



uOttawa

L'Université canadienne
Canada's university

**FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTDOCTORALES**



**FACULTY OF GRADUATE AND
POSTDOCTORAL STUDIES**

Bassel Abou Merhy

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc.A. (Génie Électrique)

GRADE / DEGREE

Département de Génie Électrique

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Segmentation par Analyse de Texture de Grilles d'Occupation Probabilistes

TITRE DE LA THÈSE / TITLE OF THESIS

P. Payeur

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

E. Petriu

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

W. Gueuib

R. Goubran

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Segmentation par Analyse de Texture de Grilles d'Occupation Probabilistes

Thèse rédigée par :

Bassel Abou Merhy

Soumise à la :

Faculté des Études Supérieures et Postdoctorales

Comme exigence partielle en but d'obtenir une :

Maîtrise en Sciences Appliquées en Génie Électrique

Institut de Génie Électrique et de Génie Informatique d'Ottawa-Carleton
École d'Ingénierie et de Technologie de l'Information
Faculté de Génie
Université d'Ottawa

©Bassel Abou Merhy, Ottawa, Canada, 2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-25735-7
Our file *Notre référence*
ISBN: 978-0-494-25735-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Table des matières

<u>TABLE DES MATIÈRES</u>	<u>II</u>
<u>TABLE DES FIGURES</u>	<u>V</u>
<u>TABLE DES TABLEAUX</u>	<u>XI</u>
<u>TABLE DES ABRÉVIATIONS</u>	<u>XII</u>
<u>RÉSUMÉ</u>	<u>XIII</u>
<u>REMERCIEMENTS</u>	<u>XV</u>
<u>CHAPITRE 1 INTRODUCTION</u>	<u>1</u>
1.1 OBJECTIFS	3
1.2 CONTRIBUTIONS	4
1.3 ORGANISATION	5
<u>CHAPITRE 2 REVUE DE LA LITTÉRATURE</u>	<u>6</u>
2.1 INTRODUCTION	6
2.2 GRILLES D'OCCUPATION PROBABILISTES	6
2.2.1 INTRODUCTION ET AVANTAGES	7
2.2.2 GRILLES D'OCCUPATION TELLES QU'INTRODUITES PAR ELFES	8
2.2.3 MODÉLISATION D'ENVIRONNEMENTS TRIDIMENSIONNELS	10
2.2.4 PHYSIQUE DE LA CONSTRUCTION DES IMAGES PROBABILISTES	10
2.2.4.1 Images de profondeur et capteur Jupiter	11
2.2.4.2 Programme de simulation en deux dimensions	13
2.3 PARTICULARITÉ DES TEXTURES	15
2.3.1 RÉGIONS EXPLORÉES NON-UNIFORMES	15
2.3.2 PRÉSENCE DE TEXTURES ALÉATOIRES	19
2.3.3 ANALYSE DE LA DISTANCE ENTRE LES FAISCEAUX ADJACENTS	21
2.4 REVUE DES APPROCHES DE LA SEGMENTATION	23
2.4.1 SEUILLAGE	24
2.4.1.1 Principes de base	24
2.4.1.2 Applicabilité sur la segmentation d'images probabilistes	27
2.4.2 CONTOURS	30

2.4.2.1	Principes de base	30
2.4.2.2	Applicabilité sur la segmentation d'images probabilistes	32
2.4.3	LES STRUCTURES PYRAMIDALES CHAÎNÉES	33
2.4.3.1	Méthodes de segmentation	34
2.4.3.2	Applicabilité sur la segmentation d'images probabilistes	37
2.4.4	LA DIVISION ET FUSION	38
2.4.4.1	Méthodes de segmentation	38
2.4.4.2	Applicabilité sur la segmentation d'images probabilistes	41
2.4.5	LES FILTRES DE GABOR MULTICANAUX	41
2.4.5.1	Choix du filtre de Gabor et de ses paramètres	42
2.4.5.2	Extraction des images caractéristiques	43
2.4.5.3	Applicabilité sur la segmentation d'images probabilistes	45
2.4.6	LA TRANSFORMÉE EN ONDELETTES	46
2.4.6.1	Les méthodes de segmentation	46
2.4.6.2	Applicabilité sur la segmentation d'images probabilistes	49
2.4.7	L'ANALYSE DE LA TEXTURE LOCALE	49
2.4.7.1	Caractérisation de la texture	50
2.4.7.2	Comparaisons des distributions LBP/C – le G-statistique	57
2.4.7.3	Algorithme de segmentation	59
2.4.7.4	Applicabilité sur la segmentation d'images probabilistes	63
2.5	CONCLUSION	68
CHAPITRE 3 SEGMENTATION PROBABILISTE BIDIMENSIONNELLE		69
3.1	INTRODUCTION	69
3.2	TENTATIVE DE RAFFINEMENT DE L'ALGORITHME D'OJALA ET PIETIKÄINEN	69
3.3	UN ALGORITHME DE SEGMENTATION LBP/C RÉVISÉ, AMÉLIORÉ ET OPTIMISÉ	75
3.3.1	DIVISION HIÉRARCHIQUE	76
3.3.2	CRÉATION DES SEGMENTS	78
3.3.3	RAFFINEMENT	83
3.4	CONCLUSION	87
CHAPITRE 4 ÉTUDE EXPÉRIMENTALE DE LA TECHNIQUE DE SEGMENTATION		88
4.1	INTRODUCTION	88
4.2	ÉTUDE COMPARATIVE DES MÉTHODES LBP/C ORIGINALE ET RÉVISÉE	88
4.3	MESURES D'ERREUR POUR ÉVALUER LE RÉSULTAT DE LA SEGMENTATION	94
4.4	EFFETS DE LA VARIATION DE L'ÉCART-TYPE ET DU PAS ANGULAIRE DES CAPTEURS	96
4.5	EFFETS DE LA VARIATION DE L'EXPLORATION DE L'ESPACE	106

4.6	EFFETS DE LA VARIATION DES PARAMÈTRES B ET S_{MIN}	113
4.7	CONCLUSION	120
<u>CHAPITRE 5</u> <u>SEGMENTATION PROBABILISTE TRIDIMENSIONNELLE</u>		122
5.1	INTRODUCTION	122
5.2	CONSTRUCTION DES ENVIRONNEMENTS PROBABILISTES TRIDIMENSIONNELS	122
5.3	L'ALGORITHME DE SEGMENTATION TRIDIMENSIONNEL	125
5.3.1	CARACTÉRISTIQUES DE LA TEXTURE TRIDIMENSIONNELLE	125
5.3.2	DIVISION HIÉRARCHIQUE	128
5.3.3	CRÉATION DES SEGMENTS	131
5.3.4	RAFFINEMENT	135
5.4	RÉSULTATS ET ANALYSE	136
5.4.1	DÉFINITION DES MODÈLES TRIDIMENSIONNELS	136
5.4.2	ÉVALUATION DES PERFORMANCES	141
5.5	CONCLUSION	148
<u>CHAPITRE 6</u> <u>EXTENSIONS ET APPLICATIONS POSSIBLES</u>		150
6.1	INTRODUCTION	150
6.2	SEGMENTATION PAR PALIERS	150
6.3	EXTRACTION DES RÉGIONS SELON LEUR PROBABILITÉ D'OCCUPATION	152
6.4	PLANIFICATION DE TRAJECTOIRE APPLIQUÉE SUR LE LBP/C RÉVISÉ	154
6.5	AUTRES APPLICATIONS	157
6.6	CONCLUSION	163
<u>CHAPITRE 7</u> <u>CONCLUSION</u>		165
7.1	RÉSUMÉ	165
7.2	CONTRIBUTIONS	166
7.3	TRAVAUX FUTURS	167
<u>RÉFÉRENCES</u>		169
<u>ANNEXE A</u> <u>SIMULATEUR DE CAPTEUR TÉLÉMÉTRIQUE</u>		175
<u>ANNEXE B</u> <u>INTERFACE USAGER DE LA SEGMENTATION BIDIMENSIONNELLE</u>		177
<u>ANNEXE C</u> <u>MODULES DE VISUALISATION 3D ET INTERFACE USAGER</u>		179

Table des figures

<i>Figure 2-1 : construction de la grille d'occupation à l'aide des mesures de profondeur. (a) et (b) interprétation d'une lecture r à l'aide d'un modèle de capteur probabiliste, (c) mise à jour de la probabilité d'occupation des proxels, (d) modèle déterministe issu de l'application de l'estimateur MAP (reproduite de [4])</i>	8
<i>Figure 2-2 : mises à jour séquentielles et intégration de grilles issues de plusieurs capteurs</i>	9
<i>Figure 2-3 : les images de profondeur fournissent une représentation 2.5D d'une réalité tridimensionnelle</i>	11
<i>Figure 2-4 : capteur Jupiter monté sur un bras robotique</i>	12
<i>Figure 2-5 : champ de vision du capteur Jupiter (reproduite de [10])</i>	13
<i>Figure 2-6 : (a) exemple d'un environnement encombré où les espaces libre et occupé sont respectivement en niveaux noir et blanc, (b) image probabiliste incorporant huit lectures de capteurs à partir de points de vue différents</i>	14
<i>Figure 2-7 : lectures dans des environnements non-encombrés avec des pas angulaires de (a) 4, (b) 2 et (c) 0.5 degrés respectivement [noir = faisceau, gris = espace non balayé]</i>	16
<i>Figure 2-8 : lectures dans un environnement encombré avec des écarts-type sur la mesure de la profondeur de (a) 1, (b) 3 et (c) 7 cm respectivement [noir = libre, blanc = occupé, gris = inconnu]</i>	18
<i>Figure 2-9 : images probabilistes incorporant respectivement (a) 2, (b) 3, (c) 4 et (d) 5 points de vue avec un pas angulaire de 3 degrés et un écart-type sur la mesure de la profondeur de 3 cm</i>	19
<i>Figure 2-10 : images qui illustrent le fait qu'une région de croisement entre un nombre fixe de points de vue contient des textures différentes : (a) et (b) contiennent respectivement deux et trois lectures avec des pas angulaires de 4 et 3 degrés</i>	20
<i>Figure 2-11 : situations particulières qui peuvent entraver et induire en erreur le processus de segmentation : (a) objet non-identifié entre deux faisceaux, (b) faisceaux adjacents couvrant deux objets distincts</i>	22
<i>Figure 2-12 : calcul de la distance maximale entre deux faisceaux adjacents en fonction du pas angulaire et de la profondeur atteinte par les rayons</i>	22
<i>Figure 2-13 : (a) image en noir et blanc incluant un seul objet, (b) histogramme de distribution de l'intensité des pixels de (a)</i>	25
<i>Figure 2-14 : (a) image incluant plusieurs objets circulaires caractérisés par des intensités différentes, (b) histogramme de distribution de l'intensité des pixels de (a)</i>	26
<i>Figure 2-15 : (a) image en noir et blanc incluant deux objets circulaires, (b) histogramme de distribution de l'intensité des pixels de (a)</i>	26
<i>Figure 2-16 : cas où la distribution des intensités dans chaque segment correspond à une distribution gaussienne (reproduite de [9])</i>	27
<i>Figure 2-17 : image probabiliste générée par la combinaison de données issues de quatre points de vue</i>	29
<i>Figure 2-18 : histogramme répartissant les pixels de la Figure 2-17 suivant leur intensité</i>	29
<i>Figure 2-19 : (a) segmentation de la Figure 2-17 en utilisant les seuils présentés dans la Figure 2-18, (b) zoom sur la région rectangulaire et caractéristiques du résultat</i>	30
<i>Figure 2-20 : un contour idéal est la liaison entre les arêtes d'un objet</i>	31
<i>Figure 2-21 : interpolation entre des éléments de contour dans le but d'obtenir un contour clos</i>	32
<i>Figure 2-22 : image probabiliste générée par la combinaison de données issues de trois points de vue</i>	33
<i>Figure 2-23 : application de la méthode de Canny [11] de détection d'arêtes sur la Figure 2-22</i>	33
<i>Figure 2-24 : organisation de la pyramide dans les « structures pyramidales chaînées »</i>	35
<i>Figure 2-25 : « enfants » et « parents » candidats pour un nœud $(i; j; l)$ donné</i>	36
<i>Figure 2-26 : diagramme en bloc de la « division » proposée par Yang et Lee [18] (reproduite de [18])</i>	40
<i>Figure 2-27 : diagramme en bloc caractérisant la segmentation multi-canal</i>	43
<i>Figure 2-28 : processus multi-étapes analyse-synthèse ou décomposition-reconstruction (reproduite de [33])</i>	47

Figure 2-29 : exemple de calcul des caractéristiques de la texture, le LBP et le C. (a) région de taille [3 x 3] de l'image, (b) seuillage, (c) poids binomiaux, (d) résultat du processus de discrétisation.....	50
Figure 2-30 : exemple de l'application du processus de discrétisation sur la valeur du contraste	52
Figure 2-31 : image probabiliste générée par combinaison de données issues de trois points de vue	53
Figure 2-32 : histogramme LBP/C correspondant à l'image probabiliste de la Figure 2-31.....	54
Figure 2-33 : exemple d'images sur lesquelles Mäenpää et al. [40] ont testé leur algorithme de classification des textures (reproduite de [42])	56
Figure 2-34 : calcul du LBPROT pour une unité de texture donnée. (a) valeurs binaires obtenues après seuillage, (b) mots de huit bits correspondants, (c) rotation du mot jusqu'à correspondance dans la table de possibilités, (d) l'indice du mot correspond au LBPROT.....	57
Figure 2-35 : modèle de subdivision dans la phase « division hiérarchique »	60
Figure 2-36 : illustration du processus de classification des pixels décrit dans [39].....	62
Figure 2-37 : (a) image probabiliste incorporant 2 points de vue, (b) résultat de la division hiérarchique montrant des subdivisions de taille minimale de [16 x 16] pixels.....	64
Figure 2-38 : résultat de la phase de fusion agglomérée appliquée sur l'image probabiliste originale représentée dans la Figure 2-37-a	65
Figure 2-39 : (a) et (d) images probabilistes de taille initiale [320 x 320], (b) et (e) résultats de la division hiérarchique, (c) et (d) résultats de la fusion agglomérée.....	66
Figure 3-1 : résultat de la division hiérarchique sur trois images avec un seuil X de subdivision égal à 2.1	70
Figure 3-2 : (a) image probabiliste, (b) résultats de la division hiérarchique, (c) et (d) sorties de la première et de la seconde étape de la phase de fusion raffinée, (e) résultat de la classification des pixels raffinée.....	72
Figure 3-3 : (a) image probabiliste, (b) résultats de la division hiérarchique, (c) et (d) sorties de la première et de la seconde étape de la phase de fusion raffinée, (e) résultat de la classification des pixels raffinée.....	73
Figure 3-4 : (a) image probabiliste, (b) résultats de la division hiérarchique, (c) et (d) sorties de la première et de la seconde étape de la phase de fusion raffinée, (e) résultat de la classification des pixels raffinée.....	73
Figure 3-5 : position des quatre subdivisions à l'intérieur de la région initiale.....	78
Figure 3-6 : localisation d'un voisin V ((a) et (b)) de droite et ((c) et (d)) du bas de la région R.....	80
Figure 3-7 : processus de construction des segments durant la phase de création des segments. (a) les subdivisions occupées recherchent les voisins de même occupation à droite et au dessous. (b) le regroupement des cellules occupées adjacents permet d'identifier en segments séparés les objets présents dans l'image probabiliste	82
Figure 3-8 : assignation de différentes couleurs aux segments occupés non adjacents. (a) image probabiliste avec deux objets, un circulaire et un autre rectangulaire, (b) résultat de la segmentation où les segments sont représentés par des couleurs différentes : libre (noir), inconnu (gris), objet circulaire (bleu) et objet rectangulaire (vert).....	83
Figure 3-9 : processus de reclassification des pixels dans la phase de raffinement entre les segments libre – occupé d'une part (à gauche) et occupé – libre de l'autre (à droite)	86
Figure 4-1 : (a) image extraite de l'album de Brodatz [42] sur laquelle Ojala et Pietikäinen [39] ont testé leur algorithme. (b), (c) et (d) résultats des trois phases de l'algorithme proposé.....	92
Figure 4-2 : (a) image extraite de l'album de Brodatz [42] sur laquelle Ojala et Pietikäinen [39] ont testé leur algorithme. (b), (c) et (d) résultats des trois phases de l'algorithme proposé.....	93
Figure 4-3 : résultats de la première phase: (a) de l'algorithme proposé, (b) du schéma d'Ojala et Pietikäinen [39] sur une image image probabiliste, jusqu'à ce point les deux schémas ne diffèrent que de la subdivision supplémentaire que nous opérons.....	94
Figure 4-4 : résultats de la première phase: (a) de l'algorithme proposé, (b) du schéma d'Ojala et Pietikäinen [39] sur une image extraite de l'album de Brodatz [42], jusqu'à ce point les deux schémas ne diffèrent que de la subdivision supplémentaire que nous opérons.....	94
Figure 4-5 : environnements simulés de taille [448 x 448] utilisés dans la génération des grilles d'occupation et des résultats du Tableau 4-2, du Tableau 4-3, du Tableau 4-4 et du Tableau 4-6. (a) Surface en U, (b) Carré et disque, (c) Forme complexe.....	99

Figure 4-6 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	101
Figure 4-7 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 5 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	101
Figure 4-8 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 7 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	102
Figure 4-9 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 9 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	102
Figure 4-10 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-b avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu et vert)	103
Figure 4-11 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-b avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 1 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu et vert)	103
Figure 4-12 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-b avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 1.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu et vert)	104
Figure 4-13 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-c avec un écart-type sur la mesure de la distance de 7 cm et un pas angulaire de 1.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	105
Figure 4-14 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-c avec un écart-type sur la mesure de la distance de 9 cm et un pas angulaire de 1.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	106
Figure 4-15 : environnements simulés utilisés dans la génération des grilles d'occupations et des résultats du Tableau 4-5 et du Tableau 4-6. (a) Prise électrique, (b) Ovale, (c) Lunettes	108
Figure 4-16 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-a et incorporant deux points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	109
Figure 4-17 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-a et incorporant trois points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	109
Figure 4-18 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-b et incorporant trois points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	110
Figure 4-19 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-b et incorporant cinq points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	110

Figure 4-20 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-a et incorporant huit points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	111
Figure 4-21 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-c et incorporant deux points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	112
Figure 4-22 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-c et incorporant cinq points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	112
Figure 4-23 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-c et incorporant huit points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	112
Figure 4-24 : environnements simulés de taille [448 x 448] utilisés dans la génération des grilles d'occupation et des résultats du Tableau 4-6. (a) Tasse, (b) Lentille divergente.....	115
Figure 4-25 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	117
Figure 4-26 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-b incorporant trois points de vue. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	117
Figure 4-27 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-a. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	118
Figure 4-28 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-a. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments inconnu sont représentés par des couleurs différentes : (gris), libre (noir) et occupé (bleu)	118
Figure 4-29 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-b. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)	119
Figure 4-30 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-b. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments inconnu sont représentés par des couleurs différentes : (gris), libre (noir) et occupé (bleu)	119
Figure 4-31 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-b. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu). La lecture supplémentaire résoud le problème de l'incohérence dans la classification de certaines régions sur la bordure de deux régions d'occupation connue.....	120
Figure 5-1 : environnement tridimensionnel de taille [M x N x 6] obtenu en superposant six images de dimension [M x N]	123
Figure 5-2 : représentation volumétrique de la superposition de plusieurs surfaces balayées par des lectures de capteur.....	124
Figure 5-3 : unité de texture cubique de taille [3 x 3 x 3].....	126
Figure 5-4 : représentation 3D de l'unité de texture adoptée.....	126
Figure 5-5 : exemple de calcul des caractéristiques de la texture tridimensionnelle.....	127
Figure 5-6 : représentation d'une région tridimensionnelle et de ses huit subdivisions.....	129
Figure 5-7 : identification et coordonnées des subdivisions issues d'une région parent.....	129
Figure 5-8 : représentation 2D d'une unité de texture 3D.....	130
Figure 5-9 : localisation d'un voisin v au-dessus d'une région r donnée.....	133
Figure 5-10 : section horizontale des environnements réel et probabiliste de taille [320 x 320 x 64] incluant un objet conique.....	137

Figure 5-11 : sections horizontales des environnements réel et probabiliste de taille [320 x 320 x 64] incluant un objet cubique. (a) et (b) les 7 premières et les 7 dernières couches horizontales de l'environnement simulé et probabiliste, (c) et (d) les 50 couches intermédiaires de l'environnement simulé et probabiliste.....	138
Figure 5-12 : section horizontale des environnements réel et probabiliste de taille [320 x 320 x 64] incluant un objet cylindrique.....	138
Figure 5-13 : section horizontale des environnements réel et probabiliste de taille [320 x 320 x 128] incluant un objet conique.....	139
Figure 5-14 : sections horizontales des environnements réel et probabiliste de taille [320 x 320 x 128] incluant un objet cubique. (a) et (b) les 14 premières et 14 dernières couches horizontales de l'environnement simulé et probabiliste, (c) et (d) les 100 couches intermédiaires de l'environnement simulé et probabiliste.....	140
Figure 5-15 : section horizontale des environnements réel et probabiliste de taille [320 x 320 x 128] incluant un objet cylindrique.....	140
Figure 5-16 : vue d'en haut sur le segment libre obtenu à la fin de la 3 ^{ième} phase, seule la pointe de l'objet cône est visible au niveau de la couche libre du dessus (environnement de taille [320 x 320 x 64] incorporant un objet conique).....	144
Figure 5-17 : vue de côté (a) et du haut (b) sur le segment occupé obtenu à la fin de la 3 ^{ième} phase (environnement de taille [320 x 320 x 64] incorporant un objet conique).....	144
Figure 5-18 : vue de côté sur le segment occupé obtenu à la fin de la 2 ^{ième} phase (environnement de taille [320 x 320 x 64] incorporant un objet conique).....	145
Figure 5-19 : vue d'en haut sur le segment libre obtenu à la fin de la 3 ^{ième} phase, l'emplacement occupé par l'objet cubique est invisible étant donné qu'il est entouré au dessus et au dessous de sept couches de cellules libres (environnement de taille [320 x 320 x 64] incorporant un objet cubique).....	145
Figure 5-20 : vue de côté (a) et du haut (b) sur le segment occupé obtenu à la fin de la 3 ^{ième} phase (environnement de taille [320 x 320 x 64] incorporant un objet cubique).....	145
Figure 5-21 : vue d'en haut sur le segment occupé obtenu à la fin de la 2 ^{ième} phase (environnement de taille [320 x 320 x 64] incorporant un objet cubique).....	146
Figure 5-22 : vue d'en haut sur le segment libre obtenu à la fin de la 3 ^{ième} phase (environnement de taille [320 x 320 x 64] incorporant un objet cylindrique).....	146
Figure 5-23 : vue de côté (a) et du haut (b) sur le segment occupé obtenu à la fin de la 3 ^{ième} phase (environnement de taille [320 x 320 x 64] incorporant un objet cylindrique).....	146
Figure 5-24 : vue d'en haut sur le segment occupé obtenu à la fin de la 2 ^{ième} phase (environnement de taille [320 x 320 x 64] incorporant un objet cylindrique).....	147
Figure 5-25 : vue de côté (a) et du haut (b) sur le segment occupé obtenu à la fin de la 3 ^{ième} phase (environnement de taille [320 x 320 x 128] incorporant un objet conique).....	147
Figure 5-26 : vue de côté (a) et du haut (b) sur le segment occupé obtenu à la fin de la 3 ^{ième} phase (environnement de taille [320 x 320 x 128] incorporant un objet cubique).....	147
Figure 5-27 : vue de côté (a) et du haut (b) sur le segment occupé obtenu à la fin de la 3 ^{ième} phase (environnement de taille [320 x 320 x 128] incorporant un objet cylindrique).....	148
Figure 6-1 : segmentation par paliers avec un intervalle de PMO de largeur 0.1.....	151
Figure 6-2 : segmentation par paliers avec un intervalle de PMO de largeur 0.15.....	152
Figure 6-3 : extractions des subdivisions dont la PMO est bornée dans l'intervalle [0;0.35].....	153
Figure 6-4 : extraction des subdivisions dont la PMO est bornée dans l'intervalle [0.52;0.7].....	153
Figure 6-5 : planification de trajectoire avec besoin de contourner un obstacle ayant une forme complexe.....	155
Figure 6-6 : planification de trajectoire avec besoin de contourner deux obstacles, un rectangulaire et un autre circulaire.....	156
Figure 6-7 : planification de trajectoire longeant la bordure de l'espace de déplacement sécuritaire.....	156
Figure 6-8 : autre cas de planification de trajectoire longeant la bordure de l'espace de déplacement sécuritaire.....	157
Figure 6-9 : (a) image rayons X de la cage thoracique d'un patient [64], (b) les tissus rigides forment un segment unique (bleu).....	158

Figure 6-10 : (a) MRI de la tête et du cerveau d'un patient [64], (b) résultat de la segmentation où les tissus à forte résonnance sont représentés par des couleurs différentes selon leur distribution spatiale.....	158
Figure 6-11 : (a) « Coronal T1W MRI » de la face d'un patient [64], (b) résultat de la segmentation où les tissus à forte résonnance sont représentés par des couleurs différentes selon leur distribution spatiale	159
Figure 6-12 : (a) image rayons X de la face d'un patient prise par L.R. Gentry [64], (b) la section du crâne qui n'est pas traversée par les rayons est identifiée comme un segment unique (bleu).....	159
Figure 6-13 : (a) image rayons X de la hanche de droite prise par D. Lopresti [64], (b) la section des os qui ne sont pas traversés par les rayons est identifiée comme un segment unique (bleu).....	160
Figure 6-14 : (a) image rayons X des hanches prise par M. Akin [64], (b) les tissus rigides qui ne sont pas traversés par les rayons sont identifiés par des segments différents	160
Figure 6-15 : (a) image aérienne d'une carrière proche de la ville de Vancouver au Canada [65], (b) les espaces où sont extraits les minerais sont identifiés dans des segments séparés (vert, bleu et orange) selon leur distribution spatiale.....	161
Figure 6-16 : (a) image aérienne d'un terrain de camping dans la ville de Miami aux États-Unis [65], (b) les routes et les espaces où nous retrouvons une forte densité de motorisés sont identifiés dans des segments séparés (bleu, vert, blanc, marron, jaune et mauve) selon leur proximité.....	161
Figure 6-17 : (a) image satellite d'une région sur le bord du lac Ontario [65], (b) l'eau (noir), la terre ferme (bleu) et l'île (vert) sont identifiés en trois segments séparés	162
Figure 6-18 : (a) image satellite de la ville de Boston aux États-Unis [65], (b) l'eau (noir) et la terre ferme (bleu) sont identifiés en deux segments séparés	162
Figure 6-19 : (a) image satellite d'une ville sur la côte Est des États-Unis [65], (b) les régions densément peuplées sont mises en évidence	162
Figure 6-20 : (a) image aérienne d'un quartier résidentiel de la ville de Las Vegas aux États-Unis [65], (b) les maisons (bleu, vert, blanc, orange, mauve, marron), les routes ainsi que les espaces libres (noir) sont identifiés dans des segments séparés.....	163
Figure 6-21 : (a) image Premiers Pas [66], (b) résultat de notre algorithme dans lequel les segments sont séparés par une ligne blanche	163
Figure A-1 : interface usager pour le programme de simulation de capteur télémétrique.....	176
Figure A-2 : (a) espace encombré, (b) image probabiliste bidimensionnelle, (c) graphe d'occupation.....	176
Figure B-1 : interface usager développée à l'aide de la librairie FLTK	177
Figure C-1 : région occupée issue de la segmentation d'un environnement incorporant un objet conique	182
Figure C-2 : région libre issue de la segmentation d'un environnement incorporant un objet cylindrique	183
Figure C-3 : Interface usager du programme de segmentation tridimensionnelle	184

Table des tableaux

<i>Tableau 2-1 : caractéristiques du champ de vision du capteur Jupiter.....</i>	<i>13</i>
<i>Tableau 2-2 : caractéristiques dimensionnelles et temporelles pour chacune des trois images probabilistes utilisées comme exemples dans cette partie suite à l'application des deux premières phases de l'algorithme proposé dans [39].....</i>	<i>67</i>
<i>Tableau 4-1 : comparaison de plusieurs textures issues des images utilisées par Ojala et Pietikäinen [39], les blocs comparés ont été extraits manuellement soit d'une région unique à texture homogène, soit de deux régions dotées d'une texture différente, ce tableau illustre l'incapacité du G-statistique à juger l'uniformité des régions de taille réduite.....</i>	<i>90</i>
<i>Tableau 4-2 : variation du temps d'exécution et des erreurs de type 1 et de type 2 en fonction de l'écart-type sur la distance et du pas angulaire du capteur.....</i>	<i>98</i>
<i>Tableau 4-3 : variation des erreurs de type 1 et de type 2 en fonction de celle de l'écart-type de l'erreur du capteur sur la distance.....</i>	<i>99</i>
<i>Tableau 4-4 : variation des erreurs de type 1 et de type 2 en fonction de celle du pas angulaire entre les faisceaux de lecture.....</i>	<i>100</i>
<i>Tableau 4-5 : variation du temps d'exécution et des erreurs de type 1 et de type 2 en fonction du nombre de points de vue et de la taille de l'environnement.....</i>	<i>107</i>
<i>Tableau 4-6 : variation du temps d'exécution et des erreurs de type 1 et de type 2 en fonction du nombre de niveaux de discrétisation du contraste (b) et de la taille minimale d'une subdivision (S_{min}).....</i>	<i>114</i>
<i>Tableau 5-1 : mesures d'erreur et temps d'exécution pour chacun des modèles traités.....</i>	<i>141</i>

Table des abréviations

LBP	Local Binary Pattern
C	Contrast
DWF	Discrete Wavelet Frames
MI	Merger Importance
MIR	Merger Importance Ratio
PMI	Probabilité Moyenne d'Occupation
MRI	Magnetic Resonance Imaging

Résumé

Le concept des grilles d'occupation et des images probabilistes a été introduit à la fin des années quatre-vingt avec le travail d'Alberto Elfes [1][2][3][4] qui se situait dans le contexte de la robotique mobile pour la construction de cartes de l'environnement. Depuis, la recherche s'est principalement concentrée sur la représentation, la fusion de données et la génération des modèles d'occupation. Malgré que ces modèles d'environnements probabilistes soient extrêmement riches en terme de contenu, peu d'efforts ont été investis dans leur traitement et dans l'extraction des données pertinentes qu'ils renferment.

Ce travail de recherche contribue à ce domaine, du fait qu'il propose un algorithme de segmentation spécialisé dans le traitement des modèles d'environnements probabilistes bidimensionnels et tridimensionnels. La méthode de segmentation proposée est non-supervisée et basée sur l'analyse et sur la différenciation entre les textures qui caractérisent les régions dotées d'une occupation donnée. La texture est représentée à l'aide de la double distribution du « motif local binaire » (Local Binary Pattern, *LBP*) et du « contraste » (Contrast, *C*). Le ratio logarithmique de probabilité, le G-statistique, est utilisé afin de mesurer le degré de similarité entre les différentes régions de l'environnement. Cette mesure pseudo-métrique compare les distributions *LBP/C* relatives aux différents segments. L'algorithme proposé qui est utilisé pour segmenter l'espace probabiliste en régions uniformes dotées d'un état d'occupation déterministe, différencie entre les divers objets présents dans l'environnement en analysant la proximité entre les segments *occupés* obtenus. Notre schéma de segmentation ouvre la voie à un grand nombre d'applications en robotique autonome. Parmi celles-ci nous retrouvons :

- ❖ La planification de la trajectoire et de mouvements d'un robot mobile;
- ❖ L'évitement des obstacles et l'interaction d'un robot avec son environnement;
- ❖ La sélection autonome de points de vue dans la construction de cartes de l'espace;
- ❖ La reconnaissance d'objets par leur forme à partir de mesures de surface incomplètes et incertaines.

Deux versions de l'algorithme sont proposées, la première est spécialisée dans la segmentation des images probabilistes bidimensionnelles tandis que la seconde correspond à une

extension capable de prendre en compte une dimension additionnelle pour ainsi traiter le cas des environnements probabilistes tridimensionnels.

Le schéma de segmentation proposé est validé expérimentalement sur un grand nombre d'images probabilistes de dimensions variées et créées à partir de capteurs dotés d'une erreur sur la mesure de la distance ainsi que d'un pas angulaire limité. Le choix des paramètres intrinsèques du schéma de segmentation est déterminé empiriquement et évalué à l'aide d'une mesure d'erreur à deux niveaux basée sur les résultats obtenus avec différents réglages.

Enfin, l'applicabilité de l'algorithme proposé est évaluée au-delà du traitement des modèles d'environnements probabilistes en le testant, sans aucune adaptation, pour la segmentation d'images aériennes et médicales.

Remerciements

La réalisation de ce travail de recherche n'aurait pas été possible sans l'effort acharné de beaucoup de personnes. Premièrement j'aimerais remercier mes parents Jaber Abou Merhy et Rosette Khoury, mes sœurs Roula et Rouba et mon frère Firas. Je dois à leur support moral, affectif et matériel, toutes les réalisations que j'ai pu atteindre jusqu'à date. Deuxièmement la présence de Claudine Girard à mes côtés a été inestimable tout au long de mon travail et mes études. Finalement, je ne peux négliger ma reconnaissance envers mon superviseur Dr Pierre Payeur pour sa patience et le temps qu'il a consacré à m'aider, me guider et me soutenir.

Chapitre 1 Introduction

Le vingtième siècle a été marqué par la conquête de l'impossible du fait que l'être humain tentait par tous les moyens dont il disposait de découvrir le monde et l'espace qui l'entourent. Ceci s'est concrétisé par toutes les explorations spatiales et océaniques qui ont été réalisées. Depuis l'émergence du concept de terrorisme mondial, les préoccupations semblent se diriger vers des priorités de survies telles la défense territoriale, la protection ainsi que la préservation de l'homme et la sécurité. En parallèle à ces préoccupations, la concurrence globale qu'a créée la mondialisation pousse de plus en plus les compagnies à combattre pour leur survie, principalement en réduisant leurs coûts et en augmentant leur productivité. Tous ces soucis ont convergé vers le développement de systèmes et de robots autonomes capables de supporter, aider et étendre les capacités de l'homme.

Afin que ces systèmes autonomes soient fonctionnels, ils doivent avoir une idée précise de l'espace dans lequel ils agissent. Une multitude de capteurs imparfaits sont souvent utilisés simultanément afin de décrire et bâtir une représentation de l'environnement en question. Les grilles d'occupation probabilistes développées à la fin des années quatre-vingt correspondent à un modèle de représentation adapté, capable de combiner plusieurs prises de vue issues de différents capteurs et de prendre en compte l'erreur intrinsèque de chacun d'entre eux. Dans le cadre de notre recherche, les termes « image probabiliste » et « environnement probabiliste » désignent une grille d'occupation probabiliste générée à l'aide d'un capteur laser de profondeur doté d'un niveau d'incertitude élevé et d'une résolution angulaire limitée. La quantité importante d'informations que les grilles d'occupation renferment, relatives à l'occupation et aux caractéristiques de l'environnement n'est pas accessible directement par les systèmes autonomes. Une phase intermédiaire de traitement, la segmentation, est requise afin subdiviser l'espace mesuré en régions dotées de caractéristiques communes.

Le but principal de cette thèse est de résoudre le problème de la segmentation des grilles d'occupation probabilistes. Étant donné que ces dernières représentent des environnements réels dans lesquels des objets et des obstacles de toutes les formes possibles peuvent être rencontrés, notre algorithme est *à priori* non supervisé. Il se base sur l'analyse de la texture qui caractérise les environnements probabilistes. Il commence par subdiviser le modèle en régions cubiques de taille variable mais de texture et d'occupation relativement uniformes. Par la suite, les subdivisions adjacentes dotées d'une occupation similaire sont fusionnées afin de créer des segments, ces derniers

correspondent à une approximation des régions qui caractérisent l'environnement. Enfin, une phase de raffinement permet de reclasser les cellules situées à la bordure de deux espaces adjacents et de générer des segments dotés d'états d'occupation déterministes soit : *libre*, *occupé* et *inconnu*. Cette classification est particulièrement utile dans le cadre d'applications en robotique autonome de planification de trajectoire et d'interaction avec l'environnement dans lesquels un robot a besoin d'identifier les objets afin de pouvoir les contourner et/ou les manipuler de façon sécuritaire.

Deux types d'erreur sont utilisés afin d'évaluer la qualité de notre segmentation, la première correspond à la proportion de cellules réellement occupées qui ont été classées comme *libres*, tandis que la seconde représente la proportion des cellules réellement libres qui ont été mal classées. Les résultats que nous avons obtenus sont très convaincants compte tenu de la nature des environnements probabilistes. Ils démontrent que les segments *occupés* englobent bien les objets auxquels ils correspondent. Ceci maximise les chances de réussite de toute application de planification de trajectoire ou d'interaction avec l'environnement.

L'algorithme proposé a été conçu en premier lieu pour traiter le cas des images probabilistes bidimensionnelles, mais en se basant sur l'observation selon laquelle les objets et l'espace ne sont jamais uniformes le long de leur élévation, nous avons étendu l'implantation pour couvrir le cas des environnements tridimensionnels. Une des particularités du schéma proposé provient de sa capacité à distinguer entre les régions occupées spatialement non adjacentes. De plus, l'algorithme est capable de reconnaître en segments uniques les objets qui ont été partiellement balayés par les capteurs. Cette dernière observation est représentative de l'opération des capteurs laser de profondeur dont l'écart entre les faisceaux adjacents est proportionnel à la distance de l'émetteur. Cet écart crée le long d'une bordure des espaces *inconnus* entre les points de lecture. La classification des objets non adjacents en segments distincts ouvre la voie à un grand nombre d'applications en robotique telles l'identification des différents objets à partir de leur forme ainsi que l'estimation et l'assignation de propriétés intrinsèques différentes aux objets afin qu'ils puissent être manipulés en conséquence.

Les résultats obtenus à l'aide de l'algorithme de segmentation tridimensionnelle proposé sont aussi convaincants que ceux obtenus dans le cas bidimensionnel. Ils démontrent que notre algorithme constitue une base solide pour toute application réelle dans laquelle un système autonome se déplace dans un espace inconnu et/ou interagit avec les objets qu'il contient.

1.1 Objectifs

Les grilles d'occupation probabilistes correspondent à des images très riches en terme d'information. Mais l'intégration de différents points de vue et de l'erreur des capteurs dans la représentation augmente considérablement la complexité des modèles. De plus, l'utilisation de capteurs laser de profondeur ne permet pas de récolter un état d'occupation probabiliste pour l'ensemble des cellules situées dans le champ de vision car les faisceaux de lecture adjacents sont séparés d'une région inconnue dont la taille dépend de la distance du capteur et de sa résolution angulaire.

Compte tenu de ces facteurs, le but principal de ce travail de recherche consiste à élaborer un algorithme de segmentation capable de subdiviser l'environnement probabiliste en régions dotées d'un état d'occupation uniforme et déterministe. L'extraction de l'information d'occupation est particulièrement utile dans le cadre d'applications en robotique autonome du fait qu'elle permet au robot de détecter, en se basant sur ses propres critères, les régions d'intérêt et d'agir en conséquence. À titre d'exemple, l'identification des régions libres et occupées d'un environnement permet à un robot mobile de planifier de façon autonome et sécuritaire sa trajectoire.

La complexité de la représentation et les caractéristiques des grilles d'occupation probabilistes imposent des critères spécifiques sur la conception de l'algorithme de segmentation. En effet, à la différence des algorithmes supervisés qui assument une connaissance préalable des caractéristiques de l'image et/ou une intervention humaine, une segmentation non supervisée est ici requise. Cette exigence est nécessaire pour les applications en robotique autonome dans lesquelles un ou plusieurs robots opèrent dans un environnement, *à priori*, inconnu d'avance.

Afin de mieux comprendre le modèle de données sur lequel l'algorithme visé agit, une revue littéraire des méthodes de génération et de représentation des grilles probabilistes est en premier lieu nécessaire. Par la suite, dans le but de choisir une plateforme stable pour la segmentation des environnements probabilistes générés à l'aide de capteurs dotés d'un niveau d'incertitude élevé et d'une résolution angulaire limitée, les principales méthodes de segmentation classiques et modernes seront analysées.

L'ensemble de la revue de la littérature constitue une source d'inspiration pour la conception d'un algorithme de segmentation capable de traiter les environnements probabilistes. Par ailleurs, ce

travail vise à développer des critères pour juger quantitativement et objectivement de la validité et de la stabilité de l'approche proposée. Un autre objectif est donc de développer des mesures d'erreur compatibles avec les modèles et avec les différentes configurations de capteurs utilisés (variation de la résolution angulaire et de l'erreur sur la distance) et différents réglages des paramètres intrinsèques à l'algorithme de segmentation.

Sachant que la troisième dimension est non négligeable pour toute application réelle, une extension de l'algorithme bidimensionnel pour couvrir le cas des environnements probabilistes tridimensionnels doit être développée. Outre la qualité des résultats de la segmentation, le schéma en trois dimensions doit chercher à minimiser l'impact de la taille des modèles traités sur la charge de calcul.

La capacité du schéma de segmentation à classifier à l'aide d'états d'occupation déterministes, tels : *libre*, *occupé* et *inconnu*, pourrait ne pas répondre aux exigences particulières de certaines applications, à titre d'exemple celles nécessitant un degré de sécurité supplémentaire dans la planification de trajectoire d'un robot mobile. Par conséquent, diverses extensions, basées sur le schéma de segmentation, doivent être envisagées afin de fournir un degré d'autonomie additionnel quant aux paramètres de la segmentation. Finalement, cette recherche vise aussi à tester la généralité du schéma de segmentation développé sur des images de nature variée et de niveau de complexité élevé.

1.2 Contributions

Les principales contributions que cette thèse apporte aux domaines de la segmentation et de la modélisation par grilles d'occupation peuvent se résumer par les points suivants :

- ❖ Elle constitue un pont entre deux domaines relativement complexes et peu explorés dans la littérature, soient la segmentation basée sur l'analyse de la texture et la représentation des images probabilistes;
- ❖ Elle analyse l'applicabilité de méthodes de segmentation modernes aux grilles d'occupation probabilistes;

- ❖ Elle propose un algorithme non supervisé spécialisé dans la segmentation des environnements probabilistes. Ce dernier est capable de distinguer en segments uniques les objets suivant leur proximité et ne nécessite aucun entraînement ni intervention humaine;
- ❖ Elle propose un algorithme de segmentation tridimensionnelle relativement performant en terme d'exécution. Sa conception ainsi que la qualité des résultats qu'il génère fournissent une base stable pour des applications en robotique des manipulateurs.
- ❖ Elle contribue au domaine du traitement des grilles d'occupations par une analyse détaillée des paramètres qui influencent la segmentation dans le contexte d'une application réelle en modélisation de l'espace;
- ❖ Elle valide expérimentalement l'approche de segmentation proposée sur des images de nature variée, ce qui ouvre la voie à un champ d'application plus large;

1.3 Organisation

Cette thèse est divisée en sept chapitres. Le premier d'entre eux introduit le travail que nous avons réalisé. Le second commence par survoler le concept des grilles d'occupations probabilistes. Par la suite, il fournit une analyse des caractéristiques texturales des images probabilistes. Finalement, il introduit les principales méthodes de segmentation développées jusqu'à date et analyse leur applicabilité sur les images probabilistes. Le troisième chapitre est consacré aux différentes étapes du processus de développement qui a mené à la conception de l'algorithme de segmentation bidimensionnelle proposé. Le quatrième chapitre est dédié à la présentation des résultats expérimentaux et à leur analyse. Le cinquième chapitre étend l'algorithme proposé pour couvrir le cas des environnements probabilistes tridimensionnels. Le sixième chapitre expose quelques extensions de notre algorithme bidimensionnel et valide cette approche dans une application de planification de trajectoire en robotique. Il évalue aussi son potentiel dans la segmentation d'images biomédicales et aériennes. Finalement, le dernier chapitre conclut notre travail de recherche et introduit nos projets futurs.

Chapitre 2 Revue de la littérature

2.1 Introduction

Ce premier chapitre présente une revue littéraire des méthodes de segmentation appliquées sur les images probabilistes. Étant donné que le concept des grilles d'occupation et celui des environnements probabilistes qui en découle ne date que de la fin des années quatre-vingt, la littérature fournit peu de ressources spécialisées dans la segmentation et le traitement de ce type de modèles. Afin de contourner cet obstacle, nous avons divisé notre exploration de la littérature en trois volets. Le premier développe les principes et la construction des modèles d'espaces probabilistes, soit les grilles d'occupation bidimensionnelles et tridimensionnelles telles qu'introduites dans [1], [2], [3], [4], [5] et [6]. Le second traite les caractéristiques texturales de ces modèles. Le dernier volet constitue une revue littéraire des principales méthodes de segmentation développées jusqu'à date. L'applicabilité de chacune d'entre elles sur le cas particulier des images probabilistes est de même analysée.

2.2 Grilles d'occupation probabilistes

Le concept des grilles d'occupation probabilistes incorpore certaines notions des théories des probabilités, de l'estimation optimale, des modèles de champs aléatoires ainsi que de la théorie des décisions. Il a été principalement développé par Alberto Elfes et Hans Moravec dans les années quatre-vingt. Le nombre d'ajouts pertinents qui ont été effectués par la suite est limité. Cette section commence par introduire les principes de base des grilles d'occupation. Par la suite les travaux de Elfes [1][2][3][4] sont exposés. La troisième section constitue une extension du modèle bidimensionnel d'Elfes et traite la modélisation d'environnements dynamiques tridimensionnels [5]. En dernier lieu, nous développons la physique de la construction des grilles probabilistes utilisées dans ce travail.

2.2.1 Introduction et avantages

Formellement, un *champ d'occupation* est un processus stochastique à états discrets défini sur un ensemble continu de coordonnées spatiales, tandis que la *grille d'occupation* est définie sur une structure réticulaire¹ spatiale discrète. Ainsi, la représentation par grille d'occupation probabiliste est un modèle de champ aléatoire qui maintient des estimés stochastiques de l'état d'occupation de chaque cellule de l'espace.

D'une manière générale et afin qu'un robot puisse construire une représentation de son environnement à partir de données de capteurs télémétriques, l'état des cellules est estimé en interprétant les données de profondeur. Cette interprétation est réalisée à l'aide des différents modèles probabilistes des capteurs mis en jeu. Plusieurs méthodes de fusion de données ont été développées afin de permettre une mise à jour incrémentale de la grille d'occupation en combinant des lectures émanant de différents capteurs et points de vue. Parmi ces méthodes, nous citons la Théorie des Probabilités [7], Dempster-Shafer et les ensembles flous.

La fusion de données issues de différentes grilles d'occupation obtenues à l'aide de différents capteurs (tels un sonar et un capteur de profondeur laser) présente plusieurs avantages [3]. Premièrement, si le modèle est bien conçu, la représentation peut exploiter la complémentarité de la couverture de l'espace ainsi que les points forts de chaque modèle de capteur, ce qui accroît considérablement la tolérance aux fautes du système global [6]. Deuxièmement, la représentation par grilles d'occupation simplifie et homogénéise le traitement des données reçues, quelque soit leur origine, étant donné qu'elles sont représentées sur une base commune [3]. En troisième lieu, les capteurs peuvent être traités d'une façon modulaire, des grilles probabilistes sont ainsi construites pour chacun d'entre eux, et à la fin les représentations intermédiaires sont fusionnées pour représenter l'espace global [4]. Finalement, le modèle probabiliste des grilles d'occupation permet de prendre en compte différents niveaux de précision dans la représentation du résultat de chaque capteur, et l'incertitude dans la position du robot suite à ses déplacements. Cette incertitude est modélisée dans [4] par une fonction gaussienne qui brouille les grilles préalablement générées.

¹ « Lattice ».

2.2.2 Grilles d'occupation telles qu'introduites par Elfes

L'inclusion du travail d'Alberto Elfes est de nature primordiale étant donné qu'il est parmi les premiers à définir une plateforme mathématique et théorique sur laquelle repose le modèle des grilles d'occupation probabilistes. Ce dernier définit dans [1], [2] et [4] les grilles d'occupation comme une représentation en mosaïque de l'espace bidimensionnel ou tridimensionnel. Dans cette représentation, chaque cellule contient un estimé probabiliste de son état d'occupation.

Elfes [4] divise conceptuellement en deux étapes la construction de la représentation de l'environnement à partir des données issues de capteurs. Dans la première, une lecture de profondeur r est interprétée par le biais d'un modèle de capteur probabiliste (Figure 2-1 a et b). Ce modèle est représenté par une fonction de densité probabiliste de la forme : $p(r|z)$, où z est la distance séparant le capteur de l'objet. Lors de la seconde étape, la lecture r est utilisée pour mettre à jour les probabilités d'occupation des *proxels* : $P[s(x)|r](x)$, où $s(x)$ correspond à l'état de la cellule x . Cette mise à jour incrémentale qui nécessite un processus de fusion des données, est réalisée à l'aide d'une procédure d'estimation Bayésienne [4] (Figure 2-1-c). Cette dernière permet d'intégrer les probabilités d'occupation en se basant sur des mesures de profondeur captées à l'aide de différents capteurs et points de vue.

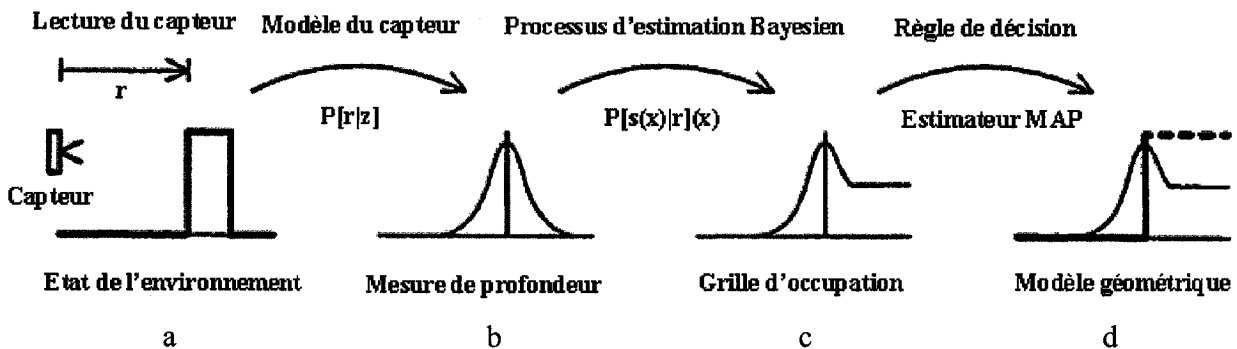


Figure 2-1 : construction de la grille d'occupation à l'aide des mesures de profondeur. (a) et (b) interprétation d'une lecture r à l'aide d'un modèle de capteur probabiliste, (c) mise à jour de la probabilité d'occupation des *proxels*, (d) modèle déterministe issu de l'application de l'estimateur MAP (reproduite de [4])

Il est important de noter qu'un modèle déterministe peut être obtenu de la grille d'occupation en appliquant une règle de décision telle que le *maximum a posteriori* (MAP) (Figure 2-1-d). Cette dernière permet d'assigner des états discrets, tels *libre*, *occupé* ou *inconnu*, aux cellules. Néanmoins,

plusieurs tâches peuvent quand même opérer directement sur la représentation en grille d'occupation (Figure 2-1-c) si aucune règle de décision n'est appliquée.

Afin d'accroître la performance et l'efficacité des systèmes robotiques en général, et autonomes en particulier, une variété de capteurs sont utilisés. Dans le cadre d'applications incorporant des robots mobiles plusieurs capteurs opérant en parallèle fournissent un niveau de sécurité et de tolérance aux fautes plus élevé du fait que chacun d'entre eux a ses propres caractéristiques opérationnelles. Ces considérations ont poussé Elfes [4] à intégrer à l'aide de la formule de mise à jour séquentielle du théorème de Bayes, la fusion de données issues de plusieurs capteurs dans la structure des grilles d'occupation probabilistes. Deux modèles d'intégration et de mise à jour des grilles d'occupation ont été identifiés. Le premier (Figure 2-2-a) concerne l'intégration de plusieurs capteurs dont l'apport est appliqué sur une grille d'occupation unique. Tandis que le second (Figure 2-2-b) traite le cas où plusieurs capteurs agissent sur des grilles séparées, celles-ci sont ultérieurement combinées afin de créer un modèle plus complet de l'environnement.

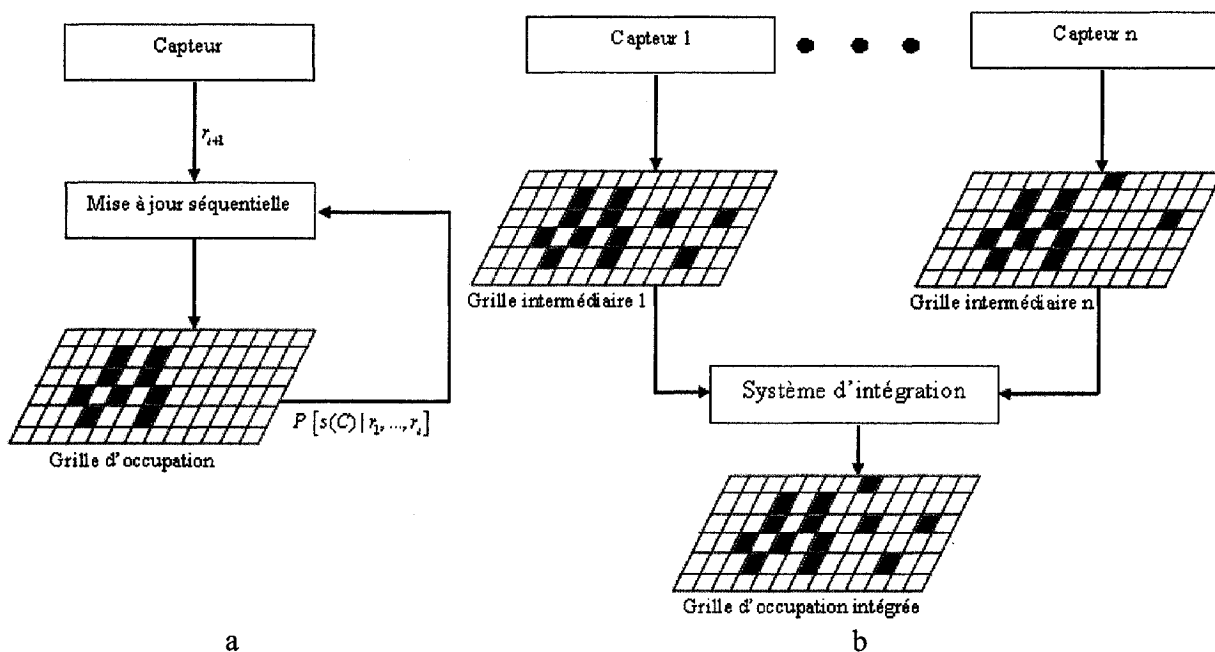


Figure 2-2 : mises à jour séquentielles et intégration de grilles issues de plusieurs capteurs

2.2.3 Modélisation d'environnements tridimensionnels

En dépit de l'hypothèse d'Elfes [4] selon laquelle les probabilités conditionnelles sont indépendantes, le modèle développé dans [1], [2], [3] et [4] nécessite le calcul de la fonction de densité probabiliste du capteur pour l'ensemble des configurations possibles. Cette évaluation est simple dans le cas bidimensionnel, mais pour un capteur doté d'une distribution tridimensionnelle (le cas gaussien est traité dans [5]), chaque cellule avoisinante à celle qui est mesurée doit être évaluée. Dans ce cas, le nombre d'évaluations à effectuer explose et conduit à des implantations inefficaces.

La méthode introduite par Payeur *et al.* [5] a pour but de résoudre ce problème et permet d'estimer la fonction de distribution du capteur ainsi que l'état global de la grille d'occupation probabiliste. Ceci consiste à trouver une approximation de la fonction de distribution probabiliste caractérisant l'occupation de l'espace (*Occupancy Probability Distribution Function OPDF*) qui résulte de l'évaluation numérique proposée par Elfes [4]. L'approximation de l'*OPDF* est utilisée afin de calculer la probabilité de l'occupation d'un volume dans l'espace tridimensionnel centré sur le point de référence du capteur. Ce concept appliqué sur un capteur de profondeur [5] conduit à des probabilités d'occupation limitées à une grille sphérique. Cette dernière qui correspond au champ de vision du capteur est centrée sur son point de référence. Une fois que le nombre de mesures désirées est atteint, les grilles sphériques résultantes sont intégrées dans la grille d'occupation cartésienne qui est à son tour encodée sous forme d'*octree*.

L'approche introduite Payeur *et al.* [5] diffère de celle développée par Elfes [4] à deux niveaux, premièrement la dépendance sur le phénomène de discrétisation créé par l'indépendance entre l'état des cellules [4] est éliminée. Deuxièmement, au lieu de suivre une approche d'intégration globale au niveau de précision le plus élevé, Payeur *et al.* [5] ont utilisé la propriété de multi-résolutions des *octrees*. Ces actions telles que présentées accroissent l'efficacité et la performance du modèle initial en terme de charge de calcul, et par conséquent le rendent extensible pour couvrir le cas des environnements tridimensionnels.

2.2.4 Physique de la construction des images probabilistes

La construction des images probabilistes dépend, comme nous l'avons introduit dans les sections précédentes, des données de profondeur générées à l'aide d'une ou de plusieurs lectures de capteurs. En pratique, les images probabilistes sont obtenues soit par l'acquisition de données de

profondeur à l'aide d'un capteur physique, soit par la simulation du modèle des capteurs ainsi que de l'environnement à l'étude. Ces deux outils de génération de grilles probabilistes ont été à notre disposition. Ceux-ci impliquent respectivement le capteur de profondeur Jupiter [10] et le programme de simulation développé par Bolzon (Annexe A).

Dans un premier lieu, nous introduisons le concept des images de profondeur [9], le capteur Jupiter [10] sera présenté comme un capteur laser typique dont le programme de simulation développé par Bolzon, et présenté dans la seconde partie, s'inspire.

2.2.4.1 Images de profondeur et capteur Jupiter

Les images de profondeur, parfois appelées images 3D fournissent une représentation 2.5D d'une réalité tridimensionnelle. La demi-dimension manquante relève d'une part de l'incapacité du capteur à intercepter les occlusions et d'autre part de la limitation des mesures à un seul point de vue. Par conséquent, comme le montre la Figure 2-3 ci-dessous, une prise de mesure unique ne permet pas de construire un modèle complet d'un environnement encombré.

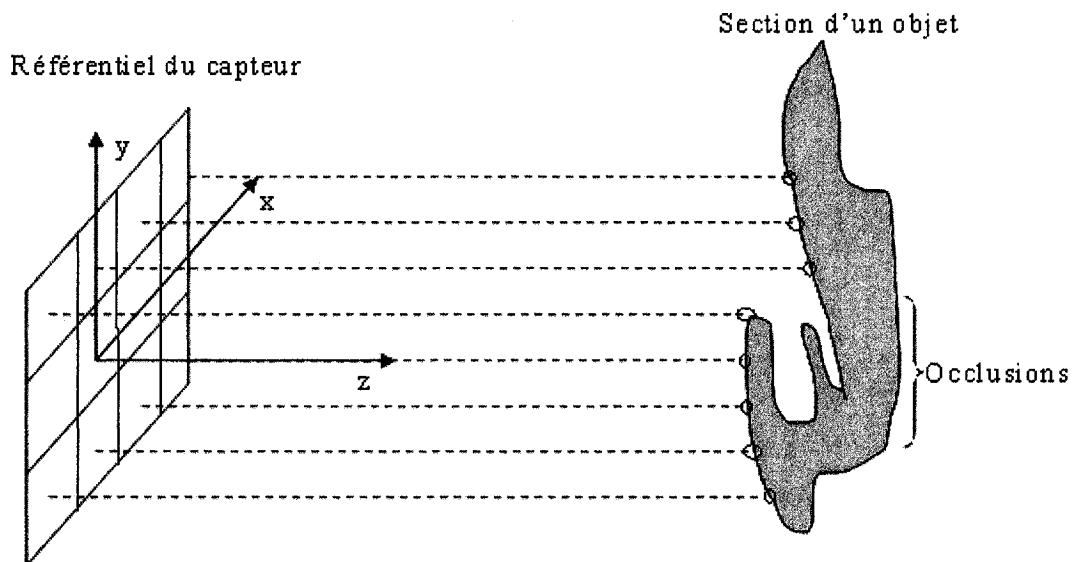


Figure 2-3 : les images de profondeur fournissent une représentation 2.5D d'une réalité tridimensionnelle

Les images de profondeur tout comme les images classiques, sont encodées sous forme de tableaux bidimensionnels. Mais à la différence des images d'intensité, la valeur des éléments correspond à la distance entre un référentiel associé au capteur et un point de surface situé dans

l'environnement. Une représentation équivalente consisterait à encoder les coordonnées cartésiennes des points mesurés.

Toutefois, la majorité des capteurs ne sont pas capables de fournir une représentation complète à partir d'un certain point de vue. En effet, certains permettent une seule prise de mesure à la fois, d'autres, tel le capteur Jupiter [10], sont capables de balayer une ligne complète sans avoir à bouger. Le besoin de déplacer les capteurs est à la source de nombreux problèmes de mises en registre. Afin de coordonner et de stabiliser les prises de mesures, les capteurs de profondeur sont souvent fixés sur des plateformes mécaniques (Figure 2-4).

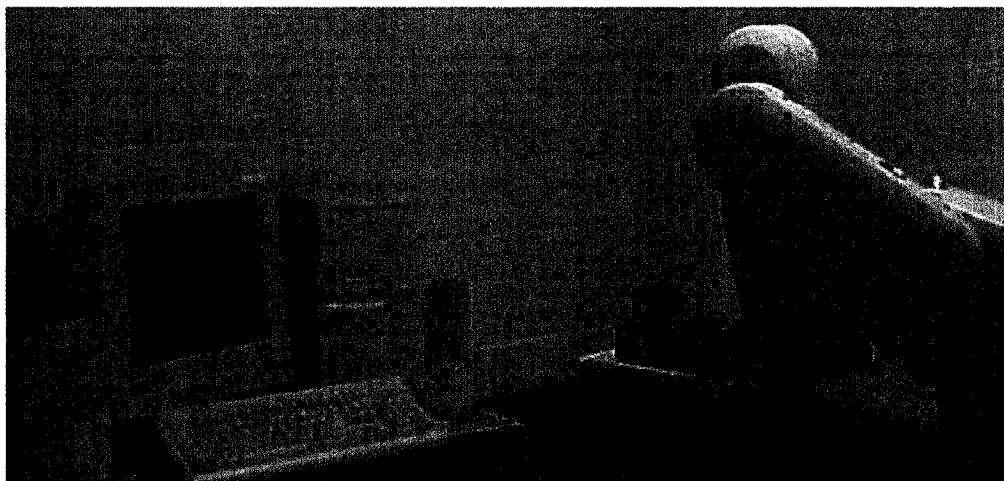


Figure 2-4 : capteur Jupiter monté sur un bras robotique

La nature du signal utilisé lors de la prise de mesures définit un capteur de profondeur comme étant soit actif, soit passif. Un capteur actif projette de l'énergie (lumière, son, ondes électromagnétiques...) sur une surface, puis estime la distance en détectant le point de réflexion ou en calculant le temps d'aller-retour. Les capteurs passifs quant à eux construisent une représentation des surfaces observées à partir d'images d'intensité (stéréoscopie, *shape from shading*...). Les capteurs actifs, dont fait partie Jupiter, se démarquent par leur insensibilité à la luminosité de l'environnement étant donné qu'ils émettent le signal utilisé pour la prise de mesures, mais le balayage de l'environnement requiert beaucoup plus de temps que l'acquisition d'une image d'intensité.

Une grande partie des capteurs actifs utilisent la triangulation [9] afin de déterminer la position d'un certain point de l'environnement par rapport au référentiel du capteur. Grâce au principe de la triangulation auto-synchronisée, Jupiter permet de collecter des mesures de profondeur

le long d'une ligne à la fois. La Figure 2-5 ainsi que le Tableau 2-1 représentent les caractéristiques du champ de vision du capteur Jupiter.

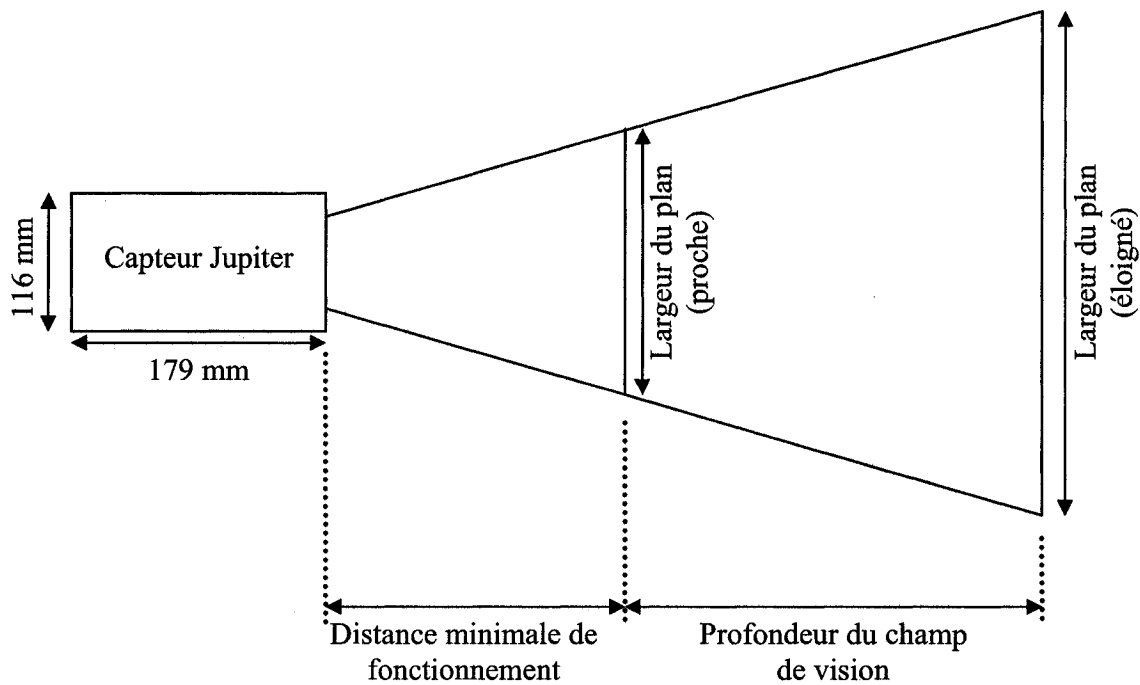


Figure 2-5 : champ de vision du capteur Jupiter (reproduite de [10])

Capteur Jupiter	
Distance minimale de fonctionnement	170 mm
Profondeur du champ de vision	1005 mm
Largeur du plan : plan proche / plan éloigné	108 mm / 579 mm
Résolution de profondeur moyenne	0.7 mm

Tableau 2-1 : caractéristiques du champ de vision du capteur Jupiter

Le capteur Jupiter opère suivant deux modes, le premier implique 256 lectures le long d'une ligne horizontale avec une fréquence de 40 lectures par seconde, tandis que le second mode implique 512 échantillons, avec une fréquence de 20 lectures par seconde. Suivant le mode choisi, Jupiter est capable de balayer une ligne horizontale complète en 0.025 et 0.05 secondes respectivement.

2.2.4.2 Programme de simulation en deux dimensions

Le programme développé par Bolzon (Annexe A) simule un robot mobile capable de détecter la présence d'objets dans un espace encombré, défini par l'utilisateur (Figure 2-6-a), imitant ainsi le fonctionnement du capteur Jupiter décrit précédemment. Lors d'une prise de mesures, le robot détecte la présence d'objets à l'aide d'un faisceau laser simulé ayant une portée maximale de deux

mètres. Si ce faisceau touche une cellule occupée par un objet, le robot calcule la distance qui le sépare des cellules situées le long du faisceau laser.

Le programme permet au robot de prendre des mesures suivant différents points de vue, chacun d'entre eux étant caractérisé par une position dans l'espace et par une certaine direction. Une fois positionné, le robot réalise avec son laser un balayage total de 30 degrés.

Ensuite, le programme se sert des distances calculées précédemment pour construire une grille d'occupation probabiliste bidimensionnelle (Figure 2-6-b) relative à l'environnement simulé (Figure 2-6-a). Cette image probabiliste est représentée par une matrice dans laquelle les cellules se voient assigner des probabilités d'occupation, comprises entre 0 et 1. Dans la Figure 2-6-b, les niveaux de gris correspondent aux différents états d'occupation, noir étant libre, gris étant inconnu et blanc étant occupé.

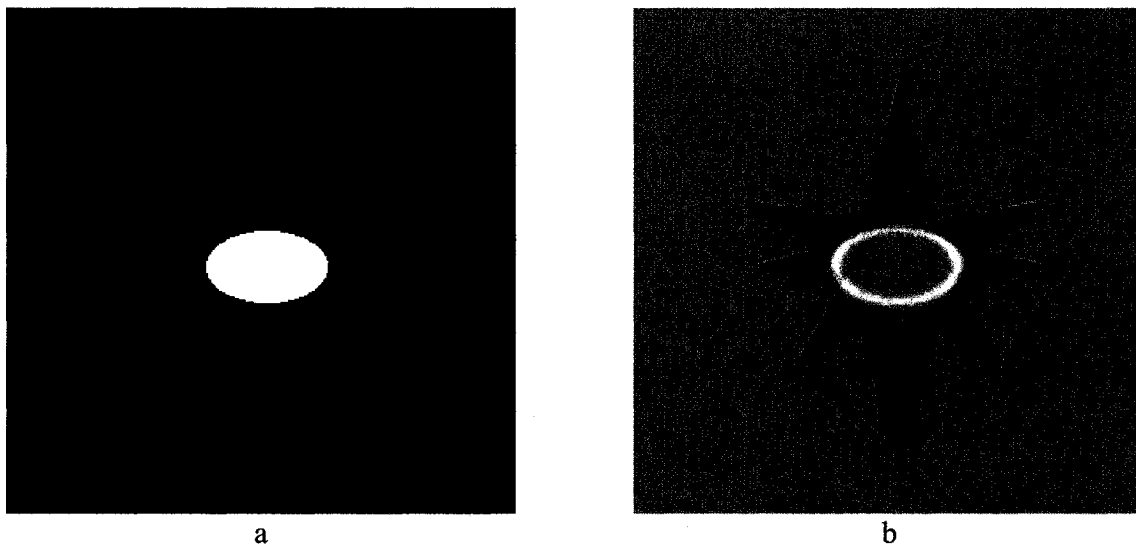


Figure 2-6 : (a) exemple d'un environnement encombré où les espaces libre et occupé sont respectivement en niveaux noir et blanc, (b) image probabiliste incorporant huit lectures de capteurs à partir de points de vue différents

Cet outil de simulation d'un capteur télémétrique laser mobile a servi de plateforme expérimentale pour le développement et l'évaluation des méthodes de segmentation de grilles d'occupation probabilistes dans le cadre des travaux de recherche présentés dans cette thèse.

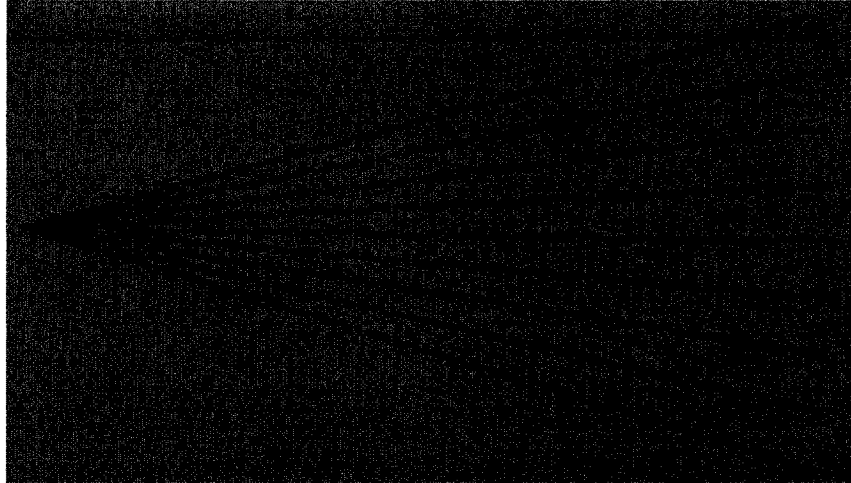
2.3 Particularité des textures

Dans le modèle des grilles d'occupation probabilistes, les pixels représentent des cellules spatiales, et leur intensité qui varie de zéro à un (de noir à blanc), correspond à la probabilité qu'elles soient occupées. Ainsi, seules les cellules spatiales qui ont été traversées par des faisceaux de lecture issus du capteur actif, ont une occupation connue. La partie de l'espace qui reste inexplorée, demeure dans un état inconnu et se caractérise par une probabilité d'occupation intermédiaire égale à 0.5 (gris moyen).

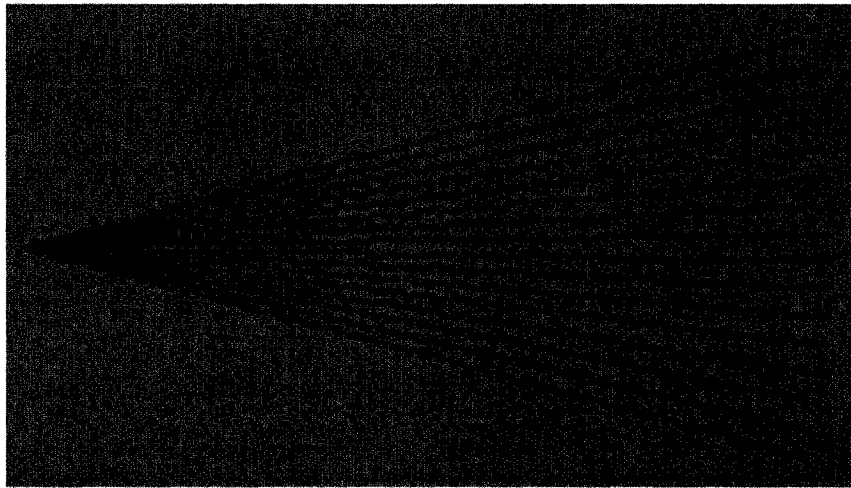
Cette section se concentre sur l'analyse, l'identification et la compréhension des caractéristiques texturales des images probabilistes obtenues à l'aide du simulateur de Bolzon. Elle se divise en trois sections, chacune d'entre elles met l'accent sur une des particularités des textures. Ainsi notre analyse permet de repérer les éléments clefs que les algorithmes de segmentation doivent considérer pour fournir une solution adéquate à notre problème.

2.3.1 Régions explorées non-uniformes

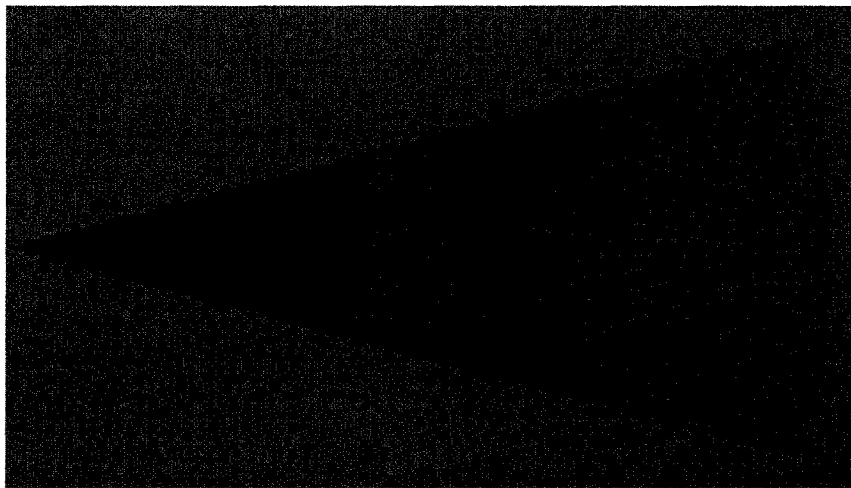
Étant donné que les faisceaux laser d'un capteur de profondeur typique tel Jupiter [10], ne couvrent pas l'espace balayé en sa totalité (Figure 2-7 a, b et c), une région spatiale explorée se caractérise par une alternance de fuseaux dont la probabilité d'occupation est connue et d'espaces non balayés ayant une occupation inconnue. De plus, étant donné que les faisceaux laser sont tous issus d'un même point et peuvent atteindre la même profondeur maximale, l'espace balayé par une seule lecture a l'air d'une section de cercle centrée sur la position du capteur. Le pas angulaire entre les différents faisceaux influence énormément la texture qui caractérise les régions explorées. Ainsi, plus ce pas est réduit, plus des composantes hautes fréquences sont introduites dans les régions proches du capteur, tandis que les régions plus éloignées sont moins touchées étant donné que les rayons sont plus espacés (Figure 2-7-b). Ces hautes fréquences peuvent être caractérisées jusqu'à un certain point par une texture, mais au-delà d'une valeur limite pour le pas angulaire, des interférences sont introduites (Figure 2-7-c). Celles-ci peuvent biaiser les résultats d'une segmentation basée sur l'analyse des textures et mal adaptée.



a



b

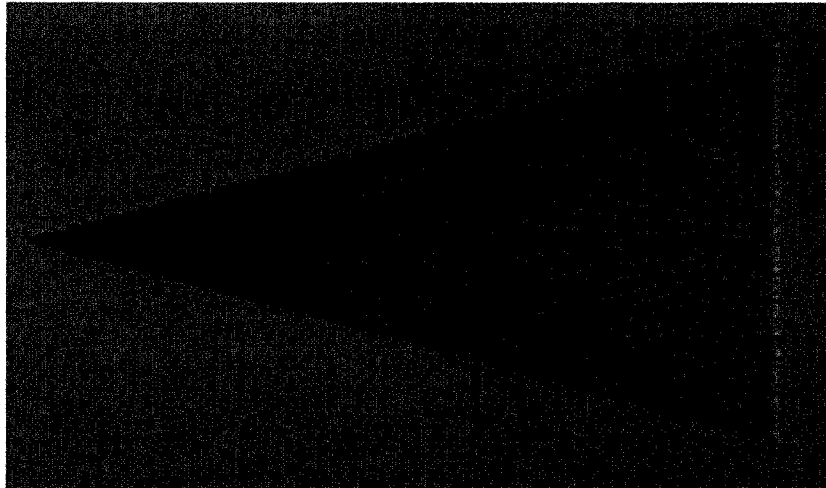


c

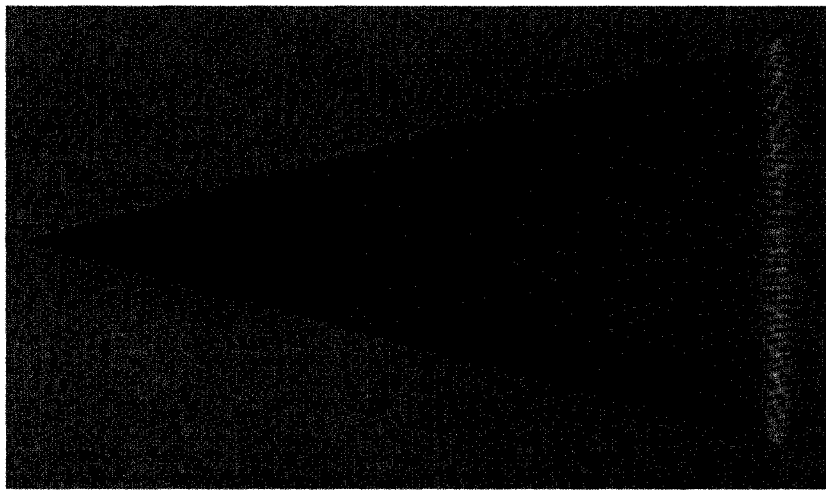
Figure 2-7 : lectures dans des environnements non-encombrés avec des pas angulaires de (a) 4, (b) 2 et (c) 0.5 degrés respectivement [noir = faisceau, gris = espace non balayé]

La présence de faisceaux de lecture influence aussi le motif de la texture dans les régions dotées d'une occupation connue (*libre* ou *occupée*). Étant donné que le but de la segmentation est principalement d'isoler les objets et les espaces libres présents dans l'environnement, il est primordial d'analyser les caractéristiques des rayons de lecture. En effet, dans les images probabilistes, l'intensité des cellules scannées définit leur occupation. Ainsi, au niveau d'un même faisceau, la transition entre la section *libre* et celle qui est *occupée* est importante car c'est au niveau de cette dernière que la bordure entre les segments *libres* et *occupés* sera positionnée. Mais les images probabilistes se caractérisent aussi par l'intégration de l'erreur du capteur dans les mesures de profondeur. Cette erreur est considérée dans notre cas comme étant gaussienne et est caractérisée par son écart-type. Plus la valeur de l'écart-type est grande, plus la transition entre les espaces libre et occupé est lente (Figure 2-8 a, b et c). De ce fait, la valeur de l'écart-type ajoute un critère supplémentaire dans l'analyse de la texture, soit celui du contraste entre les régions d'occupation différente.

Ainsi, le motif et le contraste de la texture sont influencés respectivement par le pas angulaire entre les faisceaux de lecture et par l'écart-type sur la mesure de la profondeur.



a



b



c

Figure 2-8 : lectures dans un environnement encombré avec des écarts-type sur la mesure de la profondeur de (a) 1, (b) 3 et (c) 7 cm respectivement [noir = libre, blanc = occupé, gris = inconnu]

2.3.2 Présence de textures aléatoires

En second lieu, les textures présentes dans les images probabilistes se distinguent par un patron totalement aléatoire. Comme nous pouvons le remarquer en observant les images de la Figure 2-7, une image probabiliste incorporant une seule lecture de capteur contient un nombre relativement limité de textures, celles-ci varient légèrement avec la distance de la source. La variation de texture est principalement due au fait que les faisceaux de lecture ne sont pas parallèles et que la distance entre ceux-ci augmente plus nous nous éloignons du centre de la section du cercle qui représente l'espace exploré.

Mais comme nous l'avons vu dans la section 2.2, les grilles d'occupation probabilistes se distinguent par leur capacité de bâtir un modèle global de l'environnement à partir de plusieurs lectures issues de différents points de vue et parfois de différentes sortes de capteurs [3]. Par conséquent il est possible qu'une multitude de lectures explorent un ensemble adjacent de cellules spatiales. C'est la redondance des lectures dans un espace d'intérêt donné qui augmente considérablement la complexité des textures dans l'image probabiliste résultante.

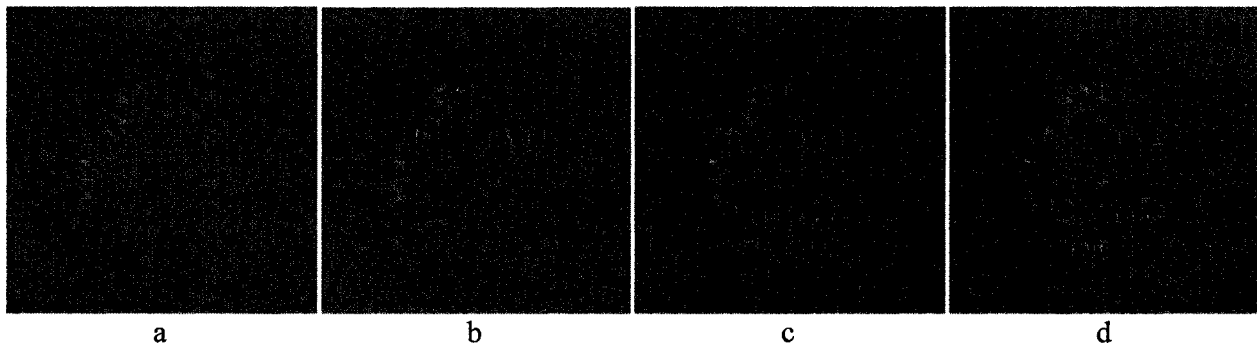


Figure 2-9 : images probabilistes incorporant respectivement (a) 2, (b) 3, (c) 4 et (d) 5 points de vue avec un pas angulaire de 3 degrés et un écart-type sur la mesure de la profondeur de 3 cm

Les images de la Figure 2-9 sont obtenues en simulant à l'aide de l'outil développé par Bolzon le même espace dans lequel nous retrouvons un objet en forme de prise électrique. Nous avons effectué un zoom dans la région encombrée afin de mieux percevoir les textures qui résultent du croisement des faisceaux issus de différentes lectures. Les images de la Figure 2-9 incorporent respectivement deux (Figure 2-9-a), trois (Figure 2-9-b), quatre (Figure 2-9-c) et cinq (Figure 2-9-d) prises de mesures. Les lectures sont caractérisées par un pas angulaire de trois degrés et un écart-type sur la distance de 3 cm. La Figure 2-9 illustre le fait que dans une même région spatiale, le nombre de

textures observées diffère énormément suivant le nombre de points de vue qui se chevauchent (par exemple la partie de gauche dans la Figure 2-9 a, b et c). Ainsi, la variation de texture au sein d'une même image probabiliste résulte de différents phénomènes :

- ❖ La distance croissante de l'objet par rapport au capteur et à la source laser implique une texture différente;
- ❖ Les régions de croisement entre les faisceaux de mesure pour des images à observations multiples ont naturellement une texture différente de celle des régions explorée par une lecture unique;
- ❖ Le fait que les rayons peuvent avoir n'importe quelle direction selon la localisation du point d'observation fait qu'une région de croisement entre un nombre fixe de points de vue n'est pas dotée d'une texture uniforme sur toute sa surface. En effet, comme le montre les images de la Figure 2-10, puisque les faisceaux issus d'un certain point de vue ne sont pas parallèles, l'intersection de plusieurs lectures crée des formes géométriques variées et non-homogènes suivant l'espace de croisement. Ainsi, dans les images ci-dessous, la différence de texture est évidente entre les régions du bas à gauche et celle en haut à droite.

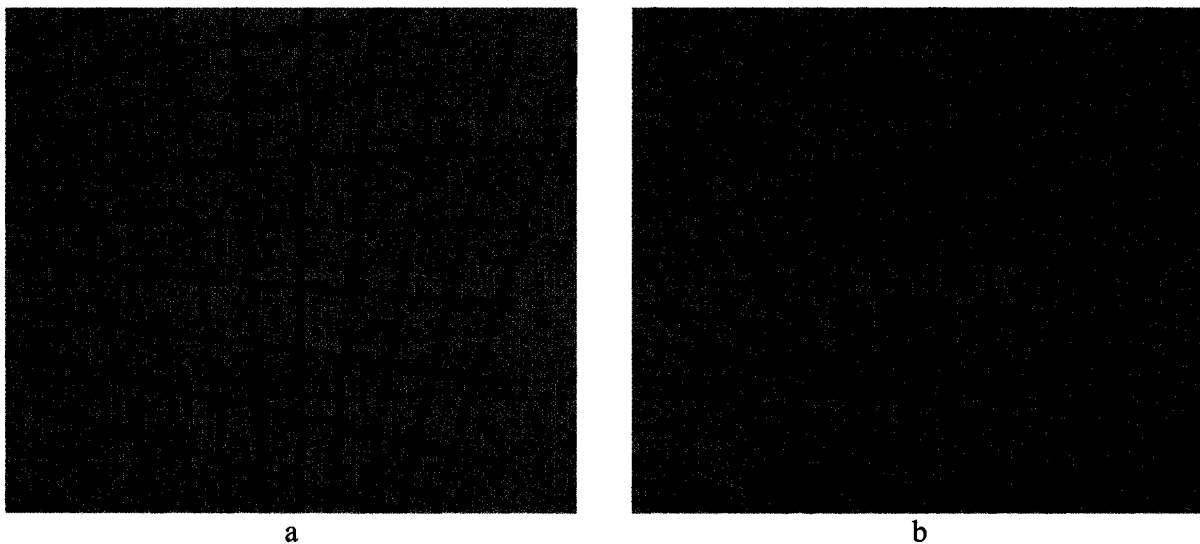


Figure 2-10 : images qui illustrent le fait qu'une région de croisement entre un nombre fixe de points de vue contient des textures différentes : (a) et (b) contiennent respectivement deux et trois lectures avec des pas angulaires de 4 et 3 degrés

Finalement, les faisceaux de lecture correspondent à des arêtes naturelles qui ne délimitent pas nécessairement les objets, mais existent seulement dans les régions explorées par les rayons des capteurs impliqués. Leur intensité dépend de deux facteurs : premièrement de l'occupation de

l'espace qu'ils traversent et deuxièmement de l'erreur qui caractérise le capteur. De plus, les rayons peuvent avoir des directions aléatoires à l'intérieur d'une section de texture donnée.

2.3.3 Analyse de la distance entre les faisceaux adjacents

Cette partie est consacrée à l'analyse de la distance entre deux faisceaux adjacents. Son importance provient du fait qu'elle influence l'applicabilité sur les images probabilistes de la segmentation basée sur l'identification des régions. En effet, dans le cas où des faisceaux d'exploration adjacents ont une occupation similaire, il serait intéressant de considérer la région en question comme étant un segment doté d'une occupation uniforme. L'extrapolation de l'information d'occupation dans les régions inconnues entre les faisceaux de lecture nous permet principalement d'obtenir des segments homogènes en terme d'occupation. Cette caractéristique est à la base de toutes les applications de planification de trajectoire et d'interaction avec l'environnement. Premièrement, la planification de la trajectoire et des mouvements nécessite un espace libre suffisamment spacieux pour que le robot puisse exécuter ses tâches. Deuxièmement, l'interaction avec l'environnement et surtout la manipulation des objets ne peut pas se baser sur des représentations discontinues et présuppose que la forme de ces espaces encombrés soit relativement bien définie. Ces observations ne seraient réalistes que si la distance entre deux faisceaux adjacents est suffisamment petite pour faire en sorte que :

- ❖ Des objets fins ne puissent pas se cacher dans les espaces inexplorés entre une paire de rayons de lecture adjacents (Figure 2-11-a);
- ❖ Deux faisceaux de lecture adjacents dotés d'une probabilité d'occupation supérieure à 50% ne couvrent pas deux objets distincts sans percevoir cette séparation (Figure 2-11-b).

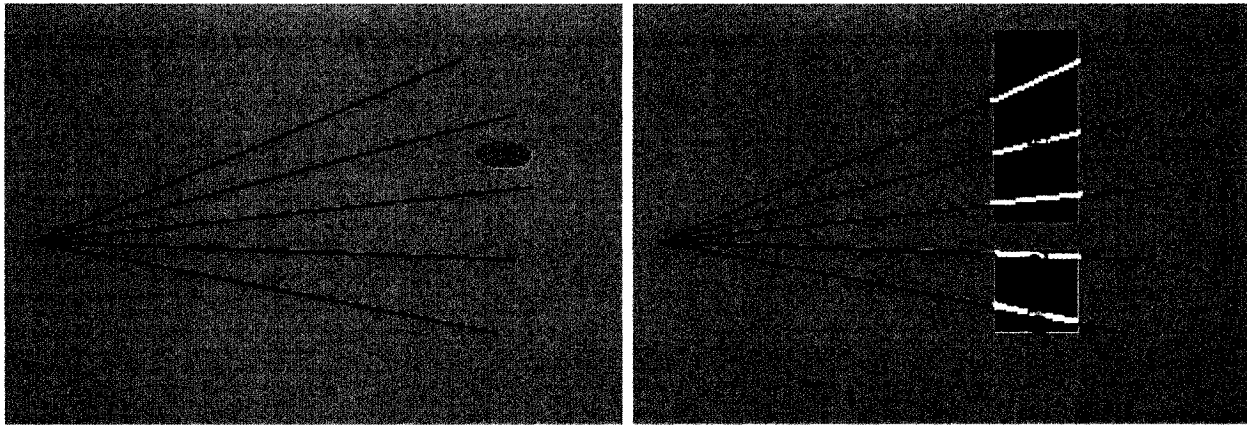


Figure 2-11 : situations particulières qui peuvent entraver et induire en erreur le processus de segmentation : (a) objet non-identifié entre deux faisceaux, (b) faisceaux adjacents couvrant deux objets distincts

La Figure 2-12 illustre le calcul de la distance maximale entre deux rayons adjacents en fonction du pas angulaire α et de la profondeur L que les faisceaux ont atteint.

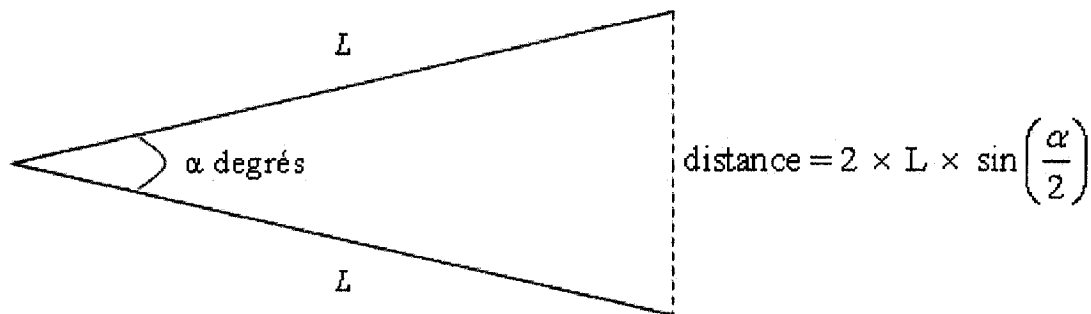


Figure 2-12 : calcul de la distance maximale entre deux faisceaux adjacents en fonction du pas angulaire et de la profondeur atteinte par les rayons

Il est important de noter que le capteur de profondeur Jupiter [10] considéré dans cette étude peut couvrir jusqu'à une profondeur de 1.175m. Selon le mode d'opération choisi, il est capable de mesurer 256 ou 512 échantillons le long d'une ligne de balayage de 57.9 cm, ce qui correspond respectivement à une distance de 0.226 cm ($= 57.9 / 256$) ou 0.113 cm ($= 57.9 / 512$) entre deux faisceaux adjacents. Ces mesures relativement faibles font en sorte que les cas illustrés dans la Figure 2-11 soient peu probables. Par contre, certaines applications nécessitent des prises de mesures à des distances assez élevées (de l'ordre des kilomètres). Même avec une réduction supplémentaire du pas angulaire, la distance entre les faisceaux de lecture demeurerait alors assez élevée. La solution la plus sécuritaire pour ce genre d'applications serait de balayer les espaces critiques sous différents points de vue pour ainsi éliminer les espaces inconnus entre les points balayés au préalable. Dans le cas des

applications qui ont lieu dans un environnement accessible, nécessitant des manipulations précises et une interaction avec l'environnement et afin de réduire la distance entre les faisceaux de lecture, un robot aura avantage à mettre à jour la représentation de l'espace dont il dispose une fois qu'il est proche de sa cible. Une approche similaire peut être adoptée dans le cas de la planification de la trajectoire, où une grille peut garder un tracé approximatif du chemin à suivre, tandis qu'une autre qui se base sur des lectures prises à faible distance, tâche de contourner les obstacles inattendus qui surviennent en cours de déplacement.

2.4 Revue des approches de la segmentation

Les algorithmes de segmentation tentent de classer les pixels d'une image en se basant sur les propriétés de ces derniers et sur leur relation avec leur entourage. Par conséquent le but de la segmentation est de diviser une image en régions dotées de propriétés homogènes. En se basant sur les caractéristiques de texture identifiées précédemment, cette section examine la pertinence des méthodes de segmentation classiques et modernes dans le contexte d'une application à des modèles d'occupation probabilistes de l'espace pour la navigation autonome de robots mobiles.

Plusieurs approches appliquées à la segmentation des images digitales ont été proposées dans la littérature. Ces méthodes sont basées sur l'identification soit des régions², soit des bordures³, soit sur une combinaison des deux. D'autre part la segmentation est généralement soit supervisée, soit non supervisée. Une segmentation non supervisée est appliquée dans le cas où, *à priori*, aucune information concernant le contenu de l'image ou les textures qui la caractérisent, n'est disponible.

Les images probabilistes qui sont au cœur de ce projet, caractérisent l'environnement en terme de son occupation. Elles sont le résultat de la fusion d'informations issues des lectures émanant d'un ou de plusieurs capteurs. Étant donné que l'environnement traité dans ces images est par hypothèse inconnu, aucune information le caractérisant n'est initialement disponible. De ce fait, le schéma de segmentation non supervisée apparaît plus adéquat pour des images probabilistes créées dans un contexte d'exploration de l'espace.

La segmentation non-supervisée vit le jour au début des années quatre-vingt, mais l'exigence des algorithmes en terme de charge de calcul a freiné son développement, toutefois les méthodes

² « Region-based segmentation ».

³ « Boundery-based segmentation ».

développées à cette époque ont été à la base des techniques modernes. Parmi ces méthodes de segmentation classiques, nous retrouvons celles basées sur : le seuillage [9], l'analyse des arêtes et des contours [9][11], les structures pyramidales chaînées⁴ [12][13][14][15], la division et fusion⁵ [16][17][18] ainsi que quelques algorithmes basés sur les arbres quadratiques [19][20].

Avec l'avènement des calculateurs plus puissants vers le début des années quatre-vingt-dix, des méthodes plus sophistiquées ont été développées, celles-ci impliquent l'utilisation des filtres de Gabor multicanaux [21][22][23][24][25], des ondelettes [26][27][28][29][30][31][32][33][34], des modèles de Markov cachés [35][36], et des dimensions fractales [37].

Les sections suivantes proposent un survol de plusieurs méthodes de segmentation et une évaluation dans le cadre spécifique de l'application considérée. Dans la première section, la segmentation basée sur le seuillage est traitée, la seconde attaque la segmentation basée sur l'analyse des contours, la troisième section développe la segmentation à l'aide des structures pyramidales chaînées. Par la suite, nous exposons la méthode de division et fusion. La cinquième partie aborde la segmentation basée sur l'analyse de la texture à l'aide des filtres de Gabor multicanaux. Finalement, nous examinons la segmentation basée sur l'analyse de la distribution des caractéristiques locales des textures.

2.4.1 Seuillage

La méthode de segmentation basée sur le seuillage se démarque par sa simplicité, elle est adaptée aux images simples qui comportent un nombre limités d'objets caractérisés par une intensité différente. Dans cette section, nous présentons la méthode de segmentation basée sur le seuillage et nous analysons son applicabilité sur le cas des images probabilistes.

2.4.1.1 Principes de base

Comme son nom l'indique, la segmentation basée sur le seuillage tente de diviser une image en régions, chacune caractérisée par un niveau d'intensité différent. Les seuils choisis correspondent aux valeurs limites qui séparent deux intervalles d'intensités. Chaque intervalle est assigné à un segment différent.

⁴ « Pyramid node linking ».

⁵ « Split and merge ».

Dans le cas des images en noir et blanc incluant un seul objet (Figure 2-13-a), un seuil unique peut être fixé manuellement à un niveau d'intensité intermédiaire, tel que 0.5 pour une distribution d'intensités comprises entre 0 et 1 (Figure 2-13-b).

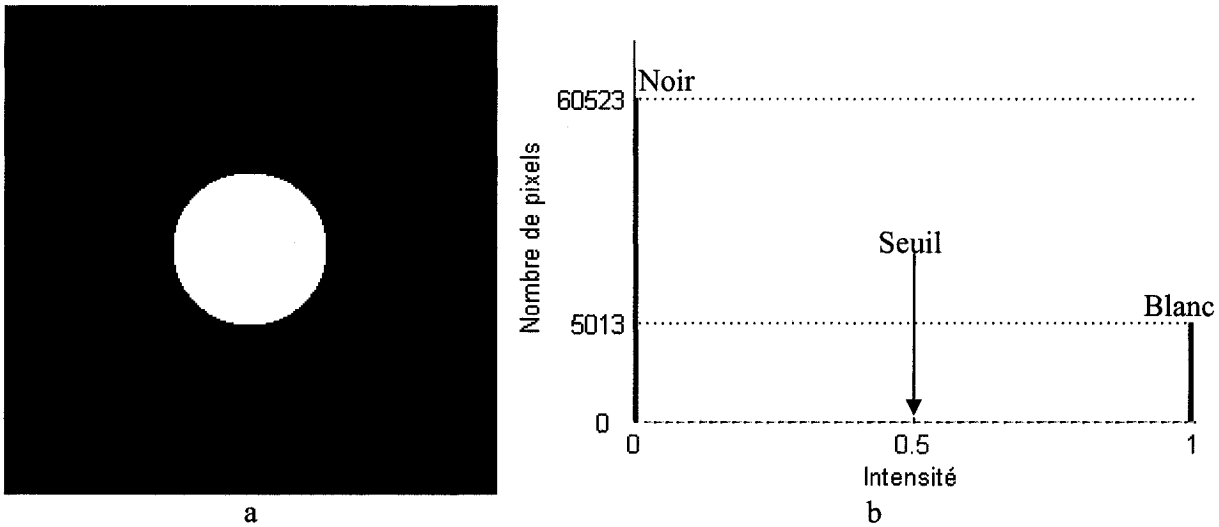


Figure 2-13 : (a) image en noir et blanc incluant un seul objet, (b) histogramme de distribution de l'intensité des pixels de (a)

Afin d'automatiser et de rendre le processus de segmentation plus robuste, les seuils sont fixés en se basant sur l'analyse de l'histogramme caractéristique de l'image (Figure 2-13-b, Figure 2-14-b et Figure 2-15-b). Cet histogramme qui représente le nombre de pixels dans l'image caractérisés par une valeur d'intensité donnée, fournit une vaste gamme d'informations :

- ❖ La largeur et la hauteur de chaque sommet donnent une idée sur la taille relative des objets dans l'image. En effet, les objets dotés d'une intensité homogène sont représentés par un sommet étroit, tandis que ceux caractérisés par une variation d'intensité correspondent aux sommets plus larges;
- ❖ Dans le cas des images incorporant plusieurs objets où chacun est caractérisé par une intensité différente (Figure 2-14), le nombre de sommets présents dans l'histogramme est plus ou moins équivalent au nombre de régions dans l'image.

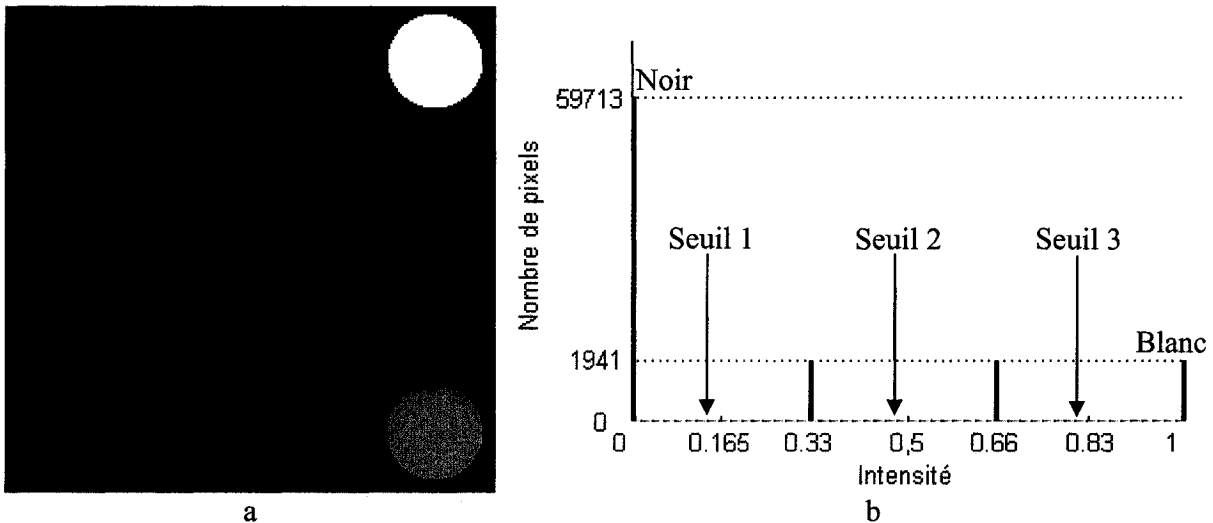


Figure 2-14: (a) image incluant plusieurs objets circulaires caractérisés par des intensités différentes, (b) histogramme de distribution de l'intensité des pixels de (a)

La principale limitation de l'histogramme provient du fait qu'il ne renferme aucune information quand à la localisation spatiale des pixels. Par conséquent, l'assignation d'un seuil unique n'est pas adaptée au cas où plusieurs objets dotés d'une intensité similaire sont présents dans l'image (Figure 2-15).

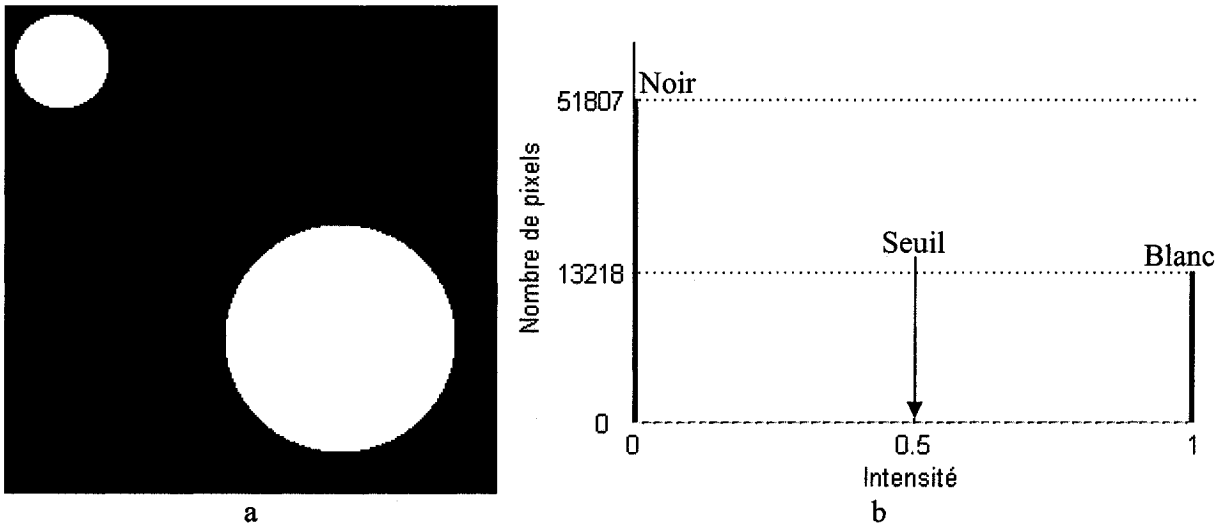


Figure 2-15: (a) image en noir et blanc incluant deux objets circulaires, (b) histogramme de distribution de l'intensité des pixels de (a)

L'analyse d'un histogramme comportant plusieurs sommets consiste à localiser les vallées entre les sommets, puis à assigner aux seuils les valeurs d'intensité de ces vallées. Mais la Figure 2-14 correspond à une simplification de la réalité, car dans le cas des images naturelles ou probabilistes, les pixels composant un segment de l'image ne sont pas tous définis par une valeur

d'intensité unique. La variation de l'intensité est due en partie au modèle d'erreur introduit par le capteur et au diverses opérations de filtrage. Si nous faisons l'hypothèse que les pixels associés à un segment sont affectés par une distribution gaussienne centrée autour d'un niveau d'intensité moyen, alors les divers segments présents dans la scène vont résulter en une superposition de distributions normales, chacune étant caractérisée par une paire de paramètres (μ_i, σ_i) (Figure 2-16). Comme signalé précédemment, l'assignation des seuils correspond à la localisation des vallées. En d'autres termes, pour un intervalle d'intensités donné, il s'agit de trouver les valeurs pour lesquelles le nombre total de pixels qui sont associés à tous les segments intersectés est au minimum. Étant donné le chevauchement entre les distributions normales, la solution à ce problème n'est pas évidente.

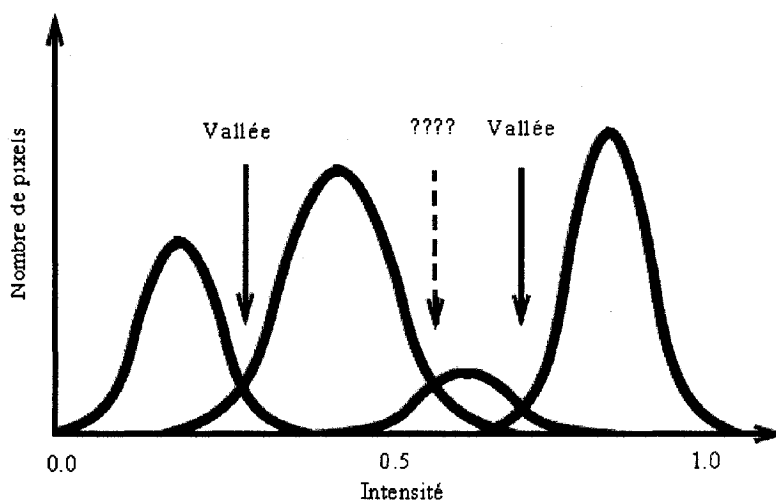


Figure 2-16 : cas où la distribution des intensités dans chaque segment correspond à une distribution gaussienne (reproduite de [9])

2.4.1.2 Applicabilité sur la segmentation d'images probabilistes

La segmentation basée sur le seuillage se caractérise par sa simplicité et par la facilité de son implantation. Mais celle-ci ne localise pas les pixels dans l'image avant de les assigner aux diverses régions. Par conséquent les segments peuvent être spatialement discontinus. La discontinuité spatiale a deux implications dans le cas où ce schéma de segmentation est appliqué aux images probabilistes :

- ❖ Les régions dont l'occupation est connue ne seront jamais uniformes étant donné que les faisceaux de lecture issus du capteur sont séparés par des régions inconnues;
- ❖ Le modèle de segmentation ne permet pas de déceler le nombre ainsi les caractéristiques des objets présents dans l'environnement, tel leur forme ou leur taille.

Les deux points énumérés ci-dessus limitent l'efficacité ainsi que la performance dans certaines applications. En effet la vérification de l'occupation d'une région continue de l'espace implique un nombre très important de mesures de capteurs. De plus, l'interaction avec les objets présents dans l'environnement ne serait pas possible suite à une telle forme de segmentation, étant donné qu'aucune information complète ne peut être extraite sur la forme des objets ainsi que sur leur taille. D'autre part, la mesure éventuelle des propriétés internes des objets, telle leur élasticité, implique une prise de mesure au niveau de chaque région occupée. Comme les espaces occupés ne sont pas spatialement regroupés en objets, un nombre très grand de prises de mesures serait nécessaire.

Afin d'illustrer les propos énumérés ci-dessus, nous avons appliqué le schéma de segmentation basé sur les seuils aux images probabilistes. La Figure 2-17 représente une image probabiliste générée par la combinaison de données issues de quatre points de vue. La Figure 2-18 quand à elle, correspond à l'histogramme de distribution des intensités des pixels de la Figure 2-17. Sur cette figure les deux seuils correspondent aux vallées de part et d'autre du sommet central, leurs valeurs respectives sont 110 et 140. Le résultat de la segmentation qui est représenté dans la Figure 2-19-a, valide les observations précédentes. Dans cette dernière, les pixels dont la valeur est inférieure au premier seuil de l'histogramme, soit 110 (région 1 dans la Figure 2-18) sur une échelle d'intensités comprises dans l'intervalle $[0;255]$, sont dotés d'une occupation *libre* (Figure 2-19-b), ceux dont l'intensité est bornée entre 110 et 140 (région 2 dans la Figure 2-18) sont *inconnus* (Figure 2-19-b), tandis que les pixels dont la valeur est supérieure au second seuil (région 3 dans la Figure 2-18) sont considérés *occupés* (Figure 2-19-b). En effet, comme illustré dans la Figure 2-19-b les segments *libre* (région 1), *occupé* (région 3) et *inconnu* (région 2) sont spatialement non-uniformes, et l'information récoltée par les faisceaux de lecture n'est pas interpolée entre ces derniers. De plus, le nombre et les caractéristiques des objets présents dans l'environnement ne sont pas interceptés car l'espace *occupé* (région 3) est discontinu.

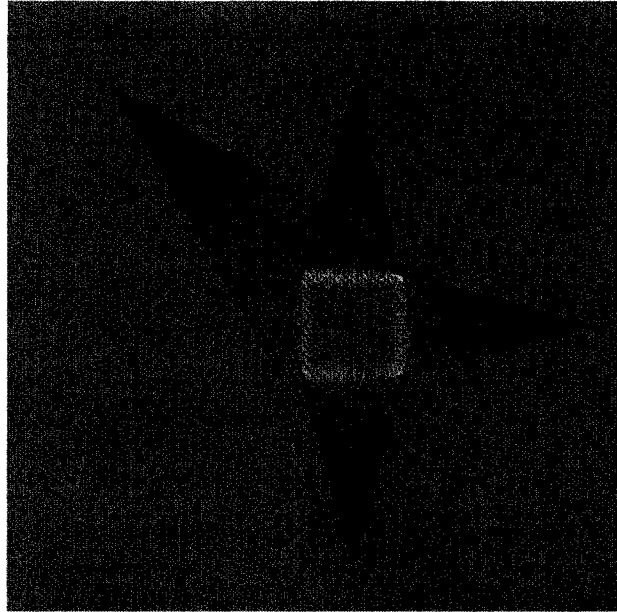


Figure 2-17 : image probabiliste générée par la combinaison de données issues de quatre points de vue

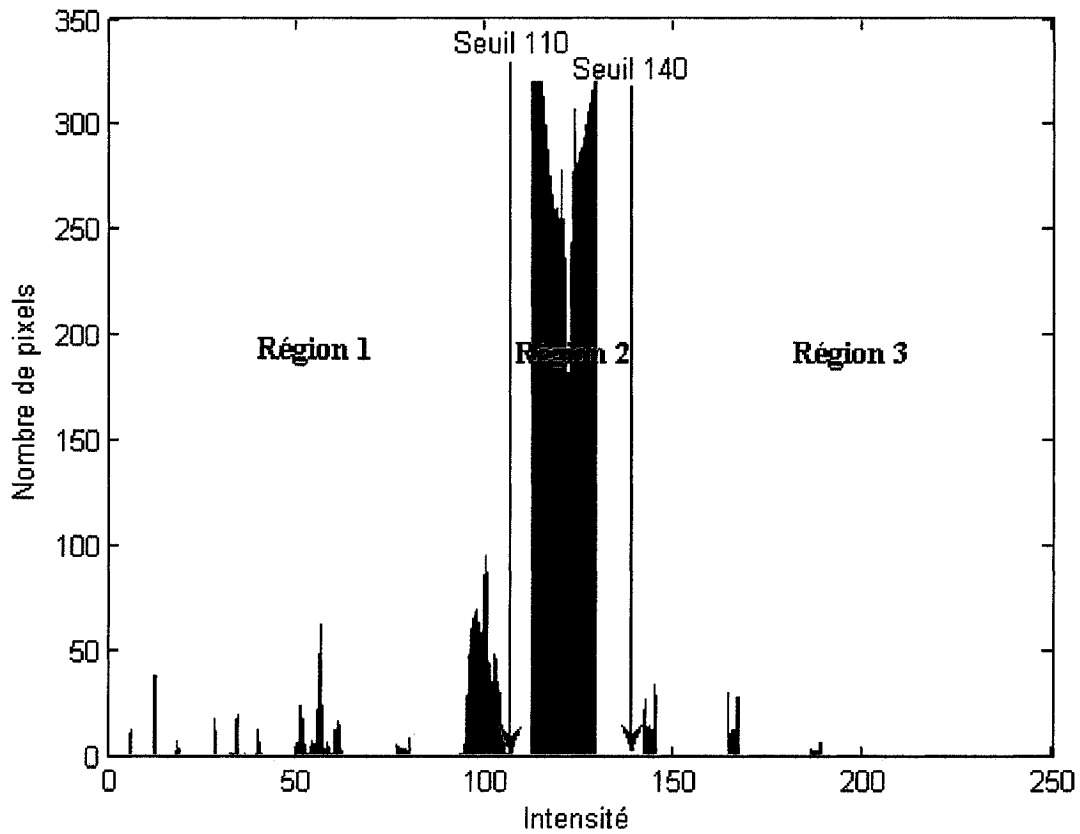


Figure 2-18 : histogramme répartissant les pixels de la Figure 2-17 suivant leur intensité

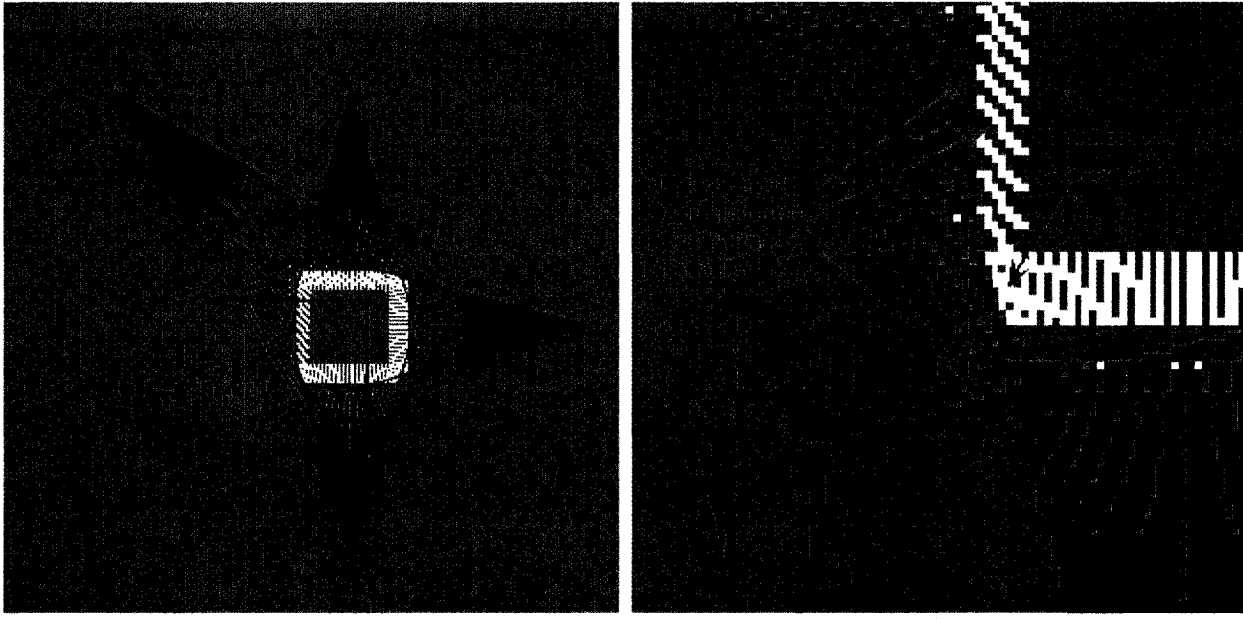


Figure 2-19 : (a) segmentation de la Figure 2-17 en utilisant les seuils présentés dans la Figure 2-18, (b) zoom sur la région rectangulaire et caractéristiques du résultat

Les limitations énumérées précédemment font que la segmentation basée sur le seuillage ne soit pas appropriée pour le traitement des images probabilistes.

2.4.2 Contours

Les limitations inhérentes à la segmentation basée sur le seuillage ont poussé la recherche vers une autre méthode basée sur les contours.

2.4.2.1 Principes de base

Les contours sont définis comme étant une suite logique d'arêtes délimitant une région dotée de caractéristiques semblables. Par conséquent et comme le montre la Figure 2-20, le processus de segmentation introduit dans cette section consiste à déterminer les contours clos entourant les différents segments présents dans une image.

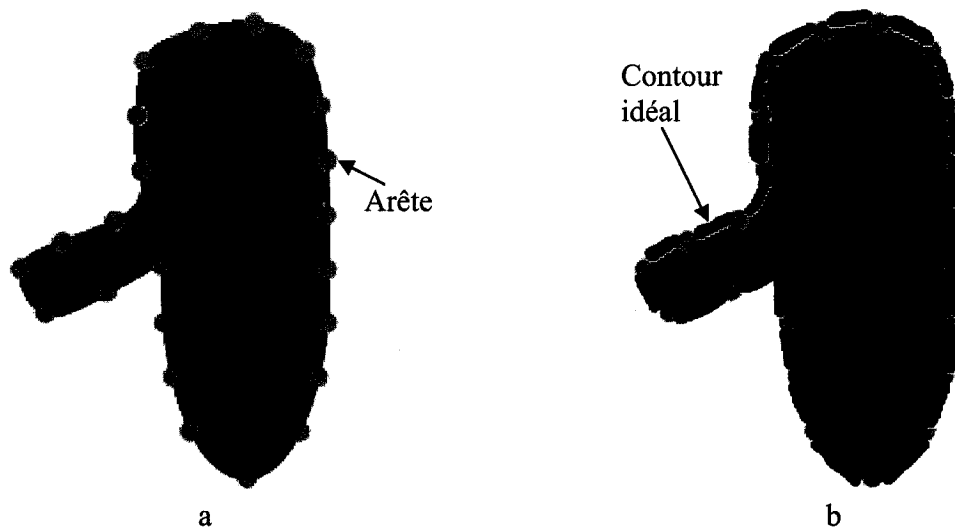


Figure 2-20 : un contour idéal est la liaison entre les arêtes d'un objet

Malheureusement, étant donné qu'une image classique est le résultat d'une interaction entre la géométrie des objets, l'orientation des surfaces, la position de la source de lumière et la position du capteur, il est extrêmement difficile d'obtenir un contour fermé pour tous les objets présents dans la scène.

Toutefois, le fait qu'un segment soit associé à une surface close, implique qu'un processus d'interpolation pourrait être utilisé afin d'obtenir un contour idéal à partir de contours incomplets. Ce processus d'interpolation nécessite la classification des différents éléments de contour comme appartenant à la bordure d'un segment unique. Comme illustré dans la Figure 2-21, diverses caractéristiques comparatives telles la symétrie, le parallélisme ou la courbure peuvent être considérées comme des paramètres de classification entre les éléments de contour.

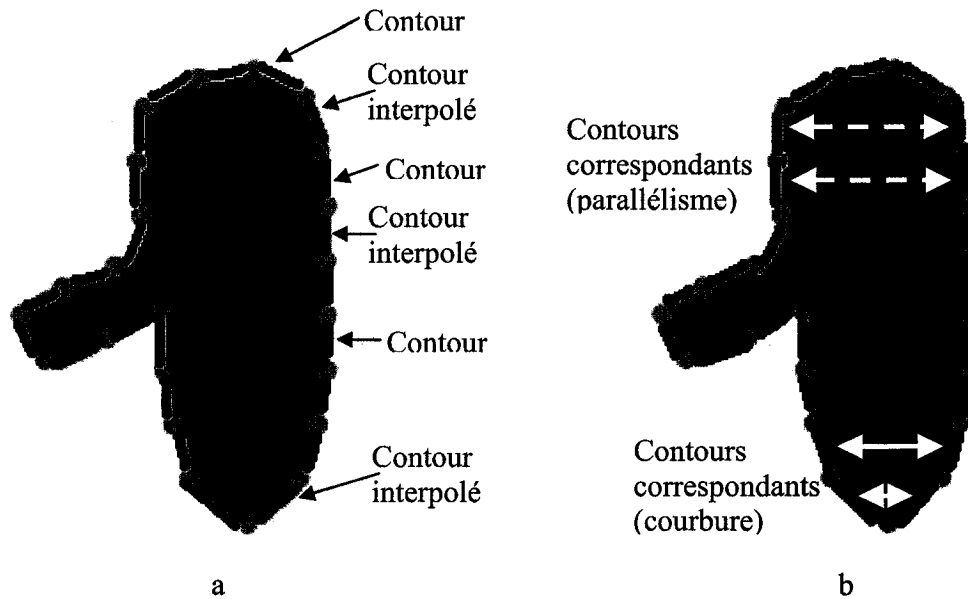


Figure 2-21 : interpolation entre des éléments de contour dans le but d'obtenir un contour clos

Dans le but d'aider à établir la correspondance entre les éléments de contours et les segments, il est usuel de considérer non seulement les caractéristiques des régions en terme d'intensité, mais aussi d'autres propriétés telle que leur forme. Par contre, la prise en compte de données supplémentaires nécessite généralement que des modèles de tous les objets qui peuvent être rencontrés dans la scène soient disponibles dans une base de données. Cette extension implique une segmentation supervisée, et par conséquent elle n'est pas appropriée pour les images probabilistes dans lesquelles aucune information concernant l'environnement n'est *à priori* disponible.

2.4.2.2 Applicabilité sur la segmentation d'images probabilistes

La méthode de segmentation basée sur les contours est fonctionnelle dans le cas où les différentes régions de l'image sont relativement uniformes. Mais dans le cas des images probabilistes, les régions dont l'occupation est connue sont caractérisées par une alternance de faisceaux de lecture et de bandes inconnues (Figure 2-22). Étant donné la différence entre leurs niveaux d'intensité respectifs, un très grand nombre d'arêtes sera détecté. Par conséquent, aucun résultat de segmentation cohérent ne peut être obtenu à l'intérieur des régions traversées par les faisceaux de lecture. Cette limitation, illustrée par le résultat médiocre de l'application de la méthode de Canny [11] (Figure 2-23) qui initie la détection de contours, démontre l'impossibilité d'appliquer la segmentation basée sur les contours sur les images probabilistes.

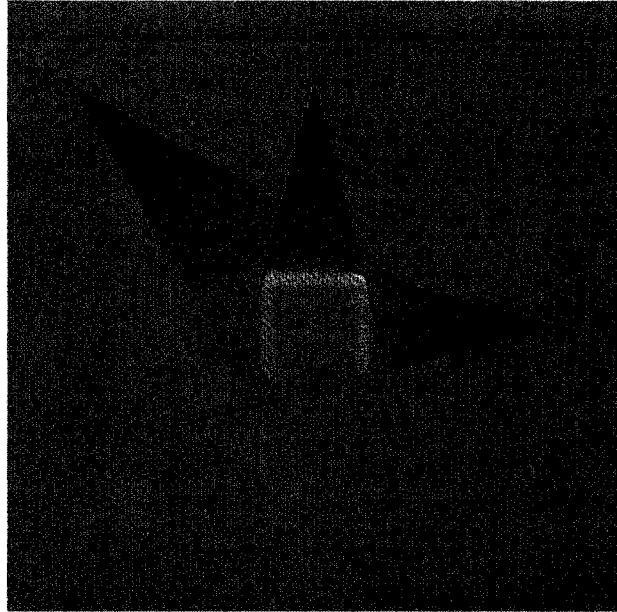


Figure 2-22 : image probabiliste générée par la combinaison de données issues de trois points de vue

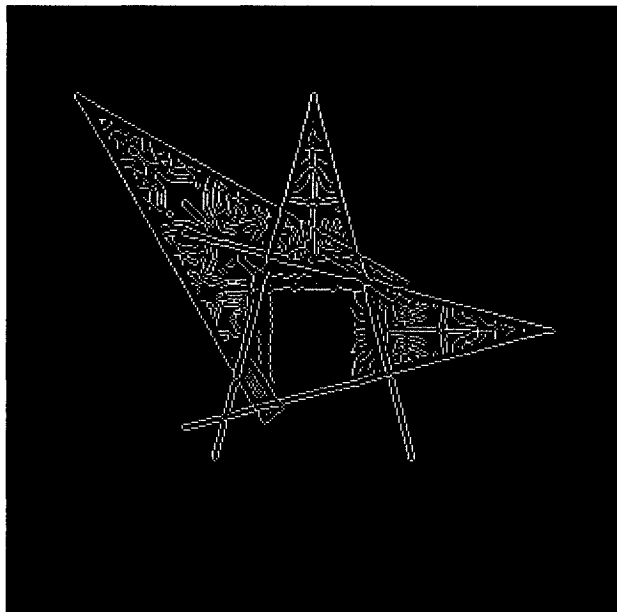


Figure 2-23 : application de la méthode de Canny [11] de détection d'arêtes sur la Figure 2-22

2.4.3 Les structures pyramidales chaînées

La première application des structures pyramidales chaînées sur la segmentation d'images a été introduite par Burt *et al.* [12] en 1981. L'année suivante, soit en 1982, divers tests et expérimentations de cette nouvelle méthode de segmentation ont été réalisés, mais la publication de

Hong *et al.* [13] expose, à travers les résultats qu'elle présente, le caractère fragile de la segmentation basée sur les structures pyramidales chaînées. L'investigation de cette méthode fut très timide par la suite. L'apport de la recherche aux structures pyramidales chaînées est aussi demeuré minime, et se caractérise par son caractère relativement empirique tel qu'illustré dans la publication de Ping *et al.* [14]. Toutefois, plusieurs applications se sont basées sur les structures pyramidales, surtout dans le domaine biomédical [15].

Cette section résume le principe de la segmentation basée sur la structure pyramidale chaînée tel que présenté par Burt *et al.* [12]. Par la suite, certaines variations de cet algorithme seront exposées. Finalement, une brève analyse évalue l'applicabilité de cette approche de segmentation sur le cas des images probabilistes.

2.4.3.1 Méthodes de segmentation

La méthode de segmentation à l'aide des structures pyramidales chaînées a été principalement développée pour contourner les problèmes rencontrés par celles qui se basent sur des fenêtres⁶. Dans ces méthodes, la taille de la fenêtre est relativement large par rapport aux caractéristiques d'une image. Cette taille excessive fait en sorte que les segments de taille réduite ainsi que les bordures entre certains segments, ne soient pas interceptés. Étant donné que la taille des segments ainsi que la nature des caractéristiques de l'image sont *à priori* inconnues, l'utilisation des fenêtres n'est pas adaptées à la segmentation non-supervisée.

Une pyramide est constituée d'un ensemble de matrices bidimensionnelles dont la taille est réduite d'un facteur de deux en passant d'un niveau donné à son supérieur. Les nœuds du niveau le plus bas, caractérisés par l'indice zéro (Figure 2-24), correspondent aux pixels de l'image à segmenter. Ainsi, pour une image initiale de taille $[2^n \times 2^n]$, le niveau n correspond au sommet de la pyramide et est formé d'un seul nœud (Figure 2-24).

⁶ « Windowing Techniques ».

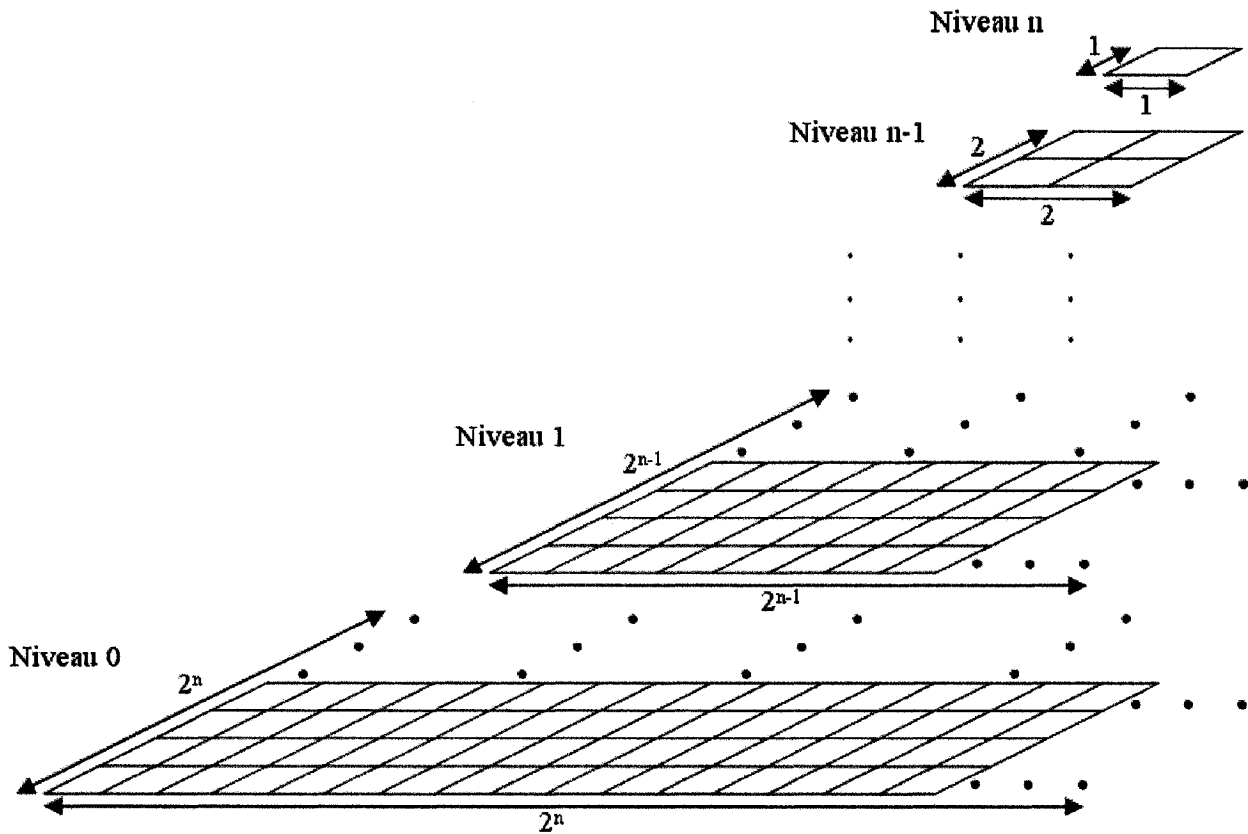


Figure 2-24 : organisation de la pyramide dans les « structures pyramidales chaînées »

Une relation de « parent – enfant » est définie entre les nœuds de niveaux adjacents, cette relation de parenté n'est pas fixe et est redéfinie d'une itération à l'autre. En effet, comme illustré dans la Figure 2-25, un nœud situé au niveau l est « candidat parent » à seize nœuds situés au niveau $l-1$. De plus, il est lui-même « enfant candidat » pour quatre nœuds au niveau $l+1$. Chaque itération assigne les nœuds à un de leur quatre « candidats parents ». Par conséquent chaque nœud situé à un niveau $l > 0$, peut avoir entre 0 et 16 « enfants légitimes », la borne inférieure (supérieure) correspond au cas où un lien s'est établi avec aucun (chacun) de ses seize « enfants candidats ». Les relations successives et « légitimes » de parenté définissent les segments de l'image. Ainsi un segment est assigné à chaque « parent légitime » N situé à un niveau l . Ce segment correspond à l'ensemble des pixels de l'image constituant la « descendance » de N . Logiquement, plus l est élevé, plus la taille du segment est grande.

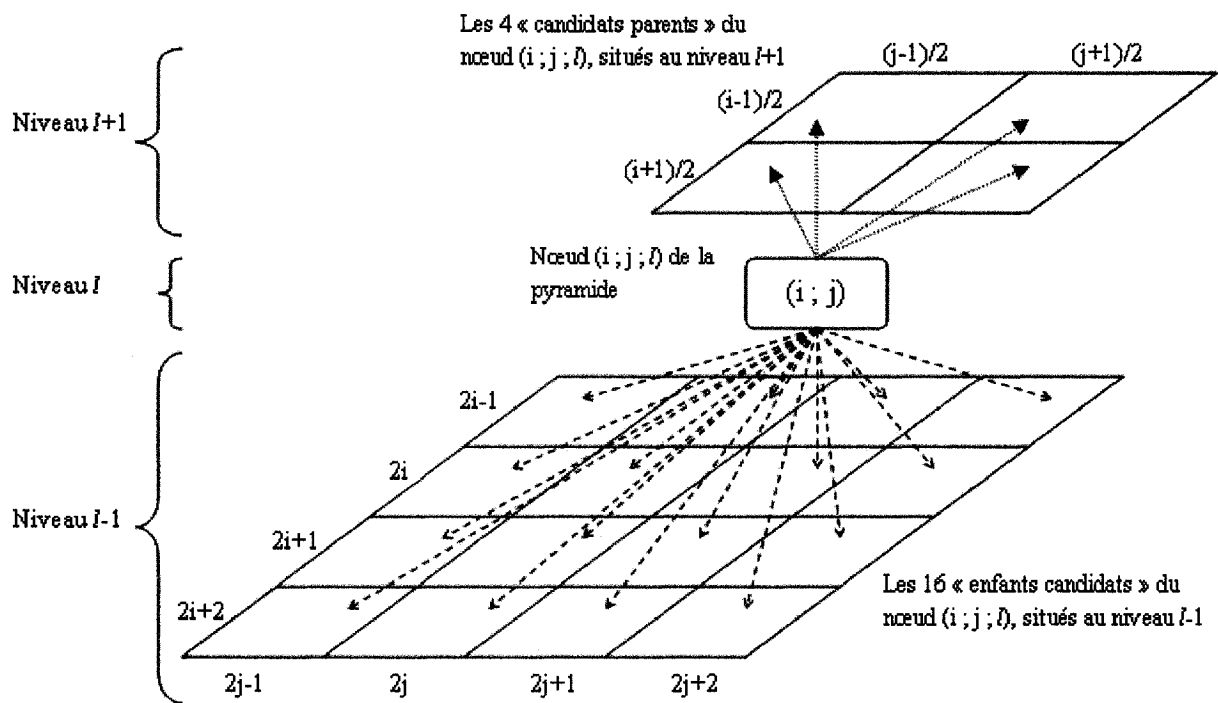


Figure 2-25 : « enfants » et « parents » candidats pour un nœud $(i; j; l)$ donné

D'autre part, une quantité variable appelée « propriété » est calculée au niveau de chaque nœud de la pyramide. Pour un nœud donné, elle est égale à la moyenne des valeurs de ses « enfants », en considérant que la « propriété » des nœuds de niveau zéro correspond à la valeur des pixels.

Pour un nœud N donné, son « parent légitime » est celui, parmi les quatre « candidats parents » possibles, dont la « propriété » est la plus proche de celle de N . Le choix d'un parent unique implique que si nous considérons un niveau l donné, chaque nœud du niveau zéro, ou en d'autres termes chaque pixel de l'image, est lié via des nœuds intermédiaires à un seul nœud au niveau considéré l . De ce fait le nœud situé au niveau le plus élevé peut être vu comme un élément central pour plusieurs structures en forme d'arbres, celles-ci partitionnant l'image en segments dotés d'une forme et d'une taille variables.

Hong *et al.* [13] proposent une série de variations. La principale d'entre elles consiste à limiter au nombre des régions présentes dans l'image, le nombre de nœuds situés aux niveaux les plus élevés sur lesquels les arbres de segmentation sont centrés. Mais cette extension qui implique une segmentation supervisée n'est pas adaptée au cas des images probabilistes étant donné que ces

dernières représentent des environnements réels incorporant un nombre initialement inconnu de régions. De plus, le schéma proposé par Hong *et al.* [13] est caractérisé par son incapacité à intercepter les segments de taille relativement grande et celles de taille réduite.

D'autre part, Ping *et al.* [14] soulignent que l'initialisation (établissement des liens « candidat parent » et « parent légitime ») d'un certain niveau dans l'algorithme de Burt *et al.* [12] se fait par le filtrage du niveau d'ordre directement inférieur. Le filtre utilisé est normalisé, de taille [4 x 4] et doté de valeurs unitaires. Ping *et al.* [14] proposent le remplacement de ce filtre par une fonction gaussienne qui garantit que les caractéristiques basses fréquences, généralement prédominantes dans les images, soient maintenues d'un niveau à l'autre.

2.4.3.2 Applicabilité sur la segmentation d'images probabilistes

La méthode de segmentation classique proposée Burt *et al.* [12] souffre, tout comme ses variations, de certaines faiblesses qui remettent en cause leur compatibilité avec les images probabilistes.

En premier lieu, ces méthodes sont partiellement supervisées, car le nombre de segments dans l'image dépend du nombre de nœuds aux niveaux supérieurs considérés et non pas des vrais caractéristiques des régions de l'image.

En second lieu, les segments obtenus à la fin de ce processus peuvent souffrir d'une discontinuité spatiale. Ceci dit que si deux espaces occupés non adjacents sont présents dans une image probabiliste, à cause du fait que leur représentation est similaire, ils seront assimilés à un même segment. La discontinuité spatiale limite la segmentation des images probabilistes à deux niveaux :

- ❖ Chaque objet dans l'environnement pourrait avoir des propriétés internes différentes de celles des autres. L'identification de tous les objets dans un seul segment implique un manque d'information qui est indésirable dans le cas de certaines applications. À titre d'exemple, dans le cas d'une application de planification de trajectoire dans un environnement contenant un ou plusieurs objets, l'identification de chacun d'entre eux serait primordiale afin que le chemin suivi par le robot mobile soit sécuritaire;
- ❖ La discontinuité spatiale implique que l'espace situé entre les rayons de mesure sera segmenté comme *inconnu*, ce qui fait qu'aucun des segments caractérisant les espaces

libres et les espaces *occupés* ne sera spatialement continu. La discontinuité de ces espaces est indésirable dans le cas où des manipulations dans l'environnement de mesure sont nécessaires.

Finalement, aucune des variations proposées par Hong *et al.* [13] et par Ping *et al.* [14] ne permet de résoudre les problèmes énumérés ci-dessus. De plus, l'utilisation d'un filtre gaussien [14] dans l'initialisation des niveaux n'est pas adaptée aux images probabilistes qui sont riches en composantes hautes fréquences. Ces composantes sont dues aux faisceaux de lecture émis par les capteurs (section 2.3.1), et représentent l'espace dont l'occupation est connue et qui doit être classifié convenablement.

2.4.4 La division et fusion

Les méthodes de segmentation d'images basées sur le concept de la division et fusion ont été parmi les premières à voir le jour et surtout parmi celles qui ont le plus influencé les méthodes de segmentation modernes. Dans cette partie, nous fournissons une vue globale de l'algorithme de division et fusion classique de Pavladis [16]. Par la suite, nous présentons deux variations proposées dans la littérature, soient celle de Wu [17] et celle de Yang et Lee [18]. Finalement, l'applicabilité des algorithmes de division et fusion sur les images probabilistes est analysée.

2.4.4.1 Méthodes de segmentation

L'algorithme développé par Pavladis [16] et basé sur le concept de la division et fusion, se divise en deux phases. La première est appelée « division » et consiste à subdiviser récursivement une image en blocs de taille égale jusqu'à ce qu'un critère d'homogénéité soit satisfait. Au départ, un test d'homogénéité est appliqué sur une image, celle-ci est à la base considérée comme une région unique. Si le résultat de ce test est insatisfaisant, la région est divisée en quatre sous-régions selon la structure des arbres quadratiques⁷ [19]. Le même processus de division est appliqué sur chacune des sous-régions obtenues, et ainsi de suite. La seconde phase, soit la « fusion », consiste à comparer en paires les régions adjacentes. Si leur union satisfait le même critère d'homogénéité utilisé dans la première phase, les deux régions sont fusionnées. Il est évident que quatre sous-régions adjacentes issues de la division d'une région parent ne peuvent pas être re-fusionnées, étant donné que c'est le

⁷ « Quadtree ».

même critère d'homogénéité qui contrôle les deux phases de l'algorithme, qui a décidé de leur division. N'empêche que trois des quatre sous-régions peuvent être unifiées si elles sont homogènes et seule la quatrième présente certaines hétérogénéités. Les critères d'arrêt tant pour la phase de « division » que pour la « fusion » n'ont pas été définis par Pavladis [16] étant donné que ces derniers dépendent de la nature de l'application.

Le désavantage majeur de cette méthode provient de son incapacité à s'adapter à la sémantique de l'image. En effet, les arêtes des segments générés par cet algorithme peuvent avoir deux orientations, soit horizontale, soit verticale, et doivent être positionnées sur les bordures d'un segment tel qu'identifié par un nœud de l'arbre quadratique. Ces limitations font qu'il est souvent impossible de faire correspondre les limites rectilignes préétablies et fixes des segments aux bordures des objets qui peuvent avoir n'importe quelle orientation et position.

Wu [17] constate qu'une segmentation réussie devrait établir un lien bidirectionnel entre les segments et les arêtes. En d'autres termes les bordures des segments doivent correspondre aux arêtes des objets. Les extensions proposées par Wu touchent principalement la phase de « division » et permettent de subdiviser une région non uniforme en deux blocs séparés par une arête à inclinaison variable. Le choix de la position de l'arête de coupure ainsi que de son inclinaison conduit à un nombre infini de bipartitions, chacune correspondant à un couple d'angle et de position possibles. Wu attaque ce problème en limitant le nombre d'angles à quatre valeurs, soient 0, 45, 90 et 135 degrés. L'inclinaison choisie est celle qui minimise la somme des erreurs quadratiques relatives aux deux subdivisions obtenues. Mais Wu ne fournit aucune solution quant à la position de l'arête de coupure. Dans ce schéma de segmentation, la récursivité est brisée une fois que l'erreur quadratique moyenne (équation (2.1)) d'un segment S_i est inférieure à un certain seuil strictement positif ε . Étant donné que l'erreur quadratique moyenne d'une région formée d'un seul pixel est nulle, la condition d'arrêt sera à ce niveau naturellement forcée.

$$d(S_i) = \frac{1}{|S_i|} \sum_{(x,y) \in S_i} [I(x,y) - \mu(S_i)]^2 \quad (2.1)$$

Où : $I(x, y)$: l'intensité du pixel de coordonnées (x, y) .

S_i : la région i de l'image.

$\mu(S_i)$: la moyenne de l'intensité des pixels dans la région S_i .

Bien que le travail de Wu [17] a permis de mieux adapter la forme des segments à celle des objets présents dans l'image, son approche ne permet pas de reproduire les bordures dotées d'orientations aléatoires. Yang et Lee [18] tentent, en reformulant la phase de « division », de résoudre ce problème.

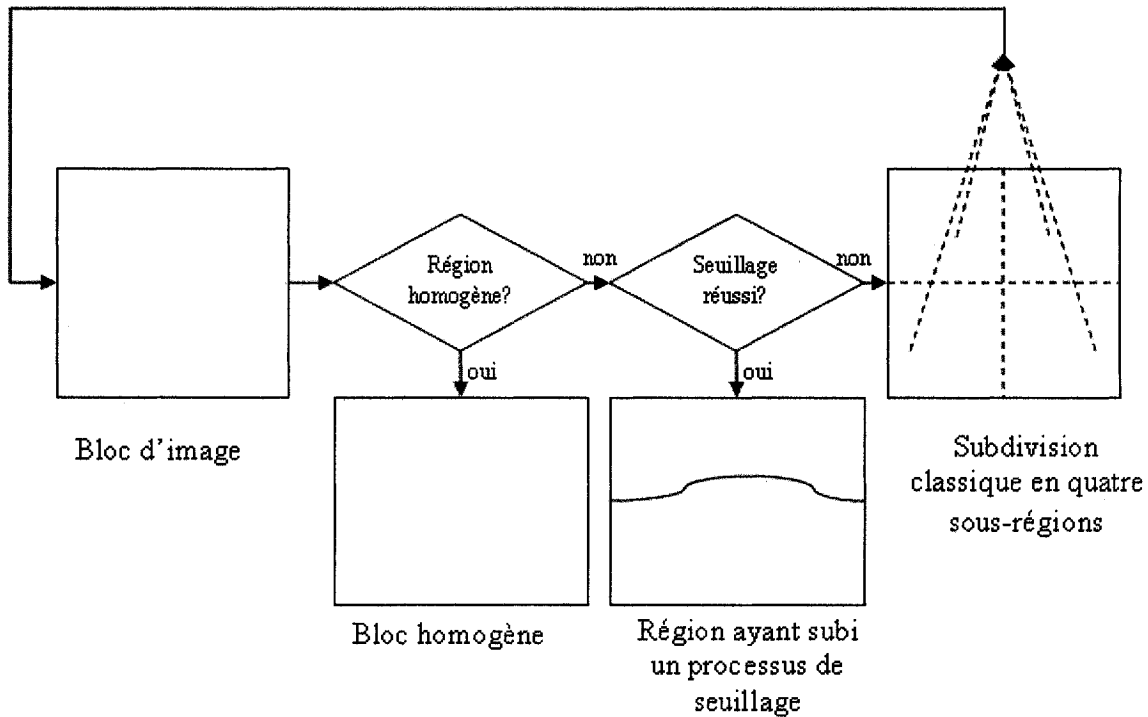


Figure 2-26 : diagramme en bloc de la « division » proposée par Yang et Lee [18] (reproduite de [18])

La Figure 2-26 présente le diagramme en bloc de la phase de « division » telle que proposée par Yang et Lee [18]. Dans cette figure, nous pouvons identifier les trois types de régions qui peuvent être générées durant la phase de « division », soit les régions homogènes, celles ayant subi un processus de seuillage et celles subdivisées en quatre sous-régions. Considérons un bloc extrait d'une image quelconque, la première étape de la phase de la « division » [18] consiste à le soumettre à un test d'homogénéité. S'il le réussit, il sera classifié comme homogène et ne subira aucune subdivision. Un bloc non-homogène passe à travers un processus de seuillage. Si une subdivision issue de ce processus est considérée uniforme, elle n'est plus subdivisée et est classifiée comme

ayant subi un processus de seuillage. Sinon elle est subdivisée en quatre sous-régions, et le processus de division s'applique itérativement sur chacune d'elles.

2.4.4.2 Applicabilité sur la segmentation d'images probabilistes

Les trois algorithmes de segmentation basés sur la méthode division et fusion souffrent de contraintes qui les rendent inapplicables sur le cas des images probabilistes considérées dans cette étude. Ainsi, la structure des espaces *occupés* et *libres* dans les images probabilistes introduit plusieurs défis qui sont insurmontables par les algorithmes de « division et fusion » décrits ci-dessus. Premièrement, ces régions ne sont pas uniformes étant donné que les espaces de certitude sont limités aux *proxels* qui ont été parcourus par un rayon issu d'une lecture de capteur. Par conséquent, la segmentation par ces algorithmes tend à identifier en région différente chaque faisceau de lecture et ne peut pas extrapoler l'information d'occupation entre les rayons adjacents. Deuxièmement, les images probabilistes correspondent à des représentations de l'environnement réel. Subséquemment les objets rencontrés ont une forme quelconque et leurs bordures ne suivent toujours pas une inclinaison prédéterminée par rapport à celle du capteur et de l'image. De ce fait, les schémas de Pavladis [16] et de Wu [17] n'arrivent pas à segmenter correctement les modèles probabilistes. Finalement, le processus de seuillage introduit par Yang et Lee [18] ne résout pas les problèmes mentionnés ci-dessus et entraîne un questionnement supplémentaire concernant la valeur et le nombre de seuils à utiliser, ce qui implique une intervention de l'opérateur. En effet le choix d'un seuil unique [18] introduit des difficultés qui résultent de l'existence de l'espace inconnu dans les images probabilistes. Par conséquent, l'utilisation d'un seuil unique dans la phase de la « division » classifie cet espace soit comme libre soit comme occupé, ce qui est incorrect.

2.4.5 Les filtres de Gabor multicanaux

Les méthodes de segmentation introduites précédemment ne fournissent pas des résultats stables dans le cas des images probabilistes. Cette limitation découle entre autre de leur incapacité à interpoler les informations qui caractérisent l'occupation de l'environnement entre les faisceaux de lecture. Les filtres de Gabor appliqués à différentes fréquences radiales et orientations permettent d'étendre ce concept en se basant sur les différences entre les arrangements spatiaux des pixels avoisinants, en d'autres termes sur la différence de texture entre les régions d'une image. L'intérêt de l'application de la segmentation basée sur l'analyse de la texture dans le cas des images

probabilistes découle de la structure même des régions qui ont été déchiffrées par une lecture issue d'un capteur à une position et orientation données.

Une définition universelle de la texture est impossible, vu la multitude de textures naturelles et artificielles existantes. Mais plusieurs méthodes permettent de détecter et de comparer les textures présentes dans une image en se basant sur des critères objectifs. Dans cette partie, nous donnons une vue d'ensemble sur l'analyse de la texture basée sur le filtrage multi-canal à l'aide de la fonction de Gabor. L'intérêt du filtrage multi-canal découle du fait que les composantes spatiales et fréquentielles de chacune des textures présentes dans une image sont différentes. Par conséquent, cette procédure permet de décomposer l'image originale en sous-images chacune dotée d'informations spectrales limitées et donc d'une texture différente.

2.4.5.1 Choix du filtre de Gabor et de ses paramètres

La fonction de Gabor qui a été introduite en 1946 par Dennis Gabor [21] décrit simultanément le comportement temporel et spectral d'un signal donné. Les filtres de Gabor que nous retrouvons dans la littérature peuvent prendre diverses formes [22][23][24][25].

Le filtrage d'une certaine image à l'aide d'une fonction de Gabor se fait à l'aide d'une opération de convolution dans le domaine spatial ou d'une multiplication dans le domaine fréquentiel. Les filtres Gabor utilisés dans un système multi-canal diffèrent par leur fréquence radiale et par leur orientation. Par conséquent la sortie de ce système pour une image d'entrée donnée correspond à une pile d'images, chacune définie à la fréquence et à l'orientation dictées par la fonction de Gabor du canal correspondant.

Les filtres de Gabor utilisés dans la littérature détectent les caractéristiques suivant quatre valeurs d'orientation, soit 0, 45, 90 et 135 degrés. Les fréquences radiales sont, quant à elles, fonction de la taille de la région sur laquelle le filtre est appliqué. Pour une région de taille $[N \times N]$, les fréquences radiales utilisées sont les suivantes :

$$1\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, \dots, \left(\frac{N}{4}\right)\sqrt{2} \text{ cycles / (unité de largeur}^8\text{)}$$

La valeur de N dépend de la structure du système de filtrage. Jain et Farrokhnia [22] appliquent un filtrage dans le domaine fréquentiel sur l'ensemble de l'image d'entrée. Tandis que Mital *et al.*

⁸ « Picture width ».

[23][24] optent pour un système similaire à celui de la transformée en cosinus discrète⁹; ils divisent l'image en blocs de taille $[32 \times 32]$ et filtrent séparément chacun de ces derniers. Le choix d'une valeur de N égale à 32 implique quatre valeurs de fréquences radiales soit : $1\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}$ cycles / (unité de largeur). À noter que Jain et Farrokhnia [22] n'utilisent pas les valeurs des fréquences radiales faibles ($1\sqrt{2}, 2\sqrt{2}$ cycles / unité de largeur) dans leur système multi-canal étant donné que ces dernières détectent des variations spatiales très larges pour correspondre à des caractéristiques de texture. De plus et afin de s'assurer que les filtres ne répondent pas à des régions dotées d'intensité constante, la composante DC des filtres de Gabor utilisés dans les systèmes multicanaux est forcée à zéro.

2.4.5.2 Extraction des images caractéristiques

Tel que représenté dans la Figure 2-27, l'extraction des images caractéristiques a lieu dans chaque canal, juste après l'application des filtres de Gabor sur l'image originale.

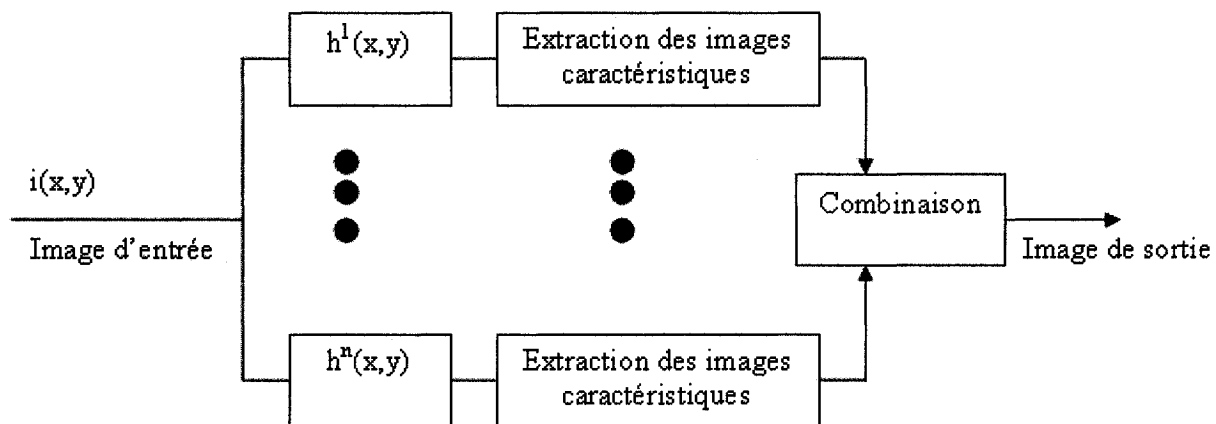


Figure 2-27 : diagramme en bloc caractérisant la segmentation multi-canal

Jain et Farrokhnia [22] considèrent que l'utilisation d'un sous-ensemble A des images filtrées peut réduire le temps d'exécution des étapes subséquentes de l'algorithme de segmentation. Ainsi, A correspond au plus petit ensemble qui conduit à un coefficient de corrélation R^2 [22], avec l'image originale, supérieur à 0.95, soit 95%. Par la suite, le calcul des images caractéristiques pour chacun des éléments de A se fait par un processus à deux étapes. En premier lieu, une opération non-linéaire [22] est appliquée sur chacun des pixels des images filtrées. En second lieu, les images

⁹ « Discrete Cosine Transform ».

caractéristiques $e_k(x, y)$ correspondantes aux images filtrées $r_k(x, y)$, tel que $k \in A$, sont calculées à l'aide de l'équation suivante :

$$e_k(x, y) = \frac{1}{M^2} \sum_{(a,b) \in W_{x,y}} |\Psi[r_k(a, b)]| \quad (2.2)$$

Où : Ψ : fonction non-linéaire.

e_k : image caractéristique d'indice k , tel que $k \in A$

r_k : image filtrée d'indice k , tel que $k \in A$

$W_{x,y}$: une fenêtre de taille $[M \times M]$ centrée sur le pixel de coordonnées (x, y) .

La taille M de la fenêtre dans laquelle la moyenne est calculée est un paramètre primordial. Des mesures plus efficaces des caractéristiques de la texture requièrent une valeur relativement large, tandis qu'une meilleure localisation des bordures des régions requiert une valeur relativement faible.

Le schéma proposé par Mital *et al.* [23][24] ne procède pas à l'élimination de certaines images filtrées. Les images caractéristiques $e_i(x, y)$ sont extraites directement des images filtrées $r_i(x, y)$ à l'aide de l'équation suivante :

$$e_i(x, y) = \frac{1}{M^2} \sum_{(a,b) \in W_{x,y}} |r_i(a, b)| \quad (2.3)$$

Où : e_i : image caractéristique d'indice i , tel que $i \in [1;16]$.

r_i : image filtrée d'indice i , tel que $i \in [1;16]$.

$W_{x,y}$: une fenêtre de taille $[M \times M]$ centrée sur le pixel de coordonnées (x, y) .

Dans l'équation (2.3), l'indice i du canal dans lequel le traitement a lieu est borné entre 1 et 16. Le nombre de canaux égal à 16 provient du fait que quatre orientations sont choisies, à chacune sont associées quatre fréquences radiales. D'autre part, la taille M de la fenêtre sur laquelle le processus d'extraction des caractéristiques se base n'est pas définie dans la littérature. Son choix demeure

empirique. Enfin, aucune des publications mentionnées dans cette section ne fournit de détails précis quant à la phase de combinaison à partir des images caractéristiques (Figure 2-27).

2.4.5.3 Applicabilité sur la segmentation d'images probabilistes

La segmentation basée sur les filtres de Gabor multicanaux combine les points forts du traitement multicanal et des filtres de Gabor. Elle permet de segmenter une image suivant les textures qui la caractérisent. La sélection des textures se base sur leur orientation ainsi que sur leur niveau de détails. Mais l'application de cette méthode sur les images probabilistes conduit à la création d'une multitude de segments dans l'espace libre et dans l'espace occupé étant donné que ces espaces sont caractérisés par une alternance de faisceaux de lectures et de régions inconnues. Chacun des faisceaux est doté d'une orientation particulière, et par conséquent sera segmenté en une région différente.

D'autre part, les capteurs desquels sont issues les lectures dans les images probabilistes, n'ont pas une position et une orientation fixes. Ceci dit que la bande passante en orientation des filtres de Gabor choisis doit être bien inférieure à 45 degrés, et pour un degré de précision relativement élevé une valeur de 5 degrés doit être choisie [22]. Cette précision se fait au détriment du nombre de canaux. En effet celui-ci est multiplié par un facteur de 9 par rapport aux implantations présentées précédemment :

$$\begin{aligned} nc_{45} &= no_{45} \times nfr \\ nc_5 &= no_5 \times nfr = (9no_{45}) \times nfr = 9 \times (nc_{45}) \end{aligned} \quad (2.4)$$

Où : nc_i : nombre de canaux étant donné une bande passante en orientation i degrés.

no_i : nombre d'orientations de i degrés.

nfr : nombre de fréquences radiales choisies.

Chaque canal incorpore une opération de filtrage qui correspond soit à une convolution dans le domaine spatial, soit à une multiplication suivie d'une transformée de Fourier inverse dans le domaine fréquentiel. Plus le nombre de canaux est élevé, plus d'opérations de filtrages sont requises et plus le processus de combinaison des images caractéristiques est chargé. En d'autres termes, l'augmentation du nombre de canaux par un facteur de 9 pour atteindre une résolution angulaire

adéquate conduit à une charge d'exécution très importante et par conséquent constitue un point faible de la méthode des filtres de Gabor.

Les points énumérés ci-dessus sont les principales causes des doutes que nous avons quand à l'applicabilité de la méthode de segmentation multi-canal à l'aide de filtres de Gabor. Néanmoins, l'investigation de cette méthode a suscité notre intérêt de recherche vers les méthodes de segmentation basées sur l'analyse de la texture.

2.4.6 La transformée en ondelettes

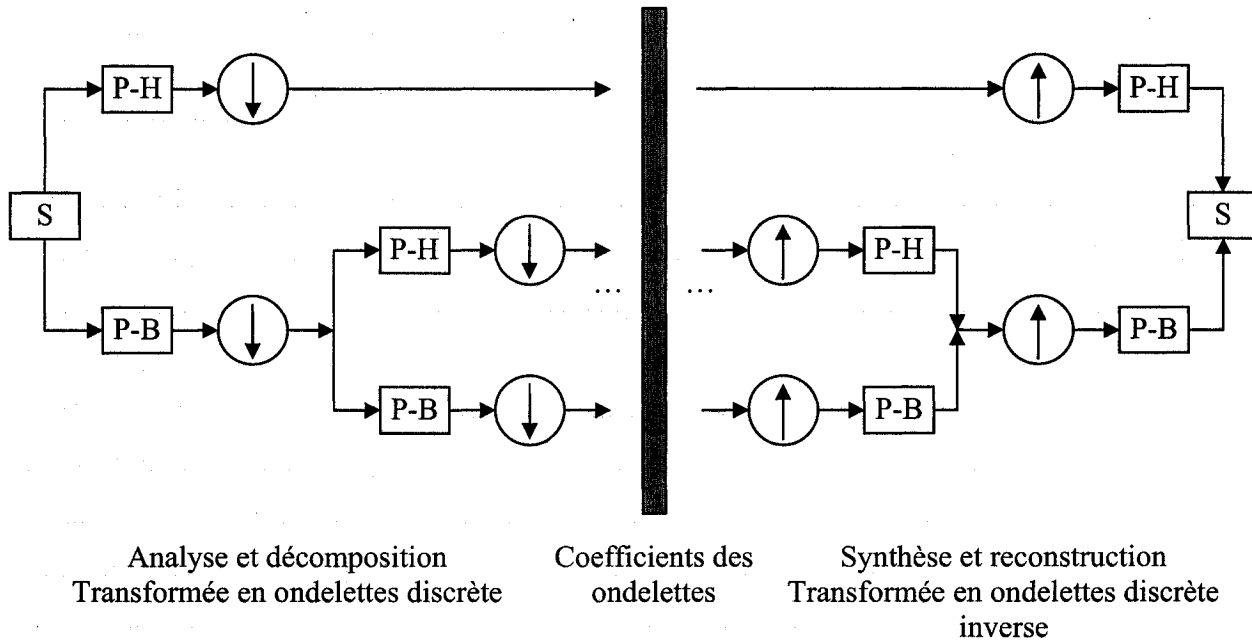
Dans le passé, les algorithmes de segmentation basés sur l'analyse de la texture opéraient à résolution unique du fait qu'ils étaient incapables de caractériser convenablement différents niveaux de texture. Vers le début des années quatre-vingt-dix, l'avènement de méthodes d'analyse multi-résolution plus sophistiquées, telles que celles basées sur les transformées en ondelettes [28][29][30], a permis une percée importante dans le domaine de l'analyse multi-résolution. Ces méthodes procèdent à une double analyse, globale et locale, de la texture présente dans une image. Elles se basent sur la capacité des ondelettes à organiser les données de texture en plusieurs ensembles d'informations caractéristiques, chacun défini à une résolution différente. Ces ensembles séparent les différentes structures physiques des textures et permettent théoriquement d'atteindre des résultats de segmentation précis. Ainsi, les résolutions grossières portent les informations relatives aux structures larges présentes dans une image, tandis que les résolutions fines contiennent les détails nécessaires pour raffiner la segmentation.

Cette partie a pour but de présenter les méthodes de segmentation basées sur la transformée en ondelettes. Pour cette fin, nous avons décidé de la diviser en deux sections. La première résume le concept des ondelettes et expose les algorithmes de segmentation et de classification des textures développés dans [26], [27] et [28]. La dernière section analyse brièvement l'applicabilité de ces algorithmes sur le cas des images probabilistes.

2.4.6.1 Les méthodes de segmentation

La Figure 2-28 illustre le fonctionnement des ondelettes. Elle met l'accent sur le processus en deux étapes de décomposition et de reconstruction. En effet, la première phase qui décompose itérativement le signal en plusieurs composantes de plus faible résolution à l'aide d'un processus de

filtrage et de sous-échantillonnage¹⁰, permet d'obtenir les coefficients des ondelettes. La seconde phase a pour but de manipuler ces coefficients par un processus inverse de sur-échantillonnage¹¹ et de filtrage afin de reconstruire le signal original. Il est important de noter que la synthèse n'est pas appliquée directement après la phase de décomposition [33]. Une fois que les coefficients des ondelettes sont obtenus, ils sont analysés et manipulés suivant les critères de l'application.



Légende

- | | | | | | |
|------------|-------------------|--|----------------------|----------|--------|
| P-H | Filtre passe-haut | | Sous-échantillonnage | S | Signal |
| P-B | Filtre passe-bas | | Sur-échantillonnage | | |

Figure 2-28 : processus multi-étapes analyse-synthèse ou décomposition-reconstruction (reproduite de [33])

Les algorithmes de classification et de segmentation basés sur l'analyse de la texture à l'aide de la transformée en ondelettes sont dans leur majorité supervisés. Dans cette section, nous exposons trois méthodologies, soient celle proposée par Chang et Kuo [26], l'algorithme de Salari et Ling [27] ainsi que le travail de Wang *et al.* [28].

¹⁰ « Downsampling ».

¹¹ « Upsampling ».

Le travail de Chang et Kuo [26] se base sur l'observation selon laquelle les sous-signaux qui sont décomposés récursivement dans les canaux des basses fréquences, ne contiennent pas nécessairement les informations de texture les plus significatives. Conséquemment les algorithmes basés sur la transformée en ondelettes traditionnelle de type pyramidal sont incapables de résoudre à coup sûr le problème de la segmentation et de la classification des textures. Afin de résoudre cette limitation, Chang et Kuo [26] proposent un algorithme itératif qui consiste à détecter et à décomposer les canaux de fréquences qui contiennent les informations les plus significatives.

Salari et Ling [27] se sont largement inspirés du travail de Laws [34] dans la conception de leur algorithme de segmentation multi-canal et multi-résolution basé sur l'analyse de la texture. Cet algorithme se divise en deux étapes, soit la décomposition en ondelettes hiérarchique et la segmentation basée sur l'analyse de la texture. La première phase consiste à appliquer les filtres en ondelettes sur l'image originale. Étant donné que ces filtres se caractérisent par une orientation différente, le résultat de la décomposition en ondelettes serait un ensemble de sous-images identifiées par les composantes de l'image originale conformes aux spécifications du filtre appliqué. La seconde phase consiste théoriquement à intégrer les caractéristiques issues de différents filtres dans le but de produire une segmentation. Afin d'atteindre leur but, Salari et Ling [27] font l'hypothèse que le nombre de segments dans l'image originale est connu d'avance.

Malheureusement, la publication de Salari et Ling [27] fournit un nombre limité de détails et d'informations relatives au processus de segmentation. En effet, les filtres de décomposition en ondelettes ne sont pas fournis et la création des images caractéristiques reste très floue, surtout en ce qui concerne les fenêtres d'analyse. L'algorithme *K-means clustering* utilisé dans la seconde phase n'est pas développé et les détails concernant la propagation de l'image d'étiquettes manquent.

Wang *et al.* [28] utilisent une approche relativement similaire à celles présentées par Chang et Kuo [26] et par Salari et Ling [27], avec quelques différences. Premièrement, la transformée en ondelettes traditionnelle est remplacée par les DWF¹² [30]. En second lieu, l'image caractéristique est filtrée afin de mieux prendre en considération les bordures des différentes régions. Finalement, la méthode de classification de Bayes est utilisée afin d'étiqueter les pixels. Outre la similarité entre les valeurs des pixels dans l'image caractéristique, leur proximité dans l'image originale est aussi prise en compte.

¹² « Discrete Wavelet Frames ».

2.4.6.2 Applicabilité sur la segmentation d'images probabilistes

Ces trois méthodes de classification des textures et de segmentation utilisant la transformée en ondelettes présentent des contraintes qui les rendent inapplicables sur les images probabilistes. En premier lieu, les trois algorithmes proposent des méthodes de segmentation supervisées qui présupposent l'existence d'une base de données incorporant toutes les textures qui peuvent être rencontrées dans les images. Cette hypothèse est particulièrement inappropriée pour les images probabilistes où une infinité de textures issues des différents schémas d'intersection des faisceaux de lecture peuvent être rencontrées (section 2.3).

En second lieu, la méthode proposée par Chang et Kuo [26] se limite à la classification des textures. Ainsi elle ne résout pas le problème des images probabilistes du fait qu'elle n'est pas capable d'extraire les régions dotées de textures uniformes.

Finalement, la publication de Salari et Ling [27] est peu détaillée, ce qui rend son analyse presque impossible. En effet, les détails qui sont exposés sont insuffisants pour juger de la vraie performance du schéma de segmentation. Mais le fait que Salari et Ling [27] proposent un algorithme supervisé limite l'applicabilité de leur méthode sur les images probabilistes.

2.4.7 L'analyse de la texture locale

Les problèmes que nous avons décelés avec la segmentation à l'aide des filtres de Gabor multicanaux et des ondelettes ont motivé une investigation plus poussée de la segmentation par analyse de la texture locale. Dans les sections précédentes, nous avons pu observer que la segmentation basée sur l'analyse de la texture est une des méthodes de choix en ce qui concerne les images probabilistes. Ceci est justifié par le fait que les régions d'occupation dans les images probabilistes sont caractérisées par des textures différentes. D'une part, les régions dont l'état d'occupation est connu sont représentées par une alternance de faisceaux de lecture issus des capteurs et d'espaces inconnus. D'autre part, les régions inconnues sont caractérisées par une distribution d'intensité homogène. Cette section explore plus particulièrement la segmentation basée sur l'analyse de la double distribution du *Local Binary Pattern* et du *Contrast* (LBP/C) qui caractérise la texture présente dans une région donnée de l'image.

La première partie développe les concepts relatifs à la représentation des textures [38][39] ainsi que les variations qui ont été proposées par Mäenpää *et al.* [40] et par Pietikäinen *et al.* [41]. La seconde partie développe le ratio logarithmique de probabilité utilisé pour comparer deux distributions des caractéristiques de la texture, chacune relative à une région différente. La troisième partie décrit le schéma de segmentation proposé par Ojala et Pietikäinen [39]. L'applicabilité de cette méthode sur le cas des images probabilistes est analysée dans la dernière partie.

2.4.7.1 Caractérisation de la texture

L'analyse de la texture présente dans une image à l'aide de la distribution de l'opérateur *Local Binary Pattern (LBP)* a été introduite par Ojala *et al.* [38]. Cet opérateur décrit la texture locale qui caractérise l'entourage de chaque pixel. Bien qu'il constitue une importante source d'informations quand aux caractéristiques des textures locales, le *LBP* ne permet pas de représenter le contraste entre les valeurs des pixels. Par la suite, Ojala *et al.* [39] ont étendu leur schéma en ajoutant un opérateur *C* de contraste pour chaque unité de texture locale. Ceci dit que la texture dans une région donnée de l'image sera caractérisée par une distribution à deux variables, soit le *LBP* et le *C*. L'analyse de cette distribution bidimensionnelle est à la base de la méthode de segmentation proposée par Ojala *et al.* [39].

Pour une région donnée, le *LBP* et le *C* sont calculés pour tous les pixels, à l'exception de ceux situés sur les bordures. Ainsi, pour chacun des pixels traités, nous considérons l'ensemble de taille [3 x 3] qui l'entoure (Figure 2-29-a). Un processus de discrétisation est appliqué sur cet ensemble, et conduit à une représentation binaire. Dans ce processus, le pixel central est utilisé comme seuil, et tout pixel avoisinant ayant une valeur supérieure ou égale à celui-ci est mis à un, sinon il reçoit une valeur nulle (Figure 2-29-b).

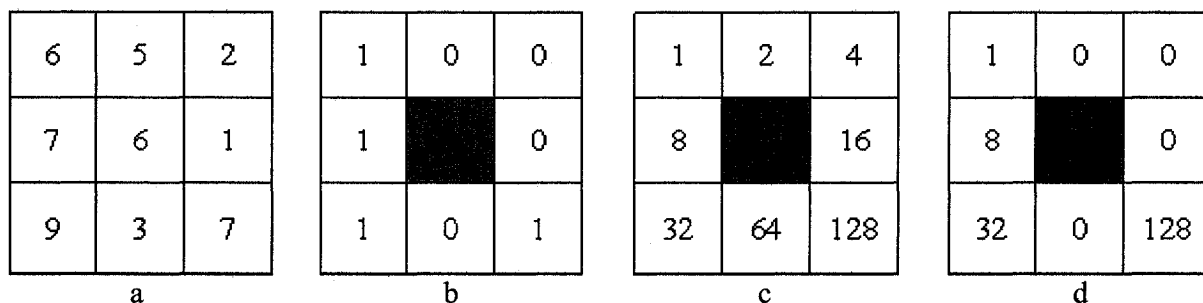


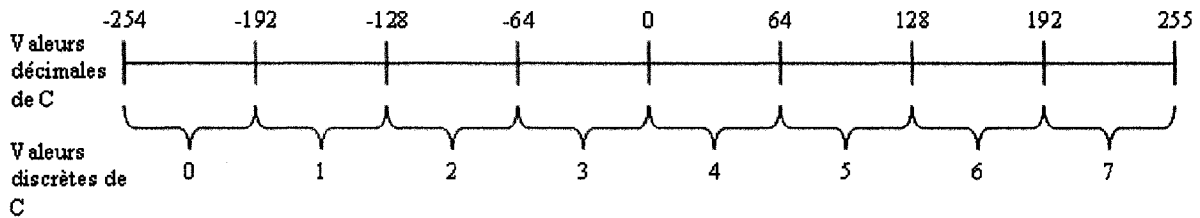
Figure 2-29 : exemple de calcul des caractéristiques de la texture, le *LBP* et le *C*. (a) région de taille [3 x 3] de l'image, (b) seuillage, (c) poids binomiaux, (d) résultat du processus de discrétisation

Les valeurs binaires obtenues sont multipliées par des poids binomiaux (Figure 2-29-c) et les résultats (Figure 2-29-d) sont additionnées afin d'obtenir le *LBP* de l'unité de texture sans considérer la valeur du pixel central. Par exemple, ici $LBP = 1 + 8 + 32 + 128 = 169$.

Ainsi, le *LBP* décrit la structure spatiale de la texture locale, mais ne fournit aucune information qui caractérise le contraste observé dans l'unité locale. Pour cette fin, le *LBP* est combiné à une simple mesure de contraste dénotée par *C*. Cette dernière n'est autre que la différence entre la moyenne de la valeur originale des pixels dont la représentation binaire (après application du seuil) est de 1 et la moyenne de la valeur originale des pixels dont la représentation binaire (après application du seuil) est de 0. La valeur du contraste relative à l'unité de texture représentée dans la Figure 2-29 est calculée comme suit : $C = (6 + 7 + 9 + 7) / 4 - (5 + 2 + 1 + 3) / 4 = 4.5$.

La distribution *LBP/C* résultante est représentée par un histogramme bidimensionnel de taille $256 * b$. Le nombre 256 provient du fait qu'un pixel central a 8 positions avoisinantes, chacune d'entre elles est représentée par une variable binaire, ce qui fait un total de 2^8 combinaisons *LBP* possibles pour une unité de texture de taille $[3 \times 3]$. D'autre part, le *b* correspond au nombre de niveaux de discrétisation du contraste *C*. Il est évident que la valeur du contraste est décimale et appartient à l'intervalle fermé $[-254; 255]$ pour une image encodée sur une échelle d'intensités de 0 à 255. La borne inférieure du contraste *C* correspond à la situation où tous les pixels avoisinants ont une valeur inférieure à celle du pixel central (normalement 255) et sont tous égaux à 254. La borne supérieure du contraste correspond à la situation où tous les pixels avoisinants au pixel central ont une valeur d'intensité égale à 255. Étant donné que la valeur de *C* est décimale, un processus de discrétisation est nécessaire afin de déterminer la correspondance entre une valeur donnée et son emplacement dans l'histogramme de distribution *LBP/C*. En se basant sur des résultats de classification expérimentaux, Ojala *et al.* [38] estiment qu'un nombre de niveaux de discrétisation *b* compris entre 4 et 16 est suffisant. La Figure 2-30 illustre pour deux valeurs de *b*, le processus de discrétisation qui assigne une valeur entière comprise entre 0 et *b*-1 pour chaque intervalle de la valeur décimale de *C*.

Exemple pour $b = 8$



Exemple pour $b = 16$

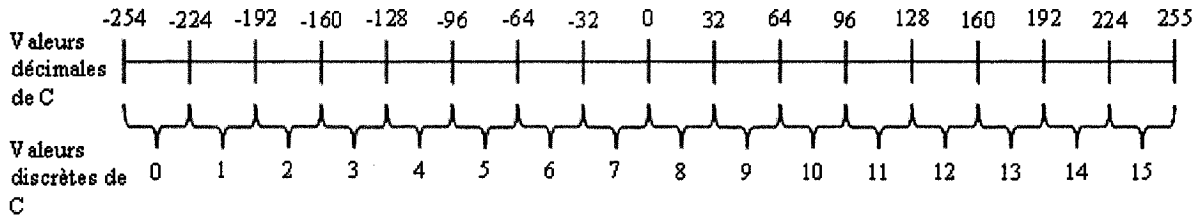


Figure 2-30 : exemple de l'application du processus de discrétisation sur la valeur du contraste

Le processus de discrétisation qu'Ojala et Pietikäinen [39] appliquent sur les valeurs du contraste réduit considérablement la taille de l'histogramme LBP/C qui caractérise la texture d'une région. Cette réduction a des implications directes sur le temps de traitement que nécessitent les étapes suivantes du processus de segmentation, étant donné que les histogrammes issus de différents sous-blocs doivent être comparés pour juger de l'uniformité de la texture dans une région parent donnée. Ojala et Pietikäinen [39] ne notent aucune variation dans les résultats de la segmentation pour des valeurs de b égales à 8 et à 16, mais d'après eux, une valeur inférieure à 8 est indésirable car elle affecte la sensibilité du schéma aux variations de contraste.

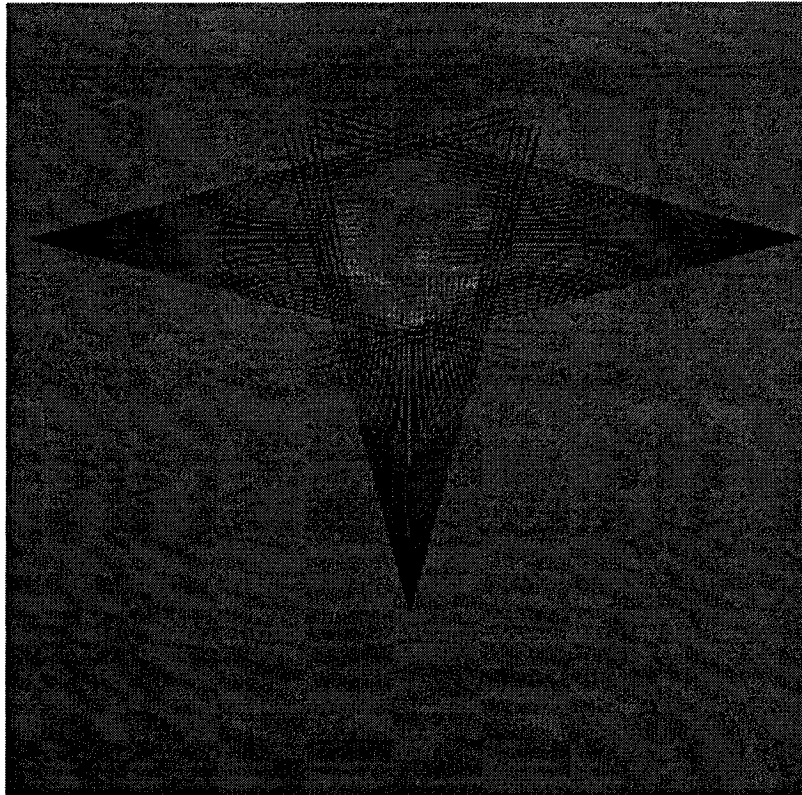


Figure 2-31 : image probabiliste générée par combinaison de données issues de trois points de vue

La Figure 2-31 correspond à une image probabiliste de taille [320 x 320], générée par la combinaison de données d'occupation issues de trois capteurs. Les cellules grises, de valeur égale à 128, représentent l'espace inconnu. Les cellules noires de valeur strictement inférieure à 128 représentent l'espace vide. Et les cellules blanches, de valeur strictement supérieure à 128 représentent l'espace occupé. La Figure 2-32 représente quand à elle, l'histogramme LBP/C de taille [256 x 8] correspondant à la Figure 2-31. Étant donné que l'espace *inconnu* homogène est prépondérant dans l'image probabiliste, la grande majorité des unités de texture soit 82 222 (Figure 2-32) sur 101 124, sont caractérisées par une valeur de contraste égale à 6 (valeur décimale du contraste égale à 128) et un *LBP* de 255. Le sommet d'amplitude 7 937 (Figure 2-32) caractérise quant à lui les cellules *libres* dont l'intensité des pixels avoisinants (*inconnus* et *libres*) est supérieure ou égale à la leur (*LBP* = 255). Le fait que les cellules *libres* ont une intensité inférieure à celles qui sont *inconnues* et qui leur sont adjacentes, réduit les valeurs décimales et discrètes ($C = 5$) du contraste par rapport aux régions *inconnues* homogènes ($C = 6$). Finalement, l'entourage de la majorité des cellules *occupées* est formé de pixels *occupés* et *inconnus* dont la différence entre les moyennes conduit à une valeur décimale du contraste positive et faible (bornée dans l'intervalle

[0;63]), ceci explique les faibles distributions des unités de texture correspondantes aux cellules occupées le long de $C = 4$ dans la Figure 2-32.

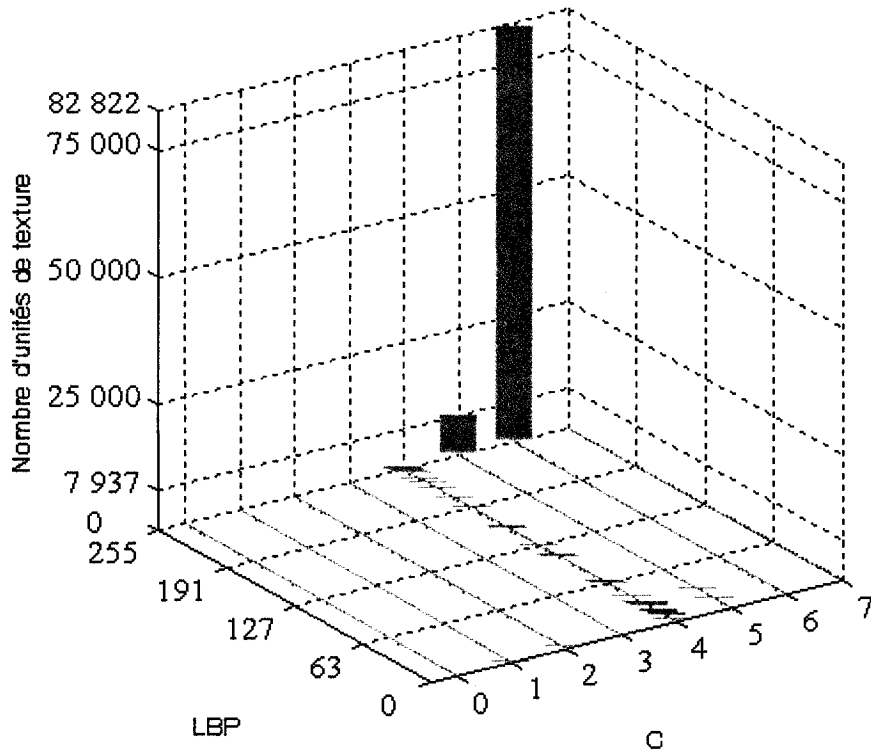


Figure 2-32 : histogramme LBP/C correspondant à l'image probabiliste de la Figure 2-31

Les histogrammes LBP/C relatifs à deux ou plusieurs régions d'une image sont comparés à l'aide d'outils statistiques afin de juger la similitude texturale des régions. Par conséquent, la taille des histogrammes qui dépend des plages de valeurs du LBP et du C , influence directement le temps d'exécution requis pour la segmentation. Comme signalé précédemment, Ojala et Pietikäinen [39] appliquent un processus de discrétisation sur la mesure décimale du contraste afin de réduire sa plage de valeur possibles à b unités. La limite minimale de b est fixée expérimentalement à 8. Au-delà de cette valeur la mesure du contraste perd son utilité [39]. Ainsi, une réduction supplémentaire de la taille de l'histogramme LBP/C nécessite des traitements sur l'opérateur LBP dont les valeurs entières sont comprises dans l'intervalle [0;255]. Mäenpää *et al.* [40] proposent dans leur algorithme de classification des textures une méthode heuristique pour la réduction du nombre de valeurs possibles du LBP . Ce schéma repose sur le choix d'un sous-ensemble de 58 valeurs parmi les 256 possibles.

Toutes les unités de texture dont le LBP n'appartient pas au sous-ensemble prédéfini sont cumulées dans la 59^{ième} colonne de l'histogramme caractéristique dont la taille devient $[59 \times b]$ éléments. Le choix de ce sous-ensemble suit un processus bien déterminé. En premier lieu, Mäenpää *et al.* [40] représentent les valeurs du LBP par leur équivalent binaire sur 8 bits. Puis ils définissent une mesure de non uniformité dénotée par $U(LBP)$. Celle-ci correspond au nombre de transitions de 0/1 et 1/0 dans la représentation circulaire (binaire) de la valeur de l'opérateur LBP . À titre d'exemple, les valeurs 0 (représentation binaire 0000 0000) et 255 (représentation binaire 1111 1111) ont une valeur de U égale à 0, tandis que les valeurs 1 (00000001₂), 2 (00000010₂), 4 (00000100₂), 8 (00001000₂), 16 (00010000₂), 32 (00100000₂), 64 (01000000₂) et 128 (10000000₂) ont une valeur de U égale à 2 étant donné que deux transitions de 0/1 et 1/0 sont observées dans leur représentation circulaire. En second lieu, Mäenpää *et al.* [40] notent que plus la valeur de U (le nombre de transitions) est faible, plus l'unité de texture est stable dans le cas d'une rotation ou d'un changement de taille de l'image. En se basant sur cette observation, seules les 58 représentations binaires du LBP dotées d'un U inférieur ou égal à deux sont prises en compte. Les valeurs binaires correspondantes sont : 0000 0000, 0000 0001, 0000 0011, 0000 0111, 0000 1111, 0001 1111, 0011 1111, 0111 1111 et 1111 1111, ainsi que les valeurs issues de la rotation circulaire de ces dernières. Le choix de 58 valeurs sur les 256 possibles est supporté par le fait que ces dernières représentent la majorité des unités de texture présentes dans les images testées durant leur expérimentation. À titre d'exemple, ces valeurs de LBP représentent plus de 94% des unités de texture dans une sélection des images de Brodatz [42], et fournissent d'excellents résultats de classification même quand l'image subit une rotation [40]. Il faut souligner que le travail de Mäenpää *et al.* [40] a été testé sur des images très particulières (Figure 2-33) contenant une seule texture très homogène. L'objectif était de reconnaître les textures dans le but de les classifier. Par conséquent la validité du schéma de classification demeure questionnable dans le contexte de segmentation.

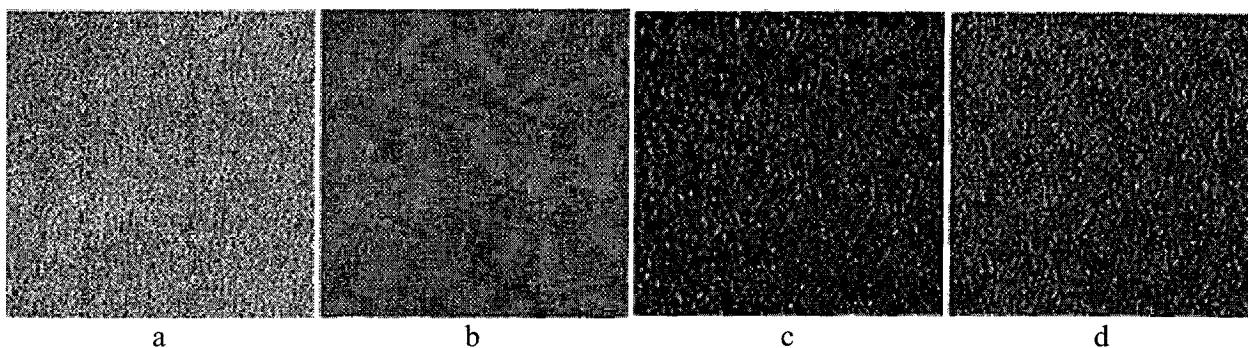


Figure 2-33 : exemple d'images sur lesquelles Mäenpää *et al.* [40] ont testé leur algorithme de classification des textures (reproduite de [42])

L'opérateur *LBP* introduit par Ojala et Pietikäinen [39] est variant aux rotations, ce qui est indésirable dans le cas de certaines applications. Pietikäinen *et al.* [41] ont introduit une version invariante aux rotations dénotée par *LBPROT*. Dans cette dernière, les valeurs binaires du voisinage obtenues après seuillage (Figure 2-34-a) sont converties en mots de huit bits (Figure 2-34-b). Par la suite, un nombre arbitraire de rotations sont opérées (Figure 2-34-c) jusqu'à ce que le mot obtenu corresponde à l'une des 36 possibilités (Figure 2-34-d) de 0 et de 1 qu'un mot de huit bits peut former sous rotation. L'indice de la possibilité correspondante détermine la valeur de l'opérateur *LBPROT* qui caractérise l'unité de texture (Figure 2-34-d).

La principale limitation du *LBPROT*, selon Pietikäinen *et al.* [41], provient de son incapacité à reconnaître les textures qui ont subi une rotation arbitraire, et ne fournit des résultats convenables que dans le cas où la rotation opérée est de $90 \times k$ degrés où $k \in \mathbb{N}$. De plus, Pietikäinen *et al.* ont aussi testé l'opérateur *LBPROT* sur des images particulières [42], similaires à celles représentées dans la Figure 2-33. À la différence des images probabilistes, celles utilisées dans [41] sont caractérisées par une texture unique et homogène. Dans le cas des images probabilistes, les faisceaux de lectures peuvent avoir n'importe quelle inclinaison, ainsi le *LBPROT* ne semble pas bien adapté à la représentation des textures qui les caractérisent. De plus, le schéma de Pietikäinen *et al.* [41] est utilisé dans la classification des textures à l'aide d'une base de données des motifs qui peuvent caractériser les régions de l'image. Par conséquent, la migration du *LBPROT* vers un schéma de segmentation implique que ce dernier soit supervisé, ce qui est indésirable dans le cas des grilles probabilistes.

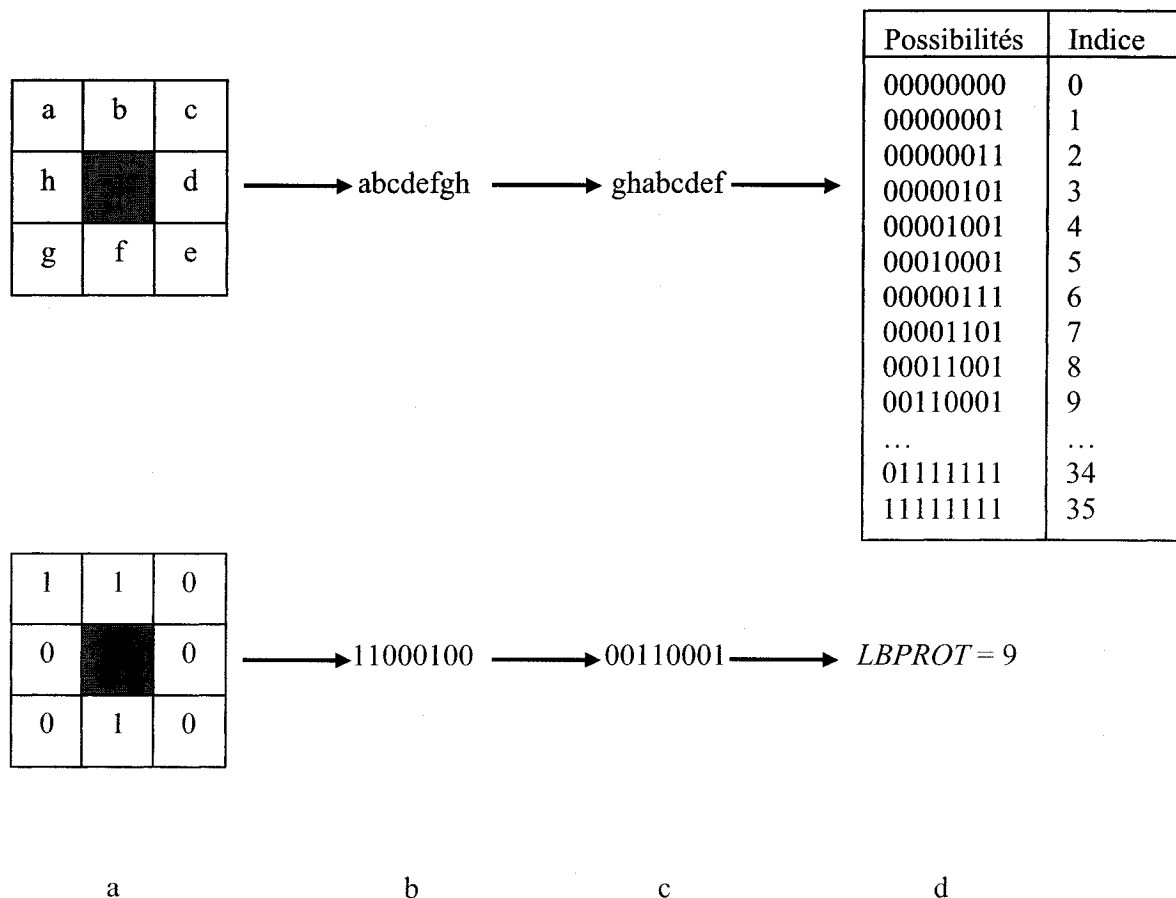


Figure 2-34 : calcul du *LBPROT* pour une unité de texture donnée. (a) valeurs binaires obtenues après seuillage, (b) mots de huit bits correspondants, (c) rotation du mot jusqu'à correspondance dans la table de possibilités, (d) l'indice du mot correspond au *LBPROT*

2.4.7.2 Comparaisons des distributions LBP/C – le G-statistique

Comme signalé dans la section précédente, la texture qui caractérise une région donnée est représentée à l'aide de la double distribution des opérateurs *LBP* et *C*. Afin de juger si deux régions sont uniformes en terme de texture, Ojala et Pietikäinen [39] proposent d'employer une mesure de similarité connue sous le nom de G-statistique [43][44]. Cette dernière est utilisée afin de comparer les histogrammes LBP/C relatifs à deux régions de l'image.

Le ratio logarithmique de probabilité G-statistique développé par Sokal et Rohlf [43][44] permet, tout comme les tests Chi carré et Kolmogorov-Smirnov [45], de vérifier la validité de la

correspondance entre deux distributions de fréquences¹³, l'une expérimentale et l'autre théorique représentant l'hypothèse testée. Mathématiquement, les distributions, qu'elles soient expérimentales ou théoriques, peuvent suivre une loi binomiale, de Poisson ou normale.

Le choix du G-statistique se base sur les avantages qualitatifs et quantitatifs qu'il offre par rapport aux autres tests [43][44] cités précédemment. La première propriété qui caractérise le ratio logarithmique de probabilité est son additivité. En effet, les résultats de plusieurs tests peuvent être additionnés pour produire un résultat pertinent sur le plan mathématique. De plus, l'ensemble des tests faits à l'aide du paramètre G-statistique peuvent être partitionnés en sous-groupes, chacun caractérisant un degré de liberté différent. Cette propriété est analogue aux tests statistiques *anova* [45][46] basés sur l'étude de la variance. Ces propriétés mathématiques dépassent le cadre de notre étude et par conséquent ne seront pas développées. Il faut néanmoins noter que, outre les avantages théoriques du paramètre G-statistique, ce dernier se caractérise par la rapidité de son calcul, que ce soit à l'aide d'outils numériques ou manuels [43][44]. Sokal et Rohlf [43][44] définissent le ratio logarithmique de probabilité *G* à l'aide de l'équation suivante :

$$\begin{aligned}
 G = & 2 \left[\sum_{s,m} \sum_{i=1}^n f_i \times \ln(f_i) \right] - 2 \left[\sum_{s,m} \left(\sum_{i=1}^n f_i \right) \times \ln \left(\sum_{i=1}^n f_i \right) \right] \\
 & - 2 \left[\sum_{i=1}^n \left(\sum_{s,m} f_i \right) \times \ln \left(\sum_{s,m} f_i \right) \right] \\
 & + 2 \left[\left(\sum_{s,m} \sum_{i=1}^n f_i \right) \times \ln \left(\sum_{s,m} \sum_{i=1}^n f_i \right) \right]
 \end{aligned} \tag{2.5}$$

Où : *s* et *m* : les deux histogrammes de distribution à comparer.

n : le nombre d'observations dans chacun des histogrammes, soit 256 x *b*.

f_i : fréquence de l'observation *i*.

Typiquement, plus les deux histogrammes *s* et *m* à comparer sont similaires, moindre sera la valeur de *G*.

¹³ Distribution de fréquences : « Graphique ou tableau montrant avec quelle fréquence chaque valeur ou chaque série de valeurs d'une variable apparaît dans un ensemble de données » [47].

Dans le cas de l'algorithme de segmentation proposé par Ojala et Pietikäinen [39], les distributions fréquentielles à comparer correspondent aux histogrammes de distribution LBP/C (section 2.4.7.1). Plus la valeur absolue du G est petite, plus grande sera la probabilité que les deux histogrammes LBP/C comparés soient issus de régions dotées d'une texture similaire. Ripley [48] ainsi que Getis et Franklin [49] fixent à 5% le seuil d'acceptation au-delà duquel les distributions de fréquences comparées à l'aide du G -statistique sont considérées comme issues de populations différentes (*i.e.* hétérogènes).

2.4.7.3 Algorithme de segmentation

Le processus de segmentation proposé par Ojala et Pietikäinen [39] consiste en trois phases : la division hiérarchique, la fusion agglomérée et la classification des pixels. La première divise hiérarchiquement l'image en régions dotées d'une texture relativement uniforme. Par la suite, la seconde phase fusionne les régions adjacentes dotées d'une texture similaire. À ce niveau, les segments obtenus correspondent à une approximation des régions présentes dans l'image. Afin d'obtenir une meilleure précision en terme de segmentation, la phase de classification des pixels est appliquée. Dans les sections suivantes, nous développons ces étapes telles que décrites dans [39].

2.4.7.3.1 Division hiérarchique

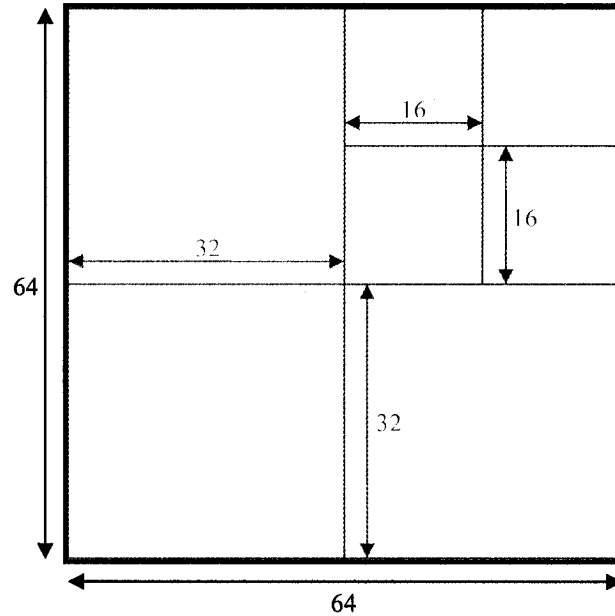
La première étape de l'algorithme d'Ojala et Pietikäinen [39] consiste à diviser l'image originale en régions de taille $[S_{max} \times S_{max}]$. Celles-ci sont ensuite divisées hiérarchiquement en blocs carrés de taille variable. La décision de subdiviser un bloc en quatre sous-blocs est basée sur le résultat d'un test d'uniformité. Pour un bloc donné, la première étape de ce test consiste à calculer les six ratios logarithmiques de probabilité entre les paires d'histogrammes LBP/C relatifs à ces sous-blocs. La plus grande valeur est notée G_{max} et la plus petite G_{min} . Le bloc parent est considéré non uniforme et subdivisé en quatre sous-blocs de taille égale si :

$$R = \frac{G_{max}}{G_{min}} > X \quad (2.6)$$

En ce qui concerne le choix de X , il vaut mieux considérer une valeur faible étant donné que si nous appliquons des divisions inutiles, la phase de la fusion agglomérée est capable de corriger l'erreur. Dans le cas contraire l'erreur est incorrigible. Ojala et Pietikäinen [39] fixent expérimentalement la valeur de X à 1.2 étant donné que 20% de différence relative entre G_{max} et G_{min}

correspond généralement à une déviation dans la texture locale. Comme signalé précédemment, Getis et Franklin [49] fixent le seuil d'acceptation pour une valeur unique de G à 5%, tandis que le seuil $X = 1.2$, introduit par Ojala et Pietikäinen [39], correspond à une valeur différente relative à un ratio entre deux valeurs du G-statistique soient le G_{max} et le G_{min} .

Les divisions itératives des blocs en sous-blocs continuent jusqu'à ce la taille du sous-bloc atteigne $[S_{min} \times S_{min}]$. Il est primordial de définir une borne inférieure S_{min} (en pratique cette valeur est soit 8 soit 16 pixels) au-delà de laquelle un bloc est indivisible. Cette limitation provient du fait qu'un bloc donné doit contenir un nombre de pixels suffisant afin que son histogramme LBP/C soit représentatif. La valeur de S_{max} choisie par Ojala et Pietikäinen [39] est de 64 pixels. La Figure 2-35 ci-dessous résume l'algorithme de la phase de division hiérarchique pour les valeurs choisies par Ojala et Pietikäinen [39] soient S_{max} égale à 64 et S_{min} égale à 16.



$$\text{Si } \left(R^{64} = \frac{G_{max}^{32}}{G_{min}^{32}} > 1.2 \right) \text{ alors divise en 4 sous-blocs de } [32 \times 32]$$

$$\Rightarrow \text{Si } \left(R_i^{32} = \frac{G_{max}^{16}}{G_{min}^{16}} > 1.2 \mid i \in [1;4] \right) \text{ alors divise en 4 sous-blocs de } [16 \times 16]$$

Figure 2-35 : modèle de subdivision dans la phase « division hiérarchique »

Si la phase de division hiérarchique est contournée en divisant directement l'image en blocs de taille $[S_{min} \times S_{min}]$, le nombre de fusions nécessaires pour l'obtention de régions homogènes

deviendra de plus en plus grand et aura un impact sur la charge de calcul, surtout dans le cas des environnements de grande dimension.

2.4.7.3.2 Fusion agglomérée

Cette phase consiste à fusionner les régions avoisinantes dotées d'une texture similaire jusqu'à ce qu'un critère de convergence soit atteint. Principalement, la fusion concerne les paires de segments adjacents dotés de la plus faible importance de fusion¹⁴ MI , celle-ci étant déterminée par l'équation suivante :

$$MI = p \times G \quad (2.7)$$

Où : p : nombre de pixels dans la plus petite des deux régions.

G : ratio logarithmique de probabilité entre deux régions, calculé dans l'équation (2.5).

C'est-à-dire que chaque fusion implique le plus petit changement possible dans l'image segmentée.

Une fois que deux régions sont fusionnées, Ojala et Pietikäinen [39] additionnent leurs histogrammes LBP/C respectifs afin de créer l'histogramme de la région globale. Le processus de fusion entre les régions et celles qui leurs sont adjacentes continue jusqu'à ce que le ratio de l'importance de fusion, MIR , atteigne un certain seuil Y défini par :

$$MIR = \frac{MI_{cur}}{MI_{max}} > Y \quad (2.8)$$

C'est-à-dire que la fusion s'arrête quand le rapport du MI actuel (MI_{cur}) sur le plus grand MI (MI_{max}) perçu jusqu'à date est supérieur à un certain seuil Y dont la valeur a été fixée expérimentalement par Ojala et Pietikäinen [39] à 2. Étant donné que la valeur du MI peut être négative, au début du processus de fusion, il est possible que le MI_{cur} et le MI_{max} soient tous les deux nuls, ce qui entraîne l'arrêt de l'opération. Par conséquent la règle d'arrêt représentée dans l'équation (2.8) n'est évaluée qu'après que 10% du nombre total de fusions ne soient réalisées.

2.4.7.3.3 Classification des pixels

La classification des pixels sur les bordures des régions permet d'améliorer le résultat de la segmentation en adaptant la forme des segments à celle des objets présents dans l'environnement.

¹⁴ « Merger Importance ».

Étant donné que cette phase n'agit que sur les bordures des segments obtenus jusqu'à date, son succès dépend de l'obtention d'un résultat cohérent à la fin de la phase de fusion agglomérée. Dans le cas positif, et afin d'améliorer la précision de la segmentation, cette phase considère les pixels qui sont sur la bordure d'au moins deux régions. Un disque de rayon r est centré sur le pixel en question (disque rouge dans la Figure 2-36) et les ratios logarithmiques de probabilité G entre la région définie par le disque et chacune des régions avoisinantes sont calculés. Le pixel est reclassé dans une nouvelle région nommée *autre*, si la valeur du G -statistique entre le disque et la région en question G_{autre} est inférieure à la valeur du G -statistique entre le disque et la région d'origine G_{origine} .

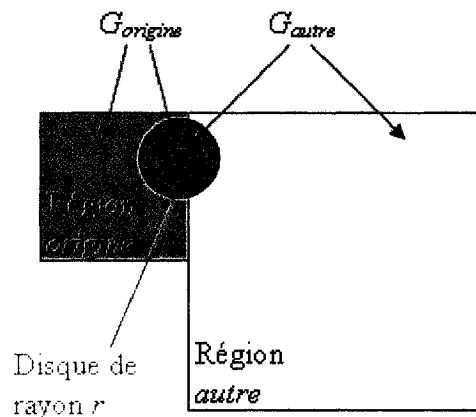


Figure 2-36 : illustration du processus de classification des pixels décrit dans [39]

A chaque itération, seuls les pixels qui ont été classés durant l'itération précédente sont traités. Ce processus continue jusqu'à ce qu'aucun pixel ne soit reclassé ou simplement quand un nombre limite d'itérations est atteint. Ce nombre limite est égal à deux fois S_{min} , étant donné que cette valeur correspond à la distance maximale d'imprécision durant la phase de fusion.

Il est important de noter la forte corrélation entre le rayon du disque choisit r et le résultat final de la segmentation. D'une part, une valeur de r très faible peut provoquer des discontinuités à l'intérieur d'une région. D'autre part, bien qu'une valeur élevée de r puisse ne pas localiser des bordures convenablement, elle est plus stable et fournit des bordures uniformes. Finalement, seuls les pixels dont le disque de rayon r est contenu à l'intérieur de l'image sont traités, ce qui implique qu'aucune adaptation de la forme des régions n'est réalisable sur les bordures de l'image segmentée.

2.4.7.4 Applicabilité sur la segmentation d'images probabilistes

Cette section fournit les caractéristiques techniques décrivant l'implantation de l'algorithme d'Ojala et Pietikäinen [39] que nous avons réalisée et les résultats ainsi que l'analyse de l'application de ce schéma sur les images probabilistes considérées dans la présente recherche.

Notre implantation de l'algorithme exposé dans [39] a été réalisée à l'aide du logiciel de traitement numérique Matlab [52] et de la librairie *Image Processing Toolbox* [53] qui lui est compatible. Bien que ce logiciel présente des limitations importantes concernant la gestion de la mémoire qui accroît considérablement le temps d'exécution par rapport aux langages de programmation de bas niveau, il fournit avec sa librairie de traitement d'images [53] une base stable et efficace pour le développement d'applications dans le domaine de l'imagerie numérique.

La première phase de l'algorithme de segmentation proposé par Ojala et Pietikäinen [39] est la division hiérarchique. Celle-ci commence par diviser l'image en régions de taille $[S_{max} \times S_{max}]$ soit $[64 \times 64]$ pixels, puis subdivise itérativement ces dernières dans le but d'obtenir des régions carrées dotées d'une texture relativement uniforme. Dans la publication [39], Ojala et Pietikäinen indiquent que des résultats cohérents sont obtenus après un maximum de deux niveaux de subdivision. Ainsi la taille minimale d'une subdivision $[S_{min} \times S_{min}]$ serait dans ce cas de $[16 \times 16]$ pixels.

La Figure 2-37-a ci-dessous correspond à l'image probabiliste complète de laquelle est issue la Figure 2-9-a. Elle incorpore deux lectures de capteur, chacune à un point de vue différent. Ces lectures sont caractérisées par un pas angulaire de 3 degrés et un écart-type de l'erreur du capteur sur la distance égal à 3 cm. Le résultat de l'application de la division hiérarchique sur l'image probabiliste ci-dessous est représenté dans la Figure 2-37-b. Dans cette image des bordures blanches séparent les différentes subdivisions, dont la taille minimale est de $[16 \times 16]$, afin de pouvoir les distinguer.

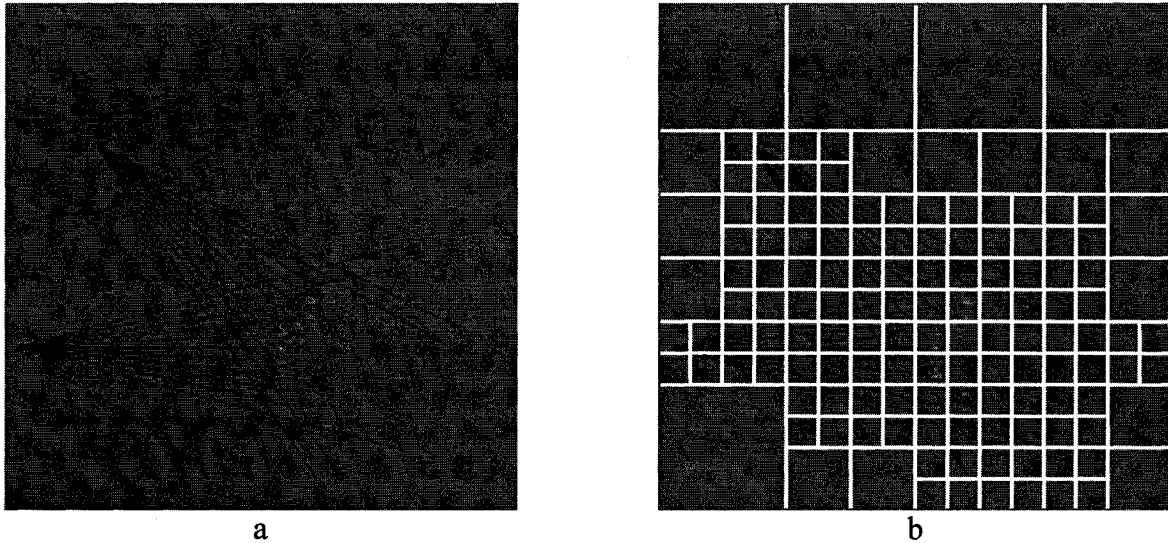


Figure 2-37 : (a) image probabiliste incorporant 2 points de vue, (b) résultat de la division hiérarchique montrant des subdivisions de taille minimale de [16 x 16] pixels

La fusion agglomérée, telle que nous l'avons décrite dans la section 2.4.7.3.2, consiste à fusionner les régions avoisinantes dotées d'une texture similaire jusqu'à ce qu'un critère de convergence soit atteint. Principalement, à chaque itération la fusion concerne la paire de régions adjacentes dotées de la plus faible importance de fusion (équation (2.7)). Ce processus continue jusqu'à ce que le rapport des *MI*s actuel et maximal soit supérieur à un seuil *Y* (équation (2.8)), dont la valeur a été fixée empiriquement par Ojala et Pietikäinen [39] à 2.0. Comme signalé dans la publication originale, le résultat de cette étape est décisif étant donné que la troisième phase de l'algorithme n'implique aucune fusion ni division, son rôle se limite à un simple raffinement des bordures des segments obtenus à la fin de l'étape actuelle.

La Figure 2-38 illustre le résultat de la seconde phase de l'algorithme de segmentation proposé par Ojala et Pietikäinen [39] sur l'image probabiliste de la Figure 2-37-a. Bien que ce résultat semble au premier abord très valable, il ne permet pas d'identifier les espaces explorés dotés d'une occupation homogène. En d'autres termes, le schéma de segmentation d'Ojala et Pietikäinen [39] est capable d'identifier les différences de texture au sein d'une région dotée de la même occupation, mais cette capacité l'empêche de fusionner les subdivisions dans les espaces explorés. Il en résulte un certain antagonisme étant donné que certaines régions adjacentes, similaires et dotées d'une occupation inconnue n'ont pas été fusionnées. Afin de bien valider ces observations, la Figure 2-39 présente les résultats de l'application des deux premières phases de l'algorithme de

segmentation proposé dans [39] sur deux images probabilistes supplémentaires (Figure 2-39-a et Figure 2-39-d).

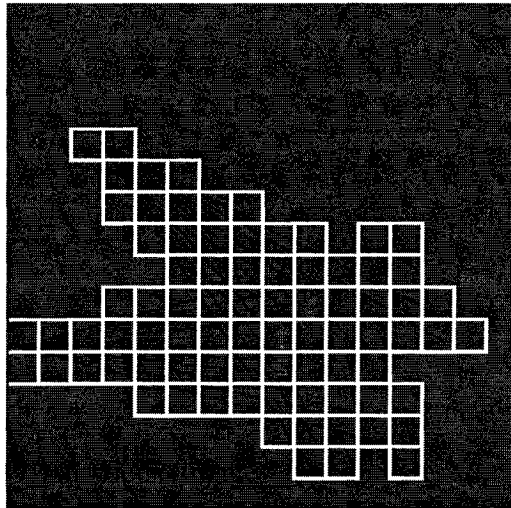


Figure 2-38 : résultat de la phase de fusion agglomérée appliquée sur l'image probabiliste originale représentée dans la Figure 2-37-a

La Figure 2-39-a incorpore un objet en forme de prise électrique mesuré par trois lectures de capteur, tandis que la Figure 2-39-d montre un objet rectangulaire mesuré à partir de quatre points de vue différents. Les résultats obtenus à la fin de la seconde phase, illustrés dans la Figure 2-39-c et la Figure 2-39-f, démontrent tout comme dans le cas de la Figure 2-38, l'incapacité du schéma d'Ojala et Pietikäinen [39] à isoler en segments distincts les régions des images probabilistes dotées d'une occupation uniforme dans les zones explorées.

Pour résumer, la première phase, soit la division hiérarchique, fournit des résultats plus ou moins optimaux étant donné qu'elle n'opère aucune subdivision dans la région dotée d'une occupation inconnue uniforme. Mais les autres régions subissent quand à elles un à deux niveaux de subdivision dus aux textures hétérogènes qui les caractérisent. La fusion agglomérée quand à elle souffre à deux niveaux :

- ❖ Premièrement, d'une hypersensibilité à la différence de textures entre les régions adjacentes dotées d'une occupation similaire;
- ❖ Deuxièmement, d'une certaine incohérence dans certains cas car elle est incapable de fusionner des régions adjacentes inconnues et totalement similaires du point de vue de l'arrangement des pixels.

Les problèmes que nous venons d'énumérer n'ont aucun lien direct avec la structure et les résultats de la première phase soit la division hiérarchique. L'obtention d'un résultat de segmentation consistant dépend donc essentiellement de la capacité de la seconde phase à fusionner les blocs dans les espaces explorés, soient les subdivisions adjacentes dotées d'une occupation similaire.

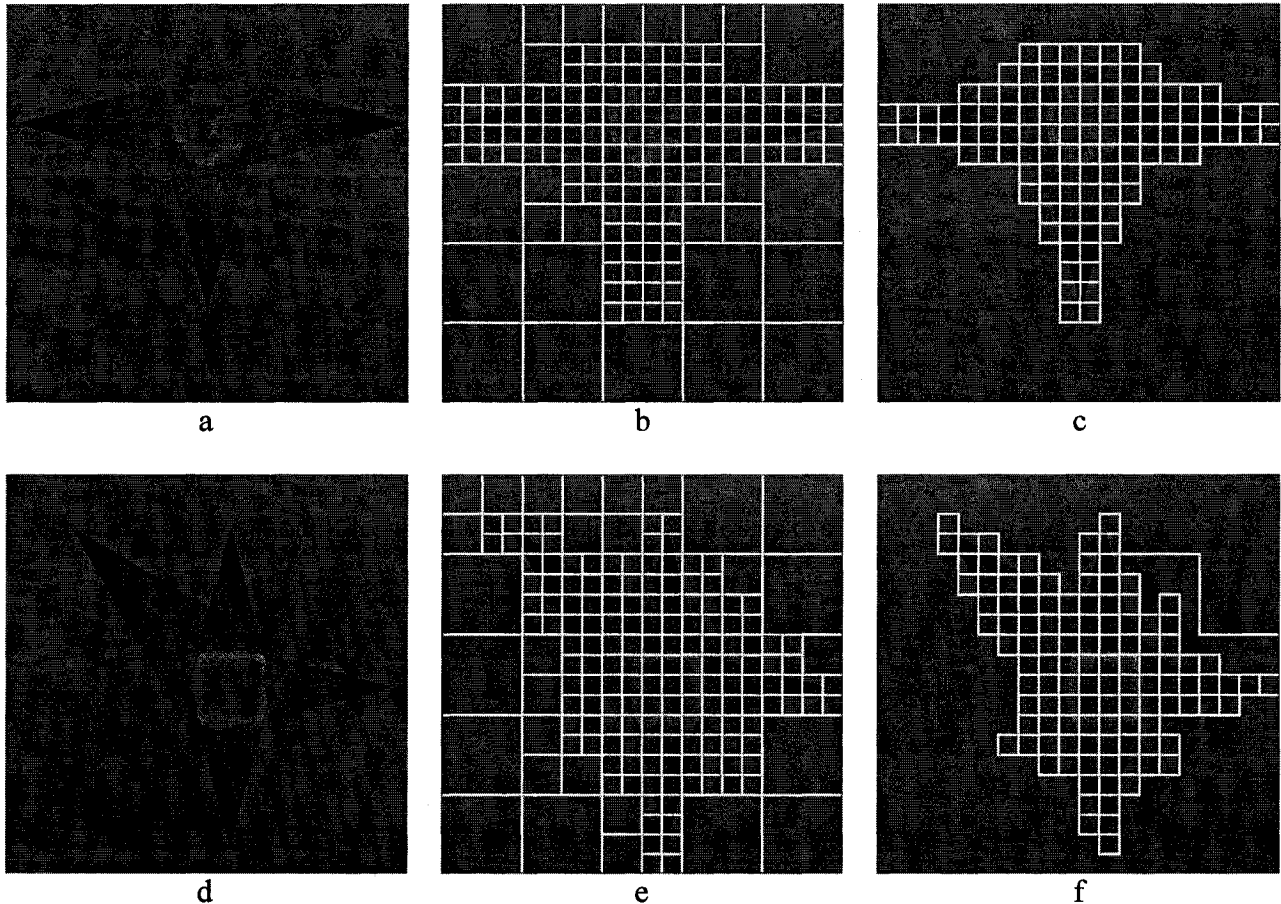


Figure 2-39 : (a) et (d) images probabilistes de taille initiale [320 x 320], (b) et (e) résultats de la division hiérarchique, (c) et (d) résultats de la fusion agglomérée

Les observations fournies dans les paragraphes précédents illustrent bien la difficulté de l'algorithme d'Ojala et Pietikäinen [39] à résoudre le problème des images probabilistes. En effet, l'application de la troisième phase, soit la classification des pixels, n'est pas considérée ici étant donné que cette dernière ne permet que de raffiner le résultat en adaptant la bordure des segments à celle des régions présentes dans l'image. Puisque les résultats que nous avons obtenus à la fin de la seconde phase ne sont pas satisfaisants, la troisième phase s'avère inutile.

Le Tableau 2-2 résume pour chacune des images probabilistes utilisées, sa taille, le nombre de subdivisions obtenues à la fin de la phase de divisions hiérarchiques, le nombre de fusions opérées

au cours de la seconde étape et la durée d'exécution des deux premières phases ainsi que la somme de ces deux valeurs. Ces résultats quantitatifs révèlent que l'algorithme de segmentation proposé par Ojala et Pietikäinen [39] est inadéquat pour des applications en temps réel.

Image probabiliste	Taille	Nombre de subdivisions initiales	Temps d'exécution de la phase 1	Nombre de fusions	Temps d'exécution de la phase 2	Temps d'exécution total
Figure 2-37-a	[256 x 256]	139	12.45 s	125	≈ 1877 s	≈ 1889.45 s
Figure 2-39-a	[320 x 320]	163	16.95 s	115	≈ 2082 s	≈ 2098.95 s
Figure 2-39-d	[320 x 320]	181	17.61 s	121	≈ 2933 s	≈ 2950.61 s

Tableau 2-2 : caractéristiques dimensionnelles et temporelles pour chacune des trois images probabilistes utilisées comme exemples dans cette partie suite à l'application des deux premières phases de l'algorithme proposé dans [39]

D'autre part, certaines faiblesses peuvent être observées au niveau de la conception du schéma de segmentation basé sur l'analyse de la texture à l'aide des histogrammes LBP/C. En effet, Ojala et Pietikäinen [39] considèrent dans la seconde phase de leur algorithme (section 2.4.7.3.2) que l'histogramme LBP/C qui caractérise une région W (LBP/C_W) issue de la fusion de deux sous-régions U et V est obtenu par la somme des deux distributions LBP/C relatives à U (LBP/C_U) et V (LBP/C_V). Cette hypothèse n'est pas exacte étant donné que la somme ($LBP/C_U + LBP/C_V$) obtenue néglige les unités de texture situées sur les bordures communes de U et de V . Le nombre d'unités de texture qui ne sont pas prises en compte augmente graduellement avec le nombre de fusions opérées étant donné qu'il est possible que la région W soit à son tour combinée avec d'autres blocs. Cet aspect mérite d'être corrigé afin que le schéma de segmentation soit plus précis.

Enfin, la méthode de segmentation LBP/C [39] repose sur le choix de deux valeurs expérimentales, soit les seuils X et Y dans les équations (2.6) et (2.8). Avec seulement deux seuils assez intuitifs, le schéma d'Ojala et Pietikäinen [39] fournit des résultats de segmentation cohérents sur certaines images extraites de l'album de Brodatz [42] et sur deux scènes naturelles. Il importe néanmoins d'observer que les régions dans les images testées ont des formes relativement simples, et la texture qui les caractérise est homogène.

Malgré les limitations mises en évidence par cette étude, le concept même d'une segmentation basée sur la mesure des caractéristiques de texture par les paramètres LBP et C , et les résultats présentés dans ces publications [38][39][40][41], ont suscité notre intérêt envers cette méthode. Cette dernière semble aux premiers abords mieux répondre aux particularités des images probabilistes que les techniques de segmentation classiques ou multicanaux examinées

préalablement. Pour ces raisons, une investigation plus profonde du travail d'Ojala et Pietikäinen [39] a été réalisée dans les étapes subséquentes de notre recherche.

2.5 Conclusion

La théorie présentée dans ce chapitre sert de base pour le développement d'une méthode de segmentation fiable pour les grilles d'occupation probabilistes bidimensionnelles et tridimensionnelles. D'une part, la formulation théorique des espaces probabilistes et de leurs caractéristiques texturales développée dans les deux premières parties nous a permis de maîtriser et de mieux comprendre le modèle de données mis en jeu. D'autre part, le développement dans la troisième partie de la théorie relative aux méthodes de segmentation développées jusqu'à date, nous donne un aperçu de l'applicabilité de chacun de ces schémas sur les images probabilistes considérées.

Chapitre 3 Segmentation probabiliste bidimensionnelle

3.1 Introduction

Ce chapitre constitue une extension naturelle du précédent du fait qu'il fournit les détails relatifs à la recherche que nous avons entreprise sur la segmentation des images probabilistes bidimensionnelles. Chacune des deux parties principales qui composent ce chapitre correspond à un stade dans l'évolution de notre travail. Dans un premier lieu, nous exposons la première étape d'adaptation de l'algorithme d'Ojala et Pietikäinen [39] ainsi que les résultats obtenus. La seconde section expose en profondeur l'algorithme de segmentation, révisé et raffiné, que nous proposons.

3.2 Tentative de raffinement de l'algorithme d'Ojala et Pietikäinen

Dans cette section, nous présentons la première étape de perfectionnement que nous avons opérée sur l'algorithme développé par Ojala et Pietikäinen [39]. Le but principal de cette adaptation est de résoudre d'un point de vue qualitatif les limitations de l'algorithme original afin de le rendre applicable sur les images probabilistes.

Lors de cette première tentative de raffinement, nous avons tenté de limiter le nombre de subdivisions que subissent les régions explorées par des lectures de capteurs. Le fait de limiter le nombre de subdivisions dans ces régions non seulement réduit la durée d'exécution des deux premières phases, mais permet aussi de mieux contrôler le fonctionnement de la fusion agglomérée car sa tâche sera plus spécialisée. Dans cette tentative, nous avons changé la valeur du seuil de subdivisions X (équation (2.6)) initialisé par défaut à 1.2 suivant les recommandations de [39]. Cette variable correspond à la valeur maximale de la différence relative tolérée, entre le G_{max} et le G_{min} au sein d'une même région, pour qu'elle ne soit pas subdivisée en quatre sous-blocs de taille similaire. Pour toutes les images probabilistes testées, jusqu'à une valeur de $X = 1.5$, les résultats de subdivisions sont demeurés inchangés comparativement à ceux obtenus en utilisant la valeur par défaut (Figure 2-39-b et Figure 2-39-e). Mais au-delà de cette borne, des résultats instables ont été observés. En effet, les images de la Figure 3-1 illustrent ce phénomène. Chacune de ces images est

issue de l'application de la division hiérarchique, avec un seuil de subdivision X égal à 2.1, sur une image probabiliste différente. Les faits suivants sont observés :

- ❖ Premièrement, à la différence de la Figure 3-1-b ainsi que de la Figure 3-1-c, et malgré l'existence d'une région inconnue au centre de la région occupée, la région où se situe l'objet dans la Figure 3-1-a est identifiée comme une région homogène et n'est pas subdivisée;
- ❖ Deuxièmement, dans la Figure 3-1-b et la Figure 3-1-c certaines régions qui chevauchent sur des espaces libres et des espaces occupés n'ont pas été subdivisées;
- ❖ Finalement, dans la Figure 3-1-a, la région de laquelle est originaire la lecture du capteur à droite, n'a pas subi de subdivision bien qu'elle se compose d'espaces dotés d'occupations entièrement différentes. Malgré cela, les autres régions de la Figure 3-1 d'où sont issues des lectures de capteurs ont toutes subi un niveau maximal de subdivisions.

Bien que ces trois points ne constituent pas une liste exhaustive des inconsistances au niveau des subdivisions perçues dans les images de la Figure 3-1, ils mettent l'accent sur le caractère fragile et intransigeant de la division hiérarchique. Par conséquent, pour des fins de stabilité nous avons choisi de conserver dans notre schéma de segmentation la valeur par défaut utilisée par Ojala et Pietikäinen [39] pour X soit 1.2.

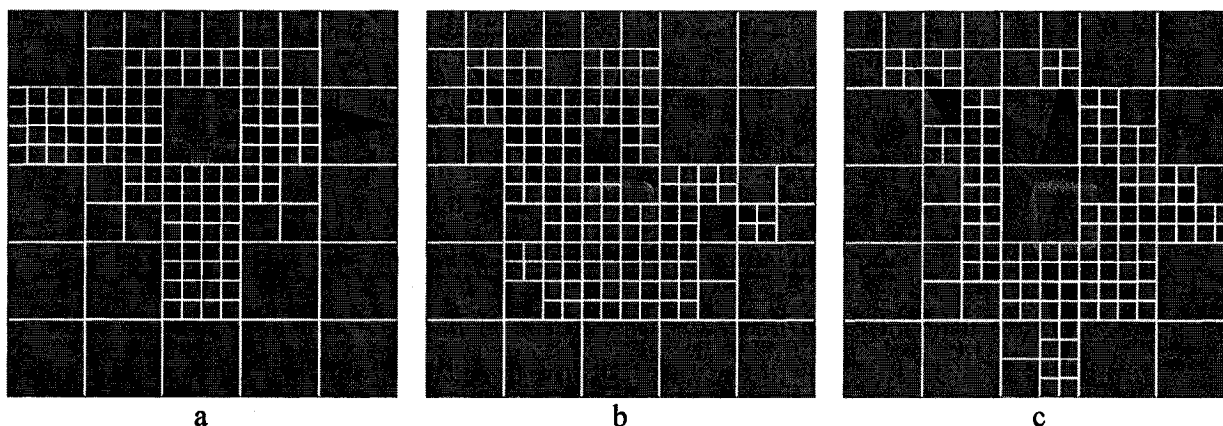


Figure 3-1 : résultat de la division hiérarchique sur trois images avec un seuil X de subdivision égal à 2.1

La seconde tentative de raffinement concerne la phase de fusions agglomérées (section 2.4.7.3.2). Notre objectif demeure similaire à celui d'Ojala et Pietikäinen [39], soit de combiner les

régions similaires adjacentes jusqu'à ce qu'un critère de convergence soit atteint. La version adaptée de la phase de fusion se divise en deux étapes :

- ❖ La première étape est similaire de part sa structure à celle développée dans l'algorithme original [39] du fait que la fusion concerne les paires de subdivisions adjacentes dotés de la plus faible importance de fusion ou MI (équation (2.7)). Mais le critère d'arrêt, défini à l'équation (2.8), a été remplacé par une valeur seuil fixée expérimentalement à 32 que le MI le plus faible ne peut pas dépasser. Ainsi l'avantage que le MI confère aux régions de taille réduite fait que les fusions entraînent le plus faible changement possible dans l'image probabiliste segmentée;
- ❖ La seconde étape se distingue de la première étape par le changement du critère de fusion. Ce dernier qui reposait sur les valeurs de MI , se base dans cette étape sur celles du ratio logarithmique de probabilité (équation (2.5)). Cette substitution fait perdre à la taille des régions son influence sur la décision de fusionner ou non. Le seuil maximal fixé expérimentalement à 1100 pour le G-statistique a été utilisé comme critère d'arrêt pour cette étape.

Il est important de noter que cette tentative de raffinement résout l'inconsistance de l'algorithme d'Ojala et Pietikäinen [39] (section 2.4.7.4) quand au calcul de l'histogramme LBP/C relatif à une région A issue de la fusion de deux subdivisions adjacentes B et C . En effet, à la différence du schéma proposé dans [39] qui considère que la distribution LBP/C de A est équivalente à la somme des distributions LBP/C respectives de B et C , notre algorithme calcule pour chacune des régions obtenues son propre histogramme de distribution.

La troisième tentative de raffinement est appliquée sur la troisième phase de l'algorithme d'Ojala et Pietikäinen [39]. Le processus de raffinement demeure similaire à celui exposé dans la section 2.4.7.3.3. Mais afin de fournir une meilleure précision, chaque itération opère le processus de classification illustré dans la Figure 2-36 sur l'ensemble des pixels situés sur la bordure de deux ou plusieurs segments adjacents, et non pas seulement sur ceux qui ont été classifiés durant l'itération précédente.

Nous illustrons dans la Figure 3-2, la Figure 3-3 et la Figure 3-4 les résultats de l'application du schéma de segmentation LBP/C, incluant les raffinements exposés dans les paragraphes précédents, sur trois images probabilistes. Les espaces occupés ainsi que le nombre de points de vue

différent d'une image à une autre. L'image probabiliste de la Figure 3-2-a implique un objet ayant la forme d'une prise électrique, tandis que les espaces occupés dans les images suivantes (Figure 3-3-a et Figure 3-4-a) correspondent à des objets rectangulaires. Les paramètres utilisés dans la segmentation sont les mêmes et correspondent à ceux décrits dans les parties précédentes. La Figure 3-2-b, la Figure 3-3-b et la Figure 3-4-b correspondent au résultat de la division hiérarchique dans laquelle les valeurs de S_{max} et de S_{min} sont respectivement de 64 et de 16 pixels. Les troisièmes (c) et quatrièmes (d) images des figures ci-dessous illustrent respectivement les résultats des deux étapes de la phase de fusion raffinée dans lesquelles les seuils d'arrêt pour le MI et le G sont respectivement de 32 et de 1100. Pour chacune des images probabilistes testées, le résultat de la phase de raffinement est représenté respectivement dans la Figure 3-2-e, la Figure 3-3-e et la Figure 3-4-e. Le nombre d'itérations choisi est de quatre, celui-ci s'avère relativement suffisant afin de bien percevoir la convergence entre la forme des segments obtenus et celle des différents espaces d'occupation dans les images probabilistes.

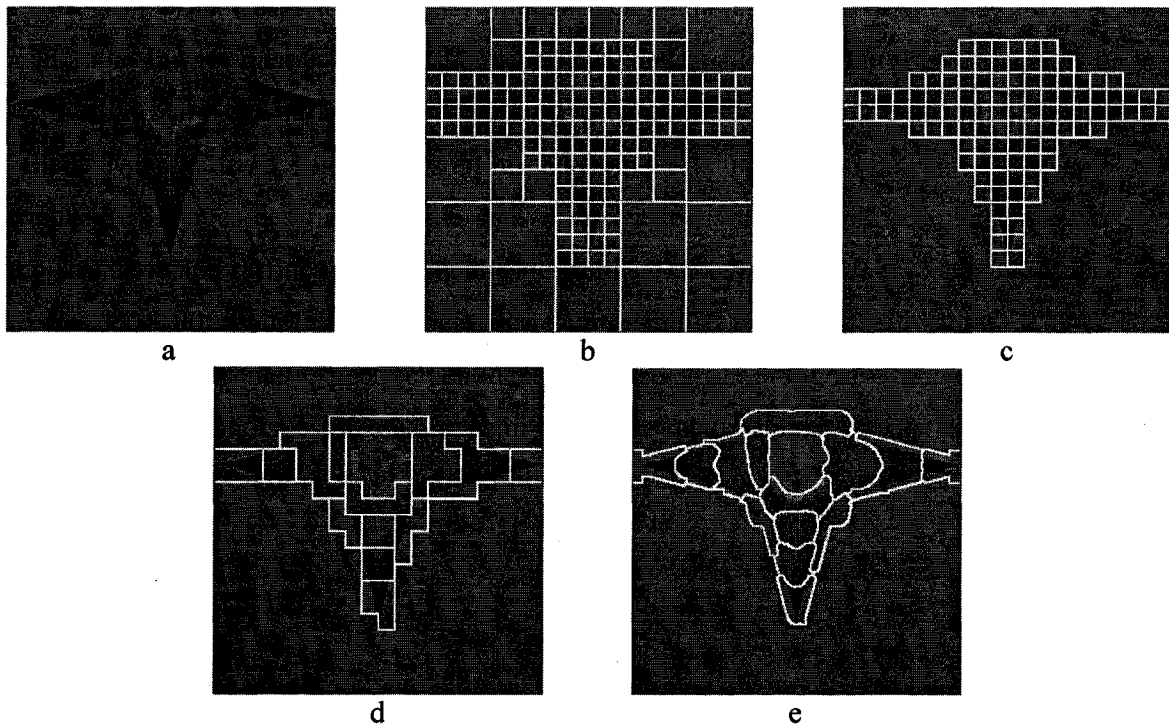


Figure 3-2 : (a) image probabiliste, (b) résultats de la division hiérarchique, (c) et (d) sorties de la première et de la seconde étape de la phase de fusion raffinée, (e) résultat de la classification des pixels raffinée

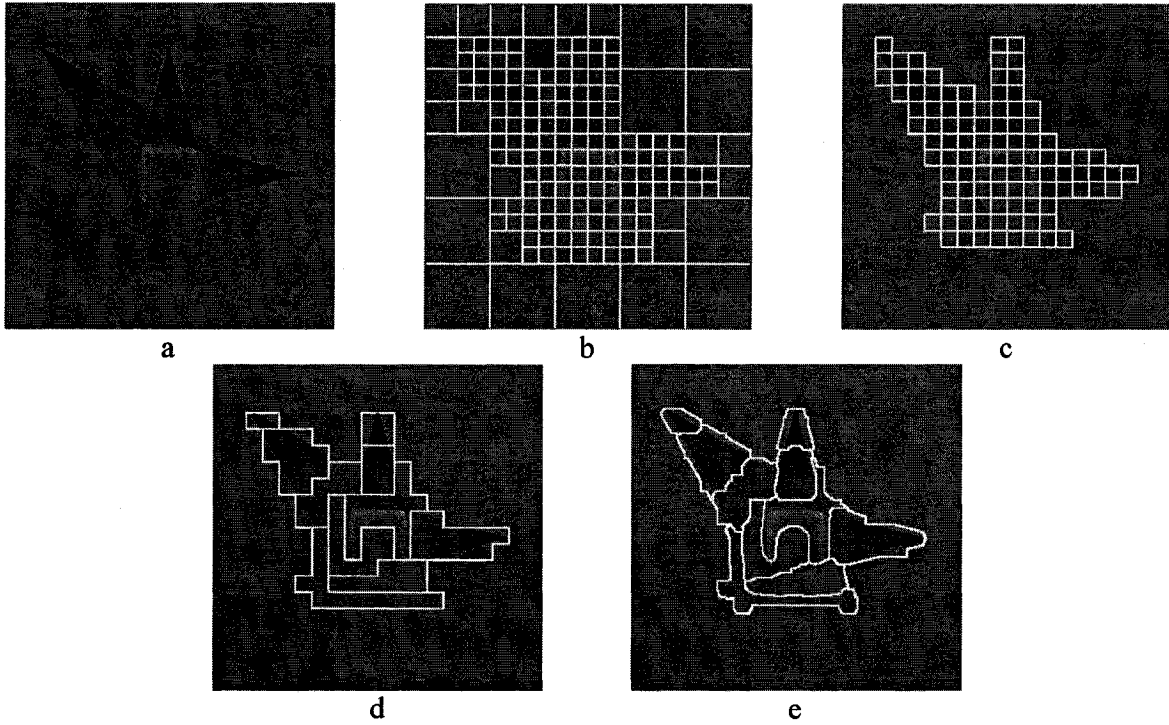


Figure 3-3 : (a) image probabiliste, (b) résultats de la division hiérarchique, (c) et (d) sorties de la première et de la seconde étape de la phase de fusion raffinée, (e) résultat de la classification des pixels raffinée

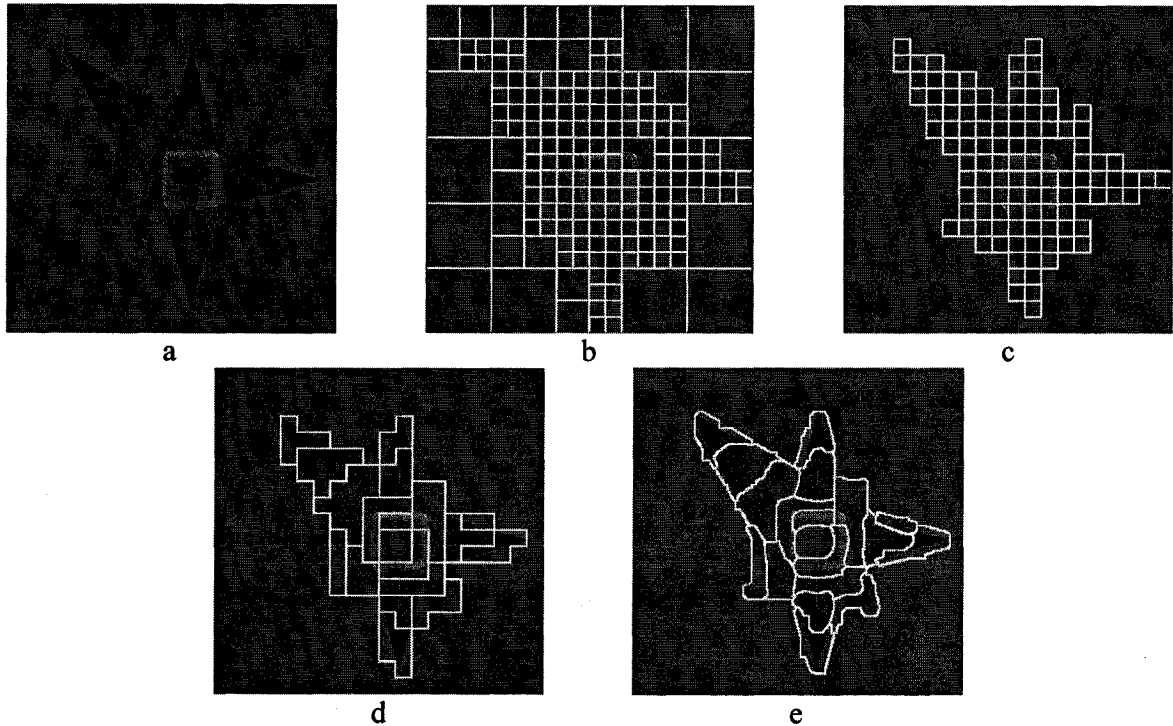


Figure 3-4 : (a) image probabiliste, (b) résultats de la division hiérarchique, (c) et (d) sorties de la première et de la seconde étape de la phase de fusion raffinée, (e) résultat de la classification des pixels raffinée

Comparés à ceux obtenus à l'aide de l'algorithme original [39], et présentés à la Figure 2-38 et à la Figure 2-39, les résultats exposés ci-dessus représentent un pas dans la bonne direction et nous permettent d'affirmer que le modèle LBP/C peut servir de base solide pour la conception d'un algorithme de segmentation spécialisé dans le traitement des images probabilistes.

Bien que la première adaptation que nous avons présentée ci-dessus n'arrive pas à produire une solution parfaite au problème de la segmentation, les conclusions que nous avons tirées de ces résultats ont servi de base à la conception de l'algorithme de segmentation que nous présentons dans la section suivante. Les trois points ci-dessous résument les principales limitations de cette implantation ainsi que les solutions que nous proposons pour les résoudre :

- ❖ Premièrement, bien que les segments prennent naissance durant la phase de fusion, aucune information quant à l'occupation de ces derniers n'est disponible. Ainsi, il serait intéressant d'étiqueter les régions suivant leur occupation. À titre d'exemple, un segment peut appartenir à l'un des trois types suivants : *occupé*, *libre* ou *inconnu*. La classification des segments suivant des états déterministes est particulièrement utile dans le cadre d'applications robotiques de planification de trajectoire et d'interaction avec l'environnement, dans lesquelles le robot a besoin d'identifier les objets afin de les contourner ou de les manipuler. La mise en place de cette modification est relativement simple. Il suffit d'assigner au cours de la division hiérarchique un état d'occupation déterministe à chacune des subdivisions créées. Par la suite, le nouveau schéma pourra combiner les segments adjacents dotés d'une occupation similaire et non pas d'une texture similaire;
- ❖ Deuxièmement, la phase de la classification des pixels fournit une convergence imparfaite, étant donné qu'elle ne peut pas traiter les bordures de l'image. Cette limitation provient du fait que pour qu'ils soient définis, les cercles entourant les pixels doivent être contenus à l'intérieur de l'image initiale;
- ❖ Le dernier point à mentionner concerne les limitations de cet algorithme du point de vue de la performance. Rappelons que l'implantation a été réalisée à l'aide du logiciel Matlab [52] sur un ordinateur doté d'un processeur Athlon 2600 XP [AMD Inc.] avec un giga-octet de mémoire vive. La durée d'exécution de la division hiérarchique est acceptable pour cette phase de développement. Elle se situe aux alentours d'une vingtaine de seconde pour les

images probabilistes testées. La deuxième étape prend en moyenne trente minutes, finalement la phase de classification requiert en moyenne six heures d'exécution pour réaliser quatre itérations. Ces données font que l'implantation actuelle de l'algorithme proposé est inadaptée aux systèmes en temps réel ou autonomes tel que les robots. Et des améliorations majeures sont requises : outre la migration du code et son optimisation en terme d'exécution, il serait cohérent de limiter les calculs opérés durant la seconde et la troisième phase. Un remaniement possible consiste à utiliser huit niveaux de discrétisation pour le contraste C au lieu de seize. Avec la réduction de la taille de l'histogramme des distributions LBP/ C de moitié, le nombre d'opérations nécessaires au calcul du ratio logarithmique de probabilité est divisé par un facteur de huit (car nous devons parcourir deux histogrammes, pour un total de quatre termes, équation (2.5)).

3.3 Un algorithme de segmentation LBP/ C révisé, amélioré et optimisé

En terme général, les méthodes de segmentation basées sur la représentation de la texture à l'aide de la double distribution LBP et C permettent jusqu'à un certain point de subdiviser l'espace suivant son degré d'occupation. Ces schémas de segmentation se sont avérés intéressants dans le cas des images probabilistes dont nous disposons, car la simulation des champs de vision des robots crée dans les espaces explorés des régions dotées de texture relativement uniforme. Dans cette partie, nous exposons l'algorithme final de segmentation appelé le LBP/ C révisé [50][51]. Ce dernier peut être considéré comme une extension et une adaptation de l'algorithme original. Bien que notre schéma se spécialise dans le traitement des images probabilistes, et tel que nous allons le voir dans le Chapitre 6, les concepts de base peuvent facilement être extrapolés pour couvrir diverses applications et types d'images.

Tout comme l'algorithme proposé par Ojala et Pietikäinen [39], le schéma de segmentation que nous avons développé peut se diviser en trois étapes, soit la division hiérarchique, la création des segments et la phase de raffinement. La première d'entre elles est relativement similaire à celle exposée dans [39]. L'innovation dans le schéma proposé provient principalement des changements majeurs que nous avons apportés à la seconde et à la troisième phase. Ces modifications non seulement adaptent l'algorithme original pour traiter les images probabilistes, mais réduisent considérablement la durée d'exécution.

La première phase divise l'image en régions caractérisées par des textures uniformes. Par la suite, l'étape de création des segments combine les régions adjacentes caractérisées par une occupation similaire. À ce niveau, les segments constituent une approximation des régions présentes dans l'image. Finalement, la phase de raffinement est appliquée dans le but d'accroître l'exactitude et la précision dans la localisation des contours.

3.3.1 Division hiérarchique

En résumé, la division hiérarchique subdivise l'image originale en blocs carrés de taille variable mais de texture relativement uniforme. Un test d'uniformité a été développé par Ojala et Pietikäinen [39] afin de déterminer si une région parent donnée contient des textures hétérogènes et doit être divisée en quatre sous-régions de taille égale. Cette procédure commence par l'identification des quatre sous-régions puis calcule le ratio logarithmique de probabilité (équation (2.5)) entre chacune des six paires possibles. La plus grande ainsi que la plus faible de ces valeurs sont dénotées respectivement par G_{max} et G_{min} . La région parent est considérée non-uniforme et par la suite subdivisée, si la différence relative entre G_{max} et G_{min} est supérieure à un certain seuil dénoté X (équation (2.6)).

Il a été démontré dans la section 3.2, ainsi que par Ojala et Pietikäinen [39], qu'une valeur réduite de X fournit des résultats plus stables. En effet, la seconde phase, soit la création des segments, est capable de corriger facilement les erreurs dues aux subdivisions supplémentaires qui touchent des régions sans différences majeures en terme de caractéristiques. Mais il serait impossible de réintroduire les segments que la première phase n'a pu identifier. Pour cette raison, une sursegmentation est toujours privilégiée dans la première phase. Dans le cas des images probabilistes bidimensionnelles et de celles de Brodatz [42], un choix de X égal à 1.2 fournit des résultats convenables.

La division hiérarchique commence par subdiviser l'image probabiliste en régions de taille $[S_{max} \times S_{max}]$ dont la valeur numérique est de $[64 \times 64]$ pixels. Si les dimensions de l'image originale ne sont pas des multiples de S_{max} , l'image originale est étendue en longueur et/ou en largeur avec des colonnes et lignes de pixels dont l'intensité est égale à 0.5. Cette valeur a été choisie de telle sorte que l'espace additionnel d'occupation *inconnue* n'interfère pas avec les régions qui ont été touchées par une lecture de capteur. De ce fait, la suppression de cet espace additionnel à la fin de la

segmentation n'affecte pas la validité des résultats. Les dimensions de l'image réajustée sont calculées à l'aide des équations suivantes :

$$M = \left\lceil \frac{M'}{64} \right\rceil \times 64 \quad (3.1)$$

$$N = \left\lceil \frac{N'}{64} \right\rceil \times 64 \quad (3.2)$$

Où : $(M; N)$: largeur et hauteur de l'image redimensionnée.

$(M'; N')$: largeur et hauteur de l'image probabiliste originale.

$\lceil a \rceil$: fonction plafond qui prend en entrée un nombre décimal a , et qui retourne le plus petit entier supérieur à a .

Pour chacun des blocs de taille $[S_{max} \times S_{max}]$, la décision d'opérer un premier niveau de subdivision dépend sur le résultat du test d'uniformité introduit à l'équation (2.6). Si le résultat du test atteint ou excède le ratio permis pour un bloc donné, ce dernier est subdivisé en quatre sous-blocs de taille $[32 \times 32]$ pixels. Dans ce cas, chacune des régions obtenues est à son tour soumise au même test d'uniformité qui décide si un second niveau de subdivision est nécessaire. Ce processus continue itérativement jusqu'à ce que la taille des subdivisions obtenues atteigne S_{min} . Ojala et Pietikäinen [39] indiquent que deux niveaux de subdivisions sont suffisants et fournissent un résultat de segmentation adéquat. Mais nos expérimentations ont démontré qu'un troisième niveau, menant à une valeur de S_{min} égale à huit, est nécessaire dans le cas des images probabilistes (section 4.6) du fait que cela fournit des résultats beaucoup plus stables et cohérents.

En terme d'implantation, une subdivision carrée donnée est représentée par une structure rectangulaire qui encapsule les coordonnées de son origine (pixel du haut à gauche), ainsi que ses dimensions (hauteur et largeur). La Figure 3-5 ci-dessous illustre les caractéristiques d'une région *subInitiale*, ainsi que ses quatre subdivisions.

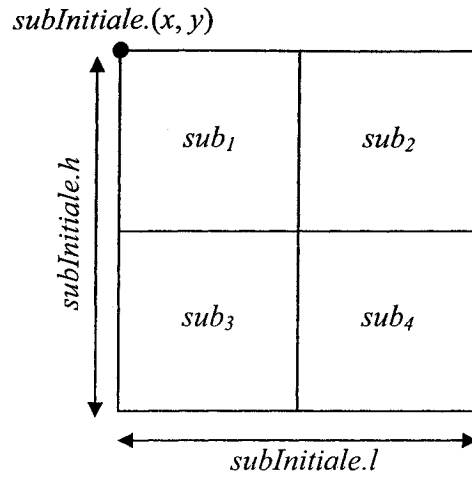


Figure 3-5 : position des quatre subdivisions à l'intérieur de la région initiale

Les dimensions ainsi que les coordonnées de l'origine qui caractérisent les subdivisions sont calculées à l'aide des équations suivantes :

$$sub_i.(l; h) = \left(\frac{subInitiale.l}{2}, \frac{subInitiale.h}{2} \right) \Big| i \in \{1; 2; 3; 4\} \quad (3.3)$$

$$sub_i.(x; y) = (subInitiale.x; subInitiale.y) \quad (3.4)$$

$$sub_2.(x; y) = (subInitiale.x + sub_1.l; subInitiale.y) \quad (3.5)$$

$$sub_3.(x; y) = (subInitiale.x; subInitiale.y + sub_1.h) \quad (3.6)$$

$$sub_4.(x; y) = (subInitiale.x + sub_1.l; subInitiale.y + sub_1.h) \quad (3.7)$$

Où : $subInitiale$: représente la région initiale.

$sub_1, sub_2, sub_3, sub_4$: les quatre subdivisions.

$s.(x; y)$: coordonnées de l'origine de la région s .

$s.(l; h)$: respectivement la largeur et la hauteur de la région s .

3.3.2 Création des segments

Le but de cette phase qui remplace la fusion agglomérée est de fusionner les régions similaires et adjacentes jusqu'à ce qu'un critère de convergence soit atteint. Principalement, la fusion implique les blocs avoisinants dont la probabilité moyenne d'occupation, dénotée par PMO ,

appartient au même intervalle. Ce paramètre qui est défini comme la moyenne de l'intensité des pixels dans une région donnée, est déterminé par l'équation suivante :

$$PMO_i = \frac{1}{M \cdot N} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} I(r_{k,l}) \mid r_{k,l} \in R_i \quad (3.8)$$

Où : PMO_i : probabilité moyenne d'occupation assignée à la région R_i .

R_i : région R d'indice i de l'image.

M et N : dimensions de la région R_i .

$I(r_{k,l})$: intensité du pixel d'indice r de coordonnées (k, l) .

La recherche de régions adjacentes à une région R (coordonnées de l'origine (x_R, y_R) et dimensions (l_R, h_R)) consiste à parcourir l'ensemble des subdivisions. Les coordonnées de l'origine du bloc V (coordonnées de l'origine (x_V, y_V) et dimensions (l_V, h_V)) sélectionné permettent de déterminer si ce dernier est bel et bien un voisin de droite (équations (3.9) et (3.10)), ou du bas (équations (3.11) et (3.12)) de la région R . Le fait d'identifier pour chaque région celles qui lui sont adjacentes à droite et en bas garantit à la fin du parcours de l'ensemble des subdivisions la prise en compte de toutes les paires possibles de régions adjacentes.

$$\left[(x_R + l_R) = x_V \right] ET \left[y_R \leq y_V \leq (y_R + h_R) \right] \quad (3.9)$$

$$\left[(x_R + l_R) = x_V \right] ET \left[y_V \leq y_R \leq (y_V + h_V) \right] \quad (3.10)$$

$$\left[(y_R + h_R) = y_V \right] ET \left[x_R \leq x_V \leq (x_R + l_R) \right] \quad (3.11)$$

$$\left[(y_R + h_R) = y_V \right] ET \left[x_V \leq x_R \leq (x_V + l_V) \right] \quad (3.12)$$

La Figure 3-6-a, la Figure 3-6-b, la Figure 3-6-c et la Figure 3-6-d illustrent respectivement les cas traités dans les équations logiques (3.9), (3.10), (3.11) et (3.12).

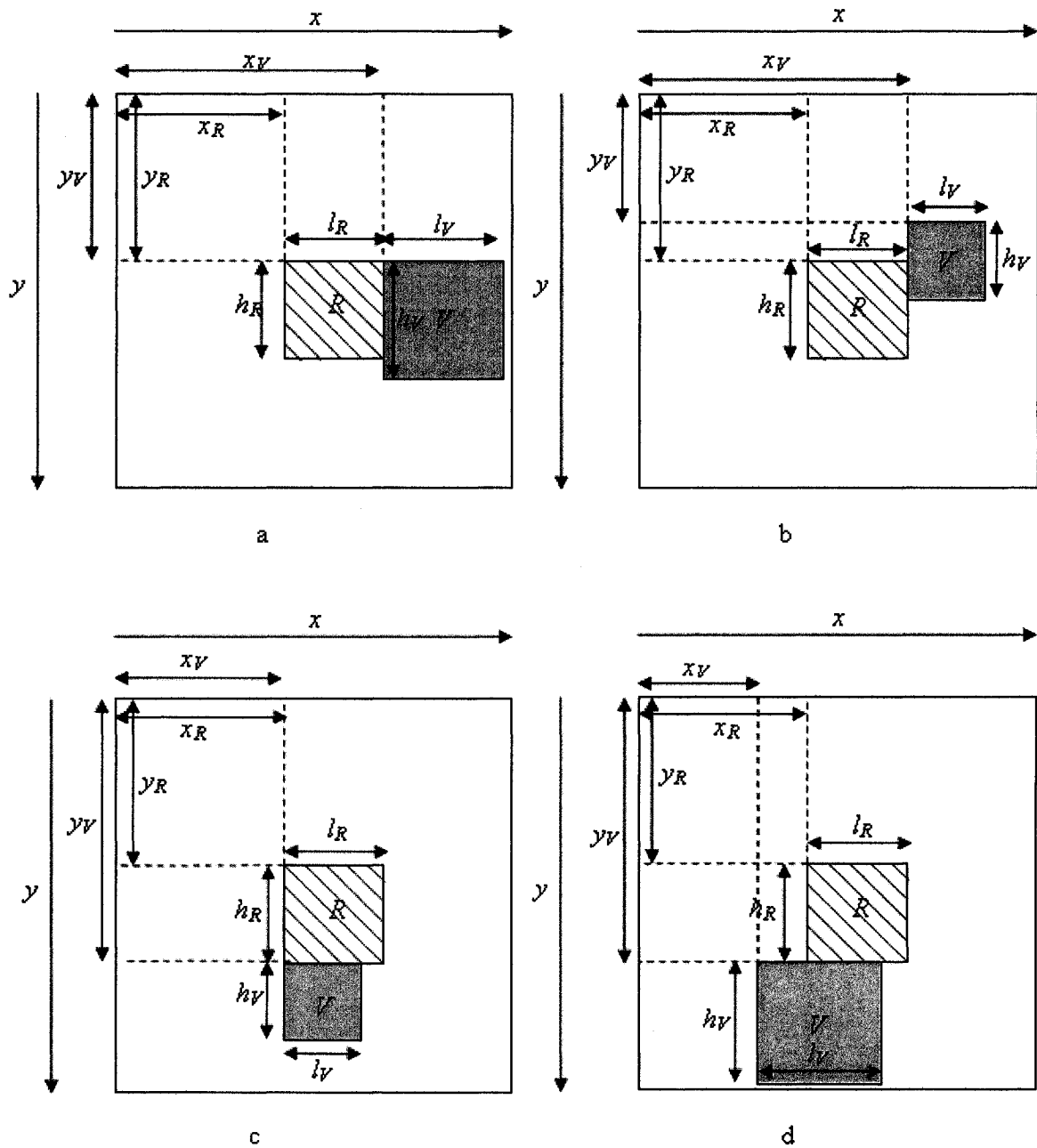


Figure 3-6 : localisation d'un voisin V ((a) et (b)) de droite et ((c) et (d)) du bas de la région R

Le choix de la PMO comme paramètre pour évaluer la similarité des textures présentes dans les régions adjacentes est relié à la structure même des images probabilistes. Dans ces dernières, la valeur des pixels correspond à leur probabilité d'occupation. Après avoir normalisé les valeurs des pixels pour les limiter à l'intervalle $[0;1]$, nous pouvons distinguer deux cas principaux :

- ❖ D'une part, si l'occupation d'une certaine région est totalement inconnue, elle sera caractérisée par une valeur de *PMO* égale à 0.5;
- ❖ D'autre part, dans le cas où une région a été explorée par une ou plusieurs lectures de capteur, trois situations se présentent :
 - La première implique une région dont la majeure partie de l'espace est *libre*, et sera caractérisée par une *PMO* inférieure à 0.5;
 - La seconde correspond au cas inverse où la majeure partie de l'espace est *occupé*, ce qui conduit à une *PMO* supérieure à 0.5;
 - Finalement, dans des cas peu probables, la partie *libre* de l'espace est exactement équivalente à celle qui est *occupée*. Cette situation particulière confère à la région considérée une *PMO* égale à 0.5.

Dans le cas des applications mettant en œuvre des robots mobiles et afin de planifier les mouvements d'une façon sécuritaire, la grille d'occupation probabiliste peut être segmentée en régions caractérisées par des états déterministes S , tel que :

$$S(R_i) \in \{\text{libre}, \text{occupé}, \text{inconnu}\} \quad (3.13)$$

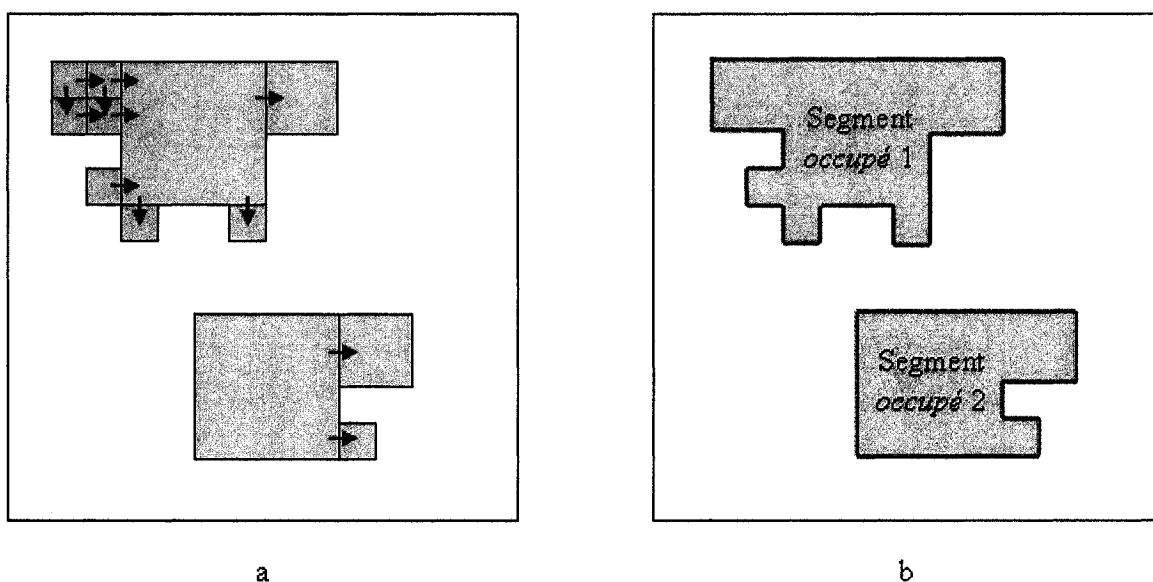
Où : S : fonction d'état déterministe.

R_i : région de l'image probabiliste dotée de l'indice i .

Les trois états *libre*, *inconnu* et *occupé* correspondent respectivement aux intervalles suivants des valeurs de la *PMO* soient $[0;0.498[$, $[0.498;0.502]$ et $]0.502;1]$. Nous avons choisi d'avoir une faible tolérance quant au choix de l'intervalle relatif à l'état *inconnu* afin de garantir que les régions majoritairement *inconnues* avec très peu de cellules *libres* ou *occupées* soient classées selon l'état des cellules prédominantes. Cette considération réduit le nombre de pixels à reclasser durant la phase de raffinement.

En se basant sur ces faits, le processus de création des segments commence par fusionner les subdivisions dont la *PMO* est bornée dans les intervalles $[0;0.498[$ et $[0.498;0.502]$ pour créer respectivement les segments *libre* et *inconnu*. À ce niveau seules les subdivisions considérées comme *occupées* ne sont pas classées.

La construction des segments *occupés* requiert de parcourir les subdivisions *occupées* non classées autant de fois qu'il y a d'objets non adjacents dans l'environnement. Au niveau de chaque parcours, la première subdivision non classée qui est rencontrée est considérée comme étant un segment *occupé* distinct. Par la suite l'algorithme procède à l'identification de ses voisins *occupés* de droite et du dessous (Figure 3-7-a). À chaque fois qu'une telle subdivision est rencontrée, elle est fusionnée avec le segment en cours de construction. Le processus continue itérativement jusqu'à ce qu'aucun ajout ne soit possible. Le fait de considérer les régions *occupées* non adjacentes comme des entités séparées (Figure 3-7-b) permet de distinguer les objets séparés et évite toute confusion possible dans leur reconnaissance et dans l'interaction d'un système robotique avec l'environnement.



Légende

↓ → Direction de l'identification d'une subdivision *occupée* avoisinante



Région *occupée* identifiée comme une entité séparée

Figure 3-7 : processus de construction des segments durant la phase de création des segments. (a) les subdivisions occupées recherchent les voisins de même occupation à droite et au dessous. (b) le regroupement des cellules occupées adjacentes permet d'identifier en segments séparés les objets présents dans l'image probabiliste

Finalement, afin de faciliter la visualisation du résultat de la segmentation à l'aide du module exposé dans l'Annexe B, cette phase assigne une couleur différente aux segments qui caractérisent l'environnement probabiliste (Figure 3-8). Ainsi, les segments *libre* et *inconnu* sont représentés respectivement en noir et en niveau de gris intermédiaire. Dix autres couleurs préprogrammées,

soient le bleu, le vert, le rouge, le marron clair, le mauve, le bleu ciel, l'orange, le marron foncé, le turquoise et le jaune foncé, permettent de caractériser respectivement les dix premiers segments *occupés* qui ont été identifiés. Dans le cas où nous disposons de plus que dix objets, les dix premiers sont représentés par les couleurs énumérées ci-dessus et le reste en blanc.

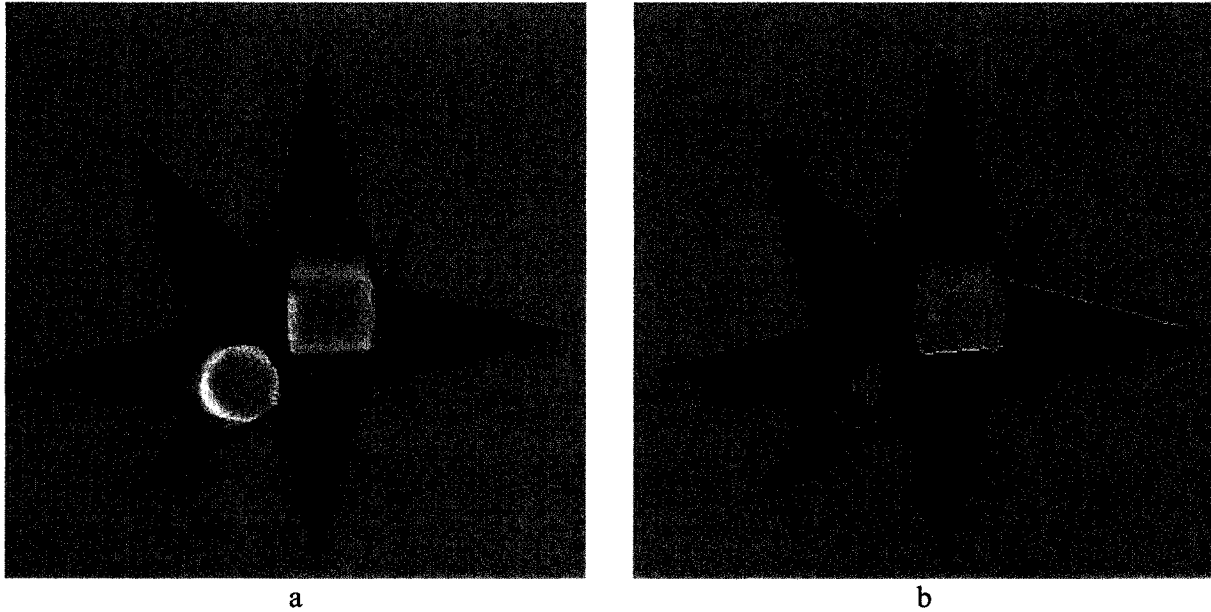


Figure 3-8 : assignation de différentes couleurs aux segments occupés non adjacents. (a) image probabiliste avec deux objets, un circulaire et un autre rectangulaire, (b) résultat de la segmentation où les segments sont représentés par des couleurs différentes : libre (noir), inconnu (gris), objet circulaire (bleu) et objet rectangulaire (vert)

3.3.3 Raffinement

Dans cette dernière phase, les résultats de la segmentation sont raffinés en réassignant les pixels localisés sur la bordure de deux ou plusieurs segments adjacents.

Dans le cas où les phases de division hiérarchique et de création des segments accomplissent avec succès leurs tâches, l'entrée de la phase de raffinement est supposée correspondre à une approximation du résultat final de la segmentation. Ainsi, la tâche principale de cette dernière étape se limite à parfaire la localisation des contours des segments.

La phase de raffinement se base sur le fait qu'une classification *inconnue* (équation (3.13)) est caractérisée par un intervalle très restreint (borné entre 0.498 et 0.502) des valeurs de *PMO*. Ainsi, dans le cas où dans une subdivision, deux espaces, l'un exploré et l'autre pas, coexistent, l'occupation globale de la région sera considérée comme étant connue (soit *libre* soit *occupée*) par la

phase de création des segments même si la majorité des pixels ont une intensité intermédiaire. Il résulte que les cellules dont l'occupation est connue dominent toujours celles qui sont inconnues, et imposent leur état sur les segments dans lesquels elles coexistent. Ceci favorise l'exploration de l'espace tel que représenté dans le modèle d'occupation.

En se basant sur l'observation précédente, un processus de réduction de taille doit être appliqué aux segments *occupés* et *libres* dans le but, premièrement de bien les délimiter, et deuxièmement, de dilater les segments *inconnus*. Au niveau de l'implantation, ce processus consiste à parcourir l'image dans chacune des quatre directions possibles : « droite – gauche », « gauche – droite », « haut – bas » et « bas – haut ». Durant chacune de ces inspections, lorsqu'une bordure est rencontrée entre deux espaces, l'un ayant une occupation connue et l'autre pas, les pixels de l'espace connu dont la valeur est 0.5 sont reclassés comme appartenant à la région inconnue. Ce processus continue jusqu'à ce qu'un pixel doté d'une valeur différente soit atteint. La procédure qui consiste à parcourir l'image dans les quatre directions possibles assure la couverture de toutes les formes possibles de bordures.

La phase de raffinement implémente ce processus de réduction de taille et de dilatation entre les segments étiquetés comme *libres* et les segments *inconnus* qui les longent. Mais, d'une part l'application d'un processus similaire ne peut améliorer le résultat de la segmentation entre les deux types d'espace caractérisés par une occupation connue (*libre* ou *occupée*) pour deux raisons :

- ❖ Les bordures entre les espaces *libres* et *occupés* ne sont pas bien définies et s'étendent selon la marge d'erreur introduite par le modèle du capteur;
- ❖ Les espaces explorés sont caractérisés par la présence des lignes de lecture et d'espaces inconnus entre ces dernières. De ce fait, la reclassification aveugle des pixels dont l'intensité est 0.5 peut créer des trous, ou même diviser une certaine région dotée d'une occupation homogène en plusieurs sections.

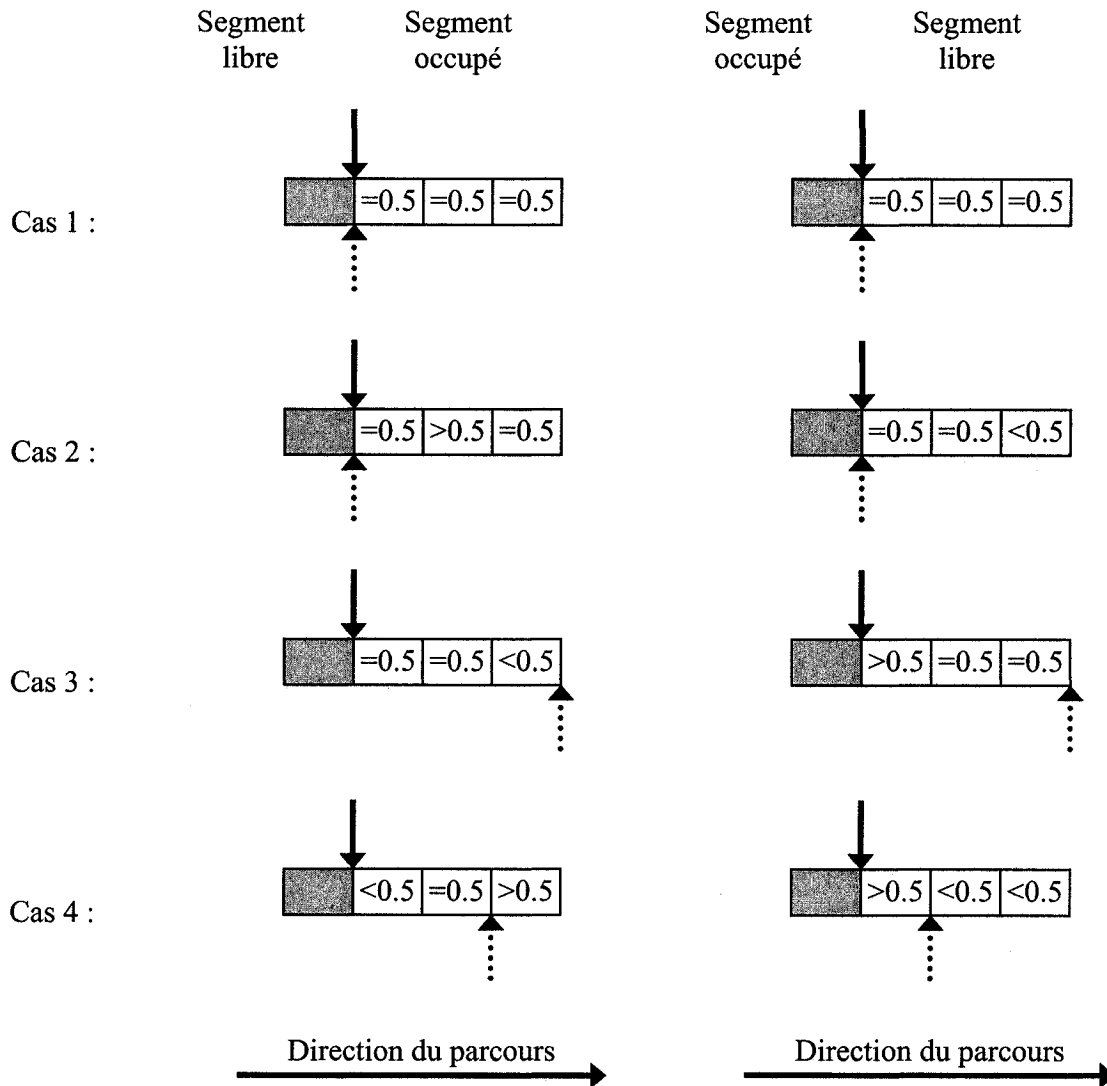
D'autre part, étant donné que les subdivisions initiales ne sont pas très précises, il s'agit de trouver une solution au problème du débordement même mineur entre les régions adjacentes de type *occupé* et *libre*.

Afin de résoudre le problème exposé dans le paragraphe précédent, nous nous sommes inspirés du processus de raffinement entre les segments *inconnus* et *libres*. Ainsi, il s'agit toujours de parcourir deux fois l'image probabiliste dans les quatre directions possibles. Lors du premier

parcours, nous considérons la bordure entre les segments *libres* – *occupés*, tandis que durant le deuxième passage la bordure entre les segments *occupés* – *libres* est examinée. Lors d'un parcours dans une direction donnée, une fois qu'une bordure *libre* – *occupé* (*occupé* – *libre*) est atteinte, nous considérons les trois pixels suivants dans la région adjacente, soit celle qui est *occupée* (*libre*). Le choix de considérer plusieurs pixels au-delà de la bordure entre deux régions d'occupation connue (*libre* ou *occupée*) est lié à l'existence de bandes inconnues entre les rayons de lecture. En effet, toute classification au niveau de ces régions *inconnues* particulières peut créer des discontinuités dans les segments et ainsi fausser la reconnaissance des objets présents dans l'environnement, surtout dans le cas où les bandes *inconnues* sont parallèles aux directions de parcours. Le choix de trois pixels fournit le meilleur rapport sécurité/performance. En terme de sécurité, nous avons voulu distinguer entre les bandes inconnues et les sections des bordures touchées par une lecture, car nous retrouvons souvent le long d'un rayon de lecture une cellule d'intensité égale à 0.5. Celle-ci est due à la transition lente des bordures dans les images probabilistes qui résulte de l'usage d'un modèle d'incertitude gaussien sur les mesures. En terme de performance, le fait de considérer plus que trois cellules au-delà de la bordure ne fournit aucun renseignement-supplémentaire utile au raffinement. Suivant la valeur des trois pixels considérés, nous avons identifié et traité quatre possibilités. La Figure 3-9 illustre les points énumérés ci-dessus, elle fournit un exemple pour chacun des quatre cas et pour chaque type de bordure possible soit entre les segments *libre* – *occupé* à gauche et *occupé* – *libre* à droite.

- ❖ Dans le cas où les trois pixels ont une intensité de 0.5 sur une échelle normalisée, aucune classification n'est appliquée (Cas 1 dans la Figure 3-9) car les chances sont que ces pixels tombent dans la bande *inconnue* entre deux rayons de lecture adjacents et leur classification peut créer des discontinuités dans le segment *occupé* (*libre*);
- ❖ Si au moins un des trois pixels a une intensité strictement supérieure (inférieure) à 0.5 et les autres ont une intensité de 0.5, aucune classification n'a lieu (Cas 2 dans la Figure 3-9). En effet, dans ce cas la bordure entre les segments d'occupation connue correspond à celle des régions libre et occupée dans l'environnement probabiliste;
- ❖ Si au moins une des cellules indique une probabilité d'occupation strictement inférieure (supérieure) à 0.5 et les autres pixels ont une intensité de 0.5 (Cas 3 dans la Figure 3-9), alors les trois pixels changent d'appartenance et sont assignés à la région *libre* (*occupée*).

Cette classification permet de traiter les débordements majeurs d'une région *occupée (libre)* sur une autre *libre (occupée)* qui lui est adjacente;



Légende



Cellule d'intensité a



Cellule dont l'intensité n'est pas évaluée dans le processus de reclassification



Position de la bordure avant le processus de reclassification



Position de la bordure après le processus de reclassification

Figure 3-9 : processus de reclassification des pixels dans la phase de raffinement entre les segments *libre – occupé* d'une part (à gauche) et *occupé – libre* de l'autre (à droite)

- ❖ Le dernier cas correspond à la situation où au moins une des trois cellules a une probabilité d'occupation strictement inférieure (supérieure) à 0.5, et au moins une autre est strictement supérieure (inférieure) à 0.5 (Cas 4 dans la Figure 3-9). Suivant la direction du parcours, les pixels dont la valeur est inférieure (supérieure) ou égale à la valeur médiane sont reclassés jusqu'à ce le premier pixel dont l'intensité n'appartient pas à cet intervalle soit atteint. Cette dernière classification opère un raffinement dans le cas où le débordement d'une certaine région sur une autre est mineur et se limite à un maximum de deux cellules.

La phase de classification des pixels complète les deux étapes précédentes de l'algorithme LBP/C révisé. Bien que le processus de raffinement qu'elle implante paraisse, à première vue, heuristique et spécialisé dans le traitement des images probabilistes, il demeure assez général et robuste pour traiter avec succès de nombreux types d'images comme cela sera démontré au Chapitre 6.

3.4 Conclusion

Dans ce chapitre, nous avons exposé les phases de développement que nous avons suivies et qui ont mené à la conception de l'algorithme LBP/C révisé [50][51]. Ce schéma, spécialisé dans la segmentation des images probabilistes bidimensionnelles correspond à une version révisée et adaptée de l'algorithme d'Ojala et Pietikäinen [39]. Le prochain chapitre fournit une analyse comparative avec le schéma original et une étude expérimentale détaillée de la technique révisée. Notre proposition constitue une plateforme solide sur laquelle diverses applications robotiques peuvent se baser, parmi celles-ci nous retrouvons la planification de trajectoire [51] (section 6.4) et l'interaction avec l'environnement. De plus, notre algorithme démontre de bonnes performances pour la segmentation d'images aériennes et médicales (section 6.5).

Chapitre 4 Étude expérimentale de la technique de segmentation

4.1 Introduction

Ce chapitre est dédié à la présentation et à l'analyse des résultats de la segmentation LBP/C bidimensionnelle. Il se divise en cinq sections, la première d'entre elles compare en terme de conception et d'applicabilité notre schéma de segmentation [50][51] avec celui développé par Ojala et Pietikäinen [39]. La seconde expose les mesures d'erreur qui sont utilisées afin de juger avec objectivité les résultats de l'algorithme proposé. Les trois dernières sections analysent les résultats obtenus ainsi que les effets de la variation de plusieurs paramètres intrinsèques et extrinsèques à l'algorithme, soient le pas angulaire entre les faisceaux de lecture, l'écart-type sur la distance, le niveau d'exploration de l'espace, le nombre de niveaux de discrétisation du contraste ainsi que le nombre de subdivisions opérées.

4.2 Étude comparative des méthodes LBP/C originale et révisée

Comme nous l'avons déjà signalé, l'algorithme de segmentation que nous avons développé est spécialisé dans le traitement des images probabilistes en général, et celles obtenues à l'aide d'un capteur laser de profondeur en particulier. Son objectif est d'extraire à partir des images probabilistes les régions dotées d'une occupation uniforme. À la différence du schéma proposé par Ojala et Pietikäinen [39], l'analyse de la texture constitue un moyen pour atteindre notre but et non pas une fin en soi. Bien que nous nous sommes basés sur le schéma exposé dans [39] afin de concevoir l'approche proposée, chacune des deux méthodes de segmentation vise un type particulier d'images. Les éléments suivants résument les principales différences entre notre algorithme et celui d'Ojala et Pietikäinen [39] :

- ❖ Les textures présentes dans les images de Brodatz [42] utilisées par Ojala et Pietikäinen [39] sont homogènes et les motifs qu'elles contiennent sont redoutants à grande échelle. Ainsi, plus la taille des régions est petite, plus la comparaison des histogrammes de distribution LBP/C à l'aide du critère G-statistique est instable. Le Tableau 4-1 illustre le fait que pour les régions de taille [16 x 16] et [8 x 8] le G-statistique ne fournit pas des résultats

cohérents du fait que la valeur du G-statistique obtenue dans le cas des régions dotées de textures hétérogènes est inférieure à celle obtenue lors de la comparaison des blocs issus d'une texture homogène. Malgré cette observation, la division hiérarchique basée sur le critère G-statistique fournit des résultats de subdivisions très satisfaisants que ce soit avec deux ou trois niveaux de subdivisions. Mais sachant qu'Ojala et Pietikäinen [39] basent la seconde phase (fusion agglomérée) de leur algorithme sur le G-statistique, il nous paraît raisonnable qu'ils aient tenté en n'opérant que deux niveaux de subdivision de minimiser l'influence de cette instabilité sur leur schéma. Il faut signaler que le LBP/C révisé [50] a pris avantage de la qualité de la subdivision hiérarchique basée sur le G-statistique. L'instabilité exposée précédemment n'influence aucunement nos résultats car la seconde étape (création des segments) de l'algorithme proposé n'implique aucune utilisation du G-statistique. Par conséquent, nous sommes permis d'opérer une subdivision supplémentaire. À l'exception des régions inconnues, les espaces touchés par des lectures dans les images probabilistes générées à l'aide de l'outil de simulation développé par Bolzon (Annexe A) contiennent des textures hétérogènes et aléatoires (section 2.3). La subdivision supplémentaire que nous appliquons augmente considérablement les chances que les subdivisions obtenues à la fin de la première phase soient homogènes en terme d'occupation. Ceci minimise la déviation entre le résultat de la seconde phase et les régions dans l'environnement probabiliste et facilite la tâche de la phase de raffinement. Cette observation est soutenue par les résultats expérimentaux que nous avons obtenus et qui sont exposés dans la section 4.6.

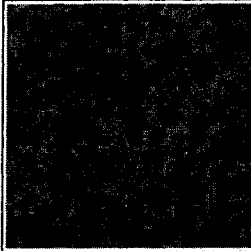

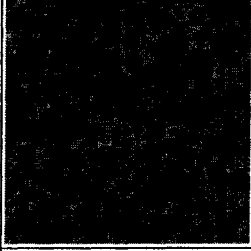
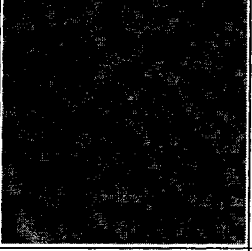
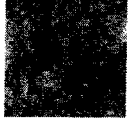

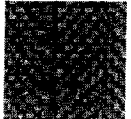









Région 1	Région 2	Dimensions	Texture	G-statistique
		[64 x 64]	Homogène	324.16
		[64 x 64]	Hétérogène	641.44
		[32 x 32]	Homogène	267.06
		[32 x 32]	Hétérogène	718.72
		[16 x 16]	Homogène	186.45
		[16 x 16]	Hétérogène	167.18
		[8 x 8]	Homogène	77.13
		[8 x 8]	Hétérogène	72.63

Tableau 4-1 : comparaison de plusieurs textures issues des images utilisées par Ojala et Pietikäinen [39], les blocs comparés ont été extraits manuellement soit d'une région unique à texture homogène, soit de deux régions dotées d'une texture différente, ce tableau illustre l'incapacité du G-statistique à juger l'uniformité des régions de taille réduite

- ❖ Les régions qui caractérisent les textures dans les images utilisées par Ojala et Pietikäinen [39] ne sont pas fortement contrastantes, mais les distributions LBP/C de leurs subdivisions sont relativement uniformes. Ainsi, l'homogénéité des textures rend l'utilisation du critère G-statistique valable pour les trois phases soient la division hiérarchique, la fusion agglomérée et la classification des pixels [39]. Les images probabilistes sont quant à elles caractérisées par des textures hétérogènes et aléatoires (section 2.3). Malgré que le G-statistique soit incapable de juger convenablement l'uniformité texturale des régions de taille réduite, son intégration dans la division hiérarchique fournit des résultats très

satisfaisants. Mais la présence de textures aléatoires dans les régions explorées de l'environnement combinée à notre choix d'opérer une subdivision supplémentaire par rapport à l'algorithme d'Ojala et Pietikäinen [39], rendent l'utilisation du G-statistique inadaptée pour la seconde étape de notre algorithme (Figure 2-39 c et f). Étant donné que les cellules des images probabilistes correspondent à la probabilité de leur occupation, un contraste suffisant est observé entre les régions *libres*, *occupées* et *inconnues*. La seconde phase de l'algorithme proposé base donc la décision de fusion sur l'occupation moyenne des régions adjacentes plutôt que sur le G-statistique. Étant donné que cette étape n'opère aucune analyse de texture, elle est capable de fusionner deux régions adjacentes de texture hétérogène mais de *PMOs* (équation (3.8)) bornées dans le même intervalle de décision (section 3.3.2).

- ❖ La troisième phase de l'algorithme proposé opère un processus de raffinement différent suivant les caractéristiques des régions traitées. Ainsi, elle élimine le débordement des segments *libres* sur les régions *inconnues* qui leurs sont adjacentes et effectue un raffinement de la bordure entre les segments *libres* et *occupés* tout en traitant les discontinuités dues à l'exploration partielle des lectures des capteurs. Le raffinement se base principalement sur l'analyse de la probabilité d'occupation des cellules dans une région de taille restreinte sur la bordure entre deux segments. Dans les images de Brodatz [42], la présence de cellules dotées d'une intensité variable au sein d'une même texture peut causer un arrêt prématuré de la phase de raffinement (Figure 4-1 c et d ainsi que Figure 4-2 c et d) de notre algorithme car cette dernière définit les bordures suivant la probabilité d'occupation des cellules rencontrées dans la direction d'un parcours donné. À titre d'exemple, la présence d'un pixel d'occupation *libre* (intensité inférieure à 0.498) ou *occupée* (intensité supérieure à 0.502) dans les images de texture [42] n'indique pas nécessairement une délimitation entre deux segments dotés de textures hétérogènes.

Jusqu'à la première phase, les deux schémas de segmentation, soient le LBP/C original [39] et le LBP/C révisé [50], ne diffèrent que de la subdivision supplémentaire que nous opérons (Figure 4-3 et Figure 4-4). L'algorithme de segmentation d'Ojala et Pietikäinen [39] procède dans ses étapes subséquentes à une analyse texturale, et fournit des résultats convenables sur les images de Brodatz [42] caractérisées par des régions à texture homogène. Son incapacité à traiter les images

probabilistes (section 2.4.7.4) nous a motivé à réviser la conception de la seconde et troisième phase qui se basent sur l'analyse de l'occupation de l'espace et non pas de la texture.

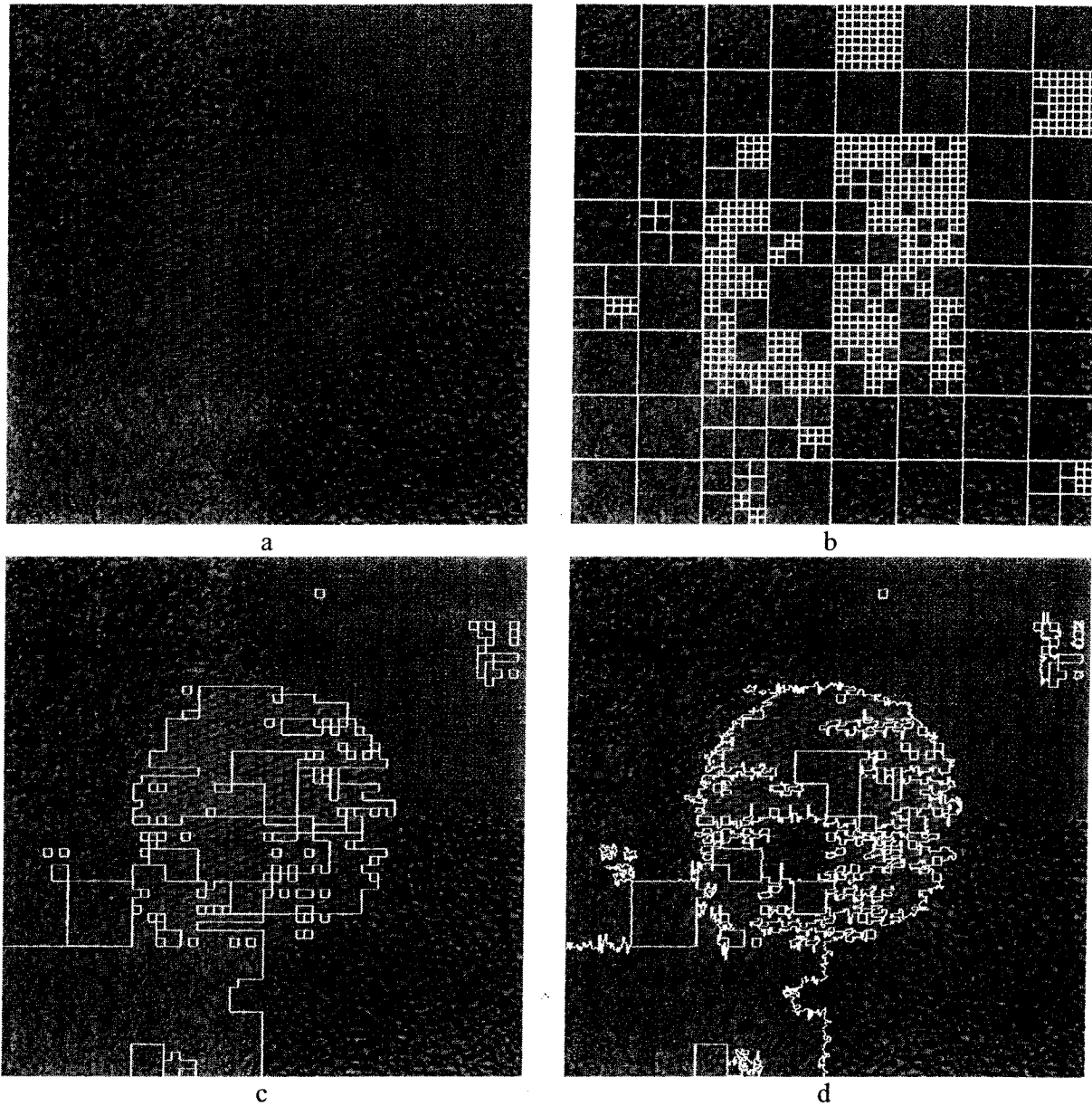
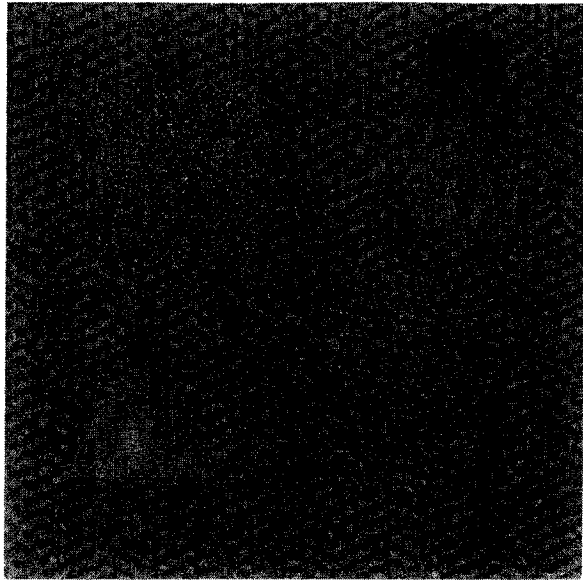
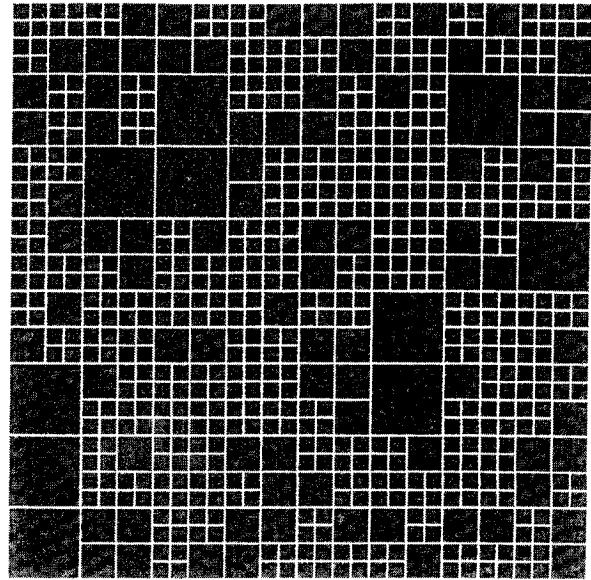


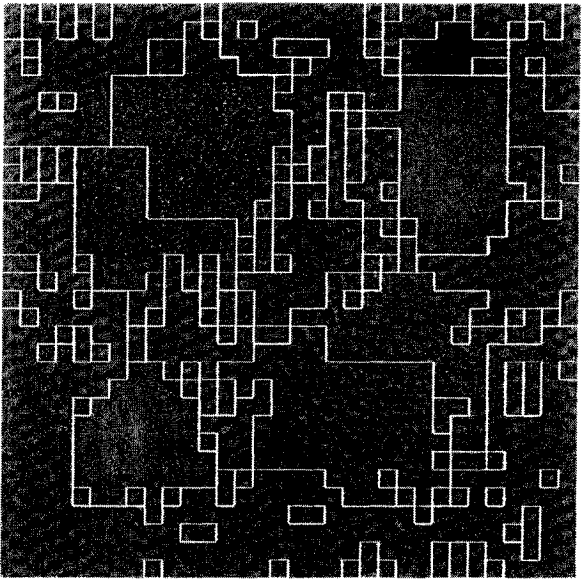
Figure 4-1 : (a) image extraite de l'album de Brodatz [42] sur laquelle Ojala et Pietikäinen [39] ont testé leur algorithme. (b), (c) et (d) résultats des trois phases de l'algorithme proposé



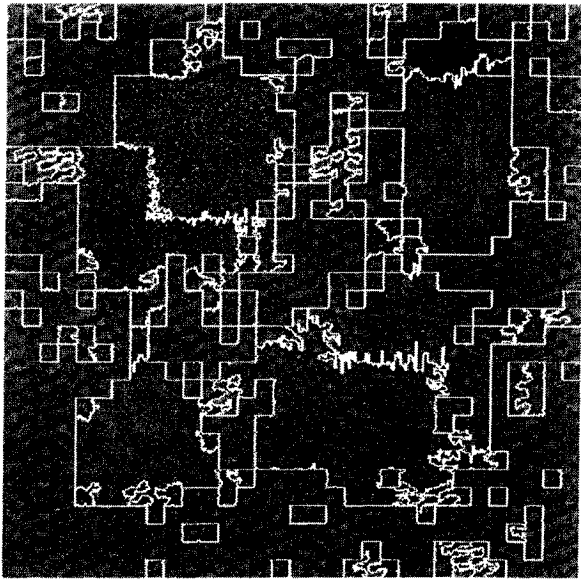
a



b



c



d

Figure 4-2 : (a) image extraite de l'album de Brodatz [42] sur laquelle Ojala et Pietikäinen [39] ont testé leur algorithme. (b), (c) et (d) résultats des trois phases de l'algorithme proposé

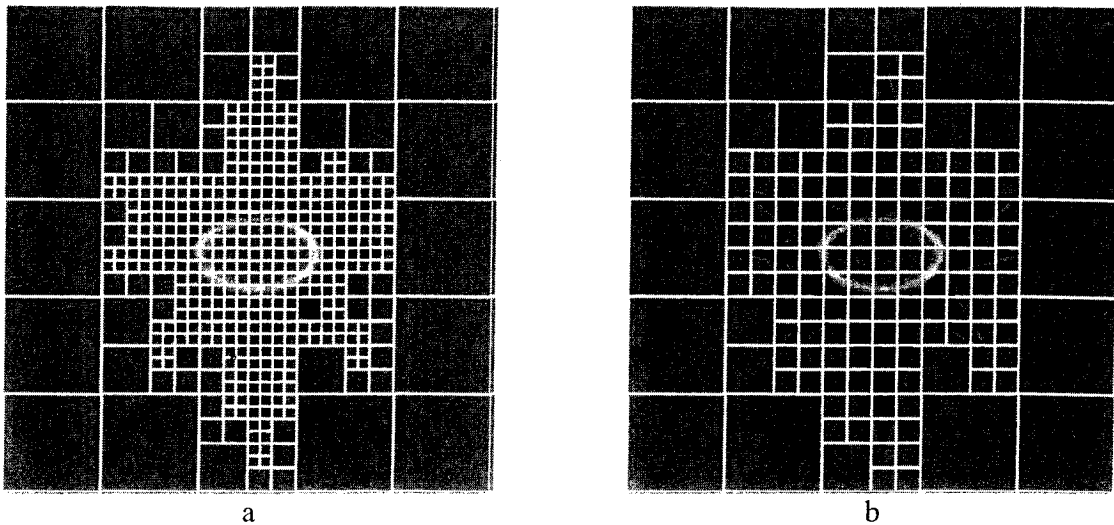


Figure 4-3 : résultats de la première phase: (a) de l'algorithme proposé, (b) du schéma d'Ojala et Pietikäinen [39] sur une image image probabiliste, jusqu'à cette phase les deux schémas ne diffèrent que de la subdivision supplémentaire que nous opérons

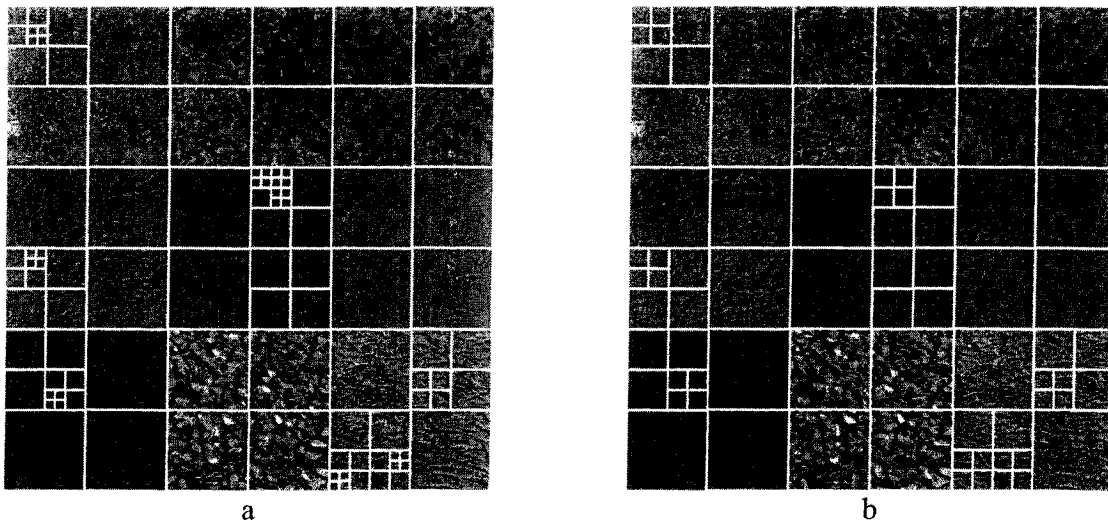


Figure 4-4 : résultats de la première phase: (a) de l'algorithme proposé, (b) du schéma d'Ojala et Pietikäinen [39] sur une image extraite de l'album de Brodatz [42], jusqu'à cette phase les deux schémas ne diffèrent que de la subdivision supplémentaire que nous opérons

4.3 Mesures d'erreur pour évaluer le résultat de la segmentation

Afin de porter un jugement quantitatif et objectif concernant les résultats de l'algorithme de segmentation proposé, nous avons développé une mesure d'erreur capable de juger le pourcentage de déviation entre les classifications obtenues et les régions présentes dans l'environnement. Étant donné que les images probabilistes segmentées ont été générées à l'aide de l'outil de simulation

développé par Bolzon (Annexe A), une représentation de l'environnement réel est disponible. De ce fait, il est possible d'analyser à l'aide d'une mesure d'erreur à deux niveaux la déviation entre les segments classifiés *occupés* et *libres* et les régions correspondantes de l'espace simulé :

- ❖ L'erreur de type 1 calcule, compte tenu de l'information récoltée par les capteurs, le nombre de cellules qui ont été classifiées comme *occupées* mais qui sont réellement libres par rapport au nombre total de cellules dont l'état a été mesuré par les capteurs. Seules les cellules réellement libres sont utilisées dans le calcul de cette mesure de déviation car un environnement simulé se caractérise par la certitude totale de son occupation. En effet, les cellules qu'il contient ont une occupation discrète, du fait que leur état est soit libre soit occupé. Cette erreur est calculée à l'aide de l'équation suivante :

$$\xi(\text{type 1}) = \frac{ol}{t} \quad (4.1)$$

Où : ol : nombre de cellules classées comme *occupées* mais qui sont réellement libres.

t : nombre total de cellules qui ont été touchées par un faisceau de lecture.

- ❖ L'erreur de type 2 correspond au cas inverse du fait qu'elle calcule, compte tenu des informations récoltées par les capteurs, le nombre de cellules qui ont été classifiées comme *libres* mais qui sont réellement occupées par rapport au nombre total de cellules dont l'état a été mesuré par les capteurs. L'observation faite pour le cas de l'erreur de type 1 est aussi valable dans ce cas. Ainsi, seules les cellules réellement occupées sont considérées, car les environnements simulés sur lesquels se base le calcul de cette mesure de déviation se caractérisent par la certitude de leur occupation. Par conséquent, une occupation intermédiaire (par exemple *inconnue*) est dans leur cas impossible. La mesure de l'erreur de type 2 est calculée comme suit :

$$\xi(\text{type 2}) = \frac{lo}{t} \quad (4.2)$$

Où : lo : nombre de cellules classées comme *libres* mais qui sont réellement *occupées*.

t : nombre total de cellules qui ont été touchées par un faisceau de lecture.

Les mesures d'erreur introduites ci-dessus renseignent sur le degré de débordement des segments *occupés* (type 1) ou *libres* (type 2) par rapport à aux régions réelles qu'ils représentent. En effet, une erreur de type 1 (type 2) élevée implique que la taille des segments classifiés *occupés* (*libres*) a été sur-évaluée par rapport aux régions réellement occupées (*libres*) correspondantes. En terme d'applications en planification de trajectoire robotique, la mesure de l'erreur de type 2 est plus critique que celle de type 1, du fait que la sous-évaluation de la taille des objets réels s'avère dangereuse étant donné qu'ils constituent une source de collisions. Par contre, dans un contexte d'interaction avec l'environnement (saisie, mesures tactiles) les deux types d'erreur prennent tout leur sens.

4.4 Effets de la variation de l'écart-type et du pas angulaire des capteurs

Comme signalé précédemment, l'algorithme proposé est principalement destiné à la segmentation des grilles d'occupation probabilistes. De plus, il se caractérise par sa capacité à traiter les discontinuités dans les régions dotées d'une occupation connue. Tout comme les paramètres intrinsèques à l'algorithme, les paramètres extrinsèques liés aux configurations des capteurs utilisés influencent premièrement les informations présentes dans les grilles d'occupation et deuxièmement le résultat de la segmentation. Dans cette partie, nous analysons l'effet de la variation de l'écart-type de l'erreur sur la distance mesurée par le capteur et celui du pas angulaire entre les faisceaux de lecture. Le Tableau 4-2 illustre les résultats de la segmentation de trois simulations d'environnements réels (Figure 4-5 a, b et c) de taille [448 x 448] pixels à l'aide de l'implantation de l'algorithme LBP/C révisé réalisée avec le langage de programmation C et la librairie OpenCV¹⁵ qui lui est compatible (Annexe B), sur un ordinateur doté d'un processeur Athlon 2600 XP [AMD Inc.] avec un giga-octet de mémoire vive. À partir de chacun des environnements simulés, nous avons généré douzes images probabilistes à l'aide du simulateur développé par Bolzon. Dans chacun des trois sous-ensembles de grilles d'occupation, la position ainsi que la direction des capteurs sont les mêmes, seules les valeurs des paramètres des capteurs ont été variées. Ainsi, quatre valeurs de l'écart-type (3, 5, 7 et 9 cm) et trois pas angulaires (0.5, 1 et 1.5 degré) ont été considérés. Il faut noter que les valeurs du pas angulaire choisies sont exagérées par rapport à la résolution angulaire des capteurs télémétriques commerciaux. En effet, le capteur Jupiter [10], duquel le simulateur de

¹⁵ « Open Computer Vision Library ».

Bolzon s'inspire, se caractérise par un pas angulaire de 0.14 ou de 0.07 degré selon le mode de fonctionnement choisi. Nous avons surestimé la valeur du pas angulaire afin de valider la capacité de l'approche proposée à résoudre le problème du balayage partiel de l'environnement, même dans des conditions atypiques.

Notre algorithme fournit des résultats de segmentation convaincants quelque soient les configurations des capteurs utilisées dans la génération des grilles d'occupation. En effet, pour l'ensemble des images, les moyennes de l'erreur de type 1 et de type 2 sont relativement faibles et se situent respectivement à 8.108% et 0.606%. Pour une configuration de l'espace donnée, le temps d'exécution demeure stable quelque soit la configuration du capteur. La variation observée d'un sous-ensemble d'images à un autre a pour origine la complexité des textures ainsi que le nombre de subdivisions requises durant la première phase (section 4.5). Les avantages de la troisième phase ne se limitent pas à la perfection visuelle du résultat, elle permet outre le raffinement entre les segments *libre* et *inconnu*, de réduire en moyenne les erreurs de type 1 et de type 2 respectivement de 0.373% et de 0.263%.

Le Tableau 4-3 illustre pour chaque valeur du pas angulaire la moyenne des variations des mesures d'erreur suite à l'incrément de l'écart-type de 2 cm. Ainsi l'erreur de type 1 s'accroît avec l'incrément de l'écart-type de l'erreur du capteur avec une moyenne de +1.471%, de +1.129% et de +3.787% respectivement pour les images probabilistes issues des environnements simulés *Surface en U* (Figure 4-5-a), *Carré et disque* (Figure 4-5-b) et *Forme complexe* (Figure 4-5-c). Cette corrélation positive implique que l'étalement de la transition lente de la probabilité d'occupation au niveau des bordures créée par une augmentation de l'écart-type sur la mesure de la distance fournie par le capteur tend à surévaluer légèrement la taille des objets. N'empêche que l'effet de l'augmentation de l'écart-type sur la distance est négligeable sur l'erreur de type 2 dont la variation maximale observée est de -0.416%. Cette valeur négative est désirable étant donné qu'elle implique une réduction du nombre de cellules réellement occupées mais qui ont été classées comme *libres*. L'algorithme a donc tendance à surestimer la taille des objets lorsque l'incertitude sur les mesures du capteur augmente, ce qui est prévisible et hautement désirable dans le contexte d'évitement des collisions en robotique considéré dans ces travaux.

Espace réel traité	Écart-type (cm)	Pas angulaire (degré)	Erreur type 1 (%)		Erreur type 2 (%)		Temps d'exécution (secondes)			
			Phase 2	Phase 3	Phase 2	Phase 3	Phase 1	Phase 2	Phase 3	Total
<i>Surface en U</i> (Figure 4-5-a)	3	0.5	1.44	1.70	0.48	0.00	9.56	0.03	0.16	9.75
		1.0	1.71	1.98	0.41	0.06	9.28	0.03	0.16	9.47
		1.5	2.16	2.65	0.60	0.14	9.29	0.03	0.16	9.48
	5	0.5	3.76	2.94	0.07	0.00	9.34	0.03	0.16	9.53
		1.0	4.26	3.53	0.08	0.01	9.18	0.03	0.16	9.37
		1.5	5.13	4.62	0.31	0.09	9.29	0.03	0.16	9.48
	7	0.5	4.66	4.11	0.03	0.00	9.65	0.04	0.16	9.85
		1.0	4.98	4.95	0.22	0.01	9.34	0.03	0.16	9.53
		1.5	6.49	6.44	0.04	0.02	9.43	0.04	0.16	9.63
	9	0.5	6.39	5.36	0.01	0.00	9.73	0.03	0.17	9.93
		1.0	7.03	6.26	0.01	0.00	9.26	0.04	0.16	9.46
		1.5	7.28	7.95	0.29	0.06	9.15	0.03	0.17	9.35
<i>Carré et disque</i> (Figure 4-5-b)	3	0.5	3.23	2.04	0.83	0.11	6.15	0.03	0.28	6.46
		1.0	4.00	2.47	1.02	0.16	6.29	0.03	0.28	6.60
		1.5	4.83	3.23	0.84	0.08	6.26	0.01	0.28	6.55
	5	0.5	3.31	3.13	0.60	0.07	6.32	0.01	0.30	6.62
		1.0	4.51	3.97	0.34	0.04	6.35	0.03	0.28	6.66
		1.5	5.46	4.80	0.55	0.12	6.48	0.03	0.28	6.79
	7	0.5	3.85	3.88	0.21	0.06	6.90	0.03	0.28	7.21
		1.0	5.00	5.21	0.31	0.05	6.90	0.02	0.28	7.20
		1.5	5.88	6.20	0.49	0.11	6.95	0.01	0.28	7.24
	9	0.5	4.46	4.50	0.22	0.06	6.88	0.03	0.28	7.19
		1.0	5.15	6.16	0.30	0.07	6.90	0.03	0.28	7.21
		1.5	7.00	7.24	0.37	0.07	6.85	0.03	0.28	7.16
<i>Forme complexe</i> (Figure 4-5-c)	3	0.5	9.59	8.74	2.15	2.02	7.10	0.03	0.51	7.64
		1.0	10.16	9.53	2.49	2.32	6.93	0.03	0.53	7.49
		1.5	12.86	11.85	2.77	2.60	7.01	0.02	0.53	7.55
	5	0.5	12.81	12.22	1.84	1.52	7.59	0.03	0.28	7.90
		1.0	13.53	12.83	1.82	1.79	7.29	0.03	0.16	7.47
		1.5	15.96	15.51	2.30	2.19	7.20	0.01	0.28	7.50
	7	0.5	15.92	15.99	2.31	1.30	7.42	0.03	0.16	7.61
		1.0	16.61	16.45	1.55	1.52	7.31	0.03	0.17	7.51
		1.5	19.39	19.22	1.95	1.86	7.39	0.03	0.16	7.58
	9	0.5	19.83	19.01	0.95	0.89	7.48	0.03	0.16	7.67
		1.0	21.08	20.41	1.09	1.12	7.51	0.03	0.16	7.70
		1.5	25.57	24.78	1.33	1.19	7.45	0.03	0.16	7.64

Tableau 4-2 : variation du temps d'exécution et des erreurs de type 1 et de type 2 en fonction de l'écart-type sur la distance et du pas angulaire du capteur

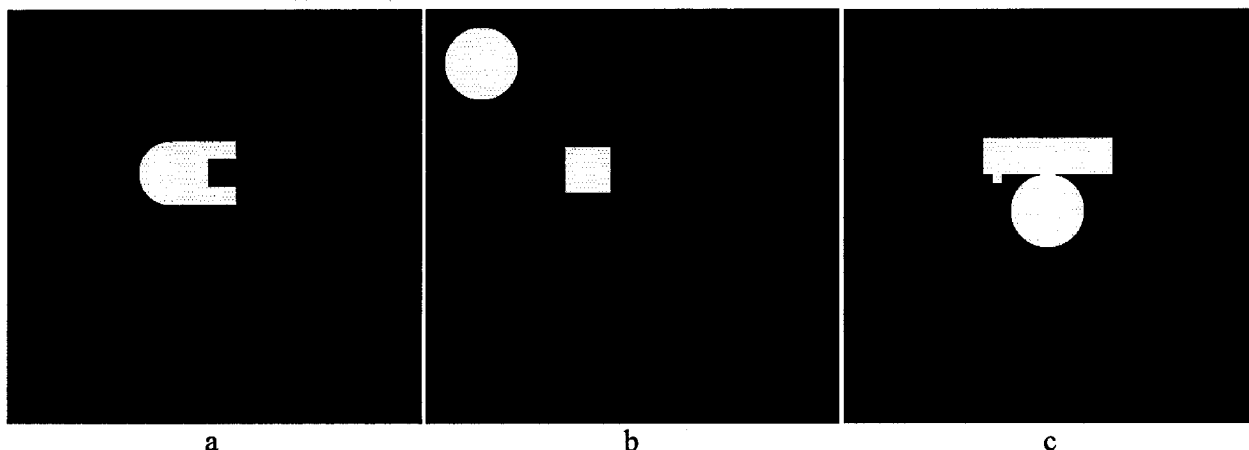


Figure 4-5 : environnements simulés de taille [448 x 448] utilisés dans la génération des grilles d'occupation et des résultats du Tableau 4-2, du Tableau 4-3, du Tableau 4-4 et du Tableau 4-6. (a) *Surface en U*, (b) *Carré et disque*, (c) *Forme complexe*

Espace réel traité	Pas angulaire (degré)	Moyenne des variations des mesures d'erreur pour les incréments de l'écart-type de 2 cm (%)		Moyenne des moyennes de variations des mesures d'erreur (%)	
		Erreur type 1	Erreur type 2	Erreur type 1	Erreur type 2
<i>Surface en U</i> (Figure 4-5-a)	0.5	+1.22	0.00	+1.47	-0.01
	1.0	+1.43	-0.02		
	1.5	+1.76	-0.02		
<i>Carré et disque</i> (Figure 4-5-b)	0.5	+0.82	-0.01	+1.13	-0.01
	1.0	+1.23	-0.03		
	1.5	+1.33	0.00		
<i>Forme complexe</i> (Figure 4-5-c)	0.5	+3.42	-0.37	+3.79	-0.42
	1.0	+3.63	-0.40		
	1.5	+4.31	-0.47		

Tableau 4-3 : variation des erreurs de type 1 et de type 2 en fonction de celle de l'écart-type de l'erreur du capteur sur la distance

Les résultats illustrés dans le Tableau 4-4 démontrent la stabilité de l'algorithme proposé quant à la variation du pas angulaire entre les faisceaux de lecture. En effet, l'erreur de type 1 qui renseigne sur la proportion de cellules réellement libres mais classées comme *occupées* varie légèrement avec les incréments du pas angulaire de 0.5 degré. Bien que cette variation augmente avec l'écart-type de l'erreur du capteur, elle demeure inférieure à +3%, et sa moyenne s'établit à +0.94%, à +0.99% et à +1.63% respectivement dans le cas des images probabilistes issues des environnements simulés *Surface en U* (Figure 4-5-a), *Carré et disque* (Figure 4-5-b) et *Forme complexe* (Figure 4-5-c). La variation de l'erreur de type 2 qui est en terme d'applications robotiques plus critique que celle de type 1, est presque négligeable avec une moyenne de +0.039%, de +0.01%

et de +0.262% pour les images probabilistes issues respectivement de *Surface en U* (Figure 4-5-a), *Carré et disque* (Figure 4-5-b) et *Forme complexe* (Figure 4-5-c).

Espace réel traité	Écart-type (cm)	Moyenne des variations des mesures d'erreur pour les increments du pas angulaire de 0.5 degré (%)		Moyenne des moyennes de variations des mesures d'erreur (%)	
		Erreur type 1	Erreur type 2	Erreur type 1	Erreur type 2
<i>Surface en U</i> (Figure 4-5-a)	3	+0.47	+0.07	+0.94	+0.04
	5	+0.84	+0.04		
	7	+1.16	+0.01		
	9	+1.29	+0.03		
<i>Carré et disque</i> (Figure 4-5-b)	3	+0.59	-0.01	+0.99	+0.01
	5	+0.83	+0.02		
	7	+1.16	+0.02		
	9	+1.37	0.00		
<i>Forme complexe</i> (Figure 4-5-c)	3	+0.39	+0.29	+1.63	+0.26
	5	+1.64	+0.33		
	7	+1.62	+0.28		
	9	+2.88	+0.15		

Tableau 4-4 : variation des erreurs de type 1 et de type 2 en fonction de celle du pas angulaire entre les faisceaux de lecture

Bien que la qualité des images probabilistes se détériore avec l'augmentation de l'écart-type sur la distance et du pas angulaire, notre algorithme fournit des résultats quantitatifs stables et cohérents. La faible variation des erreurs de type 1 et de type 2 et du temps d'exécution en sont les preuves quantitatives. Afin d'évaluer qualitativement notre schéma de segmentation, nous avons inclus les résultats de l'algorithme pour un sous-ensemble des simulations dont les résultats quantitatifs sont exposés dans le Tableau 4-2.

En premier lieu, la Figure 4-6, la Figure 4-7, la Figure 4-8 et la Figure 4-9 illustrent les images probabilistes issues de l'environnement simulé *Surface en U* (Figure 4-5-a) ainsi que les résultats de leur segmentation. Les capteurs utilisés dans ces quatre exécutions se caractérisent par le même pas angulaire, soit 0.5 degré, mais par un écart-type sur la distance variable dont les valeurs respectives sont de 3 (Figure 4-6), de 5 (Figure 4-7), de 7 (Figure 4-8) et de 9 (Figure 4-9) cm. Nous remarquons que plus la valeur de l'écart-type est élevée plus les segments *occupés* sont profonds (en terme d'épaisseur). Cette observation est due au fait que l'écart-type correspond à l'erreur du capteur sur la distance et par conséquent elle définit la lenteur de la transition sur la bordure entre les régions réellement libres et occupées. Les résultats exposés ci-dessous démontrent que l'algorithme proposé,

sans aucune connaissance préalable de l'erreur introduite par le capteur, est capable de représenter fidèlement les régions présentes dans l'environnement. En effet, la valeur de l'erreur de type 1 relative aux résultats exposés dans la Figure 4-6-d, la Figure 4-7-d, la Figure 4-8-d et la Figure 4-9-d varie avec l'écart-type et augmente de 1.7% à 5.36% (Tableau 4-2), tandis que l'erreur de type 2 demeure presque nulle. Ceci indique que le segment *occupé* englobe totalement l'objet présent dans l'environnement simulé, et déborde légèrement sur l'espace réellement libre qui l'entoure, tel que requis par la tâche de planification de trajectoire robotique.

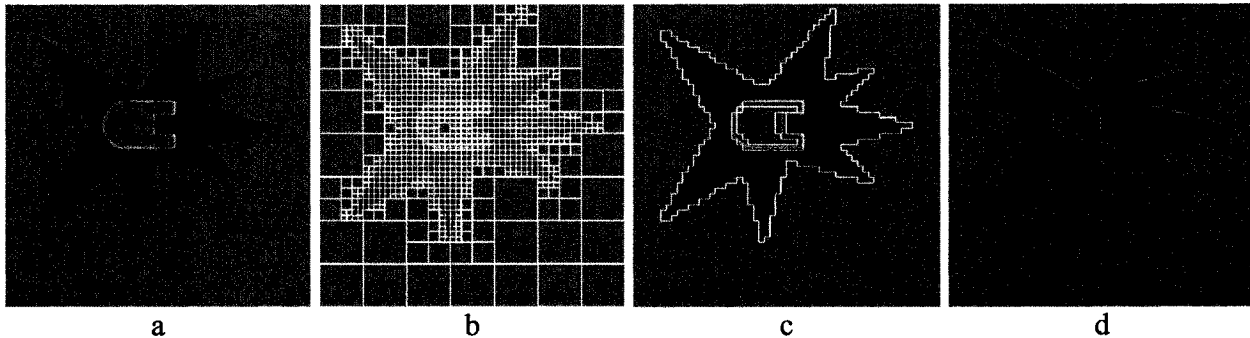


Figure 4-6 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

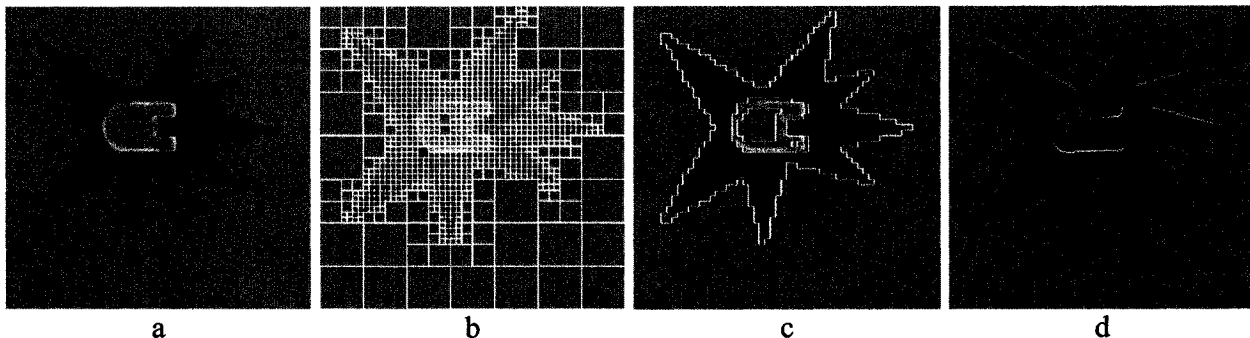
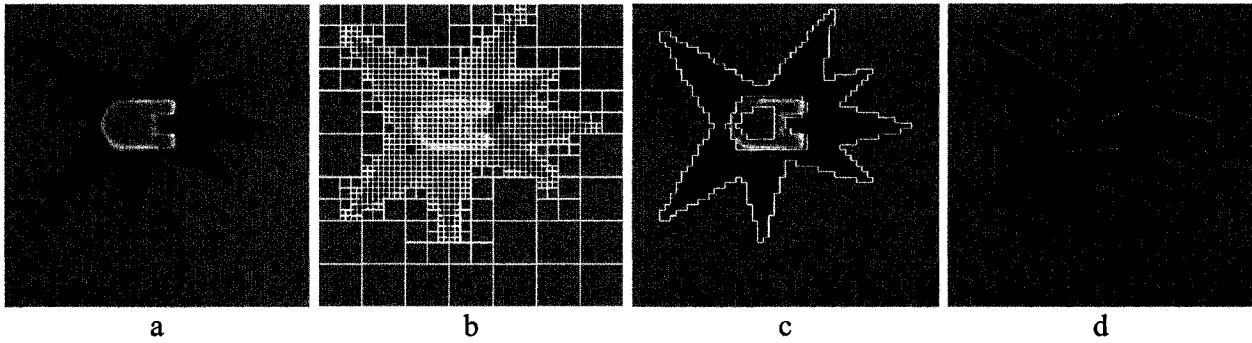
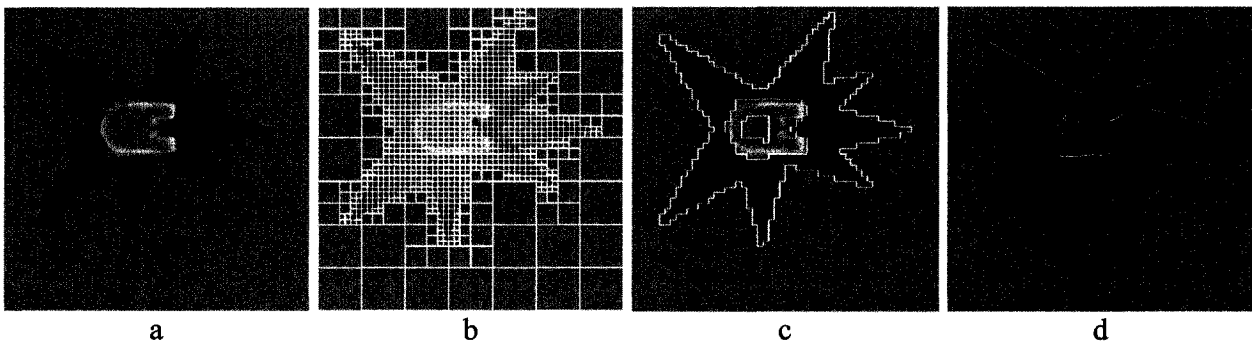


Figure 4-7 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 5 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)



a b c d
Figure 4-8 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 7 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)



a b c d
Figure 4-9 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 9 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

En second lieu, la Figure 4-10, la Figure 4-11 et la Figure 4-12 illustrent les images probabilistes issues de l'environnement simulé *Carré et disque* (Figure 4-5-b) ainsi que les résultats de leur segmentation. À la différence des quatre simulations exposées précédemment, les écarts-types des capteurs ont été fixés à 3 cm, tandis que les pas angulaires sont de 0.5, 1 et 1.5 degré respectivement dans la Figure 4-10, la Figure 4-11 et la Figure 4-12. À l'œil nu, aucune différence notable ne peut être observée entre les résultats exposés dans la Figure 4-10-d, la Figure 4-11-d et la Figure 4-12-d. Même les mesures d'erreur relatives à ces segmentations (Tableau 4-2) ne varient que légèrement. En effet, l'erreur de type 1 augmente avec le pas angulaire de 2.04% à 3.23% tandis que l'erreur de type 2 demeure négligeable (inférieure à 0.16%). Ces mesures d'erreur indiquent comme dans le cas des simulations précédentes que le segment *occupé* englobe totalement les sections de l'objet réel qu'il représente et que son débordement sur les régions réellement libres est mineur. La Figure 4-10-d, la Figure 4-11-d et la Figure 4-12-d illustrent de même la capacité de l'algorithme

proposé de reconnaître en segments séparés les régions occupées non-adjacentes de l'environnement probabiliste. En effet, les deux parties colorées dans ces images correspondent chacune à une partie des objets qu'ils représentent, soit le disque (segment coloré en bleu en haut à gauche) et le carré (segment coloré en vert dans la partie centrale), les régions *libres* et *inconnues* sont quant à elles identifiées respectivement en noir et en niveau de gris intermédiaire. Ainsi, les résultats de la segmentation extrapolent les informations d'occupation entre les faisceaux de lecture et fournissent des segments spatialement uniformes et dotés d'une occupation déterministe.

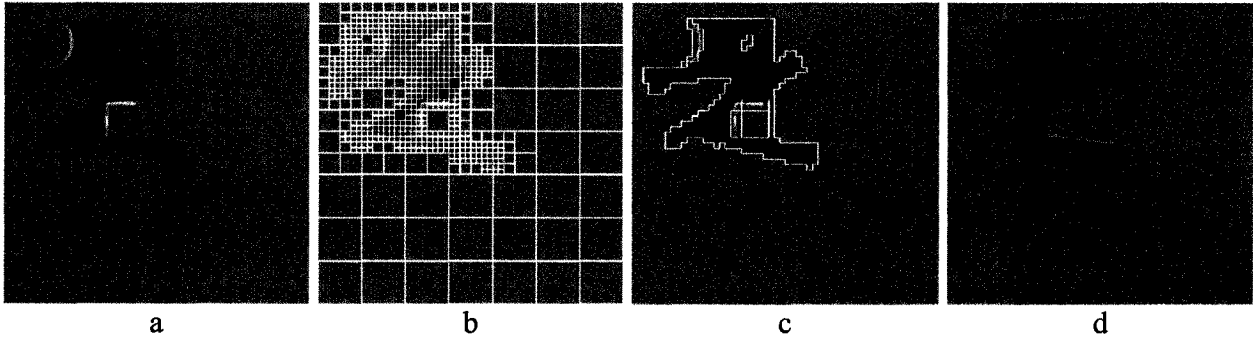


Figure 4-10 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-b avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu et vert)

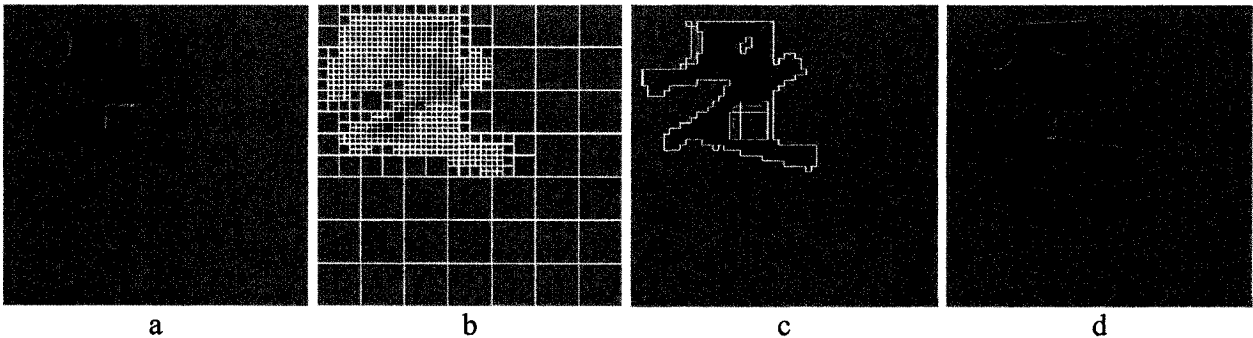
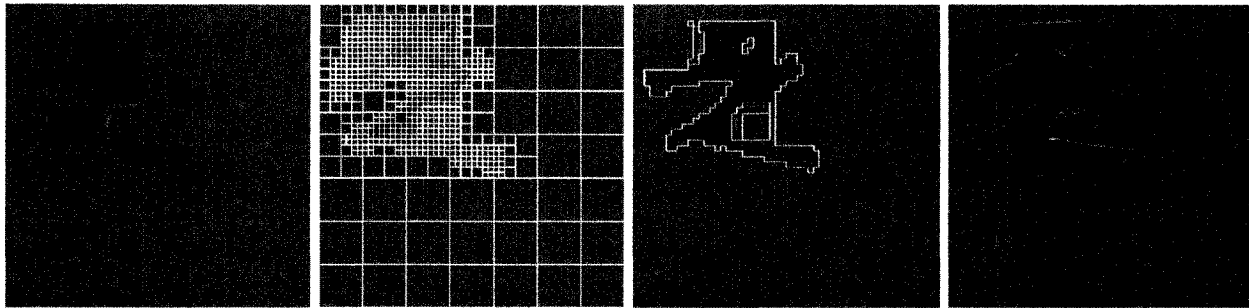


Figure 4-11 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-b avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 1 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu et vert)



a b c d
Figure 4-12 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-b avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 1.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu et vert)

En troisième lieu, nous exposons dans la Figure 4-13 et la Figure 4-14 deux cas extrêmes dans lesquels l'objet présent dans l'environnement simulé a une forme complexe (Figure 4-5-c) et où les paramètres des capteurs offrent une piètre performance. En effet, l'objet présent dans l'environnement *Forme complexe* (Figure 4-5-c) se caractérise par la présence d'une section circulaire qui empêche les deux points de vue du bas d'atteindre la partie rectangulaire du haut (Figure 4-13-a et Figure 4-14-a). De plus, la section circulaire est presque tangente à la région rectangulaire (Figure 4-5-c). Ceci fait que les régions situées des deux côtés de leur intersection sont relativement étroites et par conséquent ne sont accessibles que par un nombre très limité de faisceaux de lecture. À ceci s'ajoute la présence de la petite région rectangulaire (à gauche de l'objet) qui cache encore plus la section à angle aigu de gauche. À la complexité de l'objet présent dans l'environnement simulé (Figure 4-5-c) s'ajoutent les paramètres du capteur qui se caractérisent par un pas angulaire de 1.5 degré et un écart-type sur la distance de 7 (Figure 4-13) et de 9 (Figure 4-14) cm. Bien que la forme des segments obtenus (Figure 4-13-d et Figure 4-14-d) reflètent convenablement celle des régions de l'environnement simulé (Figure 4-5-c) qu'ils représentent, les mesures d'erreur obtenues demeurent relativement élevées par rapports aux cas exposés précédemment. En effet, l'erreur de type 1 est de 19.228% et de 24.78% respectivement pour les résultats exposés dans la Figure 4-13-d et la Figure 4-14-d. Ceci illustre le fait que la combinaison d'une valeur d'écart-type et d'un pas angulaire élevés rend l'image probabiliste très floue surtout dans les régions où les objets sont caractérisés par des coins (intersection entre la région circulaire et la zone rectangulaire au centre de l'objet, ainsi que le coin entre la petite et la grande région rectangulaire en haut à gauche de l'objet). Les images probabilistes (Figure 4-13-a et Figure 4-14-a)

obtenues à l'aide d'un capteur à faible résolution angulaire et spatiale sont incapables de reproduire certains détails présents dans l'environnement simulé (Figure 4-5-c). Ceci accentue le débordement du segment *occupé* sur la région libre qui l'entoure. Malgré cette limitation, l'erreur de type 2 demeure faible avec une valeur inférieure à 1.9%.

D'autre part, nous remarquons le long de la bordure entre les segments *occupé* et *libre* dans la Figure 4-13-d et la Figure 4-14-d la présence de certains blocs de taille réduite qui ont été classifiés comme *inconnus* (petites régions grises dans la Figure 4-13-d : une à gauche et une autre à droite de la région circulaire, et la Figure 4-14-d : deux à gauche de la région circulaire). Ces imprécisions sont dues au fait qu'une transition très lente sur la bordure (due à la valeur élevée de l'écart-type) combinée aux bandes inconnues larges entre les faisceaux de lecture (due à la valeur élevée du pas angulaire) crée des régions de niveau de gris intermédiaire uniformes et de taille $[S_{min} \times S_{min}]$ entre les segments *occupés* et *libres*. Étant donné que la *PMO* de ces régions se situe dans l'intervalle $[0.498;0.502]$, elles sont classifiées comme étant *inconnues*. Une telle classification fait ressortir le manque de mesures significatives dans ces régions et peut servir d'indicateur pour la détermination de points de vue supplémentaires requis afin de raffiner davantage le modèle de l'environnement et sa segmentation de manière automatisée.

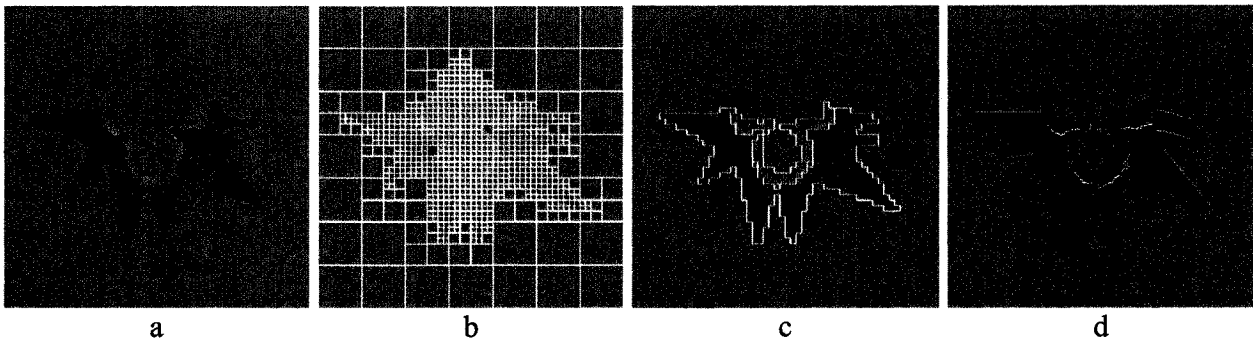


Figure 4-13 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-c avec un écart-type sur la mesure de la distance de 7 cm et un pas angulaire de 1.5 degré, (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

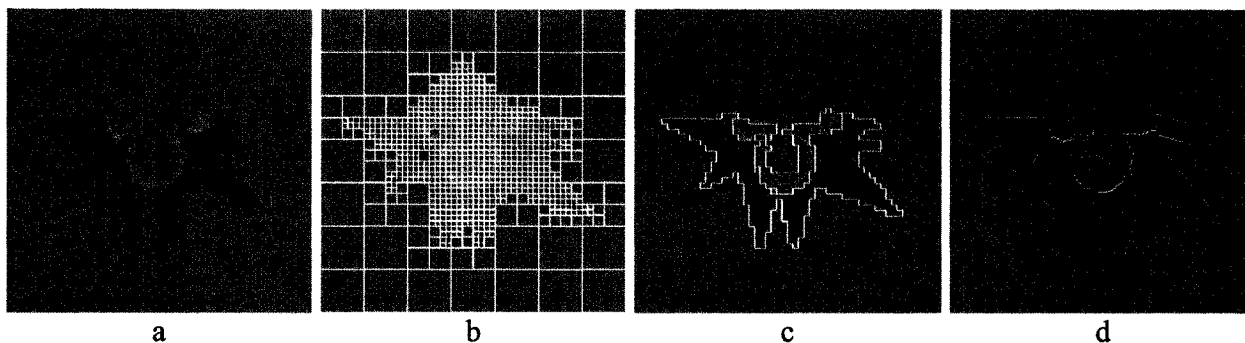


Figure 4-14 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-c avec un écart-type sur la mesure de la distance de 9 cm et un pas angulaire de 1.5 degré. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

Les résultats exposés dans cette section démontrent que pour une multitude de configurations de capteurs utilisées, le schéma proposé résout le problème de l'alternance de faisceaux de lecture et de bandes *inconnues* dans les régions balayées.

4.5 Effets de la variation de l'exploration de l'espace

Le but de cette section est d'analyser le comportement de l'algorithme proposé en fonction de la variation des dimensions et de l'exploration de l'environnement. Le Tableau 4-5 présente les données issues de l'application de notre algorithme, implanté à l'aide du langage de programmation C et de la librairie OpenCV qui lui est compatible, sur trois simulations d'environnements réels. Chacune d'entre elles a servi à la conception de douze images probabilistes qui ont été par la suite segmentées à l'aide de l'algorithme. L'analyse du temps d'exécution se base sur les deux paramètres extrinsèques à notre algorithme, soient le nombre de points de vus qui définit premièrement la complexité des textures et deuxièmement l'exploration de l'espace, ainsi que les dimensions de l'environnement. Il faut noter que l'espace additionnel qui a été ajouté afin de varier les dimensions de l'environnement est uniforme et *inconnu*. La variation du nombre de points de vue dans l'environnement permet aussi d'analyser les effets d'une lecture supplémentaire sur les résultats fournis par l'algorithme proposé.

Espace réel traité	Nombre de points de vue	Dimensions	Erreur type 1 (%)		Erreur type 2 (%)		Temps d'exécution (secondes)			
			Phase 2	Phase 3	Phase 2	Phase 3	Phase 1	Phase 2	Phase 3	Total
<i>Prise électrique</i> (Figure 4-15-a)	2	[320 x 320]	5.24	3.62	0.00	0.00	1.67	0.01	0.08	1.76
		[384 x 384]	5.24	3.62	0.00	0.00	2.12	0.00	0.12	2.24
		[448 x 448]	5.24	3.62	0.00	0.00	2.62	0.01	0.16	2.80
	3	[320 x 320]	4.47	3.01	0.00	0.00	1.76	0.00	0.08	1.84
		[384 x 384]	4.47	3.01	0.00	0.00	2.20	0.00	0.12	2.33
		[448 x 448]	4.47	3.01	0.00	0.00	2.73	0.01	0.16	2.90
	5	[320 x 320]	6.70	5.26	0.00	0.00	1.87	0.01	0.08	1.96
		[384 x 384]	6.70	5.26	0.00	0.00	2.57	0.01	0.11	2.69
		[448 x 448]	6.70	5.26	0.00	0.00	2.95	0.01	0.16	3.12
8	[320 x 320]	9.42	5.21	0.00	0.00	2.01	0.02	0.09	2.12	
	[384 x 384]	9.42	5.21	0.00	0.00	3.14	0.01	0.11	3.27	
	[448 x 448]	9.42	5.21	0.00	0.00	3.25	0.02	0.16	3.42	
<i>Ovale</i> (Figure 4-15-b)	2	[320 x 320]	7.34	6.89	0.00	0.00	2.28	0.02	0.14	2.44
		[384 x 384]	7.34	6.89	0.00	0.00	2.93	0.02	0.20	3.15
		[448 x 448]	7.34	6.89	0.00	0.00	4.93	0.06	0.28	5.27
	3	[320 x 320]	8.14	7.75	0.13	0.00	2.40	0.02	0.08	2.49
		[384 x 384]	8.14	7.75	0.13	0.00	3.07	0.01	0.11	3.19
		[448 x 448]	8.14	7.75	0.13	0.00	5.67	0.01	0.16	5.84
	5	[320 x 320]	6.68	6.14	0.18	0.00	3.06	0.01	0.08	3.15
		[384 x 384]	6.68	6.14	0.18	0.00	3.39	0.02	0.12	3.53
		[448 x 448]	6.68	6.14	0.18	0.00	5.93	0.01	0.17	6.12
8	[320 x 320]	6.61	5.70	0.15	0.00	3.67	0.02	0.08	3.76	
	[384 x 384]	6.61	5.70	0.15	0.00	3.70	0.01	0.12	3.84	
	[448 x 448]	6.61	5.70	0.15	0.00	7.29	0.03	0.16	7.48	
<i>Lunettes</i> (Figure 4-15-c)	2	[320 x 320]	3.84	3.83	1.08	0.03	2.06	0.03	0.08	2.17
		[384 x 384]	3.84	3.83	1.08	0.03	3.17	0.01	0.11	3.30
		[448 x 448]	3.84	3.83	1.08	0.03	3.29	0.01	0.16	3.46
	3	[320 x 320]	2.73	2.64	0.83	0.04	2.45	0.02	0.08	2.54
		[384 x 384]	2.73	2.64	0.83	0.04	3.92	0.01	0.11	4.04
		[448 x 448]	2.73	2.64	0.83	0.04	4.50	0.03	0.16	4.69
	5	[320 x 320]	2.93	2.61	0.35	0.05	5.06	0.01	0.08	5.15
		[384 x 384]	2.93	2.61	0.35	0.05	6.15	0.01	0.12	6.29
		[448 x 448]	2.93	2.61	0.35	0.05	9.08	0.02	0.17	9.26
8	[320 x 320]	2.51	2.37	0.19	0.00	5.70	0.03	0.08	5.81	
	[384 x 384]	2.51	2.37	1.19	0.00	7.45	0.01	0.12	7.59	
	[448 x 448]	2.51	2.37	0.19	0.00	8.18	0.03	0.16	8.37	

Tableau 4-5 : variation du temps d'exécution et des erreurs de type 1 et de type 2 en fonction du nombre de points de vue et de la taille de l'environnement

En premier lieu, le Tableau 4-5 illustre le fait que pour des objets, une configuration de capteurs et un nombre de points de vue donnés, les erreurs de type 1 et de type 2 demeurent

inchangées quelque soient les dimensions de l'environnement probabiliste traité. Ceci est une preuve supplémentaire de la stabilité de l'approche proposée. Le temps d'exécution est quant à lui affecté par deux critères, soient la taille de l'image segmentée et le nombre de points de vue qu'elle incorpore. Ainsi, une augmentation moyenne de la taille de l'environnement de 18.33% entraîne une augmentation moyenne du temps d'exécution de 26.9, 48.7 et 30.25% respectivement pour les images probabilistes issues des environnements simulés *Prise électrique* (Figure 4-15-a), *Ovale* (Figure 4-15-b) et *Lunettes* (Figure 4-15-c). Comme l'illustre le Tableau 4-5, les deux phases qui sont affectées par la taille de l'environnement sont la division hiérarchique et le raffinement, étant donné que leurs opérations nécessitent entre autre de parcourir l'image considérée afin d'analyser son contenu. De même, le temps d'exécution augmente avec le nombre de points de vue. Cette charge de calcul supplémentaire est assumée principalement par la première phase de l'algorithme qui subdivise de l'image probabiliste en régions dotées d'une texture quasi-uniforme. Ainsi, l'existence d'une ou plusieurs lectures supplémentaires accroît le nombre de subdivisions et par conséquent le temps d'exécution. En se basant sur les données du Tableau 4-5, en moyenne la segmentation d'un environnement incluant deux, trois, cinq et huit points de vue requiert respectivement 2.95, 3.31, 4.58 et 5.07 secondes, ce qui implique une augmentation moyenne de 0.38 seconde $[= (3.31-2.95) + (4.58-3.31)/2 + (5.07-4.58)/3] / 3$ par point de vue supplémentaire.

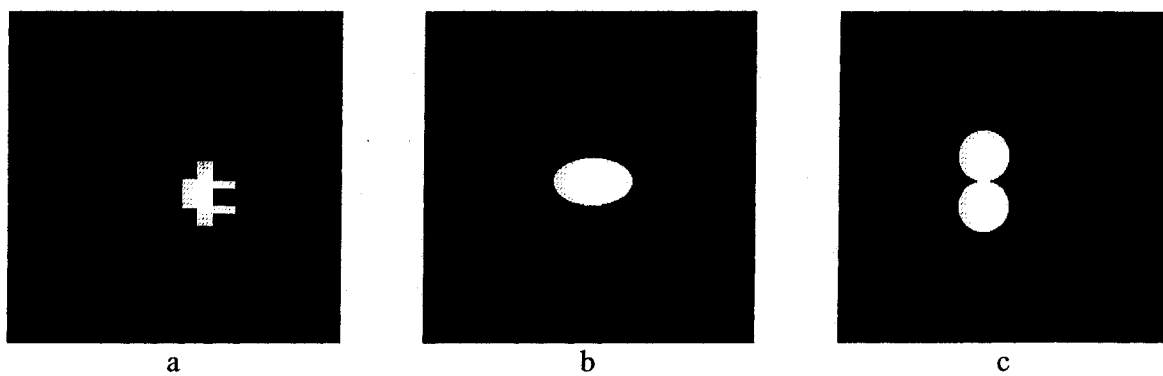
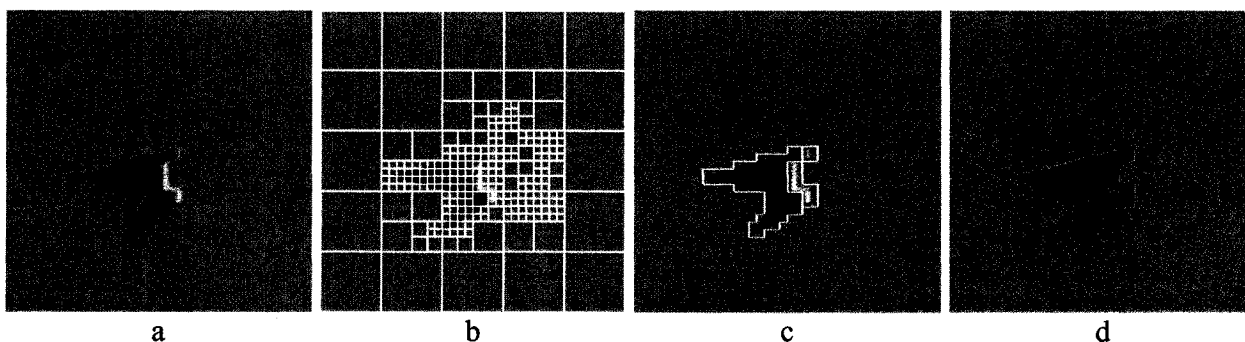


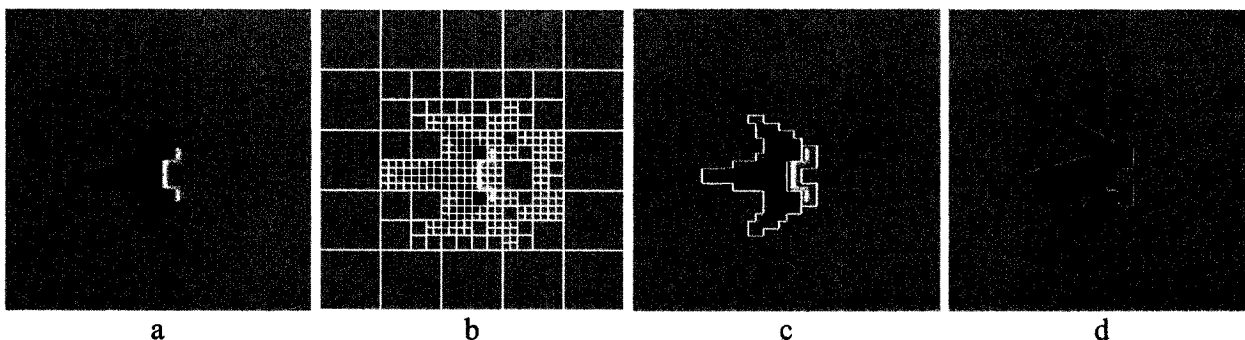
Figure 4-15: environnements simulés utilisés dans la génération des grilles d'occupations et des résultats du Tableau 4-5 et du Tableau 4-6. (a) *Prise électrique*, (b) *Ovale*, (c) *Lunettes*

L'analyse de la variation des mesures d'erreur en fonction du nombre de points de vue est loin d'être évidente. En effet, l'ajout d'une ou plusieurs lectures peut entraîner soit une hausse soit une baisse des erreurs de type 1 et de type 2. La variation des erreurs dépend directement du fait que les lectures supplémentaires touchent totalement, partiellement ou nullement une région qui a été balayée par une lecture précédente.

La Figure 4-16 et la Figure 4-17 illustrent respectivement les résultats de la segmentation des images probabilistes avec deux et trois points de vue, issues de l'environnement simulé *Prise électrique* (Figure 4-15-a). C'est grâce à la lecture additionnelle qui touche la partie supérieure à gauche de l'objet que l'erreur de type 1 est réduite de 0.601% (= 3.62 – 3.019, Tableau 4-5), alors que l'erreur de type 2 demeure nulle lorsqu'une troisième prise de mesure est ajoutée. L'absence de variation de l'erreur de type 2 implique que dans le cas de deux (Figure 4-16-d) et de trois points de vue (Figure 4-17-d), le segment occupé (en bleu) englobe entièrement la section de l'objet réel qu'il représente. La réduction de l'erreur de type 1 est due au fait que le point de vue additionnel fournit des informations supplémentaires quant à la partie qui a été balayée partiellement par les lectures précédentes, ce qui a pour effet de réduire le nombre de cellules qui n'ont pas été classées selon leur occupation réelle.



a b c d
Figure 4-16 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-a et incorporant deux points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)



a b c d
Figure 4-17 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-a et incorporant trois points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

Les résultats de la segmentation des images probabilistes incorporant trois et cinq points de vue et issues de l'environnement simulé *Ovale* (Figure 4-15-b) sont présentés respectivement dans la Figure 4-18 et la Figure 4-19. Les deux lectures additionnelles qui touchent les parties déjà balayées du bas à gauche et du haut à gauche (Figure 4-19-d) de l'objet éliminent les débordements du segment *occupé* dans ces régions (Figure 4-18-d). Ceci a pour effet de mieux adapter la forme de ce segment à celle de l'objet présent dans l'environnement simulé (Figure 4-15-b). La diminution du nombre de cellules libres mal classifiées réduit l'erreur de type 1 de 1.609% (= 7.75 – 6.141, Tableau 4-5). Tout comme dans le cas exposé dans la Figure 4-16 et la Figure 4-17, l'erreur de type 2 demeure nulle étant donné que le segment *occupé* englobe la partie de l'objet réel qu'il représente.

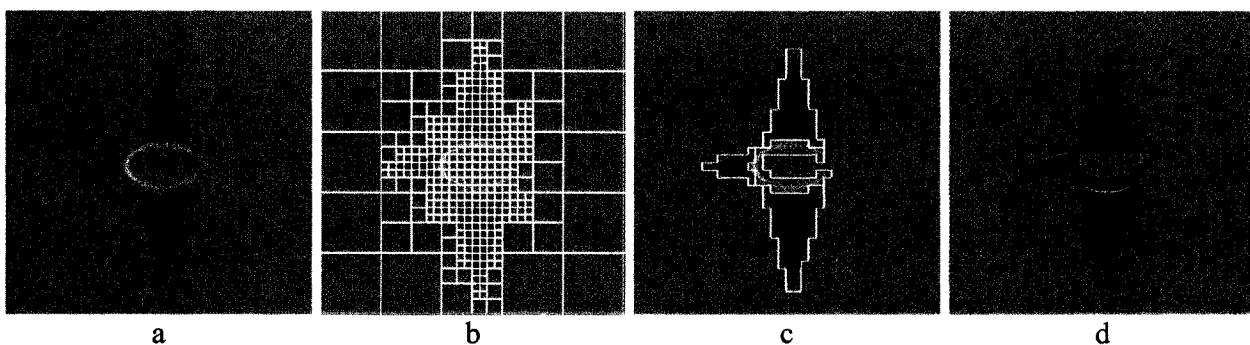


Figure 4-18 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-b et incorporant trois points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

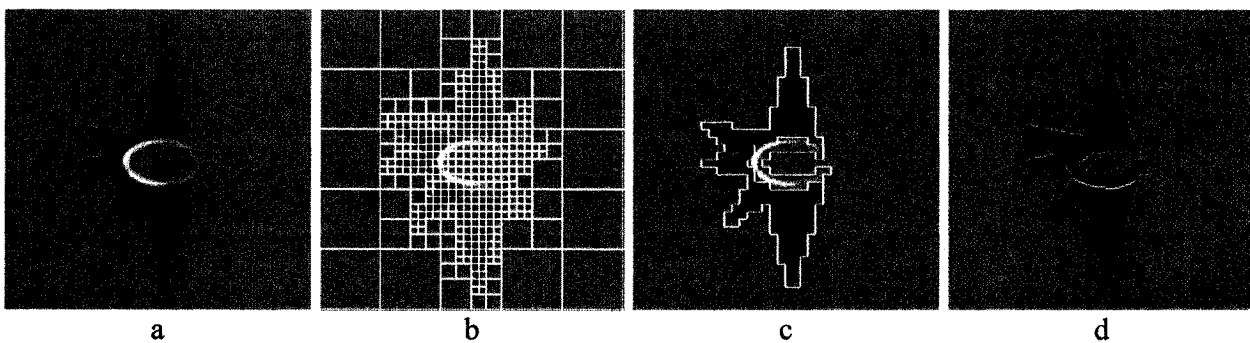


Figure 4-19 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-b et incorporant cinq points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

Le résultat de la segmentation de l'image probabiliste avec huit lectures issue de l'environnement simulé *Prise électrique* (Figure 4-15-a) est exposé dans la Figure 4-20. Ce cas constitue un contre exemple à ceux présentés précédemment du fait que l'ajout de cinq points de vue supplémentaire par rapport à l'environnement probabiliste de la Figure 4-17-a entraîne une

augmentation de 2.191% (= 5.21 – 3.019, Tableau 4-5) de l'erreur de type 1. La variation positive de la mesure d'erreur est due au fait que l'information que génèrent les lectures additionnelles est relative à des régions qui n'ont pas été explorées précédemment et n'étaient donc pas prises en compte dans le calcul de l'erreur, et dont la forme relativement complexe limite l'accès des faisceaux de lecture (partie à droite du segment occupé, Figure 4-20). Aucune variation n'a été observée quand à la valeur de l'erreur de type 2 qui demeure nulle.

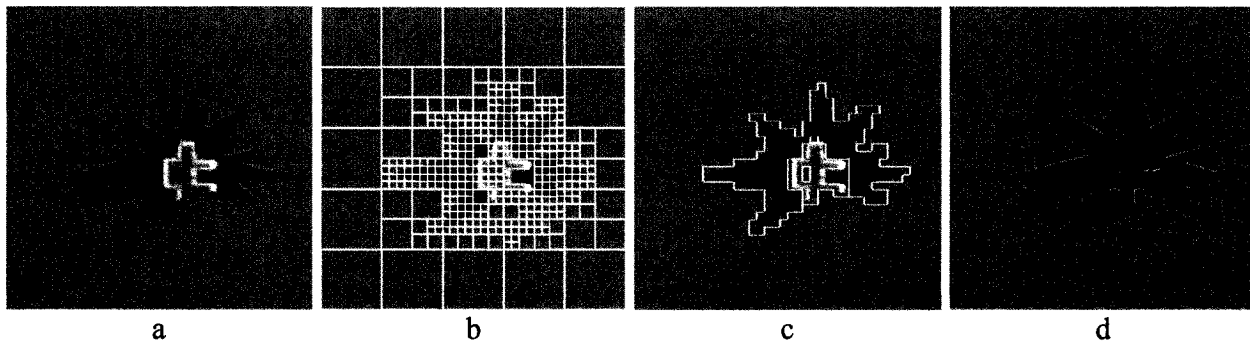
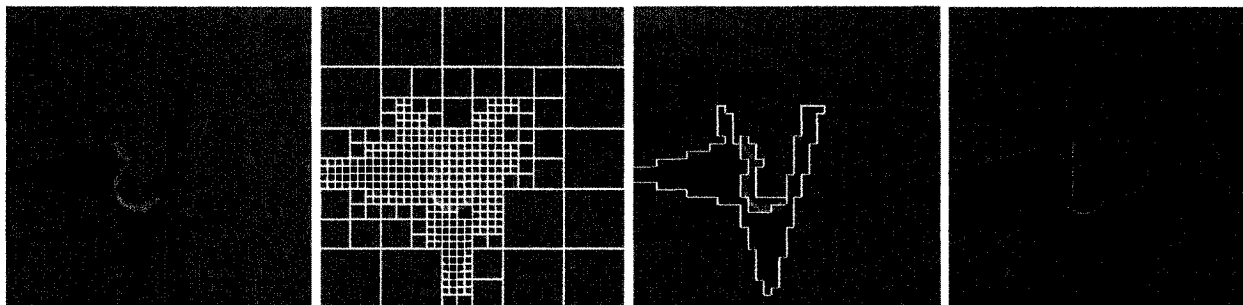


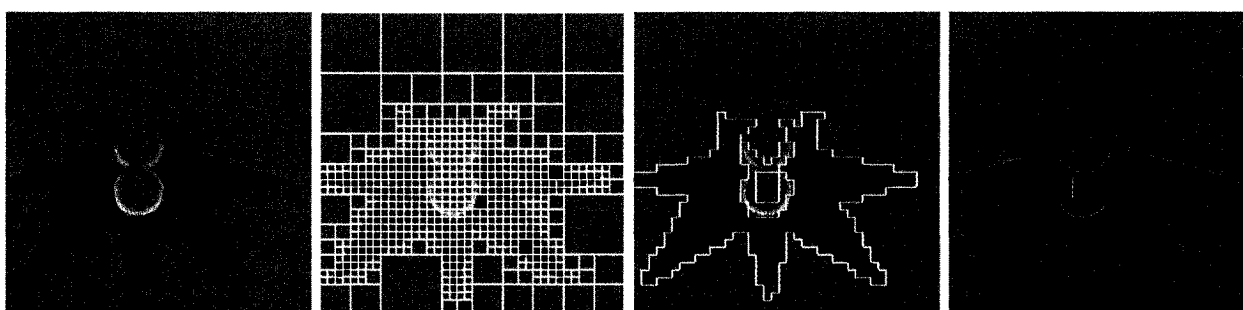
Figure 4-20 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-a et incorporant huit points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

Les résultats de la segmentation des images probabilistes incorporant deux, cinq et huit lectures et issues de l'environnement simulé *Lunettes* (Figure 4-15-c) sont présentés respectivement dans la Figure 4-21, la Figure 4-22 et la Figure 4-23. Une évaluation visuelle de ces résultats prouve, tout comme dans le cas précédent que la forme des segments obtenus correspond à celle des régions de l'environnement probabiliste qu'ils représentent. Cette observation est soutenue par les résultats quantitatifs obtenus pour par les valeurs des erreurs de type 1 et de type 2.

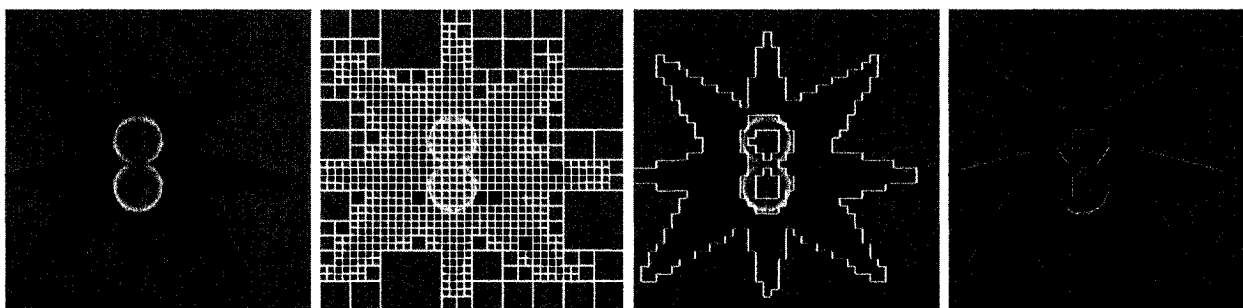
Ainsi, les moyennes des erreurs de type 1 et de type 2 pour l'ensemble des simulations exposées dans le Tableau 4-5 sont respectivement de 4.58% et de 0.01%, tandis que les moyennes des deux types d'erreur relatives aux quatre segmentations (avec deux, trois, cinq et huit lectures) des environnements probabilistes issus de *Lunettes* (Figure 4-15-c) sont respectivement de 2.86% et 0.03% (Tableau 4-5). Ces valeurs très faibles dans le cas de l'erreur de type 1 et négligeables dans le cas de l'erreur de type 2 démontrent l'efficacité de l'approche proposée qui fournit des résultats consistants quelque soient la taille et le nombre de lectures qui caractérisent l'environnement probabiliste traité.



a b c d
 Figure 4-21 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-c et incorporant deux points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)



a b c d
 Figure 4-22 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-c et incorporant cinq points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)



a b c d
 Figure 4-23 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-c et incorporant huit points de vue. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

4.6 Effets de la variation des paramètres b et S_{min}

Dans les sections 4.4 et 4.5, nous avons analysé l'effet de plusieurs paramètres extrinsèques sur les résultats de notre algorithme. La présente section traite les paramètres intrinsèques à l'algorithme, soient le nombre de niveaux de discrétisation b du contraste et la taille minimale S_{min} qu'une subdivision peut atteindre. Le premier paramètre définit une des deux dimensions des histogrammes de distribution LBP/C, tandis que le second influence le nombre de subdivisions opérées durant la première phase de l'algorithme. Le Tableau 4-6 illustre les résultats des simulations que nous avons effectuées à l'aide de l'implantation de notre algorithme avec le langage de programmation C et avec la librairie OpenCV. Les images probabilistes utilisées ont une taille de [448 x 448] pixels. Lors de la segmentation de chacune d'entre elles, nous avons varié les paramètres b et S_{min} de 8 à 16. Afin de varier les conditions de simulation et mieux démontrer la validité de nos résultats, les images probabilistes issues des espaces réels traités (Tableau 4-6) sont dotées chacune de paramètres extrinsèques différents. En effet, les capteurs utilisés dans les trois premières images probabilistes, issues de *Surface en U* (Figure 4-5-a), *Carré et disque* (Figure 4-5-b) et de *Forme complexe* (Figure 4-5-c), ont respectivement un écart-type sur la distance de 3, 5 et 7 cm et un pas angulaire de 1, 1.5 et 0.5 degré. Les cinq images suivantes issues de *Prise électrique* (Figure 4-15-a), *Ovale* (Figure 4-15-b), *Lunettes* (Figure 4-15-c), *Tasse* (Figure 4-24-a) et *Lentille divergente* (Figure 4-24-b) incorporent respectivement 8, 3, 5, 12 et 4 points de vue avec un écart-type de 4 cm et un pas angulaire de 0.5 degré.

Espace réel traité	b	S_{min}	Erreur type 1 (%)		Erreur type 2 (%)		Temps d'exécution (secondes)			
			Phase 2	Phase 3	Phase 2	Phase 3	Phase 1	Phase 2	Phase 3	Total
<i>Surface en U</i> (Figure 4-5-a)	8	8	1.71	1.98	0.41	0.06	9.34	0.03	0.16	9.53
		16	1.34	1.51	1.51	1.29	3.29	0.01	0.16	3.46
	16	8	1.19	1.90	2.88	0.35	11.48	0.05	0.16	11.68
		16	1.34	1.51	3.82	1.40	6.57	0.02	0.16	6.74
<i>Carré et disque</i> (Figure 4-5-b)	8	8	5.46	4.80	0.55	0.12	6.32	0.01	0.30	6.63
		16	4.68	4.82	5.28	0.63	2.65	0.03	0.28	2.96
	16	8	4.94	4.76	1.07	0.14	8.59	0.06	0.28	8.93
		16	4.68	4.82	5.28	0.63	5.76	0.01	0.28	6.06
<i>Forme complexe</i> (Figure 4-5-c)	8	8	5.11	4.48	0.72	0.00	7.59	0.03	0.28	7.90
		16	6.20	5.88	1.26	0.00	2.89	0.01	0.28	3.19
	16	8	5.11	5.48	0.71	0.00	8.96	0.05	0.30	9.30
		16	6.20	5.87	1.26	0.00	6.00	0.00	0.30	6.30
<i>Prise électrique</i> (Figure 4-15-a)	8	8	9.42	5.21	0.00	0.00	3.25	0.02	0.16	3.42
		16	9.55	6.16	0.61	0.00	2.37	0.02	0.16	2.55
	16	8	9.06	5.19	0.61	0.00	6.18	0.01	0.16	6.35
		16	9.54	6.16	0.61	0.00	4.98	0.02	0.16	5.15
<i>Ovale</i> (Figure 4-15-b)	8	8	8.15	7.76	0.13	0.00	7.29	0.03	0.16	7.48
		16	8.41	9.81	1.17	0.05	3.00	0.00	0.17	3.17
	16	8	7.77	7.76	0.35	0.00	8.75	0.01	0.16	8.92
		16	8.41	9.81	1.16	0.05	5.54	0.33	0.17	6.04
<i>Lunettes</i> (Figure 4-15-c)	8	8	2.93	2.61	0.35	0.04	8.18	0.03	0.16	8.37
		16	2.79	2.69	2.10	0.12	2.82	0.03	0.16	3.01
	16	8	2.84	2.61	0.27	0.04	9.35	0.03	0.16	9.54
		16	2.79	2.69	2.10	0.12	5.84	0.01	0.16	6.01
<i>Tasse</i> (Figure 4-24-a)	8	8	4.23	3.62	0.02	0.00	7.82	0.03	0.17	8.02
		16	5.17	3.85	0.36	0.01	2.90	0.02	0.16	3.07
	16	8	4.23	3.62	0.02	0.00	10.12	0.03	0.17	10.33
		16	5.17	3.85	0.36	0.01	6.08	0.02	0.16	6.25
<i>Lentille divergente</i> (Figure 4-24-b)	8	8	7.21	6.99	0.00	0.00	6.46	0.01	0.16	6.63
		16	8.09	7.17	0.20	0.04	2.70	0.00	0.17	2.87
	16	8	7.21	6.99	0.00	0.00	7.40	0.03	0.16	7.59
		16	8.09	7.17	0.20	0.04	5.45	0.02	0.16	5.62

Tableau 4-6 : variation du temps d'exécution et des erreurs de type 1 et de type 2 en fonction du nombre de niveaux de discrétisation du contraste (b) et de la taille minimale d'une subdivision (S_{min})

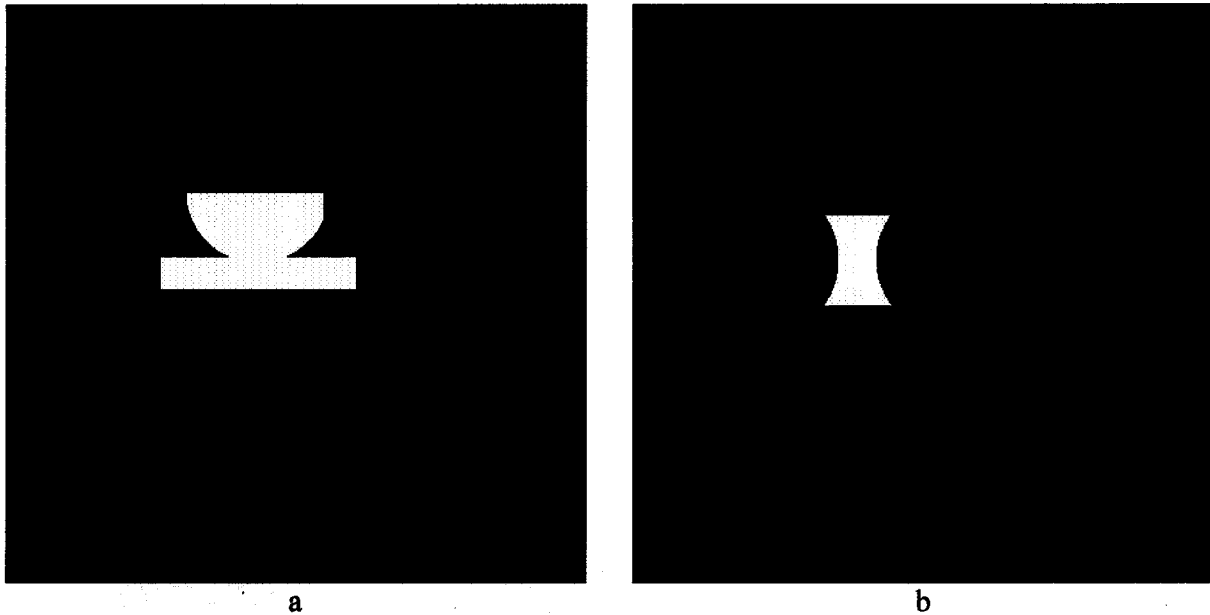


Figure 4-24 : environnements simulés de taille [448 x 448] utilisés dans la génération des grilles d'occupation et des résultats du Tableau 4-6. (a) *Tasse*, (b) *Lentille divergente*

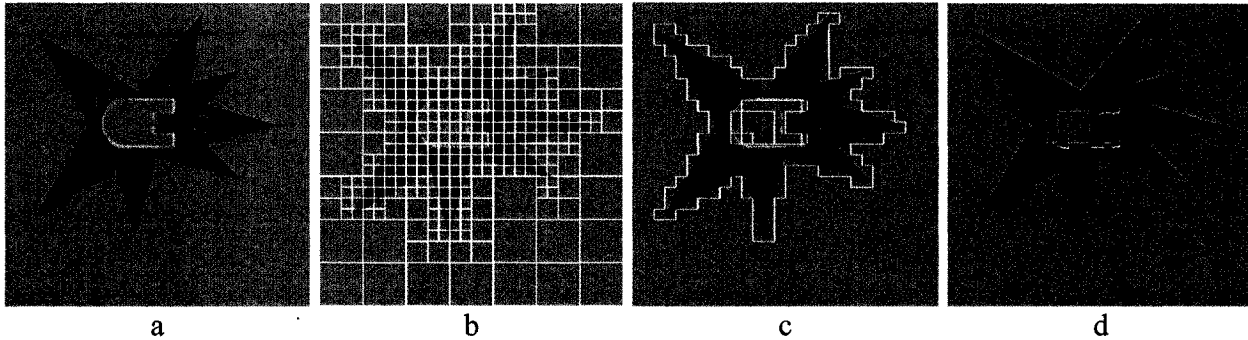
En premier lieu, le nombre de niveaux de discrétisation b du contraste n'affecte que légèrement les erreurs de type 1 et de type 2. Ainsi, seules les mesures d'erreur relatives aux résultats de la segmentation des quatre premières images du Tableau 4-6 diffèrent légèrement entre le choix d'une valeur de b égale à 8 ou à 16. En effet, dans le cas de ces simulations et pour une valeur de S_{min} égale à 8, les erreurs de type 1 et de type 2 augmentent en moyenne de 0.215% et de 0.07% entre le choix du paramètre b égal à 8 et à 16. La variation est encore plus faible dans le cas d'une valeur de S_{min} égale à 16 où la moyenne de la variation des erreurs de type 1 et de type 2 est respectivement de 0.0025% et de 0.0028%. L'effet du niveau de discrétisation du contraste est plus remarquable sur le temps d'exécution qui, pour les valeurs de S_{min} égale à 8 et à 16, augmente en moyenne respectivement de 29.65 et de 98.57% entre un choix de b égal à 8 et à 16. Cette variation qui touche principalement la première phase de notre algorithme est due au fait que le b définit une des deux dimensions des histogrammes de distribution LBP/C. Plus cette valeur est grande, plus importants sont la taille et les calculs qui sont nécessaires à la comparaison des distributions des caractéristiques des textures à l'aide du G-statistique (équation (2.5)). Le choix de 8 niveaux de discrétisation du contraste fournit de meilleurs résultats quantitatifs et qualitatifs par rapport à une valeur de 16, ce qui justifie le choix de $b = 8$ dans la conception de notre algorithme.

En second lieu, pour la valeur choisie de b égale à 8, le nombre de subdivisions (défini par S_{min}) opérées affecte qualitativement et quantitativement les résultats obtenus. En terme de mesures

d'erreur, et pour l'ensemble des segmentations opérées (Tableau 4-6), les moyennes des erreurs de type 1 et de type 2 se situent respectivement à 4.68% et à 0.03% pour une valeur de S_{min} égale à 8, et à 5.23% et à 0.27% pour une valeur de S_{min} de 16. Ainsi, la subdivision supplémentaire permet d'obtenir une meilleure convergence entre les segments et les régions correspondantes de l'environnement simulé. L'amélioration est particulièrement notable sur l'erreur de type 2. Par contre, la réduction de l'erreur de la segmentation se fait au prix du temps d'exécution. En effet, pour la valeur choisie de b égale à 8, la subdivision supplémentaire coûte en moyenne¹⁶ 4.21 secondes de calculs. Ce nombre est important sachant que les temps d'exécution moyens pour S_{min} égale à 16 et à 8 se situent respectivement à 3.03 et à 7.25 secondes (augmentation de 238%).

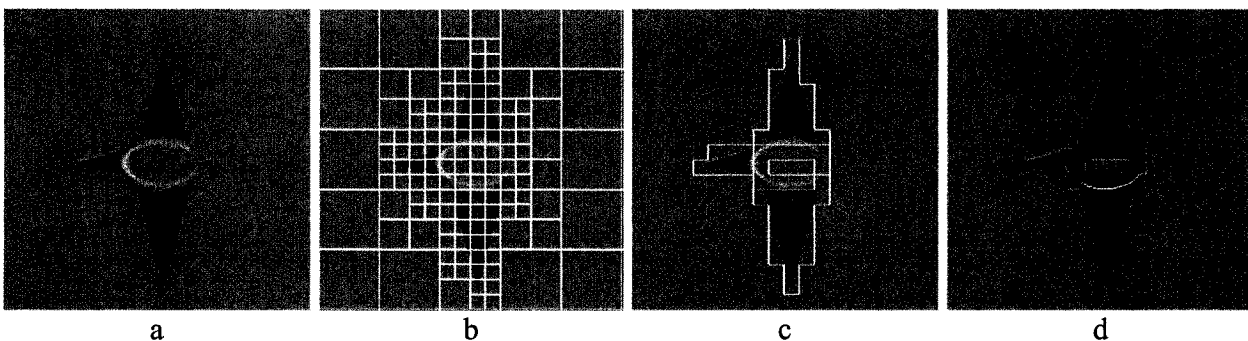
Chacun des deux critères d'évaluation cités ci-dessus (mesures d'erreur et temps de calcul) penche en faveur d'une valeur de S_{min} différente. Afin d'identifier le nombre convenable de subdivisions, nous avons inclus les résultats de la segmentation d'un sous-ensemble des images probabilistes dont les particularités sont exposées dans le Tableau 4-6. La Figure 4-6, la Figure 4-18, la Figure 4-27 et la Figure 4-29 illustrent respectivement les résultats de la segmentation des images probabilistes issues de *Surface en U* (Figure 4-5-a), *Ovale* (Figure 4-15-b), *Tasse* (Figure 4-24-a) et *Lentille divergente* (Figure 4-24-b) avec trois niveaux de subdivisions, c'est-à-dire lorsque $S_{min} = 8$. Tandis que la Figure 4-25, la Figure 4-26, la Figure 4-28 et la Figure 4-30 correspondent aux résultats de la segmentation des mêmes environnements probabilistes mais avec une valeur de S_{min} égale à 16. La comparaison de chaque paire de résultats permet d'illustrer une des limitations dont souffre la segmentation à deux niveaux de subdivisions ($S_{min} = 16$). Dans le cas des résultats de la Figure 4-6-d et la Figure 4-25-d, la subdivision à deux niveaux n'a pas intercepté l'arête inférieure de l'objet, tandis que celle à trois niveaux ($S_{min} = 8$) la représente convenablement. Ceci illustre que la subdivision supplémentaire fournit un meilleur rendement dans le cas où la transition entre les espaces libre et occupé est abrupte, c'est-à-dire quand la valeur de l'écart-type des capteurs faible (dans ce cas égale à 3 cm), ce qui cause une diminution de l'épaisseur de la région occupée.

¹⁶ Moyenne des différences des temps d'exécution entre une valeur de S_{min} égale à 16 et à 8 pour une valeur de b égale à 8 (Tableau 4-6).



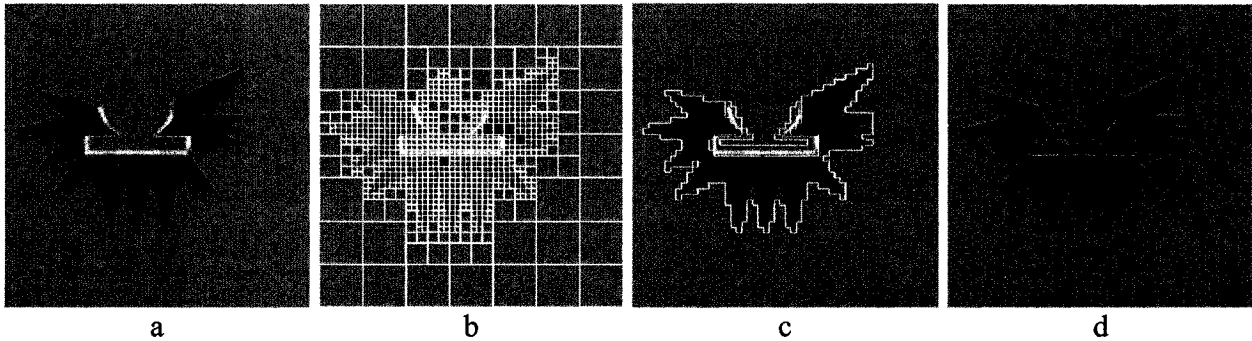
a b c d
Figure 4-25 : segmentation de l'image probabiliste (a) issue de la Figure 4-5-a avec un écart-type sur la mesure de la distance de 3 cm et un pas angulaire de 0.5 degré. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

La paire de résultats illustrés dans la Figure 4-18 et la Figure 4-26 révèle un autre problème. Bien que l'image probabiliste utilisée (Figure 4-18-a et Figure 4-26-a) n'incorpore aucune lecture du côté droit de l'objet en forme d'ovale (Figure 4-15-b), la segmentation avec une valeur de S_{min} égale à 16 identifie faussement cette région inconnue comme faisant partie du segment *occupé* (Figure 4-26-d, partie de droite du segment *occupé* représenté en bleu), ce qui n'est pas le cas lorsque $S_{min} = 8$. La subdivision supplémentaire permet donc de distinguer et de traiter correctement deux caractéristiques importantes des images probabilistes, soient la bande inconnue entre les faisceaux adjacents et une distance relativement faible entre deux régions classées *occupées* (qui peuvent dans certains cas correspondre à deux objets différents dans l'environnement simulé). L'incapacité de la segmentation à deux niveaux ($S_{min} = 16$) de détecter les régions étroites qui séparent deux régions *occupées* peut causer l'échec de nombreuses applications robotiques de planification de trajectoire et de manipulation d'objets.

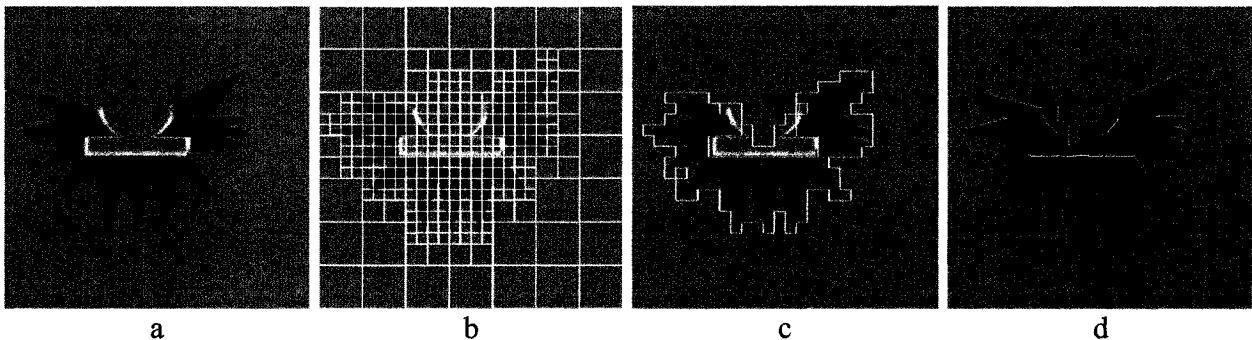


a b c d
Figure 4-26 : segmentation de l'image probabiliste (a) issue de la Figure 4-15-b incorporant trois points de vue. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

Les résultats exposés dans la Figure 4-27 et la Figure 4-28 illustrent la forte dépendance de la segmentation à deux niveaux de subdivision sur la phase de raffinement. En comparant la Figure 4-27-c et la Figure 4-28-c, nous pouvons constater que pour une valeur de S_{min} égale à 16, les débordements des segments sur les régions adjacentes d'occupation différente sont élevés après la phase de création des segments. Cet effet est nettement atténué lorsque $S_{min} = 8$.



a b c d
Figure 4-27 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-a. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments inconnu sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)



a b c d
Figure 4-28 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-a. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments inconnu sont représentés par des couleurs différentes : (gris), libre (noir) et occupé (bleu)

La seconde phase de l'algorithme proposé fusionne les subdivisions adjacentes dotées de $PMOs$ dans le même intervalle. Dans certains cas rares, une subdivision de taille $[S_{min} \times S_{min}]$ peut chevaucher sur deux régions d'occupation connue (*libre* et *occupée*) et se caractériser par une PMO bornée dans l'intervalle $[0.498;0.502]$. Étant donné que cet intervalle correspond à celui d'un segment *inconnu*, l'algorithme s'avère incapable de le classifier convenablement. La Figure 4-30-d illustre ce cas particulier. En effet, nous remarquons l'existence d'une région de taille $[S_{min} \times S_{min}]$ *inconnue* sur le côté gauche de l'objet en forme de lentille divergente. Dans ce cas, la subdivision à trois niveaux

surmonte cette mauvaise classification (Figure 4-29-d), mais il demeure possible que cette situation rare puisse arriver même dans le cas où S_{min} est égal à 8. Ce problème provient non seulement de la conception de l'algorithme mais aussi d'une exploration insuffisante de l'environnement. Ainsi, l'ajout d'une lecture supplémentaire représente la solution la plus fiable à ce problème (Figure 4-31-d). Le résultat intermédiaire d'une segmentation peut ainsi être exploité pour guider la sélection des points de vue optimaux afin de raffiner le modèle d'occupation de l'espace. Néanmoins, une subdivision à trois niveaux demeure avantageuse étant donné qu'il est toujours préférable de réduire le plus possible la taille $[S_{min} \times S_{min}]$ des régions touchées par cette incohérence. Il faut noter que dans la Figure 4-30-d et la Figure 4-31-d, le choix de deux niveaux de subdivision est incapable d'identifier la région inconnue au centre de la lentille divergente. La précision additionnelle que fournit une segmentation avec S_{min} égal à 8 permet de détecter les détails qui caractérisent les objets (la base de la tasse dans la Figure 4-27-d et le centre de la lentille divergente dans la Figure 4-29-d).

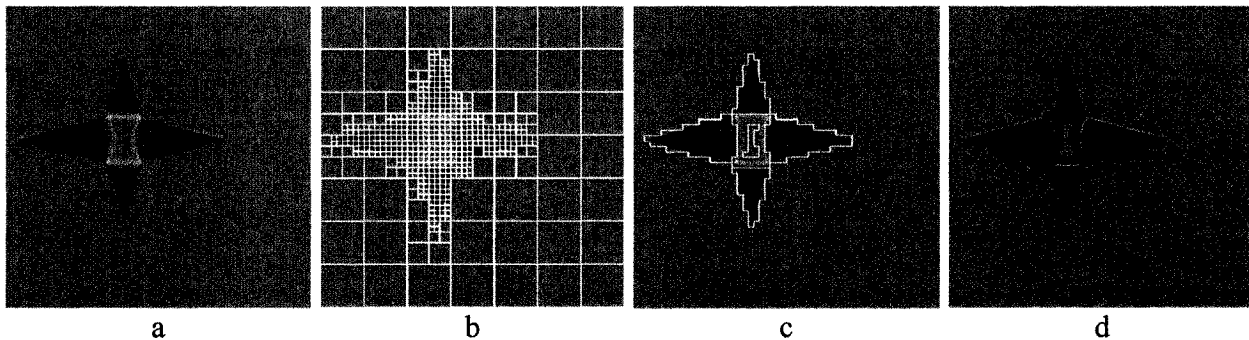


Figure 4-29 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-b. (b) division hiérarchique avec $S_{min} = 8$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu)

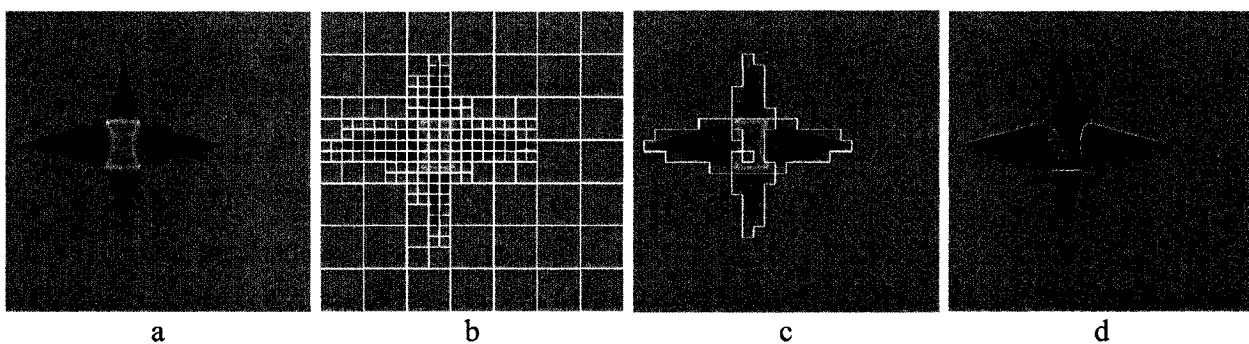
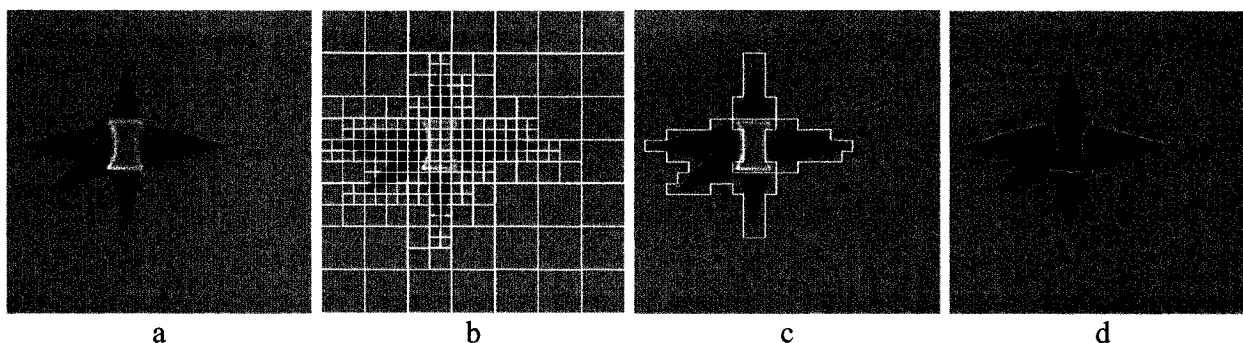


Figure 4-30 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-b. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments inconnu sont représentés par des couleurs différentes : (gris), libre (noir) et occupé (bleu)



a b c d
Figure 4-31 : segmentation de l'image probabiliste (a) issue de la Figure 4-24-b. (b) division hiérarchique avec $S_{min} = 16$, (c) création des segments, (d) raffinement où les segments sont représentés par des couleurs différentes : inconnu (gris), libre (noir) et occupé (bleu). La lecture supplémentaire résout le problème de l'incohérence dans la classification de certaines régions sur la bordure de deux régions d'occupation connue

Bien que la subdivision supplémentaire entraîne une charge de calcul additionnelle, elle permet d'améliorer considérablement les résultats de la segmentation. Ainsi, notre désir de concevoir un algorithme de segmentation stable et fiable constitue la principale raison d'opter pour une valeur de 8 pour S_{min} et pour b .

4.7 Conclusion

Ce chapitre a fourni une comparaison entre le schéma de segmentation proposé et celui d'Ojala et Pietikäinen [39]. Par la suite les résultats ainsi que les caractéristiques de la nouvelle approche ont été présentés et analysés. Ceci a conduit à la sélection des paramètres intrinsèques optimaux sur le niveau de la résolution des histogrammes de distribution LBP/C et sur le niveau de subdivision hiérarchique maximal.

Les résultats obtenus sont convaincants que ce soit visuellement ou en terme de mesures d'erreurs et de temps d'exécution. En effet, pour les paramètres de segmentation choisis soient S_{min} égal à 8 et b égal à 8, la moyenne des erreurs de type 1 et de type 2 pour l'ensemble des simulations effectuées est respectivement de 5.62% et 0.21%. Ces valeurs relativement faibles démontrent que les segments englobent presque parfaitement les régions qu'ils représentent. D'autre part, la moyenne du temps d'exécution est elle aussi très acceptable compte tenu de la qualité des résultats obtenus. En effet, la segmentation des images probabilistes de taille [448 x 448], à l'aide de l'implantation réalisée avec le langage de programmation C, requiert moins de 10 secondes (Tableau 4-2, Tableau 4-5 et Tableau 4-6), soit 49.8 μsec / pixel (= 10 s / (448 x 448) pixels), avec une

moyenne de 6.431 secondes, soit 32 μ sec / pixel. Les résultats visuels sont excellents compte tenu de la nature des images probabilistes. Il faut rappeler que c'est à partir de l'échantillonnage d'une bordure à l'aide de faisceaux de lecture que l'algorithme est capable d'identifier fidèlement les objets présents dans l'environnement. Ce chapitre démontre que l'algorithme de segmentation proposé peut servir de base stable pour des applications telles : la sélection autonome de points de vue dans la construction de cartes de l'espace, la reconnaissance d'objets par leur forme caractéristique à partir de mesures de surface incomplètes et incertaines, ainsi que la planification de la trajectoire, l'évitement des obstacles et l'interaction d'un robot avec un environnement cartographié à l'aide de capteurs dotés d'un niveau d'incertitude élevé et d'une résolution angulaire limitée.

Chapitre 5 Segmentation probabiliste tridimensionnelle

5.1 Introduction

Dans les chapitres précédents, nous avons traité la segmentation d'espaces probabilistes bidimensionnels. Ce chapitre étend ces concepts au cas des environnements tridimensionnels. Ainsi, nous commençons par exposer la physique de la construction des modèles probabilistes tridimensionnels utilisés. Par la suite, nous détaillons la généralisation de l'algorithme de segmentation au cas tridimensionnel. La dernière section est consacrée à la présentation et à l'analyse des résultats obtenus.

5.2 Construction des environnements probabilistes tridimensionnels

Dans le second chapitre (section 2.2.4), nous avons exposé la physique de la construction des images probabilistes bidimensionnelles à l'aide du capteur de profondeur Jupiter [10] (section 2.2.4.1), et du programme de simulation (section 2.2.4.2). Dans le cas tridimensionnel, outre la calibration du capteur pour balayer avec précision des plans parallèles, il faut combiner les données obtenues et les traiter avec une estimation de l'erreur du capteur suivant la théorie des grilles d'occupation introduite dans les sections 2.2.2 et 2.2.3. Ce travail, bien que théoriquement possible, se situe au-delà de l'étendue de cette thèse. D'autre part, l'extension de l'approche par simulation développée par Bolzon (Annexe A) pour traiter le cas des environnements tridimensionnels soulève certaines difficultés relatives à :

- ❖ La simulation de lectures de capteur dirigées dans toutes les directions possibles et dotées d'une composante en élévation non nulle (*i.e.* : non parallèles au plan des (x, y));
- ❖ La représentation des environnements, soient l'original fourni en entrée et le modèle probabiliste résultant.

En résumé, le développement d'une nouvelle application de simulation capable de générer directement des environnements probabilistes tridimensionnels s'est avéré, lui aussi, hors de la portée de notre travail.

Afin de surmonter ces problèmes et d'évaluer une extension tridimensionnelle de l'algorithme de segmentation proposé, nous avons choisi de construire les modèles tridimensionnels à partir d'une pile d'images bidimensionnelles de même taille. En effet, la largeur, la hauteur et la profondeur de notre environnement correspondent respectivement aux dimensions ainsi qu'au nombre d'images considérées. Ainsi, une représentation tridimensionnelle basée sur une pile d'images confère aux cellules des propriétés volumétriques, en opposition au terme pixel utilisé pour dénoter une unité de surface dans le cas des images planes. La Figure 5-1 ci-dessous représente un environnement tridimensionnel de taille $[M \times N \times 6]$ dans lequel nous retrouvons un objet conique. La base et le sommet circulaires du cône sont situés respectivement dans les images de niveau zéro et cinq. Les images intermédiaires incluent des disques de rayon décroissant qui forment le corps de l'objet. Le système de coordonnées a été choisi de telle sorte que l'origine des environnements tridimensionnels corresponde au coin en arrière à gauche de l'image au niveau zéro, soit le niveau le plus bas.

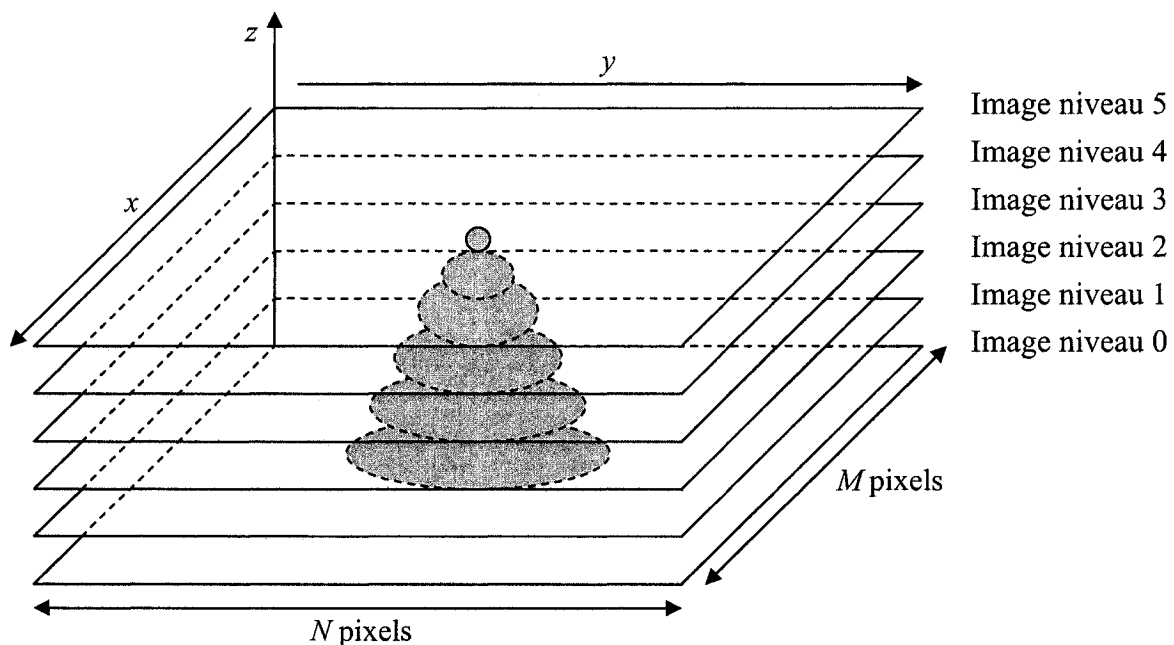


Figure 5-1 : environnement tridimensionnel de taille $[M \times N \times 6]$ obtenu en superposant six images de dimension $[M \times N]$

L'algorithme de segmentation tridimensionnelle prend deux entrées, soient l'environnement probabiliste sur lequel s'opère la segmentation et l'environnement simulé qui sert de référence pour le calcul des mesures d'erreur (équations (4.1) et (4.2)). Les images constituant l'environnement simulé sont générées et indexées manuellement. À titre d'exemple, un objet en forme de cône est

obtenu en superposant des images incorporant des disques de rayon décroissant, centrés sur les pixels de mêmes coordonnées dans le plan (x, y) . Un objet cylindrique est obtenu d'une manière similaire mais avec des disques de rayon constant. Afin de générer l'environnement probabiliste à partir de l'environnement simulé, nous fournissons chacune des images construites manuellement au programme de simulation développé par Bolzon. Les images probabilistes résultantes sont indexées dans le même ordre que les images réelles desquelles elles sont issues. Dans la série d'images probabilistes, la position ainsi que la direction des capteurs desquels émanent les lectures sont les mêmes pour tous les niveaux de la pile. Ainsi, la principale limitation de ces modèles tridimensionnels provient du fait que toutes les lectures sont parallèles au plan (x, y) et sont simplement décalées le long de l'élévation z , et que l'incertitude ne peut pas se propager directement dans la direction de l'élévation. Mais cette restriction n'affecte pas la validité de l'approche proposée étant donné que le schéma de segmentation n'opère aucune discrimination entre les trois dimensions de l'espace qui sont traités d'une manière similaire. Ainsi, tout comme les objets réels, les plans balayés par des lectures créent des espaces définis en élévation. La Figure 5-2 illustre un environnement tridimensionnel qui est obtenu en superposant six images probabilistes. La dimension supplémentaire qui est attribuée aux pixels crée une continuité volumétrique entre les plans considérés. Dans la Figure 5-2, les bordures de l'environnement probabiliste global ont été supprimées afin d'alléger la visualisation.

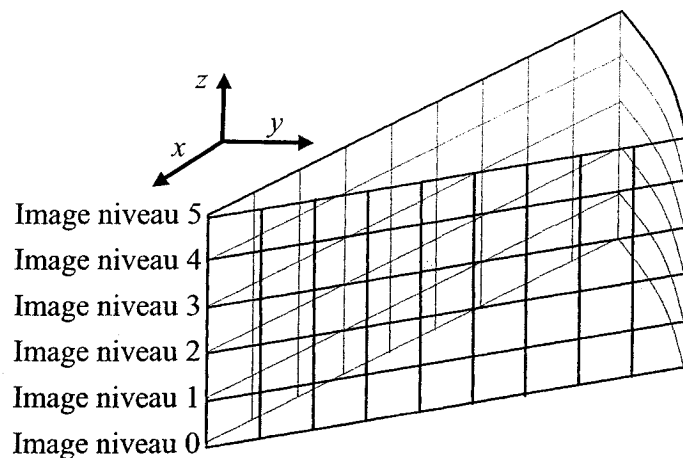


Figure 5-2 : représentation volumétrique de la superposition de plusieurs surfaces balayées par des lectures de capteur

5.3 L'algorithme de segmentation tridimensionnel

Dans le chapitre précédent, nous avons exposé l'algorithme de segmentation LBP/C révisé en deux dimensions. Les résultats convaincants que nous avons obtenus nous ont motivés à étendre les principes de cette approche pour couvrir des environnements probabilistes tridimensionnels. Néanmoins, cette adaptation est loin d'être directe, et la prise en compte de l'élévation impose des contraintes en termes de performance et de faisabilité. En effet, la redéfinition de l'unité de texture est inévitable afin de prendre en compte la dimension additionnelle. De plus, la division hiérarchique ainsi que les étapes subséquentes, autrefois opérées dans un plan doivent être reformulées et adaptées aux régions volumétriques. Dans les paragraphes suivants, les termes segment et région dénotent un volume dans l'espace. Cette partie est divisée en quatre sections. La première redéfinit et adapte le concept de la texture au cas tridimensionnel. Les sections suivantes sont dédiées respectivement aux trois étapes de l'algorithme de segmentation, soient la division hiérarchique, la création des segments et le raffinement.

5.3.1 Caractéristiques de la texture tridimensionnelle

La segmentation basée sur l'analyse de la double distribution *LBP* et *C* dépend principalement du choix de l'unité de texture. Comme introduit par Ojala et Pietikäinen [39] et repris dans le second chapitre (section 2.4.7.1), la plage de valeurs possibles de l'opérateur *LBP* définit une des dimensions de l'histogramme de distribution, la seconde dimension étant fixe du fait qu'elle dépend du nombre de niveaux de discrétisation *b* choisi pour représenter le contraste. Ainsi, l'unité de texture plane de taille $[3 \times 3]$ implique huit cellules avoisinantes pour chaque cellule centrale, soit $2^8 = 256$ valeurs possibles pour le *LBP*. Par conséquent, la taille de l'histogramme de distribution augmente d'un facteur deux pour chaque cellule avoisinante supplémentaire lorsqu'une nouvelle dimension est introduite. Cette explosion combinatoire nous motive à réduire au minimum possible la taille de l'unité de texture choisie.

Notre première idée fut de considérer des unités cubiques de taille $[3 \times 3 \times 3]$, mais cette tentative, illustrée dans la Figure 5-3, était vouée à l'échec du fait que le traitement impliquait 26 cellules avoisinantes (18 au total aux niveaux $N-1$ et $N+1$ et 8 au niveau N) pour chaque cellule centrale. Une telle représentation génère des histogrammes de distribution de dimensions $[2^{26} \times b]$ soit de $[67108864 \times b]$ éléments pour chaque région traitée. La taille excessive de ces histogrammes

et les calculs nécessaires afin de les comparer nous ont poussé à trouver des alternatives aussi stables mais moins exigeantes en terme de calcul.

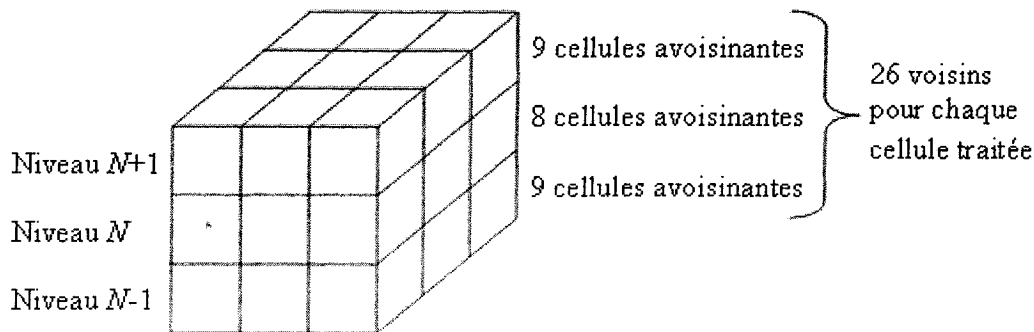


Figure 5-3 : unité de texture cubique de taille $[3 \times 3 \times 3]$

La seconde tentative de redéfinition des unités de textures fut de considérer seulement les cellules avoisinantes qui partagent une surface avec la cellule centrale (Figure 5-4). Le choix de cette unité de texture provient du fait qu'elle est, tout comme celle de la Figure 5-3, symétrique par rapport aux trois plans (x, y) , (x, z) et (y, z) . Cette symétrie fait en sorte que l'algorithme n'opère aucune discrimination entre les trois dimensions de l'espace durant l'analyse des textures présentes dans l'environnement. Le choix de l'unité de texture illustrée dans la Figure 5-4 implique des histogrammes de distribution de taille $[2^6 \times b]$ soit $[64 \times b]$ unités. Malgré la réduction de la taille des histogrammes d'un facteur de quatre par rapport au cas bidimensionnel, les résultats de la segmentation tridimensionnelle sont demeurés cohérents, tel qu'il le sera démontré dans les sections suivantes.

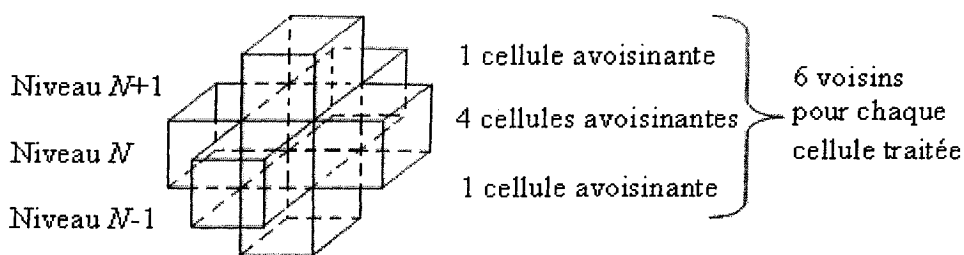


Figure 5-4 : représentation 3D de l'unité de texture adoptée

Comme dans le cas bidimensionnel, pour une région tridimensionnelle donnée, le *LBP* (*Local Binary Pattern*) et le *C* (*Contraste*) sont calculés pour chacune des cellules, à l'exception de celles qui sont situées sur les bordures de l'environnement modélisé. Pour chacune des cellules traitées, les six éléments avec qui cette dernière partage une surface sont considérés (Figure 5-4). La Figure 5-5

permet de mieux illustrer l'unité de texture. Cette figure correspond à une représentation aplatie d'une unité de texture. Dans cette figure, la cellule centrale ainsi que ses voisines de droite, de gauche, du haut et du bas se situent au niveau N , tandis que les voisines de niveaux inférieur et supérieur se situent respectivement aux niveaux $N-1$ et $N+1$. L'unité de texture considérée (Figure 5-5-a) subit un processus de discrétisation qui conduit à une représentation binaire. Dans ce processus, l'intensité de la cellule centrale est utilisée comme seuil. Toute valeur supérieure ou égale à celle-ci est mise à un, sinon elle est annulée (Figure 5-5-b). Les valeurs binaires obtenues sont multipliées par des poids binomiaux (Figure 5-5-c) et les résultats (Figure 5-5-d) sont additionnées afin d'obtenir le LBP ($LBP = 2 + 8 + 16 + 32 = 58$) de l'unité de texture tridimensionnelle. Il est important de noter que tout comme dans le cas bidimensionnel, la valeur de la cellule centrale n'est pas considérée dans cette étape.

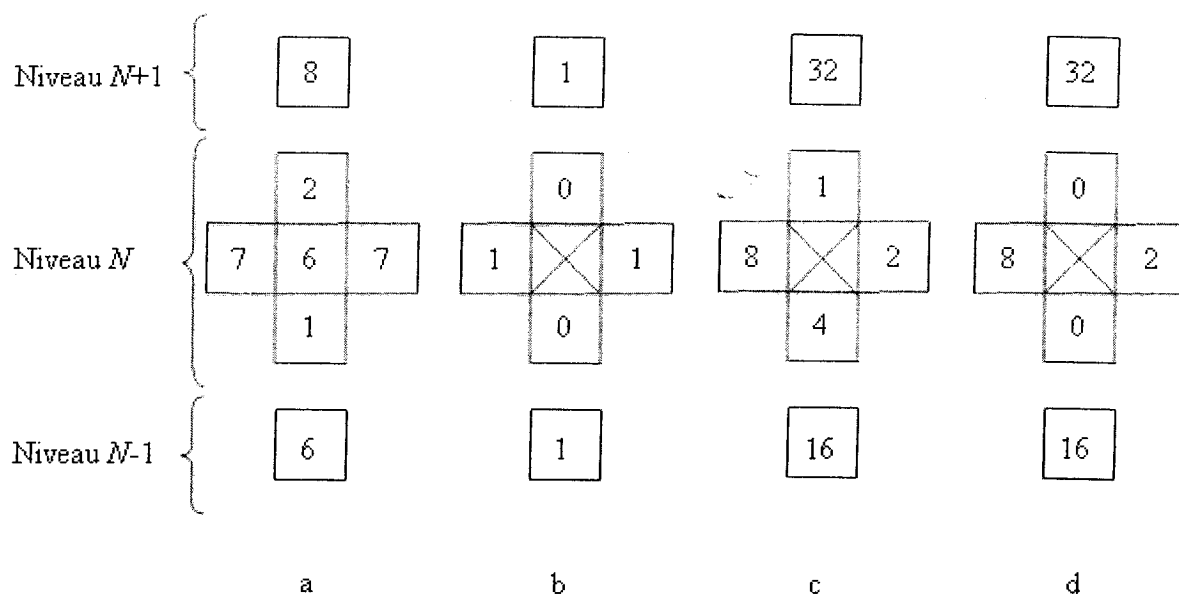


Figure 5-5 : exemple de calcul des caractéristiques de la texture tridimensionnelle

Le calcul du contraste suit le modèle développé par Ojala et Pietikäinen [39]. Ainsi, le C correspond à la différence entre les moyennes des valeurs des cellules dont la représentation binaire (après seuillage) est de 1 et celles dont la représentation binaire (après seuillage) est de 0. La valeur du contraste relative à l'unité de texture représentée dans la Figure 5-5-a est calculée comme suit : $C = (8 + 7 + 7 + 6) / 4 - (2 + 1) / 2 = 5.5$. Tel que défini précédemment, la distribution LBP/C est représentée dans un histogramme bidimensionnel de taille $64 * b$, où b correspond au nombre de niveaux de discrétisation du contraste C (Figure 2-30). Il est évident que la valeur du contraste est

décimale et appartient à l'intervalle fermé $[-254;255]$ (dans le cas où l'intensité des cellules est bornée entre 0 et 255). En se basant sur [38] et [39] et sur les résultats que nous avons obtenus dans le cas bidimensionnel, une valeur de b égale à 8 ou à 16 fournit des résultats et des erreurs de segmentation similaires. Par conséquent, nous optons pour une valeur de 8, réduisant ainsi d'un facteur 2 la taille des histogrammes de distribution LBP/C à comparer. Cependant, d'après les tests que nous avons effectués, une valeur de b égale à 4 réduit considérablement la sensibilité de l'algorithme aux différences de contraste observées entre les régions *occupées* et *libres*. Pour cette raison, cette valeur n'a pas été retenue.

5.3.2 Division hiérarchique

L'objectif ainsi que le cheminement de la division hiérarchique tridimensionnelle demeurent les mêmes que ceux de la segmentation bidimensionnelle (section 3.3.1). Ils consistent à subdiviser le modèle original en blocs tridimensionnels de taille variable mais de texture relativement uniforme.

Cette phase commence par subdiviser l'environnement probabiliste en régions cubiques de taille $[S_{max} \times S_{max} \times S_{max}]$, la valeur de S_{max} étant de 64 cellules. En terme d'implantation, une région R donnée est représentée par une structure qui encapsule les données suivantes :

- ❖ Les coordonnées x_R , y_R et z_R de son origine (cellule en arrière à gauche du niveau le plus bas);
- ❖ Ses dimensions l_R (largeur), h_R (hauteur) et e_R (élévation);
- ❖ La probabilité moyenne d'occupation PMO_R (équation (5.9)) qui la caractérise.

La probabilité moyenne d'occupation n'est pas calculée pour les subdivisions initiales car ces dernières ne subissent aucune fusion entre elles.

Toutefois, une version adaptée du test d'uniformité [39] a dû être développée afin de déterminer si une subdivision « parent » contient des textures hétérogènes et doit être subdivisée en huit sous-régions de taille égale (Figure 5-6). Ce test commence par identifier les huit subdivisions cubiques qui sont issues d'une région parent et dont les dimensions sont exactement égales à la moitié de celles de cette dernière. La Figure 5-7 illustre le principe, elle représente les huit subdivisions issues d'une région parent de coordonnées (x_p, y_p, z_p) et de dimensions (l_p, h_p, e_p) .

Comme signalé précédemment, les dimensions de ces subdivisions sont égales à $(l_p / 2, h_p / 2, e_p / 2)$.
 Le calcul de leurs coordonnées respectives est représenté dans la Figure 5-7.

Région parent P :
 coordonnées (x_p, y_p, z_p)
 dimensions (l_p, h_p, e_p)

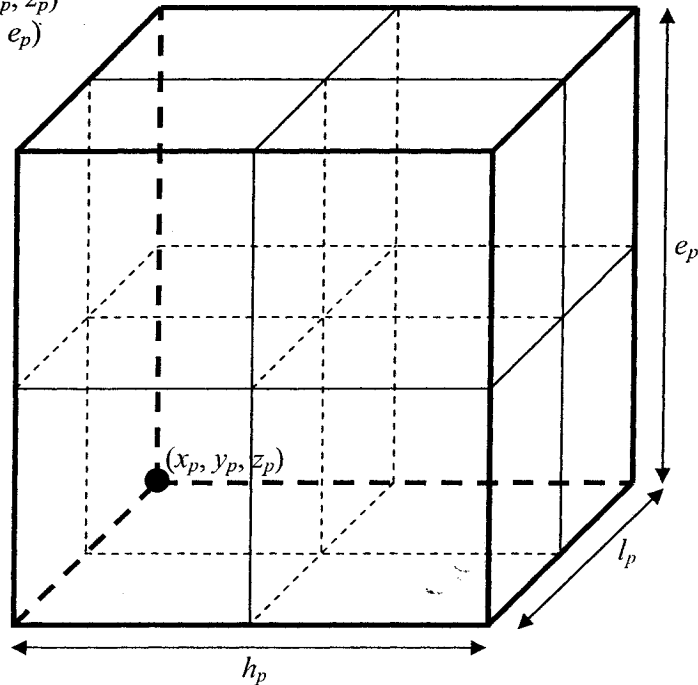


Figure 5-6 : représentation d'une région tridimensionnelle et de ses huit subdivisions

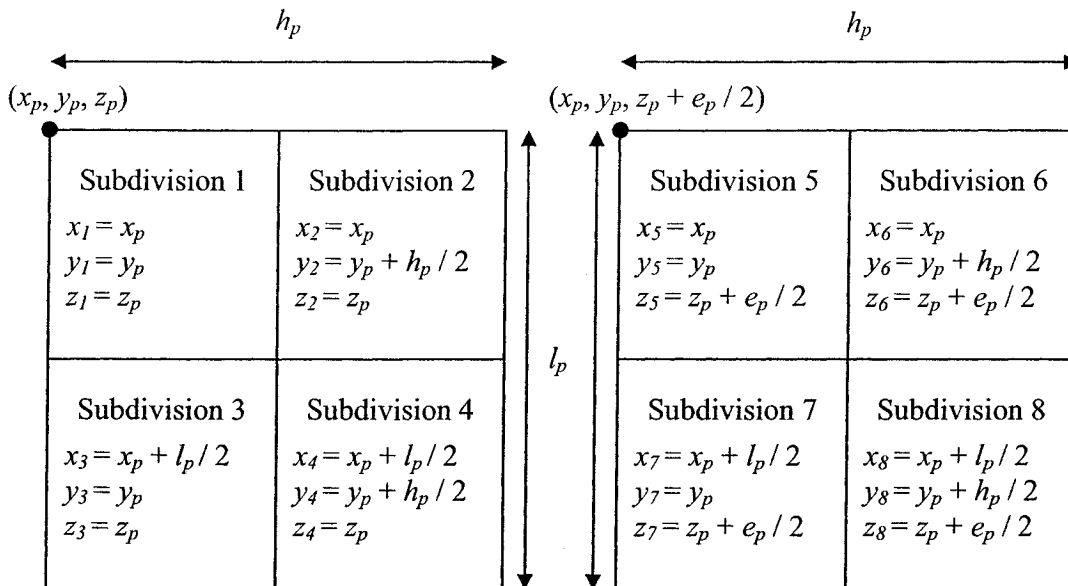


Figure 5-7 : identification et coordonnées des subdivisions issues d'une région parent

Pour chacune des subdivisions, l'histogramme de distribution LBP/C est calculé. La seule différence avec l'implantation bidimensionnelle provient de la dimension additionnelle ainsi que des changements que nous avons appliqués à l'unité de texture. À noter que les cellules sur les bordures ne sont pas traitées car leur entourage n'est pas défini dans la région en question. L'unité de texture (Figure 5-5-a), la structure binaire (Figure 5-5-b), la structure de poids binomiaux (Figure 5-5-c) et la structure de texture locale (Figure 5-5-d) sont toutes contenues dans des matrices bidimensionnelles de taille [3 x 3]. Comme le montre la Figure 5-8, ces matrices bidimensionnelles encodent la structure tridimensionnelle de l'unité de texture réelle et permettent de calculer, comme dans le cas bidimensionnel, les valeurs du *LBP* et du *C* et par conséquent de construire l'histogramme de distribution.

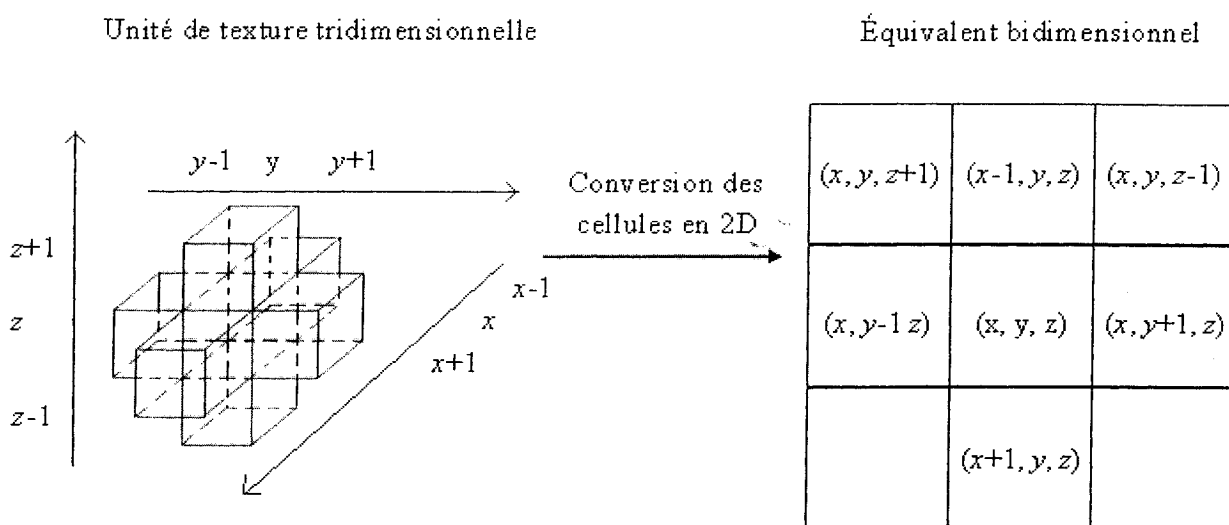


Figure 5-8 : représentation 2D d'une unité de texture 3D

Les huit histogrammes de distribution LBP/C correspondant aux huit subdivisions servent à calculer les 28 ratios logarithmiques de probabilité (équation (2.5)) entre chacune des 28 paires possibles de sous-régions. En effet, selon la loi des combinaisons sans répétition [68] (équation (5.1)), à l'aide d'un ensemble de huit éléments, nous pouvons former 28 sous-ensembles formés chacun de deux éléments distincts.

$$C_n^k = \frac{A_n^k}{k!} = \frac{n!}{k!(n-k)!} = \frac{8!}{2!(8-2)!} = 28 \quad (5.1)$$

Avec :

$$A_n^k = \frac{n!}{(n-k)!} = \frac{8!}{(8-2)!} = 56 \quad (5.2)$$

Où : n : nombre d'éléments à disposition (*i.e.* : nombre de subdivisions issues d'une région parent soit 8).

k : taille des sous-ensembles que nous voulons former à partir des n éléments à disposition (*i.e.* : notre objectif est de regrouper les subdivisions en paires afin de pouvoir calculer le G-statistique).

A_n^k : nombre d'arrangement sans répétition de n éléments pris k à k .

C_n^k : nombre de combinaison sans répétition de n éléments pris k à k .

La plus grande ainsi que la plus faible des 28 valeurs du G-statistique sont mises en évidence et dénotées respectivement par G_{max} et G_{min} . La « région parent » est considérée non-uniforme et donc subdivisée, si le rapport entre G_{max} et G_{min} est supérieur à un certain seuil X (équation (2.6)). Les concepts derrière le choix de la valeur de X (soit de 1.2) sont similaires à ceux exposés dans le troisième chapitre (section 3.3.1).

Si une région échoue au test d'uniformité, les sous-régions obtenues subissent à leur tour ce test. Le processus itératif agit sur les régions subdivisées jusqu'à ce que les blocs obtenus soient de taille $[S_{min} \times S_{min} \times S_{min}]$, la valeur de S_{min} étant de 8 cellules. Le maximum de trois niveaux de subdivision assure l'uniformité relative des textures dans chacun des blocs obtenus. Ojala et Pietikäinen [39] indiquent que deux niveaux de subdivision sont suffisants et fournissent un résultat de segmentation adéquat. Mais tout comme dans le cas bidimensionnel (section 4.6), la segmentation tridimensionnelle fournit des résultats plus stables avec un troisième niveau de subdivision menant à une valeur de S_{min} égale à huit (section 5.4).

5.3.3 Création des segments

Tout comme dans le cas de la segmentation bidimensionnelle (section 3.3.2), le but de cette phase est de fusionner les régions adjacentes d'occupation similaire jusqu'à ce qu'un critère de convergence soit atteint.

La recherche de voisins dans un environnement tridimensionnel correspond à une extension de la méthodologie bidimensionnelle (Figure 3-6). Pour une région r donnée (coordonnées de l'origine (x_r, y_r, z_r) et dimensions (l_r, h_r, e_r)), un voisin v (coordonnées de l'origine (x_v, y_v, z_v) et dimensions (l_v, h_v, e_v)) du dessus doit répondre aux deux exigences suivantes :

$$z_v = z_r + e_r \quad (5.3)$$

$$\begin{aligned} & [(x_r \leq x_v \leq x_r + l_r) \text{ ET } (y_r \leq y_v \leq y_r + h_r)] \text{ OU} \\ & [(x_r \leq x_v + l_v \leq x_r + l_r) \text{ ET } (y_r \leq y_v + h_v \leq y_r + h_r)] \text{ OU} \\ & [(x_v \leq x_r \leq x_v + l_v) \text{ ET } (y_v \leq y_r \leq y_v + h_v)] \end{aligned} \quad (5.4)$$

Tandis qu'un voisin v de droite doit répondre aux exigences suivantes :

$$y_v = y_r + h_r \quad (5.5)$$

$$\begin{aligned} & [(x_r \leq x_v \leq x_r + l_r) \text{ ET } (z_r \leq z_v \leq z_r + e_r)] \text{ OU} \\ & [(x_r \leq x_v + l_v \leq x_r + l_r) \text{ ET } (z_r \leq z_v + e_v \leq z_r + e_r)] \text{ OU} \\ & [(x_v \leq x_r \leq x_v + l_v) \text{ ET } (z_v \leq z_r \leq z_v + e_v)] \end{aligned} \quad (5.6)$$

Finalement, un voisin v du bas doit, quand à lui, répondre aux deux exigences suivantes :

$$x_v = x_r + l_r \quad (5.7)$$

$$\begin{aligned} & [(y_r \leq y_v \leq y_r + h_r) \text{ ET } (z_r \leq z_v \leq z_r + e_r)] \text{ OU} \\ & [(y_r \leq y_v + h_v \leq y_r + h_r) \text{ ET } (z_r \leq z_v + e_v \leq z_r + e_r)] \text{ OU} \\ & [(y_v \leq y_r \leq y_v + h_v) \text{ ET } (z_v \leq z_r \leq z_v + e_v)] \end{aligned} \quad (5.8)$$

En observant les équations ci-dessus nous pouvons remarquer que les exigences sur lesquelles se base la détection d'un voisin sont similaires quelque soit le coté de recherche. En effet, l'équation (5.3) ((5.5) ; (5.7)) permet de sélectionner les régions dont l'origine se situe dans le plan parallèle à (x, y) $((x, z) ; (y, z))$ longeant le coté du dessus (de droite ; du bas) de r . L'équation (5.4) ((5.6) ; (5.8)) ajoute les contraintes dans ce plan et traite les trois cas suivants :

- ❖ Le premier cas où la projection de l'origine de v dans le plan (x, y) $((x, z) ; (y, z))$ dénotée par $v_{(x, y)}$ $(v_{(x, z)} ; v_{(y, z)})$ appartient à la section du plan formé par la projection de r dans (x, y) $((x, z) ; (y, z))$ et dénotée par $r_{(x, y)}$ $(r_{(x, z)} ; r_{(y, z)})$.
- ❖ Le second cas où la projection de l'extrémité de droite-bas (du dessus-bas ; du dessous-droite) dans le plan (x, y) $((x, z) ; (y, z))$ appartient à $r_{(x, y)}$ $(r_{(x, z)} ; r_{(y, z)})$.

- ❖ Le troisième cas où $v_{(x,y)}$ ($v_{(x,z)}$; $v_{(y,z)}$) est suffisamment grande pour inclure $r_{(x,y)}$ ($r_{(x,z)}$; $r_{(y,z)}$).

La Figure 5-9 permet de visualiser les trois cas décrits ci-dessus (respectivement dans la Figure 5-9 a, b et c) pour les voisins du dessus. Dans ces images, la région r (hachurée), est exactement au dessous de v (niveau gris uniforme) dans le sens de l'axe z .

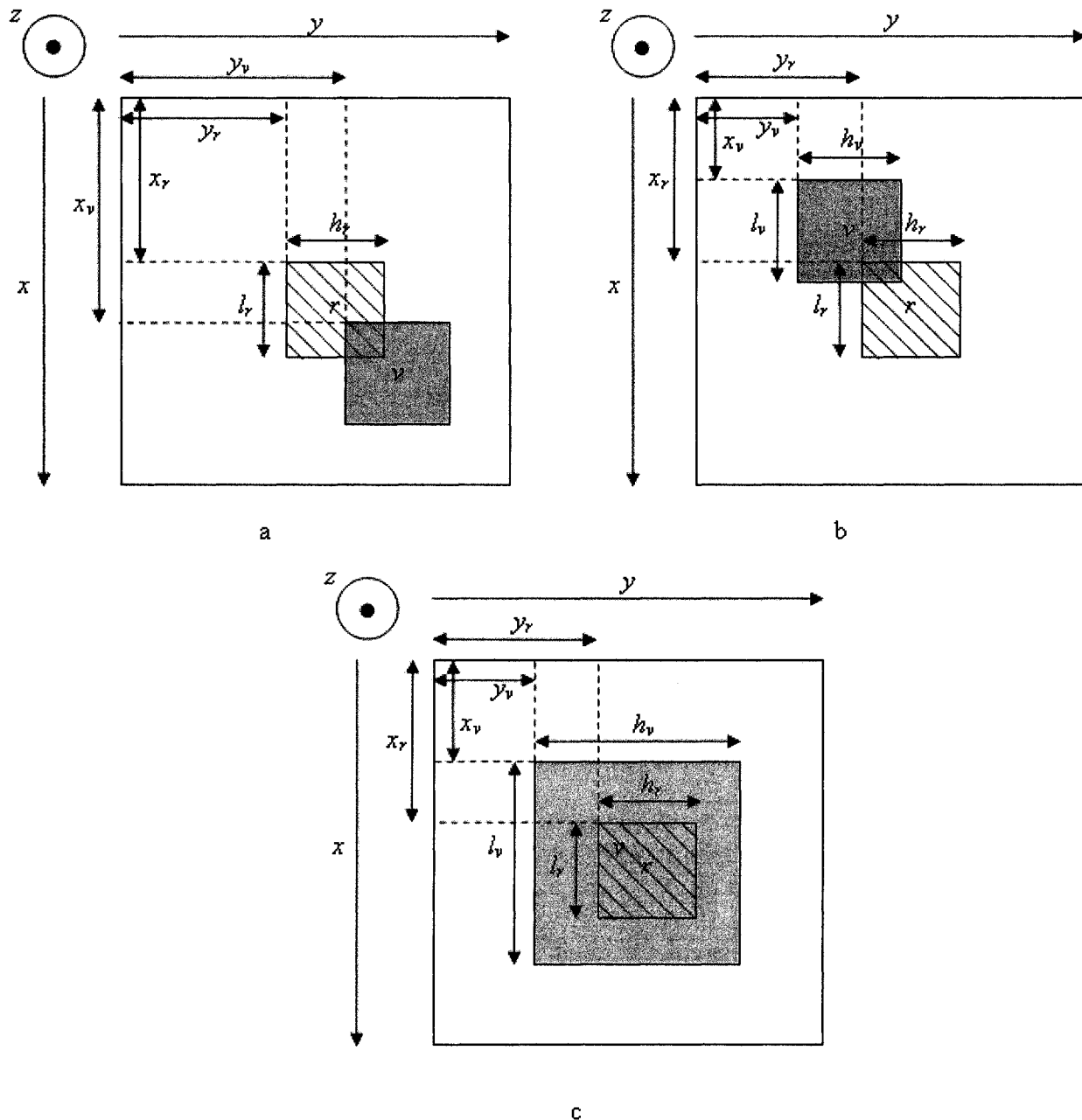


Figure 5-9 : localisation d'un voisin v au-dessus d'une région r donnée

Le fait d'identifier pour une région donnée l'ensemble de ses voisins du dessus, de droite et du bas garantit à la fin du parcours de l'ensemble des subdivisions de l'environnement tridimensionnel la prise en compte de toutes les paires possibles de régions adjacentes.

Tout comme dans le cas bidimensionnel, la fusion implique les blocs avoisinants dont les probabilités moyennes d'occupation, ou *PMOs*, appartiennent au même intervalle de mesure. Ce paramètre qui est défini comme la moyenne de l'intensité des cellules dans une région donnée, est calculé pour le cas des régions tridimensionnelles à l'aide de l'équation suivante :

$$PMO_i = \frac{1}{M \cdot N \cdot O} \sum_{l=0}^{O-1} \sum_{k=0}^{N-1} \sum_{j=0}^{M-1} I(r_{j,k,l}) \mid r_{j,k,l} \in R_i \quad (5.9)$$

Où : PMO_i : probabilité moyenne d'occupation assignée à la région R_i .

R_i : région R d'indice i .

M , N et O : dimensions de la région R_i .

$I(r_{j,k,l})$: intensité du pixel r , de coordonnées (j, k, l) .

Les motivations qui nous ont poussés à choisir la *PMO* comme critère capable d'évaluer la similarité entre les textures présentes dans des régions adjacentes sont similaires à celles présentées à la section 3.3.2, et concernent principalement la structure des environnements probabilistes.

Les intervalles qui agissent comme paramètres de décision quand à la fusion de deux ou plusieurs régions adjacentes sont aussi les mêmes que ceux choisis dans le cas de l'algorithme bidimensionnel. Ceux-ci conduisent à la segmentation de l'espace en régions caractérisées par des états déterministes (équation (3.13)). Les trois états d'occupation, soient *libre*, *inconnu* et *occupé*, correspondent respectivement aux intervalles $[0;0.498[$, $[0.498;0.502]$ et $]0.502;1]$ des valeurs de la *PMO*. En se basant sur ces faits, tout ensemble de régions adjacentes dotées de *PMOs* dans le même intervalle sont fusionnées et classifiées comme étant un segment caractérisé par un niveau d'occupation uniforme. Seuls les segments *occupés* et non-adjacents sont considérés comme des entités séparées afin d'éviter toute confusion possible dans la reconnaissance des objets et dans l'interaction avec l'environnement.

5.3.4 Raffinement

La réussite des deux premières étapes mène à une segmentation approximative, du fait que les bordures des segments obtenus ne correspondent pas parfaitement à celles qui délimitent les régions dans l'environnement probabiliste. Par conséquent, la tâche principale de cette dernière phase se limite à parfaire la localisation des contours des segments en réassignant les cellules localisées sur la bordure de deux ou plusieurs régions adjacentes. Cette dernière phase, tout comme celles qui l'ont précédée, correspond à une extension directe de celle développée dans le cas de la segmentation bidimensionnelle.

Étant donné que la structure des environnements probabilistes tridimensionnels est similaire à celle des images bidimensionnelles, les changements opérés à cette phase (section 3.3.3) sont mineurs et se limitent à son adaptation et à la prise en compte de la dimension supplémentaire.

Le premier raffinement consiste à dilater les régions *inconnues* au détriment de leurs voisins *libres*. Ceci étant motivé par le contexte de l'application des grilles d'occupation probabilistes à la planification de trajectoires exemptes de collisions en robotique. Ainsi, il s'agit de parcourir l'environnement tridimensionnel dans les six directions possibles soient : « droite – gauche », « gauche – droite », « haut – bas », « bas – haut », « dessous – dessus » et « dessus – dessous ». Il faut noter qu'en 3D, deux directions de parcours supplémentaires sont requises. Lors d'un parcours, au niveau des bordures entre les espaces *inconnu* et *libre*, les cellules *libres* (le long de la direction du parcours) dont l'intensité est de 0.5 sont reclassées comme appartenant à la région *inconnue*. Ce processus continue jusqu'à ce qu'une cellule dotée d'une valeur différente soit atteinte. Le fait de parcourir l'environnement dans les six directions possibles assure la couverture de toutes les formes possibles de bordures.

La seconde adaptation concerne la double classification entre les segments *libres – occupés* d'une part et *occupés – libres* de l'autre. Le processus de classification demeure le même que dans le cas bidimensionnel (Figure 3-9), mais la dimension additionnelle implique tout comme pour la première étape, un parcours dans les six directions possibles soient : « droite – gauche », « gauche – droite », « haut – bas », « bas – haut », « dessous – dessus » et « dessus – dessous ».

5.4 Résultats et analyse

Dans les sections précédentes, nous avons développé l'algorithme ainsi que les détails de l'implantation relatifs à la segmentation tridimensionnelle. Cette partie est consacrée à la présentation et à l'analyse des résultats obtenus à l'aide de l'implantation de l'algorithme tridimensionnel réalisée avec le langage de programmation C++ (Annexe C) sur un ordinateur doté d'un processeur Athlon 2600 XP [AMD Inc.] avec un giga-octet de mémoire vive.

5.4.1 Définition des modèles tridimensionnels

Afin de valider le processus en trois dimensions, nous avons testé l'implantation proposée sur six environnements tridimensionnels. Les trois premiers ont une taille de [320 x 320 x 64], les autres de [320 x 320 x 128]. Des objets coniques, cubiques et cylindriques sont incorporés respectivement dans un modèle de dimensions [320 x 320 x 64] et dans un autre de dimensions [320 x 320 x 128]. Les détails concernant chacun des modèles sont résumés dans les points ci-dessous :

- ❖ Le premier environnement (espace occupé réel – Figure 5-16, Figure 5-17 et Figure 5-18) est obtenu en superposant 64 images contenant des cercles centrés sur le même pixel mais de rayon décroissant. Les rayons du plus grand et du plus petit d'entre eux sont respectivement de 67 et de 4 pixels. Les images probabilistes issues ont toutes les mêmes caractéristiques du fait qu'elles incorporent six points de vue situés aux mêmes endroits et dirigés vers les mêmes directions. Le pas angulaire entre les rayons adjacents est fixé à 0.5 degré, tandis que l'écart-type de l'erreur gaussienne sur la profondeur est égal à 4 cm. L'image de la Figure 5-10-a représente la tranche du cône de rayon égal à 45 pixels. L'image probabiliste correspondante (Figure 5-10-b) incorpore les 6 prises de vue permettant de balayer la totalité de la section horizontale de l'objet.

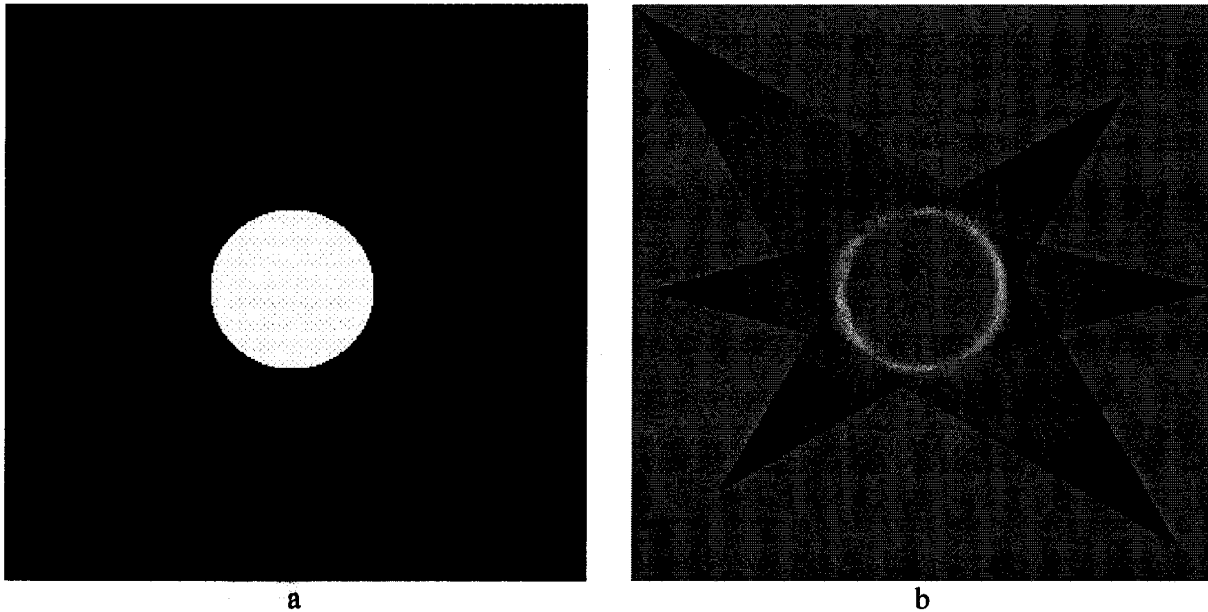
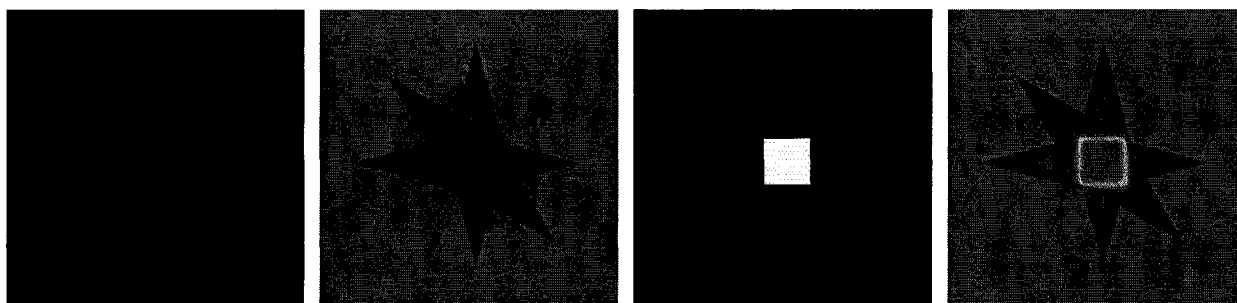


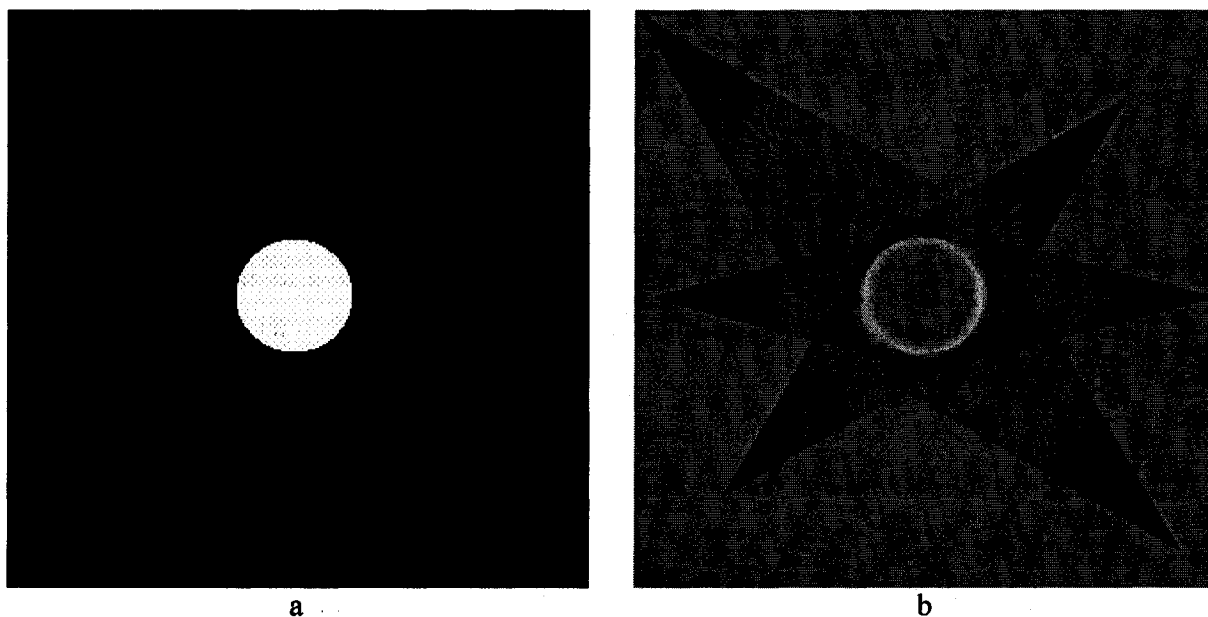
Figure 5-10 : section horizontale des environnements réel et probabiliste de taille [320 x 320 x 64] incluant un objet conique

- ❖ Le second environnement (espace occupé réel – Figure 5-19, Figure 5-20 et Figure 5-21) est obtenu de la même manière. Mais cette fois-ci, les régions occupées circulaires sont remplacées par des régions rectangulaires de taille [50 x 50] pixels (Figure 5-11-c). Cinquante de ces images sont nécessaires afin que la région occupée tridimensionnelle soit cubique et de taille [50 x 50 x 50]. Étant donné que l'environnement global est formé de 64 couches horizontales, les sept premières et les sept dernières couches sont laissées libres d'objets (Figure 5-11-a). Les images probabilistes (Figure 5-11 b et d) issues de la représentation réelle de l'environnement incluent toutes six lectures de capteur positionnées aux mêmes endroits dans le plan (x, y) mais à des élévations différentes. Tout comme le modèle précédent, l'écart-type de l'erreur sur la distance est égale à 4 cm, tandis que le pas angulaire entre deux faisceaux de lecture adjacents est de 0.5 degré.



a b c d
Figure 5-11 : sections horizontales des environnements réel et probabiliste de taille [320 x 320 x 64] incluant un objet cubique. (a) et (b) les 7 premières et les 7 dernières couches horizontales de l'environnement simulé et probabiliste, (c) et (d) les 50 couches intermédiaires de l'environnement simulé et probabiliste

- ❖ L'objet cylindrique (espace occupé réel – Figure 5-22, Figure 5-23 et Figure 5-24) représenté dans le troisième environnement est obtenu en superposant 64 régions bidimensionnelles, occupées et circulaires de rayon égal à 32 pixels (Figure 5-12-a). La position ainsi que les caractéristiques du capteur utilisé afin de générer les images probabilistes (Figure 5-12-b) sont similaires au cas de l'environnement incorporant un objet conique.



a b
Figure 5-12 : section horizontale des environnements réel et probabiliste de taille [320 x 320 x 64] incluant un objet cylindrique

- ❖ Le quatrième environnement (espace occupé réel – Figure 5-25) est obtenu en superposant 128 images contenant des cercles centrés sur le même pixel mais de rayon décroissant où, pour chaque deux couches horizontales, le rayon est décrétementé de 1 pixel. Le rayon du

plus grand et du plus petit cercles sont respectivement de 67 (Figure 5-13-a) et de 4 pixels. Les images probabilistes qui forment les couches horizontales du modèle probabiliste tridimensionnel incorporent huit points de vue (Figure 5-13-b). Les capteurs utilisés dans la construction du modèle probabiliste ont les mêmes caractéristiques que ceux utilisés dans les environnements probabilistes précédents.

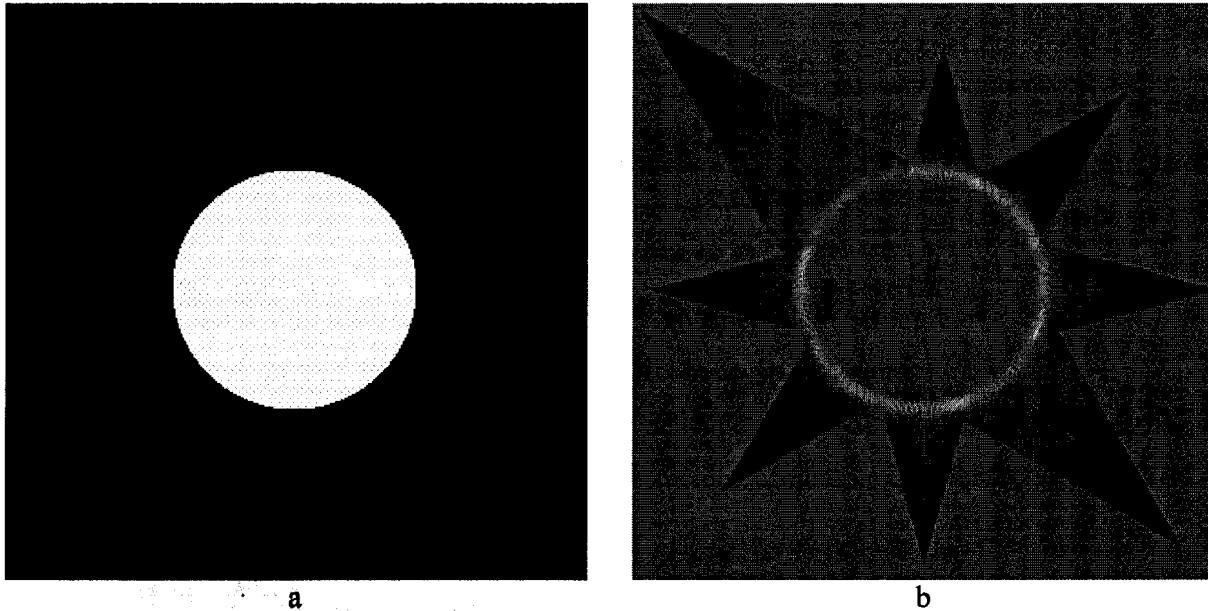
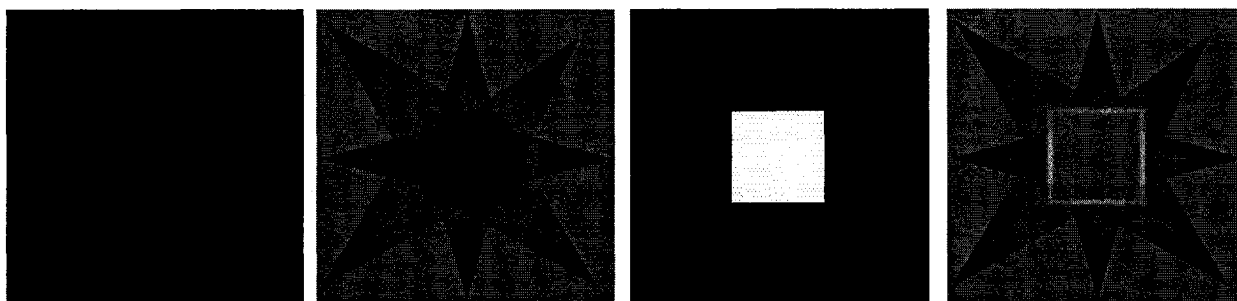


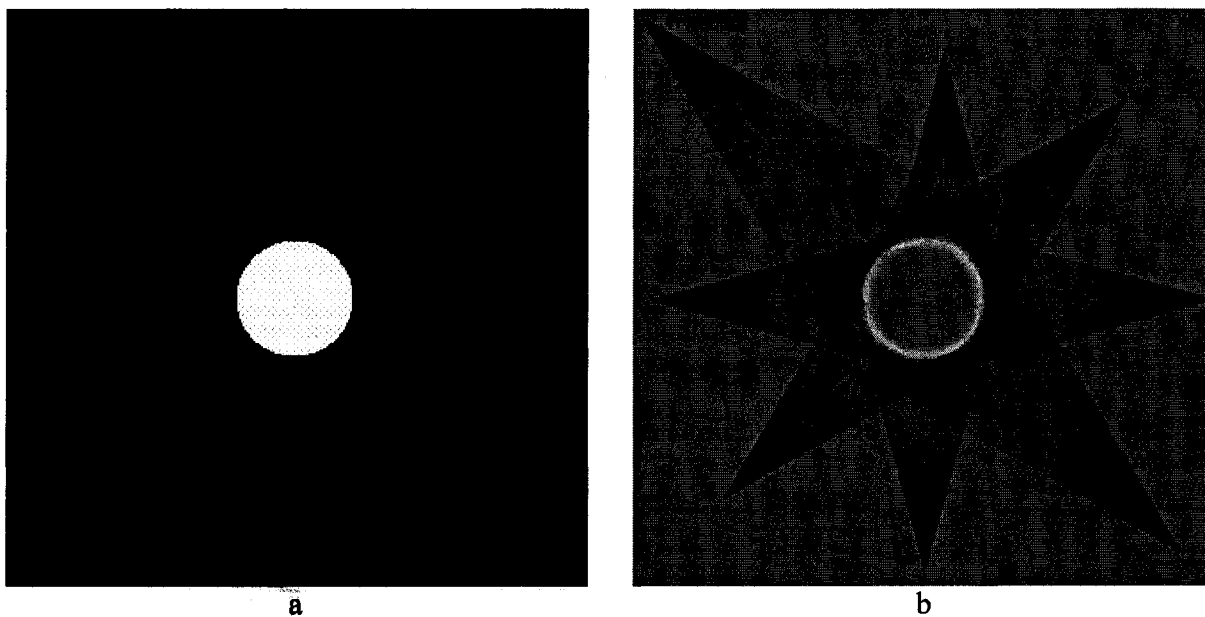
Figure 5-13 : section horizontale des environnements réel et probabiliste de taille [320 x 320 x 128] incluant un objet conique

- ❖ L'environnement de taille [320 x 320 x 128] incorporant un cube (espace occupé réel – Figure 5-26) est obtenu de la même manière que celui de taille [320 x 320 x 64], mais les régions occupées rectangulaires qui caractérisent les sections horizontales de l'objet sont de [100 x 100] pixels (Figure 5-14-c). Étant donné que l'environnement tridimensionnel est formé de 128 couches horizontales, le cube s'étale de l'image bidimensionnelle d'élévation 14 jusqu'à celle d'élévation 113, ce qui confère à l'objet une taille globale de [100 x 100 x 100] cellules. Les autres images sont complètement libres (Figure 5-14-a) et leur représentation probabiliste inclut, tout comme les autres couches, 8 points de vue centrés sur les mêmes coordonnées dans le plan (x, y) et dirigés vers les mêmes directions (Figure 5-14 b et d).



a b c d
Figure 5-14 : sections horizontales des environnements réel et probabiliste de taille [320 x 320 x 128] incluant un objet cubique. (a) et (b) les 14 premières et 14 dernières couches horizontales de l'environnement simulé et probabiliste, (c) et (d) les 100 couches intermédiaires de l'environnement simulé et probabiliste

- ❖ L'objet cylindrique (espace occupé réel – Figure 5-27) représenté dans le dernier environnement est obtenu en superposant 128 régions bidimensionnelles, occupées et circulaires de rayon égal à 32 pixels (Figure 5-15-a). Les capteurs utilisés dans la génération des images probabilistes ont les mêmes caractéristiques que ceux des environnements précédents. À la différence du modèle de taille [320 x 320 x 64] incorporant un cylindre, la représentation probabiliste de cet environnement inclue huit points de vue (Figure 5-15-b) au lieu de six (Figure 5-12-b).



a b
Figure 5-15 : section horizontale des environnements réel et probabiliste de taille [320 x 320 x 128] incluant un objet cylindrique

5.4.2 Évaluation des performances

Afin de justifier le choix des valeurs attribuées aux paramètres de l'algorithme, nous avons varié d'une exécution à l'autre le niveau de discrétisation du contraste, b , ainsi que la taille minimale, S_{min} , qu'une subdivision peut atteindre durant la première phase (Tableau 5-1). Pour chacune des segmentations, nous avons noté les valeurs des erreurs de type 1 (équation (4.1)) et de type 2 (équation (4.2)) pour les deux dernières phases, ainsi que les temps d'exécution de chacune des trois phases.

Modèle traité	Dimensions	b	S_{min}	Erreur type 1 (%)		Erreur type 2 (%)		Temps d'exécution (secondes)				
				Phase 2	Phase 3	Phase 2	Phase 3	Phase 1	Phase 2	Phase 3	Total	
Cône 1	[320 x 320 x 64]	8	8	0.67	0.63	0.06	0.00	96.97	8.60	213.17	318.74	
			16	0.47	0.63	0.42	0.03	77.67	5.94	213.60	297.21	
		16	8	0.67	0.63	0.06	0.00	100.9	8.64	213.00	322.54	
			16	0.47	0.63	0.42	0.03	78.84	5.92	213.62	298.38	
Cube 1		[320 x 320 x 64]	8	8	0.46	0.60	0.07	0.01	75.21	7.56	212.75	295.52
				16	0.28	0.38	0.69	0.25	64.21	5.87	213.73	283.81
			16	8	0.46	0.60	0.07	0.01	78.98	7.54	212.84	299.36
				16	0.28	0.38	0.69	0.25	65.28	6.00	211.72	283.00
Cylindre 1	[320 x 320 x 64]		8	8	0.64	0.52	0.02	0.00	96.82	8.76	214.45	320.03
				16	0.56	0.52	0.19	0.01	83.95	5.98	213.01	302.94
			16	8	0.64	0.52	0.02	0.00	99.93	8.76	212.56	321.25
				16	0.56	0.52	0.19	0.01	77.87	5.90	211.87	295.64
Cône 2		[320 x 320 x 128]	8	8	0.79	0.74	0.04	0.00	203.82	26.20	426.11	656.13
				16	0.67	0.74	0.32	0.01	158.59	11.95	427.05	597.59
			16	8	0.79	0.74	0.04	0.00	214.89	26.37	426.72	667.98
				16	0.67	0.74	0.32	0.01	162.34	12.00	427.39	601.73
Cube 2	[320 x 320 x 128]		8	8	1.68	1.02	0.31	0.03	207.92	42.00	427.48	677.40
				16	0.15	1.02	0.84	0.10	159.46	12.31	427.81	599.58
			16	8	1.68	1.02	0.31	0.03	233.06	41.98	427.62	702.66
				16	0.15	1.02	0.84	0.10	162.14	12.22	428.00	602.36
Cylindre 2		[320 x 320 x 128]	8	8	0.35	0.40	0.04	0.00	204.55	22.03	425.50	652.08
				16	0.18	0.40	0.32	0.00	159.36	12.00	427.42	598.78
			16	8	0.35	0.40	0.04	0.00	214.95	21.89	429.69	666.53
				16	0.18	0.40	0.32	0.00	161.05	11.89	428.67	601.61

Tableau 5-1 : mesures d'erreur et temps d'exécution pour chacun des modèles traités

Comme nous l'avons introduite dans le second chapitre, l'erreur de type 1 nous renseigne sur la proportion de cellules réellement libres mais qui ont été classifiées comme *occupées*. En observant les valeurs de cette erreur dans le Tableau 5-1, nous constatons que, dans la majorité des cas traités, elle demeure inchangée pour une valeur de S_{min} égale à 8 ou à 16. Ce n'est que dans le cas de

l'environnement tridimensionnel de taille [320 x 320 x 64] incorporant un objet cubique que l'erreur de type 1 diminue de 0.22% lorsque S_{min} passe de 8 à 16. Cette diminution qui s'accompagne d'une augmentation de 0.24% de l'erreur de type 2 démontre, tout comme dans le cas bidimensionnel (Figure 4-25), qu'une valeur de S_{min} égale à 16 tend à surévaluer les segments *libres* au détriment des régions *occupées* qui leurs sont adjacentes. La valeur relativement faible de cette mesure d'erreur (inférieure à 1.02% dans tous les cas) est rassurante du fait qu'elle indique que le débordement des segments *occupés* sur les régions *libres* qui leur sont avoisinantes est mineur.

L'erreur de type 2 est plus critique que la première, étant donné qu'elle renseigne sur la proportion des cellules réellement occupées mais qui ont été classifiées comme *libres*. La subdivision supplémentaire opérée lors de la division hiérarchique avec un $S_{min} = 8$ réduit en moyenne de 0.06% la valeur de cette erreur à la fin de la dernière phase. Ainsi, pour une valeur de S_{min} égale à 8 et pour les six environnements tridimensionnels testés, l'erreur de type 2 demeure inférieure à 0.03% et se caractérise par une moyenne de 0.006%. Cette valeur négligeable combinée à la valeur de l'erreur de type 1, implique que les segments obtenus englobent presque parfaitement les régions correspondantes de l'environnement simulé.

Le nombre de niveaux b de discrétisation du contraste n'a aucune influence sur les mesures d'erreur (Tableau 5-1), mais il influence légèrement le temps d'exécution. En effet, pour une valeur de S_{min} égale à 8, l'utilisation de 8 niveaux de discrétisation pour le contraste réduit le temps d'exécution global de 1.77% (10.07 secondes) en moyenne pour les six environnements segmentés. Cette faible réduction du temps de calcul observée touche principalement la phase de division hiérarchique étant donné que cette dernière compare les histogrammes de distribution LBP/C.

La segmentation des environnements de taille [320 x 320 x 64] et de taille [320 x 320 x 128] nécessite en moyenne et respectivement 311.43 (≈ 5 min 11 sec) et 661.87 (≈ 11 min 1 sec) secondes. À la différence de la segmentation bidimensionnelle où la troisième phase occupe en moyenne 3% du temps d'exécution global, le raffinement dans la segmentation tridimensionnelle occupe en moyenne 66.53% du temps d'exécution global. La charge croissante de calcul que requiert la troisième phase dans la segmentation tridimensionnelle a principalement deux raisons :

- ❖ La première a pour origine le nombre important de cellules qui sont accédées durant cette étape. À titre d'exemple, un environnement de taille [320 x 320 x 128] est formé de 13107200 (= 320 x 320 x 128) cellules. La phase de raffinement traverse l'environnement

tridimensionnel deux fois (une fois pour le raffinement entre les segments *inconnus - libres* et une autre entre les segments *libres - occupés* et *occupés - libres*) dans les six directions possibles. Ceci fait qu'elle accède à un minimum de 157286400 (= 13107200 x 6 x 2) cellules;

- ❖ De plus, la conception des modèles tridimensionnels n'est pas tout à fait optimisée, en effet les images qui forment les couches horizontales du modèle tridimensionnel sont contenues dans une structure de données qui est constamment traversée afin d'accéder au contenu des cellules dotées de coordonnées (x, y, z) .

Une des solutions aux deux problèmes énumérés ci-dessus serait d'optimiser l'encodage du modèle tridimensionnel en adaptant l'algorithme proposé pour opérer sur les grilles d'occupation probabilistes à résolutions multiples. Cette approche ayant déjà démontré une augmentation considérable de la performance dans des applications de modélisation similaires [5].

En comparant la durée de la segmentation des environnements de taille [320 x 320 x 64] et de taille [320 x 320 x 128], nous remarquons que le temps d'exécution augmente linéairement avec la taille du modèle tridimensionnel traité. Ainsi, en doublant la taille de l'environnement, la charge de calcul est multipliée en moyenne par un facteur de 2.12 (= 661.87 secondes / 311.43 secondes). Dans le cas des environnements testés (Tableau 5-1), les trois phases sont touchées uniformément par cette augmentation, car le volume additionnel de taille [320 x 320 x 64] n'est pas uniforme et inconnu. Il contient une région d'occupation connue qui doit subir elle aussi les processus de subdivision, de fusion et de raffinement.

Il faut rappeler que la redéfinition de l'unité de texture (section 5.3.1) joue un rôle très important dans la limitation de l'explosion du temps de calcul de la segmentation tridimensionnelle. En effet, la prise en compte pour chaque cellule de ses 6 voisines (Figure 5-4) partageant avec elle une face commune, au lieu des 26 cellules contenues dans une structure cubique de taille [3 x 3 x 3] (Figure 5-3), réduit la taille des histogrammes de distribution LBP/C à comparer d'un facteur de 1048580 [= $((2^{26} \times 8) / (2^6 \times 8))$].

Un sous-ensemble des résultats exposés dans le Tableau 5-1 est illustré dans les neuf figures ci-dessous (de la Figure 5-16 à la Figure 5-24). La Figure 5-16, la Figure 5-19, et la Figure 5-22 représentent respectivement les segments *libres* relatifs aux environnements de taille [320 x 320 x 64] incorporant un cône, un cube et un cylindre. Dans ces images, nous pouvons distinguer

clairement le nombre de points de vue utilisés. En ce qui concerne la Figure 5-19, l'objet cubique n'est pas visible au milieu de l'espace *libre* car il n'occupe pas l'ensemble de la profondeur de l'environnement, et est entouré au dessus et au dessous de sept couches de cellules *libres*. Les segments *occupés* obtenus à la fin de la seconde phase sont représentés quand à eux dans la Figure 5-18, la Figure 5-21 et la Figure 5-24 respectivement pour les trois environnements segmentés de taille [320 x 320 x 64]. La Figure 5-17, la Figure 5-20 et la Figure 5-23, la Figure 5-25, la Figure 5-26 et la Figure 5-27 illustrent les segments *occupés* obtenus à la fin de la phase de raffinement pour l'ensemble des environnements tridimensionnels traités.

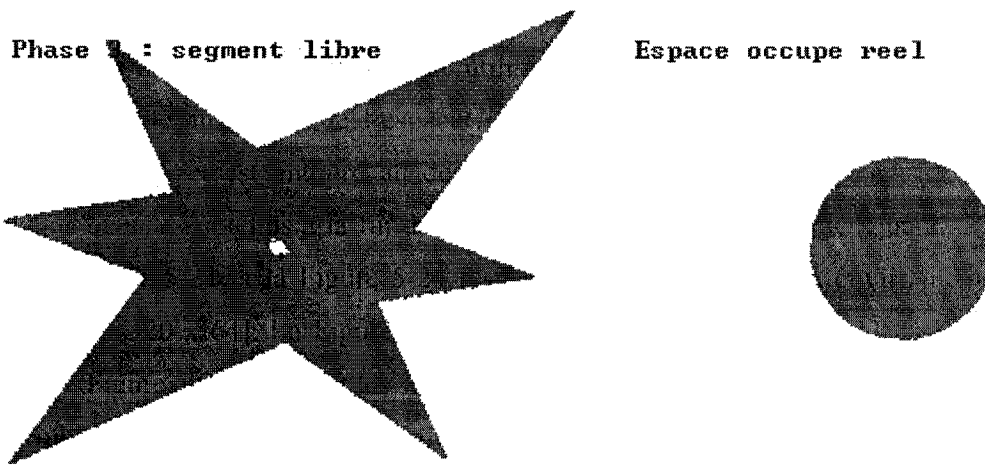


Figure 5-16 : vue d'en haut sur le segment *libre* obtenu à la fin de la 3^{ème} phase, seule la pointe de l'objet cône est visible au niveau de la couche *libre* du dessus (environnement de taille [320 x 320 x 64] incorporant un objet cône)

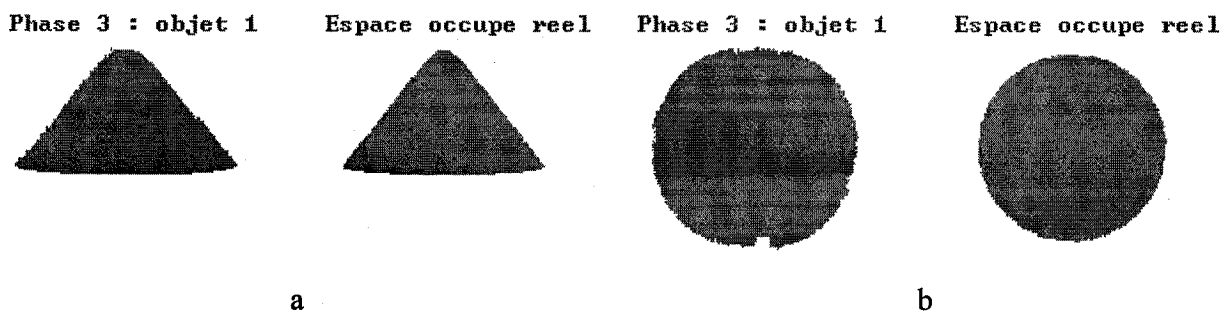


Figure 5-17 : vue de côté (a) et du haut (b) sur le segment *occupé* obtenu à la fin de la 3^{ème} phase (environnement de taille [320 x 320 x 64] incorporant un objet cône)

Phase 2 : objet 1

Espace occupe reel

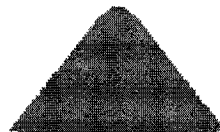
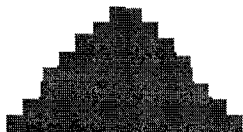


Figure 5-18 : vue de coté sur le segment *occupé* obtenu à la fin de la 2^{ème} phase (environnement de taille [320 x 320 x 64] incorporant un objet conique)

Phase 3 : segment libre

Espace occupe reel

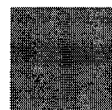
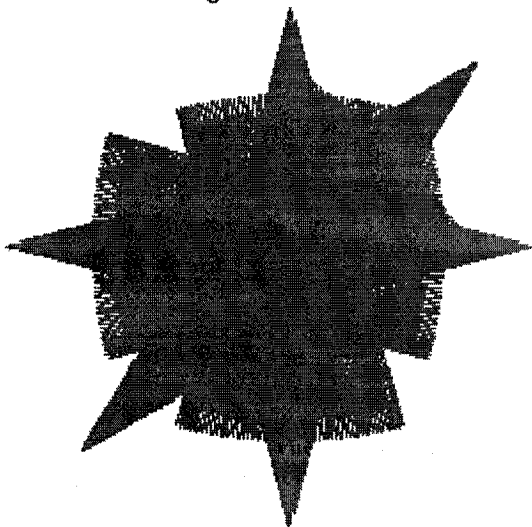


Figure 5-19 : vue d'en haut sur le segment *libre* obtenu à la fin de la 3^{ème} phase, l'emplacement occupé par l'objet cubique est invisible étant donné qu'il est entouré au dessus et au dessous de sept couches de cellules *libres* (environnement de taille [320 x 320 x 64] incorporant un objet cubique)

Phase 3 : objet 1

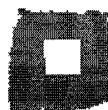
Espace occupe reel

Phase 3 : objet 1

Espace occupe reel



a



b

Figure 5-20 : vue de coté (a) et du haut (b) sur le segment *occupé* obtenu à la fin de la 3^{ème} phase (environnement de taille [320 x 320 x 64] incorporant un objet cubique)

Phase 2 : objet 1

Espace occupe reel

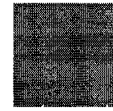
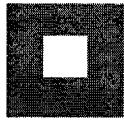


Figure 5-21 : vue d'en haut sur le segment *occupé* obtenu à la fin de la 2^{ème} phase (environnement de taille [320 x 320 x 64] incorporant un objet cubique)

Phase 3 : segment libre

Espace occupe reel

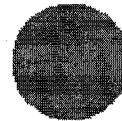
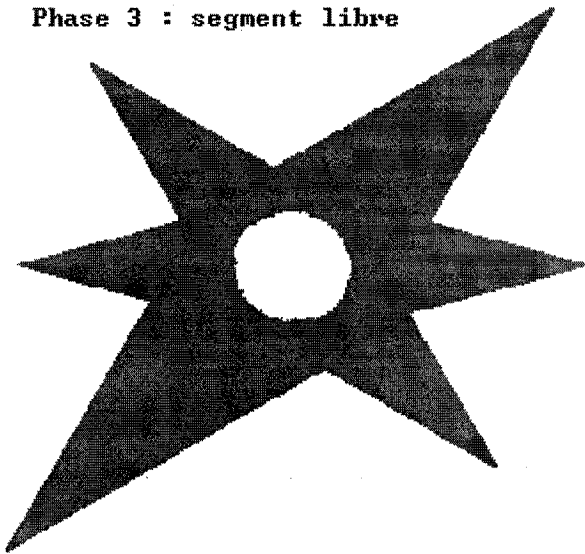


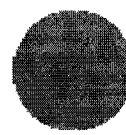
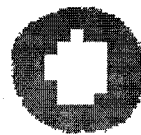
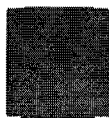
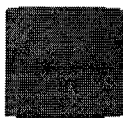
Figure 5-22 : vue d'en haut sur le segment *libre* obtenu à la fin de la 3^{ème} phase (environnement de taille [320 x 320 x 64] incorporant un objet cylindrique)

Phase 3 : objet 1

Espace occupe reel

Phase 3 : objet 1

Espace occupe reel



a

b

Figure 5-23 : vue de côté (a) et du haut (b) sur le segment *occupé* obtenu à la fin de la 3^{ème} phase (environnement de taille [320 x 320 x 64] incorporant un objet cylindrique)

Phase 2 : objet 1

Espace occupe reel

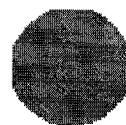
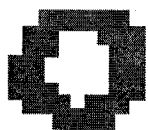


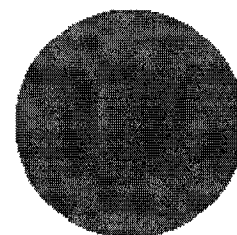
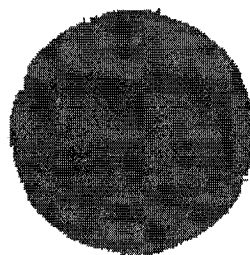
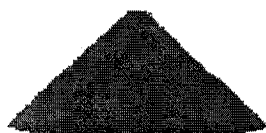
Figure 5-24 : vue d'en haut sur le segment *occupé* obtenu à la fin de la 2^{ème} phase (environnement de taille [320 x 320 x 64] incorporant un objet cylindrique)

Phase 3 : objet 1

Espace occupe reel

Phase 3 : objet 1

Espace occupe reel



a

b

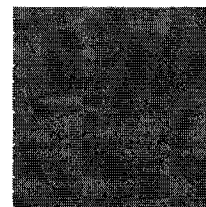
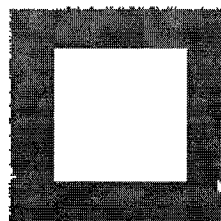
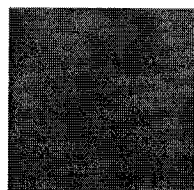
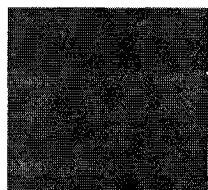
Figure 5-25 : vue de côté (a) et du haut (b) sur le segment *occupé* obtenu à la fin de la 3^{ème} phase (environnement de taille [320 x 320 x 128] incorporant un objet conique)

Phase 3 : objet 1

Espace occupe reel

Phase 3 : objet 1

Espace occupe reel



a

b

Figure 5-26 : vue de côté (a) et du haut (b) sur le segment *occupé* obtenu à la fin de la 3^{ème} phase (environnement de taille [320 x 320 x 128] incorporant un objet cubique)

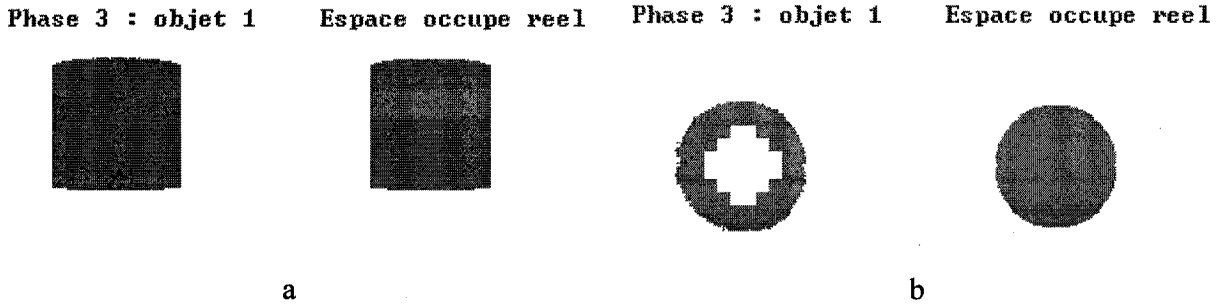


Figure 5-27 : vue de coté (a) et du haut (b) sur le segment *occupé* obtenu à la fin de la 3^{ème} phase (environnement de taille [320 x 320 x 128] incorporant un objet cylindrique)

D'un point de vue qualitatif, les résultats obtenus à l'aide de l'approche proposée sont convaincants et démontrent que l'algorithme de segmentation bidimensionnelle proposé est bel et bien extensible pour traiter le cas des environnements probabilistes 3D. Cette version évoluée de l'algorithme de segmentation peut servir de base stable pour une multitude d'applications nécessitant une interaction entre un robot fixe ou mobile et son environnement. L'extension en trois dimensions ouvre des possibilités particulièrement intéressantes pour l'utilisation avec des robots manipulateurs.

D'un point de vue quantitatif, l'algorithme de segmentation tridimensionnelle est relativement efficace en terme de temps d'exécution. Celui-ci demeure acceptable si l'application choisie ne nécessite pas des mises à jour continues de l'état de l'environnement et peut agir sur le dernier modèle en attendant la segmentation de l'environnement actuel. Bien que la segmentation d'un modèle de taille [320 x 320 x 128] requière aux alentours de 11 minutes, diverses optimisations demeurent possibles. En effet, une revue du code combinée à une implantation matérielle ou multiprocesseur peut améliorer considérablement la performance observée. De plus, le temps nécessaire à la segmentation est minime comparé à celui de la génération des modèles probabilistes tridimensionnels (récolte, traitement et combinaison des données de profondeur issues de différents capteurs et points de vue) et demeure à notre avis très acceptable compte tenu des résultats qu'il fournit.

5.5 Conclusion

Nous avons proposé dans ce chapitre une version de l'algorithme spécialisé dans la segmentation des environnements probabilistes 3D. Cette dernière constitue une extension de l'algorithme bidimensionnel exposé dans la partie précédente. En variant les paramètres intrinsèques

à notre schéma, soient la taille minimale qu'une subdivision peut atteindre et le nombre de niveaux de discrétisation du contraste, nous avons validé expérimentalement l'algorithme sur six environnements probabilistes dotés de caractéristiques différentes.

La limitation principale que nous avons rencontrée provient de la charge de calcul imposée par la dimension supplémentaire. Nous avons tenté avec succès de limiter l'augmentation du temps d'exécution en redéfinissant l'unité de texture. Étant donné la prise en compte de 6 cellules avoisinantes parmi les 26 contenues dans une unité de texture cubique, l'algorithme proposé fournit des résultats très satisfaisants.

La conception de cette méthode contribue au domaine de la segmentation tridimensionnelle qui est très peu explorée dans la littérature. Ce travail ouvre la voie à un grand nombre d'applications basées sur les grilles d'occupations.

Chapitre 6 Extensions et applications possibles

6.1 Introduction

Ce chapitre présente les extensions ainsi que quelques applications possibles de l'algorithme de segmentation LBP/C révisé. Il est divisé en quatre sections. Les deux premières décrivent respectivement la segmentation par paliers de PMO et l'identification dans une image probabiliste des régions dotées d'une PMO dans un intervalle donné. La section 6.4 présente et analyse l'applicabilité de l'algorithme de planification de trajectoire A^* [67] sur les images probabilistes segmentées. En dernier lieu, nous exposons quelques résultats issus de l'application de l'algorithme proposé sur des images aériennes et biomédicales.

6.2 Segmentation par paliers

Dans le contexte de la planification de trajectoires exemptes de collisions, l'algorithme de segmentation LBP/C révisé que nous avons présenté identifie les régions dont la probabilité moyenne d'occupation, PMO , est bornée dans trois intervalles strictes, chacun de ces derniers correspondant à une occupation différente soient *occupée*, *inconnue* et *libre*. Afin d'évaluer la généralité de l'approche proposée, nous avons reformulé la phase de création des segments de telle sorte qu'elle fusionne les subdivisions adjacentes suivant des intervalles de PMO distribués par paliers, chacun correspondant à une plage d'intensités distinctes. Ainsi, si la taille du palier choisie est P , le premier intervalle de subdivision regroupe les régions adjacentes dont la PMO est bornée dans l'intervalle $[0;P[$, la PMO du second dans $[P;2P[$, du troisième dans $[2P;3P[$, et ainsi de suite jusqu'à ce que la borne supérieure d'un intervalle atteigne 1. Le choix de la valeur supérieure provient du fait que les valeurs possibles de la PMO appartiennent à l'intervalle $[0;1]$. Les valeurs de nP sont donc normalisées.

La Figure 6-1 et la Figure 6-2 illustrent le résultat de la segmentation par paliers sur deux images probabilistes avec une largeur de l'intervalle de mesure de 0.1 et 0.15 respectivement. La première image de ces figures correspond à l'image probabiliste (Figure 6-1-a et Figure 6-2-a), tandis que les deux autres montrent le résultat de la segmentation. L'intensité des segments dans les

troisièmes images (Figure 6-1-c et Figure 6-2-c) est reliée à la valeur de *PMO* qui les caractérise. En effet, plus la *PMO* est élevée, plus le niveau de gris dans la région est pâle. Nous remarquons que pour un espace *libre*, la *PMO* devient de plus en plus élevée plus nous nous éloignons du capteur (les segments sont de plus en plus clairs). Ceci est dû au fait qu'avec la distance l'occupation de l'espace est de moins en moins certaine.

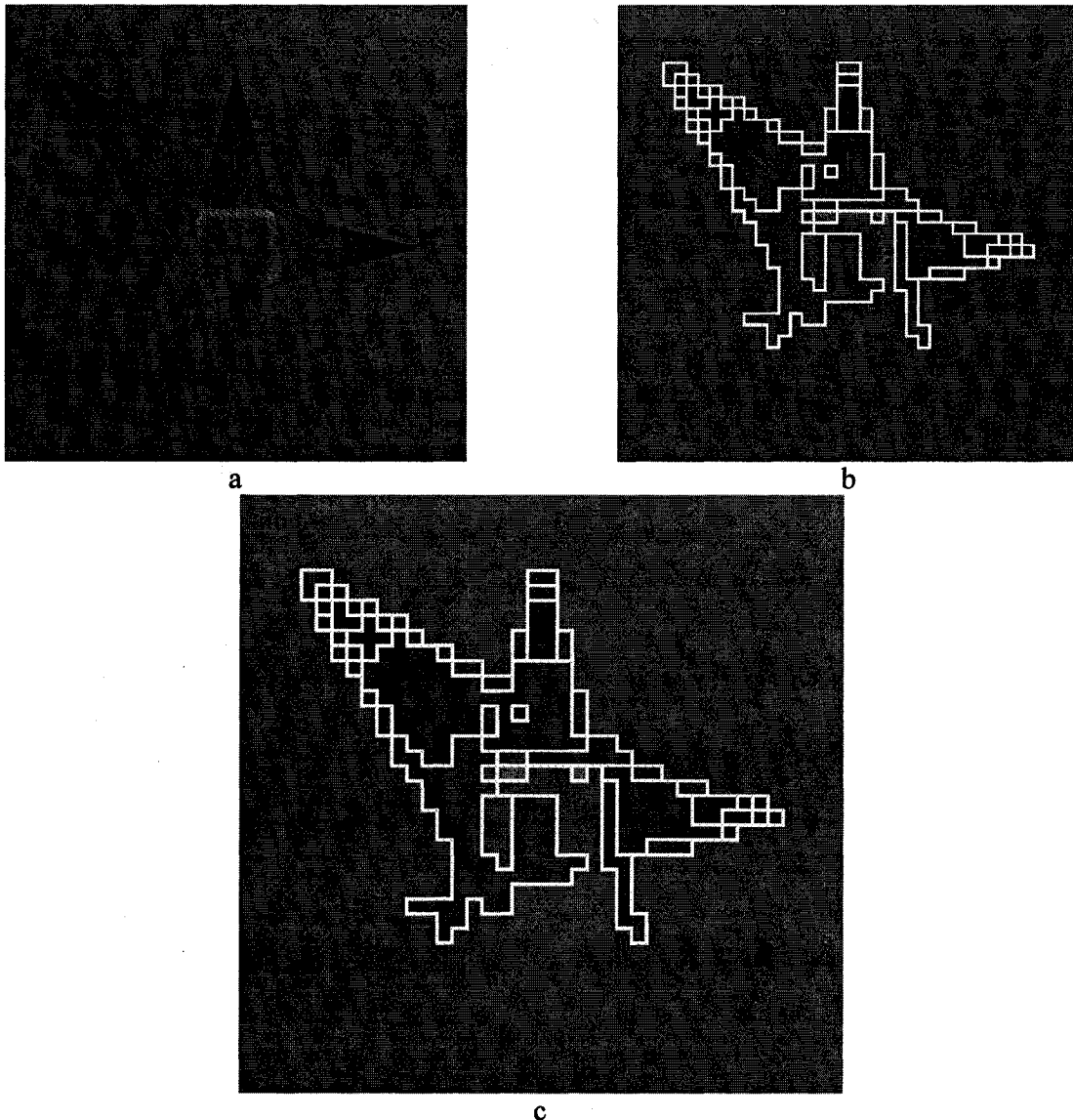


Figure 6-1 : segmentation par paliers avec un intervalle de *PMO* de largeur 0.1

En comparant la Figure 6-1-c et la Figure 6-2-c, nous remarquons que plus grande est la taille des paliers, moins nombreux seront les segments générés. Cette observation est principalement due à la réduction du nombre d'intervalles de *PMOs* et par conséquent de segments possibles.

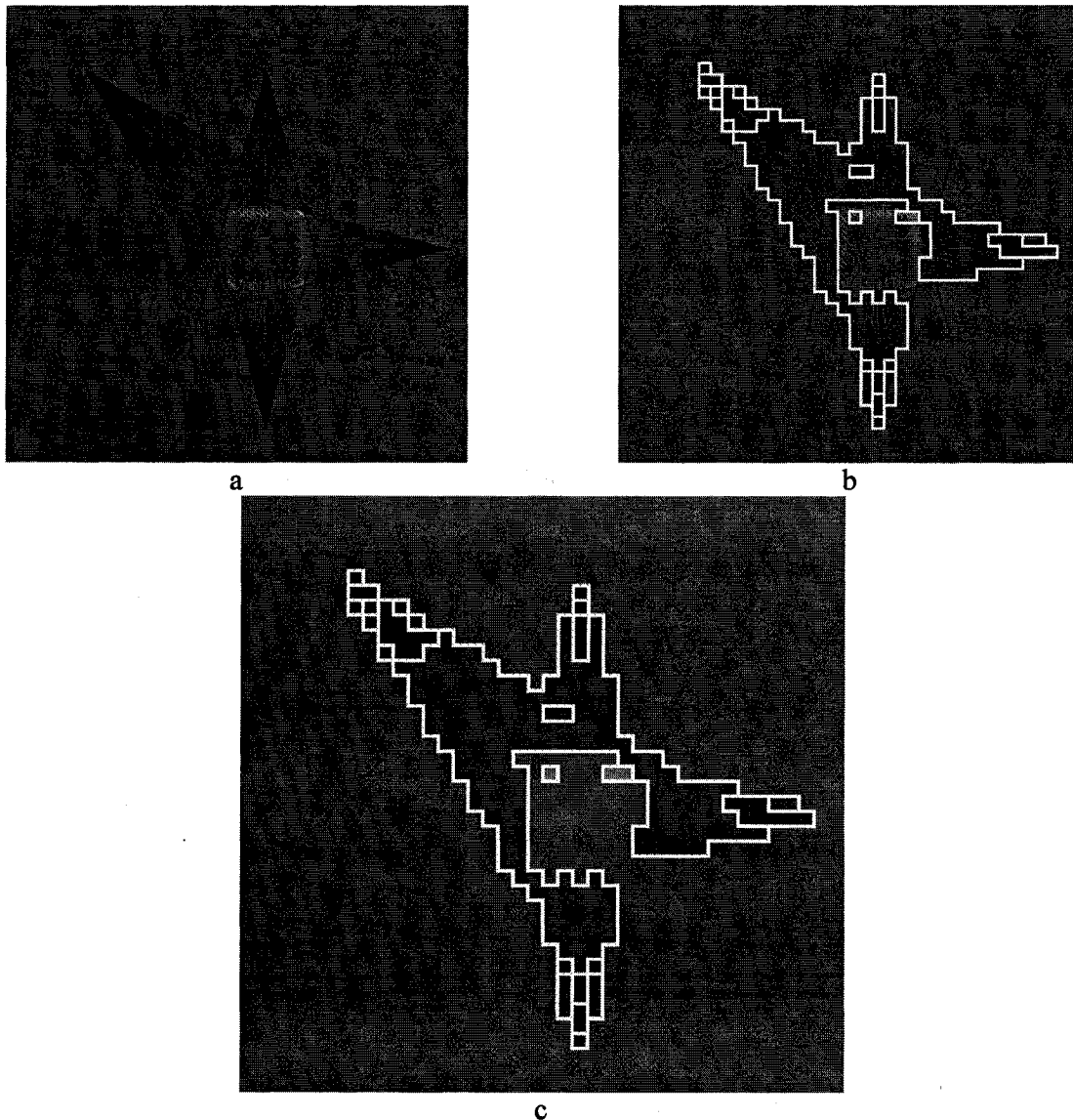


Figure 6-2 : segmentation par paliers avec un intervalle de *PMO* de largeur 0.15

6.3 Extraction des régions selon leur probabilité d'occupation

Cette extension diffère de celle présentée dans la section précédente par sa capacité à combiner dans différents segments les régions adjacentes dotées d'une *PMO* dans un intervalle choisi par l'utilisateur. En plus d'être pertinente pour la segmentation d'images à niveaux de gris génériques, cette extension est particulièrement utile dans des applications de planification de trajectoire et d'interaction avec l'environnement dans lesquels les mouvements d'un robot peuvent être limités à certaines régions dotées d'une occupation précise.

La Figure 6-3 et la Figure 6-4 présentent pour deux images probabilistes (Figure 6-3-a et Figure 6-4-a) et deux intervalles d'extractions, respectivement $[0;0.35]$ et $[0.52;0.7]$, les résultats de notre implantation. La deuxième et troisième images dans les figures ci-dessous présentent le même résultat. L'espace non inclus dans l'intervalle choisi est illustré en noir, tandis que l'espace identifié est caractérisé par un niveau de gris intermédiaire (Figure 6-3-c et Figure 6-4-c).

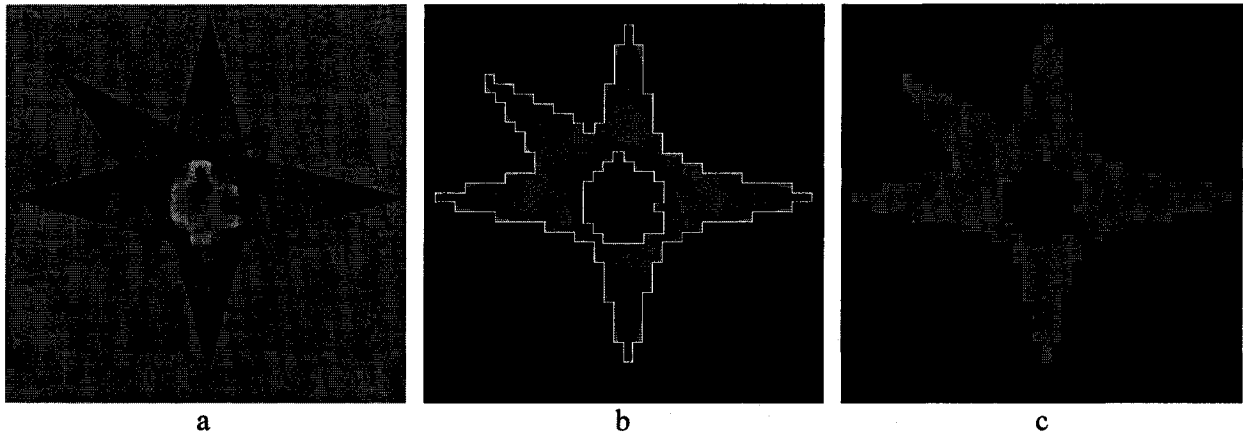


Figure 6-3 : extractions des subdivisions dont la *PMO* est bornée dans l'intervalle $[0;0.35]$

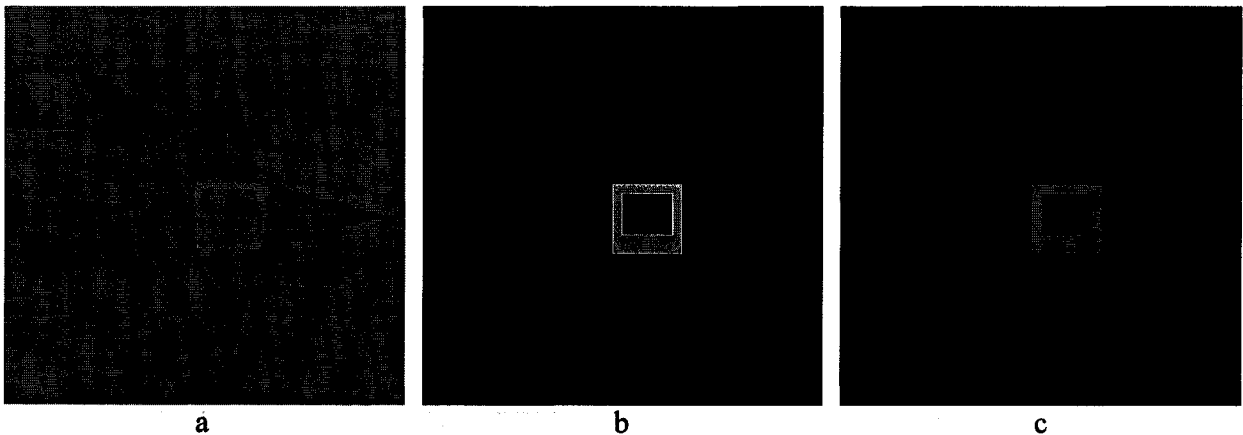


Figure 6-4 : extraction des subdivisions dont la *PMO* est bornée dans l'intervalle $[0.52;0.7]$

Finalement, il est important de noter que si deux régions non adjacentes sont caractérisées par une *PMO* dans l'intervalle choisi, chacune d'entre elles sera classée automatiquement dans un segment distinct. Cette fonctionnalité permettra à des applications telles la planification de la trajectoire de reconnaître facilement si un chemin existe entre une source et une destination, chacune étant située dans une région *libre* non adjacente.

6.4 Planification de trajectoire appliquée sur le LBP/C révisé

Dans cette section, nous présentons l'application d'une approche de planification de trajectoire classique, l'heuristique A^* [67], sur les résultats de la segmentation obtenus à l'aide du LBP/C révisé [51] afin d'évaluer le potentiel de la technique dans ce contexte spécifique. Dans un premier temps, nous développons l'heuristique A^* , par la suite nous exposons et nous analysons brièvement les résultats de l'application.

Comme défini par Latombe [67], l'heuristique A^* s'apparente aux approches dites du meilleur en premier¹⁷ et de la largeur en premier¹⁸, en ce sens qu'elle examine les cellules voisines afin de faire une sélection en optimisant un critère donné. Souvent, et comme choisi dans notre implantation, le critère considéré est celui de la distance euclidienne entre la position actuelle du robot mobile et celle de la cible. Comme l'objectif est toujours d'éviter les collisions entre le mobile et les obstacles, l'heuristique A^* offre la possibilité de suspendre temporairement la minimisation du critère afin de permettre au robot de contourner les régions *occupées* ou *inconnues*.

Ainsi, il est clair qu'un robot mobile ne peut se déplacer que dans les espaces étiquetés comme *libres*. Afin d'opérer une planification sûre et sécuritaire, le planificateur de trajectoire peut directement tirer avantage de l'extraction de régions selon leur occupation, tel que présenté à la section 6.3. Dans notre implantation, nous considérons que la région sécuritaire de déplacement est celle dont la *PMO* est bornée dans l'intervalle $[0;0.45]$. Grâce à la souplesse de l'implantation réalisée, l'algorithme est capable, avant de procéder à l'identification de la trajectoire, d'analyser si cette dernière existe. Trois situations peuvent alors être traitées :

- ❖ L'origine et/ou la destination, ne tombe(nt) pas dans la région sécuritaire;
- ❖ L'origine et la destination appartiennent chacune à des segments sécuritaires non connectés;
- ❖ Autrement, une trajectoire sécuritaire existe et l'heuristique A^* procède à son identification.

La Figure 6-5, la Figure 6-6, la Figure 6-7 et la Figure 6-8 illustrent l'application de l'algorithme de planification de trajectoire A^* sur le résultat de la segmentation de quatre images probabilistes. Dans chacune des images résultantes (Figure 6-5-b, Figure 6-6-b, Figure 6-7-b et

¹⁷ « Best-first search ».

¹⁸ « Breadth-first search ».

Figure 6-8-b), l'espace de déplacement sécuritaire qui est caractérisé par une *PMO* inférieure à 0.45 est représenté en noir, le reste de l'environnement (*occupé* et *inconnu*) est doté d'un niveau gris intermédiaire, et la trajectoire suivie par le robot mobile est de couleur blanche. L'encombrement de l'espace ainsi que les points de départ et de destination du robot varient afin d'engendrer des trajectoires de nature variée (contournement, suivi de bordure, zigzag autour de plusieurs objets). Les mesures d'erreur de type 2 négligeables que l'algorithme de segmentation permet d'obtenir (inférieures à 0.25%) impliquent que le débordement des segments *libres* sur les régions occupées de l'environnement est minime. Ainsi, la planification de trajectoire basée les grilles segmentées assure que le déplacement du robot mobile est sécuritaire et se limite aux régions libres de l'environnement. La proximité apparente qui demeure entre la trajectoire et les obstacles s'explique par la nature même de l'algorithme de planification sélectionné, A^* , qui n'optimise pas la marge de sécurité autour des obstacles. Ces résultats démontrent l'efficacité et le potentiel du schéma de segmentation proposé dans la résolution de problèmes robotiques classiques.

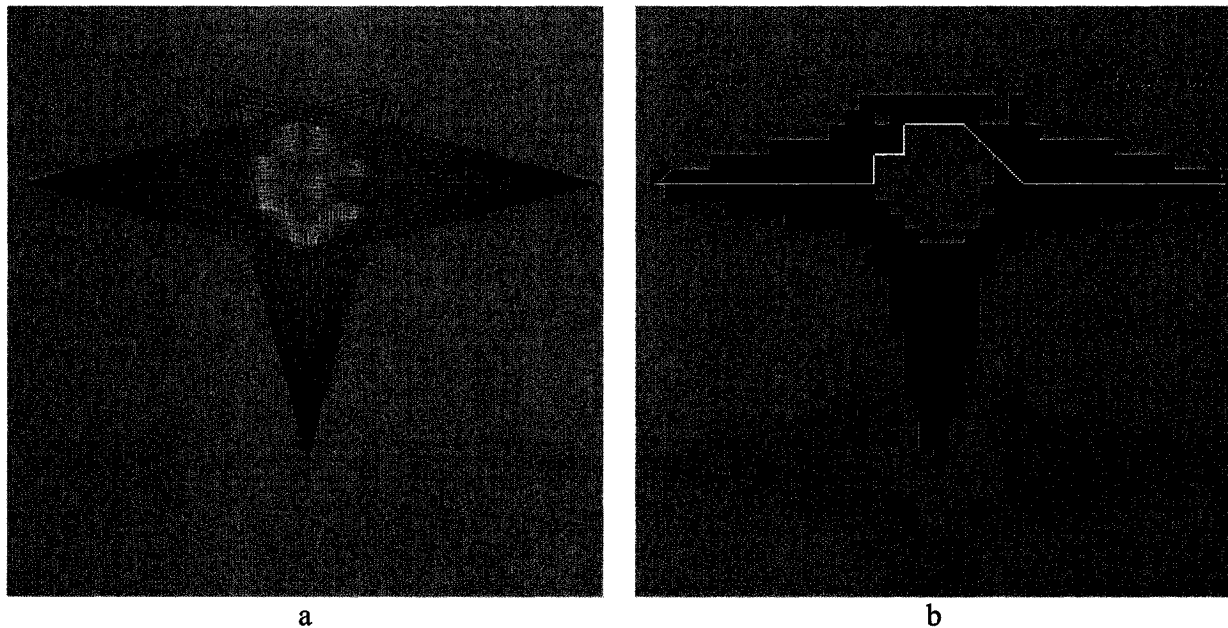
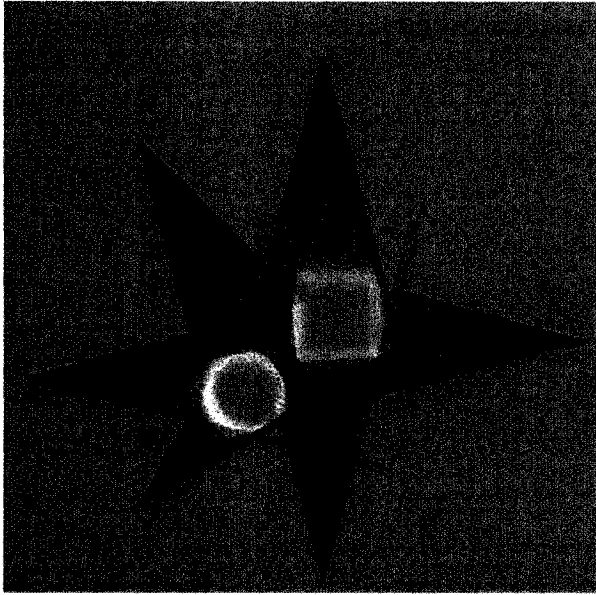
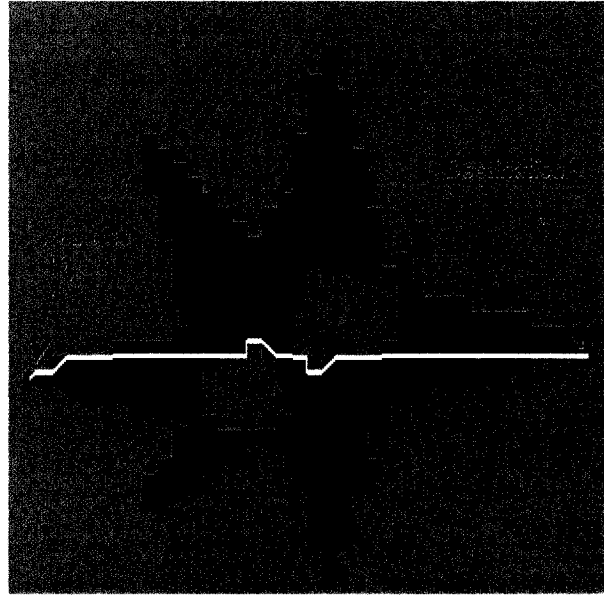


Figure 6-5 : planification de trajectoire avec besoin de contourner un obstacle ayant une forme complexe

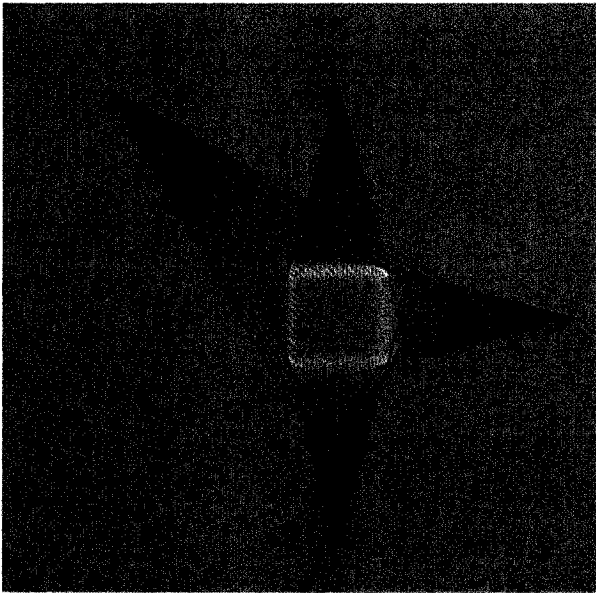


a

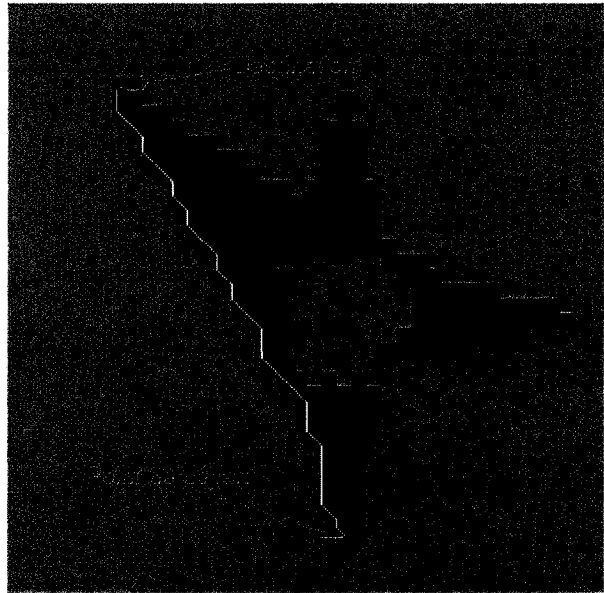


b

Figure 6-6 : planification de trajectoire avec besoin de contourner deux obstacles, un rectangulaire et un autre circulaire



a



b

Figure 6-7 : planification de trajectoire longeant la bordure de l'espace de déplacement sécuritaire

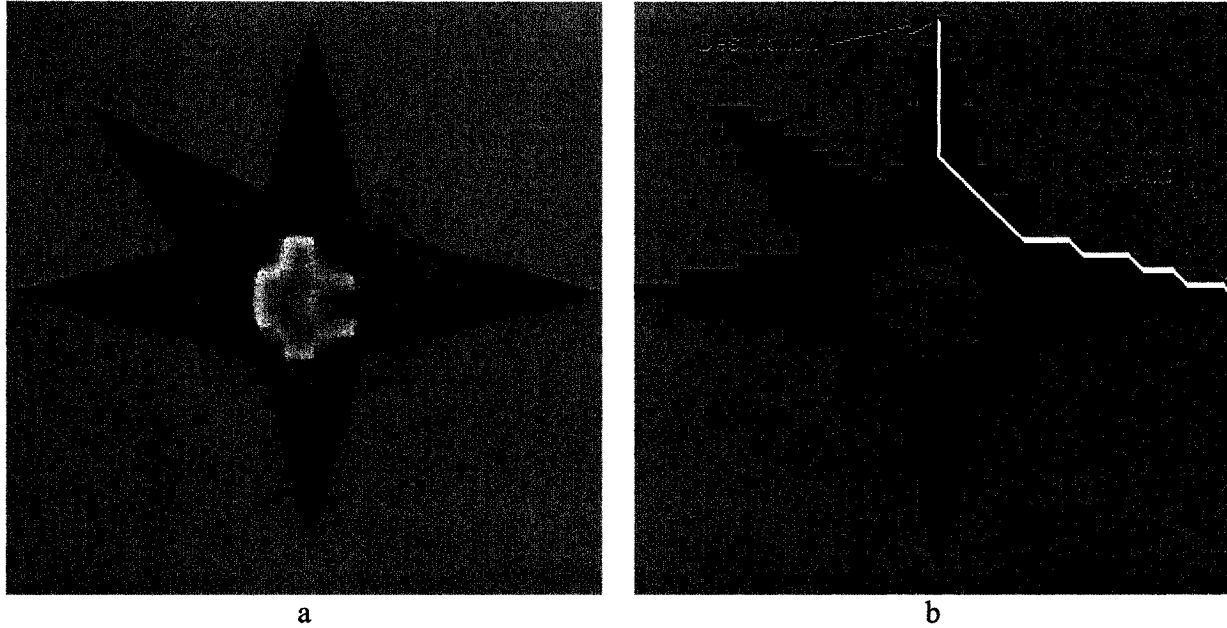


Figure 6-8 : autre cas de planification de trajectoire longeant la bordure de l'espace de déplacement sécuritaire

6.5 Autres applications

Dans le Chapitre 4, nous avons exposé et validé les résultats de l'algorithme proposé sur les images probabilistes bidimensionnelles obtenues à l'aide du simulateur d'environnement probabilistes développé par Bolzon (Annexe A). Dans la présente section, nous exposons les résultats de cette approche sur une variété d'images biomédicales et aériennes afin d'estimer le potentiel de l'approche dans un contexte moins spécifique.

La Figure 6-9 à la Figure 6-14 illustrent l'application de l'algorithme de segmentation bidimensionnelle sur une série d'images biomédicales (à rayons X et MRI) extraites de la banque d'images médicales MedPix [64]. Les résultats obtenus démontrent que l'algorithme est capable, sans aucun changement, de reconnaître en segments différents les tissus corporels rigides (partiellement pénétrés par les rayons X) ou actifs (ayant absorbé des composantes radioactives) non adjacents. Cette fonctionnalité ouvre la voie à des applications dans le domaine biomédical du fait qu'elle permet de sélectionner automatiquement les sections d'intérêt malgré la grande variation des textures dans les zones de mesure des images MRI ou rayons X originales. Les images biomédicales pourraient aussi être traitées à l'aide des extensions que nous avons proposées dans les sections 6.2 et

6.3. Celles-ci permettraient d'identifier les parties des organes selon des niveaux de réactivité préétablis.

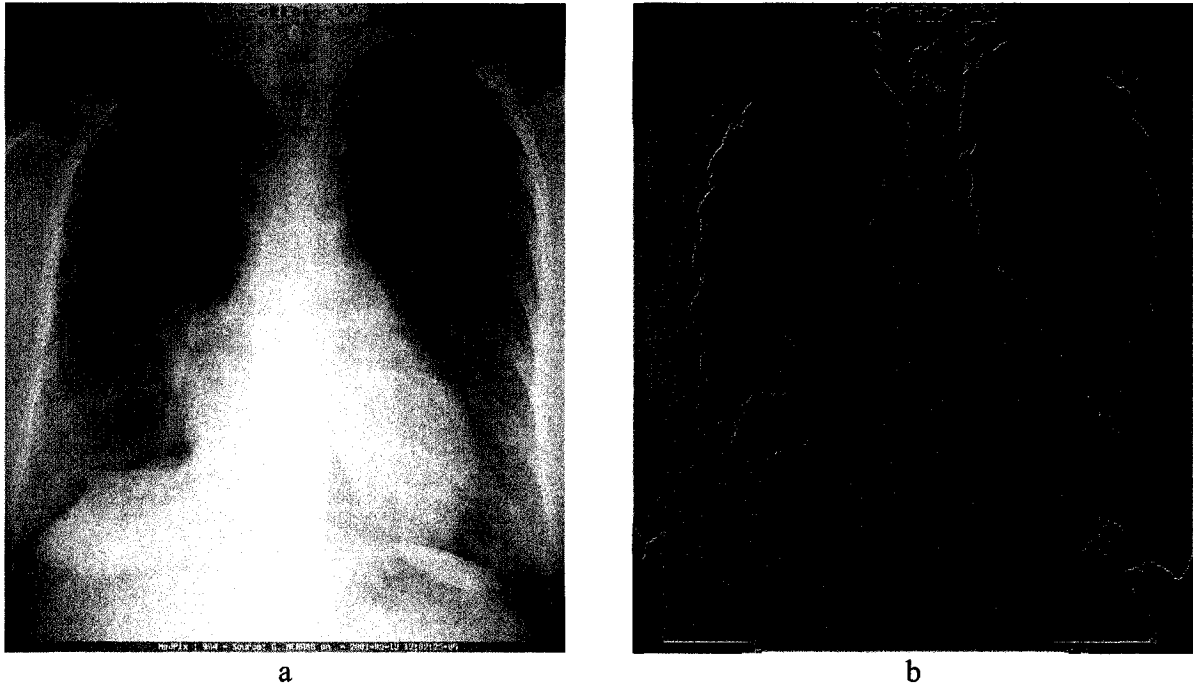
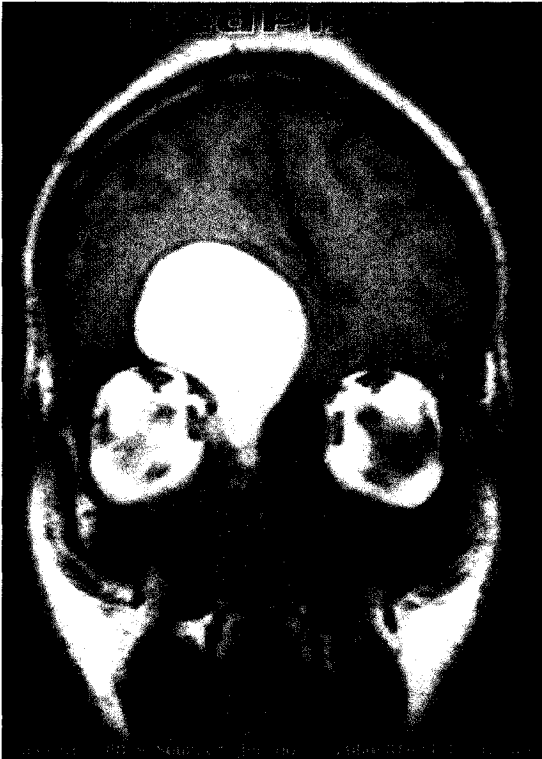


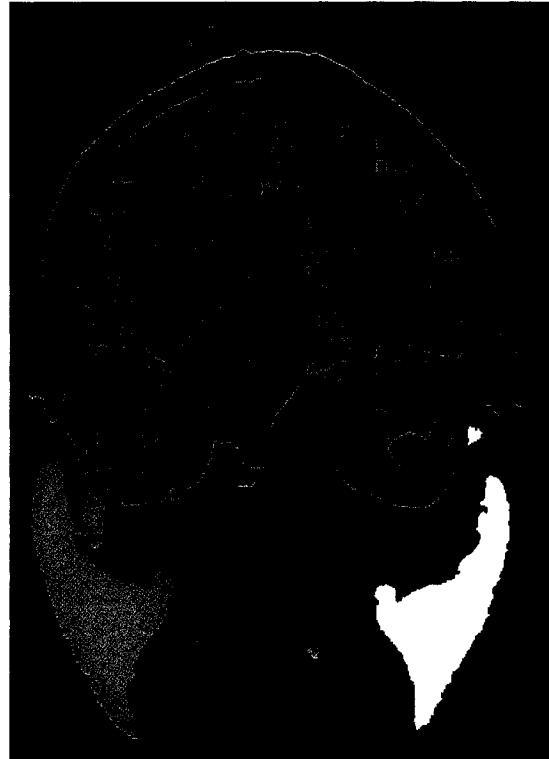
Figure 6-9 : (a) image rayons X de la cage thoracique d'un patient [64], (b) les tissus rigides forment un segment unique (bleu)



Figure 6-10 : (a) MRI de la tête et du cerveau d'un patient [64], (b) résultat de la segmentation où les tissus à forte résonance sont représentés par des couleurs différentes selon leur distribution spatiale

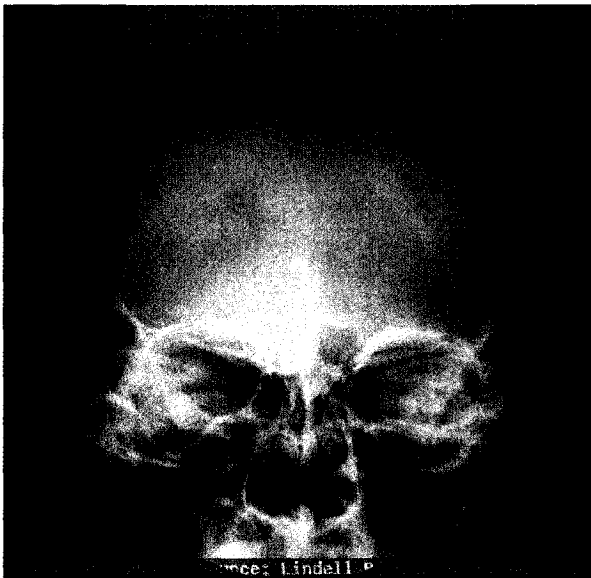


a

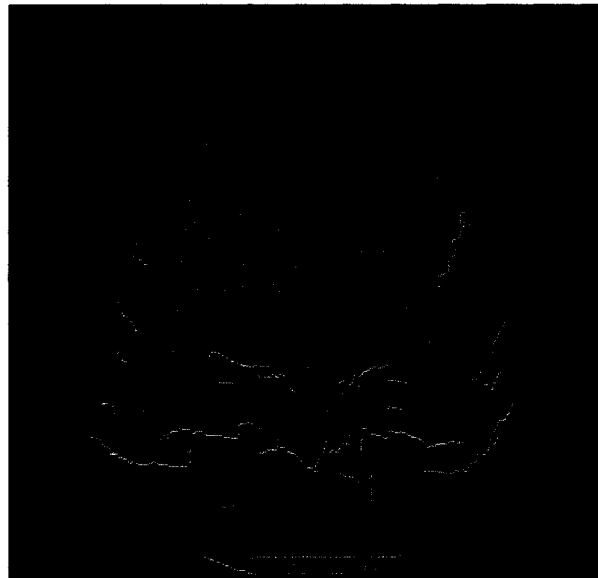


b

Figure 6-11 : (a) « Coronal T1W MRI » de la face d'un patient [64], (b) résultat de la segmentation où les tissus à forte résonance sont représentés par des couleurs différentes selon leur distribution spatiale

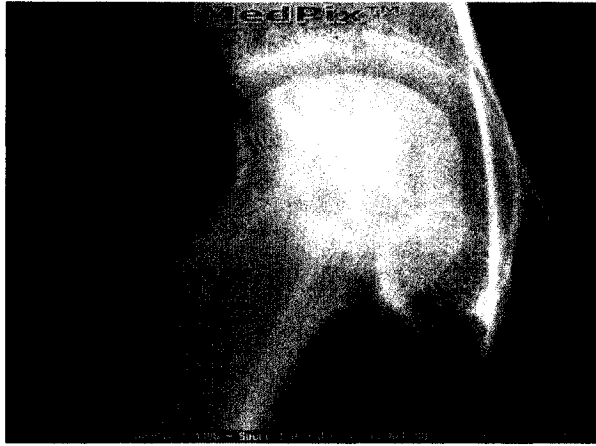


a

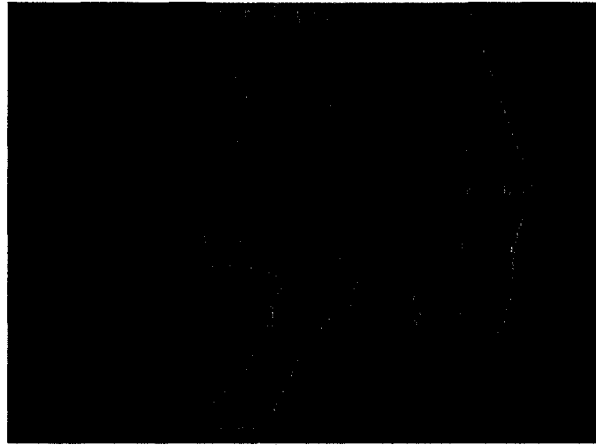


b

Figure 6-12 : (a) image rayons X de la face d'un patient prise par L.R. Gentry [64], (b) la section du crâne qui n'est pas traversée par les rayons est identifiée comme un segment unique (bleu)



a

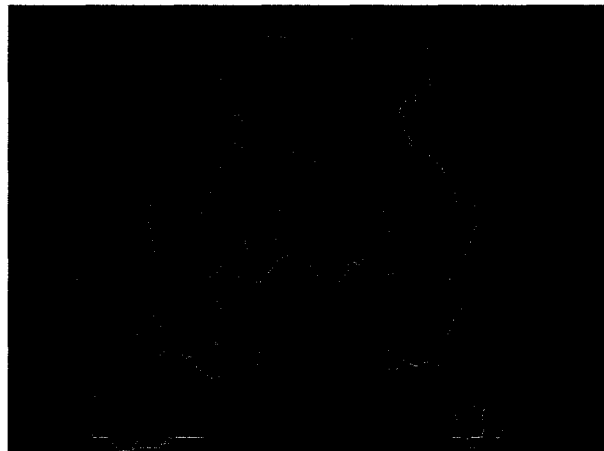


b

Figure 6-13 : (a) image rayons X de la hanche de droite prise par D. Lopresti [64], (b) la section des os qui ne sont pas traversés par les rayons est identifiée comme un segment unique (bleu)



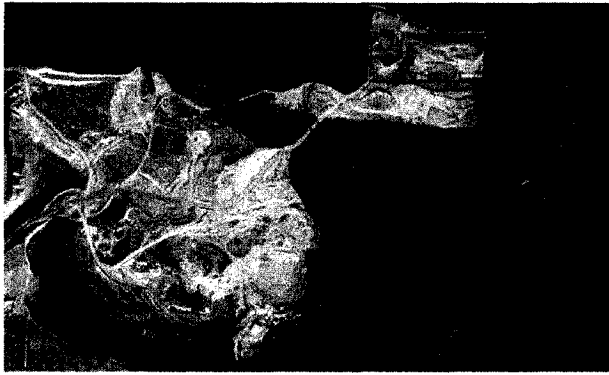
a



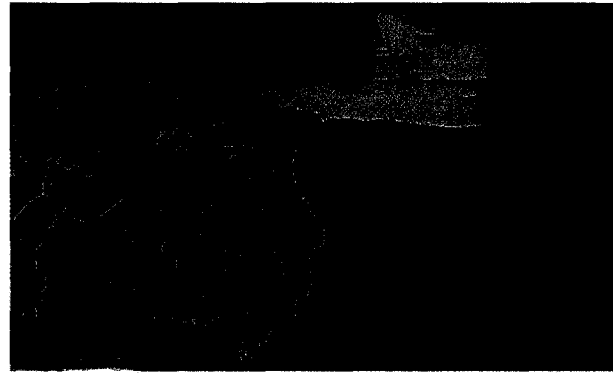
b

Figure 6-14 : (a) image rayons X des hanches prise par M. Akin [64], (b) les tissus rigides qui ne sont pas traversés par les rayons sont identifiés par des segments différents

La Figure 6-15 à la Figure 6-20 illustrent l'application de l'algorithme de segmentation sur des images aériennes et satellites extraites de Navteq [65]. Ici aussi, l'algorithme fournit des résultats convaincants quant à l'identification des différentes régions dans les images testées sans recourir à des modifications de ses paramètres. En effet, il est capable d'identifier avec précision les régions clef quand la prise de vue est éloignée de la terre (la carrière dans la Figure 6-15, la terre ferme dans la Figure 6-17 ainsi que la Figure 6-18, et la région peuplée dans la Figure 6-19). Pour les images dotées d'un degré de détails élevé, l'algorithme fournit des résultats convenables mais qui ne reproduisent pas toutes les particularités de l'espace (Figure 6-16 et Figure 6-20), tels les éléments de texture couvrant des sections trop restreintes.

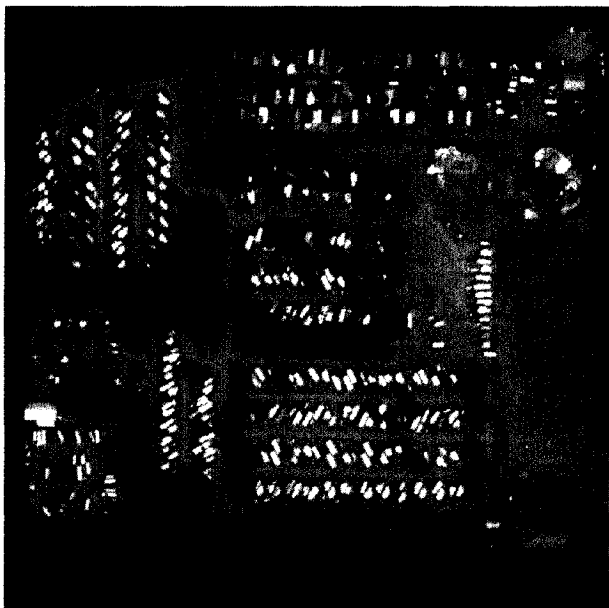


a

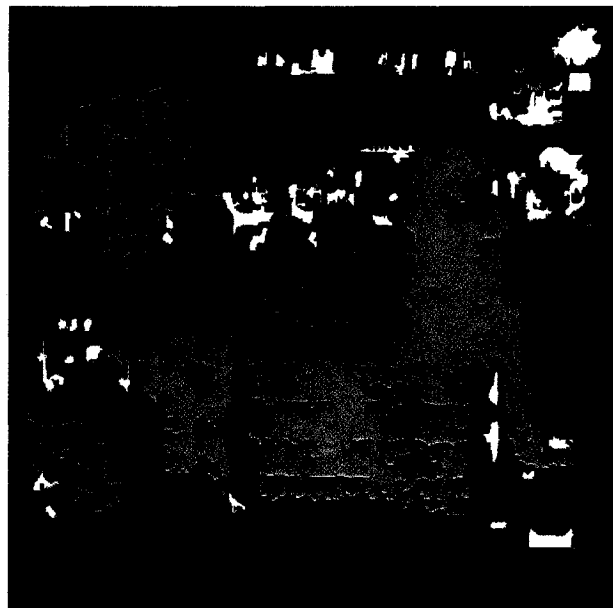


b

Figure 6-15 : (a) image aérienne d'une carrière proche de la ville de Vancouver au Canada [65], (b) les espaces où sont extraits les minerais sont identifiés dans des segments séparés (vert, bleu et orange) selon leur distribution spatiale

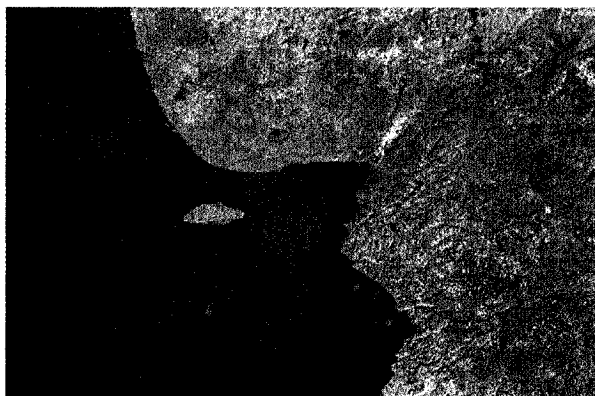


a

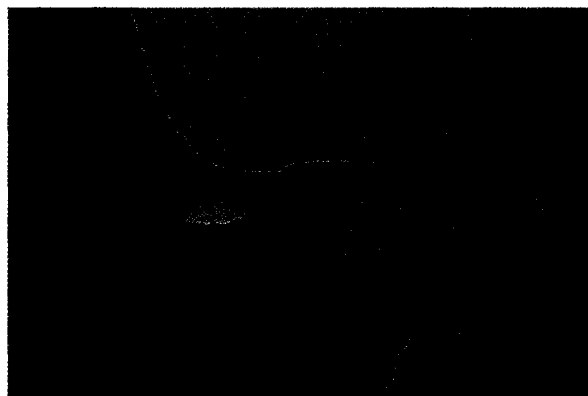


b

Figure 6-16 : (a) image aérienne d'un terrain de camping dans la ville de Miami aux États-Unis [65], (b) les routes et les espaces où nous retrouvons une forte densité de motorisés sont identifiés dans des segments séparés (bleu, vert, blanc, marron, jaune et mauve) selon leur proximité

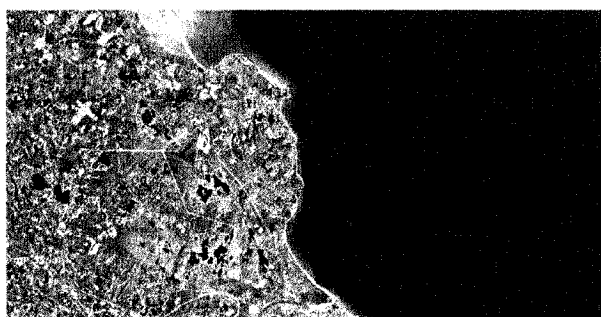


a

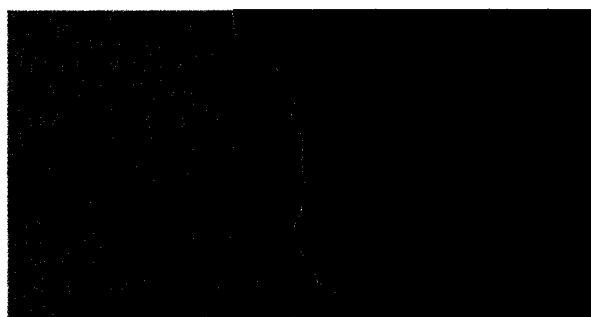


b

Figure 6-17 : (a) image satellite d'une région sur le bord du lac Ontario [65], (b) l'eau (noir), la terre ferme (bleu) et l'île (vert) sont identifiés en trois segments séparés

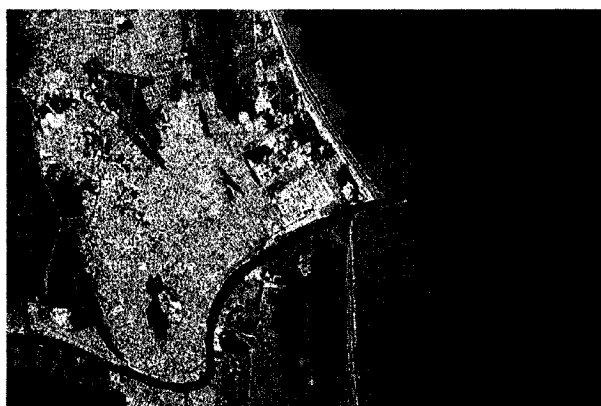


a



b

Figure 6-18 : (a) image satellite de la ville de Boston aux États-Unis [65], (b) l'eau (noir) et la terre ferme (bleu) sont identifiés en deux segments séparés



a



b

Figure 6-19 : (a) image satellite d'une ville sur la côte Est des États-Unis [65], (b) les régions densément peuplées sont mises en évidence

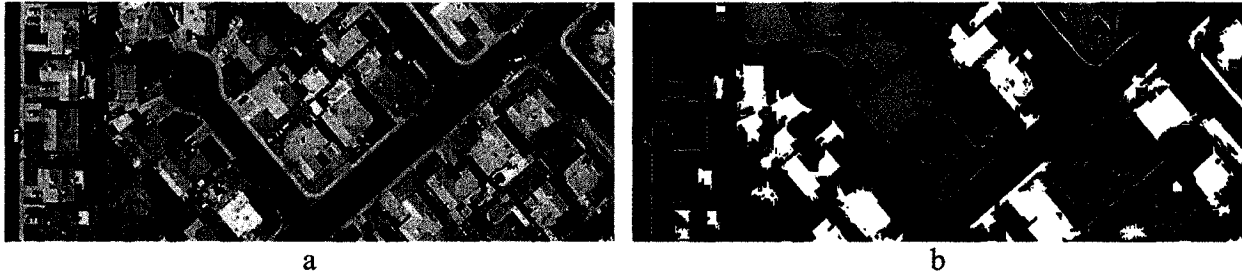


Figure 6-20 : (a) image aérienne d'un quartier résidentiel de la ville de Las Vegas aux États-Unis [65], (b) les maisons (bleu, vert, blanc, orange, mauve, marron), les routes ainsi que les espaces libres (noir) sont identifiés dans des segments séparés

Pour conclure cette section, nous exposons dans la Figure 6-21 le résultat de l'application de l'algorithme de segmentation sur une image artistique intitulée *Premiers Pas*. Il est intéressant d'observer comment la forme des segments s'adapte bien aux deux régions issues de la trace du pas dans la neige malgré la complexité des textures perçues dans la surface enneigée et dans la semelle.

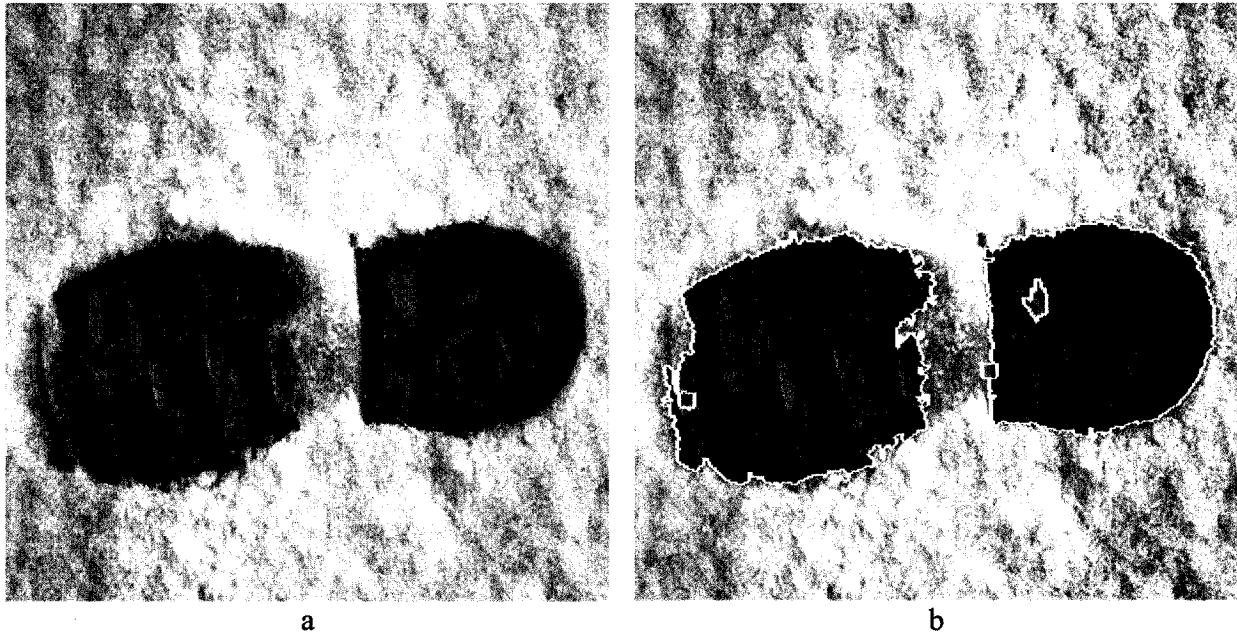


Figure 6-21 : (a) image *Premiers Pas* [66], (b) résultat de notre algorithme dans lequel les segments sont séparés par une ligne blanche

6.6 Conclusion

Dans ce chapitre, nous avons exposé différentes extensions et applications possibles de l'algorithme LBP/C révisé. Ainsi, la segmentation par paliers et l'identification de régions dotées

d'une occupation bornée dans un intervalle donné ouvrent la voie à un grand nombre d'applications en robotique autonome telle la planification de trajectoire exposée dans la section 6.4.

Finalement, nous avons démontré que l'utilité de l'algorithme proposé s'étend au-delà du traitement des grilles d'occupation probabilistes. Ainsi, sans aucune adaptation, le schéma révisé fournit des résultats intéressants pour la segmentation d'images aériennes et médicales (rayons X et MRI) ainsi que sur des images d'intensité classiques comportant des textures complexes.

Chapitre 7 Conclusion

Ce dernier chapitre est divisé en trois parties, la première résume le contenu de la thèse. La seconde met l'accent sur les contributions apportées. La dernière partie présente les travaux connexes qui peuvent compléter la recherche effectuée.

7.1 Résumé

Ce travail ayant pour objectif de contribuer à établir un pont entre deux domaines relativement complexes, soient les grilles d'occupation probabilistes et la segmentation, les avantages ainsi que la physique de la construction des modèles probabilistes ont été élaborés dans le second chapitre. Aussi, un survol des principaux travaux qui ont influencé et contribué au développement des grilles d'occupation est présenté. La seconde section analyse les particularités des textures présentes dans les images probabilistes et la dernière section est dédiée à la littérature de la segmentation. Elle revoit et analyse l'applicabilité sur les modèles probabilistes de plusieurs algorithmes.

Suite à cette évaluation des méthodes classiques, l'algorithme développé par Ojala et Pietikäinen [39] a révélé un potentiel élevé pour résoudre le problème de la segmentation des grilles d'occupation probabilistes. Malgré les résultats qualitatifs et quantitatifs inadéquats que l'approche originale fournit sur des images probabilistes, l'originalité du concept proposé nous a poussés à l'étendre, l'adapter et le rendre plus performant. Le travail que nous avons accompli en terme de segmentation bidimensionnelle est exposé dans le troisième chapitre, incluant les diverses tentatives de raffinement que nous avons effectuées directement sur l'algorithme LBP/C original ainsi qu'une description détaillée de l'algorithme LBP/C révisé.

Le quatrième chapitre est dédié à l'analyse des résultats obtenus en segmentation bidimensionnelle. Une étude comparative de l'algorithme proposé avec celui d'Ojala et Pietikäinen est proposée, celle-ci met l'accent sur les forces et faiblesses de chacune des méthodes. Par la suite, afin de juger objectivement les résultats de notre méthode, deux mesures d'erreur complémentaires sont définies. Celles-ci permettent de valider la stabilité et la précision du schéma de segmentation dans différents contextes. Une étude détaillée des effets de la variation de nombreux paramètres sur les résultats est présentée.

Afin d'étendre la technique à des applications dans le monde réel où la troisième dimension n'est pas négligeable, l'algorithme est étendu au chapitre cinq au cas des environnements probabilistes tridimensionnels. Le mode construction des environnements probabilistes tridimensionnels employés est décrit et l'unité de texture tridimensionnelle est redéfinie ainsi que les trois phases de traitement de la segmentation en fonction de la dimension supplémentaire intégrée aux modèles. La dernière partie expose et analyse les résultats de la segmentation. Les valeurs des mesures d'erreurs obtenues démontrent la validité de l'extension en trois dimensions sur le plan qualitatif et quantitatif. Bien que le temps d'exécution ne soit pas aussi intéressant que dans le cas bidimensionnel, ce dernier demeure acceptable compte tenu de la complexité et de la taille des modèles traités. De même, des solutions potentielles sont proposées pour en améliorer davantage l'efficacité.

Le sixième chapitre propose et analyse diverses extensions possibles au schéma de segmentation. Celles-ci permettent entre autre de segmenter un environnement suivant des paliers d'occupation variables et d'extraire les régions dotées d'un état d'occupation donné. En se basant sur ces variations, nous avons appliqué avec succès l'algorithme de planification de trajectoire A^* sur les résultats de la segmentation bidimensionnelle. La dernière section démontre aussi l'applicabilité de l'algorithme au-delà des grilles probabilistes en exposant les résultats de la segmentation sur plusieurs images biomédicales et aériennes au contenu très varié et complexe.

7.2 Contributions

Cette thèse propose un schéma de segmentation adapté aux particularités des environnements probabilistes bidimensionnels et tridimensionnels par l'examen de deux domaines relativement complexes et peu explorés dans la littérature, soient la segmentation basée sur l'analyse de la texture et le traitement des grilles d'occupation. Les principales contributions de ce travail sont :

- ❖ Une application des méthodes de segmentation modernes sur les grilles d'occupation probabilistes et l'analyse de leur performance;
- ❖ Un raffinement de l'algorithme de segmentation par analyse des textures d'Ojala et Pietikäinen qui conduit à une approche non-supervisée ne requérant aucune connaissance préalable de l'environnement ni d'intervention humaine. La technique est capable de

distinguer en segments uniques les objets qui ont été balayés d'une façon non-uniforme par des capteurs laser de profondeur à résolution limitée. Elle permet aussi de classifier les objets selon leur proximité spatiale;

- ❖ Une validation expérimentale quantitative de l'approche proposée pour le cas en deux dimensions où l'impact des variations des paramètres extrinsèques (écart-type de l'erreur gaussienne du capteur sur la distance et pas angulaire entre les faisceaux de lecture) et intrinsèques (nombre de niveaux de discrétisation du contraste et taille minimale qu'une subdivision peut atteindre) est analysé pour déterminer les réglages les plus stables;
- ❖ Une extension en trois dimensions de l'algorithme de segmentation proposé avec une définition optimisée de l'unité de texture qui limite l'augmentation du temps de calcul;
- ❖ Une exploration expérimentale qualitative du potentiel de l'algorithme LBP/C révisé pour la segmentation d'images complexes et de nature variée (MRI, rayons X, aériennes et artistiques), ce qui ouvre la voie à un champ d'application plus large.

Ces éléments permettent d'atteindre les objectifs de ce travail de recherche en proposant une base stable pour la segmentation de modèles probabilistes construits à partir de capteurs télémétriques représentatifs de l'état actuel de la technologie qui n'offrent qu'un échantillonnage de l'environnement et une incertitude non négligeable sur les mesures. Un tel outil ouvre la voie à de nombreuses applications en imagerie, en modélisation et en robotique autonome.

7.3 Travaux futurs

Bien que l'algorithme de segmentation proposé fournisse des résultats convaincants avec les images et les environnements probabilistes, plusieurs validations et améliorations demeurent possibles :

- ❖ La première concerne les images et environnements probabilistes sur lesquelles l'algorithme fut testé qui originaient tous d'un simulateur. Une validation de l'algorithme à partir d'images probabilistes générées à l'aide de capteurs réels, tels un sonar, un capteur laser de profondeur ou un système stéréoscopique permettrait d'approfondir encore davantage l'examen des performances;

- ❖ Un second raffinement possible pour optimiser l'encodage du modèle tridimensionnel et ainsi optimiser la performance de l'algorithme de segmentation, serait, tel que signalé dans la section 5.4, d'adapter l'algorithme proposé pour opérer sur les grilles d'occupation probabilistes à résolutions multiples;
- ❖ La troisième possibilité d'amélioration concerne la phase de raffinement, dans laquelle l'algorithme proposé applique indirectement un seuillage afin de positionner la bordure entre les segments *libres* et *occupés* le long de la transition lente émanant de l'introduction de l'erreur du capteur. Les seuils que nous avons choisis minimisent les deux mesures d'erreur, mais principalement celle de type 2. En d'autres termes, ils assurent que les objets réels soient contenus dans les segments *occupés* correspondants. Nous avons opté pour le choix le plus sécuritaire afin de garantir le succès de toutes les applications impliquant une certaine interaction entre un système autonome et son environnement. N'empêche qu'une étude plus poussée est nécessaire afin de trouver automatiquement pour chaque environnement traité, les seuils optimaux capables de minimiser les mesures d'erreur choisies;
- ❖ Finalement, la conception d'une extension à l'algorithme LBP/C révisé permettant de reconnaître les objets présents dans l'environnement à partir de la forme des segments *occupés* obtenus constitue un prolongement direct au travail présenté dans cette thèse.

Références

- [1] Elfes, A., “Using Occupancy Grids for Mobile Robot Perception and Navigation”, *Computer*, vol. 22, Issue: 6, pp. 46–57, June 1989.
- [2] Elfes, A., “A Sonar-based Mapping and Navigation System”, *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 1151–1156, April 1986.
- [3] Matthies, L., Elfes, A., “Integration of Sonar and Stereo Range Data Using a Grid-based Representation”, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 727–733, April 1988.
- [4] Elfes, A., *Occupancy Grids: A Probabilistic Framework for Spatial Robot Perception and Navigation*, PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.
- [5] Payeur, P., Hébert, P., Laurendeau, D., Gosselin, C.M., “Probabilistic Octree Modeling of a 3D Dynamic Environment”, *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1289–1296, April 1997.
- [6] Elmenreich, W., Schneider, L., Kirner, R., “A Robust Certainty Grid Algorithm for Robotic Vision”, *6th IEEE International Conference on Intelligent Engineering Systems*, pp. 25-30, Opatija, Croatia, May 2001.
- [7] Papoulis, A., “Probability, Random Variables and Stochastic Processes”, Fourth Edition, *Mc Graw Hill*, 2002.
- [8] Bolzon, B., Payeur, P., “Experimental Study of Data Merging Techniques for Workspace Modeling with Uncertainty”, *Proceedings of the IEEE International Workshop on Advanced Methods for Uncertainty Estimation in Measurement*, pp. 14-19, Niagara Falls, ON, 2005.
- [9] Payeur, P., *Machine Vision*, lectures notes for the course ELG 5163, School of Information Technology and Engineering, University of Ottawa, March 2005.
- [10] Servo-Robot Inc., *Jupiter: 3-D Laser Vision Camera, Installation and Operation Manual*, Servo-Robot Inc., 1996.
- [11] Canny, J., “A Computational Approach to Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, November 1986.

- [12] Burt, P.J., Hong, T., Rosenfeld, A., "Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, no. 12, pp. 802–809, December 1981.
- [13] Hong, T., Rosenfeld, A., Narayanan, K.A., Peleg, S., Silberberg, T., "Image Smoothing and Segmentation by Multiresolution Pixel Linking: Further Experiments and Extensions", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-12, no. 5, pp. 611–622, September 1982.
- [14] Ping, Y., Runsheng W., Diannong, L., "A New Image Segmentation Approach Based on Linked Pyramid", *3rd International Conference on Signal Processing 1996*, vol. 2, pp. 1118–1121, October 1996.
- [15] Arman, F., Pearce, J.A., "Unsupervised Classification of Cell Images Using Pyramid Node Linking", *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 6, pp. 647–650, June 1990.
- [16] Pavlidis, T., "Structural Pattern Recognition", *Springer-Verlag*, New York, 1977.
- [17] Wu, X., "Adaptive Split-and-Merge Segmentation Based on Piecewise Least-Square Approximation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, issue 8, pp. 808–815, August 1993.
- [18] Yang, H.S., Lee, S.U., "Split-and-Merge Segmentation Employing Thresholding Technique", *Proceedings of the 1997 International Conference on Image Processing*, vol. 1, pp. 239–242, October 1997.
- [19] Samet, H., "The Quadtree and Related Hierarchical Data Structures", *ACM Computing Surveys (CSUR)*, *ACM Press*, vol. 16, issue 2, June 1984.
- [20] Spann, M., Wilson, R., "A Quad-tree Approach to Image Segmentation which Combines Statistical and Spatial Information", *Pattern Recognition*, vol. 18, pp. 257–269, April 1985.
- [21] Gabor, D., "Theory of Communications", *Journal Inst. Electrical Engineering*, vol. 93, pp. 429–457, 1946.
- [22] Jain, A.K., Farrokhnia, F., "Unsupervised Texture Segmentation Using Gabor Filters", *Proceedings of the 1990 IEEE International Conference on Systems, Man and Cybernetics*, pp. 14–19, November 1990.

- [23] Mittal, N., Mital, D.P., Kap, L.C., “Features for Texture Segmentation Using Gabor Filters”, *Seventh International Conference on Image Processing and its Applications*, Volume 1, pp. 353–357, July 1999.
- [24] Mital, D.P., “Texture Segmentation using Gabor Filters”, *Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, vol. 1, pp. 109–112, September 2000.
- [25] Ji, Y., Chang, K.H., Hung, C.C., “Efficient Edge Detection and Object Segmentation Using Gabor Filters”, *Proceedings of the 42nd annual Southeast regional conference*, Huntsville, Alabama, pp. 454–459, April 2004.
- [26] Chang, T., Kuo, C.C.J., “Texture Analysis and Classification with Tree-Structured Wavelet Transform”, *IEEE Transactions on Image Processing*, vol. 2, pp. 429–441, October 1993.
- [27] Salari, E., Ling, Z., “Texture Segmentation Using Hierarchical Wavelet Decomposition”, *Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE '95*, vol. 1, pp. 216–220, July 1995.
- [28] Wang, B., Motomura, Y., Ono, A., “Texture Segmentation Algorithm Using Multichannel Wavelet Frame”, *1997 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 2527–2532, October 1997.
- [29] Mallat, S.G., “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.
- [30] Unser, M., “Texture Classification and Segmentation Using Wavelet Frames”, *IEEE Transactions on Image Processing*, vol. 4, pp. 1549–1560, November 1995.
- [31] Strang, G., Strela, V., “Short Wavelets and Matrix Dilation Equations”, *IEEE Transactions on Signal Processing*, vol. 43, pp. 108–115, January 1995.
- [32] Daubechies, I., “The Wavelet Transform, Time-Frequency Localization and Signal Analysis,” *IEEE Transactions on Information Theory*, vol. 36, pp. 961–1005, September 1990.
- [33] The Mathworks Inc., Wavelet Toolbox, “Wavelets: A New Tool for Signal Analysis”, July 2005, <http://www.mathworks.com/access/helpdesk/help/toolbox/wavelet/wavelet.html>
- [34] Laws, K.I., *Textured Image Segmentation*, PhD thesis, University of Southern California, January 1980.

- [35] Crouse, M.S., Nowak, R.D., Baraniuk, R.G., “Wavelet-Based Statistical Signal Processing Using Hidden Markov Models”, *IEEE Transactions on Signal Processing*, vol. 46, pp. 886–902, April 1998.
- [36] Choi, H., Baraniuk, R.G., “Multiscale Image Segmentation using Wavelet-Domain Hidden Markov Models”, *IEEE Transactions on Image Processing*, vol. 10, pp. 1309-1321, September 2001.
- [37] Chaudhuri, B.B., Sarkar, N., “Texture Segmentation Using Fractal Dimension”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 72-77, January 1995.
- [38] Ojala, T., Pietikäinen, M., Harwood, D., “Performance Evaluation of Texture Measures with Classification Based on Kullback Discrimination of Distributions”, *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Computer Vision & Image Processing*, vol. 1, pp. 582-585, October 1994.
- [39] Ojala, T., Pietikäinen, M., “Unsupervised Texture Segmentation using Feature Distributions”, *Pattern Recognition*, vol. 32, pp. 477-486, 1998.
- [40] Mäenpää, T., Ojala, T., Pietikäinen, M., Soriano, M., “Robust Texture Classification by Subsets of Local Binary Patterns”, *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 3, pp. 935-938, Barcelona, Spain, September 2000.
- [41] Pietikäinen, M., Ojala, T., Xu, Z., “Rotation-Invariant Texture Classification Using Feature Distributions”, *Pattern Recognition*, vol. 33, pp. 43-52, 2000.
- [42] Brodatz, P., *Textures: A Photographic Album for Artists and Designers*, Dover Publications Inc., New York, 1966.
- [43] Sokal, R. R., Rohlf, F. J., *Biometry: The Principles and Practice of Statistics in Biological Research*, Second Edition, W. H. Freeman and Company, San Francisco, 1981.
- [44] Sokal, R. R., Rohlf, F. J., *Introduction to Biostatistics*, W. H. Freeman and Company, San Francisco, U.S.A, 1973.
- [45] Physics Department, College of Saint Benedict | Saint John’s University, “Kolmogorov-Smirnov test, chi-square distribution test, ANOVA: ANalysis Of VAriance between groups”, May 2005, <http://www.physics.csbsju.edu/stats/>
- [46] VanMarcke, E., *Random Fields: Analysis and Synthesis*, MIT Press, Cambridge MA, 1983.
- [47] Statistique Canada / Statistics Canada, “Glossaire – Les Statistiques : Le Pouvoir Des Données!”, February 2006, http://www.statcan.ca/francais/edu/power/glossary/gloss_f.htm.

- [48] Ripley, B.D., “Tests of ‘Randomness’ for Spatial Point Patterns”, *Royal statistical Society, Journal of the Royal Statistical Society, Series B*, Vol. 41, No. 3, pp. 368-374, March 1979.
- [49] Getis, A., Franklin, J., “Second-Order Neighborhood Analysis of Mapped Point Patterns”, *Ecological Society of America, Ecology*, Vol. 68, No. 3, pp. 473-477, 1987.
- [50] Abou Merhy, B., Payeur, P., Petriu, E.M., “Unsupervised Texture Segmentation for 2D Probabilistic Occupancy Maps”, *Proceedings of IEEE International Workshop on Robotic and Sensors Environments*, pp. 43-48, Ottawa, ON, Sept. 2005.
- [51] Abou Merhy, B., Payeur, P., Petriu, E.M., “Application of Segmented 2D Probabilistic Occupancy Maps for Mobile Robot Sensing and Navigation”, *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference*, pp. 2342-2347, Sorrento, Italy, 24 - 27 April 2006.
- [52] Matlab v7.0, *The MathWorks Inc.*, 2001–2006.
- [53] Image Processing Toolbox v5.2, *The MathWorks Inc.*, 1994–2006.
- [54] Deitel, H. M., Deitel, P. J., *C How to Program*, Third Edition, Prentice Hall, Upper Saddle River, New Jersey, U.S.A., 2001.
- [55] Deitel, H. M., Deitel, P. J., *C++ How to Program*, Fifth Edition, Pearson Education Inc./Prentice Hall, Upper Saddle River, New Jersey, U.S.A., 2005.
- [56] Deitel, H. M., Deitel, P. J., Liperi, J. P., Yaeger, C. H., *Visual C++[®] .NET How to Program*, Pearson Education Inc./ Prentice Hall, Upper Saddle River, New Jersey, U.S.A., 2004.
- [57] Free Software Foundation Inc., “The GNU C Library Reference Manual”, December 2005, http://www.gnu.org/software/libc/manual/html_node/index.html.
- [58] Eruhimov, V., *OpenCV: Open Source Computer Vision Library*, CVPR01 Course, Intel Corporation, Russia, 2001.
- [59] Borrás, R., Open Source Computer Vision Library Community, “How to Extract a Sub-Image”, July 2005, <http://groups.yahoo.com/group/OpenCV/message/30300>.
- [60] Osten, E., University of Wisconsin, “FLTK Tutorial: CSS 559 Introduction to Computer Graphics”, December 2005, <http://www.cs.wisc.edu/graphics/Courses/f2003-cs559/ftkTutorial.htm>.
- [61] Sweet, M., Earls, C. P., Spitzak, B., Bill Spitzak and Others, “FLTK 1.1.6 Programming Manual”, December 2005, <http://www.fltk.org/doc-1.1/>.
- [62] Ercolanos, G., “Erco’s FLTK Cheat Page”. December 2005, <http://seriss.com/people/erco/fltk/>.

- [63] Kapralos, B., CSC 320 Lectures Notes, University of Toronto, “Advanced Topics in Edge detection, FLTK Sample Program 2”, December 2005, <http://www.cdf.utoronto.ca/~billk/csc320/lectureNotes/prog2.cpp>.
- [64] Irvine, H., Smirniotopoulos, J.G., USU Department of Radiology and Radiological Sciences, Uniformed Services University of the Health Sciences, “MedPix™, Medical Image Database”, April 2006, <http://rad.usuhs.mil/medpix/medpix.html?mode=default>.
- [65] Google Inc., “Navteq™ : Données Cartographiques”, April 2006, <http://maps.google.ca>.
- [66] Abou Merhi, F., “Paris Sous la Neige, Premiers Pas”, December 2005, <http://pinkb.free.fr/galerie/index.php?album=7>.
- [67] Latombe, J.-C., *Robot Motion Planning*, Kluwer Academic Publisher, 1991.
- [68] Stanley, R.P., *Enumerative Combinatorics: Volume 1*, Cambridge University Press, 1997.
- [69] Microsoft Corporation, “The MSDN Library”, February 2006, <http://msdn.microsoft.com/library/>.
- [70] Cieslak, P., Molofee, F., OpenGL Tutorials, Neon Helium Productions, “Morphing and Loading Objects from a File”, February 2006, <http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=25>.
- [71] Shreiner, D., Woo, M., Neider, J., Davis, T., *OpenGL Programming Guide: the Official Guide to Learning OpenGL*, Version 1.4, Fourth Edition, Addison-Wesley Professional, November 14, 2003.
- [72] Shreiner, D., *OpenGL Reference Manual: The Official Reference Document to OpenGL*, Version 1.4, Fourth Edition, Addison-Wesley Professional, March 16, 2004.
- [73] Grogono, P., Department of Computer Science and Software Engineering, Concordia University, “Graphics – Example Programs”, February 2006, <http://www.cs.concordia.ca/~grogono/Graphics/examples.html>.
- [74] Ramires Fernandes, A., Light House 3D, “GLUT Tutorial”, February 2006, <http://www.lighthouse3d.com/opengl/glut/>.
- [75] Kilgard, M. J., Version 3, Silicon Graphics Inc., 1996, “The OpenGL Utility Toolkit (GLUT) Programming Interface API”, February 2006, <http://www.opengl.org/resources/libraries/glut/spec3/spec3.html>.

Annexe A Simulateur de capteur télémétrique

Dans cette première annexe, nous exposons les fonctionnalités du programme de simulation de capteur utilisé pour créer les modèles probabilistes d'environnements. Ce programme, développé par Bolzon, dont l'interface usager est illustrée dans la Figure A-1, fournit plusieurs possibilités de configurations à l'utilisateur :

- ❖ Le choix de l'environnement encombré (Figure A-2-a), ce dernier correspond à une image binaire de taille [500 x 500] pixels où les pixels noirs et blancs correspondent respectivement aux régions libres et occupées de l'espace;
- ❖ Trois méthodes de fusions de données différentes permettent la réalisation de la cartographie de l'environnement. La première correspond au théorème de Bayes, la seconde à celui de Dempster-Shafer (nommé également : Bayes généralisé ou théorie de l'évidence) et la troisième au moteur d'inférence floue [8]. Ce sont ces méthodes de fusion qui assignent les probabilités d'occupation aux cellules. Elles fusionnent également les données probabilistes dans le cas où les cellules sont touchées par plusieurs faisceaux laser;
- ❖ La définition des paramètres des capteurs utilisés, tels le pas angulaire entre les faisceaux de lecture adjacents et la mesure de l'écart-type de l'erreur gaussienne sur la distance;
- ❖ L'initialisation du nombre de points de vue utilisés, leurs coordonnées ainsi que leur direction.

Une fois que l'utilisateur choisit et valide les paramètres de la simulation ainsi que l'environnement encombré (Figure A-2-a) à l'aide de l'interface usager (Figure A-1), le programme fournit les deux sorties suivantes :

- ❖ Une image bidimensionnelle en niveaux de gris représentant le taux d'occupation de l'espace (Figure A-2-b). Plus les niveaux de gris sont clairs, plus la probabilité que les cellules soient occupées est importante. Ce résultat est fourni en entrée à l'algorithme de segmentation proposé dans cette thèse;
- ❖ Un graphe d'occupation tridimensionnel (Figure A-2-c).

Il est important de signaler que les zones qui ne sont pas touchées par une lecture de capteur sont considérées comme inconnues et sont représentées par une probabilité d'occupation égale à 0.5 (niveau de gris intermédiaire dans la Figure A-2-b).

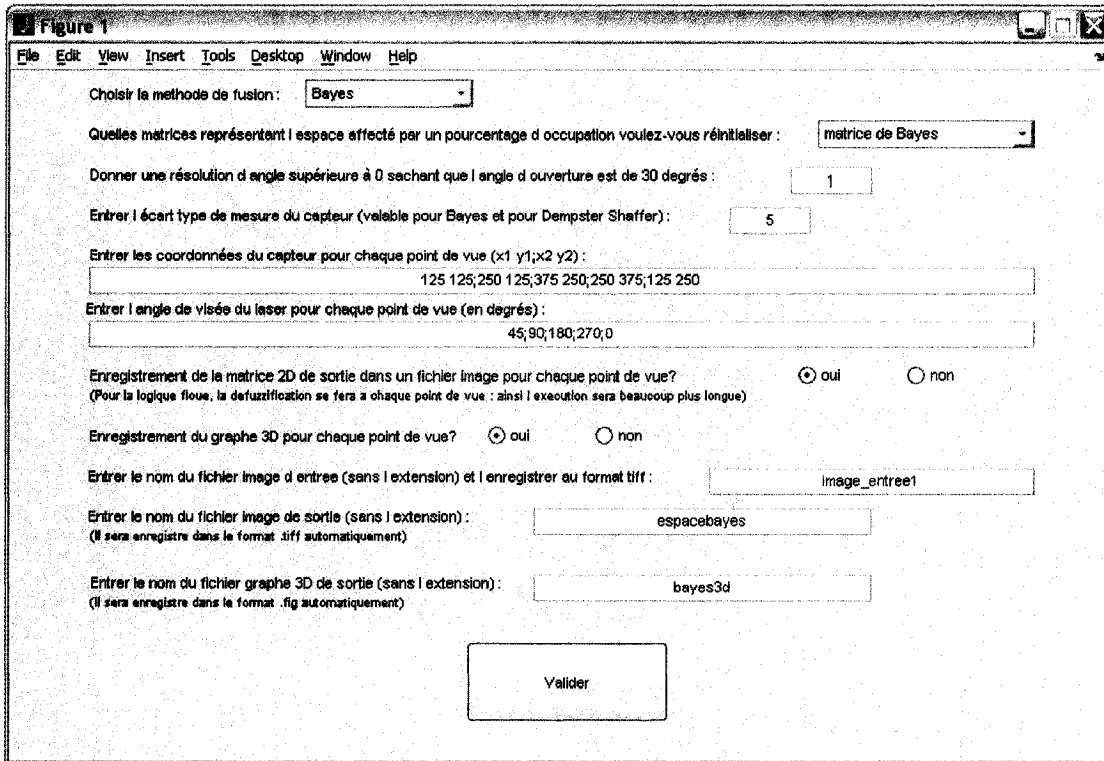


Figure A-1 : interface usager pour le programme de simulation de capteur télémétrique

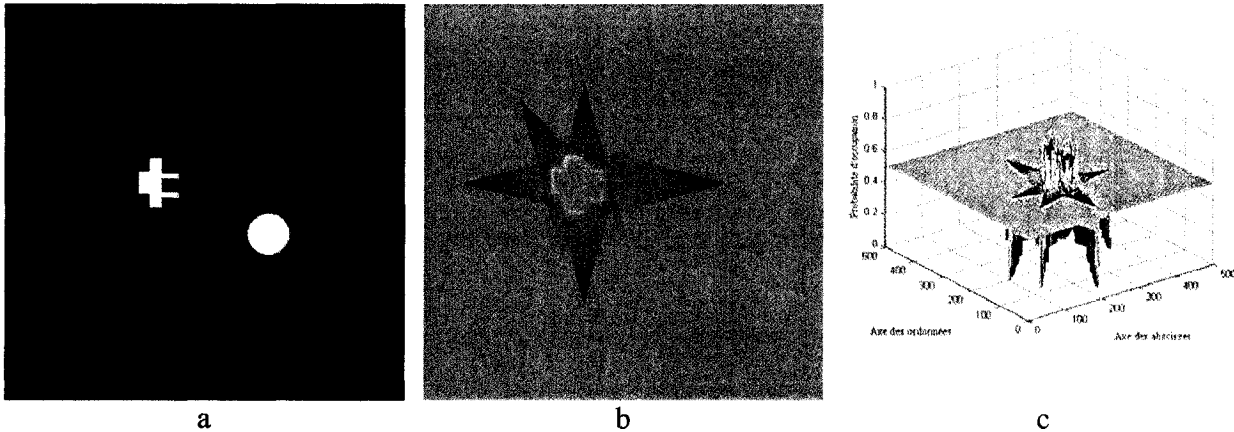


Figure A-2 : (a) espace encombré, (b) image probabiliste bidimensionnelle, (c) graphe d'occupation

Annexe B Interface usager de la segmentation bidimensionnelle

L'algorithme de segmentation bidimensionnelle a été implanté à l'aide du langage de programmation C [54][55][56][57] et de la librairie « Open Source » *OpenCV* [58][59] développée dans les laboratoires de *Intel Inc.* L'interface usager a été, quand à elle, développée à l'aide de la librairie « Open Source » *FLTK*¹⁹ [60][61][62][63]. L'avantage majeur de cette dernière outre la simplicité de son utilisation, réside dans la facilité de la migration du code vers d'autres plateformes que *Microsoft® Windows®*, tels *UNIX®* ou *Mac OS®*.

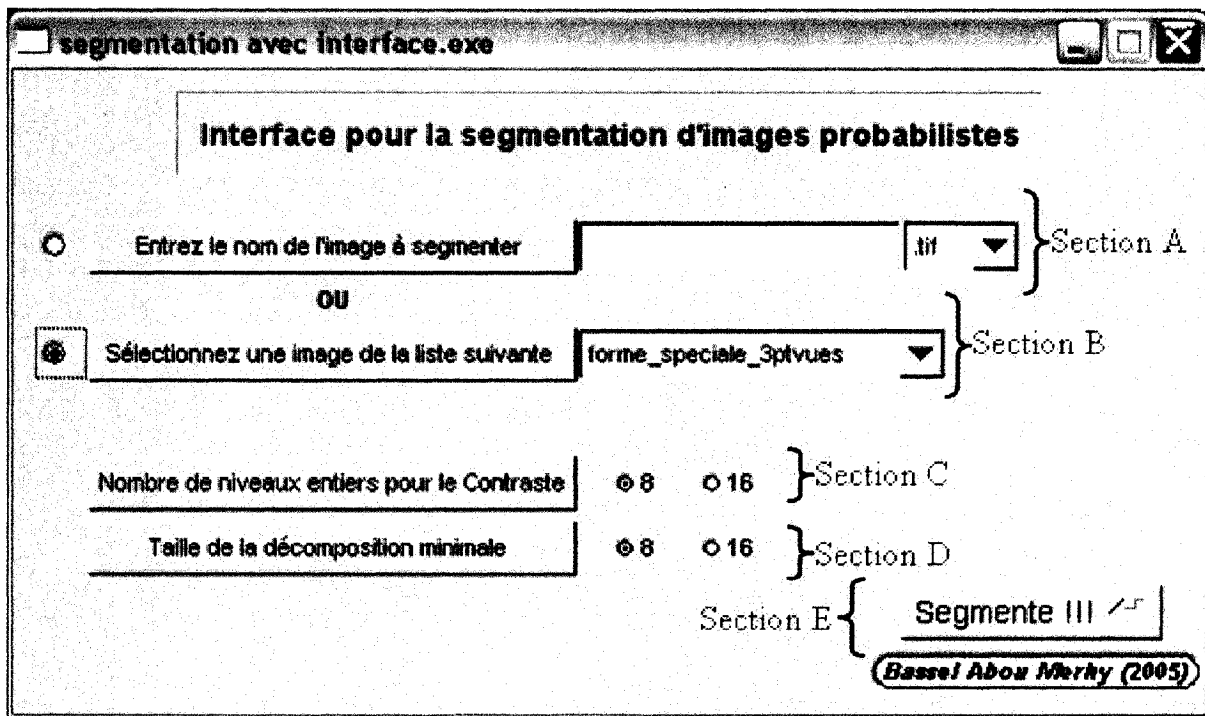


Figure B-1 : interface usager développée à l'aide de la librairie FLTK

La Figure B-1 correspond à l'interface usager que nous avons implantée. Elle comporte cinq sections numérotées par les lettres allant de « A » à « E ». Étant donné que la composante graphique ne constitue pas une priorité dans cette thèse, nous allons décrire les champs d'interaction sans rentrer dans les détails techniques de l'implantation :

- ❖ Les sections A et B correspondent à des choix exclusifs qui permettent respectivement à l'utilisateur d'entrer le nom de l'image qu'il désire segmenter, ou de choisir une des images

¹⁹ « fulltick » ou « Fast Light Toolkit ».

préexistantes. Étant donné que notre programme supporte deux formats d'images, soit le *Tagged Image File Format*[™] [Adobe Systems Inc.] (reconnu sous son extension « .tiff[™] ») et le *Bitmap* (reconnu sous son extension « .bmp »), nous avons inclus dans la section A l'option de choisir le type de l'image d'entrée.

- ❖ La section C permet à l'utilisateur de choisir entre huit et seize niveaux de discrétisations, b , pour les valeurs décimales du contraste.
- ❖ La section D quand à elle, donne le choix entre deux valeurs pour la taille minimale qu'une subdivision peut atteindre. Ainsi, les deux valeurs possibles pour le paramètre S_{min} sont de huit et de seize.
- ❖ La section E correspond au bouton qui lance le programme de la segmentation.

Annexe C Modules de visualisation 3D et interface usager

L'algorithme de segmentation LBP/C tridimensionnel a été implanté à l'aide du langage C++ [54][55][56][57][69] et de la librairie « Open Source » *OpenCV* [58][59] développée dans les laboratoires de *Intel Inc.* Le visionnement de l'environnement simulé et du résultat final de la segmentation, a été possible grâce à l'API²⁰ de graphiques 3D OpenGL²¹ [70][71][72] et à l'API GLUT²² [73][74][75] qui lui est compatible. L'interface usager qui permet à l'utilisateur de choisir le modèle à segmenter ainsi que les paramètres utilisés a été développée à l'aide de la librairie « Open Source » *FLTK* [60][61][62][63]. Le choix du langage programmation s'explique principalement par sa portabilité et par les fonctionnalités qu'il fournit en terme de structures de données prédéfinies. D'autre part, le choix des APIs OpenGL et GLUT ainsi que de l'outil FLTK, se base sur leur facilité d'intégration mutuelle et leur portabilité entre une multitude de systèmes d'exploitation tels : *UNIX*[®], *Microsoft*[®] *Windows*[®] et *Mac OS*[®].

Le résultat de l'algorithme de segmentation est un vecteur d'images dans lesquelles l'intensité des pixels correspond à l'indice du segment auquel ils appartiennent. La visualisation de cet environnement tridimensionnel nécessite un traitement préliminaire capable de doter les pixels de coordonnées en abscisse, en ordonnée et en élévation. Dans le but de réduire les ambiguïtés dans la visualisation tridimensionnelle, nous avons décidé d'afficher séparément les segments obtenus. Par conséquent, l'intensité (probabilité d'occupation) des vertex est d'une importance secondaire. D'autre part, afin que le résultat soit entièrement contenu dans la fenêtre de visualisation GLUT, nous avons décidé d'opérer une conversion sur 6 unités (allant de -3 jusqu'à 3 afin de le centrer dans la fenêtre) de la dimension la plus grande dans l'espace tridimensionnel. À titre d'exemple, si un environnement est de $[l=320 \times h=256 \times e=192]$ cellules, sa largeur sera considérée et convertie sur 6 unités, avec un pas de $6 / 320 = 0.01875$ (équation (C.1)) entre les cellules horizontalement adjacentes. Afin de réduire les distorsions visuelles des segments, les cellules de l'espace tridimensionnel sont considérées cubiques, et donc espacées d'une façon uniforme. Ainsi, la dernière observation implique que le pas utilisé pour convertir la dimension la plus grande sera imposé sur les autres dimensions. En se basant sur les données de l'exemple ci-dessus, la conversion de la hauteur

²⁰ « Application Programmée d'Interface ».

²¹ « Open Graphics Library ».

²² « OpenGL Utility Toolkit ».

sera sur $256 \times 0.01875 = 4.8$ unités (allant de -2.4 jusqu'à 2.4) tandis que celui de l'élévation sur $192 \times 0.01875 = 3.6$ unités (allant de -1.8 jusqu'à 1.8). Afin de représenter séparément les segments, les coordonnées des cellules de chacun d'entre eux sont sauvegardées dans un fichier texte différent. Le module de visualisation commence par créer ces fichiers. Par la suite, l'environnement tridimensionnel est parcouru et la valeur des cellules sert à décider dans quel fichier segment ses coordonnées seront imprimées. Trois variables décimales x_pos , y_pos et z_pos servent de coordonnées pour les vertex (ou cellules) de l'environnement, celles-ci sont initialisées selon les équations suivantes :

$$pas = \frac{6}{\max(l_{\text{environnement}}, h_{\text{environnement}}, e_{\text{environnement}})} \quad (\text{C.1})$$

$$x_pos = -\frac{pas \times l_{\text{environnement}}}{2} \quad (\text{C.2})$$

$$y_pos = -\frac{pas \times h_{\text{environnement}}}{2} \quad (\text{C.3})$$

$$z_pos = -\frac{pas \times e_{\text{environnement}}}{2} \quad (\text{C.4})$$

Où : pas : distance qui sépare deux vertex adjacents.

$\max(a, b, c)$: fonction qui sert à calculer le maximum de trois valeurs entières a, b, c .

$l_{\text{environnement}}, h_{\text{environnement}}, e_{\text{environnement}}$: largeur, hauteur et élévation de l'environnement tridimensionnel.

Les variables x_pos , y_pos et z_pos sont incrémentées d'un pas (équation (C.1)) afin de calculer les coordonnées du vertex suivant respectivement le long de la largeur, de la hauteur et de l'élévation. Une fois qu'une dimension a été traitée, la variable de coordonnée correspondante est réinitialisée à l'aide de l'équation convenable ((C.2), (C.3) ou (C.4)).

Étant donné que nous disposons de l'environnement simulé, nous avons choisi d'afficher simultanément les objets qu'il incorpore et le résultat de la segmentation. L'environnement tridimensionnel réel est quant à lui encodé dans un vecteur d'images binaires. Dans ces dernières, les cellules libres et occupées sont respectivement de niveau noir et blanc. Ainsi, un processus similaire à celui décrit dans le paragraphe précédent est appliqué sur le modèle réel. Seul l'espace réellement

occupé est pris en compte et les coordonnées des vertex équivalents à ses cellules sont imprimées dans un fichier texte.

Une fois que le modèle réel ainsi que les segments sont encodés dans des fichiers texte, le module de visionnement OpenGL est lancé. Il commence par charger dans des *Objets* spécifiques les coordonnées des vertex relatifs à chacun des modèles tridimensionnels (segments et région occupée réelle). En terme de visualisation, la fenêtre principale est divisée en deux sous-fenêtres, la première sert à afficher le résultat de la segmentation, tandis que la seconde présente l'espace occupé réel. Comme nous l'avons signalé précédemment, les segments sont affichés séparément. Par conséquent les flèches de gauche et de droite du clavier permettent de sélectionner celui à visualiser dans la sous-fenêtre de gauche. Nous avons utilisé le texte de format *bitmap* afin d'afficher premièrement le résultat des mesures d'erreur dans la partie supérieure de la fenêtre principale, et deuxièmement la description des environnements affichés (segments *inconnu*, *libre*, *occupé* et objets réels) dans les sous-fenêtres.

Les espaces tridimensionnels (segmenté et réel) sont affichés en mode de projection de perspective. Afin que le texte soit toujours en premier plan, juste durant son affichage (*rendering*) le mode de visualisation orthographique bidimensionnel est sélectionné.

Le clavier est utilisé afin de gérer l'affichage des modèles tridimensionnels :

- ❖ Comme signalé précédemment, les flèches de gauche et de droite permettent de sélectionner le segment à afficher dans la sous-fenêtre de gauche;
- ❖ Les flèches de bas et de haut sont utilisées respectivement pour zoomer arrière et avant sur les deux environnements tridimensionnels affichés dans les sous-fenêtres;
- ❖ Les touches *x*, *y* et *z*, permettent de faire pivoter les deux environnements affichés respectivement autour des axes des abscisses, des ordonnées et de l'élévation.

Nous avons décidé de contrôler simultanément les segments ainsi que les objets réels afin de les afficher sous le même angle, ce qui permet de mieux évaluer visuellement la qualité des résultats obtenus.

Deux fenêtres de visualisation sont illustrées dans la Figure C-1 et la Figure C-2. Celles-ci correspondent au résultat de la segmentation de deux environnements probabilistes, le premier incorporant un objet conique tandis que le second correspond à un objet cylindrique. Le texte en

rouge situé dans la partie supérieure de la fenêtre principale correspond aux mesures d'erreur de la seconde et de la troisième phase de notre algorithme, la sous-fenêtre de gauche permet de visualiser les segments (espace *occupé* dans la Figure C-1 et *libre* dans la Figure C-2), tandis que celle de droite affiche l'objet présent dans l'environnement simulé.

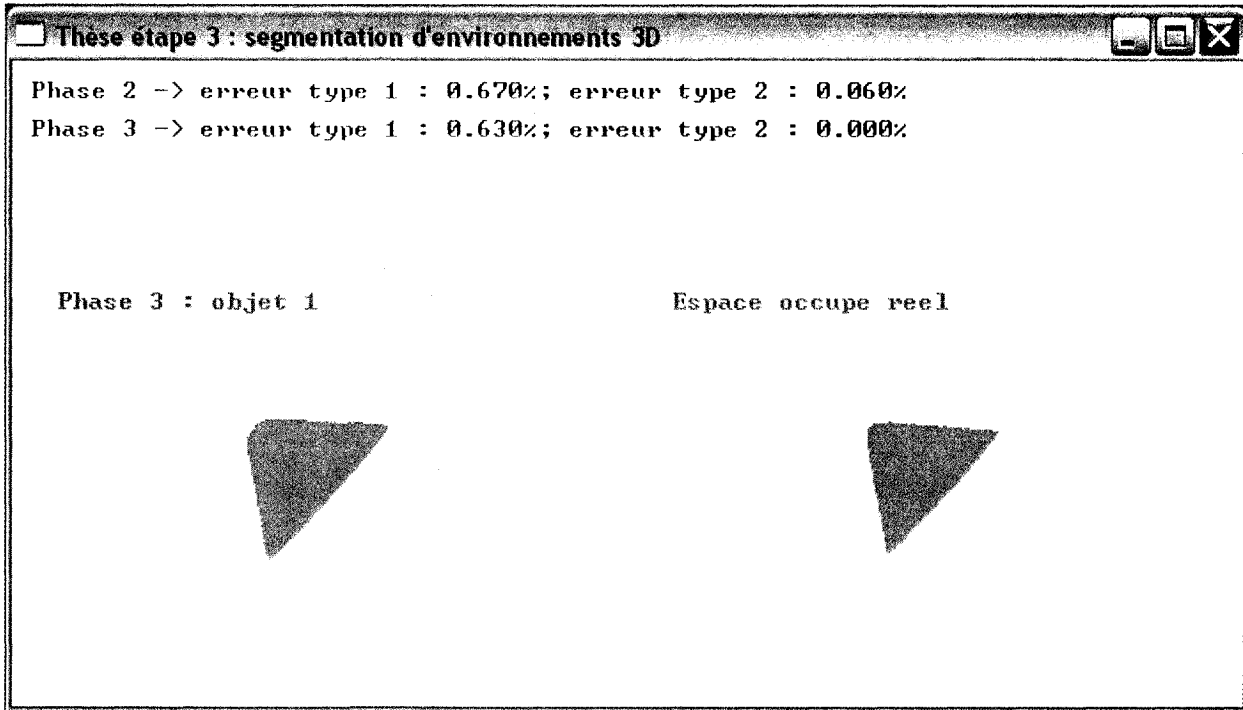


Figure C-1 : région occupée issue de la segmentation d'un environnement incorporant un objet conique

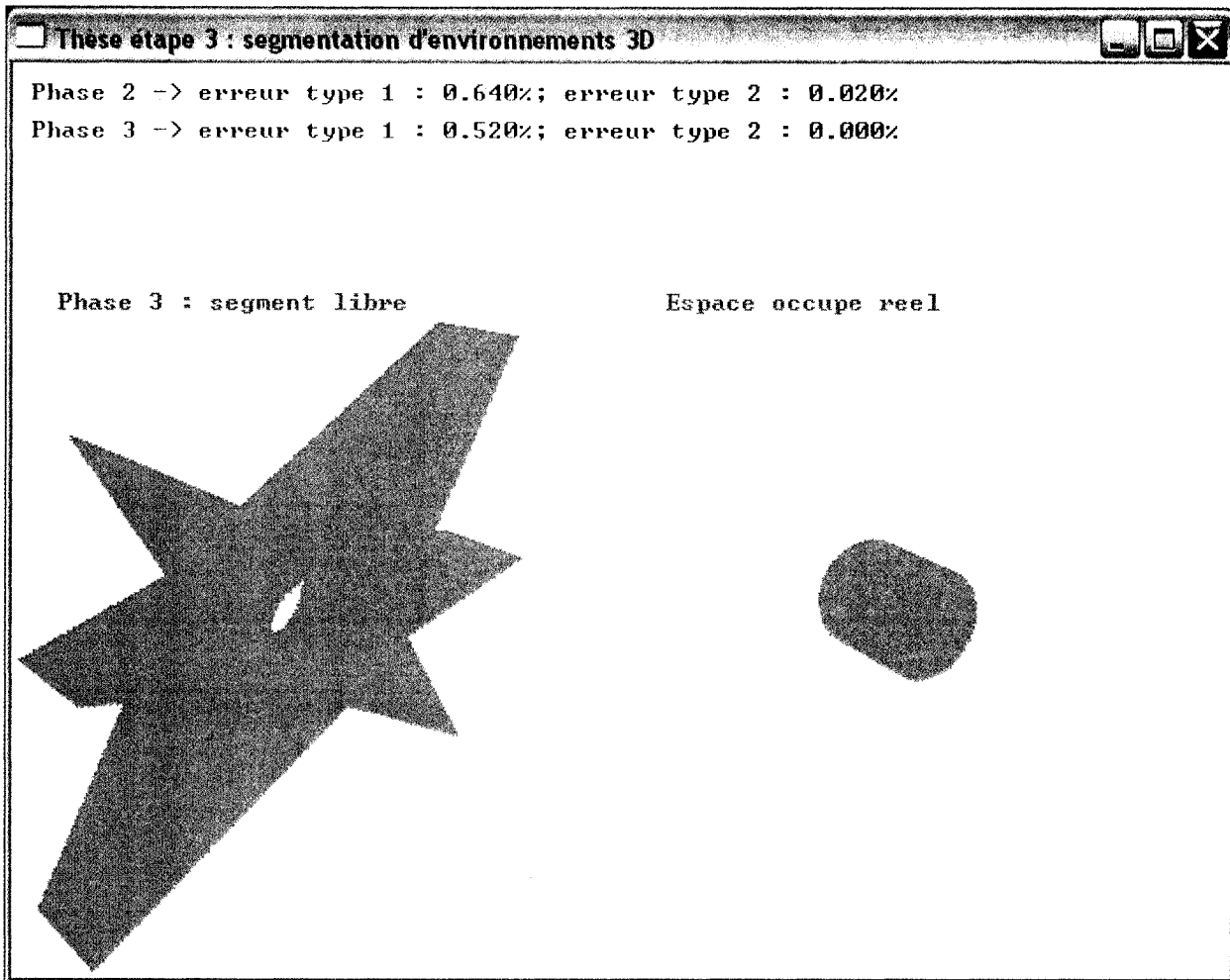


Figure C-2 : région libre issue de la segmentation d'un environnement incorporant un objet cylindrique

L'interface usager (Figure C-3) a été développée à l'aide de la librairie FLTK [60][61][62][63]. Tout comme celle de l'algorithme de segmentation bidimensionnelle, elle comporte cinq sections (numérotées de « A » à « E » dans la Figure C-3) :

- ❖ La première consiste en une liste qui permet à l'utilisateur de choisir un des six environnements tridimensionnels prédéfinis à segmenter. Dans ces espaces, les objets représentés ont respectivement une forme cubique, conique ou cylindrique.
- ❖ La section B donne le choix à l'utilisateur soit de segmenter le modèle sélectionné, soit d'afficher directement le résultat (s'il existe). Cette dernière alternative a été possible grâce à la structure de notre programme qui sauvegarde les résultats d'une segmentation donnée dans un répertoire doté d'un nom unique. Ce dernier est obtenu en concaténant le nom du

modèle en entrée avec les paramètres de la segmentation (nombre de niveaux entiers pour convertir le contraste et de la taille de la décomposition minimale (S_{min})). Étant donné que l'utilisateur contrôle tous ces paramètres à l'aide de l'interface usager, le programme est en mesure de retrouver, s'ils existent, les résultats de la segmentation antérieure du modèle.

- ❖ La troisième section, C, permet à l'utilisateur de choisir le nombre de niveaux entiers, b , utilisés dans la conversion de la valeur décimale du contraste.
- ❖ La section D quant à elle définit la taille minimale, S_{min} , qu'une subdivision peut atteindre durant la première phase de l'algorithme.
- ❖ La dernière section correspond au bouton qui permet de lancer l'application.

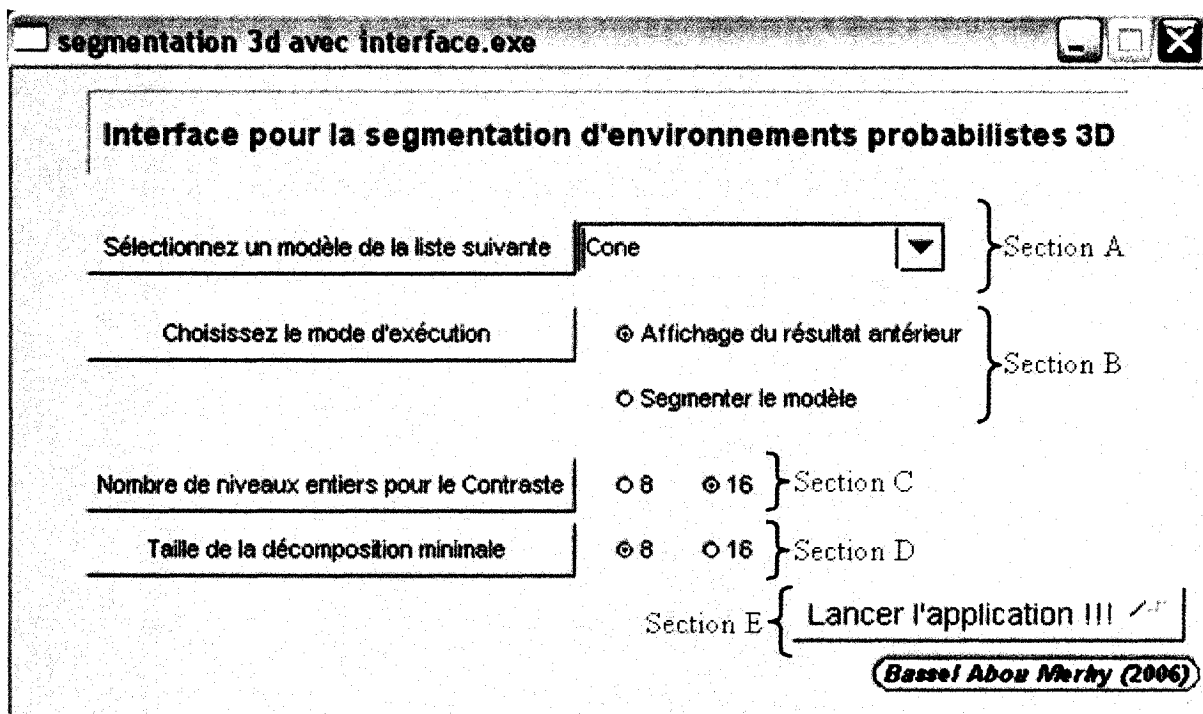


Figure C-3 : interface usager du programme de segmentation tridimensionnelle