

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ MENTOURI DE CONSTANTINE
FACULTÉ DES SCIENCES DE L'INGÉNIEUR
DÉPARTEMENT D'INFORMATIQUE

N° 18/TS/2008 - 02/INF/2008

Thèse

Présentée pour obtenir le diplôme de

Doctorat en Sciences

Spécialité : INFORMATIQUE

Soutenue publiquement par : **Smaine MAZOUZI**

le 18 mai 2008

THEME

RECONNAISSANCE DE FORMES PAR LES SYSTEMES
AUTO-ORGANISÉS

Application aux images de profondeur

Directeur de thèse : Mohamed BATOUCHE, Prof. Université de Constantine

DEVANT LE JURY :

| | | |
|------------|----------------|---|
| Président | M. BENMOHAMMED | <i>Professeur, Université de Constantine</i> |
| Rapporteur | M.K. KHOLLADI | <i>Maître de Conférences, Université de Constantine</i> |
| Examineurs | M. KHAMADJA | <i>Professeur, Université de Constantine</i> |
| | A. BOUKERRAM | <i>Maître de Conférences, Université de Setif</i> |
| | A. MOUSSAOUI | <i>Maître de Conférences, Université de Setif</i> |
| Invité | N. ZAROOUR | <i>Maître de Conférences, Université de Constantine</i> |

MAI 2008

Cette thèse est dédiée

*à ma mère et à mon père,
à mon épouse et à ma fille,
à ma tante et à mon oncle,
à mes soeurs et à mes frères,
et aux amis.*

Remerciement

Tous mes remerciements à ceux qui m'ont aidé à arriver à terme de cette thèse, et en particulier :

Monsieur Mohamed Batouche, Madame Zahia Guessoum, et Monsieur Mohamed Khireddine Kholadi, respectivement directeur et co-directeurs de cette thèse. Je tiens à exprimer une gratitude spéciale à Zahia, pour m'avoir accueilli au sein du laboratoire du LERI à Reims et pour ses précieux conseils et orientations ;

Monsieur Mohamed Benmohamed, Professeur à l'université de Constantine, qui m'a fait l'honneur de présider le jury de soutenance de cette thèse ;

Messieurs Mohamed Khamadja, Professeur à l'université de Constantine, Abdellah Boukerram et Abdelouahab Moussaoui, Maîtres de Conférences à l'université de Setif, et Nasreddinne Zaarour, Maître de Conférences à l'université de Constantine, pour avoir accepté d'évaluer mon travail et de participer au jury de soutenance.

Et à toute personne ayant contribué de près ou de loin à l'aboutissement de ce travail.

ملخص

تهتم هذه الأطروحة بإستعمال الأنظمة تلقائية التنظيم و المتمثلة في الأنظمة متعددة الأعوان و شبكات العصبونات؛ من أجل تحليل الصور و التعرف على الأشكال، و تطبيقها من أجل التعرف على الأشياء من خلال الصور ذات الأعماق. تم إقتراح مجموعة من نماذج الأنظمة متعددة الأعوان من أجل المعالجة في المستويات الدنيا كالتلميس و تقطيع الصور وكذلك من أجل المعالجة في المستويات العليا كالتمثيل النمذجي لسطوح و التعرف عل الأشياء. يعتمد النموذج الأول على مجموعة من الأعوان التفاعلية و التي تستعمل الصورة كوسط حيث تهدف الى ضهور تقطيع تجمعي للصورة و ذلك بواسطة العمليات التنافسية للأعوان. خصص النموذج الثاني، ذو المعمارية المعرفية، لتفسير محتويات الصور إعتيادا على موافقة تجمعية للأشياء و النماذج. كما تم كذلك إستضهار نموذج تشتتي لشبكة عصبونات، حيث يمكن للنموذج من إستشعار الحواف الملساء في الصور ذات العمق.

تتطرق الأطروحة كذلك الى التصويب البايزي في أنظمة الرؤية ثلاثية الأبعاد. و قد تم إعتياد هذا الوجه من أجل مواجهة المشاكل الناتجة عن توزيع المعالجات و القرارات، المميز للأنظمة المتعددة الأعوان. في هذا الإطار ندرج نمودجا جديدا للحقل العشوائي لماركوف، و الذي يمكن من تحسين نتائج تقطيع الصور الناتجة عن نظام التقطيع المتعدد الأعوان. و نقترح أيضا طريقة جديدة لتصحيح الحواف و التي تعتمد التصويب الغامض.

إن النماذج المتعددة الأعوان المطورة في إطار هذه الأطروحة أريد لها أن تكون عامة و مستقلة عن محتوى الصور و عن مجال التطبيق. و قد مكن ذلك من توفير إطار منهجي عام لتطوير الأنظمة المتعددة الأعوان الخاصة بمعالجة الصور و التعرف على الأشكال.

Résumé

La présente thèse s'intéresse à l'utilisation des systèmes auto-organisés, en l'occurrence les systèmes multi-agents et les réseaux de neurones chaotiques, pour l'interprétation d'images et la reconnaissance de formes, avec application à la reconnaissance d'objets à partir d'images de profondeur. Des modèles de systèmes multi-agents sont proposés à la fois pour les traitements de bas niveau, tels que le lissage et la segmentation d'images, et pour les traitements de haut niveau, tels que la représentation symbolique des surfaces et la reconnaissance d'objets. Le premier modèle est basé sur une population d'agents réactifs qui utilise l'image comme environnement et dont l'objectif est de faire émerger une segmentation collective de l'image par action compétitive des agents. Le second modèle, basé sur une architecture cognitive, est consacré à l'interprétation du contenu de l'image par appariement collectif objet-modèle. Un modèle de réseaux de neurones chaotiques est également présenté. Le modèle permet la détection de contours lissés dans les images de profondeur.

La thèse aborde également la régularisation bayésienne dans les systèmes de vision tridimensionnelle. Ce dernier aspect intervient pour pallier aux problèmes liés à la distribution des traitements et des décisions inhérente aux systèmes multi-agents. Dans ce cadre nous proposons un modèle du champ aléatoire de Markov qui permet d'améliorer les résultats de segmentation, produits par le système multi-agents dédié à la segmentation d'images. Nous introduisons également une nouvelle méthode de correction des contours basée sur la régularisation floue.

Les modèles multi-agents développés dans le cadre de cette thèse se veulent généraux, et ont été conçu indépendamment du contenu des images et du domaine d'application. Ceci a permis de fournir un framework général pour le développement de systèmes multi-agents pour le traitement d'images et la reconnaissance d'objets.

Abstract

The present thesis is interested in using self-organized systems, it happened, multi-agent systems and chaotic neural networks, for image understanding and pattern recognition, with application to object recognition from range images. Some multi-agent models are proposed both for low-level processing such as image smoothing and segmentation, and for high-level processing such as symbolic representation of surfaces, and object recognition. The first model is based on a population of reactive agents which use the image as an environment and where the goal is to make a collective segmentation emerges, thanks to competitive actions of agents. The second model, which is based on a cognitive architecture, is dedicated for interpretation of image contents by a collective object-model matching. A model of chaotic neural networks is also presented. The model allows the detection of smoothed edges in range images.

The thesis approaches also Bayesian regularization in three-dimensional computer vision systems. This last aspect is used to deal with problems related to the distribution of treatments and decisions, inherent to multi-agent systems. Within this part, we propose a new Markov Random Field model which allows to improve segmentation results, produced by the multi-agent system, dedicated for image segmentation. We introduce also a new method for contour correction, based on a fuzzy regularization.

The multi-agent models developed in part of this thesis claim to be general, and have been conceived to be free from image content, and application domain. This has allowed to provide a general framework for building multi-agent systems for image interpretation and object recognition.

Table des matières

| | |
|---|----------|
| Dédicace | iii |
| Remerciement | iv |
| Résumé en arabe | v |
| Résumé | vii |
| Abstract | viii |
| Table des matières | xiii |
| Table des figures | xvi |
| Liste des tableaux | xvii |
| 1 Introduction générale | 1 |
| 2 Approches multi-agents en vision par ordinateur | 8 |
| 2.1 Introduction | 8 |
| 2.2 Les systèmes multi-agents | 9 |
| 2.2.1 Définitions | 9 |
| 2.2.2 Classification | 11 |
| 2.3 Systèmes multi-agents pour l'analyse et l'interprétation d'images | 11 |
| 2.3.1 Segmentation adaptative d'images par agents distribués | 12 |
| 2.3.2 Un système immuno-multi-agents pour le traitement d'images biologiques | 14 |
| 2.3.3 Un système co-évolutionnaire pour la segmentation d'images | 16 |

| | | |
|----------|---|-----------|
| 2.3.4 | Une architecture à base d'agents situés pour la segmentation d'IRM du cerveau | 18 |
| 2.3.5 | Système multi-agents massif pour la segmentation d'IRM | 20 |
| 2.3.6 | Surveillance multi-agents de scènes dynamiques | 22 |
| 2.3.7 | Une architecture cognitive pour la segmentation des IVUS | 23 |
| 2.3.8 | Synthèse | 25 |
| 2.4 | Conclusion | 28 |
| 3 | Un système multi-agents pour la segmentation d'images de profondeur | 30 |
| 3.1 | Introduction | 30 |
| 3.2 | Images de profondeur | 31 |
| 3.3 | Modélisation et calcul des propriétés locales des surfaces | 33 |
| 3.4 | Segmentation collective d'images par agents réactifs | 35 |
| 3.4.1 | Comportement des agents | 35 |
| 3.4.2 | Détection de contours | 38 |
| 3.4.3 | Elimination du bruit | 41 |
| 3.5 | Coordination d'agents par un champ de potentiel | 42 |
| 3.5.1 | Champ de potentiel dans les systèmes multi-agents | 42 |
| 3.5.2 | Approche de coordination proposée | 47 |
| 3.5.3 | Exemple de l'évolution de la dynamique des agents | 49 |
| 3.6 | Implémentation | 50 |
| 3.6.1 | Pixel | 51 |
| 3.6.2 | Image | 52 |
| 3.6.3 | Agent | 52 |
| 3.6.4 | Scheduler | 53 |
| 3.6.5 | Analyse du coût de calcul | 54 |
| 3.7 | Conclusion | 56 |
| 4 | Régularisation bayésienne et régularisation floue des contours 3D | 59 |
| 4.1 | Introduction | 59 |
| 4.2 | Régularisation des contours par inférence bayésienne | 60 |
| 4.2.1 | Etat de l'art | 60 |
| 4.2.2 | Aperçu de la méthode | 61 |
| 4.2.3 | Modélisation de la segmentation en MAP-MRF | 61 |
| 4.2.4 | Minimisation de l'énergie | 65 |
| 4.3 | Régularisation floue des contours | 66 |

| | | |
|----------|--|-----------|
| 4.3.1 | Etat de l'art | 66 |
| 4.3.2 | Aperçu de la méthode | 67 |
| 4.3.3 | La segmentation d'images par classification floue | 68 |
| 4.3.4 | Classement flou des pixels de contours | 69 |
| 4.4 | Conclusion | 71 |
| 5 | Détection de contours par les réseaux de neurones chaotiques | 73 |
| 5.1 | Introduction | 73 |
| 5.2 | Approches connexionnistes pour la détection de contours | 74 |
| 5.3 | Les Réseaux de neurones chaotiques | 75 |
| 5.4 | Contours lissés dans les images de profondeur | 75 |
| 5.5 | Détection des contours par réduction du chaos | 78 |
| 5.5.1 | Le modèle de réseau utilisé | 79 |
| 5.5.2 | Sélection des pixels de contours | 80 |
| 5.5.3 | Couplage des paires d'éléments | 81 |
| 5.6 | Test du modèle | 82 |
| 5.7 | Conclusion | 84 |
| 6 | Une architecture coopérative pour la reconnaissance d'objets 3D | 86 |
| 6.1 | Introduction | 86 |
| 6.2 | Aperçu du système | 87 |
| 6.3 | Calcul d'appariement objet-modèle | 87 |
| 6.3.1 | Formulation du problème | 88 |
| 6.3.2 | Calcul de la BRM | 89 |
| 6.4 | Le système multi-agents proposé | 91 |
| 6.4.1 | L'agent Reconnaiseur | 92 |
| 6.4.2 | L'agent Coordonnateur | 93 |
| 6.5 | Conclusion | 94 |
| 7 | Evaluation et discussion | 97 |
| 7.1 | Introduction | 97 |
| 7.2 | Le protocole expérimental | 98 |
| 7.2.1 | Objectif et Hypothèses | 98 |
| 7.2.2 | La base d'images | 99 |
| 7.2.3 | Les méthodes utilisées dans la comparaison | 99 |
| 7.2.4 | Les métriques de performance | 100 |
| 7.2.5 | Test des hypothèses | 101 |
| 7.3 | Un framework d'évaluation de performances et de comparaison | 102 |
| 7.3.1 | Calcul des métriques de performance | 102 |

| | | |
|----------|---|------------|
| 7.3.2 | Instances de classification | 104 |
| 7.4 | Apprentissage du système et sélection des paramètres | 106 |
| 7.4.1 | Les paramètres de segmentation | 106 |
| 7.4.2 | Les paramètres de la dynamique du système | 107 |
| 7.4.3 | Les paramètres de la régularisation bayésienne | 107 |
| 7.5 | Analyse de la stabilité de la segmentation multi-agents | 108 |
| 7.6 | Résultats de segmentation obtenus par le système SIBA | 111 |
| 7.7 | Résultats après régularisation | 116 |
| 7.8 | Conclusion | 119 |
| 8 | Conclusion générale | 121 |
| | Bibliographie | 125 |

Table des figures

| | | |
|------|---|----|
| 2.1 | Comportement d'un agent en fonction du stimulus local | 13 |
| 2.2 | Agents réactifs pour la détection d'anneaux concentriques. (a) Plaice otolith de 8 ans; (b) Architecture BRIC d'un agent | 15 |
| 2.3 | Capteurs d'un agents | 15 |
| 2.4 | Co-évolution des tributs. (a) Identités initiales des agents; (b) Echange d'identités entre agents; (c) Influence mutuelle entre agents; (d) Tribus stables et homogènes | 18 |
| 2.5 | Phases de segmentation d'une IRM : Les modèles sont instanciés puis graduellement raffinés par intercalation itérative des processus d'estimation, de classification et de fusion | 19 |
| 2.6 | Réseau bayésien des agents utilisés : (a) Agent de comportement; (b) Agent de situation. | 22 |
| 2.7 | Un système Multi-agents pour la segmentation des images IVUS. (a) Un exemple d'image IVUS; (b) Architecture du système | 24 |
| 3.1 | Principe de télémétrie par triangulation | 32 |
| 3.2 | Principe du capteur utilisé | 33 |
| 3.3 | Le comportement d'un agent en fonction de son état et de sa position | 36 |
| 3.4 | Impact de la force d'altération sur les résultats de segmentation : (a) Image de profondeur (abw.test.8); (b) Résultats de segmentation avec $\xi_{min} = \xi_{max} = 4$ à $t=2500$; (c) Résultats de segmentation avec $\xi_{min} = 0.3$ et $\xi_{max} = 5$ à $t=13000$ | 39 |
| 3.5 | Amincissement de contours (image abw.test.22) : (a) Image de profondeur; (b) Pixels de contour à $t=800$; (c) Pixels de contour à $t= 8000$ | 40 |
| 3.6 | Buts et Obstacles dans un champ de potentiel | 44 |
| 3.7 | Force tangentielle de glissement | 46 |
| 3.8 | Champ de potentiel créé par l'altération des pixels par les agents. | 48 |
| 3.9 | Evolution de la dynamique des agents dans le champ de potentiel, et résultats de segmentation | 51 |
| 3.10 | Diagramme de classes simplifié | 53 |
| 3.11 | Variation de la taille de la région de bruit (A) : (a) Image de profondeur rendue réaliste; (b) à $t=5000$; (c) à $t=10000$; (d) à $t=15000$ | 55 |

| | | |
|------|---|-----|
| 3.12 | Variations théorique et observée de la taille de la région de bruit (A) | 56 |
| 4.1 | Potentiel de clique $V^2(c_{3 \times 3})$. (a) segment totalement lissé : $V^2(c_{3 \times 3}) = -\kappa$; (b) segment partiellement lissé : $V^2(c_{3 \times 3}) = 0$; autrement, le segment est non lissé : $V^2(c_{3 \times 3}) = \kappa$ | 65 |
| 4.2 | L'ensemble flou S , représentant l'ensemble des pixels appartenant à la frontière des deux régions | 69 |
| 5.1 | Contours lissés dans une image de profondeur : (a) Image de profondeur ; (b) Contours détectés par l'opérateur du laplacien | 76 |
| 5.2 | Paire de neurones : Excitateur-inhibiteur | 79 |
| 5.3 | Calcul de la valeur moyenne du laplacien de la surface. Les deux segments de pixels entre la surface courbée et les deux plans correspondent au passage par zéro de la valeur moyenne du laplacien. Ces pixels représentent les contours lissés figurant dans l'image | 81 |
| 5.4 | Test du modèle en absence de bruit : (a) Image de profondeur ; (b) Contours détectés | 83 |
| 5.5 | Test du modèle en présence de bruit avec $I_c > 0$: (a) Image de profondeur ; (b) Contours détectés | 83 |
| 6.1 | Principe du calcul de la BRM | 88 |
| 6.2 | Formulation du calcul de la BRM | 89 |
| 6.3 | Génération d'une hypothèse de correspondance objet-modèle | 92 |
| 6.4 | Hypothèses de correspondance objet-modèle (a,b) ; (c) Modèles d'objets | 94 |
| 7.1 | Appariement de segmentation GT (a) et de segmentation MS (b) de l'image abw.test.8. | 103 |
| 7.2 | Images utilisées pour le test de stabilité du système. (a) abw.test.6 ; (b) abw.test.10 ; (c) abw.test.23 | 110 |
| 7.3 | Moyennes et écart-types des nombres de régions correctement détectées en 10 exécutions (image abw.test.6) | 111 |
| 7.4 | Moyennes et écart-types des nombres de régions correctement détectées en 10 exécutions (image abw.test.10) | 111 |
| 7.5 | Moyennes et écart-types des nombres de régions correctement détectées en 10 exécutions (image abw.test.10) | 112 |
| 7.6 | Progression de segmentation. (a) Image de profondeur rendue réaliste (abw.test.6) ; (b) à $t=1000$, (c) à $t=5000$; (d) à $t=9000$; (e) à $t=13000$; (f) Image reconstruite rendue réaliste | 113 |

| | | |
|------|---|-----|
| 7.7 | Résultats de segmentation de l'image abw.test.8. (a) Image de profondeur; (b) Image de profondeur rendue réaliste; (c) Segmentation selon la Réalité terrain(GT); (d) Résultat de USF; (e) Résultat de WSU; (f) Résultat de UB; (g) Résultat de UE; (h) Résultats de SIBA (Régions extraites) | 114 |
| 7.8 | Résultats moyens des régions correctement détectées de toutes les méthodes au différentes valeurs du seuil de tolérance T ; $0.5 < T \leq 1.0$ | 116 |
| 7.9 | Résultats moyens des métriques de détection incorrecte : (a) Sur-segmentation; (b) Sous-segmentation; (c) Regions omises; (d) Regions de bruit | 117 |
| 7.10 | Un exemple de résultats après régularisation. (a) Image de profondeur; (b) Image rendu réaliste; (c) Sementation par simple accroissement de régions; (d) Segmentation par le système SIBA (avec régularisation bayésienne) | 118 |
| 7.11 | Impact de la régularisation sur les résultats de segmentation. (a) Régularisation bayésienne; (b) Régularisation floue. | 119 |

Liste des tableaux

| | | |
|-----|--|-----|
| 7.1 | Matrice des correspondances O_{mn} de l'image de la figure 7.1 | 104 |
| 7.2 | Régions correctement détectées en 10 exécutions (abw.test.6) | 108 |
| 7.3 | Régions correctement détectées en 10 exécutions (abw.test.10) | 109 |
| 7.4 | Régions correctement détectées en 10 exécutions (abw.test.23) | 109 |
| 7.5 | Comparaison des résultats de segmentation de l'image abw.test.8 avec le seuil de tolérance $T=80\%$ | 115 |
| 7.6 | Résultats moyens des cinq méthodes au seuil de tolérance $T=80\%$. . . | 115 |

Chapitre 1

Introduction générale

Dû à leurs méthodes complexes et imprécises d'acquisition, ainsi qu'aux orientations des surfaces d'objets qui y figurent, les images de profondeur, comparées aux images bidimensionnelles, sont considérées comme des images fortement bruitées ([Shin *et al.* \(2001\)](#)). La plupart des approches proposées pour l'analyse de ce type d'images ont échoué à produire une interprétation fiable, notamment quand elles sont utilisées avec des images réelles. L'analyse des différentes méthodes, ainsi que les études comparatives effectuées dans ce contexte ([Hoover *et al.* \(1996\)](#); [Jiang *et al.* \(2000\)](#)) ont reporté que la segmentation d'images était le maillon faible des méthodes d'interprétation proposées.

La segmentation d'images est nécessaire afin de produire une description compacte et opportune du contenu des images, indispensable pour les traitements d'appréhension de leur contenu. De ce fait, la qualité des résultats de l'interprétation dépend fortement des informations produites à la phase de segmentation. Segmenter une image consiste en l'affectation des pixels à des sous-ensembles homogènes et disjoints, formant une partition de l'image. Les pixels qui appartiennent à une même région partagent une propriété commune, dite critère d'homogénéité de région. A l'instar des méthodes générales de segmentation d'images, les méthodes de segmentation en imagerie de profondeur se divisent en deux catégories distinctes : les méthodes de segmentation en contours, et les méthodes de segmentation en régions. Pour la première catégorie, les pixels qui correspondent aux discontinuités de profondeur (contour en marche) ou d'orientation de surface (contour en toit) sont sélectionnés et chaînés dans le but de délimiter les régions de

l'image (Inokuchi *et al.* (1982); Fan *et al.* (1987); Jiang et Bunke (1999)). Les méthodes orientées contour sont bien connues pour leur coût de calcul faible ; cependant elles sont très sensibles au bruit et nécessitent des post-traitements complexes, pour le chaînage et la fermeture de contours (Sappa (2006)). Les méthodes orientées région utilisent les propriétés de surface pour grouper les pixels ayant les mêmes propriétés, dans des parties connexes et disjointes (Kang et Ikeuchi (1993); Li et Zhao (2003); Ding *et al.* (2005); Bab Hadiashar et Gheissari (2006)). Comparées aux méthodes orientées contour, les méthodes orientées région sont plus stables et moins sensibles au bruit. Cependant, leur efficacité dépend fortement de la sélection des graines initiales de régions. De plus, cette approche ne facilite pas la parallélisation et la distribution des traitements. Ceci rend le coût de calcul très élevé et ne permet pas à ces méthodes d'être utilisées pour des applications temps réel.

Dans les différentes approches de segmentation d'images, la plupart des méthodes développées modélisent les propriétés locales de surfaces en se basant sur le calcul des dérivées de différents ordres de la fonction image. Ceci conduit à une détection très sensible au bruit (Horaud et Monga (1995)). Ainsi, il est nécessaire d'effectuer des tâches de prétraitement, qui consistent principalement en un lissage de l'image ou un filtrage du bruit. Cependant, dans le cas d'images hautement bruitées, telles que les images de profondeur (Shin *et al.* (2001); Hoover *et al.* (1996)), un fort lissage de bruit peut conduire à l'effacement des contours, notamment les contours en toit et les contours lissés. Ces derniers correspondent à la discontinuité de la courbure de surface, et leur détection demeure un déficit en segmentation d'images (Jiang et Bunke (1999)). Cependant, si le bruit est sous-lissé, les distorsions qui restent dans l'image engendrent des résultats inexacts ou erronés. Ce problème, qui demeure toujours ouvert en traitement d'images (Li (2001); Mehrtash (2006)), est dû à la restriction du calcul et de la décision au voisinage local du pixel traité. En imagerie de profondeur, plusieurs méthodes récentes échouent parce qu'elles n'adressent et ne résolvent pas correctement ce problème (Jiang *et al.* (2000); Bab Hadiashar et Gheissari (2006)).

Pour remédier à cette difficulté, certains auteurs ont proposé des systèmes multi-agents pour la segmentation d'images. Dans ces systèmes, un agent, représentant une entité logicielle autonome, possède des capacités limitées de perception et d'action, et n'est pas conçu pour résoudre un problème dans sa totalité. Les agents s'auto-organisent

et coopèrent donc pour fournir une solution collective. Contrairement aux systèmes conventionnels, les solutions dans les systèmes multi-agents, émergent des actions collectives effectuées au sein de la population des agents (Krishnamurthy et Murthy (2006)). Cependant, la plupart des systèmes multi-agents qui ont été proposés pour la segmentation d'images procèdent selon une approche supervisée et sont spécifiques aux contenus des images traitées. Ils sont dédiés en majorité aux images à niveau de gris avec connaissance préalable du nombre et des propriétés des régions.

Dans ce manuscrit, nous présentons une approche distribuée non supervisée, basée sur le paradigme agent, pour l'interprétation des images, ainsi que l'application de cette approche aux images de profondeur. Deux systèmes multi-agents correspondant respectivement aux traitements de bas niveau (lissage et segmentation) (Mazouzi *et al.* (2004, 2005a, 2006, 2007a)), et de haut niveau (reconstruction et reconnaissance) (Mazouzi *et al.* (2005b)) seront présentés.

Le premier système est basé sur une approche réactive, où les agents sont conçus pour se déplacer sur l'image et agir sur les pixels situés sur les pourtours des régions. Les agents s'adaptent aux régions de l'image, puis lissent les pixels appartenant à ces régions. Par conséquent, ils se trouvent en compétition sur les bordures entre les régions. L'alignement alternatif des pixels des bordures, qui résulte des actions des groupes compétitifs d'agents, préserve ces bordures contre l'effacement. Les régions de bruit qui sont caractérisées par de faibles tailles ou par des profondeurs aberrantes ou aléatoires, ne permettent pas aux agents de s'adapter. Ces régions se contractent continuellement, et ce par l'alignement de leurs pixels aux vraies régions qui les entourent.

L'objectif de la mise en oeuvre de ce système est de surmonter la difficulté due à la restriction de la perception au seul voisinage immédiat du pixel traité. En effet, selon notre approche, un pixel est traité non uniquement en fonction de son voisinage, mais aussi en fonction des états des agents qui visitent ce pixel. La mémoire d'un agent représente une perception plus large qui, lorsqu'elle est combinée avec l'information locale de l'image, permet une décision plus fiable, en prenant en compte des données globales.

Dans le but d'optimiser les déplacements des agents, un champ de potentiel inspiré

du champ électrostatique est utilisé. Il permet aux agents de s'auto-organiser et de rationaliser leurs mouvements en se regroupant autour des régions d'intérêt (contours et bruit) et à concentrer leurs actions autour de ces régions.

Malheureusement, l'alignement compétitif des pixels appartenant aux pourtours de régions, ainsi que l'autonomie d'action inhérente aux agents, conduisent à la production de contours déformés et mal localisés. Ces contours seront donc corrigés en utilisant une régularisation bayésienne. Nous introduisons pour cela un nouveau modèle du champ aléatoire de Markov (Markov random field : MRF). Contrairement aux modèles de Markov, utilisés dans la littérature ([Li \(2001\)](#)), le modèle introduit utilise à la fois le smoothness des contours et le smoothness des surfaces, comme information à priori pour la régularisation bayésienne. Une seconde approche d'amélioration de contours est également proposée. Elle est basée sur la régularisation floue des contours détectés. Les deux approches d'amélioration sont ensuite discutées et comparées.

Considérés comme des systèmes auto-organisés, les réseaux de neurones artificiels, et notamment les réseaux de neurones chaotiques, ont fait l'objet d'une étude et une réalisation dans le cadre de cette thèse. Après avoir survoler brièvement les travaux de référence en segmentation d'images par les réseaux de neurones, nous exposons un modèle particulier de réseaux de neurones chaotiques, et nous montrons comment ce modèle a été appliqué pour la détection de contours dans les images de profondeurs. Le principe du modèle proposé consiste à utiliser une propriété locale de l'image comme stimulus d'une paire de neurones ([Sinha et Bazak \(1998\)](#)), afin d'agir sur le régime chaotique en le réduisant et le faire passer d'un régime chaotique à un régime périodique. Le modèle du réseau de neurones a permis la détection des contours lissés, cependant il a été constaté qu'il était très sensible au bruit. Par ailleurs, les réseaux de neurones ne nous ont pas permis la prise en charge des traitements de niveau moyen et de haut niveau en interprétation d'images. La formalisation symbolique de l'information visuelle en ces niveaux, et qui est souhaitable pour la représentation des formes et la reconnaissance des objets, ne peut être envisagée en utilisant les réseaux de neurones. Ceci nous a motivé de faire recours aux systèmes multi-agents, et d'exploiter l'aspect auto-organisationnel de ces systèmes pour proposer de nouvelles méthodes de segmentation d'images et de reconnaissance d'objets.

Dans le but de prendre en charge les traitements de haut niveau, spécifiques aux images de profondeur, nous avons proposé dans le cadre de cette thèse un second système multi-agents, qui permet la reconnaissance d'objets à partir d'images de profondeur. Utilisant les résultats de détection de contours produits en phase de segmentation, ce système procède en même temps à la reconstruction des primitives géométriques et à la reconnaissance des objets, en entretenant les deux processus. Le principe consiste à l'utilisation d'agents autonomes, dits agents reconnaisseurs, explorant l'image binaire contenant les points de contours. En reconstruisant des patterns locaux de matching composés de primitives adjacentes, chaque agent établit des hypothèses locales de matching objet-modèle. Des agents de contrôle confrontent les différentes hypothèses locales de matching et établissent une hypothèse globale de correspondance objet-modèle. Les hypothèses globales de correspondance sont utilisées par les agents reconnaisseurs de telle sorte que ces derniers puissent prévoir les primitives qui ne sont pas encore parcourues par des agents. Les agents qui sont libres seront donc orientés à parcourir ces primitives et construire de nouvelles hypothèses locales de matching. Le processus se termine lorsque toutes les primitives géométriques ont été parcourues.

L'objet du travail que nous avons mené le long de cette thèse est de montrer que malgré la simplicité des modèles numériques et géométriques utilisés pour la représentation des données image et des primitives géométriques, les approches d'interprétation d'images proposées sont meilleures, comparées aux approches conventionnelles. Nous croyons que l'interaction entre les agents fournit une solution alternative pour l'analyse et la compréhension d'images, par rapport aux méthodes classiques basées sur des modèles complexes et coûteux en temps de calcul (Li (2001)). L'utilisation d'agents réactifs et faiblement couplés permet de plus à notre approche d'être parallélisable et ainsi d'être utilisée dans des applications temps réel. Des expérimentations intensives ont été effectuées en utilisant des images réelles provenant de la base de données standard ABW (Hoover *et al.* (1996)). Les résultats expérimentaux obtenus montrent le potentiel des approches proposées pour une interprétation efficace et fiable des images de profondeur.

Les chapitres suivants de la thèse sont organisés comme suit :

Dans le chapitre 2 nous analysons quelques contributions typiques en vision par les systèmes multi-agents, en montrant les spécificités de ces contributions, ainsi que

leurs limitations majeures. Nous focalisons la discussion sur les aspects multi-agents et comment sont ils utilisés pour surmonter les problèmes liés à l'analyse et l'interprétation des images.

Le chapitre 3 est consacré au modèle multi-agents proposé pour la segmentation d'images et son application aux images de profondeur. Nous présentons également dans ce chapitre une méthode de coordination spatiale des agents, basée sur un champ de potentiel artificiel.

Le long du chapitre 4, nous abordons l'amélioration de la segmentation collective d'images. Nous présentons deux approches distinctes basées respectivement sur la régularisation bayésienne, et la régularisation floue des contours.

Le chapitre 5 est consacré à un modèle chaotique de réseaux de neurones, et son application pour la détection de contours dans les images de profondeur.

Au chapitre 6 nous présentons un modèle multi-agents pour la reconnaissance d'objets à partir d'images de profondeur. Dans ce chapitre, nous essayons de contraster notre approche, distribuée et basée sur l'entrelacement des phases de reconstruction et de matching, par rapport aux approches classiques basées sur un schéma centralisé et séquentiel.

Le chapitre 7 présente les résultats des expérimentations effectuées dans le cadre de cette thèse. Nous présentons respectivement le framework comparatif utilisé, l'apprentissage des différents paramètres, et les différents résultats obtenus. Une étude comparative est également présentée permettant de situer les performances de notre approche par rapport à d'autres approches typiques publiées dans la littérature.

Finalement, nous achevons ce manuscrit par une conclusion générale dans laquelle nous résumons nos contributions et nous traçons certaines lignes directrices pour de futurs travaux en vision par les systèmes auto-organisés, notamment les systèmes multi-agents.

Chapitre 2

Approches multi-agents en vision par ordinateur

2.1 Introduction

Les systèmes multi-agents pour la résolution de problèmes en vision par ordinateur héritent les avantages de l'approche agent, notamment la résolution d'un problème d'une manière collective et distribuée. Dans de tels systèmes, chaque agent à lui-même, par manque de compétence générale ou par manque de données globales, ne peut pas résoudre un problème dans son intégralité. Les agents sont donc appelés à interagir et à coopérer afin de combler leur insuffisance personnelle, en terme de compétence et de perception. L'évolution de la dynamique du système, qui résulte de l'interaction des agents entre eux et de leur action sur leur environnement, permet de faire émerger une solution collective au problème. Avant de procéder à l'analyse des systèmes multi-agents en vision par ordinateur, les plus référenciés dans la littérature, nous donnons un aperçu général sur les systèmes multi-agents.

2.2 Les systèmes multi-agents

Dans les systèmes multi-agents, un agent est communément considéré comme une entité software qui, contrairement à un objet, exhibe les caractéristiques suivantes (Wooldridge et Jennings (1995); Wooldridge (2002)) :

- L’agent est autonome, et possède son propre *thread* d’exécution. Il n’est par conséquent soumis à aucun contrôle centralisé d’exécution.
- Il est situé dans un environnement particulier qui peut sentir et sur lequel il peut agir.
- Un agent est proactif, et n’agit pas uniquement en réponse aux événements en provenance de son environnement. Il exhibe un comportement dirigé par un but, où l’agent est en mesure de prendre de l’initiative (Zambonelli *et al.* (2003)).
- Un agent est sociable car il interagit avec d’autres agents, directement en utilisant un langage de communication d’agents ”Agent Communication Language : ACL” (Genesereth et Ketchpel (1994)), ou indirectement via l’environnement.

2.2.1 Définitions

Afin de bien montrer la spécificité d’un agent, comparé à un programme informatique ordinaire, nous donnons plusieurs définitions, citées dans des travaux de références en agents et systèmes multi-agents. Le concept qui revient avec la plupart de ces définitions est l’autonomie.

- Selon Shoham (1993), ”Souvent, quand les gens utilise le terme ’agent’ ils se réfèrent à une entité qui fonctionne d’une manière continue et autonome, et ce dans un environnement dans lequel d’autres processus peuvent avoir lieu, et d’autres agents existent”.
- Russell stipule qu’un agent est une entité qui sent et qui agit sur son environnement (Russell (1997)).
- Selon la définition de l’agent ’MuBot’, le terme ’agent’ est utilisé pour représenter deux concepts orthogonaux. Le premier est la capacité de l’agent pour s’exécuter

d'une manière autonome. Le second est la capacité de l'agent d'exécuter le raisonnement orienté domaine d'application.

- Selon IBM, les agents intelligents sont des entités logicielles qui effectuent un certain jeu d'opérations au profit d'un utilisateur ou un autre programme, avec un certain degré de liberté ou d'autonomie et ainsi, utilisent des connaissances ou des représentations des objectifs ou des désirs des utilisateurs.
- Selon Franklin et Gasser, un agent autonome est un système placé dans un environnement dont il fait partie. L'agent sent l'environnement et y agit, dans la poursuite de son propre but ([Franklin et Graesser \(1996\)](#)).

Suite aux définitions précédentes, nous pouvons constater que certains auteurs ont essayé de donner une définition de l'agent comme une entité isolée. Leurs travaux ont principalement concerné les architectures internes des agents et leur capacité d'action ou de mobilité. Cependant, d'autres auteurs voient les agents comme entités agissant collectivement avec d'autres agents d'où le paradigme "multi-agents" ([Carabelea et al. \(2003\)](#)).

Formellement, Ferber a défini un système multi-agent comme un système caractérisé par les éléments suivants ([Ferber \(1995\)](#)) :

- Un espace métrique E , dit environnement ;
- Un ensemble d'objets O , incluant les objets passifs et les objets actifs (agents) ;
- Un ensemble d'agents A , sous ensemble de O ;
- Un ensemble de relations R , défini entre les objets de l'ensemble O ;
- Un ensemble d'opérations Op permettant aux agents d'agir sur les objets et sur l'environnement.

Selon ces éléments, dans un système multi-agents, l'agent qui peut être une entité matérielle ou logicielle dotée d'une capacité de perception et d'action, est placé dans l'environnement dans lequel il pourra à tout moment être situé. Afin d'interagir et de coopérer, les agents communiquent entre eux, et ce soit d'une manière explicite en utilisant l'information partagée ou l'envoi de message ; soit d'une manière implicite en utilisant des marques que l'on dépose dans l'environnement. Ainsi, les agents sont dotés d'un comportement autonome, conditionné par leur perception, leur connaissance, et leur compétence.

2.2.2 Classification

Suivant leur granularité et leur méthode de représentation des connaissances, les agents sont classés en deux catégories distinctes :

1. Les agents cognitifs sont caractérisés par une granularité forte. Ils utilisent un niveau élevé de représentation de connaissances ; d'où leur capacité de raisonnement et de négociation. Ces agents utilisent des langages de haut niveau, leur permettant d'échanger leurs connaissances et de coordonner explicitement leurs actions.
2. Les agents réactifs constituent l'alternatif aux agents cognitifs pour la modélisation des comportements intelligents dans les systèmes à base d'agents. Il s'agit d'entités simples, qui réagissent d'une manière réactive aux stimuli qui proviennent de leur environnement, en changeant de comportement et/ou en agissant sur les objets de l'environnement. Ces agents ne disposent pas d'une représentation symbolique de leur monde et par conséquent ne disposent pas de mécanismes complexes de raisonnement. L'intelligence dans les systèmes multi-agents réactifs, consiste en une propriété émergente de l'activité entière du système. Souvent on s'inspire des comportements de larges sociétés d'êtres simples, telles que les insectes, les poissons et les oiseaux ([Drogoul \(1994\)](#)).

2.3 Systèmes multi-agents pour l'analyse et l'interprétation d'images

La plupart des travaux, ayant proposé des systèmes multi-agents en vision par ordinateur, ont opté pour l'approche d'agents situés. Ce choix est motivé principalement, par le fait que l'information brute traitée en vision artificielle est locale, située au niveau des pixels de l'image. L'image représente l'environnement dans lequel les agents sont placés pour l'explorer à la recherche des différents objets qui représentent son contenu. Selon ce schéma, les agents qui, d'une manière ou d'une autre, parcourent l'image à la recherche de caractéristiques géométriques bien déterminées, interagissent et

coopèrent pour inter-échanger mutuellement l'information qui leur manquait, et prendre des décisions concernant les sites de l'image analysée. Dans ce qui suit, de cette section, nous présentons, l'essentiel de quelques travaux, récents, utilisant l'approche multi-agents pour la segmentation d'images.

2.3.1 Segmentation adaptative d'images par agents distribués

Un système multi-agents pour la segmentation d'images à résonance magnétique (IRM) du cerveau, était proposé par Liu et al. (Liu et Tang (1999)). L'image est considérée comme l'environnement dans lequel les agents évoluent. Au cours de leurs parcours, les agents étiquettent les pixels de l'image selon leur degré d'appartenance aux différentes régions définissant le tissu cérébral. Un nombre de classes d'agents, égale au nombre de régions homogènes, représentant les différents tissus, était utilisé. Pour les IRM du cerveau, trois classes d'agents sont utilisées et qui correspondent aux trois tissus qui figurent dans ce type d'images.

Chaque agent exhibe deux comportements réactifs, en fonction de sa perception locale, qui consiste à l'examen du pixel sur lequel il est situé (Fig. 2.1). S'il s'agit d'un pixel de la région homogène (reconnue par l'agent), ce dernier crée un nombre donné d'agents clones et les place dans l'image selon une certaine direction. Ensuite, l'agent étiquette le pixel en question et puis se détruit. Les agents créés héritent les comportements de l'agent père. Par contre, si le pixel sur lequel l'agent est situé n'appartient pas à sa région homogène, l'agent continue à parcourir l'image selon une direction calculée à partir de la direction de son agent père et des clones de sa génération ayant réussi à trouver des pixels appartenant à leur région respective.

Stimulus local d'un agent

Pour calculer l'intensité du stimulus, un agent perçoit un voisinage autour du pixel sur le quel il est situé. En considérant ce voisinage, le contraste moyen, la moyenne, et

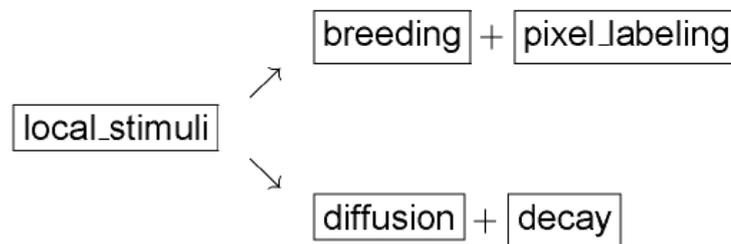


FIG. 2.1 – Comportement d’un agent en fonction du stimulus local

l’écart-type des niveaux de gris sont calculés. En fonction du stimulus perçu par l’agent, ce dernier exhibe trois comportements différents :

Génération d’agents clones

Un agent d’une génération donnée ” g ” situé sur un pixel considéré appartenant à la région homogène qui lui y est associée, génère un certain nombre d’agents clones qui forment la génération ” $g+1$ ”. Les agents créés sont placés selon une direction privilégiée.

Etiquetage de pixels

A la rencontre d’un pixel considéré appartenant à la région homogène associée à l’agent, ce dernier étiquette le pixel en lui associant l’étiquette de la région correspondante, et s’auto-détruit immédiatement.

Discussion

Dans leur article, les auteurs n’ont pas montré comment le comportement décrit des agents améliore les résultats de segmentation d’IRM du cerveau, comparé aux méthodes traditionnelles basées sur le calcul des moyennes des niveaux de gris, correspondant aux différents tissus. Par ailleurs, les agents dans ce système ne communiquent pas, que ce soit directement par envoi de messages, ou indirectement via l’environnement (l’image).

La seule interaction entre les agents consiste au fait qu'un agent peut créer des agents clones dans son voisinage.

2.3.2 Un système immuno-multi-agents pour le traitement d'images biologiques

En utilisant le langage multi-agents oRis (Harrouet *et al.* (2002)), Rodin et al. ont proposé un système multi-agents formé d'agents réactifs pour la détection de contours dans des images biologiques (Rodin *et al.* (2004)). En fonction de certains a priori sur le contenu des images, le système vise à produire une détection de contours meilleure que celle obtenue par les détecteurs traditionnels. Deux types d'agents dits respectivement agents de noircissement et agents de luminance suivent respectivement les régions sombres et les régions claires. Leurs actions visent à renforcer les régions par l'accroissement de leur contraste, permettant ainsi une détection fiable de ces régions. Un immuno-mécanisme de régulation des systèmes multi-agents était utilisé pour détecter l'état final de la dynamique du système.

Architecture des agents

Pour une application de détection d'anneaux concentriques, tels que les anneaux d'estimation de l'âge de certaines espèces biologiques (Fig. 2.2a), les agents avaient une architecture réactive représentée à l'aide du formalisme BRIC - *Block-like Representation of Interactive Components* - (Ferber (1999)). Chaque agent dispose de deux actionneurs et de trois capteurs (Fig. 2.2b).

Le premier actionneur permet à l'agent de faire des rotations autour de lui-même, ou d'avancer dans l'image. Le second lui permet d'agir sur l'image, en augmentant ou en diminuant l'illuminance d'un pixel, dans l'objectif de renforcer l'illuminance ou le noircissement de l'anneau associé. Les capteurs permettent à l'agent de mesurer l'illuminance des pixels situés dans son voisinage. Le premier ($sensor_1$) lui permet de savoir s'il est situé sur un anneau déjà détecté ou non. Les deux autres ($sensor_2$ et

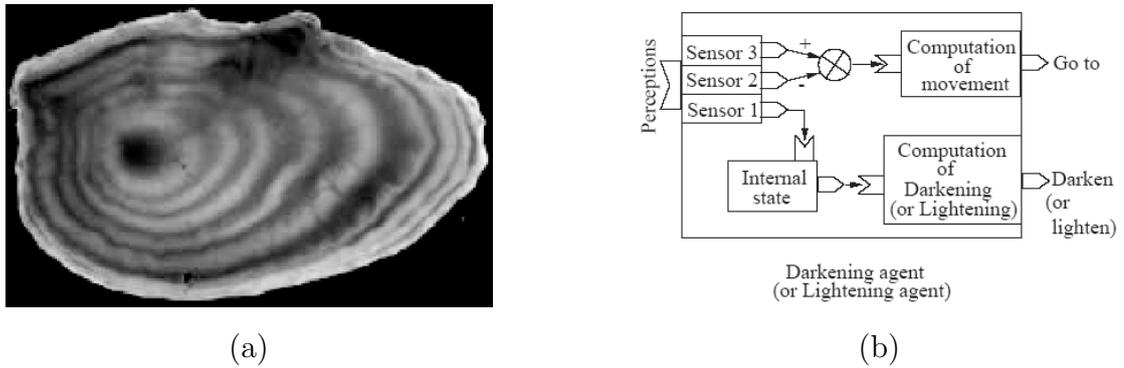


FIG. 2.2 – Agents réactifs pour la détection d’anneaux concentriques. (a) Plaice otolith de 8 ans; (b) Architecture BRIC d’un agent

$sensor_3$), permettent à l’agent de mesurer les illuminances moyennes des deux zones de pixels situées à une certaine distance (L) en avant et écartées d’un certain angle (θ) (Fig. 2.3). L’agent, en calculant l’écart des deux moyennes, est capable de maintenir son parcours sur l’anneau qu’il lui y était associé.

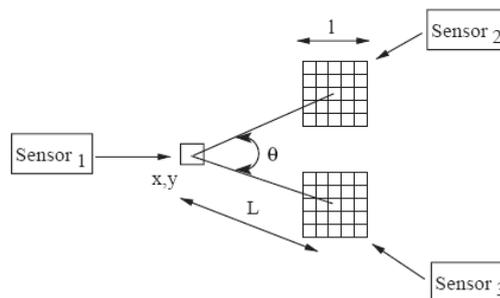


FIG. 2.3 – Capteurs d’un agents

Dynamique du système

En fonction de l’information reçue de l’environnement à l’aide de ses capteurs, et de son état interne, l’agent prend une décision et l’exécute par le moyen de ses effecteurs. Son action consiste à renforcer l’anneau sur lequel il se déplace en augmentant l’illuminance des pixels correspondants lorsqu’il s’agit d’un anneau clair, ou en diminuant l’illuminance lorsqu’il s’agit d’un anneau sombre. L’agent s’arrête et s’auto-détruit lorsqu’il revient à sa position initiale, après avoir effectué le tour de l’anneau, en fournissant le polygone correspondant à son parcours.

Discussion

Dans cette approche, les agents parcourant les différents contours sont totalement indépendants, et ne présentent aucune forme d'interaction ou de coopération entre eux. Le système paraît comme un algorithme parallèle de segmentation qui a été bien optimisé pour la détection des contours de type toit (roof edges) dans certains types d'images. Cependant, il peut échouer à détecter des contours plus simples tels que les contours de type marche. De plus, le nombre et la topologie des régions attendues doivent aussi être préalablement connus et figés dans les agents. Ceci est une limitation majeure de l'approche, car elle dépend fortement de la forme des objets qui peuvent exister dans les images traitées.

2.3.3 Un système co-évolutionnaire pour la segmentation d'images

Le travail publié par [Chen et Chen \(2005\)](#) fait partie d'un ensemble vaste de travaux ayant utilisé l'inspiration de la vie naturelle, notamment les phénomènes sociaux d'insectes, pour concevoir des systèmes robustes et parfois à fonctionnalité émergente pour l'analyse et la compréhension d'images ([Ouadfel et Batouche \(2002\)](#); [Ghosh *et al.* \(2006\)](#); [Chen et Chen \(2005\)](#)). Les auteurs ont proposé un algorithme co-évolutionnaire pour la segmentation d'images. Les agents utilisés sont placés arbitrairement dans l'image, considérée comme un écosystème. Au cours de l'évolution, les agents s'organisent en tribus et échangent entre eux les propriétés locales de l'image, perçues dans leur voisinage. Le but de la co-évolution des agents est de minimiser l'intra-variabilité des propriétés locales au sein de chaque tribu et de maximiser l'inter-variabilité de ces propriétés entre les différentes tribus. À l'équilibre du système, les tribus représentent les différentes régions de l'image.

Modélisation de l'environnement

L'image à segmenter est considérée comme un écosystème, où les segments de l'image correspondent aux différentes tribus qui habitent l'environnement. Les agents représentent les individus de l'écosystème, où chacun est caractérisé par son identité, qui est représentée par un vecteur de propriétés. Les propriétés d'un agent sont celles du pixel sur lequel il est situé, à savoir : le niveau de gris, l'illuminance, la texture, etc.

Le long de l'évolution, chaque agent communique avec ses voisins afin d'échanger avec eux les informations d'identité (Fig. 2.4). En communiquant, chaque agent essaye d'influencer les autres, en ajustant en même temps son identité par une mise à jour de son vecteur de propriétés. Après plusieurs cycles (époques), les vecteurs de propriétés des agents tendent à devenir constants, et se forment des tribus stables. A la convergence, chaque tribu correspond à une région de l'image.

Mécanisme d'évolution

Le processus d'évolution se déroule en plusieurs époques. A chaque époque, un agent qui est situé sur un pixel donné, communique avec les agents situés sur les pixels voisins, et échange de l'information avec eux en utilisant des poids de communication proportionnels aux distances entre les agents. Afin de ne pas lisser les frontières entre les segments distincts (contours), deux agents ne s'influencent mutuellement que si la différence d'identité, calculée par l'écart entre les vecteurs de propriétés correspondants, est inférieure à un certain seuil (Fig. 2.4c).

Discussion

Les systèmes inspirés de la vie naturelle sont bien connus pour leur robustesse et leur capacité d'auto-adaptation, aux imprévus qui peuvent surgir lors de l'interprétation des images. Ils sont également caractérisés par la propriété d'émergence, qui peut concerner des structures figurant dans l'image ou d'une fonctionnalité recherchée (Drogoul (1993)). Cependant, les résultats obtenus par ces systèmes sont généralement imprécis et ne

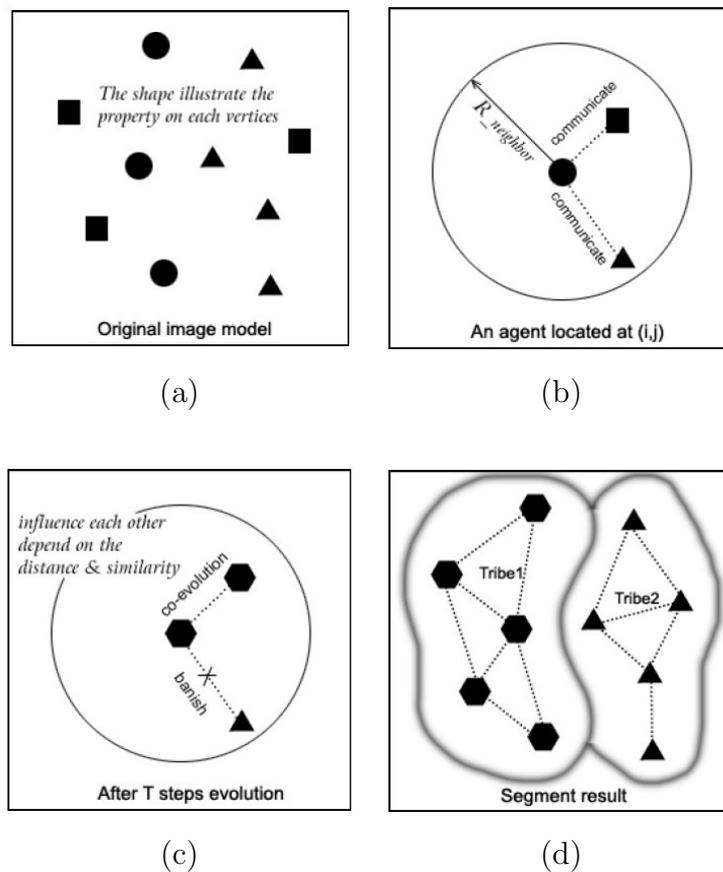


FIG. 2.4 – Co-évolution des tribus. (a) Identités initiales des agents; (b) Echange d'identités entre agents; (c) Influence mutuelle entre agents; (d) Tribus stables et homogènes

peuvent pas être directement utilisés pour réaliser des mesures exactes ou pour fournir une interprétation fiable du contenu des images.

2.3.4 Une architecture à base d'agents situés pour la segmentation d'IRM du cerveau

Dans leur approche de segmentation d'IRM du cerveau, [Richard *et al.* \(2004\)](#) ont proposé une architecture à base d'agents situés permettant de supporter l'intercalation des différentes phases de segmentation d'images, à savoir : l'estimation du modèle statistique de l'information, la classification des voxels, et le raffinement et la fusion de régions (Fig. 2.5). La distribution spatiale de l'information a permis à l'approche

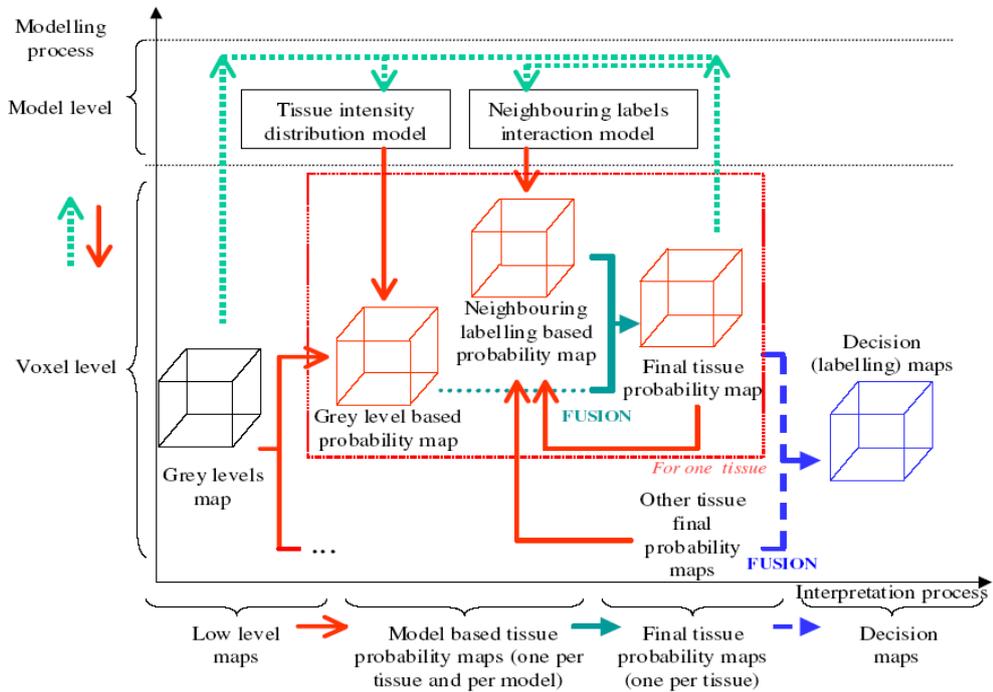


FIG. 2.5 – Phases de segmentation d’une IRM : Les modèles sont instanciés puis graduellement raffinés par intercalation itérative des processus d’estimation, de classification et de fusion

de distribuer les traitements et de les focaliser autour de certaines régions critiques telles que les frontières entre les régions représentant les différents tissus de la matière cérébrale, où l’information image est dégradée par le phénomène de mixture inhérent aux IRM (Noe et Gee (2001)).

Agents utilisés

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.

Coordination des traitements et fusion de l'information

En fonction de la disponibilité de l'information locale, les traitements sont coordonnés à l'intérieur d'un même volume local (entre agents dédiés au tissu) ou entre volumes locaux voisins (agents de contrôle local). L'estimation du modèle est renforcée en utilisant les estimations produites au niveau des volumes locaux voisins. Quand un modèle, correspondant à un volume local, est modifié, l'information est propagée aux volumes locaux voisins. A la suite de l'accroissement de régions, opéré par les agents dédiés au tissu, chaque agent de contrôle local négocie avec ses voisins pour une éventuelle fusion des volumes locaux, correspondant à une même région (d'un même tissu).

Discussion

La hiérarchisation des agents (agents de contrôle global, agents de contrôle local, et agents dédiés au tissu) a permis la distribution et la localisation des traitements, mais aussi une intégration efficace des résultats locaux de segmentation afin d'obtenir une segmentation globale, plus fiable que celles obtenues par des approches classiques. Cependant, les auteurs ont procédé à un partage ad'hoc en volumes cubiques de l'image, où la forme et le nombre de volumes locaux n'ont pas été motivés ou discutés.

2.3.5 Système multi-agents massif pour la segmentation d'IRM

Les auteurs de ce travail ([Haroun et al. \(2004\)](#)) proposent une amélioration de l'algorithme classique de segmentation par accroissement de régions, en utilisant une approche multi-agents distribuée et située. Le système multi-agents proposé, résout les problèmes d'incertitude et d'imprécision de l'algorithme classique.

Trois types d'agents associés aux trois types de tissus d'IRM du cerveau sont utilisés. Les agents sont placés d'une manière uniforme dans l'image. La position de chacun d'eux

représente la graine (seed) de la région à accroître durant le processus de segmentation. Un agent particulier dit, agent "image" (scheduler), d'une instance unique, est utilisé pour piloter les agents de segmentation, dits agents de région.

Comportement des agents

A cause de certains problèmes de mise en oeuvre (gestion des *threads*, et synchronisation des agents), les auteurs ont proposé une implémentation que nous qualifions de synchrone, pour laquelle l'agent "image" lance, à tour de rôle, les agents de région, et se met en attente à ce que le dernier agent lancé termine une tâche, puis donne le contrôle à l'agent suivant. Les agents de région exhibent les comportements suivants :

- L'agent accroît la région dont il est détenteur, par l'ajout des pixels situés dans son voisinage, si ces derniers appartiennent à la région, selon un critère d'homogénéité de région.
- Une négociation peut être initiée, par l'agent "image", entre deux agents de région pour une éventuelle fusion des deux régions, si les critères de fusion sont vérifiés. Dans ce cas, l'agent "image" crée un nouvel agent région et l'affecte à la nouvelle région.
- Les agents dont leurs régions respectives ont été fusionnées s'auto-détruisent, car ne seront plus affectés à aucune région.

Discussion

L'apport du système multi-agents n'est pas clair, du fait que le traitement est séquentiel et que les agents n'interagissent pas directement, mais passent par un agent scheduler (agent "image"). Les traitements des agents de région peuvent être délégués au agent "image" qui peut détenir des structures séparées associées aux différentes régions de l'image, ce qui amène le système au schéma classique orienté objet. L'aspect non déterministe et asynchrone relatif aux actions des agents, considéré comme inconfortable par les auteurs, est par fois un avantage recherché, du fait que les méthodes non déterministes (stochastiques) sont reconnues pour leur robustesse.

2.3.6 Surveillance multi-agents de scènes dynamiques

Les auteurs de ce travail (Remagnino *et al.* (1998)) présentent un système multi-agents pour la surveillance de scènes contenant des objets animés (piétons dans un parking) et des objets inanimés (véhicules stationnés). Pour chaque objet identifié de la scène, un agent est créé et y lui est affecté. Le système permet de suivre les objets animés et d'établir des associations en terme d'interaction entre les différents pairs d'objets situés en proximité les uns aux autres. Le système multi-agents, ainsi conçu, et en se basant sur l'information fournie par un système de détection, permet d'inférer l'interprétation la plus probable de la dynamique de la scène.

Agents du système

Les auteurs utilisent un réseau bayésien afin de représenter la dynamique des objets suivis, et l'utilisent ensuite pour inférer la meilleure (au sens bayésien) interprétation de la dynamique des objets. Selon le comportement qui exhibent et le niveau dans lequel ils opèrent, les agents du système sont divisés en deux catégories :

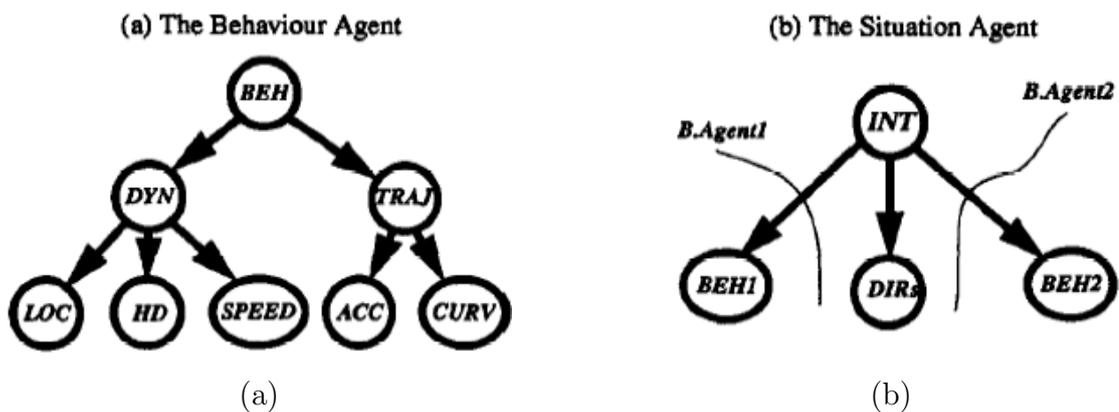


FIG. 2.6 – Réseau bayésien des agents utilisés : (a) Agent de comportement ; (b) Agent de situation.

Agent de comportement : Ce type d'agents opère au niveau objet. L'agent affecté à un objet donné de la scène est tenu de fournir une description de la cinématique de l'objet. Cette description est représentée par un réseau bayésien (Fig. 2.6a) et exprime

en premier lieu la dynamique (DYN) de l'objet en terme de sa position (LOC) et de sa vitesse (HD :head speed). En second lieu, la trajectoire de l'objet (TRAJ) est exprimée par son accélération (ACC) et sa courbure (CURV). Les noeuds cachés du réseau DYN et TRAJ représentent une interprétation de l'objet associé en terme de dynamique (DYN) et de trajectoire (TRAJ).

Agent de situation : Ce type d'agents opère au niveau inter-objets. Pour chaque paire d'objets voisins une connexion probabiliste est créée et évaluée (Fig. 2.6b). Le noeud DIRS dans la figure 2.6b représente la direction des deux objets BEH1 et BEH2. Elle exprime le fait que les deux objets correspondants sont en direction l'un vers l'autre, l'un passant à proximité de l'autre, ou sont dans la même direction.

Discussion

L'utilisation d'une approche bayésienne pour l'estimation des grandeurs exprimant la dynamique des objets dans la scène semble très pertinente du fait de l'ambiguïté d'interprétation des scènes dynamiques. Cependant, l'utilisation d'une approche multi-agents n'as pas été bien motivée. En effet, à part la distribution des traitements, en affectant à chaque objet un agent responsable de son suivi, les interactions entre les agents de comportement sont inexistantes. Pour les agents de situation, il n'a pas été explicité comment interagissent ces derniers entre eux, et où et comment l'information qui concerne l'interaction entre les objets est représentée et traitée.

2.3.7 Une architecture cognitive pour la segmentation des IVUS

En se basant sur l'architecture cognitive Soar (Newell (1990)), Bovenkamp *et al.* (2004) ont développé un système multi-agents pour la segmentation des images IntraVasculaires UltraSonores (IVUS). Leur objectif est d'établir un sous-système basé connaissances, pour le contrôle des algorithmes de traitements de bas niveau. Dans le système proposé, un agent est affecté à chaque objet prévu dans l'image. Les agents

coopèrent et adaptent dynamiquement les algorithmes de segmentation, en se basant sur la connaissance contextuelle, l'information locale, et les croyances personnelles.

Architecture du système

Afin de tirer profit de la modularisation des connaissances, chaque agent est fait responsable de la détection exclusive d'un seul type d'objets dans l'image. En considérant les différents types de régions dans de telles images, les auteurs ont défini cinq agents (Fig. 2.7b), qui sont respectivement : "lumen agent", "vessel agent", "calcified plaque agent" et "sidebranch agent", correspondant aux quatre tissus composant l'image (IVUS), ainsi qu'un agent "shadow agent" dédié aux régions d'ombre qui peuvent exister dans ce type d'images (Fig. 2.7a).

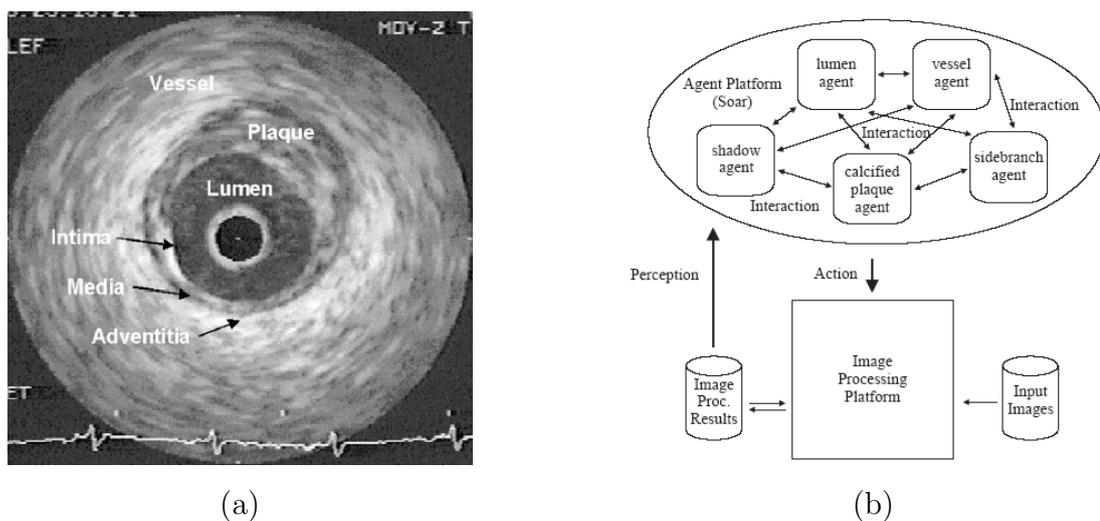


FIG. 2.7 – Un système Multi-agents pour la segmentation des images IVUS. (a) Un exemple d'image IVUS; (b) Architecture du système

Les agents s'interagissent en échangeant leurs connaissances, et agissent sur leur environnement en exécutant des tâches de traitement d'images. Ils perçoivent ce même environnement en accédant aux résultats des différents traitements effectués par les autres agents.

Les connaissances utilisées par les agents concernent les objets qui peuvent figurer dans les images traitées. Ces connaissances lorsqu'elles sont partagées permettent aux

agents de raisonner sur les différentes hypothèses possibles sur le contenu des images. Ces hypothèses peuvent être conflictuelles et impliquent donc une négociation entre les agents et un réajustement des connaissances.

Pour ce faire, un agent, en fonction de son état actuel, et des connaissances échangées avec les autres agents, s'engage à exécuter l'une des opérations suivantes :

- Raffiner : cherche à renforcer une hypothèse sur une relation de correspondance objet-image, qui a été mise en instance à cause d'un manque de connaissances ;
- Ajuster : s'exécute lorsqu'un agent reçoit une interprétation qui peut avoir une influence sur l'hypothèse actuelle de correspondance objet-image ;
- Résoudre un conflit : se déclenche lorsque l'agent reçoit une interprétation en contradiction avec l'hypothèse actuelle sur la relation objet-image ;
- Tester une hypothèse : un agent peut demander à un autre d'établir à son profit un test d'hypothèse. Ceci se produit dans le but de confirmer une hypothèse dont un agent seul ne peut le faire.

Discussion

Dans le système proposé par Bovenkamp et al., le problème du contrôle des algorithmes de traitement d'images était bien posé et résolu. Cependant, aucun agent ni comportement n'ont été proposés pour remédier au problème d'ambiguïté des données dans l'image. Le couplage fort entre les traitements de haut niveau (spécifiques au domaine) et les traitements de bas niveau (spécifiques à l'image), adopté dans ce système, ne facilite pas la réutilisation de la méthode proposée pour d'autres applications ou d'autres types d'images.

2.3.8 Synthèse

Selon les travaux cités dans la section précédente, ainsi que la plupart des travaux publiés dans la littérature, l'approche agent était plus ou moins correctement utilisée en vision par ordinateur pour résoudre les problèmes ouverts liés à l'ambiguïté d'in-

interprétation. Cette ambiguïté est due à la nature incertaine et incomplète de l'information visuelle, ainsi qu'à la présence du bruit dans les images. Par ailleurs, selon la nature de l'information traitée, les systèmes multi-agents proposés peuvent être divisés en deux catégories distinctes, à savoir : les systèmes à agents dédiés aux objets, et les systèmes à agents dédiés aux tâches.

Systèmes à agents dédiés aux objets

Pour ces systèmes, tels que ceux proposés respectivement par [Bovenkamp *et al.* \(2004\)](#), et [Remagnino *et al.* \(1998\)](#), un agent est à l'avance affecté à chaque objet attendu dans l'image. Le but de chaque agent dans ce cas est de délimiter l'objet en détectant ses contours et éventuellement assurer son suivi (en cas d'application de suivi d'objets par exemple). L'échange d'information entre les différents agents permet de renforcer les hypothèses de reconnaissance d'objets, ainsi que de bien délimiter leurs contours, et ce par négociation entre agents qui détiennent des objets adjacents ([Bovenkamp *et al.* \(2004\)](#)). Cet échange permet également aux agents de raisonner sur la cinématique des objets notamment en cas d'analyse de séquences d'images.

En terme d'architecture, les agents dans ces systèmes, sont souvent des agents cognitifs. En effet, l'agent dédié à un objet doit être doté d'une méthode de représentation des connaissances sur les objets, qui sont des connaissances de haut niveau décrivant principalement la morphologie et la topologie des objets. Ainsi ces agents sont appelés à communiquer dans des langages de communication d'agents de haut niveau ([Labrou \(1996\)](#)), et de raisonner sur les connaissances qu'ils détiennent.

Du fait que les agents dans ces systèmes sont dédiés aux objets, ces systèmes sont très liés au contenu des images pour lesquelles ils ont été conçus. Il est souvent difficile de pouvoir réutiliser ces systèmes pour d'autres types d'images, ce qui représente leur principale limitation.

Systèmes à agents dédiés aux tâches

Dans cette catégorie de systèmes, un agent est dédié à une ou plusieurs tâches concernant souvent des opérations de traitement d'images. L'image dans ces systèmes est considérée comme un environnement dans lequel les agents sont situés (Liu et Tang (1999); Haroun *et al.* (2004)). Ces agents peuvent demeurer fixes aux positions où ils ont été créés (Haroun *et al.* (2004); Richard *et al.* (2004)), comme ils peuvent se déplacer dans leur environnement (Liu et Tang (1999); Rodin *et al.* (2004)).

La perception des agents dans ce cas est liée à l'opération effectuée selon l'information visuelle traitée par l'agent. L'étendue de la perception varie d'un système à un autre. Pour certains systèmes, elle est du niveau de région, où l'agent est situé sur la graine de la région, et détient tous ses pixels (Richard *et al.* (2004); Haroun *et al.* (2004)). Pour d'autres systèmes, la perception d'un agent est limitée au voisinage du pixel sur lequel il est situé (Liu et Tang (1999); Rodin *et al.* (2004)). De tels agents sont donc contraints de se déplacer afin d'explorer toute l'image.

En terme d'architecture, les agents utilisés dans ces systèmes sont souvent de nature réactive. Ils perçoivent localement leur environnement et réagissent en fonction de leur perception. La fonctionnalité du système émerge des interactions locales entre les agents, qui communiquent usuellement par envoi de signaux à travers l'environnement. Pour les systèmes avec agents à perception plus étendue (niveau région), les agents peuvent avoir une granularité plus forte et peuvent être considérés comme agents cognitifs (Richard *et al.* (2004); Haroun *et al.* (2004)).

Les systèmes à base d'agents dédiés aux tâches sont généraux par nature, du fait que les tâches exécutées par les agents concernent l'information visuelle qui est commune à un nombre important de types d'images. Cependant, il est souvent difficile de définir l'interaction entre les agents qui permet de faire émerger la fonctionnalité désirée. Les modèles d'interaction qui ont été définis pour les systèmes vus précédemment sont ad'hoc et varient considérablement d'un type d'images à un autre.

2.4 Conclusion

Dans ce chapitre, nous avons présenté une revue des travaux les plus répandus ayant utilisé l'approche multi-agents pour l'interprétation d'images et la reconnaissance d'objets. Nous avons essayé de diversifier l'analyse afin de montrer tous les aspects caractérisant ces systèmes. Au terme de cette revue, nous avons procédé à une classification des systèmes multi-agents étudiés en deux catégories. Ceci nous a permis de ressortir les caractéristiques de chacune des catégories, mais également de bien identifier les avantages et les inconvénients qui lui sont associés. Cette étude nous a permis de bien pallier à certains problèmes lors du développement de notre approche pour la segmentation d'images de profondeur, que nous exposons en détail dans le chapitre suivant.

Chapitre 3

Un système multi-agents pour la segmentation d'images de profondeur

3.1 Introduction

Vu les problèmes spécifiques d'acquisition des images de profondeur, et vu l'inclinaison variable des surfaces qui forment les objets dans ces images, ces dernières sont considérées comme fortement bruitées, et difficile à segmenter (Hoover *et al.* (1996); Sun *et al.* (2000)). L'évaluation quantitative et rigoureuse des différentes méthodes proposées dans la littérature (Hoover *et al.* (1996); Jiang *et al.* (2000)) a montré la limitation des approches classiques à produire une segmentation correcte de ces images. Par ailleurs, les approches à base d'agents sont souvent dédiées aux images présentant un nombre fixe et prédéfini de régions, telles que les systèmes de segmentation d'IRM (Liu et Tang (1999); Richard *et al.* (2004); Haroun *et al.* (2004)). Par conséquent aucune des approches, au moins celles que nous avons examiné le long de ce travail, ne permet une segmentation des images de profondeur, car ces dernières sont caractérisées par une complexité élevée, et un nombre arbitraire de régions.

En constatant que la difficulté de segmentation pour ce type d'images est située au

niveau des contours entre les régions qui forment une image, nous avons proposé une nouvelle approche multi-agents, basée sur la compétition d'agents sur les contours de régions afin d'affecter les pixels de ces derniers aux régions dont ils forment les pourtours (Mazouzi *et al.* (2007b,a)). Nous montrons que ce pattern d'interaction, permet à la fois une élimination efficace du bruit caractéristique des images de profondeur, et la détection des contours entre les différentes régions. Les agents s'adaptent aux régions sur lesquelles ils se déplacent et coopèrent pour leur lissage. Cependant, ils se mettent en compétition sur les contours entre les régions, où chaque agent, en fonction de son état, essaye d'inclure les pixels de contour dans la région de laquelle il provient.

3.2 Images de profondeur

Une image de profondeur est une image $3D$ où à chaque pixel (x, y) correspond la distance $Z(x, y)$ entre le plan du capteur télémétrique et le point correspondant de la scène. Dans une telle image, les régions représentent les parties visibles des surfaces des différents objets figurant dans la scène. Selon la nature de données capturées, les techniques d'acquisition peuvent être classées en deux catégories distinctes (Arman *et Aggarwal* (1993)) :

- Les techniques actives, pour lesquelles un faisceau laser ou ultrasonore est utilisé. Celles-ci sont capables de mesurer directement la profondeur des objets perçus. Par conséquent, les données visuelles sont directement tridimensionnelles ;
- Les techniques passives, pour lesquelles l'énergie ambiante émise ou réfléchie, par les objets de la scène, est utilisée pour déduire les distances $3D$ de ces objets.

Les systèmes basés sur la triangulation sont des systèmes actifs utilisant la loi des sinus (Horaud *et Monga* (1995)) pour mesurer la distance vers un point de la scène (Fig. 3.1). La distance h , d'un point de l'objet, au plan de balayage est calculée comme suit :

$$h = d \frac{\tan(A) \tan(B)}{\tan(A) + \tan(B)} \quad (3.1)$$

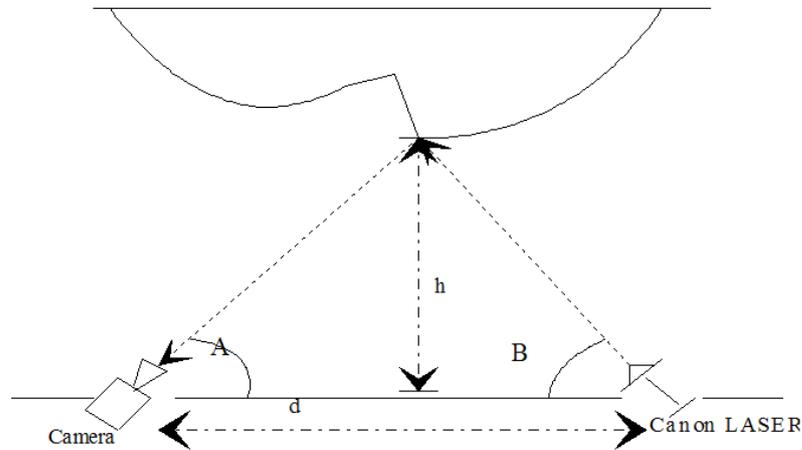


FIG. 3.1 – Principe de télémétrie par triangulation

L'acquisition consiste en l'échantillonnage spatial d'une scène observée par un faisceau de lumière concentré (généralement un faisceau laser) dont l'angle d'émission est préalablement connu. Le faisceau diffusé en chaque point éclairé de la scène, est focalisé par un objectif sur un photodétecteur (caméra), et ce dans l'objectif de mesurer l'angle de réception du faisceau réfléchi sur le point éclairé. La connaissance de l'angle d'émission et de l'angle de réception permet, par triangulation, de calculer la profondeur $Z(x, y)$ de chaque point (x, y) de la scène. Après échantillonnage de toute la scène, le système génère un tampon Z représentant l'image de profondeur de la scène observée.

Le système d'acquisition qui a été utilisé pour l'acquisition des images de la base ABW, utilisées dans notre travail, repose sur une source laser qui génère un plan de lumière qu'intersecte à son tour la scène observée selon une courbe bidimensionnelle (Fig. 3.2). La scène est observée par une caméra sous une certaine parallaxe. La déformation de la courbe dans le plan de la caméra permet, par triangulation, d'obtenir l'information tridimensionnelle. Pour balayer toute la scène, le plan de lumière est déplacé unidimensionnellement afin d'intersecter toutes les lignes visibles de la scène, ce qui permet de mesurer les profondeurs correspondantes à tous les potins appartenant à ces lignes. Le système est à base de laser et est développé à "Institute for Robotics and Intelligent Systems (IRIS)" de l'université de "Southern California" à Los Angeles (USA). Les images utilisées représentent des scènes avec des objets de complexité variables et souvent en occlusion visuelle.

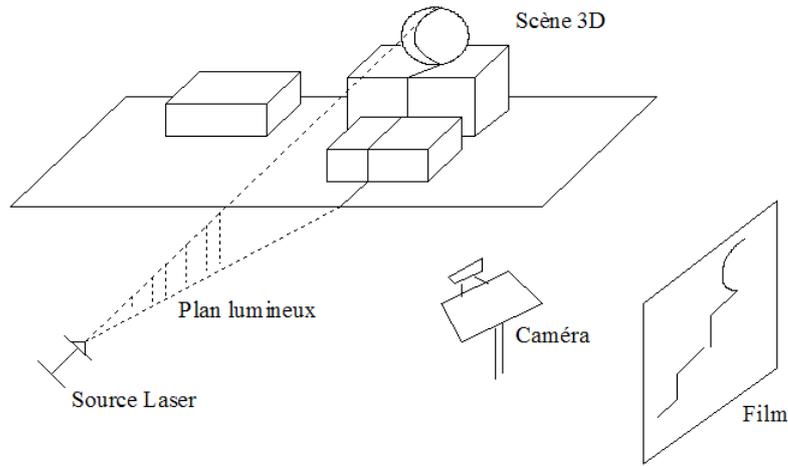


FIG. 3.2 – Principe du capteur utilisé

3.3 Modélisation et calcul des propriétés locales des surfaces

Soit $Z^*(x, y) \in R^4$, les paramètres de l'équation du plan tangent à la surface en (x, y) . L'équation du plan tangent en (x, y) est obtenue par la méthode de régression multiple en utilisant les pixels appartenant au voisinage $\chi(x, y)$ du pixel en question. Le voisinage $\chi(x, y)$ est constitué des pixels (x', y') situés dans une fenêtre de taille 3×3 centrée en (x, y) , et dont les profondeurs $Z(x', y')$ sont, à un seuil donné de profondeur (Tr_D), proches. L'équation d'un plan dans un référentiel cartésien 3D peut être exprimée comme suit :

$$ax + by + cz = d \quad (3.2)$$

où $(a, b, c)^T$ est le vecteur unitaire normal au plan ($a^2 + b^2 + c^2 = 1; c < 0$) et d est la distance orthogonale du plan à l'origine du référentiel. En premier, les coefficients α , β et γ de la surface $z = \alpha x + \beta y + \gamma$ au point (x_0, y_0) sont obtenus par la minimisation

de la fonction Φ , donnée par :

$$\Phi(\alpha, \beta, \gamma) = \sum_{(x', y') \in \chi(x_0, y_0)} [\alpha x' + \beta y' + \gamma - Z(x', y')]^2 \quad (3.3)$$

avec $\chi(x_0, y_0) = \{(x_0 + i, y_0 + j); (i, j) \in \{-1, 0, +1\} \text{ et } |Z(x_0 + i, y_0 + j) - Z(x_0, y_0)| < Tr_D\}$

Les paramètres a, b, c et d sont donc calculés comme suit :

$$(a, b, c, d)^T = \frac{1}{\sqrt{\alpha^2 + \beta^2 + 1}} (\alpha, \beta, -1, \gamma)^T \quad (3.4)$$

Les traitements effectués sur l'image se basent sur la comparaison de plans. En effet, deux plans d'équations respectives : $ax + by + cz = d$ et $a'x + b'y + c'z = d'$ sont considérés égaux s'ils ont à des seuils près, la même orientation et la même distance à l'origine du référentiel. Soit θ l'angle entre les deux vecteurs normaux, et D la distance entre les deux plans : $\sin(\theta) = \|(a, b, c)^T \otimes (a', b', c')^T\|$ et $D = |d - d'|$. Les deux plans sont donc considérés égaux si $\sin(\theta) \leq Tr_\theta$ et $D \leq Tr_D$, où Tr_θ et Tr_D sont les seuils respectivement d'angle et de profondeur, obtenus par apprentissage (voir chapitre 7). Le test d'égalité de deux plans est utilisé d'abord pour tester si le pixel à la position (x, y) appartient à une région plane, étant donnée son équation de plan. Il est également utilisé pour tester si le pixel en (x, y) est un pixel d'intérêt (de contour ou de bruit) ou non. Dans ce cas, le pixel en question est considéré comme un pixel d'intérêt si au moins un de ses voisins a une équation de plan différente, en se référant aux seuils précédents.

3.4 Segmentation collective d'images par agents réactifs

3.4.1 Comportement des agents

L'image est considérée comme un environnement dans lequel des agents sont initialisés à des positions arbitraires. Les agents utilisés ont une granularité faible, et leur nombre doit être suffisamment grand. Un agent cherche une région plane autour de sa position courante et s'adapte à cette région en mémorisant son équation de plan. Ensuite, il effectue des actions qui dépendent à la fois de son état et de l'état du pixel sur lequel il se situe. A chaque instant t , un agent est caractérisé par sa position (x_t, y_t) dans l'image, et par sa capacité C_t d'agir sur les pixels rencontrés. Au début du processus, tous les agents sont incapables d'altérer les pixels de l'image. Un agent devient capable de modifier un pixels ($C_t=true$) lorsque il détecte une région plane autour de sa position courante. Par ailleurs, si un agent vient d'altérer un pixel, il perd sa capacité d'altération ($C_t=false$) et recommence à chercher une nouvelle région plane. Chaque agent ayant modifié un pixel, enregistre dans une structure appropriée I , dite matrice d'états des pixels, à la position (x_t, y_t) le dernier état du pixel visité : $I(x_t, y_t) \in \{\text{lissé, aligné, non changé}\}$. Au lancement du system, la matrice des états des pixels I est initialisée par "non changé".

Un agent s'adapte à la région de l'image dans laquelle il se déplace en mémorisant les caractéristiques de cette région, et en adoptant le comportement adéquat en fonction des données de l'image. Nous montrons dans la suite de la section, que le comportement simple et réactif des agents permet de faire émerger les lignes de contour et d'éliminer efficacement le bruit dans l'image. La figure 3.3 schématise le comportement d'un agent en fonction de son état courant et du pixel sur lequel il est situé.

Le pseudo code exécuté par chaque agent est le suivant :

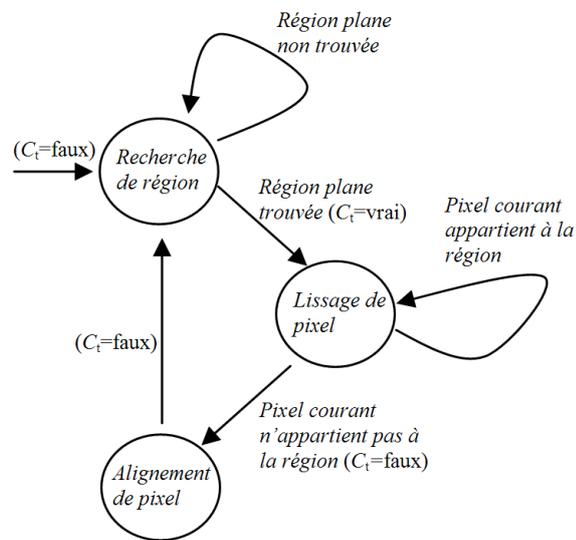


FIG. 3.3 – Le comportement d'un agent en fonction de son état et de sa position

Initialization

Repeat

Repeat

Move

Check Current Position

Until Planer region found

Memorize Current Region

Repeat

Smooth Current Pixel

Move

Check Current Pixel

Until Current pixel does not belong to the memorized region

Align current pixel to the Current Planar Region

Update Potential field

Until Process Termination

Recherche d'une région plane

Après sa création, un agent se déplace aléatoirement dans l'image et cherche une région plane autour de sa position courante. L'agent utilise les L derniers pixels parcou-

rus pour tester s'il est à l'intérieur d'une région plane ou non. L est appelée la longueur d'adaptation, et représente le degré de confiance que l'agent soit à l'intérieur d'une région plane. L'agent considère qu'il est à l'intérieur d'une région plane si les derniers L pixels parcourus appartiennent au même plan. L'agent mémorise la région trouvée et la considère désormais comme sa région plane courante. Après cela, il devient capable d'altérer le premier pixel rencontré qui n'appartient pas à cette région ($C_t=true$).

Lissage des pixels à l'intérieur d'une régions

Lors de son déplacement dans une région plane, l'agent lisse l'image au niveau du pixel sur lequel il se situe, en mettant à jour les équations respectives du plan de la région mémorisée, et du plan à la position du pixel. Ceci est fait en remplaçant les deux équations par leur moyenne pondérée. Soit (a, b, c, d) et (a', b', c', d') les paramètres respectivement du plan de la région mémorisée, et du plan au pixel courant, les paramètres résultats (a'', b'', c'', d'') du plan moyen pondéré sont obtenus comme suit :

$$(a'', b'', c'', d'') = \frac{1}{1+l}(a' + la, b' + lb, c' + lc, d' + ld) \quad (3.5)$$

où l est la longueur, en pixels, du chemin parcouru par l'agent dans la région plane.

Pour chaque pixel lissé, l'agent enregistre la valeur "lissé" dans la position correspondante dans la matrice des états des pixels I .

Alignement des pixels d'intérêt

Les pixels d'intérêt sont les pixels de contour ou les pixels à l'intérieur des régions de bruit. Un pixel d'intérêt est reconnu en testant l'égalité du plan au niveau de ce pixel, et celui de la région mémorisée au sein de l'agent (voir section 3.3). Quand un agent rencontre un pixel d'intérêt (qui n'appartient pas à sa région plane courante), il l'altère pour l'aligner partiellement à la région sur laquelle il se déplace. Les paramètres (a'', b'', c'', d'') de la nouvelle équation de plan à la position du pixel sont obtenus par la combinaison linéaire des anciens paramètres (a, b, c, d) et des paramètres de l'équation

de plan de la région mémorisée (a', b', c', d') :

$$(a'', b'', c'', d'') = \frac{1}{1 + \xi}(a + \xi a', b + \xi b', c + \xi c', d + \xi d') \quad (3.6)$$

où ξ est la force d'altération. Après, l'agent enregistre la valeur "aligné" à la position correspondante dans la matrice des états des pixels I .

Après cela, l'agent perd sa capacité d'altération ($C_t=false$) et recommence à chercher une nouvelle région plane. Suite à l'altération d'un pixel, l'agent peut passer dans une autre région ou rester dans la région précédente. Si le pixel altéré est un pixel de contour, il est plus probable que l'agent passe dans une autre région plane. Par contre, si le pixel altéré appartient à la bordure d'une région de bruit, l'agent traverse cette région et plus probablement retourne à la région plane précédente, sauf si la région de bruit est située entre deux régions planes différentes. La force d'altération ξ est un paramètre critique qui affecte la qualité des résultats et le temps de calcul. En effet, le choix de valeurs élevées pour ξ conduit à une détection rapide de régions. Cependant, les bordures des régions qui en résultent sont déformées (figure 3.4b). Le choix de valeurs basses de ξ conduit à une détection lente. Néanmoins, les bordures des régions, dans ce cas, sont bien détectées et correctement localisées (figure 3.4c). Pour accélérer le processus de segmentation sans déformation des contours, un agent choisit la force d'altération parmi ξ_{min} et ξ_{max} en fonction de l'information enregistrée dans la matrice des états de pixels I . En effet, l'agent suppose que la région courante est adjacente à une région de bruit, et alors utilise ξ_{max} comme force d'altération, s'il existe au moins 3 pixels (x', y') étiquetés "non changé" ($I(x', y')="non\ changé"$) autour de l'agent. Sinon, l'agent suppose que la région plane courante est adjacente à une autre région plane, où des agents ont étiqueté les pixels comme "lissés" ou "alignés". Dans ce cas, l'agent utilise la force d'altération ξ_{min} . Les constantes ξ_{min} et ξ_{max} sont deux paramètres du système dont les valeurs sont déterminées par apprentissage (voir chapitre 7).

3.4.2 Détection de contours

Lors de son déplacement sur l'image, un agent lisse les pixels qui appartiennent approximativement à la région plane sur laquelle il se situe actuellement. Un agent considère les pixels n'appartenant pas à cette région comme des pixels de bruit. Ces

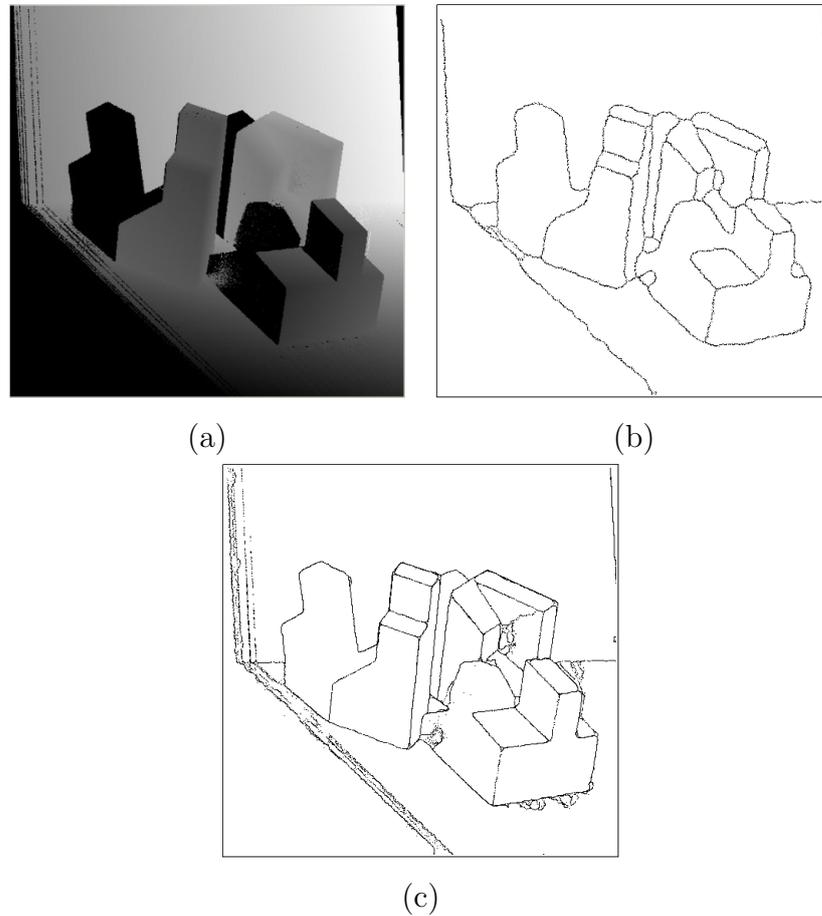


FIG. 3.4 – Impact de la force d'altération sur les résultats de segmentation : (a) Image de profondeur (abw.test.8); (b) Résultats de segmentation avec $\xi_{min} = \xi_{max} = 4$ à $t=2500$; (c) Résultats de segmentation avec $\xi_{min} = 0.3$ et $\xi_{max} = 5$ à $t=13000$

derniers sont donc automatiquement alignés à la région plane qui les entoure. Cependant, les pixels des frontières entre les régions planes sont de vrais pixels de contours et donc ne devraient pas normalement être alignés. Néanmoins, sur le segment de contour entre deux régions adjacentes, deux groupes concurrents d'agents se forment. Chaque groupe est formé d'agents qui transitent d'une région à une autre. Les agents de chaque groupe alignent les pixels de la frontière à leur région plane courante. Ainsi, ces pixels se retrouvent continuellement permutés entre les deux régions adjacentes, et demeurent ainsi distingués dans l'image. Cette compétition entre les agents permet également l'amincissement des contours de type toit et des contours lissés qui existent entre les régions adjacentes d'un même objet. Au début, un tel contour est gras, formé d'une bande de pixels qui n'appartient à aucune des deux régions adjacentes (figure 3.5b). Durant le processus, les pixels sur chaque côté de la bande sont continuellement alignés à la

région dont ils forment la bordure. Les agents qui passent à l'intérieur de cette bande ne peuvent pas s'adapter et quittent la bande sans altérer ses bordures. La bande continue donc à se rétrécir jusqu'au point où il ne reste que les pixels qui sont en son centre. Ces pixels représentent la ligne de compétition entre les deux groupes d'agents, qui ne sera alignée à aucune des deux régions (figure 3.5c).

Au bout d'un certain temps, il ne reste distingué dans l'image que les pixels de bordure entre les régions adjacentes qui continuent à se permuter entre les régions. Ce schéma d'action compétitif entre les agents permet de faire émerger les lignes de contours. Ce résultat qui n'est codé dans aucun agent, émerge de l'action collective de tous les agents sur les pixels de l'image.

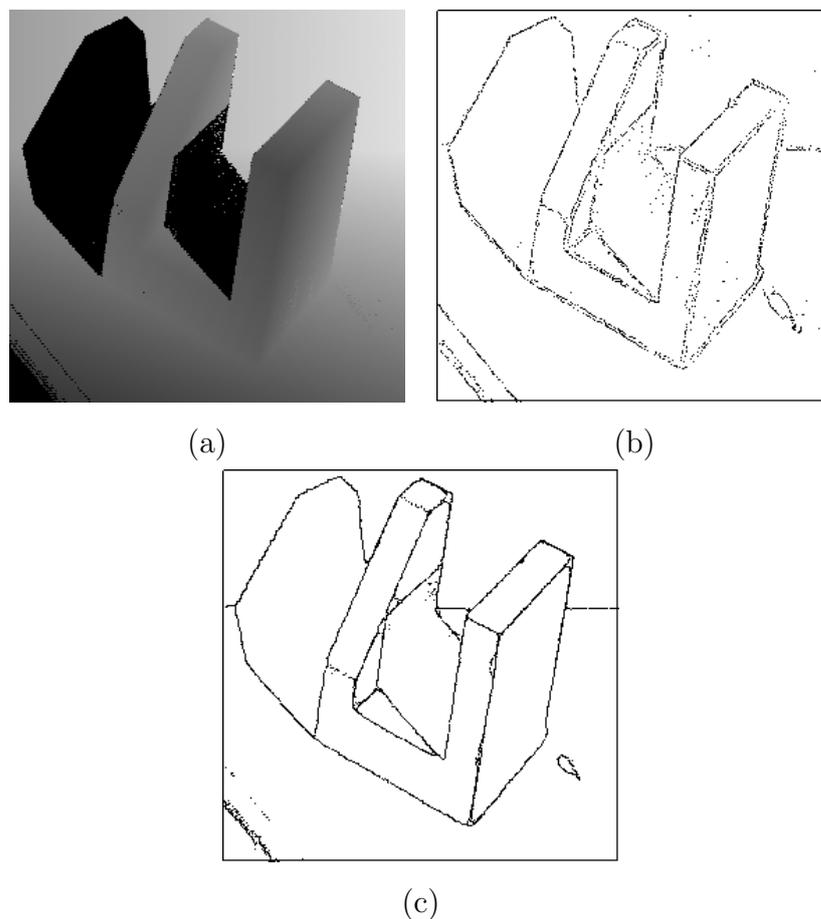


FIG. 3.5 – Amincissement de contours (image abw.test.22) : (a) Image de profondeur ; (b) Pixels de contour à $t=800$; (c) Pixels de contour à $t= 8000$

3.4.3 Élimination du bruit

Contrairement aux vraies régions de l'image, qui restent préservées contre l'effacement, les régions de bruit se contractent continuellement, et finissent par disparaître. Les bordures de ces dernières sont continuellement alignées aux vraies régions planes environnantes. Chaque agent ayant aligné un pixel appartenant à la bordure d'une région de bruit, et s'est déplacé à l'intérieur de cette région, ne sera pas capable de s'adapter (ne se trouve pas au milieu d'une région plane). Par conséquent, il ne peut aligner aucun pixel en quittant cette région. Ceci se produit dans deux cas différents : 1) lorsque la région de bruit est suffisamment large mais non plane, ou formée de pixels ayant des profondeurs aléatoires ; 2) lorsque la région est plane mais insuffisamment large pour permettre aux agents de traverser la longueur minimale L nécessaire pour pouvoir s'adapter. Dans les deux cas, l'agent quitte la région de bruit et s'adapte dans la région plane qui entoure cette dernière. A l'intérieur d'une vraie région de l'image, un agent traverse une longueur suffisante lui permettant de s'adapter à cette région. Cependant, un agent ne peut pas s'adapter à l'intérieur d'une région dont la surface n'est pas plane, et ce, quelle que soit la taille de cette région. Il ne peut également pas s'adapter à l'intérieur d'une région plane si la taille de cette dernière ne lui permet pas de traverser une longueur supérieure ou égale à la longueur minimale L . Dans les deux derniers cas, les agents ne peuvent pas s'adapter, et par conséquent ne seront pas capables d'aligner les pixels de bordure en quittant ces régions. Autrement dit, les vraies régions ont des tailles suffisamment grandes qui permettent aux agents qui sont à l'intérieur de s'adapter et d'aligner les pixels de bordure en quittant ces régions. Par contre, les régions de bruit, qui sont des régions non planes ou ayant de faibles tailles ne permettent pas aux agents de s'adapter. Par conséquent les agents n'alignent pas les pixels de bordure de ces régions en les quittant. Dans ce cas, les bordures de ces régions sont continuellement alignées de l'extérieur par inclusion de leurs pixels dans les vraies régions environnantes. Après plusieurs pas d'exécution, ces régions seront complètement effacées.

A la fin du processus, toutes les régions de l'image sont bien lissées et délimitées par les contours détectés. Un simple accroissement de régions, contrôlé par les contours détectés, permet d'extraire les régions de l'image.

3.5 Coordination d'agents par un champ de potentiel

3.5.1 Champ de potentiel dans les systèmes multi-agents

Le champ de potentiel est utilisé particulièrement pour l'évitement d'obstacles en robotique mobile et en général pour la navigation et la coordination de mouvement dans les systèmes multi-agents situés. Les principaux avantages d'utilisation du champ de potentiel sont : la rapidité de réaction, et la coordination spatiale sans échange de messages, bien approprié aux environnements inconnus, où une planification de chemin est non envisageable. Cependant, ces méthodes souffrent d'un problème des minima locaux de potentiel. Ce problème consiste à ce qu'un ou plusieurs agents se trouvent piégés dans des puits de potentiel, ou bien ils se trouvent en bouclage infini sur des cycles fermés.

Formulation générale

Le champ de potentiel - radial - V dit de type $1/d$, généré par un objet de masse (ou charge) M , en un point de l'espace, distant de d est exprimé comme suit :

$$V = \frac{k \times M}{d} \quad (3.7)$$

Où k est une constante, dépendant de la nature du champ ; exemple $k = 6,67300 \times 10^{-11} N.m^2.kg^{-2}$ pour le champ gravitationnel (constante de Newton) ; $k = 8,98 \times 10^9 N.m^2.C^{-2}$; pour le champ électrique (constante de Coulomb).

Un objet sensible au champ de potentiel (une masse dans le champ gravitationnel, ou une charge électrique dans le champ électrique), ayant une masse m , subit une force

exprimée comme suit :

$$\vec{F} = -\frac{k \times m \times M}{d^3} \times \vec{d} \quad (3.8)$$

En général, la force engendrée par un champ de potentiel est attractive ($\vec{F} \times \vec{d} < 0$). Le cas typique est le champ gravitationnel. La force électrique fait exception à cette règle, en effet, la force est répulsive si les deux charges (correspondant à m et M) sont de même signe ; et attractive si elles sont de signes contraires.

Représentation des buts et des obstacles

Dans un environnement, où les agents sont appelés à atteindre des positions données, dites buts, en évitant certains objets, dits obstacles, les méthodes basées sur le champ de potentiel offrent des solutions adéquates. Pour ce faire, un champ de potentiel attractif est créé autour des buts et un champ de potentiel répulsif est créé autour des obstacles (Fig. 3.6). Le champ de potentiel en tout point de l'environnement est la combinaison du champ attractif et du champ répulsif. La direction de la force résultante, calculé en un point, est utilisée pour inférer une nouvelle direction de d'un agent. Cette direction lui permet d'emprunter un chemin optimal vers son but, en évitant les obstacles.

La figure 3.6 montre la présence d'un obstacle sur le chemin d'un agent qui se dirige vers un but. Les lignes du champ résultant de la combinaison du champ attractif du but et du champ répulsif de l'obstacle, définissent le chemin qui sera emprunté par l'agent.

Dans la littérature, plusieurs auteurs ont utilisé le champ de potentiel pour la coordination spatiale des agents. [Koren et Borenstein \(1991\)](#) ont fait recours à un champ de potentiel du type $1/d$, pour étudier la coordination de robots par un champ de potentiel dans un environnement inconnu. Le champ est créé par toutes les cellules $\{c_{ij}\}$ appartenant aux différents obstacles, situés sur une grille rectangulaire. La force répulsive

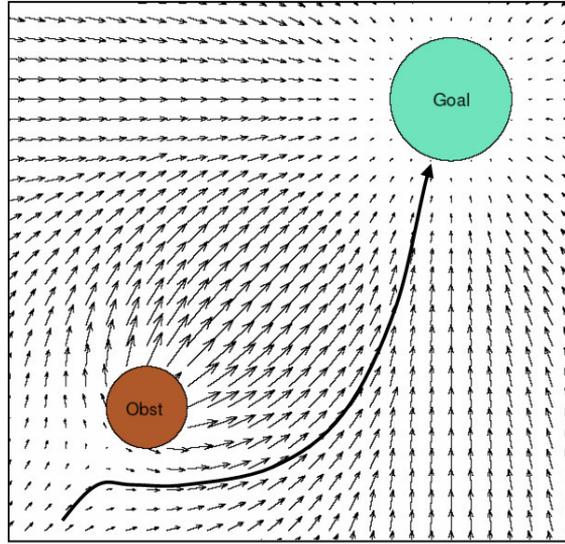


FIG. 3.6 – Buts et Obstacles dans un champ de potentiel

exercée par une cellule c_{ij} est donnée par l'expression suivante :

$$\vec{F}_{i,j} = \frac{F_{cr} W C_{ij}}{d^2(i, j)} \times \frac{\vec{x}_{ij} - \vec{x}_0 + \vec{y}_{ij} - \vec{y}_0}{d(i, j)} \quad (3.9)$$

avec (x_0, y_0) et (x_{ij}, y_{ij}) les coordonnées respectivement de la position courante du robot et de la position de la cellule c_{ij} . F_{cr} et W sont respectivement une constante de la force répulsive, et la masse du robot. C_{ij} représente la probabilité d'existence d'un obstacle à la cellule c_{ij} , et $d(i, j)$ la distance du robot à la cellule c_{ij} .

Un ensemble de cellules $\{c_{ij}\}$ exerce sur un agent situé à la position t une force répulsive F_r , égale à la somme de toutes les forces répulsives correspondant à toutes les cellules d'obstacles :

$$\vec{F}_r = \sum \vec{F}_{cr} \quad (3.10)$$

Cependant, une force attractive exercée par un but situé à la position (x_t, y_t) , est

donnée par :

$$\vec{F}_t = F_{ct} \times \frac{\overrightarrow{x_t - x_0} + \overrightarrow{y_t - y_0}}{d_t} \quad (3.11)$$

où F_{ct} est une constante de la force attractive, d_t est la distance du robot au but et (x_t, y_t) les coordonnées du but. La force résultante, $\vec{R} = \vec{F}_r + \vec{F}_t$ est utilisée pour réorienter le robot. La déviation subit au robot est donnée par :

$$\Omega = k(\delta - \theta) \quad (3.12)$$

où δ et θ sont respectivement la direction courante du robot, et la direction de la force, et k est une constante inversement proportionnelle à l'inertie du robot.

Le problème de minima locaux, était discuté par [Zeghal et Ferber \(1993\)](#), qui rapportaient que ce problème est engendré à cause du comportement de fuite des agents face aux obstacles. Ils proposent un comportement d'évitement ou de contournement au lieu de celui de fuite. Pour cela ils introduisent la force de glissement dont la direction est tangentielle à l'équipotentielle du champ. La force résultante \vec{A}_i , obtenue par la combinaison de la force attractive du but i , A_{goal_i} et des forces répulsives $A_{i,j}$ de N obstacles $\{j\}$, est calculée d'une manière conventionnelle [Simonin \(2001\)](#) :

$$\vec{A}_i = \overrightarrow{A_{goal_i}} + \sum_{j \neq i}^N \beta_{ij} \vec{A}_{ij} \quad (3.13)$$

Cependant, la force, dite de glissement, utilisée pour calculer la direction de l'agent est obtenue par la projection de la force résultante (attraction+répulsion), sur la droite tangentielle à l'équipotentielle (voir Fig. 3.7).

Afin d'améliorer les méthodes basées sur le champ de potentiel, plusieurs heuristiques ont été introduites par d'autres chercheurs tels que M. Khatib ([Khatib \(1996\)](#)), qui propose les améliorations suivantes :

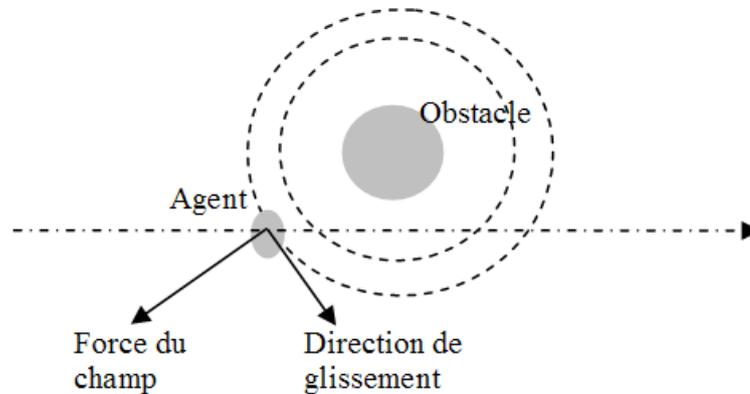


FIG. 3.7 – Force tangentielle de glissement

- La force est calculée en fonction de la rotation du robot ;
- Seulement les obstacles qui sont en face du robot (appartenant au secteur visible par le robot), sont pris en compte.

Balch et Arkin (1994) ont proposé un paradigme dit "schema-based control" pour la coordination dans les systèmes multi-agents réactifs. Dans ce paradigme, chaque comportement réactif "motor schema" calcule sa réaction à son stimulus en utilisant la méthode du champ de potentiel. Cependant, et contrairement au formalisme général, le champ de potentiel est calculé en considérant uniquement la perception locale et la position courante du robot. Ils introduisent le concept de la perception orientée action "action-oriented perception". Ceci permet de ne prendre en compte, lors du calcul du champ de potentiel, que les informations relatives à la tâche à exécuter.

Ensuite, les auteurs ont testé l'impact de l'introduction de trois modes de communication sur la performance du système. Ils ont pu différencier les tâches où la communication améliore la performance, et les tâches où la communication paraît non nécessaire. Il a été reporté que si elle s'avère nécessaire, la communication de bas niveau paraît plus efficace que celle de haut niveau.

3.5.2 Approche de coordination proposée

Afin de doter nos agents d'un mécanisme d'auto-organisation spatiale, un champ de potentiel artificiel inspiré du champ électrostatique est créé et mis à jour autour des pixels alignés. Il permet aux agents de se regrouper autour des régions d'intérêt dans l'image et de concentrer leurs actions sur les contours de ces régions.

Création du champ de potentiel

Contrairement à tous les travaux vus dans la littérature, où le champ de potentiel est créé à des positions prédéfinies correspondant aux objets de l'environnement (buts et obstacles) (Ferber (1995); Tsuji *et al.* (2002); Simonin (2005); Flacher (2005)), dans notre cas, le champ de potentiel résulte de l'interaction des agents avec les objets de l'environnement (pixels de l'images). L'intensité $\Psi(x, y)$ du champ à la position (x, y) , créée par un ensemble de P pixels préalablement alignés $\{(x_i, y_i), i = 1..P \wedge I(x_i, y_i)=\text{aligné}\}$ est donnée par :

$$\Psi(x, y) = \sum_{i=1}^P \frac{k}{\sqrt{(x - x_i)^2 + (y - y_i)^2}}, k \in R^+ \quad (3.14)$$

où k est la constante de la force électrostatique, prise égale à 1.

Un agent, capable d'altérer des pixels ($C_t=\text{true}$), et situé à la position (x, y) , subit une force d'attraction \vec{F} (Fig. 3.8). Cette force s'exprime à l'aide du vecteur gradient du champ de potentiel :

$$\vec{F} = \begin{cases} -\vec{\nabla} \Psi(x, y) & \text{si } C_t=\text{true} \\ \vec{0} & \text{sinon} \end{cases} \quad (3.15)$$

Ainsi, les déplacements de l'agent qui sont de nature stochastique sont pondérés par la force attractive exercée par le champ de potentiel. Les agents sont ainsi poussés à se diriger vers les pixels d'intérêt, tout en maintenant un déplacement stochastique qui permet d'explorer toutes les régions de l'image.

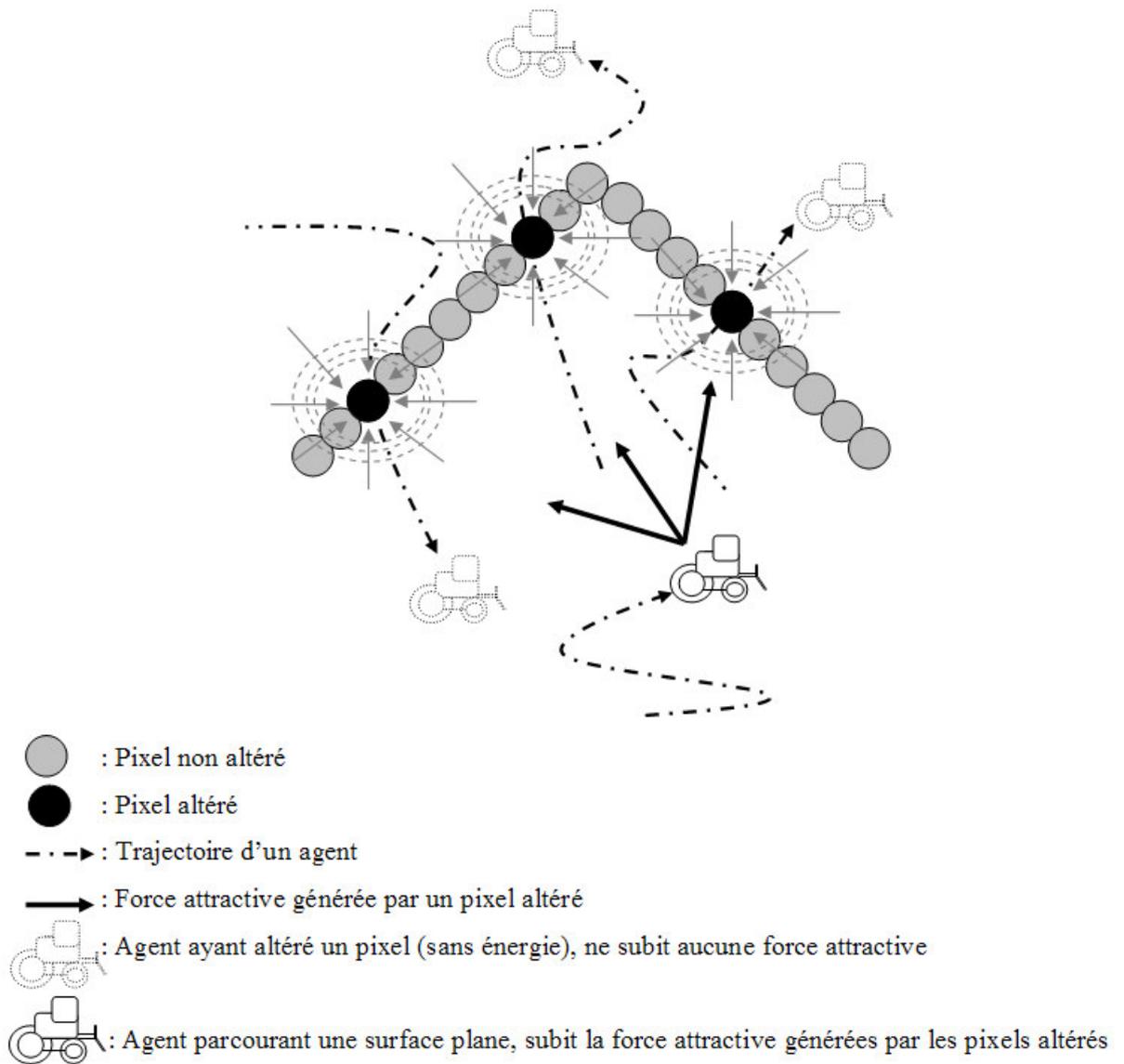


FIG. 3.8 – Champ de potentiel créé par l'altération des pixels par les agents.

Amortissement du champ de potentiel

Un mécanisme de relaxation du champ de potentiel est introduit. Il permet aux agents groupés autour des pixels d'intérêt de se libérer et d'explorer d'autres régions dans l'image. Autour d'un pixel donné, l'intensité du champ décroît après chaque altération de ce pixel. L'équation de la dynamique de relaxation est exprimée comme

suit :

$$\Psi_{t+1}(x, y) = \mu \times \Psi_t(x, y), \mu < 1 \quad (3.16)$$

$\Psi_0(x, y)$ correspond au champ créé après la première altération du pixel. La constante μ initialisée à 0.9, représente le taux de décroissance de l'intensité du champ. Après plusieurs altérations d'un pixel donné, le champ généré autour de ce pixel tend à s'annuler. A la limite de la dynamique, le champ de potentiel est complètement relaxé dans toute l'image. Ceci représente l'état final du processus pour lequel l'intérieur des régions est totalement lissé et les contours sont précisément détectés. Par ailleurs, l'amortissement du champ permet de résoudre le problème des minima locaux, inhérent à la coordination par le champ de potentiel. En effet, la variation du potentiel en un point donné de l'environnement, dû à l'altération des pixels de l'image par les agents, déplace les minima locaux dans l'image, et permet ainsi au agents de se libérer de ces minima.

3.5.3 Exemple de l'évolution de la dynamique des agents

La figure 3.9 montre un exemple de l'évolution de la dynamique des agents sur une image réelle prise de la base ABW (Mazouzi *et al.* (2006)). L'affichage de l'image (Fig. 3.9a) est réalisé par un simple algorithme de rendu réaliste, basé sur la technique du lancé de rayon, en utilisant l'orientation du vecteur normal à la surface. Une population de 1000 agents était lancée dans l'image. La figure 3.9b montre le résultat de détection de contours obtenu en utilisant un détecteur spatial, basée sur la variation de l'orientation du vecteur normal à la surface. Dû au niveau élevé de bruit, présent dans les images réelles, les résultats de détection, ainsi obtenus sont insatisfaisants et inexploitable. Les figures 3.9c, et 3.9d montrent respectivement les positions des agents, et les pixels alignés après 20 itérations. Les figures 3.9e et 3.9f permettent de constater l'influence du champ de potentiel sur les agents après 200 itérations. A ce stade, les agents attirés par le champ de potentiel ont acquis une capacité d'altération ($C_t = \text{true}$).

Après 800 itérations, des amas d'agents sont formés (Fig. 3.9g, 3.9h) autour des régions d'intérêt (contours et régions de bruit). La figure 3.9i montre l'état du système

après relaxation du champ de potentiel dans toute l'image. Les contours finaux détectés après 8000 itérations sont montrés dans la figure 5j. A ce stade les agents reprennent un mouvement aléatoire libre de toute influence du champ de potentiel qui s'est totalement relaxé. La figure 3.9k présente l'état des surfaces et des arrêtes à la fin du processus. Nous pouvons constater que les surfaces ont été lissées et que les arrêtes ont été bien améliorées et amincies.

3.6 Implémentation

Plusieurs plates-formes ont été proposées pour le développement des systèmes multi-agents (Minar *et al.* (1996); Guessoum (2000); Gutknecht *et al.* (2001); Gasser et Kakugawa (2002)), dont la plupart sont implémentées en JAVA. Le langage JAVA permet la portabilité des solutions et la facilité d'intégration. Cependant, il n'est pas recommandé en cas d'applications nécessitant un minimum de temps d'exécution, telles que l'interprétation d'images en vision robotique. Certaines plates-formes sont spécialement orientées simulation, telle que SWARM (Minar *et al.* (1996)). Swarm existe en objective-C, cependant cette version est de moins en moins supportée, et l'expérience s'accumule de plus en plus avec la version en JAVA. Nos expérimentations avec quelques une de ces plates-formes ont révélé des temps de calcul excessifs. Par ailleurs, le C++ reste le langage qui regroupe la plupart des développeurs en traitement d'images et reconnaissance d'objets. En plus de l'expérience accumulée avec ce langage, le C++ génère un code dont le temps d'exécution est plus rapide. De ce fait, nous avons opté pour le développement d'un framework dédié à notre approche multi-agents pour la segmentation d'images, en utilisant le langage de programmation C++. Le framework était utilisé pour la segmentation d'images de profondeur. Néanmoins, sa décomposition fonctionnelle permet sa réutilisation pour la segmentation de d'autres types d'images. Par ailleurs, ce framework est facile à utiliser par des développeurs qui n'ont pas nécessairement une compétence particulière en multi-agents. Le développeur est appelé à instancier uniquement les éléments qui sont spécifiques aux images traitées. Le framework développé est composé de plusieurs classes, dont les principales sont : la classe Pixel, la classe Image, La classe Agent, et la classe Scheduler (figure 3.10).

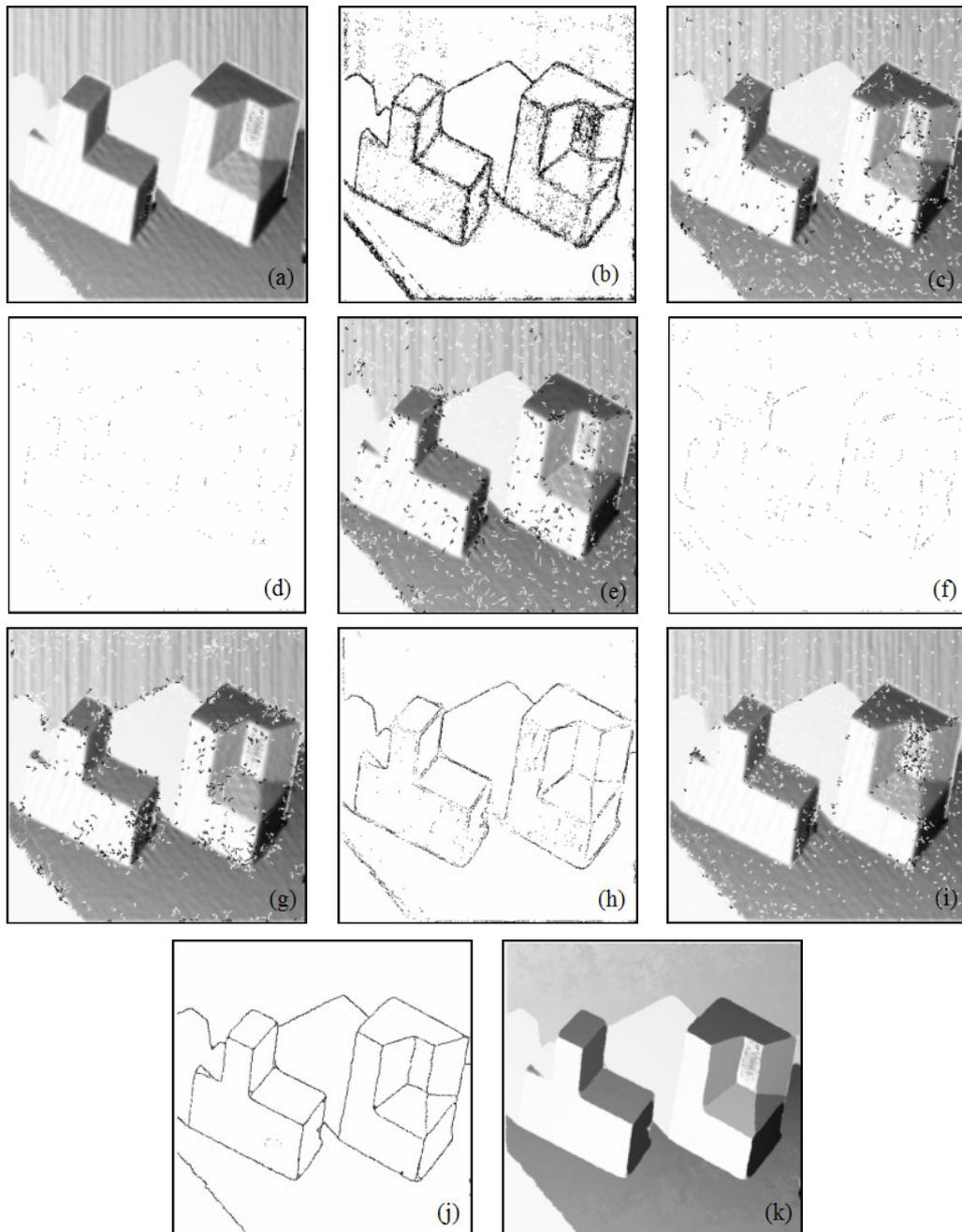


FIG. 3.9 – Evolution de la dynamique des agents dans le champ de potentiel, et résultats de segmentation

3.6.1 Pixel

Un pixel est caractérisé par une profondeur (z), les paramètres de l'équation du plan correspondant (a, b, c, d), un état (*state*) (lissé, aligné, inchangé), et l'intensité du

champ de potentiel psi à la position du pixel. Les principales opérations qui peuvent être exécutées sur un pixel sont :

- *computePlaneEquation()*, permettant le calcul des paramètres de l'équation du plan tangent ;
- *isPixelOfInterest()*, pour tester si le pixel est un pixel d'intérêt ;
- *isInPlane()*, pour tester si le pixel appartient à un plan ;
- *smoothPixel()*, pour le lissage du pixel ;
- *alignPixel()*, pour l'alignement du pixel.

3.6.2 Image

La classe Image regroupe les attributs globaux des images traitées, et offre l'interface nécessaire pour la manipulation des images. Une image est caractérisée par sa taille ($w \times h$), les différents seuils utilisés dans les traitements de l'image ($trTeta, trD, xiMin, xiMax$), ainsi qu'une matrice de références aux différents objets pixels (*Pixels*) composant l'image. Les principales opérations effectuées sur une image sont :

- *loadRawImage()*, permettant le chargement d'une image à partir d'un fichier ;
- *convertImage()*, permettant de convertir une image de profondeur (Z) en une image de plan (Z^*) ;

3.6.3 Agent

Chaque agent est caractérisé par sa position courante sur l'image (x, y) et la capacité d'altération de pixels (c). L'activité d'un agent est initialisée par la méthode *create()*. La méthode *step()* permet à un agent de se déplacer d'un pas sur l'image et d'exécuter l'action appropriée sur le pixel courant. Elle permet aussi de créer et de mettre à jour l'intensité du champ de potentiel aux pixels voisins du pixel en cours.

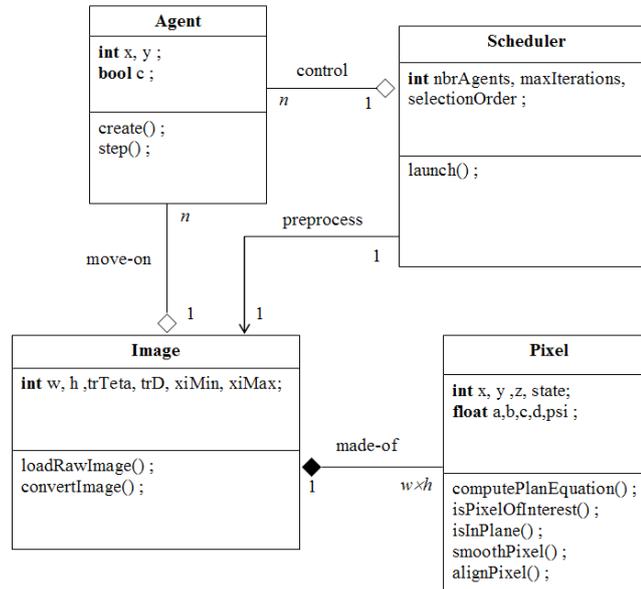


FIG. 3.10 – Diagramme de classes simplifié

3.6.4 Scheduler

Le multi-threading était aussitôt abandonnée dès les premières expérimentations. En effet, nous avons constaté qu'il n'était pas possible de lancer un nombre de threads à hauteur de 2500, nécessaires pour le bon fonctionnement de notre système. Nous avons donc opté pour l'activation des agents par un scheduler d'exécution. Il s'agit du pilote du système qui permet d'initialiser les différents objets, et d'activer les agents. Le scheduler, après l'initialisation de tous les agents, sélectionne à chaque itération un agent et lui passe le contrôle pour effectuer un seul pas. Aussitôt le pas effectué, le contrôle est retourné au scheduler pour sélectionner et activer un autre agent. L'ordre de sélection des agents peut être séquentiel ou aléatoires. Nos expérimentations ont montré que les résultats étaient indépendants de l'ordre choisi. Les principaux attributs de la classe Scheduler sont :

- le nombre d'agents (*nbrAgents*) ;
- le nombre maximal d'itérations (*maxIterations*) ;
- l'ordre de sélection (*selectionOrder*) ;
- une référence à l'objet image (*image*).

La principale méthode du scheduler est *launch()* qui permet d'initialiser et de lancer le scheduler.

3.6.5 Analyse du coût de calcul

En considérant la dynamique des agents, nous pouvons constater que le temps de calcul, représenté par le nombre de pas effectués par chaque agent, dépend uniquement de la densité de distribution des agents sur l'image. En effet, les agents nécessitent un minimum de temps de traitement pour pouvoir complètement amincir les contours des vraies régions, et lisser les régions de bruit. En dehors de l'influence du champ de potentiel, nous pouvons supposer que les agents restent uniformément distribués dans l'image. Dans ce cas, le temps de traitement est indépendant du nombre et de la distribution des contours et des régions de bruit.

Pour illustrer l'estimation du coût de calcul, nous considérons une image de synthèse de taille $W \times H = 256 \times 256$, contenant une unique région de bruit A de forme circulaire et d'un rayon initial (à $t=0$), $R_0 = 25$ pixels (figure 3.11a). Cette région est entourée d'une unique région plane B qui ne contient aucun pixel d'intérêt. Soit le nombre d'agents $N = 800$ et soit $\xi_{min} = \xi_{max} = \infty$. Ces deux dernières valeurs de ξ_{min} et ξ_{max} permettent d'aligner un pixel à une région plane après une seule altération par un agent (alignement total). Sous ces conditions d'exécution, les agents alignent les pixels de la bordure de la région A lorsqu'ils se déplacent à l'intérieur de cette région. Comme il s'agit d'une région de bruit, les agents qui transitent à l'intérieur ne peuvent pas s'adapter et par conséquent n'alignent pas les pixels de la bordure lorsqu'ils quittent cette région. La région A se contracte donc continuellement jusqu'à ce qu'elle ait complètement disparu (figure 3.11b, 3.11c, 3.11d). Le nombre de pas effectués par les agents avant l'effacement total de la région A représente le temps total de calcul, noté Γ . Au temps t , le nombre d'agents situés à l'intérieur de la région A est : $\pi\rho R_t^2$; où R_t est le rayon de la région au temps t et ρ est la densité de distribution des agents. Cette distribution dépend de la taille de l'image et du nombre d'agents utilisés : $\rho = \frac{N}{WH}$. Ainsi, le nombre d'agents situés sur la bordure de la région au temps t , peut être approximativement estimé par $2\pi\rho R_t$, exprimant le nombre d'agents sur la circonférence d'un cercle de rayon R_t , avec une densité d'agents ρ . Evidemment, seulement un taux τ

de ces agents alignent des pixels sur la bordure de la région A. Il s'agit des agents avec une capacité $C_t = \text{true}$ et passant de la région B vers la région A. Nous déduisons donc, le nombre de pixels alignés au temps t : $2\pi\tau R_t\rho$, et par conséquent, la nouvelle taille de la région A deviendra : $\pi R_{t+1}^2 = \pi R_t^2 - 2\pi\tau\rho R_t$. Le taux de décroissance du rayon R_t de la région de bruit A peut être exprimé comme suit : $R_{t+1}^2 = R_t^2 - \eta R_t$, où $\eta = \frac{2\tau N}{WH}$. Le temps minimum Γ nécessaire pour complètement lisser la région de bruit est obtenu lorsque $R_\Gamma \leq 0$.

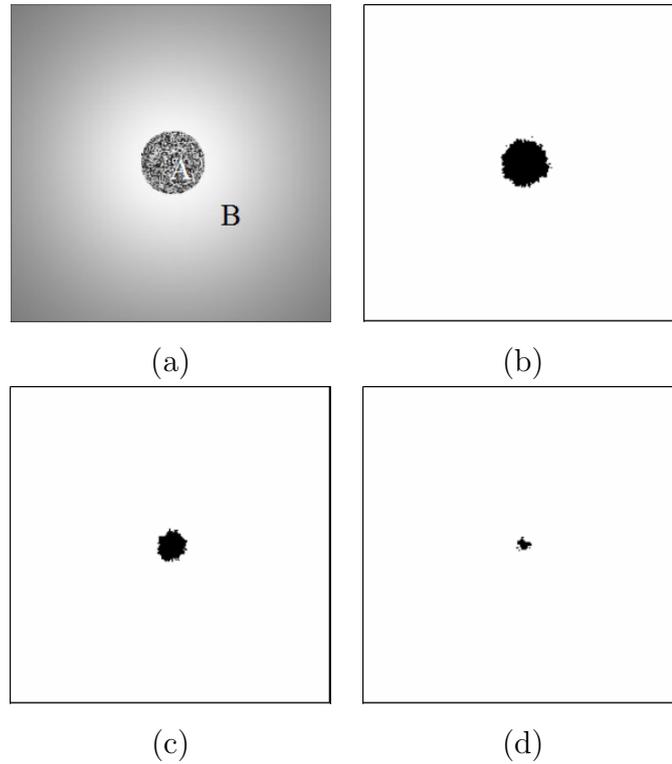


FIG. 3.11 – Variation de la taille de la région de bruit (A) : (a) Image de profondeur rendue réaliste ; (b) à $t=5000$; (c) à $t=10000$; (d) à $t=15000$

Dans le but de comparer le temps de calcul théorique au temps de calcul observée, nous avons enregistré la taille de la région A après chaque pas d'exécution. La figure 3.12 montre les courbes respectives de la variation estimée et observée de la taille de la région de bruit, avec le paramètre $\tau = 0.24$. La corrélation des résultats nous permet de valider le modèle théorique d'estimation du coût de calcul.

Le temps moyen de segmentation par image sur l'ensemble des 30 images de test (voir chapitre 7) était de 14 secondes sur un mono-processeur PC Compaq 8220. Cette moyenne de temps de segmentation est meilleure que celles de la majorité des méthodes

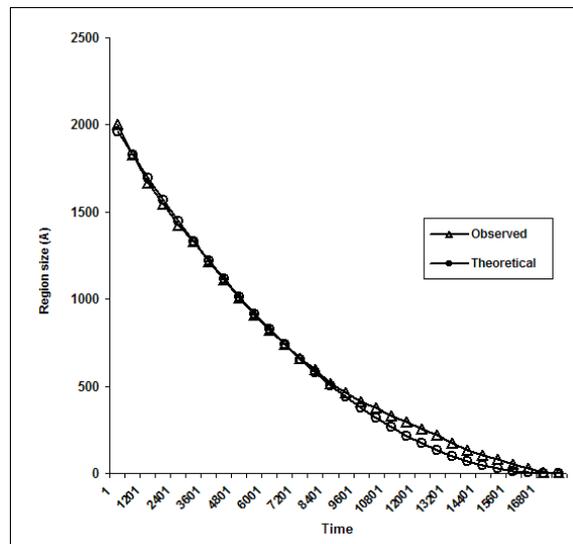


FIG. 3.12 – Variations théorique et observée de la taille de la région de bruit (A)

publiées (Hoover *et al.* (1996)). Par ailleurs, grâce au couplage faible des agents et au fait qu'ils ne communiquent pas directement, il est possible de paralléliser les traitements en les distribuant sur des unités de calcul autonomes partageant une mémoire commune qui contiendra l'image à segmenter. Ceci permettra une segmentation d'images rapide, souhaitable pour les applications temps réel.

3.7 Conclusion

Dans ce chapitre, nous avons proposé une approche multi-agents pour la segmentation d'images de profondeur. Les interactions indirectes entre les agents autonomes qui se déplacent dans l'image permettent à la fois, une élimination efficace du bruit, et une extraction fiable des régions. La compétition entre les agents qui s'auto-organisent autour des bordures des régions ont permis de faire émerger les lignes de contours dans l'image. Ces contours, dont aucune détection explicite n'est codée dans les agents, émergent des interactions et des actions collectives au sein de la population d'agents. Les résultats obtenus sont meilleurs que ceux obtenus par des algorithmes conventionnels de segmentation, tel que l'accroissement de région, ou la détection explicite de contours. Les agents utilisés sont faiblement couplés et communiquent indirectement via leur environnement (l'image). Ceci permet des implémentations parallèles souhaitables pour

une haute efficacité de calcul, nécessaire pour l'interprétation en temps-réel d'images, notamment en vision robotique. Les résultats d'expérimentation (voir chapitre 7) obtenus avec les images réelles de la base ABW ont été comparés à ceux obtenus par quatre algorithmes typiques de segmentation d'images de profondeur. Les résultats de comparaison révèlent une excellente performance de la méthode proposée et ce, à la fois, en terme d'efficacité de détection et d'exactitude des résultats.

Chapitre 4

Régularisation bayésienne et régularisation floue des contours 3D

4.1 Introduction

Nous avons présenté dans le chapitre précédent une nouvelle approche de segmentation d'images, et nous avons détaillé son application à la segmentation d'images de profondeur. Malheureusement, l'alignement compétitif des frontières entre les régions, opéré par des agents autonomes, conduit à la production de contours déformés et mal localisés. Afin de remédier à ce problème nous décrivons dans ce chapitre deux méthodes de correction, basées sur une approche de régularisation. Dans les deux méthodes, nous supposons que la segmentation de l'image, effectuée par le système multi-agents est correcte dans le sens où les régions sont correctement extraites ; ainsi les pixels pourtours de ces régions qui ont été affectés arbitrairement à ces régions seront revus dans le but de les réaffecter, en se basant sur une approche formelle.

La première méthode de correction de contours est basée sur une régularisation bayésienne, en faisant recours à un apprentissage bayésien des paramètres du modèle statistique des régions de l'image ([Mazouzi et Batouche \(2007a,b\)](#)). A cette fin, nous utilisons un nouveau champ aléatoire de markov (Markov Random Field : MRF). Le MRF introduit est utilisé pour modéliser le smoothness des pourtours de régions, considéré

comme à priori pour la régularisation des contours. La seconde méthode est basée sur une régularisation floue des pourtours de régions. Elle procède à la correction de la frontière entre deux régions adjacentes, en considérant ces deux dernières comme deux ensembles flous ([Mazouzi et Batouche \(2007c\)](#)).

4.2 Régularisation des contours par inférence bayésienne

4.2.1 Etat de l'art

Peu de travaux ont intégré l'inférence bayésienne pour la segmentation d'images de profondeur. [LaValle et Hutchinson \(1993\)](#) ont utilisé un test bayésien pour la fusion de régions. En se basant sur certains vecteurs d'observation, et certaines connaissances à priori sur les régions, la fusion de deux régions est conditionnée par la probabilité que la région engendrée est homogène. [Jain et Nadabar \(1990\)](#) ont proposé une méthode de segmentation d'images de profondeur, basée contour. Ils se basent sur le modèle du Line Process (LP) ([Geman et Geman \(1984\)](#)) afin d'étiqueter des points de contours, en considérant le smoothness des régions comme information à priori. [Wang et Wang \(2004\)](#) ont présenté un schéma hybride pour la segmentation d'images de profondeur. En premier, ils ont proposé une estimation bayésienne jointe des étiquettes des pixels et de l'ensemble des patches de surface. En suite, la solution est améliorée en combinant l'algorithme du ScanLile de détection de contours ([Jiang et Bunke \(1999\)](#)), et le modèle Markovien de la logistique multi-niveau (MLL) ([Li \(2001\)](#)). Leur objectif était de réduire la complexité du modèle, en estimant le nombre et les paramètres des régions dans une image. Li propose dans [Li \(2000\)](#) un modèle du champ aléatoire de Markov pour le lissage d'une image de profondeur avec préservation des contours. L'application du framework MAP-MRF (Maximum à posteriori - Markov random field) en considérant l'à priori de smoothness par partie de la surface, a permis un bon lissage des régions avec préservation des contours à la fois de type toit et de type marche.

Malgré l'apport des travaux cités précédemment, certains aspects inhérents à la

segmentation d'images de profondeurs ont été omis. En effet, la plupart des travaux utilisent des modèles markoviens qui se basent exclusivement sur le smoothness de la surface. Par ailleurs, les méthodes proposées procèdent par une classification des pixels dans des clusters sans assurer la continuité de chaque cluster. Typiquement, dans la méthode proposée par Wang et Wang (2004), les pixels appartenant à des régions coplanaires peuvent être classés indifféremment dans n'importe laquelle de ces régions. La contrainte de la continuité spatiale des régions ne semble pas être prise en compte.

4.2.2 Aperçu de la méthode

La régularisation ne concerne que les pixels situés sur les frontières entre les régions. Ces pixels sont étiquetés en se basant sur une estimation bayésienne en utilisant deux à priori distincts. Le premier consiste au smoothness par partie de la surface, et est représenté par le modèle MLL (Li (2001)). Le second que nous introduisons dans ce travail est le smoothness par partie des contours. Le modèle Markovien utilisé pour représenter cet à priori est basé sur un voisinage d'ordre élevé, et suppose que les pixels de contours appartiennent à des courbes lisses, représentant les bords des régions. L'utilisation de l'algorithme de l'ICM (Iterated Conditional Modes) (Besag (1986)) pour la recherche de la solution optimale, nous a permis de formuler la continuité des régions, et ce par la mise en place d'une contrainte sur les étiquettes possibles qu'un pixel donné peut avoir. En effet, un pixel ne peut avoir une étiquette que parmi celles correspondant aux régions dont il est proche.

4.2.3 Modélisation de la segmentation en MAP-MRF

Le smoothness par partie des régions de l'image ainsi que le smoothness par partie des lignes de contours sont les deux propriétés que nous avons utilisées pour modéliser les distributions des étiquettes dans les images de profondeur. Soit S le trillel de l'image. A chaque pixel $(x, y) \in S$, $D(x, y)$ représente la profondeur en ce pixel et $D^*(x, y)$ représente les paramètres de son équation de plan; $z = ax + by + c$: $D^*(x, y) = (a(x, y), b(x, y), c(x, y))$. L'ensemble des étiquettes qu'un pixel peut avoir est $L = \{l_1, ..l_M\}$,

où M est le nombre de régions dans l'image. En supposant la markovianité de distribution de l'ensemble des variables représentant les étiquettes des pixels $F = \{f(x, y), (x, y) \in S\}$, le problème de segmentation de S selon le framework bayésien revient à maximiser la probabilité à posteriori de $F : P(F/D)$ sachant les données de profondeur bruitées D . Selon la règle de bayes, la probabilité à posteriori est définie comme suit :

$$P(F/D) = \frac{p(D/F)P(F)}{p(D)} \quad (4.1)$$

avec $Z = \sum_F e^{-U(F)}$, une constante de normalisation dite fonction de partition. $P(F) = \frac{1}{Z}e^{-U(F)}$ représente la probabilité à priori de F . L'énergie à priori $U(F)$ est obtenue par la somme des potentiels $V_c(F)$ de toutes les cliques c de l'ensemble des cliques possibles $C : U(F) = \sum_{c \in C} V_c(F)$.

Dans notre travail, nous avons considéré deux ensembles de cliques : l'ensemble C_1 des cliques formées chacune de deux sites voisins, et l'ensemble C_2 des cliques où chacune est formée des 9 sites situés dans une fenêtre de taille 3×3 . Le potentiel V^1 pour les cliques de C_1 est défini à l'aide du paramètre ζ , ($\zeta < 0$) comme suit :

$$V^1(f(x, y), f(x', y')) = \begin{cases} \zeta, & \text{si } f(x, y) = f(x', y') \\ -\zeta, & \text{sinon} \end{cases} \quad (4.2)$$

Pour pouvoir définir le potentiel dans les blocks de 3×3 pixels, nous considérons les notations suivantes :

Soit $c_{3 \times 3}$ une clique formée de 9 sites et centrée en (x, y) :

$$c_{3 \times 3}(x, y) = \{f(x + i, y + j), (i, j) \in \{-1, 0, 1\}\} \quad (4.3)$$

Définissons la transformée T , tel que :

$$T(c_{3 \times 3}(x, y)) = (f(x - 1, y - 1), \dots, f(x + 1, y + 1)) \quad (4.4)$$

Le potentiel des cliques de l'ensemble C_2 peut donc être exprimé à l'aide du paramètre κ , ($\kappa < 0$) comme suit (Fig. 4.1) :

$$V^2(T(c_{3 \times 3}(x, y))) = \begin{cases} \kappa, & \text{si } \exists (x', y') \text{ et } (x'', y'') \text{ tel que } f(x, y) = f(x', y') = f(x'', y'') \\ & \text{et } (x', y'), \widehat{(x, y)}, (x'', y'') = \pi \\ 0, & \text{si } \exists (x', y') \text{ et } (x'', y'') \text{ tel que } f(x, y) = f(x', y') = f(x'', y'') \\ & \text{et } (x', y'), \widehat{(x, y)}, (x'', y'') = 2\pi/3 \\ -\kappa, & \text{sinon} \end{cases} \quad (4.5)$$

Le potentiel V^1 modélise le smoothness des régions, cependant le potentiel V^2 modélise le smoothness des lignes de contours (Fig. 4.1). Le choix de V^2 varie selon le type des surfaces modélisées. Pour les images contenant des objets polyédriques, considérées dans ce travail, la frontière entre deux surfaces adjacentes est formée des pixels appartenant à un même segment de droite.

La fonction de distribution de D sachant F est obtenue en considérant que les observations D sont dégradées par un bruit suivant une distribution gaussienne identique et indépendant (i.i.d) :

$$d(x, y) = a_{f(x,y)}x + b_{f(x,y)}y + c_{f(x,y)} + e(x, y) \quad (4.6)$$

avec $e(x, y) \sim N(0, \sigma_{f(x,y)}^2)$

La fonction de distribution de D sachant F est donnée par l'équation suivante :

$$p(D/F) = \frac{1}{\prod_{(x,y) \in S} \sqrt{2\pi\sigma_{f(x,y)}^2}} e^{-U(D/F)} \quad (4.7)$$

avec l'énergie de probabilité $U(D/F)$ exprimée comme suit :

$$U(D/F) = \sum_{(x,y) \in S} (a_{f(x,y)}x + b_{f(x,y)}y + c_{f(x,y)} - d(x, y))^2 / (2\sigma_f^2(x, y)) \quad (4.8)$$

Notons que $p(D)$ est constante pour D fixé, et sert comme facteur de normalisation. La solution F^* est obtenu donc en maximisant la probabilité à posteriori $P(F/D) \times p(D/F)P(F)$. Ceci est équivalent à la minimisation de l'énergie à posteriori : $U(F/D) = U(D/F) + U(F)$:

$$F^* = \operatorname{argmin}\{U(D/F) + U(F)\} \quad (4.9)$$

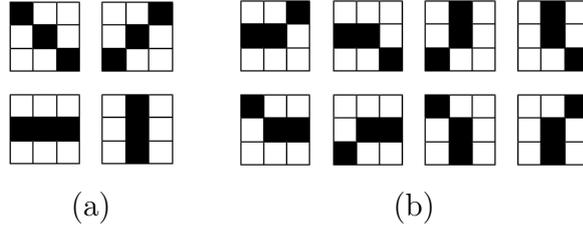


FIG. 4.1 – Potentiel de clique $V^2(c_{3 \times 3})$. (a) segment totalement lissé : $V^2(c_{3 \times 3}) = -\kappa$; (b) segment partiellement lissé : $V^2(c_{3 \times 3}) = 0$; autrement, le segment est non lissé : $V^2(c_{3 \times 3}) = \kappa$

4.2.4 Minimisation de l'énergie

En supposant la markovianité de F , et en considérant que les observations $D(x, y)$ sont conditionnellement indépendantes, nous avons choisi l'algorithme du ICM (Iterated Conditional Modes) pour minimiser l'énergie à posteriori $U(F/D)$. En exprimant $U(F/D)$ comme la somme des énergies de toutes les sites :

$U(F/D) = \sum_{(x,y) \in S} U(f(x, y)/d(x, y))$, nous pouvons la séparer en deux termes :

$$U(F/D) = \sum_{(x,y) \in S'} U(f(x, y)/d(x, y)) + \sum_{(x,y) \in S-S'} U(f(x, y)/d(x, y)) \quad (4.10)$$

où S' est l'ensemble des sites appartenant aux bordures des régions, obtenues durant la phase de segmentation par le système multi-agents :

$$S' = \{(x, y) \in S \mid \exists (x', y'), (x' - x, y' - y) \in \{-1, 0, 1\}^2 \wedge f_{(x,y)} \neq f_{(x',y')}\}$$

En supposant le correcteness de l'étiquetage de $S - S'$ (effectué en première étape), le terme $\sum_{(x,y) \in S-S'} U(f(x, y)/d(x, y))$ est donc constant.

La minimisation de $U(F/D)$ revient donc à minimiser l'énergie $U'(F/D)$ correspondant aux sites dans S' :

$$U'(F/D) = \sum_{(x,y) \in S'} U(f(x, y)/d(x, y)) \quad (4.11)$$

La supposition du correcteness de l'étiquetage de $S - S'$ nous permet aussi l'introduction d'une contrainte sur l'ensemble des valeurs qu'un site de S' peut prendre durant de déroulement de l'algorithme de l'ICM. En effet, l'étiquette $f(x, y)$ d'un site (x, y) est choisie parmi l'ensemble $L'(x, y) \subset L$ contenant les étiquettes des sites préalablement étiquetés, et qui sont situés dans une fenêtre de taille $W \times W$, centrée en (x, y) . Formellement, $L'(x, y)$ peut être exprimé comme suit :

$$L'(x, y) = \{l | \exists (x', y') \in S - S', (x' - x, y' - y) \in [-W/2, W/2]^2 \wedge f(x', y') = l\} \quad (4.12)$$

Dans l'algorithme de l'ICM, ceci permettra, lors de la sélection de l'étiquette $f^k(x, y)$ à l'itération k , de ne considérer que les étiquettes de l'ensemble $L'(x, y)$.

Les deux heuristiques précédentes, permettent à la fois, l'accélération de la recherche du minimum de l'énergie à posteriori $U'(F/D)$, et aussi de satisfaire la contrainte de continuité des régions. En effet, en supposant que la distance entre les bords de deux régions coplanaires R et R' est supérieure à W , les étiquettes l_R , et $l_{R'}$, correspondant respectivement à R et R' , ne peuvent pas appartenir en même temps à un même ensemble $L'(x, y)$. Si on suppose par exemple que le site (x, y) est plus proche de R , son étiquette $f(x, y)$ ne peut avoir la valeur $l_{R'}$, et ce malgré l'égalité des énergies à posteriori $U'(l_R/d(x, y))$ et $U'(l_{R'}/d(x, y))$.

4.3 Régularisation floue des contours

4.3.1 Etat de l'art

Plusieurs auteurs ont proposé le classement flou pour améliorer les méthodes de segmentation, notamment les méthodes dépendantes du domaine, telles que les méthodes de segmentation d'images à résonance magnétique (IRM) du cerveau (Xue *et al.* (2000); Shen *et al.* (2005); Colliot *et al.* (2006)). La classification floue est bien appropriée à la segmentation des IRM, parce que dans ce type d'images, les classes sont préalablement

connues et correspondent aux différentes parties anatomiques du cerveau. Cependant, peu d'auteurs ont utilisé la classification floue dans la segmentation d'images indépendamment du domaine d'application. Pour les images de profondeur, les classes correspondent aux différentes parties visibles des surfaces d'objets figurant dans la scène; leur nombre ainsi que leurs prototypes sont donc inconnus. Frigui et Krishnapuram (1996) ont présenté un algorithme de classification floue et son application à la segmentation d'images de profondeur. Dit "Robust Competitive Agglomeration", l'algorithme approxime l'image de profondeur par une surface quadratique. Afin d'estimer le nombre de classes, l'algorithme procède par une sur-segmentation de l'image. Due à la nature compétitive de l'algorithme, seulement un seul prototype de chaque surface va survivre. Burgiss et al. procèdent par incorporation de plusieurs caractéristiques d'une multitude de types d'images, dans le but de mettre en oeuvre un détecteur robuste de contours pour l'analyse de scènes tridimensionnelles (Burgiss *et al.* (1998)). Leur approche de segmentation est basée sur l'application de l'algorithme du watershed à une carte de caractéristiques floues. Soodamani et Liu (2000) ont utilisé des attributs flous avec un algorithme génétique pour augmenter la performance de reconnaissance d'un système basé sur le modèle, et utilisant des images de profondeur. Les auteurs prétendent que l'exactitude de reconnaissance d'instances d'objets préalablement connus, et la capacité de généralisation par reconnaissance d'instances inconnues d'objets connus, sont largement améliorées.

4.3.2 Aperçu de la méthode

La régularisation floue des contours consiste à l'étiquetage des pixels sur les frontières de régions en utilisant l'algorithme du "Fuzzy C-Mean" (FCM) (Bezdek (1981)). L'algorithme FCM est appliqué à chaque paire de régions adjacentes, pour assigner chaque pixel de la frontière, à une des deux régions. Ainsi, contrairement aux méthodes floues existantes, nous utilisons seulement deux ensembles avec des prototypes initiaux bien définis. La classification de contours proposée a permis de formuler la continuité des régions en définissant une contrainte sur les ensembles possibles auxquels un pixel peut appartenir. En effet, l'étiquette d'un pixel donné est choisie parmi les deux étiquettes correspondant aux régions dont la frontière contient le pixel.

4.3.3 La segmentation d'images par classification floue

La segmentation floue d'images est basée sur la théorie des ensembles flous, introduite par Zadeh dans Zadeh (1965). Dans la segmentation floue, les régions d'une image sont considérées comme des ensembles flous, pour lesquelles chaque pixel de l'image a un degré d'appartenance à chacune des régions. Notez que pour la définition conventionnelle de la segmentation d'images, qui est basée sur une classification précise, à un moment donné, un pixel appartient exclusivement à une seule région de l'image. Plusieurs algorithmes de classification floue ont été utilisés en reconnaissance de formes ; cependant, le fameux algorithme du FCM (Fuzzy C-Mean) (Bezdek (1981)) demeure largement utilisé en segmentation d'images (Baraldi et Blonda (1999)). L'algorithme du FCM consiste en la partition d'un ensemble de N individus représentés par un ensemble de vecteurs de caractéristiques $\{p_1, p_2, \dots, p_N\}$, en C classes, représentées par leurs prototypes $\{v_1, v_2, \dots, v_C\}$. Le but est de minimiser une fonction objective J définie comme suit (Bezdek (1981)) :

$$J = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m D(v_i, p_j) \quad (4.13)$$

où $D(v_i, p_j)$ représente une certaine distance entre l'individu j et le centre de la classe i , et u_{ij} est le degré d'appartenance de l'élément j à la classe i . Le paramètre m représente le facteur de fuzzification, initialisé à la valeur typique 2. La meilleure classification selon le framework de la classification floue, est obtenue par le calcul de la matrice $U = [u_{ij}]$, $i = 1 \dots C$; $j = 1 \dots N$, qui minimise la fonction objective J . Donc, à chaque itération t de l'algorithme du FCM, les prototypes et les degrés d'appartenance sont mis à jour comme suit :

$$u_{ij} = \left[\sum_{k=1}^C \left(\frac{D(v_i, p_j)}{D(v_k, p_j)} \right)^{\frac{2}{m-1}} \right]^{-1} \quad (4.14)$$

$$v_i = \frac{\sum_{j=1}^N u_{ij}^m p_j}{\sum_{j=1}^N u_{ij}^m} \quad (4.15)$$

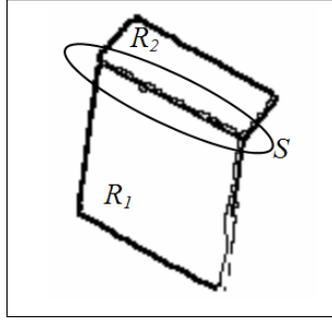


FIG. 4.2 – L'ensemble flou S , représentant l'ensemble des pixels appartenant à la frontière des deux régions

Les itérations sont répétées jusqu'à la convergence de la fonction J :

$$|J^{(t)} - J^{(t-1)}| \leq \varepsilon \quad (4.16)$$

où ε est un certain seuil, initialisé à 10^{-4} dans nos expérimentations.

4.3.4 Classement flou des pixels de contours

Dans notre cas, nous exécutons l'algorithme du FCM pour chaque paire de régions adjacentes. Soient R_1 et R_2 deux régions adjacentes ($C = 2$), et soit S l'ensemble de pixels appartenant à la frontière entre les deux régions (Fig. 4.2). S est obtenu par l'examen de tous les pixels appartenant au voisinage de la frontière entre les deux régions R_1 et R_2 , en considérant les pixels avec une distance plane à une des deux régions inférieure à un seuil donné (W pixels).

$$\begin{aligned} S = & \{p/p \notin R_1 \wedge \exists q \in R_2 : \max\{|x_p - x_q|, |y_p - y_q|\} < W\} \cup \\ & \{p/p \notin R_2 \wedge \exists q \in R_1 : \max\{|x_p - x_q|, |y_p - y_q|\} < W\} \cup \\ & \{p/p \notin R_1 \cup R_2 \wedge \exists q \in R_1 \cup R_2 : \max\{|x_p - x_q|, |y_p - y_q|\} < W\} \end{aligned} \quad (4.17)$$

Soit $p_j = (x_j, y_j, z_j) \in S$, et soit $v_i = (a_i, b_i, c_i)$ le prototype du cluster i ; $i = 1, 2$. Le prototype de chaque classe i est initialisé par le barycentre r_i de la région R_i comme

suit :

$$v_i = r_i = \frac{1}{|R_i|} \sum_{k=1}^{|R_i|} p_{ik} \quad (4.18)$$

où $p_{ik} = (a_{ik}, b_{ik}, c_{ik})$ sont les paramètres de l'équation du plan au pixel k de la région R_i . La distance entre un pixel p_j dans S et le centre v_i d'une région R_i est définie comme suit :

$$D(v_i, p_j) = \frac{(a_i x_j + b_i y_j + c_i - d_{x_j, y_j})^2}{2\sigma_i^2} \quad (4.19)$$

Après chaque itération, l'ensemble S est partitionné en R'_1 et R'_2 , comme suit :

$$R'_1 = \{p_k / u_{1k} > u_{2k}; k = 1 \cdots N\} \quad (4.20)$$

$$R'_2 = S - R'_1 \quad (4.21)$$

Le nouveau centre r_i de la région R_i ; $i = 1, 2$ est calculé comme suit :

$$r_i = \frac{r_i |R_i| + v_i |R'_i|}{|R_i| + |R'_i|} \quad (4.22)$$

La fonction objective est calculée en considérant les deux ensembles R'_1 et R'_2 :

$$J = \sum_{i=1}^2 \sum_{j=1}^N u_{ij}^2 D(v_i, p_j) \quad (4.23)$$

A la convergence, chaque pixel de l'ensemble S est de façon optimale proche à l'une des deux régions, au sens de minimisation de la fonction objective J . Après defuzzification en utilisant la méthode du maximum, les degrés précis d'appartenance sont obtenus, et chaque pixel de la frontière est assigné à la région correspondante. L'algorithme du FCM est exécuté pour chaque paire de régions adjacentes. Finalement, tous les contours sont régularisés d'une façon optimale, au sens de la classification floue.

L'étiquetage des pixels de contours, en ne considérant que les deux régions adjacentes associées, a permis de satisfaire la contrainte de continuité de régions. En effet, sans cette contrainte, les pixels appartenant à des régions coplanaires sont indifféremment assignés à n'importe laquelle de ces régions, parce qu'ils sont à une même distance à chacune d'elles. Selon notre schéma d'étiquetage, si nous supposons que la distance entre deux régions coplanaires R_1 et R_2 est supérieure à W , les pixels de R_1 et ceux de R_2 ne peuvent pas être dans le même ensemble S , formé des pixels proches de la frontière entre les deux régions R_1 et R_2 .

4.4 Conclusion

Afin d'améliorer les résultats de segmentation des images et prévoir un traitement spécial des contours de régions, nous avons proposé dans ce chapitre deux méthodes de régularisation de contours. Pour la première méthode, basée sur l'inférence bayésienne, la segmentation produite par le système multi-agents est raffinée en procédant à un étiquetage bayésien des pixels appartenant aux pourtours des régions. Pour ce faire, nous avons introduit un nouveau modèle du champ aléatoire de Markov qui permet à la fois, la modélisation du smoothness des régions et des lignes de contours. Pour la seconde méthode, le raffinement de la segmentation initiale est fait en utilisant une version adaptée de l'algorithme du FCM. Les régions extraites en première étape ont permis d'obtenir le nombre et les centres de classes. Ceci a permis une classification floue supervisée, avec une bonne initialisation des paramètres utilisés.

Chapitre 5

Détection de contours par les réseaux de neurones chaotiques

5.1 Introduction

Dans ce chapitre, nous montrons comment une paire de neurones : excitateur et inhibiteur est utilisée pour construire un réseau de neurones à dynamique complexe pour la détection de contours dans les images de profondeur. L'image dense, est considérée comme une surface $3D$ sur laquelle les contours sont situés aux frontières entre les différentes *patches* de surfaces. A l'état de repos, ou loin des pixels de contours, les paires du réseau, exhibent un comportement chaotique, par contre, la dynamique est réduite sur les pixels séparant les différentes *patches* de surfaces. Pour réduire la dynamique en ces pixels, une caractéristique de l'image doit être utilisée comme stimulus de la paire de neurones afin d'agir sur le régime chaotique en le réduisant et le basculant en un régime périodique. Dans ce travail, nous traitons les surfaces complexes, pour lesquelles les opérateurs conventionnels, basés sur le gradient, ne peuvent être utilisés directement pour détecter certains contours, notamment les contours lissés. Dans ce travail, nous procédons par reconstruction, localement, des surfaces $3D$, à partir d'un voisinage donné, puis nous utilisons une propriété locale de la surface construite, pour la sélection des pixels de contours.

5.2 Approches connexionnistes pour la détection de contours

La discontinuité ou l'homogénéité des informations visuelles contenues dans une image sont les éléments de base sur lesquels sont fondées les différentes approches et techniques de segmentation d'images. Les contours sont souvent générés aux frontières physiques des différents objets d'une scène, et représentent de puissants descripteurs des images (Heraud et Monga (1995)). Ces contours peuvent être apparents, visiblement observés aux frontières des surfaces ; ou lissés, se situant sur les intersections des surfaces ayant la même orientation sur les frontières qui les séparent. Plusieurs opérateurs ont été utilisés pour la détection des contours apparents (de types marche et toit). Ces opérateurs considèrent la profondeur de la même manière comme pour le niveau de gris. Ceci ne permet pas de prendre en compte les spécificités géométriques des surfaces 3D. Des approches connexionnistes ont été ultérieurement étudiées afin d'effectuer une panoplie de tâches relevant du domaine de la perception, et ce, en utilisant les résultats de recherche en neurobiologie sur la façon utilisée par le système cérébral chez les mammifères. Dans le cortex cérébral de ces derniers, plusieurs régions formées de réseaux de neurones, fortement connectés, ont été identifiées comme étant les berceaux des différentes facultés de perception telles que la vision et l'odorat (Freeman et Skarda (1987)).

Depuis cela, plusieurs travaux reposant sur l'approche connexionniste, dans le domaine de la vision artificielle, ont été publiés. La faculté de perception des arrêtes qui séparent les régions d'objets d'une scène est une tâche fondamentale d'un système de vision. Elle se traduit, au niveau du cortex cérébral, par une phase de synchronisation d'une activité neuronale oscillatoire (Gray *et al.* (1989)). Ce phénomène représente le mécanisme du "lien visuel" (Visual binding) avec lequel les stimuli, qui correspondent aux propriétés locales d'un objet visuel, et après traitement parallèle dans les différentes régions du cortex visuel, sont intégrés d'une façon cohérente dans des zones plus hautes du cerveau. La tâche devient plus complexe lorsque les régions d'objets sont séparées par des contours lissés. Nous proposons, dans notre méthode de détection, l'utilisation du laplacien de l'image comme stimulus d'un réseau de neurones particulier afin de détecter, à la fois, les contours apparents et les contours lissés.

5.3 Les Réseaux de neurones chaotiques

Suite aux travaux de Freeman ([Freeman et Skarda \(1987\)](#)), la présence du chaos dans l'activité cérébrale est devenue une certitude chez les connexionnistes. Selon les expériences de Freeman, le chaos est réduit en présence de stimuli externes connus. En dehors de l'utilisation des réseaux de neurones classiques (sans dynamiques complexes) dans le traitement d'images, plusieurs travaux utilisant des réseaux à dynamiques complexes pour la segmentation d'images ont été publiés. La majorité de ces travaux focalisaient leurs investigations sur la séparation de l'objet du fond de l'image et utilisaient des images à niveaux de gris. En effet, pour la séparation objet-fond, [Grossberg et Somers \(1990\)](#), utilisaient un réseau de neurones à dynamique complexe. Certains neurones votent pour l'objet, et les autres pour le fond. [Han et al. \(1998\)](#) ont utilisé un réseau de type Hopfield, un apprentissage de patterns permet de segmenter successivement toute entrée, qui sera considérée comme superposition de plusieurs patterns stockés (appris). Un réseau de paires de neurones a été utilisé par [Sinha et Bazak \(1998\)](#) pour la séparation objet-fond dans une image à niveaux de gris. Une configuration plane de paires de neurones excitateurs et inhibiteurs permettait de séparer les objets du fond de l'image.

5.4 Contours lissés dans les images de profondeur

Nous entendons par contour lissé l'ensemble des pixels d'une image de profondeur formant l'intersection de deux surfaces différentes et qui ont la même orientation à la frontière qui les sépare. Contrairement aux contours lissés, les contours apparents, correspondent à une discontinuité d'une mesure locale, de l'information 3D ([Horaud et Monga \(1995\)](#)), prise comme étant un niveau de gris, et peuvent être détectés par les opérateurs conventionnels dérivés du gradient. Ces méthodes ne peuvent pas être utilisées avec les contours lissés, car d'une part, elle ne permettent pas la détection des contours lissés, et elles engendrent des pixels de contours pour tous les points appartenant aux surfaces courbées, d'une autre part. Les pixels d'un contour lissé sont caractérisés par une variation de la courbure dans le sens perpendiculaire au contour ([Horaud et Monga \(1995\)](#)).

Dans la figure 5.1a, l'arc situé entre l'hémisphère et le cylindre correspond à un contour lissé. A chaque pixel de cet arc il y a transition de la courbure, d'une valeur non nulle pour l'hémisphère à une valeur nulle pour le cylindre, dans le sens perpendiculaire à l'arc (parallèle au cylindre). La figure 5.1b montre le résultat de détection de contours par l'opérateur du laplacien opéré directement sur les données 3D de l'image, après avoir considéré un seuil de détection qui a permis de ne pas ressortir tous les points des deux surfaces courbées. Les pixels détectés sont ceux appartenant uniquement aux contours apparents.

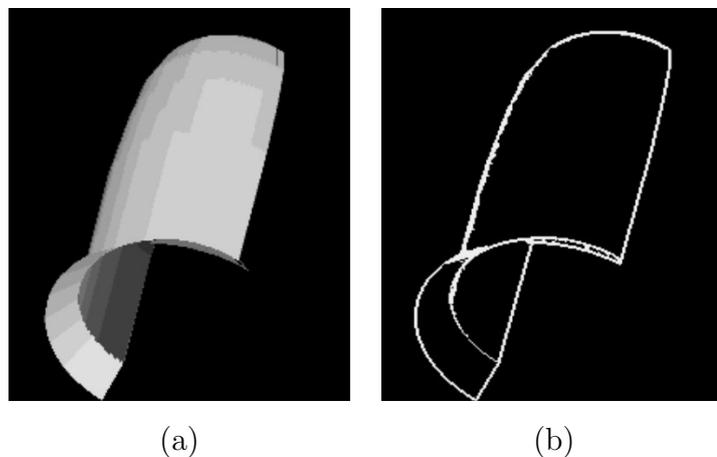


FIG. 5.1 – Contours lissés dans une image de profondeur : (a) Image de profondeur ; (b) Contours détectés par l'opérateur du laplacien

Dans notre travail, nous nous sommes basés sur le laplacien de la surface locale pour définir un nouvel opérateur permettant la détection des contours lissés. Notons ici, que les contours apparents sont également détectés par ce nouvel opérateur, car à leur niveau, il y a également variation -dans ce cas, brutale- de la courbure. Afin de pouvoir définir les propriétés géométriques locales au niveau d'un pixel (x, y) , en utilisant son voisinage immédiat, nous avons opté pour l'approximation de la surface 3D par une fonction polynomiale de degré 3, utilisée par [Haralick et Shapiro \(1985\)](#), et dont le principe est exposé dans la suite de la section.

Soit une image 3D I , son approximation par une fonction polynomiale, I^* , sur un

voisinage $[-R \cdots R, -C \cdots C]$ est définie comme suit :

$$I^*(r, c) = f_0 + f_1 r + f_2 c + f_3(r^2 - 2/3) + f_4 r c + f_5(c^2 - 2/3) + f_6(r^2 - 2/3) + f_7(c^2 - 2/3) + f_8(r^2 - 2/3)(c^2 - 2/3) \quad (5.1)$$

Pour le voisinage immédiat d'un pixel (x, y) ; $\{(x + r, y + c), r = -1.. + 1, c = -1 \cdots + 1\}$, les coefficients f_i du polynôme sont obtenus par convolution de l'image, respectivement avec les masques suivants :

$$f_0 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, f_1 = \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, f_2 = \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$f_3 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix}, f_4 = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}, f_5 = \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix}$$

$$f_6 = \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix}, f_7 = \frac{1}{6} \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix}, f_8 = \frac{1}{9} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Le nouvel opérateur, qu'on propose dans ce travail, est obtenu par la seconde dérivation du polynôme selon les deux directions principales (0° et 90°). Après calcul et simplification, nous obtenons les masques de convolutions, correspondant respectivement aux deux dérivées secondes :

$$\frac{\partial^2 I^*}{\partial r^2} = 2f_3 + 2f_6 c + 2f_8(c^2 - 2/3) \quad (5.2)$$

$$\frac{\partial^2 I^*}{\partial c^2} = 2f_5 + 2f_7 r + 2f_8(r^2 - 2/3) \quad (5.3)$$

Le masque du laplacien moyen, selon les deux direction, est obtenu par la variation

de r et c dans l'intervalle $[-1 \cdots 1]$, et est donné par :

$$\begin{bmatrix} 2 & -1 & 2 \\ -1 & 4 & -1 \\ 2 & -1 & 2 \end{bmatrix}$$

Les pixels de contours, sont détectés au passage par zéro de la valeur obtenue par la convolution de l'image avec le masque précédent. La particularité de l'opérateur réside dans le fait, qu'il procède à la détection des pixels de contours, en estimant les nuages de points (voisinages de 9 pixels) par une surface 3D. Ceci permettra une prise en compte des formes courbées 3D, problème qui est mal solutionnée, en considérant la profondeur comme niveau de gris. La valeur du laplacien, ainsi définie sera utilisée comme stimulus pour le réseau de neurones proposé, afin de stabiliser sa dynamique sur les pixels de contours.

5.5 Détection des contours par réduction du chaos

Dans ce travail, nous allons reprendre le modèle de base, utilisé dans [Sinha et Bazak \(1998\)](#), et qui consiste à une paire de neurones : l'un excitateur et l'autre inhibiteur, pour construire un réseau de neurones chaotique, permettant, non seulement la séparation objet-fond, mais permettra aussi l'extraction des contours d'objets dans une image de profondeur. Le réseau de neurones, en absence de stimuli externes, exhibe une dynamique complexe. La dynamique, est contrôlée par l'image d'entrée, dont chaque pixel est considéré comme stimulus, et la complexité de la dynamique est réduite au niveau des éléments qui correspondent aux pixels n'appartenant pas aux arêtes entre les différentes surfaces (pixels à l'intérieur des régions homogènes),(Fig. 5.1).

5.5.1 Le modèle de réseau utilisé

La brique de base du réseau utilisé, consiste à une paire couplée de neurones : excitateur et inhibiteur (Fig. 5.2).

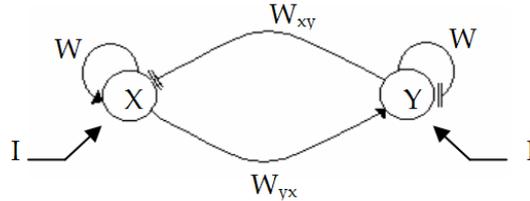


FIG. 5.2 – Paire de neurones : Excitateur-inhibiteur

I et I' sont les stimuli externes respectivement de l'excitateur X et l'inhibiteur Y . La fonction de transfert est une sigmoïde de la forme :

$$F_u(z) = \begin{cases} 1 - \exp -uz, z \geq 0 \\ 0, \text{ sinon} \end{cases} \quad (5.4)$$

La dynamique est calculée en temps discret. L'évolution de la dynamique suit les équations suivantes :

$$\begin{aligned} x_{n+1} &= F_a(W_{xx}x_n - W_{xy}y_n + I_n) \\ y_{n+1} &= F_b(W_{yx}x_n - W_{yy}y_n + I'_n) \end{aligned} \quad (5.5)$$

Considérons : $W_{xx} = W_{xy} = W_{yx} = W_{yy} = 1$, et soit :

$$z_n = x_n - y_n$$

nous obtenons l'équation générale de la dynamique de la paire de neurones :

$$z_{n+1} = F_a(z_n + I_n) - F_b(z_n + I'_n) \quad (5.6)$$

En dynamique spontanée ($I = 0$), la variation de a et b engendre des transitions du régime, de la paire de neurones, du point-fixe au régime chaotique en passant par des régimes périodiques. En dynamique contrainte, avec l'introduction du stimulus externe I , nous retrouvons les 3 types de régime, néanmoins, la transition ne dépend pas uniquement des paramètres a et b , mais aussi de la grandeur du stimulus I . Une analyse théorique ([Sinha et Bazak \(1998\)](#)) de l'équation caractéristique :

$$z_{n+1} = -e^{-az_n+I} + e^{-bz_n+I} \quad (5.7)$$

a permet la déduction d'une valeur critique I_c du stimulus I en fonction de a et b , pour laquelle le régime de la paire transite du régime point-fixe au régime chaotique et vice-versa.

$$I_c = \frac{1 - 2a/b}{b^{a/b} - a^2/b} + \frac{1}{b} \ln(b/e) \quad (5.8)$$

En effet, les éléments ayant une entrée I inférieure à I_c ($I < I_c$), exhibent une dynamique complexe (chaotique). Par contre ceux ayant une entrée supérieure à I_c , montrent une dynamique point-fixe. Les éléments aux frontières entre les deux régimes précédents exposent une dynamique cyclique.

5.5.2 Sélection des pixels de contours

Dans ce travail, nous avons considéré comme stimulus externe, la valeur moyenne du laplacien de la fonction polynomiale approximant localement la surface au point (i, j) , en considérant son voisinage immédiat. Il est à noter qu'il est possible de définir d'autres

variantes de mesure des propriétés locales d'une surface 3D, telles que la courbure de Gauss et la courbure moyenne.

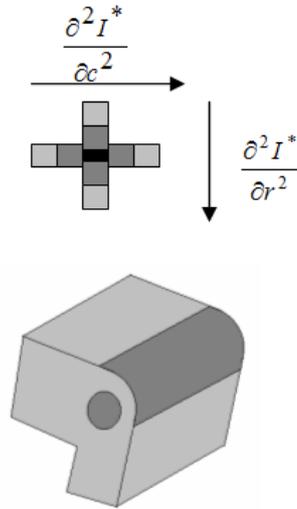


FIG. 5.3 – Calcul de la valeur moyenne du laplacien de la surface. Les deux segments de pixels entre la surface courbée et les deux plans correspondent au passage par zéro de la valeur moyenne du laplacien. Ces pixels représentent les contours lissés figurant dans l'image

Pour choisir les paramètres caractéristiques a et b de la paire de base, nous avons fixé le stimulus critique I_c à zéro (moyenne nulle). En suite, et pour des valeurs arbitrairement choisies de a , nous déduisons les valeurs correspondantes de b . De ce fait, les éléments $z^{i,j}$ du réseau qui correspondent à des pixels de contours vont se voir osciller dans un régime chaotique. Par contre, les éléments $z^{i,j}$ qui correspondent aux pixels à l'intérieur des surfaces vont converger et se stationner sur un régime point-fixe.

5.5.3 Couplage des paires d'éléments

La mise en oeuvre précédente nécessite un près traitement qui consiste en le calcul explicite de la moyenne du laplacien et de la considérer comme stimulus à la paire de neurones (excitateur et inhibiteur). Ceci élimine, en fait, l'intérêt du couplage des paires entre elles, et donc l'intérêt du réseau lui-même. Pour remédier à ce problème (de près traitement), le couplage de paires voisines permet d'avoir un réseau de neurones

permettant la détection des pixels de contours sans le calcul préalable de la moyenne du laplacien. En effet, la moyenne sera implicitement représentée en réalisant un couplage adéquat des neurones voisins. Le couplage est réalisé en se basant sur le masque précédemment obtenu, approximant la moyenne du laplacien de la surface.

Soit l'élément $z^{i,j}$; son équation caractéristique est donnée par :

$$z_{n+1}^{i,j} = \frac{F_a(z_n^{i,j} + |\sum_{k=-1}^1 \sum_{l=-1}^1 A_{k,l} z_n^{i+k,j+l}|) - F_b(z_n^{i,j} + |\sum_{k=-1}^1 \sum_{l=-1}^1 A_{k,l} z_n^{i+k,j+l}|)}{1} \quad (5.9)$$

avec $z_0^{i,j} = I(i, j)$ et

et

$$A = \begin{bmatrix} 2 & -1 & 2 \\ -1 & 4 & -1 \\ 2 & -1 & 2 \end{bmatrix}$$

5.6 Test du modèle

Pour nos tests, nous avons utilisé des images de profondeur avec une résolution de 256×256 , où à chaque pixel (x, y) , $I(x, y)$ représente sa profondeur. Une matrice de 256×256 de paires de neurones (excitateurs et inhibiteurs) représente le réseau de neurones. Les paires sont couplées entre elles dans un voisinage de rayon unitaire. Après convergence de la dynamique, les pixels de contours exhibent une dynamique chaotique. Initialement, nous avons pris une moyenne nulle, pour la valeur critique du stimulus ($I_c = 0$), et plusieurs couples (a, b) ont été testés. En absence de bruit, les résultats étaient satisfaisants (Fig. 5.4), mais en présence de bruit les performances du système se dégradent significativement.

Nous avons repris le calcul des couples (a, b) , pour une moyenne critique I_c non nulle

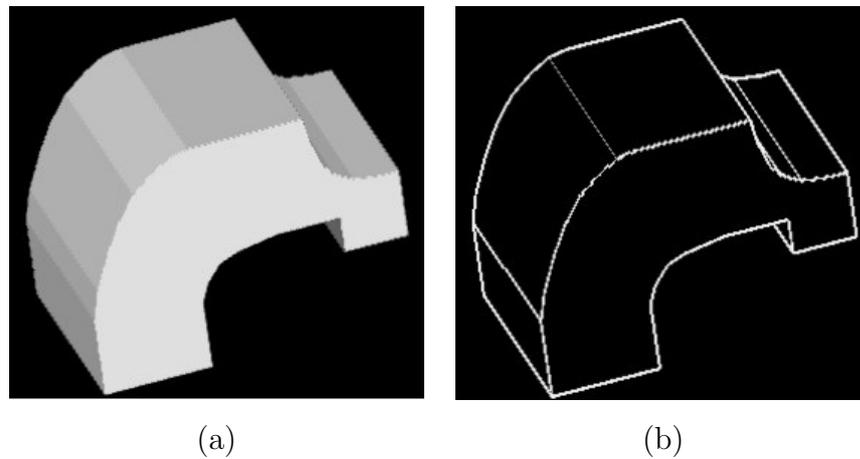


FIG. 5.4 – Test du modèle en absence de bruit : (a) Image de profondeur ; (b) Contours détectés

mais suffisamment proche de zéro. Pour les mêmes valeurs de a , prises en absence de bruit, nous procédons, de la même sorte, au calcul des valeurs b correspondantes en utilisant, cette fois-ci, une moyenne critique $I_c \simeq 0$. En effet, si on considère la moyenne de distorsion ΔI de l'image, due à un bruit additif, nous concluons l'incertitude absolue $\Delta I_c = 16\Delta I$ de la moyenne des valeurs du laplacien (somme des valeurs absolues des éléments du masque). La valeur ΔI_c sera prise comme valeur critique I_c . Les résultats, même en présence de bruit, ont été significativement améliorés (Fig. 5.5).

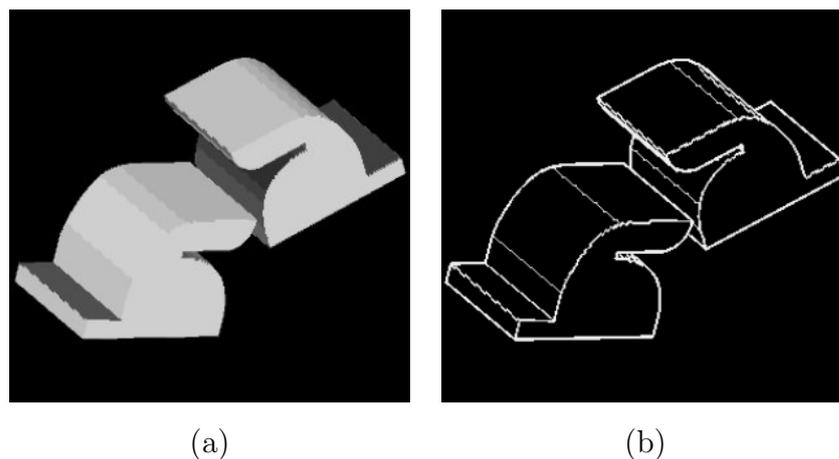


FIG. 5.5 – Test du modèle en présence de bruit avec $I_c > 0$: (a) Image de profondeur ; (b) Contours détectés

5.7 Conclusion

Dans ce chapitre, nous avons utilisé un modèle particulier de réseaux de neurones pour la détection de contours dans une image de profondeur. Le réseau de neurones utilisé exhibe une dynamique chaotique qui est contrôlée par l'image en entrée. La morphologie locale des surface 3D était modélisée afin d'être utilisée comme stimulus et dont le résultat consiste en la réduction de la dynamique des éléments correspondant aux pixels à l'intérieur des surfaces des objets. En perspective à ce travail, il est possible d'envisager d'autres variantes d'expressions des propriétés locales d'une surface 3D telle que la moyennes selon quatre directions, la courbure de Gauss, ou la courbure moyenne. Ce qui permettra la généralisation de la méthode pour des surfaces complexes. Néanmoins, les expressions non-linaires de certaines expressions rendent la tâche difficile et nécessitent, une investigation théorique poussée.

Chapitre 6

Une architecture coopérative pour la reconnaissance d'objets 3D

6.1 Introduction

Tout système de vision tridimensionnelle doit générer une description symbolique des objets contenus dans la scène. La description symbolique générée est utilisée pour établir les hypothèses de correspondance objet-modèle. Cependant, la partition du processus de vision, basé sur le modèle (Horaud et Monga (1995)), en étapes conventionnelles (détection, représentation symbolique, et appariement objet-modèle) est non plus commode pour élaborer des systèmes fiables et robustes en vision artificielle. Les erreurs dues à la nature incertaine de l'information visuelle, ainsi que les approximations faites lors du passage d'une étape à une autre (sans retour sur les étapes précédentes) font aboutir souvent à un échec d'interprétation. Avec un environnement sans modèle ou mal modélisé, tel qu'une image, la coopération d'entités distribuées opérant d'une façon autonome peut s'avérer une solution efficace, pour laquelle un processus continue à se dérouler correctement dans les situations où il devrait s'arrêter ou produire de faux résultats. Le problème majeur en interprétation d'images réside dans l'incapacité de prévoir l'environnement (image) dans lequel s'exécutent les programmes d'interprétation. Le recours à une approche basée agent, permet d'adapter les traitements et la décision à la situation courante à la fois de l'environnement et des agents.

6.2 Aperçu du système

Dans ce chapitre, nous présentons une architecture coopérative basée sur une approche multi-agents pour l'interprétation de scènes à partir d'images de profondeur. Un ensemble d'agents, dits agents reconnaisseurs sont lancés pour explorer l'image dans le but de poursuivre les contours. Suite à cela ils établissent les différentes hypothèses de correspondance objet-modèle. Les différentes étapes conventionnelles du processus de vision sont entrelacées. A un moment donné, la prédiction d'une correspondance objet-modèle permet de prévoir les segments de contours qui ne sont pas encore parcourus, et par conséquent, les agents libres qui explorent l'image, sont orientés pour parcourir les segments prévus. La coopération des agents qui parcourent les contours d'objets consiste à ajuster les hypothèses de correspondance objets-modèle. Lors du parcours d'un segment de contour, et au passage par un sommet (intersection de deux ou plusieurs segments), un agent communique ce nouvel état et ce, dans le but de réajuster les différentes hypothèses de correspondance relevant du sommet atteint. Certaines seront maintenues, tandis que toutes les hypothèses qui sont incompatibles à ce nouvel état seront écartées. Les étapes respectivement de détection et de reconstruction de segments, ainsi que celle de mise en correspondance, sont entrelacées. Le long du processus, les agents libres sont orientés vers les segments non parcourus, et ce en considérant les hypothèses de correspondance les plus probables.

6.3 Calcul d'appariement objet-modèle

Nous avons utilisé une méthode d'appariement objet-modèle dit appariement par génération de bases relatives aux modèles (BRM), que nous avons développé dans (Benlamri *et al.* (1993, 1994)). Le principe de la méthode consiste en la comparaison d'un objet, exprimé par rapport au référentiel absolu de la scène, aux différents modèles candidats, exprimés chacun par rapport à sa propre base relative (BRM) (Fig. 6.1). Cette base qui représente un référentiel relatif au modèle, est générée suite à une hypothèse d'appariement établie par la mise en correspondance géométrique entre un sous-ensemble de primitives (modèle local de matching) de l'objet et un sous-ensemble de primitives du modèle.

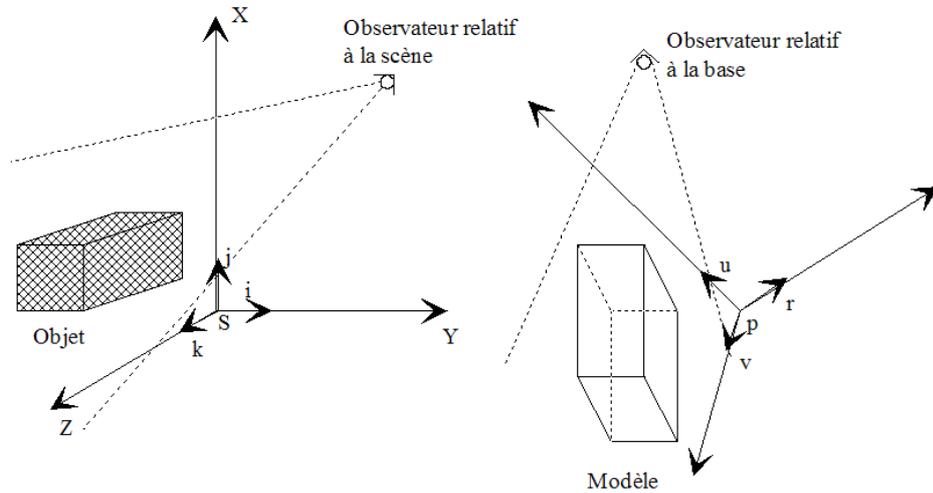


FIG. 6.1 – Principe du calcul de la BRM

6.3.1 Formulation du problème

Soit un objet O_i , extrait de la scène et exprimé par rapport au référentiel absolu (S, i, j, k) . Le problème de la reconnaissance consiste à désigner un ou plusieurs modèles correspondant à l'objet en question et de déterminer pour chacun d'eux, les transformations géométriques (translations et rotations) appliquées à ces modèles, pour qu'il y ait superposition totale avec l'objet. Le matching est basé sur un seul modèle local associé à l'objet à faire correspondre à plusieurs modèles locaux d'un modèle donné. Pour ce qui est de l'objet, le modèle local est constitué de deux segments adjacents entièrement visibles, soit (S_i, S_j) pour lesquelles on définit les paramètres géométriques suivants (Fig. 6.2) :

- N_i : le vecteur unitaire du segment S_i
- N_j : le vecteur unitaire du segment S_j
- A_{ij} : l'angle entre N_i et N_j
- G_{ij} : le point d'intersection des deux segments N_i et N_j (le point en commun des deux segments)

Pour un modèle donné, les mêmes paramètres sont calculés pour toute correspondance du couple de segments de l'objet (S_i, S_j) et toutes les couples compatibles du

modèle. Pour ce dernier, il s'agit de déterminer tous les couples (S_k, S_l) de segments adjacents et vérifiant les conditions suivantes :

- $A_{kl} = A_{ij}$
- $\|S_i\| = \|S_k\|$
- $\|S_j\| = \|S_l\|$

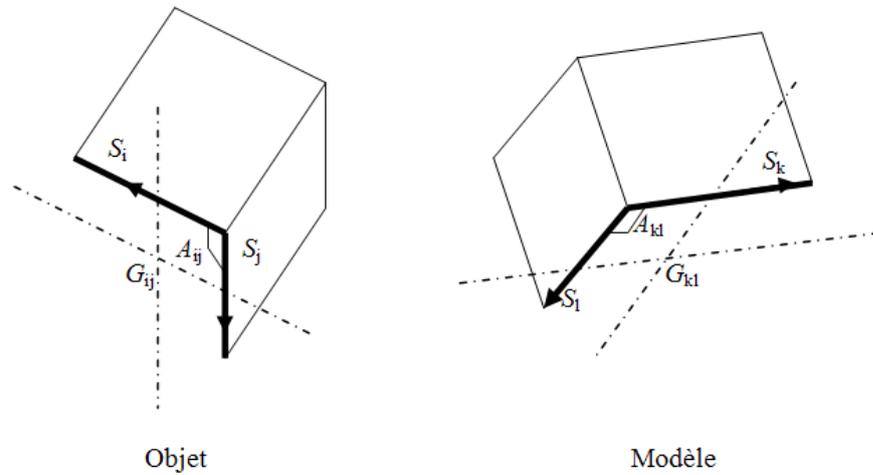


FIG. 6.2 – Formulation du calcul de la BRM

6.3.2 Calcul de la BRM

Soit B la base orthonormée formée par les vecteurs r, u et v qui représentent les vecteurs unitaires de la BRM. La base $B = (r, u, v)^T$ est déterminée de telle sorte que les expressions de N_k, N_l , et $N_k \otimes N_l (N_{kl})$, exprimées dans cette base, soient respectivement égales à celles de N_i, N_j et $N_i \otimes N_j (N_{ij})$ exprimées dans la base absolue $(i, j, k)^T$.

Soit :

$$N_i = (N_{ix}, N_{iy}, N_{iz})^T$$

$$N_j = (N_{jx}, N_{jy}, N_{jz})^T$$

$$N_{ij} = (N_{ijx}, N_{ijy}, N_{ijz})^T$$

$$N_k = (N_{kx}, N_{ky}, N_{kz})^T$$

$$N_l = (N_{lx}, N_{ly}, N_{lz})^T$$

$$N_{kl} = (N_{klx}, N_{kly}, N_{klz})^T$$

La superposition des deux modèles locaux de matching permet d'obtenir les 3 vecteurs unitaire de la base B , et ce en résolvant le système d'équations suivant :

$$N_{k\cdot}(r, u, v)^T = N_i$$

$$N_{l\cdot}(r, u, v)^T = N_j$$

$$N_{kl\cdot}(r, u, v)^T = N_{ij}$$

L'origine de la BRM désignée par le point P , est déterminée de telle sorte que les coordonnées du point G_{ij} associé à l'objet, exprimées par rapport au référentiel (S, i, j, k) , soient égales à celles du point G_{kl} associé au modèle, exprimées par rapport au référentiel (P, r, u, v) . Autrement dit, le centre P est déterminé de telle sorte que le vecteur PG_{kl} exprimé par rapport à la Base B soit identique au vecteur SG_{ik} exprimé par rapport à la base absolue (S, i, j, k) . On aura donc :

$$PG_{kl\cdot}(r, u, v)^T = SG_{ij}$$

d'où P est obtenu en résolvant le système d'équations suivant :

$$PG_{kl\cdot}r = G_{ijx}$$

$$PG_{kl\cdot}u = G_{ijy}$$

$$PG_{kl\cdot}v = G_{ijz}$$

En notation matricielle, la BRM peut être exprimée comme suit :

$$BRM = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -P_x & -P_y & -P_z & 1 \end{bmatrix} \begin{bmatrix} r_x & r_y & r_z & 0 \\ u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6.4 Le système multi-agents proposé

En considérant les opérations qui doivent être exécutées au sein du système, à savoir les opérations de reconnaissance d'objets associées à toute hypothèse de correspondance objet-modèle; et les opérations de confrontation des hypothèses de matching et du contrôle sur les agents, deux types d'agents ont été définis. Le premier représente les agents reconnaisseurs, qui prennent en charge la tâche du suivi des contours, de la reconstruction des segments, et de l'établissement des hypothèses de correspondance. Le second type d'agent existe en une seule instance et représente l'agent coordonnateur qui examine et valide les différentes hypothèses de mise en correspondance, générées et envoyées par les agents reconnaisseurs. Une hypothèse de correspondance objet-modèle permet de prédire l'ensemble des segments qui ne sont pas parcourus par aucun agent reconaisseur. Certains agents libres qui sont au voisinage de ces segments, et qui sont en mouvement aléatoire sont, alors orientés à se placer sur les segments prédits afin de les parcourir. Les paramètres de lissage et de détection sont spécifiques à chaque agent et sont calculés par l'agent coordonnateur en fonction de l'état des agents ([Mazouzi et al. \(2004\)](#)). Les paramètres sont choisis de telle sorte que les agents libres procèdent à un fort lissage et une détection de contours peu sensible. Cependant, ils sont choisis de telle sorte que les agents qui suivent les lignes de contours procèdent, contrairement aux agents libres, par un faible lissage et une détection de contours plus sensible. Les résultats de ces opérations sont représentés au niveau des agents reconnaisseurs, et par conséquent l'image n'est pas affectée. Le long du processus, les agents reconnaisseurs s'organisent en clusters qui représentent les différents objets de la scène. Les agents d'un cluster donné parcourent les segments formant le même objet.

En fonction des hypothèses de correspondance objet-modèle, établies par les agents reconnaisseurs, l'agent coordonnateur calcule la meilleure direction pour chacun des agents reconnaisseurs libres. Plusieurs méthodes peuvent être envisagées pour le calcul

de la direction. Dans notre travail, nous avons considéré la direction associée au segment le plus proche de l'agent en question. Noter que cette direction peut être modifiée suite à l'établissement d'une nouvelle hypothèse de correspondance objet-modèle, car une hypothèse actuellement valide peut ne pas y rester, suite à l'établissement de d'autres hypothèses qui peuvent lui être incompatibles.

6.4.1 L'agent Reconnaisseur

Caractérisé par un comportement autonome, l'agent reconnaisseur parcourt les contours d'objets. Quand un agent reconnaisseur parcourt deux segments adjacents, il génère une hypothèse de correspondance objet-modèle. En dehors des segments de contours un agent reconnaisseur se déplace aléatoirement, ou dans la direction qui lui a été communiquée par l'agent coordonnateur suite à la prédiction de segments par ce dernier. Pour chaque couple de segments parcourus par un agent reconnaisseur, ce dernier procède à l'appariement de l'objet associé, formé par l'ensemble des segments parcourus par l'agent, aux différents modèles candidats. Pour chaque correspondance, le modèle associé est mémorisé au niveau de l'agent reconnaisseur et l'hypothèse de correspondance est communiquée à l'agent coordonnateur. La mise en correspondance objet-modèle consiste en la comparaison des segments de l'objet, exprimés par rapport au référentiel absolu, aux segments du modèle, exprimés par rapport à la BRM associée (Fig. 6.3).

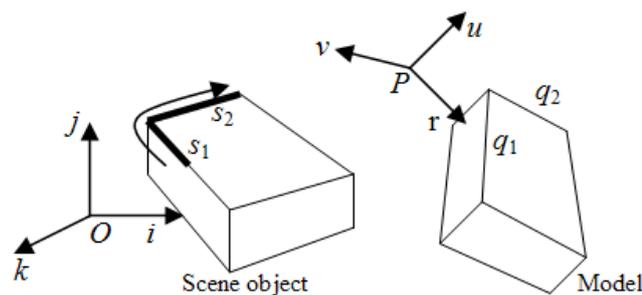


FIG. 6.3 – Génération d'une hypothèse de correspondance objet-modèle

Une hypothèse de correspondance objet-modèle est retenue et communiquée au agent coordonnateur si tous les segments parcourus par l'agent reconnaisseur sont superposés, même partiellement, aux segments du modèle ; projeté dans son référentiel

relatif.

6.4.2 L'agent Coordonnateur

L'agent coordonnateur assure la cohérence des différentes hypothèses de correspondance objet-modèle qui ont été établies par les agents reconnaisseurs. L'évolution des clusters d'agents reconnaisseurs, qui correspondent aux différents objets de la scène, tend à se stabiliser, quand les clusters deviennent constants. A la stabilité des clusters, chacun contient les agents qui parcourent les segments d'un même objet. Le meilleur cluster auquel un agent libre sera affecté est celui qui correspond à la distance minimale entre son centre de gravité, calculé par les positions des différents agents reconnaisseurs formant le cluster, et la position de l'agent libre en question. Une fois le cluster sélectionné, le segment du modèle, projeté dans la BRM, le plus le plus proche de l'agent représente le segment vers lequel l'agent libre sera dirigé.

Chaque hypothèse de correspondance objet-modèle consiste en un couple (M, Pos) du modèle M correspondant à l'objet, et de la position Pos de ce dernier par rapport au modèle. Cette dernière est exprimée simplement par la BRM qui a permis de générer l'hypothèse de correspondance. A la réception de toute nouvelle hypothèse de correspondance, l'agent coordonnateur procède à l'élimination de certaines hypothèses qui sont incompatibles avec l'hypothèse récemment reçue. Une hypothèse stockée (M', Pos') est éliminée par l'hypothèse reçue (M, Pos) si les deux positions Pos et Pos' désignent la même position spatiale, et le modèle M contient au moins un segment parcouru, qui ne correspond à aucun segment dans le modèle M' . L'élimination d'une hypothèse implique l'élimination de l'agent qui l'a créé du cluster correspondant.

Dans la figure 6.4a les agents reconnaisseurs A_1 et A_2 communiquent respectivement à l'agent coordonnateur les hypothèses suivantes :

$$A_1 \rightarrow \{(M_1, Pos), (M_2, Pos)\}$$

$$A_2 \rightarrow \{(M_1, Pos), (M_2, Pos)\}$$

A ce stade, aucune des hypothèses précédentes n'est éliminée parce qu'elles sont simultanément compatibles. Après que chaque agent passe par son sommet correspondant (Fig. 6.4b), les hypothèses communiquées à l'agent coordonnateur sont les suivantes :

$$A_1 \rightarrow \{(M_1, Pos), (M_2, Pos)\}$$

$$A_2 \rightarrow \{(M_1, Pos)\}$$

Cette fois-ci, la seule hypothèse générée par l'agent A_2 ((M_1, Pos)) élimine l'hypothèse (M_2, Pos) générée par l'agent A_1 , parce que les deux modèles M_1 et M_2 sont à la même position Pos , et un segment du modèle M_1 qui n'existe pas dans le modèle M_2 , était parcouru par l'agent A_2 .

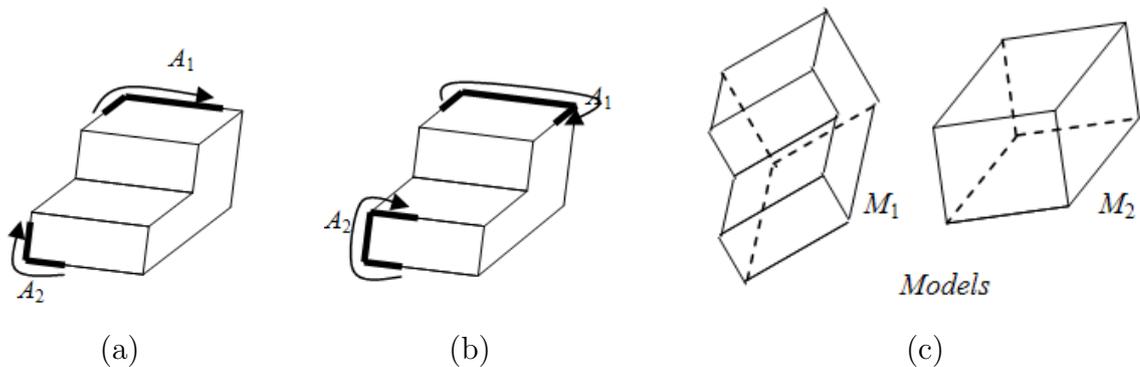


FIG. 6.4 – Hypothèses de correspondance objet-modèle (a,b) ; (c) Modèles d'objets

6.5 Conclusion

Dans ce chapitre, une approche à base d'agents coopératifs était proposée pour la reconnaissance d'objets à partir d'images de profondeur. L'auto-organisation qui s'établit au sein de la population des agents reconnaisseurs permet l'émergence de

clusters stables d'agents, qui correspondent chacun à un objet de la scène. L'imprecision et les erreurs qui peuvent se produire lors de la segmentation sont palliées par une auto-correction, qui consiste à l'élimination de certaines hypothèses qui sont dues à de faux segments générés par les agents. Une limitation que nous avons constatée lors de nos expérimentations réside dans le nombre élevé de modèles locaux de matching qui correspondent aux différents couples de segments adjacents. Pour remédier à ce problème, il est possible d'utiliser les couples de surfaces adjacentes. Cependant, il sera nécessaire dans ce cas de procéder au traitement des parties cachées des objets tridimensionnels ; ce qui représente une tâche fastidieuse en temps de calcul, et non souhaitable pour un système de vision temps réel.

Chapitre 7

Evaluation et discussion

7.1 Introduction

Dans ce chapitre nous présentons une évaluation rigoureuse de performances de l'approche proposée pour la détection de contours dans les images de profondeur. Les résultats, obtenus en utilisant des images réelles, sont comparés, avec ceux obtenus par d'autres systèmes, et ce en utilisant un framework, spécialement conçu pour la comparaison des algorithmes de segmentation d'images de profondeur. La comparaison des résultats de segmentation est une tâche pénible. Ceci est dû, en fait, à la difficulté d'implémenter les algorithmes proposés par de tierces personnes. Les auteurs n'apportent pas suffisamment de détails, nécessaires à l'expérimentation de leurs méthodes. Dans plusieurs cas, ce n'est pas possible de pouvoir reproduire les résultats publiés en utilisant les algorithmes tels qu'ils sont présentés par leurs auteurs. Cependant, et malgré cette difficulté, l'évaluation subjective par l'observation visuelle doit céder place à une évaluation plus rigoureuse, basée sur la quantification de certains critères de performance. Dans le travail publié par ([Hoover *et al.* \(1996\)](#)), qui est le seul à notre connaissance, les auteurs ont développé une méthodologie d'évaluation d'algorithmes de segmentation d'images de profondeur. Pour ce faire, ils ont définie un ensemble de métriques de performance, qui se calculent à partir des nombres d'instances, respectivement de détections correctes, de sur-segmentations, de sous-segmentations, de régions omises, et de régions de bruit. Pour l'ensemble des images utilisées, une segmentation

manuelle correspondant à la réalité terrain GT (Ground Truth) était également réalisée. Le comportement stochastique des agents utilisés rend notre système indéterministe. Ceci nous a obligé à montrer que l'indéterminisme inhérent à notre système, n'influe pas négativement sur les résultats de détection. En effet, une analyse des résultats est faite selon le nombre d'agents et le nombre d'exécutions. Nous présentons respectivement dans la suite du chapitre, le protocole expérimental adopté, le framework comparatif et les résultats des différentes expérimentations.

7.2 Le protocole expérimental

7.2.1 Objectif et Hypothèses

L'objectif de l'expérimentation est d'évaluer la performance du système de segmentation proposé en le comparant à d'autres systèmes. Le framework utilisé est proposé dans Hoover *et al.* (1996). A travers cette expérimentation, nous essayons de vérifier les hypothèses suivantes :

Hypothèse 1 : Malgré le caractère non déterministe du système proposé, dû aux trajectoires semi-arbitraires des agents, les résultats de plusieurs exécutions ne sont pas significativement différents.

Hypothèse 2 : Pour des valeurs optimales des paramètres du système proposé, les résultats obtenus sont significativement équivalents à ceux des meilleurs systèmes utilisés dans la comparaison.

Hypothèse 3 : La correction des contours par les méthodes de régularisation, en l'occurrence la régularisation bayésienne et la régularisation floue, permettent d'améliorer significativement les résultats de segmentation. Les deux méthodes ne présentent pas de différences significatives.

7.2.2 La base d'images

La base d'images est divisée en deux parties distinctes. La première correspond à 10 images d'apprentissage. Elle est utilisée pour l'entraînement des méthodes de segmentation afin d'estimer les meilleures valeurs des paramètres respectifs aux différentes méthodes. La deuxième partie contient 30 images, utilisées comme images de test. Toutes les images sont des images réelles de taille 512×512 . La complexité des images varie d'images contenant un simple objet, aux images contenant plusieurs objets en occlusion visuelle. En nombre de régions, les images varient de 9 à 27 régions avec une moyenne de 15.23 et un écart-type de 5.15. Pour chacune des images, nous disposons des résultats, respectivement de la segmentation correspondant à la réalité terrain (Ground Truth : GT), et de ceux de chacune des méthodes utilisées dans la comparaison.

7.2.3 Les méthodes utilisées dans la comparaison

Nous avons utilisé les résultats des 4 méthodes citées dans Hoover *et al.* (1996) afin de comparer nos résultats à ceux produits par ces méthodes. Toutes les méthodes citées sont dédiées à la segmentation d'images de profondeur. Nous présentons un bref descriptif de chacune de ces méthodes.

USF de "University of South Florida, USA" : le principe de l'algorithme d'USF est basé sur l'estimation du plan tangent à la surface en chaque pixel de l'image. La méthode procède ensuite à un accroissement de régions par ajout de pixels ayant les mêmes plans tangents.

UB de "University of Bern, Switzerland" : l'algorithme est basé sur le principe d'accroissement de régions. Cependant, l'étiquetage de tout pixel est renforcé par l'hypothèse que les pixels appartenant aux segments de la ligne de scan appartiennent à la même surface. Les pixels sont regroupés en prenant en compte leur appartenance à la ligne de scan, et non plus considérés individuellement.

UE de "University of Edinburgh, Scotland" : suivant le même chemin que l'algorithme d'USF, en améliorant les résultats de l'accroissement de régions par un traitement d'amélioration spécifique, des pourtours des régions.

WSU de "Washington State University, USA" : l'algorithme procède à l'estimation des vecteurs normaux aux surfaces par une analyse en composantes principales. Les composantes de position et d'orientation sont regroupées et classifiées, en spécifiant le nombre maximal de classes (régions). Certaines régions adjacentes seront fusionnées, et ce, en fonction de leur degré de coplanarité.

7.2.4 Les métriques de performance

Nous utilisons les métriques de performance définies dans le framework introduit par Hoover *et al.* (1996) (voir section 7.3), afin d'évaluer les performances de notre système, et pour comparer nos résultats à ceux obtenus par les systèmes précédemment introduits. Les métriques, au nombre de 5, se calculent à partir de la comparaison des régions issues de la segmentation d'une image, aux régions de la segmentation manuelle correspondant à la réalité terrain (supposée idéale). Ces métriques sont :

- le nombre d'instances de détections correctes
- le nombre d'instances de sur-segmentations
- le nombre d'instances de sous-segmentations
- le nombre d'instances de régions omises
- le nombre d'instances de régions de bruit.

Un seuil de tolérance de comparaison de régions est utilisé. Les métriques précédentes sont calculées pour une série de valeurs de ce seuil, comprises entre 0.5 et 1.0. Afin d'assurer la fiabilité de la comparaison, les valeurs des métriques sont calculées par un outil software offert par les concepteurs du framework (Hoover *et al.* (1996)).

7.2.5 Test des hypothèses

Hypothèse 1

Dix exécutions sont effectuées pour un ensemble de 3 images, de complexité variable. Pour chacune de ces images, la métrique de détection correcte est calculée en variant le seuil de tolérance T de 0.5 à 0.95. Les valeurs obtenues sont utilisées pour calculer les moyennes et les écart-types de cette métrique.

Nous nous attendons à ce que les moyennes calculées soient représentatives de l'échantillon des 10 exécutions (écart-types faibles). En effet, nous nous attendons à ne pas trouver des valeurs aberrantes, concluant à la non représentativité des moyennes.

Hypothèse 2

Pour chacune des 30 images de la base d'images et pour chaque valeur du seuil de tolérance, la segmentation effectuée est comparée à la segmentation manuelle GT. Pour chaque segmentation, nous enregistrons les nombres respectivement ; de cas de détection correcte, de cas de sur-segmentation, de cas de sous-segmentation, de cas d'omission et de cas de bruit.

Nous calculons pour chaque segmentation manuelle, la moyenne, le minimum et le maximum des nombres de régions. Ceux-ci seront utilisés pour calculer les pourcentages de la moyenne du minimum et du maximum pour chaque valeur du seuil de tolérance T . Pour chacune des métriques, trois graphiques sont à tracer et à comparer : le premier correspond aux moyennes, le second aux minimums, et le troisième, aux maximums.

Hypothèse 3

La courbe de détection correcte sans régularisation est comparée aux courbes de détection correcte avec régularisation. Les courbes correspondant respectivement à la

régularisation bayésienne, et à la régularisation floue sont également comparées.

7.3 Un framework d'évaluation de performances et de comparaison

Un framework d'évaluation et de comparaison, dédié aux méthodes de segmentation d'images de profondeur, a été proposé par Hoover et al. dans Hoover *et al.* (1996) et est largement utilisé par les auteurs d'algorithmes dans le domaine (Jiang et Bunke (1999); Jiang *et al.* (2000); Li et Zhao (2003); Ding *et al.* (2005); Bab Hadiashar et Gheissari (2006)). Le framework consiste en une base d'images de profondeur, et un ensemble de métriques objectives de performance. Il permet de comparer une segmentation générée par une machine de segmentation (MS) à une segmentation manuelle, supposée idéale, et représentant la réalité terrain (GT). Les métriques de performance les plus importantes sont le nombre de régions correctement détectées, de régions sur-segmentées, de régions sous-segmentées, de régions omises et de régions de bruit. La classification de régions est effectuée en fonction d'un seuil de tolérance T ; $50\% < T \leq 100\%$, qui reflète la rigueur de la classification. Notez que le framework ne peut être utilisé que si les résultats de segmentation sont présentés sous forme de matrices d'étiquettes, où que les pixels appartenant à la même région auront la même étiquette.

7.3.1 Calcul des métriques de performance

Soit une machine de segmentation MS, segmentant une image donnée en M régions. Et soit GT, la réalité terrain, qui consiste à un ensemble de N régions, obtenu par une segmentation manuelle, supposée idéale, de la même image. Pour chaque région R_m des M régions de MS, et pour chaque région R_n des N régions de GT, les tailles en pixels, P_m , et P_n des régions sont calculées.

Soit O_{mn} , le nombre de pixels en communs entre les deux régions R_m et R_n , O_{mn} est le cardinal de l'ensemble $R_m \cap R_n$. Selon la valeur de O_{mn} , différents cas se présentent :

- $O_{mn} = 0$: les deux régions R_m et R_n sont totalement disjointes ;
- $O_{mn} = P_m = P_n$: les deux régions sont parfaitement superposées ;
- $O_{mn} > 0$; $O_{mn} < P_m$ ou $O_{mn} < P_n$: les deux régions R_m et R_n sont partiellement superposées.

L'appariement des régions de MS et de GT permet d'obtenir une matrice $[O_{mn}]$ où chaque élément O_{mn} représente le nombre de pixels en commun entre la région R_m de MS et la région R_n de GT (Fig. 7.1). La première ligne et la première colonne de la table 7.1, représentent les tailles respectivement des régions de GT et de MS. A l'intersection de la ligne m et la colonne n est stockée la valeur O_{mn} qui représente le nombre de pixels en commun entre les deux régions R_m et R_n .

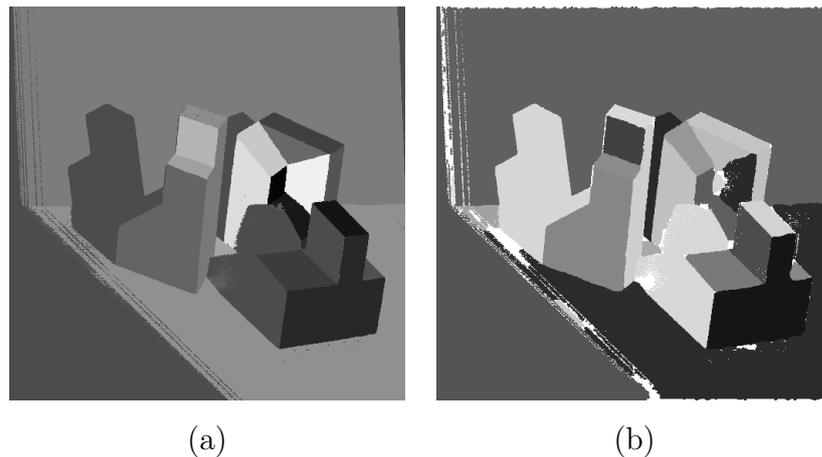


FIG. 7.1 – Appariement de segmentation GT (a) et de segmentation MS (b) de l'image abw.test.8.

Pour chaque instance (R_m, R_n) , deux valeurs sont calculées et utilisées pour décider de la classification de l'instance :

- O_{mn}/P_m : qui représente le pourcentage de couverture de $R_m \cap R_n$ dans R_m .
- O_{mn}/P_n : qui représente le pourcentage de couverture de $R_m \cap R_n$ dans R_n .

TAB. 7.1 – Matrice des correspondances O_{mn} de l'image de la figure 7.1

| | | GT | | | | | | | | | | | | | |
|----------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|-----|-----|-----|
| O_{mn} | | 85216 | 68586 | 13955 | 7836 | 5009 | 4175 | 4122 | 1939 | 1700 | 1660 | 1177 | 765 | 600 | 560 |
| | 81964 | 81950 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 52828 | 93 | 52631 | 79 | 0 | 15 | 0 | 2 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| | 23694 | 0 | 12445 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 13263 | 0 | 1 | 13211 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 48 | 0 | 0 |
| | 7554 | 0 | 0 | 0 | 7537 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 4556 | 0 | 2 | 0 | 0 | 4550 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| MS | 3918 | 0 | 21 | 0 | 0 | 55 | 42 | 3768 | 5 | 0 | 0 | 0 | 0 | 0 | 27 |
| | 3870 | 0 | 0 | 0 | 0 | 0 | 3868 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1735 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1735 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1519 | 0 | 0 | 0 | 0 | 0 | 7 | 0 | 0 | 1512 | 0 | 0 | 0 | 0 | 0 |
| | 1427 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 1417 | 0 | 0 | 0 | 0 |
| | 662 | 0 | 0 | 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 622 | 0 | 2 | 0 |
| | 535 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 522 | 0 |
| | 432 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 432 |

7.3.2 Instances de classification

En se basant sur la matrice $[O_{mn}; m = 1..M, n = 1..N]$, les auteurs du framework ont proposé 5 types d'instances de classification de régions entre MS et GT.

Instances de détection correcte

On note une instance de détection correcte, pour le couple de régions R_m de MS et R_n de GT si :

1. $O_{mn} \geq T \times P_m$;
2. $O_{mn} \geq T \times P_n$.

Où T est le seuil de tolérance de comparaison, pris entre 0.5 et 1.0. Ceci traduit le fait que les deux régions R_m et R_n sont superposées à un pourcentage supérieur ou égal

à T .

Instances de sur-segmentation

Une sur-segmentation est constatée lorsque k régions $\{R_i, i = 1 \dots k\}$ de MS, correspondent à une seule région R_n de GT, à un pourcentage T :

1. $\forall i = 1 \dots k, O_{in} \geq T \times P_i$;
2. $\sum_{i=1}^k O_{in} \geq T \times P_n$.

Instances de sous-segmentation

Ces instances correspondent aux cas où une région R_m de MS correspond au regroupement d'un sous ensemble de l régions $\{R_j, j = 1 \dots l\}$ de GT :

1. $\forall j = 1 \dots l, O_{mj} \geq T \times P_j$;
2. $\sum_{j=1}^l O_{mj} \geq T \times P_m$.

Instances d'omission

Une région R_n de GT est dite omise, si elle n'a pas participé à la classification d'aucune instance. Ceci signifie que la machine de segmentation n'a pas pu détecter cette région.

Instances de bruit

Une région R_m de MS est dite région de bruit, si elle n'a pas participé à aucune correspondance avec les régions de GT. Il s'agit d'une région détectée par la machine de segmentation, mais qui n'as pas de correspondant dans l'image réelle.

7.4 Apprentissage du système et sélection des paramètres

7.4.1 Les paramètres de segmentation

Pour notre système basé agent de segmentation d'images, que nous baptisons SIBA/2ARIS pour Segmentation d'Images à Base d'Agents/*Agent-based Approach for Range Image Segmentation*, six paramètres doivent être fixés : ξ_{min} , ξ_{max} , Tr_{θ} , Tr_D , N , et L . Cet ensemble de paramètres est divisé en deux sous-ensembles. ξ_{min} , ξ_{max} , Tr_{θ} , et Tr_D représentent respectivement les forces d'alignement et les seuils d'angle et de profondeur, et sont utilisés dans les opérations de test et d'alignement des pixels. Les paramètres N et L représentent respectivement le nombre d'agents, et la longueur d'adaptation. Ces deux paramètres influencent la dynamique des agents. Pour le premier sous-ensemble de paramètres, nous avons considéré 256 combinaisons à savoir $(\xi_{min}, \xi_{max}, Tr_{\theta} \text{ et } Tr_D) \in \{0.5, 0.3, 0.1, 0.05\} \times \{1.0, 3.0, 5.0, 7.0\} \times \{15^{\circ}, 18^{\circ}, 21^{\circ}, 24^{\circ}\} \times \{12, 16, 20, 24\}$, pour faire exécuter sur les images d'apprentissage. Le critère de performance choisi pour ces quatre paramètres est le nombre moyen de régions correctement détectées dans les images d'apprentissage, avec le seuil de tolérance $T=80\%$.

Suite à l'exécution des 256 combinaisons, les paramètres de segmentation ont été initialisés comme suit :

Les forces d'alignement ξ_{min} et ξ_{max} ont été initialisées respectivement à 0.3 et 5.0. Ces valeurs assurent une bonne localisation des pixels de contour en un temps d'exécution raisonnable.

Le seuil Tr_{θ} est initialisé à 21° . Nous avons constaté que des valeurs plus grandes de ce paramètre sous différencient les pixels selon leurs vecteurs normaux, et conduisent à une sous-segmentation. Cependant, des valeurs significativement inférieures à 21° sur différencient les pixels et conduisent à une sur-segmentation de l'image. Ceci se traduit par la génération d'un nombre élevé de fausses petites régions.

Finalement, le seuil Tr_D est initialisé à 16. Les valeurs significativement supérieures à 16 peuvent conduire à une fausse fusion de régions parallèles et proches. Cependant, si la valeur de ce paramètre est significativement inférieure à 16, les régions hautement inclinées ne peuvent pas être détectées comme régions planes (Jiang et Bunke (1999)). Ceci conduit à un taux élevé de régions omises.

7.4.2 Les paramètres de la dynamique du système

Il s'agit du nombre d'agents utilisés N , et de la longueur d'adaptation L qui sont critiques et doivent être minutieusement sélectionnés. Un mauvais choix de ces deux paramètres peut conduire à un taux élevé d'erreurs de segmentation. En effet, un nombre insuffisant d'agents, pour une taille donnée des images traitées (512×512 pixels), conduit à un sous-lissage de l'image et à un sous-amincissement des contours. Les régions qui en résultent sont amputées d'un nombre important de pixels qui auraient dû être inclus dans ces régions. Une faible valeur de la longueur d'adaptation L conduit à la prise en compte de petites régions planes qui devraient être considérées comme régions de bruit. Par contre, une valeur élevée de L peut conduire à l'omission de certaines vraies régions planes et qui sont relativement étroites (voir section 3.4.3). Pour fixer les paramètres N et L , 25 combinaisons de ces paramètres, soit $(N, L) \in \{1500, 2000, 2500, 3000, 3500\} \times \{3, 5, 7, 9, 11\}$ ont été exécutées sur les images d'apprentissage. Dans ce cas, le critère de performance choisi est le nombre moyen des régions de bruit, avec le seuil de tolérance $T=80\%$. Le calcul a abouti aux valeurs de $N=2500$ et $L=7$.

7.4.3 Les paramètres de la régularisation bayésienne

Il s'agit des deux paramètres ζ et κ qui représentent les potentiels des cliques, utilisés dans la régularisation bayésienne de la segmentation. Ces paramètres sont estimés par la méthode des "Codings" (Besag (1974)). Pour chaque image de l'ensemble d'apprentissage une paire des deux paramètres est calculée. Les deux moyennes sont retenues comme valeurs finales des deux paramètres. Nous avons utilisé un seul coding de $S - S'$ qui correspond aux cliques définies par le voisinage $c_{3 \times 3}$. En effet, nous considérons que

TAB. 7.2 – Régions correctement détectées en 10 exécutions (abw.test.6)

| Exec. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Moyenne | Ecarte- type |
|-------|----|----|----|----|----|----|----|----|----|----|---------|-----------------|
| 0,5 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14,00 | 0,00 |
| 0,6 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14,00 | 0,00 |
| 0,7 | 14 | 14 | 14 | 13 | 14 | 14 | 14 | 13 | 14 | 14 | 13,80 | 0,42 |
| 0,8 | 13 | 13 | 14 | 13 | 13 | 13 | 13 | 14 | 13 | 13 | 13,20 | 0,42 |
| 0,9 | 11 | 11 | 11 | 11 | 10 | 11 | 11 | 11 | 12 | 11 | 11,00 | 0,47 |
| 0,95 | 5 | 5 | 6 | 5 | 4 | 5 | 5 | 5 | 5 | 5 | 5,00 | 0,47 |

la régularisation des lignes de contours est plus importante pour l'étiquetage des pixels de contours, du fait que ces derniers sont proches des vrais bords de régions. L'optimum (ζ^*, κ^*) pour chaque image est calculé en utilisant l'algorithme du recuit simulé (Kirkpatrick *et al.* (1987)) avec un échantillonneur de Gibbs (Geman et Geman (1984)). Les valeurs de ζ et κ obtenues étaient respectivement -0.37×10^{-4} and -0.21×10^{-4} .

7.5 Analyse de la stabilité de la segmentation multi-agents

Du fait de la nature aléatoire des mouvements des agents, ainsi que leurs actions compétitives, qui sont dues à leur tour à leur distribution et leur autonomie, il est nécessaire de démontrer la stabilité du système, en terme de production des mêmes résultats lorsque l'exécution est répétée pour les mêmes images. Pour ce faire, nous avons considéré 3 images de différentes complexités et ayant des nombres de régions variables (Figure 7.2).

Pour chacune des trois images, nous avons enregistré les résultats de segmentation, en terme de détection correcte, de 10 exécutions. Les tableaux et les graphiques suivants montrent les résultats de détection correcte des 10 exécutions respectives à chacune des images.

TAB. 7.3 – Régions correctement détectées en 10 exécutions (abw.test.10)

| Exec. T | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Moyenne | Ecarte- type |
|------------|---|---|---|---|---|---|---|---|---|----|---------|-----------------|
| 0,5 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8,00 | 0,00 |
| 0,6 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8,00 | 0,00 |
| 0,7 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8,00 | 0,00 |
| 0,8 | 8 | 8 | 8 | 8 | 8 | 8 | 7 | 8 | 8 | 8 | 7,90 | 0,32 |
| 0,9 | 6 | 6 | 7 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6,10 | 0,32 |
| 0,95 | 3 | 3 | 4 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3,00 | 0,47 |

TAB. 7.4 – Régions correctement détectées en 10 exécutions (abw.test.23)

| Exec. T | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Moyenne | Ecarte- type |
|------------|----|----|----|----|----|----|----|----|----|----|---------|-----------------|
| 0,5 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15,00 | 0,00 |
| 0,6 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15,00 | 0,00 |
| 0,7 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15,00 | 0,00 |
| 0,8 | 15 | 15 | 15 | 15 | 15 | 15 | 14 | 15 | 15 | 15 | 14,90 | 0,32 |
| 0,9 | 11 | 12 | 11 | 11 | 11 | 11 | 10 | 11 | 11 | 11 | 11,00 | 0,47 |
| 0,95 | 8 | 9 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8,10 | 0,32 |

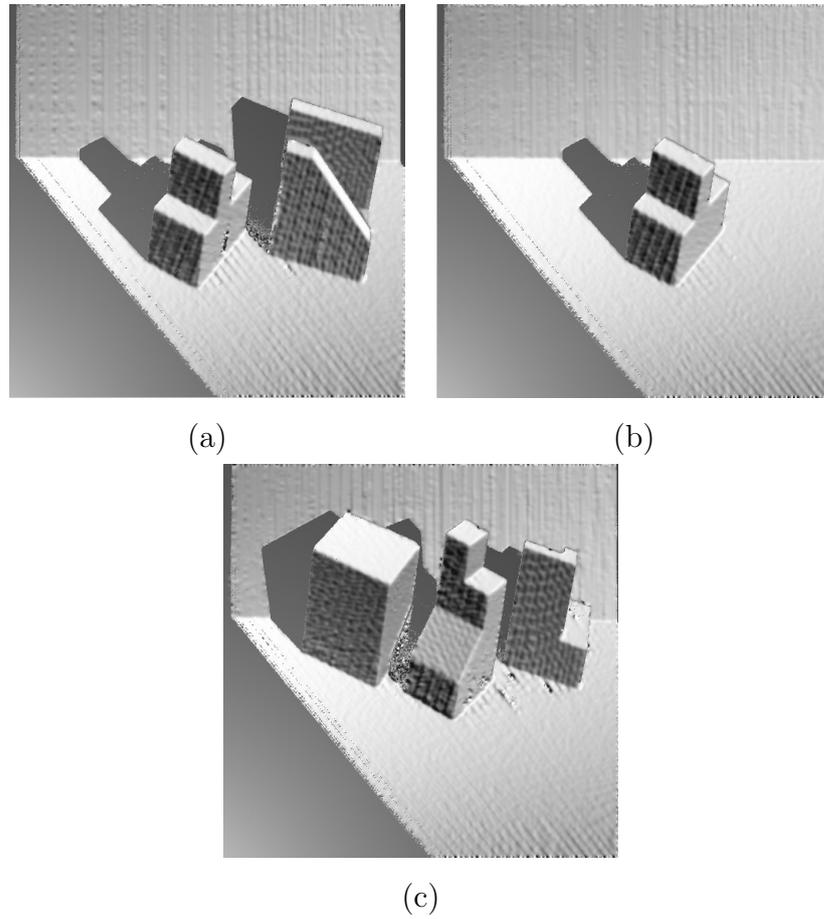


FIG. 7.2 – Images utilisées pour le test de stabilité du système. (a) `abw.test.6`; (b) `abw.test.10`; (c) `abw.test.23`

Selon les moyennes et les écart-types calculés, le système reproduit statistiquement le même résultat après chaque exécution. En effet, on constate que les écart-types correspondant aux différentes valeurs de la tolérance T sont faibles comparés aux moyennes. Ce qui signifie que les nombres de régions correctement détectées sont regroupés étroitement autour de leurs moyennes respectives.

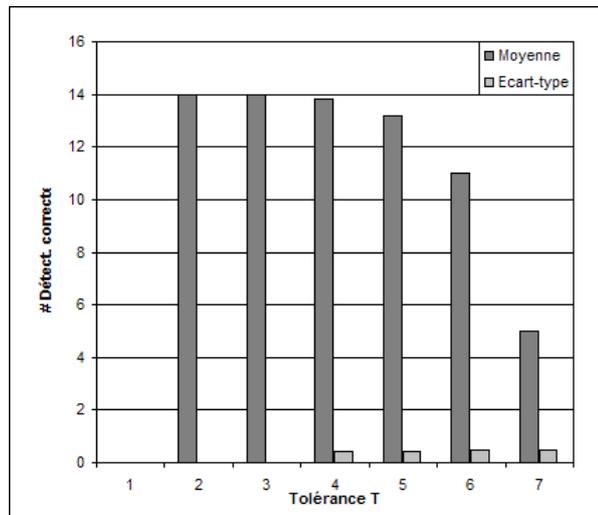


FIG. 7.3 – Moyennes et écart-types des nombres de régions correctement détectées en 10 exécutions (image abw.test.6)

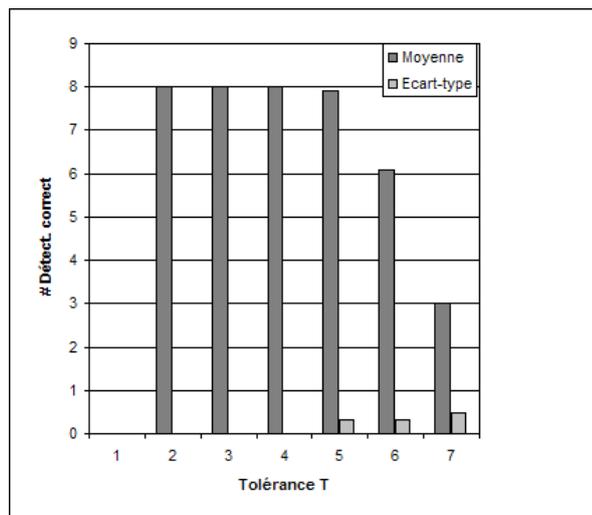


FIG. 7.4 – Moyennes et écart-types des nombres de régions correctement détectées en 10 exécutions (image abw.test.10)

7.6 Résultats de segmentation obtenus par le système SIBA

La figure 7.6 montre un exemple de l'évolution du processus de segmentation dans le temps. Le temps t représente le nombre de pas effectués par chaque agent depuis le début du processus. L'affichage d'image de profondeur par un simple algorithme de

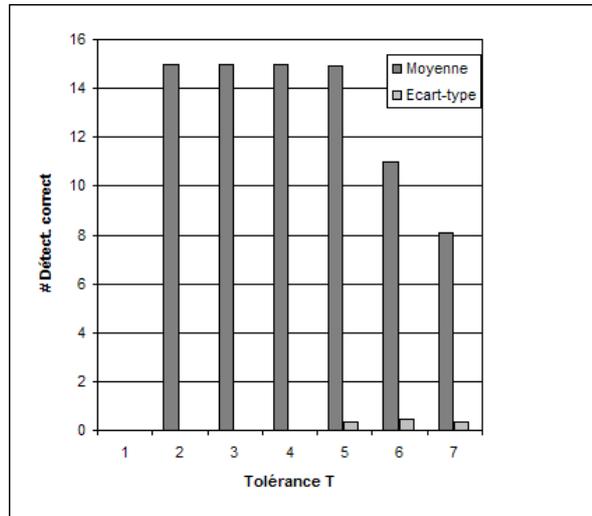


FIG. 7.5 – Moyennes et écart-types des nombres de régions correctement détectées en 10 exécutions (image abw.test.10)

rendu réaliste (figure 7.6a), permet de constater le niveau élevé de bruit dans les images utilisées. Les figures 7.6b, 7.6c, 7.6d et 7.6e montrent l'ensemble des pixels d'intérêt (pixels de contours ou de bruit) respectivement à $t=1000$, 5000, 9000 et 13000. Les régions sont progressivement lissées par l'alignement des pixels de bruits qui sont à l'intérieur de ces régions, aux régions environnantes. Les contours entre les régions adjacentes sont progressivement amincis. A la fin, ces contours sont formés de lignes minces de 1 pixel de largeur (figure 7.6e).

La figure 7.7 montre les résultats de segmentation de l'image abw.test.8 obtenus par les différentes méthodes, avec le seuil de tolérance $T=80\%$. Cette image était prise comme une image typique pour comparer les méthodes impliquées (Hoover *et al.* (1996); Ding *et al.* (2005)). Les figures 7.7a, 7.7b et 7.7c montrent respectivement l'image de profondeur, l'image rendue réaliste, et la segmentation manuelle (GT).

Les figures 7.7d, 7.7e, 7.7f et 7.7g montrent respectivement les résultats de segmentation des méthodes USF, WSU, UB, et UE, tandis que la figure 7.7h présente le résultat de segmentation obtenu par le système proposé. Les métriques correspondantes dans la table 7.5 montrent que toutes les régions de l'image détectées par la meilleure méthode (UE) ont été détectées par notre méthode. A l'exception de la région ombrée qui n'a été détectée par aucune des méthodes, toutes les régions d'objets ont été bien détectées.

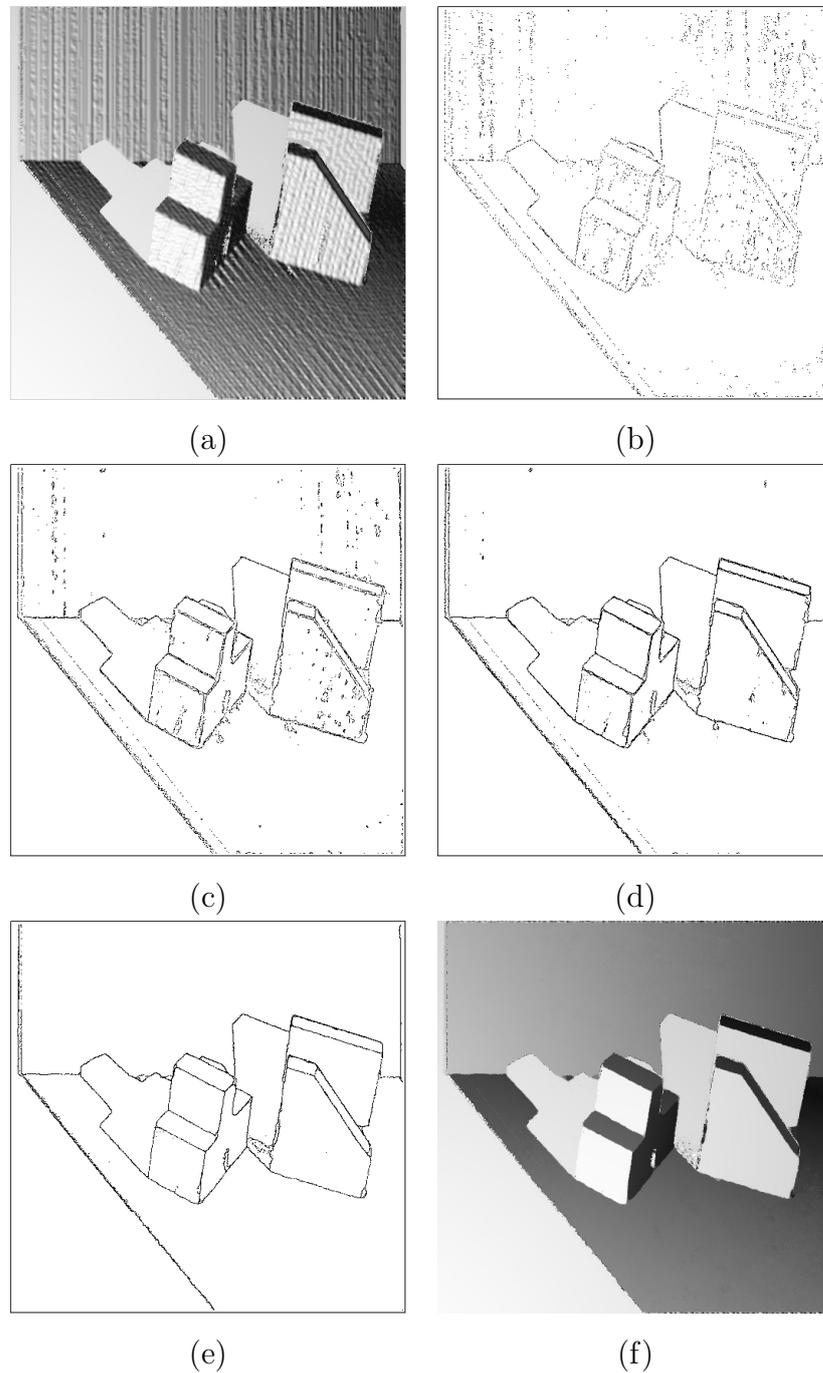


FIG. 7.6 – Progression de segmentation. (a) Image de profondeur rendue réaliste (abw.test.6); (b) à $t=1000$, (c) à $t=5000$; (d) à $t=9000$; (e) à $t=13000$; (f) Image reconstruite rendue réaliste

Les régions incorrectement détectées dans cette image sont dues à une sur-segmentation de la région plane représentant le support de la scène. Comparées aux autres méthodes, les différentes valeurs des métriques de détection incorrecte sont excellentes avec cette

image.

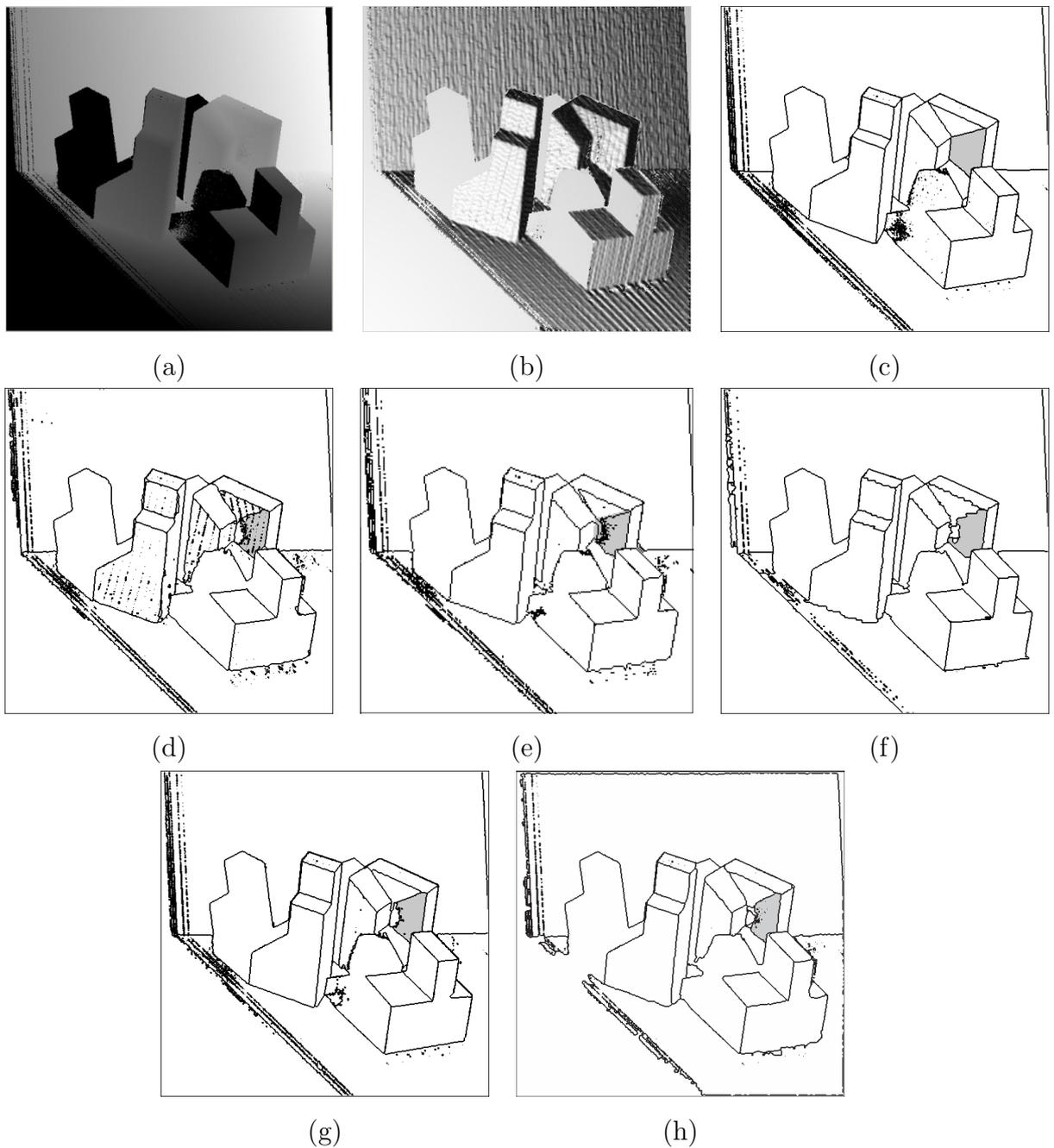


FIG. 7.7 – Résultats de segmentation de l'image *abw.test.8*. (a) Image de profondeur ; (b) Image de profondeur rendue réaliste ; (c) Segmentation selon la Réalité terrain (GT) ; (d) Résultat de USF ; (e) Résultat de WSU ; (f) Résultat de UB ; (g) Résultat de UE ; (h) Résultats de SIBA (Régions extraites)

La table 7.6 présente les résultats moyens obtenus avec toutes les images de test et pour toutes les métriques de performance. Le seuil de tolérance étant fixé à la valeur

TAB. 7.5 – Comparaison des résultats de segmentation de l'image abw.test.8 avec le seuil de tolérance $T=80\%$

| Méthode | Régions GT | Détection Correcte | Sur- segmentation | Sous- segmentation | Omise | Bruit |
|---------|---------------|-----------------------|----------------------|-----------------------|-------|-------|
| USF | 21 | 17 | 0 | 0 | 4 | 3 |
| WSU | 21 | 12 | 1 | 1 | 6 | 4 |
| UB | 21 | 16 | 2 | 0 | 3 | 6 |
| UE | 21 | 18 | 1 | 0 | 2 | 2 |
| SIBA | 21 | 18 | 2 | 0 | 1 | 1 |

TAB. 7.6 – Résultats moyens des cinq méthodes au seuil de tolérance $T=80\%$

| Méthode | Régions GT | Détection Correcte | Sur- segmentation | Sous- segmentation | Omise | Bruit |
|---------|---------------|-----------------------|----------------------|-----------------------|-------|-------|
| USF | 15.2 | 12.7 | 0.2 | 0.1 | 2.1 | 1.2 |
| WSU | 15.2 | 9.7 | 0.5 | 0.2 | 4.5 | 2.2 |
| UB | 15.2 | 12.8 | 0.5 | 0.1 | 1.7 | 2.1 |
| UE | 15.2 | 13.4 | 0.4 | 0.2 | 1.1 | 0.8 |
| SIBA | 15.2 | 13.0 | 0.5 | 0.1 | 1.7 | 0.9 |

typique 80%. La figure 7.8 montre les moyennes des nombres de régions correctement détectées pour toutes les images de test aux différentes valeurs du seuil de tolérance T ; $T \in \{51\%, 60\%, 70\%, 80\%, 90\%, 95\%\}$. Les résultats obtenus montrent que le nombre de régions correctement détectées par le système proposé est en moyenne meilleur que celui des méthodes USF, UB et WSU. Par exemple, notre méthode enregistre un score meilleur que celui de WSU pour toutes les valeurs du seuil de tolérance T . Elle enregistre un score meilleur que celui de la méthode USF pour le seuil de tolérance $T \in \{80\%, 90\%, 95\%\}$, et meilleur que celui de UB pour $T \in \{51\%, 60\%, 70\%, 80\%\}$. Pour toutes les métriques de détection incorrecte, notre méthode a enregistré des scores équivalents à ceux de UE et USF, qui enregistrent des scores meilleurs que ceux des méthodes UB et WSU (figure 7.9).

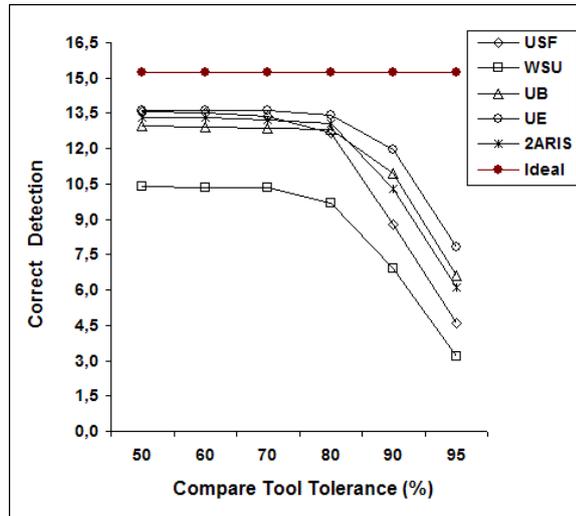


FIG. 7.8 – Résultats moyens des régions correctement détectées de toutes les méthodes au différentes valeurs du seuil de tolérance T ; $0.5 < T \leq 1.0$

7.7 Résultats après régularisation

La segmentation par le système SIBA fourni de meilleurs résultats comparés à ceux obtenus par un simple algorithme d'accroissement de régions (Fig. 7.10c,d). Cependant, la segmentation résultante reste souvent insatisfaisante. Dans la figure 7.10c, nous pouvons noter que la plupart des bords ne sont pas bien lissés localement, parce que les pixels sur les bords ne sont pas assignés d'une façon optimale aux régions respectives.

La figure 7.11 illustre l'impact de la régularisation des pixels appartenant aux bords des régions sur les résultats de segmentation. Nous avons comparé pour chacune des méthodes proposées de régularisation les résultats obtenus directement avec le système SIBA, et ceux obtenus après régularisations respectivement par la méthode bayésienne et la méthode floue.

L'écart entre les deux graphes des figures 7.11a,b correspondant respectivement à la méthode bayésienne et la méthode floue montre que les résultats de segmentation ont été significativement améliorés. Il est à noter que l'amélioration a concerné spécifiquement les résultats correspondant aux valeurs élevées du seuil de tolérance de comparaison T . En effet, la régularisation de bord de régions dans les images de profondeur permet d'améliorer l'exactitude de la segmentation par un assignement optimal des pixels

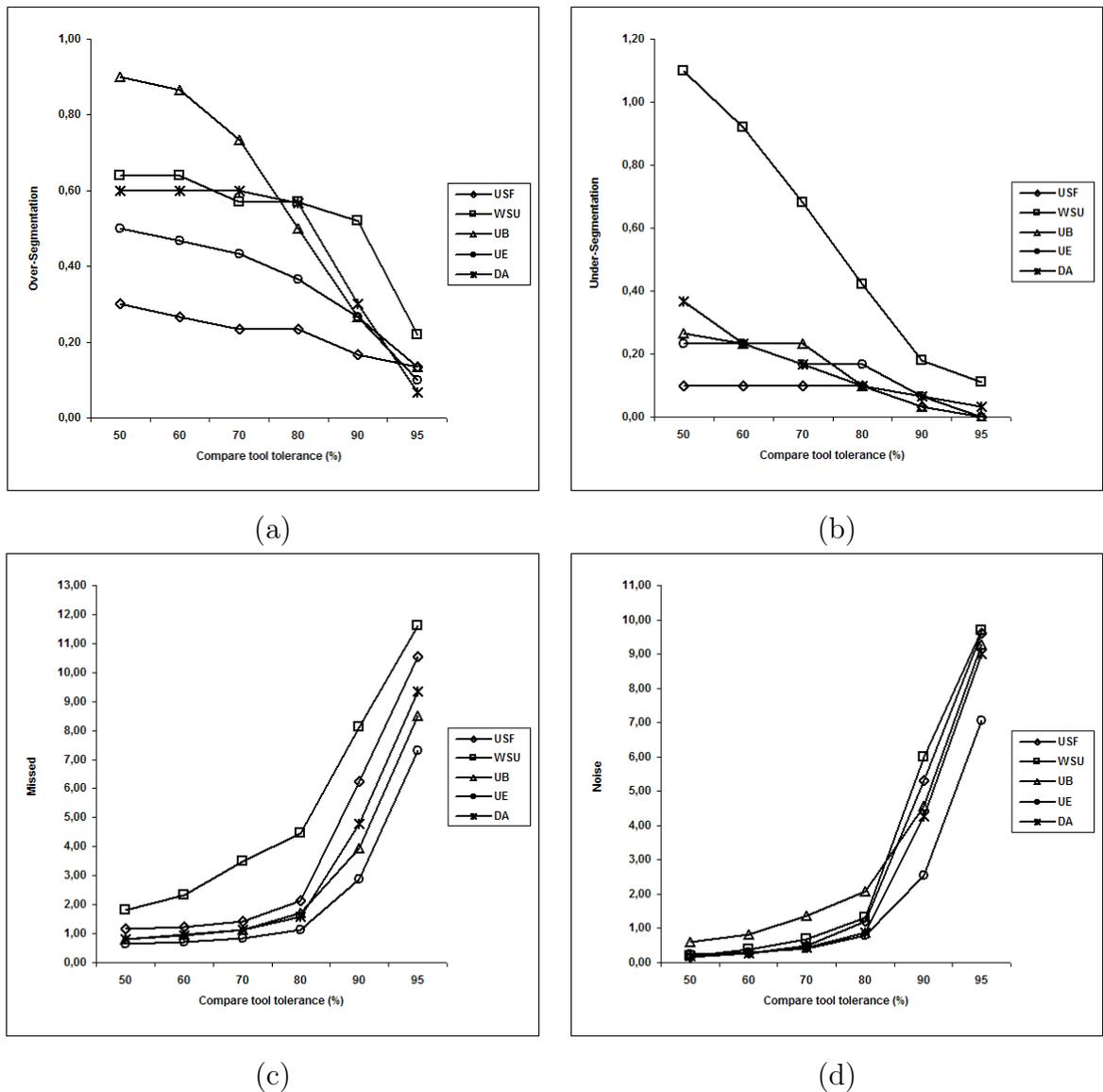


FIG. 7.9 – Résultats moyens des métriques de détection incorrecte : (a) Sur-segmentation ; (b) Sous-segmentation ; (c) Regions omises ; (d) Regions de bruit appartenant aux bordures des régions.

Il a été reporté par plusieurs auteurs, tels que dans Hoover *et al.* (1996) et Jiang *et al.* (2000) qu’une méthode de segmentation d’images de profondeur fourni des résultats meilleurs lorsqu’elle prête une attention particulière aux traitements des bordures de régions ; ce que nous avons pu confirmer par les méthodes de régularisation des contours dans ce type d’images.

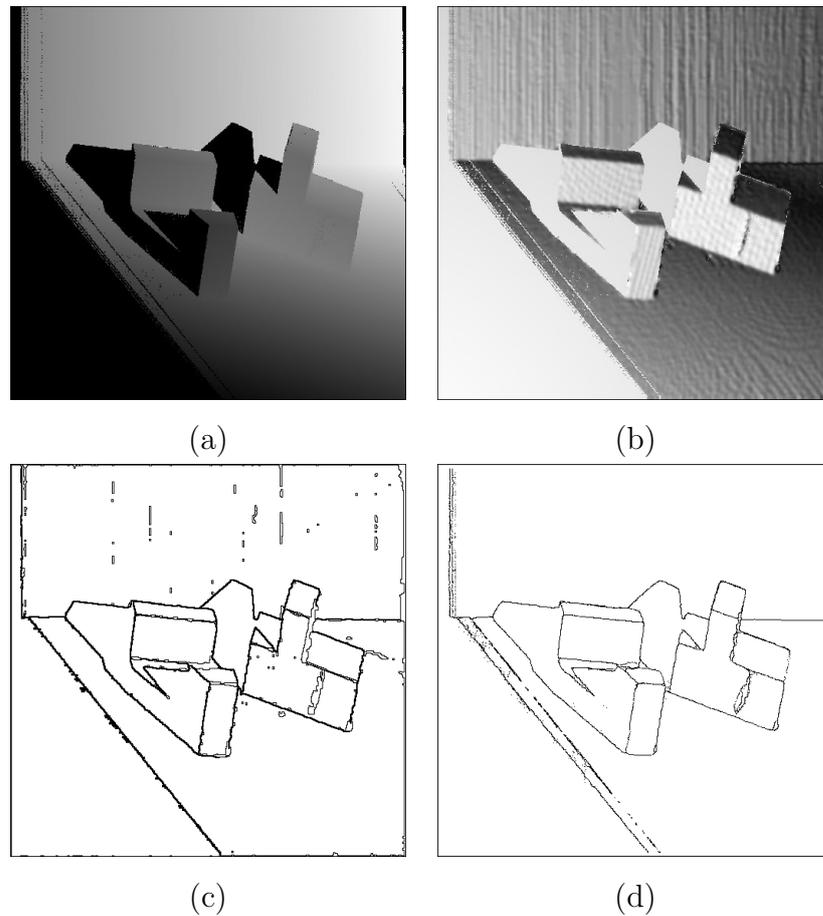


FIG. 7.10 – Un exemple de résultats après régularisation. (a) Image de profondeur ; (b) Image rendu réaliste ; (c) Segmentation par simple accroissement de régions ; (d) Segmentation par le système SIBA (avec régularisation bayésienne)

Selon les résultats obtenus, il est à noter qu'il y a pas de différence significative entre la régularisation bayésienne et la régularisation floue dans les images de profondeur. En effet, le problème l'incertitude dans l'étiquetage des pixels de contours, dont l'approche bayésienne est mieux appropriée pour y remédier, peut être expliqué par le problème de l'imprécision des mesures effectuées lors de l'acquisition et du traitement des images, dont l'approche floue est la plus appropriée, dans ce cas, pour y remédier.

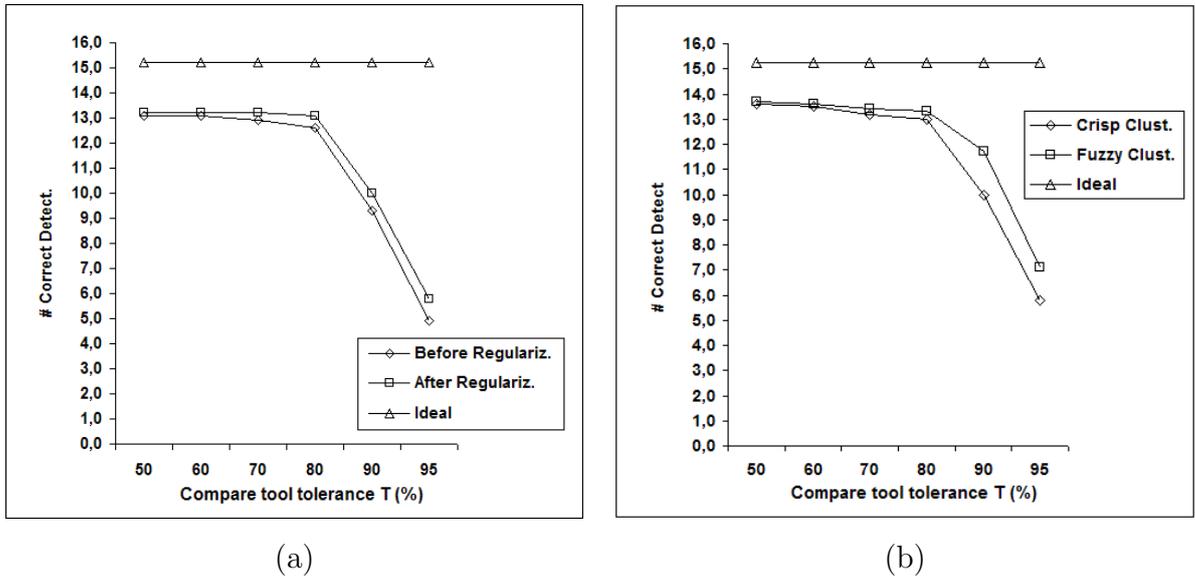


FIG. 7.11 – Impact de la régularisation sur les résultats de segmentation. (a) Régularisation bayésienne; (b) Régularisation floue.

7.8 Conclusion

Le long de ce chapitre, nous avons présenté une évaluation qui nous a permis de valider nos approches de segmentation d'images de profondeur, et de régularisation de contours, proposées dans le cadre de cette thèse. Afin de pouvoir quantifier l'apport de nos contributions et pouvoir comparer nos résultats à ceux publiés dans la littérature, nous avons fait recours à un framework dédié à l'évaluation et la comparaison de méthodes de segmentation d'images de profondeur. L'objectivité des métriques de performance utilisées dans ce framework, nous a permis de bien quantifier les résultats de nos approches, mais également de situer leurs avantages et leurs insuffisances. En effet, dès les premières expérimentations effectuées sur la segmentation multi-agents, nous avons constaté, à l'aide des métriques d'évaluation, une carence des résultats malgré leur aspect visuel qui était satisfaisant. Ceci nous a conduit à chercher l'origine de cette carence et dont la conclusion consistait à l'affectation arbitraire des pixels de bordures des régions, ce qui nous a poussé à développer des méthodes d'amélioration, basées sur la régularisation des contours.

Chapitre 8

Conclusion générale

Dans cette thèse nous nous sommes intéressés à l'utilisation des systèmes auto-organisés pour le traitement d'images et la reconnaissance des formes. Nous nous sommes penchés sur les systèmes multi-agents, dont nous avons proposé deux approches multi-agents, respectivement pour la segmentation d'images, et pour la reconnaissance d'objets. L'approche de segmentation multi-agents d'images consiste en l'utilisation d'une population d'agents réactifs pour le lissage d'image et la détection des contours. Contrairement aux méthodes traditionnelles de segmentation d'images, les résultats de détection dans notre cas, émergent des interactions locales des agents, notamment la compétition des agents sur les contours de régions. Une nouvelle méthode de coordination d'agents, basée sur un champ de potentiel artificiel, était également introduite. L'originalité de cette dernière résulte dans le fait que le champ de potentiel est créé et est mis à jour, en fonction des interactions des agents avec les objets de l'environnement, qui sont les pixels d'intérêt dans le cas de traitement d'images.

La segmentation d'images de profondeur, est une tâche complexe, qui demeure non résolue d'une manière définitive. Les résultats expérimentaux obtenus, en utilisant un benchmark dédié, ont montré un fort potentiel de l'approche proposée pour une segmentation efficace et exacte des images de profondeur. Néanmoins, l'approche proposée reste générique et peut être adaptée à tous les types d'images, où des contours et des régions peuvent être définis. Pour ce faire, il suffit de modéliser les régions qui définissent les images, et d'inculquer le modèle obtenu aux agents, afin qu'ils puissent s'adapter

aux régions homogènes et détecter les pixels qui n'appartiennent pas à ces régions.

L'utilisation de la nouvelle approche de segmentation avec les images de profondeur a révélé un inconvénient qui consiste en la déformation des pourtours des régions, qui est due à la délocalisation des pixels de contours. En effet, ce phénomène est dû à l'autonomie des agents et à leurs actions asynchrones et distribuées, dont aucun contrôle central n'est utilisé. Dans le but de proposer des solutions à ce problème, nous avons consacré une partie de notre travail à la régularisation des contours dans les images de profondeur. Dans cette thématique, nous avons proposé de nouvelles méthodes de correction de contours, basées respectivement sur la régularisation bayésienne, et la régularisation floue. Contrairement aux méthodes bayésiennes publiées dans la littérature, qui utilisent exclusivement le smoothness de la surface comme à priori pour la segmentation d'images ; nous utilisons également le smoothness des contours de régions comme nouvel à priori dont, pour le représenter, nous avons introduit un nouveau modèle du champ aléatoire de markov.

En ce qui concerne la régularisation floue, nous avons proposé une nouvelle version de l'algorithme du FCM pour la classification floue des pixels de contours, en considérant une segmentation initiale, obtenue par le système multi-agents. Les classes sont donc définies à l'avance ; cependant, leurs prototypes et leurs frontières sont mis à jour par l'algorithme proposé. Le résultat d'exécution consiste à une localisation optimale aux sens de la régularisation floue des pixels de contours. Les résultats de la régularisation des contours, selon les deux approches ont montré que la segmentation était significativement améliorée. Nous avons proposé également un modèle de réseau de neurones chaotiques pour la détection de contours dans les images de profondeur, en considérant que les réseaux de neurones représentent une classe des systèmes auto-organisés.

L'aspect haut niveau en vision par ordinateur était également abordé dans le cadre de cette thèse. En effet, nous avons proposé une architecture coopérative d'agents autonomes pour la reconnaissance d'objets à partir des images de profondeur. Nous avons introduit le principe d'entrelacement de la détection et la reconstruction des primitives géométriques, et de l'appariement objet-modèle. L'entrelacement de la détection et de l'appariement permet de prédire certaines primitives géométriques et ce en fonction du résultat courant du processus de reconnaissance d'objets.

Dans le cadre de cette thèse, plusieurs concepts relevant des systèmes auto-organisés étaient revus, étendus et utilisés dans le but de proposer de nouvelles approches pour le traitement d'images et la reconnaissance d'objets. Les systèmes multi-agents, la régularisation basée sur l'inférence bayésienne, et la régularisation floue, représentaient les principaux axes autour desquels ce travail s'est déroulé.

Les expérimentations intensives, en utilisant des images réelles, ont montré l'efficacité des méthodes proposées pour une interprétation fiable des images de profondeur. Cependant, plusieurs extensions du présent travail, peuvent être envisagées. En effet, en terme d'architecture des agents, il serait plus pertinent l'utilisation d'agents génériques indépendants du type des images traitées. Dans ce cas, les connaissances opératoires des agents leur seront dynamiquement inculquées en fonction du type d'images considéré. Dans cette optique, il est envisageable d'interfacer les systèmes multi-agents aux ontologies de connaissances, spécifiques aux domaines d'application.

Concernant la régularisation des contours, et plus particulièrement la régularisation bayésienne, il reste possible la conception de modèles du champ aléatoire de Markov plus généraux, qui prennent en compte les surfaces et contours complexes, et qui peuvent être appliqués aux différents types d'images.

Bibliographie

- ARMAN, F. et AGGARWAL, J. K. (1993). Model-based object recognition in dense-range images—a review. *ACM Comput. Surv.*, 25(1):5–43.
- BAB HADIASHAR, A. et GHEISSARI, N. (2006). Range image segmentation using surface selection criterion. *IEEE Transactions on Image Processing*, 15(7):2006–2018.
- BALCH, T. et ARKIN, R. C. (1994). Communication in reactive multiagent robotic systems. *Auton. Robots*, 1(1):27–52.
- BARALDI, A. et BLONDA, P. (1999). A survey of fuzzy clustering algorithms for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 29(6):778–785.
- BENLAMRI, R., MAZOUZI, S. et BATOUCHE, M. (1994). An efficient method for recognising polyhedral scenes using rangedata. *Dans IEEE International Conference on Systems, Man, and Cybernetics, 1994*, volume 3, pages 2278–2283, San Antonio, USA.
- BENLAMRI, R., MAZOUZI, S. et BOULCINA, Y. (1993). Object recognition for polyhedral scenes using range data. *Dans Stochastic and Statistical Methods in Hydrology and Environmental Engineering*, Toronto, Canada.
- BESAG, J. (1974). Spatial interaction and statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36:192–236.
- BESAG, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48:259–302.
- BEZDEK, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

- BOVENKAMP, E. G. P., DIJKSTRA, J., BOSCH, J. G. et REIBER, J. H. C. (2004). Multi-agent segmentation of IVUS images. *Pattern Recognition*, 37(4):647–663.
- BURGISS, S. G., LESTER, E. D., WHITAKER, R. T. et ABIDI, M. A. (1998). Scene segmentation from vector-valued images using anisotropic diffusion. *Dans 17th Intelligent Robots and Computer Vision*, volume 3522, pages 527–538, Boston, USA.
- CARABELEA, C., BOISSIER, O. et FLOREA, A. (2003). Autonomy in multi-agent systems : A classification attempt. *Dans Agents and Computational Autonomy*, pages 103–113.
- CHEN, Y. W. et CHEN, Y. Q. (2005). An evolutionary system and its application to automatic image segmentation. *Dans International Conference on Natural Computation (2)*, pages 705–709.
- COLLIOT, O., CAMARA, O. et BLOCH, I. (2006). Integration of fuzzy spatial relations in deformable models-application to brain mri segmentation. *Pattern Recognition*, 39(8):1401–1414.
- DING, Y., PING, X., HU, M. et WANG, D. (2005). Range image segmentation based on randomized hough transform. *Pattern Recognition Letters*, 26(13):2033–2041.
- DROGOUL, A. (1993). When ants play chess (or can strategies emerge from tactical behaviours?). *Dans Modelling Autonomous Agents in a Multi-Agent World*, pages 13–27.
- DROGOUL, A. (1994). *De la simulation multi-agents à la résolution collective de problèmes*. Thèse de doctorat, Université Paris VI, France.
- FAN, T., MEDIONI, G. et NEVATIA, R. (1987). Segmented description of 3-D surfaces. *IEEE Journal of Robotics and Automation*, 3(6):527–538.
- FERBER, J. (1995). *Les systèmes multi-agents : vers une intelligence collective*. Informatique, Intelligence Artificielle. InterÉditions.
- FERBER, J. (1999). *Multi-Agent Systems : An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- FLACHER, F. (2005). *Génération Ascendante de Coordination Spatiale. Vers une conception automatisée du contrôle de coordination spatiale*. Thèse de doctorat, LIP6 - université Paris 6, France.

- FRANKLIN, S. et GRAESSER, A. C. (1996). Is it an agent, or just a program ? : A taxonomy for autonomous agents. *Dans Agent Theories, Architectures, and Languages (ATAL)*, pages 21–35.
- FREEMAN, W. et SKARDA, C. (1987). How brains make chaos in order to make sense of the word. *Behav. Brain*, 10:161–195.
- FRIGUI, H. et KRISHNAPURAM, R. (1996). Surface approximation and range image segmentation through robustcompetitive clustering. *Dans Fifth IEEE International Conference on Fuzzy Systems*, pages 1414–1419, New Orleans, LA, USA.
- GASSER, L. et KAKUGAWA, K. (2002). Mace3j : fast flexible distributed simulation of large, large-grain multi-agent systems. *Dans AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 745–752, New York, NY, USA. ACM Press.
- GEMAN, S. et GEMAN, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.
- GENESERETH, M. R. et KETCHPEL, S. P. (1994). Software agents. *Commun. ACM*, 37(7):48–ff.
- GHOSH, S., KOTHARI, M. et GHOSH, A. (2006). Aggregation pheromone density based image segmentation. *Dans Indian Conference on Computer Vision, Graphics and Image Processing*, pages 118–127.
- GRAY, C. M., KÖNING, P., ENGEL, A. et SINGER, W. (1989). Oscillatory response in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 333:334–337.
- GROSSBERG, S. et SOMERS, D. (1990). Synchronized oscillations during cooperative feature linking in a cortical model of visual perception. *Neural networks*, 4:7200–7204.
- GUESSOUM, Z. (2000). A multi-agent simulation framework. *Transactions of the Society for Computer Simulation International*, 17(1):2–11.
- GUTKNECHT, O., FERBER, J. et MICHEL, F. (2001). Integrating tools and infrastructures for generic multi-agent systems. *Dans AGENTS '01 : Proceedings of the fifth*

international conference on Autonomous agents, pages 441–448, New York, NY, USA. ACM Press.

HAN, S. K., KIM, W. S. et KOOK, H. (1998). Temporal segmentation of the stochastic oscillator neural network. *Phys. Rev. E*, 58:2325–2334.

HARALICK, R. et SHAPIRO, S. (1985). Survey, image segmentation technics. *Dans computer graphics and images processing*.

HAROUN, R., BOUMGHAR, F., HASSAS, S. et HAMAMI, L. (2004). A massive multi-agent system for brain mri segmentation. *Dans MMAS*, pages 174–186.

HARROUET, F., TISSEAU, J., REIGNIER, P. et CHEVAILLIER, P. (2002). oRis : un environnement de simulation interactive multi-agents. *Technique et Science Informatiques*, 21(4):499–524.

HOOVER, A., JEAN-BAPTISTE, G., JIANG, X., FLYNN, P. J., BUNKE, H., GOLDFOF, D. B., BOWYER, K. W., EGGERT, D. W., FITZGIBBON, A. W. et FISHER, R. B. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689.

HORAUD, R. P. et MONGA, O. (1995). *Vision par ordinateur : outils fondamentaux*. Editions Hermès, Paris, France.

INOKUCHI, S., NITA, T., MATSUDA, F. et SAKURAI, Y. (1982). A three dimensional edge-region operator for range pictures. *Dans 6th International Conference on Pattern Recognition*, pages 918–920, Munich.

JAIN, A. et NADABAR, S. (1990). MRF model-based segmentation of range images. *Dans International Conference on Computer Vision*, pages 667–671, Osaka, Japan.

JIANG, X., BOWYER, K. W., MORIOKA, Y., HIURA, S., SATO, K., INOKUCHI, S., BOCK, M., GUERRA, C., LOKE, R. E. et du BUF, J. M. H. (2000). Some further results of experimental comparison of range image segmentation algorithms. *Dans 15th International Conference on Pattern Recognition*, volume 4, pages 4877–4882, Barcelona, Spain.

JIANG, X. et BUNKE, H. (1999). Edge detection in range images based on Scan Line approximation. *Computer Vision and Image Understanding*, 73(2):183–199.

- KANG, S. et IKEUCHI, K. (1993). The complex EGI : A new representation for 3-D pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):707–721.
- KHATIB, M. (1996). *Sensor-based motion control for mobile robots*. Thèse de doctorat, LAAS-CNRS December, 1996.
- KIRKPATRICK, S., GELATT, C. D. et VECCHI, M. P. (1987). Optimization by simulated annealing. *Readings in Computer Vision : Issues, Problems, Principles, and Paradigms*, pages 606–615.
- KOREN, Y. et BORENSTEIN, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. *Dans IEEE International Conference on Robotics and Automation*, pages 1398–1404.
- KRISHNAMURTHY, E. V. et MURTHY, V. K. (2006). Distributed agent paradigm for soft and hard computation. *Journal of Network and Computer Applications*, 29(2):124–146.
- LABROU, Y. (1996). *Semantics for an agent communication language*. Thèse de doctorat. Director-Timothy Finin.
- LAVALLE, S. M. et HUTCHINSON, S. A. (1993). Bayesian region merging probability for parametric image models. *Dans IEEE Conference on Computer Vision and Pattern Recognition*, pages 778–779, New York, USA.
- LI, S. et ZHAO, D. (2003). Gradient-based polyhedral segmentation for range images. *Pattern Recognition Letters*, 24(12):2069–2077.
- LI, S. Z. (2000). Roof-edge preserving image smoothing based on MRFs. *IEEE Transactions on Image Processing*, 9(6):1134–1138.
- LI, S. Z. (2001). *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- LIU, J. et TANG, Y. Y. (1999). Adaptive image segmentation with distributed behavior-based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):544–551.

- MAZOUZI, S. et BATOUCHE, M. (2007a). A new bayesian method for range image segmentation. *Dans EMMCVPR, LNCS*, China.
- MAZOUZI, S. et BATOUCHE, M. (2007b). Range image segmentation by randomized region growing and bayesian edge regularization. *Journal of Computer Science*, 3(5): 410–417.
- MAZOUZI, S. et BATOUCHE, M. (2007c). Range image segmentation improvement by fuzzy edge regularization. *Journal of Information Technology*, 6(8):1316–1322.
- MAZOUZI, S., BATOUCHE, M. et GUESSOUM, Z. (2004). A self-adaptive multi-agent system for segmentation and reconstruction of 3d scenes. *Dans AISTA 2004*, Kirchberg, Luxembourg.
- MAZOUZI, S., BATOUCHE, M. et GUESSOUM, Z. (2005a). Un système multi-agents pour la segmentation et la reconstruction de surfaces complexes. *Dans SETIT 2005*, Sousse, Tunisie.
- MAZOUZI, S., GUESSOUM, Z. et BATOUCHE, M. (2005b). A self-adaptive multi-agent system for 3d scene recognition. *Dans AAMAS 2005*, Paris, France.
- MAZOUZI, S., GUESSOUM, Z., BATOUCHE, M. et MICHEL, F. (2006). A potential field-based multi-agent model for contour detection in range images. *Dans IEEE ICCP 2006*, Romania.
- MAZOUZI, S., GUESSOUM, Z., MICHEL, F. et BATOUCHE, M. (2007a). A multi-agent approach for range image segmentation. *Dans CEEMAS, LNCS*, Leibzig, Germany.
- MAZOUZI, S., GUESSOUM, Z., MICHEL, F. et BATOUCHE, M. (2007b). A multi-agent approach for range image segmentation with bayesian edge regularization. *Dans ACIVS, LNCS*, Delf, Netherland.
- MEHRTASH, A. (2006). Fuzzy edge preserving smoothing filter using robust region growing. *IEEE International Conference on Fuzzy Systems*, pages 1748–1755.
- MINAR, N., BURKHART, R., LANGTON, C. et ASKENAZI, M. (Santa Fe Institute, 1996). The swarm simulation system, a toolkit for building multi-agent simulations.
- NEWELL, A. (1990). *Unified theories of cognition*. Harvard University Press, Cambridge, MA, USA.

- NOE, A. et GEE, J. C. (2001). Partial volume segmentation of cerebral mri scans with mixture model clustering. *Dans IPMI '01 : Proceedings of the 17th International Conference on Information Processing in Medical Imaging*, pages 423–430, London, UK. Springer-Verlag.
- OUADFEL, S. et BATOUCHE, M. (2002). Ant colony system for image segmentation. *Dans ANTS 2002*, LNCS Conferences, pages 294–295, Brussels.
- REMAGNINO, P., TAN, T. et BAKER, K. (1998). Multi-agent visual surveillance of dynamic scenes. *Image Vision Comput.*, 16(8):529–532.
- 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.
- RODIN, V., BENZINO, A., GUILLAUD, A., BALLE, P., HARROUET, F., TISSEAU, J. et BIHAN, J. L. (2004). An immune oriented multi-agent system for biological image processing. *Pattern Recognition*, 37(4):631–645.
- RUSSELL, S. J. (1997). Rationality and intelligence. *Artificial Intelligence*, 94(1-2):57–77.
- SAPPA, A. (2006). Unsupervised contour closure algorithm for range image edge-based segmentation. *IEEE Transactions on Image Processing*, 15(2):377–384.
- SHEN, S., SANDHAM, W., GRANAT, M. et STERR, A. (2005). Mri fuzzy segmentation of brain tissue using neighborhood attraction with neural-network optimization. *IEEE Transactions on Information Technology in Biomedicine*, 9(3):459–467.
- SHIN, J., SUN, Y., PAIK, J. et ABIDI, M. A. (2001). Adaptive regularized noise smoothing of dense range image using directional laplacian operators. *Dans SPIE*, volume 4298, pages 119–126.
- SHOHAM, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92.
- SIMONIN, O. (2001). *Le modèle satisfaction-altruisme : coopération et résolution de conflits entre agents situés réactifs, application à la robotique*. Thèse de doctorat, LIRMM-UM2 - Montpellier, France.

- SIMONIN, O. (2005). Construction of numerical potential fields with reactive agents. *Dans AAMAS '05 : Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1351–1352, New York, NY, USA. ACM Press.
- SINHA, S. et BAZAK, J. (1998). Response of an excitatory-inhibitory neural network to external stimulation : Application to object-background segregation. *Dans Proc. of MUT*, Calcutta, India.
- SOODAMANI, R. et LIU, Z.-Q. (2000). Ga-based learning for a model-based object recognition system. *International Journal of Approximate Reasoning*, 23(2):85–109.
- SUN, Y., PAIK, J. K., PRICE, J. R. et ABIDI, M. A. (2000). Dense range image smoothing using adaptive regularization. *Dans International Conference on Image Processing*.
- TSUJI, T., TANAKA, Y., MORASSO, P., SANGUINETI, V. et KANEKO, M. (2002). Biomimetic trajectory generation of robots via artificial potential field with time base generator. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 32(4):426–439.
- WANG, X. et WANG, H. (2004). Markov random field modeled range image segmentation. *Pattern Recognition Letters*, 25(3):367–375.
- WOOLDRIDGE, M. (2002). *Introduction to MultiAgent Systems*. John Wiley & Sons.
- WOOLDRIDGE, M. et JENNINGS, N. R. (1995). Intelligent agents : Theory and practice. 10(2):115–152.
- XUE, J.-H., RUAN, S., MORETTI, B., REVENU, M., BLOYET, D. et PHILIPS, W. (2000). Fuzzy modeling of knowledge for mri brain structure segmentation. *Dans International Conference on Image Processing*, volume 1, pages 617–620, British Columbia, Canada.
- ZADEH, L. (1965). Fuzzy sets. *Information and Control*, 8:338–353.
- ZAMBONELLI, F., JENNINGS, N. R. et WOOLDRIDGE, M. (2003). Developing multiagent systems : The gaia methodology. *ACM Trans. Softw. Eng. Methodol.*, 12(3):317–370.
- ZEGHAL, K. et FERBER, J. (1993). A reactive approach for distributed air traffic control. *Dans Avignon'93*, pages 381–390.