

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
UNIVERSITE FARHAT ABBAS – SETIF

MEMOIRE

Présenté à la Faculté des Sciences de l'Ingénieur
Département d'Informatique
Pour l'Obtention du Diplôme de

MAGISTER

École doctorale d'informatique STIC
Option : Ingénierie des Systèmes d'Information

Par :

Youssef BOULKHIOUT

THEME

**Architecture multi-agents adaptative pour
la détection et la localisation de structures :
Application à la segmentation d'images**

Soutenu le : 14 avril 2010 devant la commission d'examen :

Dr. BOUKERRAM Abdallah MC à UFAS Président

Dr. MOUSSAOUI Abdelouahab MC à UFAS Rapporteur

Dr. TOUAHRIA Mohamed MC à UFAS Examineur

Promotion 2007-2008

Remerciement

Tout d'abord, louanges à dieu pour nous avoir permis d'avoir les capacités mentales et physiques, nous permettant de réaliser ce modeste mémoire et nous l'implorons afin que notre travail soit couronné de succès.

Je remercie mon encadrant monsieur MOUSSAOUI Abdelouahab, pour avoir initié ce projet et pour ses efforts, ses coups de pousses, ses commentaires constructifs... entre autres.

Un merci chaleureux pour notre Messieurs les membres de jury ayant accepté de juger ce modeste travail. Leur relecture attentive ainsi que leur regard à la fois critique et bienveillant.

Un merci tout particulier

Je viens également de remercier : L. Toumi, M. Semcheddine, B. Bemessahal... ayant par de nombreuses discussions, contribué à certaines idées citées dans ce Mémoire, pour leur support apporté à cette réflexion, pour leur coopération tout au long de la réalisation de ce modeste mémoire, et pour les échanges enrichissants que j'ai pu avoir avec eux

Je remercie donc les différentes personnes avec qui m'a partagé cette passion et qui ont ainsi, indirectement mais indéniablement, contribué à ce mémoire

Je remercie aussi mes enseignants et toute personne qui me a fait apprendre même une lettre depuis la naissance,...

Dédicace

*Je dédie ce modeste résultat à ma chère et tendre mère pour leur
bienveillance durant ces longues années.*

*Pour leur amour, leur patience, leur courage et leur sacrifice. Leur affection et
leur soutien.*

🚩 *A mes frères. Brahim, AbdEnnour, Hmida, SidAli, Mounir et.*

🚩 *A mes sœurs. Rabeia, Amina, Fatiha, Loubna et Imane*

🚩 *A ma grande mère et mon grand père.*

A toute la famille.

🚩 *A tous mes Amis, Abderrahim, Omar, Lazher, Mehfoud, Mourad,
Djamel, Eldjamai, Elnouri, Abdelgani,.....*

🚩 *A tous ceux que j'aime*

Et qui m'aiment...

Où qu'ils soient.

Sommaire :

Introduction générale-----	1
CHAPITRE 1 : L'IRM ET L'ANATOMIE CEREBRALE-----	4
Introduction-----	5
1. Quelques éléments d'anatomie cérébrale -----	5
1.1. Le système nerveux central -----	5
<i>1.1.1. La moelle épinière</i> -----	5
<i>1.1.2. Le tronc cérébral</i> -----	6
<i>1.1.3. Le cerveau :</i> -----	7
1.2. Les principaux tissus cérébraux -----	7
<i>1.2.1. Le liquide céphalo-rachidien</i> -----	7
<i>1.2.2. La matière grise</i> -----	7
<i>1.2.3. La matière blanche</i> -----	9
1.3. Le système ventriculaire -----	9
2. Observation du cerveau-----	10
3. IRM (Imagerie par Résonance Magnétique)-----	11
3.1. Principe générale -----	11
3.2. Principe physique -----	12
<i>3.2.1. Champ et moment magnétique</i> -----	12
<i>3.2.2. La Résonance Magnétique Nucléaire (RMN)</i> -----	12
<i>3.2.3. Contraste des images</i> -----	12
3.3. Images pondérées -----	14
<i>3.3.1. L'image pondérée en T1</i> -----	14
<i>3.3.2. L'image pondérée en T2</i> -----	14
<i>3.3.3. L'image de densité de protons ou ρ-pondérée</i> -----	14
3.4. Caractéristiques des images-----	15
3.5. Défauts des images IRM (artefacts) -----	15
<i>3.5.1. Bruit</i> -----	15
<i>3.5.2. Effets de volume partiel</i> -----	15
<i>3.5.2. Mouvement</i> -----	16
<i>3.5.4. Variations du champ magnétique (inhomogénéité RF)</i> -----	16
3.6. Avantages et inconvénients -----	16
Conclusion-----	17

CHAPITRE 2 : METHODES DE SEGMENTATION D'IMAGES IRM	18
Introduction.....	19
1. Définition	19
2. Applications de la segmentation.....	19
3. Approches de segmentation d'images	20
3.1 Approches régions	20
3.1.1. Seuillages	21
3.1.2. Croissance de régions	21
3.1.3. Morphologie mathématique	22
3.2 Approches contours.....	23
3.2.1. Méthodes dérivatives	23
3.2.2. Modèles déformables	24
2.3. Segmentation par classification	25
3.3.1. Méthodes supervisées	25
3.3.1.1. Segmentation bayésienne.....	25
3.3.1.2. Segmentation Markovienne	27
3.3.1.3. La méthode des k plus proches voisins (kppv)	28
3.3.1.4. Réseaux de Neurones	29
3.3.2. Méthodes non supervisées	31
3.3.2.1. Algorithmes de classification non flous	31
3.3.2.2. Algorithme des C-moyennes floues (Fuzzy C- Means « FCM »)	32
3.3.2.3. Algorithme de classification possibiliste	33
3.3.2.4. Algorithme semi-supervisée :	34
2.4 Autres approches	35
2.4.1. Les modèles pyramidaux	35
2.4.2. Les modèles mixtes	35
3. Comparaison entre méthodes de segmentation.....	35
Conclusion.....	37
CHAPITRE 3 : LES METHODES DE SEGMENTATION PAR CLASSIFICATION	38
Introduction.....	39
1. Les machines à vecteurs supports (SVM).....	39
Introduction.....	39
1.1. Notions fondamentales	39
1.1.1. La classification	39

1.1.2. L'apprentissage	39
1.1.3. La séparation	40
1.2. Le principe de fonctionnement des SVM	41
1.2.1. L'hyperplan séparateur	41
1.2.2. La marge	41
1.2.3. Fondements mathématiques	42
1.2.3.1. SVM linéaire	42
1.2.3.2. SVM non linéaire	44
1.3. Les noyaux	45
1.3.1. Définition de noyau	45
1.3.2. Types de noyaux	46
1.3.2.1. Noyaux linéaires	46
1.3.2.2. Noyaux polynomiaux	46
1.3.2.3. Noyaux gaussiens	46
1.3.2.4. Noyaux exponentiels	46
1.3.2.5. Noyaux sigmoïdaux	46
1.4. Classes des SVM	47
1.4.1. Les SVM bi-classes	47
1.4.2. SVM multi-classes	47
1.5. Application à segmentation d'image	48
Conclusion	49
2. Les fourmis artificielles	49
Introduction	49
2.1. Notions fondamentales	49
2.1.1. L'émergence	49
2.1.2. Les phéromones	50
2.1.3. La Stigmergie	50
2.2. Le comportement de fourmis	50
2.2.1. Activités de regroupement	50
2.2.2. Recherche de nourriture	50
2.2.3. Choix de plus courts chemin	51
2.2.4. S'adapter aux changements	51
Le rôle des phéromones :	52
2.3. Modélisation et implémentation	52

2.3.1. <i>Algorithme de retour à la fourmière</i>	53
2.3.2. <i>Algorithme de recherche</i>	54
2.4. L'algorithme <i>AntClustering</i>	55
2.4.1. <i>L'environnement des fourmis</i>	55
2.4.2. <i>Le fonctionnement de l'algorithme AntClustering</i>	55
2.4.2.1. Dépôt du pixel	57
2.4.2.2. Ramassage d'un pixel	57
2.4.2.3. Mémoire des fourmis	58
2.3. Application à segmentation d'image	59
Conclusion	59
CHAPITRE 4 : LES SYSTEMES MULTI-AGENTS	60
Introduction	61
1. Définitions	61
1.1. Agent	61
1.2. Systèmes multi-agents (SMA)	61
1.3. Domaines d'application des SMA	62
2. Typologie des agents	62
2.1. Les agents cognitifs	62
2.2. Les agents réactifs	63
2.3. Les agents hybrides	63
2.4. Les agents fixes	63
2.5. Les agents mobiles	63
3. Les caractéristiques des SMA	63
3.1. Organisation sociale	64
3.2. Coopération	64
3.3. Engagement/Planification	65
3.4. Communication	65
3.5. Adaptativité	65
4. Le développement des SMA	65
4.1. Les méthodologies de développement des SMA	65
4.1.1. <i>La méthodologie AAI</i>	65
4.1.2. <i>La méthodologie GAIA</i>	66
4.1.4. <i>La méthodologie Cassiopea</i>	66
4.1.4. <i>La méthodologie AUML</i>	66

4.1.5. <i>La méthodologie MAS-CommonKADS</i> :-----	67
4.2. Les plates-formes des systèmes multi-agents -----	67
4.2.1. <i>Agent Builder</i> -----	67
4.2.2. <i>Madkit</i> -----	67
4.2.4. <i>Jade</i> -----	68
4.2.4. <i>Zeus</i> -----	68
5. Apports des SMA pour la segmentation d'images-----	68
5.1. La représentation et l'organisation des informations-----	69
5.2. L'extraction de nouvelles informations -----	69
5.3. La focalisation des traitements -----	70
5.4. La coordination et le contrôle de traitements-----	70
5.5. Les approches multi-agents proposées pour la segmentation -----	70
5.5.1. <i>Mise en oeuvre d'un SMA pour la segmentation</i> -----	70
5.5.2. <i>Agents contours/régions</i> -----	71
5.5.3. <i>Image de type quelconque</i> -----	72
5.5.4. <i>Image IRM</i> -----	72
Conclusion-----	73
CHAPITRE 5 : IMPLEMENTATION PAR LES SYSTEMES MULTI-AGENTS -----	74
Introduction-----	75
1. Choix de l'algorithme-----	75
1.1. L'algorithme supervisé SVMs-----	75
1.1.1. <i>Classification linéaire</i> -----	76
1.1.2. <i>Classification non linéaire</i> -----	77
1.1.3. <i>Les noyaux</i> -----	77
1.1.4. <i>Variantes des SVMs</i> -----	78
1.1.4.1. Les SVMs bi-classes -----	78
1.1.4.2. Les SVMs multi-classes -----	78
1.1.5. <i>Les SVMs et imagerie par résonance magnétique</i> -----	79
1.2. L'algorithme non supervisée <i>AntClustering</i> -----	79
1.2.1. <i>Formalisation du problème</i> -----	79
1.2.1.1. L'algorithme principal -----	80
1.2.1.2. Déplacements des fourmis-----	81
1.2.1.3. Ramassage d'un pixel -----	81
1.2.1.4. Dépôt du pixel-----	82

1.2.2. <i>Les paramètres de AntClustering</i>	82
1.2.3. <i>AntClustering et imagerie par résonance magnétique</i>	82
2. Choix des paramètres des algorithmes	83
2.1. Initialisation des algorithmes	83
2.2. Détermination du nombre de classes	83
2.3. Choix des vecteurs forme	83
3. Algorithmes utilisés	84
3.1. Algorithmes SVM	84
3.2. Algorithmes <i>AntClustering</i>	88
3.3. Algorithme général de l'approche	89
4. Pourquoi une architecture multi-agents ?	90
5. Les Plates Formes multi-agents	91
5.1. FIPA (Foundation for Intellegent Physical Agents)	91
5.2. Critères de choix d'une plate forme MA	92
5.3. La plate forme JADE	92
5.3.1. <i>Architecture logiciel de la plate-forme JADE</i>	92
5.3.2. <i>Langage de communication de la plate-forme JADE</i>	93
5.3.3. <i>Comportements des agents dans la plate-forme JADE</i>	94
5.3.4. <i>Outils de JADE</i>	95
5.3.5. <i>Evaluation de JADE</i>	95
5.4. Les avantages de JADE	96
Conclusion	96
CHAPITRE 6 : VALIDATION DES RESULTATS	97
Introduction	98
1. Les images de test	98
1.1. Les images réelles	98
1.2. Les images de synthèse	98
2. Evaluation et étude comparative	100
2.1. Le processus d'évaluation sur image	100
2.2. Le paramètres d'évaluation	100
2.2.1. <i>Taux de recouvrement</i>	100
2.2.2. <i>La distance de Hausdorff</i>	101
2.2.3. <i>Les trois coefficients de validation des systèmes d'imagerie</i>	102
2.2.4. <i>Autres mesures d'évaluation</i>	102

2.3. Résultats -----	103
3. Résultats -----	105
3.1. Images de synthèses-----	105
3.2. Images réelles-----	107
Conclusion-----	113
Conclusion générale-----	114
Références -----	116
Webliographie :-----	123

Liste des figures :

Figure 1.1: Le système nerveux central-----	5
Figure 1.2 : Coupe transversale de la moelle épinière-----	6
Figure 1.3: Le tronc cérébral. -----	6
Figure 1.4: Structures anatomiques de la matière grise.-----	8
Figure 1.4: Différentes structures du cerveau. -----	9
Figure 1.5: Rapports noyaux/ventricules. -----	10
Figure 1.6: Les trois axes de coupe pour la visualisation du cerveau. -----	10
Figure 1.7 : Acquisition par IRM. -----	11
Figure 1.8: L'angle de basculement-----	13
Figure 1.9 : L'inhomogénéité RF. (a) : Image affectée par une inhomogénéité RF, (b) : L'artefact RF isolé, (c) : Image sans artefact -----	16
Figure 2.1 : Image IRM segmentée par une approche région.-----	23
Figure 2.2 : Segmentation, approche contour. (a) : image IRM originale, (b :) image segmentée (approche contour) -----	24
Figure 2.3 : Segmentation par modèle déformable. (a) : Image originale, (b) : Cerveau isolé	25
Figure 2.4 : Segmentation par méthode bayésienne. -----	27
Figure 3.1: Sur apprentissage et sous apprentissage. -----	40
Figure 3.2: Données séparables (et non séparables) linéairement -----	41
Figure 3.3 : Hyperplan. -----	41
Figure 3.4 : Meilleur et mauvais hyperplan. -----	42
Figure 3.5 : Hyperplan séparateur et vecteurs de support. -----	43
Figure 3.6 : Redescription et séparation linéaire. -----	44
Figure 3.7 : Hyperplan optimal généralisé. -----	45
Figure 3.8 : Solution du XOR par changement d'espace.-----	47
Figure 3.9 : Le regroupement des cadavres chez les fourmis.-----	50
Figure 3.10 : Recherche de nourriture. -----	51
Figure 3.11 : L'adaptation aux changements.-----	51
Figure 3.12 : Collecte de nourriture : avec ou sans phéromones. -----	52
Figure 3.13 : Dessin du monde.-----	52
Figure 3.14 : L'environnement des fourmis artificielles -----	55
Figure 4.1: Représentation d'un agent en interaction avec son environnement et les autres agents-----	62

Liste des figures

Figure 5.1. La fonction de similarité pour $ng_i = 128$ et $g_k \in [0,255]$, $\beta = 50$ -----	80
Figure 5.2 : les étapes de l'approche proposée. -----	89
Figure 5.3 : l'interaction entre agents dans l'approche développée.-----	90
Figure 5.4 : Plateforme d'agent de FIPA -----	93
Figure 6.1 : Processus de construction des images simulées. -----	99
Figure 6.2 : Processus d'évaluation sur des images fantômes. -----	100
Figure 6.3 : Recouvrement entre les deux régions A et B -----	101
Figure 6.4 : Définition des valeurs de VP, VN, FP et FN par comparaison de l'image de référence (ensemble A) et de l'image segmentée (ensemble B). -----	102

Liste des tableaux :

Tableau 2.1 : Comparaison entre méthodes de segmentation -----	36
Tableau 4.1 : Comparaison entre les agents cognitifs et les agents réactifs -----	63
Tableau 4.2 : Les modèles de coopération. -----	64
Tableau 5.1 : détermination des noyaux. -----	77
Tableau 5.2 : les paramètres de <i>AntClustering</i> . -----	82
Tableau 5.3 : évaluation de JADE. -----	95
Tableau 6.1 : Paramètres de SVMs et de <i>AntClustering</i> . -----	103
Tableau 6.2: Evaluations de la segmentation sur 08 coupes successives du volume.-----	104
Tableau 6.3 : Comparaison des mesures obtenues par algorithmes. -----	104
Tableau 6.4 : Comparaison des images segmentées par différentes approches de classification -----	106
Tableau 6.5 : résultats de segmentation des images réelles sur différentes coupes, différentes pondérations et par différentes algorithmes. -----	112
Tableau 6.6 : Temps de segmentation en secondes. -----	113

Introduction générale

Le domaine du traitement de l'image est un domaine qui est en pleine expansion et qui pénètre tous les secteurs d'activités, des environnements industriels et professionnels au monde des applications grand public. Cependant, malgré les progrès significatifs réalisés ces dernières années dans les systèmes d'imagerie (depuis la caméra jusqu'aux machines de vision sont de plus en plus performants), et dans tous les secteurs (médicales, industriels,...), l'image reste aujourd'hui encore un sujet difficile et les solutions actuelles ne permettent pas toujours de résoudre de manière satisfaisante de nombreux problèmes importants.

L'analyse d'image par ordinateur passe par la décomposition des problèmes complexes en problèmes plus simple. On fait généralement la distinction entre les traitements de bas niveau et les traitements de haut niveau. Les premiers travaillent directement sur les valeurs attachées aux pixels de l'image, tandis que les seconds opèrent sur des entités symboliques constituant une interprétation de la réalité extraite de l'image.

Le travail réalisé dans cette thèse concerne la partie bas niveau et plus particulièrement la phase de segmentation destinées à extraire la forme d'un objet présent dans une image or la détection et la localisation des structures dans une image.

Dans ce travail on a essayé, de mettre en oeuvre les méthodes de segmentation d'image par classification en se concentrant sur la méthode supervisée basé sur *les machines à vecteurs de supports (SVMs)* et la méthode non supervisée basé sur les fourmis artificiels *AntClustering*. D'après l'étude détaillée des différentes méthodes de segmentation des images, on a décelé des inconvénients et des avantages pour chacune. Ce qui nous a permis d'affirmer qu'il n'existe pas de méthode de segmentation idéale malgré les améliorations apportées. Les méthodes de segmentation supervisées difficiles à mettre en œuvre car ils ne sont utilisables que dans des contextes limités et demandent l'intégration d'informations *a priori* sur les zones recherchées. Par contre les approches non supervisées reposent sur des fondements théoriques et mathématiques et ne nécessitent aucune information *a priori* sur les zones recherchées.

Le choix du domaine d'application n'a pas été le sort de l'improvisiste mais l'apanage des différents avantages de l'IRM. Il s'agit d'une image à très haute résolution issue d'un signal non invasif qui reproduit exactement l'état de la matière, dont on peut capter et interpréter, il est donc possible de mesurer les signaux par rapport à une table des signaux que les médecins connaissent, d'ailleurs on parle d'hypo ou hyper signal en matière d'IRM.

La prise en considération des différentes structures du corps humain sa prendre du temps dont le but n'est pas segmenter tous le corps humain mais d'appliquer une méthode sur une région précise, de cet effet on a limiter l'application au cerveau humain qui est la région la

mieux explorer par l'IRM. Le cerveau humain est un ensemble de tissus composés de cellules (neurones et cellules accessoires), organisé en structures anatomiques et fonctionnelles. Seulement pour notre étude on a choisis de limiter le travail pour trois structures qui ont une interprétation significative dans l'image IRM : Matière grise, Matière Blanche et Liquide cérebrospinal. Ces trois substances représentent en IRM trois structures avec un signal différent, ainsi que la répartition anatomique bien individualisé nous permet de vérifier dans un autre temps l'exactitude des résultats sur un atlas ou auprès d'un expert dans la matière (radiologue).

La mise en oeuvre de notre approche se fait sur des fondements de l'Intelligence Artificielle Distribuée (IAD), c'est-à-dire principalement aux *systèmes multi-agents* (SMA), qui sont des ensembles d'entités autonomes et adaptatives (*les agents*) qui collaborent de façon décentralisée, pour résoudre un problème donné (ici l'interprétation d'une image). Le caractère distribué du processus d'interprétation d'une image, en termes de buts à atteindre, de zones de l'image à explorer et de méthodes à appliquer, fait en effet naître certains besoins relatifs à la coordination des traitements ou à la gestion des conflits, pour lesquels les systèmes multi-agents proposent des solutions en termes de formalismes, de mécanismes de coordination et de communication, de méthodologies de conception, d'outils d'Interface Homme/Machine...

Le mémoire est organisé de la façon suivante :

1. Le premier chapitre présente le cadre applicatif qui a servi de support à nos recherches. Il rappelle quelques notions d'anatomie du cerveau humain, les principes fondamentaux de la technique de l'imagerie par résonance magnétique, ainsi que les différents paramètres qui ont influence sur la qualité des images IRM.
2. Le second chapitre propose un état de l'art sur la segmentation en IRM cérébrale. Il est organisé en trois parties. Les approches région puis les approches contours et enfin, les approches procédant par classification. Pour chaque approche, on présente les différentes méthodes et techniques utilisées tout en évoquant les avantages et les inconvénients.
3. Le troisième chapitre présente les différentes méthodes de segmentation par classification qui ont été introduites dans le système développé. Elles sont au nombre de deux. La première est la méthode qui se base sur les machines à vecteurs de supports (SVMs) pour la détection des contours. La deuxième est une méthode qui se base sur les fourmis artificielles (*AntClustering*) pour la détection des régions.
4. Le quatrième chapitre présente le paradigme multi-agents, où on introduit pour cela quelques notions de base sur les systèmes multi-agents et sur la conception de systèmes

ditions de segmentation, tout en montrant comment ces notions peuvent être appliquées à la segmentation des IRM.

5. Le cinquième chapitre décrit l'ensemble des processus de l'approche proposée par les SVMs et *AntClustering* et précise les techniques mises en oeuvre pour réaliser un système de caractérisation des structures ou bien dite de segmentation des IRM.
6. Le sixième chapitre propose une validation de l'ensemble de l'approche proposée à travers l'utilisation d'images simulées développées au Montréal Neurological Institute. Il présente également des résultats obtenus sur des images réelles.

Une conclusion et perspectives de ce travail seront présentées à la fin du mémoire.

CHAPITRE 1 : L'IRM ET L'ANATOMIE CEREBRALE

Introduction

L'objectif de ce chapitre est de présenter quelques éléments d'anatomie du système nerveux central, afin de préciser le vocabulaire et les notions qui seront utilisés dans de ce mémoire. Puis, nous présentons brièvement les principes physiques de l'IRM (l'imagerie par résonance magnétique) en insistant plus particulièrement sur les difficultés qui découlent de cette technique d'acquisition.

1. Quelques éléments d'anatomie cérébrale

1.1. Le système nerveux central

Le système nerveux central (figure 1.1) est constitué d'un réseau complexe de communications dont les influx règlent le fonctionnement des organes. Il est principalement formé de substance blanche et de substance grise. Il joue un rôle essentiel dans le comportement conscient de l'individu que pour les processus inconscients [JAG 98].

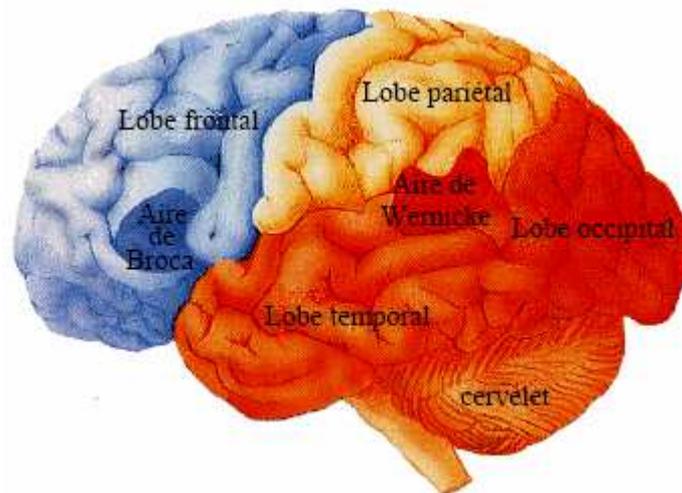


Figure 1.1: Le système nerveux central [JAG 98].

L'anatomie le décompose en trois parties : la moelle, le tronc cérébral et le cerveau.

1.1.1. La moelle épinière

La moelle épinière est la partie de système nerveux central situé dans le canal rachidien. Elle présente la forme d'une tige cylindrique enveloppée par des méninges. La substance blanche est formée de fibres axoniques. La substance grise de la moelle épinière dessine une forme de papillon autour du canal de l'épendyme, nom du canal central dans la moelle épinière. les cornes latérales de la substance grise sont représentées en bleu (figure 1.2), au milieu se trouve le canal épendymaire qui est relié aux cavités du cerveau remplies de liquide céphalo-rachidien.

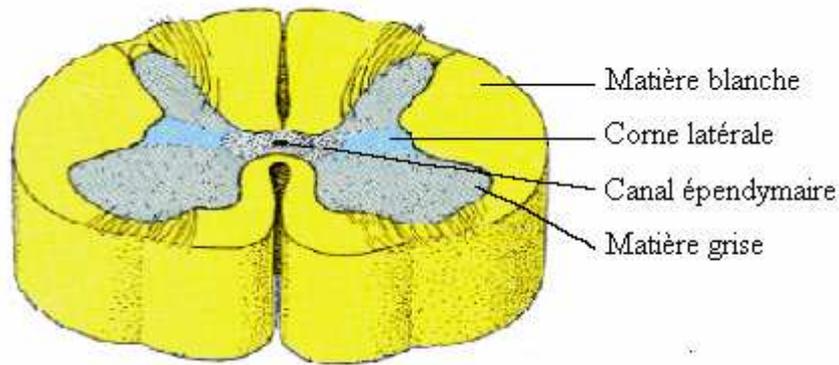


Figure 1.2 : Coupe transversale de la moelle épinière. [JAG 98]

1.1.2. Le tronc cérébral

Le tronc cérébral (figure 1.3) se compose de trois parties : le myélocéphale (ou moelle allongée ou bulbe rachidien), le mésencéphale (ou isthme) et le métencéphale (pont ou protubérance). Il est formé de substance blanche externe et de substance grise interne morcelée en de multiples noyaux de substance grise ou noyaux gris. Le tronc cérébral a donc un rôle conducteur mais il est également responsable de certaines fonctions inconscientes liées à la déglutition et à la respiration.

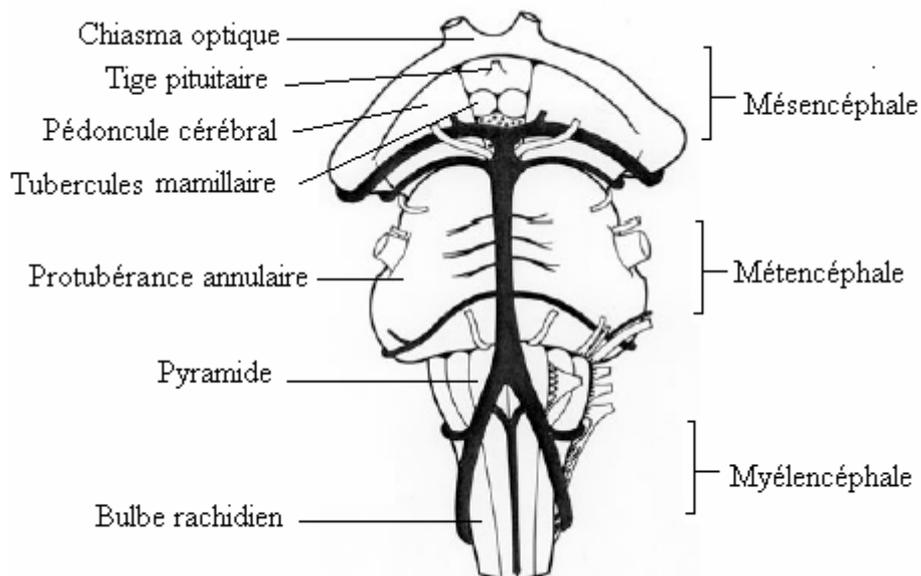


Figure 1.3: Le tronc cérébral.

Le cervelet, également constitué de substance blanche et de substance grise, est relié à chacune de ses trois parties par des pédoncules. Il possède trois lobes, comportant de nombreux sillons de plus petites tailles que ceux du cerveau, et a la charge du maintien général de la coordination (contractions musculaires et équilibre du corps). Il dispose, à sa surface, d'un cortex d'environ 1 mm d'épaisseur [JAG 98].

1.1.3. Le cerveau :

Dans le Petit Larousse le cerveau est la partie antérieure de l'encéphale formée des hémisphères cérébraux et des structures qui les unissent. Le cerveau est la partie la plus volumineuse et l'étage le plus élevé dans la hiérarchie fonctionnelle du système nerveux central, son poids moyen varie entre 1400 à 1800 grammes. Il est placé dans la boîte crânienne mais n'est pas directement au contact de la cavité osseuse. Il se situe dans une enceinte liquidienne (*Liquide Céphalo-rachidien*) qui a la particularité de pénétrer également à l'intérieur du cerveau dans les cavités du *système ventriculaire*. Il est constitué de deux hémisphères principaux. Les hémisphères sont reliés par différentes structures cérébrales comme *le corps calleux* ou *le thalamus*. Il est composé de trois tissus principaux (la Matière Blanche (MB), la Matière Grise (MG), le Liquide Céphalo-rachidien (LCR) [SEB 03].

1.2. Les principaux tissus cérébraux

1.2.1. Le liquide céphalo-rachidien

Le liquide céphalo-rachidien ou cérébro-spinal est un liquide clair, incolore, entoure le cerveau, remplit les ventricules cérébraux et l'espace sous-arachnoïdien. Il soutient le tissu nerveux ainsi protégé des coups, des frottements et des compressions. Le liquide céphalo-rachidien remplit par ailleurs des fonctions importantes, il permet de protéger le cerveau des chocs en jouant un rôle d'amortisseur, et remplit des fonctions importantes dans les échanges entre le sang et les tissus nerveux [JAG 98], [SEM 08].

1.2.2. La matière grise

La matière grise est le cœur du traitement de l'information nerveuse, elle se trouve principalement au niveau du cortex cérébral, du cortex cérébelleux, dans les noyaux gris centraux (figure 1.4), et dans les noyaux du tronc cérébral et à l'intérieur de la moëlle épinière [SEM 08].

Le cortex : Le cortex recouvre la totalité du cerveau, est caractérisé par de nombreuses dépressions appelées sillons. Il est formé de cellules, dont le nombre est estimé à plus de dix milliards, et qui sont disposées en couches superposées. Son épaisseur moyenne au sommet d'une circonvolution est de 2,6 mm, son épaisseur maximale pouvant atteindre plus de 4,5 mm. Le cerveau (télencéphale) est séparé en deux hémisphères par un sillon profond, les sillons les plus profonds, sillons primaires, définissent les lobes cérébraux (frontal, pariétal, temporal et occipital). Les moins profonds, sillons secondaires, délimitent les circonvolutions ou gyri et sont plus variables que les sillons primaires. Les sillons tertiaires finalement n'obéissent à aucune systématisation et sont différents d'un cerveau à l'autre [NAT 04].

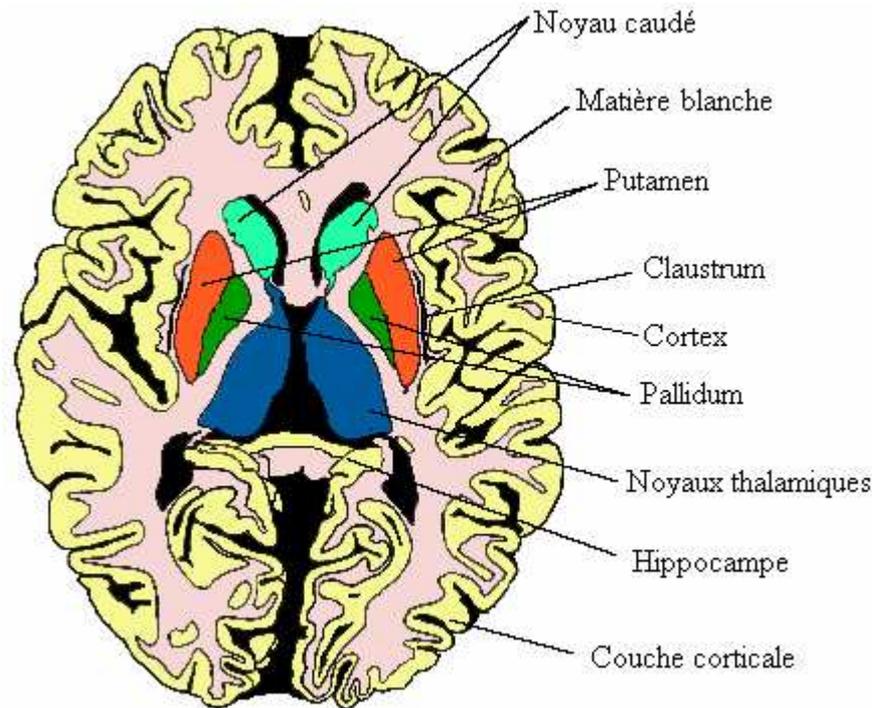


Figure 1.4: Structures anatomiques de la matière grise.

- **Les noyaux de base** : Les noyaux centraux, qui sont avec le cortex les seules structures de substance grise du cerveau, sont également formés de corps cellulaires neuroniques mais avec une densité moins importante que dans le cortex. Ils sont composés des noyaux du télencéphale, ces noyaux sont appelés les noyaux de base (ou corps strié), parmi eux, on distingue les noyaux caudés et les noyaux lenticulaires.
 - **Le noyau caudé** : En forme de virgule à grosse extrémité il est presque complètement enroulé autour du thalamus. Il longe à peu près dans toute son étendue le ventricule latérale. On lui distingue trois parties : la tête, le corps et la queue [GER 99].
 - **Noyaux lenticulaires** : Le noyau lenticulaire se situe en dehors du noyau caudé et du thalamus. Sa forme est celle d'une lentille biconvexe, triangulaire sur les coupes axiales et coronales. Le noyau lenticulaire se compose du putamen (externe), et du pallidum (interne).
- **Les noyaux du diencephale**: parmi ces noyaux :
 - **Le thalamus** : le thalamus est une masse grise qui flanque la cavité du troisième ventricule, aboutissent toutes les sensations avant qu'elles soient projetées vers la conscience. Le thalamus est connecté aux centres moteurs et coordinateurs [SEM 08].
 - **L'hypothalamus** : l'hypothalamus est la paroi inférieure du troisième ventricule, se prolonge jusqu'à l'hypophyse. De petite dimension, il a la charge des équilibres physiologiques du corps. Les nouveaux atlas associent dorénavant le pallidum au groupe de noyaux gris du diencephale.

1.2.3. La matière blanche

C'est la partie interne du cerveau, elle est constituée de fibres à myéline, organisées en fibres d'association reliant deux régions d'un même hémisphère, en fibres commissurales inter hémisphériques passant par *le corps calleux* et *le trigone cérébral* et en fibres de projection entre les différentes parties du corps et le télencéphale. Toutes ces fibres, sensibles ou motrices, aboutissent aux centres nerveux. Ces centres sont les corps striés, les noyaux gris du diencephale, le cervelet, le tronc cérébral et la moelle épinière [JAG 98].

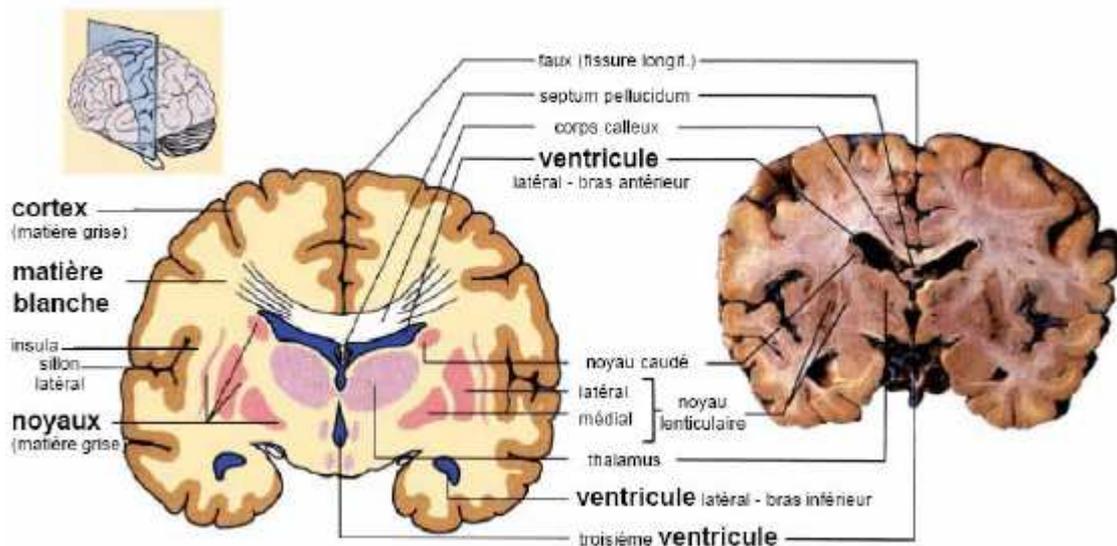


Figure 1.4: Différentes structures du cerveau. [SEM 08]

1.3. Le système ventriculaire

Le système ventriculaire (figure 1.5) comporte quatre cavités qui communiquent entre elles : deux ventricules latéraux de part et d'autre du plan inter hémisphérique et les troisième et quatrième ventricules. Le liquide céphalo-rachidien passe de l'un à l'autre par le trou de Monro et l'aqueduc de Sylvius. Le quatrième ventricule se prolonge jusqu'au canal épendymaire et possède trois connexions avec l'espace sous-arachnoïdien [JAG 98].

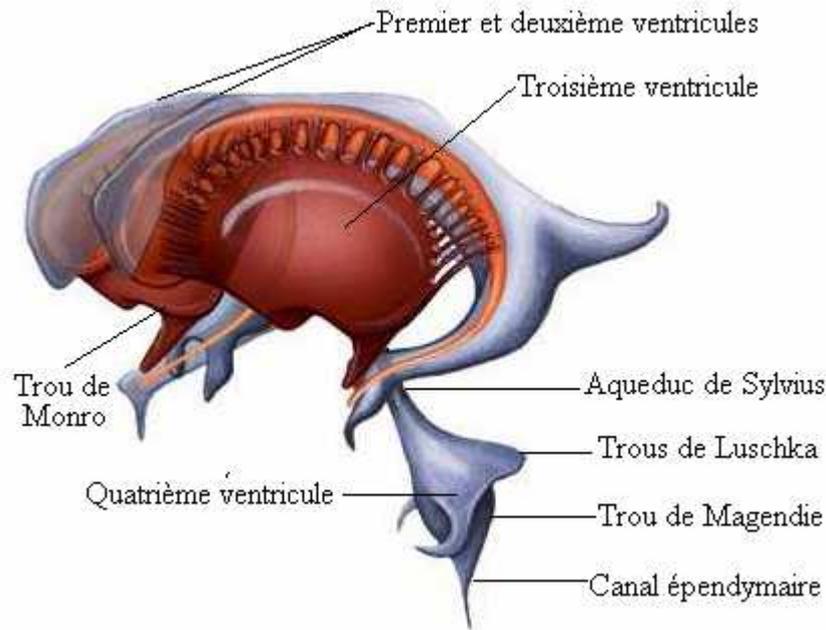


Figure 1.5: Rapports noyaux/ventricules.

2. Observation du cerveau

L'observation par des coupes en deux dimensions du cerveau peut être effectuée selon plusieurs angles de vue. Ainsi, on distingue trois axes anatomiques pour réaliser les différentes coupes (figure 1.6) [GER 99].

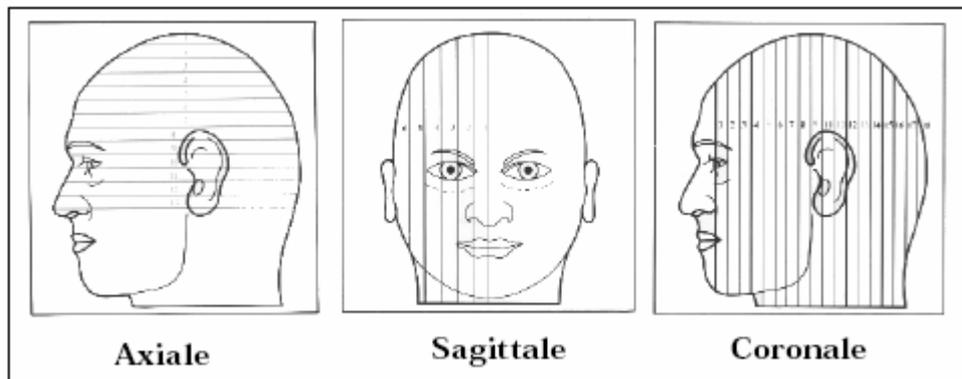


Figure 1.6: Les trois axes de coupe pour la visualisation du cerveau. [GER 99]

- a) **Coupes axiales** : Ces coupes correspondent quasiment à un plan horizontal. En imagerie par résonance magnétique, elles correspondent à un plan perpendiculaire à l'axe du champ magnétique principal.
- b) **Coupes sagittales** : Ces coupes sont prises dans des plans parallèles au plan inter hémisphérique. Il s'agit des vues latérales du cerveau.
- c) **Coupes coronales** : Se sont des coupes perpendiculaires aux coupes axiales et sagittales.

3. IRM (Imagerie par Résonance Magnétique)

L'imagerie par Résonance magnétique est une technique récente qui peut fournir des informations sur la composition des tissus par la production des images anatomiques dans des plans multiples. La première acquisition d'IRM a été produite en 1973 par Lauterbur et Damadian. Cette technique a permis d'arriver à des avancées technologiques et scientifiques majeures, en particulier concernant les états pathologiques de l'homme [SEM 05].



Figure 1.7 : Acquisition par IRM.

3.1. Principe générale

La technique d'IRM utilise la résonance des *noyaux d'hydrogène* lorsque, placés dans un *champ magnétique*, ils sont excités par une *onde radiofréquence*. Les atomes d'hydrogènes peuvent être considérés comme des petits *dipôles magnétiques* dont l'orientation spatiale est aléatoire en dehors d'un champ magnétique significatif. Placés dans un aimant, ces dipôles s'orientent tous dans le même sens en fonction de la direction du champ magnétique créé. Deux orientations sont possibles : un *état stable* de faible énergie, que va adopter la grande majorité des dipôles et un *état instable* d'énergie supérieure, correspondant à la direction opposée, que va adopter, au hasard, une minorité des dipôles.

L'adjonction d'un *rayonnement électromagnétique* de fréquence bien précise va faire entrer en *résonance* les atomes d'hydrogène qui vont se retourner et passer de l'état stable à un état instable. L'arrêt du rayonnement entraîne une *relaxation* qui se caractérise par un retour progressif à l'état stable en même temps que l'émission d'une onde radiofréquence qui constitue le *signal RMN*. Le signal RMN dépend de *la structure moléculaire* dans laquelle est impliqué l'atome d'hydrogène en question. Les informations recueillies sont transmises et analysées par un ordinateur qui reconstruit la topographie du signal RMN dans la structure observée. Donc le signal RMN émis par les protons, lors de leur retour à leur état de base, est à l'origine de la création des images. Les images obtenues sont traitées par de puissants logiciels qui permettent de reconstruire des coupes transversales, sagittales et longitudinales. Ces images sont visualisées sur un écran puis transformées en clichés radiographiques [BAR 02].

3.2. Principe physique

3.2.1. Champ et moment magnétique

Le moment magnétique est une grandeur vectorielle caractérisée un circuit électrique qui permet d'étudier son comportement dans un champ magnétique. Ainsi, un électron qui décrit une orbite autour d'un noyau constitue une boucle de courant, de moment magnétique $\vec{\mu}$ perpendiculaire à l'orbite. Lorsqu'un proton est placé dans un champ magnétique \vec{B}_0 statique, son moment magnétique est animé d'un mouvement de précession (figure 1.8) autour de l'axe de vecteur directeur \vec{B}_0 et sa fréquence de rotation est exprimée par l'équation $f_0 = (\gamma/2\pi)\vec{B}_0$, où γ est le rapport gyromagnétique spécifique au noyau étudié. Dans le cas du proton, la mécanique quantique n'autorise que deux états possibles de spin $(-1/2, +1/2)$, $\vec{\mu}$ sera donc orienté soit parallèlement à \vec{B}_0 , soit dans la direction opposée à \vec{B}_0 .

Comme la population ayant un spin de $1/2$ est très légèrement supérieure à la population ayant un spin de $-1/2$, le moment magnétique résultant \vec{M} sera orienté dans le sens de \vec{B}_0 ; sa composante en Z, \vec{M}_Z est petite devant \vec{B}_0 . Pour mesurer ce moment magnétique il faut donc le basculer dans le plan transverse XY. Pour cela, on applique une onde radiofréquence (RF) égale à f_0 et perpendiculaire à \vec{B}_0 [GER 99].

3.2.2. La Résonance Magnétique Nucléaire (RMN)

La résonance magnétique nucléaire (RMN) est une technique fondée sur l'échange d'énergie entre les spins nucléaires des atomes ayant un nombre impair de nucléons et un système de bobinage. En présence d'un champ magnétique intense \vec{B}_0 , les spins, considérés comme des moments magnétiques tournant autour d'un axe, sont répartis sur au moins deux niveaux d'énergie distincts. Les spins peuvent passer à un niveau énergétique supérieur en absorbant l'énergie fournie par un champ magnétique RF, à leur fréquence de précession, qui est généré par le système de bobinage. La relaxation des spins vers un niveau d'énergie inférieur produit une énergie magnétique captée par le système de bobinage sous forme d'un signal « RMN » [NAT04].

3.2.3. Contraste des images

En IRM, le signal étudié est l'énergie libérée par l'ensemble des protons lors de leur retour à l'état stable dont le paramètre d'excitation RF joue un rôle essentiel car la composante transverse \vec{M}_{XY} dépend de la durée et de l'amplitude de la RF. Les images résultant de la mesure

de l'aimantation \vec{M} en tout point d'un volume dépendent principalement de trois paramètres intrinsèques :

- T_1 : (temps de relaxation longitudinale ou encore temps de relaxation spin-réseau) C'est le temps nécessaire au retour de \vec{M} dans la direction de \vec{B}_0 .
- T_2 : (temps de relaxation transverse ou temps de relaxation spin-spin) C'est le temps de relaxation qui décrit le retour à 0 de la composante \vec{M}_{xy} de \vec{M} .
- ρ : La densité de protons, qui définit la quantité de signal considéré.

Des paramètres extrinsèques liés à la séquence d'acquisition utilisée sont essentiels :

1. TR : (temps de répétition) représente le temps qui sépare deux excitations RF successives. Si TR est long ($>3s$), tous les tissus auront le temps de revenir à leur état initial et le contraste sera quasiment indépendant de T_1 . Si au contraire, TR est court, on observera des différences de contraste entre les tissus ayant un T_1 court qui pourront revenir à l'équilibre, et les tissus ayant un T_1 long qui ne reviendront pas à l'équilibre (3s pour le liquide cérébro-spinal). Le contraste des images sera donc dépendant des T_1 respectifs des tissus étudiés. Ce contraste est souvent utilisé pour construire des images anatomiques.
2. TE : (temps d'écho) représente le temps qui sépare l'impulsion RF du maximum du signal de résonance magnétique détecté. Ce signal est appelé écho. Si TE est court ($<20s$), la décroissance du signal issue des différents tissus est négligeable et le contraste est quasiment indépendant de T_2 . Si TE est long ($>50s$), le signal sera plus élevé pour les tissus ayant un T_2 plus long.
3. α représente l'angle de basculement de \vec{M} dans le plan XY [GER 99].

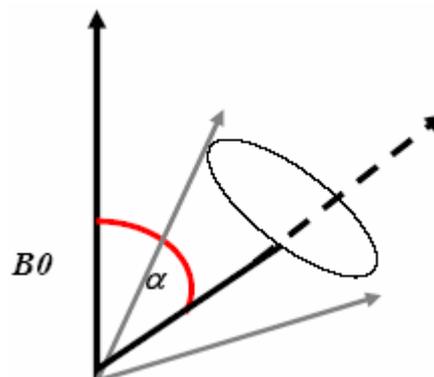


Figure 1.8: L'angle de basculement

3.3. Images pondérées

3.3.1. L'image pondérée en T_1

L'image d'IRM pondérée en T_1 est créée typiquement en employant des temps courts de TR et de TE , dont le contraste est principalement déterminé par des différences de relaxation T_1 . Par la suite, nous désignerons par T_1 cette image pondérée en T_1 [DOU 02]. Si les paramètres de la machine sont choisis avec un TR égal ou inférieur au T_1 (typiquement < 500 ms) dans le produit des fonctions exponentielles on peut négliger le second terme en exp^{-t/T_2} et donc $\vec{M}_{xy} = \vec{M}_{xy}(0) \frac{TR}{T_1}$, Ainsi l'expression devient peu dépendante du T_2 et dépendante surtout du TR et du T_1 .

3.3.2. L'image pondérée en T_2

Une image d'IRM pondérée en T_2 est créée typiquement en employant un TR nettement plus grand que T_1 (typiquement ≥ 2000 ms) et TE plus grand ou égal au T_2 (typiquement ≥ 100 ms), le contraste d'image T_2 est pondéré avec un long TR comparé au T_1 du tissu (pour réduire la contribution T_1 au contraste de l'image) et un TE compris entre le plus long et le plus court T_2 du tissu d'intérêt. Pour un TR 3 fois à 5 fois plus grand que le plus long des T_1 les différences liées à T_1 sont inférieures à 5%. En raison de l'éventail des valeurs de T_1 , de T_2 et de densité de protons et des tissus qui peuvent être trouvés dans le corps, une image qui est pondérée T_2 pour certains tissus ne l'est pas forcément pour d'autres. La graisse a un temps de T_2 court et se relaxe plus rapidement que l'eau. Puisque la quantité de magnétisation transversale en graisse est petite, la graisse produit un signal très petit sur une image de contraste T_2 et semble foncée. Le T_2 prend davantage d'importance avec un plus long TE . L'eau a une valeur très élevée de T_2 , et donc un signal très élevé en T_2 et semble plus en hypersignal sur une image de contraste T_2 .

3.3.3. L'image de densité de protons ou ρ -pondérée

Une image d'IRM pondérée en ρ est créée lorsque en réduire au minimum les effets du T_1 et du T_2 , pour obtenir une image pondérée en densité de protons, il faut utiliser une valeur de TR (typiquement ≥ 2000 ms) beaucoup plus grande que le temps T_1 spécifique du tissu, et une valeur de TE (typiquement ≤ 30 ms) inférieure au temps T_2 spécifique du tissu considéré.

Le contraste en densité de protons est quantitativement dépendant du nombre de protons par unité de volume. Plus le nombre de protons dans une unité du tissu donnée est élevé, plus la composante transversale de l'aimantation est grande, et plus le signal sur l'image de contraste en

densité de protons est en hypersignal. Réciproquement plus le nombre de protons dans une unité donnée du tissu est bas, moins la magnétisation transversale est importante et plus le signal sur l'image de densité de protons est en hypersignal [DOU 02].

3.4. Caractéristiques des images

Les images d'IRM sont des coupes de la région étudiée reconstruites par l'ordinateur dans le plan demandé. Différentes *séquences d'excitation et de relaxation* peuvent être créées pour faire varier les conditions d'émission du signal, contrairement à la radiologie, l'intensité du signal n'est pas toujours la même *et l'aspect des structures varie en fonction des séquences utilisées*. Le liquide peut être blanc sur une séquence et apparaître noir sur une autre. *La résolution en contraste des images d'IRM est excellente* et en fait tout son intérêt. L'utilisation rationnelle de séquences différentes permet d'explorer à peu près tous les tissus organiques à condition qu'ils *contiennent de l'eau*. Seul l'os cortical émet un signal très faible quelles que soient les séquences. *La résolution spatiale* des images dépend de l'intensité du champ magnétique et de la forme de l'antenne utilisée pour émettre et recevoir les ondes radiofréquence. L'antenne est placée sur le patient le plus proche de la région à étudier [BAR 02].

3.5. Défauts des images IRM (artefacts)

Dans l'imagerie par résonance magnétique, un certain nombre d'autres facteurs viennent affecter la qualité des images. Les erreurs (artefacts) d'acquisition en IRM sont de natures différentes de celles observées dans d'autres domaines du traitement d'images. On distingue essentiellement quatre effets: le bruit, le mouvement, les variations de champ et les effets de volume partiel [GER 99].

3.5.1. Bruit

Le bruit a des origines multiples, liées en partie au bruit de l'appareillage. Dans les images par résonance magnétique, l'objectif est d'augmenter le contraste entre les tissus tout en conservant une bonne résolution et un rapport signal/bruit élevé, ces caractéristiques sont cependant contradictoires et il est nécessaire de trouver un bon compromis entre résolution et bruit. Ainsi, on peut doubler la taille des pixels pour multiplier le rapport signal/bruit d'un facteur p , mais la résolution est divisée par deux. Le choix d'acquisition est donc un facteur déterminant [SEM 08].

3.5.2. Effets de volume partiel

Cet effet se manifeste principalement à la frontière des principales matières (matière blanche/matière grise et liquide céphalo-rachidien/matière grise). Si la surface entre plusieurs objets intersectées un pixel, la mesure radiométrique sur ce pixel résulte d'un mélange des

contributions radiométriques des objets présents dans ce pixel, Il se produit également lorsque des structures sont trop fines pour être visibles à la résolution de l'image (par exemple pour certains vaisseaux sanguins) [NAT 04].

3.5.2. *Mouvement*

Le mouvement diminue la qualité de l'image et pose des problèmes d'interprétation, il peut provenir de plusieurs sources. Il peut être lié au métabolisme comme la circulation sanguine ou la respiration (déplacement chimique) et peut également être lié au mouvement du patient pendant l'acquisition. Le mouvement les mouvements de la tête, sont responsables d'artefacts dans les IRM cérébrales [SEM 08].

3.5.4. *Variations du champ magnétique (inhomogénéité RF)*

Les variations de champ ont pour conséquence une variation des intensités d'un même tissu dans une direction quelconque de l'image. Ce phénomène est dû au fait que le champ magnétique n'est pas parfaitement homogène spatialement et temporellement pendant une acquisition. Il existe de plus des non-linéarités de gradient de champ magnétique.

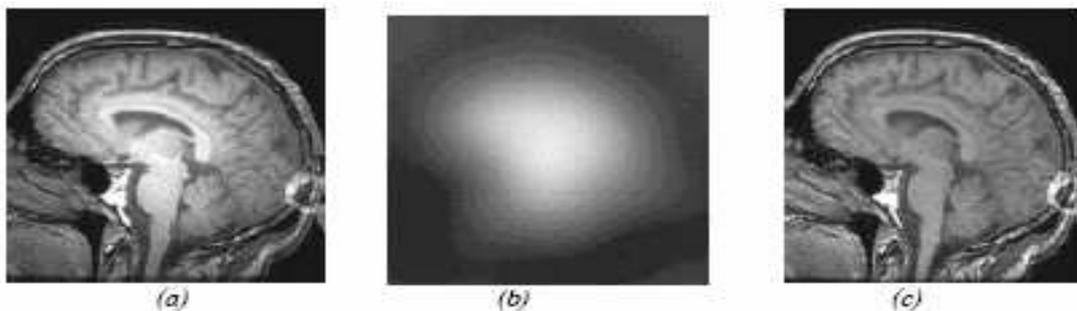


Figure 1.9 : L'inhomogénéité RF. (a) : Image affectée par une inhomogénéité RF, (b) : L'artefact RF isolé, (c) : Image sans artefact [SEM 08]

3.6. Avantages et inconvénients

+ Les avantages :

L'IRM donne dans la plupart des cas des images de meilleure qualité et contenant plus d'informations anatomiques que les autres méthodes d'imagerie médicale, au moins dans l'étude des organes fixes tels que le cerveau ou les membres. Cela est dû à l'*excellente résolution en contraste* qui en fait le système d'imagerie le plus performant dans ce domaine, et l'*absence de superposition* des structures caractéristiques des systèmes d'imagerie en coupe [BAR 02].

– Les inconvénients :

Parmi leurs inconvénients en site : Le coût très élevé des appareils haut champ et de leur maintenance rend la procédure très onéreuse pour le propriétaire. La durée des examens est encore majorée en médecine vétérinaire par le temps d'anesthésie et de mise en place du

patient, ce qui majore encore le coût de l'examen et limite fortement la rentabilité de l'appareil. Les appareils bas champ sont souvent limités par *la résolution spatiale* et le *bruit* important des images dues à un manque d'intensité du signal. La faiblesse du champ magnétique est alors compensée par une acquisition plus longue qui augmente la durée de l'examen [BAR 02].

Conclusion

Nous avons présenté dans ce chapitre les notions essentielles liées à l'anatomie du cerveau ainsi que les principes d'acquisition des images par résonance magnétique qui apporte une aide consistante en termes de diagnostic précoce et de caractérisation des tissus cérébraux.

On a remarqué qu'il y a des problèmes liés à l'anatomie du cerveau humain du fait qu'elle est compliquée dans le sens où elle présente de nombreuses structures cérébrales enchevêtrées, très diverses dans leurs formes et dans leurs compositions tissulaires. On peut se rendre compte des difficultés que rencontre le praticien (hétérogénéité des tissus, bruit, l'effet du volume partiel, ...etc.), pour cela le médecin doit nécessairement connaître avec exactitude les changements survenus, donc, il est parfois nécessaire d'analyser plusieurs images avant d'aboutir à la décision finale. Pour cela plusieurs méthodes de segmentation sont proposées dans la littérature.

CHAPITRE 2 : METHODES DE SEGMENTATION D'IMAGES IRM

Introduction

Dans ce chapitre, nous allons décrire les trois approches de segmentation d'image. Dans un premier temps nous présenterons de façon succincte l'approche *région* et l'approche *contour* de la segmentation, puis nous présentons de façon un peu plus précise l'approche de segmentation par *classification*. Dans un second temps, nous présentons brièvement les principaux algorithmes pour chaque une en insistant plus particulièrement sur les difficultés qui découlent de ces techniques.

1. Définition

La segmentation est un traitement de bas niveau qui consiste à créer une partition de l'image observée en un certain nombre de régions selon un critère déterminé (couleur, texture, niveau de gris...) qui pourront par la suite être traitées de manières différentes [GER 99]. Lorsque les informations traitées se composent de plusieurs images ou volumes obtenus sous différents paramètres d'acquisition, le traitement est appelé multimodal par opposition au traitement monomodal.

La segmentation fait référence aux notions de différence et de similarité perçues par le système visuel humain. Ces notions ont donné naissance à deux approches principales : l'approche contour et l'approche région. L'approche classification est une autre approche qui se base sur des fondements mathématiques, et d'autres approches moins utilisées telles que les contours actifs, les modèles pyramidaux et les modèles mixtes [AME 05].

2. Applications de la segmentation

Les principales applications de la segmentation d'image sont la morphométrie, la cartographie fonctionnelle et la visualisation.

1. *La morphométrie* est la mesure quantitative des positions, formes et tailles de structures cérébrales, elle nécessite la segmentation préalable de ces structures. Elle peut permettre d'identifier, comprendre et suivre la progression des pathologies comme *L'autisme*, *l'Alzheimer* ou la *schizophrénie*.
2. L'essentiel de l'activité fonctionnelle mesurable est situé dans le cortex dont la topologie est bidimensionnelle, la segmentation de la structure bidimensionnelle est alors nécessaire pour localiser les signaux, pour les cartographier et pour les visualiser.
3. *La visualisation* des structures anatomiques (par exemple en chirurgie assistée), se fait naturellement à l'aide de la visualisation de leurs surfaces extérieures. Il est alors nécessaire de segmenter ces structures.

Toutes ces applications permettent de définir des objectifs et des propriétés primordiales pour un système de segmentation. Dans le domaine d'imageries cérébrales, segmenter ou

délimiter les structures cérébrales est une étape fondamentale pour l'analyse. L'objectif important est l'automatisation complète car la segmentation purement manuelle est extrêmement laborieuse. Le changement de comportement par l'opérateur humain, la fatigue le fera commettre des erreurs et le résultat manque de fiabilité et de robustesse. Donc l'automatisation complète de la segmentation paraît indispensable pour le traitement des images médicales [SEM 08].

La segmentation de l'IRM cérébrale présente des particularités par rapport à d'autres domaines d'applications de segmentation, ces particularités sont liées aux spécificités des objets à segmenter (l'anatomie du cerveau) et au processus d'acquisition IRM caractérisées par:

- **Volume partiel** : le terme *volume partiel* évoque une vision d'un pixel qui contiendrait un mélange de plusieurs tissus, le pixel à ce niveau est appelé *mixel*. Donc le terme *volume partiel* est utilisé dans le milieu de l'imagerie médicale pour désigner les problèmes dus à l'échantillonnage (ou résolution) de l'image. En effet, si on considère une portion d'image à la frontière de deux tissus différents, l'échantillonnage donnera des pixels qui portent une information provenant de plusieurs pixels voisins.
- **Information d'intensité insuffisante** : l'intensité observée sur d'images IRM n'apporte qu'une information incomplète sur les structures anatomiques sous-jacentes c'est-à-dire plusieurs structures peuvent avoir la même intensité, par exemple, l'intensité d'IRM lors de la transition entre la MG et la MB est complexe, et peut être variable. Une solution qui permet de palier à l'insuffisance des informations présentées dans l'image, est l'utilisation simultanée de plusieurs modalités.
- **Artefacts d'acquisition (erreurs)** : les principaux artefacts apparaissant en IRM ont été présentés dans le premier chapitre. Lors de la segmentation, l'artefact le plus souvent cité est l'artefact d'inhomogénéité de la radio fréquence RF qui se manifeste par une variation spatialement lente de l'intensité des tissus sur l'image. La plupart des méthodes de segmentation prend en compte cet artefact, soit en l'éliminant par un prétraitement approprié, soit en le modélisant explicitement au cours de la segmentation [SEM 08].

3. Approches de segmentation d'images

3.1 Approches régions

Les approches de segmentation *régions* visent à créer une partition de l'image en un ensemble de régions homogènes au sens d'un ou plusieurs critères [ANN 02]. On recherche donc à rassembler les pixels par critères de similarité. Dans ce qui suit nous intéressons particulièrement aux méthodes à base de seuillages, puis aux méthodes par croissance de régions et finalement aux méthodes fondées sur la morphologie mathématique.

3.1.1. Seuillages

Les méthodes de segmentation par seuillages visent à discriminer les individus (pixels) en fonction de leurs niveaux radiométriques (niveaux de gris). Elles supposent donc, de manière implicite, que l'intensité des structures est suffisamment discriminante pour permettre leur bonne séparation. Le plus souvent, les seuils sont déterminés à l'aide d'une analyse d'histogrammes, localement ou globalement à l'image ou au volume [ANN 02].

Les méthodes de segmentation par seuillages sont des méthodes rapides et simples à mettre en œuvre. Néanmoins, elles échouent lorsque les régions de recherche sont insuffisamment contrastées, lorsque les niveaux radiométriques des structures anatomiques se recouvrent. Toutefois, des solutions peuvent être mises en place. En particulier, l'effet de la dérive peut être réduit par sa prise en compte lors de prétraitements ou simultanément à la segmentation [LIM 89]. De plus, l'absence de prise en compte de contraintes de voisinage rend ces méthodes très sensibles au bruit, les régions obtenues pouvant alors être sémantiquement peu cohérentes. Ceci est alors particulièrement pénalisant dans le cadre de la recherche de tumeurs cérébrales.

Donc, les méthodes de segmentation par seuillages ne sont pas adaptées elles seules à segmenter les principales structures anatomiques et les éventuelles pathologies. Leur utilisation nécessite, en particulier, la prise en compte d'informations locales, liées au voisinage d'un pixel [ANN 02].

3.1.2. Croissance de régions

Les méthodes de segmentation par de croissance de régions sont fondés sur l'agrégation progressive des pixels à partir d'un point initial appelé *germe*, selon un critère. Les points voisins du germe sont successivement examinés ; s'ils vérifient le critère d'agrégation, ils sont inclus à la région et deviennent à leur tour un germe ; dans le cas contraire, ils sont rejetés. Le critère est le plus souvent un critère local basé sur les niveaux radiométriques des points [ANN 02].

Dans [SCH 01], Schnack et al ont segmentés le ventricule latéral et le troisième ventricule par croissance de régions et morphologie mathématique. La croissance de régions est guidée par des connaissances anatomiques *a priori* sur le système ventriculaire. Les tests montrent un taux de recouvrement de 98% entre les segmentations obtenues et la vérité terrain. Dans [WAR 95], Warfield et al. Présentent une méthode d'identification des tumeurs de la matière blanche. Cette méthode suppose la segmentation préalable du cortex, de la matière grise et de la matière blanche. En particulier, le cortex est segmenté par croissance de régions, guidée par des connaissances *a priori* sur la forme de la structure, ainsi que par des contraintes d'intensité et de gradient. Les lésions sont ensuite détectées par un maximum de vraisemblance.

La segmentation par croissance de régions est simple à mettre en œuvre, mais parfois coûteuse en temps de calcul. Leur avantage réside dans la prise en compte des informations contextuelles par le mécanisme de croissance et dans la possibilité d'introduire des informations de plus haut niveau. Néanmoins, leur inconvénient est lié à la disposition des germes initiaux. En effet, celle-ci nécessite quasi-systématiquement l'intervention d'un opérateur, en particulier lorsqu'il s'agit de détecter des tumeurs, qui, par définition, n'ont pas de localisation *a priori* dans le cerveau. Enfin, nous notons leur utilisation restreinte au contexte simple-écho. [ANN02]

3.1.3. Morphologie mathématique

La morphologie mathématique est une branche des mathématiques fondée sur l'étude des ensembles permettant d'analyser les structures géométriques. Ses bases théoriques ont été établies par Matheron [MAT 65] et Serra [SER 82].

Les outils de base de la morphologie mathématique (érosion, dilatation, fermeture, ouverture. . .) se retrouvent fréquemment dans des méthodes de segmentation de l'enveloppe cérébrale. Ainsi dans [BRU 93], Brumer utilise des érosions, dilatations et étiquetages en composantes connexes à la suite de seuillages afin de déconnecter les structures non cerveau de la structure cerveau et d'affiner le masque initial. Cette méthode est reprise dans [GER 98] Géraud propose, à la suite de la segmentation du cerveau, de coupler seuillages et morphologie mathématique pour segmenter successivement le liquide céphalo-rachidien (ventricules et liquide des sillons) et la matière grise (cortex et noyaux de gris centraux) sur des images pondérées en T_1 . Les résultats obtenus sont jugés très corrects par l'auteur.

La morphologie mathématique s'appuie sur des informations *a priori* concernant la forme des objets à segmenter, donc elle ne semble donc pas adaptée à la recherche de zones tumorales, celles-ci étant, par nature, caractérisées par une grande variabilité de forme. La morphologie mathématique est rarement utilisée seule pour segmenter les structures cérébrales mais fait généralement partie d'une chaîne de traitements plus complexe. Alors qu'il n'existe pas de méthode directe de segmentation d'images IRM par morphologie mathématique [ANN 02].

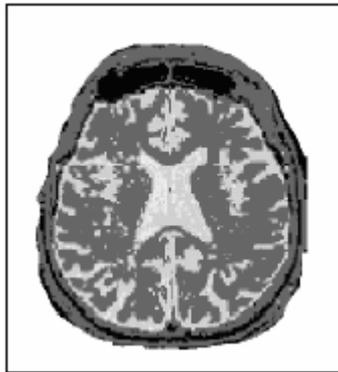


Figure 2.1 : Image IRM segmentée par une approche région.

3.2 Approches contours

Les approches contours se fondent sur la dissimilarité présentée par les pixels de contour par rapport à ces voisins. Or, ils travaillent sur les discontinuités des images afin de déterminer les contours des régions. Divers traitements puisés dans les méthodes propres à l'analyse morphologique sont souvent nécessaires pour obtenir des contours fermés. Toutes ces approches sont bien adaptées aux images qui présentent des régions uniformes au sens des niveaux de gris et offrent une bonne localisation spatiale. Mais, elles restent très sensibles aux bruits et aux contours mal définis (bruit généré par les capteurs, à l'éclairage et aux phénomènes d'ombrage). Dans la suite, nous nous intéressons plus particulièrement aux méthodes dérivatives et aux méthodes fondées sur les modèles déformables.

3.2.1. Méthodes dérivatives

Les méthodes dérivatives permettent de détecter de manière simple les ruptures dans une image ou un volume. Les contours sont assimilés à des points de forts gradients ou de dérivées secondes nulles. On retrouvera alors des opérateurs tels que les filtres de Roberts [ROB 65], de Sobel [SOB 78] ou de Prewitt [PRE 70] pour le gradient et le Laplacien pour les dérivées seconde, et parmi la diversité des opérateurs, celui de Marr-Hildreth [MAH 80] qui fournit des contours fermés, est plébiscité par de nombreux auteurs.

Pour la segmentation des images IRM cérébrales, les méthodes dérivatives, employées sont relativement peu nombreuses dans la littérature. Bien que rapides et ne nécessitant pas d'apport d'information a priori, elles ne permettent pas, en général, d'obtenir des contours fermés. Elles sont en effet très sensibles au bruit et à la dérive du champ radiofréquence. Leur efficacité repose sur l'existence de contrastes entre les structures recherchées. Si ceux-ci existent effectivement au niveau de la peau ou du système ventriculaire, ils sont beaucoup moins importants au niveau de la transition matière blanche - matière grise ou encore au niveau du liquide céphalo-rachidien périphérique [DEL 91].

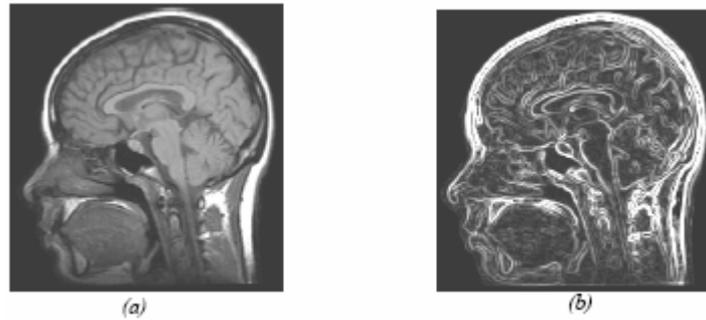


Figure 2.2 : Segmentation, approche contour. (a) : image IRM originale, (b) : image segmentée (approche contour)

3.2.2. Modèles déformables

Les méthodes de segmentations fondées sur les modèles déformables dérivent des méthodes de contours dynamiques. Le principe de base est de faire évoluer un objet d'une position initiale vers une position d'équilibre par la minimisation itérative d'une fonctionnelle. Cette dernière permet de prendre en compte, par l'intermédiaire de forces externes et internes, des informations *a priori* sur la forme de l'objet à détecter. Ils ont l'avantage, par rapport aux méthodes dérivatives, de fournir des contours ou surfaces fermés. Les différentes méthodes de la littérature sont généralement dédiées, non pas à la segmentation de l'ensemble des principales structures du cerveau, mais à celle d'une structure particulière [ANN 02].

Dans [GHA 98], une surface dynamique discrète permet la segmentation de l'hippocampe dans le cadre du diagnostic de troubles tels que l'épilepsie. L'originalité de la méthode est d'introduire une nouvelle force externe fondée sur la recherche de minima locaux dans l'image permettant une déformation correcte du contour même lorsque l'objet recherché présente des discontinuités. Notons que la surface initiale est calculée à partir d'un ensemble de contours tracés par un opérateur. La segmentation a été validée par comparaison avec une segmentation manuelle experte.

Les méthodes de segmentation par modèles déformables permettent en effet de s'adapter à la forme complexe des structures grâce à l'enrichissement des contraintes de déformation. Elles s'avèrent efficaces pour la segmentation de structures anatomiques particulières. Cependant, ces méthodes restent relativement sensibles à leur initialisation. Face à ce problème, les auteurs tendent à inclure des procédures d'initialisation automatique de façon à réduire l'influence des opérateurs et augmenter la reproductibilité des résultats. De plus, nous notons leur utilisation restreinte au contexte monomodal [GER 99].

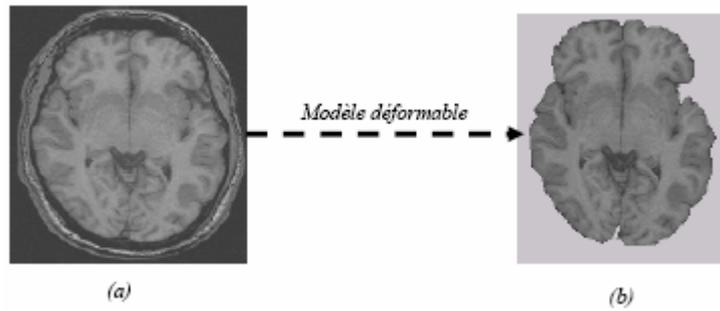


Figure 2.3 : Segmentation par modèle déformable. (a) : Image originale, (b) : Cerveau isolé

2.3. Segmentation par classification

L'approche de segmentation par classification permet de regrouper les échantillons d'une population en sous populations, en fonction de n'importe quel critère de similarité. Dans l'objectif est de faire une classification des tissus cérébraux (principalement le liquide cérébro-spinal, la matière blanche et la matière grise). Ces méthodes peuvent être classées en deux catégories de méthodes : les méthodes supervisées et les méthodes non supervisées (automatiques) [SEM et al].

3.3.1. Méthodes supervisées

Dans tout type des méthodes de segmentation d'image, Les données à analyser $X = \{x_j \in \mathbf{R}^p, 1 \leq j \leq N\}$, où X_j est un vecteur forme représentant le pixel j , pour les méthodes supervisées les données sont donc décomposées en un ensemble d'apprentissage et un ensemble de test. Les données segmentées de l'ensemble d'apprentissage proviennent d'un étiquetage manuel des images ou des régions d'intérêt en C classes de tissus $C_1 \dots C_C$ par un ou plusieurs experts (radiologues). Chaque classe C_i se voit donc affecter un ensemble d'apprentissage E_i , et les données de l'ensemble de test sont segmentées en fonction des E_i . Puisque la structure anatomique d'un cerveau est différente d'un patient à l'autre, l'étiquetage doit être renouvelé pour chaque patient ou groupe de patients, ce qui représente un inconvénient major de ces méthodes.

3.3.1.1. Segmentation bayésienne

Le principe de base de segmentation bayésienne est le calcul de probabilités d'appartenance à un tissu donne pour chaque pixel de l'image afin de la segmenter. Cette approche utilise la règle de Bayes combinée à des modèles généraux décrivant les distributions des niveaux de gris des tissus recherches. C'est à dire pour chaque vecteur forme X_j , calculer la probabilité conditionnelle $P(C_i|x_j)$ pour chacune des C classes (C_1, \dots, C_C) à l'aide de la règle de Bayes :

$$(\forall i \in \{1 \dots C\}) P(C_i | X_j) = \frac{P(C_i) \cdot P(X_j | C_i)}{\sum_{k=1}^C P(C_k) \cdot P(X_j | C_k)} = \frac{P(C_i) \cdot P(X_j | C_i)}{P(X_j)}$$

Les lois $P(C_i)$ et $P(x_j/C_i)$, donnant la probabilité de trouver un individu de classe C_i parmi des individus de vecteurs forme X_j , sont souvent inconnues. Si la probabilité $P(C_i)$ peut être assez facilement déduite de la base d'apprentissage, la recherche de la loi $P(x_j/C_i)$ est plus délicate, elle peut être effectuée par estimation.

a- **Segmentation bayésienne paramétrique** : ici, l'intensité d'un pixel est considérée comme une combinaison linéaire de probabilités d'appartenance attachées à chaque classe. La probabilité conditionnelle $P(x_j/C_i)$ est modélisée par une fonction dépendant d'un vecteur de paramètre q . Le problème est alors pour chaque classe C_i d'estimer le meilleur paramètre q connaissant de l'ensemble d'apprentissage E_i de cette classe. De nombreux estimateurs peuvent être utilisés dans ce but, mais l'estimateur de maximum de vraisemblance est le plus fréquemment rencontré. Celui-ci détermine la valeur de q qui maximise :

$$P(E_i|\theta) = \prod_{k=1}^{|E_i|} P(X_j|C_i, \theta)$$

b- **Segmentation bayésienne non paramétrique** : Dans ce type d'approche, les probabilités conditionnelles $P(x_j/C_i)$ sont supposées quelconques. La forme non paramétrique doit permettre de rendre compte le plus fidèlement possible de la réelle distribution statistique des niveaux de gris dans l'image. Une méthode souvent utilisée consiste en la modélisation des probabilités conditionnelles par des fenêtres de Parzen [DUD 73]. Pour chaque classe C_i d'ensemble d'apprentissage $E_i = \{n_{ij}\}$, la loi $P(x_j/C_i)$ est obtenue en centrant une gaussienne de largeur s faible autour de chaque vecteur d'entraînement :

$$P(X_j|C_i) = \frac{1}{|E_i|} \sum_{k=1}^{|E_i|} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_j - n_{ij})^2}{2\sigma^2}}$$

Une limite importante de cette approche est qu'elle est fondée sur une hypothèse d'uniformité des niveaux de gris pour un même tissu dans toutes les zones de l'image, hypothèse non vérifiée dans la réalité. Par conséquent, des erreurs de classification peuvent résulter du traitement choisi. Selon [BEZ 93], les hypothèses de ce type sont mal adaptées à la segmentation d'images médicales, IRM en particulier, où l'intensité d'un pixel ne peut être dérivée d'une combinaison linéaire de probabilités distinctes attachées aux classes.

En contrepartie, l'introduction de connaissance *a priori* sous forme de probabilité permet de simplifier les calculs pour la classification et d'éviter le recours à un processus d'isolement du cerveau par rapport au reste de l'image.

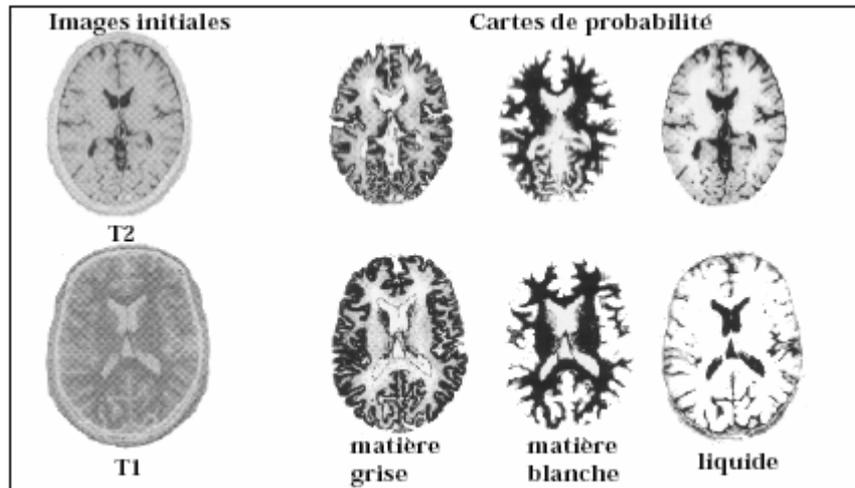


Figure 2.4 : Segmentation par méthode bayésienne.

3.3.1.2. Segmentation Markovienne

La segmentation Markovienne se base la minimisation d'une fonction d'énergie définie sur l'image à segmenter. Grâce à l'introduction de notions de voisinages entre pixels, cette approche permet d'envisager la recherche de modèles locaux de distributions selon un principe équivalent localement à celui de la classification bayésienne.

a) Principe

Les approches utilisant les champs de Markov sont fondées sur une théorie utilisant des notions de voisinages et de cliques que nous rappelons ici. Ou une image de taille $N \times N$ est définie par l'ensemble de ses pixels P_{ij} et noté : $I = \{P_{ij} = (i, j), i \leq l, j \leq n\}$ et à chaque pixel est associée une variable aléatoire A_j dont les valeurs appartiennent à un ensemble Ω donné. L'image est considérée comme un vecteur aléatoire $A = (A_i, i \in I)$ et $a = (a_i, i \in I)$ est une réalisation de A . Un système de voisinage V est défini sur cette image de la façon que tout voisinage V vérifie deux propriétés qui sont :

- Un pixel n'appartient pas à son voisinage.
- La relation de voisinage est symétrique : si un pixel S appartient au voisinage d'un pixel T alors T appartient au voisinage v_s de S .

La définition d'un champ markovien associe un champ aléatoire A à un système de voisinage V à la condition suivante (la condition de positivité étant nécessaire seulement pour que le champ soit un champ de *Gibbs*) :

$$\begin{cases} P(a) > 0 \\ P(a_i | a_j, j \in I - \{i\}) = P(a_i | a_j, j \in v_i) \end{cases}$$

Ce qui revient à considérer que l'influence d'un pixel est limité à une zone locale de l'image, située autour de ce pixel. La définition des cliques est nécessaire pour la formalisation.

Une clique est un ensemble de pixels tel que deux pixels quelconques de l'ensemble sont voisins au sens de V .

• **Mesure de Gibbs :**

D'après le théorème d'Hammersley-Clifford un champ aléatoire A associé au système de voisinage V est un champ markovien si et seulement si sa distribution de probabilité $p(A=a)$ est une mesure de Gibbs :

$$\forall A \in \Omega^{N \times N}, P(a) = \frac{\exp(-u(a))}{\sum_{a \in \Omega^{N \times N}} \exp(-u(a))}$$

Où U est une fonction d'énergie correspondant à la somme des potentiels de chaque clique de l'image.

b) Application à la segmentation d'image

Les modalités anatomiques se prêtent tout type de méthode de segmentation qui prend en charge des contraintes de voisinages. Pour cela Jaggi et al. [JAG 97] propose une approche par champs de Markov permettant de prendre en compte les effets de volume partiels en IRM en estimant les coefficients d'un modèle de mélange à 5 classes (LCS, MG, MB, MB/MG. Held et al. [HEL 97] qui segmentent des images IRM pondérées en T_2 et densité de protons à l'aide d'un champ de Markov modifié permettant d'estimer les hétérogénéités dans l'image simultanément à la segmentation.

3.3.1.3. La méthode des k plus proches voisins (kppv)

La méthode des k plus proches voisins est une méthode de classification qui est considérée supervisée (non supervisée pour certains auteurs) introduite dans [DUD 73] qui repose sur un principe simple et intuitif de regroupement d'individus en fonction de leur voisinage et qui est largement utilisée en classification d'images.

a) Principe

Cette méthode se base sur le calcul de la distance entre le vecteur non étiqueté X_j à chacun des vecteurs des E_i , avec E_i de cardinaux assez grands. La probabilité *a posteriori* d'appartenance de X_j à une classe est estimée d'après la fréquence des classes présentes parmi les k plus proches voisins de ce vecteur. La règle de décision classique consiste à affecter X_j à la classe la plus représentée parmi ses k plus proches voisins. Seuls la distance et k doivent être déterminés. La première détermine la forme du voisinage de chaque vecteur X_j , et le choix de k est arrêté lorsqu'une valeur supérieure ne modifie pas la classification.

b) Application à la segmentation d'images

La méthode de kppv est largement utilisée en classification d'images cérébrales. Prenant l'exemple de Clarke et al. [CLA 93] qui choisissent un k égale à 7. Ces auteurs utilisent cet

algorithme comme méthode de référence car elle s'avère être selon eux, la plus stable en termes de variabilité. Toutefois, dans des articles postérieurs à cette étude, des chercheurs de la même équipe précisent, dans la partie concernant la sélection de l'ensemble d'apprentissage, que si les résultats obtenus sont visiblement anatomiquement incorrects, un nouvel ensemble d'apprentissage doit être choisi jusqu'à obtention d'une bonne segmentation [VEL 95]. Le choix de l'ensemble d'apprentissage n'est pas donc évident.

Cette méthode caractérisée par un temps de calcul élevé, du au calcul pour chaque pixel de l'image n'appartenant pas à l'ensemble E_i la distance à chacun des pixels de l'ensemble d'apprentissage, puis les ordonner avant de sélectionner la classe la plus représentative. Vinitski et al. [VIN 99] ont proposé de réduire le temps de calcul en excluant les points "redondants" et "marginiaux" de l'ensemble d'apprentissage de chaque classe. Ôter un élément redondant de l'ensemble d'apprentissage d'une classe de tissu n'est pas nécessairement judicieux, puisque cela revient à augmenter ainsi le risque de mauvaises classifications.

3.3.1.4. Réseaux de Neurones

a) Description formelle d'un réseau de neurones

Un réseau de neurones est le résultat de l'interconnexion d'un ensemble d'unités élémentaires s'appel *noeuds* à l'aide d'une fonctions d'activation linéaires ou non linéaires selon de nombreux modèles de réseaux (perceptrons multicouches, réseaux de Hop Field,...).

Ces noeuds sont regroupés pour les réseaux multicouches en un sous-ensemble d'entrée, un autre de sortie et éventuellement un ensemble de neurones cachés. les différents noeuds étant complètement ou partiellement interconnectés aux autres. L'ensemble des liens convergeant vers un noeud constitue les connexions entrantes du noeud. Ceux qui divergent vers d'autres noeuds sont les connexions sortantes. A chaque connexion entre des noeuds i et j , est associé un poids W_{ij} représentant la force de l'influence du noeud i sur le noeud j . L'ensemble des poids est regroupé dans un vecteur de poids W . Un vecteur de scalaires A présenté à tous les noeuds d'entrée est appelé exemple. A cet exemple sont aussi associées les valeurs y (le vecteur de sortie) que l'on désire apprendre. Les poids des connexions sont éventuellement modifiés au cours d'un cycle d'apprentissage.

Modifier la sortie des noeuds à partir de leurs entrées consiste tout d'abord à calculer l'activation présente à l'entrée du noeud puis à calculer la sortie du noeud suivant la fonction d'activation qu'elle possède. Un réseau de neurones peut ainsi être défini pour chaque noeud par quatre éléments :

- La nature de ses entrées, qui peuvent être binaires ou réelles.
- La fonction d'entrée totale e , qui définit le pré-traitement $e(a)$ effectué sur les entrées.
- La fonction d'activation f du noeud qui définit son état de sortie en fonction de la valeur de e .

- La nature de ses sorties, qui peuvent être binaires ou réelles.

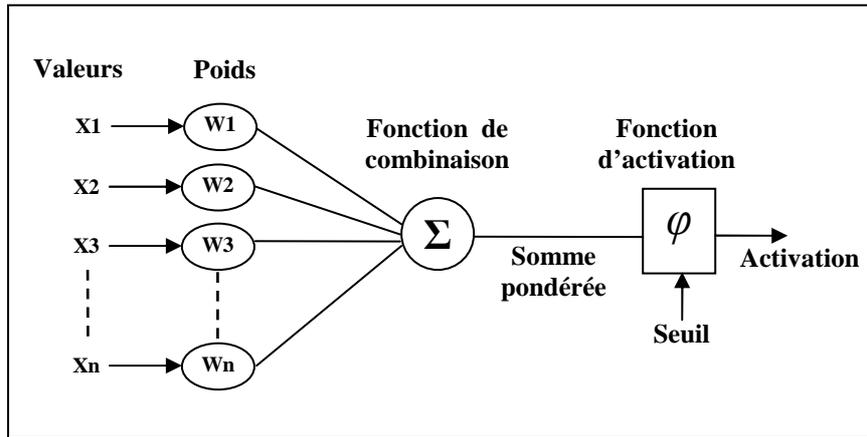


Figure 2.5 : Structure d'un neurone artificiel.

Le fonctionnement du réseau de neurones est subdivisé en deux étapes:

a) Première étape (Apprentissage) :

L'adaptation des paramètres du réseau de neurones pour donner une réponse désirée à une entrée donnée s'appelle apprentissage. La méthode classique en apprentissage supervisé consiste à se procurer un ensemble d'exemples entrées/sorties $\{(a_k, y_k), k \in K\}$. La fonction calculée par le réseau est alors décrite sous une forme paramétrique, $f(a_k, w)$, et désigne la sortie du réseau quand le vecteur a_k est présenté en entrée et que le réseau utilise le vecteur de poids w . Une distance d sur l'espace vectoriel de sortie est également choisie et permet de mesurer l'erreur commise $d(f(a_k, w), y_k)$ en un point par le réseau. Le but de l'apprentissage est alors de trouver la valeur de w qui minimise l'erreur totale commise par le réseau pour l'ensemble $\{(a_k, y_k), k \in K\}$.

b) Deuxième étape (Généralisation) :

La deuxième étape consiste à généraliser la représentation construite lors de la phase d'apprentissage pour des données quelconques. L'erreur de généralisation ainsi mesurée dépend essentiellement du nombre d'exemples utilisés pour l'apprentissage, de l'architecture du réseau et de la complexité du problème sous-jacent.

b) Application à la segmentation des tissus

Magnotta et al [MAG 99] ont segmenté certaines structures cérébrales (putamen, noyau caudé, corps calleux) sur des images IRM en utilisant des réseaux de neurones. L'apprentissage a été effectué par une segmentation manuelle sur 2/3 des images disponibles, le reste étant segmenté par le réseau entraîné qui donne un taux de réussite moyen de 88.

Les réseaux de neurones sont très sensibles à la base d'entraînement proposée à l'entrée du réseau. Si le réseau de neurones doit être utilisé, il impose une intervention constante de l'opérateur qui doit sélectionner pour chaque image des données d'entraînement pertinentes.

Cela a été démontré dans [CLA 93], Clarke *et al* ont analysé les résultats d'un réseau de neurones pour la caractérisation de tissus (MB, MG, LCS, muscle, graisse, fond) en IRM, avec une base d'entraînement constituée à partir d'images pondérées en T_1 , T_2 et en densité de protons.

3.3.2. Méthodes non supervisées

Les méthodes non supervisées ou dite automatiques sont des méthodes qui ne nécessitent aucune base d'apprentissage et aucune tâche préalable d'étiquetage manuel. La seule intervention de l'expert se situe à la fin du processus de segmentation pour identifier les tissus en comparant les classes calculées avec les classes biologiques. Les algorithmes non supervisés les plus répandus tendent à minimiser une fonction coût, dépendant de la distance de chaque pixel aux prototypes (ou noyaux) des classes [SEM et al].

Notons $X=(x_j, j=1..N)$ l'ensemble des vecteurs forme de R_p , avec $x_j=[x_{j1},x_{j2},\dots,x_{jp}]^T$, et $B=(b_1,\dots,b_c)$ un ensemble de vecteurs prototypes inconnus, où b_i caractérise la classe i . La génération de fonctions d'appartenance consiste à construire automatiquement des fonctions permettant d'affecter chaque x_j à une ou plusieurs classes selon certains critères. Dans la suite de ce chapitre nous proposons quelques algorithmes de classification non supervisés.

3.3.2.1. Algorithmes de classification non flous

Dans la méthode HCM (Hard Clustering Means) un élément de X est attribué à une classe et une seule parmi les C proposées. Dans ce cas, la fonctionnelle à minimiser est :

$$J(B,U,X) = \sum_{i=1}^C \sum_{j=1}^N u_{ij} d^2(x_j, b_i)$$

et les solutions au problème s'écrivent :

$$u_{ij} = \begin{cases} 1 & \text{ssi } d^2(x_j, b_i) < d^2(x_j, b_k) \forall k \neq i \\ 0 & \text{sin on} \end{cases}$$
$$b_i = \frac{\sum_{j=1}^N u_{ij} x_j}{\sum_{j=1}^N u_{ij}}$$

La classification des éléments de X s'effectue de manière itérative en alternant l'étape de classification et l'étape de mise à jour des centres, jusqu'à stabilisation de la segmentation ou de la fonction objectif.

Dans une méthode comme HCM, les éléments sont classés de façon certaine comme appartenant à une classe et une seule. Quelle que soit la modalité d'imagerie, cette assertion ne reflète pas la réalité physique de l'échantillon étudié (bruit, volume partiel, hétérogénéité de

champ) [SEM 06]. Les méthodes présentées dans les paragraphes suivants permettent d'obtenir une segmentation floue qui prend en compte ces aspects imprécis et incertains.

Dans une variante nommée ILSC (Iterative Least Squares Clustering) [VEL 95], les vecteurs b_i sont recalculés à chaque ajout d'un élément dans une classe. Ces auteurs rapportent un bon taux de reconnaissance des tissus cérébraux en IRM, mais cette modification entraîne à la fois un temps de calcul plus long et une dépendance de la partition finale à l'ordre de parcours de l'image [PEN 99].

3.3.2.2. Algorithme des C-moyennes floues (Fuzzy C- Means « FCM »)

a) Présentation de l'algorithme FCM

La classification floue assigne à un pixel son degré d'appartenance à chacune des classes et non pas l'étiquette relative à une classe unique. ou le degré d'apparence est la valeur exprime l'appartenance incertaine d'un pixel à une région et sont appelées degrés d'appartenance. Ce dernier se situe dans l'intervalle $[0, 1]$ et les classes obtenues ne sont pas forcément disjointes. Dans ce cas, les données x_j ne sont plus assignées à une classe unique, mais à plusieurs par l'intermédiaire de degrés d'appartenance u_{ij} du vecteur x_j à la classe i . Le but des algorithmes de classification est non seulement de calculer les centres de classe B mais aussi l'ensemble des degrés d'appartenance des vecteurs aux classes.

Si u_{ij} est le degré d'appartenance de x_j à la classe i , la matrice $U_{C \times N}[u_{ij}]$ est appelée matrice de C-partitions floues si et seulement si elle satisfait aux conditions :

$$\forall i \in \{1 \dots C\}, \forall j \in \{1 \dots N\} \left\{ \begin{array}{l} u_{ij} \in [0,1] \\ 0 < \sum_{j=1}^N u_{ij} < N \end{array} \right.$$

$$\forall j \in \{1 \dots N\} \sum_{i=1}^C u_{ij} = 1$$

La fonctionnelle à minimiser et les solutions au problème du FCM sont décrites par les formules suivantes :

$$J(B, U, X) = \sum_{i=1}^C \sum_{j=1}^N u_{ij} d^2(x_j, b_i)$$

$$u_{ij} = \left[\sum_{k=1}^C \left(\frac{d^2(x_j, b_i)}{d^2(x_j, b_k)} \right)^{2/(m-1)} \right]^{-1}$$

$$b_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m}$$

L'algorithme des C-moyennes floues (FCM) consiste alors en l'application itérée des deux dernières équations jusqu'à stabilité des solutions. Le critère d'arrêt des itérations, définissant cette stabilité, peut par exemple consister en l'étude de la norme de la matrice U ou en la stabilité des centres de classe sur deux itérations successives.

b) Application à la segmentation des tissus

De nombreux travaux ont notamment été effectués en IRM cérébrales. Clark et al. [CLA 94] ont utilisé le FCM comme étape d'initialisation à un système expert pour segmenter des volumes de tumeur ou d'oedème sur des images pondérées en T_1 , T_2 et en densité de protons, Baumgartner et al. [BAU 98] ont ainsi utilisé le FCM pour segmenter les régions activées (simulation et aires motrices) du cerveau. Les pixels étant représentés par leur niveau de gris, les auteurs ont montré non seulement que l'algorithme avait des performances comparables à l'analyse de corrélation standard (avec l'avantage de ne nécessiter aucune connaissance *a priori* sur le paradigme), mais aussi que le FCM détectait des zones effectivement activées qui restaient silencieuses avec la corrélation. En IRM traditionnelle.

3.3.2.3. Algorithme de classification possibiliste

a) Présentation de l'algorithme PCM

Cet algorithme de classification appelé possibilistic c-means (PCM), dans lequel l'équation $\forall j \in \{1 \dots N\} \sum_{i=1}^c u_{ij} = 1$ ne doit pas contraindre les appartenances d'un vecteur quelconque x_j si les classes représentées par les nuages sont considérées comme ces sous-ensembles flous sur le domaine $X=(x_j, j=1 \dots N)$. Les degrés d'appartenance doivent seulement appartenir à l'intervalle $[0,1]$. Ainsi, un nouvel ensemble de contraintes est défini :

$$\forall i \in \{1 \dots C\}, \forall j \in \{1 \dots N\} \quad u_{ij} \in [0,1]$$

$$\forall j \in \{1 \dots N\} \left\{ \begin{array}{l} 0 < \sum_{i=1}^c u_{ij} = 1 < N \\ \max_i u_{ij} > 0 \end{array} \right.$$

Cette condition assure simplement que la partition floue résultante de l'algorithme recouvre le domaine X . En toute rigueur, la matrice U résultante n'est plus une C-partition floue, puisque la contrainte $\forall j \in \{1 \dots N\} \sum_{i=1}^c u_{ij} = 1$ n'est plus satisfaite.

La fonctionnelle à minimiser et les solutions de l'algorithme PCM deviennent :

$$J(B, U, X) = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d^2(x_j, b_i) + \sum_{i=1}^C \eta_i \sum_{j=1}^N (1 - u_{ij})^m$$

$$\forall i \in \{1 \dots C\} \quad \forall j \in \{1 \dots N\} \quad u_{ij} = \left[1 + \left(\frac{d^2(x_j, b_i)}{\eta_i} \right)^{1/(m-1)} \right]^{-1}$$

$$\forall i \in \{1 \dots C\} \quad b_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m}$$

où η_i est un réel positif déterminant la distance à laquelle le degré d'appartenance d'un vecteur à la classe i est égal à 0.5.

Pour classer des nuages de points très proches la fonctionnelle à minimiser peut être remplacé par la fonctionnelle: $J(B, U, X) = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d^2(x_j, b_i) + \sum_{i=1}^C \eta_i \sum_{j=1}^N u_{ij} (\log(u_{ij}) - 1)$

L'algorithme résultant de l'application itérée des solutions constitue l'algorithme de classification possibiliste, noté dans la suite PCM.

b) Application à la segmentation des tissus

Cet algorithme est peu utiliser pou la segmentation d'image cérébrales. Masulli et Schenone [MAS 99] ont proposé de combiner une approche par réseaux de neurones avec l'algorithme PCM pour segmenter les tissus cérébraux et des entités pathologiques (méningiomes). Barra et Boire [BAR 00] ont également appliqué PCM en IRM sur des vecteurs forme x_j .

3.3.2.4. Algorithme semi-supervisée :

De nombreuses variations de l'algorithme FCM sont possibles, en changeant la fonctionnelle à minimiser, la métrique d ou les contraintes à appliquer. Nous présentons ici la *méthode FCM semi-supervisée* qui est une variante plus particulièrement utilisée pour la caractérisation des tissus cérébraux.

Dans [BEN 94], Bensaïd et al ont alors proposé une version semi-supervisée de l'algorithme des C-moyennes floues. Un ensemble de vecteurs d'apprentissage A est défini par un expert dont chaque élément appartient de façon exclusive à l'une des classes. De plus, une pondération est attribuée à ces vecteurs, qui permet de créer des copies des vecteurs d'apprentissage et ainsi de guider la segmentation vers un résultat anatomiquement correct. Le calcul des degrés d'appartenance des vecteurs n'appartenant pas à A reste inchangé par rapport à un FCM conventionnel. En revanche, la mise à jour des centres de classe nécessite d'intégrer les vecteurs d'apprentissage au calcul :

$$b_i = \frac{\sum_{x \in A} p_i u'_{ik}{}^m x + \sum_{x \notin A} p_i u_{ik}^m x}{\sum_{x \in A} p_i u'_{ik}{}^m + \sum_{x \notin A} p_i u_{ik}^m}$$

où p_i est un facteur de pondération attribué aux vecteurs d'apprentissage de la classe i et U_{jk} est égal à 1 si l'élément k appartient à la classe d'apprentissage i et 0 sinon. Cet algorithme présente deux inconvénients majeurs : il nécessite une intervention humaine pour définir l'ensemble d'apprentissage et introduit les p_i comme paramètres supplémentaires. Dans certaines applications, l'intervention de l'utilisateur peut être évitée. Suckling et al. Dans [SUC 99] initialisent ainsi les vecteurs d'apprentissage pour la matière grise et la matière blanche d'après l'histogramme de l'image. Ce type d'initialisation non supervisée est possible dans des cas simples où, malgré un chevauchement des paramètres caractéristiques, les structures sont identifiables sur l'histogramme. Si le facteur de pondération est trop fort, le centre du nuage restera égal au barycentre des vecteurs d'apprentissage de la classe tout au long des itérations. S'il est trop faible, le comportement de l'algorithme semi-supervisé sera similaire à celui d'un FCM conventionnel. L'algorithme semi-supervisé obtient selon ses auteurs de meilleurs résultats qu'un algorithme de ($kppv$) utilisé avec le même ensemble d'apprentissage. Il est donc moins sensible au choix d'un ensemble d'apprentissage médiocre et produit dans ce cas de meilleures segmentations.

2.4 Autres approches

2.4.1. Les modèles pyramidaux

Ces modèles consistent à utiliser des images réduites par deux à chaque fois par rapport à l'image initiale. On peut donc choisir la hauteur de la pyramide en fixant le nombre de divisions que l'on autorise (demi image, quart d'image, huitième d'image, ...). Un trop petit nombre de division revient à n'opérer que sur l'image initiale et n'est pas intéressant en pratique. Trop de divisions donnent une image microscopique sur laquelle on ne peut plus trouver le contour. Pour effectuer la segmentation, on recherche tout d'abord le contour dans une image au 1/8ème par exemple, où le traitement est extrêmement rapide vu la faible taille de l'image. Ensuite, dans l'image au 1/4, on recherche le contour en initialisant ce dernier à partir de l'image au 1/8ème. Puis, on affine au fur et à mesure le tracé du contour pour finalement obtenir une segmentation de l'image à sa taille initiale [AME 05].

2.4.2. Les modèles mixtes

Ils résultent de la combinaison des approches contour et région.

3. Comparaison entre méthodes de segmentation

D'après l'étude des différentes méthodes citées dans ce chapitre, nous avons relevé (Tableau 2.1) des inconvénients et des avantages pour chacune d'entre elles. Ce qui permet de faire une comparaison entre ces méthodes et de dire qu'il n'existe pas de méthode de segmentation complète qui donne des résultats optimaux.

Chapitre 2 : Méthodes de segmentation d'images IRM

		Approches	Structures à segmenté	Contraintes	inconvénients	Avantages
Contours		Méthodes dérivatives [MAT 80]	Système ventriculaire	Contrastes entre les structures	-contours non fermés. - très sensibles au bruit La sous- et sur-segmentation	Rapides et ne nécessitant pas d'information a priori
		Modèle déformable [KAS 88]	Hippocampe Surface corticale	Sensibles à leur initialisation	Non adaptée aux différentes structures	Contours ou surfaces fermés.
Régions		Seuillage [BRU 93]	MG, MB, LCR	Contraste entre Structures	Nécessitent souvent des post traitements	Simple à mettre en œuvre, rapide
		Croissance de régions [THI 97]	MG, Mb, LCR	Disposition des germes initiaux.	Coûteuses en temps de calcul	Simple à mettre en œuvre
Statistiques (Classification)	Supervisées	Champs de Markov [HEL 97]	MG, MB, LCR, MB/MG LCS/MG	Base d'apprentissage	Sensibilité	Prenent En compte les relations spatiales
		Kppv [DUD 73]	MG, Mb, LCR	Le paramètre K	Temps de calcul élevé	Simplicité
		Réseaux de neurones [MAG 73]	Putamen, noyau caudé, Corps calleux	Base d'apprentissage	Intervention d'opérateur	Apprentissage pour chaque image
	Automatiques	HCM [BEZ 81]	MG, MB, LCR	Centres des classes	Ne prend pas l'incertain et l'imprécis	Automatique
		FCM [BEZ 81]	MG, MB, LCR	Centres des classes	Degrés d'appartenance relatifs	Prend l'incertain et l'imprécis
		PCM [KRI 96]	MG, MB, LCR	Centres des classes	Choix des paramètres	Degrés d'appartenance absolus

Tableau 2.1 : Comparaison entre méthodes de segmentation [SEM 08]

Conclusion

Dans ce chapitre nous avons présenté l'ensemble des approches de segmentation des images IRM cérébrales et nous avons mis en évidence la diversité des méthodes pour les différentes approches. Cette diversité reflète la complexité du problème de segmentation des images IRM cérébrales, ces difficultés sont liées, d'une part, aux objets traités et, d'autre part, à la particularité des images IRM.

Le cerveau humain est une entité complexe composée de différentes structures mélangées, à caractères variables importants. Concernant les images IRM, elles fournissent une représentation de l'anatomie cérébrale de qualité mais elles sont sujettes à des incertitudes et des imprécisions vues précédemment, liées, en particulier, aux bruits d'acquisition, au phénomène de dérive du champ radiofréquence, aux effets de volume partiel . . . etc.

CHAPITRE 3 : LES METHODES DE SEGMENTATION PAR CLASSIFICATION

Introduction

Les méthodes de segmentation par classification des pixels affectent chaque pixel à une classe, en fonction d'un ou plusieurs attributs de ce pixel. La classification est dite supervisée lorsque des connaissances *a priori* sont utilisées pour la construction des classes sous la forme d'un ensemble d'apprentissage. Dans le cas où aucune connaissance *a priori* n'est disponible, on parle de classification non supervisée. Dans les deux types de méthodes, de nombreux algorithmes de classification existent dans la littérature.

Dans ce chapitre, nous avons présenté une méthode pour chaque type, dans un premier temps nous avons présenté les machines à vecteurs supports (SVM) puis nous avons présenté les fourmis artificielles (*AntClustering*) en insistant sur leurs applications pour la segmentation des images IRM cérébrales.

1. Les machines à vecteurs supports (SVM)

Introduction

Les machines à vecteurs supports (SVMs) sont apparues au début des années 90 sur le principe de minimisation du risque structurel. Elles reposent sur l'existence d'un classificateur linéaire dans un espace approprié en exploitant les concepts relatifs à la théorie de l'apprentissage statistique. Le compromis entre la capacité d'apprentissage et la capacité de généralisation pour ces machines est respectivement accompli en minimisant l'erreur empirique et au même temps, en essayant de maximiser une marge géométrique. Ces fameuses machines ont été inventées en 1992 par Boser et al, mais leur dénomination par SVM n'est apparue qu'en 1995 avec Cortes et al. Depuis lors, de nombreux développements ont été réalisés pour proposer des variantes traitant le cas non linéaire [BEN 07]. D'où l'utilisation de fonctions dites noyau (kernel) qui permettent une séparation optimale des données [MOH 06].

1.1. Notions fondamentales

«Les SVMs sont des classificateurs linéaires à large marge dans un espace à noyau» est la définition des SVMs la plus courante. Pour mieux éclairer cette définition, on doit mettre l'accent sur quelques notions fondamentales:

1.1.1. La classification

Classer un ensemble d'objets, c'est attribuer à chacun une parmi plusieurs classes définies à l'avance. Cette tâche est appelée « *classification* » ou « *discrimination* ». Un algorithme qui réalise automatiquement une classification est appelé classificateur.

1.1.2. L'apprentissage

Le but de l'apprentissage est de parvenir à classer correctement les futures observations grâce à la connaissance apprise à propos d'un échantillon limité de données. Une question

centrale de l'apprentissage est donc de savoir « *comment utiliser les données pour avoir une bonne performance en généralisation, c'est à dire sur les observations inconnues à venir* ». Les données sont générées par un modèle linéaire en situation de *sous apprentissage*. Alors que le modèle de haut degré est en situation de *sur apprentissage* (*apprentissage par coeur*) comme illustrés dans la figure 3.1.

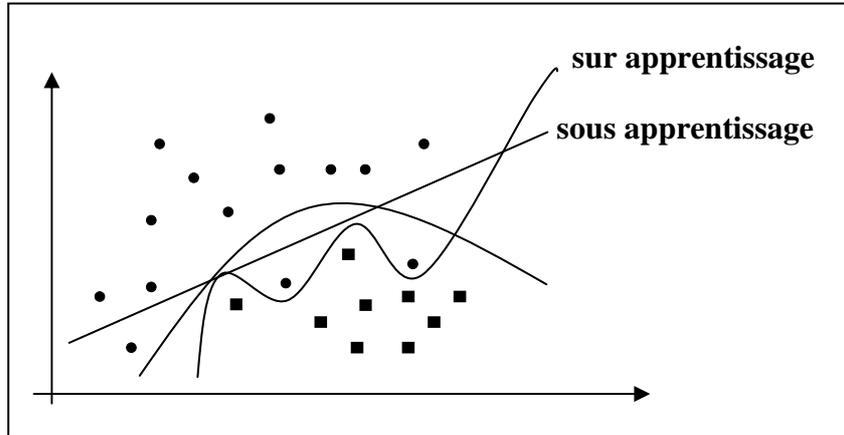


Figure 3.1: Sur apprentissage et sous apprentissage.

- **La théorie d'apprentissage statistique :** L'objectif de la théorie de l'apprentissage statistique [BEN 07] est l'étude d'un modèle conceptuel basé sur le principe de minimisation du risque empirique dit aussi erreur d'apprentissage qui constitue la perte moyenne sur un nombre fini d'exemples de données.

1.1.3. La séparation

Un ensemble de données est dite *séparables linéairement* (figure 3.2-A-) si tous les points associés aux données peuvent être séparés correctement par une frontière linéaire (hyperplan) [JAR] sinon elles sont des données *non séparables linéairement* (figure 3.2-B-).

- 1. Séparateurs Linéaires :** sont des algorithmes qui trouvent une frontière linéaire quand les données sont linéairement séparables, ce type d'algorithmes sont simples puisque il ont peu de paramètres à régler mais les données dans l'espace de grande dimension sont souvent non linéairement séparables.
- 2. Séparateurs non Linéaires :** sont des algorithmes qui ne trouvent pas de frontière linéaire quand les données ne sont pas linéairement séparables, ces algorithmes sont plus précis mais il sont plus compliqués puisqu'ils ont plus de paramètres à régler.

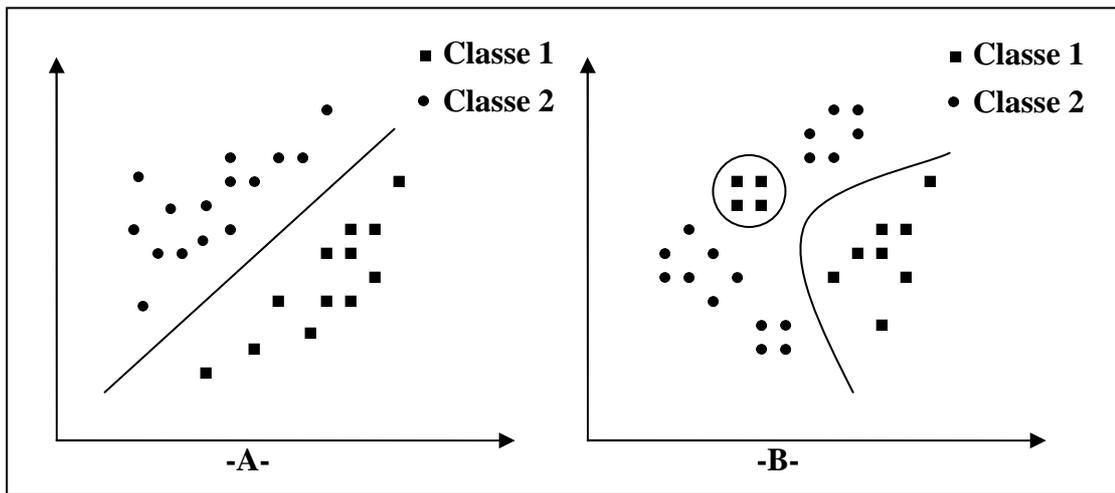


Figure 3.2: Données séparables (et non séparables) linéairement

Note : On peut utiliser des algorithmes linéaires pour des problèmes non linéaires en utilisant des fonctions s'appellent noyaux.

1.2. Le principe de fonctionnement des SVM

1.2.1. L'hyperplan séparateur

Le but de SVM est de trouver un classificateur qui va séparer les données en deux classes, ce classificateur linéaire appelé hyperplan. Dans le schéma en figure 3.3, H est un exemple d'hyperplan H séparant deux ensembles de points. Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

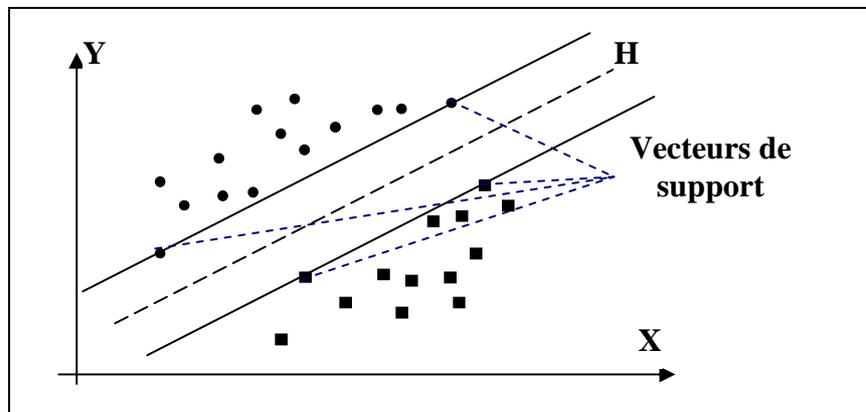


Figure 3.3 : Hyperplan.

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal (sur). Cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale.

1.2.2. La marge

On appelle « *marge* » la distance entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on

parlera de *séparateurs à vaste marge*. Intuitivement, plus une marge large plus la confiance confier à la classification d'un nouvel exemple augmente.

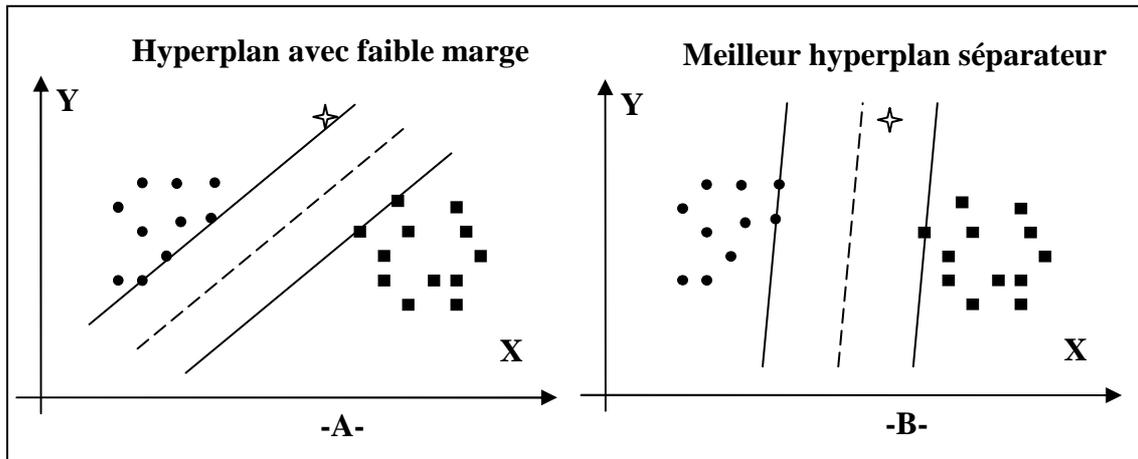


Figure 3.4 : Meilleur et mauvais hyperplan.

Dans le schéma en figure 3.4-B- montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la figure 3.4-A-, qu'avec une plus petite marge, l'exemple se voit mal classé.

Note : La classification d'un nouvel exemple inconnu est généralement donnée par sa position par rapport à l'hyperplan optimal. Dans le schéma en figure 3.4-B-, le nouvel élément sera classé dans la classe des rectangles.

1.2.3. Fondements mathématiques

On cherche à déterminer une fonction f qui, à partir d'une entrée x , produit une sortie $y=f(x)$ par apprentissage d'un échantillon d'apprentissage qui constitue un ensemble de couples entrée-sortie $\{(x_i, y_i) : i = 1, \dots, n\}$ afin de prédire d'autres événements et de bien les classer. On considère des couples $(x, y) \in X \times Y$ avec $X \in \mathbb{R}^d$ et $Y = \{-1, 1\}$.

On veut construire $f : X \rightarrow \mathbb{R}$ (ensemble des réels) de façon que la classe est donnée par le signe de f ; $sign(f)$ qui nous donne la valeur de $y (y = sign(f))$.

1.2.3.1. SVM linéaire

En classification linéaire la fonction f est linéaire en x_i . Donc, elle prend la forme générale suivante: $f(x_i) = \langle w \cdot x_i \rangle + b$ où (w, b) sont les paramètres de $\mathbb{R}^d \times \mathbb{R}$ à estimer de la fonction de décision f , l'opérateur $\langle \cdot \rangle$ désigne le produit scalaire usuel dans \mathbb{R}^d et b est le biais d'un modèle linéaire. Ces paramètres doivent être estimés à partir des données d'apprentissage. L'hyperplan séparateur a donc pour équation $w \cdot x + b = 0$ (Figure 3.5).

La distance d'un point au plan est donnée par $d(x) = |w \cdot x + b| / \|w\|$ et l'hyperplan optimal est celui pour lequel la distance aux points les plus proches (marge) est maximale [BEN 07].

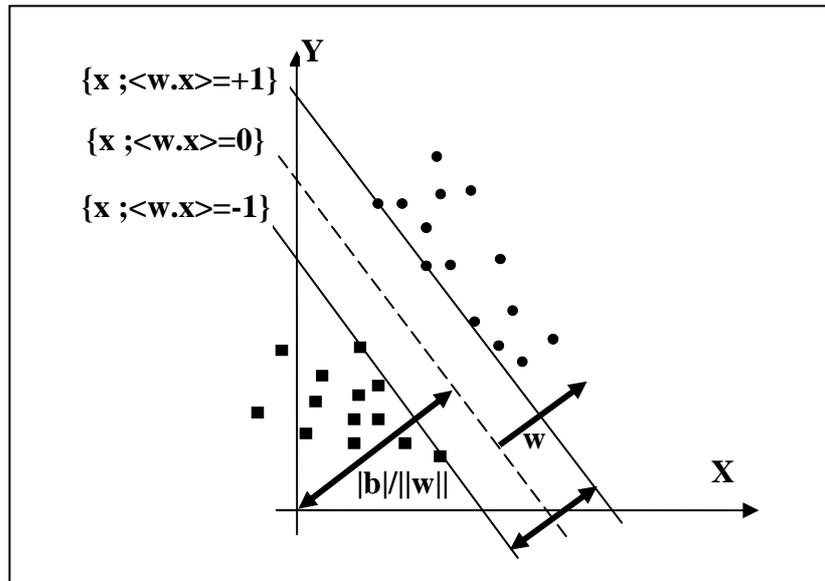


Figure 3.5 : Hyperplan séparateur et vecteurs de support.

Soient x_1 et x_2 deux points de classes différentes c'est-à-dire $f(x_1)=+1$ et $f(x_2)=-1$ alors on obtient : $(w.x_1)+b=+1$ et $(w.x_2) +b=-1$ donc $(w.(x_1-x_2))=2$, d'où : $(w/||w||).(x_1-x_2)=2/||w||$. On peut donc déduire que maximiser la marge revient à minimiser $||w||$. La minimisation de $||w||$ revient à la minimisation de $||w||^2$.

Pour qu'un point (x, y) soit bien classé, le produit $yf(x)$ doit être strictement supérieur à 0. On s'impose que $yf(x) \geq 1$ et le problème de minimisation se fera sous les contraintes suivantes :

$$\begin{cases} \min \frac{1}{2} ||w||^2 \\ \forall i, y_i (w.x_i + b) \geq 1 \end{cases}$$

Ce problème connu sous le nom problème primal est un problème convexe d'optimisation qui peut se résoudre soit par méthode directe d'optimisation quadratique si possible soit par méthode des multiplicateurs de Lagrange qui nous ramène au problème dual d'optimisation.

Problème dual : Il est réalisé en introduisant des multiplicateurs de Lagrange pour chaque contrainte du problème primal. On obtient une contrainte pour chaque exemple d'apprentissage. [MOH 06]

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i . x_j \\ \forall i, \alpha_i \geq 0 \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases}$$

C'est un problème de programmation quadratique de dimension n qui peut être résolu en utilisant des méthodes standard de programmation quadratique. Une fois la solution optimale

$\alpha^* = (\alpha^*_1, \dots, \alpha^*_n)$ du problème obtenue, le vecteur de poids de l'hyperplan à marge maximale recherché s'écrit : $w^* = \sum_{i=1}^n \alpha^*_i y_i x_i$

On appelle *vecteurs supports* les éléments x_i de l'échantillon d'apprentissage pour lesquels les coefficients α^*_i sont non nuls.

La fonction de décision associée est donc : $f(x) = \sum_{i=1}^n \alpha^*_i y_i x_i \cdot x + b$

La règle de décision est donc donnée par $sign(f(x_i))$.

1.2.3.2. SVM non linéaire

Dans la plupart des problèmes, l'hypothèse de linéarité est trop restrictive et le séparateur optimal doit pouvoir prendre une forme plus compliquée. La méthode du noyau est un moyen efficace pour traiter ce problème [ZAM 04], on transfère les données de l'ensemble de départ vers un ensemble de dimension supérieure dans lequel le problème devient probablement séparable linéairement. Soit $\Phi : \mathbf{R}^d \rightarrow F ; x \rightarrow \Phi(x)$ avec $card(F) > d$.

Le passage par une transformation des données comme le montre la figure 3.6 peut permettre une séparation linéaire des exemples.

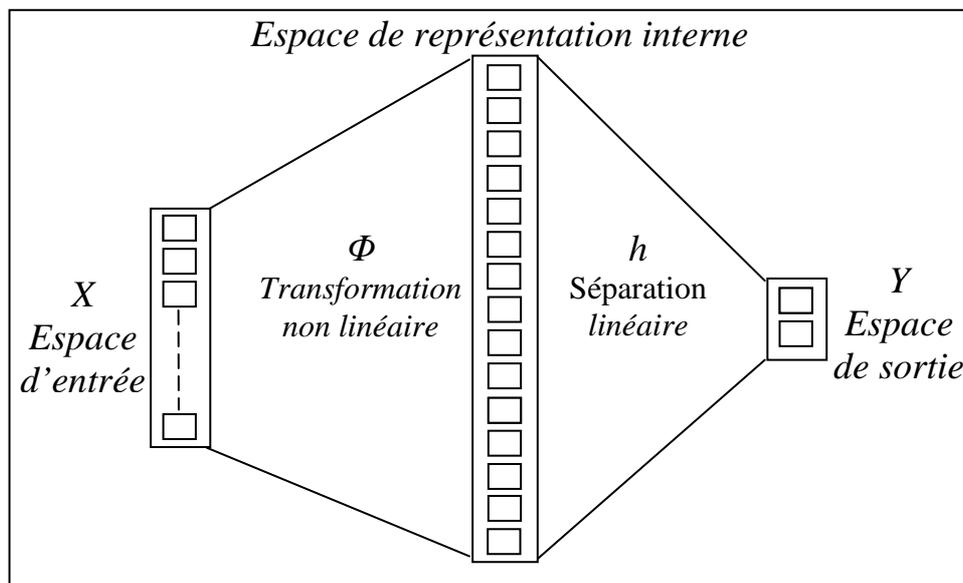


Figure 3.6 : Redescription et séparation linéaire.

On applique aux vecteurs de l'ensemble des données d'entrée une transformation non linéaire Φ , on obtient leurs redescriptions dans le nouvel espace caractéristique F , L'hyperplan séparateur obtenu dans l'espace F est appelé hyperplan optimal généralisé, il nous donne deux ensembles de points E_1 et E_2 montrés dans la figure 3.7.

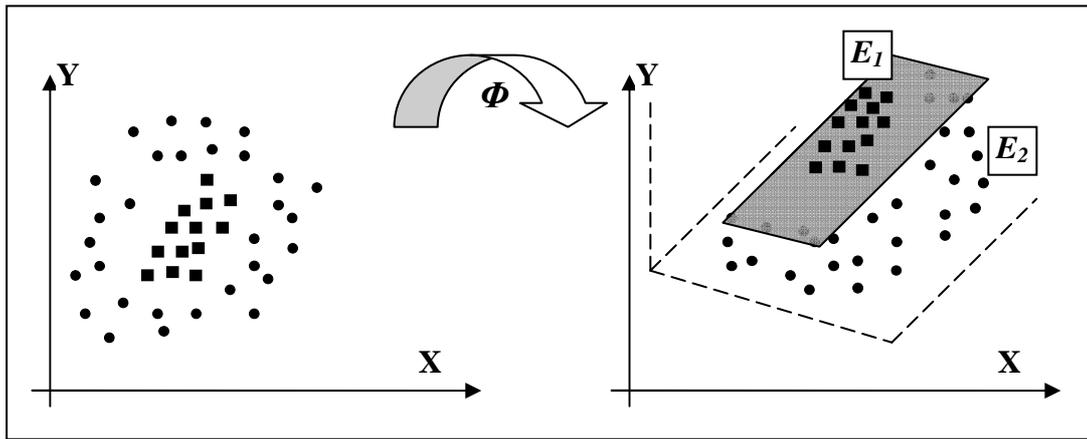


Figure 3.7 : Hyperplan optimal généralisé.

Pour optimiser le séparateur, on doit donc résoudre le problème dual suivant:

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(x_i) \cdot \phi(x_j) \\ \forall i, \alpha_i \geq 0 \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases}$$

Et la solution sera de la forme : $f(x) = \sum_{i=1}^n \alpha_i^* y_i \phi(x_i) \cdot \phi(x) + b$

Ce problème et sa solution ne dépendent que du produit scalaire $\Phi(x_i) \cdot \Phi(x_j)$ plutôt que de choisir la transformation non-linéaire $\Phi: X \rightarrow F$, on peut choisir directement une fonction $k : X \times X \rightarrow R$ appelée *fonction noyau*.

1.3. Les noyaux

1.3.1. Définition de noyau

Un noyau définit une similarité entre exemples. Cette similarité correspond à un produit scalaire dans un nouvel espace (de dimension supérieure). Le noyau définit donc implicitement une projection dans un nouvel espace, ce qui nous donne une projection qui sépare linéairement les données [GAU].

Pour représenter des données par similarité, on considère un ensemble $X = \{x_1, \dots, x_n\}$ et on peut représenter ces données dans une matrice de similarité $K : K_{ij} = K(x_i, x_j)$ où K est une fonction de similarité.

Toute fonction qui satisfait les conditions de Mercer peut être utilisée comme fonction noyau.

- **Condition de Mercer :** Pour toute fonction $k(x,z)$ symétrique et vérifiant : $\int k(x,z) \cdot f(x) \cdot f(z) \cdot dx \cdot dz \geq 0$ et pour toute fonction f telle que $\int f(x)^2 dx$ soit finie, il existe une fonction Φ telle que pour tout $x,z : k(x,z) = \Phi(x) \cdot \Phi(z)$ [MOU 08].

1.3.2. Types de noyaux

Le choix du noyau et de ses paramètres se fait généralement d'une manière heuristique lors de tentatives du type essai-erreur [FAH]. Il existe plusieurs type de noyaux, on site :

1.3.2.1. Noyaux linéaires

Les noyaux linéaires sont de la forme : $K(x_1, x_2) = (x_1 \cdot x_2)$. Ce noyau peut être utilisé pour le test de la non linéarité dans un ensemble d'apprentissage, aussi bien comme référence dans la classification à noyau.

1.3.2.2. Noyaux polynomiaux

Les noyaux polynomiaux sont de la forme : $K(x_1, x_2) = (c + \langle x_1 \cdot x_2 \rangle)^p$. Dans ce cas, on peut montrer que la règle de décision repose sur une statistique polynomiale de degré q . Parce qu'ils sont fonction du produit scalaire des observations, de tels noyaux sont dits projectifs.

1.3.2.3. Noyaux gaussiens

Les noyaux gaussiens sont des noyaux de type radial, indiquant qu'ils dépendent de la distance $\|x_1 - x_2\|$ entre les observations. Ces noyaux sont définies par :

$K(x_1, x_2) = \exp(-\|x_1 - x_2\|^2 / \beta_0)$, où β_0 est appelé largeur de bande.

Ce noyau conduit à une dimension infinie de l'espace transformé T . La frontière obtenue en utilisant un noyau gaussien ayant comme largeur de bande $\beta_0=1$ est plus complexe que celle obtenue avec un noyau polynomial de degré 2.

1.3.2.4. Noyaux exponentiels

Le noyau exponentiel est un autre exemple de noyau de type radial. Il est défini par :

$K(x_1, x_2) = \exp(-\|x_1 - x_2\| / \beta_0)$ Ce noyau fournit une surface de décision linéaire par morceaux dans l'espace des observations.

1.3.2.5. Noyaux sigmoïdaux

Le noyau sigmoïdal est défini comme étant : $K(x_1, x_2) = \tanh(\alpha_0 \langle x_1 \cdot x_2 \rangle + \beta_0)$. L'utilisation d'un tel noyau est équivalente à celle d'un réseau de neurones à une couche cachée. Ce noyau dépend de deux paramètres α_0 et β_0 , ce qui peut poser des problèmes lors de sa mise en oeuvre.

Exemple : dans le cas du XOR [MOH 06], si on prend une fonction $\Phi : (x, y) \rightarrow (x, y, x \cdot y)$ qui fait passer d'un espace de dimension 2 (figure 3.8-A-) à un espace de dimension 3, on obtient un problème en trois dimensions (figure 3.8-B-) linéairement séparable :

$$\Phi(0,0) = (0,0,0), \Phi(0,1) = (0,1,0), \Phi(1,0) = (1,0,0) \text{ et } \Phi(1,1) = (1,1,1).$$

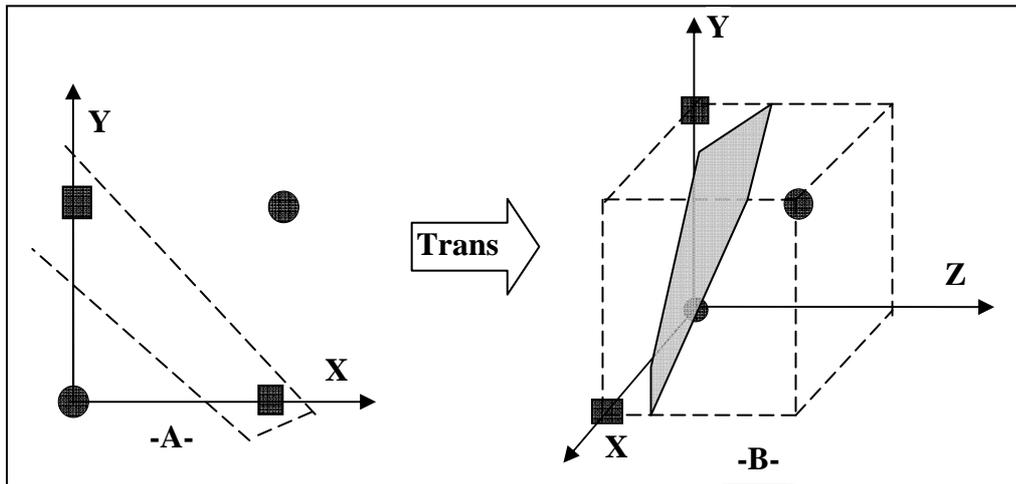


Figure 3.8 : Solution du XOR par changement d'espace.

1.4. Classes des SVM

1.4.1. Les SVM bi-classes

Une des originalités de la méthode des SVM est de produire une fonction de décision binaire par l'utilisation d'un sous-ensemble de la base d'apprentissage à partir de la minimisation d'un problème quadratique sous contraintes.

Soit $Z^A = \{z_1, z_2, \dots, z_m\} = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ une base d'apprentissage composé de m couples (vecteur d'attributs, classe) avec $x_i \in R^d$ le vecteur d'attributs, $y_i \in \{-1, +1\}$ la classe et $z_i = (x_i, y_i)$ la concaténation des deux types d'information caractérisant l'exemple numéro i .

Le problème quadratique à optimiser est le suivant :

$$\min_{\alpha} \left\{ \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y_i y_j \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^m \alpha_i \right\}$$

sous les contraintes $\begin{cases} 0 \leq \alpha_i \leq C, (1 \leq i \leq m) \\ \sum_{i=1}^m y_i \alpha_i = 0 \end{cases}$

Si α^*, b^* est la solution optimale la fonction de décision soit : $f(x) = \text{sign} \left(\sum_{i=1}^m y_i \alpha_i^* x_i + b^* \right)$

Le compromis entre erreur de classification et complexité du modèle dépend du choix du paramètre C . Dans [PLA 99], Platt a proposé un algorithme efficace pour résoudre ce problème s'appelle SMO.

1.4.2. SVM multi-classes

Les SVM étant des classificateurs binaires, la résolution d'un problème multi-classes est réalisée en le transformant en une combinaison de problèmes binaires [HSU 02], toutes les démonstrations des résultats présentés dans cette section se trouvent dans [DAR 05].

Chapitre 3 : les méthodes de segmentation par classification

Soit $Y = \{C_1, \dots, C_k, \dots, C_Q\}$ à Q classes. Une SVM multi-classes réalise des fonctions vectorielles $h = (h_k)_{1 \leq k \leq Q}$ dont les fonctions composantes $h_k(\cdot) = \langle w_k, \Phi(\cdot) \rangle + b_k$ sont construites comme les fonctions définissant les SVM bi-classes. Un individu est affecté à la classe associée à la sortie la plus élevée. Ici encore, le paramétrage est obtenu par la résolution de problème d'optimisation (3.10) suivant :

$$\min_{\alpha} \left\{ \frac{1}{2} \left\{ \sum_{i \equiv j} \sum_{k=1}^Q \sum_{l=1}^Q \alpha_{ik} \alpha_{jl} k(x_i, x_j) - 2 \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^Q \alpha_{ik} \alpha_{jy_i} k(x_i, x_j) + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^Q \alpha_{ik} \alpha_{jk} k(x_i, x_j) \right\} - \sum_{i=1}^m \sum_{k=1}^Q \alpha_{ik} \right\}$$

sous les contraintes $\begin{cases} 0 \leq \alpha_{ik} \leq C, (1 \leq i \leq m), (1 \leq k \leq Q, k \neq y_i) \\ \sum_{x_i \in C_k} \sum_{l=1}^Q \alpha_{il} - \sum_{i=1}^m \alpha_{ik} = 0, (1 \leq k \leq Q-1) \end{cases}$

La notation $i \equiv j$ signifiant l'appartenance de x_i et x_j à une même classe.

La formulation de ce problème d'optimisation fait implicitement intervenir, dans un but de simplification, des pseudo-variables α_{iy_i} , ($1 \leq i \leq m$) toutes égales à 0.

L'expression des w_k en fonction des α_{ik} et des données d'apprentissage est :

$$w_k = \sum_{x_i \in C_k} \sum_{l=1}^Q \alpha_{il} \Phi(x_i) - \sum_{i=1}^m \alpha_{ik} \Phi(x_i)$$

Les b_k se déduisant à l'optimum des conditions de Kuhn-Tucker.

1.5. Application à segmentation d'image

Les SVM sont bien adaptés aux problèmes de classification en imagerie, dans ce contexte [ZHA et al] ont proposé une méthode de segmentation de l'image médicale basée sur les SVMs, cette méthode incorpore une forme d'information sur l'objet à segmenter pour le processus de segmentation. Les résultats de segmentation sont validés par des images synthétiques, des images IMR et ultrasoniques images.

Dans [OSU et al], les auteurs ont proposé une méthode appelée 2C-SVM pour les applications de segmentations où l'erreur de classification d'un individu est plus élevée par rapport aux autres. Une variante de cette méthode est présentée dans [CHE et al] appelé 2v-SVM apporte des poids différents pour des classes différents, où les valeurs de σ et $(v+, v)$ sont prédéterminées et tout 2v-SVM donne une segmentation, un algorithme appelé STAPLE estime la solution qualifiée juste parmi l'ensemble des segmentations. [ART et al]

Dans [COL et al], les auteurs ont testé la robustesse de leur méthode de segmentation d'image qui se base sur les SVMs pour générer des contours de blessure avec 23 images acquises par des simples dispositifs et qui ne sont pas utilisées pour l'apprentissage, le taux d'erreur moyen est estimé à 5.8 %.

Conclusion

Dans cette section, on a tenté de présenter le concept des machines à vecteurs de support. Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des ensembles de données en maximisant la marge et en minimisant le nombre d'erreurs d'apprentissage. On a vu qu'il existe deux cas : les cas linéairement séparables et les cas non linéairement séparables qui nécessitent l'utilisation de fonction noyau.

2. Les fourmis artificielles

Introduction

Avec la complexité croissante des systèmes informatiques et le besoin d'avoir des systèmes adaptatifs et dynamiques, les méthodes classiques de résolution de problèmes sont devenues inefficaces. Ce qui a conduit les chercheurs à explorer de nouvelles voies et de nouveaux outils de développement. D'où l'utilisation de la notion d'émergence pour la conception de nouveaux types de systèmes artificiels. L'objectif est de concevoir des systèmes complexes constitués de petites entités en interaction entre elles et avec leur environnement et dont le comportement global est émergent et peut être qualifié d'intelligent. Ces systèmes s'inspirent en large partie de l'observation des systèmes naturels et en particulier du comportement de groupes et d'animaux sociaux.

Parmi les systèmes artificiels collectifs à fonctionnalité émergente, on trouve les algorithmes de fourmis artificielles ou les fourmis sont considérées comme des agents réactifs capables de se déplacer et de modifier leur environnement par l'intermédiaire des phéromones. Ils ont la capacité de s'adapter à un changement de l'environnement. Leur simplicité, décentralisation ainsi que le parallélisme dans leur comportement ce qui permet de résoudre efficacement des problèmes considérés NP-complets surtout problèmes modélisables sous forme d'un graphe.

Dans cette section nous présentons en détail l'algorithme de fourmis artificielles *AntClustering*.

2.1. Notions fondamentales

2.1.1. L'émergence

Le phénomène d'émergence peut être défini comme étant l'apparition d'une propriété au niveau macroscopique d'un système sans qu'elle soit préalablement programmée d'une manière explicite ni qu'elle puisse déduite à partir des propriétés des niveaux microscopiques. Un moyen de mise en œuvre de l'émergence est l'auto-organisation qui fait référence à un processus au cours duquel le système se restructure, se maintient sans nécessiter une contrainte explicite qui provient de l'extérieur du système.

2.1.2. Les phéromones

Les phéromones sont des substances chimiques (mélange d'hydrocarbures) spécifiques d'une espèce ayant une action incitative, leur transmission se fait de diverses façons. Elles se répartissent en deux grandes catégories : les phéromones de signalisation qui provoquent une réaction comportementale généralement immédiate, et les phéromones d'amorçage qui entraînent un changement physiologique à long terme.

2.1.3. La stigmergie

La stigmergie est le mécanisme de communication utilisé par les fourmis. Les fourmis utilisent l'environnement comme support de communication, elles échangent indirectement de l'information en déposant des phéromones. Plusieurs comportements collectifs chez les fourmis naturelles ont été source d'inspiration pour l'Intelligence Artificielle, on peut citer : [QUI 04]

2.2. Le comportement de fourmis

2.2.1. Activités de regroupement

Le regroupement des cadavres chez les fourmis (Figure 3.9) se fait de manière que chaque fourmi commence à chercher au hasard. Si elle rencontre un cadavre, elle le prend, et continue de se déplacer au hasard. Si elle rencontre encore un cadavre, elle recherche une place libre à côté de lui et y pose le sien. Aucun dépôt de phéromone n'est nécessaire dans ce cas. Au bout d'un certain temps, on est certain qu'il ne reste qu'un seul tas de cadavres.

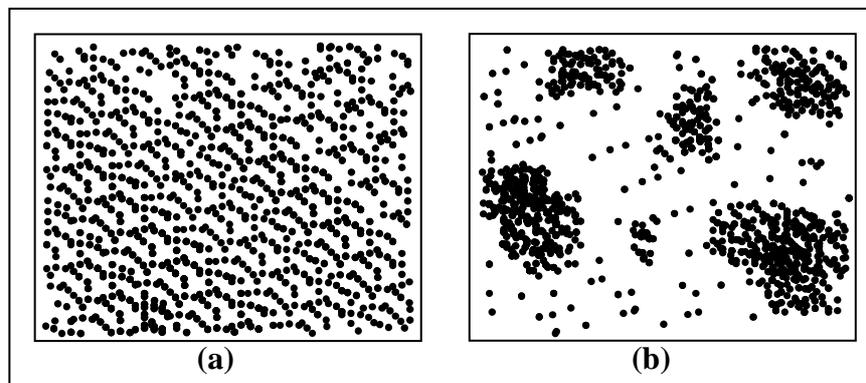


Figure 3.9 : Le regroupement des cadavres chez les fourmis.

2.2.2. Recherche de nourriture

Il s'agit cette fois de ramener le plus efficacement possible de la nourriture au nid, la recherche se fait suivant les étapes présentées dans la figure 3.10 ou une fourmi *appelée éclaireuse* parcourt plus ou moins au hasard l'environnement autour de la colonie. Si celle-ci découvre une source de nourriture, elle rentre plus ou moins directement au nid. En rentrant au nid, la fourmi laisse sur son chemin une piste de phéromones. Ces phéromones étant attractives, vont attirer les fourmis qui sont dans le nid qui vont suivre cette piste vers la source

Chapitre 3 : les méthodes de segmentation par classification

de nourriture et même les fourmis passant à proximité vont avoir tendance à suivre, de façon plus ou moins directe, cette piste. En revenant au nid, ces mêmes fourmis vont renforcer la piste.

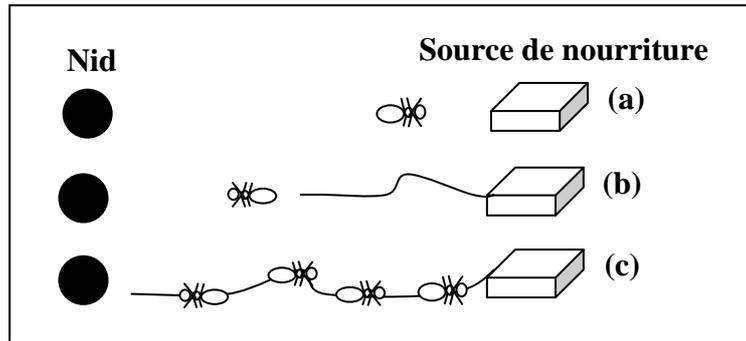


Figure 3.10 : Recherche de nourriture.

2.2.3. Choix de plus court chemin

Quand plusieurs chemins entre le nid et la nourriture sont possibles avec de longueurs inégales, les fourmis choisissent au hasard et déposent de la phéromone qui leur permet de retrouver le chemin du nid. A cause du phénomène d'évaporation, il y aura plus de phéromone sur le plus court des chemins, donc c'est la manière naturelle qu'une colonie de fourmis trouve le plus court chemin de la nourriture au nid.

2.2.4. S'adapter aux changements

C'est sur ce point que les comportements collectifs ont les qualités les plus spectaculaires. En effet, si pour une raison quelconque, les données du problème changent (le vent disperse les cadavres, une pierre tombe sur la route des fourmis,...) La colonie va continuer *comme si de rien n'était*, plus précisément va poursuivre sa tâche à partir de la nouvelle configuration du problème. La colonie va donc s'adapter naturellement à la nouvelle situation, sans avoir besoin d'une nouvelle configuration. La figure 3.11, nous montre l'adaptation des fourmis à la nouvelle situation, et elle nous montre que les fourmis choisissent le plus court chemin.

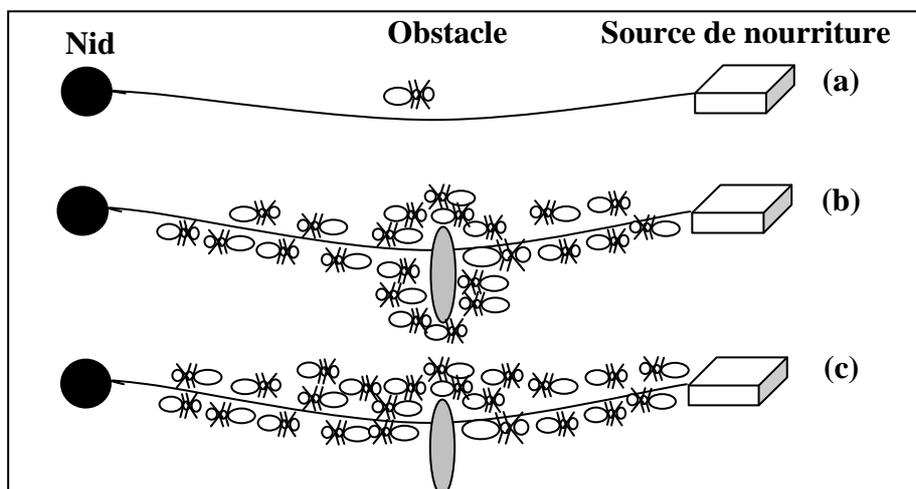


Figure 3.11 : L'adaptation aux changements.

Le rôle des phéromones :

La figure 3.12 donne une comparaison entre la collecte de nourriture avec et sans phéromone. On constate que l'amélioration induite par les phéromones est radicale, en particulier pour le début de la collecte. La collaboration entre les fourmis permet en effet d'optimiser les déplacements de celles qui sont en mode recherche. Donc, le déplacement de la majorité des fourmis se fait sur des routes de phéromones vers les zones de nourriture, ce qui donne une performance de collecte. En effet, l'absence de phéromones conduit à une répartition de façon plus uniforme des fourmis dans le monde. Certaines d'entre elles sont éloignées de toute nourriture. Donc, Les phéromones assurent une meilleure efficacité globale.

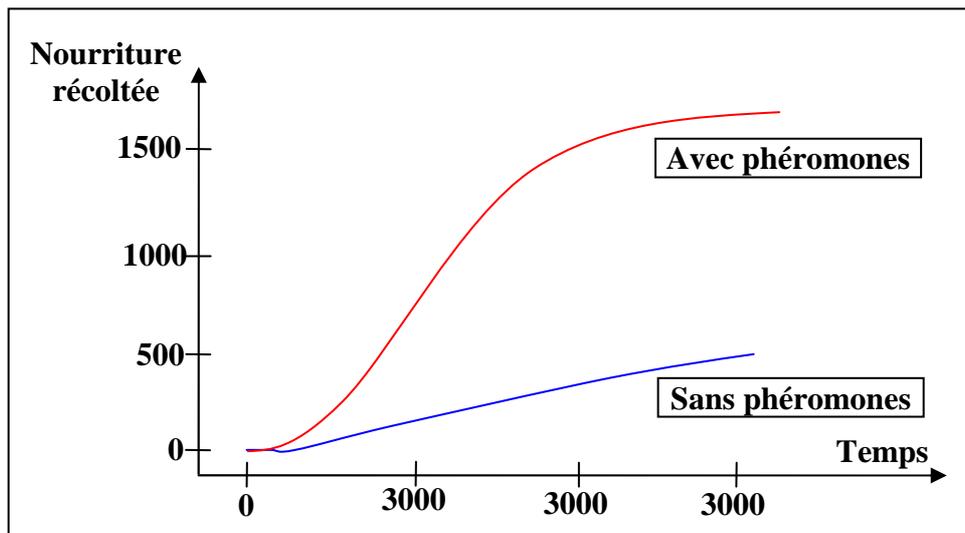


Figure 3.12 : Collecte de nourriture : avec ou sans phéromones.

2.3. Modélisation et implémentation

Dans le modèle informatique, le monde peut être constitué d'une grille de taille finie dont chaque case peut contenir une ou plusieurs fourmis, ainsi qu'un nombre arbitraire d'unités de nourriture. Certaines cases peuvent contenir des obstacles infranchissables. La fourmilière occupe une case unique [ROS 03].

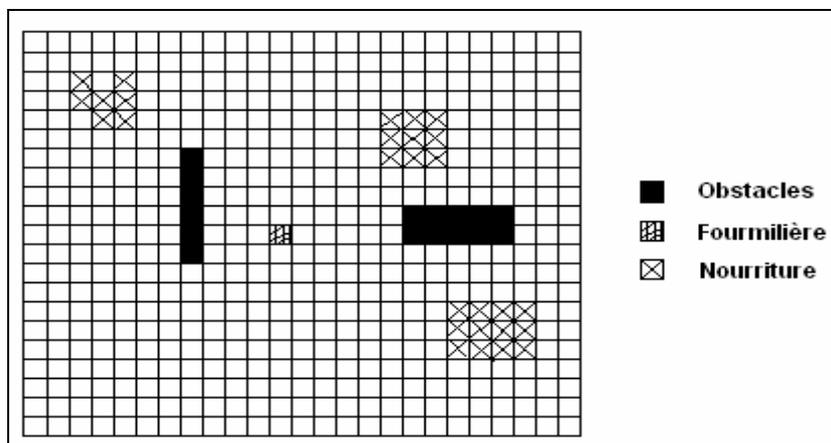


Figure 3.13 : Dessin du monde.

Quand une fourmi se déplace, elle passe directement d'une case à une des cases voisines. Le temps s'écoule de façon discrète, c'est-à-dire qu'on passe de l'instant t directement à l'instant $t+1$ en réalisant séquentiellement les mouvements de toutes les fourmis étudiées.

Une fourmi possède une orientation, c'est-à-dire qu'elle regarde une des 8 cases voisines et ne peut se déplacer que vers la case observée. En d'autres termes, pour se déplacer, la fourmi commence par s'orienter, puis avance dans la direction choisie. Une fourmi peut prendre une décision de mouvement en fonction de son environnement. Par exemple, elle peut décider de se déplacer vers de la nourriture si elle en voit dans les cases qui l'entourent. En effet plus l'évolution de la quantité de nourriture stockée dans la fourmilière au cours du temps est rapide plus la collecte de nourriture par les fourmis est efficace.

Une implémentation peut simuler plusieurs types de fourmis (stupides, l'intelligence collective, l'intelligence individuelle). Dans la nature, les fourmis exhibent un comportement collectif en utilisant un outil de communication très efficace, les phéromones qui sont aussi utilisées pour coordonner la colonie. Les fourmis utilisent en effet des phéromones pour laisser des pistes que les autres fourmis vont suivre pour s'orienter. Plus précisément, on suppose qu'une fourmi peut déposer des gouttes de phéromones sur le sol. C'est ce qu'elle va faire en mode retour à la fourmilière, l'algorithme peut donc être :

2.3.1. Algorithme de retour à la fourmilière

1. *déposer une goutte de phéromones*
2. *se tourner vers la fourmilière*
3. *avancer d'un pas*
4. *si la case atteinte est la fourmilière :*
 - (a) *déposer la nourriture*
 - (b) *faire demi-tour*
 - (c) *passer en mode recherche*
5. *sinon, retourner en 1*

Une fourmi qui porte de la nourriture fabrique donc une piste entre l'emplacement du bloc de nourriture et la fourmilière. Les fourmis en mode recherche peuvent alors profiter de la piste ainsi obtenue pour se diriger vers la nourriture. De plus, le niveau de phéromones dans une case du monde est une indication du nombre de fourmis transportant de la nourriture qui sont passées par cette case. Les fourmis en recherche ont donc intérêt à se diriger vers des cases qui contiennent beaucoup de phéromones. La nouvelle direction d'une fourmi en mode recherche est déterminée par l'algorithme suivant :

2.3.2. Algorithme de recherche

1. *si au moins une des 8 cases voisines contient de la nourriture :*
 - (a) *déterminer les cases contenant le plus de nourriture*
 - (b) *se tourner aléatoirement vers une de ces cases*
2. *sinon, si au moins une des 8 cases voisines contient des phéromones:*
 - (a) *déterminer la case contenant des phéromones la plus éloignée de la fourmilière*
 - (b) *en cas des distances égaux, choisir la case contenant le plus de phéromones*
 - (c) *se tourner vers la case choisie*
3. *choisir la prochaine direction au hasard à partir de la direction actuelle*

On constate que l'algorithme est assez évolué, mais possède des caractéristiques très intéressantes :

- l'algorithme est local : la fourmi étudie seulement les cases qui l'entourent. On peut bien sûr critiquer l'utilisation de la distance à la fourmilière, mais c'est en fait une façon simple d'exprimer le fait qu'une fourmi va naturellement suivre une piste de phéromones dans le sens qui l'éloigne de la fourmilière. Comme la fourmi connaît la direction de la fourmilière, on pourrait exprimer l'algorithme en terme de cette direction. Il serait alors moins lisible mais équivalent. Notons de plus que l'information donnant la position de la fourmilière peut être maintenue dynamiquement par la fourmi en prenant en compte ses propres déplacements ;
- la communication entre fourmis est indirecte : toute la communication est basée sur les phéromones, il n'y a pas de liaison entre les fourmis ;
- une fourmi n'a pas de mémoire, elle ne sait pas où elle est déjà passée, ou elle a déjà ramassé de la nourriture, etc. En fait, le sol joue le rôle d'une mémoire globale et répartie pour l'ensemble de fourmis.

Pour éviter que les pistes persistent quand une source de nourriture est épuisée, on donne une durée de vie aux phéromones, c'est-à-dire que celles-ci s'évaporent progressivement. Pour qu'une piste soit conservée, il faut donc qu'elle soit entretenue, c'est-à-dire utilisée par des fourmis portant de la nourriture.

Dans la suite de ce chapitre, nous présentons l'algorithme *AntClustering* ainsi que les améliorations que nous avons introduites puis nous décrivons en détail les différentes étapes de l'algorithme ainsi que leur applications sur des images IRM cérébrales afin de montrer leur efficacité.

2.4. L'algorithme *AntClustering*

AntClustering est une transposition du modèle comportemental de tri de couvain observé chez les fourmis réelles pour la résolution collective du problème de classification des images à niveau de gris. Lors du développement de l'algorithme *AntClustering*, nous avons repris les travaux existants concernant le problème de classification non supervisé [DEN 90], [LUM 94], [MON 99] et notamment [OUA 06] pour y introduire un certain nombre de modifications susceptibles d'améliorer les performances de l'algorithme en terme de coût et de qualité de classification. Dans la suite de ce chapitre, nous détaillons les éléments de base de l'algorithme *AntClustering* en terme d'environnement de fourmis et leurs comportements.

2.4.1. L'environnement des fourmis

Dans *AntClustering*, l'environnement des fourmis est un tableau de N cellules reliées chacune à un emplacement représentant le nid de la colonie afin de faciliter les déplacements des fourmis d'une cellule à une autre (Figure 3.14). Initialement les N pixels de l'image I à segmenter sont placés sur le tableau de telle sorte qu'une cellule ne contienne qu'un pixel à la fois. Durant le processus de classification une cellule peut correspondre à un ou plusieurs pixels de l'image d'origine. A la fin de l'algorithme le nombre de cellules non vides représente le nombre possible de classes présentes dans l'image. [OUD06]

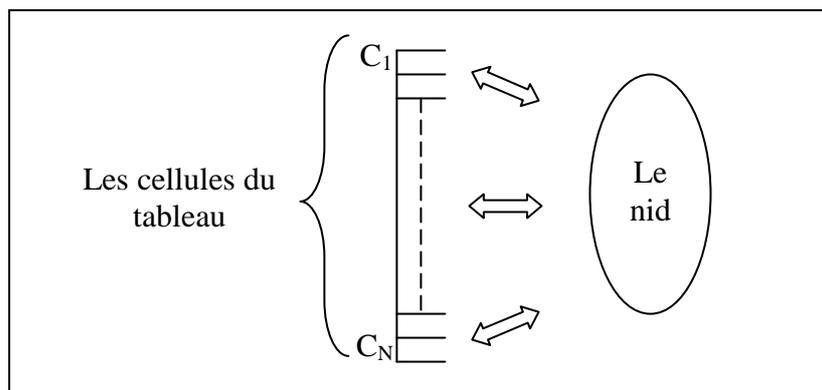


Figure 3.14 : L'environnement des fourmis artificielles

2.4.2. Le fonctionnement de l'algorithme *AntClustering*

Soit une population $A = \{a_1, a_2, \dots, a_k\}$ de K fourmis et un tableau T de N cellule contenant les N pixels de l'image I à classer. Les fourmis coopèrent ensemble et communiquent par stigmergie en déplacent les pixels de leurs cellules en fonction d'un critère de similarité entre un pixel donné et les pixels présents dans sa cellule pour former des classes homogènes.

Algorithme AntClustering

/ Initialisation*/*

Pour chaque pixel p_i faire

Placer p_i dans une cellule du tableau;

Finpour

Pour chaque fourmi a_i faire

Placer a_i dans une cellule choisie aléatoirement et lui affecter son pixel;

Etat[a_i] := porteuse;

Finpour

Déplacer toutes les fourmis vers le nid

/ Boucle principale*/*

Pour $t=1$ à t_{max} faire

Pour chaque fourmi a_i faire

Si état[a_i] = porteuse alors

Déplacer a_i vers une cellule c_k ;

Dépôt := déposer le pixel p_i qu'elle transporte dans c_k avec une probabilité $p_{dépôt}(p_i, c_k)$;

Si Dépôt = vrai alors Etat[a_i] := libre ;Finsi

Sinon

Choisir aléatoirement un pixel p_i ;

Déplacer a_i vers la cellule c_k contenant p_i ;

Porter := Porter p_i de sa cellule avec une probabilité $p_{porter}(p_i, c_k)$;

Si Porter = vrai alors Etat[a_i] := porteuse ;Finsi

Déplacer les fourmis vers le nid

Finpour

Finpour

Retourner la partition obtenue

L'algorithme *AntClustering* commence avec la phase d'initialisation dans laquelle les N pixels sont placés sur les N cellules du tableau en vérifiant qu'une cellule ne contient qu'un seul pixel, chaque fourmi choisit aléatoirement un pixel et retourne vers le nid. A la suite de cette étape, un processus de classification itératif commence : c'est un cycle simple exécuté pour chaque fourmi sélectionnée aléatoirement.

Comme dans les précédents travaux, les items comportementaux d'une fourmi sont le déplacement d'un pixel, le dépôt d'un pixel et, en plus, un comportement de retour vers le nid.

Chapitre 3 : les méthodes de segmentation par classification

La fourmi porteuse évalue l'adéquation en terme de niveau de gris qu'il y a entre le pixel qu'elle porte et les pixels contenus dans la cellule du tableau où elle voudrait le déposer à l'aide d'un critère de similarité. Plus ce critère de dissimilarité est important, plus la probabilité de le déposer sera importante. Le déplacement d'un pixel obéit à un raisonnement inverse. La fourmi évalue le critère de similarité entre le pixel considéré et les pixels présents dans sa cellule. Plus ce critère est faible, plus la fourmi aura de la chance de le déplacer de sa cellule. De cette manière, les fourmis génèrent dynamiquement des classes de niveau de gris homogènes à travers une coopération inconsciente.

Le critère d'adéquation entre un pixel p_i et les pixels d'une cellule c_k contenant n_k pixels est alors défini par la fonction de similarité suivante :

$$f(p_i, c_k) = \frac{1}{n_k} \sum_{p_j \in c_k} \frac{\alpha^2}{\alpha^2 + d(p_i, p_j)^2}$$

$$\text{Avec : } \alpha = \frac{1}{N(N-1)} \sum_{p_i \in I} \sum_{p_j \in I} d(p_i, p_j)$$

Où ng_i et ng_j sont les niveaux de gris respectifs des pixels p_i et p_j . α est une constante représentant la distance moyenne entre deux pixels de l'image I et la distance euclidienne des niveaux de gris des deux pixels: $d(p_i, p_j) = (ng_i - ng_j)/255$

La fonction de similarité f atteint son maximum quand le niveau de gris du pixel p_i est très proche de celui des pixels présents dans la cellule c_k .

2.4.2.1. Dépôt du pixel

Une fourmi porteuse effectue un certain nombre de déplacements entre son nid et le tableau afin d'explorer les cellules non vides et choisir d'une manière probabiliste la cellule où elle déposera le pixel qu'elle transporte. La probabilité de déposer un pixel p_i dans une cellule c_k est donnée par la formule suivante :

$$P_{\text{dépôt}}(p_i, c_k) = 1 - \cos^2\left(\frac{\pi}{2} f(p_i, c_k)\right)$$

Ainsi, plus la fonction de similarité entre le pixel p_i et les éléments de c_k est faible ($f(p_i, c_k) \rightarrow 0$), plus la probabilité de dépôt sera faible ($P_{\text{dépôt}}(p_i, c_k) \rightarrow 0$). Dans le cas contraire ($f(p_i, c_k) \rightarrow 1$), la fourmi a une grande chance de déposer le pixel dans la cellule ($P_{\text{dépôt}}(p_i, c_k) \rightarrow 1$).

2.4.2.2. Ramassage d'un pixel

Tout fourmi libre recherche automatiquement le prochain pixel à transporter. Afin d'éviter les déplacements infructueux des fourmis et accélérer la convergence de l'algorithme, cette recherche n'est pas aléatoire. Elle est guidée en utilisant un index commun contenant les pixels

libres (non transportés par d'autres fourmis) ainsi que la position de leurs cellules dans le tableau. Initialement, l'index contient tous les pixels de l'image.

La fourmi choisit aléatoirement un pixel p_i de l'index et se dirige directement de son nid vers la position de sa cellule c_k dans le tableau. Elle évalue localement la fonction de similarité et décide de ramasser le pixel considéré de sa cellule avec la probabilité de porter définie par la formule suivante :

$$P_{\text{porter}}(p_i, c_k) = \begin{cases} 1 & \text{si } |c_k| = 1 \\ q & \text{si } |c_k| = 2 \\ \cos^2\left(\frac{\pi}{2} f(p_i, c_k)\right) & \text{sin on} \end{cases}$$

Où $|c_k|$ est le nombre de pixels dans la cellule c_k . Ainsi si $(p_{\text{porter}}(p_i, c_k) \rightarrow 0)$, le pixel a une grande chance de rester dans sa cellule. Dans le cas contraire, $(p_{\text{porter}}(p_i, c_k) \rightarrow 1)$, le pixel a une grande chance de soit ramasser par la fourmi. Que cette décision soit négative ou positive, la fourmi revient vers son nid pour terminer son cycle.

2.4.2.3. Mémoire des fourmis

Afin d'accélérer le processus de regroupement et donc la convergence de l'algorithme, la fourmi n'a pas un mouvement complètement désordonné. Pour cela, [OUA 06] a proposé un algorithme où chaque fourmi mémorise les m derniers objets qu'elle a ramassés ainsi que leurs emplacements sur tableau. Quand une fourmi transporte un pixel p_i , elle consulte sa mémoire et évalue la possibilité de le placer dans la cellule d'un des pixels qu'elle a déjà transporté. Pour cela, elle calcule la fonction de similarité f pour chacune des cellules mémorisées. La cellule candidate à recevoir le pixel p_i sera celle pour laquelle la fonction f est maximum.

Dans l'approche de Lumer et Faieta [LUM 94] chaque fourmi mémorise les m derniers objets qu'elle a ramassés ainsi que leurs emplacements sur la grille. A chaque fois qu'elle ramasse un nouvel objet, il est comparé aux objets contenus dans sa mémoire afin de biaiser la direction qu'elle va prendre. La fourmi a tendance à se diriger vers l'emplacement où elle a déposé auparavant un objet similaire à celui qu'elle transporte actuellement.

Ce mécanisme a été étendu dans [HAN 03] en remplaçant la distance euclidienne entre deux objets par la fonction de voisinage appliquée aux positions actuelles de tous les objets contenus dans la mémoire de la fourmi. Dans [MON 99], Monmarché reprend les idées de Lumer et Faieta et remplace la comparaison des objets sur la distance les séparant par la distance entre le centre de gravité du tas transporté par la fourmi et les tas qu'elle a mémorisé (puisque dans son approche, les fourmis peuvent transporter plus qu'un objet à la fois) pour choisir le prochain emplacement de l'objet (ou du tas) qu'elle transporte.

2.3. Application à segmentation d'image

Les fourmiers artificiels sont beaucoup utilisés dans la segmentation d'image parce qu'ils sont bien adaptés à ce genre d'application. Pour cela Ouadfel [OUA] a présenté l'algorithme *AntClust* extrait automatiquement les classes présentes dans l'image sans connaître le nombre de classe *a priori* et son partition de départ. Il diffère des autres en deux points : les fourmis ne se déplacent pas sur une grille mais sur un tableau et leurs déplacements pour le ramassage et le dépôt de pixels sont guidés par leurs expériences personnelles et celles de leurs congénères. L'utilisation de principe de recherche stochastique permet d'obtenir une partition de meilleure qualité. Les expériences effectuées montrent que *AntClust* fournit de très bons résultats sur des images synthétiques et s'avère très compétitif à l'algorithme classique FCM.

Conclusion

AntClustering est un algorithme de classification non supervisée des images. Il est inspiré du comportement de tri de couvain observé chez les fourmis réelles. Dans *AntClustering*, l'environnement des fourmis est un tableau de N cases contenant les pixels de l'image et reliée chacune à un nid collectif afin de faciliter le déplacement des fourmis. À travers cet environnement, les fourmis interagissent ensemble et construisent des partitions de bonne qualité en utilisant des règles simples de placement et de déplacement de pixels. *AntClustering* utilise une fonction de similarité locale qui mesure la similarité d'un pixel en terme de niveau de gris avec les autres pixels présents dans une classe. L'intérêt principal de cet algorithme est sa capacité à extraire automatiquement les classes présentes dans l'image sans connaître le nombre de classe *a priori* et sans partition de départ. De plus cet algorithme est distribué donc facilement parallélisable.

CHAPITRE 4 : LES SYSTEMES MULTI-AGENTS

Introduction

Les systèmes informatiques contemporains sont de plus en plus compliqués et font intervenir un grand nombre de sous-systèmes hétérogènes, qui se trouvent souvent répartis dans l'espace physique (télécommunication, transport et distribution d'énergie, conduite des processus industriels, systèmes d'aide à la décision...). Ces systèmes ont montré les limites de l'approche classique de l'Intelligence Artificielle (IA) qui s'appuie sur une centralisation de l'expertise au sein d'un système unique. Pour remédier aux insuffisances de l'approche classique, une nouvelle discipline est née : l'Intelligence Artificielle distribuée (IAD).

L'IAD a pour but la distribution de l'expertise sur un ensemble de composants qui travaillent ensemble pour atteindre un objectif global. Ces composants sont appelés agents, et les systèmes correspondants sont appelés les systèmes multi-agents. Dans ce chapitre, nous allons essayer de donner une vue panoramique sur les systèmes multi-agents.

1. Définitions

1.1. Agent

Selon Ferber (1995), un agent est une entité virtuelle ou physique capable d'agir sur son environnement afin de satisfaire ses besoins, possédant une représentation partielle de cet environnement et pouvant communiquer avec d'autres agents [FEB 95].

1.2. Systèmes multi-agents (SMA)

Un système multi-agents est un ensemble d'agents autonomes capables de communiquer et d'agir sur leur environnement pour effectuer une ou plusieurs tâches. Ferber définit un système multi-agents (SMA), comme étant un système composé d'un environnement E , un ensemble d'agents A ($A \subseteq O$), un ensemble des objets O (ces objets peuvent être perçues, créés, détruits, ou modifiés par les agents), ces objets sont reliés entre eux par un ensemble de relations R , un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer, et manipuler les objets de O et des opérateurs chargés de représenter l'application des opérations et la réaction du monde à cette tentative de modification que l'on appellera les lois de l'univers [FEB 95] .

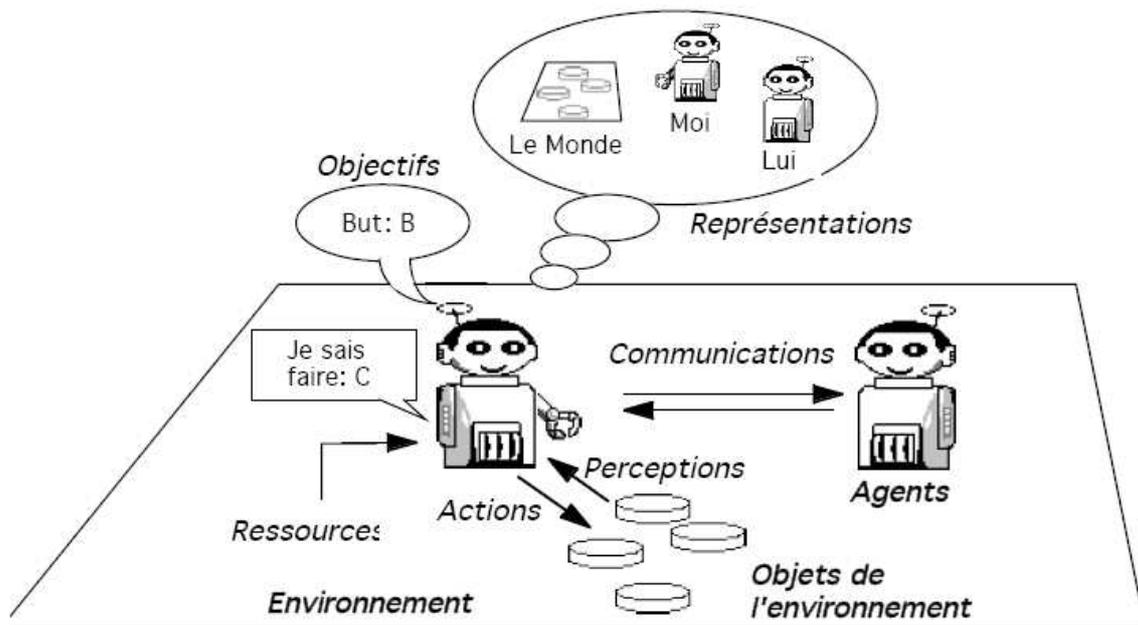


Figure 4.1: Représentation d'un agent en interaction avec son environnement et les autres agents [FEB 95]

1.3. Domaines d'application des SMA

Les systèmes multi-agents sont largement utilisés dans plusieurs domaines tel que :

- L'industrie : l'automatisation du processus de production, les robots coopératifs;
- L'enseignement à distance : gestion des cours, évaluation,...
- La santé : la supervision des malades, les systèmes de support ;
- L'information : l'assistance personnelle, recherche d'information ;
- L'énergie : la gestion des réseaux, le support d'un centre de crise ;
- Traitement d'image : la reconnaissance de forme, la segmentation, la classification...etc.

2. Typologie des agents

Il existe plusieurs classifications d'agents, et cela est dû au choix du critère sur lequel se base la classification. Donc on peut classer les agents en fonctions du degré de granularité en agents cognitifs, agents réactifs et agents hybrides, comme on peut aussi classer les agents selon leur mobilité en agents fixes et agents mobiles.

2.1. Les agents cognitifs

Les systèmes d'agents cognitifs sont basés sur la coopération d'agents capables de réaliser des opérations complexes. Ce genre de système comporte un nombre limité d'agents. Chaque agent possède une base de connaissances comprenant l'ensemble des informations et des savoir-faire aidant à la réalisation sa tâche et à la gestion de la communication avec d'autres agents. Chaque agent est assimilable à système expert, on parle alors d'agent de forte granularité.

2.2. Les agents réactifs

Cette approche se base sur l'idée qu'il n'est pas nécessaire que les agents soient intelligents individuellement pour que le système ait un comportement global intelligent. Donc, l'intelligence du système et du à la coordination et la coopération de l'ensemble d'agent de système. Dans les systèmes réactifs les agents sont de faible granularité mais nombreux. Les agents actifs sont de plus bas niveau, ils ne disposent qu'un protocole d'un langage de communication réduits, leurs capacité répandent uniquement à la loi stimulus / réponse [LAB 93].

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

Tableau 4.1 : Comparaison entre les agents cognitifs et les agents réactifs [LAB 93]

2.3. Les agents hybrides

L'agent hybride est un mélange des deux.

2.4. Les agents fixes

Un système d'agents fixe est le système le plus facile à implémenter, il est caractérisé par la possibilité d'exécution des divers agents en parallèle (sous forme de threads) ce qui permet une indépendance d'exécution des agents. Les agents de ce système d'agents peuvent être de différents types (réactif, cognitif,...etc.) et communiquer grâce à un protocole et un système de communication.

2.5. Les agents mobiles

Les agents mobiles sont des processus capables de se déplacer dans les réseaux tel que internet, interagissant avec plus des hôtes, recueillant des informations pour leurs propriétaires et accomplissant des tâches qui sont confiées. L'exemple d'agent pour la réservation de vols d'avions et l'exemple d'agent pour l'administration des réseaux de télécommunication peuvent nous donner un éclairage sur les applications de ce type d'agent.

3. Les caractéristiques des SMA

Chaque système multi-agents se distingue des autres par les choix effectués pour résoudre les problèmes qui apparaissent dans une société d'agents tels que : l'organisation sociale, le contrôle et prise de décision, la résolution des conflits, la communication et la coopération.

3.1. Organisation sociale

L'organisation sociale d'un système multi-agents est la manière dont le groupe est constitué, à un instant donnée, pour pouvoir fonctionner. Il existe deux structures d'organisation pour les sociétés d'agents : *la structure horizontale* dans laquelle, tous les agents sont au même niveau, il n'y a pas de relation maître/esclave entre les agents, et *la structure verticale* dans laquelle, les agents sont structurés par niveaux, dans un même niveau on retrouve localement une structure horizontale, les agents de cette société fonctionnent de manière qu'un agent reçoit le problème à résoudre d'un autre agent qui lui est supérieur dans la hiérarchie.

Dans les sociétés horizontales, le contrôle est le résultat des mécanismes de contrôle locaux aux agents. Par contre dans les sociétés hiérarchisées certains agents superviseurs peuvent prendre eux même le contrôle de quelques éléments de la société.

Les mécanismes de décision font référence à tout ce qui est allocation de tâches ou de ressources, en relation avec la résolution de conflit. Pour ce faire, les agents peuvent coordonner leurs activités et négocier leurs actions pour arriver à une situation cohérente. Une bonne négociation est caractérisée par un nombre faible d'agents impliqués et un protocole minimal. Ils regroupent tous les parties décisionnelles des protocoles de négociation [LAB 93].

3.2. Coopération

La résolution distribuée d'un problème est le résultat de coopération d'un nombre d'agents. Quelque soit la structure de la société d'agents, un agent peut coopérer selon les modèles décrites dans le tableau suivant : [LAB 93]

<i>modèle</i>	<i>description</i>
<i>Coopération par commande</i>	Un agent supérieur A décompose le problème en sous-problèmes qu'il répartit entre les autres agents X_i .
<i>Coopération par partage de tâches et de résultats</i>	Un agent A partage leurs tâches et leurs résultats avec autres agents, avec la possibilité de prendre en compte localement les plans des autres.
<i>Coopération par appel d'offre</i>	Un agent A décompose le problème en sous-problèmes dont il diffuse la liste. Chaque agent X_i qui le souhaite avoir une offre, A choisit parmi celles-ci et distribue les sous-problèmes.
<i>Coopération par compétition</i>	Un agent A décompose et diffuse la liste des sous-problèmes, chaque agent X_i résout un ou plusieurs sous-problèmes et envoie les résultats à A qui à son tour fait le tri.

Tableau 4.2 : Les modèles de coopération.

3.3. Engagement/Planification

Un agent coopératif planifie ses actions par coordination et négociation avec les autres agents. En construisant un plan pour atteindre un but, l'agent se donne les moyens d'y parvenir et donc s'engage à accomplir les actions qui satisfont ce but : l'agent croit qu'il est en mesure d'exécuter tout le plan qu'il a élaboré, ce qui le conduit (ainsi que les autres agents) à agir en conséquence.

3.4. Communication

La résolution coopérative des problèmes est basée sur la communication. Elle permet de synchroniser les actions des agents et résoudre les conflits par négociation. Il existe deux modes de communication dans les SMA :

- **Par partage de ressources :** Les agents qui utilisent ce mode de communication ne sont pas directement liés entre eux. Ils communiquent à travers une zone de données commune.
- **Par envoi des messages :** Dans les systèmes fonctionnant par envoi de messages, chaque agent peut communiquer directement avec les autres en utilisant souvent des langages pour structurer le message tel que : KQML, FIPA ACL.

3.5. Adaptativité

Un agent adaptatif est un agent capable de contrôler ses aptitudes (communicationnelles, comportementales, etc.) selon l'agent avec qui il interagit. Autrement, un agent adaptatif est un agent d'un haut niveau de flexibilité. [LAB 93]

4. Le développement des SMA

4.1. Les méthodologies de développement des SMA

Pour qu'on puisse mettre en place un bon système multi-agents, différentes méthodologies de développement des systèmes multi-agents ont été proposées, on peut les classer en deux catégories : la première regroupe les méthodologies s'inspirant des méthodologies de développement orienté objet telle que : la méthodologie AAI, GAIA, et AUML, la deuxième regroupe les méthodologies s'inspirant des systèmes à base de connaissances telle que : la méthodologie MAS-CommonKADS.

4.1.1. La méthodologie AAI

La méthodologie AAI (Australian Artificial Intelligence Institute) a été développée par l'institut australien de l'intelligence artificielle, elle est basée sur l'architecture BDI (beliefs, desires, intentions) et sur un système de raisonnement procédural. Dans cette méthodologie, le SMA a deux modèles : un modèle interne représente l'implémentation des agents et un modèle externe définit les agents et les relations entre agents, les classes d'agents et les instances

associées, les rapports d'héritage entre classe d'agents et les instanciations de ces classes lors de l'exécution.

La méthodologie AAI commence par une étape qui identifie les rôles de l'application et développer une hiérarchie des classes d'agents, puis identifier les responsabilités à chaque rôle, les services demandés et fournis par chaque rôle et déterminer les objectifs associés à chaque service, puis pour chaque rôle, déterminer les plans associés suivant les conditions d'exécution et enfin, déterminer la structure des croyances du système. [1]

4.1.2. La méthodologie GAIA

La méthodologie GAIA était développée par Wooldredge (2001) pour permettre au concepteur des systèmes multi-agents de passer d'un cahier des charges aux détails de l'implémentation du système, c'est-à-dire elle permet de passer des concepts abstraits (les rôles, les permissions, les responsabilités, les protocoles) aux concepts concrets (les types d'agents, les services et les connaissances) [MBA 03].

Cette méthodologie est basée sur le concept de rôle qui est défini par quatre attributs (responsabilités, permissions, activités et protocoles). Les responsabilités indiquent la responsabilité de l'entité. Afin de remplir les responsabilités, le rôle dispose d'un ensemble de permissions qui définit les ressources accessibles au rôle. Les activités d'un rôle sont les traitements que ce dernier peut effectuer sans communiquer avec d'autres agents.

4.1.4. La méthodologie Cassiopea

Contrairement aux méthodologies AAI et GAIA, Cassiopea a une vision par le bas. Donc on commence par spécifier les comportements nécessaires pour réaliser les tâches. Cette méthodologie comporte trois étapes :

- Identification des comportements élémentaires de l'ensemble du système ;
- Identification des relations entre les comportements élémentaires ;
- Identifications des comportements organisationnels de l'ensemble du système.

4.1.4. La méthodologie AUML

UML est un langage unifiant les différents paradigmes d'analyse et conception orienté objet des systèmes. UML est un langage non une méthodologie, mais il y a une méthodologie associée à UML qui s'appelle « Rational Unified Process ». Plusieurs chercheurs ont vu en UML un point de départ pour développer des outils et méthodes pour les systèmes multi-agents. Pour cela, Odell et d'autres chercheurs ont proposé des extensions à UML afin de modéliser les agents.

Les modifications proposées à UML permettent la représentation des protocoles d'interactions entre agents et la définition du concept de rôle qui permet de modéliser un agent qui joue plusieurs rôles [MBA 03].

4.1.5. La méthodologie MAS-CommonKADS :

La méthodologie MAS-CommonKADS est une extension du modèle défini dans CommonKADS qui rajoute des techniques pour modéliser les agents et les interactions entre agents. La méthodologie définit les modèles décrits ci-dessous pour l'analyse et la conception d'un système pour chaque modèle, la méthodologie définit les constituants et les rapports entre ces derniers. La méthodologie MAS-CommonKADS comporte les modèles suivants :

- Le modèle d'agents : décrit les caractéristiques principales des agents ;
- Le modèle de tâches : définit les tâches devant être exécutées par les agents ;
- Le modèle d'expertise : décrit les connaissances nécessaires aux agents pour réaliser les tâches ;
- Le modèle de coordination : définit les itérations entre agents ;
- Le modèle d'organisation : définit l'organisation de la société d'agents ;
- Le modèle de communication : décrit les interactions entre l'agent humain et le logiciel.
- Le modèle de conception : c'est le modèle globale du système qui regroupe tous les modèles précédents. [1]

4.2. Les plates-formes des systèmes multi-agents

Une plate-forme multi-agents est un outil qui assiste le développeur tout au long du processus de construction d'un SMA (analyse, conception, développement et déploiement). Parmi les plates-formes les plus connues on a : Agent Builder, Jade, Madkit et Zeus.

4.2.1. Agent Builder

Agent Builder est une suite d'outils permettant de construire des agents intelligents. Elle a été développée en Java par Rectular System. La création du comportement des agents se fait à base du modèle BDI et au langage AGENT-O. Le langage de communication utilisé KQML (Knowledge Query Manipulating Language). Agent Builder est composé d'une interface graphique et d'un langage orienté agent permettant de définir les croyances, les engagements et les actions [RIC 01].

4.2.2. Madkit

Madkit (Multi Agent Development Kit) est une plate-forme multi-agents libre développée en 1996 par Olivier Getknechi et Michel Ferber au laboratoire d'informatique, de robotique et de microélectronique de Montpellier (LIMM) de l'université de Montpellier II dans le langage Java selon le modèle d'organisation Alaadin (Agent, Groupe, Rôle).

Elle est développée en elle est doté d'environnement de développement graphique qui permet facilement la construction des applications. Dans cette plate-forme, chaque agent a un rôle et peut appartenir à un groupe. [1]

4.2.4. Jade

Jade (Java Agent DEvelopment Framework) est une plate-forme multi-agents développé par le laboratoire TILAB dans le langage Java pour le développement des applications multi-agents conforme à la norme FIPA. Elle fournit des classes pour la définition du comportement des agents. Dans la plate-forme Jade, un agent est un processus autonome et indépendant communicant entre eux par envoi du message structurée en FIPA-ACL. Jade possède trois modules principaux: [FOU 02]

- DF « Director Facilitator » fournit un service de « pages jaunes » à la plate-forme ;
- ACC « Agent communication Channel » gère la communication entre les agents ;
- AMS « Agent Management System » supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

4.2.4. Zeus

Zeus est une plate-forme multi-agents conçue et réalisée par British Telecom. Zeus est écrit dans le langage Java conformément au norme de la FIPA(Fondation International of Processing Agent). Elle regroupe principalement les composantes suivantes :

- Une boîte aux lettres et un gestionnaire de messages qui analyse les messages de la boîte aux lettres et les transmet aux composantes appropriées ;
- Un moteur de coordination ;
- Un planificateur qui planifie les tâches de l'agent en fonction des décisions du moteur de coordination, des ressources disponibles et des spécifications des tâches ;
- Plusieurs bases de données représentant les plans connus par l'agent, les ressources et l'ontologie utilisée ;
- Un contrôleur d'exécution qui gère l'horloge locale de l'agent et les tâches actives. [1].

5. Apports des SMA pour la segmentation d'images

Les SMA permettent de faire cohabiter des agents de toutes natures, depuis des agents réactifs jusqu'à des agents cognitifs évolués, et de les faire évoluer en parallèle. Ces considérations ont conduit à concevoir et à étudier différentes *architectures logicielles* qui doivent proposer des solutions pour la représentation et l'organisation les informations manipulées par le système, l'extraction de nouvelles informations, la focalisation des différents traitements nécessaires à l'interprétation des images, enfin la coordination et le contrôle

d'exécution de ces traitements. Un état de l'art de ces différents aspects est présenté notamment dans [BOU 99] et [GER 00].

5.1. La représentation et l'organisation des informations

Les connaissances sont toutes les informations qui sont utilisées dans les processus de décision. On peut distinguer les connaissances opératoires qui permettent de définir les actions à entreprendre, les connaissances descriptives données *a priori*, ou les informations produites pendant le processus d'interprétation qui sont utilisées, à une étape de traitement donnée, pour produire de nouvelles informations.

La façon d'associer les différentes informations entre elles est un problème clé. La difficulté réside alors dans le choix des modes de représentation des informations qui s'effectuent donc en fonction de deux objectifs, d'une part intégrer de la façon la plus riche possible les informations (créer des liens), d'autre part préserver leur accessibilité, et faciliter leur partage, afin notamment de permettre la formalisation de contexte d'exécution pour les opérateurs et de favoriser les capacités de raisonnement du système. Pour segmenter une image, le système a en effet souvent besoin de connaissances de haut niveau pour effectuer ses traitements de plus bas niveau, alors que ces connaissances sont la plupart du temps des résultats de traitements, et ne sont pas nécessairement disponibles immédiatement, d'où la recherche de meilleurs interconnexions entre les niveaux de représentation. Des représentations sous forme de réseaux sémantiques et de graphes ont été utilisées, car elles permettent de formaliser les liens existant entre connaissances. Les SMA répondent bien à ces objectifs [GER 99].

5.2. L'extraction de nouvelles informations

Les informations manipulées par le système sont soit fournies *a priori* par le concepteur ou par un module d'apprentissage, soit acquises par le système lui-même en cours d'exécution. Au cours du processus de segmentation, ces informations doivent être accumulées de manière incrémentale, afin d'augmenter progressivement les connaissances du système et d'adapter constamment les traitements en fonction des informations déjà récoltées.

Les images IRM sont parfois incomplètes, imprécises et potentiellement erronées. Pour pouvoir les utiliser afin de produire de nouvelles informations de façon robuste, il est nécessaire d'évaluer et de gérer leur incertitude. Différents formalismes numériques modélisant l'incertain ont été proposés, tels que les approches probabilistes, ou encore les approches floues [BLO 96]. Ils permettent d'introduire des mécanismes d'inférence de façon à tirer des conclusions provisoires (dont l'incertitude est quantifiée), nécessaires à la progression de l'interprétation et à l'estimation des modèles.

5.3. La focalisation des traitements

La segmentation d'image est un processus distribué en termes de buts à atteindre, de zones de l'image à explorer et de méthodes à appliquer. Les traitements sont focalisées en faisant des choix sur la zone d'image à explorer, les informations à chercher et sur les algorithmes et les méthodes et les stratégies à utiliser et en utilisant un certain nombre de connaissances organisées suivant cet espace tridimensionnel pour produire de nouvelles informations.

La distribution des traitements va de pair avec la mise en oeuvre de *coopérations* par le système, qui doit combiner plusieurs sources d'informations ou modalités de traitement pour accomplir une tâche donnée, et tirer parti, par exemple, des complémentarités entre méthodes ou opérateurs et augmenter la robustesse du processus de segmentation. Un état de l'art peut être trouvé dans [GER 00].

5.4. La coordination et le contrôle de traitements

La nature distribuée du processus de segmentation (distribution spatiale, opératoire, ...) fait naître des besoins de coordination des traitements et de maintien de la cohérence entre les informations et traitements locaux et globaux. Pour cela le contrôle peut être centralisé ou décentralisé, sachant que les décisions sont prises globalement par un coordinateur central ou localement par différents acteurs autonomes en interaction (Agents). La décentralisation du contrôle est notamment un moyen pour mettre en oeuvre une stratégie réactive et opportuniste. Les systèmes multi-agents permettant de mettre en oeuvre un contrôle décentralisé.

5.5. Les approches multi-agents proposées pour la segmentation

5.5.1. Mise en oeuvre d'un SMA pour la segmentation

Lors de la mise en oeuvre d'un système multi-agents pour la segmentation d'image, le principal effort réside dans la répartition des tâches entre les agents ainsi que la définition de leurs comportements pour obtenir la solution du problème posé. Il est donc nécessaire de définir un modèle d'agent, le comportement associé ainsi qu'un protocole de communication inter agents. En effet, la cohérence des comportements doit s'exprimer à travers des similarités de comportement entre des agents générés dans le but de réaliser une même tâche (segmentation distribuée d'un tissu par plusieurs agents par exemple).

Les aspects *d'adaptativité* peuvent également être pris en compte dans de tels systèmes, ce qui est intéressant dans le contexte de l'inhomogénéité des niveaux de gris en IRM. L'agent doit s'adapter aux situations rencontrées. Cet aspect concerne plutôt la spécialisation des agents qui doit leur permettre d'avoir un comportement en adéquation avec le problème posé. En IRM, les agents doivent garder le même comportement quelle que soit la séquence d'acquisition choisie. Enfin, les agents doivent exécuter une fonctionnalité, c'est-à-dire

participer à la résolution d'un des buts du problème posé, comme la détection d'un tissu ou d'un contour.

Dans [GER 98], Germond a retenu 3 axes principaux qui lui semblent essentiels dans les systèmes multi-agents pour une application à la segmentation :

- 1- **Distribution** : la distribution de l'activité dans le système permet de répartir les tâches et de spécialiser localement les agents mis en oeuvre afin d'obtenir une meilleure réaction à l'environnement [FER 95]. elle permet également d'exploiter la simultanéité d'exécution (*parallélisme*) d'un ensemble d'agents et de tirer parti de l'émergence progressive d'une solution.
- 2- **Adaptation** : dans le cadre d'un système de segmentation, la proposition des agents fonctionnent en réaction à un environnement dynamique semble tout à fait appropriée. En effet, un ensemble d'images possèdent des caractéristiques variées constitue un environnement changeant dans lequel toute modélisation complète est quasi-impossible.
- 3- **Contrôle** : Les modèles de systèmes multi-agents proposent généralement d'abandonner l'idée d'un contrôle global. Plus précisément, l'absence de contrôle global revient à laisser une population d'agents s'auto organiser afin d'aboutir à une solution. Cela suppose une grande capacité de communication entre agents. La communication directe entre agents, permettant en théorie une absence de contrôle global, l'ajustement des comportements des agents est assuré par le partage des connaissances. Par exemple, le schéma architectural de Brooks [BRO 91] propose de laisser émerger des fonctionnalités à partir d'un ensemble de comportements par opposition à une architecture planifiée qui définit des comportements à partir de fonctionnalités.

5.5.2. Agents contours/régions

De nombreux travaux exploitent deux grands types d'agents de segmentation : des agents de croissance de contours et des agents de croissance de région. Pour les agents de type région le principe est de définir un pixel initial appelé germe. Puis, itérativement, d'examiner la frontière de la zone en cours de construction et de choisir d'agréger à la région un pixel selon des critères à définir. La croissance d'une région cesse lorsque aucun des pixels de la frontière ne vérifie les critères d'agrégation.

Les agents de type contours sont également définis à partir d'un germe et effectuent une croissance en examinant les pixels situés aux extrémités du contour en cours de construction.

A chaque étape, le pixel vérifiant au mieux un critère d'agrégation est sélectionné parmi les candidats potentiels et agrégés au contour. Comme pour les agents région, la croissance cesse lorsque aucun des pixels candidats ne satisfait le critère d'agrégation.

5.5.3. Image de type quelconque

Dans le cadre de segmentation d'image, plusieurs de systèmes multi-agents ont été proposés. Ces systèmes sont généralement destinés à des applications spécifiques, qui ont pour objectif de traiter, d'interpréter et d'analyser des images ayant des caractéristiques propres nécessitant des opérateurs, des modèles et des traitements adaptés. Différents types d'agents sont utilisées : agents cognitifs [BOI 93], agents réactifs [LIU 99] et agents situés [BOU 99].

Dans [MAZ et al], *Mazouzi et al* présentent et évaluent une approche qui consiste à l'utilisation d'une population d'agents autonomes pour la segmentation d'une image de profondeur en ses différentes régions. Les agents s'adaptent aux régions sur lesquelles ils se déplacent, puis effectuent des actions coopératives et compétitives produisant une segmentation collective de l'image. Les résultats expérimentaux obtenus par des images réelles montrent le potentiel de l'approche proposée pour l'analyse des images de profondeurs, et ce vis-à-vis de l'efficacité de segmentation et de la fiabilité des résultats.

Dans [KHO et al], Khouadjia et al présentent une approche adaptative de segmentation d'images basée sur une architecture multi-agents. Dans cette approche, les agents sont déployés sur l'image tel que chaque agent est équipé d'une habilité à estimer l'homogénéité d'une région dans une localité prédéfinie. Chaque agent exhibe plusieurs comportements réactifs en réponse au stimulus local. Cette dernière peut migrer, se reproduire, ou bien se diffuser au sein de l'image. L'interaction entre plusieurs comportements permet l'émergence d'une nouvelle fonctionnalité, à savoir la segmentation de l'image. Les résultats obtenus sont très satisfaisant et montrent l'efficacité de l'approche.

Dans [BOI 93], l'auteur propose ainsi un système d'agents cognitifs, dans lequel les agents ont la possibilité de raisonner et de définir leurs propres buts et sont organisés de façon explicite au moyen de structures de coordination.

Dans [BOU 99], une société d'agents situés est développée pour segmenter et analyser une séquence d'images de cellules en mouvement. Chaque agent est lancé à une position donnée de l'image, avec un but donné (segmenter une partie de la cellule par exemple son noyau ou son cytoplasme). Ils sont spécialisés à l'aide de connaissances *a priori* sur les objets à segmenter et dotés de plusieurs comportements, de perception (d'exploration et de marquage de l'image par croissance de région), d'interaction (de négociation pour fusionner les régions) et de reproduction (dans la même image et dans les images suivantes de la séquence temporelle).

5.5.4. Image IRM

Dans le cadre de segmentation d'image IRM, aussi plusieurs de systèmes multi-agents ont été proposés. Certaines de ces approches reposent sur l'introduction d'agents très simple et

réactifs, qui sont situés dans l'image où ils doivent explorer et sur lequel ils déposent des marques, les étiquettes formant l'interprétation de l'image.

Dans [SEM 06] les auteurs proposent une approche floue possibiliste implémentée dans un SMA qui a donné de bons résultats lors de la classification des structures cérébrales (MG, MB, LCR). L'approche décrite dans [HAR 04] propose un système multi-agents basé sur une approche de segmentation floue \ croissance de régions appliquée à des IRM cérébrales.

Laurence dans [GER 99] a distribué les tâches de segmentation sur trois agents spécialisés, un *agent_matière grise* pour segmenter la matière grise, *agent_matière blanche* pour segmenter la matière blanche, et *agent_LCR* pour segmenter le liquide céphalo-rachidien.

Dans [SCH et al], Scherrer et al ont proposé une approche markovienne coopérative qui se fonde sur le raffinement mutuel des segmentations en tissus et en structures. La connaissance *a priori* nécessaire à la segmentation des structures est apportée par une description floue de l'anatomie cérébrale. Cette approche markovienne dynamique et coopérative est implémentée dans un environnement multi-agents : des entités autonomes distribuées dans l'image estiment des modèles markoviens locaux et coopèrent pour assurer leur cohérence. L'évaluation est réalisée à la fois sur des images fantômes et sur des images réelles acquises à 3 Tesla.

Dans le système proposé par [LIU 99], des agents réactifs sont conçus pour étiqueter des structures fines et homogènes, comme le cortex cérébral. Ces agents sont dotés de différents comportements, de perception, d'étiquetage, de reproduction et de diffusion.

Conclusion

Dans ce chapitre, nous avons essayé de donner une vue globale sur le monde des systèmes multi-agents. Au début, nous avons justifié l'existence de ce genre de systèmes. Puis, on a donné une définition de ce type de système et ses caractéristiques, les différentes méthodologies et plates-formes pour la création des systèmes multi-agents. Enfin et dans le contexte de la segmentation des IRM cérébrales, nous savons que les systèmes multi-agents ont montré leur efficacité lorsque plusieurs méthodes interviennent, donc l'utilisation d'une approche adaptative à plusieurs méthodes pour segmenter des IRM, en plus l'hétérogénéité des informations, nécessite un SMA. En effet, la construction de ce type de systèmes comporte toutes les difficultés inhérentes aux systèmes répartis, auxquelles s'ajoute le caractère de flexibilité des interactions entre agents. En conclusion, on peut dire que les systèmes multi-agents gagnent encore du terrain dans la résolution distribuée des problèmes. Cela est dû aux apports de cette approche tel que : la modularité, le parallélisme, la coopération, la réutilisation.

Dans la suite de ce mémoire nous allons présenter l'approche proposée, et nous justifions le choix des outils et les paramètres de chaque algorithme.

CHAPITRE 5 : IMPLEMENTATION PAR LES SYSTEMES MULTI-AGENTS

Introduction

Différentes méthodes de segmentation d'images cérébrales ont été présentées dans les sections précédentes, dans cette partie nous allons présenter notre approche, qui consiste à concevoir une architecture multi-agents adaptative pour la détection et la localisation des structures. Pour cela nous avons choisi deux types d'algorithmes, un algorithme supervisé «*SVMs*» et un algorithme non supervisé «*AntClustering*» pour profiter les avantages de chaque type.

Pour mettre en oeuvre cette approche, les traitements sont programmés en utilisant le paradigme «*multi-agents*» décrit dans le chapitre précédent, ces systèmes étant construits à partir d'agents «*adaptatifs*», pour chaque algorithme, une population d'agents est lancée dans l'image à segmentée, chaque agent fait classifier un ou plusieurs pixels de l'image, l'ensemble des résultats obtenu par les agents interprète l'image segmentée. L'architecture des agents, les choix des paramètres de chaque algorithme seront détaillés le long de ce chapitre.

1. Choix de l'algorithme

Pour notre approche, puisque nous aurons dans la suite à segmenter des images IRM de différentes modalités, l'emploi d'un algorithme supervisé fournit des très bons résultats mais il nécessite, comme nous l'avons vu dans le deuxième chapitre, une base d'apprentissage pour chaque classe et pour chaque patient. La création de cette base peut s'avérer fastidieuse pour les experts. De plus, les méthodes supervisées en segmentation d'images médicales peuvent être très dépendantes de la base d'apprentissage. Cela constitue l'inconvénient majeur de ce type de méthodes. Par contre l'emploi d'un algorithme non supervisé ne pose aucun problème pour les spécialistes, leur intérêt est qu'elles ne nécessitent aucune phase d'apprentissage ou d'étiquetage manuel préalable. La seule intervention de l'expert se situe à la fin du processus pour identifier les classes calculées avec les classes biologiques [SEM 08].

Ces raisons nous ont fait préférer de combiner l'algorithme supervisé basé sur les SVM et l'algorithme non supervisé *AntClustering*.

1.1. L'algorithme supervisé SVMs

Un SVM est un classificateur binaire qui sépare les données d'une population en deux classes, ce classificateur appelle hyperplan séparateur qui est caractérisé par une fonction de décision soit linéaire ou non linéaire d'ou l'utilisation d'une fonction dite noyau qui permet une séparation optimale des données. Cette fonction est obtenue par apprentissage d'un sous ensemble d'une base d'apprentissage.

1.1.1. Classification linéaire

Dans le cas où les données sont linéairement séparables, la fonction de décision f s'écrit sous la forme linéaire suivante: $f(x_i) = w \cdot x_i + b$ avec $(w, b) \in R^d \times R$.

L'opérateur $\langle \cdot \rangle$ désigne le produit scalaire usuel dans R^d , w est le vecteur des poids et b est le biais de modèle linéaire. Les paramètres (w, b) doivent être estimés à partir des données d'apprentissage.

L'hyperplan optimal est celui pour lequel la distance aux points les plus proches est maximale. Cette distance est donnée par : $d(x) = |w \cdot x + b| / \|w\|$

Soient x_1 et x_2 deux points de classes différentes c'est-à-dire $(w \cdot x_1) + b = +1$ et $(w \cdot x_2) + b = -1$ donc $(w \cdot (x_1 - x_2)) = 2$; d'où : $(w \cdot (x_1 - x_2)) / \|w\| = 2 / \|w\|$. On peut donc déduire que maximiser la marge revient à minimiser $\|w\|$. La minimisation de $\|w\|$ revient à la minimisation de $\|w\|^2$.

Pour qu'un point (x, y) soit bien classé, le produit $y \cdot f(x)$ doit être strictement supérieur à 0. On s'impose que $y \cdot f(x) \geq 1$ et le problème de minimisation se fera sous les contraintes suivantes :

$$\begin{cases} \min \frac{1}{2} \|w\|^2 \\ \forall i, y_i (w \cdot x_i + b) \geq 1 \end{cases}$$

Ce problème connu sous le nom problème primal. Leur résolution se fait par méthode des multiplicateurs de Lagrange nous ramène au problème dual d'optimisation qui est réalisé en introduisant des multiplicateurs de Lagrange pour chaque contrainte du problème primal. On obtient une contrainte pour chaque exemple d'apprentissage [MOH 06].

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ \forall i, \alpha_i \geq 0 \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases}$$

Ce problème peut être résolu par des méthodes standard de programmation quadratique.

Si α^* est la solution optimale, le vecteur de poids de l'hyperplan à marge maximale s'écrit :

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i$$

La fonction de décision associée est donc :

$$f(x) = \sum_{i=1}^n \alpha_i^* y_i x_i \cdot x + b$$

La règle de décision est donc donnée par $sign(f(x_i))$.

1.1.2. Classification non linéaire

Dans le cas où les données ne sont pas linéairement séparables, on transfère les données de l'ensemble de départ vers un ensemble de dimension supérieure dans lequel le problème devient probablement séparable linéairement par l'utilisation d'une fonction dite noyau.

Soit $\Phi : R^d \rightarrow F ; x \rightarrow \Phi(x)$ avec $card(F) > d$.

Pour optimiser le séparateur, on doit donc résoudre le problème suivant:

$$\left\{ \begin{array}{l} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(x_i) \cdot \phi(x_j) \\ \forall i, \alpha_i \geq 0 \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{array} \right.$$

Et la fonction f sera de la forme :

$$f(x) = \sum_{i=1}^n \alpha_i^* y_i \phi(x_i) \cdot \phi(x) + b$$

1.1.3. Les noyaux

Un noyau définit une similarité entre exemples, cette similarité correspond à un produit scalaire dans un nouvel espace (de dimension supérieure).

Soit un ensemble $x = \{x_1, \dots, x_n\}$, K est une fonction de similarité s'il existe une fonction Φ telle que pour tout $x_i, x_j : K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. Cette fonction peut être représentée par une matrice de similarité $K : K_{ij} = K(x_i, x_j)$

Une détermination des noyaux est représentée dans le tableau suivant:

Type de noyaux	Forme générale $K(x_1, x_2) =$	Cas d'utilisation
Noyaux linéaires	$(x_1 \cdot x_2)$	Ce noyau peut être utilisé pour le test de la non linéarité dans un ensemble d'apprentissage.
Noyaux polynomiaux	$(c + \langle x_1; x_2 \rangle)^p$	Dans ce cas, on peut montrer que la règle de décision repose sur une statistique polynomiale de degré p .
Noyaux gaussiens	$exp(-\ x_1 - x_2\ ^2 / \beta_0)$ où β_0 est appelé largeur de bande.	Ce noyau conduit à une dimension infinie de l'espace transformé. La frontière obtenue lorsque $\beta_0 = 1$ est plus complexe que celle obtenue avec un noyau polynomial de degré 2
Noyaux exponentiels	$exp(-\ x_1 - x_2\ / \beta_0)$	Ce noyau fournit une surface de décision linéaire par morceaux dans l'espace des observations.
Noyaux sigmoïdaux	$tanh(\alpha_0 \langle x_1; x_2 \rangle + \beta_0)$	L'utilisation d'un tel noyau est équivalente à celle d'un réseau de neurones à une couche cachée.

Tableau 5.1 : détermination des noyaux.

1.1.4. Variantes des SVMs

Comme nous avons vu dans le troisième chapitre, les SVMs servent à classifier les données en deux classes, dans leur version originale. D'autres variantes sont introduites pour interpréter le multi classes dans beaucoup de travaux.

1.1.4.1. Les SVMs bi-classes

Dans les SVMs bi-classes, on cherche à classifier les données en deux classes.

Soit $Z_A = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ une base d'apprentissage composé de m couples (vecteur d'attributs, classe) avec $x_i \in \mathbb{R}^d$ le vecteur d'attributs et $y_i \in \{-1, +1\}$ la classe.

Le problème quadratique à optimiser est le suivant :

$$\min_{\alpha} \left\{ \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y_i y_j \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^m \alpha_i \right\}$$

$$\text{sous les contraintes : } \begin{cases} 0 \leq \alpha_i \leq C, (1 \leq i \leq m) \\ \sum_{i=1}^m y_i \alpha_i = 0 \end{cases}$$

Si α^*, b^* est la solution optimale la fonction de décision soit :

$$f(x) = \text{sign} \left(\sum_{i=1}^m y_i \alpha_i^* x_i + b^* \right)$$

1.1.4.2. Les SVMs multi-classes

Dans les SVMs multi-classes, on cherche à classifier les données en Q classes.

Soit $Y = \{C_1, \dots, C_k, \dots, C_Q\}$ a Q classes. Un SVM multi-classes réalise des fonctions vectorielles $h = (h_k)_{1 \leq k \leq Q}$ dont les fonctions composantes $h_k(\cdot) = \langle w_k, \Phi(\cdot) \rangle + b_k$ sont construites comme les fonctions définissant les SVM bi-classes. Un individu est affecté à la classe associée à la sortie la plus élevée. Ici encore, le paramétrage est obtenu par la résolution de problème d'optimisation suivant :

$$\min_{\alpha} \left\{ \frac{1}{2} \left\{ \sum_{i \neq j} \sum_{k=1}^Q \sum_{l=1}^Q \alpha_{ik} \alpha_{jl} k(x_i, x_j) - 2 \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^Q \alpha_{ik} \alpha_{jy_i} k(x_i, x_j) + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^Q \alpha_{ik} \alpha_{jk} k(x_i, x_j) \right\} - \sum_{i=1}^m \sum_{k=1}^Q \alpha_{ik} \right\}$$

$$\text{sous les contraintes } \begin{cases} 0 \leq \alpha_{ik} \leq C, (1 \leq i \leq m), (1 \leq k \leq Q, k \neq y_i) \\ \sum_{x_i \in C_k} \sum_{l=1}^Q \alpha_{il} - \sum_{i=1}^m \alpha_{ik} = 0, (1 \leq k \leq Q-1) \end{cases}$$

La notation $i \equiv j$ signifiant l'appartenance de x_i et x_j à une même classe.

La formulation de ce problème d'optimisation fait implicitement intervenir, dans un but de simplification, des pseudo-variables α_{iy_i} , ($1 \leq i \leq m$) toutes égales à 0.

L'expression des w_k en fonction des α_{ik} et des données d'apprentissage est :

$$w_k = \sum_{x_i \in C_k} \sum_{l=1}^Q \alpha_{il} \Phi(x_i) - \sum_{i=1}^m \alpha_{ik} \Phi(x_i)$$

Les b_k se déduisant à l'optimum des conditions de Kuhn-Tucker.

1.1.5. Les SVMs et imagerie par résonance magnétique

Les images médicales présentent des caractéristiques spécifiques et le niveau de gris ne permet pas une représentation complète des tissus. L'utilisation d'un algorithme de classification supervisé (SVMs) pour caractériser les tissus cérébraux présente l'inconvénient de nécessité d'une base d'apprentissage. Afin d'obtenir cette base, une manière simple consiste à demander à un expert de déterminer des régions d'intérêt de construire une base d'apprentissage. De plus, la qualité des résultats de SVMs dépendre de la qualité de la base d'apprentissage utilisée. Tous ça, nous a conduit à l'utilisation d'une méthode non supervisées (*AntClustering*) pour caractériser les tissus cérébraux.

1.2. L'algorithme non supervisée *AntClustering*

Dans cette section, nous présentons l'algorithme *AntClustering* pour la segmentation d'IRM. Cet algorithme est basé sur une population des agents fournis non intelligents capables de s'auto-organiser pour créer un comportement global complexe et faire émerger collectivement une segmentation optimale de l'IRM par l'intermédiaire des interactions qu'ils entretiennent avec leur environnement.

AntClustering utilise les principes d'exploration stochastique et distribuée d'une population de fourmis artificielles pour fournir une segmentation d'une image en des classes pertinentes sans disposer d'une partition de départ et sans connaître le nombre de classes qui seront nécessaires.

1.2.1. Formalisation du problème

Nous considérons une image IRM comme un ensemble de N pixels $\{p_1, \dots, p_N\}$ que l'on désire regrouper en des classes aussi homogènes que possible en terme de niveau de gris. Nous considérons aussi une population A de K fourmis $\{a_1, \dots, a_k\}$ qui coopèrent ensemble et communiquent par stigmergie pour fournir une classification optimale.

Initialement on a N classes constituées chacune d'un pixel enregistrée dans une case d'un tableau de N cases. Au cours du processus de classification les fourmis déplacent les pixels d'une classe à une autre afin de regrouper dans une même classe le maximum de pixels similaires en terme de niveau de gris par l'évaluation de la mesure de similarité entre un pixel

et le centre de gravité de la classe k suivante : $f(p_i, c_k) = \frac{1}{1 + \left(\frac{ng_i - ng_k}{\beta} \right)^2}$

β est un paramètre qui contrôle la dilatation de la fonction f .

Un schéma représentant l'évolution de la fonction f est donné dans la figure 5.1. [OUA]

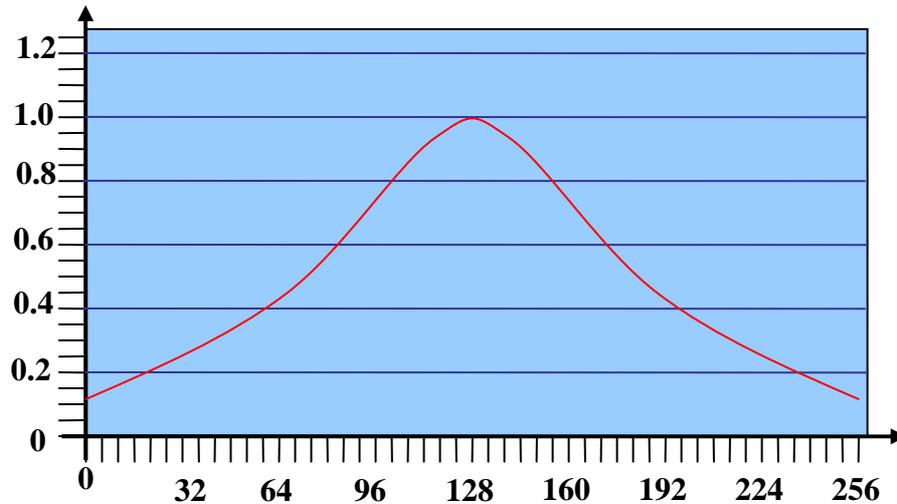


Figure 5.1. La fonction de similarité pour $ng_i = 128$ et $g_k \in [0, 255]$, $\beta = 50$

Ce tableau possède les propriétés suivantes: [OUA]

- Chaque case du tableau est reliée au nid des fourmis ce qui facilite le déplacement des fourmis sur le tableau et peut contenir un nombre illimité de pixels similaires.
- Initialement il y'a sur le tableau autant de case que de pixels à regrouper et chaque case ne contient qu'un seul pixel.

1.2.1.1. L'algorithme principal

L'algorithme commence avec une phase initiale dans laquelle (1) les N pixels sont placés aléatoirement sur les cases du tableau; (2) les A fourmis sont déplacées de leur nid et disposées aléatoirement sur les cases du tableau en vérifiant qu'une case ne peut contenir qu'une seule fourmi à la fois; et (3) la fourmi ramasse un pixel de la case où elle se trouve. A la suite de cette étape, le processus de classification commence : c'est une boucle simple, dans laquelle (4.1) une fourmi est sélectionnée aléatoirement ; (4.1.1) elle revient vers son nid et se déplace vers une case guidée par une information heuristique; et (4.1.2) la fourmi décide d'y déposer le pixel qu'elle transporte selon une règle probabiliste. (4.1.3) une fois qu'elle devient libre, elle effectue de nouveaux déplacements entre le nid et les cases du tableau afin de rechercher le prochain pixel à transporter. (4.1.4) le ramassage d'un pixel est aussi effectué selon une règle probabiliste. Cette boucle est répétée pour chacune des fourmis. A la fin du processus de classification, le nombre de classes intéressantes de l'image correspond au nombre de cases non vides présentes sur le tableau. [OUA]

Le schéma général de l'algorithme *AntClustering* est le suivant :

1. Placer les N pixels de l'image chacun dans une case du tableau;
2. Initialiser aléatoirement les positions des K fourmis;
3. pour chaque fourmi, ramasser le pixel de la case où elle se trouve;
4. Pendant T_{max} itérations faire
 - 4.1. Pour chaque fourmi faire
 - 4.1.1. Revenir au nid;
 - 4.1.2. Choisir la prochaine case c vers laquelle elle se déplacera et y déposer le pixel qu'elle transporte avec une probabilité;
 - 4.1.3. Revenir au nid;
 - 4.1.4. Se déplacer vers la case contenant le prochain pixel à transporter et le ramasser avec une probabilité;

Dans la suite nous allons décrire en détail les règles de déplacements, de ramassage et de dépôt de pixels que les fourmis vont utiliser sur le tableau pour classer les pixels de l'image.

1.2.1.2. Déplacements des fourmis

Durant le processus de classification, les fourmis vont se déplacer régulièrement entre leur nid et les cases du tableau pour transporter ou bien déposer un pixel. Afin d'accélérer le processus de regroupement et donc la convergence de l'algorithme, la fourmi n'a pas un mouvement complètement désordonné.

Quand une fourmi transporte un pixel, on l'autorise à accéder à son voisinage immédiat. Elle calcule la fonction de similarité pour chacune des cases des 8 voisins du pixel qu'elle transporte et évalue ainsi directement la possibilité de le déposer dans une des ses cases candidates. Le meilleur emplacement sera celui pour lequel la fonction de similarité est maximum. La fourmi décide alors de déposer son pixel sur cet emplacement avec une probabilité. Si cette décision est négative, la fourmi garde le pixel qu'elle transporte, et essaye d'autres cases choisies aléatoirement jusqu'à ce qu'elle arrive à le déposer.

1.2.1.3. Ramassage d'un pixel

La probabilité de transporter un pixel p_i de sa cellule c_k est définie par la formule suivante :

$$P_{\text{porter}}(p_i, c_k) = \begin{cases} 1 & \text{si } |c_k| = 1 \\ q & \text{si } |c_k| = 2 \\ \frac{k_p}{k_p + f(p_i, c_k)} & \text{sin on} \end{cases}$$

Où $|c_k|$ est le nombre de pixels dans la cellule c_k . Ainsi si la cellule c_k du pixel sélectionné ne contient que lui, il est systématiquement ramassé par la fourmi. Si la cellule contient deux

pixels, la fourmi a une probabilité q de ramasser le pixel p_i . Enfin si la cellule contient plus de deux pixels : la probabilité p_{porter} de transporter le pixel p_i est importante ($p_{porter}(p_i, c_k) \rightarrow I$), quand la fonction de similarité entre le pixel p_i et les pixels présents dans sa cellule est faible ($f(p_i, c_k) \rightarrow 0$). Dans le cas contraire ($f(p_i, c_k) \rightarrow I$), ($p_{porter}(p_i, c_k) \rightarrow 0$), le pixel a une grande chance de rester dans sa cellule [OUA 06].

1.2.1.4. Dépôt du pixel

Si une fourmi transporte un pixel, elle explore son voisinage immédiat pour choisir la case vers laquelle elle se déplacera pour y déposer avec une probabilité donnée par la formule suivante :

$$P_{\text{dépôt}}(p_i, c_k) = \begin{cases} 1 & \text{si } f(p_i, c_k) \leq \min_{p_i \in c_k} (f(p_i, c_k)) \\ \frac{f(p_i, c_k)}{f(p_i, c_k) + k_d} & \text{sin on} \end{cases}$$

Ainsi, plus la fonction de similarité entre le pixel p_i et les éléments de c_k est faible ($f(p_i, c_k) \rightarrow 0$), plus la probabilité de dépôt sera faible ($p_{dépôt}(p_i, c_k) \rightarrow 0$). Dans le cas contraire si la fonction de similarité est importante ($f(p_i, c_k) \rightarrow I$), alors ($p_{dépôt}(p_i, c_k) \rightarrow I$) la fourmi a une grande chance de déposer le pixel dans la cellule [OUA 06].

1.2.2. Les paramètres de AntClustering

Un certain nombre de paramètres qui ont une influence direct sur la performance de *AntClustering* dont les valeurs peuvent dépendre ou non de l'image à segmenter. Le tableau suivant résume les valeurs des paramètres de *AntClustering* avec lesquels nous avons obtenu de nos résultats.

Paramètre	Description	Valeur
A	Nombre de fourmis	20
k_d	contrôle la probabilité de déplacer un pixel d'une case	0.3
k_p	contrôle la probabilité de déposer un pixel dans une case	0.1
Q	Contrôle la probabilité de choisir un pixel dans une case contenant deux pixels	0.7

Tableau 5.2 : les paramètres de *AntClustering*.

1.2.3. AntClustering et imagerie par résonance magnétique

En raison de la nature complexe des interfaces entre tissus (MB/MG et MG/LCS par exemple), un nombre significatif de pixels contient un mélange de deux ou trois tissus. De plus, les IRM sont affectées par un bruit inhérent à la méthode d'acquisition et aux méthodes de

reconstruction. Pour cela, nous nous sommes orientés vers l'algorithme de classification supervisé SVMs pour caractériser les tissus cérébraux.

2. Choix des paramètres des algorithmes

Dans notre approche, Nous utilisons les algorithmes de classification SVMs et *AntClustering* pour la localisation des structures cérébraux dans les images IRM. Pour ce la, il nous faut définir les différents paramètres gouvernant ces méthodes, à savoir les valeurs de C , le choix des poids η_i , la métrique utilisée et enfin les vecteurs forme représentant les pixels des images.

2.1. Initialisation des algorithmes

Le problème de la classification par SVMs s'exprime comme une minimisation d'une fonctionnelle, sous certaines contraintes. Puisque les SVMs sont supervisées, elles nécessitent une base d'apprentissage. Pour obtenir cette base, plusieurs stratégies ont été proposées dans la littérature. La plus simple consiste à demander à un expert de déterminer des régions d'intérêt de construire une base d'apprentissage. Comme tout algorithme de classification supervisé, les résultats de SVMs vont évidemment dépendre de l'étape d'apprentissage.

Par contre la classification par *AntClustering* s'exprime par convergence vers une solution plus ou moins optimale. *AntClustering* consiste à lancer une population de fourmis dans l'image à segmentée. Chaque fourmi agit sur son environnement (image) par déplacement des pixels et laisser des marques, l'image segmenter émerge du comportement collectif des fourmis. Les résultats de *AntClustering* ne sont pas nécessairement correcte et vont évidemment dépendre de temps d'émergence et de nombre de fourmis.

2.2. Détermination du nombre de classes

La localisation des structures cérébrales se fait selon deux grands approches : l'approche région ou nous cherchons donc à segmenter la matière blanche, la matière grise, le liquide cérébro-spinal et le fond d'écran. Donc le nombre de classe C sera pris égal à quatre. L'approche contour où nous cherchons les contours des structures cérébrales et le fond d'écran. Donc le nombre de classe C sera pris égal à deux. Cependant, et comme nous cherchons des contours des structure pour les localiser, le nombre des classes prises dans les SVMs est de deux ($C=2$) et Quatre ($C=4$) pour les *AntClustering*.

2.3. Choix des vecteurs forme

Le choix des vecteurs forme est fondamental puisque leur pertinence va permettre de discriminer les pixels entre eux. Ce choix est défini suivant le type de modalité. L'image anatomique que nous utilisons est L'imagerie par résonance magnétique qui est une modalité d'imagerie multi-spectrale donnant accès à un grand nombre de paramètres et donc de vecteurs forme. Le signal est la caractéristique la plus intéressante qui peut être exploitée principalement

par l'intermédiaire d'images pondérées en T_1 , T_2 et en densité de protons. Le vecteur forme x_j d'un pixel j est alors formé des niveaux de gris de ce pixel dans toutes les images. Cette information est très largement utilisée en segmentation d'images, mais elle est dans ce cas très sensible aux variations du signal dues à l'instrumentation.

Kiviniitty [KIV 84] affirme que les paramètres T_1 et T_2 suffisent à discriminer correctement les tissus sains dans des images IRM. Just *et al.* [JUS 88] partagent cette opinion, mais notent que certaines entités pathologiques (tumeurs, oedèmes) sont caractérisées par un grand nombre de valeurs dans les images pondérées en T_1 , T_2 et en densité de protons, et que cette variété peut affecter l'analyse de ces entités [SEM 08].

3. Algorithmes utilisés

3.1. Algorithmes SVM

a) L'algorithme primal :

Entrée: S : l'ensemble d'apprentissage linéairement séparables, $\eta \in \mathbb{R}^+$: la vitesse

d'apprentissage ;

$w_o \leftarrow 0$; $b_o \leftarrow 0$; $k \leftarrow 0$; $R \leftarrow \max_{1 \leq i \leq \ell} \|x_i\|$;

Répéter

Pour $i=0$ à ℓ faire

Si $(y_i(\langle w_k, x_i \rangle + b_k) \leq 0)$ alors

$w_{k+1} \leftarrow w_k + \eta \cdot y_i \cdot x_i$;

$b_{k+1} \leftarrow b_k + \eta \cdot y_i \cdot R^2$;

$k \leftarrow k + 1$;

Finsi

Finpour

Jusqu'à $(k > k_{max})$;

Retourner (w_k, b_k) ;

b) L'algorithme dual :

Entrée: S : l'ensemble d'apprentissage linéairement séparables;

$\alpha \leftarrow 0$; $b \leftarrow 0$; $R \leftarrow \max_{1 \leq i \leq \ell} \|x_i\|$;

Répéter

Pour $i=0$ to ℓ faire

Si $(y_i(\sum_{j=1}^{\ell} \alpha_j y_j \langle x_j, x_i \rangle + b) \leq 0)$ alors

$\alpha_i \leftarrow \alpha_i + 1$;

$b \leftarrow b + y_i \cdot R^2$;

Finsi

Finpour

Jusqu'à ($k > k_{max}$);

Retourner (a, b) pour définie la fonction $h(x)$;

L'algorithme d'apprentissage des SVMs résultant dans le cas non-biais est le suivant : Noter que chaque exemple d'apprentissage doit avoir leur propre vitesse d'apprentissage η_i .

c) L'algorithme (3) :

Entrée: *S: l'ensemble d'apprentissage linéairement séparables, $\eta \in \mathbb{R}^\ell$: la vitesse d'apprentissage ;*

$\alpha \leftarrow 0$;

Répéter

Pour $i=0$ to ℓ faire

$$\alpha_i \leftarrow \alpha_i + \eta_i \cdot (1 - y_i \sum_{j=1}^{\ell} \alpha_j y_j \cdot K(x_j, x_i)) ;$$

Si ($\alpha_i < 0$) alors $\alpha_i \leftarrow 0$;

sinon $\alpha_i \leftarrow C$;

Finsi

Finpour

Jusqu'à (condition);

Retourner α ;

d) l'algorithme d'optimisation :

Entrée: *S = $\{(x_1, y_1) \dots ; (x_\ell, y_\ell)\}$: l'ensemble d'apprentissage linéairement séparables;*

L'hyperplane (w, b) qui résoudre le problème d'optimisation en dessous réalise la maximisation de la marge géométrique $\gamma = 1 / \|w\|^2$.

minimise $w, b \langle w, w \rangle$;

subject to $y_i (\langle w, x_i \rangle + b) \geq 1 / i=1 \dots \ell$;

Ou bien

Entrée: *S = $\{(x_1, y_1) \dots ; (x_\ell, y_\ell)\}$: l'ensemble d'apprentissage qui soit linéairement séparables dans le future espace définie par le noyau $K(x, z)$ et supposant que les paramètres α^* et b^* résoudre le problème d'optimisation quadratique suivant :*

$$\text{maximise } \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_j \cdot y_i \cdot \alpha_j \cdot \alpha_i \cdot K(x_j, x_i)$$

$$\text{subject to } \sum_{i=1}^{\ell} y_i \cdot \alpha_i = 0 / \alpha_i \geq 0, i = 1 \dots \ell$$

La règle de décision est donnée par: $\text{sgn}(f(x))$, avec $f(x) = \sum_{i=1}^{\ell} y_i \cdot \alpha_i^* \cdot K(x_i, x) + b^*$.

e) Le Pseudo code de l' Algorithme SMO [NEL 00]

```
target = desired output vector
point  = training point matrix
procedure takeStep(i1, i2)
  if (i1 == i2) return 0
  alph1 = Lagrange multiplier for i1
  y1 = target[i1]
  E1 = SVM output on point[i1] - y1 (check in error cache)
  s = y1 * y2
  Compute L, H
  if (L == H) return 0
  k11 = kernel(point[i1] ,point[i1])
  k12 = kernel(point[i1] ,point[i2])
  k22 = kernel(point[i2] ,point[i2])
  eta = 2*k12-k11-k22
  if (eta < 0) {
    a2 = alph2 - y2*(E1-E2)/eta
    if (a2 < L) a2 = L
    else if (a2 > H) a2 = H
  }else {
    Lobj = objective function at a2=L
    Hobj = objective function at a2=H
    if (Lobj > Hobj+eps) a2 = L
    else if (Lobj < Hobj-eps) a2 = H
    else a2 = alph2
  }
  if (|a2-alph2| < eps*(a2+alph2+eps)) return 0
  a1 = alph1+s*(alph2-a2)
  Update threshold to reflect change in Lagrange multipliers
  Update weight vector to reflect change in a1 & a2, if linear SVM
  Update error cache using new Lagrange multipliers
  Store a1 in the alpha array
  Store a2 in the alpha array
  return 1
endprocedure
procedure examineExample(i2)
```

```
y2 = target [i2]
alph2 = Lagrange multiplier for i2
E2 = SVM output on point [i2] - y2 (check in error cache)
r2 = E2*y2
if ((r2 < -tol && alph2 < C) || (r2 > tol && alph2 > 0)){
  if (number of non-zero & non-C alpha > 1){
    i1 = result of second choice heuristic
    if takeStep(i1,i2) return 1
  }
  loop over non-zero and non-C alpha, starting at random point {
    i1 = identity of current alpha
    if takeStep(i1, i2) return 1
  }
  loop over all possible i1, starting at a random point {
    i1 = loop variable
    if takeStep(i1, i2) return 1
  }
}
return 0
endprocedure
main routine:
  initialize alpha array to all zero
  initialize threshold to zero
  numChanged = 0
  examineAll = 1
  while (numChanged > 0 | examineAll){
    numChanged = 0
    if (examineAll)
      loop I over all training examples
        numChanged += examineExample(I)
    else
      loop I over examples where alpha is not 0 & not C
        numChanged += examineExample(I)
    if (examineAll == 1)    examineAll = 0
    else if (numChanged == 0)examineAll = 1
  }
//John Platt, reprinted with permission [PLA 99].
```

3.2. Algorithmes *AntClustering*

a) *l'algorithme principal (AntClustering)* : [OUA 06]

Pour chaque pixel p_i faire /* Initialisation*/

Placer p_i dans une cellule du tableau;

Finpour

Pour chaque fourmi a_i faire

Placer a_i aléatoirement dans une cellule et lui affecter son pixel;

Etat[a_i] := porteuse;

Finpour

Déplacer toutes les fourmis vers le nid;

Pour $t=1$ à t_{max} faire /* Boucle principale*/

Pour chaque fourmi a_i faire

Si état[a_i] = porteuse alors

Déplacer a_i vers une cellule c_k ;

Dépôt := déposer le pixel p_i qu'elle transporte dans c_k ; avec une
probabilité $p_{dépôt}(p_i, c_k)$;

Si Dépôt = vrai alors Etat[a_i] := libre; Finsi

Sinon Choisir a_i aléatoirement un pixel p_i ;

Déplacer a_i vers la cellule c_k contenant a_i ;

Porter := Porter p_i de sa cellule avec une probabilité $p_{porter}(p_i, c_k)$;

Si Porter = vrai alors Etat[a_i] := porteuse; Finsi

Déplacer les fourmis vers le nid;

Finpour

Finpour

Retourner la partition obtenue

b) *Algorithme de retour à la fourmilière* :

1. déposer une goutte de phéromones;

2. se tourner vers la fourmilière;

3. avancer d'un pas;

4. si la case atteinte est la fourmilière:

4.1. déposer la nourriture;

4.2. faire demi-tour;

4.3. passer en mode recherche;

5. sinon, retourner en 1;

c) Algorithme de recherche :

4. si au moins une des 8 cases voisines contient de la nourriture:
 - (a) déterminer les cases contenant le plus de nourriture;
 - (b) se tourner aléatoirement vers une de ces cases;
5. sinon, si au moins une des 8 cases voisines contient des phéromones:
 - (a) déterminer la case contenant des phéromones la plus éloignée de la fourmilière;
 - (b) en cas d'égalité, choisir la case contenant le plus de phéromones;
 - (c) se tourner vers la case choisie;
6. choisir la prochaine direction au hasard à partir de la direction actuelle;

3.3. Algorithme général de l'approche

Etape 01 : fixer les paramètres.

Les entrées: $X=(x_j, j=1..N)$ l'ensemble des vecteurs forme, N : nombre de classes. k_d : contrôle la probabilité de déplacer un pixel d'une case, k_p : contrôle la probabilité de déposer un pixel d'une case, A : nombre de fourmis ;

Etape 02 : lancer l'algorithme d'initialisation ;

Etape 03 : lancer l'algorithme SVMs et/ou lancer l'algorithme AntClustering;

Les sorties : image segmentée.

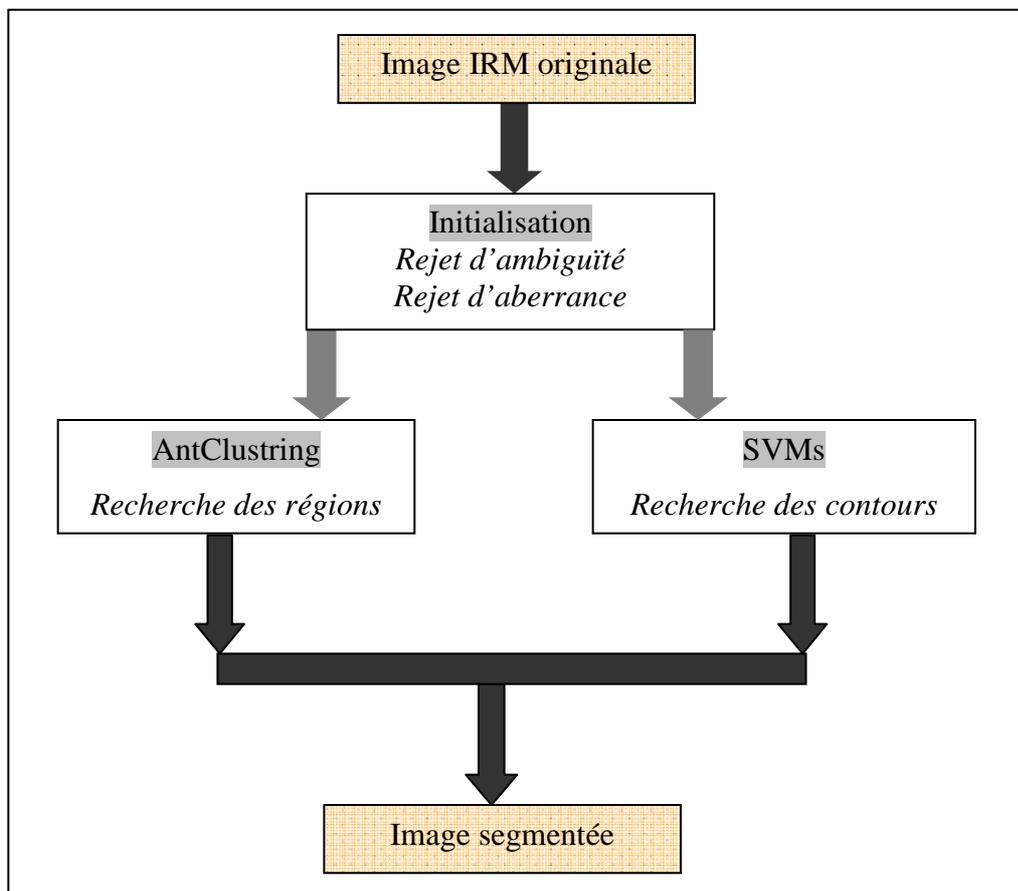


Figure 5.2 : les étapes de l'approche proposée.

La figure précédente illustre les étapes de l'approche que nous avons proposée pour la détection et la localisation des structures dans l'imagerie médicale (IRM).

4. Pourquoi une architecture multi-agents ?

Dans notre approche, les agents sont les entités de base qui permettent de localiser des structures dans les images médicales. Dans chaque algorithme de l'approche, une population d'agents est lancée, ou chaque agents fait segmentée une partie de l'image (c'est à dire classifier un ou plusieurs pixels). L'image segmentée résulte de fusion des résultats des agents. Donc, ces agents permettent d'effectuer la classification des principaux tissus cérébraux.

La figure 5.3 illustre l'interaction entre agents dans l'approche développée. Différentes classes d'agents ont été définies pour répondre à des besoins spécifiques, tels que la modularité, la robustesse...etc.

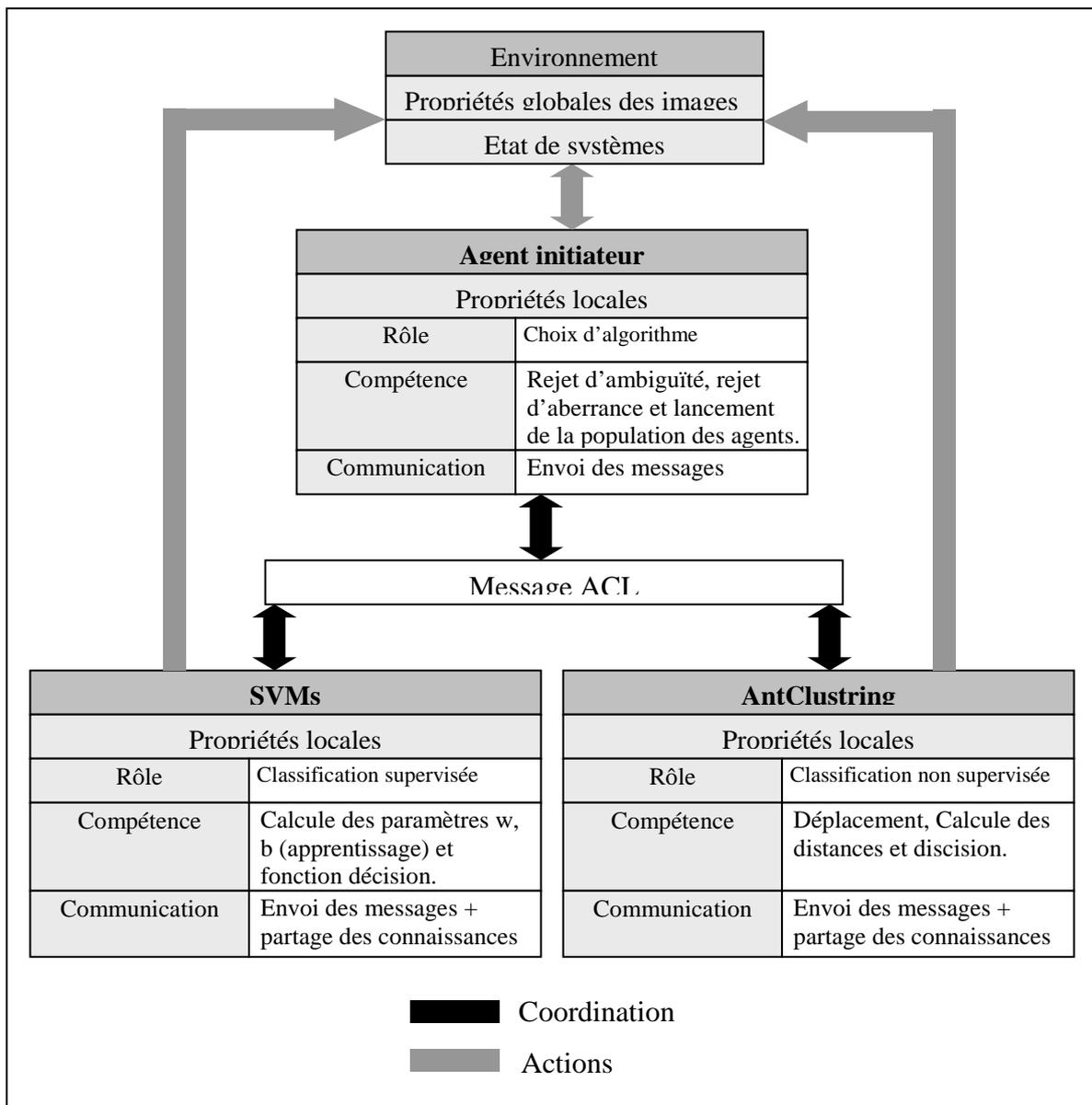


Figure 5.3 : l'interaction entre agents dans l'approche développée.

5. Les Plates Formes multi-agents

Une plate-forme multi-agents est un ensemble d'outils nécessaire à la construction et à la mise en service d'agents au sein d'un environnement spécifique. Ces outils peuvent servir également à l'analyse et au test du SMA ainsi créé. Ces outils peuvent être sous la forme d'environnement de programmation (API) et d'applications permettant d'aider le développeur [FERG].

5.1. FIPA (Foundation for Intelligent Physical Agents)

La FIPA est une organisation à but non lucratif fondée en 1996 dont l'objectif est de produire des standards pour l'interopération d'agents logiciels hétérogènes. Par la combinaison d'actes de langages, de logique des prédicats et d'ontologies publiques, la FIPA cherche à offrir des moyens standardisés permettant d'interpréter les communications entre agents de manière à respecter leur sens initial.

Afin d'atteindre ce but, la FIPA émet des standards couvrant cinq catégories et à chaque catégorie sont associés une ou plusieurs spécifications et un ou plusieurs TCs (Technical Committees) ou WGs (Working Groups) [SKR, SOU].

- *Les applications agents (Agents Applications)* : applications nomades, agent de voyage personnel, gestion de réseaux, assistant personnel...
- *L'architecture abstraite (Abstract Architecture)* : Elle identifie les éléments qui peuvent être décrits d'une manière abstraite tels que le transport des messages, le langage de communication entre agents et les services de localisation.
- *Le transport des messages (Agent Message Transport)* : traite de la représentation (textuelle, binaire ou XML) et de la livraison des messages à travers plusieurs types de réseaux. Un message consiste en une enveloppe et un corps exprimé en FIPA ACL.
- *La gestion des agents (Agent Management)* : Elle établit le modèle selon lequel les agents sont créés, sont enregistrés, sont localisés, communiquent, migrent et sont détruits. Les spécifications des services de pages blanches (*Agent Management System ou AMS*) et de pages jaunes (*Directory Facilitator ou DF*) sont aussi données.
- *La communication entre Agents (Agent Communication)* : définit Les langages d'interaction (ACL), et la séquence des échanges entre les agents permettant ainsi de structurer les protocoles interactions (SL, CCL, KIF ou RDF).

5.2. Critères de choix d'une plate forme MA

Ces critères sont naturellement en relation avec nos attentes d'un système multi-agents et des caractéristiques des agents. Ainsi, la plate forme devra répondre aux contraintes suivantes :

- Les agents doivent pouvoir communiquer entre eux. Il ne s'agira pas pour un agent de répondre mécaniquement à un message mais de pouvoir analyser plusieurs possibilités de réponses avant de réagir.
- La nature de la plate forme est importante : libre ou propriétaire (Nous pourrons accéder aux codes sources).
- La plate forme doit répondre à plusieurs fonctionnalités et offrir une large gamme de bibliothèques, il faut aussi tenir compte de nature de l'interface de développement (vu le temps court alloué au développement des agents).
- La plate forme doit être répandue, elle doit avoir répondu à plusieurs projets de développement de systèmes multi-agents.
- Le langage de programmation sous jacent et la nature des messages échangés est aussi un point important .Un langage plus épandu favorisera une meilleure compréhension des codes sources.

La recherche de la documentation relative aux plates formes que nous avons procédés nous permettre de trancher. En fonction des fonctionnalités offertes par chacune d'entre elles, en définitif **JADE** est la plate forme qui se rapproche plus de nos critères.

5.3. La plate forme JADE

La plate forme JADE (Java Agent DEvelopment Framework) est un environnement de développement d'agents implanté totalement dans le langage Java par le laboratoire TILAB à l'université de Parma. Elle facilite la mise en place d'un système multi agents répondant aux spécifications de FIPA à travers un ensemble d'outils. La plate-forme JADE peut être répartie sur plusieurs serveurs.

Le concept d'agent est vu par JADE comme un processus autonome et indépendant qui a une identité, qui requiert la communication à travers des messages représentés en FIPA-ACL avec les autres agents dans le but de remplir ses missions.

5.3.1. Architecture logiciel de la plate-forme JADE

JADE reprend donc l'architecture de l'*Agent Management Reference Model* proposé par FIPA. Les différents modules présentés dans la figure suivante sont présentés sous forme de services. Les services de base proposés sont le *Directory Facilitator(DF)* et l'*Agent Management System(AMS)*. Il est possible de lui demander de tenir en plus le service de Message Transport *Service (MTS)* pour communiquer entre plusieurs plates-formes.

Un *Agent Identifier (AID)* identifie un agent de manière unique. Le DF offre le service des pages jaunes qui permet de mettre en relation les agents avec leurs compétences. Un agent peut enregistrer ses compétences dans le DF ou interroger le DF pour connaître les compétences proposées par les autres agents.

L'AMS offre le service de pages blanches qui effectue la correspondance entre l'agent et l'AID. il contrôle l'accès et l'utilisation de la plate-forme et maintient un répertoire contenant les adresses de transport des agents de la plate forme. Il n'y a qu'un AMS par plate-forme et chaque agent doit s'enregistrer à un AMS pour avoir un AID.

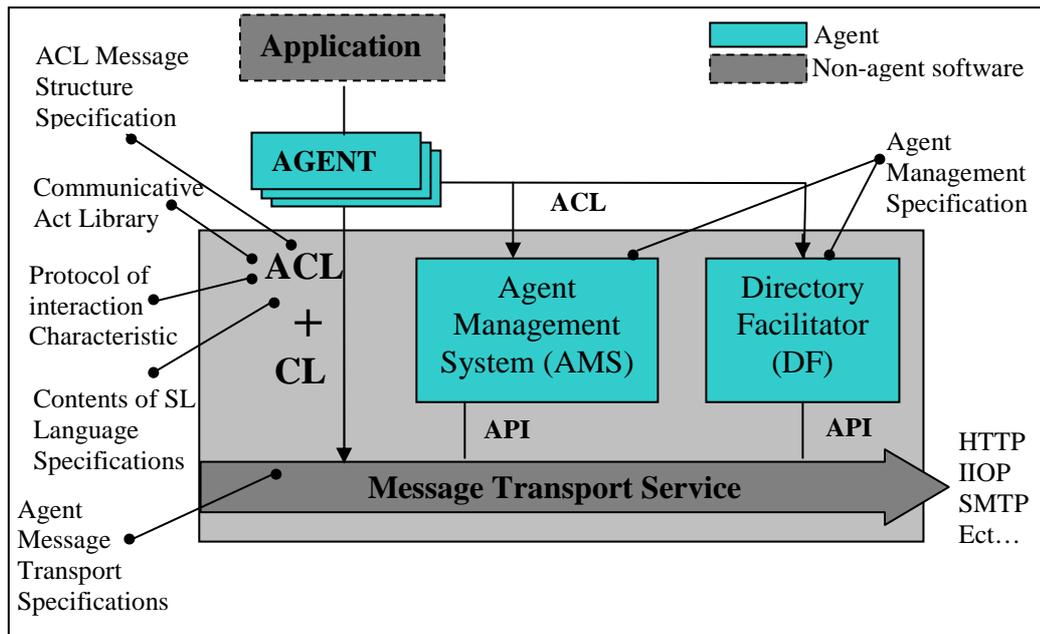


Figure 5.4 : Plateforme d'agent de FIPA [GRE 04].

5.3.2. Langage de communication de la plate-forme JADE

Le langage de communication de la plate-forme JADE est FIPA-ACL (Agent Communication Language). La classe `ACLMessage` représente les messages qui peuvent être échangés par les agents. Lorsqu'un agent souhaite envoyer un message, il doit créer un nouvel objet `ACLMessage`, compléter ces champs avec des valeurs appropriées et enfin appeler la méthode `send()`. Lorsqu'un agent souhaite recevoir un message, il doit employer la méthode `receive()` ou la méthode `blockingReceive()`.

Un message ACL dispose obligatoirement les champs: *Sender* : expéditeur du message ; *Receiver* : destinataire du message ; *reply-to* : participant de la communication ; *content* : contenu du message ; *language* , *encoding* et *ontology* : description du contenu ; *protocol* , *conversation-id* , *reply-with* , *in-reply-to* et *reply-by* : contrôle de la communication. Tous les attributs de la classe `ACLMessage` peuvent être obtenus et modifiés par les méthodes `set/get()`. Le contenu des messages peut être aussi bien du texte que des objets car la sérialisation Java est supportée [FERG].

5.3.3. Comportements des agents dans la plate-forme JADE

Un agent doit être capable de gérer plusieurs tâches de manière concurrente en réponse à différents événements extérieurs. Afin de rendre efficace cette gestion chaque agent de JADE est composé d'un seul thread et plusieurs comportements. Afin d'implémenter un comportement, le développeur doit définir un ou plusieurs objets de la classe Behaviour, les instancier et les ajouter à la file des tâches « ready » de l'agent. Il est à noter qu'il est possible d'ajouter des comportements et sous-comportements à un agent ailleurs que dans la méthode setup() [FERG].

Tout objet de type Behaviour dispose d'une méthode action() (qui constitue le traitement à effectuer par celui-ci) ainsi que d'une méthode done() (qui vérifie si le traitement est terminé). Dans les détails, l'ordonnanceur exécute la méthode action() de chaque objet Behaviour présent dans la file des tâches de l'agent. Une fois cette méthode terminée, la méthode done() est invoquée. Si la tâche a été complétée alors l'objet Behaviour est retiré de la file. Enfin, il existe également quelques méthodes supplémentaires afin de gérer les objets Behaviour :

- reset() qui permet de réinitialiser le comportement;
- onStart() qui définit des opérations à effectuer avant d'exécuter la méthode action();
- onEnd() qui finalise l'exécution de l'objet Behaviour avant qu'il ne soit retiré de la liste des comportements de l'agent;

JADE fournit sous forme de classes un ensemble de comportements et sous-comportements prêt à l'emploi. Elle peut les exécuter selon un schéma prédéfini. On cite :

- Classe SimpleBehaviour (abstraite): modélise un comportement simple. Sa méthode reset() n'effectue aucune opération.
- Classe CompositeBehaviour (abstraite) : modélise un comportement composé. Les actions effectuées par cette classe sont définies dans les comportements enfants.
- Classe FSMBehaviour : Cette classe hérite de CompositeBehaviour et exécute des comportements enfants suivant un automate à états finis défini par l'utilisateur.
- Classe SenderBehaviour : elle étend la classe OneShotBehaviour et encapsule une unité atomique qui effectue une opération d'envoi de message.
- Classe ReceiverBehaviour : Elle encapsule une unité atomique qui effectue une opération de réception de message. Ce comportement s'arrête dès qu'un message a été reçu.
- On peut aussi citer d'autres classes par exemples : CyclicBehaviour (abstraite), ParallelBehaviour, SequentialBehaviour, OneShotBehaviour (abstraite).

5.3.4. Outils de JADE

Pour supporter la tâche difficile des applications multi-agents, des outils ont été développés dans la plate-forme JADE. Chaque outil est empaqueté comme un agent, obéissant aux mêmes règles, aux mêmes possibilités de communication et aux mêmes cycles de vie d'un agent générique [FERG].

- *Agent RMA (Remote Management Agent)* : Le RMA permet de contrôler le cycle de vie de la plate-forme et tous les agents la composant. L'architecture répartie de JADE permet le contrôle à distance d'une autre plate-forme. Plusieurs RMA peuvent être lancés sur la même plate-forme du moment qu'ils ont des noms distincts.
- *Agent Dammy* : L'outil DummyAgent permet aux utilisateurs d'interagir avec les agents JADE d'une façon particulière. L'interface permet la composition et l'envoi de messages ACL et maintient une liste de messages envoyés et reçus et le sauvegarde des messages sur le disque et renvoyé plus tard.
- *Agent DF (Direcory Facilitator)* : L'interface du DF peut être lancée à partir du menu du RMA. Elle peut être juste vue sur l'hôte où la plate-forme est exécutée. Elle permet à l'utilisateur d'interagir avec le DF qui fournit un service de pages jaunes à la plateforme.
- *Agent Sniffer* : Quand un utilisateur décide d'épier un agent ou un groupe d'agents, il utilise un agent sniffer. Chaque message partant ou allant vers ce groupe est capté et affiché sur l'interface du sniffer.
- *Agent Inspector* : Cet agent permet de gérer et de contrôler le cycle de vie d'un agent s'exécutant et la file de ses messages envoyés et reçus.

5.3.5. Evaluation de JADE

Nous allons donner dans ce qui va suivre des résultats publiés (par Mikko Laikanen) qui étaient le fruit d'une série de tests sur la plateforme JADE 1.2. [RAM]

aspects logiciel (documentation, design, ...)		Conformité à la norme FIPA	
Interaction avec l'utilisateur	1/1	conditions accomplies	27/28
Documentations	3.3/4		
facilité d'installation	3/3		
Disponibilité des fichiers binaires	5/5	%	96%
Disponibilité des sources	0/4	options accomplies	6/15
techniques mérites	3.7/4.6		
facilité de mettre en application des agents	4/4		
Développement ultérieur	1/5	%	40%

Tableau 5.3 : évaluation de JADE.

5.4. Les avantages de JADE

- JADE permet de développer facilement des agents selon les cas, les agents peuvent tourner sur une même machine ou sur des machines physiquement distribuées sur un réseau.
- JADE est libre et facile à apprendre car ses composants ont été distingués clairement et ont été entièrement intégrés: protocoles d'interaction, ACL, langues, schémas de codage, protocoles de transport...
- L'architecture de communication offre la transmission de messages flexibles et efficaces selon le modèle de communication FIPA. Elle crée et contrôle une file d'attente des messages entrants pour chaque agent.
- Le mécanisme de transport fonctionne comme un caméléon. Il s'adapte à chaque situation, en choisissant de manière transparente le meilleur protocole disponible.
- La solution du multi-threading offerte directement par Java, Jade supporte également la gestion des comportements coopératifs.
- L'extension JadeX permet le développement des SMA adaptatif et des agents cognitifs.

Conclusion

Dans ce chapitre nous avons décrit l'approche proposée, avec la présentation de l'architecture générale des agents utilisés et les paramètres de chaque algorithme développé.

Le problème majeur de l'algorithme SVM est sa sensibilité aux choix de la base d'apprentissage et au nombre de classe. Pour plus de robustesse et plus de facilité nous avons choisir un nombre de classe limité à deux (point de contour ou font d'écran) et une base d'apprentissage pour chaque composant étudié. Par contre l'algorithme *AntClustering* présente le problème qu'il est sensible aux points aberrantes et au bruit de l'instrumentation lors réquisition des images.

Toutes les interactions et communications se déroulent dans une plateforme *JADE* qui facilite la création et la destruction des agents, et permet la mise en place d'une communication entre ces agents.

Dans le chapitre suivant nous allons présenté les résultats obtenu par nos algorithmes, afin d'évaluer l'efficacité du système développé.

CHAPITRE 6 : VALIDATION DES RESULTATS

Introduction

Dans le but de fournir une évaluation quantitative et qualitative des résultats (validation) de segmentation d'images, nous distinguerons principalement les approches exploitant les résultats de segmentation réalisées manuellement par des experts sur des images réelles et les approches exploitant des images de synthèse (fantômes) pour lesquelles les références sont parfaitement connues.

Dans ce chapitre nous présentons les résultats obtenus par notre système, qui sont évalués sur des images réelles et simulées (mises à disposition sur le site internet brainweb4).

1. Les images de test

Afin de tester les performances de nos algorithmes nous avons utilisé des images réelles et des images de synthèse présentées dans la figure 6.1. L'intérêt des images des images de synthèse est qu'on connues de manière précise le nombre exact de classes ainsi que la qualité de la segmentation de référence. Ces informations ne sont évidemment pas transmises à nos algorithmes et ne sont utilisées que pour tester la capacité de ces algorithmes.

L'application de nos algorithmes pour la segmentation des images réelles conduit à une évaluation des résultats obtenus par rapport aux résultats réalisés manuellement par des algorithmes de références (l'algorithme *AntClust* proposé dans [OUA 06] et l'algorithme de détection des contours de Sobel [SOB 78] avec un seuil égale à 100) sur ces images en utilisant des mesures de validation.

1.1. Les images réelles

Les images réelles sur lesquelles nous avons travaillé ont été acquises dans le cadre de la collaboration entre le laboratoire LSI (Laboratoire Systèmes Intelligents : équipe image et signaux) de l'université Ferhat Abbas de Sétif et le Centre de Recherche en Sciences et Technologies de l'Information et de la Communication (CReSTIC) équipe LAM (Traitement d'images) IUT de Troyes. Il s'agit d'images pondérées en T_1 , T_2 et en densité de proton pour des patients de différents âges (taille pixel =1mm, taille de matrice 256X 256).

1.2. Les images de synthèse

Afin d'évaluer quantitativement le système développé, nous avons utilisé des images de synthèse générées à l'aide du simulateur proposé par BrainWeb [2]. Il s'agit d'images de synthèse dont il est possible de contrôler les paramètres d'épaisseur de coupe, de bruit et d'hétérogénéité, etc.

Les images sont créées à partir de 27 acquisitions réalisées sur un même individu et caractérisées par un rapport signal/bruit élevé. Le volume initial utilisé pour la construction d'un fantôme de cerveau est constitué par la moyenne de ces acquisitions.

Les principales étapes mises en oeuvre afin de transformer le volume initial en un fantôme permettant de générer des simulations sont les suivantes :

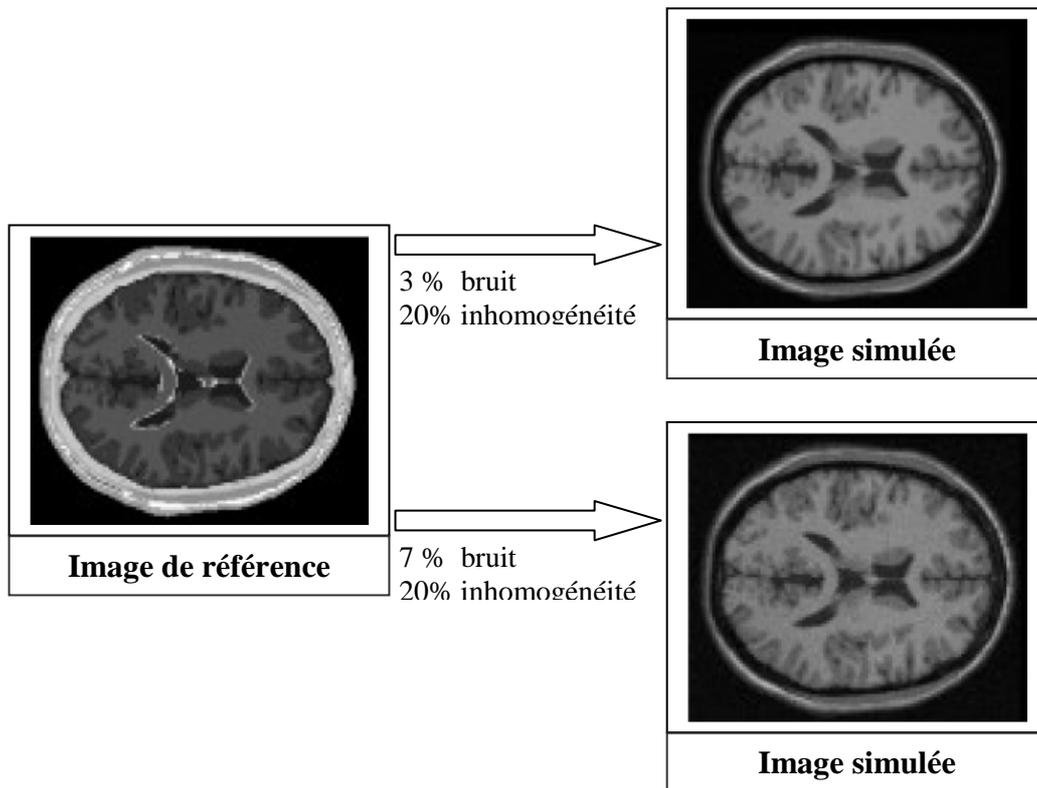


Figure 6.1 : Processus de construction des images simulées.

- *La correction des non-uniformités* : afin de réduire au maximum les risques d'erreurs de classification, un algorithme de correction des non-uniformités des niveaux de gris a été appliqué avant tout traitement du volume initial.
- *La classification* : la classification a été réalisée à partir d'un ensemble d'apprentissage décrivant des exemples de pixels appartenant aux différents tissus recherchés et marqués par un expert.
- *Amélioration des résultats*: un ensemble de corrections manuelles ou semi-automatiques a été réalisé afin d'améliorer les résultats de la segmentation.
- *Les simulations* : les niveaux de gris simulés pour chaque pixel du volume ont été obtenus par résolution des équations de Bloch [GER 99].

Nous avons choisi des volumes avec une épaisseur de coupe de 1 mm, taille couramment utilisée pour générer des images anatomiques. Le choix du bruit proposé sur le site BrainWeb est compris entre 0% et 9% et celui des valeurs du paramétrage d'hétérogénéité entre 0% et 40%. Pour des raisons de réalisme, la valeur de 0% n'a pas été retenue. De même, la valeur de 9% produit des images à l'allure très artificielle et nous l'avons écartée. Nous avons utilisé dans nos simulations la valeur de 3%. Les valeurs du paramètre d'inhomogénéité ont été choisies entre 0% et 20%. La valeur de 40% a été écartée car elle perturbe trop les distributions des

niveaux de gris dans les images et que dans la réalité l'inhomogénéité se situe plutôt aux environs de 10%.

2. Evaluation et étude comparative

Dans la segmentation d'image, les performances de nos algorithmes ont rapport avec la qualité de la segmentation obtenue. Pour un type de données, une *bonne* segmentation est définie par l'appréciation de l'utilisateur (qualité visuelle) et le but recherché. C'est pour cette raison que nous avons mesuré et quantifié les performances de notre segmentation de l'ensemble de l'encéphale.

2.1. Le processus d'évaluation sur image

La comparaison quantitative de la segmentation obtenue par rapport à un référentiel est facilitée par l'utilisation des images de synthèses ce qui permet d'étudier la robustesse des systèmes de segmentation pour différents niveaux de bruit et d'hétérogénéité d'intensité des images simulées. Pour un même système de segmentation, il est possible de tester quantitativement l'influence du choix de certains paramètres sur les obtenues.

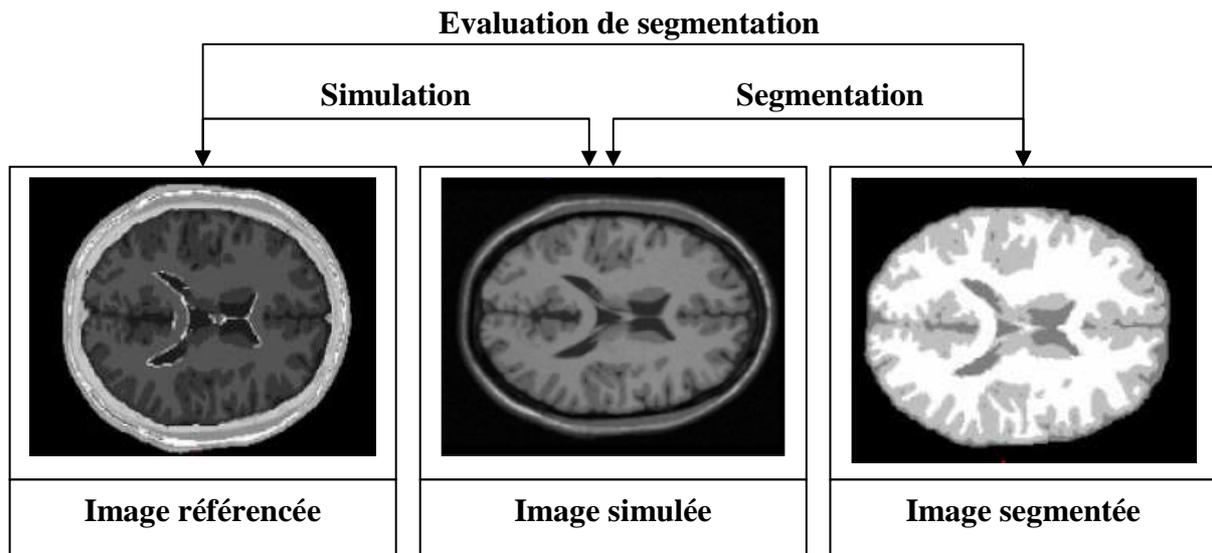


Figure 6.2 : Processus d'évaluation sur des images fantômes.

2.2. Le paramètres d'évaluation

2.2.1. Taux de recouvrement

Le taux de recouvrement $R(\%)$ entre l'ensemble A et B est défini par : $R(\%) = \frac{A \text{ et } B}{A \text{ ou } B}$

où "**et**" et "**ou**" signifient les opérations "*et*" et "*ou*" logique.

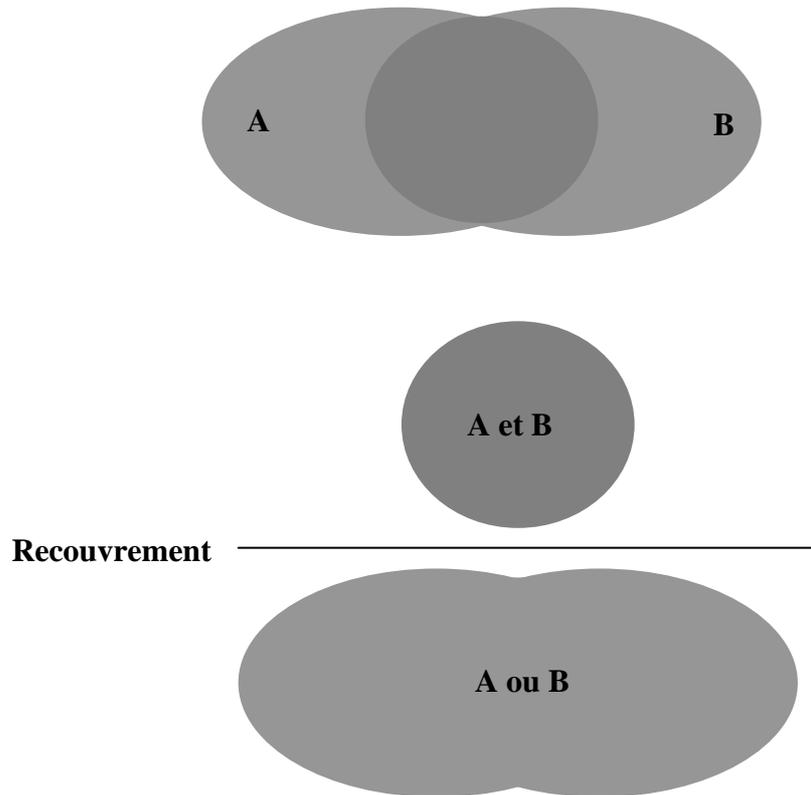


Figure 6.3 : Recouvrement entre les deux régions A et B

Pour tester la validité de nos méthodes et pour mesurer quantitativement la qualité de notre segmentation, une approche classique présentée par Vinet [VIN 91] consiste à calculer le pourcentage des pixels mal appariés entre deux segmentations. Cette méthode présente l'inconvénient qu'elle ne fait pas intervenir d'information spatiale sur la localisation de ces pixels.

2.2.2. La distance de Hausdorff

Soit X l'ensemble des pixels de l'image, N le nombre total de pixels, A et B deux régions de X . La distance de Hausdorff entre A et B est définie par Haacke et Tkach [HAA 90]:

$$H(A,B) = \max\{\sup\{d(x,B)/x \in A\}; \sup\{d(x,A)/x \in B\}\}$$

Avec $d(x,A) = \inf\{p(x,a)/a \in A\}$ ou $p(x,a)$ est une distance définie entre deux points de X . $H(A,B)$ est la distance maximale théoriquement car elle a une relation directe avec les opérations de base de la morphologie mathématique. En remarquant que la distance de Hausdorff entre deux régions peut s'écrire sous la forme :

$$H(A,B) = \sup\{|d(x,A) - d(x,B)| / x \in X\}$$

où A , et B sont deux régions de X , Finalement, si deux segmentations sont identiques, la distance est nulle; dans le cas contraire, elle tend vers 1. Leur inconvénient est que H est sensible au bruit à cause de l'utilisation de \sup dans sa définition.

2.2.3. Les trois coefficients de validation des systèmes d'imagerie

Nous avons préféré utiliser les coefficients décrits dans [SHU 99], définis pour valider des systèmes d'imagerie. Les trois coefficients proposés sont le Building Detection Percentage (BDP), le Branching Factor (BF) et le Quality Percentage(QP) ou le taux de recouvrement avec :

- *Le Building Detection Percentage* : est en fait équivalent à la mesure de sensibilité et quantifie la fraction de pixels qui ont été correctement classifiés par le système. Dans le cas idéal, ce coefficient vaut 1.
- *Le Branching Factor* : quantifie la sur détection de pixels n'appartenant pas au tissu recherché dans la carte de référence. Dans le cas idéal ce coefficient vaut 0.
- *Le Quality Percentage* : représente la qualité globale absolue du système de segmentation. Dans le cas idéal ce coefficient vaut 1.

Ces coefficients définis respectivement par les équations suivantes :

$$BDP = \frac{VP}{VP + FN} , BF = \frac{FP}{VP} , QP = \frac{VP}{VP + FP + FN}$$

où le nombre de Vrais Positifs (**VP**) désigne le nombre de pixels détectés à raison, le nombre de Faux Positifs (**FP**) désigne le nombre de pixels détectés à tort, le nombre de Vrais Négatifs (**VN**) désigne le nombre de pixels non détectés à raison et le nombre de Faux Négatifs (**FN**) désigne le nombre de pixels non détectés à tort, sont généralement calculés pour chaque classe de tissus .

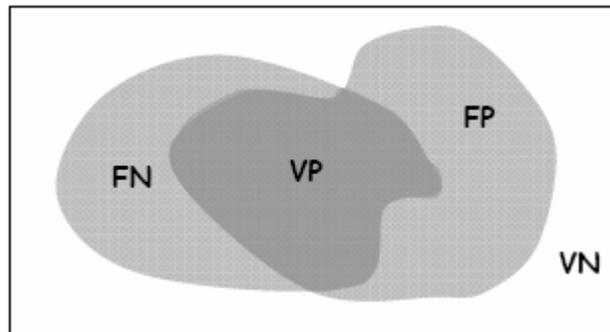


Figure 6.4 : Définition des valeurs de VP, VN, FP et FN par comparaison de l'image de référence (ensemble A) et de l'image segmentée (ensemble B).

2.2.4. Autres mesures d'évaluation

Soit une image segmentée *Seg* et une image de référence *Ref*. l'index de Rand *R* qui détermine le taux de pixels (de l'image *Seg*) bien classés (par rapport à l'image *Ref*) est défini

comme suit : $R = \frac{a + d}{a + b + c + d}$ avec *a*, *b*, *c* et *d* des paramètres calculés pour tous les couples

de pixels et de la façon suivante: si $c_{ref}(i)$, $c_{ref}(j)$, $c_{seg}(i)$ et $c_{seg}(j)$ sont les classes de p_i et p_j dans l'image de référence et l'image segmentée, on a :

$$a = \left\{ \left\{ i, j / c_{ref}(i) = c_{ref}(j) \wedge c_{seg}(i) = c_{seg}(j) \right\} \right\} \quad b = \left\{ \left\{ i, j / c_{ref}(i) = c_{ref}(j) \wedge c_{seg}(i) \neq c_{seg}(j) \right\} \right\}$$

$$c = \left\{ \left\{ i, j / c_{ref}(i) \neq c_{ref}(j) \wedge c_{seg}(i) = c_{seg}(j) \right\} \right\} \quad d = \left\{ \left\{ i, j / c_{ref}(i) \neq c_{ref}(j) \wedge c_{seg}(i) \neq c_{seg}(j) \right\} \right\}$$

R prend ses valeurs dans l'intervalle $[0,1]$. Il atteint son maximum quand tous les couples de pixels sont classés ensemble dans l'image de référence et l'image segmentée. [OUA 06]

- La variance intra-classe calculée par l'équation suivante : $V = \sum_{k=1}^K \sum_{i \in c_k} (ng_i - ng_k)^2$

Où K est le nombre de classes trouvé par l'algorithme de classification et ng_k est le centre de gravité de la classe c_k . Une bonne classification permet de minimiser la variance intra-classe.

2.3. Résultats

Cette section présente les résultats des évaluations conduites par les deux algorithmes de classification sur 8 images de jeu de test utilisé. Pour chacune d'elles, ces testes permet d'avoir une estimation du comportement moyen de chacun des algorithmes. Pour chacune des mesures d'évaluation, nous avons calculé les valeurs moyennes.

Les SVMs		AntClustering	
Nombre de SVMs	65536	Nombre de fourmis	A=20
Nombre de classes	02	contrôle la probabilité de déplacer un pixel d'une case	$k_d=0.3$
Nombre d'exemples d'apprentissage	21	contrôle la probabilité de déposer un pixel dans une case	$k_p=0.1$
Taux de passage moyen des bases d'apprentissages	97.46%	Contrôle la probabilité de choisir un pixel dans une case contenant deux pixels	Q=0.7

Tableau 6.1 : Paramètres de SVMs et de *AntClustering*.

Ce tableau résume les valeurs des paramètres de l'algorithme *AntClustering* et des SVMs avec lesquels nous avons obtenu les résultats montrés dans le tableau 6.2 et le tableau 6.3 pour la segmentation des images synthétiques en utilisant les coefficients de validation des systèmes d'imagerie décrits dans [SHU 99], l'index de Rand et la mesure de l'intra-classe comparés avec ceux des approches classiques. Les résultats montrent clairement que l'algorithme *AntClustering* bien qu'il ne nécessite pas une connaissance du nombre probable de classe, arrive à identifier un nombre égal (ou très proche) du nombre correct de classes. D'un

autre coté, la qualité de la classification générée par l’algorithme SVMs est très proche de la qualité optimale fournit par la base d’apprentissage.

Algos Coupes	Les SVMs					AntClustering						
	Fond		contour		QP (%)	MG		MB		LCR		QP (%)
	BF	BDP	BF	BDP		BF	BDP	BF	BDP	BF	BDP	
Coupe 01	0,0002	0,999	0,002	0,998	99,67	0,03	0,91	0,08	0,92	0,05	0,93	94,45
Coupe 02	0,0004	0,999	0,002	0,995	99,36	0,06	0,96	0,09	0,91	0,06	0,93	94,35
Coupe 03	0,0001	0,999	0,001	0,999	99,85	0,04	0,95	0,10	0,92	0,06	0,94	95,12
Coupe 04	0,0001	0,999	0,001	0,999	99,77	0,09	0,97	0,09	0,93	0,07	0,93	94,89
Coupe 05	0,0002	0,999	0,003	0,998	99,53	0,07	0,93	0,11	0,92	0,05	0,93	95,07
Coupe 06	0,0003	0,999	0,004	0,998	99,4	0,06	0,95	0,08	0,91	0,07	0,92	94,26
Coupe 07	0,0001	0,999	0,003	0,999	99,61	0,08	0,94	0,07	0,93	0,10	0,93	93,98
Coupe 08	0,0001	0,999	0,002	0,999	99,74	0,09	0,95	0,11	0,89	0,08	0,94	93,87
Moyenne	0,0002	0,999	0,002	0,998	99,62	0,07	0,95	0,09	0,92	0,07	0,93	94,50

Tableau 6.2: Evaluations de la segmentation sur 08 coupes successives du volume.

En comparant les résultats des deux systèmes, nous constatons que :

- La similarité des résultats obtenus pour les valeurs du *BDP* de MG et du *BDP* de MB.
- Pour toutes les images, la différence entre les valeurs moyennes de ces coefficients est inférieure à 0.09, ce qui justifié notre choix d’écarter les pixels ambigus dans le processus de classification (entre MG, MB)

Dans le cas de notre application, le coefficient *BF* est faible à cause du rejet des pixels aberrants par rapport aux centres des classes, ce facteur est un coefficient qui permet d’évaluer les surdétections de pixels.

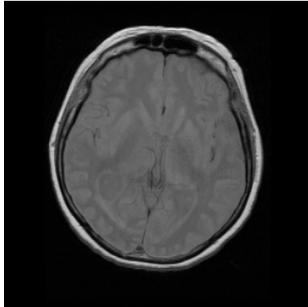
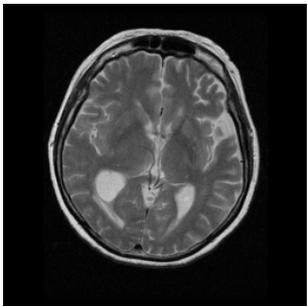
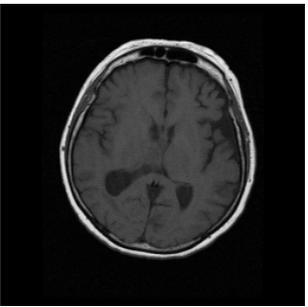
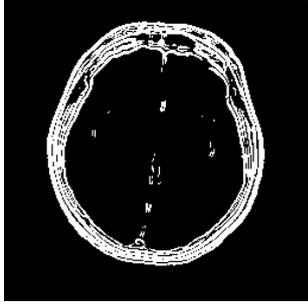
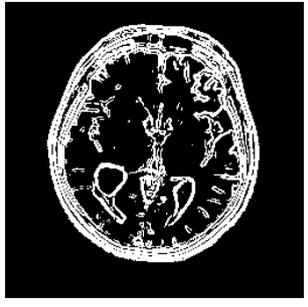
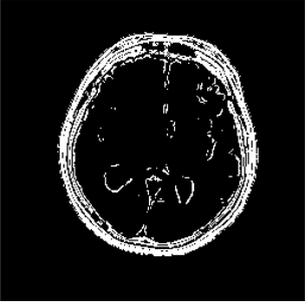
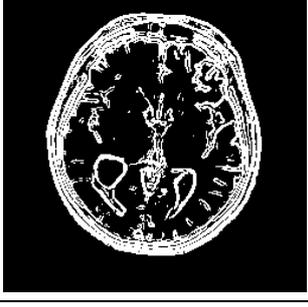
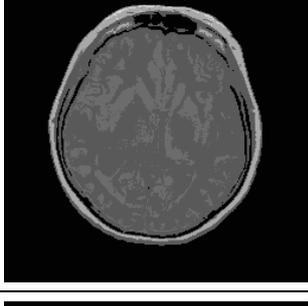
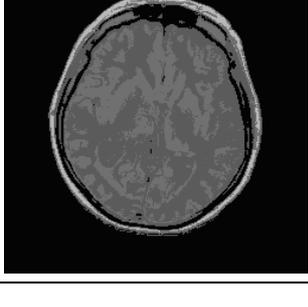
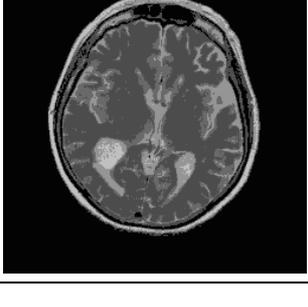
Algorithmes Mesures	Les SVMs	Filtre de Sobel (seuil = 100)	AntClustering	AntClust
Le taux de recouvrement	Contour = 0.95 Fond = 0.98	Contour = 0.96 Fond = 0.98	LCR = 0.93 MB = 0.74 MG = 0.75	LCR = 0.94 MB = 0.73 MG = 0.75
L’index de rand R	0.98	0.98	0.96	0.96
La variance intra-classe	10^8	10^8	10^6	10^6
L’écart type intra-classe	10^4	10^4	10^3	10^3

Tableau 6.3 : Comparaison des mesures obtenues par algorithmes.

3. Résultats

3.1. Images de synthèses

Le Tableau 6.4 représente les résultats de l'application de nos algorithmes sur des images de synthèse :

		Image en T_1	Image en T_2	Image en DP
images saines	Images originales			
	Images segmentées par SVMs			
	Images segmentées par filtre de Sobel			
	Images segmentées par AntClustering			
	Images segmentées par AntClust			

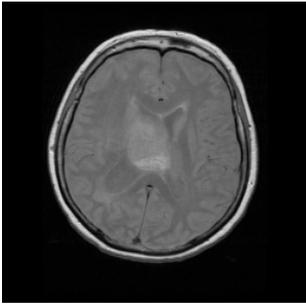
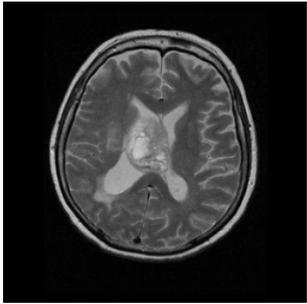
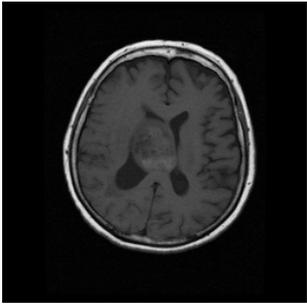
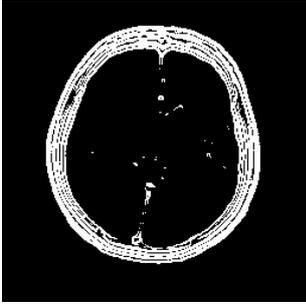
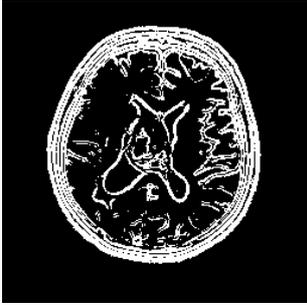
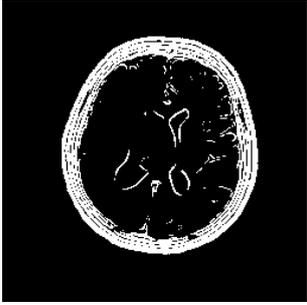
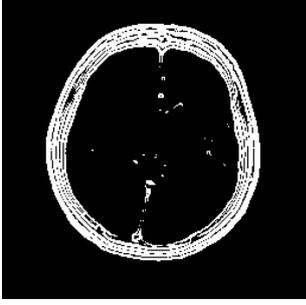
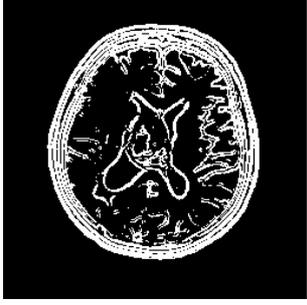
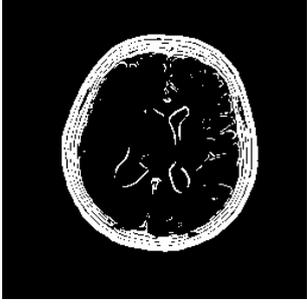
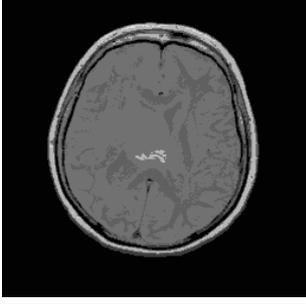
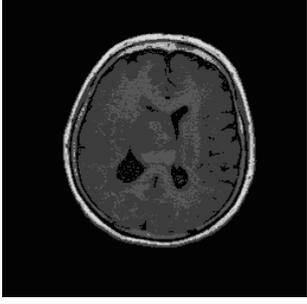
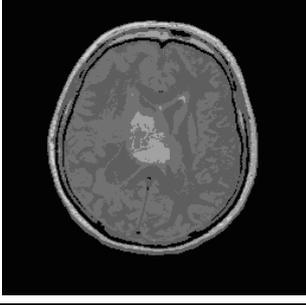
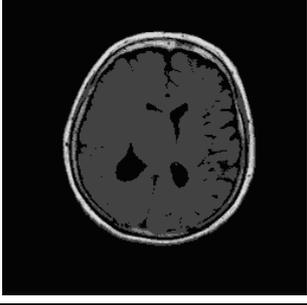
images pathologiques	Images originales			
	Images segmentées par SVMs			
	Images segmentées par filtre de Sobel			
	Images segmentées par AntClustering			
	Images segmentées par AntClust			

Tableau 6.4 : Comparaison des images segmentées par différentes approches de classification

En comparant les résultats obtenues par nos algorithmes avec les résultats obtenues par des algorithmes de références sur les mêmes images (images de Tableau 6.4), nous avons établi le bilan suivant :

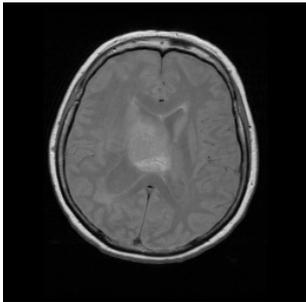
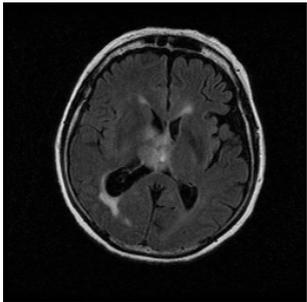
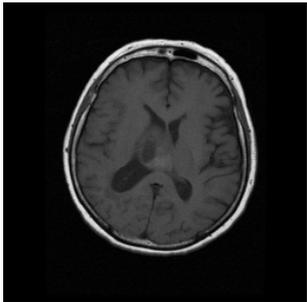
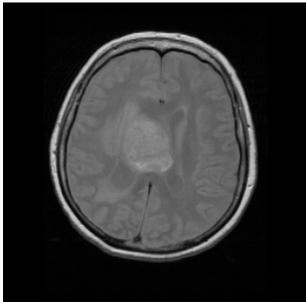
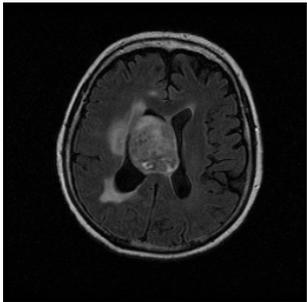
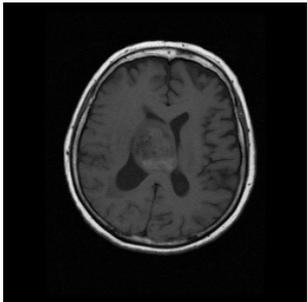
- *Pour les SVMs* : les contours sont très semblables à celles de l'image segmentée par l'algorithme de référence (filtre de Sobel [SOB 78] avec un seuil égale à 100). Le taux moyen de pixels biens classées par notre algorithme par rapport à celles résultantes de l'algorithme de référence est de 97.04 %.
- *Pour les AntClustering* : les régions sont semblables à celles de l'image segmentée par l'algorithme de référence (*AntClust* [OUA 06]). Le taux moyen de pixels biens classées par notre algorithme par rapport à celles résultantes de l'application de l'algorithme de référence sur la même image est de 94.31 %.

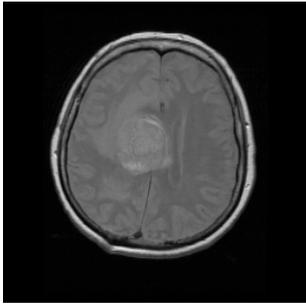
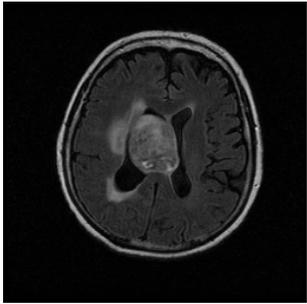
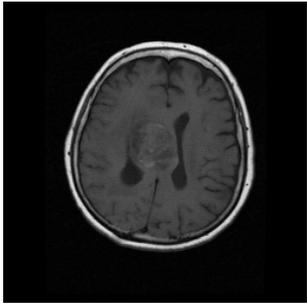
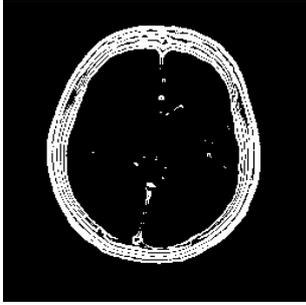
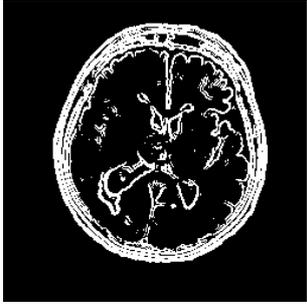
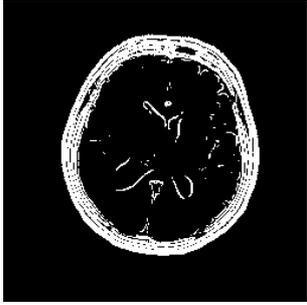
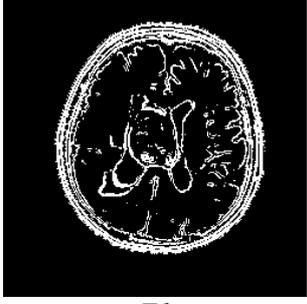
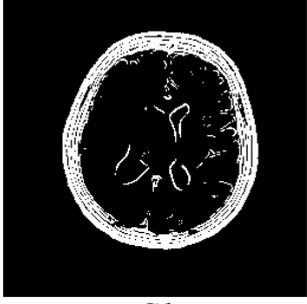
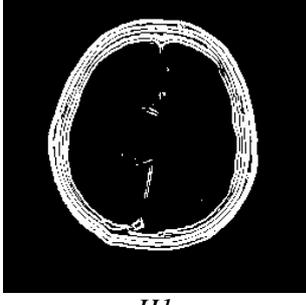
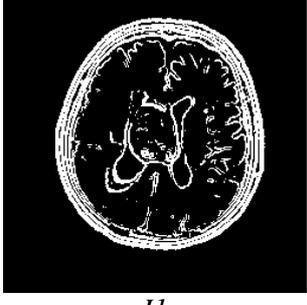
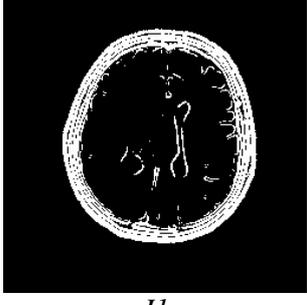
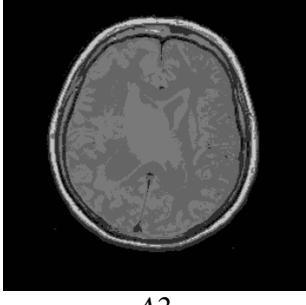
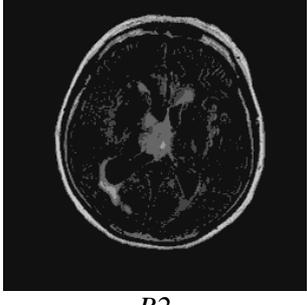
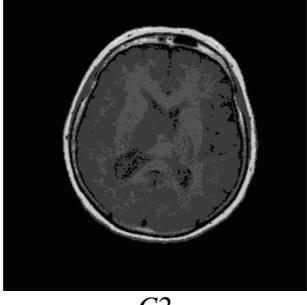
3.2. Images réelles

Les résultats illustrés sur le tableau 6.5 du à la segmentation des images réelles du cerveau appliquée avec succès. Ces images ont les propriétés suivantes :

- **Nature** : IRM Cérébrales;
- **Format** : BMP;
- **Taille** : 256*256;
- **Coupes** : axiales, sagittales, coronales;
- **Pondérations** : T₁, T₂, densité de protons;
- **Patients** : enfants, hommes, femmes;
- **Date d'acquisition** : .../.....

Ces images sont présentées sur différentes coupes, différentes pondérations et par différentes algorithmes :

		Image en T ₁	Image en T ₂	Image en DP
Coupes axiales	Image originale	 A	 B	 C
		 E	 F	 G

			
	<i>H</i>	<i>I</i>	<i>J</i>
Image segmentée par SVMs			
	<i>A1</i>	<i>B1</i>	<i>C1</i>
			
	<i>E1</i>	<i>F1</i>	<i>G1</i>
			
	<i>H1</i>	<i>I1</i>	<i>J1</i>
Image segmentée par AntClustering			
	<i>A2</i>	<i>B2</i>	<i>C2</i>

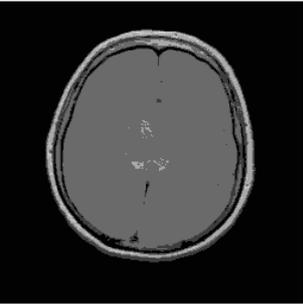
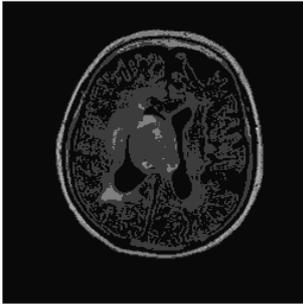
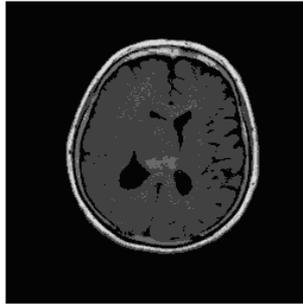
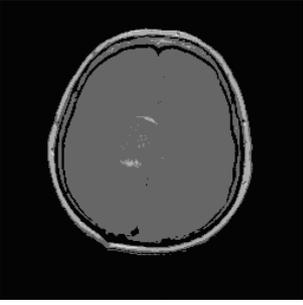
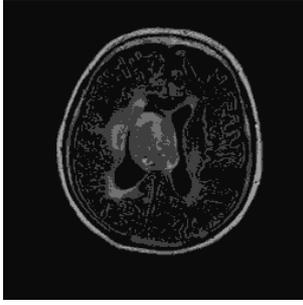
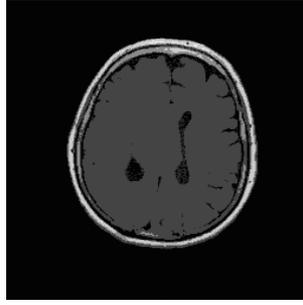
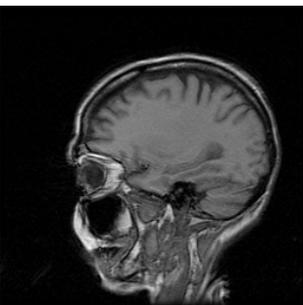
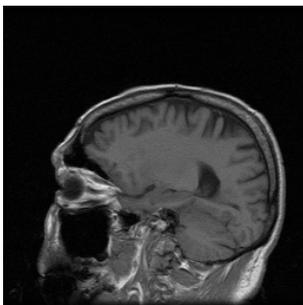
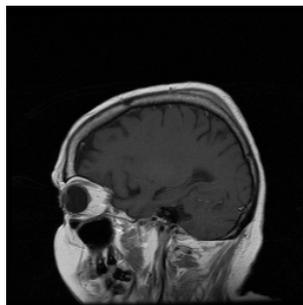
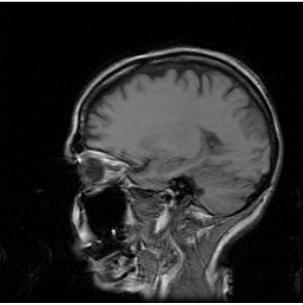
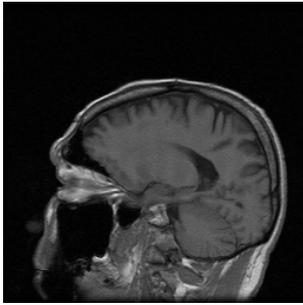
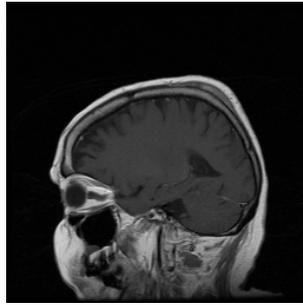
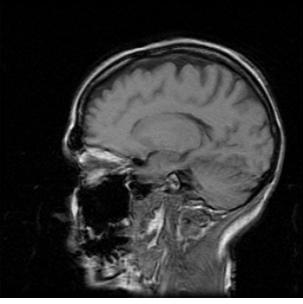
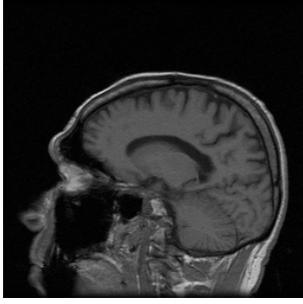
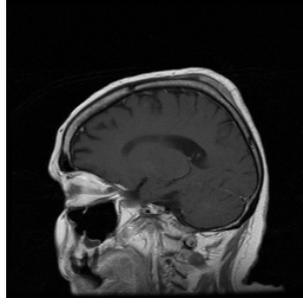
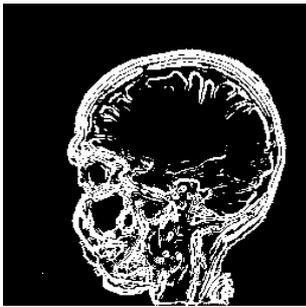
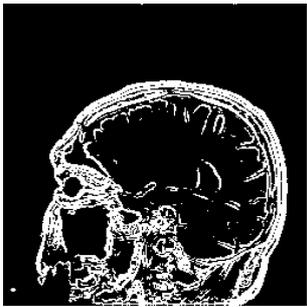
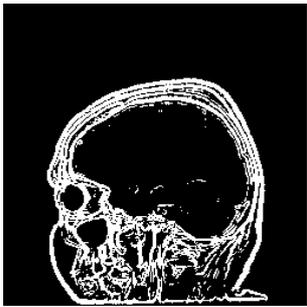
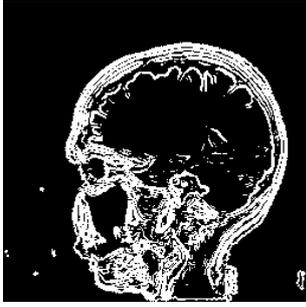
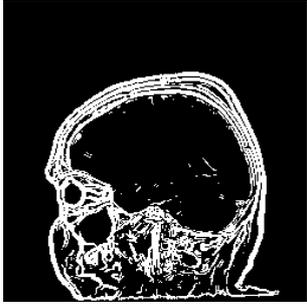
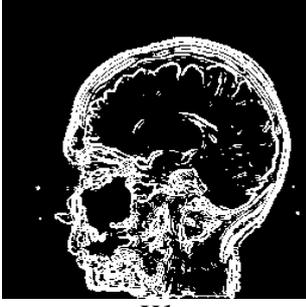
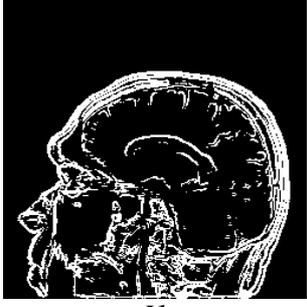
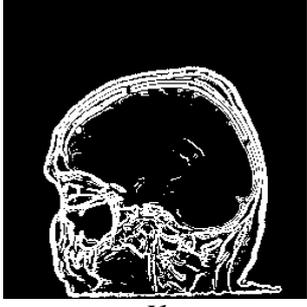
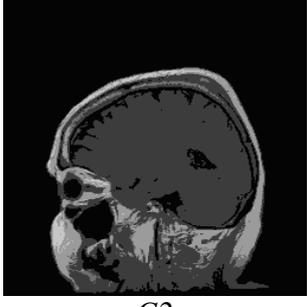
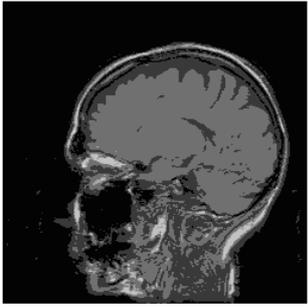
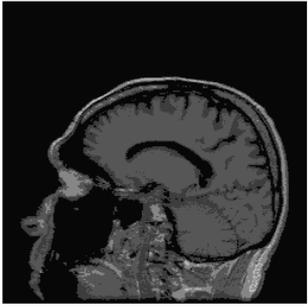
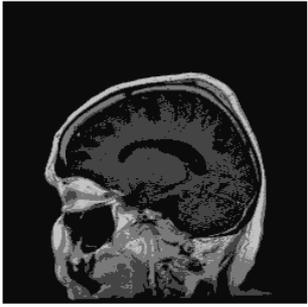
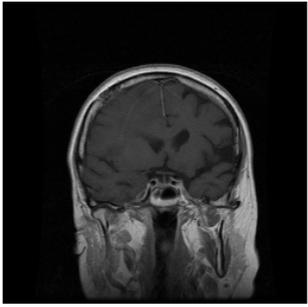
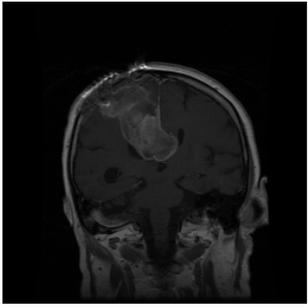
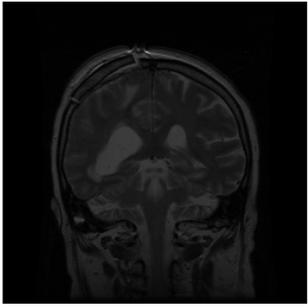
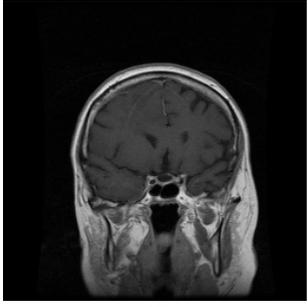
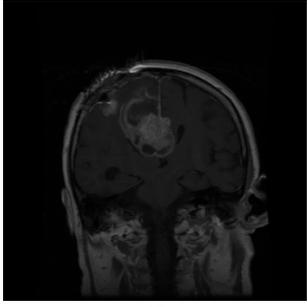
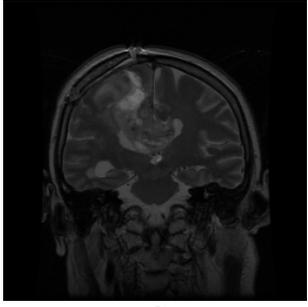
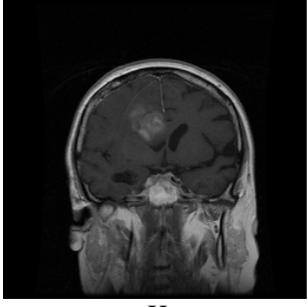
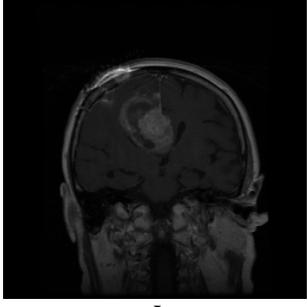
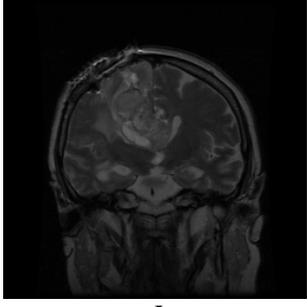
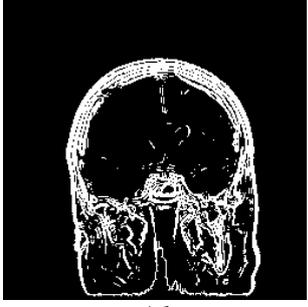
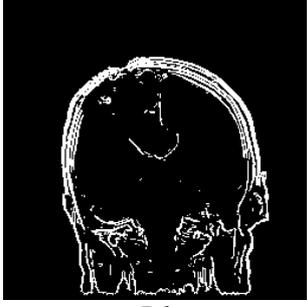
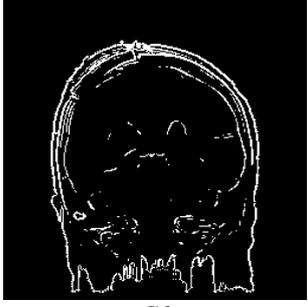
		 <p><i>E2</i></p>	 <p><i>F2</i></p>	 <p><i>G2</i></p>
		 <p><i>H2</i></p>	 <p><i>I2</i></p>	 <p><i>J2</i></p>
<p>Coupes sagittales</p>	<p>Image originale</p>	 <p><i>A</i></p>	 <p><i>B</i></p>	 <p><i>C</i></p>
		 <p><i>E</i></p>	 <p><i>F</i></p>	 <p><i>G</i></p>
		 <p><i>H</i></p>	 <p><i>I</i></p>	 <p><i>J</i></p>

Image segmentée par SVMs			
	<i>A1</i>	<i>B1</i>	<i>C1</i>
			
	<i>E1</i>	<i>F1</i>	<i>G1</i>
			
	<i>H1</i>	<i>I1</i>	<i>J1</i>
Image segmentée par AntiClustering			
	<i>A2</i>	<i>B2</i>	<i>C2</i>
			
	<i>E2</i>	<i>F2</i>	<i>G2</i>

		 <i>H2</i>	 <i>I2</i>	 <i>J2</i>
Coupes coronales	Image originale	 <i>A</i>	 <i>B</i>	 <i>C</i>
		 <i>E</i>	 <i>F</i>	 <i>G</i>
		 <i>H</i>	 <i>I</i>	 <i>J</i>
	Image segmentée par SVMs	 <i>A1</i>	 <i>B1</i>	 <i>C1</i>

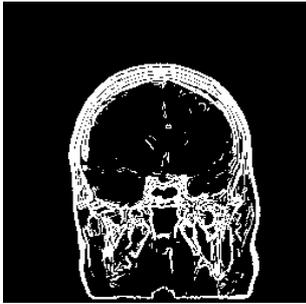
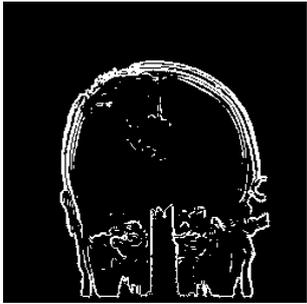
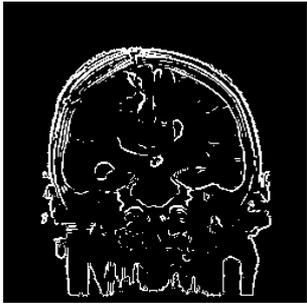
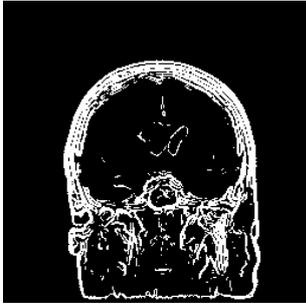
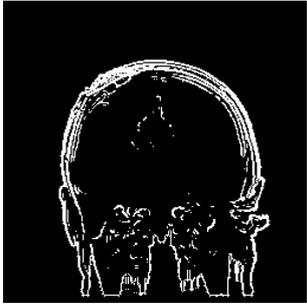
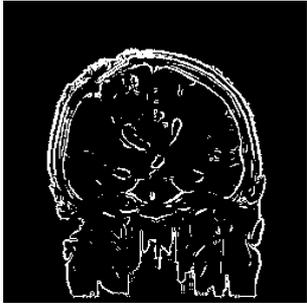
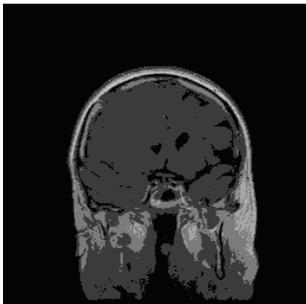
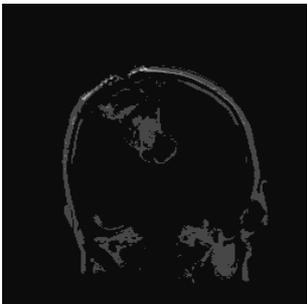
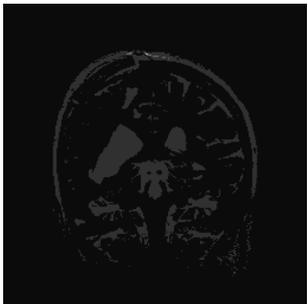
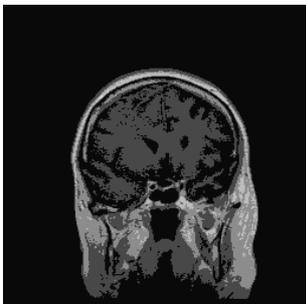
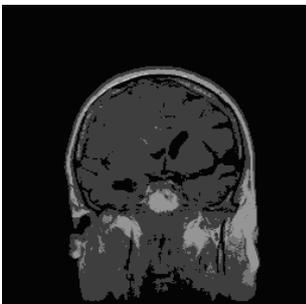
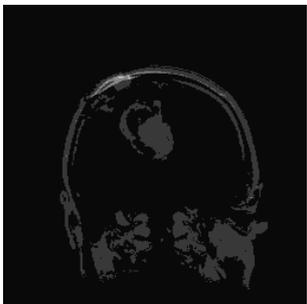
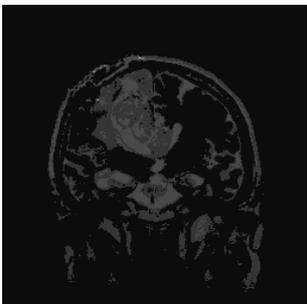
			
	<i>E1</i>	<i>F1</i>	<i>G1</i>
			
	<i>H1</i>	<i>I1</i>	<i>J1</i>
Image segmentée par <i>AntClustering</i>			
	<i>A2</i>	<i>B2</i>	<i>C2</i>
			
	<i>E2</i>	<i>F2</i>	<i>G2</i>
			
	<i>H2</i>	<i>I2</i>	<i>J2</i>

Tableau 6.5 : résultats de segmentation des images réelles sur différentes coupes, différentes pondérations et par différents algorithmes.

Le tableau 6.6 présente les temps d'exécutions enregistrés pour chaque segmentation :

images	axiales			sagittales			coronales		
	SVMs		Ant- Clustering	SVMs		Ant- Clustering	SVMs		Ant- Clustering
	Ta	Td		Ta	Td		Ta	Td	
1	340,94	96,24	1.37	360,59	98,16	1.17	362,27	99,10	1.24
2	351,02	98,32	1.25	351,86	102,70	1.22	466,50	100,51	1.51
3	458,63	95,56	1.58	352,13	97,33	1.35	478,13	94,71	1.89
4	392,06	94,12	1.23	455,00	97,54	1.48	447,31	95,83	1.34
5	486,11	96,18	1.18	373,97	96,32	1.27	494,83	98,85	1.55
6	381,58	93,78	1.32	363,41	98,63	1.38	403,91	96,21	1.37
7	395,78	99,07	1.21	363,83	91,24	1.44	511,16	95,16	1.78
8	433,66	90,78	1.43	361,13	101,55	1.57	342,49	96,17	1.29
9	362,75	93,17	1.30	360,25	93,44	1.48	388,52	99,14	1.68
Moyenne	400,28	95,25	1.32	371,35	97,43	1.37	432,79	97,30	1.52

Tableau 6.6 : Temps de segmentation en secondes.

Pendant tout le processus de segmentation des images réelles, nous avons remarqué que le temps nécessaire pour segmenter une image par les SVMs (temps d'apprentissage + temps de décision) est très élevé par rapport au référentiel, par contre le temps nécessaire pour segmenter une image par les *AntClustering* (temps d'émergence) est très faible par rapport au temps de segmentation de la même image par l'algorithme de référence AntClust.

Remarque : Les algorithmes développés tournent sur une machine Intel (R) Pentium (R) 4 CPU 2.26GHz-2.28GHz, 224 Mo de RAM.

Conclusion

En conclusion, on peut dire que nos algorithmes de détection et de localisation de structures sont robustes. Une expertise des résultats a montré que ces derniers étaient très proches par rapport au référentiel. Nous pourrions conclure, en comparant la segmentation par notre approche avec celles fournies par les approches classiques avec un coefficient global de qualité avoisinant les 95%, donc la qualité de la segmentation est en faveur de l'architecture proposée.

Malgré tous les avantages des approches développées, ces dernières souffrent cependant du problème de temps d'exécution qui reste élevé cela est principalement lié à la taille des images, la nature des calculs effectués sur ces dernières et le temps de lancement des agents.

Conclusion générale

Dans l'objectif que nous avons fixé au début de cette thèse, qui était de proposer une architecture multi-agents adaptative pour la détection et la localisation des structures dans des images médicales de type IRM, L'étude bibliographique que nous avons procédé au deuxième chapitre a montré la diversité des approches qui existent pour aborder la segmentation des images IRM cérébrales. Cette diversité nous a guidé à développer une architecture multi-agents adaptative qui combine entre un algorithme supervisé (SVMs) et un algorithme non supervisé (*AntClustering*) pour profiter des avantages de chacun, afin que la segmentation caractérise mieux les différents structures cérébraux (matière grise, matière blanche, liquide céphalorachidien) et permet aussi de faire une comparaison ces deux types de méthodes.

L'ensemble du travail que nous avons réalisé au cours de ce mémoire est fondé sur l'utilisation des machines à vecteurs de supports pour la détection des contours (puisque les SVMs sont bien adaptés à des application ou la décision est binaire) et l'algorithme de classification non supervisé *AntClustering* avec les améliorations que nous avons introduit pour la détection des régions homogènes par le biais de générer une population de fourmis qui déplacent les pixels de l'image de manière que les fourmis réalisent des groupes de pixels. Ces derniers représentent des zones homogènes de l'image. Pour rendre nos algorithmes robustes face aux données ambiguës et aberrantes nous avons introduit la notion de rejet avec deux types de rejet : le rejet d'ambiguïté qui concerne les individus situés entre les frontières des différentes classes et le rejet d'aberrance concernant les individus dont les similarités par rapport à toutes les classes sont très éloignées.

La mise en place de notre approche nous a conduit à l'utilisation des systèmes multi-agents qui présentent la faculté d'apprentissage de nouvelles compétences pour gérer l'incertitude liée aux informations manipulées (images IRM). Notre processus de segmentation fait donc intervenir, à différents niveaux. Un agent initiateur (superviseur) est chargé de rejet d'ambiguïté et d'aberrance avant de lancer une population d'agents pour chaque algorithme et gérer toutes les interactions entre ces agents. Toutes les interactions et communications se déroulent dans une plateforme Jade qui permet de masquer les aspects techniques de communication entre agents, et facilite la création et la destruction des agents.

Les premiers résultats de la segmentation qui a été réalisée sur des IRM cérébrales en utilisant les algorithmes développés sont présentés dans le sixième chapitre. Ces résultats obtenus après segmentation sur un ensemble de données sont satisfaisants et comparés de façon favorable avec les résultats des autres méthodes comme l'algorithme *AntClust* [OUA 06], sur le même ensemble de données sur tous en terme de temps d'exécution (minimisation de

temps d'exécution des *AntClustering* de 1:30h à 120 secondes), Cependant l'approche que nous avons proposée admet aussi des faiblesses concernant le temps de lancement des agents sur tous pour les SVMs qui est trop long par rapport aux temps de décision. Les perspectives qui pourraient être envisagées en réponse à ces difficultés dans le cadre de futurs travaux sont :

- Le temps d'exécution est un problème qu'il serait optimisé par l'utilisation du mode distribué de Jade ou par l'utilisation d'une architecture parallèle pour minimiser le temps de lancement et d'apprentissage pour les machines à vecteurs de supports ou la réduction des messages envoyés entre agents de segmentation.
- Utiliser la capacité d'apprentissage des machines à vecteurs de supports pour entraîner des fournis artificielles pour la segmentation d'image

Enfin, autres perspectives indépendantes des difficultés rencontrées au cours de ces recherches sont :

- L'application des opérateurs de morphologie mathématique pour extraire le cerveau du crâne.
- L'utilisation du recalage pour les images de taille différente dans la segmentation multimodale.
- Développer un système expert capable de donner un diagnostic à travers les résultats de classification effectuée par notre application.
- Utiliser un autre mode de coopération (régions, contours, ...etc).
- Développer une ontologie spécifique à la segmentation des IRM cérébrales.
- Elargir l'espace d'application pour des images pathologique, avec une classe pathologie.
- Séparer l'os du cerveau en utilisant la morphologie mathématique.

Références

1. [AME 05] : S.Ameur, Z.Ameur, 2005, » *Revue des approches de segmentation d'image texturées : exemple des images morphologiques* » Setit, Université Mouloud Mammeri, Tizi-Ouzou, Algérie.
2. [ANN 02] : Anne-Sophie Capelle, 2002, « *Segmentation d'images IRM multi-échos tridimensionnelles pour la détection des tumeurs cérébrales par la théorie de l'évidence* », Thèse de doctorat, Université de Poitiers, France.
3. [ART et al]: Yusuf Artan, Xiaolei Huang, » *combining multiple 2-SVM classifiers for tissus segmentation*», Université de Lehigh, Suisse.
4. [BAR 02] : Paul Barthez, 2002-03. « *La tomodynamométrie (TDM) et imagerie par résonance magnétique (IRM) en médecine ventriculaire* ».
5. [BAR 00]: V.BARRA, JY.BOIRE, 2000, « *Tissue Characterization on MR Images by a possibilistic Clustering on a 3D Wavele Representation*», Journal of Magnetic Resonance Imaging, **11**:267-278.
6. [BAU 98]: M. Brandt, T. Bohan, L. Kramer, J. Fletcher, 1994, « *Estimation of CSF, White and Gray Matter Volumes in Hydrocephalic Children Using Fuzzy Clustering of MR Images*», Computerized Medical Imaging and Graphics, 18:25-34.
7. [BEN 07]: Anis Ben Ishak, 2007, » *Sélection de variables pas les machines à vecteurs de supports pour la discrimination binaire et multi classe en grande dimension*», Thèse doctorat, Université de Tunis, Tunisie.
8. [BEZ 81]: J .C. Bezdek, 1981, « *Pattern Recognition with Fuzzy Objective Function Algorithms*», Plenum Press, New- York.
9. [BEZ 93]: J .C Bezdek, 1993, « *Review of MR Image Segmentation Techniques using Pattern Recognition* », Medical Physics.
10. [BEN 94]: A. Bensaid, L. Hall, J. Bezdek, L. Clarke, 1994, « *Fuzzy Cluster Validity in Magnetic Resonance Images*», SPIE Medical Imaging, **2167**:454-464.
11. [BLO 96]: I. Bloch, 1996, « *Information combination operators for data fusion: A comparative review with classification*», IEEE Trans. Systems, Man, and Cybernetics 26(52-67).
12. [BOI 93] : O. Boissier, 1993, « *Problème du contrôle dans un système intégré de vision* », Thèse de doctorat, Institut National Polytechnique de Grenoble.
13. [BOU 99] : A. Boucher, 1999, « *Une approche décentralisée et adaptative de la gestion d'information en vision* », Thèse de doctorat, Université Joseph Fourier, Grenoble.

14. [BRO 91]: R. A. Brooks, 1991, «*Intelligence without representation*», Artificial Intelligence, 47, pp. 139–159.
15. [BRU93]: Brummer, M.E. Mersereau, R.M. Eisner, R.L. Lewine, 1993, «*Automatic detection of brain contours in MRI data sets* », Medical Imaging, IEEE Transactions on (12), Page(s):153 – 166.
16. [CAG 93]: S. Cagnoni, 1993, «*Neural network segmentation of magnetic resonance spin echo images of the brain* », Journal of Biomedical Engineering.
17. [CHE et al]: H.G. Chew, R.E. Bougner, C.C. Lim, 2001, «*Dual- ν support vector machine with error rate and training size biasing*», IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP).
18. [CLA 93]: L. Clarke, R. Velthuizen, S. Phuphanich, 1993, «*MRI: Stability of three Supervised Segmentation Techniques* », Magnetic Resonance Imaging, **11**:95-106.
19. [CLA 94]: M. Clark, 1994, «*MRI Segmentation Using Fuzzy Clustering Techniques* », IEEE Engineering in Medicine and Biology.
20. [MAS 99]: F. Masulli, A. Schenone, 1999, «*A Fuzzy Clustering based Segmentation System as Support to Diagnosis in Medical Imaging*», Artificial Intelligence in Medicine, **16**:129-147.
21. [COC 95]: S. Philipp, J-P. Cocquerez, 1993, «*Fusion de régions sur critères géométriques* », 14ème Colloque GRETSI.
22. [COL et al]: Marina Kolesnik and Ales Fexa, «*How robust is the SVM wound segmentation?* », Université de Karlsruhe, Allemagne.
23. [DAR 05]: Y. Darcy, Y. Guermeur, 2005, «*Radius-margin Bound on the Leave-one-out Error of Multi-class SVMs*», Technical Report RR-5780, INRIA.
24. [DEL 91]: S. Dellepiane, 1991, «*Image segmentation : Errors, sensitivity and uncertainty*», In 13th IEEE Engineering in Medicine and Biology Society, volume 13, pages 253–254.
25. [DUD 73]: R. Duda, 1973, «*Pattern Classification and Scene Analysis* », Edition Wiley.
26. [DOU 02] : Mme Weibei Dou, 2002, «*Segmentation d'images multispectrales basée sur la fusion d'informations : application aux images IRM* » Thèse de doctorat, Université Caen.
27. [FAH]: Fahed Abdallah, «*Les séparateurs à vaste marge (SVM)*», Centre de recherche Royallieu, Université de Technologie de Compiègne (UTC), France.
28. [FEB 95] : J. Ferber, 1995, «*Les systemes multi-agents, vers une intelligence collective* » Editeurs: G.M.P.
29. [FER] : Ferguen Adel, «*La plate-forme JADE* ».

30. [FOU 02] : Olivier Fourdrinoy, 2002, « *Création d'une librairie de Balises pour l'insertion d'agents JADE dans des pages Jsp* », Maîtrise d'informatique, Faculté Jean Perrin, Université d'Artois, Lens.
31. [GAU]: Eric Gaussier, «*Introduction aux SVMs*», http://www-clips.imag.fr/mrim/User/eric.gaussier/M1_AS/svm.pdf.
32. [GER 99] : Germond Laurence, 1999, « *Trois principes de coopération pour la segmentation en imagerie de résonance magnétique cérébrale* », Thèse de doctorat L'université Joseph Fourier.
33. [GER 98]: Germond Laurence, 1998, « *Coopération entre processus guidés par les données et les modèles pour la segmentation en IRM* », Technical Report RR1005-I, TIMCIMAG.
34. [GER 99]: Germond Laurence, 1999, « *Trois principes de coopération pour la segmentation en imagerie de résonance magnétique cérébrale* », Thèse de doctorat L'université Joseph Fourier.
35. [GER 00]: L. Germond, M. Dojat, C. Taylor, C. Garbay, 2000, «*A cooperative framework for segmentation of MRI brain scans*», Artificial Intelligence in Medicine, 20(1),
36. [GRE, 04]: Dominic Greenwood, 2004, «*FIPA: the Foundation for Intelligent, Physical Agents*», Whitestein Technologies AG.
37. [GHA 98]: A. Ghanei, H. Soltanian-Zadeh, J.P. Windham , 1998, «*Segmentation of the hippocampus from brain MRI using deformable contours*», Computerized Medical Imaging and Graphics, 22:203–216.
38. [GER 98]: L. Germond, 1998, « *Coopération entre processus guidés par les données et les modèles pour la segmentation en IRM* », Technical Report RR1005-I, TIMCIMAG.
39. [HAS 07]: D. Hasboun, 2006–2007, «*Neuroanatomie*», Université de Pierre Marie Curie, Paris.
40. [HAN 03]: J. Handl, 2003, «*Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative techniques*», Masters Thesis, universität Erlangen-Nürnberg, Erlangen, Germany.
41. [HEL 97]: Karsten Held, Elena Rota Kops, J. Bernd, 1997, « *Markov Random Field Segmentation of Brain MR Images* », IEEE Transactions on Medical Imaging, 16:878- 886.
42. [HAA 90]: Haacke E, Tkach J, 1990, «*Fast MR Imaging: «Techniques and Clinical Applications*», AJR 155, pp. 951-964.

43. [HAR 04] : R. Haroun, L. Hamami, F. Boumghar, 2004, « *Segmentation d'images médicales IRM, par un système hybride flou - croissance de régions, dans un système multiagents* », Journées d'études sur l'imagerie médicale JETIM Blida.
44. [HSU 02]: C.-W. Hsu, C.-J. Lin. A, 2002, «*comparison of methods for multiclass support vector machines*», Dans IEEE Transactions in Neural Networks, volume 13.
45. [JAG98] M. Cyril JAGGI, 1998, « *segmentation par méthode Markovienne de l'encéphale humain en imagerie par résonance magnétique : théorie, mise en œuvre et évaluation* ». Thèse de doctorat, Université Caen.
46. [JAR]: Michèle Jardino, «*Fouille de données dans les corpus de textes classification supervisée SVM*», Groupe Langues, Information et Représentations,<http://www.limsi.fr/Recherche/LIR>.
47. [JUS 88]: M. Just, M. Thelen, Tissue 1988, « *Characterization with T1, T2 and Proton Density Values:Results in 160 Patients with Brain Tumors* », Radiology, **169**:779-785.
48. [KAS 88]: M Kass, A Witkin, D Terzopoulos 1988, « *Snakes: Active contour models* », International Journal of Computer Vision, Springer
49. [KHO et al]: M. Khouadjia, H. Khanfouf, S. Meshoul, « *Une Approche adaptative pour la segmentation d'images : Implémentation sur la plate-forme Multi-agents NetLogo* », Laboratoire LIRE, Université Mentouri, Constantine, Algérie.
50. [KIV 84]: K. Kiviniitty, 1984, «*NMR relaxation Times in NMR Imaging*», Annals of Clinical Research, **40** : 4-6.
51. [KRI 96]: R. Krishnapuram, J. Keller, 1996. «*The Possibilistic C-Means Algorithm: Insights and Recommendations*», IEEE transactions on Fuzzy Systems.
52. [LAB 93] : S. Labidi, W. Lejouad, 1993, « *De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents* », Rapport de recherche, INRIA, N° 2004.
53. [LIM 89]: K.O. Lim, A.J. Pfefferbaum, 1989, «*Segmentation of MR brain images into cerebro-spinal fluid spaces, white and gray matter*», Journal of Computer Assisted Tomography, 13:588–593.
54. [LIU 99]: J. Liu, Y. Y. Tang, 1999, «*Adaptive Image Segmentation With Distributed Behavior-Based Agents*», IEEE Trans. Pattern Analysis And Machine Intelligence 21(6): 544-551.
55. [LUM 94]: E.D Lumer, et B. Faieta, 1994, «*Diversity and adaptation in populations of clustering ants*», in D. Cli®, P. Husbands, J. Meyer and S. Wilson (eds), « *From Animals to Animats 3, Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior*», The MIT press/Bradford Books.

56. [DEN 90]: J.L Deneubourg, S Aron, S. Goss, et J.M Pasteels, 1990, «*The self-organizing exploratory pattern of the argentine ant. Dans Journal on insect Behavior*».
57. [MAH 80]: D. Mahr, 1980, « *Theory of edge detection. In Proceedings of the Royal Society of London* », pages 197–217.
58. [MAG 99]: V. Magnotta, 1999, « *Measurement of Brain Structures with Artificial Neural Networks: 2D and 3D Applications* », *Radiology*.;211:781-790.
59. [MAT 65]: G. Matheron, 1965, « *Random Sets and Integral Geometry* », Edition Wiley
60. [MAZ et al]: S. Mazouzi, Z. Guessoum, F. Michel, « *Une approche multi-agents pour la segmentation d'images de profondeur* », Laboratoire LERI, IUT de Reims, B.P. 1035, 51687, Reims, France.
61. [MBA 03] : Aloys MBALA HIKOLO, 2003, « *Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance* » Thèse de doctorat, Université de Franche-Comté.
62. [MOH 06]: Mohamadally Hasan, Fomani Boris, 2006, «*SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges*», BD Web, Versailles St Quentin, France.
63. [MOM 99]: N. Monmarché, 1999, «*On Data Clustering with Artificial Ants. In:Freitas AA, editor, Data Mining with Evolutionary Algorithms: Research Directions*», Papers from the AAI Workshop, AAAI Press.
64. MON 00]: Nicolas Monmarché, 2000, «*Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation*», Thèse doctorat, Université de Tours.
65. [MOU 08]: Fabien Moutarde, 2008, «*Brève introduction aux Support Vector Machines (SVM)*», Centre de Robotique, Ecole des Mines de Paris.
66. [NAT 04] : Richard Nathalie, 2004, « *Une approche située, coopérative et décentralisée pour l'interprétation d'images cérébrales par RMN* », Thèse de doctorat de l'INPG, Grenoble.
67. [NEL 00]: Nello Cristianini, John Shawe-Taylor, 2000, «*An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*», Cambridge University Press.
68. [OSU et al]: E. Osuna, R. Freund, F. Girosi, 1997, « *Support Vector Machines: Training and Applications* », Laboratoire de l'Intelligence Artificiel MIT.
69. [OUA 06]: Ouadfel Salima, 2006, « *Contributions à la Segmentation d'images basées sur la résolution collective par colonies de fourmis artificielles* », thèse doctorat, université de Batna.

70. [OUA]: Salima Ouadfel, Mohamed Batouche, « *Des fourmis pour la segmentation des images* », université de Batna.
71. [PEN 99]: J. Peña, 1999, « *An empirical Comparison of four Initialization Methods for the k-means Algorithm* », Pattern Recognition Letters.
72. [PLA 99]: J. Platt, 1999, « *Fast training of support vector machines using sequential minimal optimization, advances in kernel methods-support vector learning* », Dans MIT Press.
73. [PRE 70]: J.M.S Prewitt, 1970, « *Picture Processing and Psychophysics* », Academic Press. pages 75–149.
74. [QUI 04]: Joël Quinqueton, 2004, « *Aspects socio-organisationnels dans les systèmes multiagents : L'intelligence artificielle en essaim* », Journée FING, Paris.
75. [RAM] : Lotfi Ramdani, « *JADE (Développement et Implémentation de systèmes Multi-Agents)* », Master recherche Informatique _LRI_ Université Paris Sud XI.
76. [RIC 01] : Pierre-Michel Ricordel, 2001, « *Programmation Orientée Multi-Agents : Développement et Déploiement de Systèmes Multi-Agents Voyelles* », Thèse de doctorat, laboratoire Leibniz.
77. [ROB 65]: G. Roberts, 1965, « *Machine perception of threedimensional solids* », Edition Cambridge. pages 159–197.
78. [ROS 03]: Fabrice Rossi, 2003 « *L'intelligence collective* ». <http://apiacoa.org/>.
79. [SCH 01]: H.G. Schnack, H.E. Hulshoff Pol, W.F.C. Baare, M.A. Viergever, and R.S. Kahn. , 2001, « *Automatic Segmentation of the Ventricular System from MR Images of the Human Brain* », NeuroImage, 14:95–104.
80. [SCH et al]: Benoit Scherrer, Michel Dojat, Florence Forbes, Catherine Garbay, « *Une Approche SMA pour la Segmentation Markovienne des Tissus et Structures Présents dans les IRM Cérébrales* », Laboratoire Jean Kuntzmann, Université de Grenoble (MISTIS)
81. [SEB 03]: Sébastien Gorges, 2003 « *Segmentation des Structures Sous Corticales Par Fusion Floue d'Informations sur des Images RMN* » INSERM, Grenoble.
82. [SEM 05]: M. Semchedine, L. Toumi, A. Moussaoui, 2005, « *Classification multimodale hybride des images IRM Cérébrales sous le système multi-agents MAGIQUE* », Mémoire de l'ingénieur, Université Ferhat Abbas, Sétif.
83. [SEM 06]: M. Semchedine, L. Toumi, A. Moussaoui, 2006, « *Nouvelle Approche de Classification Multimodale Hybride d'Images IRM dans un SMA* », Journées internationales de l'informatique graphique JIG .Tébessa.

- 84. [SEM 08]:** Moussa SEMCHEDINE, 2008, « *Système Coopératif Hybride de Classification dans un SMA : Application à la segmentation d'images IRM* », Thèse de Magister, Université Ferhat Abbas, Sétif.
- 85. [SEM et al]:** M. Semchedine, L. Toumi, A. Moussaoui, « *Un Système Multi-Agents Pour La Classification Coopérative D'images IRM Cérébrales* » Université Ferhat Abbas, Sétif.
- 86. [SER 82]:** J. Serra. 1982. « *Image Analysis and Mathematical Morphology* », volume 1. Academic Press, London.
- 87. [SKA 80]:** Christof Skrzpczyk, Toni Soueid , 1980, «*LEAP (Lightweight Extensible Agent Platform) : Présentation générale* ».IEEE Transactions on System, Man and Cybernetics, SMC, Vol. 11 No.161-70.
- 88. [SHU 99] :** J.A. Shufelt, 1999, «*Performance Evaluation and Analysis of Monocular Building Extraction from Aerial Imagery*», IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(4), pp. 311–326.
- 89. [SUC 99]:** J. Suckling, 1999, « *A modified Fuzzy Clustering Algorithm for Operator Independent Brain Tissue Classification of Dual Echo MR Images* » Magnetic Resonance Imaging.
- 90. [SOB78]:** I. Sobel. 1978. « *Codage des images binaires et détection des contours* », Image Processing. , 8:127–135, 1978.
- 91. [THI 97]:** J. P. Thiran, V. Warscotte, B. Macq, 1997, « *A queue-based region growing algorithm for accurate segmentation of multi-dimensional digital images* », Signal processing vol. 60, no1, pp. 1-10.
- 92. [VAP 98]:** V. N. Vapnik, 1998, «*Statistical learning theory*», Wiley, New York.
- 93. [VAP 95]:** V. N. Vapnik, 1995, «*The Nature of Statistical Learning Theory*», Springer Verlag, New York.
- 94. [VAP 71]:** V. N. Vapnik and A. Y. Chevonenkis, 1971, «*On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications*».
- 95. [VAP 91]:** V. N. Vapnik and A. Y. Chevonenkis, 1991, «*The necessary and sufficient conditions for consistency in the empirical risk minimisation method. Pattern Recognition and Image Analysis*».
- 96. [VAP 00]:** V. N. Vapnik and O. Chapelle, 2000, «*Bounds on error expectation for support vector machines*», Neural Computation.
- 97. [VEL 95]:** R. Velthuizen, 1995, « *Unsupervised Measurement of Brain Tumor Volume on MR Images* », Journal of Magnetic Resonance Imaging.

98. [VIN 91] : L. Vinnet. 1991, « *Segmentation et mise en correspondance de régions de paires d'images stéréoscopiques* », Université de Paris IX-Dauphine.
99. [VIN 99]: S. Vinitiski, 1999, « *Fast Tissue Segmentation Based on a 4D Feature Map in Characterization of intracranial Lesions* », Journal of Magnetic Resonance Imaging .
100. [WAR 95]: S. Warfield, J. Dengler, J. Zaers, C. Guttman, W. Wells III, G.J. Ettinger, J. Hiller, and R. Kikinis, 1995, «*Automatic Identification of Grey Matter Structures from MRI to Improve the Segmentation of White Matter Lesions*», The Journal of Image Guided Surgery, 1(6):326– 338.
101. [ZAM 04]: Olivier Zammit, Xavier Descombes, Josiane Zerubia, 2004, «*Apprentissage non supervisé des SVM par un algorithme des K-moyennes entropique pour la détection de zones brûlées*», projet commun ARIANA, INRIA.
102. [ZHA et al]: ZHANG Zhao, ZHANG Su, ZHANG Chen-xi, CHEN Ya-zhu, 2005, « *SVM for density estimation and application to medical image segmentation* », Université de science B.

Webliographie :

103. [1] : Introduction aux agents et systèmes multi-agents
<http://turing.cs.pub.ro/auf2/html/chapters/chapters.html>
104. [2] : brainweb www.bic.mni.mcgill.ca/brainweb

ملخص:

في هذه المذكرة ذكرنا مختلف طرق تقسيم صور الرنين المغناطيسي للدماغ، مع تعدد هذه الطرق لاحظنا انه لا توجد طريقة مثلى وكل طريقة تعطي نتائج مرضية في سياق معين فقط وهو ما دفعنا إلى التفكير في إنشاء هندسة متعددة الأعوان ذاتية التأقلم تمزج بين الخوارزمي الأوتوماتيكي النمل الاصطناعية (*AntClustering*) و الخوارزمي غير الأوتوماتيكي المصنفات عريضة الهوامش (SVMS) من اجل تقسيم صور الرنين المغناطيسي واستغلال التكامل الموجود بين مختلف الطرق. اقترحنا في هذه الطريقة استعمال خوارزمي المصنفات عريضة الهوامش من اجل الكشف عن حدود الأجسام و خوارزمي النمل الاصطناعية من اجل الكشف عن المناطق المتجانسة في الأجسام لصور الرنين المغناطيسي للدماغ ومن اجل التقليل من تأثير الأفراد (البيكسال) الغامضة والأفراد البعيدة أدخلنا مفهومي الرفض: الأول متعلق برفض الأفراد التي توجد بين مختلف الأقسام والثاني متعلق بالأفراد البعيدة كل البعد عن وسط التقسيم.

مفاتيح:

التقسيم، التصنيف الأوتوماتيكي، النمل الاصطناعية، المصنفات عريضة الهوامش، صور الرنين المغناطيسي، النظم متعددة الأعوان.

Résumé:

Dans ce mémoire on a essayé, de mettre en oeuvre un état de l'art sur les différentes méthodes de segmentation d'images IRM cérébrales. Cette diversité nous a guidé à développer une architecture multi-agents adaptative qui combine entre un algorithme supervisé (SVMS) et un algorithme non supervisé (*AntClustering*) pour profiter des avantages de chacun, afin que la segmentation caractérise mieux les différents structures cérébraux (MG, MB, LCR) et permet aussi de faire une comparaison ces deux types de méthodes.

On a proposé les machines à vecteurs de supports pour la détection des contours et les fourmis artificielles pour la détection et la localisation des régions homogènes avec les propositions que nous avons introduit pour améliorer la qualité et la rapidité de approche. Afin de diminuer l'influence des données aberrantes et ambiguës, nous avons introduit deux concepts de rejet : le rejet d'ambiguïté qui concerne les individus situés entre les frontières des différentes classes et le rejet d'aberrance concernant les individus qui sont très éloignés par rapport à toutes les classes.

Mots clé : Segmentation, Classification non supervisée, les fourmis artificiels, les machines à vecteurs de supports, Imagerie par Résonance Magnétique, les systèmes multi-agents.

Abstract:

In this memory we present different methods of segmentation of cerebral images MRI (Magnetic resonance imaging). this diversity of methods, we have conduct to develop an adaptative multi-agents architecture that combine between the supervised algorithm (SVMS) and the automatic algorithm (*AntClustering*) to profit the advantages of each one, in order that the segmentation characterize the different cerebrals structures (MG, MB, LCR) and permit also a comparison between the two types of methods.

We have made the supports vectors machines to detect outlines and the *AntClustering* to detect and to localize homogeneous regions and the modification that we have introduced to improve the quality and the rapidity of our approach. In order to make the algorithm more robust for the ambiguous and the aberrant data, two types of rejections were introduced: the rejection of ambiguity and the rejection of aberrance. We test the results on images MRI in which we proved the convergence of our algorithms for the various anomalies which these images can have.

Key words: Segmentation, Clustering, *AntClustering*, supports vectors machines, Resonance Magnétique imaging, multi-agents system.