'imagerie numérique. De nombreuses techniques ont été développées et font toujours l'objet de recherches dans les laboratoires du monde entier. Du fait de cette diversité, il est difficile de déterminer quelles sont les techniques les plus efficaces. Elles partagent néanmoins, le fait qu'elles soient en grande majorité spécifiques à une catégorie d'images.

La difficulté de présenter toutes les techniques de segmentation réside dans leur grand nombre et dans le fait que certaines techniques sont des hybrides. Nous allons cependant présenter une description des techniques les plus répandues dans les publications scientifiques de ces dernières années. Ces techniques seront organisées selon l'approche qui les régit, ceci d'après la classification présentée par Jérémy Lecœur et Christian Barillot dans le rapport de recherche « Segmentation d'images cérébrales » et illustré dans la figure 2.5 [3].

2.3.2 L'approche contour

Les contours dans une image proviennent des discontinuités de la fonction de l'intensité, ainsi que des discontinuités de profondeur. Dans la pratique, ils représentent les

frontières entre les différentes partitions ou régions dont est composée une image.

Les approches contours cherchent donc à extraire les contours présents dans l'image, en se basant sur l'étude des changements abrupts de la fonction de luminance ou sur la discontinuité des propriétés des ensembles [16]. Parmi ces approches, on peut distinguer les modèles dérivatifs et les modèles d'espace-échelle.

2.3.2.1 Les modèles dérivatifs

Les modèles dérivatifs consistent à modéliser les contours ou des zones d'images et supposent que l'image numérique provient de l'échantillonnage d'une fonction scalaire à support borné et dérivable en tout point [3]. Un contour représente généralement une variation d'intensité entre deux régions connexes d'une image. La variation de la fonction scalaire que représente le contour correspond généralement aux extrema locaux du gradient ou aux zéro du Laplacien de la fonction ; ainsi, le gradient est une fonction vectorielle des pixels $[i,j]$:

$$\nabla f[i, j] = \left(\frac{\delta f}{\delta x}[i, j], \frac{\delta f}{\delta y}[i, j] \right) \quad (2.1)$$

alors que le Laplacien est une fonction scalaire de $[i,j]$:

$$\nabla^2 f[i, j] = \left(\frac{\delta^2 f}{\delta x^2}[i, j], \frac{\delta^2 f}{\delta y^2}[i, j] \right) \quad (2.2)$$

Les méthodes dérivatives sont cependant moins adaptées au traitement d'images médicales que d'autres méthodes, car elles se basent principalement sur la présence de contraste entre les régions de l'image. Ces techniques ont aussi l'inconvénient d'être hautement sensibles au bruit présent dans l'image, ce qui est souvent le cas dans les images médicales.

2.3.3 L'approche région

2.3.3.1 Classification déterministe supervisée

2.3.3.1.1 Les réseaux de neurones

Un "neurone formel" (ou simplement "neurone") est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie"[17]. La période d'apprentissage consiste à calculer les différents coefficients, de sorte à ce que les sorties du réseau soient égales aux valeurs souhaitées (à un seuil d'erreur près) pour les exemples utilisés.

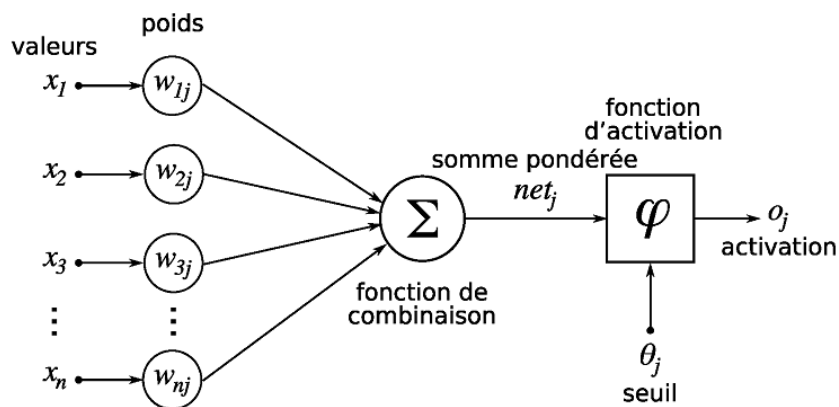


FIGURE 2.6 – Structure d'un neurone artificiel [3]

Les réseaux de neurones sont habituellement utilisés en imagerie médicale comme classificateurs, ayant pour entrées les images médicales à segmenter et pour sortie, les différentes classes de régions possibles sur ces images. On utilise un certain nombre d'exemples manuellement segmentés pour la phase d'apprentissage du réseau.

Cette période d'apprentissage est le principal inconvénient de la méthode, car l'efficacité du système sera proportionnelle au nombre d'exemples manuellement traités, ainsi qu'à leur diversité.

2.3.3.2 Classification déterministe non supervisée

2.3.3.2.1 L'algorithme des K-moyennes

L'algorithme des K-moyennes (en anglais K-means) est une méthode de classification automatique, qui a pour objectif le partitionnement des données d'une image (pixels) en K classes (ou clusters). Chaque pixel étant caractérisé par un vecteur de I attributs [18]. À partir d'une partition initiale on améliore itérativement la partition de l'espace en minimisant la variance intra-classe et en maximisant l'écartement entre les classes. Les étapes principales de l'algorithme K-Moyennes sont [19] :

1. Choix aléatoire de la position initiale des K classes ;
2. (Ré-) affecter les objets à une classe suivant un critère de minimisation des distances ;
3. Une fois tous les objets placés, recalculer les K centroïdes ;
4. Répéter les étapes 2 et 3 jusqu'à ce qu'aucune réaffectation ne soit faite.

La classification des éléments de l'image se fait de manière itérative en alternant l'étape de classification et l'étape de mise à jour des centres, jusqu'à la stabilisation de la segmentation [19].

2.3.3.2.2 L'algorithme des C-moyennes floue (fuzzy c-means)

L'algorithme des c-moyens flous (fuzzy c-means) est un algorithme de classification floue fondé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité [20]. Le nombre de classes à déterminer est initialisé avant le début des traitements. Les classes sont ensuite générées par un processus itératif en minimisant une fonction objectif. Ainsi, il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance à une région donnée.

L'algorithme utilise l'ensemble des vecteurs caractéristiques des pixels $A = \{X_1, X_2, \dots, X_n\}$ et le nombre de classes c qui correspond au nombre de régions à identifier sur l'image.

Il utilise aussi l'ensemble des vecteurs représentant les centres inconnus des classes $V = \{V_1, V_2, \dots, V_c\}$. Les valeurs des degrés d'appartenance sont regroupées dans une matrice :

$$U = [\mu_{ik}] \text{ avec } 1 \leq i \leq n, 1 \leq k \leq c$$

où $\mu_{ik} \in [0, 1]$
 et $\sum_{k=1}^c \mu_{ik} = 1 \quad \forall i$, désigne le degré d'appartenance du pixel i à la classe k .

L'algorithme FCM a donc pour objectif de modifier les partitions en minimisant la fonction suivante :

$$J_m(U, V, X) = \sum_{i=1}^n \sum_{k=1}^c (\mu_{ik})^2 (d_{ki})^2 \quad (2.3)$$

$m \in [0, \infty + [$ est le facteur flou. Il contrôle l'étendue de recouvrement entre les classes, donc le flou de la partition qui croît avec la valeur de m . L'algorithme se déroule donc de la manière suivante :

1. Initialiser V et U aléatoirement (en satisfaisant les contraintes précédemment citées).
2. initialiser $m > 1$ (en général $m = 2$).
3. Pour $i = 1..n$ et $j = 1..c$:

$$U_{ij} = \frac{1}{\sum_{k=1}^c \frac{\|X_i - V_j\|^{2/(m-1)}}{\|X_i - V_k\|^{2/(m-1)}}} \quad (2.4)$$

et pour $j = 1..c$:

$$V_j = \frac{\sum_{i=1}^n U_{ji}^m X_i}{\sum_{i=1}^n U_{ji}^m} \quad (2.5)$$

4. Le traitement s'arrête lorsque la différence entre deux itérations de U_{ji} courante et précédente est inférieure à un certain seuil E ; sinon refaire l'étape 3.

2.3.3.2.3 L'algorithme mean shift

La méthode « mean shift » se base sur un estimateur du gradient de densité non

paramétrique développé par Fukunaga et Hostetler [21] et exploité par Comaniciu et Meer [22] pour le traitement d'images. Elle est basée sur la recherche du point de plus haute densité d'une distribution de données, basée sur une estimation de densité par les noyaux de Parzen [23] dont la formule est :

$$\hat{f}(x) = \frac{k}{N.V} \quad (2.6)$$

Où x : point de la distribution ; V : volume autour de x ; N : cardinal de la distribution et k : le nombre de points dans V .

La procédure de segmentation dans ce cadre est la suivante [3] :

1. Considérer les images en terme de caractéristiques (via couleur, gradient, mesures de texture, etc.).
2. Choisir une répartition uniforme des fenêtres de recherche initiales.
3. Calculer le centroïde des données pour chaque fenêtre.
4. Centrer la fenêtre de recherche sur le centroïde de l'étape 3.
5. Répéter les étapes 3 et 4 jusqu'à convergence.
6. Fusionner les fenêtres se trouvant au même point final.
7. Grouper les données traversées par les fenêtres fusionnées.

2.3.3.3 Classification Probabiliste Non-Paramétrique

2.3.3.3.1 Champs Aléatoires de Markov

Le formalisme des champs de Markov permet d'effectuer une segmentation de l'image en prenant en compte les interactions avec les pixels voisins. Pour la présentation de cette méthode, nous aurons besoins des définitions suivantes[24] :

L'image (un vecteur de variables aléatoires) : Soit une image sur une maille dont les sommets constituent un ensemble S de sites s (les pixels de l'image) tels que $S = \{s = (i, j); 1 \leq i, j \leq N\}$.

A chaque site s est associé une variable aléatoire Y_s , dont les valeurs appartiennent aux différents niveaux de gris que peuvent prendre les pixels. L'image ($N \times N$) est donc représentée par un vecteur de variables aléatoires $Y = \{Y_s, s \in S\}$ dont le vecteur $y = \{y_s, s \in S\}$, est appelé configuration (ou réalisation).

$P_Y(y)$ est la probabilité de la configuration y , c'est la probabilité $P_Y(y_s, s \in S)$.

$P_Y(y_s)$ est la loi marginale de Y_s .

Système de voisinage : Un système de voisinage $v = (v_s, s \in S)$ est formé de parties

v_s de S , vérifiant les propriétés suivantes :

- $s \notin v_s$
- $s \in v_t \Leftrightarrow t \in v_s$

L'ensemble v_s est appelé le voisinage de s . t est dit voisin de s si $t \in v_s$.

Champ Markovien : Un champs aléatoire x est dit champ Markovien associé à ce système de voisinage seulement si

- $P_x(x) > 0$
- $P_{X_s|X_r}(x_s | x_s, r \neq s) = P_{X_s|X_r}(x_s | x_s, r \in v_s)$

Clique : Un sous ensemble c de S est appelé une clique relative au système de voisinages v si

- c est un singleton,
- 2 pixels quelconques de c sont voisins (au sens de v),
- l'ordre d'une clique correspond au nombre de ses éléments.

Mesure de Gibbs et champ Markovien : Le théorème de Hammersley-Clifford démontre que X est un champ Markovien seulement si $P(X)$ est une mesure de Gibbs définie par

$$\forall x \in \Omega, \quad P_X(x) = \frac{1}{Z} \exp(-U(x))$$

où Ω est l'ensemble des configurations possible pour X et U est une fonction

appelée énergie tel que

$$U(x) = \sum_{s \in S} \left[\frac{(y_s - \mu_{x_s})^2}{2\delta x_s} + \frac{1}{2} \log(2\pi)\delta x_s \right] + \beta \sum_{C=\{s,t\}} \phi(x_s - x_t)$$

β joue le rôle de coefficient de régularisation. Plus β est grand, plus la transition entre régions est pénalisée et plus les régions obtenues sont grandes. Le choix de la fonction φ influe grandement sur le résultat (notamment en ce qui concerne la convexité de $U(x)$), ainsi, le choix d'une forme quadratique pure qui correspond à la régularisation de Tikhonov pénalise les forts gradients, donnant alors une segmentation plus lisse. Une estimation du champ des étiquettes peut se faire suivant le critère du Maximum a posteriori. Cela se fait en minimisant l'énergie grâce à l'algorithme des modes conditionnels itérés ou du recuit simulé[3].

2.3.3.3.2 Machines à Vecteur de Support (SVM)

Les SVM sont un ensemble de techniques d'apprentissage supervisé développées dans les années 90 par Vladimir Vapnik [25] et destinées à résoudre des problèmes de discrimination (classification) et de régression (prédiction de la valeur d'une variable) par l'intermédiaire de fonctions dites « noyau ».

Pour deux classes d'exemples données, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan. Il existe cependant un grand nombre d'hyperplans pouvant être déterminés, l'avantage des SVM est de pouvoir déterminer l'hyperplan optimal appelé aussi séparateur à vaste marge, dont la distance minimale aux exemples d'apprentissage est maximale.

L'utilisation de cette technique de classification pour la segmentation d'images a fait l'objet de plusieurs publications parmi les quelles l'article de Z. Lao et al. intitulé « Segmentation automatique des lésions de la substance blanche sur des IRM 3D du

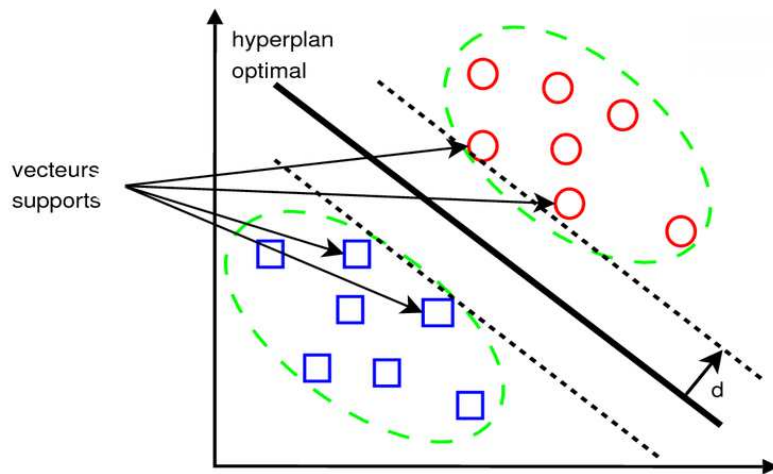


FIGURE 2.7 – Hyperplan optimal d'une machine à vecteur de support

cerveau » dans lequel les SVM sont utilisées pour segmenter des image IRM pour détecter d'éventuelles lésions dans la substance blanche du cerveau [26].

L'inconvénient des SVM est qu'elles se basent sur un ensemble de données d'exemples pour la période d'apprentissage, les résultats obtenus dépendent donc de ces données utilisées lors de l'apprentissage. L'application de cette technique à la segmentation d'images médicales augmente la gravité de cet inconvénient car l'anatomie de chaque individu est unique.

2.3.4 L'approche Structurelles

2.3.4.1 Gradient Morphologique

La morphologie mathématique est une théorie de traitement non linéaire de l'information introduit par Mathero et Serra utilisée dans le domaine du traitement d'images, basée non pas sur le traitement du signal mais sur la théorie des ensembles. La morphologie mathématique se base sur la détection de contours et ce au travers d'un ensemble de tests utilisant les opérations de base d'érosion et de dilatation.

Soit B un élément structurant et B_x cet élément centré en un pixel x . L'érosion consiste à poser en chaque pixel x d'un objet X , la question : " B_x est-il contenu entièrement dans X ?". L'ensemble des positions x correspondant à une réponse positive forme le nouvel ensemble $\epsilon_{B(X)}$, appelé érodé de X par B . Autrement dit [3] :

$$\epsilon_{B(X)} = \{x | B_x \subseteq X\}. \quad (2.7)$$

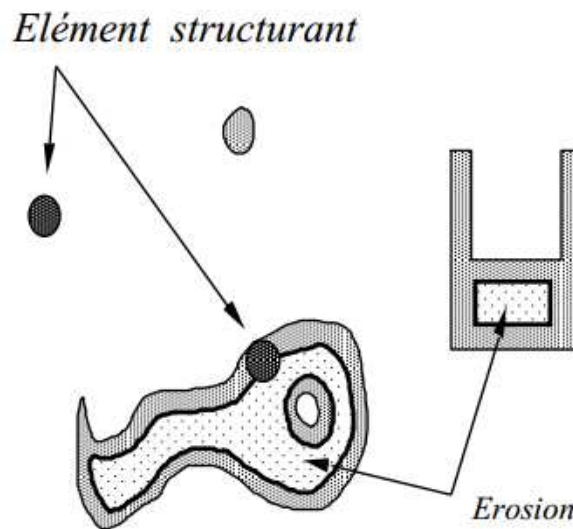


FIGURE 2.8 – Érosion de plusieurs objets d'une scène par un élément structurant

L'opération de dilatation se définit de manière analogue à l'érosion. En prenant le même élément structurant B , on pose pour chaque point z de l'image la question " B_z touche-t-il l'ensemble X ?".

C'est à dire, y a-t-il une intersection non vide entre B_z et X ? L'ensemble des points de l'image correspondant aux réponses positives forme le nouvel ensemble $\delta_{B(X)}$, appelé dilaté de X par B . C'est-à-dire [3] :

$$\delta_{B(X)} = \{z | B_z \cap X \neq \emptyset\} \quad (2.8)$$

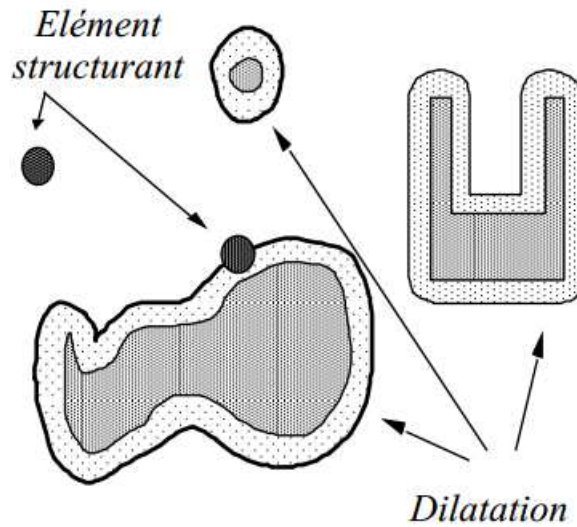


FIGURE 2.9 – Dilatation de plusieurs objets d'une scène par un élément structurant

La différence symétrique entre l'image dilatée et érodée par le même élément structurant de taille unitaire donne le gradient morphologique qui est un opérateur de détection de contours et qui peut se résumer par l'équation [3] :

$$mgrad_{B_1}(X) = \delta_{1B}(X) / \epsilon_{1B}(X) \quad (2.9)$$

Le gradient morphologique a été utilisé dans plusieurs publications dans le domaine de l'imagerie médicale. Parmi ces publications nous citons l'algorithme proposé par Hsiao et al. pour la segmentation de tumeurs cérébrales qui couple le gradient morphologique et la croissance et fusion de région pour segmenter les différentes zones du cerveau et détecter les tumeurs [27].

2.3.4.2 Ligne de partage des eaux

L'algorithme de la ligne de partage des eaux est une technique de segmentation d'images inspirée de la topographie géographique et utilisée pour la segmentation d'images

à niveaux de gris. L'algorithme de la ligne de partage des eaux considère l'image comme étant un relief où, le niveau de gris d'un pixel correspond à son altitude. Cette représentation topographique de l'image laisse apparaître un certain nombre de bassins et de crêtes. En considérant ce relief comme étant troué aux emplacements de ses minimas, L'immersion de celui-ci dans de l'eau provoquerait une inondation des bassins formant ainsi des lacs. Les lignes de partages des eaux correspondent donc aux points où deux lacs disjoints se rejoignent.

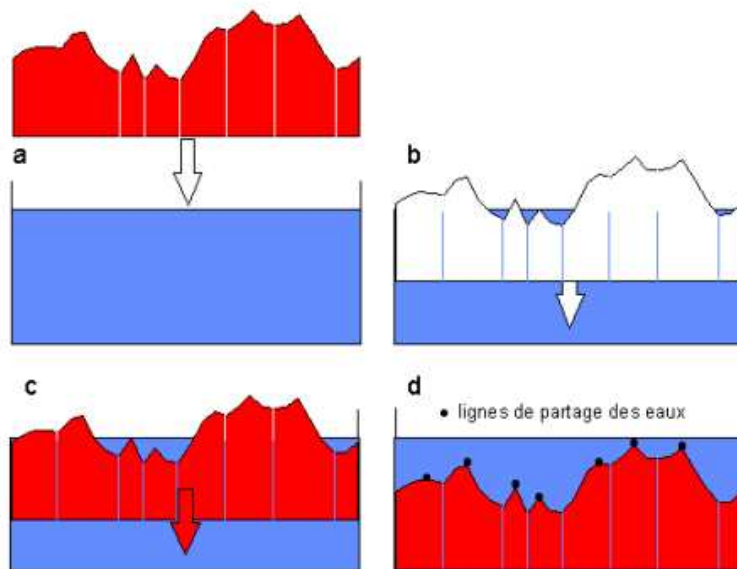


FIGURE 2.10 – Principe de la Ligne de Partage des Eaux

Une des difficultés rencontrées lors de l'implémentation d'une analogie intuitive comme celle-ci est qu'elle laisse beaucoup de liberté quant à sa formalisation. Il existe deux classes principales d'implémentation : l'une est basée sur un algorithme récursif d'immersion et l'autre basée sur l'utilisation de fonctions de distances géodésiques [3].

L'algorithme de la ligne de partage des eaux est une technique de segmentation assez simple à implémenter mais elle a l'inconvénient majeur de générer une sur-segmentation

de l'image, ce qui est très gênant dans le domaine de l'imagerie médicale. C'est pour cette raison que dans la majorité des publications, elle est couplée à une autre technique comme la fusion de régions par exemple.

2.3.5 L'approche Forme

2.3.5.1 Recalage d'atlas

La segmentation basée sur l'atlas repose sur l'existence d'une image de référence (l'atlas) dans laquelle les objets d'intérêt ont été soigneusement segmentés, manuellement ou avec l'aide d'une méthode de segmentation semi-automatique. Le résultat de cette segmentation est un ensemble de masques créés, chacun d'entre eux correspondant à une zone d'intérêt de l'image de référence. Pour segmenter une nouvelle image, on réalise une transformation de l'atlas pour faire correspondre celui-ci avec l'image à segmenter ceci par l'intermédiaire de l'utilisation d'un algorithme de mise en correspondance. Cette transformation sera ensuite utilisée pour déformer les masques des objets d'intérêt à partir de l'atlas et déterminer ainsi leurs positions dans l'image à segmenter [4].

En imagerie médicale, l'approche basée sur l'atlas aligne l'atlas à l'image cible par une transformation globale. Puis, l'affinement local de chaque partie de l'atlas est effectué de façon précise pour extraire les contours des parties anatomiques d'intérêt.

Réaliser un recalage d'atlas basé sur une seule image de référence peut s'avérer problématique. Tout d'abord, l'unique image sélectionnée peut ne pas être un exemple typique. Deuxièmement, la nature unique de l'anatomie humaine ainsi que la diversité de pathologies possiblement rencontrées rendent donc l'utilisation d'un seul atlas (une seule image de référence) difficile pour déterminer si une forme a été correctement transformée ou non. Pour résoudre ce problème une nouvelle solution, appelée « Atlas probabiliste », a été utilisée. Cette nouvelle solution est basée sur un ensemble d'atlas d'apprentissage.

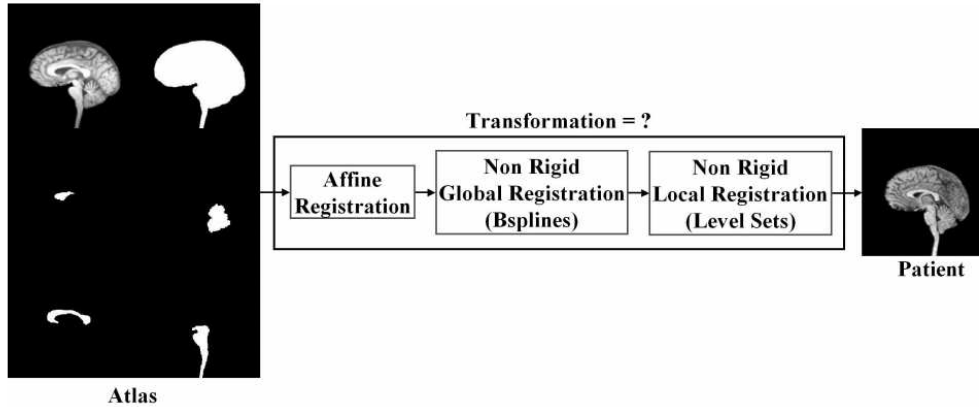


FIGURE 2.11 – Processus de segmentation par l'approche d'atlas utilisé dans [4]

Il représente la distribution spatiale de la probabilité qu'un pixel appartient à un organe particulier [28].

Le recalage d'atlas est une approche très répandue dans la segmentation d'images médicales, car elle permet d'isoler les différentes structures anatomiques comme les organes et les tissus. On trouve ainsi des publications utilisant l'approche du recalage d'atlas traitant différents types d'images médicales (images cérébrales [29], image cardiothoracique [30], IRM de la prostate [31]...etc.). Bien qu'ayant un grand potentiel dans la segmentation d'images médicales, le recalage d'atlas a cependant un inconvénient majeur qui est la constitution de l'atlas. Le choix des images et leurs segmentations manuelles ou semi-automatiques sont les principaux facteurs influençant les performances de l'approche.

2.3.5.2 Transformation dans un sous-espace

2.3.5.2.1 Analyse par ondelettes

Une ondelette est une fonction Ψ de $L^2(R)$ (espace des fonction carrée intégrable

réelles) dont la moyenne est nulle et dont l'énergie tend vers zéro à l'infini.

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \quad (2.10)$$

À partir de cette unique fonction qu'on appellera "Ondelette Mère", on construit une famille de fonctions appelées atomes de base :

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) \quad (2.11)$$

avec u et s sont appelés facteur d'échelle et facteur de translation. L'analyse par ondelettes des signaux permet de prendre en compte des événements de taille différentes par décomposition d'un signal $f(t)$ sur une famille d'ondelette obtenu par dilatation et translation de la fonction mère, la décomposition du signal ce fait à différents niveaux de résolutions. La transformée en ondelettes est donc définie par :

$$Wf(u, s) = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t-u}{s}\right) dt \quad (2.12)$$

où l'atome de base y est une fonction de moyenne nulle, centrée au voisinage de 0 et d'énergie finie.

Cette procédure est appliquée sur un signal digitalisé en utilisant un banc de filtre passe bas et passe haut. Les coefficients résultant de l'analyse se composent de coefficients approximés qui représente l'image originale au n^{eme} niveau de résolution et des coefficients d'ondelettes qui représentes les détails horizontaux, verticaux et diagonaux perdus entre deux niveaux successif. Cette démarche qui réalise récursivement la décomposition des coefficients issus du banc de filtres est appelée analyse pyramidale [32].

2.3.5.2.2 Modèle à formes actives

Le modèle à formes actives est une classe de modèles déformables, qui intègrent dans le processus de segmentation, des connaissances a priori statistiques sur la forme et les modes de déformation de la structure étudiée, introduite initialement par Cootes et al [33].

Pour localiser des objets déformables dans des images médicales, elle se base principalement sur une connaissance a priori statistique sur la forme et les modes de déformation des structures d'intérêt. Ces connaissances sont déduites, à partir de N échantillons d'images présentant les variations possibles de la structure étudiée. Dans un premier temps, ces échantillons sont manuellement étiquetés. Sur chaque image, n points caractéristiques sont positionnés sur le contour de la région d'intérêt. Chaque forme sera alors modélisée par un vecteur construit par concaténation des coordonnées des points placés sur son contour [34] :

$$V = (x_{i1}, y_{i1}, x_{i2}, y_{i2}, \dots, x_{in}, y_{in}) \quad (2.13)$$

avec (x_{ij}, y_{ij}) sont les coordonnées du point j dans l'image i .

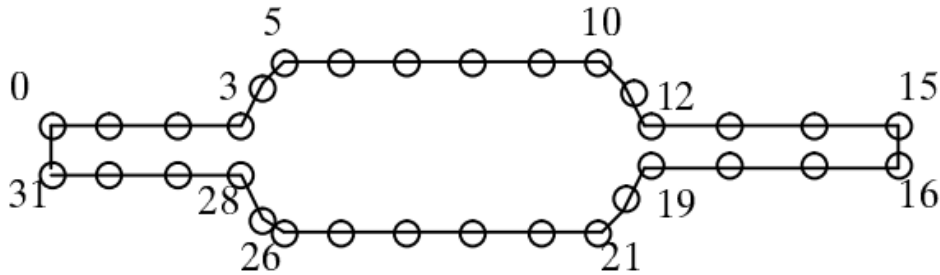


FIGURE 2.12 – Étiquetage manuel des formes par N points

Ensuite, un algorithme itératif permet d'aligner les formes obtenues. La détermination de la forme moyenne ainsi que les principaux modes et amplitudes de déformation est réalisée par une analyse en composantes principales (ACP). Finalement, un modèle de formes est obtenu en décrivant les variations autorisées de la forme. Ce modèle est défini par l'équation suivante :

$$v = \bar{v} + P_f b_f \quad (2.14)$$

Avec v la forme moyenne, principaux modes de déformation de la forme et b_f de une matrice des poids, représentant la projection de la forme v dans la base P_f . La phase de

localisation s'appuie sur ce modèle pour localiser la structure recherchée. Elle consiste à commencer par une estimation initiale, placée proche du contour cible sur l'image à segmenter (généralement la forme moyenne). Il faudrait ensuite, déplacer itérativement cette estimation vers les frontières de la structure étudiée, en se basant sur les propriétés de luminance de l'image. A chaque itération, la nouvelle P_f la matrice des formes obtenues, doit appartenir à l'espace autorisé imposé par le modèle de formes cité ci-dessus, jusqu'à convergence [34].

Les modèles à formes actives ont été utilisés dans plusieurs publications pour la segmentation d'images médicales, parmi lesquelles celle de Nicolae Duta et Milan Sonka [5] qui ont proposé une méthode améliorée d'ASM pour la segmentation et le traitement d'IRM cérébrales. Zhao et Teoh [35] ont quand a eux proposé une technique de modèle à formes actives pour la segmentation de structures cérébrales en trois dimensions n'utilisant qu'un nombre restreints d'exemples d'apprentissage.

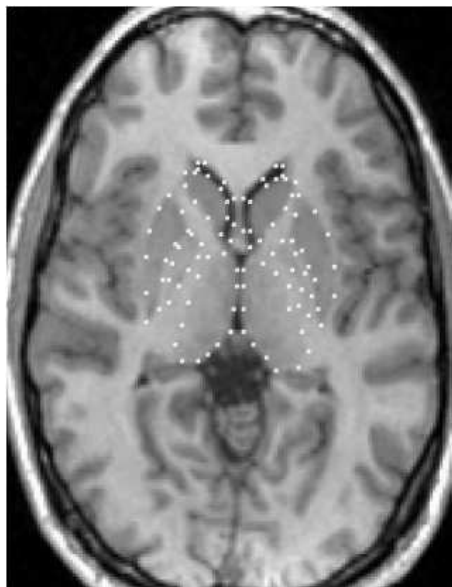


FIGURE 2.13 – Étiquetage manuel des formes Sur une IRM cérébrale [5]

La méthode ASM a pour principal inconvénient la nécessité d'utiliser un échantillon

lors de la phase d'apprentissage. Cet échantillon doit être le plus représentatif possible des différentes structures qui pourront être rencontrées lors de la segmentation. Toute forme ne faisant pas partie du modèle défini lors de l'apprentissage aura peu de chance d'être extraite correctement.

2.3.5.2.3 Modèles actifs d'apparence

Les modèles actifs d'apparence (en anglais Active Appearance Model ou AAM) peuvent être considérés comme un perfectionnement des modèles à formes actives. Ils permettent de construire un modèle statistique de l'objet à segmenter incluant à la fois la forme comme c'est le cas dans l'ASM, ainsi que les niveaux de gris (la texture) de cet objet.

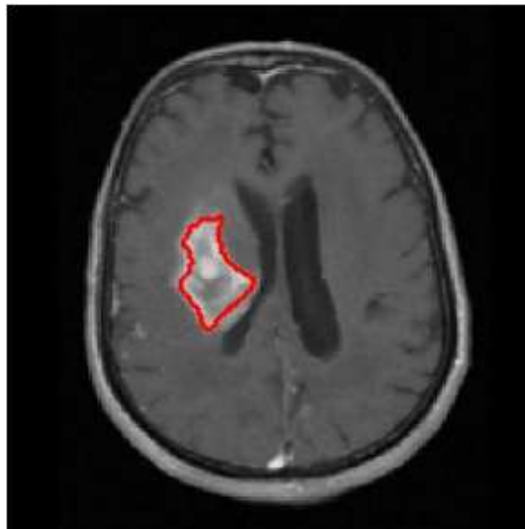


FIGURE 2.14 – Isolation d'une tumeur dans une IRM cérébrale à l'aide de la segmentation AAM [6]

Comme les ASM, les AAM nécessitent une période d'apprentissage où un certain nombre de points caractérisant les différentes formes sont placés sur les images d'entraînement. La forme moyenne ainsi que ses principaux modes de variation sont ensuite calculés par ACP. Pour constituer le modèle de texture, les images d'entraînement sont

préalablement déformées par un algorithme de triangulation de manière à mettre en coïncidences les points du modèle de forme avec ceux de la forme moyenne. Les niveaux de gris des pixels situés à l'intérieur des contours sont ensuite échantillonnés. Une analyse en composante principale permet ensuite de calculer les modes principaux de variation de la luminance. Enfin, le modèle d'apparence est obtenu en concaténant les paramètres de forme et de luminance de chaque image. La convergence du modèle d'apparence se fait par un algorithme itératif. Les paramètres sont progressivement ajustés de manière à minimiser la distance entre l'image de départ et l'image synthétisée par combinaison linéaire [36].

Les modèles actifs d'apparence ont comme les ASM l'inconvénient majeur de leur période d'apprentissage statistique. L'élaboration d'un ensemble d'images manuellement segmentées utilisées pour la phase d'entraînement reste une opération fastidieuse, ceci est d'autant plus vrai pour les images médicales où chaque image est en général composée d'un nombre conséquent de slices et ne peut être correctement segmentée que par un spécialiste.

2.3.5.3 Formes paramétriques

2.3.5.3.1 Contours actifs

Les contours actifs, ou snakes, ont été introduits par Kass en 1987 [37] et sont toujours utilisés aujourd'hui sous des formes plus évoluées. Ce modèle présente le problème de la segmentation directement comme un problème de minimisation de fonction d'énergie. Dans le modèle des contours actifs, le snake est modélisé par une courbe paramétrée C telle que :

$$C = \{v(s) = (x(s), y(s); s \in [a, b])\} \quad (2.15)$$

S est l'abscisse curviligne. Le contour initial v_0 est initialisé par l'utilisateur près des frontières de l'objet à détecter. La courbe est ensuite déplacée et déformée au cours d'un

processus itératif cherchant à minimiser une fonction d'énergie E :

$$E(v(s)) = \int_{\Omega} \alpha v'(s)^2 ds + \int_{\Omega} \beta v''(s)^2 ds - \lambda \int_{\Omega} |\nabla I(v(s))|^2 ds \quad (2.16)$$

E_{int} : L'énergie interne dépend uniquement de la forme du snake .Elle regroupe des notions comme la courbure du contour ou la régularité d'espacement des points. Elle est constituée de $v'(s)$ et $v''(s)$ les dérivées premières et secondaires du contour, Avec les constantes réelles α et β sont respectivement les coefficients d'élasticité et de rigidité de la courbe. Le choix des paramètres α et β impose les caractéristiques de déformation du contour.

E_{ext} : L'énergie externe est définie par l'utilisateur selon les spécificités du problème et associe le contour aux données de l'image (comme l'intensité), Elle est constituée de $\nabla I(v(s))$ un gradient de luminance de l'image, Avec la constante réelle λ permettant de pondérer l'influence extérieure [6].

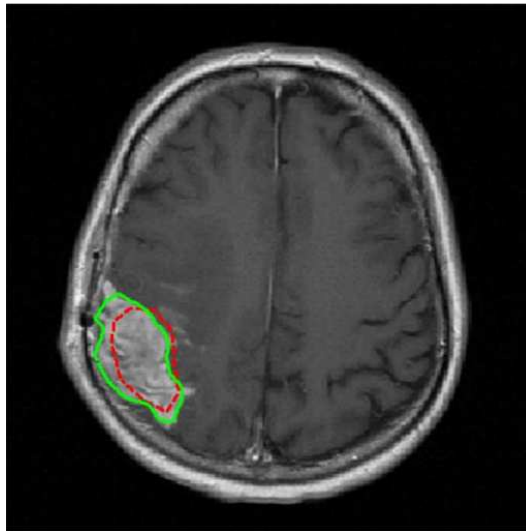


FIGURE 2.15 – Isolation d'une tumeur dans une IRM cérébrale à l'aide d'un modèle de contours actifs basé sur le GVF [7]

Dans le modèle de contours actifs, une courbe fermée ou ouverte est initialisée à proximité de l'objet à détecter. Le système commence alors à déformer et déplacer cette courbe de manière itérative et ce suivant les contraintes définies au sein du modèle. L'évolution du contour s'effectue selon un processus itératif de déformations contrôlées par un test de convergence.

Bien qu'étant très répandu dans la littérature, le modèle de contours actifs présente des inconvénients parmi lesquels la sensibilité à la présence de bruit dans l'image, la difficulté de détermination des différentes constantes du modèle et l'inconvénient principal qui est l'obligation d'initialiser le contour aux abords de l'objet à extraire.

Pour trouver une solution à cet inconvénient Xu et Prince [38, 39] ont développé un nouveau type de snake qui permet une initialisation éloignée de l'objet et arrive à segmenter de façon correcte en levant l'obstacle des concavités. Ce nouveau snake est basé sur un nouveau type de champ de force externe, appelé gradient « vector flow » ou GVF. Ce champ est calculé comme la diffusion spatiale du gradient d'une carte de bords (ou edge map). Combiné aux forces usuelles, ce GVF améliore grandement la qualité de la segmentation, notamment pour les images cardiaques et cérébrales où les frontières sont très complexes [3].

2.3.5.4 Forme non-paramétrique

2.3.5.4.1 Level set

Le principe des surfaces de niveau est inspiré de la théorie de la propagation des courbes subissant une force normale. Elle consiste à considérer une courbe C se propageant dans un l'environnement qu'est l'image et définie par :

$$C : \begin{cases} [0, \infty[& \rightarrow R^n \\ t & \mapsto C(t) \end{cases} \quad (2.17)$$

comme étant l'ensemble de niveau zéro d'une hypersurface, ou fonction définie dans un espace de dimension $n + 1$, notée Ψ et tel que :

$$\Psi : \begin{cases} \mathbb{R}^n \times [0, \infty[& \rightarrow \mathbb{R} \\ (x, t) & \mapsto \Psi(x, t) \end{cases} \quad (2.18)$$

L'expression de l'interface C peut être alors exprimée de la manière suivante :

$$C(t) = \{x \in \mathbb{R}^2 | \Psi(x(t), t) = 0\} \forall t \in [0, \infty[\quad (2.19)$$

Au lieu de modéliser directement l'évolution de C , on s'intéresse à l'évolution de Ψ , en faisant l'hypothèse qu'à chaque instant, on pourra retrouver C en prenant l'ensemble de niveau zéro de Ψ et ce, en utilisant l'équation d'évolution de l'hypersurface par la formule de Hamilton-Jacobi, tel que :

$$\frac{\partial \Psi}{\partial t} + F(k)|\nabla \Psi| = 0 \quad (2.20)$$

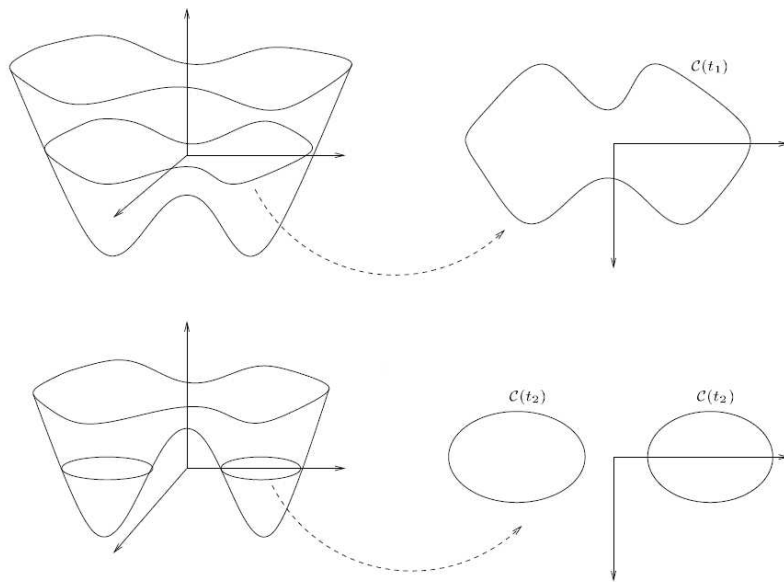


FIGURE 2.16 – Changement de topologie dans les surfaces [8]

Parmi les principaux avantages de la méthode des surfaces de niveau, nous pouvons d'abord citer sa capacité à permettre des changements de topologie (séparation ou fusion de deux interfaces) sans implémentation particulière ni paramétrisation du contour de l'interface. Cette méthode permet aussi l'utilisation en trois dimensions et peut aisément calculer les caractéristiques géométriques de l'interface C implicitement représentées par Ψ .

Cette méthode nécessite la ré-actualisation de l'hypersurface à chaque itération pour l'ensemble des points de la grille. Cela conduit à une complexité en $O(N^n)$, où N est le nombre de points de la grille et n le nombre d'itérations nécessaires pour converger. En 3D, cela correspond à un temps de calcul rédhibitoire, même sur des machines puissantes [40].

2.3.6 L'approche théorie des graphes

Dans les approches de segmentation d'images utilisant la théorie des graphes, on considère l'image comme un ensemble d'éléments (pixels) liés entre eux par un certain nombre de relations (composante spatiale et radio-métriques). La création d'un graphe représentant l'image permet l'utilisation de toutes les méthodes d'optimisation et outils développés dans le domaine de l'optimisation et de la recherche opérationnelle.

Parmi les approches de segmentation d'images utilisant la théorie des graphes, nous nous intéresserons à celles des Hypergraphes, ainsi qu'à celle des graphe-cuts.

2.3.6.1 Les Hypergraphes

Les hypergraphes représentent une généralisation de la définition des graphes, où les arêtes ne relient plus un ou deux sommets, mais un nombre compris entre un et le nombre de sommets de l'hypergraphe. Soient $V = \{v_1, v_2, \dots, v_n\}$ un ensemble, $E = \{E_1, E_2, \dots, E_m\}$ une famille de parties de V , avec $(m, n) \in \mathbb{N}^{*2}$ deux entiers non nuls.

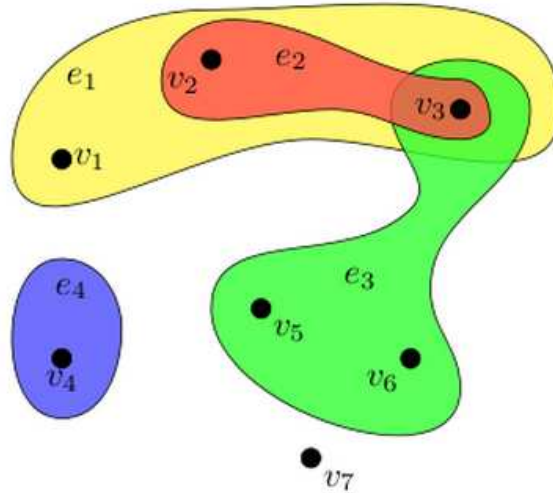


FIGURE 2.17 – Exemple d'un Hypergraphe

Un hypergraphe H est un couple (V, E) tel que :

$$\begin{aligned} \forall i \in ([1, m] \cap N) \quad E_i \neq \emptyset \\ \bigcup_{i=1}^m E_i \subseteq V \end{aligned} \tag{2.21}$$

À l'instar des graphes classiques, on dit que :

- Les éléments de V sont les sommets de H .
- Le nombre de sommets n est l'ordre de l'hypergraphe.
- Les éléments de E sont les arêtes ou hyper-arêtes de H .

La tâche principale de l'approche utilisant les hypergraphes, est la définition des relations qui déterminent les hyper-arêtes reliant les sommets (pixels) du graphe. La première relation utilisée fut la relation de voisinage d'ordre n des pixels, la seconde relation utilisée fut la relation de voisinage colorimétrique où on ajoute la notion de distance radiométrique, les pixels sont ainsi reliés entre eux par rapport à leurs positions spatiales et leurs similarités en terme de couleurs et d'intensités lumineuses.

Parmi les différentes publications sur la segmentation d'images par hypergraphes, on retrouve celle de Rital et al. [41] qui propose une approche basée sur le partitionnement d'un hypergraphe pondéré pour la segmentation d'images à niveaux de gris (comme les images médicales).

2.3.6.2 Graphe Cuts

Comme pour les hypergraphes, la segmentation d'images par Graph-Cuts est une méthode qui ramène le processus de segmentation à un problème de coupe minimale d'un graphe de voisinage pondéré représentant l'image. Les pixels de l'image sont donc représentés par des sommets et sont liés à leurs voisins par des arêtes. Ces arêtes sont pondérées et leur pondération est inversement proportionnelle au degré de similarité entre pixels.

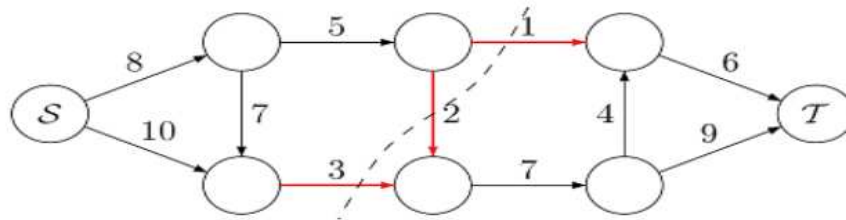


FIGURE 2.18 – Exemple d'une coupe minimale d'un graphe

L'image est donc représentée par un graphe $g = (V, E)$ orienté, où V est l'ensemble des sommets du graphe (pixels), et E l'ensemble de ses arêtes. Nous définissons aussi une fonction de poids $w(e) > 0$ utilisée pour la pondération des arêtes. Il est aussi nécessaire de distinguer deux types particuliers de sommets qui sont les sommets sources notés s (associés aux pixels composant les objets de l'image) et les sommets puits notés t (associés aux pixels composant le fond de l'image).

On appelle coupe dans un graphe une partition (V_1, V_2) de l'ensemble des sommets V

tel que : $V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$, $s \in V_1$ et $t \in V_2$. La coupe minimale pour un graphe G sera donc une coupe qui minimisera la somme du poids des arêtes reliant V_1 à V_2 .

Lors d'une segmentation d'images par Graph Cuts, l'utilisateur initialise manuellement un certain nombre de sources à l'intérieur de l'objet à segmenter (Sommets s), et un certain nombre de puits à l'extérieur de l'objet. Les contours de l'objet sont ainsi calculés en déterminant la coupe minimale du graphe G à l'aide d'un des différents algorithmes réalisés à cet effet.

Dans la littérature, la méthode des Graph Cuts n'est généralement pas utilisée toute seule, mais plutôt couplée à d'autres techniques de segmentation. Nous pouvons citer par exemple les travaux de Xu et al. [42] qui associent contours actifs et Graph Cuts pour la segmentation d'images cérébrales, ou encore le travail de Grosgeorge et al.[43] qui proposent une segmentation par Graph Cuts basée sur un modèle de formes pour segmenter des IRM cardiaques.

2.4 Conclusion

Il nous a paru essentiel de bien expliciter les caractéristiques particulières des images médicales, c'est pour cela que nous avons entamé ce chapitre par des notions de base sur les images IRM, un type bien rependu d'images médicales particulièrement difficiles à traiter. Nous y avons entre autres défini le système d'acquisition de ces images ainsi que la manière dont elles sont numériquement codées.

Après cela, nous avons défini de façon formelle ce qu'était le processus de segmentation d'une image, nous avons ensuite passé en revue les différentes techniques de segmentation d'images présentes dans la littérature parmi lesquelles les approches basées sur les contours, sur les formes, sur les régions, sur les graphes, etc. Nous avons pu ainsi constater la grande diversité qu'offre ce domaine en termes d'approches possibles ainsi que de l'attention particulière dont il fait l'objet à l'heure où les images prennent une

place importante dans nos vies de tous les jours.

Grâce aux informations collectées et présentées dans ce chapitre, il nous a paru plus judicieux de faire coopérer plusieurs techniques dans un même système, plutôt que d'utiliser une méthode classique. Cette coopération se fera aux sien de ce qu'on appelle système multi-agent, nous allons présenter ces systèmes ainsi que leurs caractéristiques dans le chapitre suivant.

Chapitre 3

Les systèmes multi-agents et la segmentation d'images

3.1 Introduction

Les systèmes multi-agents ont émergé de la collaboration de plusieurs domaines que sont l'intelligence artificielle, les systèmes distribués et le génie logiciel. Cette nouvelle discipline peut être assimilée à une mise en pratique du célèbre proverbe « l'union fait la force », elle se base ainsi sur la conception de systèmes composés d'un ensemble d'unités en interaction et qui coopèrent à la réalisation d'une fonction bien déterminée. Les approches multi-agents offrent de ce fait, une vision collaborative pour la résolution de problèmes.

Nous allons donc introduire dans ce chapitre quelques notions sur les agents et sur les systèmes multi-agents, nous étudierons ensuite différentes publications parues dans la littérature et proposant des approches de segmentation d'images par l'utilisation d'un système multi-agents. Nous terminerons par une conclusion dans laquelle nous discuterons des avantages qu'apportent les systèmes multi-agents à la segmentation d'images, ainsi que des obstacles auxquelles ils doivent faire face.

3.2 Les Systèmes Multi-Agents

3.2.1 Définition des systèmes multi-agents

La théorie derrière le concept de systèmes multi-agents n'est pas complètement figée, elle a été en constante évolution tout au long de ces dernières années. Les scientifiques de part le monde ont développé différents formalismes pour représenter les propriétés statiques et dynamiques des agents. Certains de ces scientifiques ont vu les agents comme des systèmes autonomes possédant des comportements ou des attitudes humaines, comme les croyances, le libre arbitre, les intentions, la conscience, les capacités, les engagements, ou des désirs. Cependant, il semble que la représentation de ces concepts de façon descriptive soit une tâche très difficile [44].

Un système multi-agents est un ensemble d'agents autonomes en interaction, capables de s'organiser d'une manière dynamique et adaptative. Les agents disposent des propriétés suivantes [45] :

- Un agent doit être autonome (fonctionne sans intervention directe des humains ou d'autres systèmes, il a le contrôle sur ses actions et son état interne) ;
- Un agent doit être sociable (interagit par le biais d'un langage de communication agent avec les autres agents du système) ;
- Un agent doit être réactif (perçoit l'environnement dans lequel il se trouve et répond rapidement à ses changements) ;
- Un agent doit être Proactif (il prend des initiatives pour atteindre son but).

Les agents sont autonomes et peuvent être de nature hétérogène. Par conséquent, les systèmes multi-agents sont considérés habituellement comme une collection, une organisation ou une société d'agents individuels qui interagissent dans un environnement commun, chacun d'eux ayant ses propres connaissances, ses buts, et sa capacité d'action [46]. Ces agents peuvent former des groupes d'intérêts, où ils peuvent agir ensemble pour un objectif commun.

3.2.2 Architectures des systèmes multi-agents

Selon une classification largement acceptée par la communauté des SMA, et faite par Woolridge et Jennings [47], les systèmes multi-agents peuvent être divisés en trois catégories selon leurs architectures : les architectures cognitives, les architectures réactives (délibératives) et les architectures hybrides.

3.2.2.1 L'architecture cognitive (délibérative)

L'architecture cognitive (délibérative) est définie dans [47]. Les agents y sont considérés comme un type particulier de systèmes basés sur la connaissance et où le raisonnement est purement logique. Dans ces systèmes, les agents utilisent leurs raisonnements pour réaliser leur but, ils coopèrent généralement entre eux à l'aide de messages, partagent ainsi leurs connaissances et améliorant leurs perceptions du monde qui les entoure.

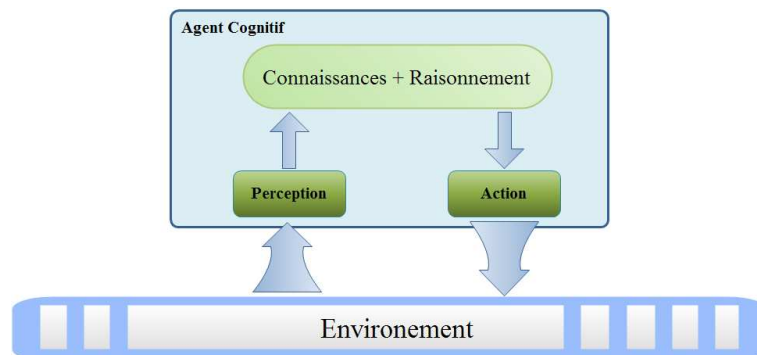


FIGURE 3.1 – Agent cognitif

3.2.2.2 L'architecture réactive

Cette approche alternative a été développée afin de pallier au problème d'acquisition de connaissances des agents délibératifs. Elle a été développée par Rodney Brooks l'un

des scientifiques les plus influents de ces dernières années dans le domaine des systèmes multi-agents. Brooks a proposé trois postulats qui ont guidé son travail [48] :

- Un comportement intelligent peut émerger sans une représentation explicite des connaissances.
- Un comportement intelligent peut émerger sans un système de raisonnement complexe.
- L'intelligence est une propriété émergeant des interactions entre des composants très simples.

L'architecture réactive est définie donc comme celle ne comprenant aucune sorte de modèle de connaissance et n'utilisant pas de raisonnement complexe. La complexité émergera de l'interaction des agents entre eux et avec l'environnement qui les entoure.

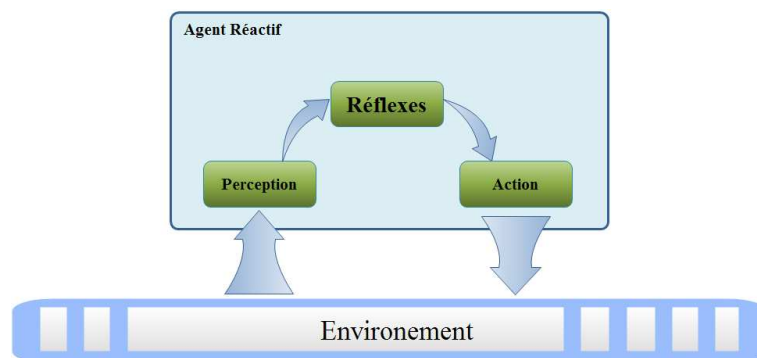


FIGURE 3.2 – Agent reactif

3.2.2.3 L'architecture hybride

L'architecture hybride essaye de combiner les deux architectures précédemment citées pour tirer les points forts de chacune d'entre elles. Cette architecture consiste en l'utilisation d'un système cognitif et d'un système réactif simultanément, le système réactif ayant une architecture simple peut réagir rapidement et peut être utilisé pour offrir des réponses urgentes, pendant que le système cognitif peut être utilisé pour pré-

voir des actions à plus ou moins long terme, ceci grâce à sa capacité de raisonnement et ses connaissances. Ces systèmes sont généralement utilisés sous forme d'architectures multi-couches, les couches de bas niveau utilisant des agents réactifs, et les couche de haut niveau utilisant un système cognitif plus complexe.

3.3 Les Systèmes Multi-Agents pour la segmentation d'images médicales

L'imagerie médicale est devenue de nos jours un outil presque indispensable aux différentes disciplines que compte la médecine moderne. Les IRM, Scanner et échographie sont différents supports qui aident les praticiens à établir et à affiner le diagnostic d'un grand nombre de pathologies. Étant victimes de leurs succès, les différentes cliniques et services d'imagerie médicale se retrouvent saturés et ne peuvent faire face à la demande de la société. C'est donc dans une logique de désengorgement de ces services et d'amélioration des délais de traitement et de diagnostic des pathologies que sont apparus les systèmes de traitement automatiques des images médicales et parmi eux figurent les systèmes de segmentation d'images. Les spécialistes en traitement automatique d'images en général, et en segmentation d'images en particulier commencent à explorer la piste des systèmes multi-agents pour résoudre les différents problèmes et contraintes auxquels ils doivent faire face. En effet la distributivité des tâches et la capacité de traitement d'une grande quantité de données dans un temps réduit fait de l'architecture multi-agents la candidate idéale pour l'implémentation de système de segmentation automatique d'images.

Généralement, dans le contexte du traitement d'images par ordinateur, l'approche agent a été associée à l'utilisation de calculs évolutifs. Les agents dans le contexte de tels systèmes peuvent utiliser, par exemple, les algorithmes génétiques, ils divisent ainsi entre eux le volume de données à traiter et (l'image par exemple) de telle sorte que chaque

agent opère sur son propre sous-ensemble des données indépendamment des autres. Les agents dans de tels systèmes n'ont généralement pas besoin de collaborer et de partager les résultats avec les autres. Après l'achèvement des travaux sur leurs sous-ensembles de données, les résultats sont ensuite agrégés par un composant du système, et présentés à l'utilisateur pour l'évaluation. Cette division d'une tâche de traitement d'images en sous-tâches indépendantes est souvent utile, mais ne fait pas entièrement usage de la capacité d'une approche multi-agents [49].

Une des variantes de cette approche est l'utilisation d'une hiérarchie d'agents, selon laquelle les agents au niveau le plus bas de la hiérarchie opèrent sur des pixels individuels, les agents au niveau suivant opèrent sur des régions (dont certains peuvent se chevaucher), jusqu'à ce qu'un agent au plus haut niveau de la hiérarchie combine les résultats. L'utilisation d'une telle hiérarchie est utile pour soutenir certains types d'activités de traitement d'images (comme la détection de certains types d'objets dans une image), mais ils sont difficiles à généraliser à d'autres types de résolution de problèmes. Dans ces approches hiérarchiques, les agents ont besoin de partager leurs résultats entre eux (principalement des agents plus haut dans la hiérarchie), afin de générer la réponse finale [49].

Un agent peut également entreprendre des tâches de bas niveau, telles que le traitement d'images couleur, la détection de teinte, la détection des contours, etc. Les résultats que fournissent ces agents d'analyse multiples peuvent ensuite être agrégés. Ainsi, un agent peut être amené à opérer au niveau du pixel sur une image, et au niveau d'une région sur une autre. Cette activité est équivalente à l'application d'un certain nombre de filtres pour une image. Les résultats de ces agents individuels peuvent aussi être combinés pour obtenir la réponse collective du filtre. Bien qu'utile, une telle approche n'utilise pas pleinement les avantages d'une approche basée sur des agents. Dans ce cas, chaque agent agit comme un filtre passif sur une image initiale, et donc ne fournit aucune valeur ajoutée autre que l'analyse statique de l'ensemble ou des parties de l'image

[49].

Ces dernières années ont donc vu l'apparition de plusieurs tentatives d'implémentation de systèmes multi-agents pour l'imagerie médicale. Nous essayerons donc d'en traiter quelques unes de ces implémentations dans ce qui suit.

3.3.1 Une approche biomimétique pour la segmentation d'images

L. Djerou et M. Batouche utilisent une approche biomimétique pour traiter les problèmes liés aux techniques de croissance de régions. Le système est basé donc sur une hybridation de deux techniques de l'intelligence en essaim (l'Optimisation par Essaim de Particules et la stigmergie qualitative). Il se compose d'un ensemble d'agents réactifs se déplacent sur l'image en construisant des régions homogènes ; par le regroupement des pixels connexes présentant une certaine similarité. L'OEP est une méthode d'optimisation faisant appel à une population d'agents appelés particules. Chaque particule est caractérisée par sa position et un vecteur de changement de position (appelée vitesse). En se déplaçant les particules sont influencées par leurs comportements passés et par les comportements de leurs voisins. L'OEP sera utilisée dans le système pour optimiser la recherche de pixels germes ainsi que pour estimer de façon locale les valeurs de seuils d'homogénéités aux différentes parties de l'image.

La stigmergie par contre [50] est un mécanisme de communication indirecte se servant de l'environnement comme médium. Un exemple de stigmergie très connu dans le domaine informatique est celui de l'algorithme des colonies de fourmis.

3.3.1.1 Spécification de l'approche

Les agents présents dans le système peuvent être divisés en deux populations distinctes, la population d'agents Détecteurs, liés au marquage de régions homogènes, et la population d'agents système permettant la gestion de l'environnement. Les agents Détecteurs sont des entités autonomes, mobiles qui possèdent des mémoires simples,

des capacités de calcul limitées et des perceptions locales de leur environnement, ils mettent en oeuvre des comportements simples comme Exploration, Agrégation, Fusion et Reproduction, en fonction des contraintes de l'environnement. Ces agents peuvent communiquer directement ou bien indirectement via l'environnement partagé [51].

Lors de la phase d'initialisation, les agents détecteurs se déplacent dans l'image à la recherche de positions idéales pour les points germes tout en estimant les valeurs locales des seuils d'homogénéité ceci en utilisant une variante de l'OEP [52]. Au terme de cette phase, les pixels germes seront marqués.

Après l'apparition de quelques régions marquées, les déplacements des agents Détecteurs sont guidés les champs de phéromones produit par les régions marquées

L'adéquation E_g d'un pixel à une région est déterminée par une somme pondérée des deux évaluations : E_1 , E_2 . L'évaluation E_1 a pour effet de favoriser une croissance homogène de la région marquée, en mesurant la différence entre le niveau de gris du pixel et le niveau de gris moyen de la région marquée. L'évaluation E_2 a pour effet de favoriser une croissance compacte de la région marquée, en mesurant le nombre de voisins du pixel appartenant déjà à la région marquée. Les pondérations sont fixées en fonction de l'importance que l'on attribut à chaque évaluation. Les évaluations fournissent des valeurs comprises entre 0 et 1. Donc, il apparaît dans l'environnement (image) plusieurs régions de propagation de phéromones[51].

$$E_g(p) = \alpha E_1(p) + \beta E_2(p) \quad (3.1)$$

3.3.1.2 Comportements des agents

3.3.1.2.1 Agent Système

Les agents système s'occupent de gérer la partie stigmergique du système, c'est-à-dire à la gestion des variations de concentrations des phéromones dans l'environnement. La dynamique de phéromones est caractérisée par les opérations : de propagation, d'évapo-

ration, de concentration et construction de champs du gradient, tous cela géré par les agents système. La région de propagation de la phéromone R_p dans l'environnement est donc, l'ensemble de pixels situés aux voisinages de la région marquée R_m et dont leurs évaluations globales suivant le critère d'homogénéité est au-dessus du $\delta 1$. La réduction et l'augmentation de la taille de la région de propagation d'une phéromone correspond respectivement à l'évaporation et la propagation de la phéromone dans le temps. La valeur de la concentration $S(p_i)$ de phéromone dans un pixel p_i est donc le rapport entre sa valeur d'évaluation globale du critère d'homogénéité $E_g(p)$ et la somme m et n du nombre de pixels respectivement dans la région marquée R_m et dans la région de propagation de phéromones R_p .

$$S(p_i) = \frac{E_g(p_i)}{m + n} \quad (3.2)$$

3.3.1.2.2 Agent Détecteurs

Les agents détecteurs sont initialisés aléatoirement dans l'image, ils ont chacun une couleur de marquage, un numéro de génération, une mémoire de la meilleure estimation locale du critère d'homogénéité $\delta 1$ et l'emplacement correspondant à cette estimation. Chaque agent détecteur peut prendre lors de la segmentation de l'image, l'un des comportements suivant :

1. Comportement Exploration

La phase d'exploration est en générale successive à la phase d'initialisation, et en l'absence de phéromones (dans sa région de perception), l'agent explore l'image à la recherche d'un germe (un pixel de départ). Il parcourt ainsi l'image en calculant le seuil d'homogénéité local est $\delta 1$. Après un certain nombre de changements N_c de sa valeur $\delta 1$, l'agent peut revenir à l'emplacement correspondant à la meilleure valeur d'estimation local $\delta 1$, et commence à marquer son emplacement.

2. Comportement Agrégation

L'agent peut suivre le comportement d'agrégation, si un des pixels de son en-

vironnement local est couvert par un des champs de phéromones (l'état de son environnement local est "TrouverPhéromone"). Dans ce cas, l'agent sélectionne son emplacement suivant l'attraction relative f_i , puis marque cet emplacement par sa propre couleur.

$$f_i = \frac{S_i}{\sum_{p_j \in vA(p)} S_j} \quad (3.3)$$

Avec : f_i : est l'attraction relative du pixel p_i , S_i : la concentration de phéromone dans le pixel p_i et $vA(p)$: l'environnement local de l'agent.

3. Comportement Fusion

L'agent peut suivre le comportement de fusion, si un des pixels de son environnement local est couvert, au moins, par deux champs de phéromones (l'état de son environnement local est PhéromonesSuperposées). Dans ce cas l'agent adapte sa couleur de marquage à la couleur de la région marquée produisant le champ de phéromone qui l'attire puis il fait un tour en arrière en suivant la trace de son ancien marquage, et il marque son emplacement par la nouvelle couleur de marquage.

4. Comportement Reproduction

L'agent suit le comportement de reproduction, si tous les pixels de son environnement local sont marqués et ne sont pas couverts par aucun champ de phéromones (l'état l'environnement local est SituationDeBlocage). Dans ce cas, l'agent crée un agent fils qui hérite la même couleur de marquage de son père et un numéro de génération incrémenté de " 1 " puis il meurt. Si le numéro de génération de l'agent est supérieur à la valeur Maxgénération, l'agent ne peut pas reproduire un descendant.

3.3.1.3 Discussion

Dans cet article, les auteurs n'ont pas procédé à une comparaison de leur approche avec les techniques existantes de segmentation d'images (multi-agents ou non), de ce

fait on ne peut juger de son efficacité ou de l'apport qu'elle pourrait offrir. L'approche proposée offre aussi très peu de tolérance au bruit, il ne dispose pas en outre, d'un mécanisme pour éliminer le bruit (zone de non intérêt) dans l'image segmentée.

3.3.2 Une architecture pyramidale agent pour la segmentation d'images : application à l'extraction d'une zone lobulaire issue d'une mammographie

Jacquelet et al. ont utilisé une approche basée sur une pyramide irrégulière adaptative d'agent pour la segmentation d'image. En effet, le système définit des agents région et des agents contour comme sommets d'un graphe d'adjacence et associe une accointance à chaque arc de ce graphe. Les accointances d'un agent représentent donc la liste de ses régions voisines et des contours voisins. Cette transposition contribue à l'augmentation du degré d'adaptabilité des sommets du graphe, conduisant à une gestion plus efficace des activités locales de traitement d'image, entre autre le choix des fusions [9].

3.3.2.1 Spécification de l'approche

La pyramide peut être vue comme un ensemble de traitements itératifs, lors de ces traitements, on procède à la fusion de sommets du graphe de niveau N pour obtenir un graphe de niveau $N + 1$. Définir la pyramide agent revient donc en premier lieu à créer une première population d'agents constituant ainsi la base de la pyramide. Il est nécessaire aussi de définir la manière dont le graphe évolue itérativement d'une étape N à une étape $N + 1$. Enfin, il est obligatoire de préciser la condition d'arrêt stabilisant la population d'agents. Le traitement pyramidal se divise en trois étapes clés :

1. L'image originale est fragmentée (arbre quaternaire) initialement en régions homogènes. Chaque région R_i est alors associée à un agent A_i , les agents de contours sont quant à eux issus d'une image de gradient.

2. Après la création du niveau initial, commence le traitement de l'image qui se traduit par l'interaction des agents en eux, ils adoptent ainsi différents comportements pour établir la liste des meilleures fusions de sommets possibles. Ces traitements génèrent un certain nombre de conflits (lorsque au moins deux agents désirent fusionner avec un troisième, une élection distribuée règle le problème en désignant parmi les agents en conflits ceux qui pourront fusionner ensemble). Au terme des fusions et résolutions de problèmes on passe au niveau suivant.
3. la condition d'arrêt sera en rapport avec le degré de similarité entre deux niveaux successifs, les traitements s'arrêteront donc lorsque la différence entre ces deux niveaux sera trop faible.

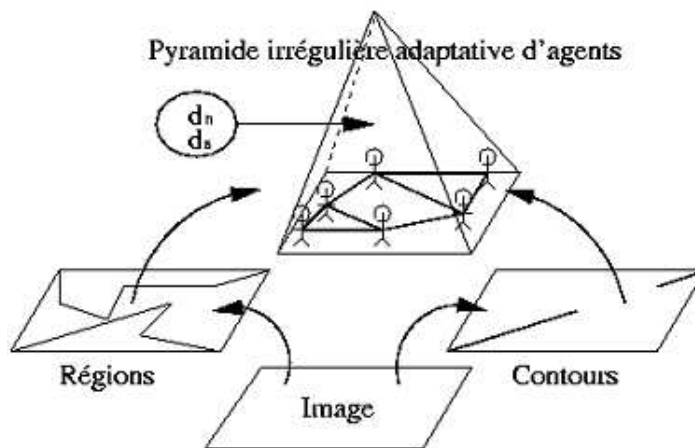


FIGURE 3.3 – Origine des agents (régions et contours) [9]

3.3.2.2 Comportements des agents

Dans le système, les sommets du graphe sont représentés par des agents qui interagissent entre eux pour définir les fusions à effectuer et passer ainsi à un niveau supérieur. Lors de ces traitements, l'agent adopte les comportements suivants [9] :

3.3.2.2.1 Exploration

L'exploration de l'agent se déroule principalement en deux étapes : en premier lieu l'agent procède à l'analyse locale de sa région pour en extraire le vecteur qui la caractérise, ensuite il passe à l'examen des affinités avec les régions voisines, ceci en transmettant son vecteur caractéristique et en recevant celui des transmis par les agents voisins. L'affinité inter-agent joue un rôle pivot car les agents se basent sur celle-ci pour établir leur décision de fusion.

3.3.2.2.2 Établissement du plan local de fusion

Dans ce comportement, l'agent établit des hypothèses de fusions avec chacun des agents présent dans sa liste de voisinage sur la base de similarités. Les agents voisins seront donc divisés en trois classes : "Régions pas suffisamment similaire", "Régions suffisamment similaire" et "Régions à délibérer". l'affinité (d) entre deux agent sera donc la différence entre leurs deux vecteurs caractéristiques. L'hypothèse de fusion sera donc rejetée si ($d < d_s$), incertaine si ($d_s < d < d_n$) ou acceptée si ($d_n < d$). L'ouverture $|d_n - d_s|$ représente en quelque sorte le réglage du niveau délibératif des agents. Une grande ouverture déterminera l'émergence de grandes régions, une petite ouverture l'apparition de nombreuses régions.

3.3.2.2.3 Affinage du plan local de fusions par collaboration

Le renforcement ou l'affaiblissement du désir de fusion entre agents est obtenu grâce aux critiques de leurs voisins (région ou contours). Ainsi, le désir d'un agent A_s de se réunir avec un agent A_i n'est pas forcément commutatif (désir $A_{si} \neq$ désir A_{is}). La critique n'est, en fait, qu'une appréciation scalaire portant sur une fusion incertaine. Chaque agent A_s ajoute à son désir initial de fusion pour A_i la somme algébrique des critiques des agents voisins et seuille cette valeur par rapport à d_n : $desir_{si} = desir_{si} + \sum_j c_{si}^j$. A cette étape du traitement, l'agent connaît parfaitement les hypothèses de fusions rejetées et acceptées. Ces hypothèses représentent le plan de fusions de notre

agent, un plan qui lui est propre.

3.3.2.2.4 Négociation et résolution des conflits

Après le calcul des plans locaux de fusions, vient la phase de résolution des conflits qui peuvent apparaître au sein de la population d'agents. Pour cela les auteurs ont eu recours à une variante de l'algorithme d'élection distribué de Meer [22], où la probabilité qu'un agent soit un maxima local dans le réseau de conflit est proportionnelle à son désir de fusion avec les autres agents. Ce traitement se divise en deux parties : la coopération agents région-région (si pour deux régions voisines l'une apparaît dans le plan de fusion de l'autre mais l'inverse n'est pas vrai, cela diminuera la probabilité de fusion de ces deux régions) et la coopération agents région-contours (un agent contour se trouve entre deux agents régions sur le point de fusionner diminue grandement la probabilité de fusion). Après cela, les fusions seront effectuées, et on passera ainsi d'un niveau (K) à un niveau ($K+1$) de la pyramide agent.

3.3.2.3 Discussion

Dans cet article, le processus de segmentation initial de l'image n'a pas été clairement défini, la manière dont l'image est divisé pour obtenir la première liste de régions qui seront plus tard affectées aux agents semble négligée bien qu'étant une étape cruciale dans le processus de segmentation. D'autre part, le réglage du niveau délibératif des agents reste manuel et doit faire l'objet d'ajustements pour chaque image traitée.

3.3.3 Segmentation 3d multi-objets d'images de scanner cardiaques : une approche multi-agents

J. Fleureau et al. proposent un système multi-agents de segmentation d'images 3D appliqué à l'extraction de structures cardiaques en imagerie scanner multi-barrettes, cette approche est caractérisée par un couplage de processus de croissance de régions

3D avec un apprentissage supervisé au sein d'un système multi-agents, où les propriétés de coopération et de compétition sont pleinement exploitées pour contraindre et contrôler l'extraction parallèle d'objets connexes dans le volume traité [10].

L'architecture élaborée repose sur la centralisation des opérations autour d'un agent contrôleur appelé aussi agent communiquant, ayant pour rôle de contrôler une population d'agents travailleur situés dans l'image. La segmentation de l'image émergera de la coopération et de la compétition des agents travailleurs qui parcourent l'image à la recherche de points appartenant à leurs zones respectives.

3.3.3.1 Spécification de l'approche

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. L'utilisateur suit dans son choix d'objet les contraintes suivantes :

- L'homogénéité de l'objet en termes d'intensité et de texture ;
- Une connexité spatiale 3D de telle sorte qu'un organe est considéré comme composé de sous-parties connexes et fermées ;
- Une similitude de niveaux de gris et de texture au sein de l'objet.

Un germe sélectionné par l'utilisateur sera le point de départ d'un agent travailleur, l'utilisateur affectera donc à chaque agent travailleur un germe (position initiale de l'agent dans l'image) ainsi qu'une étiquette anatomique représentant l'objet identifié à segmenter. Plusieurs agents avec la même étiquette anatomique peuvent coexister, leur but étant d'associer leur étiquette commune à une région de l'image.

Après cela commence le processus d'apprentissage réalisé dans un premier temps sur les informations de texture et d'intensité concernant l'objet auquel les agents sont associés. Chaque agent placé au sein de l'image et associé à un objet, analyse son voisinage pour extraire ses informations caractéristiques (textures et intensités) qu'il utilisera

comme référentiel lors de son parcours de l'image.

Au terme de la période d'apprentissage, chaque agent parcourt l'image à segmenter depuis le germe auquel il a été assigné, en se déplaçant dans son voisinage avec pour contrainte de ne suivre que les voxels (points de l'image) voisins qui satisfont le critère d'homogénéité (intensité et texture) identifié lors de la période d'apprentissage. L'agent émet des requêtes pour assigner à l'étiquette à laquelle il appartient, les voxels rencontrés lors de son parcours de l'image.

L'agent contrôleur est l'entité chargée de traiter les requêtes des agents travailleurs et de l'affectation des voxels suivant un certain nombre de contraintes. La segmentation de l'image résulte donc de l'interaction des agents qui peuvent être en compétition ou en coopération pour l'affectation d'un voxel, tous cela sous la surveillance de l'agent contrôleur.

3.3.3.2 Comportements des agents

3.3.3.2.1 Les agents travailleurs

Les agents travailleurs sont des entités créés au sien de l'image coopérant et entrant en compétition de manière à, par croissance de région, maximiser leurs territoires (segment de l'image) selon la vérification de différents critères (texture et d'intensité). Le comportement de ces agents passe par deux étapes essentielles :

1. Après avoir été assigné à un germe et à une étiquette, l'agent travailleur commence l'apprentissage de l'environnement qui l'entoure pour extraire les informations de texture et d'intensité qui caractérisent l'objet auquel il est assigné, et pourra par la suite utiliser ces informations lors de l'étape suivante.
2. À la fin de l'étape d'apprentissage commence l'étape de segmentation de l'image, en effet chaque agent travailleur parcourt l'image à la recherche de voxels appartenant à l'objet auquel il a été assigné. Chaque agent gère son déplacement en utilisant l'algorithme de Dijkstra et ne parcourt que les pixels qui satisfont leurs

critères de comparaison. À l'identification d'un voxel pouvant faire partie de l'objet auquel l'agent travailleur est assigné, il dépose une requête d'acquisition auprès de l'agent contrôleur qui débouchera vers un ajout simple, une coopération ou une compétition sur le voxel.

3.3.3.2 Les agents Inhibiteurs

C'est dans le but d'éviter les déplacements des agents travailleurs sur des zones ne faisant pas partie de l'objet auquel ils ont été assignés, que la notion d'agents inhibiteurs fut introduite dans le système. Ces agents sont en faits de simples agents travailleurs mais assigné non pas à un objet d'intérêt mais à une structure de non intérêt qui à cause de sa proximité ou de sa similarité avec un objet pourrai faussait le résultat de la segmentation. La compétition de ces agents inhibiteur avec les agents travailleurs permet d'éviter le débordement des zones d'objets d'intérêt, améliorent ainsi le résultat de la segmentation.

3.3.3.3 L'agent contrôleur

L'agent contrôleur (appelé aussi agent communiquant) est le coordonnateur du système, ayant pour rôle principale l'attribution ou non d'un voxel à l'agent ayant émit une requête pour son acquisition. L'agent contrôleur lors de son processus décisionnel fait appel à deux cartes, « la carte de segmentation » qui est une copie étiquetée de l'image originale et une « carte de voyage » qui enregistre, pour chaque voxel, la longueur du chemin parcouru par l'agent qui a acquis ce voxel. Il utilise ainsi ces deux cartes pour répondre aux différentes requêtes de la manière suivante :

- Acquisition Simple : si le voxel dont l'agent travailleur a émis une requête d'acquisition n'appartient à aucun autre agent, l'agent contrôleur accepte l'acquisition. Il devra ensuite mettre à jour les deux cartes (segmentation et voyage) et autorisera ainsi l'agent travailleur à poursuivre son parcourt.
- Coopération : si le voxel dont l'agent travailleur a émis une requête d'acquisition

appartient déjà à un autre agent du même groupe (même étiquette), le contrôleur compare dès lors, les distances parcourues par les deux agents pour atteindre le voxel, l'agent choisi étant celui ayant parcouru le plus court chemin. Il affectera donc le voxel à ce dernier et mettra à jour les deux cartes de segmentation et de voyage.

- **Compétition** : si le voxel dont l'agent travailleur a émis une requête d'acquisition appartient déjà à un autre agent appartenant à un autre groupe, le contrôleur choisit l'agent ayant parcouru le plus court chemin pour atteindre le voxel.

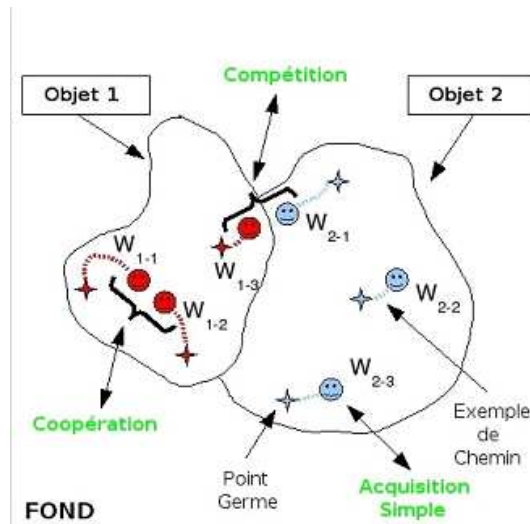


FIGURE 3.4 – Comportement des agents dans le système [10]

3.3.3.3 Discussion

Bien que d'après des résultats positifs montrés par l'expérimentation, l'approche proposée reste spécifique à un type bien précis d'images (cardiaques 3D) et n'as pas été testée sur d'autres type d'image 3D. Cette approche a aussi un autre inconvénient qui est de faire appel à des connaissances anatomiques poussées de la part de l'utilisateur, ceci lors du positionnement des points germes pendant l'initialisation du traitement.

3.3.4 Application de l'apprentissage par renforcement à la segmentation d'images

Farhang Sahba et al. présente une méthode de segmentation d'image utilisant un système multi-agents basé sur l'apprentissage par renforcement. Les auteurs ont utilisé une approche multi-agents pour optimiser la détection des valeurs locale des seuils pour la segmentation de l'image. Un agent utilise une image et sa segmentation manuelle pour réaliser un apprentissage. Il réalise, en effet, des actions sur l'image initiale et compare le résultat avec l'image manuellement segmentée. Suivant le résultat obtenu, l'agent fera l'objet d'une récompense ou d'une punition. Les résultats obtenus de l'apprentissage pourront ensuite être utilisés lors de segmentations d'images du même type [11].

3.3.4.1 Spécification de l'approche

Dans l'approche proposée, l'image est divisée en un sous ensemble d'images de taille $R_s \times C_s$ (R_s lignes et C_s colonnes) sur lesquelles l'agent d'apprentissage travaillera séparément. L'agent d'apprentissage aura pour objectif de déterminer le seuillage dans chaque image partielle ainsi que les différents objets s'y trouvant. Lors de l'utilisation de systèmes basés sur les agents d'apprentissage par renforcement, un grand nombre de paires (état-action) sont traités ce qui peut amener à l'apparition de deux problèmes qui sont les suivant :

- Une utilisation excessive de la mémoire.
- Un temps de traitement trop longs pour le parcours de toutes les paires (état-action).

Pour remédier à cela, l'approche a recours à l'apprentissage basé sur l'opposition. L'apprentissage correspond plus exactement en l'élaboration d'une matrice (appelée Q-matrice dans l'apprentissage par renforcement) ayant pour lignes l'ensemble des états d'une image, et pour colonnes l'ensemble des actions que l'agent peut appliquées sur

cette image. Les différentes valeurs qui rempliront la matrice seront quand à elles les différentes récompenses que recevra l'agent pour les différentes actions qu'il appliquera aux différents états. Il est donc essentiel de définir ces trois composantes :

- Les états : un état est un ensemble de paramètres devant refléter la qualité de l'image à traiter, cet ensemble de paramètre est étroitement lié à la nature du problème à résoudre. Pour l'approche étudié l'état de l'image partielle est défini par un ensemble de caractéristique qui sont les suivante : la surface du plus grand objet appartenant à l'image, sa compacité, les coordonnées relative de l'image partielle et le nombre d'objets contenus dans cette image.
- Les actions : l'ensemble des actions utiliser dans cette approche est le changement des seuils de l'image ainsi que de la taille des éléments qui la compose.
- Les récompenses/punitions : les récompenses et punitions peuvent être considérés comme des critères quantifiant la qualité de segmentation de l'image après l'application d'une action. Dans l'approche étudiée, les auteurs on choisit la différence de qualité entre une image avant et après sa segmentation comme quantificateur. La fonction de récompense sera donc défini comme suit :

$$reward = \begin{cases} \epsilon_1 D_{\Delta} & si \quad D_{\Delta} \geq 0 \\ \epsilon_2 D_{\Delta} & si \quad D_{\Delta} \leq 0 \end{cases} \quad (3.4)$$

3.3.4.2 Comportements des agents

L'agent d'apprentissage commence par le traitement d'une image d'apprentissage (une image échographique dans le cas présent) et sa version manuelle segmenté. L'image est divisée en un sous ensemble d'images partielles qu'il traitera une à une. Il explore ainsi l'espace des solutions possibles pour chaque image partielle, utilisant comme référence la version manuellement segmentée de celle-ci. L'agent d'apprentissage change l'état de l'image en modifier les seuils de cette dernière ou en modifiant la taille des objets la composant, il reçoit ainsi une récompense ou une punition pour chaque ac-

tion effectuée, il met alors à jour la valeur de la Q-matrice correspondant à cette paire (état-action). L'agent d'apprentissage explore ainsi un grand nombre d'actions et essaye d'exploiter celle qui offre le plus de récompenses. Cette méthode est particulièrement efficace lorsque le traitement s'effectue sur plusieurs images ayant intrinsèquement les mêmes caractéristiques, ce qui est le cas dans le domaine de l'imagerie médicale.

L'apprentissage par opposition est utilisé par l'agent pour limiter au maximum le temps nécessaire à son apprentissage. En effet, pour chaque état E recevant une action A l'agent reçoit une récompense R , on peut donc logiquement déduire que l'application de l'action \bar{A} (l'inverse de A) sur l'état E conduit à une récompense \bar{R} (l'inverse de R). De la même manière l'application de l'action A sur l'état \bar{E} (inverse de E) conduit à une récompense \bar{R} (l'inverse de R)...etc. Cette méthode permet le remplissage des valeurs de la Q-matrice dans des délais réduits mais elle a comme principale inconvénient de choisir un ensemble d'états et un ensemble d'actions qui soient symétriques (pour chaque état E il existe un opposé \bar{E} et pour chaque action A il existe une action opposée \bar{A}).

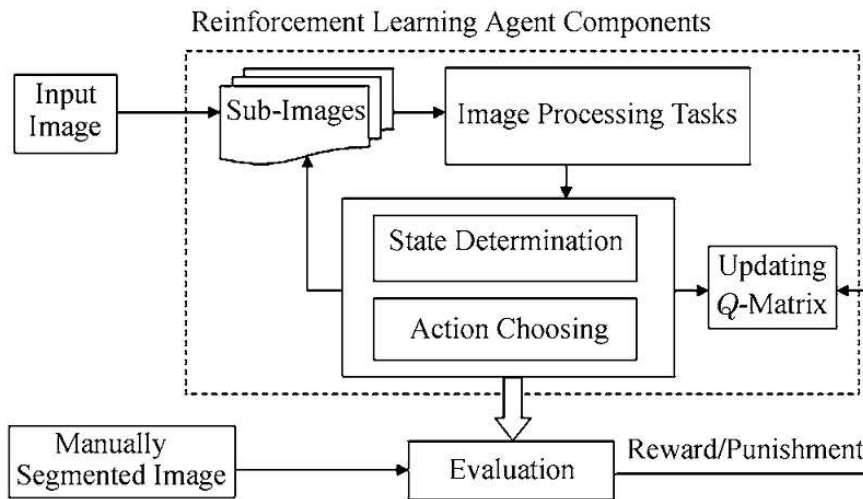


FIGURE 3.5 – Comportement de l'agent d'apprentissage dans le système [11]

3.3.4.3 Discussion

Bien qu'intéressant, l'approche proposée dans cet article tient plus du domaine du parallélisme que de celui des systèmes multi-agents. Le système repose en effet sur un seul type d'agents (les agents d'apprentissage) et il n'existe aucune interaction entre ces agents tout au long du processus de segmentation d'image, ni interaction directe, ni interaction indirecte (par l'intermédiaire de l'environnement). L'approche n'utilise ainsi aucune sorte de communication entre agents qui est pourtant l'un des principaux intérêts d'utiliser un système multi-agent.

3.3.5 Une approche distribuée basée sur la logique flou et les algorithmes génétiques pour la segmentation d'images

K. E. Melkemi and S. Foufou présentent un nouvel algorithme de segmentation image (appelé FDGA-Seg) basé sur une combinaison de la logique floue, des systèmes multi-agents et des algorithmes génétiques. Ils y proposent d'utiliser une représentation floue des images en introduisant une certaine imprécision dans les valeurs des tons de gris. La distributivité de l'algorithme FDGA-Seg vient du fait qu'il est conçu autour d'un système multi-agents (SMA) utilisant deux architectures différentes (l'architecture maître-esclave et l'Island model) [12].

3.3.5.1 Spécification de l'approche

Dans le modèle maître-esclave, le système multi-agents est composé d'un ensemble d'agents de segmentation (esclaves) connectés à un agent coordinateur (maître). Pendant la phase d'initialisation, chaque agent crée un ensemble flou d'images obtenu à partir de l'image originale à l'aide des K-moyennes et d'une fonction d'appartenance chaotique. Le comportement de chaque agent dépend de ses propres données initiales. Pendant le cycle de l'évolution dans l'algorithme génétique, chaque agent effectue une classification

à l'aide de l'algorithme ICM (Iterated Conditional Modes) sur sa propre image initiale, puis transmet son image initial, sa version segmentée et la valeur sélective (fitness value) à l'agent coordinateur qui sélectionne et enregistre la meilleure segmentation, effectue les opérateurs génétiques, retransmet ensuite les nouvelles images initiales à tous les agents de segmentation pour un autre cycle de segmentation.

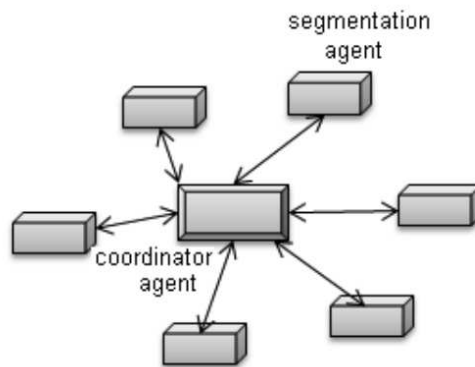


FIGURE 3.6 – Communication des agents dans l'architecture maitre-esclave [12]

L'Island Model est quant à lui un moyen efficace pour mettre en oeuvre les algorithmes génétiques et améliorer leurs comportements. Dans ce modèle, la population de l'algorithme génétique est en cluster, dans un ensemble de sous-populations appelées îlots. Les différentes îles maintiennent un certain degré d'indépendance. Les agents y explorent différents espaces de recherche, et partagent leurs informations par l'intermédiaire des opérateurs de la migration. L'algorithme FDGA-Seg propose de combiner les concepts de logique floue et l'Island model basé sur les systèmes multi-agents.

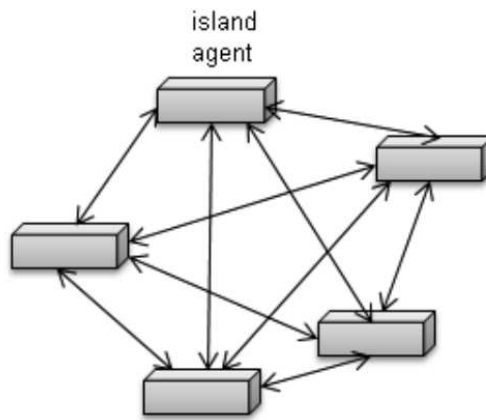


FIGURE 3.7 – Communication des agents dans l'architecture Island model [12]

3.3.5.2 Comportements des agents

3.3.5.2.1 l'architecture maitre-esclave

L'architecture maitre-esclave se compose d'un ensemble d'agents de segmentation (esclaves) connectés à un agent coordonnateur. Le FDGA-Seg utilise cette architecture pour déterminer la meilleure segmentation possible en utilisant le croisement génétique d'une population d'images sur plusieurs générations. Le comportement des agents dans cette architecture est le suivant :

1. Phase d'initialisation

- Appliquer l'algorithme des K-Moyennes sur l'image initiale.
- Appliquer l'algorithme de logique flou pour altérer l'image obtenu à l'issue de l'application des K-Moyennes.
- Éclaircir l'image altérer pour pouvoir commencer le traitement.

2. Cycle d'évolution

- Chaque agent de segmentation effectue l'ICM (Iterated Conditional Modes) sur sa propre image initiale, il retourne ensuite l'image segmentée, l'image initiale ainsi que la valeur de fitness à l'agent coordonnateur

- L'agent coordonnateur
 - * Reçoit les messages des agents de segmentation.
 - * Enregistre la meilleure segmentation ainsi que sa valeur de fitness.
 - * Effectue les croisements et mutations et retransmet les images descendantes aux agents de segmentation.
- 3. Les étapes 2 et 3 se répètent jusqu'à stabilisation du système

3.3.5.2.2 L'architecture basée sur l'Island Model

Dans ce modèle, la population est considérée comme un ensemble flou et les îles comme un sous ensemble flous. L'utilisation de la logique floue permet la diversification des données d'initialisation et offre ainsi un plus grand domaine de résolution. Le comportement des agents dans cette architecture est le suivant :

1. Lors de la phase d'initialisation chaque agent île
 - Applique l'algorithme des K-Moyennes sur l'image initiale.
 - Applique l'algorithme de logique floue pour altérer l'image obtenu à l'issue de l'application des K-Moyennes.
 - Éclaircit l'image altérée pour pouvoir commencer le traitement.
 - Applique l'ICM sur l'image résultant des traitements précédents
 - Transmet l'image issue des traitements aux autres agents îles et définit ainsi la sous population initiale.
2. Lors du cycle d'évolution chaque agent île
 - Reçoit des individus provenant des autres agents îles.
 - Applique l'algorithme génétique sur les sous population
 - Applique l'ICM sur la meilleure descendance
 - Met à jour la meilleure segmentation et la valeur de fitness

3.3.5.3 Discussion

Dans l'approche présentée, la communication entre les différents agents et le volume immense de donnée échanger lors des traitements peut être un grand inconvénient en termes de performances pour le système. Le fait que chaque agent esclave doit retransmettre l'image initiale, l'image segmentée ainsi que la valeur de fitness à l'agent maître ; et que celui-ci retransmet la meilleure segmentation à tous les esclaves, tout cela rend immense la quantité de données transférées lors du traitement. Cet inconvénient peut être aussi un frein à une éventuelle implémentation de l'approche sur une architecture distribuée.

3.3.6 Un nouveau mécanisme d'essaim basé sur les colonies d'araignées sociales pour la segmentation d'image

Christine Bourjot et al. proposent dans leur article un système multi-agents pour la segmentation d'images à niveaux de gris, basé sur un mécanisme d'essaim, inspiré de l'activité collective de tissage de certaines araignées sociales. Ces araignées coopèrent pour le tissage de leur toile qui peut atteindre jusqu'à 10 m^3 , ainsi que pour d'autres activités comme la chasse ou la maintenance de cette toile. Christine Bourjot et al. proposent donc un système multi-agents où l'environnement (l'image) est modélisé sous forme d'une grille carrée, chacune des cellules de cette grille (pixel) représente un piquet. Les piquets peuvent avoir différentes hauteurs (le niveau de gris du pixel) et leurs sommets peuvent être reliés par des fils de soie. Les agents se déplacent dans l'environnement et tissent une toile sur une région qu'il considère comme homogène. L'ensemble des toiles résultant à la fin du processus d'exploration représentera la partition de l'image. [13]

3.3.6.1 Spécification de l'approche

L'approche proposée se base sur le tissage de toiles par les agents pour l'extraction des régions de l'image ayant un niveau de gris similaire. Les agents (araignées) sont divisés en plusieurs groupes de manière à ce que les membres d'un même groupe partagent des paramètres communs qui caractérisent la région qu'ils doivent détecter. Ils se déplacent ainsi dans l'image en fixant leurs fils de soie sur les pixels d'intérêt.

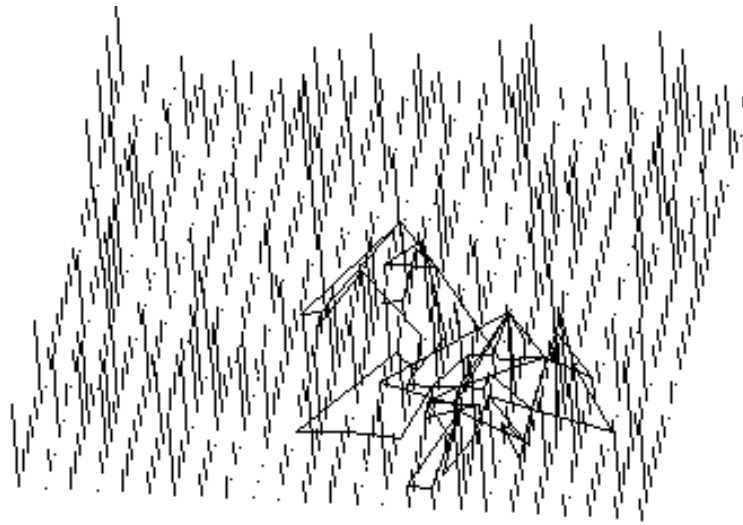


FIGURE 3.8 – Modèle de l'environnement des agents [13]

L'environnement est donc modélisé comme étant une grille de pixels, chaque pixel étant caractérisé par son niveau de gris, ainsi que par la liste des fils de soie fixés sur ce pixel. Chaque fil de soie est quant à lui caractérisé par son pixel de fin et par le label du groupe auquel l'agent qu'il l'a tissé appartient. Les agents se déplacent dans l'environnement et choisissent l'action à réaliser en utilisant un système de probabilités basé sur des valeurs à initialiser. Chaque agent embarque un certain nombre de caractéristiques qui spécifient son comportement ainsi que son état. Ces caractéristiques peuvent être triées selon leurs influences sur le comportement des agents en deux catégories :

- Influence sur le comportement d'exploration : les paramètres concernés sont le rayon perception de l'agent (appelé R) qui définit les pixels considérés comme voisins de la position de l'agent, sa tendance à retourner sur sa toile (appelé *Back-Probability*), l'attraction de sa soie et l'attraction de la toile des autres groupes.
- Influence sur le comportement de tissage de la toile : les paramètres concernés sont le niveau de gris de la région à détecter (appelé *RefLevel*) et la tolérance de sélection des pixels (appelée *Selectivity*).

L'état de l'agent est quant à lui caractérisé par sa position (appelé *CurrentP*) ainsi que par la position du dernier pixel de fixation du fil de soie (appelé *Last Fixed*).

Au début d'une exécution du système, l'environnement est dénué de fil de soie. Les agents sont alors placés au sien de cet environnement et leurs caractéristiques sont initialisées avec les paramètres propres aux régions souhaitant être détectées (initialisation des *RefLevel* des agents et de la tolérance), la tendance à retourner sur sa toile, l'attraction de sa soie ainsi que l'attraction des toiles des autres groupes sont quant à elle les mêmes pour tous les agents. Le système évolue par cycles. Dans chacun d'eux, chaque agent est activé et sélectionne une des trois actions (mouvement, fixer le fil de soie ou le retour sur la toile) et ceci selon un système de probabilités. L'exécution se termine après un nombre de cycles déterminé par l'utilisateur.

3.3.6.2 Comportements des agents

Le comportement des agents est géré par un système probabiliste, la probabilité que l'agent exécute une action donnée peut être calculée selon un certain nombre de caractéristiques (action du mouvement par exemple), ou être considéré comme une valeur constante, choisie à l'initialisation du système. L'agent pourra donc effectuer l'une des actions suivantes, lors de ses différents cycles d'exécution :

3.3.6.2.1 Le mouvement

Le mouvement des agents est influencé par les fils de soie qui exercent une attraction sur ces agents, cette attraction est implémentée en utilisant un système de pondération des positions accessibles par un agent de manière à ce que tous les pixels adjacents à la position actuelle de l'agent soient de même poids, les positions accessibles par le biais d'un ou plusieurs fils de soie sont pondérés suivant le nombre de fils menant à cette position, suivant aussi la nature du fil (appartenant au même groupe que l'agent ou à un autre groupe). Après normalisation, on obtient une distribution de probabilités, La position ayant donc la plus grande probabilité d'être atteinte par l'agent sera donc choisie comme nouvelle position.

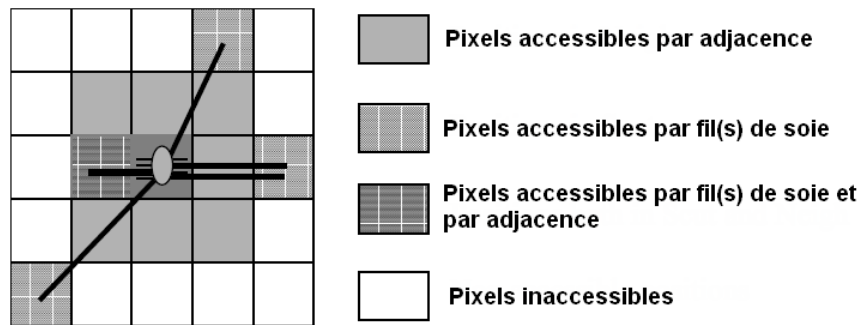


FIGURE 3.9 – Exemple de possibilité de déplacement d'un agent à un instant t [13]

3.3.6.2.2 La fixation du fil de soie

L'agent décide de fixer ou non le fil de soie sur sa position courante en s'appuyant sur une probabilité, calculé à partir d'une distribution gaussienne ayant pour moyenne $RefLevel$ et dont l'écart type est $1/Selectivity$. L'action de fixation aura pour effet de mettre à jour la liste des fils de soie du pixel courant ainsi que celle du dernier pixel où le fil a été fixé. La position actuelle est elle aussi enregistrée comme position du dernier point de fixation du fil de soie.

3.3.6.2.3 Le retour sur la toile

La probabilité de cette action est constante (`BackProbability`), et consiste à remplacer la position actuelle de l'agent par celle du dernier point de fixation du fil de soie.

3.3.6.3 Discussion

Bien qu'étant très intéressante, l'approche proposée reste néanmoins très peu performante par rapport aux techniques existantes, ceci à cause du grand nombre de paramètres à initialiser au lancement du système (`RefLevel`, `BackProbability`, `Selectivity...etc.`). L'extraction des régions n'est généralement que partielle et l'utilisateur doit aussi choisir le nombre d'agents ainsi que leurs positions sur l'image. Tous ces inconvénients rendent cette approche non utilisable dans des applications réelles, mais elle reste néanmoins prometteuse.

3.4 Conclusion

Dans ce chapitre, nous avons présenté les systèmes multi-agents qui sont des systèmes composés d'un ensemble d'entités appelées agent, en interaction pour la réalisation d'une tâche déterminée et qui peuvent être cognitifs ou réactifs. Nous avons ensuite fait un tour d'horizon des différentes approches multi-agents pour la segmentation d'images. Ces approches plus différentes les unes des autres, se basent sur des mécanismes très distincts, certains empruntent des comportements présents dans la nature, comme cela est le cas par exemple pour le système d'araignées sociales, là où d'autres sont quant à eux basés sur de l'apprentissage.

Bien qu'étant très différents, les systèmes présentés dans ce chapitre présentent la propriété commune d'être spécifiques à des types particuliers d'images et ont beaucoup de difficultés à généraliser leurs approches à plusieurs types d'images. Grâce à ce chapitre nous avons amélioré notre perception de ce que pouvait être un système multi-agents

pour la segmentation d'images. Cela a permis de contribuer à la réalisation de notre propre approche présentée dans le chapitre suivant.

Chapitre 4

Un SMA pour la segmentation d'images médicales

4.1 Introduction

L'objectif de ce chapitre est de décrire une approche pour la segmentation d'images, basée sur un système multi-agent. Ce système sera constitué d'un ensemble d'agents réactifs situés, ayant pour environnement l'image à segmenter et utilisant l'accroissement de régions pour déterminer les différentes zones homogènes composant l'image. Le système exploitera les propriétés de coopération et de concurrence entre agents pour aboutir à une segmentation collective de l'image en un ensemble de régions et de contours.

Lors de l'exécution, les agents composant le système auront pour objectifs d'élargir le territoire (région) auquel ils sont associés. Cette région sera initialement composée d'un seul pixel appelé « Germe » et sera au fur et à mesure étendue par l'agent aux pixels de voisinage satisfaisant le critère d'appartenance à la région. Cette élargissement pourra aussi s'exécuter par l'intermédiaire d'opération de fusions avec d'autres régions, ou par réaffectation de pixels déjà assimilés par une région connexes associée à un autre agent.

Nous entamerons donc ce chapitre par une description de la dynamique du système et son évolution lors du processus de segmentation de l'image. Nous présenterons ensuite le comportement des agents dans le système ainsi que les différentes interactions entre ces derniers à l'intérieur du système.

4.2 Spécification de l'approche

Au démarrage du système, les agents sont dispersés de manière aléatoire dans leur environnement, qui n'est autre que l'image à segmenter. Chaque agent parcourt dès lors l'image, à la recherche d'un pixel qui sera le point de départ d'une nouvelle région. Un pixel ne pourra donc être considéré par un agent comme germe possible d'une région, que si son intensité est comprise entre la moyenne des intensités de ses voisins plus ou moins leurs écart-type, cela afin d'augmenter les probabilités d'appartenance du pixel à une zone homogène de l'image.

Une fois positionné, l'agent crée une nouvelle région en phase d'initialisation, contenant le pixel courant et ayant comme contour l'ensemble des pixels voisins à ce dernier. L'agent entame alors la croissance de sa région. Il assimile ainsi, les pixels de contour satisfaisant le critère d'appartenance à la région et recalcule ensuite ce contour. Le critère d'appartenance utilisé par l'agent lors de la phase d'initialisation diffère de celui utilisé lors de la croissance d'une région mature. Cette différence a pour but de préparer la région à la phase suivante de croissance et sera détaillée dans la suite de la section.

Chaque agent aura aussi la possibilité de fusionner sa région courante avec une région adjacente à la sienne. Cette fusion ne sera possible que si la région résultante sera moins hétérogène que la sienne. Le cas échéant, tous les pixels seront assimilés par la région courante de l'agent et le contour sera par la suite recalculé.

L'accroissement parallèle des régions opéré par les agents présents sur l'image, donne lieu à l'apparition d'un comportement de concurrence en ce qui concerne l'assimilation

des pixels. De ce fait, un pixel déjà assimilé par un agent au sien de sa région courante, peut par la suite, être réaffecté par un autre agent et ainsi assimilé à une autre région.

Lorsque l'agent n'aura plus de possibilité d'accroître sa région, que ce soit par ajout de pixels non assimilés, par fusion ou par réaffectation à partir de pixels d'autres régions, deux scénarios sont alors possibles. Le premier scénario est que la région associée à l'agent est encore en cours d'initialisation, c'est-à-dire, qu'elle n'a pas atteint la taille minimale de confiance. Dans ce cas précis, l'agent considère que la taille de sa région n'est pas suffisante pour être prise en compte. Il supprime la région et réaffecte l'étiquette de non marqué à tous les pixels qui lui appartenaient et ceux qui composaient son contour. Le deuxième scénario est que la région à laquelle l'agent est associé n'est plus en phase d'initialisation. Cela voudra dire que la taille de la région est supérieure ou égale à la taille minimale de région, l'agent opérera alors une opération de suppression du bruit éventuellement présent au sien de cette région et qui sera décrite plus loin dans la section, avant de l'enregistrer dans la liste des régions détectées.

Au terme des deux scénarios précédemment cités, l'agent repart à la recherche d'un nouveau pixel germe qui sera le point de départ d'une nouvelle région, cela, tant que tous les pixels de l'image ne seront pas tous marqués ou que le nombre maximum de cycles d'exécution ne soit pas atteint, dans le cas contraire, les différents agents stoppent leur exécution jusqu'à l'arrêt total du système.

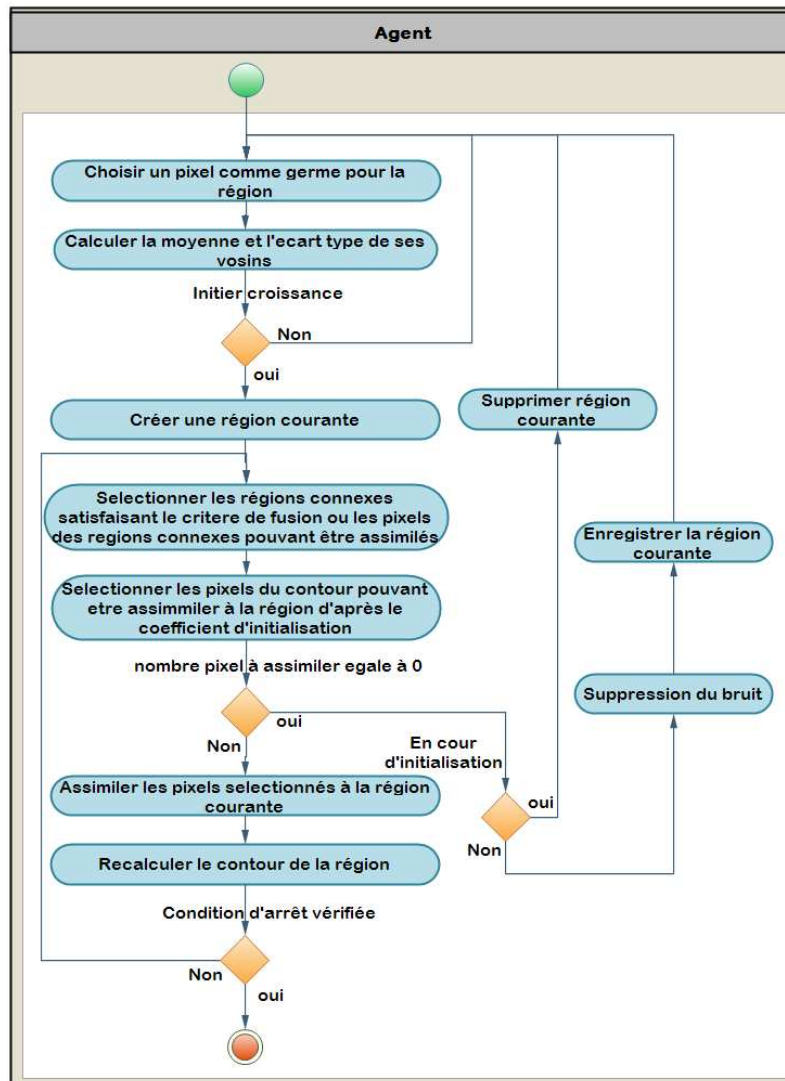


FIGURE 4.1 – Diagramme d'activité d'un agent

4.3 Comportement des agents

Après sa création, l'agent se déplace aléatoirement dans son environnement qui est l'image, à la recherche d'un pixel pouvant faire office de point de départ d'une région.

Le choix de ce pixel se fait en fonction de son homogénéité avec ses voisins, nous allons donc introduire d'abord quelques notions sur les pixels et le voisinage des pixels.

Une image en informatique, se compose d'un ensemble de pixels caractérisés par un couple de coordonnées (x, y) et où à chacun de ces pixels correspond une intensité que nous noterons G . Nous pouvons donc calculer des distances entre pixels en utilisant leurs coordonnées. Parmi ces distances on trouve la distance de Manathan, la distance Euclidienne ainsi que la distance d'Echiquier qui nous intéresse plus particulièrement. Soit deux pixels $P_1(x_1, y_1)$ et $P_2(x_2, y_2)$. La distance d'Echiquier $d(p_1, p_2)$ est calculée de la manière suivante :

$$d(P_1, P_2) = \text{Max}(|x_1 - x_2|, |y_1 - y_2|) \quad (4.1)$$

A l'aide de cette distance, nous pouvant maintenant introduire la notion de voisinage d'ordre k d'un pixel et qui est définie comme suit :

$$V_k(P) = \{P_i \text{ telle que : } 0 < d(P, P_i) < k\} \quad (4.2)$$

Dans l'approche proposée, nous utilisons le voisinage d'ordre "1" pour la détermination de l'ensemble des voisins d'un pixel. La notion de voisins d'un pixel P dans les sections suivantes fera donc référence à l'ensemble des pixels P_i ayant une distance d'Echiquier de P , égale à 1.

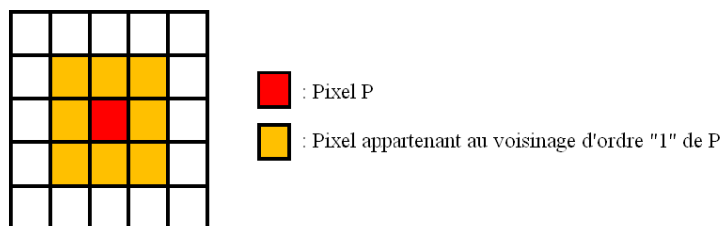


FIGURE 4.2 – Le voisinage d'ordre "1" d'un pixel

Lors de la phase de recherche d'un pixel germe, l'agent se déplace de pixel en pixel et à chaque pixel P exploré, il réalise les opérations suivantes :

1. Déterminer l'ensemble des pixels voisins de P .
2. Calculer l'intensité moyenne des pixels voisins de P tel que :

$$E_{V(P)}[G] = \frac{\sum_{P_i \in V(P)} G_{P_i}}{\text{Card}(V(P))} \quad (4.3)$$

avec $\text{Card}(V(P))$ étant le cardinal de l'ensemble des voisins de P .

3. Calculer l'écart type des pixels voisins de P tel que :

$$\sigma_{V(P)} = \sqrt{E_{V(P)}[G^2] - E_{V(P)}[G]^2} \quad (4.4)$$

4. Vérifier la condition :

$$E_{V(P)}[G] - \sigma_{V(P)} < G_P < E_{V(P)}[G] + \sigma_{V(P)} \quad (4.5)$$

avec G_P étant le niveau d'intensité du pixel P .

Si cette condition n'est pas vérifiée, l'agent se déplace aléatoirement sur un autre pixel et revient à l'étape 1. Dans le cas contraire, il crée une nouvelle région, en phase d'initialisation et composée seulement de son pixel germe P . Le contour de la région sera quant à lui Composé de l'ensemble des pixels appartenant au voisinage de P .

4.3.1 Accroissement d'une région

A chaque cycle d'exécution, l'agent essaye d'élargir la région à laquelle il est associé. Cette région est définie par un identifiant unique, un ensemble R de taille $\text{Card}(R)$ de tous les pixels appartenant à cette région, ainsi que d'un ensemble C de taille $\text{Card}(C)$ de tous les pixels constituant le contour de la région. Pendant ce cycle, l'agent parcourt un à un l'ensemble des pixels contenus dans C et vérifie la possibilité de leur appartenance à sa région courante. Cette vérification se fait par l'intermédiaire d'un critère d'homogénéité qui diffère selon que la région soit ou non en phase d'initialisation.

Au début du processus d'accroissement de la région, l'ensemble R ne contient que le pixel P , germe de celle-ci, alors que l'ensemble C n'est quant à lui composé que des

pixels appartenant au voisinage d'ordre "1" de P . La région est considérée alors par l'agent comme étant en phase d'initialisation.

Cette phase se déroule entre la création de la région et le moment où cette dernière atteint la taille minimale de confiance L_{min} définie par l'utilisateur et ceci, à l'aide de résultats expérimentaux. Cette taille représente le nombre minimal de pixels que doit contenir une région pour être considérée comme appartenant à une zone homogène. Lors de cette phase d'initialisation, le critère d'homogénéité utilisé pour déterminer l'appartenance ou non d'un pixel à une région est le suivant :

$$E_R[G] - (E_R[G] * \varepsilon_{Tolerance}) < G_P < E_R[G] + (E_R[G] * \varepsilon_{Tolerance}) \quad (4.6)$$

Avec $\varepsilon_{Tolerance}$ est le seuil de tolérance. Il représente la différence maximale entre l'intensité d'un pixel pouvant être assimilé et la moyenne de la région .

Lorsqu'une région atteint la taille minimale de confiance, elle quitte alors la phase d'initialisation. Dès lors, le critère d'homogénéité précédemment cité est remplacé par le critère suivant :

$$E_R[G] - \sigma_R < G_P < E_R[G] + \sigma_R \quad (4.7)$$

L'intensité du pixel n'est plus comparée à l'aide du seuil de tolérance. En effet, les pixels retenus par l'agent sont ceux permettant la stabilisation ou l'amélioration de l'homogénéité de la région associée à cet agent.

À chaque fois qu'un pixel de contour vérifie le critère d'homogénéité (Lors de la phase d'initialisation ou après), il est ajouté à la liste des pixels à assimiler par la région et sera pris en compte lors du calcul du critère d'homogénéité des pixels suivant.

Une fois que tous les pixels appartenant au contour C soient parcourus par l'agent, on obtient alors l'ensemble de tous les pixels pouvant être ajoutés à la région courante et que l'on notera A . Cette liste sera donc utilisée par l'agent pour mettre à jour l'ensemble R (ensemble des pixels de la région associée à l'agent) et ceci en ajoutant à R tous les pixels de cette liste n'appartenant pas à une autre région et en manquant ces derniers

avec l'identifiant de la région courante à laquelle est associé l'agent. Ce marquage des pixels sera par la suite, utilisé pour identifier les pixels d'une région à l'intérieur de l'image. Il permet aussi à chaque agent d'identifier toutes les régions connexes à la sienne.

Après que l'ensemble R soit mis à jour, l'agent doit recalculer le contour de sa région. Il parcourt donc un à un les pixels contenus dans A et pour chacun d'eux, il réalise les opérations suivantes :

- Retirer le pixel P contenu dans A de l'ensemble C .
- Calculer $V(P)$, l'ensemble des pixels voisins de P .
- Ajouter à l'ensemble C tous les pixels P_i de $V(P)$ vérifiant la condition suivante :

$$(P_i \in V(P)) \wedge (P_i \notin R) \wedge (P_i \notin C)$$

Une fois les ensembles R et C mis à jour, l'agent vide l'ensemble A et recommence ensuite son cycle d'exécution. Le cycle d'exécution peut donc être assimilé à un processus itératif d'agrégation, où chaque agent sélectionne progressivement les pixels de contour qui vérifient le critère d'homogénéité de la région à laquelle il est associé. Il retire ensuite les pixels sélectionnés du contour pour les inclure dans la région. L'agent recalcule ensuite le contour de sa région et recommence ensuite le processus jusqu'à atteindre l'un des critères d'arrêt définis plus loin dans cette section. Ce mécanisme de croissance permet de déterminer les régions homogènes présentes dans une image ainsi que leur contour, aussi bien le contour externe qu'interne d'une région.

4.3.2 Fusion de région

Après le démarrage du système les agents parcourent l'image à la recherche d'un pixel germe sur lequel se positionner. Après s'être positionné, chaque agent commence à accroître sa région en assimilant les pixels de contour satisfaisant le critère d'homogénéité comme cela est expliqué dans la sélect précédente.

La croissance des régions associées aux agents se fait donc simultanément. De cette manière les différentes régions se rencontrent lors de leur croissance. Ainsi, lorsque deux régions R_1 et R_2 deviennent connexes l'une à l'autre, les agents auxquels ces régions sont associés opèrent (si la condition de fusion est vérifiée) la fusion des deux régions en une seule région et cela suivant le processus décrit dans ce qui suit.

Lors du processus de croissance, chaque agent parcourt la liste des pixels du contour de la région à laquelle il est associé pour l'assimiler éventuellement à sa région. Si ce pixel de contour est déjà marqué par l'identifiant d'une autre région, dans ce cas l'agent 1 considère ce pixel comme faisant partie d'une autre région connexe à la sienne et essaye alors de fusionner les deux régions.

Cet agent crée ainsi une nouvelle région temporaire R_* contenant l'ensemble des pixels de sa région, ainsi que l'ensemble des pixels de la région 2 connexe à région 1 :

$$R_* = R_1 \cup R_2 \quad (4.8)$$

L'agent 1 calcule ensuite σ_{R_*} l'écart type des intensités des pixels de R_* et le compare à l'écart type σ_{R_1} de la région à laquelle il est associé. Si $\sigma_R < \sigma_{R_1}$, l'agent conclura que la fusion des deux régions aura pour conséquence une diminution de l'hétérogénéité des pixels de sa région. La fusion sera dans ce cas acceptée. L'agent 1 créera donc une nouvelle région ayant pour ensemble de pixel R_* et pour ensemble de contours C_* tel que :

$$C_* = C_1 \cup C_2 \quad (4.9)$$

L'agent sera associé à la région résultant de la fusion dont il poursuivra l'accroissement, l'agent 2 ayant perdu la région à laquelle il était associé lors de la fusion, sera de nouveau libre. Il retournera donc à l'étape de recherche d'un pixel germe.

4.3.3 Concurrence entre régions

Pendant le processus de croissance de la région et plus particulièrement lorsque l'agent parcourt son contour à la recherche de pixels à ajouter à sa région, il arrive qu'il rencontre des pixels appartenant à une autre région connexe à la sienne. L'agent tente à ce moment-là de fusionner ces deux régions comme cela est décrit dans la section précédente. Dans le cas où la fusion des deux régions n'est pas possible et si le pixel satisfait le critère d'homogénéité de la région associée à l'agent, il essaye dans ce cas de le réaffecter à cette dernière et cela de la manière décrite dans ce qui suit.

Pour pouvoir assimiler le pixel d'une autre région, l'agent doit d'abord calculer les valeurs absolues des écarts d'intensités entre le pixel et la région à laquelle il appartient ainsi qu'entre le pixel et la moyenne de la région de l'agent tel que :

$$E_i = |E[R_i] - G_p| \quad (4.10)$$

Il compare ces deux valeurs et le pixel sera alors affecté à la région ayant la moyenne la plus proche de l'intensité du pixel en litige. Cette approche est logique mais s'avère peu efficace car lors de la croissance des régions effectuée par les agents, on observe que les régions de faibles tailles empêchent la croissance de régions de plus grandes tailles et ceci en subtilisant leurs pixels, on obtient alors une sur-segmentation de l'image. Pour remédier à cela, on divise l'écart entre intensité d'un pixel et moyenne d'une région par le nombre de pixels de la région, on notera cette valeur comme E_i^* tel que :

$$E_i^* = \frac{|E[R_i] - G_p|}{Card(R_i)} \quad (4.11)$$

L'agent comparera alors les E^* des deux régions en litige, le pixel sera alors affecté à celle ayant la valeur E^* la plus faible. Cette méthode permet ainsi de diminuer l'effet de sur segmentation et d'améliorer la détection des contours des régions lors du processus de segmentation.

4.3.4 Épilogue de l'accroissement

L'épilogue de l'accroissement représente le moment où tous les pixels du contour ne satisfont pas le critère d'homogénéité de la région, donc ne peuvent être assimilés et il n'y a aucune possibilité de fusion avec une région connexe, à ce moment-là, deux cas de figure sont possibles.

1. Région en initialisation : Le premier cas de figure représente l'arrêt de la croissance de la région associée à l'agent alors que celle-ci est encore en phase d'initialisation, c'est-à-dire que le nombre de pixels contenus dans l'ensemble R de la région est inférieur à la taille minimale de confiance notée T_{min} , taille minimale nécessaire pour que la région associée à l'agent soit considérée comme zone homogène de l'image. Dans ce cas précis, l'agent considère que sa région n'est pas assez vaste pour être considérée comme zone homogène de l'image, il supprime alors cette région et réaffecte l'étiquette de non-marqué (libre) à tous les pixels qui en faisaient partie. L'agent repart ensuite à la recherche d'un germe pour une nouvelle région.
2. Région n'étant plus en phase d'initialisation : si à l'arrêt de la croissance de la région, sa taille est supérieure ou égale à la taille minimale de confiance (T_{min}), l'agent la considère donc comme étant une zone homogène de l'image. Il entame alors l'affaiblissement du bruit présent au sein de la région. Pour ce faire, l'agent parcourt les pixels du contour de sa région et assimile à cette dernière tous les pixels vérifiant la condition suivante :

$$(P \in C \wedge \forall P' \in V(P)) \Rightarrow P' \in R$$

Une fois cette tâche terminée, l'agent enregistre sa région associée et repart à la recherche d'un pixel pouvant faire office de germe pour une nouvelle région

4.4 Conclusion

Dans ce chapitre, nous avons présenté notre approche de segmentation d'images basée sur l'utilisation d'un environnement multi-agents à base d'agents réactifs. Cette approche fait coopérer les techniques de croissance et de fusion de régions, cela en y ajoutant une notion de concurrence sur les pixels aux frontières communes des régions. Les agents distribués de manière aléatoire sur l'image, recherchent des germes pour leurs régions, puis entament leur croissance. Lors de cette phase, les agents interagissent entre eux en fusionnant et en se subtilisant des pixels afin de maximiser la surface de leurs régions respectives. Cette approche fait ainsi émerger un comportement complexe qu'est la segmentation d'images, par l'interaction d'entités à comportement simple que sont les agents. L'approche proposée a été donc implémenter pour pouvoir tester son efficacité sur différents types d'images. L'implémentation et les résultats expérimentaux des testes sont présentés dans le chapitre 5.

Chapitre 5

Implémentation et résultats expérimentaux

5.1 Introduction

Il existe un grand nombre de Frameworks de développement pour les systèmes multi-agent, offrant pour la plupart, des prototypes d'agent ainsi que des systèmes de communication inter-agent utilisés par les développeurs pour l'implémentation des différentes approches SMA. Ces Framework disponibles sont majoritairement open source et ont comme principal objectif, la portabilité à tous les systèmes d'exploitation et configurations matérielles. Cette portabilité a néanmoins un coût en termes de fiabilité, de performance et de sécurité.

Un système multi-agents pouvant être utilisé en entreprise par exemple se doit de disposer d'un mécanisme de conservation de l'état du système et des agents en cas de pannes éventuelles. La gestion de l'administration des privilèges des agents doit être elle aussi effectuée d'une manière efficace, cela en vue d'éventuelles interactions entre le SMA et d'autres environnements opérationnels ou solutions logicielles. Pour cela le SMA doit être en liaison étroite avec le système d'exploitation sur lequel il s'exécute

[53].

Dans notre travail, l'implémentation de notre approche est réalisée sous environnement Microsoft Windows à l'aide de la technologie .Net. Microsoft .Net est une plateforme d'outils pour le développement, le déploiement et l'exécution d'applications de divers types. Nous allons donc débiter ce chapitre par une brève présentation de la technologie .Net utilisée dans notre travail, nous présenterons l'implémentation de l'approche que nous avons proposée. Nous passerons ensuite à la présentation de quelques résultats expérimentaux et nous terminerons par une conclusion.

5.2 Le Framework .Net

Le Framework .NET est un ensemble de technologies servant de base pour développement d'applications. Il est composé de deux parties :

- Un système d'exécution virtuel appelé CLR (Common Language Runtime) équivalant au JVM de JAVA et qui se trouve au coeur de toute la structure .NET et sur laquelle tout repose. Grâce à ce système, le code des programmes n'est plus directement compilé en code natif, mais dans un langage intermédiaire nommé MSIL (Microsoft Intermediate Language). Le CLR est chargé de la prise en charge et la compilation du code ainsi que la gestion de la mémoire et la durée de vie des objets lors de l'exécution grâce au Garbage Collector. Le .NET Framework n'étant pas sensible à un langage particulier, la programmation de différents modules peut être réalisée avec différents langages parmi lesquels le c sharp.
- Une vaste bibliothèque de classes, offrant aux développeurs un large panel de services et de fonctionnalités simplifiant et réduisant le travail de programmation.

La technologie .Net offre de nombreux avantages pour le développement d'applications parmi lesquels on peut citer :

- La simplification du développement grâce à l'utilisation de sa bibliothèque.

- La simplicité de déploiement grâce l’effort fourni par Microsoft pour faciliter l’interaction et l’intégration des applications .Net dans l’environnement Windows.
- La simplicité de la gestion des versions grâce à un système de manifeste qui enregistre les données concernant la version d’une assembly, permettant ainsi à plusieurs versions d’une même assembly de coexister sans conflit.

Le Framework .Net offre donc des facilités pour le développement de solutions logicielles en général, ce qui est le cas aussi pour les applications multi-agents comme cela est expliqué dans [53]. C’est pour cela que nous avons donc choisi cette technologie pour l’implémentation de notre modèle, cette implémentation sera détaillée dans les sections suivantes.

5.3 Implémentation de l’approche

D’après les constatations initiées dans [54], les résultats de la simulation d’un système multi-agents sont étroitement liés à la manière dont l’approche a été implémentée. L’un des principaux facteurs pouvant influencer sur l’implémentation d’une approche SMA est ce qu’on appelle « la politique d’exécution des agents » et qui a un impact direct sur l’évolution du modèle implémenté.

Le choix de la politique d’exécution est donc une étape importante dans l’implémentation d’un système multi-agents, car de celle-ci résulte de la gestion du temps d’activation des agents. Le cycle de temps d’une simulation impacte sur la granularité de l’action que l’agent peut effectuer et l’activation sur le moment où cette action peut être réalisée. La plupart des plates-formes de simulation multi-agents ont une gestion du temps discrète où le temps et les actions sont discrétisés. Dans ce cadre, un des problèmes récurrents est la synchronisation des agents [55].

Il existe plusieurs politiques d’exécution dans l’implémentation des systèmes multi-agent, nous avons choisi dans notre travail d’utiliser celle étant la plus répandue et

plus simple à réaliser, appelée « simulations à pas de temps constant ». Cette politique consiste en l'introduction d'un agent appelé « scheduler » ayant pour seule fonction l'activation séquentielle des agents lors de l'exécution.

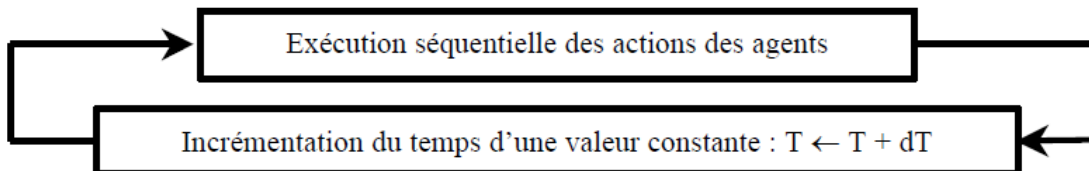


FIGURE 5.1 – Principe d'une simulation à pas de temps constant [14]

Cette technique a pour avantage la suppression des conflits d'accès à d'éventuelles variables ou objets partagés par les agents et contraint les agents à n'agir qu'une seule fois par pas (cycle). En utilisant cette politique d'exécution, nous avons abouti à une implémentation composée des classes suivantes :

5.3.1 Scheduler

La classe scheduler est nécessaire à la politique d'exécution de simulation à pas de temps constant. Il s'agit en effet d'une sorte de pilote du système. S'occupant de l'initialisation des différents objets et activant de manière séquentielle, l'ensemble des agents lors d'un pas d'exécution. À chaque pas du système, le scheduler parcourt l'ensemble des agents en les activant un par un. Après chaque activation, l'agent activé exécute sa tâche selon son contexte, puis retourne la main au scheduler pour qu'il puisse activer l'agent suivant et cela, jusqu'à ce que tous les agents soient activés. Il incrémente ensuite le pas et entame une nouvelle itération. Cette politique permet d'éviter les conflits entre agents sans avoir recours à des mécanismes complexes de synchronisation. Les principaux attributs qui composent cette classe sont :

- *ImageASegmenter* : fait référence à l'image traitée par le système.

- *NombreAgents* : contenant le nombre d’agents initialisés dans l’image.
- *NombreIteration* : contenant le nombre de pas d’exécution maximum.

Les principales méthodes de cette classe sont quant à elles les suivantes :

- *Initialiser()* : pour initialiser et positionner la population des agents.
- *Activer()* : pour lancer le processus itératif d’activation des agents.

5.3.2 Pixel

Le pixel est l’élément de base dont se compose une image numérique matricielle. La classe pixel que nous avons utilisée comprend les attributs suivants :

- *Colonne* : représente la coordonnée X du pixel dans l’image.
- *Ligne* : représente la coordonnée Y du pixel dans l’image.
- *NiveauGris* : représente l’intensité du pixel.
- *IdentifiantRegionAppartenance* : contient l’étiquette de la région à laquelle appartient le pixel, elle est égale à null tant que le pixel reste non marqué.

La principale méthode de la classe Pixel est *TrouverVoisins()* et qui, comme son nom l’indique, permet de déterminer la liste des voisins du pixel.

5.3.3 ImageSlice

La classe ImageSlice contient toutes les informations relatives à une tranche d’image IRM (image 2D) extraite d’une image IRM 3D, ainsi que les différents seuils utilisés pour sa segmentation. Elle comprend les attributs suivants :

- *TypeTranche* : axiale, coronale ou sagittale.
- *NombreColonnes* : représente la largeur de l’image en termes de pixels.
- *NbreLignes* : représente la hauteur de l’image en termes de pixels.
- *Pixels* : représente la liste de tous les pixels contenus dans l’image.
- *SeuilTolerance* : représente la différence maximale entre l’intensité d’un pixel pouvant être assimilé à une région et la moyenne de cette région.

- *Lmin* : représente la taille minimale en termes de nombre de pixel pour qu'une région de cette image puisse être considérée comme zone homogène.

La principale méthode de la classe *ImageSlice* est *ChargerImageSlice()* qui permet de charger l'image à partir du fichier utilisé.

5.3.4 Region

La Classe *Region* représente une zone homogène de l'image. Elle se compose d'un ensemble de pixels partagent et vérifiant un critère d'homogénéité. Elle possède les attributs suivants :

- *Identifiant* : représente l'identifiant unique de la région.
- *Initialisation* : un booléen indiquant si la région est ou non encore en phase d'initialisation.
- *Pixels* : représentant la liste des pixels composant la région.
- *PixelsContour* : représentant la liste des pixels composant le contour de la région.

Les principales méthodes de cette classe sont, quant à elles, les suivantes :

- *Moyenne()* : calcule l'intensité moyenne des pixels de la région.
- *EcartType()* : calcule l'écart type des intensités des pixels de la région.

5.3.5 Agent

L'agent est l'entité active du système. Il se déplace sur l'image, créer des régions, les fait croitre, les fusionnent...etc. A chaque pas du système il exécute ces différentes tâches suivant le modèle défini dans les sections précédentes. La classe agent possède les attributs suivants :

- *PixelCourant* : représente la position de l'agent.
- *ImageCourante* : représente l'image sur laquelle se déplace l'agent.
- *RegionCourante* : représente la région à laquelle l'agent est associé.

Ses méthodes, sont quant à elles, les suivantes :

- *PixelGerme()* : Vérifie si le pixel sur lequel est positionné l'agent peut faire office de germe.
- *PixelVerifieCritere(Pixel P)* : Teste si le pixel en question, vérifie ou non le critère d'homogénéité de la région associée à l'agent.
- *AssimilerPixelRegion(Pixel P)* : Ajoute le pixel en question à la région associée à l'agent.
- *AjouterPixelAuContour(Pixel P)* : Ajouter le pixel en question au contour de la région.
- *SupprimerBruit()* : Élimine le bruit à la fin de la croissance de la région.
- *Deplacer()* : Déplace l'agent aléatoirement sur l'image.
- *Fusionner(Region R)* : Fusionne la région concernée avec celle de l'agent dans le cas où le critère de fusion est satisfait.
- *Step()* : Elle est la méthode principale de la classe agent. Elle représente un pas d'exécution de l'agent dans le système et détermine le comportement qu'il va suivre.

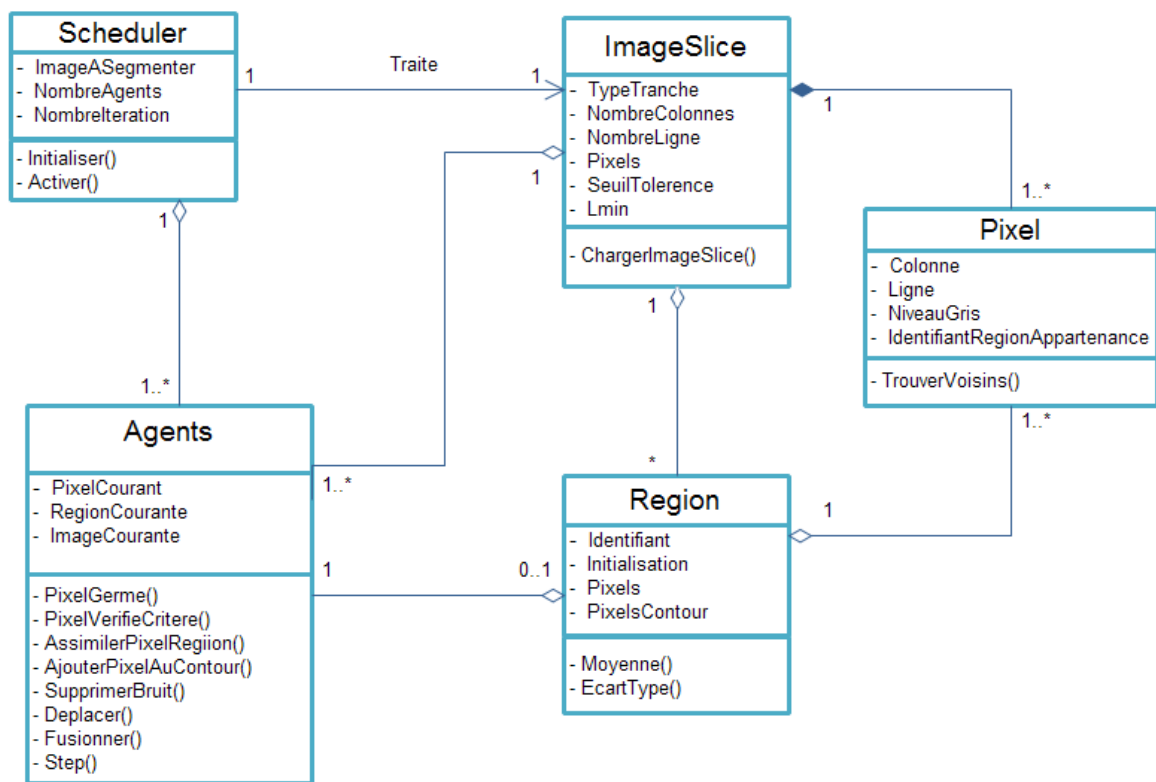
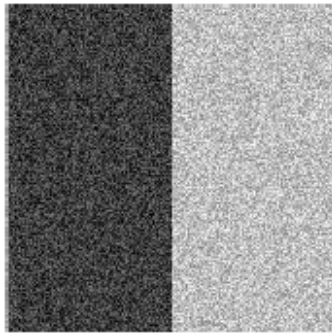


FIGURE 5.2 – Diagramme de Classes

5.4 Résultats expérimentaux

Pour nos expérimentations, nous avons appliqué notre système à des images à niveaux de gris de différentes natures. D'abord, le système est testé sur des images synthétiques hautement bruitées. Il est parvenu ainsi, comme le montre la figure 5.3, à segmenter cette image, détectant les deux zones homogènes qui la composent et éliminant le bruit présent dans l'image améliorant de ce fait sa qualité.



(a) images synthétiques hautement bruité



(b) image après segmentation

FIGURE 5.3 – segmentation d'une image synthétique hautement bruitée

Le deuxième test que nous avons effectué est pratiqué sur une image photographique à niveau de gris de grains de riz éparpillés sur une surface plane. Sur cette image, le système parvient à détecter les contours des différents grains de riz ainsi que les zones homogènes de l'image, comme cela est illustré dans la figure 5.4.

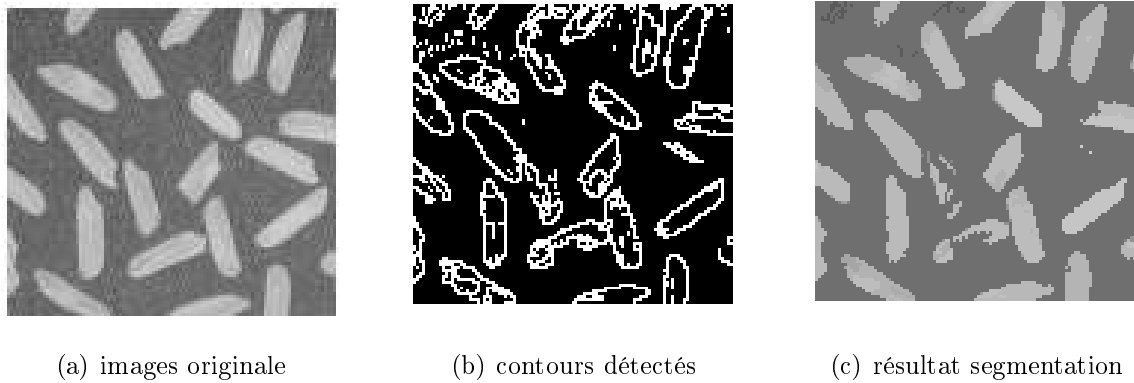


FIGURE 5.4 – segmentation d’une image photographique de grains de riz

Après avoir testé notre système sur des images synthétiques et des images photographiques, nous passons à un type tout à fait différent d’images que sont les images médicales, nous allons pour cela utiliser une image de scanner.

L’image utilisée pour nos expérimentations représente une coupe de scanner de la cage thoracique d’un patient humain. La segmentation de cette image par notre système offre des résultats satisfaisants. Plusieurs régions homogènes de l’image ont pu ainsi être déterminées précisément par le système. Ces différentes régions sont illustrées en rouge dans la figure 5.5. Cette segmentation pourrait par la suite être utilisée dans l’interprétation de ce type d’images.

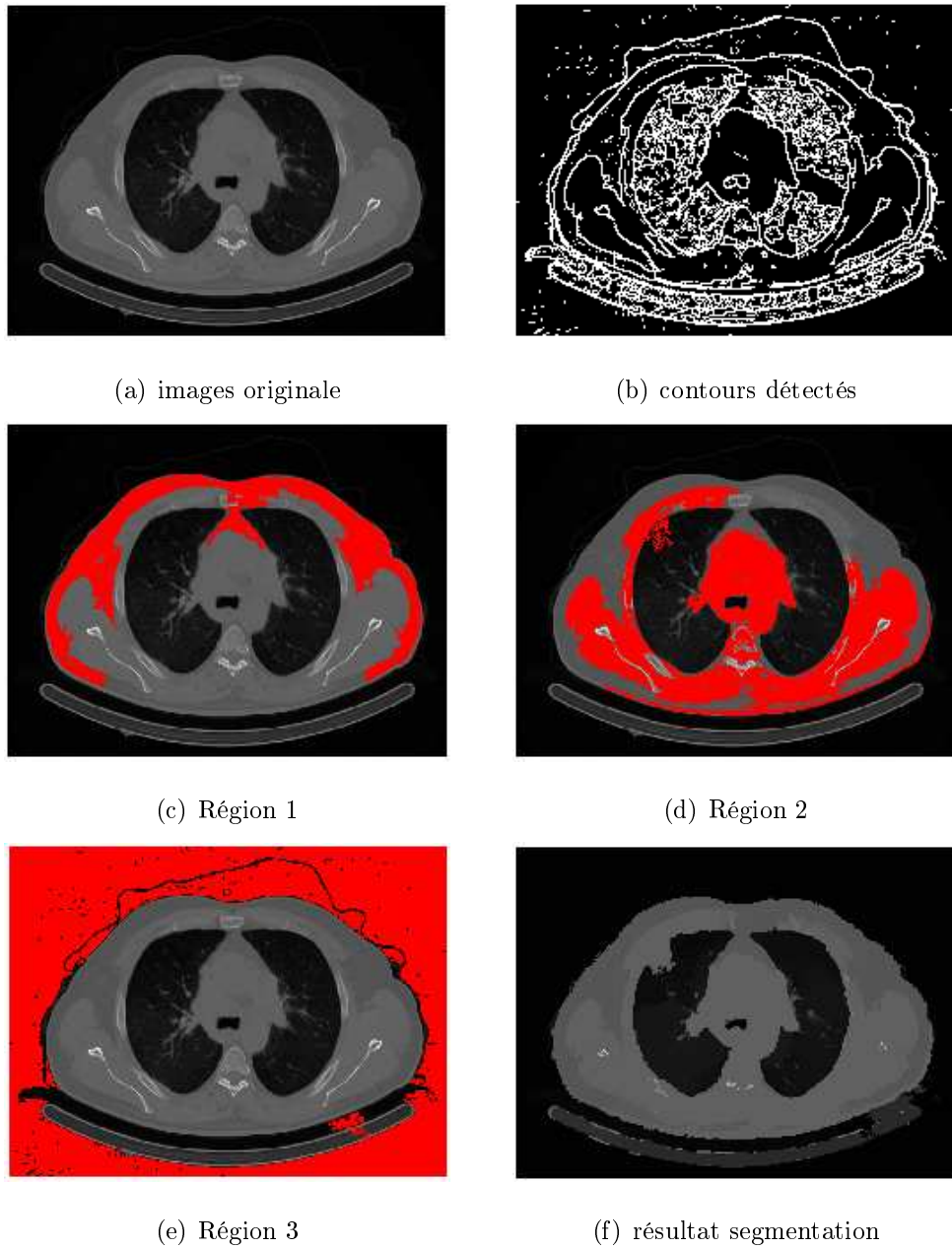


FIGURE 5.5 – segmentation d’une coupe de scanner thoracique

La dernière partie de nos expérimentations utilise des images IRM. Ce type d'images est par nature sujet à la présence de bruit, augmentant ainsi la difficulté à les segmenter. Nos tests seront donc effectués sur une IRM cérébrale d'un patient adulte, les images utilisées sont des coupes axiales, en pondération T1 et de résolution 256x256 en niveaux de gris. Nos expérimentations ont ainsi montré que le système parvient à détecter efficacement les contours de l'image et à lisser les différentes zones homogènes de celle-ci et cela en dépit du bruit.

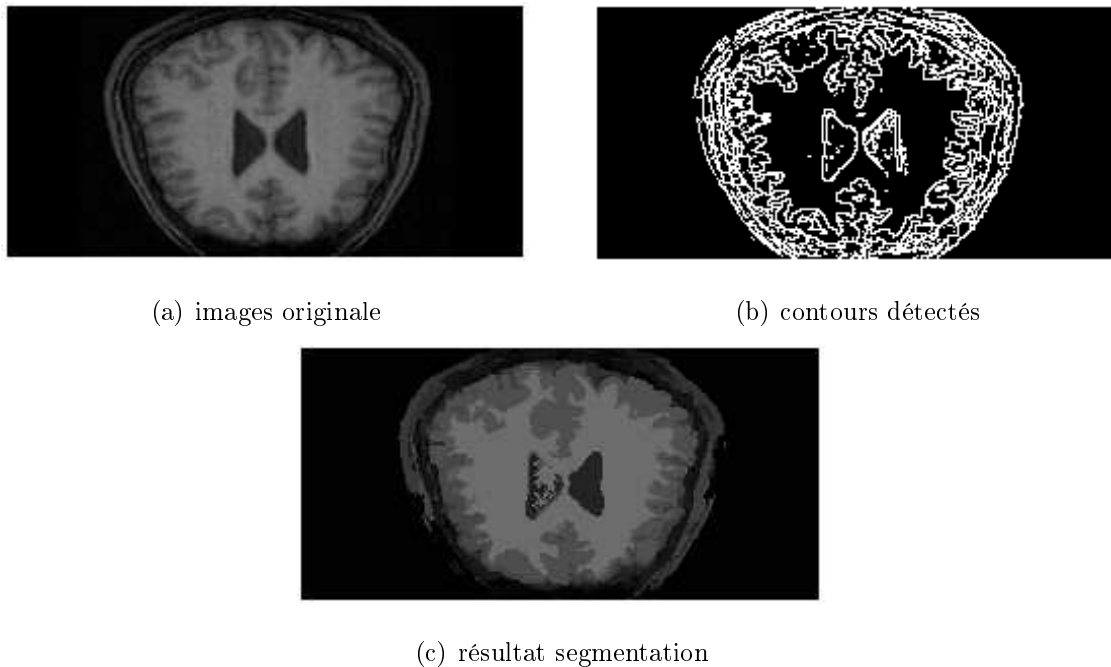


FIGURE 5.6 – segmentation d'une coupe axiale d'une IRM cérébrale en pondération T1

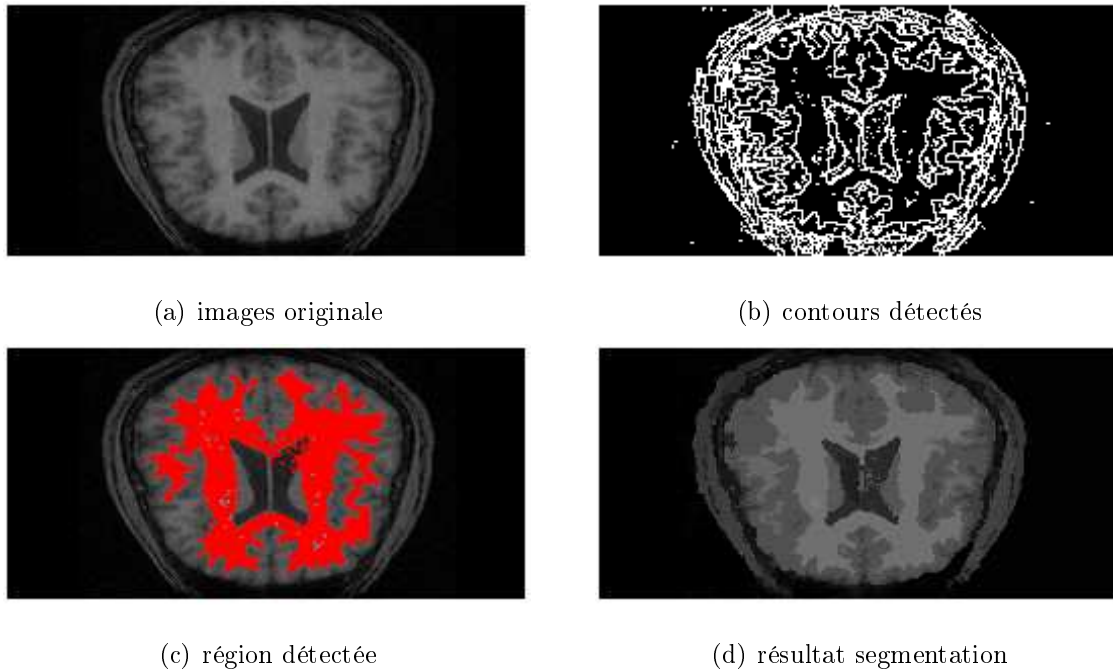


FIGURE 5.7 – segmentation d’une autre coupe axiale d’IRM cérébrale

5.5 Conclusion

Dans ce chapitre nous avons décrit la technologie ainsi que les différentes classes que nous avons utilisées pour l’implémentation de l’approche que nous avons proposée. L’interaction des agents bien qu’ayant un comportement simple, permet l’émergence une segmentation efficace, amoindrissant le bruit de l’image et permettant d’en extraire les zones homogènes. Les différents tests ont montré l’efficacité de l’approche pour la segmentation d’images de différents types, ils ont aussi permis d’exposer sa robustesse quant à la présence de bruit sur l’image.

Bien qu’offrant une segmentation efficace des images, le système reste néanmoins dépendant de l’ajustement des seuils que sont le nombre d’agent présent sur l’image, la taille minimale d’une région L_{min} et le seuil de tolérance lors de l’initialisation. Ces diffé-

rentes valeurs doivent être ajustées selon le type de l'image et le niveau de bruit qu'elle présente. Par ailleurs, l'implémentation en pseudo parallélisme utilisant le Scheduler, Bien que simple à implémenter, amoindrie de manière significative les performances du système en en termes de temps d'exécution qui peut aller de plusieurs dizaines de seconde à plusieurs minutes. Une implémentation parallèle, bien que plus complexe, offrira probablement de meilleures performances.

Malgré ces quelques inconvénients, le système offre des résultats satisfaisant pour la segmentation d'images de niveaux de gris de différentes natures et cela par l'interaction d'agent réactif ayant un comportement relativement simple. En outre, une amélioration du système peut être envisagée dans de futurs travaux par l'ajout d'une notion d'auto-adaptation aux images traitées, éliminant ainsi le problème des seuils.

Chapitre 6

Conclusion générale

L'objectif de notre travail est la réalisation d'un système permettant la segmentation des différents types d'images médicales et cela par l'utilisation de la technologie des systèmes multi-agents.

Nous avons donc entamé ce rapport par la présentation d'un type d'images médicales qui est l'imagerie par résonance magnétique (IRM) ainsi que la procédure d'acquisition de ce type d'images. Nous avons ensuite poursuivi par la présentation de l'état de l'art concernant les différentes techniques de segmentation d'images nous permettant ainsi de constater la diversité de ce domaine ainsi que l'activité des recherches de cette discipline. Lors de cette étude bibliographique, nous nous sommes plus particulièrement penchés sur les approches présentes dans la littérature utilisant les systèmes multi-agents pour la segmentation d'images. Cela a permis de souligner que bien qu'étant encore peu répondue, les systèmes multi-agents commencent à être de plus en plus utilisés dans le domaine du traitement d'images, mais les solutions proposées restent en général spécifiques à un type d'image et ne sont pas généralisables à tous les types d'images.

L'approche que nous avons proposée et implémentée se base sur les techniques de croissance et de fusion de régions. Les agents parcourent leur environnement qui est l'image à la recherche d'un germe pour entamer la croissance de sa région. Lors de

cette croissance, les agents fusionnent entre eux et essaient d'assimiler le plus de pixels possibles satisfaisant le critère d'homogénéité de leur région. L'implémentation fut effectuée en utilisant la technologie du Framework .net de Microsoft et le langage c sharp, un langage orienté objet issu du c++ et du java. Dans cette implémentation, nous avons utilisé la politique d'exécution appelée « simulation à pas de temps constant » utilisant un Scheduler pour activer de manière itérative les agents présents dans le système. Cette politique nous a permis d'implémenter simplement l'approche que nous avons proposée et ce, en évitant les conflits d'accès à d'éventuels objets partagés par les agents.

La dernière partie de notre travail concerne quant à elle, le test de l'efficacité de notre système pour la segmentation d'images de différents types. Nous avons entamé les tests par une image synthétique hautement bruitée et que notre système est parvenu à segmenter de manière efficace, cela fut aussi le cas pour une image photographique. Nous avons achevé nos expériences par le test de notre système sur la segmentation d'images médicales. Ces expérimentations utilisent les IRM cérébrales. Le système a pu ainsi extraire les zones homogènes et délimiter les contours présents dans les images.

Notre système bien qu'étant parvenu à segmenter efficacement différents types d'images médicales, reste néanmoins dépendant de l'ajustement de ses seuils. Pour parvenir à une segmentation satisfaisante, ces différentes valeurs doivent pour l'instant déterminer de manière expérimentale. Cela reste un inconvénient non négligeable, mais qui pourrait éventuellement être corrigé par l'ajout d'une notion d'apprentissage. Cet apprentissage pourrait être réalisé par l'intermédiaire d'un agent utilisant l'apprentissage par renforcement qui se chargera d'adapter les seuils du système suivant le type d'images segmentées et cela d'après l'expérience acquise au fil des exécutions. La politique d'exécution choisie a aussi l'inconvénient d'augmenter le temps d'exécution nécessaire à la segmentation des images affaiblissant ainsi les performances du système.

Nos perspectives pour d'éventuels travaux futurs seraient donc l'amélioration de notre approche par l'ajout de la notion d'apprentissage ainsi que par l'implémentation distribuée de l'approche pour augmenter les performances du système. Cela pourrait à terme permettre d'aboutir à un système auto-adaptatif de segmentation d'images pouvant par la suite, être intégré dans un outil médical d'aide au diagnostique.

Bibliographie

- [1] A. Capelle, *Segmentation d'images IRM multi-échos tridimensionnelles pour la détection des tumeurs cérébrales par la théorie de l'évidence*. PhD thesis, Université de Poitiers, 2004.
- [2] B. Nazarian, "Imagerie médicale 3d visualisations, segmentations et reconstructions," tech. rep., CNRS, 2002.
- [3] J. Lecoœur and C. Barillot, "Segmentation d'images cérébrales : État de l'art," tech. rep., INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE, 2008.
- [4] V. Duay, N. Houhou, and J. Thirane, "Atlas-based segmentation of medical images locally constrained by level sets," tech. rep., Signal Processing Institute, Ecole Polytechnique Federale de Lausanne, 2005.
- [5] N. Duta and M. Sonka, "Segmentation and interpretation of mr brain images : An improved active shape model," *IEEE Transactions on Medical Imaging*, pp. 1049–1062, 1998.
- [6] A. Meziane, "Étude et comparaison des méthodes de segmentation d'images cérébrales," Master's thesis, Université Abou-Bakr Belkaid-Tlemcen, 2011.
- [7] *Biomedical Image Analysis : Segmentation*. Morgan and Claypool, 2009.
- [8] K. Renaud, *Equations aux Dérivées Partielles, Evolutions de Courbes et de Surfaces et Espaces d'Echelle : Applications à la Vision par Ordinateur*. PhD thesis, Ecole

- nationale française des ponts et chaussées, 1997.
- [9] Y. Jacquélet, J. Montois, E. Duchesnay, M. Garreau, and A. Kinie, “Architecture pyramidale agent pour la segmentation d’image : Application à l’extraction d’une zone lobulaire issue d’une mammographie,” in *GRETSI, Groupe d’Etudes du Traitement du Signal et des Images*, 2001.
- [10] J. Fleureau, M. Garreau, D. Boulmier, C. Leclercq, and A. Hernandez, “Segmentation 3d multi-objets d’images scanner cardiaques : une approche multi-agents,” *IRBM*, vol. 30, no. 3, pp. 104–113, 2009.
- [11] F. Sahba, H. R. Tizhoosh, and M. M. A. Salama, “Application of opposition-based reinforcement learning in image segmentation,” in *IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)*, pp. 246–251, 2007.
- [12] K. E. Melkemi and S. Foufou, “Fuzzy distributed genetic approaches for image segmentation,” *Journal of Computing and Information Technology - CIT*, pp. 221–231, 2010.
- [13] C. Bourjot, V. Chevrier, and V. Thomas, “A new swarm mechanism based on social spiders colonies : From web weaving to region detection,” *WEB INTELLIGENCE AND AGENT SYSTEMS*, vol. VOL 1, pp. 47–64, 2003.
- [14] F. Michel, O. Gutknecht, and J. Ferber., “Une méthodologie pour la conception de simulateur multi-agents basée sur l’organisation,” in *Plate-forme AFIA 2001 - Atelier SMA*, 2001.
- [15] *Analyse d’images : filtrage et segmentation*. Enseignement de la physique, 1995.
- [16] H. Laguel, *Déploiement sur une Plateforme de Visualisation d’un Algorithme Coopératif pour la Segmentation d’Images IRM basé sur les Systèmes Multi-Agents*. PhD thesis, Université des Sciences et de la Technologie Houari Boumediene, 2010.

BIBLIOGRAPHIE

- [17] G. Dreyfus, “Les réseaux de neurones,” *Mécanique Industrielle et Matériaux*, septembre 1998.
- [18] *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [19] W. Aribi, A. Kalfallah, N. Elkadri, L. Farhat, W. Siala, J. Daoud, and M. S. Bouhelel, “Évaluation de techniques pyramidales de fusion multimodale (irm/tep) d’images cérébrales,” in *5th International Conference : Sciences of Electronic, Technologies of Information and Telecommunications*, 2009.
- [20] R. Cannon, J. Dave, and J. Bezdek, “Efficient implementation of the fuzzy c-means clustering algorithms,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.*, pp. 248–255, 1996.
- [21] K. Fukunaga and L. Hostetler, “The estimation of the gradient of a density function with applications in pattern recognition,” *IEEE Transactions on Information Theory*, vol. IT-21, pp. 32–40, 1975.
- [22] D. C. and P. Meer, “Mean shift analysis and applications,” in *Proceedings of the 1999 7th IEEE International Conference on computer Vision (ICCV’99)*, (Greece), 1999.
- [23] E. Parzen, “On estimation of a probability density function and mode,” *Annals of Mathematical Statistics.*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [24] M. Mignotte, “Traitement d’image (segmentation d’image),” tech. rep., Département d’informatique et de recherche opérationnelle université de Montréal.
- [25] *The Nature of Statistical Learning Theory*. Springer, 1995.
- [26] Z. Laoa, D. Shena, A. Jawadb, B. Karacalia, D. Liua, E. Melhema, R. Bryana, and C. Davatzikos, “Automated segmentation of white matter lesions in 3d brain mr images using multivariate pattern classification,” in *Proceedings of the 2006 IEEE*

BIBLIOGRAPHIE

- International Symposium on Biomedical Imaging : From Nano to Macro*, pp. 307–310, 2006.
- [27] Y. Hsiao, C. Chuang, J. Jiang, and C. Chien, “A contour based image segmentation algorithm using morphological edge detection,” in *2005 IEEE International Conference on Systems, Man and Cybernetics*, (Waikoloa, Hawaii), pp. 2962–2967, October 2005.
- [28] H. Park, P. H. Bland, and C. R. Meyer :, “Construction of an abdominal probabilistic atlas and its application in segmentation,” *IEEE Transactions on Medical Imaging*, *Volume 22*, vol. Volume 22, pp. 483–492, April 2003.
- [29] X. Han and B. Fischl, “Atlas renormalization for improved brain mr image segmentation across scanner platforms,” *IEEE TRANSACTIONS ON MEDICAL IMAGING*, *VOL. 26, NO. 4, APRIL*, vol. VOL. 26, pp. 479–486, Avril 2007.
- [30] F. Ding, W. K. Leow, and S. Wang, “Segmentation of 3d ct volume images using a single 2d atlas,” *In Computer Vision for Biomedical Image Applications*, 2005.
- [31] S. Kleina, U. van der Heideb, B. Raaymakersb, A. Kotteb, M. Staringa, and J. Pluima, “Segmentation of the prostate in mr images by atlas matching,” in *Biomedical Imaging : From Nano to Macro, 4th IEEE International Symposium on*, 2007.
- [32] V. Chen and C. Cudel, “Segmentation d’image textiles par l’analyse multiresolution en ondelettes biorthogonales,” in *Quinzième colloque GRETSI. JUAN-LES-PINS*, 1995.
- [33] T. Cootes, C. Taylor, D. Cooper, and J. Graham, “Active shape model (thier training and application),” *Computer Vision and Image Understanding*, vol. Vol 61, pp. 38–59, 1995.
- [34] S. Ettaieb, K. Nawrès, and K. Hamrouni, “Asm+d : Nouveau modèle de formes actives intégrant une relation spatiale de distance,” in *XXIe Colloque GRETSI*,

BIBLIOGRAPHIE

- Dijon France, 2009*, septembre 2009.
- [35] Z. Zhao and E. K. Teoh, “A novel 3d statistical shape model for segmentation of medical images,” *Lecture Notes in Computer Science*, pp. 638–647, 2006.
 - [36] N. Eveno, *Segmentation des Lèvres par un Modèle Déformable Analytique*. PhD thesis, INSTITUT POLYTECHNIQUE DE GRENOBLE, 2003.
 - [37] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes : Active contour models,” *International Journal of Computer Vision*, pp. 321–331, 1987.
 - [38] C. Xu and J. Prince, “Gradient vector flow : a new external force for snakes,” *International Conference on Computer Vision and Pattern Recognition*, pp. 66–71, 1997.
 - [39] C. X. et J. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE Transactions on Image Processing*, pp. 359–369, 1998.
 - [40] C. Ciofolo, *Segmentation de Formes Guidée par des Modèles en Neuro-Imagerie : Intégration de la Commande Floue dans une Méthode de Segmentation par Ensembles de Niveau*. PhD thesis, Université de Rennes 1, 2005.
 - [41] S. Rital, H. Cherifi, and S. Miguët, “Weighted adaptive neighborhood hypergraph partitioning for image segmentation,” *PATTERN RECOGNITION AND IMAGE ANALYSIS : Lecture Notes in Computer Science*, 2005.
 - [42] N. Xu, R. Bansal, and N. Ahuja, “Object segmentation using graph cuts based active contours,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 03)*, 2003.
 - [43] D. Grosgeorge, C. Petitjean, and S. Ruan, “Segmentation d’images par coupe de graphe avec a priori de forme,” in *congrès francophone sur la Reconnaissance des Formes et l’Intelligence Artificielle (RFIA 2012)*, 2012.

BIBLIOGRAPHIE

- [44] A. Rares, M. Reinders, and E. Hendriks, “Mapping image analysis problems on multi-agent systems,” tech. rep., Information and Communication Theory Group TU Delft, November 1999.
- [45] *An Introduction to MultiAgent Systems*. John Wiley and Sons, 2002.
- [46] A. Florea, “Cognitive interactions in a multi-agent system,” in *POLITEHNICA*, University of Bucharest, Computer Science and Engineering Faculty, 1998.
- [47] M. Wooldridge and N. R. Jennings, “Intelligent agents : Theory and practice,” *The Knowledge Engineering Review*, vol. 10, pp. 115–152, 1995.
- [48] R. A. Brooks, “Intelligence without reason,” in *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, (Sydney, Australia), pp. 569–595, 1991.
- [49] P. L. Rosin and O. F. Rana, “Agent-based computer vision,” *Pattern Recognition*, vol. 37, pp. 627–629, 2004.
- [50] P. Grassé, “La construction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes,” in *La théorie de la stigmergie : essais d'interprétation du comportement des termites constructeurs*, vol. 6, pp. 41–84, 1959.
- [51] L.Djerou and M. Batouche, “Une approche biomimetique pour la segmentation d'image,” *Courrier du Savoir*, pp. 33–38, Mars 2009.
- [52] Y. T. J. Liu, “Adaptive image segmentation with distributed behavior based agents,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 544–551, 1999.
- [53] A. Grosso and A. Boccalatte, “An agent programming framework based on the c sharp language and the cli,” in *1st Int. Workshop on C Sharp and .NET Technologies on Algorithms, Computer Graphics, Visualization, Computer Vision and Distributed Computing*, (Plzen, Czech Republic), 2003.

BIBLIOGRAPHIE

- [54] R. L. Axtell, “Effects of interaction topology and activation regime in several multi-agent systems,” in *Second Workshop on Multi Agent Based Simulation. MABS-2000*, 2000.
- [55] F. Badeig and F. Balbo, “Un cadre conceptuel et opérationnel pour la simulation multi-agents,” in *10e Rencontres des Jeunes Chercheurs en Intelligence Artificielle RJCIA*, 2011.

Annexe A

Format Analyse 7.5

A.1 Description du format Analyse 7.5

Mayo Clinic Analyser 7.5 format est un format de fichier pour stocker des données d'IRM. Un ensemble de données Analyser 7.5 se compose de deux fichiers :

- Un fichier d'entête doté d'une extension `.hdr` et qui contient toutes les informations relatives au fichier IRM comme les dimensions, l'orientation.
- Un fichier doté d'une extension `.img` contenant le ou les volumes de l'IRM suivant le schéma décrit dans le fichier d'entête.

A.1.1 Description du fichier d'entête

Le fichier d'entête est représenté comme une structure qui décrit les dimensions et l'historique des données. La structure d'en-tête se compose de trois sous-structures :

- **header key** : qui décrit la structure du fichier d'entête.
- **image dimension** : qui décrit les caractéristiques physique de l'image.
- **data history** : structure optionnel contenant des information sur l'historique des données et des données sur le patient.

comme la partie "data history" contient essentiellement des information sur l'historique, nous n'allons pas définir sa structure dans ce document. Nous allons cependant décrire les valeurs d'un de ses champs (hist.orient) qui sera utiliser pour spécifier l'ordre des voxels dans le fichier image.

Offset	Nom du champs	Type	Description
0	sizeof header	int32	Indique la taille du fichier d'entête, généralement égale à 348 et indique un entête étendu
4	data type	unchar[10]	aucune
14	db name	unchar[18]	aucune
32	extents	int32	oit être égale à 16384
36	session error	int16	aucune
38	regular	uchart	aucun doit être égale à "r" ce qui indique que toutes les images et les volumes sont de la même taille
39	hkey un0	int32	aucune

TABLE A.1 – structure du Header key

A.1.2 Description du fichier .img

Le fichier image Analyse 7.5 se compose d'un long flux d'intensités de voxels. dans ce flux l'ordre des voxel est d'une importance cruciale. Il existe ainsi un ordre par default des voxels dans ce type de fichier, on peut aussi un autre ordre ceci en spécifiant le champ hist.orient. voichtbpi un tableau illustrant les différentes valeurs que peut prendre le champs et les ordres de voxels correspondants :

Offset	Nom du champs	Type	Description
40	dim[0]	int16	Indique le nombre de dimensions dans l'image, généralement égale à 4
42	dim[1]	int16	nombre de voxels dans une ligne de l'image (dimension X)
44	dim[2]	int16	nombre de lignes dans une tranche de l'image (dimension Y)
46	dim[3]	int16	nombre de tranches dans un volume de l'image (dimension Z)
48	dim[4]	int16	nombre de volume dans une image
50	dim[5]	int16	aucune
52	dim[6]	int16	aucune
54	dim[7]	int16	aucune
56	vox units	uchar[4]	unités de mesure spatiales des voxels
60	cal units	uchar[8]	unité de calibrage
68	unused1	int16	aucune
70	data Type	int16	type de donnée du voxel
72	BitPix	int16	nombre de bits d'un occupés par un voxel
74	dim un0	int16	aucune
76	PixDim[0]	float	aucune
80	PixDim[1]	float	largeur d'un voxel en mm
84	PixDim[2]	float	hauteur d'un voxel en mm
88	PixDim[3]	float	épaisseur d'une tranche en mm
92	PixDim[4]	float	aucune
96	PixDim[5]	float	aucune
100	PixDim[6]	float	aucune
104	PixDim[7]	float	aucune
108	vox offset	float	le numéro de l'octet par lequel commence la matrice de voxel dans le fichier .img, peut être négative pour indiquer un offset absolu
112	funused1	float	échelle
116	funused2	float	intensité d'image, égale à zéro à l'origine

Data Type	Valeur	Description
DT NONE	0	Aucune
DT UNKNOWN	0	Aucune
DT BINARY	1	1 bit par voxel
DT UNSIGNED CHAR	2	8 bit par voxel
DT SIGNED SHORT	4	16 bit par voxel
DT SIGNED INT	8	32 bit par voxel
DT FLOAT	16	32 bit par voxel
DT COMPLEX	32	2 x 32 bit par voxel
DT DOUBLE	64	64 bit par voxel
DT RGB	128	Aucune
DT ALL	255	Aucune

TABLE A.3 – Les différents types de codage d'un voxel

Val. hist.orient	Nom de l'ordre	index 0	index 1	index 2
0 (default)	Transversal non renversé	Droite à Gauche	Postérieur à Anté- rieur	Inférieur à Supérieur
1	Coronaire non renversé	Droite à Gauche	Inférieur à Supérieur	Postérieur à Anté- rieur
2	Sagittale non renversé	Postérieur à Anté- rieur	Inférieur à Supérieur	Droite à Gauche
3	Transversal renversé	Droite à Gauche	Antérieur à Posté- rieur	Inférieur à Supérieur
4	Coronaire renversé	Droite à Gauche	Supérieur à Inférieur	Postérieur à Anté- rieur
5	Sagittale renversé	Postérieur à Anté- rieur	Supérieur à Inférieur	Droite à Gauche

TABLE A.4 – Les différentes valeurs de Hist.orient

A.2 Ordre des voxels et terminologie d'orientation utilisée

Le traitement des données d'images IRM nécessite de se familiariser avec les conventions et la terminologie utilisés pour décrire l'orientation de ses images. par exemple les notions Avant, Arrière, haut et bas ne sont pas utilisés car cet terminologie peut prêter à confusion. Il est nécessaire aussi de savoir que les directions employées sont toutes par rapport au patient.

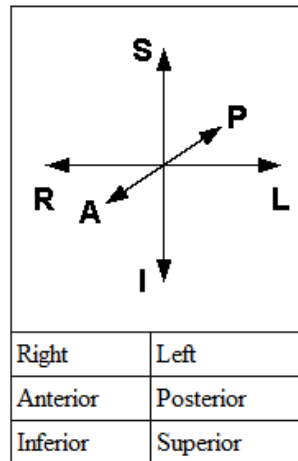


FIGURE A.1 – Axes utilisés dans les images IRM

Pour parler de zones dans le cerveau, il est nécessaire de bien définir les axes employés et quelles sont les directions positives, et l'ordre dans lequel ils seront énumérer pour décrire les coordonnées d'un point. On utilise trois axes, qui pourraient être utilisés dans n'importe quel ordre, et où les deux sens peut être positif, ce qui nous donne un total de $3 * 2 * 1 * 8 = 48$ combinaisons d'axes possibles.

Un certain nombre de formats de fichiers IRM impliquent le stockage des intensi-

tés de voxels simplement comme un flux d'octet représentant les intensités des voxels dans l'image. Ces formats nécessitent l'enregistrement d'un certain nombre de caractéristiques du fichier image, y compris pour voxel, le type de codage utilisé pour l'intensité, et d'autres attributs relatifs aux conditions d'acquisition d'image, les étapes de traitement qui ont été effectuées ainsi que l'ordre dans lequel les voxel sont enregistrés.

En générale, les voxels sont stockés dans une séquence de ligne, ligne après ligne et tranche après tranche. Il faut donc définir les directions dans lesquels sont les lignes et les tranches. Comme dans le cas des trois axes, nous avons à nouveau un choix de 48 possibilités de stockages différentes. on aura donc par exemple pour un positionnement droite à gauche, postérieur à antérieur et inférieur à supérieur l'ordre des voxels sera le suivant :

- Voxels ordonnés de droite à gauche pour stocker une ligne.
- Lignes ordonnées du postérieur vers l'antérieur pour stocker une tranche.
- Tranches stockées à partir de l'inférieur vers le supérieur pour stocker un volume