

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master 16/925

Filière : Informatique

Option : Ingénierie des Médias

Thème :

**Maîtrise du comportement d'un Système
Multi-agents adaptatif**

Encadré Par :

Benhamza Karima

Présenté par :

Boudjehem Rochdi

Halimi Houssam

Juin 2016

Résumé

La segmentation d'images est une étape essentielle dans les processus de traitement d'image. L'approche de segmentation par contour consiste à localiser les frontières des régions. Alors que les approches de segmentation par région quant à elles agissent en partitionnant l'image en un ensemble de régions où chacune désigne un ou plusieurs objets connexes.

Dans la pratique les meilleurs résultats de segmentation sont obtenus en combinant conjointement des méthodes distinctes. En faisant cela nous obtenons des approches hybrides plus solides et plus efficaces, car la limite d'une méthode peut être surpassée par une autre, ou bien sa force peut être renforcée.

Dans ce travail, Une nouvelle approche hybride de segmentation d'images couleurs est proposée. C'est une approche adaptative basée sur une coopération région-contour au sein d'un système multi-agents.

La segmentation procède par l'élaboration d'un ensemble de régions et de contours initiaux qui vont être améliorés mutuellement offrant ainsi une vision à la fois globale et locale au processus de segmentation.

La conception d'un tel système s'avère très difficile et le lien entre le comportement collectif et celui des individus qui composent ce système est difficile à établir.

Nous défendons le principe de maîtriser le comportement d'un système multi-agents par une approche de contrôle, pour cela, nous agissons sur lui à partir d'informations relatives à ses comportements globaux.

Remerciements

Au premier lieu, je tiens à remercier vivement l'encadreur de ce mémoire, Mme Benhamza Karima, pour avoir dirigé ce travail avec compétence, patience et professionnalisme, pour tous les conseils précieux qu'elle m'a prodigué, pour l'intérêt qu'elle porte au développement de la recherche, pour les fortes impulsions qu'elle s'est si bien donner.

Je remercie sincèrement Messieurs les membres du jury pour avoir accepté de jurer ce modeste travail.

Je remercie Mr Hallaci Samir, pour toutes ces explications et conseils dans le domaine de traitement d'images.

Que toute personne ayant contribué, de près où de loin, à la mise en œuvre de ce travail trouve ici nos sincères remerciements.

Table des matières

Table des matières

Introduction générale	10
Chapitre I : Les systèmes multi-agents	13
I Introduction	13
II Agent	14
II.1 Définitions de l'agent	14
II.1.1 Définition de Wooldrige	14
II.1.2 Définition de Ferber	14
II.2 Caractéristiques d'un agent	15
II.3 Types d'agent	15
II.3.1 Agent réactif	15
II.3.1.1 Agent à réflexe simple	16
II.3.2 Agent cognitif	17
II.3.2.1 Agent ayant des buts	17
II.3.2.2 Agent BDI	18

II.3.3	Agent hybride	20
III	Systèmes multi-agents (SMA)	20
III.1	Définition d'un SMA	20
III.1.1	Domaines d'application des SMA	21
III.2	SMA réactifs et cognitifs	22
III.3	Caractéristiques d'un SMA	22
III.4	Interaction et coopération entre les agents	23
IV	Conclusion	25

Chapitre II : Dynamique de comportement d'un SMA 27

I	Introduction	27
II	Notion de l'émergence	27
II.1	Environnement d'un SMA	28
II.2	Dynamique d'un SMA	28
II.2.1	Ouverture et perturbations	29
II.2.2	Décentralisation	29
II.3	Emergence	30
II.4	Difficultés pour diriger le comportement d'un SMA	32
III	Approches de maîtrise du comportement global d'un SMA	34
III.1	Approches par construction	34
III.1.1	Méthodes de conception	35

III.1.2	Approches par ajustement des paramètres	36
III.1.2.1	Utiliser la connaissance du système	36
III.1.2.2	Explorer les paramètres en parallèle	36
III.1.3	Limite des approches par construction	37
III.2	Approches par contrôle	37
III.2.1	Contrôle d'un système dynamique	38
III.2.1.1	Objectif du contrôle	38
III.2.2	Approche de contrôle au niveau global	39
III.2.2.1	Intuition du contrôle	39
IV	Problématique et objectif ciblé	40
V	Conclusion	40

Chapitre III : Segmentation d'image 43

I	Introduction	43
II	Segmentation d'image	43
II.1	Définition	43
II.2	Domaines d'application	44
II.3	Différentes approches de segmentation	45
II.3.1	Approches Contour	45
II.3.1.1	Principe	45
II.3.1.2	Techniques d'extraction de contours	46

II.3.1.3	Limites de segmentation par contour	47
II.3.2	Approches Région	48
II.3.2.1	Principe	48
II.3.2.2	Limites de segmentation par région	49
II.3.3	Seuillage	49
II.4	Formes de coopération	51
II.4.1	Coopération séquentielle	51
II.4.2	Coopération des résultats	52
II.4.3	Coopération mutuelle	53
III	Travaux Connexes	53
IV	Conclusion	57

Chapitre IV : Conception et implémentation de notre système 59

I	Introduction	59
II	Description et objectifs de l'application	59
II.1	Choix de l'application	59
II.2	Objectif de l'application	60
III	Conception	60
III.1	Modèle proposé pour la segmentation coopérative	61
III.2	Modélisation de problème	61

III.2.1	Modélisation de l'environnement	61
III.2.2	Modélisation d'agents	62
III.2.2.1	Agents contour	62
III.2.2.2	Agents région	63
IV	Réalisation :	64
IV.1	Environnement de développement (NetBeans)	64
IV.2	Présentation de l'application	65
IV.3	Processus d'exécution	67
V	Discussion de résultats	70
V.1	Base d'images	70
V.2	Comparaison des résultats	71
VI	Conclusion	73
	Conclusion générale et perspectives	75

Liste de figures

Table des figures

I.1	Schéma d'un agent à réflexes simples.....	16
I.2	Schéma d'un agent ayant des buts.....	18
I.3	Diagramme d'une architecture BDI.....	19
I.4	Schéma d'un système multi agent	21
III.1	Segmentation dans un processus d'analyse d'image.....	44
III.2	Les approches de segmentation d'image.....	45
III.3	Le processus classique de détecteur de contours.....	46
III.4	Coopération séquentielle	52
III.5	Coopération des résultats	52
III.6	Coopération mutuelle.....	53
IV.1	Modèle proposé pour la segmentation coopérative.....	61
IV.2	Interface principale de l'application.....	65
IV.3	Menu chargement d'image et sauvegarde de résultats.....	66
IV.4	Menu lancement des méthodes de segmentation classiques.....	66
IV.5	Menu lancement de notre méthode	67

IV.6	Boîte de dialogue d'ouverture d'image.....	67
IV.7	Aperçu de l'image choisi	68
IV.8	Détecteur de contour canny	69
IV.9	Segmentation en utilisant la méthode de croissance de région.....	69
IV.10	Résultat de la segmentation en utilisant notre méthode.....	70
IV.11	Comparaison des résultats expérimentaux	72

Introduction générale

Introduction générale

Ce travail s'inscrit dans le domaine des systèmes multi-agents adaptatifs et s'applique sur une branche du traitement d'images (segmentation d'image).

On définit généralement un système multi-agents (SMA) comme étant un modèle informatique composé d'entités de base (les agents). On s'intéresse notamment au phénomène d'émergence issue des différentes interactions entre les agents dans un environnement ouvert et décentralisé. Au sein de ces systèmes, il est presque impossible de prédire l'état du système à un moment donné ainsi que l'état final de ce système.

Malgré qu'on peut créer des agents dont on connaît à l'avance le comportement de chacun d'entre eux, on ne peut pas maîtriser le comportement du système entier à cause des différentes interactions entre ces agents. Il est aussi très difficile de créer un système capable d'aboutir à un résultat souhaité. Par contre, on peut diriger le comportement du système pour qu'il puisse aboutir au résultat demandé.

Dans ce contexte, l'objectif de notre travail est la conception d'un système multi-agents adaptatif à fonctionnalité émergente composé d'agents réactifs. La réalisation de différents tests permettra la découverte des réponses du système aux problèmes du dynamisme de l'environnement, ainsi que le contrôle de ce dernier. Le but est de mieux comprendre et de diriger le comportement d'un SMA réactif, en particulier lorsqu'il n'est pas possible d'assurer le comportement optimal souhaité lors de la construction du système. Ce dernier a été appliqué sur un problème de segmentation d'images.

Le travail présenté est organisé en quatre chapitres :

Le premier chapitre traite les concepts d'agents, leurs caractéristiques, leurs archi-

tectures et leurs différents types d'agents. Puis, il présente les systèmes multi-agents, leurs caractéristiques, ainsi que les différentes interactions et coopérations entre ses agents.

Le deuxième chapitre est consacré à l'étude de la dynamique de comportement d'un SMA ainsi que les différentes approches pour maîtriser le comportement ce dernier.

Le troisième chapitre présente le domaine d'application : la segmentation d'images en commençant par citer les différentes approches classiques et les formes de coopérations existantes.

Le quatrième chapitre présente la conception et l'implémentation de notre modèle avec une discussion et comparaison des résultats obtenus.

Finalement, ce travail se termine par une conclusion générale et quelques perspectives souhaitées.

Chapitre I :
Les systèmes multi-agents

Chapitre I

Les systèmes multi-agents

I Introduction

L'intelligence artificielle (IA) s'intéresse à la conception de machines pouvant se substituer à la pensée humaine. Elle est directement issue des concepts cognitivistes. C'est pourquoi certains auteurs préfèrent le terme de computation pour désigner cette orientation théorique. L'IA a connu de nombreux succès dans des applications variées et particulièrement dans la conception de systèmes experts. L'un de ces objectifs le plus important est la définition de systèmes capables de représenter des connaissances, de raisonner, et de planifier des actions afin de résoudre des problèmes pouvant être très complexes. Elle cherche ainsi à obtenir des résultats comparables à ce que feraient des êtres humains dans des cas similaires, mais sans forcément utiliser les mêmes moyens.

Dans sa notion classique, l'IA consiste à regrouper le comportement Intelligent dans une seule base de connaissances, cette modélisation a vite rencontré un certain nombre de difficultés, comme la nécessité d'incorporer, au sein d'une même base de connaissances, l'expertise, les aptitudes et les connaissances d'entités différentes qui dans la réalité, communiquent et coopèrent à la réalisation d'un but ordinaire. Cette insuffisance a permis l'avènement d'une nouvelle discipline qui est l'intelligence artificielle distribuée.

Cette Approche a permis la modélisation des comportements intelligents qui sont le produit de l'activité coopérative entre plusieurs agents, d'où la réalisation des systèmes

dits : « multi-agents ». En effet, le but principal d'un système multi-agents (SMA) est de faire collaborer un certain nombre d'agents afin de résoudre un problème.

Ce chapitre présente les notions d'agents et de systèmes multi-agents. Ensuite, il expose leurs architectures et notamment leurs caractéristiques.

II Agent

II.1 Définitions de l'agent

Le concept d'agent a été utilisé dans différentes disciplines : dans les systèmes à base de connaissance, la robotique et autres domaines de l'IA, mais aussi dans des domaines comme la philosophie et la psychologie. Dans la littérature, on trouve une multitude de définitions. On peut citer :

II.1.1 Définition de Wooldridge

Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome pour atteindre les objectifs(buts)pour lesquels il a été conçu [Wooldridge *et al.*, 1995].

II.1.2 Définition de Ferber

Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents[Ferber, 1995].

II.2 Caractéristiques d'un agent

En utilisant quelques définitions de la littérature, il est possible de créer un ensemble de caractéristiques d'agent :

Situé : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement ;

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 ;

Proactif : l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment ;

Capable de répondre à temps : l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis ;

Social : l'agent doit être capable d'interagir avec des autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs.

II.3 Types d'agent

Les agents peuvent être classés en deux catégories principales selon leur comportement :

II.3.1 Agent réactif

Ce sont des agents qui réagissent uniquement à leurs perceptions de l'environnement et qui agissent en fonction de cette perception. C'est le niveau de complexité le plus bas des agents. Ce type d'agents ne peut pas vraiment être qualifié d'intelligent, étant donné que leur fonctionnement est principalement basé sur le principe du stimulus/action. Ce principe permet aux agents d'agir grâce à des réflexes totalement conditionnés. À la perception d'un stimulus particulier, l'agent fournit une réponse stéréotypée. La communication qu'il a avec d'autres agents et avec son environnement est très rudimentaire. Néanmoins, ils

présentent des caractéristiques intéressantes [Drogoul 93] : simplicité de la description du comportement local et émergence de comportements globaux au-delà de la simplicité de l'agent.

Les principaux travaux utilisant des agents réactifs concernent les robots [Bureau *et al.*, 2001], [Steels et Brooks, 1994], la planification [Gallone J-M., 1994] ou la résolution collective de problèmes [Deneubourg *et al.*, 1991]. On utilise parfois le terme d'animats. La sous-section suivante décrit un modèle qui peut servir à la conception d'agents réactifs.

II.3.1.1 Agent à réflexe simple

Ce type d'agent agit uniquement en se basant sur ses perceptions courantes. Il utilise un ensemble de règles prédéfinies, du type Si condition alors action, pour choisir ses actions.

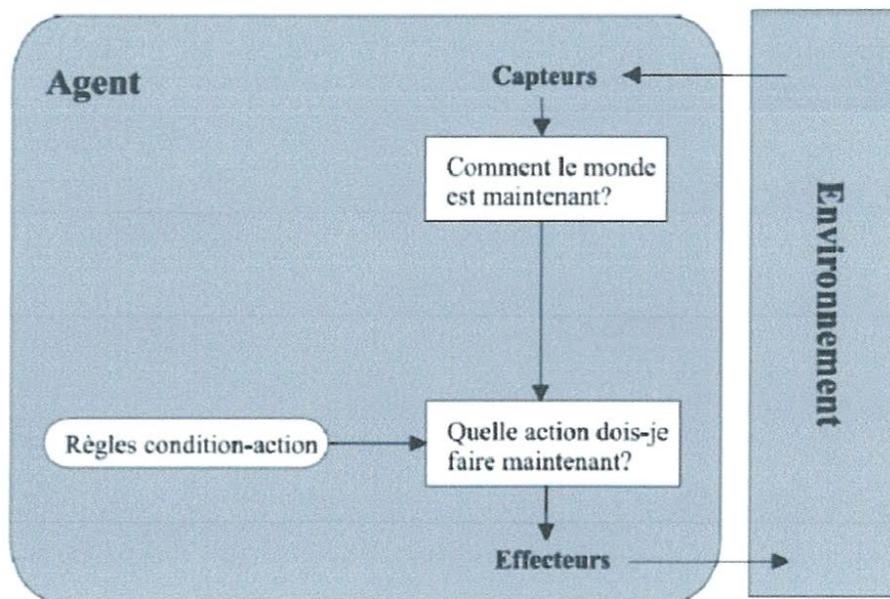


FIGURE I.1 Schéma d'un agent à réflexes simples.

Comme on peut le constater, ces règles permettent d'avoir un lien direct entre les perceptions de l'agent et ses actions. Le comportement de l'agent est donc très rapide, mais peu réfléchi. à chaque fois, l'agent ne fait qu'exécuter l'action correspondant à la règle activée par ses perceptions (Figure I.1)

II.3.2 Agent cognitif

C'est un type d'agents plus complexe [Wooldridge, 2009]. De tels agents sont non seulement capables de percevoir et d'agir sur leur environnement, mais en plus ils ont des capacités de cognition leur permettant de raisonner sur les autres ou sur l'avancement de la résolution. Ils font souvent appel à des modes de communication plus complexes qu'une simple perception. Ils communiquent généralement grâce à des structures de données partagées ou par des communications directes.

Pouvant utiliser leurs capacités cognitives, l'agent cognitif échange des données plus riches au niveau sémantique que les agents réactifs. En effet, il est souvent pourvu de connaissances et de savoir-faire leur permettant des raisonnements plus élaborés (inférences, filtrage des informations, accointances [Djemame et Batouche, 2009], représentation neuronale...). La gestion de leurs interactions s'en trouve donc différente [Chevrier, 1993]. Bien souvent, les agents cognitifs disposent d'une représentation explicite des autres agents et de la manière d'interagir avec eux. Cette représentation est désignée par le terme de réseau d'accointance. Les deux sous-sections suivantes décrivent un modèle qui peut servir à la conception d'agent cognitif.

II.3.2.1 Agent ayant des buts

Contrairement à l'agent réactif, l'agent cognitif, qui raisonne sur les buts, tient compte d'une certaine projection dans le futur. Il se pose des questions comme « Qu'est-ce qui va arriver si je fais telle ou telle action ? » et « Est-ce que je serai satisfait si cela se produit ? ». Bien entendu, l'agent raisonnant sur ses buts prend, en général, beaucoup plus de temps à agir qu'un agent réactif. Il offre en revanche beaucoup plus de flexibilité.

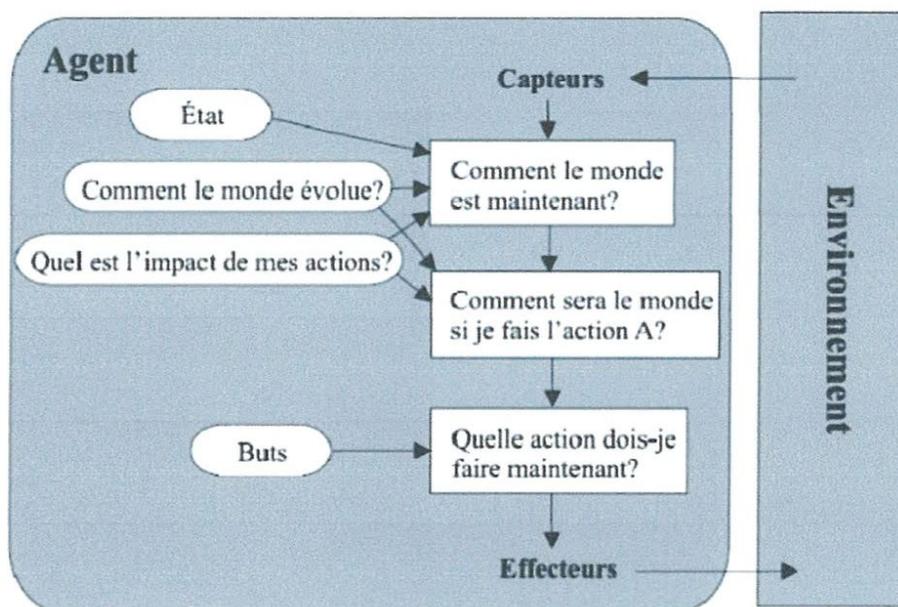


FIGURE I.2 Schéma d'un agent ayant des buts.

La figure I.2 montre la structure d'un agent basé sur les buts. Comme on peut le constater, il est identique à l'agent réactif gardant une trace de l'environnement, sauf qu'il se projette dans le futur pour voir l'impact de ses actions et qu'il choisit ses actions en se basant sur ses buts, contrairement à l'agent réactif qui ne faisait qu'appliquer des règles prédéfinies pour relier ses perceptions à ses actions.

II.3.2.2 Agent BDI

L'architecture BDI est une autre approche utilisée dans la conception des agents cognitifs. BDI est un acronyme qui signifie, en anglais, Belief, Desire, Intentions. Ce qui se traduit en français par croyances, désirs et intentions. Les agents se basent donc sur ces trois aspects pour choisir leurs actions. Dans ce cadre, Wooldrige [Wooldridge *et al.*, 1995] propose une architecture ayant sept composantes, telles que présentées sur la figure I.3 :

- Un ensemble de croyances courantes, représentant les informations que l'agent possède à propos de son environnement courant ;
- Une fonction de révision des croyances, qui prend les entrées des capteurs et les croyances actuelles de l'agent et qui détermine un nouvel ensemble de croyances ;

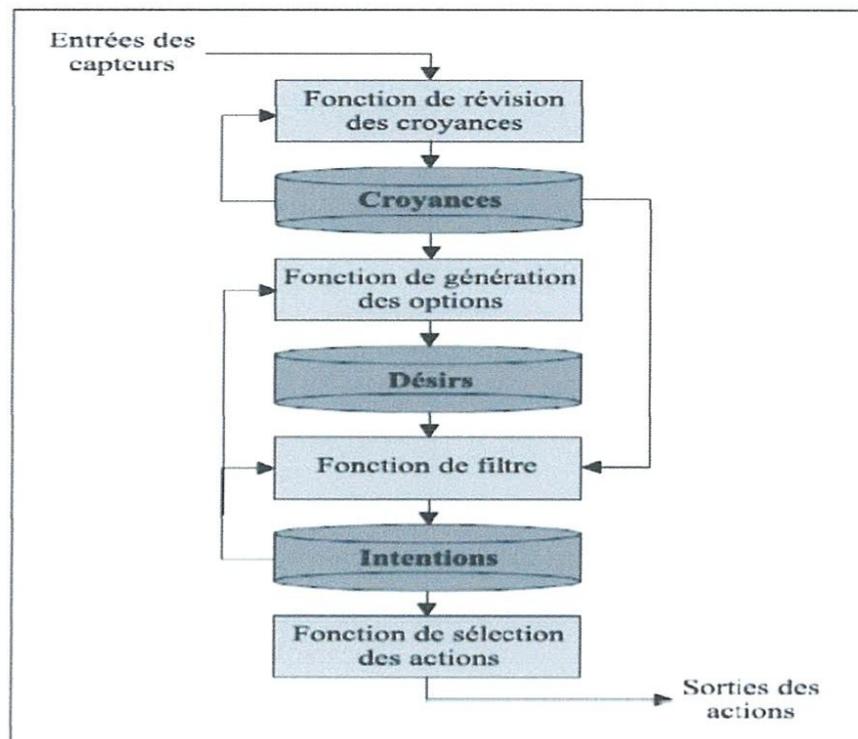


FIGURE I.3 Diagramme d'une architecture BDI.

- Une fonction de génération des options, qui détermine les options disponibles pour l'agent (i.e. ses désirs), en se basant sur les croyances courantes de l'agent à propos de son environnement et sur ses intentions courantes ;
- Un ensemble de désirs, représentant les options disponibles à l'agent ;
- Une fonction de filtre, qui représente le processus de délibération de l'agent et qui détermine les intentions de l'agent en se basant sur ses croyances, ses désirs et ses intentions courantes ;
- Un ensemble d'intentions courantes, représentant le centre d'attention actuel de l'agent, c'est-à-dire les buts envers lesquels il s'est engagé et envers lesquels il a engagé des ressources ;
- Une fonction de sélection des actions, qui détermine l'action à effectuer en se basant sur les intentions courantes de l'agent.

En résumé, un agent BDI doit donc mettre à jour ses croyances avec les informations qui lui proviennent de son environnement, décider quelles options lui sont offertes, filtrer ces

options afin de déterminer de nouvelles intentions et poser ses actions en se basant sur ses intentions

II.3.3 Agent hybride

Un agent hybride possède des composants réactifs et aussi des composants cognitifs pour garantir un raisonnement de qualité [Sycara, 1989]. Dans ce cas, un agent est composé de plusieurs couches arrangées selon une hiérarchie. La plupart des architectures considèrent trois couches. Au plus bas niveau, on trouve une couche purement réactive, qui prend ses décisions en se basant sur des données brutes en provenance des senseurs. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une version qui se situe au niveau des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement (raisonnement tenant compte des autres agents).

III Systèmes multi-agents (SMA)

III.1 Définition d'un SMA

Nous avons vu qu'un agent est une entité relativement autonome présente dans un système qui lui permet d'évoluer, de communiquer etc. Ce système est appelé un système multi-agents.

Un système multi-agents peut être défini [Ferber, 1995] comme un système composé des éléments suivants :

- un environnement E identifié et muni d'un système de repérage dans l'espace (souvent Euclidien).
- un ensemble d'objets O passifs pouvant être perçus, créés, modifiés ou détruits par des agents.
- un ensemble de relations R qui unissent des objets entre eux.

- un ensemble d'opérations Op offrant la possibilité aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.
- un ensemble de lois universelles qui sont des opérateurs chargés de représenter l'application des actions des agents sur le monde et la réaction du monde à ces actions. Un SMA peut-être ouvert (cas d'un magasin où des clients entrent et sortent librement) ou fermé (l'ensemble d'agents reste le même comme dans un match de football).

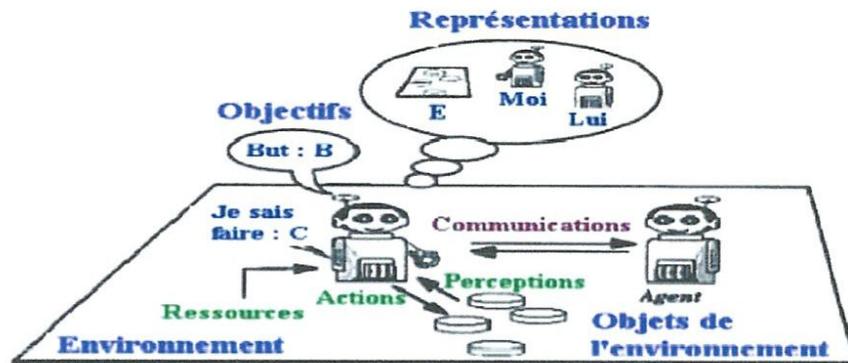


FIGURE I.4 Schéma d'un système multi agent[Ferber, 1995].

Un SMA peut être homogène, c'est-à-dire que tous les agents sont construits sur le même modèle ou hétérogène, c'est-à-dire que les agents sont de modèles différents et/ou de granularité différentes.

III.1.1 Domaines d'application des SMA

Les SMA peuvent être différenciés en trois familles d'application [Boissier *et al.*, 2004a] :

La simulation de systèmes, par exemple biologiques [Thomas *et al.*, 2002] ou sociaux [Amblard, 2003],

La résolution de problèmes,

L'intégration de l'informatique avec les êtres humains et les systèmes mécaniques, c'est-à-dire des SMA ayant une raison d'être propre (réseaux P2P [Siebert *et al.*, 2008], systèmes multi-robots)[Simonin *et al.*, 2002].

III.2 SMA réactifs et cognitifs

Un SMA composé d'agents réactifs (respectivement cognitifs) est lui-même appelé réactif (cognitif). Un système réactif comporte souvent une ou quelques populations composées de nombreux agents identiques, tandis qu'un système cognitif comporte quelques agents souvent hétérogènes. La distinction réside avant tout dans l'objectif lié à l'étude du système, comme indiqué dans [Parunak 99] :

- L'étude des SMA cognitifs cherche à améliorer les comportements individuels des agents en s'intéressant à leur intelligence individuelle, leur modèle cognitif, et aux communications. Elle met l'accent sur l'agent et ses capacités.
- L'étude des SMA réactifs cherche à comprendre le fonctionnement du système comme un tout, en se focalisant sur les interactions et la dynamique qui en résulte, donc sur les aspects collectifs du système.

III.3 Caractéristiques d'un SMA

Il existe des caractéristiques propres aux SMA, par rapport aux autres systèmes informatiques. Nous en fournissons une liste issue de la littérature, proposée par O. Boissier, S. Gitton et P. Glize dans [Boissier *et al.*, 2004a].

Un SMA possède la plupart des caractéristiques suivantes :

Distribution : le système est modulaire, l'élément de base étant l'agent.

Autonomie : un agent est en activité permanente et prend ses propres décisions en fonction de ses objectifs et de ses connaissances.

Décentralisation : les agents sont indépendants, il n'y a pas de décisions centrales valables pour tout le système.

Charge de connaissance : les agents sont capables de communiquer entre eux, selon des langages plus ou moins élaborés.

Interaction : les agents ont une influence localement sur le comportement des autres agents, généralement sur un pied d'égalité (il n'y a pas d'ordres, seulement des requêtes).

Organisation : les interactions créent des relations entre les agents, et le réseau de ces relations forme une organisation qui peut évoluer au cours du temps.

Situation dans un environnement : les agents sont ancrés dans un environnement, source de données, de contraintes et d'incertitude, lieu d'actions et d'influences entre agents. L'évolution du SMA est la combinaison des évolutions des agents et de l'environnement.

Délégation : l'utilisateur accepte de ne pas maîtriser le comportement de l'application globale, à défaut de pouvoir supporter la complexité liée à l'ensemble des décisions prises par les agents dans le système. Il délègue une partie du contrôle de l'application globale aux agents.

Personnalisation : lorsqu'un agent représente un utilisateur, typiquement dans un SMA appartenant à la famille des systèmes intégrés dans un contexte plus large, il s'adapte à lui.

Intelligibilité : les SMA proposent une manière naturelle de modéliser d'autres systèmes ou de mettre en oeuvre des applications, ce qui les rend simples à appréhender pour un utilisateur extérieur.

III.4 Interaction et coopération entre les agents

Les agents peuvent interagir en communiquant directement entre eux ou par l'intermédiaire d'un autre agent ou même en agissant sur leur environnement.

Le fait d'avoir plusieurs agents actifs au même moment dans le système implique de nouveaux phénomènes :

1. Les effets des actions de chaque agent et les lois d'évolution de l'environnement se combinent, et peuvent parfois s'amplifier, s'annuler, se perturber...

2. Les agents peuvent percevoir les effets provoqués par les actions des autres.

Ainsi, un agent peut avoir connaissance du fait qu'il n'est pas seul dans son environnement. Ces deux remarques nous amènent à la notion d'interaction qui peut être définie de la façon suivante : Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'actions réciproques [Ferber, 1995].

- **La coopération** : La coopération est une attitude des agents qui décident de travailler en commun. Dans ce cas, les agents coopèrent s'ils s'engagent dans une action commune après avoir identifié un but commun. Ferber [Ferber, 1995] considère que plusieurs agents coopèrent, pour atteindre un objectif commun, est vérifiée l'ajout d'un nouvel agent accroît différentiellement les performances du group et s'il existe des conflits potentiels ou actuels d'accession à des ressources et l'action des agents sert à éviter ou à sortir de tels conflits.
- **Communication entre les agents** : La communication est essentielle dans la résolution coopérative des problèmes [Ferber, 1995].

On distingue généralement deux modèles de communication :

La communication par partage d'information : Les messages transitent par un système qui les partage ensuite. La communication est centralisée et les agents communiquent indirectement entre eux.

La communication par envoi de message : Cette communication distribuée va être réutilisée en partie par les agents qui seront alors connectés directement entre eux. Il y'a donc dans ce modèle aucune supervision de la communication. Les agents peuvent communiquer avec n'importe lequel des autres agents dans la mesure où il connaît son existence.

- **La coordination** : D'après Ferber [Ferber, 1995], la coordination définit la manière dont les actions des différents agents doivent être organisées dans le temps et l'espace de manière à réaliser les objectifs. La coordination permet d'organiser un ensemble d'actions ce qui évite que les agents se gênent mutuellement. Selon Durfee et Montgomery le but de la coordination [Moraitis, 1994] est de trouver, parmi un ensemble

de comportements d'agents qui interagissent, une collection de comportements qui réalise d'une façon satisfaisante les objectifs les plus importants des agents.

IV Conclusion

Dans ce chapitre nous avons essayé de définir ce qu'est un agent et ce qu'est un système multi-agents avec leurs notions élémentaires, car l'idée de base des SMA est de faire cohabiter des entités autonomes, capable de percevoir et d'agir sur l'environnement en les faisant interagir et collaborer pour résoudre des problèmes complexes et distribués.

Dans le prochain chapitre, nous présenterons la dynamique de comportement d'un SMA.

Chapitre II :

Dynamique de comportement d'un SMA

Chapitre II

Dynamique de comportement d'un SMA

I Introduction

Un SMA est un système composé d'entités informatiques, appelées des agents, qui évoluent et interagissent dans un environnement commun. La notion d'interaction entre agents est essentielle car chacun d'eux est impliqué dans une dynamique commune, au lieu d'évoluer parallèlement et indépendamment aux autres.

Dans ce chapitre, nous nous attaquons d'abord à définir c'est quoi le concept d'émergence dans un système multi-agent, et celui de comportement global, nous identifions les problèmes fondamentaux soulevés lorsqu'un utilisateur cherche à maîtriser ce comportement.

II Notion de l'émergence

Avant de présenter la notion d'émergence, il est impérative de donner quelques définitions des notions fondamentales liées à ce concept concernant l'environnement et la dynamique d'un SMA. En se basant sur les caractéristiques des SMA déjà mentionnés dans le premier chapitre, on se focalisera ensuite sur deux notions particulières qui sont « l'ouverture » et « la décentralisation ».

II.1 Environnement d'un SMA

Avec celle des comportements individuels, la caractérisation de l'environnement permet de définir la dynamique d'un SMA.

Selon [Weyns *et al.*, 2004], les différents rôles de l'environnement sont de permettre la communication entre agents, d'être le support des actions des agents en définissant les règles et en renforçant ces actions, d'être observable par les agents, et enfin de prendre en charge l'activité propre des objets et des ressources présents en son sein.

Selon [Boissier *et al.*, 2004b], un SMA possède un environnement dans lequel plusieurs agents évoluent, communiquent, perçoivent et agissent. Il peut être essentiel au système, par exemple si les agents se déplacent spatialement en son sein (on dit alors que les agents sont situés) ou s'il possède une dynamique propre, par exemple s'il contient des objets qui évoluent selon des lois physiques, indépendamment des agents. Il peut au contraire être très peu présent, jusqu'à se limiter aux messages échangés par des agents cognitifs.

Ainsi, l'environnement détient une importance fondamentale pour les agents réactif, car il joue le rôle d'intermédiaire à leurs interactions. Comme ces agents sont dépourvu de la faculté de communiquer directement entre eux, ils s'influencent mutuellement soit par leur position s'ils sont situés, soit par l'intermédiaire d'objets qu'ils perçoivent et modifient.

II.2 Dynamique d'un SMA

Selon [Klein, 2009], Un SMA en fonctionnement évolue au cours du temps, sous l'influence des comportements individuels des agents et de la dynamique propre à l'environnement. Le modèle influences-réaction de [Ferber et Müller, 1996] permet d'exprimer et de résoudre les conflits entre des actions contradictoires des agents.

Ce modèle définit l'état dynamique d'un SMA un couple $\langle \sigma, \gamma \rangle$ dont les membres appartiennent respectivement à un ensemble d'états Σ de l'environnement et à un ensemble d'influences Γ des agents sur l'environnement.

Une fonction opérateur génère cette influence, tandis qu'une fonction traduit les lois

qui régissent les réactions de l'état de l'environnement à une influence . Des fonctions d'exécution et de réaction prennent en compte ces données pour mettre à jour le système.

La dynamique du SMA se traduit par une fonction Cycle : $\Sigma \times \Gamma \Rightarrow \Sigma \times \Gamma$ qui met à jour son état dynamique $\langle \sigma, \gamma \rangle$ de façon régulière. À tout instant, le SMA se trouve dans un état, et son évolution est l'ensemble des états successifs.

II.2.1 Ouverture et perturbations

Pour un SMA, l'ouverture désigne la capacité d'ajouter ou de retirer dynamiquement dans le système des agents [Sichman, 1995], ou des fonctionnalités et des services [Vercouter, 2001, Vercouter, 2004] de ces agents. Pour [Wooldridge et Ciancarini, 2001], c'est l'impossibilité de savoir lors de la conception quels seront les composants du SMA ni comment ils vont interagir les uns avec les autres.

Selon [Klein, 2009], la notion d'ouverture d'un SMA est l'existence potentielle de perturbations exogènes au système. Leur origine est facile à expliquer pour des SMA intégrés dans un environnement plus large, qui contient des utilisateurs humains ou une dynamique physique propre.

Pour les SMA utilisés en résolution de problèmes, [Gechter et Simonin, 2005] présente les contraintes du problème comme des perturbations de l'environnement. Pour la simulation de systèmes, enfin, les perturbations sont celles qui peuvent intervenir sur le système modélisé.

II.2.2 Décentralisation

L'évolution d'un SMA résulte des différentes interactions entre ses agents. Ainsi, sa dynamique est définie au niveau des comportements individuels de ces agents. Cela lui fournit la robustesse requise lorsqu'il est utilisé pour résoudre des problèmes, son fonctionnement est indépendant d'aucun agent particulier. Le changement du comportement d'un agent (i.e : en cas de dysfonctionnement) ne va pas altérer le fonctionnement du SMA dans son ensemble, à partir du moment où les agents sont remplaçables.

En contre partie, cette décentralisation impose au SMA de fortes contraintes. Elle va imposer pour l'observateur de l'environnement des contraintes de localité similaires à celles des agents : il ne peut percevoir et agir que sur une partie du système.

Aussi, l'état courant du SMA n'est pas forcément connu intégralement, seuls quelques indicateurs limités sur son état sont disponibles, lui permettant son observation partielle, et toute action destinée à influencer sur le SMA pourra ne pas s'appliquer à l'ensemble du système, ou pas de manière synchronisée.

II.3 Emergence

Après avoir exposé les différentes définitions concernant l'environnement et la dynamique d'un SMA, nous allons présenter la problématique ciblée.

L'observation de l'évolution d'un SMA permet en générale de connaître les différents comportements de la plupart de ses agents. Ces comportements jouent un rôle important dans cette étude. Le comportement du SMA est imprévisible, car il ne peut être défini avec précision au sein du système, au contraire de celui des agents, qui est explicitement défini au départ. Le comportement global du SMA émerge des interactions lors de l'évolution du système. Le terme d'émergence détient plusieurs définitions dans la littérature.

[Jean *et al.*, 1997] ont proposé une définition de l'émergence dans un SMA à deux niveaux :

- Au niveau micro du système, un ensemble d'agents sont en interaction, selon une dynamique rapide, au sein d'un environnement qui sert à la fois de médium de connaissance et d'ensemble de contraintes.
- Au niveau macro, un phénomène collectif est produit, suivant une dynamique plus lente donc plus observable que le niveau micro.

L'émergence correspond à la description de ce phénomène global, soit par l'observateur, soit par les agents.

Ainsi, le phénomène d'émergence n'est pas défini explicitement, mais résulte des comportements des agents et de leurs interactions. Il a lui-même une influence sur les agents. Le lien entre un comportement collectif et les individus qui le composent est difficile à établir. Obtenir un comportement collectif particulier en influant le niveau des agents est donc aussi difficile.

[Deguet, 2008] définit l'émergence comme un avis plus ou moins consensuel, sans fondement formel, regroupant tout ou partie des critères suivants :

Sens commun : c'est l'apparition soudaine et importante d'une chose qui était cachée ou inexistante.

Causalité descendante : l'émergent, ou épiphénomène, est perçu indépendamment des entités qui composent le système, même s'il survient d'elles, et a une causalité de son niveau global sur le niveau local de ces entités.

Observation et prédiction : l'émergence facilite la description de l'état du système ou de son évolution future. Mais il n'y a pas de réduction possible : on ne peut pas déduire le niveau global du niveau local. L'observation du système est nécessaire, et le meilleur moyen de prévoir son évolution est de le simuler.

Complexité à partir de la simplicité : l'émergence est une description simple d'un comportement complexe.

Interprétation : l'émergence est l'interprétation d'un phénomène résultant d'une interaction entre les agents et l'environnement. L'observation et l'interprétation peuvent être faites par une personne extérieure au système (émergence faible) ou par les entités locales elles-mêmes (émergence forte).

Émergence et auto-organisation : bien qu'elles se rencontrent souvent ensemble, ces notions ne font pas consensus. Pour [De Wolf et Holvoet, 2004], elles sont indépendantes mais toutes deux positives, et c'est leur combinaison qui fait leur force. Pour [Shalizi *et al.*, 2001], au contraire, l'auto-organisation génère la complexité du système, ce qui incite à rechercher des régularités émergentes pour la réduire.

Le tout est supérieur à la somme des parties : cette phrase est classique de la caractérisation de l'émergence, et traduit le gain apporté au système par l'interaction de ses composants locaux.

Point de vue pragmatique : un phénomène émergent est surprenant pour un utilisateur extérieur, même s'il connaît les règles de comportement locales.

Ces critères expriment l'idée que la connaissance du niveau local ne suffit pas pour connaître le phénomène émergent que va apparaître, ou pour expliquer ce phénomène. Réciproquement, obtenir un phénomène émergent en définissant ou en influant sur le niveau local d'un SMA est un problème difficile.

Selon [Drogoul *et al.*, 2004], dans un système d'entités en interaction, l'émergence est la production d'un phénomène (processus, état stable ou invariant) global au regard des entités, et observé soit par un observateur extérieur, soit par les entités elles-mêmes. Pour un SMA réactif, l'observateur est généralement extérieur au système car les agents eux-mêmes n'ont pas la capacité d'identifier des comportements globaux.

La construction d'un SMA vise à la réalisation d'objectifs globaux en définissant son niveau local. Son évolution dépend ensuite de données fournies par l'environnement, d'une initialisation particulière, ou de la modification d'agents due à l'ouverture du système. Il y a émergence si le SMA atteint un état non définissable a priori pour chaque évolution.

II.4 Difficultés pour diriger le comportement d'un SMA

L'une des questions principales liées à l'émergence et aux SMA est de déterminer comment concevoir le niveau local du système (Agents, Interactions et Organisation) pour garantir l'obtention d'un comportement émergent particulier au niveau global.

Généralement, la démarche consiste en une élaboration préalable d'un SMA grâce à des considérations heuristiques, suivie d'une phase empirique composée d'une succession de tests et d'adaptations permettant de parvenir à une configuration voulue. Il n'y a pas de vérification formelle possible des propriétés du SMA.

Des protocoles expérimentaux permettent toutefois de couvrir largement l'espace

des réponses, mais se heurtent à la difficulté de spécifier formellement les comportements émergents. La phase d'élaboration préalable du SMA repose sur des principes d'ingénierie qui cherchent à reproduire des phénomènes naturels connus, en particulier le bio-mimétisme et le socio-mimétisme.

Différents travaux ont visé la recherche à diriger les phénomènes émergents d'un SMA, et parlent de comportements émergents. Parmi eux, [De Wolf *et al.*, 2005] explique qu'il est difficile de prouver que le comportement est bien maîtrisé, par manque de démonstrations formelles. Ainsi, on peut extraire les trois problèmes qui mènent à la difficulté de lier le niveau local où est défini le SMA à ses comportements globaux observés :

Mesurer le comportement global : C'est-à-dire déterminer celui qui est présenté par un SMA à un instant donné. Ce problème dépend essentiellement de la nature du SMA étudié.

Maîtriser un SMA : C'est-à-dire faire en sorte qu'il présente effectivement un comportement global désiré, diriger le système.

Certifier qu'une solution de maîtrise d'un SMA donne les résultats attendus : Ou encore évaluer cette solution pour savoir si elle est utile.

Ainsi, le comportement d'un SMA à un instant donné peut être défini comme une description selon le point de vue de l'observateur de l'ensemble des phénomènes émergents qu'il observe. Ce comportement a un sens pour l'observateur, et peut être désiré ou non. Plusieurs comportements différents peuvent être atteints par un même SMA, en fonction de conditions initiales, de perturbations, ou du caractère aléatoire de l'évolution du système.

La question la plus importante est de savoir comment l'utilisateur d'un SMA (par exemple son concepteur), peut l'influer afin d'assurer qu'il présente le comportement désiré. De la même manière que [Drogoul *et al.*, 2004], nous considérons donc l'observateur comme un utilisateur : il cherche à agir sur le SMA pour atteindre, ou au moins favoriser, un comportement souhaité.

En outre, il est difficile de prouver que ce comportement est bien maîtrisé, par manque de démonstrations formelles, comme indiqué dans [De Wolf *et al.*, 2005]. Ces pro-

blèmes se posent que l'on cherche à diriger le comportement d'un SMA préexistant, ou même que l'on construise un SMA pour qu'il présente un comportement désiré. L'utilisation d'approches par mimétisme, qui copient des systèmes connus en espérant que les phénomènes qui s'y trouvent soient transférés dans le SMA construit, montre à quel point il est difficile de maîtriser l'émergence en définissant les comportements individuels.

III Approches de maîtrise du comportement global d'un SMA

Différentes approches de maîtrise du comportement global d'un SMA ont été proposées, nous distinguons deux grandes familles de catégories [Klein, 2009] :

Les approches par construction : Elles se situent à la conception du système et permettent de construire un SMA qui présente un comportement désiré. Ces approches sont en générale insuffisantes pour assurer le comportement global d'un SMA.

Les approches qui cherchent à contrôler le système : Elles se concentrent sur le contrôle du comportement global lors de l'évolution du SMA. Ces approches amènent le système à passer d'un comportement non souhaité à un comportement souhaité, appelé cible.

Pour chacune d'entre elles, nous allons présenter les apports techniques, les difficultés rencontrées par les auteurs et leurs recommandations pour y remédier.

III.1 Approches par construction

Ces approches concernent les travaux qui vise à concevoir ou à construire des SMA qui présentent un comportement désiré.

[Edmonds, 2004] explique qu'il existe deux manières d'obtenir un SMA qui se comporte conformément aux spécifications : l'ingénierie et l'adaptation. Il est montré dans

[Edmonds et Bryson, 2004] que l'ingénierie seule n'est généralement pas suffisante pour parvenir aux spécifications attendues et qu'une approche expérimentale est nécessaire.

Deux types d'approches existent dans la littérature. Elles peuvent être utilisés successivement :

- Des méthodes de conception, qui gèrent à la fois la phase d'ingénierie et la phase de révision. La seconde phase consiste à remettre en question, les fondements du comportement des agents, en suivant des modèles sociaux ([Boissier *et al.*, 2004b]).
- Des approches par ajustement des paramètres du SMA, qui recherchent les valeurs de paramètres optimales d'un système déjà conçu mais pas encore calibré pour assurer un comportement souhaité.

III.1.1 Méthodes de conception

Une analyse de ces méthodes est donnée dans [Arlabosse *et al.*, 2004], et [Campagne, 2005] ournit une liste de Quarante en en détaillant neuf d'entre elles. Leur but est de définir entièrement le comportement des agents, sans cadre formel qui délimite leurs capacités : le concepteur du système est libre de créer ses agents comme il l'entend. On comprend qu'elles s'appliquent plus naturellement à des SMA cognitifs que réactifs.

Dans les méthodes de conception, un «intervenant humain» contribue à déterminer et à corriger les comportements des agents grâce à son expérience personnelle. Le lien entre le niveau local de définition du SMA et son comportement global est défini par un cerveau humain, ce qui rend ces méthodes faibles car un humain n'a pas la capacité d'effectuer de nombreux tests et calculs comme une machine, et ne peut pas assurer de penser à toutes les améliorations possibles. Une autre limite est qu'il présuppose qu'il existe un moyen de toujours assurer un comportement désiré. Ces méthodes ne considèrent pas le cas où le système subit des perturbations extérieures.

III.1.2 Approches par ajustement des paramètres

Un champ entier de la recherche sur les SMA est lié à la calibration d'un système existant ([Culioli, 1994]), le but étant l'obtention de valeurs de paramètres du système qui permettent d'optimiser l'une de ses caractéristiques.

Cette optimisation se fait expérimentalement, en explorant l'espace des paramètres du SMA à calibrer, et en retenant la meilleure combinaison de valeurs. Typiquement, il s'agit de trouver des valeurs qui permettent d'obtenir un comportement global quelles que soient les conditions initiales. La section suivante présente quelques approches :

III.1.2.1 Utiliser la connaissance du système

L'article [Fehler 04] propose des techniques qui exploitent la connaissance qu'on a d'un système pour calibrer les paramètres d'un SMA qui le simule. Trois problèmes propres à la calibration d'un SMA sont donnés pour justifier le recours à ces techniques :

1. Le nombre de paramètres à calibrer est particulièrement élevé dans un SMA.
2. Le comportement global dépend des paramètres de façon complexe, à cause des interactions entre agents.
3. L'évolution d'une simulation multi-agents présente un important coût de calcul.

Dans cette proposition, l'espace des paramètres à calibrer est décomposé de manière à résoudre des sous-objectifs, choisis en fonction de la connaissance du système. Le problème de calibration est successivement résolu pour des sous-espaces de plus faible dimension ce qui limite l'explosion combinatoire de l'espace des paramètres.

III.1.2.2 Explorer les paramètres en parallèle

Une autre solution de calibrage proposée dans [Calvez et Hutzler, 2005] se base sur un principe similaire à la proposition précédente, à savoir diviser l'espace des paramètres.

Il s'agit à la fois d'explorer en parallèle cet espace par les différents agents du SMA¹, et de déterminer les zones à explorer en traitant indépendamment chaque paramètre. Cela permet d'optimiser un paramètre en fonction uniquement des meilleures valeurs trouvées pour les autres. Cette approche est justifiée par le nombre élevé de paramètres à calibrer dans le SMA.

III.1.3 Limite des approches par construction

Le but principal des approches par construction est de créer un SMA tout en assurant qu'il présente un comportement global désiré, Alors qu'un SMA présente de nombreuses sources d'imprévu : son évolution complexe peut être stochastique, soumis à des perturbations et analytiquement imprévisible.

Quelle que soit la méthode de construction choisi, il est difficile de garantir analytiquement qu'un SMA présente un comportement désiré comme le signale [De Wolf *et al.*, 2005]), dès que l'une de ces sources d'imprévisibilité entre en jeu, même si la construction d'un SMA permet d'atteindre les spécifications dans la plupart des cas, elle n'indique pas quoi faire lorsqu'une évolution particulière l'amène dans un comportement non désiré. Comme ce comportement est par définition stable, la méthode de construction aura échoué à remplir son rôle de maîtrise du comportement.

III.2 Approches par contrôle

Le contrôle d'un système consiste à agir au cours de son évolution afin de diriger son comportement. C'est une notion formelle issue de l'étude des systèmes dynamiques.

Il existe des approches empiriques qui se réclament le contrôle pour diriger le comportement des SMA, appliquées soit au niveau local, soit au niveau global du système.

D'après [Drogoul *et al.*, 2004], l'étude de l'évolution d'un SMA réactif peut être exprimée dans le paradigme des systèmes dynamiques. Selon [Manneville, 2001], un système dynamique est un système qui évolue au cours du temps de façon déterministe donnée par

1. Cela n'est donc possible que pour l'optimisation de paramètres propres aux agents.

en ensemble de règles F ou un système d'équations différentielles dans le cas continu.

L'état du système à tout instant est donné par les valeurs de ses variables d'état X . Les règles d'évolution s'appliquent à ces variables d'état :

$$X_{t+1} = F(X_t) \text{ dans le cas discret, } = F(X_t) \text{ et } \frac{dX}{dt} = F(X)$$

L'état du système peut être représenté dans son espace des phases X , dont chaque dimension correspond à l'une des variables d'état. L'évolution du système est alors équivalente à une trajectoire dans cet espace. Le système est appelé linéaire si la fonction F est linéaire, et il est appelé non linéaire dans le cas contraire.

III.2.1 Contrôle d'un système dynamique

Un problème de contrôle est défini par trois données :

- La cible, ou consigne, à atteindre et maintenir,
- Des moyens d'action sur le système : ses paramètres ou commandes,
- Des moyens d'observation du système.

III.2.1.1 Objectif du contrôle

Le problème du contrôle recouvre plusieurs objectifs différents :

Le premier est l'étude de la contrôlabilité, c'est-à-dire vérifier s'il est possible d'atteindre la cible avec les moyens d'action à disposition.

Le second objectif est l'optimisation du contrôle. Si l'on se donne une mesure de performances, qui représente par exemple le temps nécessaire pour atteindre la cible, il s'agit de trouver une suite de valeurs, qui optimise cette mesure.

Une étape clef, en général passée sous silence car peu difficile, est de comparer la connaissance issue de l'observation de l'état courant du système à la cible ou à la consigne. Cela sert d'une part à évaluer le contrôle, en sachant quand la cible est atteinte, et d'autre

part à guider les actions de contrôle en déterminant l'état du système à tout instant, en fonction de l'écart entre l'état et la cible.

Pour trouver la suite de valeurs qui répond à l'objectif de contrôle fixé, trois questions se posent, bien que seule la dernière pose réellement problème en général :

- trouver l'information utile pour le contrôle,
- sélectionner les types de commandes ou les paramètres qui permettent de contrôler,
- choisir la commande en fonction de l'information retenue.

III.2.2 Approche de contrôle au niveau global

On trouve une proposition de contrôle au niveau global d'un SMA dans [Campagne *et al.*, 2004b] et [Campagne *et al.*, 2004a], [Campagne, 2005]. Il s'agit de l'unique travail de recherche qui traite explicitement ce problème trouvé dans la littérature, c'est pourquoi nous le développons tout particulièrement.

III.2.2.1 Intuition du contrôle

L'auteur indique que « les recherches [...] qui s'intéressent à la construction [des SMA] se concentrent sur les méthodes de type génie logiciel et les méthodes d'analyse, mais le problème du contrôle n'est pas vraiment abordé. » Il ajoute que le contrôle est utile lorsqu'on constate qu'une méthode uniquement auto-adaptative n'est pas envisageable pour arriver au même résultat, c'est-à-dire si le comportement du SMA est susceptible de subir des dérives qui ne pourront pas être réparées par le seul comportement des agents. Il propose donc de réallser un système dont le comportement est approxlmativement correct, puis de moduler ce comportement par un contrôle extérieur.

IV Problématique et objectif ciblé

Notre travail se situe dans le cadre des systèmes multi-agents (SMA) réactifs, et traite le contrôle et la maîtrise du comportement global du système.

L'objectif étant de construire des systèmes robustes, dynamiques, capables de s'exécuter seuls et d'adapter leurs comportements face à des circonstances variées. En fait, le comportement des SMA découle des multiples interactions entre les entités qui le composent qui sont définis localement. Ce comportement est souvent difficile à expliquer.

Notre travail est donc la conception de systèmes multi-agents adaptatif à fonctionnalité émergente. Comme il est composé de simple agents réactifs qui n'ont qu'une perception local, la réalisation de différents tests permettra la découverte des réponses du système au problèmes du dynamisme de l'environnement, ainsi que le contrôle de ce dernier. Le but est de mieux comprendre et de diriger le comportement d'un SMA réactif, en particulier lorsqu'il n'est pas possible d'assurer le comportement optimal souhaité lors de la construction du système.

Pour mieux comprendre le travail, le choix d'une application est incontournable. Le domaine de l'imagerie a été ciblé et l'axe de recherche s'est orienté vers la segmentation d'image vu que les recherches dans ce dernier restent toujours ouvertes.

V Conclusion

Un phénomène émergent dans un SMA réactif est un phénomène global, difficilement prévisible à partir de la connaissance du niveau local du système, interprété par un observateur extérieur. Il est global en espace, car il fait intervenir collectivement plusieurs agents, et en temps, puisque sa dynamique plus lente le fait apparaître stable.

La problématique de maîtriser le comportement global d'un SMA fait l'objet de notre étude et le choix de notre application s'est orienté vers la segmentation d'image. ainsi, il est impérative d'introduire les notions de base dans ce domaine. Au début, nous allons citer et comparer les différents techniques basés sur les algorithmes de segmentation

d'image, ainsi que les différentes combinaisons possibles entre elles.

Chapitre III :

Segmentation d'image

Chapitre III

Segmentation d'image

I Introduction

Dans une image, les informations les plus directes à distinguer sont les contours (variation rapide et claire d'intensité) et les régions (homogénéité des zones de l'image). C'est ainsi que les méthodes de segmentation ont été classées primitivement en deux grandes approches : les approches contours et les approches régions.

Cependant, l'utilisation séparée de ces deux approches a montré rapidement leurs insuffisances. Ce qui a encouragé l'apparition d'autres techniques plus évoluées : approches coopératives Région/Contour.

Dans ce chapitre, nous présentons les principales approches contour et région les plus utilisées, ensuite nous donnerons le principe des méthodes coopératives.

II Segmentation d'image

II.1 Définition

Segmenter une image signifie décrire son contenu d'une façon plus représentative et plus exploitable qu'un ensemble de points [Kermad, 1997].

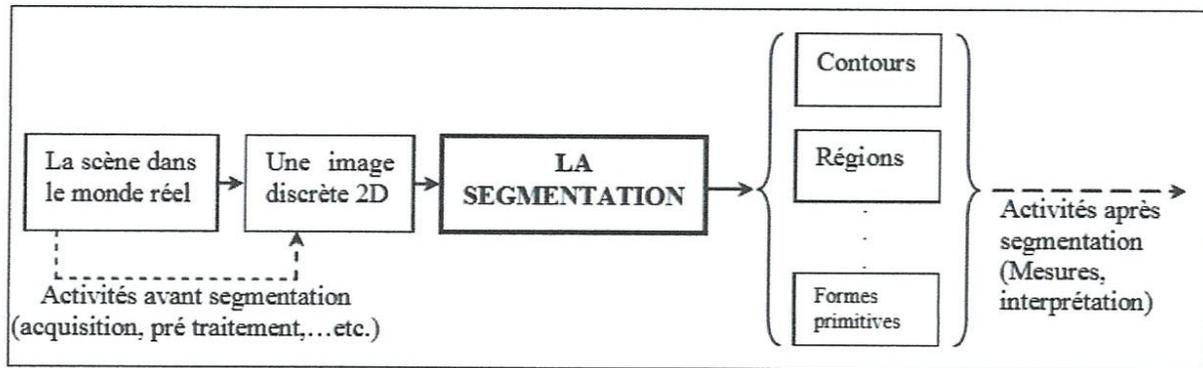


FIGURE III.1 La segmentation dans un processus d'analyse d'image.[Chan et Shen, 2005]

phrase Introduction point suivants

II.2 Domaines d'application

La segmentation d'image est fréquemment utilisé dans plusieurs domaines Certaines des applications pratiques de la segmentation d'image sont :

- Simulation de chirurgie virtuelle.
- Navigation intra-chirurgie.
- Détection d'objets.
- Détection de piétons.
- Détection facial.
- Détection de la lumière de frein.
- Localiser des objets dans des images satellite (routes, forêts, cultures, etc.).
- Tâches de reconnaissance.
- Reconnaissance de visage.
- La reconnaissance d'empreintes digitales.
- Reconnaissance de l'iris.
- Vidéosurveillance

II.3 Différentes approches de segmentation

Beaucoup de méthodes de segmentation sont basées sur deux propriétés de base des pixels par rapport à leur voisinage local : discontinuité et similitude. La discontinuité de pixels est utilisée par les méthodes de segmentation par approche contours, tandis que la similitude de pixels est utilisée par les méthodes de segmentation par approche région. Dans la première approche, on s'intéresse aux frontières des régions et dans la deuxième on s'intéresse au contenu des régions (voir figure III.2).

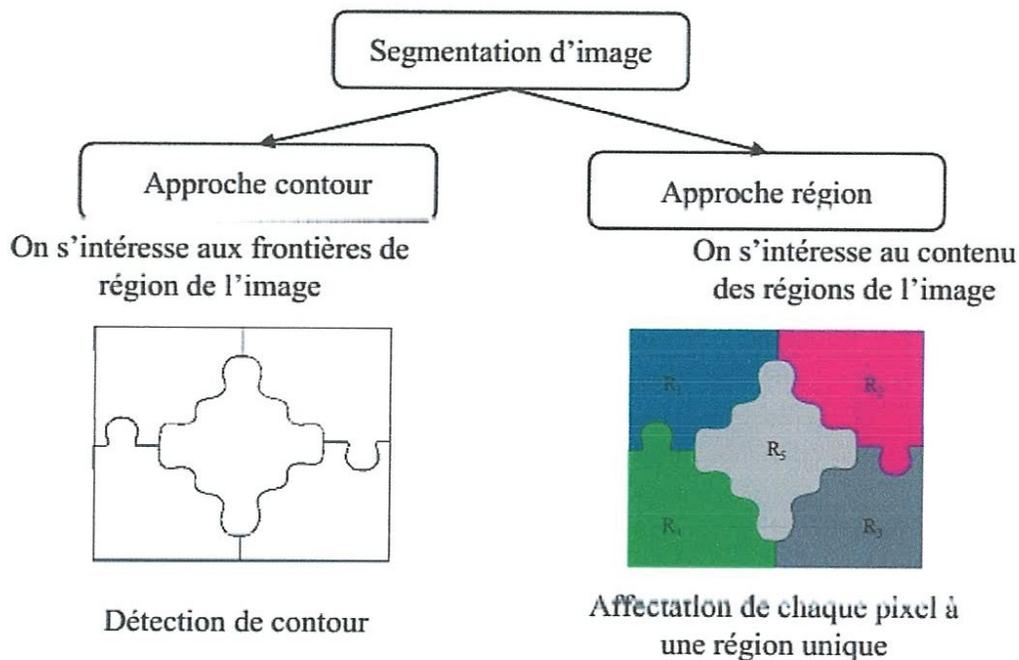


FIGURE III.2 Les approches de segmentation d'image.

Dans ce qui suit nous allons détailler chacune de ces approches.

II.3.1 Approches Contour

II.3.1.1 Principe

L'approche contour consiste à identifier les changements entre les régions. En général, un élément du contour est un point de l'image appartenant à la frontière de deux ou plusieurs objets ayant des niveaux de gris différents [Horaud et Monga, 1995].

Le processus classique de détecteur de contours suit les étapes suivantes. [Meliani, 2012] :



FIGURE III.3 Le processus classique de détecteur de contours.

La mise en évidence des contours : C'est une première étape important, car elle permet de marquer les contours. Cette mise en évidence s'obtient par exemple par une différentiation (dérivation) de l'image.

La réduction des contours : Elle agit de manière à ce que les contours mis en évidence ne fassent qu'un seul pixel d'épaisseur.

La binarisation des contours : Elle permet d'obtenir une image contours binaire. La binarisation est une étape clef. Elle définit quels sont les contours considérés comme importants et ceux qui sont rejeté.

La description des contours : Elle permet d'organiser les points de contours en structures simples tels les segments de droites, arcs de cercle, etc., et de détecter les contours fermés afin de les séparer.

La détection des contours dans les images a débuté d'une façon extrêmement empirique par des opérateurs locaux qui se basent généralement sur des méthodes dérivatives. Par la suite, d'autres approches analytiques ont été mise au point afin d'obtenir des contours plus significatifs [Bloch *et al.*, 2004].

II.3.1.2 Techniques d'extraction de contours

De nombreuses techniques d'extraction de contours existent dans la littérature. Elles peuvent être classées comme suit :

Les algorithmes basés sur le gradient (méthodes dérivatives du premier ordre), Laplacien (méthodes dérivatives du second ordre). 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 résultat est une image binaire constituée de points de contours et de points non-contours.

Les méthodes analytiques pour la détection de contours ils sont apparues dans les années quatre-vingts, elles ont permis une meilleure compréhension des conditions d'une bonne détection de contours, ce qui a conduit à des détecteurs de bonne qualité. Les approches les plus intéressantes, et les plus utilisés sont ; le filtre de 'Canny', le filtre de 'Deriche' et le filtre de 'Shen et Castan', et qui sont des filtres monodimensionnels inspirés par les méthodes du traitement de signal.

II.3.1.3 Limites de segmentation par contour

Les principales limites des méthodes de détection de contour sont les suivantes [Acharya et Ray, 2005] :

- Les contours extraits selon les méthodes classiques souvent ne correspondent pas nécessairement à la limite des objets. Dans de nombreux images de basse qualité, quelques-unes des méthodes produisent des faux contours.
- Les techniques de détection de contour dépendent de l'information contenue dans le voisinage local de l'image. Il n'y a pas d'information globale.
- Après l'extraction des points de contours, ces derniers sont reliés afin de déterminer les frontières. Le processus de fermeture des contours peut parfois conduire à des discontinuités et des lacunes dans l'image.
- Il est souvent difficile d'identifier et de classer les contours parasites.

II.3.2 Approches Région

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).

II.3.2.1 Principe

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, homogénéité ...). Il existe plusieurs méthodes :

Segmentation par croissance de régions : description

Segmentation par fusion de régions .

Segmentation par division de régions .

Segmentation par division-fusion .

Pour obtenir une bonne segmentation par approche région, il existe quelques règles à suivre [Haralick et Shapiro, 1985] :

- Les régions doivent être uniformes et homogènes par rapport à certaines caractéristiques (niveau de gris, écart type, gradient).
- Leurs intérieurs doivent être simple et sans beaucoup de petits trous (des parties de région non segmentés).
- Les régions adjacentes doivent avoir des valeurs très différentes par rapport à la caractéristique prise en compte dans la segmentation.
- Les limites de chaque région doivent être simples et spatialement précises.

II.3.2.2 Limites de segmentation par région

Les inconvénients de la segmentation par région se situent à trois niveaux [Belhamra et Melzi, 2010] :

- Les régions obtenues ne correspondent pas, dans tous les cas, aux objets représentés dans l'image.
- Les limites des régions obtenues sont généralement imprécises et ne coïncident pas exactement avec les limites des objets de l'image.
- La difficulté d'identifier les critères pour agréger les pixels ou pour fusionner et diviser les régions.

II.3.3 Seuillage

Dans les deux approches on relève un problème majeur qui est le choix pour la détection de contours et de régions, dans ce qui suit nous allons présenter les deux formes de seuillage existants :

a) Seuillage global

Le principe du seuillage global est d'utiliser une valeur « seuil » à partir de laquelle on peut choisir la classe d'appartenance du pixel. La transformée utilisée peut s'écrire ainsi [Gupta *et al.*, 2007] :

$$\forall i, j \in N * M \quad I(i, j) = \begin{cases} 1 & \text{si } f(i, j) \geq S, \\ 0 & \text{sinon} \end{cases} \quad (\text{III.1})$$

Avec

- $N * M$: nombre de colonnes et de lignes de l'image ;
- I : image binarisée ;

- f : valeur fonction de l'image d'origine ;
- S : seuil de binarisation.

Il existe plusieurs méthodes pour calculer le seuil global les plus utilisés sont :

- Méthode d'Otsu [Otsu, 1975] ;
- Méthode de Kapur [Kapur *et al.*, 1985] ;
- Méthode de Cheng et Chen [Cheng *et al.*, 1998] ;
- Méthode de Li-Lee [Li et Lee, 1993] ;

b) Seuillage local

Le principe du seuillage local est d'utiliser une étude localisée autour du pixel pour déterminer quel seuil approprié. Pour réaliser cette étude locale, les techniques utilisent une fenêtre d'étude centrée sur le pixel ciblé. Cette fenêtre peut avoir différentes tailles. Le premier à proposer une technique donnant de bons résultats fut Bernsen en 1986 [Bernsen, 1986].

Mathématiquement, le calcul du seuil peut s'écrire ainsi :

$$S(i, j) = (\max(i, j) + \min(i, j))/2 \quad (\text{III.2})$$

Avec :

- $S(i, j)$: seuil à appliquer pour le point i, j ;
- $\text{Max}(i, j)$: valeur du niveau de gris maximal dans une fenêtre centré en (i, j) de taille $N + M$;
- $\text{Min}(i, j)$: valeur du niveau de gris minimal dans une fenêtre centré en (i, j) de taille $N * M$;
- N et M appartenant à N .

Il existe plusieurs méthodes pour calculer le seuil local, parmi le :

- Méthode de Niblack [Kefali *et al.*, 2009];
- Méthode de Sauvola [Sauvola *et al.*, 1997];
- Méthode de Wolf [Wolf *et al.*, 2002];
- Méthode Nick [Khurshid *et al.*, 2009];
- Segmentation hiérarchique floue [Gadi et Benslimane, 2000].

Bien que chaque approche ait ses avantages, elle présente également des inconvénients. Cela a poussé les chercheurs à s'intéresser aux approches coopératives. Dans ce type d'approche, la complémentarité des deux types de segmentation (approche contour et approche région) est exploitée.

La segmentation coopérative est une segmentation dans laquelle on corrèle l'extraction de plusieurs types de primitives ou d'informations (le gradient, écart type). Elle combine les avantages de chacune des méthodes prise séparément : la précision et la rapidité d'une segmentation contour, et la fermeture des frontières et la densité de l'information extraite, d'une segmentation région.

II.4 Formes de coopération

L'intégration des deux types de segmentation (contour, région) peut être réalisée à différents niveaux, qui peuvent être catalogués en trois classes : coopération séquentielle, coopération des résultats et coopération mutuelle [Sebari et He, 2007].

II.4.1 Coopération séquentielle

Dans cette méthode l'une des techniques de segmentation est réalisée en premier (région ou contour), et son résultat va être exploité par l'autre technique pour renforcer la définition des critères ou des paramètres de la segmentation.

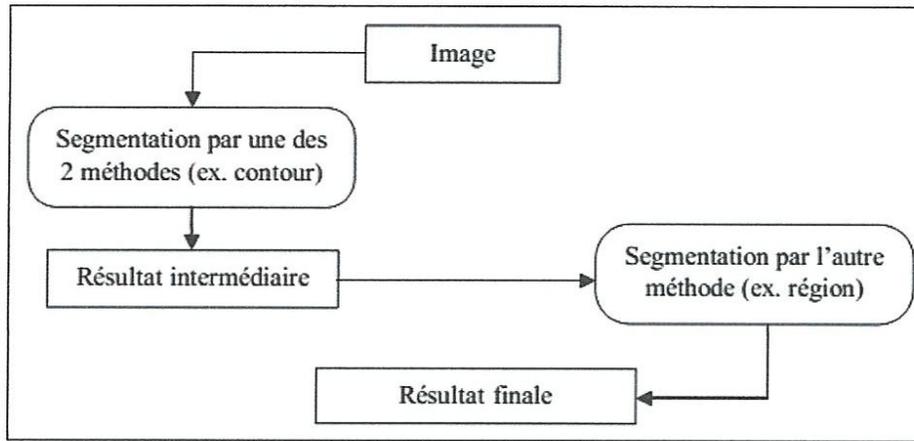


FIGURE III.4 Coopération séquentielle [Sebari et He, 2007].

La coopération la plus intuitive de ce genre de coopération est l'utilisation d'une carte de contours déjà construite pour guider un processus d'élaboration de régions.

Un autre exemple est d'utiliser des régions déjà construites dans un processus contour (i.e contour actif) où les frontières des régions seront les contours initiales qui vont être évoluées par la force d'attraction des points de fort gradient.

II.4.2 Coopération des résultats

Les deux types de segmentation seront réalisés indépendamment, la coopération concernera leurs résultats qui seront intégrés afin d'atteindre une meilleure segmentation.

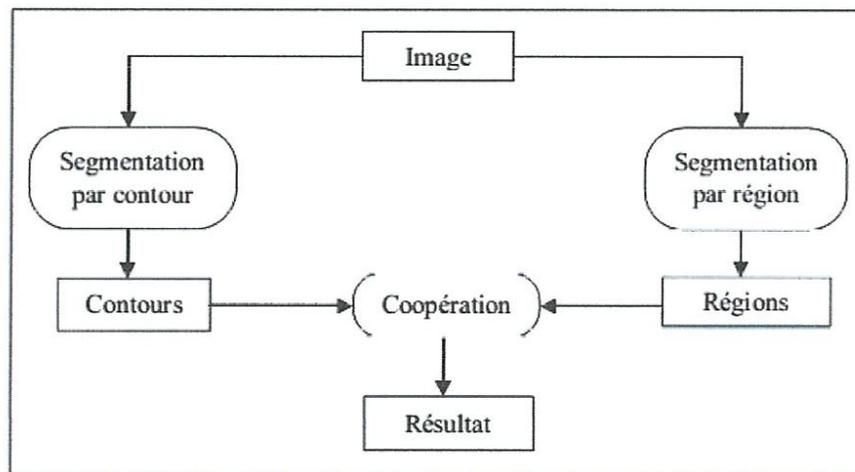


FIGURE III.5 Coopération des résultats [Sebari et He, 2007].

II.4.3 Coopération mutuelle

Les deux types de segmentations coopéreront mutuellement au cours de leur processus d'exécution.

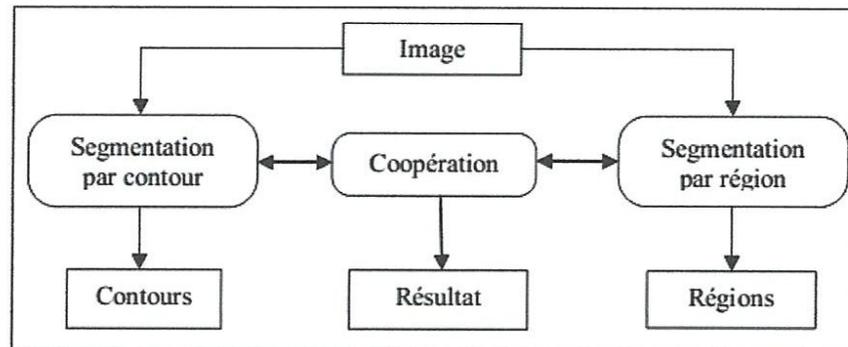


FIGURE III.6 Coopération mutuelle [Sebari et He, 2007].

La coopération mutuelle présente des avantages par rapport aux deux type de coopération précédentes. Elle permet de guider la progression d'un processus particulier et de participer à l'enrichissement du résultat global. Néanmoins, il existe relativement peu d'approches proposant un tel type de coopération pour la segmentation. Cela est dû aux difficultés rencontrés lors de la mise en oeuvre d'une telle approche.

III Travaux Connexes

Les travaux les plus anciennes sont ceux R.Bajcsy décrits dans [Bajcsy *et al.*, 1986] et [Anderson *et al.*, 1987]. L'algorithme décrit est un « système expert de segmentation » qui modifie au cours de son déroulement ses paramètres de contrôle de sorte que les résultats des deux segmentations soient globalement compatibles.

Chacun des deux modules (détecteurs de contours et de régions) peut être amené à fonctionner un nombre important de fois jusqu'à convergence, ce qui est difficilement compatible avec une implémentation future temps réel sur système autonome. De plus le système expert nécessite des connaissances, ce qui rend la segmentation trop spécifique à un type restreint d'applications.

La méthode proposée par J.Benois [Benois et Barba, 1992] reprend un schéma algorithmique de coopération entre les informations issues des contours et des régions. L'originalité réside dans l'implantation de techniques de morphologie mathématique pour la partie "détection de régions". En effet, des points particuliers du squelette (construit à partir des points de contours) seront centres de croissance dans une procédure de croissance de régions.

Le détecteur de facettes proposé par H.Chabbi [Chabbi, 1993] permet une fusion élégante des informations issues des contours et des régions. Les segments de contours extraits les premiers permettent de guider correctement la croissance de régions, en influant sur le seuil contrôlant la fusion des régions adjacentes. Ensuite, les segments de contours sont regroupés en facettes. Une facette comprend : la région et les segments de contour qui la bordent. Bien que le principe soit simple à énoncer, le nombre de cas à prendre en compte est impressionnant.

Parmi les travaux qui traitent la segmentation d'images en utilisant les agents on peut citer :

Bellet [Bellet *et al.*, 1995] propose un système multi-agents spécialisé pour la segmentation d'images tel que certains agents exécutent une croissance de régions alors que d'autres exécutent un processus de détection de contours. La coopération entre les deux types d'agents est dynamique et permet de propager une information quand elle devient nécessaire pour une prise de décision.

[Liu et Tang, 1999] ont présenté un système basé sur des agents réactifs pour la segmentation d'IRM du cerveau. 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 trouver des pixels d'une région homogène spécifique, créent des agents clones à l'intérieur de leurs régions voisines. Ces derniers sont initialisés de telle sorte qu'ils deviennent capables de reconnaître les pixels de la même région, pour laquelle ils ont été créés. Les auteurs affirment que leur système est plus robuste et plus efficace que les algorithmes classiques d'éclatement et de fusion de régions. Cependant, dans ce système les agents n'interagissent pas directement entre eux, et n'agissent pas sur l'image. Leurs actions dépendent uniquement de leur perception locale de l'image. Néanmoins, ils sont créés et placés de telle sorte qu'ils soient susceptibles de trouver plus de pixels de la région pour

laquelle ils ont été créés.

Pour le même type d'images, [Richard *et al.*, 2004] ont proposé une architecture hiérarchique d'agents situés et coopératifs pour la segmentation d'images. Trois types d'agents ont été utilisés : agent de contrôle global, agent de contrôle local, et agent dédié au tissu. Le rôle de l'agent de contrôle global est de partitionner le volume de données en territoires adjacents et d'affecter à chaque territoire un agent de contrôle local. Le rôle de ce dernier est de créer les agents dédiés au tissu, dont le rôle est d'effectuer un accroissement de région à l'intérieur du volume local. Les paramètres de distribution des données sont mis à jour par coopération entre les agents voisins. Par l'utilisation de différents types d'agents, le système proposé a pris en compte les traitements d'images de bas niveau, ainsi que le contrôle sur les tâches de segmentation. Cependant, il résulte des patrons complexes d'interaction dont la gestion est épineuse et coûteuse.

[Settache, 2002] a utilisé des agents région et contour qui coopèrent pour la segmentation d'IRM. Il a utilisé une carte de région (quadtree) et une carte de contour (Deriche) initiaux, pour démarrer un processus de coopération agent pour le raccordement des contours et la fusion des régions.

[Idir *et al.*, 2005] à utiliser la structure d'une pyramide duale irrégulière pour réaliser son système de segmentation. Elle a utilisé une carte de contour pré-calculé (filtre de Canny) afin de guider le processus de fusion de régions dans la pyramide.

[Mobahi *et al.*, 2006] se sont inspirés des comportements de groupes d'animaux, notamment des bancs de poissons, pour proposer un système multi-agent pour l'initialisation des contours actifs. Les agents se déplacent selon un modèle cinématique basé sur 3 forces ; à savoir :

1. la force de charge qui tire un agent vers les points de contour ;
2. la force d'évitement permettant de garder les agents distants les uns des autres ;
3. la force d'opportunisme, permettant aux agents d'explorer l'image afin de se trouver dans les régions d'intérêt (contours).

[Benamrane et Nassane, 2007] proposer une approche pour segmentation d'image

médicale basée sur un système multi-agent. L'idée consiste à fusionner les régions suivantes plusieurs critères et avec une population massive d'agents situés qui coopèrent, négocient l'aide de protocoles d'interaction et de communiquer en faisant passer des messages asynchrones.

[Redjimi et Amri, 2012] proposée une méthode consiste à coupé l'espace de l'image à traiter dans un ensemble de sous-espaces (partitions de l'image), dans lequel plusieurs agents sont créés pour détecter les contours des objets puis de les suivre (ces agents sont appelés détecteurs - suiveurs agents). Ces agents adapter un algorithme très efficace de détection et de suivre le plan selon les caractéristiques de la région qu'ils évoluent. Les informations sont transmises à des niveaux d'agents de supervision (agents partitions) qui prennent soin qu'ils à collecter les informations émises par le détecteur d'agents, Une mise en œuvre de cette approche par l'utilisation de Madkit.

[Meddeber, 2013] a proposé une approche permettant un suivi rapide d'un objet dans une séquence d'images médicales. L'approche proposée définit trois agents : Le premier utilise une analyse multi-résolution pyramidale afin de décomposer les images de la séquence en des résolutions multiples. Le deuxième agent détecte l'objet à suivre dans les images à basse résolution par les contours actifs paramétriques. Le troisième agent réalise la projection des contours détectés dans les images à basse résolution vers les images à haute résolution.

[Djerou et Batouche, 2013] a proposé un système à base d'agents réactifs capable de segmenter des images en niveau de gris. Dans cette approche l'auteur a combiné le mécanisme de stigmergie qualitative observé chez les insectes sociaux avec un autre type de comportement d'essaim qui est l'Optimisation par l'Essaim de Particules (OEP). Les agents se déplacent sur l'image en construisant des régions homogènes ; par le regroupement des pixels connexes présentant une certaine similarité, sous une même marque. La formation d'une région commence à partir d'un pixel appelé pixel germe, puis sa croissance suit un processus itératif de regroupement des pixels voisins et connexes vérifiant un critère d'homogénéité. Le choix du pixel germe et l'estimation locale de l'homogénéité de la région sont traités par l'OEP, cependant, le contrôle du processus de construction de la région est réalisé par le mécanisme stigmergique.

La plupart des méthodes présente des difficultés, l'utilisation des méthodes coopératives est relativement rudimentaire, malgré les nombreux avantages présentés par ces méthodes.

IV Conclusion

Dans ce chapitre, nous avons présenter les différentes approches classiques de segmentation d'image et nous avons montré la difficulté voir l'impossibilité d'avoir un résultat proche à une analyse humaine, car la quasi-totalité de ces approche sont conçus pour travailler dans des conditions parfaits. Nous avons exposer les travaux connexes relatives au domaine de la segmentation d'images tout en soulignant l'utilisation des systèmes multi-agents.

Dans le chapitre suivant, nous allons présenter le système proposé pour une segmentation coopérative d'image avec contrôle et maîtrise du comportement global.

Chapitre IV :

Conception et Implémentation

Chapitre IV

Conception et implémentation de notre système

I Introduction

Dans le présent chapitre, nous allons présenter le modèle proposé qui consiste à effectuer une segmentation d'image à l'aide d'une coopération entre deux SMA, le premier va effectuer une segmentation contour et l'autre va effectuer une segmentation région, les deux systèmes vont coopérer indirectement pour atteindre le résultat amélioré.

II Description et objectifs de l'application

II.1 Choix de l'application

La conception d'un système capable d'interpréter automatiquement les images « quelque soit leur type, format ou nature » d'une manière similaire à une interprétation humaine constitue l'un des défis les plus populaires dans les travaux de recherche actuels qui focalisent sur le traitement d'image.

Le passage par l'opération de segmentation est impératif pour y arriver à implémenter

un tel système, ce qui constitue un tâche qui n'est pas facile, dû aux raisons qui suit :

- Il est difficile de choisir un seuil adapté à toutes les images vu la différence de couleur d'une image à une autre et d'une partie à une autre au sein d'une même image.
- L'influence des conditions d'acquisitions sur la qualité d'image à segmenter.
- L'absence d'un algorithme complet et raisonnable par rapport à la croissance d'utilisation d'images comme source d'information.
- L'information initiale offerte par l'image est limitée (position et intensité des pixels) par rapport aux résultats attendus par un algorithme de segmentation.

La présence de tels raisons, a laissé les portes grande ouvertes sur la recherche dans ce domaine.

II.2 Objectif de l'application

La méthode que nous proposons consiste à utiliser la coopération mutuelle entre l'approche contour et l'approche région. Cette méthode présente à la fois l'avantage d'aboutir à un résultat de segmentation plus précis et plus fidèle que celui obtenu à l'aide des autres techniques tout en offrant une modélisation distribuée approprié à l'utilisation des systèmes multi-agents adaptatifs ciblés dans notre travail. Toutefois, elle soulève une importante difficulté d'implémentation.

III Conception

Notre application est constituée essentiellement de deux systèmes multi-agents, l'un dédié à la sementation contour et l'autre à la segmentation région. Ces deux systèmes fonctionnent indépendamment l'un de l'autre tout en partageant un Environment commun.

La maîtrise du système global se fait par contrôle des deux systèmes qui s'ajustent mutuellement pour aboutir à des résultats améliorés.

III.1 Modèle proposé pour la segmentation coopérative

Notre idée consiste à décomposer le processus de segmentation en deux étapes principales qui s'exécutent en parallèle continuellement jusqu'à aboutir à un critère d'arrêt.

La figure IV.1 présente l'architecture du modèle proposé :

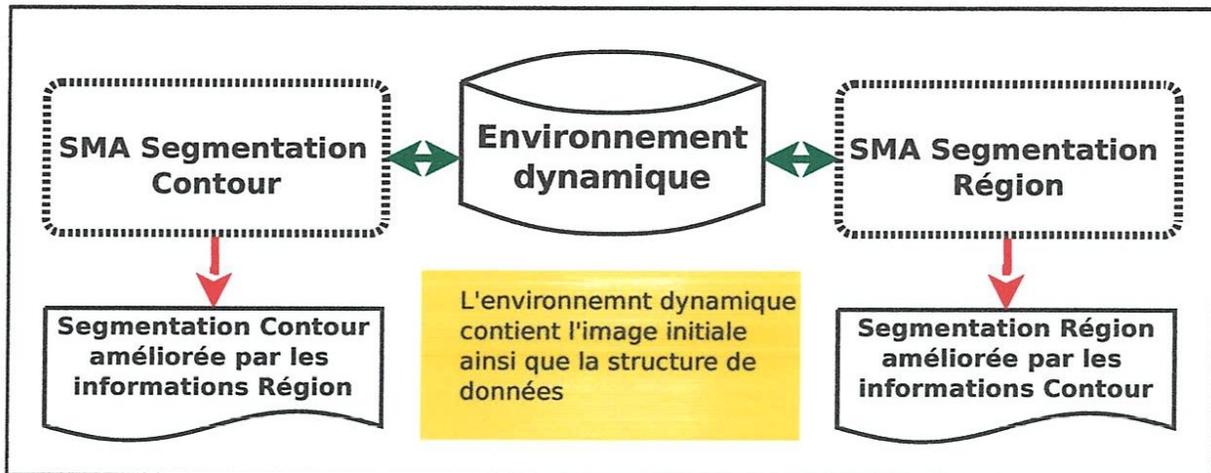


FIGURE IV.1 Modèle proposé pour la segmentation coopérative

III.2 Modélisation de problème

Les éléments essentiels qui constituent notre modèle sont :

- L'environnement.
- Les Agents.

III.2.1 Modélisation de l'environnement

L'ensemble des pixels constituant l'image source représente l'environnement sur lequel les agents perçoivent et exécutent leur comportement. En plus de l'image source notre environnement est doté d'une structure de données partagée entre les deux SMA. Cette dernière contient principalement les champs suivants :

idBloc, seuilLocal, seuilHaut, seuilBas, homogeneite, visiteContour, visiteRegion, contour, candidatContour, region, candidatRegion, idRegion.

L'image va être divisée en petites zones, permettant le calcul d'un seuil homogène local.

Pour les valeurs de seuil utilisé pour la détection de contour, nous avons utiliser deux seuils qu'on va appeler *seuilHaut* et *seuilBas* qui sont calculés lors de l'initialisation de l'environnement à partir du *seuilLocal* selon les formules suivantes.

- $seuilHaut = seuilLocal * 0.03$
- $seuilBas = seuilLocal * 0.02$

Ces deux formules sont issues des différentes essais effectués pendant la phase de développement là ou on a constater que ces valeurs donnent les meilleurs résultats

III.2.2 Modélisation d'agents

Ce sont des agents réactifs qui interagissent entre eux en réponse à des stimuli de leur environnement et qui possèdent un nombre restreint de fonctionnalités. Ces fonctionnalités changent selon l'information locale disponible (intensité lumineuse, pixels voisins).

III.2.2.1 Agents contour

En interagissant avec l'environnement cette entité suit les règles comportementales suivantes :

- si *pixelContour()* alors Marquer le *Pixel* comme *contour*
- sinon si *pixelCandidatContour()* alors faire :
 - marquer le pixel *candidatContour*
 - si *existeVoisinContour()* marquer *contour*

- si *pixelCandidatContour()* alors :
 - si le pixel est marqué *region* démarquer *candidatContour*
 - sinon marquer le pixel *contour* et démarquer *candidatContour*
- si nombre *candidatContour* est égale à 0 arrêter l'agent.

La fonction *pixelContour()* permet de tester le changement brutal de l'intensité lumineuse au niveau du pixel (fonction de dissimilarité) et decide si on doit marquer le pixel *contour* ou pas, en comparant avec la valeur *seuilHaut*.

La fonction *pixelCandidatContour()* permet de tester le changement brutal de l'intensité lumineuse au niveau du pixel (fonction de dissimilarité) et decide si on doit marquer le pixel *candidatContour* ou pas, en comparant avec la valeur *seuilBas*.

III.2.2.2 Agents région

En interagissant avec l'environnement cette entité suit les règles comportementales suivantes :

- marquer tous les pixels *candidatRegion*
- si *pixelRegion()* alors Marquer le pixel *Region* et démarquer le pixel *CandidatRegion*.
- sinon :
 - si le pixel est marqué *contour* démarquer *candidatRegion*
 - sinon marquer le pixel *region* et démarquer *candidatRegion*
- si nombre *candidatRegion* est égale à 0 arrêter l'agent.

La fonction *pixelRegion()* permet de tester si le pixel fait partie d'une région ou pas, en testant la valeur *homogeneite*.

IV Réalisation :

IV.1 Environnement de développement (NetBeans)

Un environnement de développement intégré (Integrated Development Environment ou IDE) est indispensable pour un programmeur, car il offre des outils permettant à ce dernier d'écrire, de compiler et d'exécuter des programmes. Un IDE fournit aussi un utilitaire d'Aide qui décrit tous les éléments du langage et permet de trouver et de corriger plus facilement les erreurs dans les programmes. Alors que la plupart des IDE sont très chers, il en existe un excellent, gratuit : NetBeans.

NetBeans est un IDE, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native, ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

L'environnement de base comprend les fonctions générales suivantes⁴ :

- configuration et gestion de l'interface graphique des utilisateurs,
- support de différents langages de programmation,
- traitement du code source (édition, navigation, formatage, inspection ..),
- fonctions d'import/export depuis et vers d'autres IDE, tels qu'Eclipse ou JBuilder,
- accès et gestion de bases de données, serveurs Web, ressources partagées, gestion de tâches (à faire, suivi ...),

- documentation intégrée.

IV.2 Présentation de l'application

L'interface graphique de l'application regroupe un ensemble d'options qui permettent l'accès facile aux différentes fonctionnalités disponibles, ainsi que l'affichage immédiat des résultats.

Dans la figure IV.2 nous avons décrit les différentes fonctionnalités de l'interface graphique de notre application.

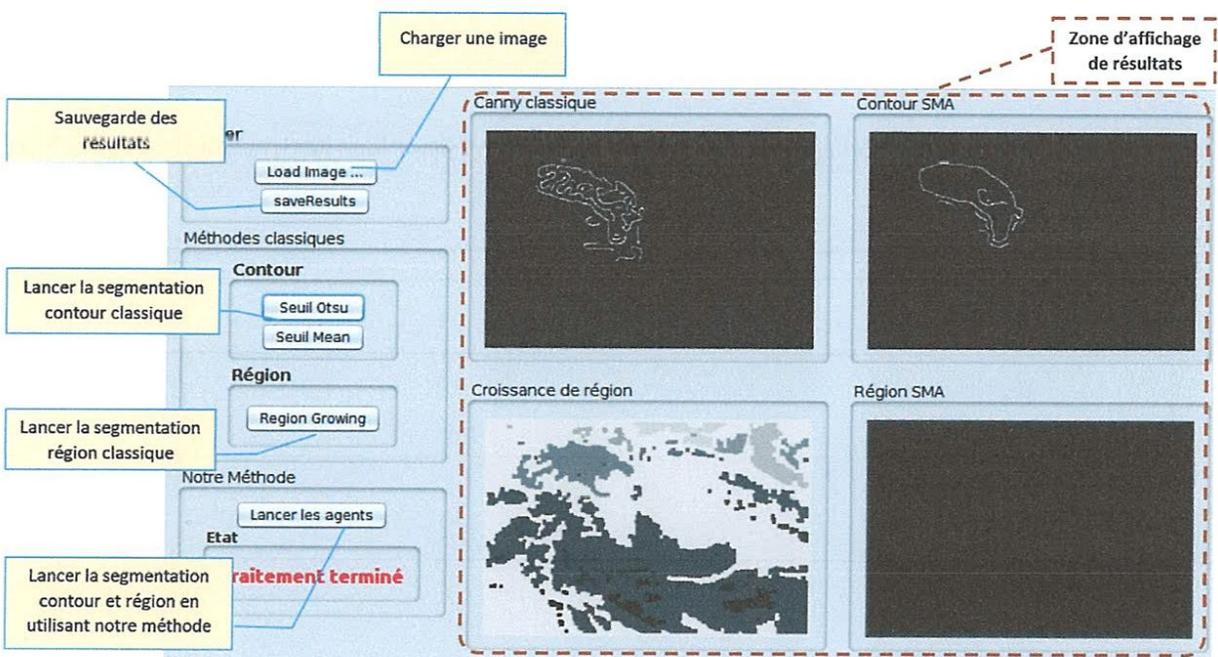


FIGURE IV.2 Interface principale de l'application.

Les principaux menus présentés dans l'interface principale sont les suivants :

Menu de gestion de fichiers

Ce menu regroupe deux boutons le premier « Load image ... » permet de charger une image à partir du disque dur. L'autre bouton « save results » permet de sauvegarder les différents résultats, chacune dans un fichier distinct.

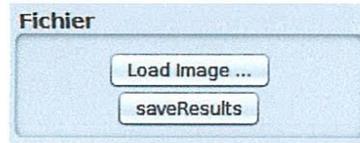


FIGURE IV.3 Menu chargement d'image et sauvegarde de résultats

Menu de lancement des méthodes de segmentation classiques

Dans ce menu Les deux boutons « Seuil Otsu » et « Seuil Mean » du panneau « Contour » lance chacun une méthode de segmentation contour différente qui utilisent le détecteur de contour «Canny» [Canny, 1986], la première utilise un seuil local calculé en utilisant la méthode d'Otsu [Otsu, 1975], et la deuxième utilise un seuil local calculé à partir de la moyenne des intensités des pixels qui constituent une zone donnée.

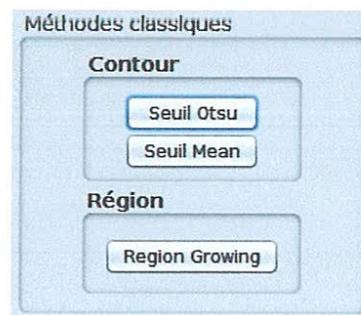


FIGURE IV.4 Menu lancement des méthodes de segmentation classiques

Le panneau « Région » contient un seul bouton « Region Growing » qui lance une segmentation région en utilisant la méthode de croissance de région.

Menu de lancement de la méthode proposée

Ce menu ne comporte qu'un seul bouton : « Lancer les agents », qui va lancer les deux SMAs qui vont s'exécuter en parallèle pour aboutir à deux résultats différents :

- Une segmentation contour améliorée à l'aide des informations région.
- Une segmentation région améliorée à l'aide des informations contour.

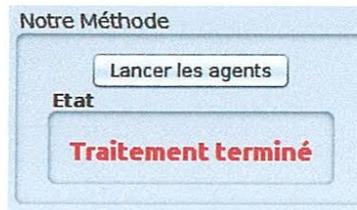


FIGURE IV.5 Menu lancement de notre méthode

IV.3 Processus d'exécution

Pendant le démarrage de l'application une fenêtre qui demande de choisir l'image à segmenter est aussitôt lancée (figure IV.6), il est toutefois possible de choisir une autre image à l'aide du bouton «load image ...».

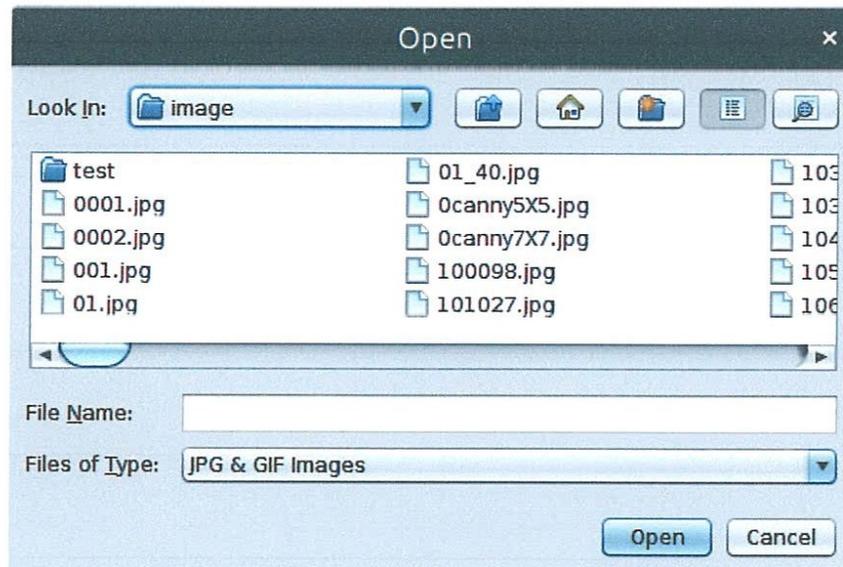


FIGURE IV.6 Boite de dialogue d'ouverture d'image

Un aperçu de l'image est affiché dans la zone d'affichage, pour permettre à l'utilisateur de voir l'image originale choisi en taille réelle (figure IV.7).

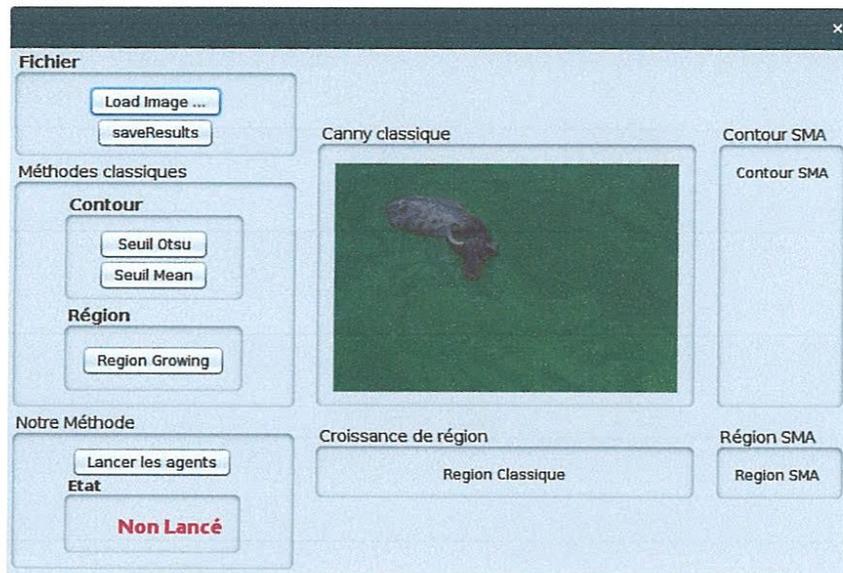


FIGURE IV.7 Aperçu de l'image choisi

L'application donne la possibilité d'avoir une segmentation classique contour ou région au choix pour pouvoir les comparer avec les résultats de la méthode proposée.

Pour les contours notre application propose la méthode de canny avec deux choix pour le calcul de seuil automatique en utilisant la moyenne des valeurs d'intensités de pixel ou bien on utilisant la méthode de Otsu [Otsu, 1975]. Pour les régions notre application propose la méthode de croissance de région.

La figure IV.8 montre les deux résultats obtenus, à droite l'image segmentée en utilisant le seuil d'Otsu, à gauche l'image segmentée en utilisant le seuil calculé à partir de la moyenne (mean) :



FIGURE IV.8 Détecteur de contour canny

Pour les régions notre application propose la méthode de croissance de région classique. La figure IV.9 montre les résultats obtenu.

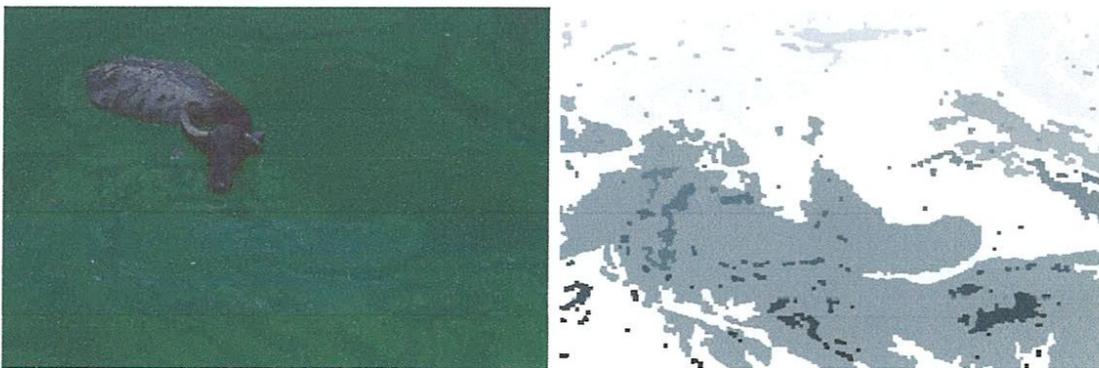


FIGURE IV.9 Segmentation en utilisant la méthode de croissance de région

En fin, l'exécution de la méthode proposée par nous donne les résultats dans la figure IV.10.



FIGURE IV.10 Résultat de la segmentation en utilisant notre méthode

V Discussion de résultats

V.1 Base d'images

Lors du développement de notre application, nous avons utilisé la base d'images de l'université de Berkeley [Arbelaez *et al.*, 2007], qui a pour but de fournir une base empirique pour la recherche sur la segmentation d'image et détection des contours. Elle a recueilli 12.000 segmentations marquées à la main de 1000 Corel images du jeu de données à partir de 30 sujets humains. Les images sont divisées en un ensemble de formation de 200 images, et un ensemble de 100 images de test.

V.2 Comparaison des résultats

Afin d'évaluer les résultats obtenus, nous avons opté pour une comparaison entre les résultats obtenus par notre méthode et ceux obtenus en appliquant les méthodes classiques présentés précédemment (pour contour canny (otsu et mean), pour région croissance de région). Les essais ont été effectués sur différents types d'images (complexe, couleur, niveau de gris, image médicale).

La figure IV.11 présente une comparaison visuelle entre les résultats obtenus par les méthodes classiques et le modèle proposé.

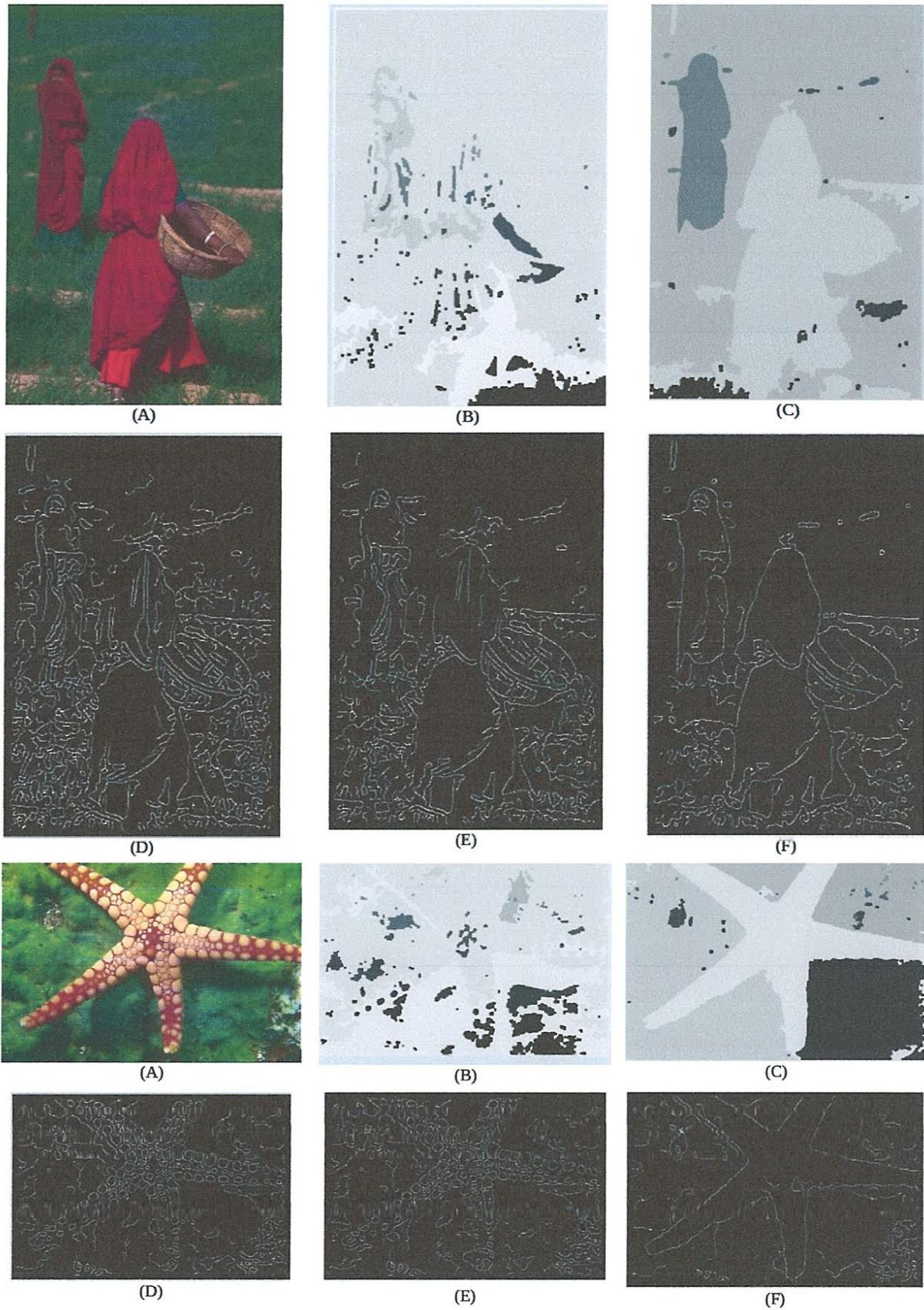


FIGURE IV.11 Comparaison des résultats expérimentaux. A : Image source, B : Résultat segmentation Région classique, C : Résultat région avec notre méthode, D : Résultat Canny avec Otsu, E : Résultat Canny avec mean, F : Résultat contour avec notre methode

D'après ces résultats nous observons que les contours détectés par le filtre de canny présentent une distorsion de bruit, et ne sont pas bien précis c'est le cas du sur-segmentation lorsqu'il s'agit d'une image complexe.

Pour les régions les zones détectées ne correspondent pas aux objets représentés dans l'image, et présentent des imprécisions c'est à dire ne coïncident pas réellement aux limites des objets.

VI Conclusion

Dans ce chapitre nous avons présenté une méthode de segmentation d'image basée sur l'utilisation d'une coopération mutuelle entre l'approche contour et l'approche région, l'algorithme a été testé sur des images de plusieurs natures, il a montré son efficacité par rapport aux approches classiques.

En contrepartie, notre méthode présente des faiblesses envers les images complexes, là où nous avons remarqué que l'application passe un temps considérable par rapport aux améliorations qu'elle porte sur le résultat, et cela est dû au choix de seuil adaptatif.

Conclusion générale et perspectives

Conclusion générale et perspectives

L'objectif principal de notre travail de recherche était de créer un système multi-agents adaptatif pour la résolution de problèmes de nature difficile pour être résolu par les approches algorithmiques classiques. Nous avons défendu l'idée que la résolution des problèmes complexes avec des agents réactifs était possible, et nous permet même d'obtenir des solutions très satisfaisantes.

Nous avons choisi la segmentation d'image comme application de cette méthodologie de résolution de problèmes. Les agents réactifs interagissent entre eux en réponse à des stimuli de leur environnement ce qui va permettre de produire un comportement global qualifié comme intelligent.

L'analyse de l'adaptation de l'algorithme a été effectuée sur différentes images. L'algorithme s'adapte avec succès à ces images, et donne des résultats satisfaisants.

En effectuant une comparaison entre les résultats obtenus par les méthodes classiques et les résultats obtenus par notre méthode. On constate que ces derniers sont relativement meilleurs que ceux obtenus par les méthodes classiques dans la plupart des images. D'autre part, ces résultats sont mieux que ceux obtenus par le filtre « Canny » concernant les images contenant beaucoup d'ouvertures, et celles prises dans des conditions de luminosité différentes.

En conclusion, nous pouvons dire que l'approche de segmentation par les Systèmes Multi-Agents adaptatifs est l'un des éléments essentiels dans l'analyse et l'interprétation d'images.

En perspective, notre système peut être amélioré par calcul adaptatif du seuil global

de l'image, par le choix automatique des paramètres d'application tels que critères d'arrêt et critères de convergence par une application plus concrète dans le domaine médicale ou de vidéo surveillance.

Bibliographie

- [Acharya et Ray, 2005] ACHARYA, T. et RAY, A. K. (2005). *Image processing : principes and applications*.
- [Amblard, 2003] AMBLARD, F. (2003). *Comprendre le fonctionnement de simulations sociales individus-centrées : Application à des modèles de dynamiques d'opinions*. Thèse de doctorat, Université Blaise Pascal-Clermont-Ferrand II.
- [Anderson et al., 1987] ANDERSON, H. L., BAJCSY, R. et MINTZ, M. (1987). *A modular feedback system for image segmentation*. University of Pennsylvania, School of Engineering and Applied Science, Department of Computer and Information Science.
- [Arbelaez et al., 2007] ARBELAEZ, P., FOWLKES, C. et MARTIN, D. (2007). The berkeley segmentation dataset and benchmark. see <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds>.
- [Arlabosse et al., 2004] ARLABOSSE, F., GLEIZES, M.-P. et OCCELLO, M. (2004). Méthodes de conception de systèmes multi-agents. *Systèmes Multi-Agents-Collection ARAGO-N 29*.
- [Bajcsy et al., 1986] BAJCSY, R., MINTZ, M. et LIEBMAN, E. (1986). *A Common Frame Work for Edge Detection and Region Growing*. University of Pennsylvania, Department of Computer and Information Science.
- [Belhamra et Melzi, 2010] BELHAMRA, H. et MELZI, S. (2010). *La segmentation image par croissance de région*. *ESI*. Thèse de doctorat.

- [Bellet *et al.*, 1995] BELLET, F., SALOTTI, M. et GARBAY, C. (1995). Une approche opportuniste et coopérative pour la vision de bas niveau. *TS. Traitement du signal*, 12(5):479-494.
- [Benamrane et Nassane, 2007] BENAMRANE, N. et NASSANE, S. (2007). Medical image segmentation by a multi-agent system approach. *In German Conference on Multiagent System Technologies*, pages 49-60. Springer.
- [Benois et Barba, 1992] BENOIS, J. et BARBA, D. (1992). Image segmentation by region-contour cooperation for image coding. *In Pattern Recognition, 1992. Vol. III. Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on*, pages 331-334. IEEE.
- [Bernsen, 1986] BERNSEN, J. (1986). Dynamic thresholding of grey-level images. *In International conference on pattern recognition*, pages 1251-1255.
- [Bloch *et al.*, 2004] BLOCH, I., GOUSSEAU, Y., MAÎTRE, H., MATIGNON, D., PESQUET-POPESCU, B., SCHMITT, F., SIGELLE, M. et TUPLIN, F. (2004). Le traitement des images. *Polycopié du cours ANIM, version*, 5.
- [Boissier *et al.*, 2004a] BOISSIER, O., GITTON, S. et GLIZE, P. (2004a). Caractéristiques des systèmes et des applications. *Systèmes multi-agents/Observatoire français des techniques avancées, ARAGO 29*, pages pp 25.
- [Boissier *et al.*, 2004b] BOISSIER, O., GITTON, S. et GLIZE, P. (2004b). Caractéristiques des systèmes et des applications. *Systèmes multi-agents/Observatoire français des techniques avancées, ARAGO 29*, pages pp 25.
- [Bureau *et al.*, 2001] BUREAU, A., GARBAY, C. et DOJAT, M. (2001). Coopération entre deux populations d'agents pour la segmentation.
- [Calvez et Hutzler, 2005] CALVEZ, B. et HUTZLER, G. (2005). Automatic tuning of agent-based models using genetic algorithms. *In Multi-agent-based simulation VI*, pages 41-57. Springer.
- [Campagne, 2005] CAMPAGNE, J.-C. (2005). *Systèmes multi-agents et morphologie*. Thèse de doctorat, Université Pierre et Marie Curie - Paris VI.

- [Campagne *et al.*, 2004a] CAMPAGNE, J.-C., CARDON, A., COLLOMB, E. et NISHIDA, T. (2004a). Massive multi-agent system control. In *FAABS III 2004, IEEE Workshop on Formal Approaches on Agents-based Systems, LNCS 3228*,. NASA Goddard Space Center, Greenbelt MA, USA.
- [Campagne *et al.*, 2004b] CAMPAGNE, J.-C., CARDON, A., COLLOMB, E. et NISHIDA, T. (2004b). Using morphology to analyse and control a multi-agent system, an example. In *STAIRS 2004 : Proceedings of the Second Starting AI Researchers' Symposium*, volume 109, page 229. IOS Press.
- [Canny, 1986] CANNY, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679 698.
- [Chabbi, 1993] CHABBI, H. (1993). Checking 3d planar surfaces using projective geometry. In *PROCEEDINGS OF THE SCANDINAVIAN CONFERENCE ON IMAGE ANALYSIS*, volume 2, pages 1023 1023. PROCEEDINGS PUBLISHED BY VARIOUS PUBLISHERS.
- [Chan et Shen, 2005] CHAN, T. F. et SHEN, J. J. (2005). *Image processing and analysis : variational, PDE, wavelet, and stochastic methods*. Siam.
- [Cheng *et al.*, 1998] CHENG, H., CHEN, J.-R. et LI, J. (1998). Threshold selection based on fuzzy c-partition entropy approach. *Pattern recognition*, 31(7):857 870.
- [Chevrier, 1993] CHEVRIER, V. (1993). *Etude et mise en oeuvre du paradigme multi-agents : de Atome a Gtmas*. Thèse de doctorat.
- [Culioli, 1994] CULIOLI, J. (1994). Introduction à l'optimisation, editor : ellipses. ISBN : 2729894284.
- [De Wolf et Holvoet, 2004] DE WOLF, T. et HOLVOET, T. (2004). Emergence versus self-organisation : Different concepts but promising when combined. In *Engineering self-organising systems*, pages 1 15. Springer.
- [De Wolf *et al.*, 2005] DE WOLF, T., SAMAEY, G. et HOLVOET, T. (2005). Engineering self-organising emergent systems with simulation-based scientific analysis. In *Proceedings of the Fourth International Workshop on Engineering Self-Organising Applications*, pages 146 160.

- [Deguet, 2008] DEGUET, J. (2008). *Intégration de l'émergence au sein des systèmes multi-agent Une étude appliquée à la recherche heuristique*. Thèse de doctorat, Université Joseph-Fourier-Grenoble I.
- [Deneubourg *et al.*, 1991] DENEUBOURG, J.-L., GOSS, S., FRANKS, N., SENDOVA-FRANKS, A., DETRAIN, C. et CHRÉTIEN, L. (1991). The dynamics of collective sorting robot-like ants and ant-like robots. *In Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, pages 356 363.
- [Djemame et Batouche, 2009] DJEMAME, S. et BATOUCHE, M. (2009). Une approche biomimétique pour la segmentation d'images inspiration des araignées sociales. *In CIIA*.
- [Djerou et Batouche, 2013] DJEROU, L. et BATOUCHE, M. (2013). Une approche biomimétique pour la segmentation d'images.
- [Drogoul *et al.*, 2004] DROGOUL, A., FERRAND, N. et MÜLLER, J. (2004). Emergence : l'articulation du local au global. *Observatoire Français des Techniques Avancées : Systèmes Multi-Agents Série ARAGO*, 29.
- [Edmonds, 2004] EDMONDS, B. (2004). Using the experimental method to produce reliable self-organised systems. *In Engineering Self-Organising Systems*, pages 84 99. Springer.
- [Ferber, 1995] FERBER, J. (1995). Les systèmes multi-agents : vers une intelligence collective. informatique. *Intelligence Artificielle. InterÉditions*.
- [Ferber et Müller, 1996] FERBER, J. et MÜLLER, J.-P. (1996). Influences and reaction : a model of situated multiagent systems. *In Proceedings of Second International Conference on Multi-Agent Systems (ICMAS-96)*, pages 72 79.
- [Gadi et Benslimane, 2000] GADI, T. et BENSLIMANE, R. (2000). Segmentation hiérarchique floue. *Traitement du signal*, 7(1).
- [Gallone J-M., 1994] GALLONE J-M., C. F. e. C. V. (1994). *Un modèle d'agent pour un raisonnement contraint par le temps*. Thèse de doctorat.
- [Gechter et Simonin, 2005] GECHTER, F. et SIMONIN, O. (2005). Conception de SMA réactifs pour la résolution de problèmes : une approche basée sur l'environnement. *In JFSMA*, pages 157 173.

- [Gupta *et al.*, 2007] GUPTA, M. R., JACOBSON, N. P. et GARCIA, E. K. (2007). Ocr binarization and image pre-processing for searching historical documents. *Pattern Recognition*, 40(2):389 397.
- [Haralick et Shapiro, 1985] HARALICK, R. M. et SHAPIRO, L. G. (1985). Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100 132.
- [Horaud et Monga, 1995] HORAUD, R. et MONGA, O. (1995). *Vision par ordinateur : outils fondamentaux*. Editions Hermès.
- [Idir *et al.*, 2005] IDIR, K., MEROUANI, H. et TLILI, Y. (2005). *Image Segmentation Through Dual Pyramid of Agents*, pages 360 366. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Jean *et al.*, 1997] JEAN, M. *et al.* (1997) Emergence et sma. *IA distribuée et systèmes multi-agents, JFIADSMA*, 97:323 341.
- [Kapur *et al.*, 1985] KAPUR, J. N., SAHOO, P. K. et WONG, A. K. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3):273 285.
- [Kefali *et al.*, 2009] KEFALI, A., SARI, T. et SELAMI, M. (2009). Evaluation de plusieurs techniques de seuillage d'images de documents arabes anciens. *In 5ème symposium international Images Multimédias Applications Graphiques et Environnements*, pages 123 134.
- [Kermad, 1997] KERMADE, C. (1997). *Segmentation d'image : recherche d'une mise en oeuvre automatique par coopération de méthodes*. Thèse de doctorat, Université Rennes 1.
- [Khurshid *et al.*, 2009] KHURSHID, K., SIDDIQI, I., FAURE, C. et VINCENT, N. (2009). Comparison of niblack inspired binarization methods for ancient documents. *In IS&T/SPIE Electronic Imaging*, pages 72470U 72470U. International Society for Optics and Photonics.
- [Klein, 2009] KLEIN, F. (2009). *Contrôle d'un Système Multi-Agents Réactif par Modélisation et Apprentissage de sa Dynamique Globale*. Thèse de doctorat, Université Nancy 2.

- [Li et Lee, 1993] LI, C. H. et LEE, C. (1993). Minimum cross entropy thresholding. *Pattern Recognition*, 26(4):617 625.
- [Liu et Tang, 1999] LIU, J. et TANG, Y. Y. (1999). Adaptive image segmentation with distributed behavior-based agents. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(6):544 551.
- [Manneville, 2001] MANNEVILLE, P. (2000-2001). Dynamique non-linéaire appliquée au chaos et à son contrôle. *Lecture notes/notes de cours, DEA de Dynamique des Fluides et des Transferts (Paris-Sud) et DEA de Mécanique (Paris VI)*.
- [Meddeber, 2013] MEDDEBER, H. (2013). *Segmentation d'images Par Les Contours Actifs*. Thèse de doctorat, usto.
- [Meliani, 2012] MELIANI, M. (2012). Segmentation d'image par coopération régions-contours.
- [Mobahi et al., 2006] MOBAHI, H., AHMADABADI, M. N. et ARAABI, B. N. (2006). Swarm contours : A fast self-organization approach for snake initialization. *Complexity*, 12(1):41 52.
- [Moraïtis, 1994] MORAÏTIS, P. (1994). *Paradigme multi-agent et prise de décision distribuée*. Thèse de doctorat.
- [Otsu, 1975] OTSU, N. (1975). A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23 27.
- [Redjimi et Amri, 2012] REDJIMI, M. et AMRI, S. (2012). An adaptative multi-agent system approach for image segmentation. *International Journal of Computer Applications*, 51(12).
- [Richard et al., 2004] RICHARD, N., DOJAT, M. et GARBAY, C. (2004). Automated segmentation of human brain mr images using a multi-agent approach. *Artificial Intelligence in Medicine*, 30(2):153 176.
- [Sauvola et al., 1997] SAUVOLA, J., SEPPÄNEN, T., HAAPAKOSKI, S. et PIETIKÄINEN, M. (1997). Adaptive document binarization. In *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, volume 1, pages 147 152. IEEE.

- [Sebari et He, 2007] SEBARI, I. et HE, D.-C. (2007). Les approches de segmentation d'image par coopération régions-contours. *Revue télédétection*, 7(1-2-3-4):499 506.
- [Settache, 2002] SETTACHE, H. (2002). *Une plateforme multi-agent pour la segmentation d'images : Application dans le domaine de IRM cérébrale 2D*. Thèse de doctorat.
- [Shalizi et al., 2001] SHALIZI, C. R. et al. (2001). *Causal architecture, complexity and self-organization in the time series and cellular automata*. Thèse de doctorat, University of Wisconsin Madison.
- [Sichman, 1995] SICHMAN, J. S. (1995). *Du raisonnement social chez les agents : une approche fondée sur la théorie de la dépendance*. Thèse de doctorat.
- [Siebert et al., 2008] SIEBERT, J., CIARLETTA, L. et CHEVRIER, V. (2008). Impact du comportement des utilisateurs dans les réseaux pair-à-pair (p2p) : modélisation et simulation multi-agents. *In 16es Journées Francophones des Systèmes Multi-Agents-JFSMA '08*, pages 129 138.
- [Simonin et al., 2002] SIMONIN, O., MICHEL, F., CHAPELLE, J. et FERBER, J. (2002). Un simulateur de systèmes multi-robots dans madkit. *In JFSMA*, pages 167 170.
- [Steels et Brooks, 1994] STEELS, L. et BROOKS, R. (1994). The artificial life route to artificial intelligence. building situated embodied agents.
- [Sycara, 1989] SYCARA, K. (1989). Multiagent compromise via negotiation. distributed artificial intelligence ii, ed., gasser, l. and huhns, m.
- [Thomas et al., 2002] THOMAS, V., BOURJOT, C., CHEVRIER, V. et DESOR, D. (2002). Mas and rats multi-agent simulation of social differentiation in rats' groups. interest for the understanding of a complex biological phenomenon. *In International Workshop on Self-Organization and Evolution of Social Behaviour, Monte Verita*.
- [Vercouter, 2001] VERCOUTER, L. (2001). Une gestion distribuée de l'ouverture dans un système multi-agent. *In Journées Francophones IAD et SMA 2001*, pages pp 177.
- [Vercouter, 2004] VERCOUTER, L. (2004). MAST : Un modèle de composants pour la conception de SMA. *Journées Multi-Agents et Composants (JMAC'04)*, pages 18 31.

- [Weyns *et al.*, 2004] WEYNS, D., PARUNAK, H. V. D., MICHEL, F., HOLVOET, T. et FERBER, J. (2004). Environments for multiagent systems state-of-the-art and research challenges. In *Environments for multi-agent systems*, pages 1 47. Springer.
- [Wolf *et al.*, 2002] WOLF, C., JOLION, J.-M. et CHASSAING, F. (2002). Extraction de texte dans des vidéos : le cas de la binarisation. *Proceedings of RFIA 2002*, 1:145 152.
- [Wooldridge, 2009] WOOLDRIDGE, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.
- [Wooldridge *et al.*, 1995] WOOLDRIDGE, M., JENNINGS, N. R. *et al.* (1995). Intelligent agents : Theory and practice. *Knowledge engineering review*, 10(2):115 152.
- [Wooldridgey et Ciancarini, 2001] WOOLDRIDGEY, M. et CIANCARINI, P. (2001). Agent-oriented software engineering : The state of the art. In *Agent-oriented software engineering*, pages 1 28. Springer.