

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de la Manouba

Institut Supérieur des Arts Multimédias



PROJET DE FIN D'ETUDES

Code IngRV09

Préparé en vue de l'obtention du diplôme d'Ingénieur en Informatique et Multimédia

**UNE APPROCHE MULTI-AGENTS POUR LA
DETECTION DE CONTOURS**

Réalisé par

Slim NAMOUCHI

Encadré par

Imed Riadh FARAH- ISAMM
Abdesslam BENZINOU – Lab-STICC-ENIB

Année Universitaire 2014-2015

Résumé

Le présent rapport a été élaboré dans le cadre du projet de fin d'études pour l'obtention du diplôme d'Ingénieur en Informatique et Multimédias de l'ISAMM. Ce projet est proposé par le laboratoire de recherche Lab-STICC à École Nationale d'Ingénieurs de Brest ENIB, intitulé «Une approche multi-agents pour la détection de contours», vise à développer un système multi-agents pour segmenter les images par approche contour. Les développements informatiques sont réalisés en Java, tout en profitant de tous les services offerts par la plateforme multi-agents Jade.

Mots clés : segmentation d'image, système multi-agents, approche contour.

Abstract

This report was prepared as part of the final project studies for obtaining the engineering degree in Computer Science and Multimedia of ISAMM. This project is proposed by the research laboratory Lab-STICC in National School of Engineers of Brest ENIB, entitled "A multi-agent approach for edge detection", aims to develop a multi-agent system to segment the images by Contour approach. IT developments are made in Java, while enjoying the services provided by the MAS platform Jade.

Keywords : image segmentation, multi-agent system, outline approach.

Dédicaces

Dieu source de toute connaissance

*A celui qui a toujours garni mes chemins de force et lumière... mon trop
cher père*

A la plus belle perle au monde... ma tendre mère

A mes frères

A ma sœur

*A toute ma famille pour l'amour et le respect qu'ils m'ont toujours accordés
A tous mes amis*

A toute personne qui m'a aidé à franchir un horizon dans ma vie...

Slim

Remerciements

Au terme de ce travail, je voudrais adresser mes sincères remerciements à Monsieur André PERENNOU, responsable du site Lab-STICC-ENIB, pour m'avoir accueillie et avoir accepté de me confier la tâche de réaliser ce projet dans un cadre convivial.

Je remercie mon encadrant Monsieur Abdesslam BENZINU, maître de conférences à École Nationale d'Ingénieurs de Brest ENIB, pour m'avoir suivi de près tout au long de ce projet, pour sa disponibilité et ses conseils toujours avisés.

Je remercie, également, mon superviseur Monsieur Imed Riadh Farah, maître de conférences et directeur de l'Institut Supérieur des Arts Multimédias de la Manouba, qui a bien voulu assurer la direction de ce travail, pour sa patience, son assistance et ses précieuses recommandations.

Je tiens également à exprimer toute ma gratitude à mon co-encadrant Monsieur Vineet Nagrath, post-doctorant à l'ENIB, pour ces précieux conseils et sa disponibilité tout au long de mon stage.

Un grand merci à tous les membres du laboratoire RIADI et du laboratoire Lab-STICC-ENIB pour leur chaleureux accueil et leur complicité.

Mon dernier mot s'adresse à tous les membres du jury pour l'honneur qu'ils m'ont donné de juger mon travail. Qu'ils trouvent ici ma reconnaissance et mon respect, sans oublier mes enseignants à l'ISAMM, je voudrais leur témoigner ma plus profonde gratitude pour la formation qu'ils m'ont prodiguée tout au long de mon cursus universitaire au sein de cet honorable institut.

Table des matières

Introduction générale	12
1 Contexte du projet	14
Introduction	14
1.1 Problématique	14
1.2 Objectifs	15
1.3 Cadre et Environnement du stage	15
1.3.1 Laboratoire RIADI	15
1.3.2 Laboratoire Lab-STIC-ENIB	16
Conclusion	17
2 Traitement numérique de l'image et systèmes multi-agents	19
Introduction	19
2.1 Difficultés principales dans la segmentation	19
2.2 Détection de contours	21
2.2.1 Les méthodes dérivatives	21
2.2.2 Les modèles déformables	24
2.3 Les systèmes multi-agents	25
2.3.1 Problématique	25
2.3.2 De l'Intelligence Artificielle Distribuée (IAD) aux systèmes multi-agents	26
2.3.3 Notion d'agent	27
2.3.4 Interaction et organisation des agents	28

2.3.5	Les types d'agents	29
2.4	Plate-formes multi-agents	31
2.4.1	La plate-forme JADE	31
2.4.2	La plate-forme MADKIT	31
2.4.3	La plate-forme Zeus	33
2.5	Présentation de la plate-forme multi-agents JADE	33
2.5.1	FIPA	33
2.5.2	Éléments de la plateforme	33
2.5.3	Le gestionnaire d'agent de JADE	33
2.5.4	Communication entre agents	34
	Conclusion	35
3	État de l'art	36
	Introduction	36
3.1	Approches contours	36
3.1.1	Approche gradient : calcul direct des dérivées	37
3.1.2	Approche gradient : masque de Sobel et filtre MIRA .	39
3.1.3	Seuillage	40
3.1.4	Discussion	43
3.2	Approches régions	43
3.2.1	classification Floue Fuzzy C-Mean	43
3.2.2	Segmentation par agrégation de pixels	46
3.2.3	Discussion	47
3.3	Approches coopérative régions-contours	47
3.3.1	Coopération mutuelle	48
3.3.2	Coopération des résultats	50
3.3.3	Coopération séquentielle	51
3.3.4	Discussion	52
3.4	Synthèse	52
	Conclusion	54

4 Approche proposée pour la détection de contours	55
Introduction	55
4.1 Les types d'agents	55
4.1.1 Agent contrôleur du système « Maître»	56
4.1.2 Agent de détection « Détecteur»	56
4.1.3 Agent de suivi « Suiveur »	57
4.2 Comportement des agents	58
4.2.1 Comportement de supervision	59
4.2.2 Comportement d'exploration	59
4.2.3 Comportement de suivi de contour	61
4.3 Spécification et conception	62
4.3.1 Besoins fonctionnels	62
4.3.2 Besoins non fonctionnels	63
4.3.3 Diagramme de classes	63
Conclusion	65
5 Réalisation et expérimentation	66
Introduction	66
5.1 Environnement logiciel	66
5.1.1 Latex	66
5.1.2 Visio	67
5.1.3 Matlab	67
5.1.4 Eclipse	67
5.1.5 JADE	68
5.2 Évaluation de la détection	68
5.2.1 Paramètres gérant le comportement des agents	68
5.2.2 Influence du bruit	74
5.2.3 Influence du nombre d'agents	74
5.3 Résultats	74
Conclusion	77

Conclusion générale et perspectives	78
5.4 Conclusion générale	78
5.5 Perspectives	79

Table des figures

1.1	Logo laboratoire RIADI	16
1.2	Logo Laboratoire LabSTICC	17
1.3	Organigramme Lab-STICC	18
2.1	Les approches de segmentation d'image	20
2.2	Quelques profils de contours	21
2.3	Dérivée première et seconde d'un signal.	22
2.4	Filtres de Roberts	23
2.5	Filtres de Prewitt	23
2.6	Filtres de Sobel	23
2.7	Un exemple de l'application des filtres de Roberts, Prewitt, et Sobel	24
2.8	Représentation schématique d'un SMA, du point de vue d'un observateur extérieur	25
2.9	Agent cognitif	30
2.10	Agent réactif	30
2.11	Plateforme JADE	32
2.12	Architecture Générale du MadKit.	32
2.13	Cycle de vie d'un agent JADE	34
2.14	Le paradigme de passation de messages asynchrones chez JADE	35
3.1	Type d'agents utilisés dans le système	37
3.2	Résultat de la segmentation sur une coupe 2D de deux poumons	38

3.3	Système multi-agents dans BRIC (foot note), (b) un agent dans BRIC	38
3.4	Les résultats obtenus sur les otolithes de poissons âgés de 1 à 8 ans	39
3.5	L'organisation du système multi-agents est articulée autour de trois groupes	40
3.6	Exemples de quelques résultats de segmentation	41
3.7	Architecture d'un SMA pour la reconstruction de scènes 3D . .	42
3.8	Résultat de la segmentation	42
3.9	Résultat de la segmentation	44
3.10	Résultats de la segmentation sur des images réelles	45
3.11	Résultat de la segmentation	45
3.12	Principe de diffusion des agents	46
3.13	Résultat de la segmentation, les différents contours et régions .	47
3.14	Boucle de perception-action des agents avec l'environnement et interactions entre des agents régions et contours qui coopèrent	49
3.15	Segmentation d'une image synthétique	49
3.16	Résulat de la segmentation sur des images de synthèses	50
3.17	Principe général de la segmentation réalisée par l'approche proposée	51
3.18	Principe de la génération successive des agents de segmentation à partir du contour du cerveau, pour la matière grise, puis la matière blanche	52
3.19	Vision s hématique du comportement de marquage du territoire	53
3.20	(a) Image originale, (b) Pyramide, (c) Division quaternaire, (d) Segments de contours	53
4.1	Archietcture de l'approche proposée	56
4.2	Les huit directions	56
4.3	Forme des capteurs avec une direction donnée	57
4.4	Dans (a), l'agent de détection trouve un contour. Il détermine deux directions possibles pour suivre le contour. Dans (b), le même agent crée alors deux agents de suivi de contour	57

4.5	Capteurs d'un agent	58
4.6	Diagramme d'états-transitions décrivant le comportement des agents	60
4.7	Forme de capteur d'agent détecteur	61
4.8	L'agent se déplace sur les points contour	62
4.9	Diagramme de classes de système	64
5.1	Logo Latex	67
5.2	Logo Microsoft Visio	67
5.3	Logo Matlab	67
5.4	Logo Eclipse	68
5.5	Logo JADE	68
5.6	Image de synthèse utilisée pour l'évaluation	69
5.7	Contours détectés pour différentes valeurs de L	70
5.8	Critères de détection pour différentes valeurs de L	71
5.9	Agent avec 5 capteurs de longueur $L = 4$	71
5.10	Contours détectés pour différentes valeurs de N	72
5.11	Critères de détection pour différentes valeurs de N	72
5.12	Contours détectés pour différentes valeurs de S	73
5.13	Critères de détection pour différentes valeurs de S	73
5.14	Contours détectés pour différents niveaux de bruit	74
5.15	Critères de détection obtenus pour les contours détectés	75
5.16	Critères de détection selon le nombre d'agents	75
5.17	Interface principale	76
5.18	Résultat de segmentation sur des contours en anneaux	76
5.19	Résultat de segmentation sur des contours en anneaux	77
5.20	Résultat de segmentation pour des contours en formes géométriques	77

Introduction générale

Pour un ordinateur, une image est une matrice dont les valeurs appelées pixels représentent des intensités lumineuses. Une image est donc vue comme un “tout” par un ordinateur et non pas comme une composition d’éléments. Par exemple, une image représentant un clavier d’ordinateur posé sur une table est perçue par un ordinateur comme étant une matrice de pixels. Si on veut mener une analyse sur la partie de l’image associée au clavier, on doit pouvoir séparer les pixels qui représentent le clavier de ceux représentant la table. C’est un problème de traitement d’image connu sous le nom de segmentation d’images. Cette segmentation est basée sur la détection d’indices visuels primaires. Les indices visuels primaires les plus utilisés sont les régions (points ayant une luminosité proche) et les contours (points présentant des changements de luminosité remarquables). Ainsi, on distinguera les méthodes de segmentation par détection de régions et / ou de contours. Les méthodes traditionnelles de traitement d’image utilisent une approche séquentielle consistant à traiter l’image pixel après pixel. Il existe d’autres approches qui consistent à utiliser les systèmes multi-agents pour traiter l’image. Un système multi-agent consiste en un certain nombre d’agents qui interagissent. Un agent est capable de percevoir son environnement, d’agir de manière autonome, d’interagir avec les autres agents, d’apprendre de ses expériences et de s’adapter à son environnement. Selon leur définition, l’utilisation des agents offre une souplesse dans la segmentation par rapport aux approches séquentielles. En effet, répartir les agents sur la surface d’une image peut présenter les avantages suivants.

- **Distribution** : répartition et spécialisation locale afin d’obtenir une meilleure réaction à l’environnement.
- **Adaptation** : capacité d’adaptation à un environnement dynamique.
- **Contrôle global** : du fait de la communication et de la coopération entre les agents.

Cependant, un agent en tant que composant logiciel agit de façon autonome selon l'implémentation de son comportement. Dans le cadre de ce PFE, notre but est de concevoir et développer un système multi-agents dédiés au traitement d'image et d'en dégager les apports.

Notre sujet s'inscrit donc dans un cadre d'un projet de fin d'études en vue de l'obtention du diplôme national d'ingénieur :

Le premier chapitre est consacré à l'étude du contexte du projet. Dans ce chapitre nous posons d'abord la problématique relative à notre sujet d'étude. Ensuite, nous présentons nos objectifs. Enfin, nous clôturons ce chapitre par une présentation du cadre et environnement de notre stage.

Le deuxième chapitre présente dans un premier volet les différents aspects de segmentation d'images afin de mieux comprendre le contexte de projet, ensuite, dans un second volet, nous présentons le concept des systèmes multi-agents.

Le troisième chapitre présente un état de l'art des différentes méthodes de segmentation à base des systèmes multi-agents afin de bien positionner le contexte de notre étude.

Le quatrième chapitre est consacré au système proposé avec une approche multi-agents. Nous y présentons une description détaillée du système.

Le cinquième chapitre présente les résultats expérimentaux obtenus.

Chapitre 1

Contexte du projet

Introduction

Dans ce chapitre introductif, nous allons commencer d'abord par poser la problématique et essayer de définir les enjeux et les contraintes que soulève notre sujet. Ensuite, nous expliciterons les objectifs que nous nous sommes fixés dans le but d'accomplir notre mission. Enfin, nous clôturerons cette partie par une présentation du cadre de notre stage et de l'organisme d'accueil.

1.1 Problématique

La plupart des méthodes classiques de traitement d'images traitent les pixels séquentiellement, en partant du premier pixel de l'image et en allant jusqu'au dernier. Une autre approche consiste à utiliser un système multi-agent afin d'effectuer une tâche complexe de manière distribuée.

Utiliser des algorithmes de segmentation ayant une approche globale sur toute l'image, peut conduire le système à trop de rigidité et à un manque d'adaptabilité face à un contexte particulier. C'est une des raisons pour lesquelles nous avons envisagé l'utilisation de systèmes multi-agents. Chaque agent peut alors traiter localement l'information dans l'image, et délibérer selon les perceptions qu'il a à la fois de l'image et des différents agents qui l'entourent. Les systèmes multi-agents sont un bon moyen pour distribuer la résolution d'un problème. De plus, cette distribution permet de donner une certaine robustesse au système : l'erreur produite par un agent est pondérée par le travail effectué par chaque agent. À défaut de donner des solutions

optimales, les systèmes multi-agents offrent de bonnes solutions robustes.

1.2 Objectifs

Le but principal de ce stage est d'étudier et implémenter un système multi-agent pour le traitement d'images, en particulier dans la résolution des problèmes classiques comme le prétraitement, la restauration, la détection de contours ou la segmentation en général. Les points importants du système comme l'organisation, l'interaction entre agents, la rapidité et le parallélisme du traitement, l'adaptabilité et la modélisation des connaissances a priori, seront essentiellement étudiés. Cette étude fait suite à plusieurs travaux réalisés au sein de l'équipe de recherche au LabSTIC-ENIB, en particulier ceux décrits dans [Benzinou et al., 2004], [Guillaud et al., 2002] et [Guillaud, 2000].

1.3 Cadre et Environnement du stage

Le stage s'intègre dans le cadre d'un projet de fin d'études en vue de l'obtention du diplôme national d'ingénieur en informatique et multimédias de l'institut supérieur des arts multimédias de la Manouba (ISAMM).

Étant proposé par le laboratoire LabSTIC-ENIB (site de Brest - France) et en Co-encadrement avec le laboratoire de recherche RIADI-SIVIT (Tunisie), notre stage, d'une durée globale de six mois, s'est décomposé en trois principales phases, de deux mois chacune :

- une phase de documentation et d'étude bibliographique qui s'est tenu au sein du Laboratoire de recherche RIADI-SIVIT.
- une deuxième phase de développement et d'expérimentation qui s'est concrétisé par un stage pratique au sein du laboratoire LabSTIC à l'École Nationale d'Ingénieurs de Brest ENIB.
- une dernière phase portant sur la synthèse et l'évaluation des résultats obtenues qui s'est déroulé au sein du Laboratoire de recherche RIADI.

1.3.1 Laboratoire RIADI

Le laboratoire RIADI a été habilité en 1999 par le Ministère de l'Enseignement Supérieur, de la Recherche Scientifique et de la Technologie et par le Secrétariat d'État à la Recherche Scientifique et à la Technologie sous le

N° de code IT01 pour la période 1999-2002. Le laboratoire a une grande expérience d'enseignement, de recherche et de collaboration socio-économique particulièrement dans les domaines des technologies d'information et de communication.



FIGURE 1.1 – Logo laboratoire RIADI

Acronyme de système intelligent en vision, imagerie et télédétection, le pôle SIVIT du laboratoire RIADI est une branche interdisciplinaire du secteur des sciences et technologies de l'information et de la communication fédéré par l'imagerie.

Il développe des travaux fondamentaux et appliqués dans un large spectre de l'informatique, du traitement d'images et de la télédétection. Dans tous ces domaines, l'image - réelle ou de synthèse, en 2 ou 3 dimensions, fixe ou animée, multimodale - joue un rôle important.

L'image est l'objet même des études en modélisation géométrique et animation, en visualisation et en analyse d'images ou de séquences d'images. Elle est un type de données complexe privilégié dans les travaux sur l'algorithme, la programmation, la classification, la fouille de données et l'asservissement visuel. Son codage et son transport alimentent les travaux sur les réseaux, l'optimisation de son traitement mobilise les compétences en calcul parallèle. Sa capture et son analyse sont essentielles pour le pilotage de robots et elle est l'une des bases fondamentales des études en télédétection.

1.3.2 Laboratoire Lab-STIC-ENIB

Lab-STICC, **Laboratoire des Sciences et Techniques de l'Information, de la Communication et de la Connaissance**, créé en janvier 2008, le Lab-STICC est un laboratoire structurant la recherche en STIC en Bretagne océane. Il résulte de la fusion de quatre laboratoires :

- LEST (UMR CNRS 6165)
- TAMCIC (UMR CNRS 2872)
- LESTER (FRE CNRS 2734)
- SABRES (UBS EA 3371)



FIGURE 1.2 – Logo Laboratoire LabSTICC

Organigramme

L'équipe d'accueil EA3883 a regroupé jusqu'en fin 2012 les enseignants-rechercheurs de l'Université de Bretagne Occidentale (UBO), de l'École Nationale d'Ingénieur de Brest (ENIB), et de l'Ecole Nationale Supérieure d'Ingénieur des Etudes et Techniques d'Armement (ENSIETA), travaillant dans le domaine de l'informatique des systèmes complexes. Depuis Janvier 2012, les membres du EA 3883 ont majoritairement intégré le Lab-STICC/UMR 6285, cet laboratoire regroupe, sous le thème principal « des capteurs à la connaissance », trois pôles de recherche :

- **le pôle MOM** : Micro-ondes, Optoélectronique, Matériaux
- **le pôle CID** : Connaissance, Information, Décision
- **le pôle CACS** : Communications, Architectures, Circuits et Systèmes

Conclusion

Le présent chapitre nous a permis de présenter le projet dans son environnement général et de définir les obligations de travail. Nous présenterons dans le chapitre suivant d'abord les différents aspects de segmentation d'images, ensuite, dans un second volet, nous présenterons le concept des systèmes multi-agents.

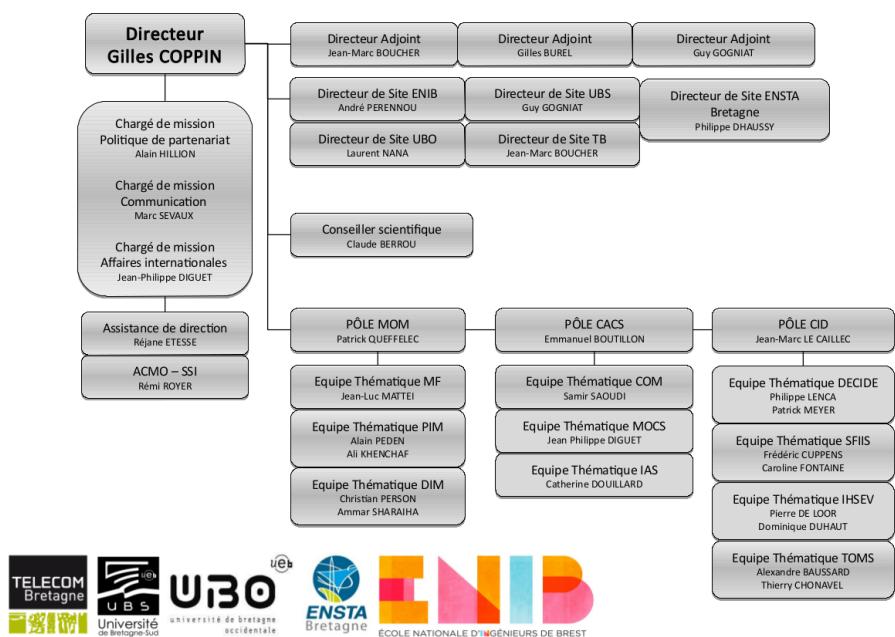


FIGURE 1.3 – Organigramme Lab-STICC

Chapitre 2

Traitement numérique de l'image et systèmes multi-agents

Introduction

La segmentation est une opération qui consiste à séparer une image en régions et en contours (figure 2.1). La complexité de la tâche est très variable. Cela dépend de la nature de l'information à extraire mais aussi de celle de l'image à analyser. De plus, il existe un nombre important d'algorithmes de segmentation. Ce nombre est en lien avec les différents types d'informations à extraire en fonction de la scène analysée. Chacune d'entre-elles nécessite des stratégies de segmentation adaptées. Cela dépend aussi du cadre d'application étudié : il peut s'agir d'imagerie médicale, de photographie réelle, d'images de radar/sonar/satellites ou autres. Beaucoup de ces algorithmes n'ont pas de fondements théoriques communs et participent à un développement horizontal du domaine. Une première difficulté vient donc du flou qui règne dans la discipline. Bien souvent, l'algorithme est conçu directement à partir du résultat souhaité sans passer par de quelconques axiomes. Son utilisation dépend de leur cadre d'application.

2.1 Difficultés principales dans la segmentation

Le traitement numérique de l'image peut se résumer à l'extraction de deux primitives complémentaires : les contours qui définissent les limites des régions et inversement, les régions (opération de segmentation) qui définissent les contours. En ne considérant que le domaine spatial (à l'instar des domaines

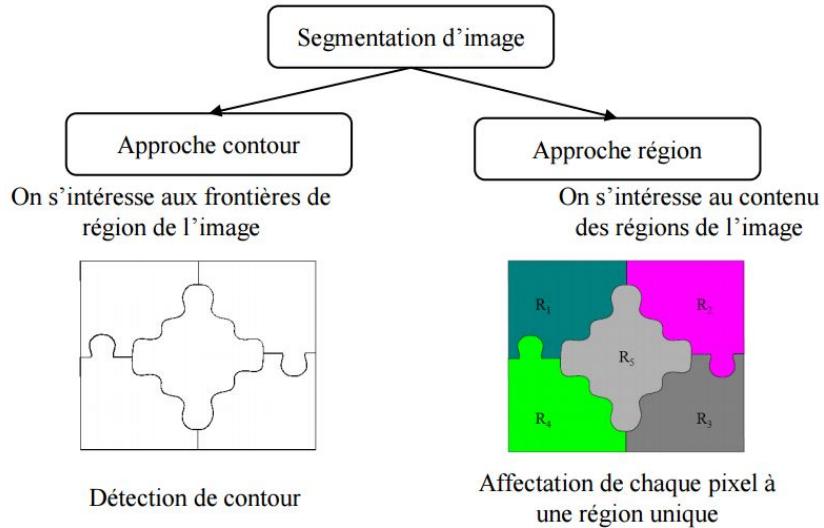


FIGURE 2.1 – Les approches de segmentation d’image

fréquentiels, spatio fréquentiels, paramétriques, etc...), trois principales approches sont utilisées :

- La segmentation de contours (voir la section 2.2 pour la détection des contours) : assez souvent, les résultats obtenus présentent une sous-segmentation de l’image. La méthode utilisée n’a pas su détecter tous les contours et donc toutes les régions.
- La détection de régions : à l’opposé, les résultats obtenus présentent une sur-segmentation de l’image.
- Le mélange des deux, où la détection de contours peut être utilisée pour initialiser les algorithmes de détection de régions. La méthode est judicieuse, car pour combler les limites présentées par une approche, il s’agit d’utiliser l’approche opposée mais complémentaire. Cependant les problèmes inhérents (sur ou sous-segmentation) aux limites de ces deux approches ne sont aucunement supprimés. Ils sont juste mieux considérés.

Les outils traditionnels utilisés pour la détection des contours ou des régions présentent plusieurs problèmes. D’une part, comme nous l’avons déjà présenté, la résolution influence directement les algorithmes utilisés, ou leurs paramètres. Ensuite, l’utilisation d’outils de dérivation (Ex : gradient, laplacien) est souvent mal adaptée au bruit présent dans l’image. Avec suffisamment de bruit, ces opérateurs perdent de leur efficacité :

- Le bruit crée des transformations dans la détection des contours.

- Les contours peuvent être pris pour du bruit et peuvent donc être ignorés.

2.2 Détection de contours

La segmentation par approche contour consiste à identifier les changements entre les régions. En général, un élément de contours est un point de l'image appartenant à la frontière de deux ou plusieurs objets ayant des niveaux de gris différents. Les variations d'intensité de lumière et de couleurs sont très bien perçues par le système visuel humain. En effet, une frontière est définie comme un endroit de l'image où le changement en niveaux de gris est le plus important. Ces frontières constituent le contour des objets. Le contour s'appuie alors sur les transitions plus ou moins importants (de type marche, toit, pointe) (voir la figure 2.2), ou encore sur les frontières entre zones homogènes distinctes ou non (présence d'une ligne).

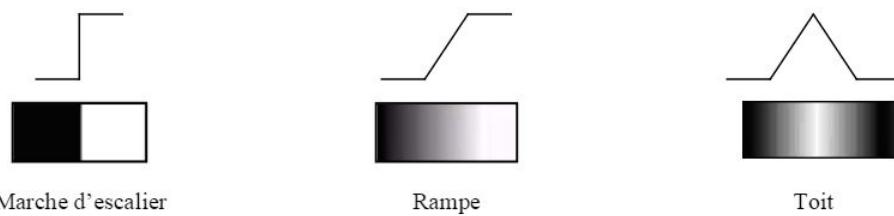


FIGURE 2.2 – Quelques profils de contours

2.2.1 Les méthodes dérivatives

Les méthodes dérivatives sont les plus utilisées pour détecter des transitions d'intensité par différenciation numérique (Première et deuxième dérivé). Ce sont des méthodes locales qui balayent l'image avec un masque définissant la zone d'intérêt. A chaque position, un opérateur est appliqué afin de détecter les transitions significatives au niveau de l'attribut de discontinuité choisi. Le gradient permet de calculer la pseudo-dérivée d'une image sur les axes x et y. La dérivée permet de détecter les contours en déterminant les passages par 0 dans la dérivée seconde. La figure 2.3 illustre cette détection : le passage par zéro dans la dérivée seconde correspond à un changement brusque de valeur (significatif de la présence d'un contour) dans le signal.

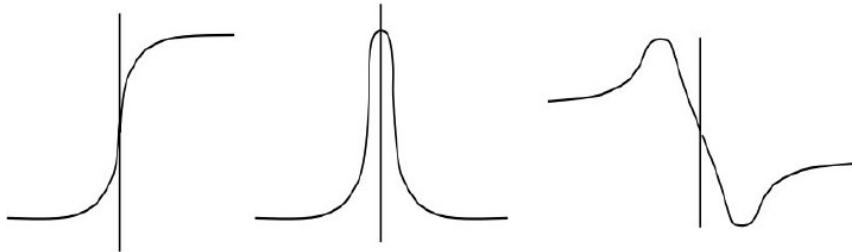


FIGURE 2.3 – Dérivée première et seconde d'un signal.
Vous pouvez trouver à gauche le signal source, au centre la dérivée première de ce dernier, et à droite la dérivée seconde.

Le gradient d'une image

Le gradient d'une image se note comme suit :

$$Ix(x, y) = \frac{\partial If(x, y)}{\partial x} \quad (2.1)$$

$$Iy(x, y) = \frac{\partial If(x, y)}{\partial y} \quad (2.2)$$

En chaque (x, y) de l'image, il est possible de calculer le vecteur gradient. Le module et la direction de ce vecteur sont donnés par :

$$amplitude : G = \sqrt{Ix^2 + Iy^2} \quad (2.3)$$

$$G \approx \max(Ix, Iy) \quad (2.4)$$

$$G \approx \max(|Ix|, |Iy|) \quad (2.5)$$

La norme du gradient correspond à la valeur de la dérivée, et la direction du gradient permet de connecter la direction verticale correspondant au contour.

Filtres déivateurs

Comme nous venons de le voir, détecter les fortes variations d'un signal (les contours) revient à détecter les pics de sa dérivée (Gradient) ou les passages par zéro de sa dérivée seconde (Laplacien). Généralement, on effectue la dérivée sur chaque axe de l'image en utilisant un filtre déivateur-lisseur. Par exemple, on utilise les filtres de Roberts (voir figure 2.4), de Prewitt (voir figure 2.5) ou de Sobel (voir figure 2.6). Puis on stocke le module de la valeur renvoyée par le gradient de chaque axe. Cette méthode a toutefois des limites dans la mesure où il est difficile de déterminer arbitrairement le seuil correct à appliquer. Vous trouverez un exemple de l'application de ces filtres sur une image dans la figure 2.7.

$$\begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$$

FIGURE 2.4 – Filtres de Roberts

$$\begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

FIGURE 2.5 – Filtres de Prewitt

$$\begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

FIGURE 2.6 – Filtres de Sobel

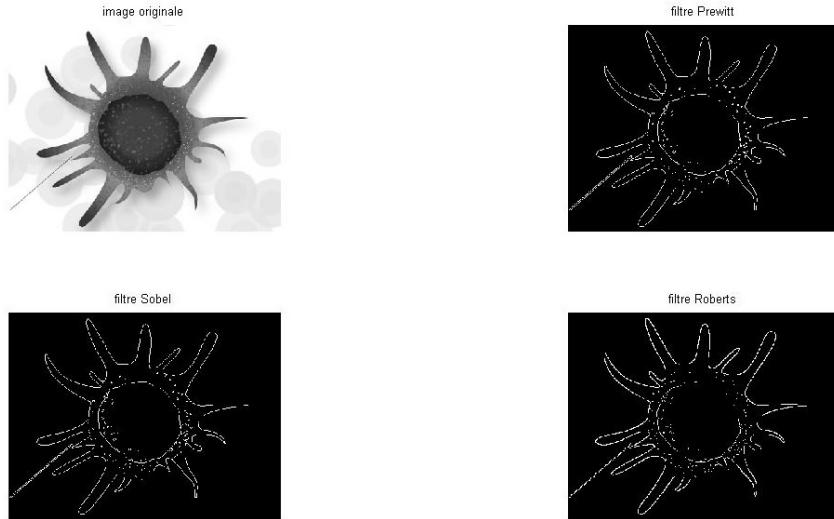


FIGURE 2.7 – Un exemple de l’application des filtres de Roberts, Prewitt, et Sobel

Le cas du laplacien

On utilise la deuxième dérivé pour calculer le laplacien. Les points de contour sont situés aux passages par zéro du laplacien

$$\Delta If(x, y) = \frac{\partial If(x, y)}{\partial x^2} + \frac{\partial If(x, y)}{\partial y^2} \quad (2.6)$$

Le calcul du laplacien a des avantages. Il est proche du mécanisme de la vision humaine et utilise un seul paramètre (le masque). Il n’utilise pas le seuil de significativité de l’amplitude. Parmis ces inconvénient, on peut citer : sa grande sensibilité au bruit et le fait qu’il n’est pas d’information sur l’orientation du contour

2.2.2 Les modèles déformables

Les algorithmes de segmentation fondés sur les modèles déformables ont l’avantage, par rapport aux méthodes dérivatives, de fournir des contours ou surfaces fermés. Parmis ces méthodes, on trouve les contours actifs et les ensembles de niveaux

2.3 Les systèmes multi-agents

Un système multi-agents est un ensemble d'agents localisés dans un environnement virtuel ou réel et interagissant entre eux selon une certaine organisation (figure 2.8), plus ou moins dynamique et plus ou moins auto-adaptative. L'auto-adaptation se manifeste par le fait que les entités/agents interagissent ensemble pour s'adapter à leur environnement et aux contraintes que ce dernier lui impose. C'est la spécialisation de chaque agent et l'interaction complexe qui naît entre eux qui fait émerger un résultat global. Ce résultat global rétroagit sur chaque agent du système les faisant converger vers une solution commune pourtant non comprise par un agent individuel. La dynamique fournie par le système et sa « logique » se résume comme étant un phénomène émergent donnant une fonctionnalité supplémentaire que chaque agent ne peut fournir individuellement. Par exemple, l'organisation collective qu'ont les fourmis entre elles leur permet de trouver le plus court chemin entre leur fourmilière et un point de nourriture [Mahdjoub, 2011].

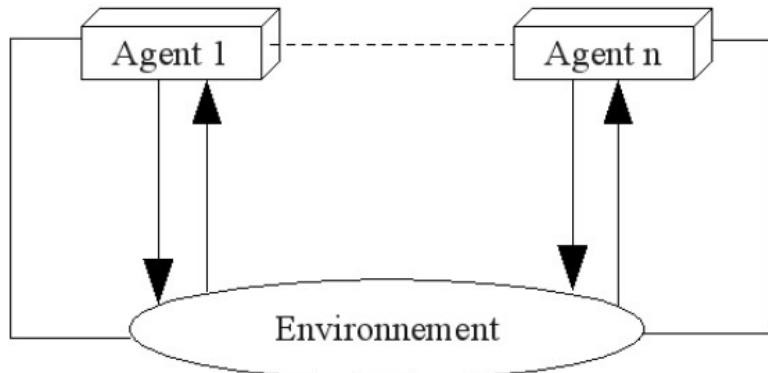


FIGURE 2.8 – Représentation schématique d'un SMA, du point de vue d'un observateur extérieur

2.3.1 Problématique

Interpréter une image consiste à produire une description symbolique de cette image, c'est-à-dire à reconnaître et décrire les différentes entités qui la composent. Ce problème de « vision artificielle » s'apparente aux problèmes classiques de l'intelligence artificielle. Il s'agit en effet d'un problème de prise de décision : tout au long du processus de vision, des choix doivent être faits.

2.3.2 De l'Intelligence Artificielle Distribuée (IAD) aux systèmes multi-agents

Le but de l'Intelligence Artificielle Distribuée est de concevoir des approches de développement de communautés d'agents intelligents qui puissent interagir de façon coopérative mais aussi conflictuelle ou concurrente afin de résoudre des problèmes complexes.

Les systèmes de vision sont composés d'un grand nombre d'entités traduisant des connaissances hétérogènes, tant descriptives qu'opératoires. Dans le but de fournir un cadre favorisant une intégration harmonieuse de ces composants, ils utilisent l'une des approches de l'Intelligence Artificielle Distribuée comme architecture de contrôle. Les justifications d'un tel choix sont nombreuses [Haroun, 2005] :

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

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

Avant d'aborder les systèmes multi-agents, il est utile d'évoquer quelques notions de base sur les agents, c'est l'objet des sections suivantes.

2.3.3 Notion d'agent

Le concept d'agent a fait l'objet d'études pour plusieurs décennies dans différentes disciplines. Il a été non seulement utilisé dans les systèmes à base de connaissances, la robotique, le langage naturel et d'autres domaines de l'intelligence artificielle, mais aussi dans des disciplines comme la philosophie et la psychologie.

Il existe de multiples définitions pour un agent selon le point de vue privilégié. Par exemple, [Ferber, 1995], a donné une définition minimale d'agent : on appelle agent une entité physique ou virtuelle :

- capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- mue par un ensemble de tendances sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser,
- qui possède des ressources propres,
- capable de percevoir mais de manière limitée son environnement,
- qui ne dispose que d'une représentation partielle de cet environnement, et éventuellement aucune,
- qui possède des compétences et qui offre des services,
- qui peut éventuellement se reproduire,
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Pour notre travail, nous retenons la définition adoptée par [Haroun, 2005] d'après les travaux de M.Wooldridge : « Un agent est un système informatique, situé dans un environnement, qui agit de façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu ».

Les notions « *situé* », « *autonome* » et « *flexible* » sont définies comme suit :

- **situé** : L'environnement fournit aux agents des entrées sensorielles qui leurs permettent d'agir sur lui. Dans notre contexte, l'environnement des agents est l'image.
- **autonome** : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.

- **flexible** : l'agent dans ce cas est :
 - capable de répondre à temps : l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans les temps requis ;
 - proactif : un agent ne fait pas que réagir à son environnement mais il est aussi capable de produire de lui-même des actions motivées par des buts.
 - social : l'agent doit être capable d'interagir avec les autres agents (logiciels et humains), quand la situation l'exige, afin de compléter ses tâches ou aider d'autres agents à accomplir les leurs.

Bien entendu, selon les applications, certaines propriétés sont plus importantes que d'autres, il peut même s'avérer que pour certains types d'applications, des propriétés additionnelles soient requises. Il convient cependant de souligner que la présence des propriétés qu'on vient de voir comme l'autonomie, la flexibilité et la sociabilité, donne naissance au paradigme agent tout en le distinguant des systèmes conventionnels comme les systèmes distribués [Haroun, 2005].

2.3.4 Interaction et organisation des agents

Outre la notion d'environnement, qui comme on l'a expliqué, est fondamentale dans la définition des SMA, deux autres concepts caractérisent ces derniers. Le concept d'interaction qui permet aux agents d'échanger et le concept d'organisation, qui les structure.

Interactions entre agents

L'interaction est une notion importante dans les systèmes multi-agents, car un objectif dans la définition d'une société d'agents est de créer des échanges entre ces agents afin d'augmenter les capacités du groupe. L'interaction peut être définie comme la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques [Ferber, 1995].

Les interactions sont basées sur :

- **La coopération** : travailler ensemble dans un but commun.
- **La coordination** : organiser la solution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées.
- **La négociation** : parvenir à un accord acceptable pour toutes les parties concernées.

Organisation des agents

Les organisations permettent de maîtriser et de structurer les agents, qui n'ont qu'une vision partielle de l'ensemble afin d'éviter les incohérences et les redondances au niveau système.

2.3.5 Les types d'agents

Les agents peuvent être classés selon leurs degrés d'autonomie, de coopération et d'adaptation, caractéristiques généralement considérées comme principales en IAD. Deux types d'agent ont longtemps été distingués dans les SMA :

Les agents cognitifs

Les agents cognitifs sont généralement « intentionnels » c'est-à-dire qu'ils possèdent des objectifs et des plans explicites leur permettant d'accomplir leurs buts. Ils agissent suivant un cycle (perception/décision/action). La figure 2.9 représente un agent cognitif.

Ces agents disposent d'une base de connaissances comprenant les diverses informations liées à leurs domaines d'expertise et à la gestion des interactions avec les autres agents et leur environnement. Cette base de connaissance regroupe ce qu'on appelle les états mentaux de l'agent. Un état mental ou cognition est une structure cognitive élémentaire, il concerne les croyances, les désirs et les intentions Fig. 2.9 [Hadjer, 2010] :

On peut définir ces trois notions comme suit :

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

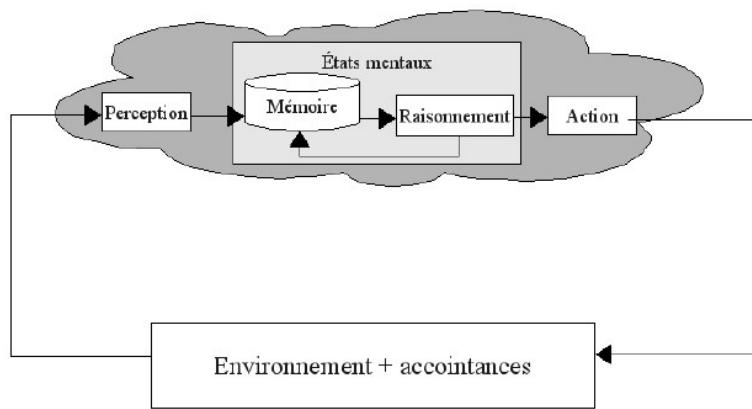


FIGURE 2.9 – Agent cognitif

Les agents réactifs

Les agents réactifs au contraire des agents cognitifs, ne sont pas « intelligents » pris individuellement. Ils ne peuvent que réagir à des stimuli simples provenant de leur environnement (Figure 2.10 [Hadjer, 2010]). Leur comportement est donc simplement dicté par leur relation à leur entourage sans qu'ils ne disposent d'une représentation des autres agents ou de leur environnement.

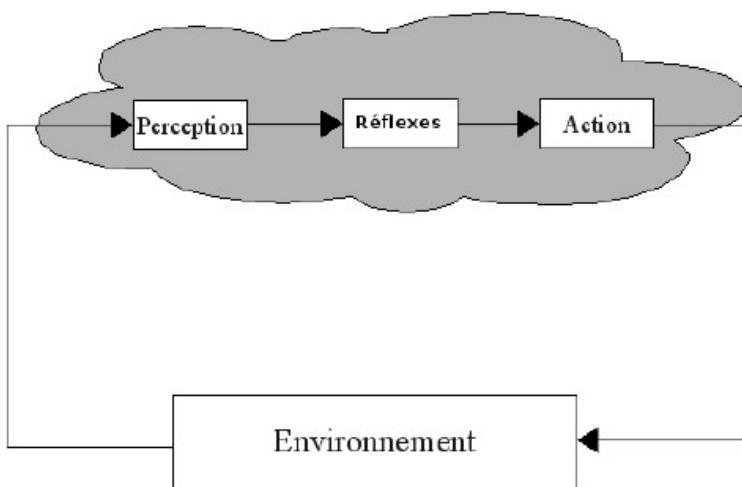


FIGURE 2.10 – Agent réactif

2.4 Plate-formes multi-agents

Le meilleur moyen pour construire un système multi-agent(SMA) est d'utiliser une plateforme multi-agent. Une plate-forme multi-agent 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 se présenter sous la forme d'environnement de programmation (API) et d'applications permettant d'aider le développeur. Nous présentons les principes de trois plates-formes. Nous avons choisi un ensemble de plates-formes populaire et régulièrement mis à jour, couvrant autant d'aspects des systèmes multi-agents, y compris les modèles d'agent, interaction, coordination, organisation,... qui ont une bonne documentation, la disponibilité de téléchargement, et fondés sur des modèles d'études bien connus. Donné ces critères, nous avons choisi trois plates-formes :

- MadKit.
- JADE.
- Zeus.

2.4.1 La plate-forme JADE

JADE (Java Agent DEvelopment), est un plat-forme de développement de systèmes multi-agents, open-source et basé sur le langage Java. Il offre en particulier un support avancé de la norme FIPA-ACL, ainsi que des outils de validation syntaxique des messages entre agents. (figure 2.11).

2.4.2 La plate-forme MADKIT

MadKit (Multi agent development Kit), est une plate-forme de systèmes multi-agents, figure 2.12, [Ferber, 1995] on choisit le langage Java pour des raisons de portabilité, la richesse des bibliothèques de bases, les possibilités de sérialisation d'objets et de présence de réflexivité. MadKit, en tant qu'outil de développement de SMA, a été motivé par le besoin d'avoir une plate-forme la plus souple possible, et capable de s'adapter à différents modèles d'agents et domaines d'applications.

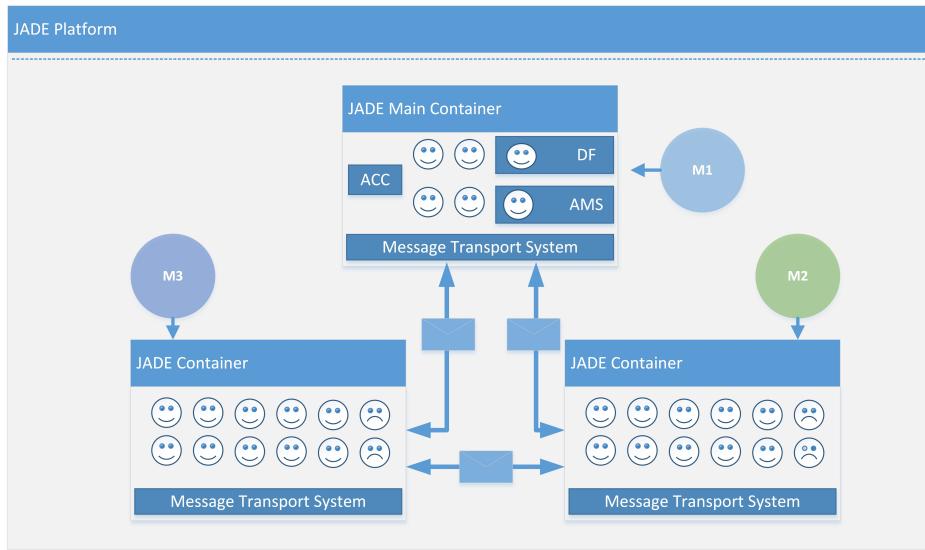


FIGURE 2.11 – Plateforme JADE

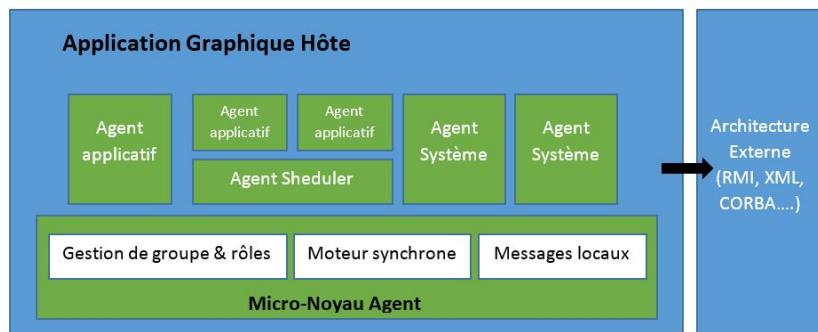


FIGURE 2.12 – Architecture Générale du MadKit.

2.4.3 La plate-forme Zeus

Développer par le programme de recherche d'agent du laboratoire de recherche des systèmes intelligent de British Telecom. L'atelier Zeus permet de concevoir des systèmes multi-agents répartis [Nwana et al., 1999]. Cet atelier, développé en Java, génère automatiquement le code Java à partir des agents spécifiés graphiquement.

2.5 Présentation de la plate-forme multi-agents JADE

Dans ce paragraphe, nous introduisons avec plus de détails la plate-forme multi-agents JADE qui tire avantage d'un nouveau modèle de communication basé sur les protocoles de messagerie instantanée.

2.5.1 FIPA

La FIPA (Foundation for Intelligent Physical Agents) est une organisation de normalisation qui été formée dans le but de prévenir la prolifération des systèmes d'agents incompatibles et de promouvoir la technologie à base d'agents.

2.5.2 Éléments de la plateforme

JADE possède trois modules principaux (nécessaire aux normes FIPA), ces trois modules sont activés à chaque démarrage de la plate-forme.

- **DF « Directory 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.

2.5.3 Le gestionnaire d'agent de JADE

JADE offre un modèle prédéfini à ses agents. Ce modèle est composé principalement d'un mécanisme de connexion à la plateforme, d'un dispatcheur

de message, et d'un ensemble de comportements divers appelés « Behavior »

Un agent JADE est une classe qui hérite de la classe Agent et qui redéfinie des méthodes qui définissent le cycle de vie de l'agent dans la plateforme. (voir figure 2.13, [Giovanni, 2003] :

- La méthode **setup** est la première méthode qui sera appelée après instanciation de l'agent par le container.
- la méthode **doDelete** permet de demander au container de détruire l'agent.
- Avant que l'agent soit détruit, la méthode **takeDown** est appelée.
- Un agent peut se déplacer d'un container à l'autre.
- Avant chaque opération de migration, la méthode **beforeMove** est appelée
- Après chaque opération de migration, la méthode **afterMove** est appelée

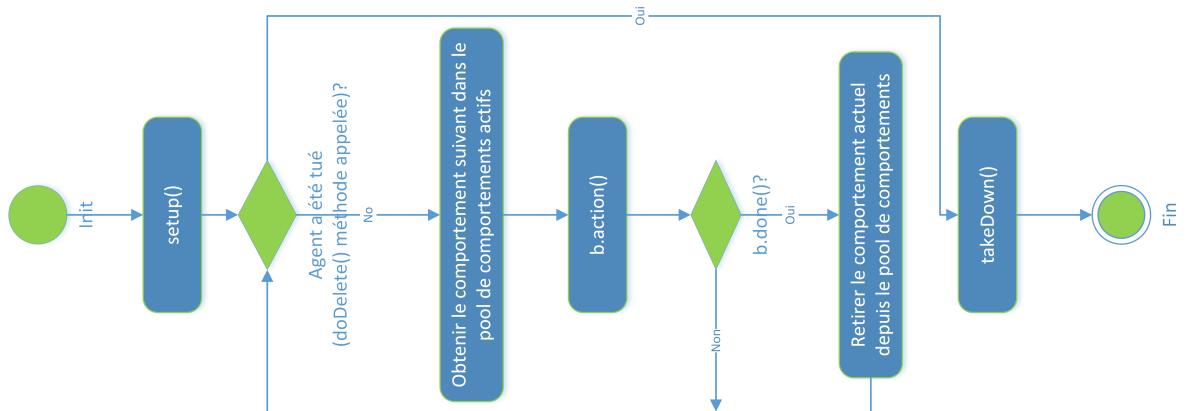


FIGURE 2.13 – Cycle de vie d'un agent JADE

2.5.4 Communication entre agents

Une des caractéristiques les plus importantes que les agents de JADE fournissent est la capacité de communiquer, la figure 2.14, [Giovanni, 2003]. Le paradigme de la communication adoptée est le passage de message asynchrone. Chaque agent a une sorte de boîte aux lettres (la file d'attente de messages de l'agent) où JADE poste les messages envoyés par d'autres agents. Chaque fois qu'un message est écrit dans la file d'attente de l'agent récepteur est notifiée. Si et quand l'agent prend effectivement le message de la file d'attente de message à traiter, il est tout à fait au programmeur cependant.

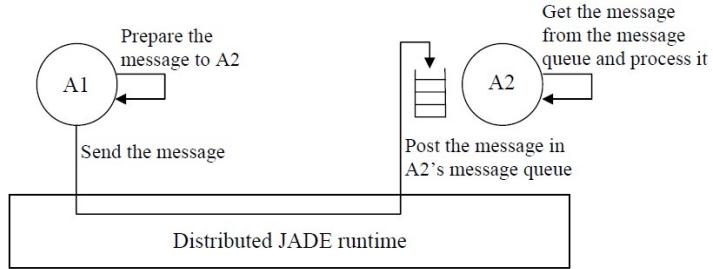


FIGURE 2.14 – Le paradigme de passation de messages asynchrones chez JADE

Conclusion

Ce chapitre a servi à présenter les différentes approches pour la segmentation en contours. L'utilisation d'un tel algorithme pose des problèmes, ces algorithmes ne s'adaptent pas à l'information présente dans l'image. Ils traitent l'image toute entière de la même manière, ce qui force le concepteur à se reporter sur des heuristiques, ou sur des algorithmes plus avancés.

Dans une deuxième partie, nous avons présenté les systèmes multi-agents, qui sont des systèmes permettant la modélisation distribuée des traitements et des connaissances, différents modes de coopération, etc. Nous avons vu qu'il existait deux types d'agents, réactifs et cognitifs, les agents du deuxième type sont utilisés dans peu de travaux, ils concernent des systèmes d'une certaine complexité, ceci n'exclut pas l'utilisation des agents réactifs dans la résolution de problèmes complexes. Le meilleur moyen pour construire un système multi-agent(SMA est d'utiliser une plateforme multi-agent. Dans le cadre de notre travail, c'est la plateforme multi-agents JADE qui a été choisie, une brève présentation de cette dernière est faite dans ce chapitre.

Chapitre 3

État de l'art

Introduction

Plusieurs systèmes multi-agents ont été proposés dans le domaine du traitement d'images. Ces systèmes introduisent des solutions intéressantes pour remédier à différents problèmes tels que la manipulation de connaissances hétérogènes, le contrôle de haut niveau sur les traitements de bas niveau, et la parallélisations et la distribution des traitements. Dans cet état de l'art, nous ne considérons que les travaux qui ont adressé une solution basée agent pour la segmentation d'images.

Nous avons séparé les approches multi-agents en trois domaines, les systèmes qui se basent sur une approche de segmentation en contour, d'autres qui se base sur une segmentation en régions et des approches qui fusionnent les deux principes région-contour

3.1 Approches contours

Les contours correspondent généralement à des changements brusques des propriétés physiques et géométriques de l'image perçue, qui sont causés principalement par la différence naturelle entre les régions adjacentes (les frontières des régions), et par le bruit.

3.1.1 Approche gradient : calcul direct des dérivées

Travaux relatifs à [Mahdjoub et al., 2006]

Principe Dans [Mahdjoub et al., 2006], les autres ont proposé un système multi-agent capable de segmenter les poumons d'un patient sur une coupe de scanner. Les agents sont influencés par un champ de vecteurs (Gradient Vector Flow), les dirigeants vers les contours. Lorsqu'ils détectent un contour, ils se mettent à le suivre. Ensuite, ils interagissent ensemble afin de former un maillage relativement précis (suivant un coefficient). Quatre types d'agents sont concernés :

- les agents de recherche de contours.
- les agents de suivi de contours.
- les agents noeud (leur seul rôle est de lier un agent avec un autre en formant des segments pour le maillage).
- les agents de fin de contours (à la différence des agents noeuds, les agents de fin de contours ne sont liés qu'avec un seul agent).

Le travail locale de tous ces agents rend émergents la représentation des bords par segments (voir la figure 3.1).

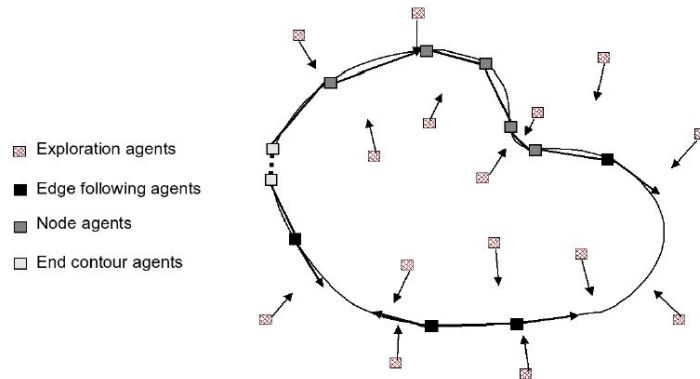


FIGURE 3.1 – Type d'agents utilisés dans le système

Résultat Détection de contours à travers le système proposé sur une coupe 2D de deux poumons, figure 3.2

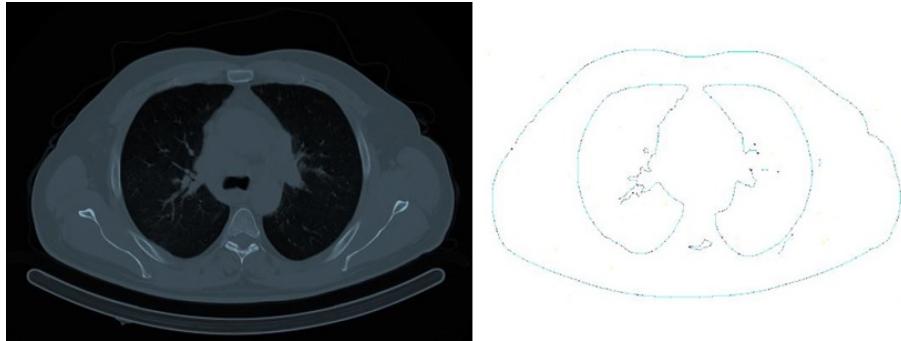


FIGURE 3.2 – Résultat de la segmentation sur une coupe 2D de deux poumons

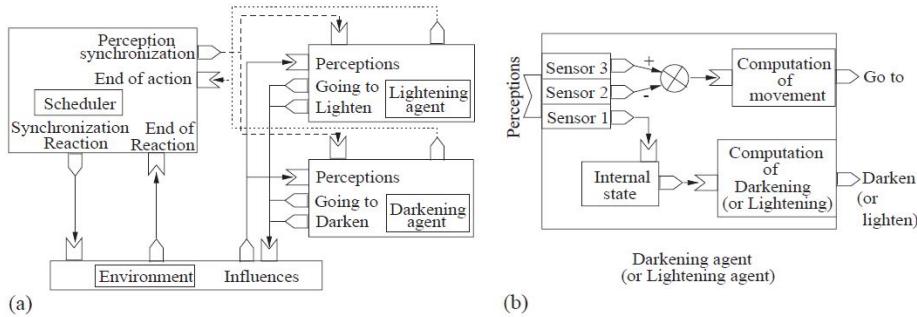


FIGURE 3.3 – Système multi-agents dans BRIC (foot note), (b) un agent dans BRIC

1

Travaux relatifs à [Benzinou et al., 2004]

Principe Dans [Benzinou et al., 2004], les auteurs ont proposé un système composé d'agents réactifs, dont le but est de détecter les contours d'images biologiques, plus précisément pour calculer l'âge des poissons en essayant de calculer le nombre de cercles dans ces otolites. Le système est basé sur le langage multi-agent oRis. Deux types d'agents sont utilisés :

- agent de noircissement (pour les régions sombres).
- agent de luminance (pour les régions claires).

Leurs actions visent à renforcer les régions par l'accroissement de leur contraste. Les agents parcourant les contours sont totalement indépendants les uns des autres. Ils sont dotés de capteurs pour les régions sombres et des capteurs pour les régions claires. (figure 3.3).

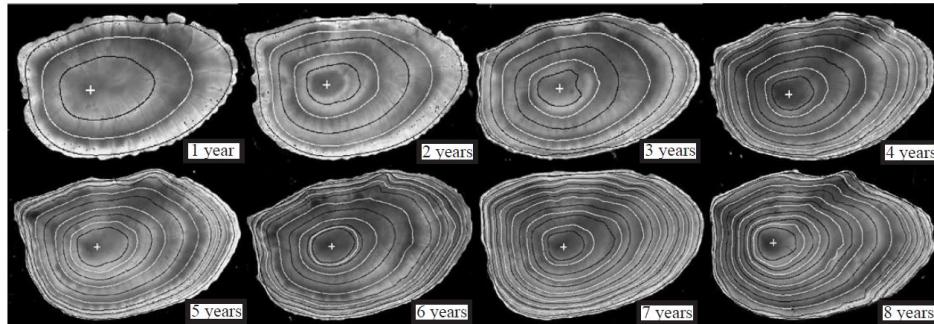


FIGURE 3.4 – Les résultats obtenus sur les otolithes de poissons âgés de 1 à 8 ans

Résultat La figure 3.4 représente un ensemble de résultats obtenus en utilisant cette méthode de détection d’âge des poissons.

3.1.2 Approche gradient : masque de Sobel et filtre MIRA

Travaux relatifs à [Mohammed and Said, 2012]

Principe [Mohammed and Said, 2012] a proposé un système multi-agent en se basant sur 3 types d’agents (voir la figure 3.5) :

- **Agent master** : effectue les traitements de haut niveau, il supervise tout le système ; il évalue la complexité de l’image afin de créer un nombre de partitions, crée et initialiser les agents de partition et reconstruire l’image segmentée.
- **Agents de partition** : joue les rôles suivants ; étiqueter les régions à l’intérieur de la partition et mettre les partitions de l’image aux agents détecteurs-suiveurs.
- **Agents détecteurs-suiveurs** : Ces agents travaillent au niveau inférieur de l’image afin de détecter et de suivre les points de contours, Ils utilisent des méthodes appropriées pour la détection et le suivi du plan. L’agent choisit son opérateur de la segmentation selon l’étiquette de la zone où il est.

Résultat La Figure 3.6 montre quelques exemples d’application de la méthode dans la segmentation d’images puis leur reconstruction. Le type d’images (a, b : circuit électronique ; c, d : la médecine ; e, f : visage).

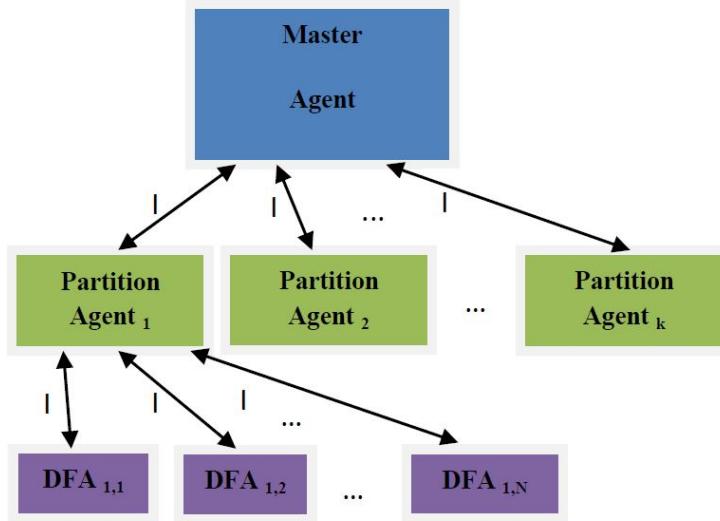


FIGURE 3.5 – L’organisation du système multi-agents est articulée autour de trois groupes

3.1.3 Seuillage

Une fois la norme du gradient calculée en chaque point de l’image, avant, il y avait un seuil unique qui amenait soit à des fausses alarmes, soit à des lacunes. Il était très difficile, voir impossible, de trouver un seuil adapté à toute l’image.

Travaux relatifs à [Mazouzi et al., 2005]

Principe Dans [Mazouzi et al., 2005], leur solution consiste à guider continuellement le processus de segmentation par une reconstruction partielle des primitives géométriques, qui sont obtenus à leur tour à partir des résultats courants de la segmentation. Les agents détecteurs de contour, qui représentent les agents de segmentation, adaptent, chacun, ses seuils de lissages et de détection en fonction des agents qui lui sont interdépendants. De ce fait, un point de l’image n’est plus analysé uniquement en se basant sur sa valeur et sur celles de ses points voisins, mais en se basant aussi sur sa position par rapport aux agents détecteurs qui opèrent dans son voisinage. (Figure 3.7).

Résultat Les résultats de segmentation et de reconstruction sont exposés et comparés dans deux situations différentes. La première correspond

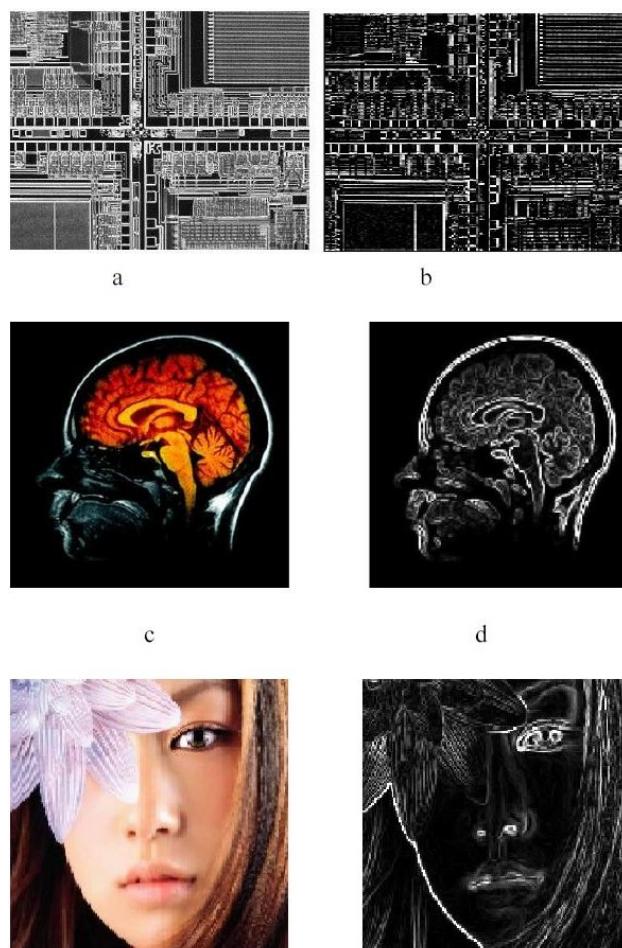


FIGURE 3.6 – Exemples de quelques résultats de segmentation

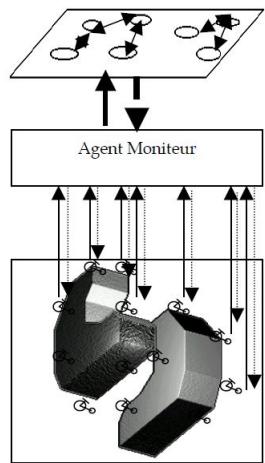


FIGURE 3.7 – Architecture d'un SMA pour la reconstruction de scènes 3D

au cas où les agents détecteurs sont en totale autonomie sans aucun mécanisme de coopération. L'agent moniteur était désactivé, et dans ces conditions les agents détecteurs vont évoluer indépendamment de toute organisation. Chaque agent continue le traitement avec les seuils initiaux de lissage et de détection (Figure 3.8). La seconde situation correspond au cas où les agents détecteurs adaptent leurs comportements en fonction de leurs interdépendances.

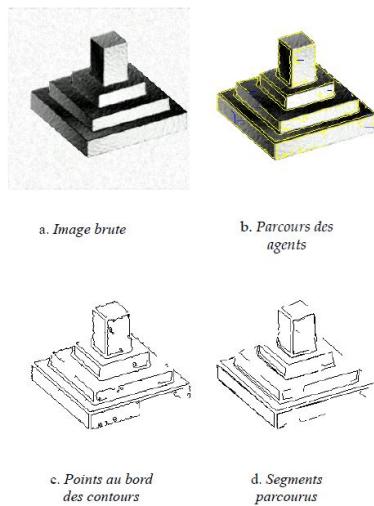


FIGURE 3.8 – Résultat de la segmentation

3.1.4 Discussion

La segmentation par contours permet de détecter les limites entre les régions de l'image, les détecteurs de contours utilisés peuvent être simples, comme les opérateurs de Sobel ou de Roberts. La détection de contours permet de repérer dans les images les objets qui s'y trouvent avant d'appliquer le traitement uniquement sur ces objets. Cette approche présente quelques avantages au niveau de la précision et la rapidité d'une segmentation, mais elle reste plus sensible au bruit.

3.2 Approches régions

L'approche région consiste à partitionner l'image en un ensemble de régions (un ensemble groupé de pixels). Ces régions doivent coller le mieux possible avec les objets de l'image concernée pour que la segmentation soit pertinente. Usuellement, chaque objet élémentaire de l'image est caractérisé par des attributs relativement homogènes, d'où la segmentation en régions agit en exploitant ces attributs pour donner des régions aussi homogènes.

3.2.1 classification Floue Fuzzy C-Mean

Travaux relatifs à [Haroun et al., 2004]

Principe Une approche similaire [Haroun et al., 2004] portent sur la segmentation d'image IRM du cerveau. Les auteurs y présentent une approche coopérative mettant en œuvre deux algorithmes de segmentation d'image, que sont la classification floue Fuzzy C-Mean (pour gérer l'incertitude et l'imprécision) et la croissance de région (pour agir localement sur l'image). Un système multi-agents est introduit dans la phase de croissance de région afin d'améliorer la qualité de la segmentation. Il existe deux types d'agents :

- un agent de type "contrôleur".
- plusieurs agents de type "Croissance" déployés sur l'image.

Résultat La Figure 3.9 montre la différence entre une segmentation classique par FCM (a et b) et une segmentation coopérative avec un système multi-agents.

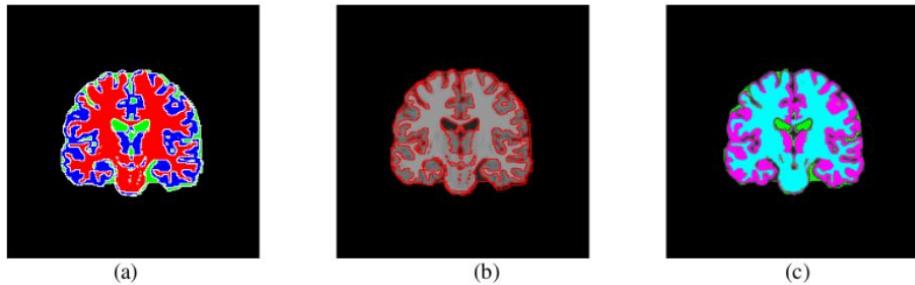


FIGURE 3.9 – Résultat de la segmentation

Travaux relatifs à [Hadjer, 2010]

Principe Dans [Hadjer, 2010], l’objectif de ce travail est de déployer sur une plate-forme de visualisation 3D, un système de segmentation d’images par résonance magnétique du cerveau.

La segmentation a pour but d’extraire les trois principaux tissus composant le cerveau : la matière blanche (MB), la matière grise (MG) et le liquide céphalo-rachidien (LCR). Dans ce travail, l’auteur fait coopérer différentes méthodes de segmentation, afin de tirer parti des avantages de chacune d’elles. La première méthode appliquée est une méthode de classification, qui utilise l’image dans sa globalité pour la segmenter, alors que la deuxième méthode permet de tenir compte des caractéristiques locales de l’image. Cette coopération est bâtie dans un environnement multi-agents.

Résultat Les résultats présentés dans la figure 3.10 montrent que l’approche permet d’avoir une classification plus nette, c’est-à-dire que l’on ne constate pas l’apparition de pixels dans la matière blanche, classifiés comme appartenant à la matière grise.

Travaux relatifs à [MANGOUA, 2008]

Principe Dans [MANGOUA, 2008], son approche est basée sur la logique floue. Pour cela, l’auteur utilise des agents de segmentation en région qui se répartissent à la surface de l’image en agrégant des pixels à partir d’un germe selon le comportement général donné par un algorithme dont les grandes lignes sont les suivantes.

- Sélection de tous les pixels candidat.
- Evaluation des pixels : l’adéquation de chaque pixel candidat doit être numériquement mesurée. Cela se fait souvent à l’aide d’heuristiques.

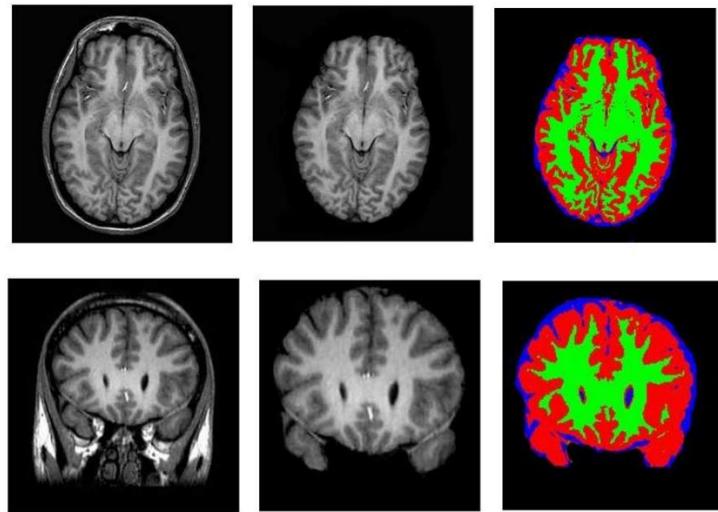


FIGURE 3.10 – Résultats de la segmentation sur des images réelles

- Tri des pixels candidats par rapport à la précédente évaluation. Le meilleur candidat sera retenu.
- L'intégration du meilleur pixel candidat. Les caractéristiques de la primitive peuvent alors être mises à jour.

Résultat Le système a été testé grâce aux images fournies avec la bibliothèque fournie et ceux de la base d'image [Z James, 2015].



FIGURE 3.11 – Résultat de la segmentation

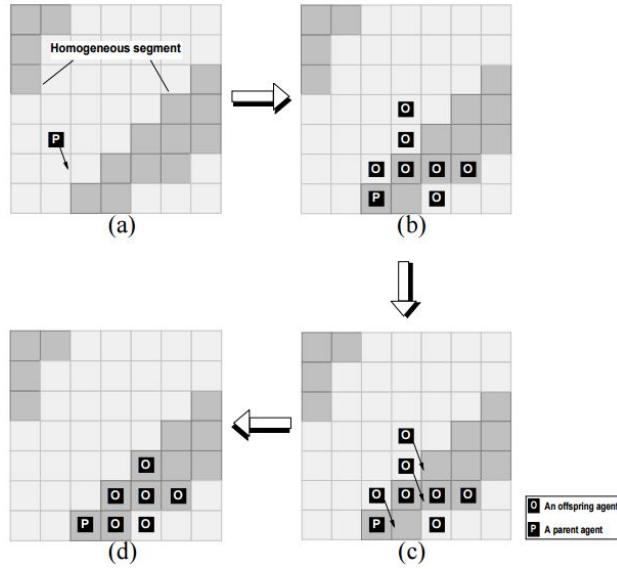


FIGURE 3.12 – Principe de diffusion des agents

3.2.2 Segmentation par agrégation de pixels

Le principe de l'agrégation de pixel est le suivant :

- on choisit un germe et on fait croître ce germe tant que des pixels de son voisinage vérifient le test d'homogénéité.
- Lorsqu'il n'y a plus de pixel candidat dans le voisinage, on choisit un nouveau germe et on itère le processus.

Travaux relatifs à [Liu and Tang, 1999]

Principe [Liu and Tang, 1999] ont proposé un système multi-agent pour la segmentation d'IRM du cerveau. Les agents se basent directement sur des critères de contrastes, de valeurs maximums/minimums, relatifs aux pixels avoisinants. Quatre types d'agents sont utilisés pour étiqueter les pixels de l'image en fonction de leur degré d'appartenance aux différentes régions. Les agents qui réussissent à détecter une région homogène se dupliquent sur leur voisinage afin d'explorer l'intégralité de la région (voir figure 3.12 image (b)). Pour ce faire, les agents dupliqués présentent les mêmes propriétés que leurs pairs. Les agents peuvent se déplacer à des pixels adjacents, ou disparaître dans l'image. A cet égard, nous considérons le comportement des agents comme réactif, comme il est entièrement activé et donc déterminée par l'environnement local des agents.

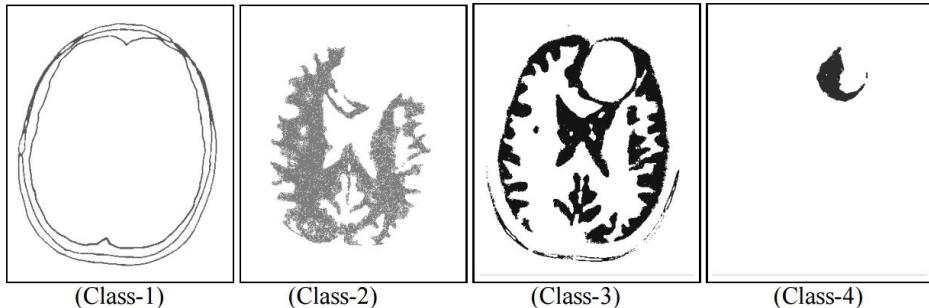


FIGURE 3.13 – Résultat de la segmentation, les différents contours et régions

Résultat Une vue séparée des contours identifiés et régions homogènes est donné dans la figure 3.13, 4 classes sont distinguées :

- Classe 1 : les contours.
- Classe 2 : région ramification.
- Classe 3 : région enfermant.
- Casse 4 : Région de la tumeur.

3.2.3 Discussion

Les méthodes de cette approche s'intéressent au contenu de la région (des informations locales ou globales). Elles regroupent les pixels vérifiant des propriétés communes (niveau de gris, écart-type,...). Il existe plusieurs méthodes telles que la segmentation par croissance de région, par division de région, et par fusion de région. Cette approche présent ces deux principaux avantages : la fermeture des frontières et la densité de l'information extraite, mais elle possède des inconvénients ; la difficulté d'identifier les critères pour agréger les pixels ou pour fusionner et diviser les régions et que les limites des régions obtenues sont généralement imprécises et ne coïncident pas exactement avec les limites des objets de l'image.

3.3 Approches coopérative régions-contours

La segmentation par coopération régions-contours exploite les avantages de ces deux types de segmentation afin d'aboutir à un résultat de segmentation plus précis et plus fidèle que celui obtenu à l'aide d'une seule technique. En effet, les algorithmes combinant les techniques de segmentation basées sur les régions et celles basées sur les contours prennent avantage de la nature complémentaire de l'information sur la région et sur le contour.

3.3.1 Coopération mutuelle

Les deux types de segmentation coopèrent mutuellement au cours de leur processus d'exécution.

Travaux relatifs à [Porquet et al., 2003]

Principe Dans [Porquet et al., 2003] des cartes de régions, ainsi que des cartes de contours sont préalablement générées en utilisant des algorithmes classiques en traitement d'images. L'algorithme du quadtree est utilisé pour déterminer des régions homogènes en termes de luminance. Le filtre de Shen est utilisé pour déterminer les contours. Chaque région et chaque contour sont ensuite pris en charge par un agent spécialisé. Cette approche est basée sur le principe de coopération des systèmes multi-agents (Figure 3.14) :

- **Coopération région-région** pour la fusion de régions : il doit exister un désir mutuel de fusion entre les deux régions pour que la fusion soit autorisée.
- **Coopération contour-région** pour la fusion ou la division de régions : il ne doit pas exister de contour entre les deux régions qui souhaitent fusionner/diviser.
- **Coopération contour-contour** pour le raccordement de contours : il doit exister un désir mutuel de raccordement entre deux contours proches pour que le raccordement soit autorisé.

Résultat Dans la figure 3.15, en haut à gauche, l'image originale ; en haut à droite, la carte des contours obtenues par la méthode de Deriche ; en bas à gauche, la segmentation en régions initiales obtenue par la méthode des quadtrees ; en bas à droite, la segmentation finale obtenue par le SMA.

Travaux relatifs à [Bellet, 1998]

Principe Le système proposé par [Bellet, 1998] est destiné à la segmentation d'images par coopération d'opérateurs contours et régions. Son travail suggère de construire simultanément les contours et les régions en se basant sur une approche incrémentale utilisant un système à bases de tâches. Cette approche incrémentale permet de construire la solution progressivement. Une gestion efficace de l'information peut être mise en œuvre ce qui se traduit par des prises de décisions motivées par l'information existante, l'utilisation

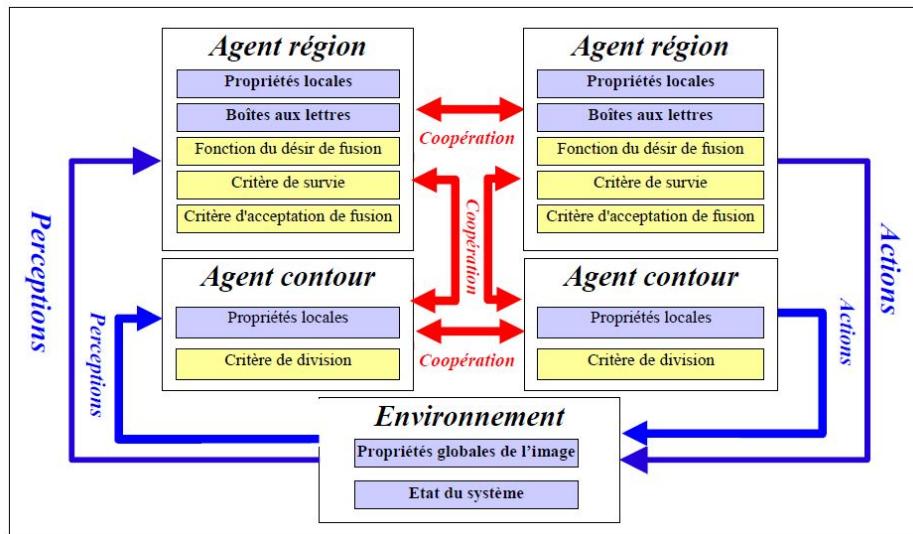


FIGURE 3.14 – Boucle de perception-action des agents avec l'environnement et interactions entre des agents régions et contours qui coopèrent

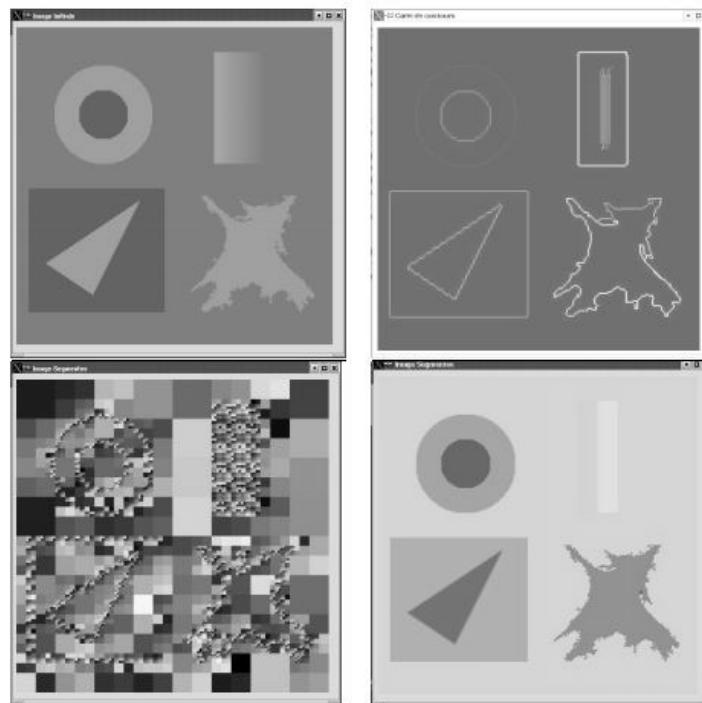


FIGURE 3.15 – Segmentation d'une image synthétique

au maximum de l'information disponible et la création de l'information manquante si le besoin se fait sentir.

Résultat Pour l'évaluation de son système [Bellet, 1998] a utilisé des images de synthèse. La figure 3.16 présente l'image originale, ainsi que la carte de référence des contours.

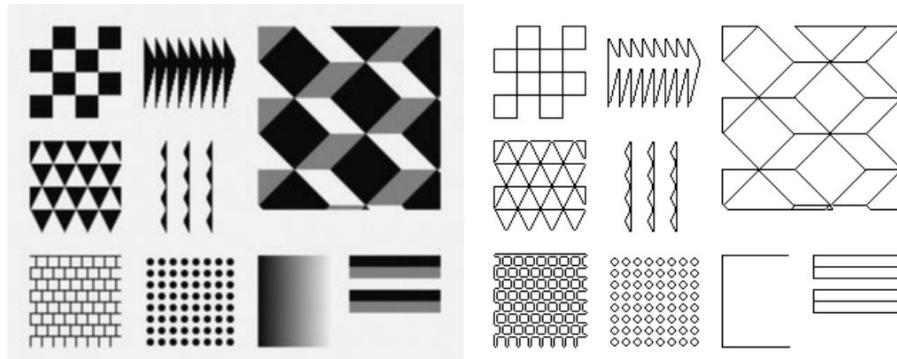


FIGURE 3.16 – Résulat de la segmentation sur des images de synthèses

3.3.2 Coopération des résultats

Les deux types de segmentation sont exécutés indépendamment. La coopération se fait sur les résultats obtenu afin d'atteindre une meilleure segmentation

Travaux relatifs à [Germond, 1999]

Principe Dans [Germond, 1999] propose différentes coopérations définies entre un modèle déformable, un système multi-agents et un détecteur de contour. Le modèle déformable est utilisé pour extraire l'encéphale. On peut distinguer quatre phases essentielles dans le fonctionnement de système. La première concerne la construction du modèle déformable, la seconde la segmentation en régions, la troisième la segmentation de contours et la quatrième le processus de rétroaction. Figure 3.17.

Résultat Plusieurs agents sont placés sur l'encéphale de telle façon que les agents de matière grise soient placés en périphérie, et ceux de la matière

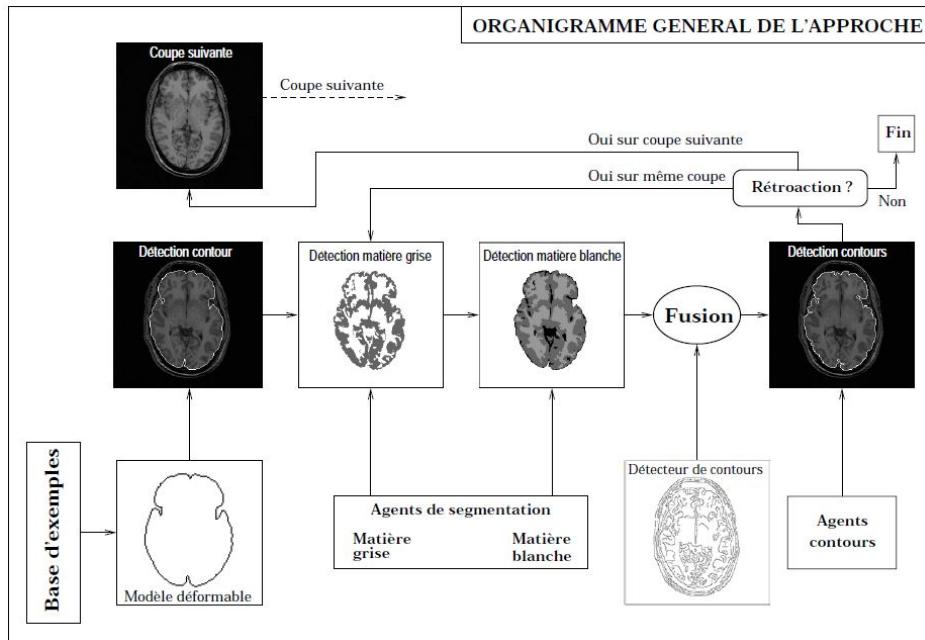


FIGURE 3.17 – Principe général de la segmentation réalisée par l’approche proposée

blanche à l’intérieur (voir la figure 3.18).

3.3.3 Coopération séquentielle

Une des techniques de segmentation (région ou contour) est exécutée, son résultat est exploité par la deuxième technique pour améliorer les critères ou les paramètres de la segmentation ;

Travaux relatifs à [Duchesnay, 2001]

Principe [Duchesnay, 2001], dans le cadre de sa thèse, a réalisé une plateforme multi-agents destinée à la segmentation d’images. Comme pré-classification, l’auteur utilise la décomposition en « split and merge », qui est une segmentation qui ne donne que les zones homogènes dans l’image. Un agent est ensuite placé dans chaque zone homogène et le processus de la segmentation basé sur les SMA débute. Ils adoptent les pyramides irrégulières comme élément organisationnel de la population d’agents (Figure 3.19) ; la progression du processus se faisant de la base de la pyramide jusqu’au sommet. Dans ces travaux, la segmentation est appliquée à des images de

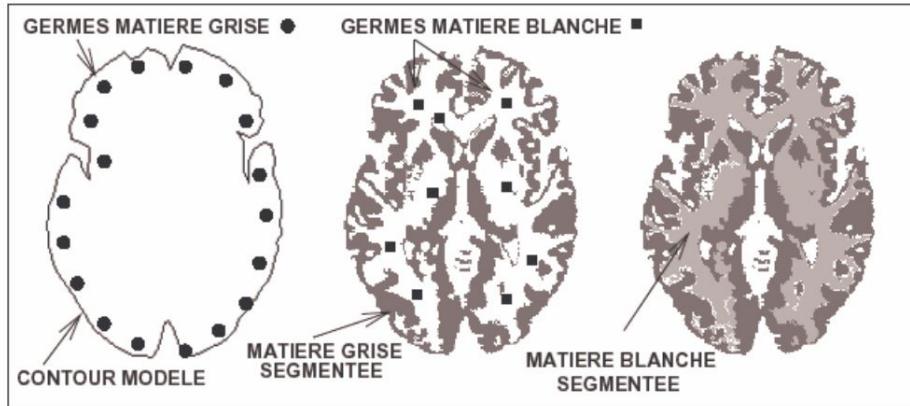


FIGURE 3.18 – Principe de la génération successive des agents de segmentation à partir du contour du cerveau, pour la matière grise, puis la matière blanche

scanner du sein.

Résulat Dans la figure 3.20 on a :

- (a) : image originale : la tumeur se développe dans la partie lobulaire.
- (b) : représente la pyramide avec pour chaque niveau le nombre d'agents utilisés et le temps de calcul en ms.
- (c), (d) : on visualise la répartition initiale des agents régions et contours de la base de la pyramide

3.3.4 Discussion

Le mélange des deux approches par régions et par contours, où la détection de contours peut être utilisée pour initialiser les algorithmes de détection de régions. La méthode est judicieuse, elle combine les avantages de chacune prise séparément : la précision et la rapidité d'une segmentation en contours, et la fermeture des frontières et la densité de l'information extraite, d'une segmentation en régions.

3.4 Synthèse

Nous avons différentes approches pour la segmentation en contours ou en régions. Nous avons conclu qu'il n'y a pas de détecteur parfait ; il faut essayer de formaliser l'objectif final du traitement où la recherche des contours

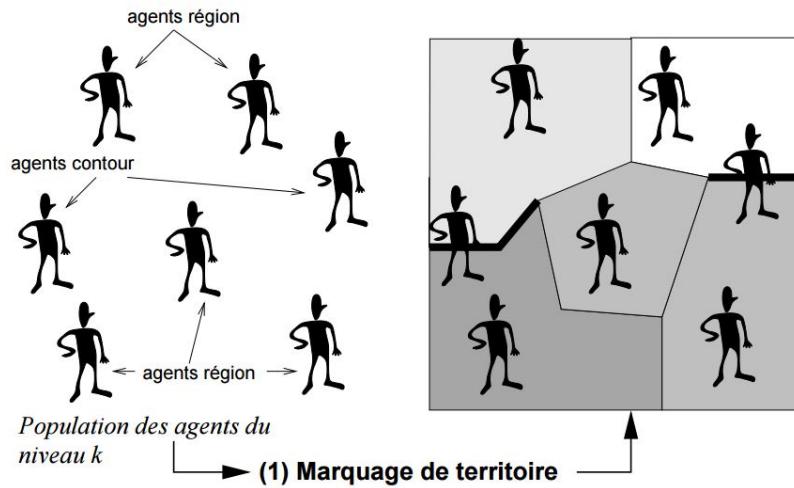


FIGURE 3.19 – Vision s hématique du comportement de marquage du territoire

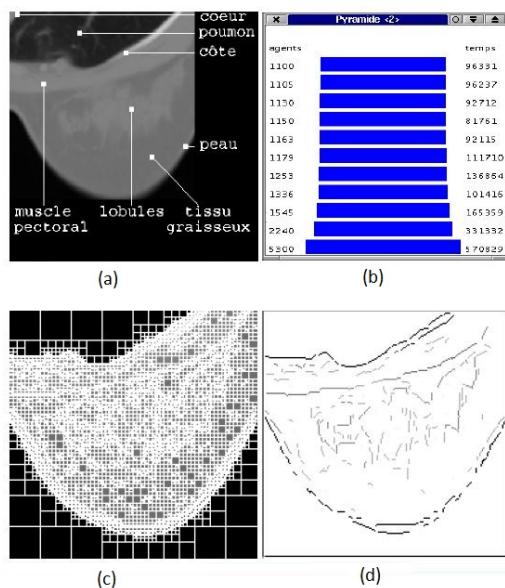


FIGURE 3.20 – (a) Image originale, (b) Pyramide, (c) Division quaternaire, (d) Segments de contours

n'est qu'une étape intermédiaire et, relativement à cet objectif, évaluer les performances d'un détecteur :

- proportion de détections correctes,
- de détections manquées,
- et de fausses détections.

Pour ce stage, notre objectif final est d'appliquer la solution proposée sur des échantillons multi-capteurs d'environnement marin, comme nous avons vue dans [Benzinou et al., 2004] le type de contour étudié est un peu spécifique, d'où une approche régions ne sera pas adaptable pour notre cas malgré ces nombreux avantages, une approche hybride est inutile aussi, il nous reste l'approche contours qui semble le mieux adoptable pour ces types de contours.

Conclusion

Dans ce chapitre, Nous avons exploré les principales approches de segmentation multi-agents en les classant en trois catégories : les approches région, les approches contour et le approches coopératives régions-contours. Pour les approches contours, nous avons cité les travaux des SMA qui sont basées sur le calcul du gradient et qui sont basées sur les calculs analytiques. Pour la catégorie région, on a exposé les travaux de segmentation par croissance de région et par algorithme k-means. Nous avons aussi parlé des approches SMA hybrides qui sont basés la coopération des deux approches régions et contours. Enfin, nous avons conclure que notre approche sera une approche multi-agents par approche contours vue qu'on vise adopter notre approche pour un type de contour spécifique qui est les otolithes des poissons pour calculer leurs ages.

Chapitre 4

Approche proposée pour la détection de contours

Introduction

Nous avons vu dans le chapitre précédent, qu'il existe plusieurs méthodes pour la segmentation par contour et par région. Bien que chaque approche ait ses avantages, elle présente également des inconvénients. Suite à cette état de l'art que nous avons abordé, nous avons décidé d'adopter une solution de segmentation par approche contour. De ce fait, nous aborderons dans le présent chapitre les différents éléments de l'approche de segmentation proposée. Ensuite, ensuite nous allons présenter les besoins fonctionnels et non fonctionnels. Enfin nous détaillons l'approche proposée d'un point de vue architecture logiciel.

4.1 Les types d'agents

Nous détaillons dans cette section les différents éléments de l'approche de segmentation proposée. Notre système est composé de trois types d'agents. Comme il est indiqué dans la figure 4.1, nous aurons un agent "contrôleur", plusieurs agents « de détection » et plusieurs agents "de suivi". Ces types d'agents partagent certaines caractéristiques mais se différencient par d'autres.

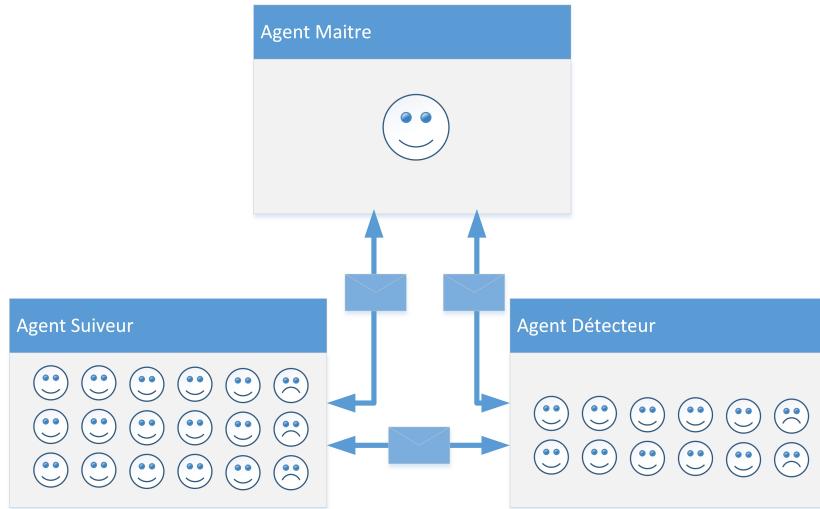


FIGURE 4.1 – Architetture de l'approche proposée

4.1.1 Agent contrôleur du système « Maître »

On peut définir le contrôleur du système comme étant un super agent, qui se charge de lancer l'exécution des autres agents. Il initialise les agents de détections (voir la figure 4.2) en lui donnant les positions et directions initiales, et reçoit les résultats de segmentation des différents agents de suivis pour afficher le résultat final à l'utilisateur.

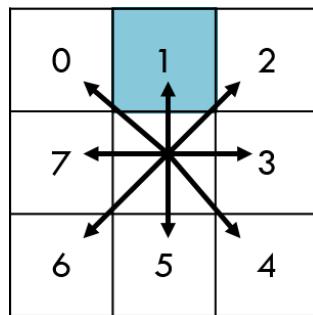


FIGURE 4.2 – Les huit directions

4.1.2 Agent de détection « Détecteur »

Cet agent possède un comportement d'exploration consiste à rechercher un contour non visité dans l'image. Pour cela, les agents se déplacent selon les directions indiquées par l'agent de contrôleur de système, par exemple si

la direction donnée est 4 avec un nombre de capteurs égal à 3 les capteurs de l'agent seront comme il est indiqué dans la figure 4.3 (a), et si le nombre des capteurs est égal à 5, les capteurs seront comme il est indiqué dans la même figure (b).

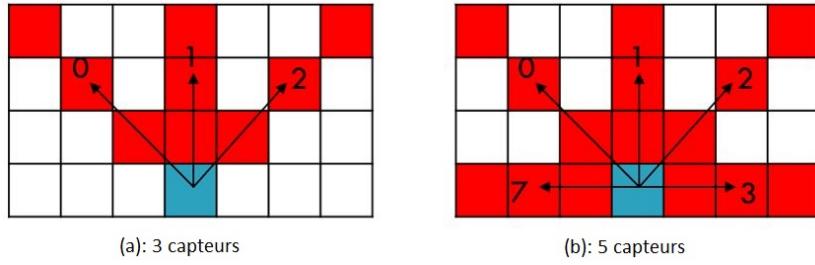


FIGURE 4.3 – Forme des capteurs avec une direction donnée

L'agent vérifie s'il perçoit un contour non exploré par d'autres agents. Si c'est le cas, l'agent détermine les différentes directions possibles pour suivre ce contour (figure 4.4). Il crée par la suite autant d'agents de suivi de contour que de directions détectées. Ces agents créés suivront le contour détecté selon une certaine direction (figure 4.2). Après avoir détecté un contour, l'agent exploration continue la procédure de détection.

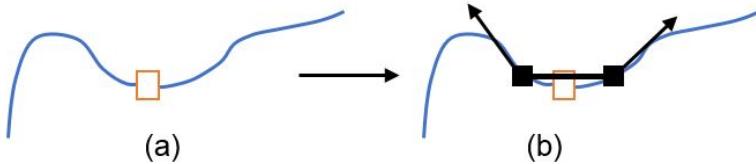


FIGURE 4.4 – Dans (a), l'agent de détection trouve un contour. Il détermine deux directions possibles pour suivre le contour. Dans (b), le même agent crée alors deux agents de suivi de contour

4.1.3 Agent de suivi « Suiveur »

Les agents de suivi de contour sont créés par les agents de détection qui ont trouvé un contour non visité. L'agent de suivi de contour est lié à un autre agent de suivi qui explore le même contour dans la direction opposée (figure 4.4). Les agents, initialement présentés dans [Bellet, 1998] se déplacent dans leur environnement, qui est ici l'image à traiter. Ils peuvent détecter

localement les niveaux de gris de l'image leur permettant de s'orienter pour suivre la direction d'un contour. Il existe un modèle unique d'agents pour la recherche de contours de type saut d'amplitude. Ils sont munis de capteurs en barre, comme il est indiqué dans la figure 4.5, sur les niveaux de gris de l'image (en nombre supérieur ou égal à trois), et d'un capteur unitaire leur permettant de lire sur une image commune à tous les agents afin de savoir si un contour a déjà été détecté sur le pixel où ils se situent.

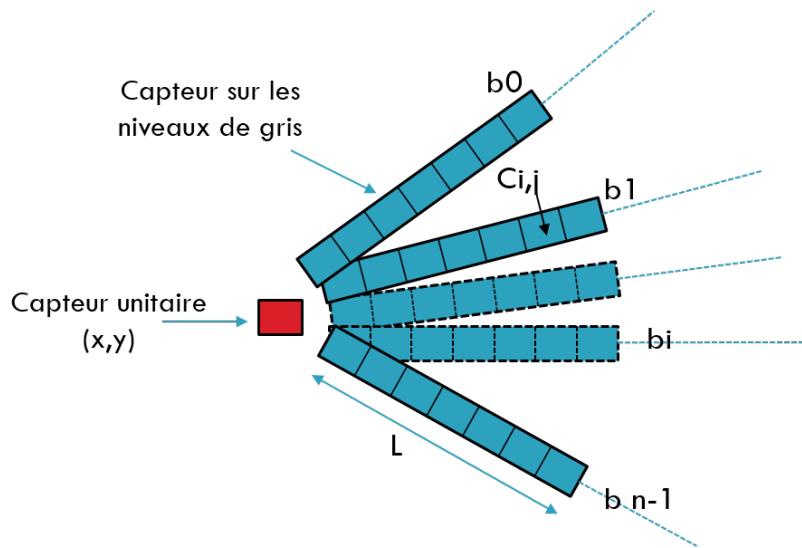


FIGURE 4.5 – Capteurs d'un agent

La figure 4.5 fait apparaître différentes variables :

- $i \in [0, n-1]$ est l'indice d'une barre,
- n est le nombre de capteurs
- L est la longueur des barres en pixels
- $j \in [0, L-1]$ est l'indice d'un capteur unitaire dans une barre
- $C_{i,j}$ est la valeur lue par un capteur unitaire j d'une barre i

4.2 Comportement des agents

Les agents sont de type réactif (voir section 2.3.5), ils réagissent aux stimuli perçus par l'intermédiaire de leurs capteurs. Leur comportement peut être résumé comme suit :

- L'agent maître crée un nombre fini des agents de détections ;
- ces agents d'explorations sont d'abord placés aléatoirement sur l'image, puis ils se déplacent en suivant la direction affecté par l'agent maître

(voir la figure 4.2) ;

- après la détection d'un contour probable, l'agent de détection crée deux agents de suivi pour suivre le contour détecté ;
- les agents de suivi se déplacent en suivant la direction déterminée par la direction médiane entre les deux capteurs en barre présentant un gradient d'intensité maximum, si ce gradient est supérieur à un seuil **S** déterminé. Sinon ils gardent leur direction courante.

Une présentation plus détaillée de leur comportement peut être illustrée par un diagramme d'états-transition dans la figure 4.6.

Les transitions entre les différents états sont présentés dans la figure 4.6 et sont détaillés ci-dessous :

4.2.1 Comportement de supervision

- **État Créeur** : lors de sa création, l'agent de supervision reçoit la matrice d'image à segmenter, il initialise un nombre fini des agents de détections avec des positions et une direction aléatoire sur l'image.
- **État Superviseur** : après initialiser tous les agents sur l'image, l'agent maître passe à un état d'écoute des messages venant des agents de suivi afin d'afficher le résultat de segmentation à l'utilisateur.

4.2.2 Comportement d'exploration

- **État Init** : l'agent est initialisé, on lui affecte une position et une direction aléatoires dans l'image.
- **État Renifleur** : cet état consiste à rechercher un contour non visité dans l'image. Pour cela, les agents se déplacent selon les directions indiquées muni d'un capteur (voir la figure 4.7) pour détecter le changement d'intensité sur l'image. L'agent vérifie s'il perçoit un contour potentiel non exploré par d'autres agents. Si c'est le cas il passe à l'état **orienteur**.
- **État orienteur** : l'agent détermine les différentes directions possibles pour suivre ce contour (figure 4.4), ensuite il passe à l'état **créateur**.
- **État créateur** : après la détection des différentes directions possible pour le contour détecté, l'agent crée autant d'agents de suivi de contour que de directions détectées. Ces agents créés suivront le contour détecté selon une certaine direction (figure 4.2). Après avoir créé les agents de suivi, l'agent de détection passe à nouveau à l'état **renifleur** pour explorer à nouveau des contours.

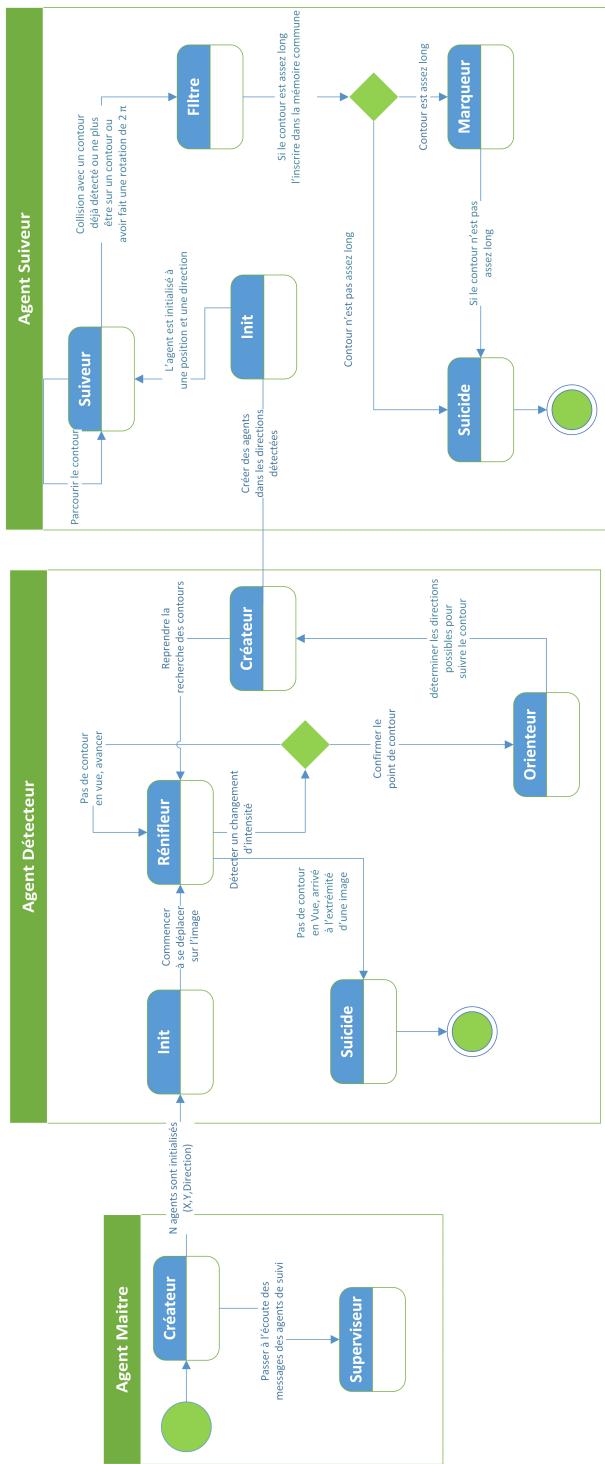


FIGURE 4.6 – Diagramme d'états-transitions décrivant le comportement des agents

- **État suicide** : finalement, si l'exploration de l'agent n'aboutit à aucun contour au bout d'un certain temps ou bien il arrive à l'extrémité de l'image où il n'y pas d'autres pixels à explorer, l'agent disparaît.

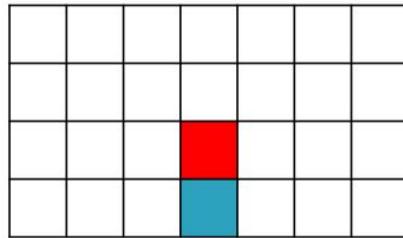


FIGURE 4.7 – Forme de capteur d'agent détecteur

4.2.3 Comportement de suivi de contour

- **État Init** : l'agent est initialisé, on lui affecte une position et une direction aléatoires dans l'image, une valeur pour les paramètres présentés dans la figure 4.5 (L , n), ainsi que le seuil S et la longueur minimale d'un contour pour qu'il soit validé. Il passe alors à l'état de suiveur.
- **État suiveur** : comme dans la figure 4.8, l'agent avance tout droit tant que la différence de niveau de gris maximum entre deux barres b_i et b_{i+1} est inférieure au seuil S . Ce gradient est normalisé en calculant la différence entre la moyenne des intensités b_i de deux capteurs contigus. Si cette différence dépasse le seuil S , l'agent passe à l'état filtre. Le gradient maximum est calculé à partir des intensités lues par les capteurs comme suit :

$$b_i = \sum_{j=0}^L -1C_{i,j} \quad (4.1)$$

$$\text{gradient} = b_{i+1} - b_i \quad (4.2)$$

$$\text{gradient}_{\max} = \max(\text{gradient}_i) \quad (4.3)$$

- **État filtre** : l'agent va évaluer la longueur du contour parcouru ; si elle est supérieure à une valeur seuil, l'agent passe à l'état **Marqueur** où le contour sera inscrit dans la mémoire commune, sinon l'agent passe à l'état de suicide et sa tâche est terminée.

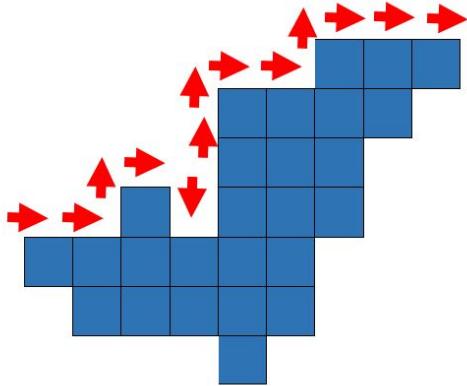


FIGURE 4.8 – L’agent se déplace sur les points contour

- **État marqueur** : l’agent inscrit le contour mémorisé en mémoire commune en y traçant un polynôme reliant les pixels consécutifs composant sa trajectoire, en utilisant un niveau de gris spécifique qui sera reconnu par les autres agents pour détecter la présence d’un contour.
- **État suicide** : finalement, si le contour suivie n’était pas assez long ou l’agent a marqué le contour suivie, l’agent disparaît.

4.3 Spécification et conception

Dans cette section, nous présentons les fonctionnalités que doit assurer notre système. Chaque fonctionnalité est décrite afin de mieux clarifier les besoins.

4.3.1 Besoins fonctionnels

Dans ce paragraphe nous listons les fonctionnalités de notre système. Ces fonctionnalités sont les suivantes :

- Segmentation : le système doit présenter clairement le résultat de segmentation où il devra repérer les contours dans les images d’entrée
- Évaluation de résultat de traitement : pour étudier l’influence des différents paramètres sur le comportement des agents, nous devons évaluer la détection effectuée.
- ouverture/affichage/sauvegarde des images : Notre système manipule des fichiers images et en particulier des images de format TIFF. Cette

fonctionnalité est imposée par une bibliothèque utilisée pour le développement de notre approche. De même le système doit être capable de sauvegarder une image après une série de traitements pour pouvoir comparer entre l'image d'entrée et l'image de sortie.

4.3.2 Besoins non fonctionnels

- **Temps de réponse** : le temps de calcul a toujours été un défi pour les développeurs dans le domaine du traitement d'images. Un traitement sur une image est souvent synonyme de parcours de tous les pixels de l'image. De ce fait, il est clair que le temps de réponse est directement lié à la masse de données traitées, à savoir la taille de l'image et sa résolution, mais avec notre solution multi-agents, nous avons la possibilité de percevoir une partie de la segmentation en temps réel. Le système doit afficher l'image résultante d'un traitement dans un délai raisonnable (l'ordre de secondes mais ce délai dépend de la taille de l'image traitée).
- **Robustesse** : l'application doit prendre en charge les entrées inattendues ou incorrectes et renvoyer des erreurs correctement gérées à l'utilisateur.
- **Ré-utilisabilité** : le code doit être modulaire, et donc réutilisable et améliorable. Faciliter l'ajout de nouveaux formats d'image ou l'implémentation de nouveaux algorithmes concernant le suivi de contours.

4.3.3 Diagramme de classes

Dans cette partie nous définirons en détails le diagramme de classes de notre système, ensuite nous essayerons de présenter quelques scénarios du système. La figure 4.9 illustre la vue conceptuelle classique du système via son diagramme de classes. Ainsi, la représentation d'un tel diagramme permet de bien comprendre la composition.

Description des classes de système

La plate-forme multi-agents de JADE a été réalisée suivant le paradigme Objet qui consiste à définir et à assembler des objets. Chaque objet représente un concept, une idée ou une entité physique. Dans un modèle Orienté Objet, les entités sont définies à partir de classes et de relations entre ces classes. Chaque classe est caractérisée par des attributs et des comportements. Nous

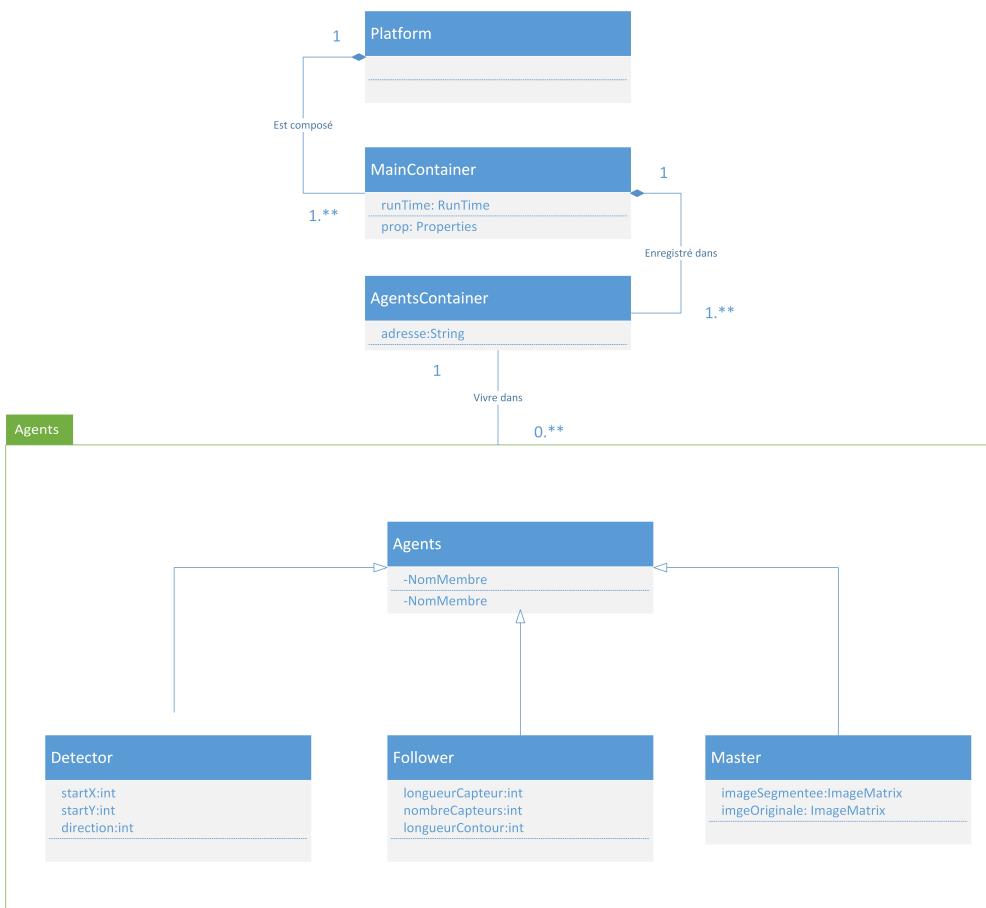


FIGURE 4.9 – Diagramme de classes de système

désignons une relation d'héritage entre les classes lorsqu'une classe est spécialisée en sous-classes et une relation de composition ou d'agrégation lorsqu'une classe contient des instances d'autres classes. L'intérêt de cette modélisation est de favoriser la conception de modules génériques faciles à étendre pour des besoins spécifiques.

- Un ensemble de conteneurs constituent une **Plateforme**.
- Chaque plateforme doit contenir un conteneur spécial appelé **Main-Container** et tous les autres conteneurs s'enregistrent auprès de celui-là dès leur lancement.
- Chaque instance du JADE est appelée conteneur **Container**, et peut contenir plusieurs agents.
- Les classes **Master**, **Follower** et **Detector** sont des agents qui héritent des propriétés de la classe Agent. (le cycle de vie d'un agent JADE et son comportement sont plus détaillés dans le chapitre 2, voir la figure 2.13).

Conclusion

Nous avons présenté dans ce chapitre les différents types d'agents qui composent notre SMA. Ensuite, nous avons détaillé les comportements de chaque agent ainsi que l'interaction entre eux, et finalement nous avons effectué une analyse de besoins et l'architecture logiciel de notre système. Dans le chapitre suivant, nous allons détailler l'implémentation de notre approche et les résultats obtenus.

Chapitre 5

Réalisation et expérimentation

Introduction

Dans ce chapitre nous allons exposer le travail achevé. Nous présenterons en premier lieu l'environnement logiciel dans lequel notre application a été développée en indiquant les technologies utilisées. Ensuite, nous allons évaluer la détection effectuée. Nous clôturons ce chapitre par un aperçu sur l'état d'avancement du projet et sur quelques captures d'écran traduisant le déroulement de l'application.

5.1 Environnement logiciel

Dans cette partie, nous présentons l'environnement logiciel qui a accompagné la réalisation du projet. Nous citerons les différents outils utilisées pour le développement de système, pour l'évaluation de l'approche et pour la rédaction de rapport.

5.1.1 Latex

L^AT_EXest un langage de description de document, permettant de créer des documents écrits de grande qualité : principalement livres et articles, mais aussi, courriers, présentations projetées... (figure 5.1)

LATEX

FIGURE 5.1 – Logo Latex

5.1.2 Visio

Microsoft Visio est un logiciel de diagrammes et de synoptique, produit initialement développé par Visio Corporation et racheté par Microsoft en 2000. (figure 5.2).



FIGURE 5.2 – Logo Microsoft Visio

5.1.3 Matlab

Pour l'évaluation de notre approche et effectuer des prétraitements sur les images, nous avons utilisé Matlab qui est un environnement logiciel pour la génération et l'exécution de scripts pour l'analyse et la simulation de données mathématiques et techniques. (figure 5.3).



FIGURE 5.3 – Logo Matlab

5.1.4 Eclipse

Eclipse IDE est un environnement de développement intégré, libre et extensible, polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions. La spécificité d'Eclipse IDE vient du fait de

son architecture totalement développée autour de la notion de plugin. (figure 5.4).



FIGURE 5.4 – Logo Eclipse

5.1.5 JADE

Il existe plusieurs plateformes offrant un cadre d'exécution aux agents. Pour notre travail, il a fallu trouver une plateforme qui permet de développer des agents en JAVA , et qui offre un certain nombre de services. La plateforme JADE nous a semblé la plus adaptée. Les principales caractéristiques de cette plate-forme ont été présentées au chapitre 3, section 2.4. (figure 5.5).



FIGURE 5.5 – Logo JADE

5.2 Évaluation de la détection

Dans cette section, nous allons nous intéresser à l'aspect de performances des agents en tant que détecteurs de contour, notamment en fonction des valeurs des différents paramètres. Pour cela nous utiliserons des images de synthèse comprenant des contours de type saut d'amplitude noyés dans un bruit gaussien, inspirées de celles qui sont proposées comme images tests dans l'ouvrage [Guillaud, 2000] pour comparer des extracteurs de contours. Ainsi l'image de la figure 5.6 (a), de taille 128*128 pixels, comprend un carré de niveau de gris 26 sur un fond de niveau de gris 20, et un bruit gaussien d'écart-type 4 y a été ajouté en Figure 5.6 (b).

5.2.1 Paramètres gérant le comportement des agents

Pour étudier l'influence des différents paramètres sur le comportement des agents, nous avons évalué la détection effectuée en s'inspirant des critères de

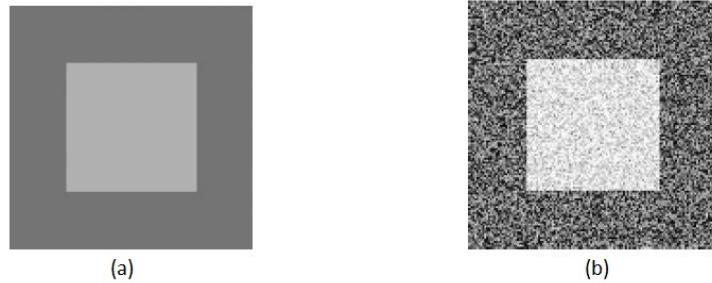


FIGURE 5.6 – Image de synthèse utilisée pour l'évaluation

[Guillaud, 2000]. Nous avons ainsi calculé deux coefficients permettant de quantifier respectivement la sur-détection et la sous-détection définis dans les équations suivantes :

$$C1 = 100 * \frac{\text{nombre_de_pixels_détectés_n'étant_pas_des_points_contours}}{\text{nombre_de_pixels_non_contours}} \quad (5.1)$$

$$C2 = 100 * \frac{\text{nombre_de_pixels_contours_non_détectés}}{\text{nombre_de_pixels_contours}} \quad (5.2)$$

- **C1 : Erreur de sur-segmentation** : Les algorithmes de segmentation peuvent ne pas affecter des pixels à un contour ou à une région correctement.
- **C2 Erreur de sous-segmentation** : Lorsque la différence de niveau de gris (ou de couleur) entre deux régions voisines n'est pas très important, leur gradient est très petit, et le contour peut ne pas être détecté. Par conséquent, certains pixels appartenant à un contour ne seront pas pris, on parle alors de sous segmentation

Dans les paragraphes suivants nous calculerons les valeurs obtenues pour ces critères selon les valeurs des différents paramètres. Nous présenterons également le résultat de manière visuelle en superposant sur une image les points contours détectés (en bleu) sur la position réelle du contour (carré en rouge), et la superposition des deux couleurs (contour totalement détecté) est en couleur violet, comme en figure 5.7.

Paramètre L

Plus L est grand moins les agents seront sensibles au bruit, cependant une valeur importante de L augmente le temps de calcul, et dans le cas de

contours très irréguliers une trop grande valeur de L peut entraîner une erreur de jugement chez les agents qui pourraient être influencés par des différences de niveau de gris ne reflétant pas l'existence du contour que les agents sont en train de suivre.

Nous avons également testé l'influence du paramètre L sur le traitement de l'image présentée dans la figure 5.7.

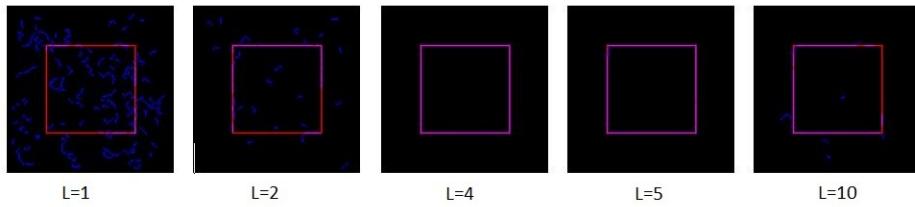


FIGURE 5.7 – Contours détectés pour différentes valeurs de L

Comme l'illustrent la figure 5.7 et la courbe dans la figure 5.8 pour $L = 1$ on obtient une importante sur-détection, car les agents ayant des capteurs unitaires peuvent prendre un gradient dû au bruit pour la position d'un contour. Ensuite quand L augmente sur-détection et sous-détection diminuent, les agents devenant moins sensibles au bruit. Mais si L devient trop grand (par exemple $L = 10$) la sous-détection augmente et les contours sont moins précis. Il semble donc nécessaire de réaliser un compromis entre une valeur faible de L permettant de mieux localiser les contours, mais rendant la détection sensible au bruit, et une valeur plus élevée de L permettant de limiter la sensibilité au bruit, par un lissage qui risque toutefois de nuire à une bonne localisation des contours.

Nombre de capteurs N

Plus le nombre de capteurs est grand, plus la position du contour pourra être évaluée précisément, mais aussi plus le temps de calcul sera long. Ce nombre n est également limité par les dimensions d'un agent, comme le montre l'exemple de la figure 5.9 ; les images étant numériques, le nombre d'inclinaisons possibles pour un capteur entre l'horizontale et la verticale est limité. n doit bien sûr être strictement supérieur à 2, sinon l'agent ne pourra pas effectuer de comparaison entre différents gradients possibles.

Nous avons testé l'influence du nombre de capteurs sur la détection des contours de l'image figure 5.6. La figure 5.11 représente les résultats obtenus

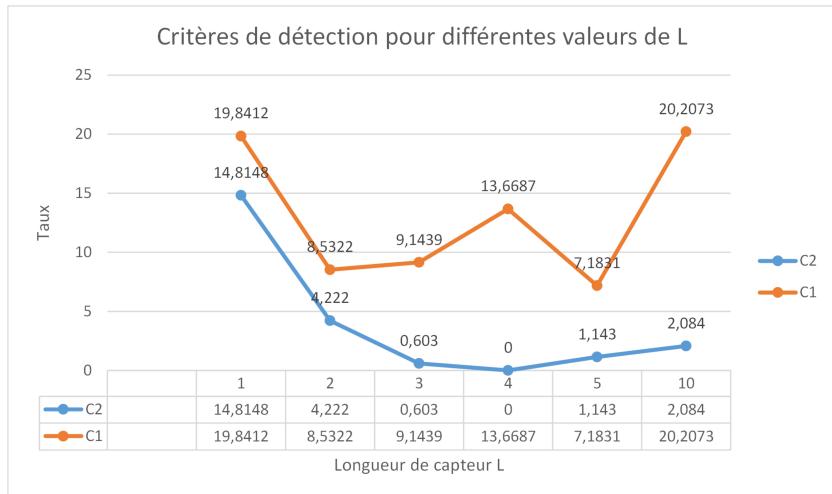


FIGURE 5.8 – Critères de détection pour différentes valeurs de L

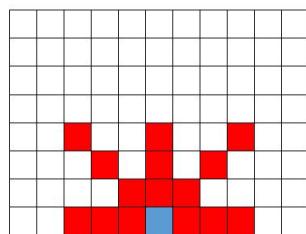


FIGURE 5.9 – Agent avec 5 capteurs de longueur L = 4

avec différentes valeurs de n . la figure 5.10 indique les critères de détection obtenus pour ces différentes images.

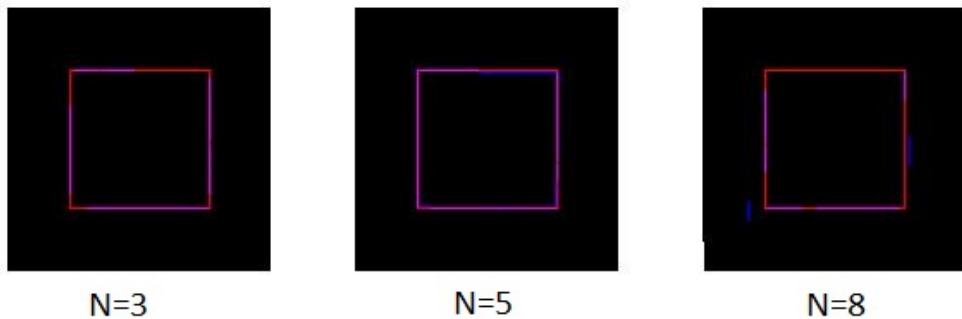


FIGURE 5.10 – Contours détectés pour différentes valeurs de N

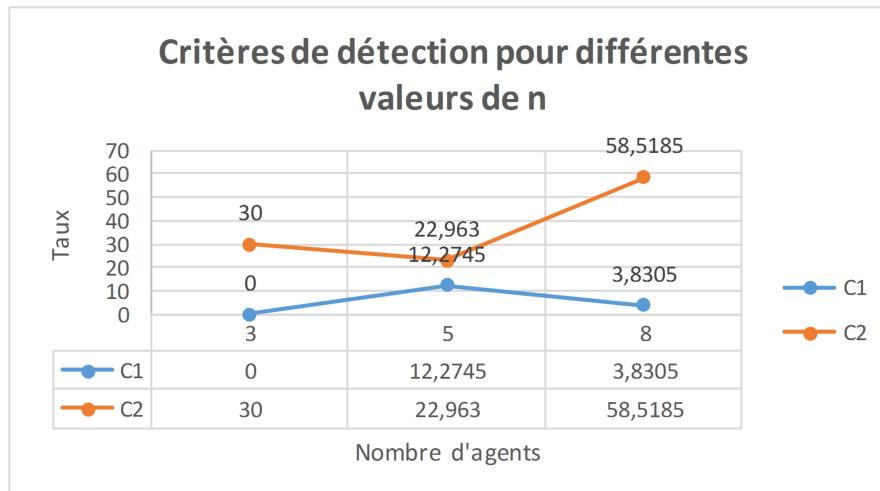


FIGURE 5.11 – Critères de détection pour différentes valeurs de N

Pour les raisons données dans ce paragraphe, augmenter le nombre de capteurs n'améliore pas les performances des agents, donc on fixera le nombre des capteurs à 5 pour limiter le nombre de calculs tout en effectuant un échantillonnage suffisant de l'image.

Seuil pour le gradient S

Les agents considèrent qu'ils se trouvent en présence d'un contour si le gradient maximum entre deux de leurs capteurs est supérieur à un seuil S .

Donc si ce seuil est trop élevé un agent peut manquer un contour peu marqué et s'il est trop faible l'agent sera plus sensible au bruit. Ainsi sur la figure 5.12 pour $S=1$ et $S=2$ on observe une forte sur-détection de points contours correspondant à du bruit. En effet l'écart-type du bruit étant égal à 2 les agents ont pu trouver un gradient supérieur à 1 ou 2 ne correspondant pas à un contour réel. Ensuite pour $S = 3$ ou $S = 4$ la détection de contours dus au bruit a été évitée, mais pour des valeurs supérieures de S on a une importante sous-détection du contour réel. Pour l'image en figure 5.12 le seuil $S = 7$ est supérieur au gradient du contour réel (égal à 6) donc la probabilité que les agents détectent un contour est très faible.

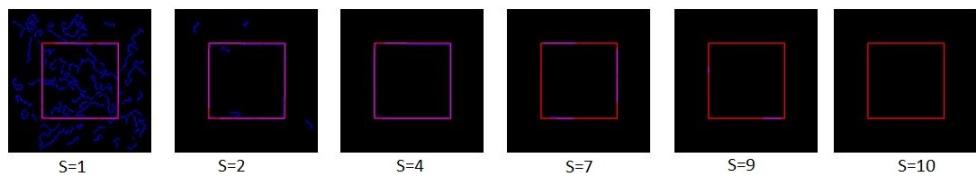


FIGURE 5.12 – Contours détectés pour différentes valeurs de S

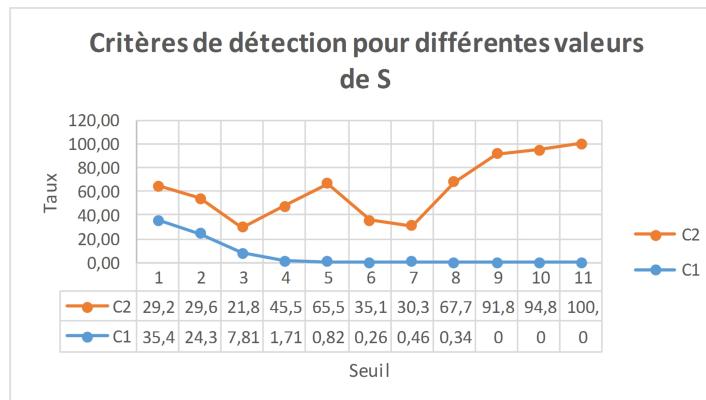


FIGURE 5.13 – Critères de détection pour différentes valeurs de S

L'influence du paramètre S semble donc être primordiale sur le résultat de la détection. S doit être ajusté pour que sa valeur soit assez importante pour limiter l'influence du bruit, et suffisamment inférieure au gradient des contours à rechercher pour éviter une sous-détection. Pour satisfaire ce compromis dans le cas présent, nous avons pris $S = 4$ afin de traiter l'image de la figure 5.6. Le saut d'amplitude est en effet de 6 niveaux de gris, et l'écart-type du bruit égal à 2.

5.2.2 Influence du bruit

Nous avons ajouté un bruit gaussien d'écart type variable à l'image originale du carré, présenté sur la figure 5.6. Jusqu'à un écart-type $\delta = 3$ on obtient une détection à peu près correcte du carré, avec une faible sur-détection (figure 5.14). Au-delà de cette valeur on a une forte sur-détection de points dus au bruit et le contour en lui-même est mal détecté. Une plus grande valeur pour le paramètre L permet alors de limiter la sur-détection, mais les contours détectés sont peu précis. Si on définit le rapport signal à bruit comme étant 10 fois le logarithme de la hauteur du contour divisé par l'écart-type du bruit, on obtient pour $\delta = 4$ un rapport signal à bruit de 1.76 dB, ce qui est assez faible. On peut donc dire que les agents montrent une robustesse satisfaisante face au bruit.

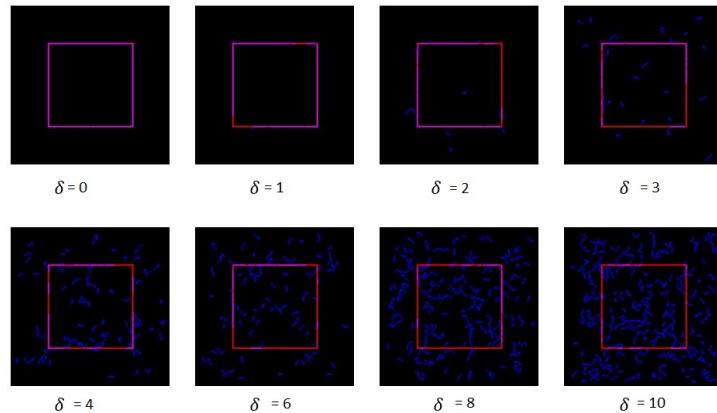


FIGURE 5.14 – Contours détectés pour différents niveaux de bruit

5.2.3 Influence du nombre d'agents

Les agents peuvent être placés à plusieurs dans une image, afin d'en explorer simultanément différentes zones. Nous nous sommes intéressés à l'influence du nombre d'agents Nb sur la qualité du traitement effectué, pour un même temps de calcul (figure 5.16).

5.3 Résultats

Pour l'aspect graphique, nous avons utilisé la bibliothèque Swing qui fait partie de jade, pour la création d'interfaces graphiques. La figure 5.17 illustre

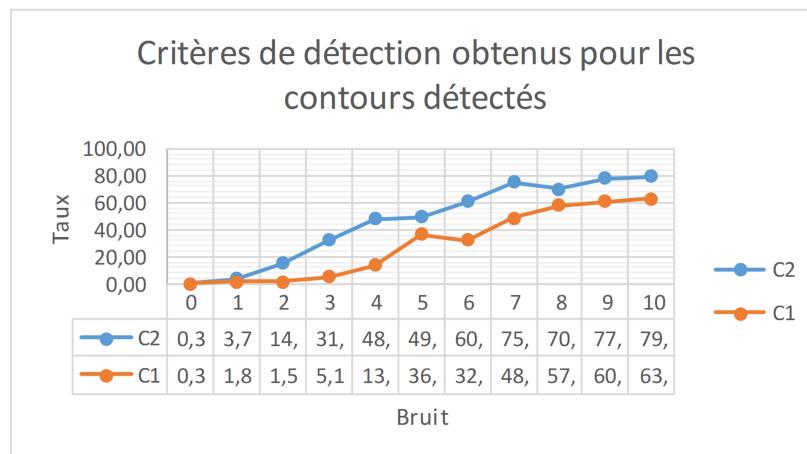


FIGURE 5.15 – Critères de détection obtenus pour les contours détectés

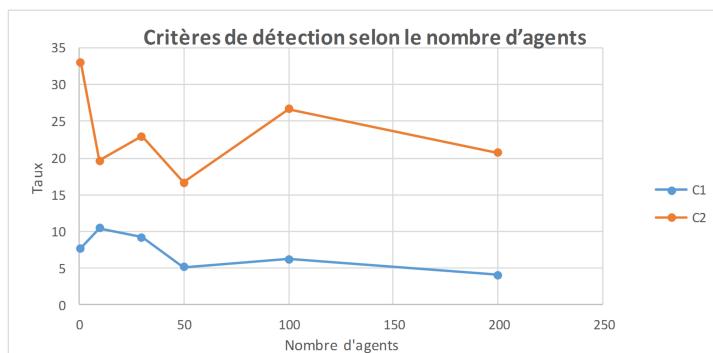


FIGURE 5.16 – Critères de détection selon le nombre d'agents

l'interface de notre système, à gauche l'image originale sera affichée dans cette zone après que l'utilisateur charge son fichier et à droite on affiche le résultat de traitement.

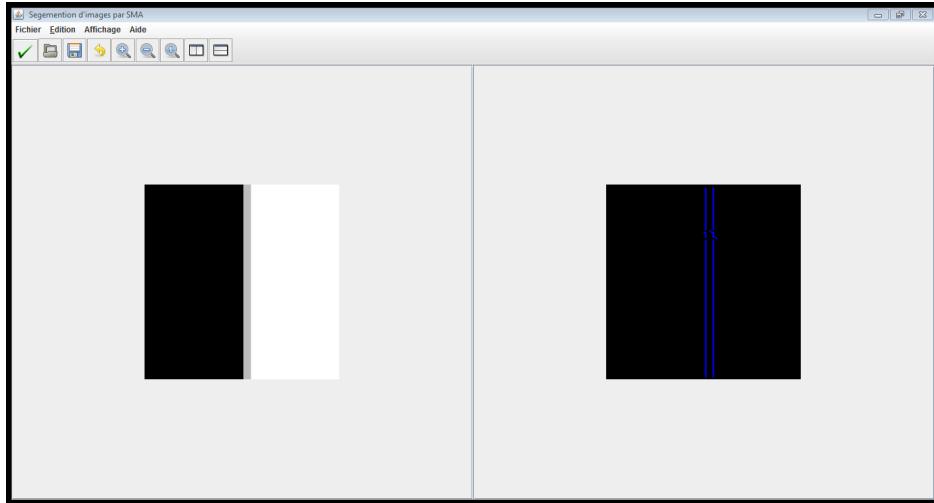


FIGURE 5.17 – Interface principale

Détection de contours en anneaux

Nous avons testé le système sur plusieurs images. La détection de contours à travers notre SMA s'avère efficace. Néanmoins, le système ne parvient pas à détecter tous les contours, dans la figure 5.18 et la figure 5.19, le système montre une robustesse face au contour en anneaux.

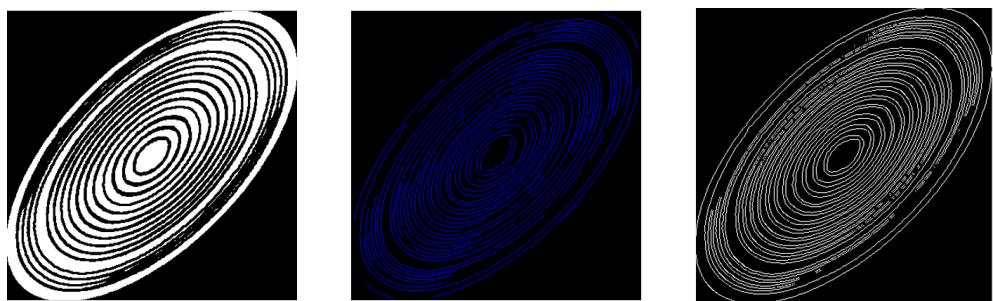


FIGURE 5.18 – Résultat de segmentation sur des contours en anneaux

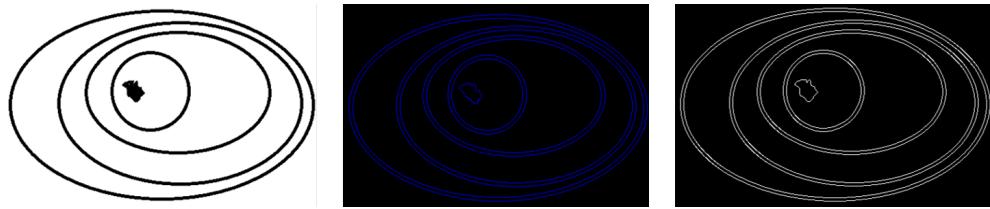


FIGURE 5.19 – Résultat de segmentation sur des contours en anneaux

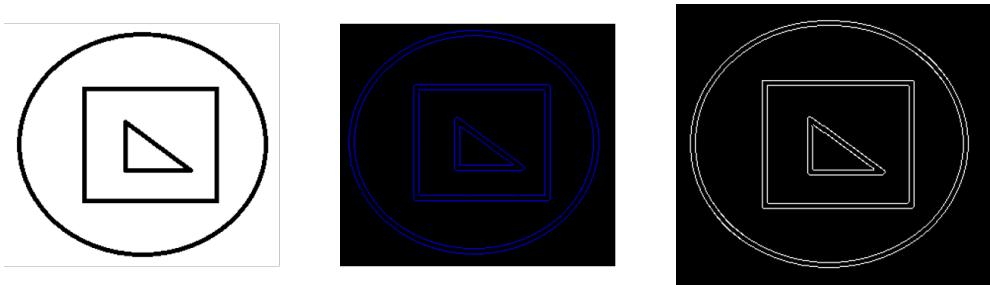


FIGURE 5.20 – Résultat de segmentation pour des contours en formes géométriques

Détection de contours en forme géométrique

Notre système détecte proprement les formes géométriques. (voir la figure 5.20).

Conclusion

Dans ce chapitre, nous avons présenté l'environnement de développement de notre approche. Ensuite, nous avons analysé l'influence des différents paramètres essentiels à l'agent qui peuvent moduler son comportement. Le choix de la longueur L des capteurs et du seuil S pour le gradient semble être le plus influent sur le résultat de la détection. Une valeur élevée pour ces deux paramètres permet de limiter la sensibilité des agents au bruit, mais risque également de nuire à la détection des contours de faible amplitude. Un compromis doit alors être trouvé entre ces deux positions. Des paramètres extérieurs aux agents peuvent également influer sur leur action. Nous étudierons ainsi la réaction des agents selon leur nombre face à différents niveaux de bruit.

Conclusion générale et perspectives

5.4 Conclusion générale

La segmentation d'images est une étape cruciale dans tout processus d'analyse d'images. Elle consiste à préparer l'image afin de la rendre plus exploitable par un processus automatique telle que l'interprétation. Il existe deux grandes approches purement locales. L'approche contour consiste à localiser les frontières des régions, elle est basée sur la notion de dissimilarité. Parmi ces point fort : sa simplicité et sa rapidité mais elle donne parfois des contours ouverts. L'approche région consiste à réunir les pixels connexes dans une région homogène, elle est basé sur la notion de similarité. Elle est simple et rapide mais l'utilisation uniquement des informations locales donne parfois de mauvais résultats (sous-segmentation, sur-segmentation). La coopération de ces deux approches qui sont par nature dual améliore le résultats de la segmentation.

L'objectif de notre travail a concerné la conception et le développement d'un système multi-agent pour la segmentation des images. Afin de mener à bien ce travail, nous avons mené une étude de l'art sur la question. Cette étude nous a aidé à inspirer une architecture pour notre approches et à choisir une plate-forme multi-agents pour nous aider à simplifier la tâche de développement du système et profiter des différents outils offerts par la plate-forme. Ensuite, nous avons réalisé l'étape de conception qui a consisté à dégager l'architecture générale et détaillée de notre système. Enfin, nous avons analysé les propriétés du système proposé qui est dédié à l'extraction de contours de type saut d'amplitude, en étudiant l'importance des différents paramètres influant sur le comportement des agents. Nous avons observé les avantages et les limites d'un tel système en tant qu'extracteur de contours. Des améliorations seraient nécessaires pour rechercher des contours plus complexes,

que l'on peut trouver dans des images de scènes réelles. Du point de vu de résultat de segmentation, le système est loin d'être parfait. Il reste encore des points à traiter.

Malgré des difficultés dues à nos connaissances minimes dans le domaine de traitement d'image, à savoir les fonctions essentielles pour sa mise en place et les technologies qui s'y réfèrent, ce stage nous a représenté une grande opportunité pour toucher de près le domaine de recherche scientifique. En effet, ce stage nous a été d'une grande utilité afin d'enrichir nos connaissances techniques, découvrir et maîtriser des nouvelles technologies, d'acquérir de nouvelles connaissances dans le domaine de traitement d'images, et d'étudier en particulier une étape importante du traitement à savoir l'étape de segmentation. Il a contribué aussi à l'enrichissement de nos relations humaines ainsi que notre aptitude d'intégration dans le domaine professionnel.

5.5 Perspectives

Comme perspectives à notre travail, nous essayerons d'intégrer la logique floue dans le comportement des agents pour résoudre les cas d'imprécisions, la logique floue qui permet d'admettre des valeurs autres que le "vrai" et le "faux" absolu. En traduisant une idée d'appartenance partielle d'un élément à plusieurs classes, la logique floue permet de modéliser les connaissances incertaines et imprécises à travers les ensembles flous. Autre perspective, nous pensons à adapter le comportement des agents sur de images réelles. Enfin, les traitements effectués sur les images sont développé en Java ce qui est un peu difficile et nécessite le développement de toutes les fonctionnalités à zéro, nous pensons à profiter des fonctionnalités offerts par Matlab, et combiner Matlab avec la plate-forme multi-agents Jade.

Bibliographie

- [Bellet, 1998] Bellet, F. (1998). *An incremental approach based on cooperative and adaptive processes for grey level image segmentation.* Theses, Institut National Polytechnique de Grenoble - INPG.
- [Benzinou et al., 2004] Benzinou, A., Rodin, V., Guillaud, A., Ballet, P., Harrouet, F., Tisseau, J., and Bihann, J. L. (2004). An immune oriented multi-agent system for biological image processing. "Pattern Recognition.
- [Duchesnay, 2001] Duchesnay, E. (2001). *Multi-agent system organized in an irregular Pyramid : Application for image segmentation.* Theses, Université Rennes 1.
- [Ferber, 1995] Ferber, J. (1995). *Les Systèmes multi-agents : vers une intelligence collective.* I.I.A. Informatique intelligence artificielle. InterEditions.
- [Germond, 1999] Germond, L. (1999). *Three principles of cooperation for the segmentation of magnetic resonance images of the brain.* Theses, Université Joseph-Fourier - Grenoble I.
- [Giovanni, 2003] Giovanni, C. (2003). *JADE PROGRAMMING FOR BEGINNERS.* JADE TUTORIAL. TILAB, formerly CSELT.
- [Guillaud, 2000] Guillaud, A. (2000). *Un système multi-agents pour la reconnaissance de formes sur des pièces calcifiées, Aide à l'estimation de l'âge de poissons et de céphalopodes.* Thèse de doctorat, Université de Bretagne Occidentale, Brest, France.
- [Guillaud et al., 2002] Guillaud, A., Troadec, H., Benzinou, A., Le Bihann, J., and Rodin, V. (2002). A multiagent system for edge detection and continuity perception on fish otolith images. *EURASIP Journal on Applied Signal Processing.*
- [Hadjer, 2010] Hadjer, L. (2010). *Déploiement sur une plateforme de visualisation, d'un algorithme coopératif pour la segmentation d'images IRM basé sur les systèmes multi agents.* Thèse de doctorat, Université des sciences de la technologie de Houari Boumedienne.

- [Haroun, 2005] Haroun, R. (2005). *Segmentation des tissus cérébraux sur des images par résonance magnétique*. Thèse de doctorat, Université des sciences et de la technologie Houari Boumediène.
- [Haroun et al., 2004] Haroun, R., Boumghar, F., Hassas, S., and Hamami, L. (2004). A massive multi-agent system for brain MRI segmentation. In *Massively Multi-Agent Systems I, First International Workshop, MMAS 2004, Kyoto, Japan, December 10-11, 2004, Revised Selected and Invited Papers*.
- [Liu and Tang, 1999] Liu, J. and Tang, Y. Y. (1999). Adaptive image segmentation with distributed behavior-based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6).
- [Mahdjoub, 2011] Mahdjoub, J. (2011). *Vers un système de vision auto-adaptatif à base de systèmes multi-agents*. Thèse de doctorat, Université de Reims Champagne-Ardenne, France.
- [Mahdjoub et al., 2006] Mahdjoub, J., Guessoum, Z., Michel, F., and Herbin, M. (2006). A multi-agent approach for the edge detection in image processings. In *Fourth European Workshop on Multi-Agent Systems*, Lisbon, Portugal.
- [MANGOUA, 2008] MANGOUA, W. (2008). Utilisation de la logique floue pour traitement d'images par système multi-agents. In *EvoWorkshops*.
- [Mazouzi et al., 2005] Mazouzi, S., Batouche, M. C., and Guessoum, Z. (2005). Un système multi-agents auto-adaptatif pour la segmentation et la reconstruction de scènes 3d. In *STIT*, Sousse, Tunisie.
- [Mohammed and Said, 2012] Mohammed, R. and Said, A. (2012). Article : An adaptative multi-agent system approach for image segmentation. *International Journal of Computer Applications*, 51(12). Full text available.
- [Nwana et al., 1999] Nwana, H. S., Ndumu, D. T., Lee, L. C., and Collis, J. C. (1999). Zeus : A toolkit and approach for building distributed multi-agent systems. In *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, pages 360–361, New York, NY, USA. ACM.
- [Porquet et al., 2003] Porquet, C., Settache, H., Ruan, S., and Revenu, M. (2003). Une plate-forme multi-agent pour la segmentation d'images. etude des stratégies de coopération contour-région. In *ORASIS'2003*, Gérardmer, France.
- [Z James, 2015] Z James, W. (2015). Wang's research group.