

UNIVERSITE SAAD DAHLAB DE BLIDA

Facultés des Sciences de l'ingénieur

Département d'Electronique

MEMOIRE DE MAGISTER

En Electronique

Spécialité : Traitement d'Images et de la Parole

SEGMENTATION ADAPTATIVE DES IMAGES MONO-SPECTRALES PAR COOPERATION DE METHODES

Par

ZEGAOUA Fatna

Devant le jury composé de:

Mme NADJIA BENBLIDIA	MCA, U. de Blida	Présidente
Mr M. HALIMI	MR, CSC	Examineur
Mr ABDERRAHMANE NAAMANE	MCA, U. de Blida	Examineur
Mr ABDERREZAK GUESSOUM	Professeur, U. de Blida	Directeur de thèse
Mr YACINE KEBIR	MAA, U. de Blida	Invité

Blida, Mars 2012

RESUME

La segmentation d'images est une étape incontournable dans tout processus d'analyse d'image. C'est un traitement de bas niveau qui précède l'étape de mesure, de compréhension et de décision. Son objectif consiste à partitionner l'image en régions connexes et homogènes au sens d'un critère d'homogénéité difficile à définir surtout dans le cas de régions texturées.

La plupart des méthodes de segmentation publiées dans la littérature nécessitent l'ajustement de plusieurs paramètres de contrôle afin d'obtenir de bons résultats. De plus, ces méthodes sont fortement liées à l'application visée. La recherche d'une méthode flexible adaptable à tout type d'images reste un problème ouvert en analyse d'images. De ce fait, la segmentation adaptative par coopération de méthodes suscite un grand intérêt ces dernières années.

La méthode proposée a l'avantage d'adapter le traitement au contexte selon le type d'image à segmenter (uniforme, faiblement ou fortement texturée).

Pour arriver à notre but nous avons étudié l'hétérogénéité de l'image en question pour déterminer son type et par la suite les attributs à utiliser. L'algorithme de Mean-Shift est en suite appliqué à l'espace des caractéristiques pour définir les centroides initiaux à utiliser dans la dernière étape de classification par l'algorithme de Fuzzy-C-Mean.

L'avantage major de notre méthode réside dans l'adaptation des traitements au contexte de l'image avec le minimum de connaissances a priori. Elle exploite l'efficacité des matrices de cooccurrence et l'analyse multi-échelle des filtres de Gabor dans le cas des images texturées pour aboutir à un résultat de segmentation plus précis et plus fidèle.

ABSTRACT

Segmentation is the process of extracting meaningful objects from an image. This task, although easily achieved by the human visual system, is actually complex and still a true challenge for the image processing community despite several decades of research. Its aim is to divide the image to separate regions according to a similarity criterion.

Generally, the existing segmentation methods need to adjust lot of parameters to reach satisfactory results. Also, those methods depend on their final application.

Finding a flexible and adaptive method that deals with all images types is a difficult problem to solve in image analysis. By consequent, over recent years, there has been growing interest in the use of cooperative approaches in image segmentation.

The proposed framework has the advantage of adapting the processing operation to the image type. We have analyzed the image heterogeneity in order to define its type (uniform, low or high textured). Knowing that, this pretreatment will help us to determine the appropriate features to be used for the segmentation process. The Mean-shift algorithm is then applied to these features space to get the clusters centers that are used as initial centroids in the last stage of classification by the Fuzzy-C-Mean algorithm.

The main advantage of our approach is found by taking advantage of the efficiency of the co-occurrence matrix and multi-resolution analysis provided by the Gabor filters in case of textured images to reach more reliable results.

The efficiency of our new system is demonstrated by the segmentation results.

REMERCIEMENTS

Je tiens à exprimer tous mes remerciements et ma vive gratitude au Professeur A.GUESSOUM d'avoir d'abord accepté de diriger ma thèse et pour sa patience, sa rigueur scientifique, ses conseils et son soutien pendant tous ces années de recherche.

J'adresse mes vives gratitudeux aux membres de jury qui ont acceptés de juger mon mémoire.

J'adresse mes sincères remerciements au Dr. KHEDAM. Radja de l'université de Houari Boumediene pour son aide et son soutien dans les moments les plus difficiles dans ma recherche et de m'avoir ouvert sa porte.

Je tiens à rendre hommage au Dr. OUKIL. Assia de l'université de Houari Boumediene pour ses conseils avisés, son aide, son suivi et ses corrections même sans me voir.

Mesdames, trouvez ici le témoignage de ma gratitude pour tous ce que vous avez fait pour moi.

J'adresse mes remerciements au Dr. KEBIR. Yacine de l'université de Saad Dahleb pour son aide et son soutien.

Je tiens à exprimer mes remerciements à Mr.TIRARI et Mr. BESKRI de l'université de Haouari Boumediene pour leur aide et leur orientation, qui sans eux, ce travail n'aura jamais vu la lumière.

A mes anges : mes parents, mes frères et sœurs qui m'ont soutenue pendant tous ces années, que Dieu les protège.

TABLE DES MATIERES

RESUME	
ABSTRACT	
REMERCIEMENTS	
TABLE DES MATIERES	
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	
INTRODUCTION	6
1. ETAT DE L'ART EN SEGMENTATION D'IMAGES	
1.1. Introduction	10
1.2. Méthodes de segmentation non-coopérative	13
1.2.1. Méthodes adaptées aux images uniformes	13
a. Détection de contours	13
b. Croissance de régions	14
c. Méthodes de segmentation par seuillage	15
1.2.2. Méthodes adaptées aux images texturées	18
a. Les méthodes structurales	19
b. Les méthodes statistiques	19
c. Les méthodes basées sur l'étude des formes	20
d. Les spatio-fréquentielles	20
e. Les méthodes fractales	20
1.3. Méthodes de segmentation coopératives	21
1.3.1. Méthodes séquentielles	21
a. Coopération contour-région	21
b. Coopération contour-modèle surfacique	22
c. Coopération division-fusion et contours actifs	22
d. Coopération multiseuillage et fusion de régions	22
1.3.2. Méthodes parallèles	23
a. Coopération contour-région	23
b. Extraction des régions et contours par matrices de cooccurrence	23
c. Fusion de résultats de segmentation	24
1.3.3. Méthodes hybrides	24
a. Fusion de plusieurs segmentations	24

b. Coopération active	24
c. Coopération multi-agents	25
d. Coopération adaptative	25
1.4. Conclusion	25
2. PRETRAITEMENTS ET IDENTIFICATION DU TYPE D'IMAGE A SEGMENTER	
2.1. Introduction	26
2.2. Environnement du travail	26
2.3. Etapes d'élaboration du système proposé	29
2.3.1. Organigramme de l'approche proposée	30
2.4. Prétraitements	31
2.4.1. Le seuillage	31
2.4.2. Types de seuillage	31
2.4.3. Notion d'histogramme	32
2.4.4. Seuillage d'Otsu	33
2.4.5. Multi-seuillage adopté	34
a. Choix de taille de la fenêtre d'analyse	36
2.5. Identification du type d'image à segmenter	38
2.5.1. Classement d'image à segmenter	40
a. Choix de taille de la fenêtre d'analyse	40
b. Choix du seuil pour des blocs 4x4 pixels	41
2.6. Conclusion	44
3. SEGMENTATION ADAPTATIVE DES IMAGES UNIFORMES	
3.1. Introduction	45
3.2. Algorithme de Mean-Shift	45
3.2.1. Aperçu historique	45
3.2.2. Principe de fonctionnement	46
3.2.3. Emploi du meanshift dans l'approche proposée	49
3.3. Segmentation des images uniformes	50
3.3.1. Estimation du nombre de modes	51
3.3.2. Limitations du meanshift	54
a. Effets de largeurs de bande	54
b. Effet du seuil de fusion	59
3.4. Conclusion	59

4. SEGMENTATION ADAPTATIVE DES IMAGES TEXTUREES	
4.1. Introduction	61
4.2. Analyse de texture	61
4.2.1. Notion de texture	62
4.2.2. Notion de voisinage	62
4.2.3. Effet d'échelle	63
4.2.4. Techniques d'analyse de texture	63
4.3. Descripteurs de texture	64
4.3.1. Matrice de cooccurrence	65
a. Formulation mathématique	65
b. Choix des paramètres (d et θ)	66
c. Indices de Haralik	69
4.3.2. Matrice de cooccurrence indexée	70
a. Fondement	70
b. Apports obtenus	75
c. Attributs de texture de la matrice indexée	76
4.3.3. Banc de Filtres de Gabor	77
a. Formulation mathématique	77
b. Fondement d'un banc des filtres de Gabor	79
c. Choix des paramètres	79
d. Attributs de Gabor	80
4.4. Classification par Fuzzy-C-Means	82
4.4.1. Notion de classification	82
4.4.2. Types de classification	82
4.4.3. Critères d'agrégation	83
4.4.4. Méthode de clustering adoptée	83
4.5. Segmentation des images texturées	84
4.5.1. Choix des attributs	85
4.5.2. Application du meanshift	86
a. Estimation du nombre de modes	86
4.5.3. Application du FCM	87
4.5.4. Apport de coopération proposée	88
a. Illustrations démonstratives	88
4.6. Conclusion	93
5. RESULTATS ET EVALUATIONS	

5.1. Introduction	94
5.2. Méthodes d'évaluation	94
5.2.1. Evaluation supervisée	95
5.2.2. Evaluation non supervisée	95
5.3. Evaluation des résultats du système proposé	95
5.3.1. Formulation mathématique	96
a. Critère d'uniformité intra-région de Levine et Nazif	96
b. Contraste de Zeboudj	96
5.3.2. Résultats et évaluation	97
5.4. Conclusion	106
CONCLUSION ET PERSPECTIVES	107
APPENDICE	
A. Statistiques du premier et deuxième ordre	110
B. Etude comparative des attributs de cooccurrence	112
REFERENCES	115

LISTE DES ILLUSTRATIONS, GRAPHIQUES et TABLEAUX

Figure 2.1	Exemple des images synthétiques utilisées	27
Figure 2.2	Exemple des images créées et images réelles utilisées	28
Figure 2.3	Fenêtre principale de l'interface développée	28
Figure 2.4	Organigramme de l'approche proposée	30
Figure 2.5	Exemples d'histogramme, (a) « Histogramme d'une image uniforme », (b) « Histogramme de la même image en introduisant des régions texturées »	32
Figure 2.6	Exemple du seuillage global d'Otsu (seuil= 211)	34
Figure 2.7	Exemples de multi-seuillage, (a) « Images originales », (b) «leur multiseuillage ».	35
Figure 2.8	Image originale à 247 niveaux de gris et son histogramme	36
Figure 2.9	Fenêtre du choix de la taille des blocs	36
Figure 2.10	: Image multi-seuillée et son histogramme, Nbr de NG= 292	37
Figure 2.11	Image multi-seuillée et son histogramme, Nbr de NG= 123	37
Figure 2.12	Image multi-seuillée et son histogramme, Nbr de NG= 61	37
Figure 2.13	Image multi-seuillée, Nbr de NG= 23	37
Figure 2.14	Figure 2.14 : Image multi-seuillée, Nbr de NG= 3	38
Figure 2.15	Exemple de localisation des zones uniformes et texturées par mesure de variance (blocs 4 x 4 pixels et T= 16), (a) « Image originale », (b) « Zones uniformes en blanc et texturées en noir », (c) « Superposition des zones texturées à l'image originale »	39

Figure 2.16	Image originale	42
Figure 2.17	Image originale	43
Figure 3.1	Algorithme du Meanshift	48
Figure 3.2	Fenêtre de segmentation des images uniformes	50
Figure 3.3	Segmentation par le mean-shift ($hs=7$, $hr=8$), (a) «Image originale <u>parfaitement uniforme</u> à 4 régions », (b) « Image segmentée »	50
Figure 3.4	Segmentation par le mean-shift ($hs=7$, $hr=23$), (a) «Image originale <u>uniforme bruitée</u> à 5 régions », (b) « Image segmentée »	51
Figure 3.5	Segmentation par le mean-shift ($hs=15$, $hr=13$) , (a) «Image originale à 12 régions uniformes bruitées», (b) « Image segmentée »	52
Figure 3.6	Segmentation par le mean-shift ($hs=7$, $hr=8$) , (a) «Image originale à 6 régions parfaitement uniformes», (b) « Image segmentée »	52
Figure 3.7	Segmentation par le mean-shift ($hs=7$, $hr=13$) , (a) «Image originale à 9 régions uniformes bruitées», (b) « Image segmentée »	53
Figure 3.8	Image originale uniformes bruitée à 6 classes	54
Figure 3.9	Images segmentées avec valeur fixe de hs , (a) « Image segmentée à 4 Ngs, $hs=7$, $hr=21$ », (b) « Image segmentée à 4 Ngs, $hs=7$, $hr=45$ », (c) « Image segmentée à 2 Ngs, $hs=7$, $hr=81$ »	54
Figure 3.10	Images segmentées avec valeur fixe de hr , (a) « Image classifiée à 5 Ngs, $hs=13$, $hr=11$ », (b) « Image classifiée à 5 Ngs, $hs=49$, $hr=11$ », (c) « Image classifiée à 2 Ngs, $hs=75$, $hr=11$ »	55
Figure 3.11	Image originale parfaitement uniforme à 6 classes	55

Figure 3.12	Images segmentées avec valeur fixe de hs , (a) « Image segmentée à 4 Ngs, $hs=9$, $hr=21$ », (b) « Image segmentée à 4 Ngs, $hs=9$, $hr=67$ », (c) « Image segmentée à 2 Ngs, $hs=9$, $hr=100$ »	55
Figure 3.13	Images segmentées avec valeur fixe de hr , (a) « Image segmentée à 4 Ngs, $hs=21$, $hr=6$ », (b) « Image segmentée à 4 Ngs, $hs=55$, $hr=6$ », (c) « Image segmentée à 3 Ngs, $hs=75$, $hr=6$ »	56
Figure 3.14	Image cameraman originale	56
Figure 3.15	Images segmentées avec différentes valeurs de largeurs de bande	57
Figure 3.16	Image originale à droite et image attribut à gauche (normalisée de 0 à 255)	57
Figure 3.17	Images segmentées avec différentes valeurs des largeurs de bande	58
Figure 3.18	Image originale et image attribut (normalisée de 0 à 255)	58
Figure 3.19	Images segmentées avec différentes valeurs des largeurs de bande	59
Figure 4.1	Exemples de textures (10 premières extraites de l'Album de Brodatz°)	62
Figure 4.2	Illustration des connexités	63
Figure 4.3	Matrice de cooccurrence ($d=2, \theta=90^\circ$), (a) « Image originale texturée à 125 Ngs », (b) « son image de cooccurrence »	66
Figure 4.4	Matrice de cooccurrence ($d=1, \theta=45^\circ$), (a) « Image originale uniforme à 142 Ngs », (b) « son image de cooccurrence »	66
Figure 4.5	Matrice de cooccurrence ($d=1, \theta = 45^\circ$), (a) « Image originale », (b) « Image de cooccurrence », (c) « valeurs	67

	numériques de la matrice »	
Figure 4.6	Matrice de cooccurrence ($d=1, \theta = 0^\circ$), (a) « Image originale », (b) « Image de cooccurrence », (c) « valeurs numériques de la matrice »	68
Figure 4.7	Matrice de cooccurrence ($d=1, \theta = 90^\circ$), (a) « Image de cooccurrence », (b) « valeurs numériques de la matrice »	68
Figure 4.8	Matrice de cooccurrence ($d=1, \theta = 135^\circ$), (a) « Image de cooccurrence », (b) « valeurs numériques de la matrice »	69
Figure 4.9	Illustration du calcul de la matrice de cooccurrence	70
Figure 4.10	Illustration du calcul de la matrice de cooccurrence indexée	71
Figure 4.11	Matrice de cooccurrence indexée ($d=1, \theta=0^\circ$), (a) « Image originale », (b) « Image de cooccurrence initiale », (c) « Image de cooccurrence indexée », (d) « Valeurs numériques de la matrice indexée »	72
Figure 4.12	Matrice de cooccurrence indexée ($d=1, \theta=45^\circ$), (a) « Image originale uniforme bruitée », (b) « Image de cooccurrence », (c) « Valeurs numériques de la matrice indexée »	73
Figure 4.13	Matrice de cooccurrence indexée ($d=1, \theta=45^\circ$), (a) « Image originale faiblement texturée », (b) « Image de cooccurrence », (c) « Valeurs numériques de la matrice indexée »	74
Figure 4.14	Matrice de cooccurrence indexée ($d=1, \theta=90^\circ$), (a) « Image originale faiblement texturée », (b) « Image de cooccurrence », (c) « Valeurs numériques de la matrice indexée »	74
Figure 4.15	Matrice de cooccurrence indexée ($d=1, \theta=135^\circ$), (a) « Image originale faiblement texturée », (b) « Image de cooccurrence », (c) « Valeurs numériques de la	75

	matrice indexée »	
Figure 4.16	Matrice de cooccurrence ($d=1, \theta = 45^\circ$), (a) « Image originale à 91Ngs », (b) « son Image de cooccurrence indexée »	76
Figure 4.17	Matrice de cooccurrence ($d=1, \theta = 45^\circ$), (a) « Image multiseuillée à 27 Ngs », (b) « son Image de cooccurrence indexée »	76
Figure 4.18	Exemple d'un Filtre de Gabor, (a) « partie réelle », (b) « partie imaginaire »	81
Figure 4.19	Application d'un filtre de Gabor à une image texturée, (a) « Image originale », (b) « Image filtrée 'partie réelle' », (c) « Image filtrée 'partie imaginaire' »	81
Figure 4.20	Application d'un filtre de Gabor à une image uniforme bruitée, (a) « Image originale », (b) « Image filtrée 'partie réelle' », (c) « Image filtrée 'partie imaginaire' »	81
Figure 4.21	Fenêtre de segmentation des images faiblement texturées	85
Figure 4.22	Fenêtre de segmentation des images fortement texturées	86
Figure 4.23	Fenêtre de résultat segmentation	88
Figure 4.24	Segmentation par matrices de cooccurrence, (a) « Image originale faiblement texturée », (b) « Image segmentée »	89
Figure 4.25	Segmentation par filtres de Gabor, (a) « Image originale faiblement texturée », (b) « Image segmentée »	90
Figure 4.26	Segmentation par coopération, (a) « Image originale faiblement texturée », (b) « Image segmentée »	90
Figure 4.27	Segmentation par matrices de cooccurrence, (a) « Image originale fortement texturées », (b) « Image segmentée »	91
Figure 4.28	Segmentation par filtres de Gabor, (a) « Image originale fortement texturées », (b) « Image segmentée »	92

Figure 4.29	Segmentation par coopération, (a) « Image originale fortement texturées », (b) « Image segmentée »	92
Tableau 2.1	Localisation des zones texturées et uniformes ($T=45$)	40
Tableau 2.2	Localisation des zones texturées et uniformes ($T=20$)	41
Tableau 2.3	Localisation des zones texturées et uniformes	42
Tableau 2.4	Localisation des zones texturées et uniformes pour différents seuils	43
Tableau 5.1	Résultats et évaluations de segmentation des images uniformes par « Niveau de gris »	98
Tableau 5.2	Résultats et évaluations de segmentation des images uniformes par « Autocorrélation »	99
Tableau 5.3	Résultats et évaluations de segmentation des images faiblement texturées par « Variance »	100
Tableau 5.4	Résultats et évaluations de segmentation des images faiblement texturées par « Filtres de Gabor »	101
Tableau 5.5	Résultats et évaluations de segmentation des images faiblement texturées par « Coopération »	102
Tableau 5.6	Résultats et évaluations de segmentation des images fortement texturées par « Attributs de cooccurrence »	103
Tableau 5.7	Résultats et évaluations de segmentation des images fortement texturées par « Filtres de Gabor »	104
Tableau 5.8	Résultats et évaluations de segmentation des images fortement texturées par « Coopération »	105

INTRODUCTION

La qualité de la vision humaine est exceptionnelle. Elle s'avère être malheureusement une mécanique extrêmement complexe dont nous n'avons qu'une idée générale du fonctionnement. Des expériences ont été réalisées et des théories ont été proposées mais elles ne donnent qu'une vue d'ensemble de la vision naturelle.

L'information visuelle est sans doute la plus riche des différentes sources d'informations disponibles. De ce fait, la conception des systèmes de vision pour l'interprétation automatique de scènes suscite un intérêt sans cesse croissant. De nombreux chercheurs se sont penchés sur ce problème et ont mis en œuvre plusieurs approches en vue de la conception de ces systèmes.

L'image tient aujourd'hui une place importante dans des secteurs aussi variés que l'audiovisuel, la robotique, l'exploitation des ressources terrestres et marines, la médecine, la biologie, ...etc. Elle est pluridisciplinaire par ses connexions avec les autres thèmes méthodologiques : traitement du signal, optique, informatique, électronique, télécommunications, mathématiques.

L'objectif des premières étapes de la plus part des systèmes de vision est de décrire l'importance quantité d'informations contenues dans l'image en recherchant des indices visuels ou des primitives pertinents et discriminants permettant de la représenter sous une forme plus condensée et facilement exploitable.

La thématique de recherche dans laquelle s'inscrit cette thèse est celle de la segmentation d'images. Cette opération fondamentale et primordiale vient en amont de toute autre opération dans les systèmes de traitement d'images et c'est à ce titre qu'elle reste encore l'un des principaux thèmes de recherches en vision artificielle.

La segmentation constitue l'un des problèmes les plus importants en analyse d'images car le résultat obtenu à l'issue de cette étape conditionne

fortement la qualité finale de l'interprétation. Bien que le sujet ait déjà fait l'objet d'une vaste littérature, il n'existe pas de méthode générale de segmentation, les techniques employées restent généralement dépendantes :

- ✦ Des spécificités de l'image à traiter : richesse en textures d'orientations et d'échelles différentes, transitions floues entre régions, contours occultés,
- ✦ Du type d'indices visuels à extraire : contours ; régions uniformes au sens des niveaux de gris, textures, points caractéristiques, formes : segments de droite, angles,
- ✦ De la nature du problème à résoudre en aval de la segmentation : reconstruction 3D; reconnaissance de formes, interprétation, contrôle de qualité, suivi automatique d'objets,
- ✦ De contraintes d'exploitation : complexité algorithmique, fonctionnement en temps réel, contraintes matérielles liées aux systèmes d'acquisition, capacité de stockage...etc.

Tous ces contraintes font naissance à plusieurs méthodes de segmentation qu'on peut les grouper en deux catégories fondamentales :

- Méthodes non coopératives, qui contiennent des méthodes adaptées aux images uniformes comme ceux de détection de contour ou de croissance de régions et des méthodes adaptées aux images texturées qui sont composées de méthodes structurelles et méthodes statistiques comme les approches par extraction de paramètres et classification, ou bien les approches par modélisation de texture.
- Méthodes de segmentation coopératives, qui contiennent des méthodes séquentielles telles que les méthodes basées sur la coopération contour-région. Aussi, des méthodes parallèles comme l'extraction des régions et des contours par matrice de cooccurrence et les méthodes hybrides comme la fusion de résultats des plusieurs segmentations.

Il est important également de mentionner que l'on peut considérer deux formes de segmentation, une segmentation dynamique qui vise à effectuer de la détection d'objets dans des séquences vidéo et une segmentation statique qui tente d'atteindre les mêmes objectifs mais sur des images fixes.

La littérature propose de nombreuses techniques de segmentation [4] [28] [36]. Beaucoup de ces méthodes donnent de bons résultats mais se restreignent à un type d'images particulier ou nécessitent plusieurs paramètres dont le réglage exige souvent l'expertise humaine ou encore des connaissances a priori qui sont dans la majorité des cas difficilement accessibles.

Malgré que beaucoup de travaux menés ont permis de développer des systèmes d'analyse adaptables à plusieurs catégories d'images et coopèrent plusieurs méthodes de segmentation, cette dernière reste un problème ouvert, surtout dans le cadre de l'analyse d'images réelles riches en détails et contenant une grande variété de régions de nature uniforme et/ou texturée. Le problème major de ces approches est leur incapacité à s'adapter au contexte.

L'objectif de notre travail est de développer un système de segmentation permettant d'adapter les traitements d'une image étant donné son contexte. Ces traitements se font en deux étapes : la première consiste à identifier le type de l'image à segmenter (uniforme ou texturée) et la seconde permet de la segmenter avec l'ensemble approprié des attributs issus des diverses méthodes adoptées.

Ce mémoire est organisé en cinq chapitres, le premier chapitre est consacré à une présentation générale des différentes approches de segmentation proposées dans la littérature.

Le deuxième chapitre explique les différentes étapes de segmentation proposée et les premières étapes élaborées : le multiseuillage et l'identification du type d'image à traiter. La segmentation des images uniformes, en particulier, l'algorithme adopté du Meanshift, font l'objet du troisième chapitre.

Le quatrième chapitre présente la segmentation des images texturées en exploitant les matrices de cooccurrence et les filtres de Gabor avec la classification par l'algorithme Fuzzy-C-Means.

Les étapes du processus mis en œuvre sont testées et présentées aux chapitres 2, 3 et 4. Une fois les différents ajustements effectués, nous donnons et commentons les résultats de segmentation obtenus au dernier chapitre. Nous terminons ce mémoire par une conclusion et des perspectives au travail réalisé.

CHAPITRE 1

ETAT DE L'ART EN SEGMENTATION D'IMAGES

1.1. Introduction

Les méthodes de segmentation ont fait l'objet de nombreuses recherches, car elles débouchent sur de multiples applications pratiques. L'objectif de cette partie est d'analyser les différentes méthodes de segmentation existantes pour pouvoir exploiter leurs avantages et éventuellement pallier les inconvénients liés à celles que nous choisirons de mettre en œuvre.

Avant de présenter ces méthodes, il est nécessaire de définir l'image et la segmentation qui sont les deux éléments fondamentaux dans cette étude. Une image « I » est, en informatique, une matrice (soit un tableau à deux dimensions) où :

- Chaque case de cette matrice correspond à un pixel. Le pixel étant la plus petite zone affichable, c'est une unité de taille
- Chaque case possède une couleur (codée sur plusieurs composantes (RGB)) ou un niveau de gris (0 - 255).

La segmentation est un traitement bas-niveau qui, à partir de l'image originale ou de l'image filtrée, s'attache à reconstruire les régions qui correspondent aux objets présents dans l'image. Elle transforme l'image à traiter en une image où chaque pixel est caractérisé par une étiquette correspondant à la région à laquelle il appartient. C'est une opération qui a pour but de séparer des objets au fond et entre eux, ou d'une manière plus générale de séparer les différents pixels ayant des propriétés communes.

La segmentation est le partitionnement d'une image I en N régions « R_i » (ensembles de pixels, $i=1, \dots, N$) disjointes et homogènes selon un critère « C »

déterminé. On définit une région comme étant un ensemble de points (pixels) ayant des propriétés communes (intensité, texture, ...) qui les différencient de ceux des régions voisines. La segmentation est caractérisée par :

- ✦ $I = \bigcup_{i=1}^N R_i$, chaque pixel de l'image doit appartenir à une région et l'union de toutes les régions correspond à l'image entière.
- ✦ Les sous ensembles R_i sont connexes. Cette condition est relative à la structure des régions. Elle définit une région comme un sous-ensemble de pixels connexes.
- ✦ $C(R_i) = \text{vrai}, \forall i = 1, \dots, N$, chaque région doit respecter un prédicat d'uniformité.
- ✦ $R_i \cap R_j = \emptyset, \forall (i, j) \text{ tel que } i \neq j, i, j \in \{1, \dots, N\}$, cette condition implique la non-réalisation de ce même prédicat pour toute réunion de deux régions adjacentes.

Remarque

- Une segmentation qui vérifie ces propriétés ne garantit pas une partition unique de l'image.
- $C(R_i)=\text{vrai}$ signifie que le critère C défini sur les attributs de l'ensemble des points de la région R_i est vérifié.

La segmentation d'images a été l'application la plus étudiée dans le domaine d'analyse d'images, car elle a des applications dans de nombreux domaines pratiques, elle a des débouchés dans la reconnaissance de motifs, la détection d'objets, l'analyse d'image médicale [1], la robotique, ou encore dans le domaine des images par satellites [2], pour les applications de recherche d'image par le contenu [3] [4] et bien d'autres encore.

Dans le but d'avoir une segmentation efficace, de nombreux travaux utilisent plusieurs techniques et adaptent chacune d'elles à la nature de la zone de l'image à traiter.

Les méthodes adaptées aux images uniformes au sens du niveau de gris sont apparues les premières tels que : croissance de régions, détection de contours, ...etc. Ensuite, l'analyse de texture a permis de développer d'autres approches pour segmenter les images texturées. Et comme l'utilisation d'une seule méthode ne donne pas généralement des résultats satisfaisants étant donné la complexité du problème, les approches récentes tentent de faire coopérer plusieurs méthodes de segmentation afin d'exploiter leurs complémentarité ou combiner leurs avantages [5] [6][1].

Dans cette discipline, le système visuel humain reste le modèle de référence dans la conception de beaucoup de méthodes de segmentation d'images. Les mécanismes de la vision humaine permettent l'interprétation instantanée de scènes très complexes, font appels à des connaissances a priori et des stratégies de commande de nos capteurs visuels (fixation, suivi d'objets) avec l'utilisation de plusieurs informations tel que l'intensité lumineuse réfléchiée par l'objet. Les principales informations dans l'interprétation du message visuel pour un observateur humain sont les contours et/ou les textures.

Le but de la vision artificielle est de doter les machines d'un système visuel leur permettant, à partir d'images, d'effectuer des tâches complexes comme par exemple, la manipulation d'objets par robots, la surveillance des forêts par l'analyse des images satellitaires et la prévision météorologiques.

Il y a de nombreuses méthodes de segmentation, on distingue, cependant, deux familles principales, selon l'hypothèse admise sur la formation des régions.

La première suppose que les régions adjacentes représentant des objets différents présentent des discontinuités locales de leurs propriétés à leurs frontières. Les méthodes liées à cette approche, dite approche "contours", tentent de rechercher les pixels contours correspondant aux hautes variations entre ces régions.

La seconde suppose qu'une région est constituée de pixels connexes partageant des propriétés similaires. Les méthodes relatives à cette approche, dite approche "régions", recherchent dans l'image des sous-ensembles de pixels connexes et homogènes au sens d'un prédicat précis.

La segmentation peut être réalisée soit par l'application d'une seule approche (*segmentation non coopérative*), soit en combinant deux ou plusieurs techniques (*segmentation coopérative*) :

1.2. Méthodes de segmentation non coopératives

Dans ce type de méthodes, on distingue celles adaptées aux régions uniformes ou faiblement texturées et celles adaptées aux régions fortement texturées [1] [4].

1.2.1. Méthodes adaptées aux images uniformes

Principalement, ce type de méthodes consiste à regrouper des pixels ayant des caractéristiques similaires où les régions correspondent aux parties connexes de ces ensembles, ou bien de détecter des variations entre deux régions différentes. L'information sur les propriétés colorimétriques est une information suffisante pour réaliser cette tâche.

a. Détection de contours

La détermination des limites des objets est un sujet de recherche très important. Les approches liées à cette catégorie recherchent des discontinuités des intensités lumineuses où les contours se manifestent dans l'image par une forte transition des valeurs de ces intensités. Deux catégories existent, celles qui emploient le filtrage différentiel ou adaptatif (tels que les points de fort gradient) et celles qui utilisent les outils de la programmation dynamique tels que la morphologie mathématique ou les approches probabilistes [7] [8]. Parmi les méthodes existantes, on décrit les suivantes:

1) Méthodes dérivatives

Le principe est de détecter une transition d'un signal à partir de sa dérivée d'ordre 1 ou 2. Parmi les opérateurs du premier ordre les plus courants, on peut citer ceux de Prewitt [9] et de Sobel [10] et l'opérateur de Kirsch [11]. Ces opérateurs sont sensibles au bruit et nécessitent l'utilisation d'un seuil heuristique. D'autres opérateurs combinent des opérateurs de lissage avec ceux de dérivation. Ces derniers ont l'avantage d'être rapides en temps de calcul [8].

Les opérateurs du second ordre localisent les contours par la détermination des passages par zéro de l'opérateur Laplacien ou Laplacien-Gaussien (l'image est convoluée avec une Gaussienne avant d'appliquer le filtre Laplacien) [12].

Des résultats satisfaisants peuvent être obtenus par ces opérateurs dans le cas des images simples. Néanmoins, ils sont moins précis pour la localisation des contours engendrant des fausses détections pour les images complexes. Pour pallier certains inconvénients, Canny [13], Deriche [14] et Shen et Castan [15] ont formalisé mathématiquement des approches optimales (bonne détection, bonne localisation et non multiplicité des réponses).

2) Méthodes utilisant la modélisation surfacique

Une approximation polynomiale sur l'image intensité ou Laplacien est faite dans ce type de méthodes. L'approche de Huertas et Medioni [16] est connue dans ce type où les contours sont détectés avec une localisation sub-pixellique. Les passages par zéro sont calculés analytiquement sur le polynôme approximant.

3) Approches variationnelles

Au contraire des méthodes précédentes, ces approches ne modélisent pas le contour à détecter. Des courbes élastiques sont déterminées et une énergie est y associée où ses minima locaux correspondent aux contours. C'est le modèle de contours actifs "Snakes"[17]. L'inconvénient majeur de cette méthode est l'initialisation manuelle de la courbe de départ.

b. Croissance de régions

Zucker [5] introduit les premiers travaux relatifs à la croissance de région. Les relations spatiales entre les points et leurs caractéristiques sont prises en considération. L'objectif principal de ce type de méthodes est d'obtenir des régions significatives et homogènes au sens d'un critère donné qui présente dans beaucoup des cas le majeur problème à surmonter : comment définir un tel critère. Les pixels ayant des propriétés communes sont fusionnés. Cette fusion peut se faire à plusieurs niveaux, le plus élémentaire étant le pixel [18].

1) Agrégations de pixels

Ces méthodes procèdent par comparaison des propriétés des points avec celles des voisins et décident du regroupement si ces propriétés sont suffisamment proches (le double critère d'homogénéité et d'adjacence). Des germes sont initialement définis et le processus regroupe les pixels similaires aux germes. Les seuils d'agrégation influent beaucoup sur la qualité du résultat [19] [20].

Dans ce type d'approche, des méthodes de multi-seuillage ou par classification d'attributs peuvent être intégrées. L'agrégation peut être aussi faite par des informations locales (calcul des attributs sur un petit voisinage).

2) Relaxation

La relaxation est une technique permettant de faire évoluer une partition de l'image en régions en n'intégrant que deux types d'informations : un attribut du pixel traité et la classe de ses voisins. Le processus est itératif et converge vers une partition de l'image. Cette technique est utilisée en segmentation d'images uniformes en considérant des histogrammes locaux ou des moments sur une fenêtre de petite taille [21].

c. Méthodes de segmentation par seuillage

La segmentation par seuillage est l'opération qui associe à chaque point de l'image $I(i,j)$ une classe de luminance (étiquette e_i , $i=1, \dots, C$) après comparaison de ses attributs (niveaux de gris) à un ou plusieurs seuils qui peuvent être déterminés globalement ou localement. Une (ou plusieurs) valeur de seuil optimal est recherchée en se basant sur certains critères [22].

1) Approches globales

Dans un tel cas, le seuil est utilisé pour segmenter l'image entière. Plusieurs méthodes existent :

➤ Analyse des concavités de l'histogramme

L'histogramme h des niveaux de gris est la fonction qui associe à chaque

niveau de gris g_i (i compris entre 0 et $NG-1$ où NG est le nombre total des niveaux de gris) son occurrence, i.e $h(g_i)$ le nombre de pixels dans l'image ayant la même intensité [23].

L'analyse des concavités de l'histogramme lorsque les vallées sont mal définies, sert à déterminer les seuils correspondant aux maxima locaux de la différence de l'histogramme h et son convexe \bar{h} (le plus petit polynôme contenant h).

➔ Méthodes basée sur la théorie d'information

L'entropie de l'histogramme est utilisée dans certains types de ces approches tels que la méthode de Pun et la méthode de Kapur [20] [24].

➔ Méthode basée sur l'analyse discriminante

La méthode d'Otsu [25] segmente l'image en C classes en maximisant un critère de séparabilité. Le seuil optimal qui maximise la variance inter-classes dans le cas de deux classes C_0 et C_1 est fourni par la relation (1.1):

$$\sigma^2(s^*) = \max_{s=1, \dots, NG-1} \{\sigma^2(s)\} = P_s(\mu_0 - \mu)^2 + (1 - P_s) \cdot (\mu_1 - \mu)^2 \quad (1.1)$$

où : P_s est la probabilité qu'un point appartient à C_0 , μ , σ sont la valeur moyenne et l'écart type des intensités de l'image respectivement.

$$p_i = h(g_i), \mu_0 = \frac{\sum_{i=0}^s i \cdot p_i}{P_s} \text{ et } \mu_1 = \frac{1 - \sum_{i=0}^s i \cdot p_i}{1 - P_s} \quad (1.2)$$

➔ Méthodes basées sur les matrices de cooccurrence

Les matrices de cooccurrence ont été introduites par Haralick [26] pour l'analyse de texture. Cette matrice est l'estimée de la densité de probabilité d'avoir deux niveaux de gris g_i et g_j séparés par une distance d dans l'orientation θ . D'une autre manière, c'est la fréquence d'apparition du g_i parmi les 8 voisins de g_j quand $d=1$. Différentes mesures (contraste, homogénéité, ...) sont définis à partir de cette matrice.

➔ Méthodes basées sur la notion des ensembles flous

Les méthodes floues de classification ont été étudiées par plusieurs auteurs [27] [28]. La classification floue permet de moduler la décision finale d'étiquetage des points en fonction des connaissances a priori sur le contenu de l'image. Un pixel appartient à plusieurs classes avec différents degrés avant la décision finale optimisant un critère. Parmi les algorithmes de cette classification, on trouve le "Fuzzy-C-Means" utilisant comme critère la somme pondérée des écarts quadratiques intra-classes [29] [30].

2) Approches locales

Dans ce type de méthodes, un seuil est déterminé pour chaque point dépendant de l'histogramme de son voisinage. L'image est subdivisée en blocs de tailles égales pouvant être chevauchants ou non. Pour chaque bloc, un seuil optimal est calculé en utilisant l'une des méthodes citées précédemment.

Pour remédier aux problèmes créés par les frontières des sous images voisines, deux approches sont développées par Chehikian et Chafik [31].

➔ L'approche multi-résolution

Une étude récente [32] a montré les avantages de l'utilisation de l'information multi-résolution pendant les phases d'analyse d'images satellitaires. L'analyse « multi-résolution » d'une scène permet de tenir compte de toutes les informations disponibles et aussi, de leur complémentarité et leur hiérarchie.

Une convolution avec un filtre gaussien est faite, en premier lieu, pour réduire la fréquence maximale contenue dans l'image. Une séquence de filtrage est, ensuite, faite permettant de produire une suite d'images de différentes résolutions. Une remise à l'échelle est faite, en dernier lieu, pour classifier les points en se basant sur les différents Laplaciens calculés.

➔ L'approche par apprentissage

Cet algorithme est basé sur l'apprentissage d'un seuil local et la classification des pixels est basée sur ce seuil. Cette méthode s'est avérée très

efficace sur des images avec des objets sombres sur un fond clair ou vice-versa. Néanmoins, elle nécessite le choix du contraste minimal [4].

1.2.2. Méthodes adaptées aux images texturées

La segmentation d'images est une tâche extrêmement importante dans les applications d'analyse d'images ou de vision par ordinateur. Une des techniques les plus pointues pour y aboutir est la segmentation par textures. Cette dernière est considérée comme un aspect clé dans le système visuel humain. Dans la littérature, il existe un grand nombre de recherches liées à ce concept telles que la classification, l'analyse de texture et sans doute la segmentation. L'imagerie texturale a trouvé une large application en télédétection, imagerie médicale, contrôle de qualité. ...etc [33].

Malgré tous ces travaux, la texture reste sans une définition exacte. La définition la plus répandue identifie la texture comme étant une région de l'image dont l'observation se traduit par une impression visuelle d'homogénéité pour toutes les transitions possibles à l'intérieur de cette région. Elle est considérée comme une "répétition spatiale d'un même motif dans différentes directions de l'espace" [6].

La texture se manifeste par une information visuelle qui permet de la décrire qualitativement à l'aide des adjectifs suivants: grossière, fine, lisse, granuleuse, régulière, ...etc. Une définition de texture qui est appropriée dans le contexte de la segmentation est que c'est une région où l'ensemble des statistiques locales ou d'autres propriétés locales sont constantes, de variation lente ou presque périodique.

Haralick [26] élargit la définition en décrivant une texture comme un phénomène à deux dimensions: la première concerne la description d'éléments de base ou primitives (le motif) à partir desquels est formée la texture; la deuxième dimension est relative à la description de l'organisation spatiale de ces primitives.

En pratique, on distingue deux grandes classes de textures, qui correspondent à deux niveaux de perception:

- ❖ Les macrotextures qui présentent un aspect régulier, sous formes de motifs répétitifs spatialement placés selon une règle précise telles que la peau de lézard.
- ❖ Les microtextures présentant des primitives "microscopiques" distribuées de manière aléatoire telles que le sable, laine tissée ou l'herbe.

Due à la diversité de textures naturelles et artificielles, un grand nombre d'approches ont été développés, que l'on peut grouper dans cinq catégories: statistiques, géométriques, structurales, basées sur des modèles et basées sur le traitement de signal.

Le but de l'analyse de texture est de formaliser les descriptifs de la texture par des paramètres mathématiques qui serviraient à l'identifier. Dans ce sens, les critères visuels qui ont été retenues pour la texture peuvent être: le contraste, l'orientation, la forme, la finesse et la régularité. Une multitude de méthodes, de variantes et de combinaisons de méthodes ont déjà été proposées dans la littérature et éprouvées en pratique [34].

a. Les méthodes structurales

Ces méthodes considèrent la texture comme un agencement spatial d'un ensemble de motifs de base suivant certaines règles de placement. Elles tiennent compte de l'information structurelle et contextuelle d'une forme et sont particulièrement bien adaptées aux textures macroscopiques. Les étapes d'analyse sont d'abord l'identification des éléments constitutifs, puis la définition des règles de placement. Les deux structures les plus importantes sont les structures de graphe et les structures syntaxiques [35].

b. Les méthodes statistiques

Elles doivent être employées dans le cadre d'une vision microscopique des textures et s'appliquent aux textures à caractère anarchique mais homogène. La texture est considérée comme la réalisation d'un processus stochastique stationnaire. Des paramètres statistiques sont estimés pour chaque pixel de l'image.

Suivant la modalité des images à étudier, la signature la plus discriminante de la texture est à rechercher soit dans des méthodes qui exploitent directement les propriétés statistiques de la texture (matrices de cooccurrences, matrice de longueurs de plages, fonction d'auto-corrélation, modèle de Markov,...etc), soit dans des méthodes qui exploitent les propriétés statistiques à partir d'un plan transformé dans lequel on réécrit l'image de texture (densité spectrale, méthode des extrema locaux, méthodes de transformation de Fourier ou avec des filtres numériques) [35] [36].

c. Les méthodes basées sur l'étude des formes

Elles se trouvent au croisement de la reconnaissance des formes, de la caractérisation de défauts et de l'analyse macrotexturale. Les régions texturales de l'image épousent des formes particulières et peuvent être caractérisées par des paramètres dits de formes [37].

d. Les méthodes spatio-fréquentielles

Les représentations spatio-fréquentielles préservent à la fois les informations globales et locales donc elles sont bien adaptées aux signaux quasi périodiques. En effet, les textures sont des signaux quasi périodiques qui ont une énergie fréquentielle localisée.

Ces méthodes permettent de caractériser la texture à différentes échelles par analyse de leur décomposition dans le domaine de Fourier, et surtout, sur une base d'ondelettes [36] [38].

e. Les méthodes fractales

Du point de vue de la synthèse d'image, les méthodes fractales [39] sont à part car elles permettent de synthétiser des images très proches de la réalité. En analyse de texture, la dimension fractale, qui est une mesure du degré d'irrégularité d'un objet, décrit une certaine propriété de la texture. Le modèle fractal est basé essentiellement sur l'estimation par des méthodes spatiales de la dimension fractale de la surface représentant les niveaux de gris de l'image.

1.3. Méthodes de segmentation coopératives

Pour remédier aux limites des méthodes citées auparavant, on combine certaines méthodes pour profiter de leurs avantages et compenser leurs inconvénients [40] [41]. Les représentations qui restent les plus utilisés dans les approches coopératives sont celles des contours et des régions. Ces deux types de primitives se fondent respectivement sur des notions de dissimilarité et de similarité des propriétés de l'image. Elles sont donc théoriquement duales et ont des caractéristiques complémentaires.

On trouve dans cette branche, trois approches de coopération : méthodes séquentielles, méthodes parallèles et méthodes hybrides.

1.3.1. Méthodes séquentielles

Chronologiquement, ces méthodes sont apparues pour combiner les méthodes de détection de contours et de croissance de régions. L'extraction des contours est préalablement effectuée pour contrôler le processus de croissance de régions.

a. Coopération contour-région

L'idée est d'introduire un processus de détection des contours à l'intérieur de la croissance de régions pour les utiliser comme des contraintes guidant l'extraction des régions [42]. Une approche connue est celle de Wrobel et Monga [43]. Ils utilisent la fusion de régions en premier lieu et présentent le résultat sous forme de graphe d'adjacence. Un regroupement final des couples de régions voisines est fait en se basant sur un critère dépendant de la variance de la région totale et du pourcentage de points contours sur la frontière commune aux deux régions à fusionner.

Une autre méthode développée [44] adopte une approche multi-résolution pour assurer la robustesse vis-à-vis du bruit. Les étapes de cette méthode peuvent être résumées comme suit :

- Lissage de l'image par une pyramide passe bas,

- Création d'un graphe d'adjacence des régions à partir de la pyramide créée précédemment et après, un autre graphe pour les contours,
- Fusion des informations contenues dans chaque graphe.

b. Coopération contour-modèle surfacique

Dans l'approche de Gombotto [45], la région est modélisée par une surface plane bruitée. L'agrégation des pixels au germe initial (choisi par l'utilisateur) se fait itérativement et un segment (ligne d'épaisseur d'un pixel) est annexé à la région s'il est statiquement proche du modèle de la région. L'arrêt de la croissance est lié à l'information contour qui est le gradient moyen calculé à la frontière de la région en formation.

c. Coopération division-fusion et contours actifs

Pavlidis et Liow [46] proposent une approche qui combine l'algorithme de division-fusion avec une correction des contours par une méthode de déformation des contours. La méthode procède en 4 phases :

- Application d'une méthode de division et fusion par quad-tree fournissant une sur-segmentation,
- Extraction des frontières des régions obtenues,
- Elimination des fausses frontières ; chaque frontière est conservée ou enlevée en fonction de son contraste, ses changements de direction et sa longueur,
- Modification des contours en maximisant une fonction de coût. La position des frontières significatives conservées est corrigée à l'aide d'une approche par contour actif.

d. Coopération multi-seuillage et fusion de régions

C'est une méthode automatique [47] qui combine une classification à partir des histogrammes locaux avec la fusion des régions étiquetées. C'est une méthode non supervisée et donne de bons résultats pour des images texturées.

1.3.2. Méthodes parallèles

Dans cette catégorie, on trouve généralement, la fusion de plusieurs résultats de segmentation de la même image.

a. Coopération contour-région

Une méthode qui combine une détection de contours avec une classification de pixels par CLUSTER, est proposée par Farrokhnia et al [48]. Les différentes étapes sont :

- Calcul d'images attributs à partir des filtres de Gabor,
- Classification des pixels par Clustering,
- Utilisation d'un filtre adaptatif de Canny pour extraire les contours de l'image attributs,
- Intégration de résultats

L'inconvénient de cette méthode est le coût élevé en temps de calcul.

Une autre méthode proposée par Bonnin et Zavidovique [49] fait extraire les contours en premier lieu. Une première segmentation en régions est, ensuite, effectuée où les frontières des régions sont obtenues par fusion de pixels adjacents à l'intérieur des régions. Une confrontation des résultats est faite, en dernier lieu, pour éliminer les points contours non significatifs.

b. Extraction des régions et des contours par matrice de cooccurrence

Le partitionnement statistique de la matrice de cooccurrence est exploité par Haddon et Boyce [50]. Ils définissent deux types de transitions de niveau de gris dépendant de la situation vis-à-vis de la diagonale principale de la matrice. Ces transitions sont utilisées pour classer les pixels de l'image comme points régions ou contours. La pertinence de la classification issue est ensuite vérifiée dans une seconde étape sur la base d'un critère local utilisant une mesure d'entropie.

c. Fusion de résultats de segmentation

La théorie des ensembles flous introduite par Zadeh [29] consiste à quantifier un ensemble par un degré d'appartenance s'interprétant comme une probabilité de l'évènement correspondant. L'inconvénient de cette méthode est le choix de la fonction d'appartenance.

Une approche générale d'intégration de résultats de plusieurs segmentations a été suggérée utilisant le principe de segmentation floue [51].

1.3.3. Méthodes hybrides

L'approche fusionnant des résultats issus par des approches de détection des contours et par approches d'extraction des régions, est dite hybride. On peut citer quelques méthodes :

a. Fusion de plusieurs segmentations

Chabbi et Masini [52] proposent une méthode opérant par fusion des informations extraites par des détecteurs des régions et des contours, pour extraire des facettes 2D. La combinaison est faite comme suit :

- Regrouper les segments extraits en chaînes connexes sans contrôler la formation des régions,
- Pour chaque chaîne connexe, plusieurs segmentations en régions sont envisagées,
- Validation des chaînes connexes (chacune n'englobe qu'une seule région et ne traverse aucune autre région),
- Contrôler l'extraction de régions par les contours et, ensuite, faire une fusion pour obtenir le résultat final.

b. Coopération active

Bajcsy [53] propose une coopération alternative en boucle fermée jusqu'à ce que le résultat soit satisfaisant. Plusieurs paramètres de contrôle sont fournis pour la détection des contours (filtre de Canny) et des régions (par croissance). Le

résultat obtenu pour chaque itération est utilisé dans l'ajustement des paramètres à l'itération suivante.

c. Coopération multi-agents

Chaque agent est un petit système expert autonome lié aux autres agents. Baujard et Garbay [54] proposent un système de vision complet servant à combiner les contours détectés par le détecteur de Deriche avec les régions extraits par la technique de classification de Fisher. Un agent est dédié à chacun des processus et l'analyse et la comparaison des résultats est faite par un autre agent spécifié.

Une autre illustration de ce type de méthode [40] utilise plusieurs agents qui coopèrent de manière interactive utilisant l'estimation du bruit et celle des paramètres de la texture. Cette coopération s'appuie surtout sur une architecture dédiée permettant le choix de la méthode appropriée compte tenu du contexte.

d. Coopération adaptative

Salzenstein [55] propose une méthode coopérative supervisée où le choix de la méthode la plus adaptée est conditionné en fonction des caractéristiques de l'image en entrée et aussi dépend du taux d'erreur de la classification, exploitant la vérité terrain.

1.4. Conclusion

Au terme de cet état de l'art sur les méthodes de segmentation, nous avons retenu que pour établir une segmentation précise et fiable, il est nécessaire de faire coopérer plusieurs techniques pour tirer avantage de chacune d'entre elles tout en palliant leurs insuffisances. C'est dans ce sens que se sont orientés nos travaux de recherche.

Notre souci majeur a été de trouver une technique pouvant s'adapter à tout type d'image qu'elle soit uniforme, peu ou fortement texturée. La technique que nous proposons fait l'objet des prochains chapitres.

CHAPITRE 2

PRETRAITEMENT ET IDENTIFICATION DU TYPE D'IMAGE A

SEGMENTER

2.1. Introduction

Dans une application d'analyse, la segmentation est considérée comme une tâche difficile qui donne des résultats très variables selon le type d'image à analyser et la méthode utilisée pour la segmenter. Afin de remédier à la contrainte d'avoir une méthode avec de bons résultats mais restreinte à une application précise, la conception des systèmes possédant la flexibilité et l'adaptabilité au diverses tâches et contextes, est devenue une solution indispensable.

Notre but été d'avoir une méthode non supervisée qui s'adapte au contexte de l'image à segmenter. L'objectif de ce chapitre est de présenter les démarches clés de l'approche de segmentation proposée et les premières étapes élaborées : la réduction des niveaux de gris à manipuler par multi-seuillage et la détermination du type d'image à segmenter qui soit uniforme, faiblement ou fortement texturée.

Avant d'entamer le détail de notre méthode, on présente ci-après un aperçu sur le matériel et les outils utilisés pour son implémentation :

2.2. Environnement du travail

Pour arriver à notre but, un certain nombre d'outils est exploité :

- Images utilisées

La base des images sur laquelle nous avons travaillé est constituée de deux ensembles :

- Le premier ensemble est téléchargé d'un site web [56]. C et ensemble contient des images synthétiques à différentes régions uniformes et/ou texturées comme montre l'exemple suivant (Figure 2.1):

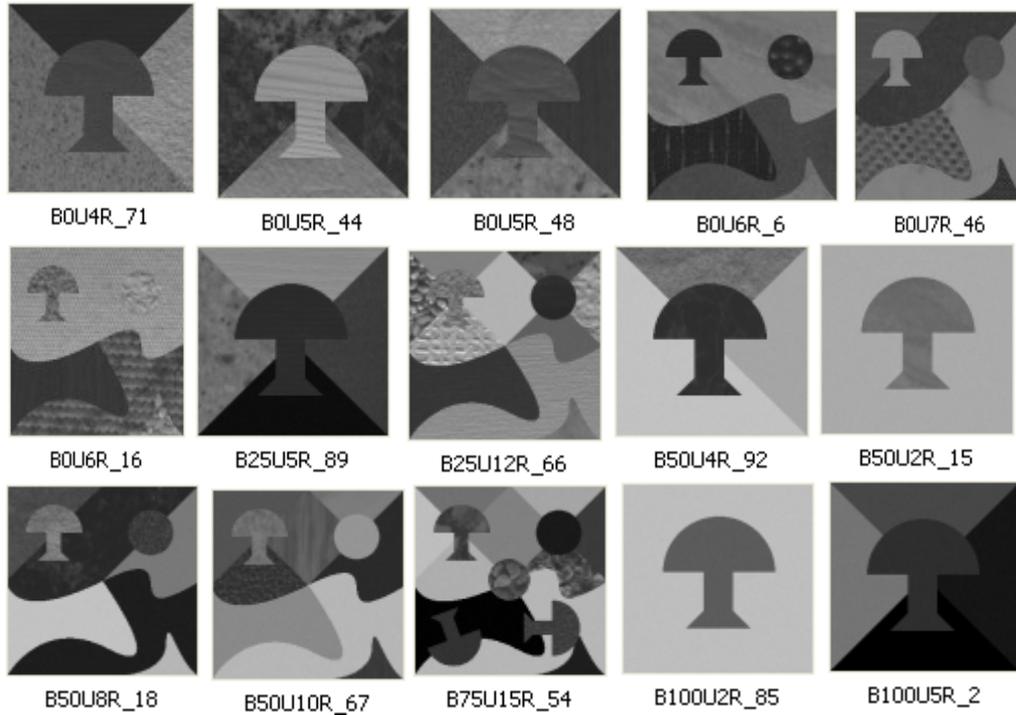


Figure 2.1 : Exemple des images synthétiques utilisées

- Le deuxième ensemble contient des images médicales réelles et d'autres images que nous avons créé avec l'application « *Paint* » en utilisant l'album de Brodatz [57] (Figure 2.2) :

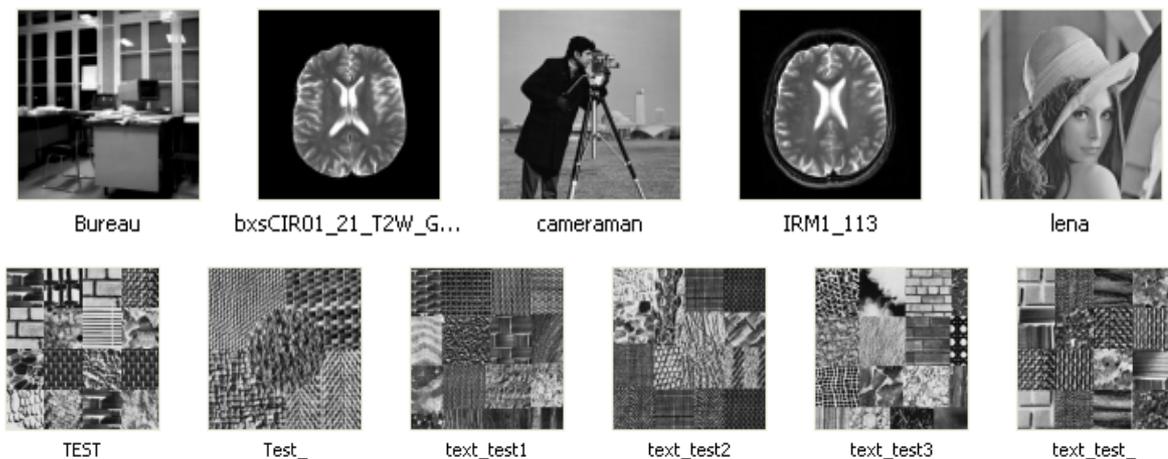




Figure 2.2 : Exemple des images créées et images réelles utilisées

- Matériel utilisé

L'exécution du programme développé est faite sur un PC DELL de 320G de disque dur ayant un processeur Inter Core 2 DUO de 2.1 GHz.

- Logiciel de programmation

Le langage utilisé est le C++ Builder6.

- Interface développée

L'interface contient plusieurs fenêtres de choix des paramètres et visualisation des résultats. La fenêtre principale est illustrée par la Figure 2.3.

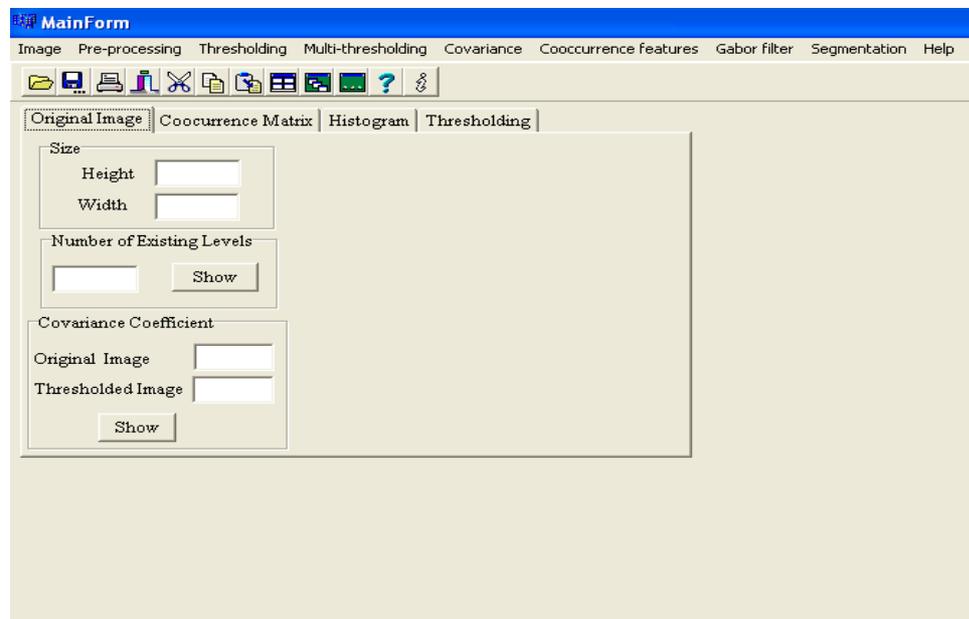


Figure 2.3 : Fenêtre principale de l'interface développée

Le reste des fenêtres vont être présentés au fur et à mesure de la présentation des tâches associées.

2.3. Etapes d'élaboration du système proposé

Nous décrivons, dans ce qui suit, le principe de fonctionnement de la méthode proposée :

- ❖ Une première étape simplifie l'image par une méthode de multi-seuillage afin de minimiser le nombre de niveaux de gris existant dans l'image sans altérer l'information pertinente. Cette étape est très importante dans le cas des images texturées dont le calcul des matrices de cooccurrence devient très coûteux.
- ❖ En suite, un paramètre de décision est calculé pour déterminer le type d'image en cours de segmentation. Ce paramètre a été développé lors de nos tests qui nous ont montré qu'un seul ensemble d'attributs ne peut pas être performant pour tous types d'images.
- ❖ Selon le type d'image, un ensemble d'attributs est choisi. Le calcul est fait directement sur l'image originale pour le cas des images uniformes, alors que l'image multi-seuillée est utilisée pour le cas des images texturées. Pour ce dernier type, les attributs sont issus des matrices de cooccurrence et des filtres de Gabor.
- ❖ Après la sélection des attributs, on détermine le nombre de modes existant dans l'image via l'application de l'algorithme de Mean-shift. Cette étape assure l'obtention d'un système de segmentation non- supervisé.
- ❖ En dernière étape, une classification est faite pour affecter les pixels aux différentes classes déterminées auparavant en adoptant l'algorithme de FCM. Ce dernier est choisi vu sa simplicité d'implémentation et ses résultats robustes. Sa rapidité est assurée dans notre cas par l'initialisation faite à l'aide du Meanshift qui nous donne des centroïdes initiaux fiables.

2.3.1. Organigramme de l'approche proposée

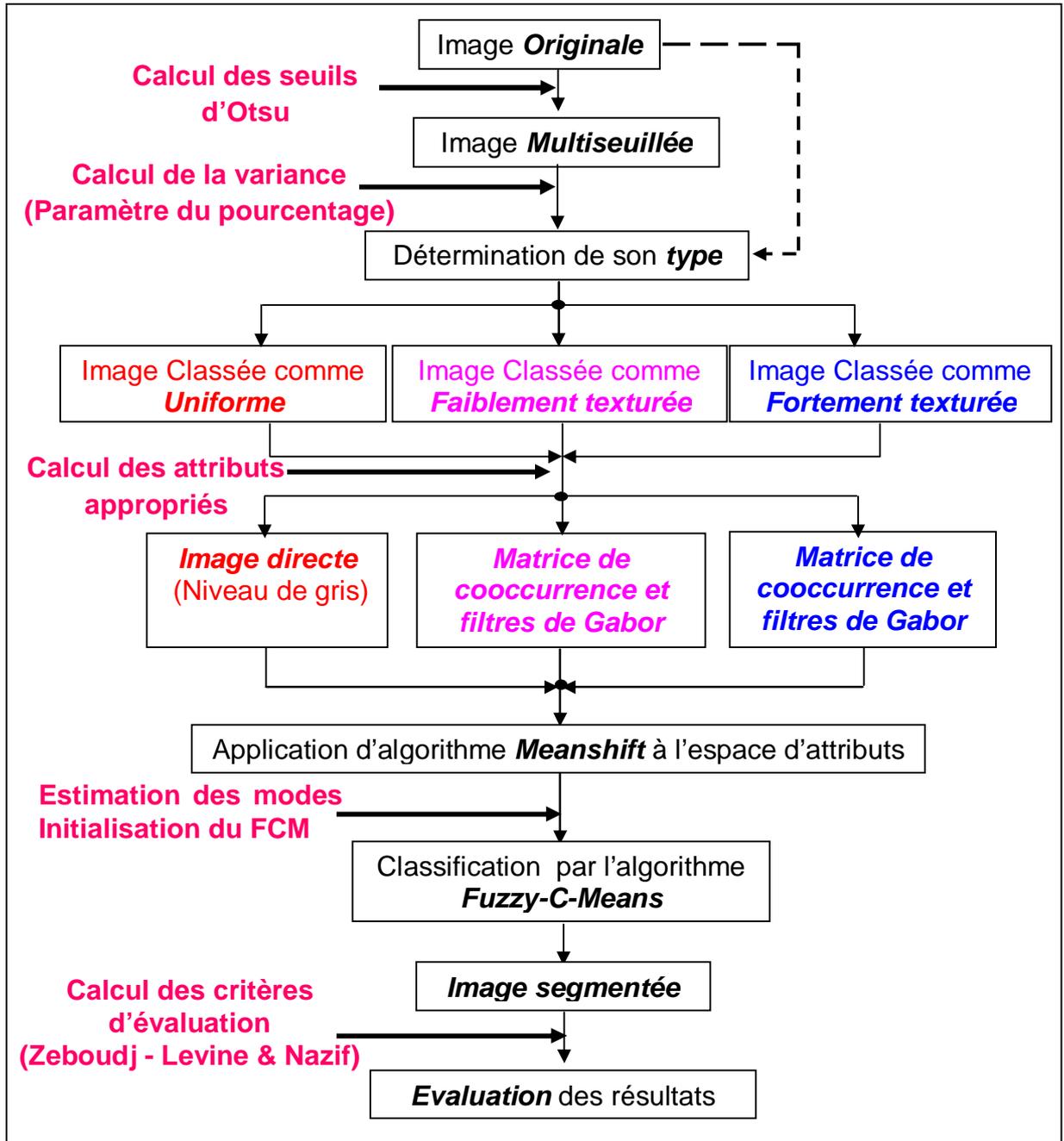


Figure 2.4 : Organigramme de l'approche proposée

2.4. Prétraitements

Afin de développer une méthode de segmentation adaptative et non supervisée, nous avons utilisé un certain nombre d'outils et algorithmes. La première phase élaborée concerne la réduction du nombre de niveaux de gris à manipuler.

Pour mieux mettre en évidence cette étape, on rappelle quelques notions :

2.4.1. Le seuillage

Le seuillage d'une image est la méthode la plus simple de sa segmentation. À partir d'une image en niveau de gris, le seuillage crée une image binaire à deux valeurs.

L'ensemble des techniques qui sont regroupées ici s'appuient sur l'analyse de la distribution des niveaux de gris. On peut considérer qu'elles sont abusivement considérées comme des techniques de segmentation car ne tenant pas compte – pour la majorité d'entre elles – d'une quelconque information spatiale.

2.4.2. Types de seuillage [24] [58]

- ✚ Seuillage simple : L'opération dite de "*seuillage simple*" consiste à mettre à zéro tous les pixels ayant un niveau de gris inférieur à une certaine valeur (appelée *seuil*, en anglais *threshold*) et à la valeur maximale les pixels ayant une valeur supérieure. Il permet de mettre en évidence des formes ou des objets dans une image. Toutefois la difficulté réside dans le choix du seuil à adopter.
- ✚ Seuillage multiple : Par opposition au "*seuillage simple*", il est possible de définir deux ou plusieurs seuils afin de segmenter une image ou de réduire son contenu des variations en niveaux de gris.
- ✚ Seuillage automatique : Le seuillage n'est pas nécessairement une opération qui s'effectue manuellement. La valeur de seuil peut être obtenue de manière automatique. Ce type de seuillage est basé sur l'analyse de la distribution en fréquence (cumulée ou histogramme) des niveaux de gris

associée à l'image. L'objectif des méthodes de recherche de seuil automatique est de détecter deux populations sur l'histogramme, et de proposer la valeur qui permet de les séparer au mieux [59].

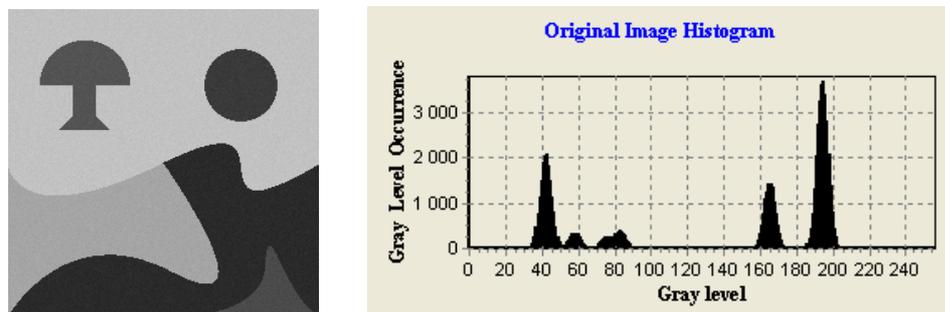
2.4.3. Notion d'histogramme

Avant de présenter le seuil qu'on a adopté, on rappelle que l'*histogramme* d'une image I est la fonction H qui associe à tout niveau de gris g , le nombre de points ayant le même niveau de gris. H est définie sur l'intervalle $\{0, \dots, M\}$ dont M est le maximum que prend g dans I .

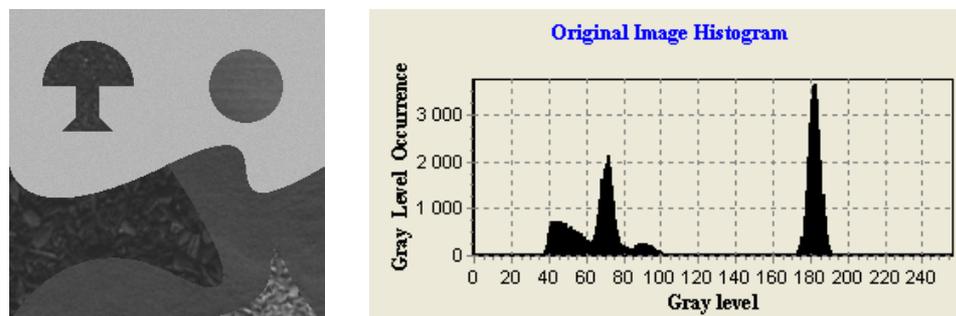
La probabilité d'occurrence d'un pixel de niveau g est:

$$p_x(g) = p(x = g) = \frac{n_g}{n}, 0 \leq g \leq M, \quad (2.1)$$

avec n le nombre total de pixels de l'image, et p_x définit alors l'histogramme normalisé (Figure 2.5).



(a)



(b)

Figure 2.5 : Exemples d'histogramme, (a) « Histogramme d'une image uniforme », (b) « Histogramme de la même image en introduisant des régions texturées »

En analysant ces histogrammes, on peut constater que l'histogramme de l'image uniforme (au sens du niveau de gris) présente généralement des pics séparés identifiant les différentes régions existences, alors que pour la même image en introduisant des régions texturées, l'histogramme devient plus élargi dont les pics se chevauchent vu les multiples variations locales des textures. La connaissance du nombre de régions dans telle image devient plus difficile et ne peut se réaliser par une simple analyse de son histogramme.

2.4.4. Seuillage d'Otsu

Le principe des méthodes de seuillage automatique est d'utiliser des méthodes de traitement des données (maximisation d'entropie, maximisation ou minimisation de la variance inter ou intra-classes...) pour trouver automatiquement la valeur de seuil qui sépare le mieux les objets à partir de l'histogramme des niveaux de gris. La méthode d'Otsu est l'une d'elles. Le nom de cette méthode provient du nom de son initiateur, Nobuyuki Otsu [25].

L'algorithme suppose que l'image à binariser ne contient que deux classes de pixels, puis calcule le seuil optimal qui sépare ces deux classes afin que leur variance intra-classe soit minimale [60].

Rappelons que la moyenne et la variance d'une classe C à n échantillons sont formulées comme suit:

$$\text{- La moyenne } \mu \text{ est donnée par: } \mu = \frac{1}{n} \cdot \sum_{i=0}^n C(i) \quad (2.2)$$

$$\text{- La variance } \sigma^2 \text{ est donnée par : } \sigma^2 = \frac{1}{n} \sum_{i=0}^n (C(i) - \mu)^2 \quad (2.3)$$

σ étant l'écart type de cette classe.

Dans la méthode d'Otsu, le seuil qui minimise la variance intra-classe est recherché à partir de tous les seuillages possibles.

$$\sigma_w^2(t) = w_1(t) \cdot \sigma_1^2(t) + w_2(t) \cdot \sigma_2^2(t) \quad (2.4)$$

les poids w_i représentent la probabilité d'être dans la $i^{\text{ème}}$ classe, chacune étant séparée par un seuil t .

Otsu montre que minimiser la variance intra-classe revient à maximiser la variance inter-classe

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = w_1(t) \cdot w_2(t) \cdot [\mu_1(t) - \mu_2(t)]^2 \quad (2.5)$$

qui est exprimée en termes des probabilités de classe w_i et des moyennes de classes μ_i qui à leur tour peuvent être mises à jour itérativement. Cette idée conduit à un algorithme efficace (Figure 2.6).

Algorithme du seuillage d'Otsu

1. Calculer l'histogramme et les probabilités de chaque niveau d'intensité
2. Définir les $w_i(0)$ et $\mu_i(0)$ initiaux
3. Parcourir tous les seuils possibles $t=1, \dots, \text{intensité max}$
 - Mettre à jour w_i et μ_i
 - Calculer $\sigma_b^2(t)$,
4. Le seuil désiré correspond au $\sigma_b^2(t)$ maximum.



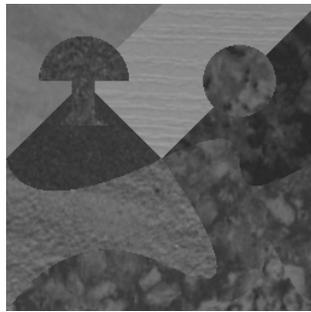
Figure 2.6 : Exemple du seuillage global d'Otsu (seuil= 211)

2.4.5. Multi-seuillage adopté

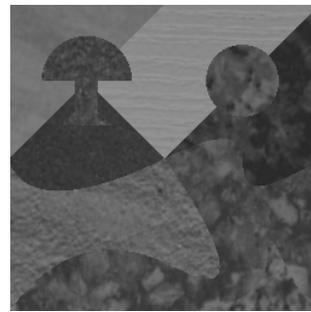
L'extension de la méthode originale pour faire du seuillage à plusieurs niveaux est appelée Multiseuillage d'Otsu « **Multi Otsu method** » [58]. Nous avons choisi d'utiliser le seuil d'Otsu pour son efficacité et sa simplicité de mise en œuvre.

Le multi-seuillage a été sélectionnée pour être la première étape lors de l'élaboration de notre système. Cela est justifié par l'importance du temps de calcul nécessaire pour segmenter une image à plusieurs niveaux de gris et celui pour calculer les matrices de cooccurrence. Le nombre de niveaux de gris à manipuler sera diminué tout en conservant l'information significative.

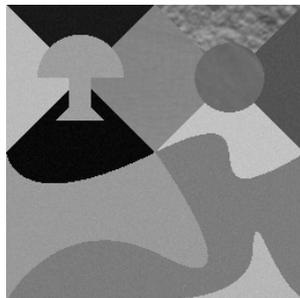
La méthodologie utilisée est basée sur le calcul du seuil d'Otsu sur des blocs non chevauchants de l'image originale. Les seuils résultants sont utilisés pour avoir une image multi-seuillée : les pixels sont affectés selon leurs distances euclidiennes minimales par rapport à ces seuils après avoir éliminé leurs redondances (Figure 2.7).



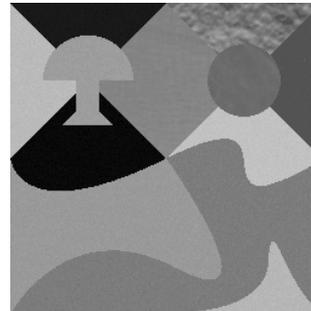
(a) « Image originale (à **123** Ngs) »



(b) « Image multi-seuillée (à **91** Ngs) »



(a) « Image originale (à **169** Ngs) »



(b) « Image multi-seuillée (à **79** Ngs) »

Figure 2.7 : Exemples de multi-seuillage, (a) « Images originales », (b) « leur multiseuillage ».

L'impact de cette étape peut être mesuré en quantifiant la réduction des niveaux de gris dans des images à un nombre élevé. Cette réduction, qui peut dépasser les 50%, est relative et fortement liée au type et contenu de l'image en cours.

Au cours d'achèvement de cette étape, plusieurs tests ont été accomplis pour choisir les paramètres les plus adéquats :

a. Choix de taille de la fenêtre d'analyse

Pour faire une analyse robuste, nous présentons le résultat de multi-seuillage pour la même image 'cameraman' (Figure 2.8). Cette image est très utilisée pour tester la performance d'une méthode de segmentation. Les résultats obtenus nous aide à décider quelle taille doit être appliquée.

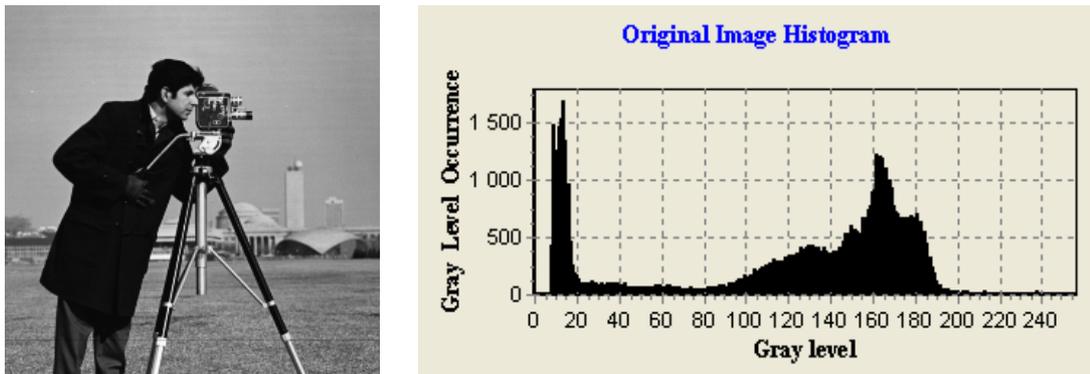


Figure 2.8 : Image originale à **247** niveaux de gris et son histogramme

La fenêtre à utiliser (Figure 2.9) pour faire le choix de taille est la suivante :

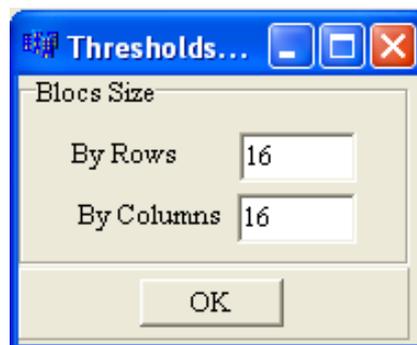


Figure 2.9 : Fenêtre du choix de la taille des blocs

Les résultats de multi-seuillage pour différentes tailles de la fenêtre sur laquelle le seuil d'Otsu est calculé (Figures 2.10 à 2.14), sont présentés ci-après :

➤ Pour une fenêtre 8 x8

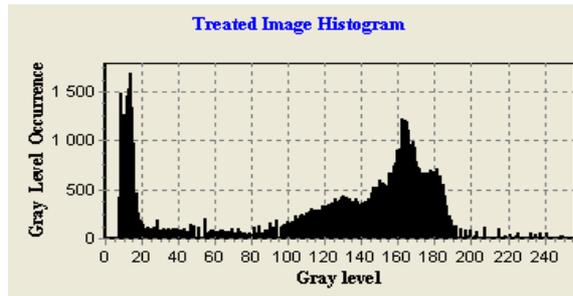


Figure 2.10 : Image multi-seuillée et son histogramme, Nbr de NG= 292

➤ Pour une fenêtre 16 x16

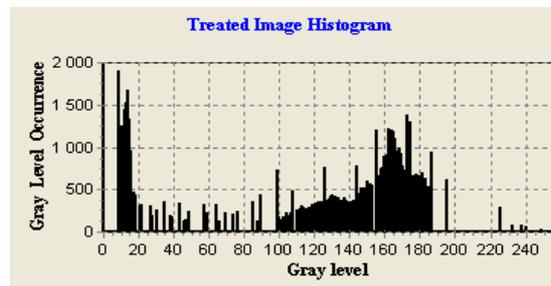


Figure 2.11 : Image multi-seuillée et son histogramme, Nbr de NG= 123

➤ Pour une fenêtre 32 x32

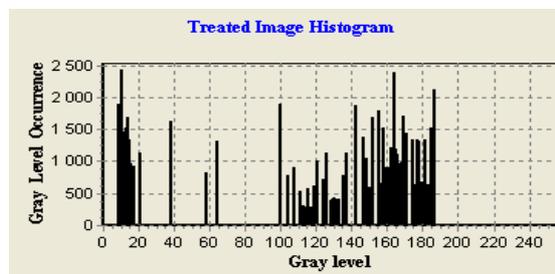


Figure 2.12 : Image multi-seuillée et son histogramme, Nbr de NG= 61

➤ Pour une fenêtre 128 x32

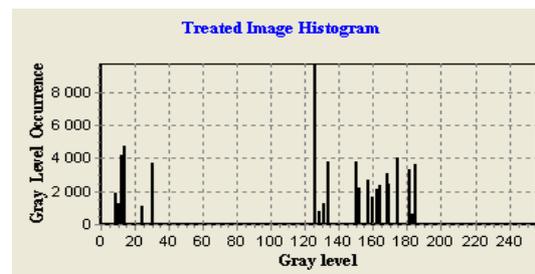


Figure 2.13 : Image multi-seuillée, Nbr de NG= 23

➤ Pour une fenêtre 128 x128

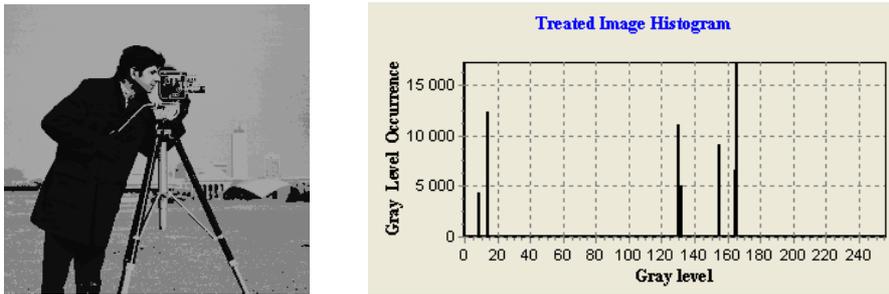


Figure 2.14 : Image multi-seuillée, Nbr de NG= 3

Les seuils obtenus pour cet exemple (Figure 2.14) sont :

165	9	166	155	132	14	130
-----	---	-----	-----	-----	----	-----

Après sélection de différentes images en appliquant la même procédure, nous avons constaté qu'une taille petite de la fenêtre d'analyse produit une réduction faible du nombre de niveaux de gris alors qu'une taille trop grande peut éliminer l'information utile.

Sachant que cette étape ne doit pas altérer l'information pertinente et elle doit bien jouer le rôle de prétraitement qui aide à réduire le temps de calcul et rendre le système plus rapide et puissant, nous avons retenu une taille de **16 x 16 pixels**.

2.5. Identification du type d'image à segmenter

Les statistiques du premier ordre sont estimées sur les niveaux de gris sans prendre en compte leur distribution relative. Beaucoup d'attributs sont proposés tels que les moments et les moyennes d'espace ainsi que les attributs qu'on peut extraire de l'histogramme de niveau de gris [61].

Parmi ces paramètres, la covariance est connue comme l'une des caractéristiques statistiques qui permettent de résumer l'information contenue dans un nuage de points. Si x et y sont deux caractères mesurés sur n individus, \bar{x} et \bar{y} désignent leurs moyennes empiriques, on appelle covariance de x et y , la quantité C_{xy} donnée par:

$$C_{xy} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y}) \quad (2.6)$$

L'idée de base était de trouver un facteur qui possède des valeurs différentes en passant d'une image homogène à une autre hétérogène.

Dans notre travail, nous avons utilisé le terme de la variance " C_{xx} " comme paramètre de décision. L'image est divisée en petits blocs non chevauchants pour lesquels on calcule leur variance. Les blocs sont par la suite affectés à l'une des régions 'texturée' ou 'uniforme' selon un seuil « T » de la variance calculée (Figure 2.13).

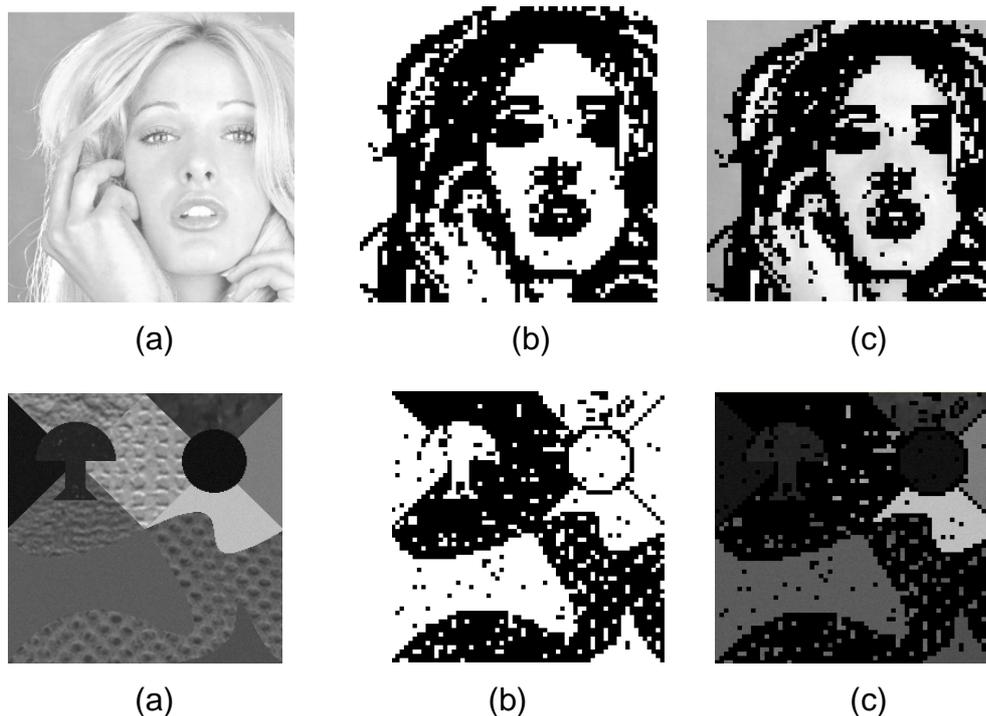


Figure 2.15 : Exemple de localisation des zones uniformes et texturées par mesure de variance (blocs 4×4 pixels et $T= 16$), (a) « Image originale », (b) « Zones uniformes en blanc et texturées en noir », (c) « Superposition des zones texturées à l'image originale »

NB : L'affichage en noir correspond au bloc texturé dans les images (b) et (c).

Lors de nos expérimentations, nous avons découvert qu'un coefficient de **pourcentage « P »** des blocs texturés par rapport au nombre global des blocs dans l'image nous donne une indication sur le type d'image à classer qu'on peut l'affecter dans l'une des trois catégories : **uniforme, faiblement texturée ou fortement texturée** :

2.5.1. Classement de l'image à segmenter

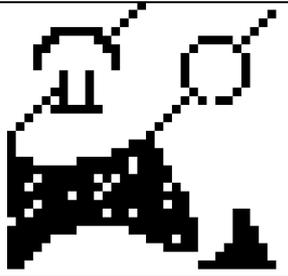
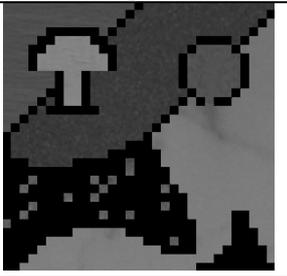
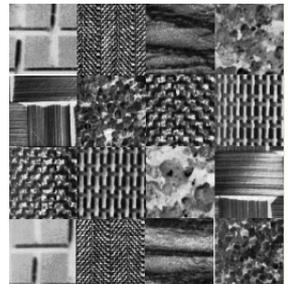
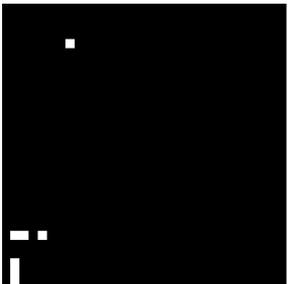
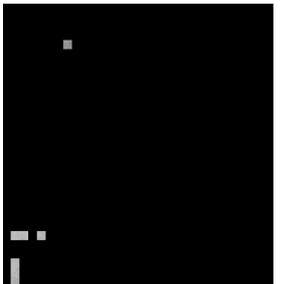
Pour trouver la bonne combinaison (taille, seuil) qui nous mène à une décision convenable, nous avons opté pour la visualisation des images résultantes pour différentes valeurs de seuil et différentes tailles de la fenêtre d'analyse.

a. Choix de taille de la fenêtre d'analyse

Pour avoir un bon compromis entre la valeur de la variance à prendre comme seuil et la taille de la fenêtre d'analyse correspondantes, nous avons fait plusieurs expérimentations dont on présente quelques unes :

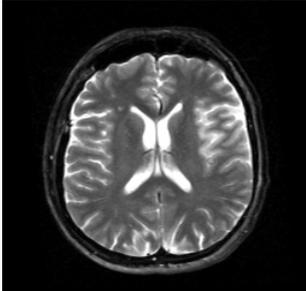
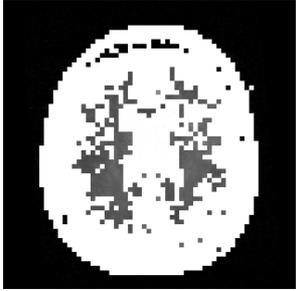
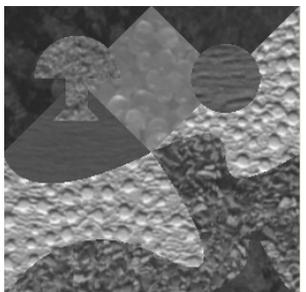
- Pour des blocs à une taille de 8 x 8 pixels

Tableau 2.1 : Localisation des zones texturées et uniformes ($T=45$)

Image originale	Classement selon le seuil T	Superposition du résultat à l'originale
		
		
		

➤ Pour des blocs de 4 x 4 pixels

Tableau 2.2 : Localisation des zones texturées et uniformes ($T=20$)

Image originale	Classement selon le seuil T	Superposition du résultat à l'originale
		
		
		

NB : nous avons inversé l'affichage pour la première image médicale pour visualiser la superposition.

Nous avons conclu à la fin de cette étape qu'une taille petite donne une localisation plus affinée et meilleure. Nous avons choisi la taille de **4 x 4 pixels** et procédé à la fixation du seuil.

b. Choix du seuil pour des blocs de 4 x 4 pixels

En fixant cette taille, nous avons essayé d'analyser la valeur du paramètre P pour chaque cas afin de déterminer la base finale de notre classement (identification du type)

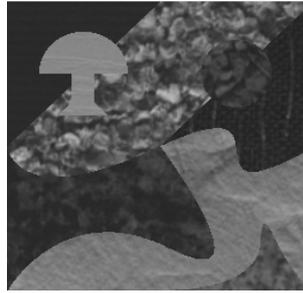
1) Image de test à 8 régions texturées

Figure 2.16 : Image originale

Tableau 2.3 : Localisation des zones texturées et uniformes

Image résultante	Seuil « T »	Pourcentage « P »
	50	29%
	35	34%
	20	46%
	5	76%

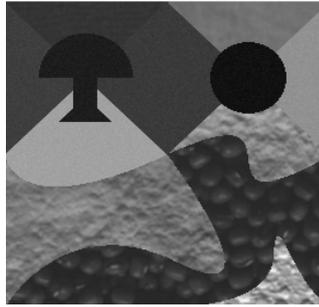
2) Image de test à 4 régions texturées et 8 régions uniformes

Figure 2.17 : Image originale

Tableau 2.4 : Localisation des zones texturées et uniformes pour différents seuils

Image résultante	Seuil	Pourcentage
	60	23%
	35	30%
	15	42%
	3	85%

Examinant les résultats obtenus, on conclut que des valeurs élevées du seuil conduit à une mauvaise localisation des régions texturées surtout pour celles fines ou microscopiques alors que de très petites valeurs marquent des régions uniformes comme texturées. Nous avons retenu la valeur **$T=15$** comme un seuil adéquat à une localisation satisfaisante des zones uniformes et texturées pour notre application.

Remarque

Nous avons appliqué cette procédure aux images originales ainsi qu'à celles multi-seuillées, il n'y a pas de grandes différences à noter. La classification aboutit par notre approche reste la même.

2.6. Conclusion

L'élément important que nous avons retenu lors d'une analyse comparative sur plusieurs images, est que le pourcentage « **P** » peut être un indicateur judicieux du type d'image.

Nous avons retenu un seuil de **$T=15$** calculé sur des blocs de **4×4 pixels** qui nous a donné une bonne classification de l'image en question dans l'un des trois niveaux :

- *Globalement uniforme*, pour $p \leq 14\%$
- *Faiblement texturée*, pour $14\% < p < 60\%$
- *Fortement texturée*, pour $p \geq 60\%$

Cette classification joue un rôle primordial dans le choix du type et nombre d'attributs à utiliser dans les étapes en aval. *Notre segmentation est alors adaptative*.

Ce choix influe considérablement le temps de calcul nécessaire pour arriver à notre finalité.

CHAPITRE 3

SEGMENTATION ADAPTATIVE DES IMAGES UNIFORMES

3.1. Introduction

Après avoir déterminé le type d'image à segmenter, celle classée comme uniforme est segmentée en utilisant l'algorithme du Meanshift. Ce dernier est appliqué à l'espace des descripteurs, c-à-d « image originale directe » pour ce type d'images.

3.2. Mean-Shift

L'algorithme de Mean-shift représente une procédure itérative de montée du gradient qui a comme objet le clustering non supervisé et/ou la détermination des modes.

3.2.1. Aperçu historique

Cette méthode a été proposée par Fukunaga [62] en 1975, dans le but de proposer une estimation intuitive du gradient de la densité d'un nuage de points et de l'utiliser pour les problèmes de reconnaissance de formes.

Elle a été très peu utilisée jusqu'à sa généralisation par Cheng [63] en 1995, mais elle est de plus en plus rencontrée depuis les travaux de Comaniciu et Meer [64] à partir de 1997.

Keselman et Micheli-Tzanakou ont montré dans [65] que cet algorithme était applicable pour l'extraction et la caractérisation de régions d'intérêts dans les images biomédicales. Mayer et coll. [66] utilisent ce paradigme de manière adaptative pour segmenter les IRM cérébrales. Jimenez-Alaniz et coll. proposent dans [46] de faire une estimation de densité non-paramétrique par un mean-shift conjointement avec une carte d'autres paramètres.

3.2.2. Principe de fonctionnement

Le principe de la procédure meanshift est de trouver **le mode** d'un ensemble de n points multidimensionnels. Le mode d'un nuage de points correspond à un maximum local de sa densité, ce qui implique qu'en ce point *le gradient soit égal au vecteur nul*.

si on note $\{x_i\}_{i=1, \dots, n}$ le nuage des points dans l'espace de dimension d , la densité en tout point $x \in \mathbf{R}^d$ est estimée par la formule suivante :

$$\hat{f}(x) = \frac{1}{n \cdot h^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (3.1)$$

K étant une fonction noyau et h le rayon d'une boule centrée en x (connu comme largeur de bande « Bandwidth »).

La recherche des modes se fait en résolvant itérativement l'équation suivante :

$$\nabla \hat{f}(x) = 0 \quad (3.2)$$

Comme on ne connaît pas la densité réelle du nuage, on ne peut pas calculer directement son gradient. L'idée de Fukunaga est d'estimer le gradient de la fonction de densité avec le gradient de l'estimation de la fonction de densité, c'est-à-dire :

$$\hat{\nabla} f(x) = \nabla \hat{f}(x) = \frac{1}{n \cdot h^d} \sum_{i=1}^n \nabla K\left(\frac{x-x_i}{h}\right) = \frac{2 \cdot C}{n \cdot h^{d+2}} \sum_{i=1}^n (x - x_i) \cdot k' \left[\left\| \frac{x-x_i}{h} \right\|^2 \right] \quad (3.3)$$

La minimisation de l'erreur quadratique moyenne intégrée (l'écart type entre la fonction de densité estimée et la fonction de densité réelle) aboutit à une fonction noyau notée K_E telle que :

$$K_E(x) = \begin{cases} \frac{1}{2} C_d^{-1} (d+2) \cdot (1 - \|x\|^2) & \text{si } \|x\| < 1 \\ 0 & \text{si non} \end{cases} \quad (3.4)$$

qui est connu comme **noyau d'Epanechnikov**. C_d est le volume de la sphère unité de d dimensions. Pour ce noyau, une fonction nommée **profil d'Epanechnikov** est donnée par:

$$k_E(x) = \begin{cases} 1 - x, & 0 \leq x < 1 \\ 0, & x \geq 1 \end{cases} \quad (3.5)$$

Satisfaisant

$$K_E(x) = C \cdot k_E(\|x\|) \quad (3.6)$$

avec C est une constante de normalisation.

Le développement de l'équation (3.3) conduit à définir une nouvelle fonction profile $g(x)$ dont $g(x) = -k'(x)$. Cette définition donne un nouveau noyau $G(X)$ dont la formule est : $G(X) = C \cdot g(\|X\|^2)$ avec C une constante de normalisation.

En introduisant cette fonction à (3.3), on obtient :

$$\widehat{\nabla} f(x) = \frac{2 \cdot C}{n \cdot h^{d+2}} \cdot \sum_{i=1}^n (x_i - x) \cdot g\left(\left\|\frac{x - x_i}{h}\right\|^2\right) = \frac{2 \cdot C}{n \cdot h^{d+2}} \cdot g\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \cdot \left[\frac{x_i \cdot g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)} - x \right] \quad (3.7)$$

Le terme donné par

$$m_{h,G}(x) = \left[\frac{x_i \cdot g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)} - x \right] \quad (3.8)$$

est connu par **le vecteur mean-shift**. Cette équation correspond à la moyenne des points à l'intérieur d'une hyper-sphère de rayon h centrée en x .

La recherche des modes à l'aide du meanshift revient à itérer une procédure ascendante du gradient de la densité estimée à partir de chaque point échantillon des données dans l'espace d'attributs, jusqu'à convergence. Lors des itérations, chaque point est translaté vers une nouvelle position en se basant sur la moyenne de tous les points appartenant à une hyper-sphère de rayon h centrée à ce point.

Le schéma suivant (Figure3.1) illustre une étape de ce processus :

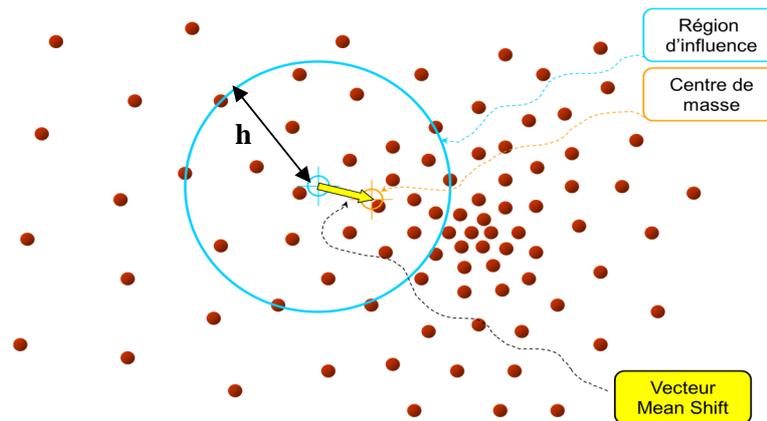


Figure 3.1 : Algorithme du Meanshift

Si on note par $\{y_i\}_{i=1,2,\dots}$; la séquence des locations successives du noyau \mathbf{G} , la formule du nouveau centre est donnée par:

$$y_{i+1} = \left[\frac{x_i \cdot g\left(\left\|\frac{y_i - x_i}{h}\right\|^2\right)}{g\left(\left\|\frac{y_i - x_i}{h}\right\|^2\right)} \right] \quad (3.9)$$

Les points stationnaires de cette procédure ($y_{i+1} - y_i = 0$) sont les modes de la distribution du nuage considéré. En outre, les points associés au même mode appartiennent approximativement au même cluster.

Algorithme

Les étapes du processus général de segmentation que la recherche des modes soit séquentielle ou parallèle pour tous les points échantillons, sont les suivantes :

1. Considérer les images en termes de caractéristiques (via couleur, gradient, mesures de texture, etc.)
2. Choisir une répartition uniforme des fenêtres de recherche initiales (si l'application est parallèle)
3. Calculer le centroïde des données pour chaque fenêtre.
4. Centrer la fenêtre de recherche sur le centroïde de l'étape 3.

5. Répéter les étapes 3 et 4 jusqu'à convergence.
6. Fusionner les fenêtres se trouvant au même point final.
7. Grouper les données traversées par les fenêtres fusionnées.

3.2.3. Emploie du Mean-shift dans l'approche proposée

Cet algorithme a prouvé sa puissance et son efficacité pour atteindre de bons résultats selon la finalité de son emploi. La principale caractéristique de cette procédure qui nous a attiré est cette convergence assurée vers les maxima locaux pour un nuage de points multidimensionnels.

Nous avons utilisé cette méthode en considérant les paramètres définis dans les deux domaines **espace** « **Space** » (coordonnées du pixel) et **valeur** « **Range** » (attributs). Dans ce cas, l'expression du noyau devient:

$$K_{h_s, h_r}(x) = \frac{C}{h_s^2 h_r^{d-2}} k \left[\left\| \frac{X_s}{h_s} \right\|^2 \right] k \left[\left\| \frac{X_r}{h_r} \right\|^2 \right] \quad (3.10)$$

où : X_s est la partie espace du chaque pixel et X_r est sa partie attribut, h_s et h_r sont les valeurs des rayons d'influence (largeurs de bande) dans ces deux domaines successivement. Dans notre cas, h_s est un nombre **impair** afin d'avoir une fenêtre centrée au pixel concerné [67].

Deux emplois de cet algorithme ont été fondé dans notre travail. Nous avons l'utilisé :

- Pour une **segmentation directe** des images uniformes pour lesquelles il a conduit à des bons résultats (Figure 3.3).
- En revanche, pour les images faiblement ou fortement texturées, l'algorithme montre que le résultat de segmentation par le meanshift seul ne suffit pas. A cet effet, on a admis que l'utilisation de ce dernier doit restreindre à **l'estimation du nombre de modes** et la détermination de la position des centroides initiaux qui seront employés dans l'algorithme du Fuzzy-C-Means. Cette association avec le FCM permet de réduire le

nombre des itérations nécessaires pour que celui-ci converge et nous aide à contourner le besoin de connaissance à priori du nombre de classes.

3.3. Segmentation des images uniformes

On a opté pour la segmentation d'images classées comme uniformes en utilisant seulement l'algorithme du meanshift et le seul descripteur « *niveau de gris* ». La fenêtre utilisée est montrée par la Figure (3.2)

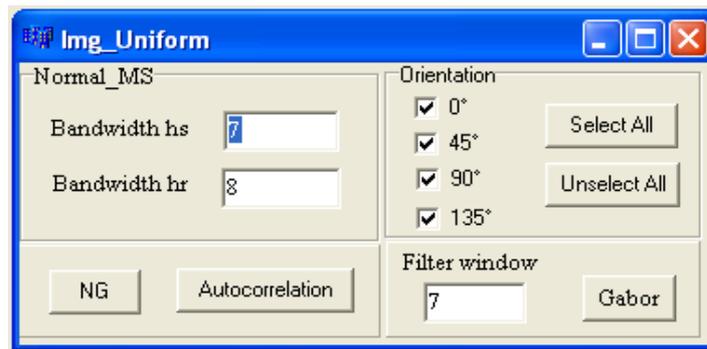


Figure 3.2: Fenêtre de segmentation des images uniformes

Les largeurs de bande à utiliser pour des données normalisées ont leurs valeurs dans l'intervalle [0, 1].

Dans notre application, les données sont multipliées par 100 pour éviter la manipulation de très petites valeurs. A cet effet, nos largeurs de bande ont des valeurs dans l'intervalle [0, 100]. Les exemples suivants (Figures 3.3 et 3.4) présentent quelques aboutissements :



Figure 3.3: Segmentation par le mean-shift ($hs=7$, $hr=8$), (a) «Image originale parfaitement uniforme à 4 régions », (b) « Image segmentée »

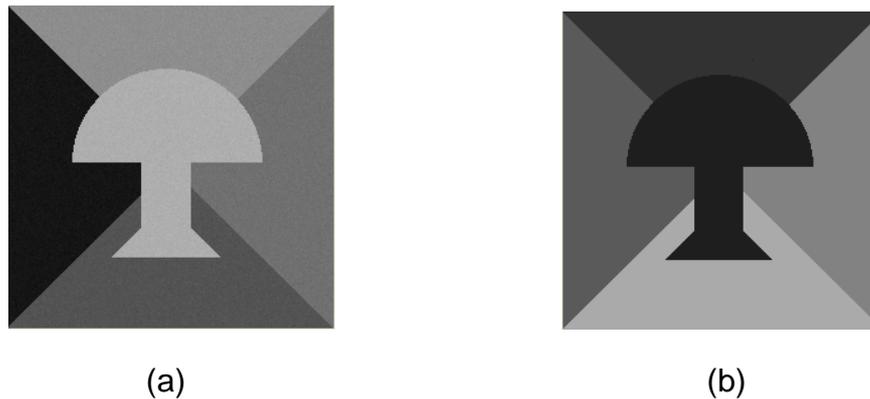


Figure 3.4: Segmentation par le mean-shift ($hs=7$, $hr=23$), (a) « Image originale uniforme bruitée à 5 régions », (b) « Image segmentée »

Nous avons implémenté l'algorithme du meanshift en utilisant **le profile d'Epanechnikov**. Les résultats obtenus mettent en évidence le choix d'un tel algorithme. Une simple classification des différents pixels par rapport aux niveaux de gris déterminés comme centres de classes, à l'aide d'une minimisation de la distance Euclidienne, nous donne des résultats satisfaisants. En revanche, les résultats obtenus ont montré sa sensibilité aux valeurs choisies des largeurs de bandes hs et hr . Le pouvoir de ce processus de trouver le nombre correct des classes est très lié à ces valeurs. Pour éclaircir ce phénomène, on présente l'analyse suivante :

3.3.1. Estimation du nombre de modes

Ce qui est nécessaire à accentuer dans cette étape, est **le sens de la classe** lui-même ! *Est-ce que deux objets séparés ayant la même couleur (niveaux de gris) sont considérés comme deux classes différentes ?*

Dans une image uniforme, le nombre de classe à vérifier revient réellement à la détection de ses différentes couleurs. Le sens des classes vis-à-vis leurs niveaux de gris est ambigu. Une étape d'étiquetage est essentielle pour séparer les différentes entités dans l'image. **On s'intéresse à cette étape au sens pur du niveau de gris.**

Nous avons appliqué le processus du meanshift à des images parfaitement uniformes et autres bruitées. En comparant seulement la similarité de la couleur, les résultats obtenus sont satisfaisants mais l'estimation correcte des modes est

mitigée. Des exemples (Figures 3.5 et 3.6) illustrant cet aspect sont présentés dans ce qui suit :

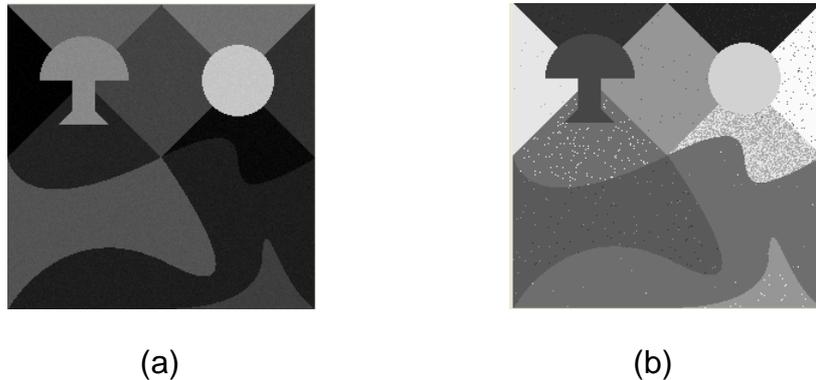


Figure 3.5: Segmentation par le mean-shift ($hs=15$, $hr=13$), (a) «Image originale à 12 régions uniformes bruitées», (b) « Image segmentée »

Remarquons que cette image contient 12 régions distinctes, elle n'a en réalité que **8 niveaux différents**. Sa segmentation par le meanshift ne donne que 10 niveaux différents avec un résultat final acceptable.

Aussi, pour l'image suivante qui contient 5 objets sur un fond, elle n'a que 4 niveaux distincts. Sa segmentation conduit à 4 classes.



Figure 3.6: Segmentation par le mean-shift ($hs=7$, $hr=8$), (a) «Image originale à 6 régions parfaitement uniformes», (b) « Image segmentée »

De plus, pour quelques images, les valeurs détectées comme modes ont dépassé le nombre exact des régions dans l'image originale ce qui montre que certains modes n'ont conduit à aucune région. En contrepartie, le résultat reste acceptable. Ce phénomène est illustré par les résultats (Figure 3.7 – 3.8) ci-après :

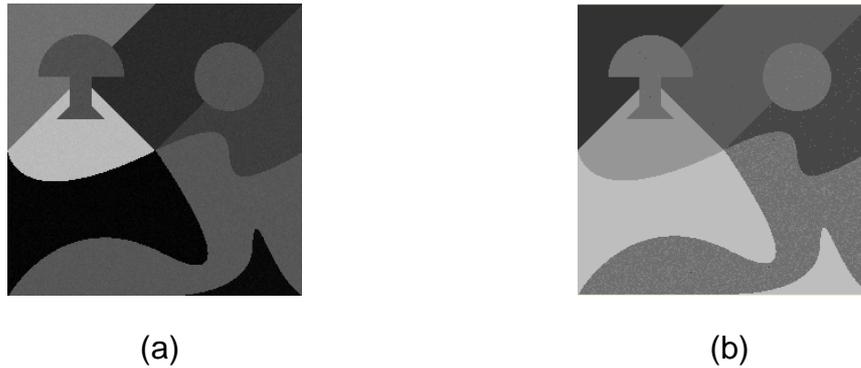


Figure 3.7: Segmentation par le mean-shift ($hs=7$, $hr=13$), (a) «Image originale à 9 régions uniformes bruitées», (b) « Image segmentée »

La segmentation de cette image a aboutit à 10 modes qui est **supérieur à 9** :

0	1	2	3	4	5	6	7	8	9
1,9896	2,0001	2,1205	2,0000	7,4444	29,316	90,555	97,915	133,71	134,85
1,9896	17,815	134,35	189,04	253,35	56,534	57,574	79,834	0,5000	120,04
130,67	112,96	67,360	42,195	80,870	98,901	196,52	212,64	6,0911	26,834

Position du mode (i,j) }
 Valeur du mode →

En remarquant l'image résultat, on trouve qu'il y a 7 niveaux différents seulement dont 3 modes détectés n'ont conduit à aucune région.

Malgré que les résultats paraissent satisfaisants, il est nécessaire de préciser qu'une telle affectation des pixels au mode le plus proche ne peut pas être considérée comme une segmentation qui divise l'image en régions distinctes. Une étape d'étiquetage est nécessaire.

En concluant que le sens du nombre de modes pour ces images est loin de sa propre signification.

A ce stade-là, on retient qu'on ne peut valoriser le pouvoir de détection du nombre correct de modes qu'avec des images ayant différentes textures ou régions parfaitement différentes. Il est incontestable que le niveau de gris seul ne donne le sens de classe seulement s'il n'y a pas de répétition de celui-ci dans l'image à classifier.

En plus, avec cette implémentation en utilisant des largeurs de bande fixes, quelques limitations ont été rencontrées. On les présente dans ce qui suit :

3.3.2. Limitations du Mean-shift

Afin de présenter les insuffisances de l'algorithme, nous l'avons appliqué à la segmentation de quelques images à l'aide d'un seul attribut (le niveau de gris et celui de l'autocorrélation).

a. Effet de largeurs de bande

Il est important de mentionner que tous les résultats obtenus sont liés aux valeurs de hr et hs choisis. Des valeurs trop petites ou trop élevés conduisent à une mauvaise classification. On procède à illustrer l'effet de la valeur hs (taille de fenêtre espace) pour une valeur fixe de hr (largeur de bande pour attributs) et celui de hr pour une valeur fixe de hs . (Figures 3.8 à 3.13):

1) Cas d'une image uniforme bruitée

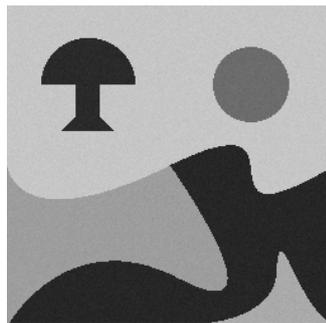


Figure 3.8: Image originale uniformes bruitée à 6 classes

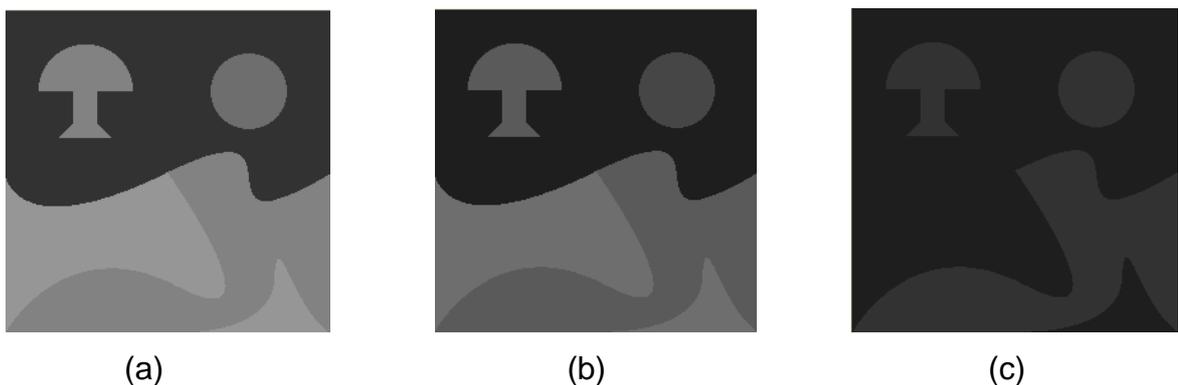


Figure 3.9: Images segmentées avec **valeur fixe de hs** , (a) « Image segmentée à 4 Ngs, $hs=7$, $hr=21$ », (b) « Image segmentée à 4 Ngs, $hs=7$, $hr=45$ », (c) « Image segmentée à 2 Ngs, $hs=7$, $hr=81$ »

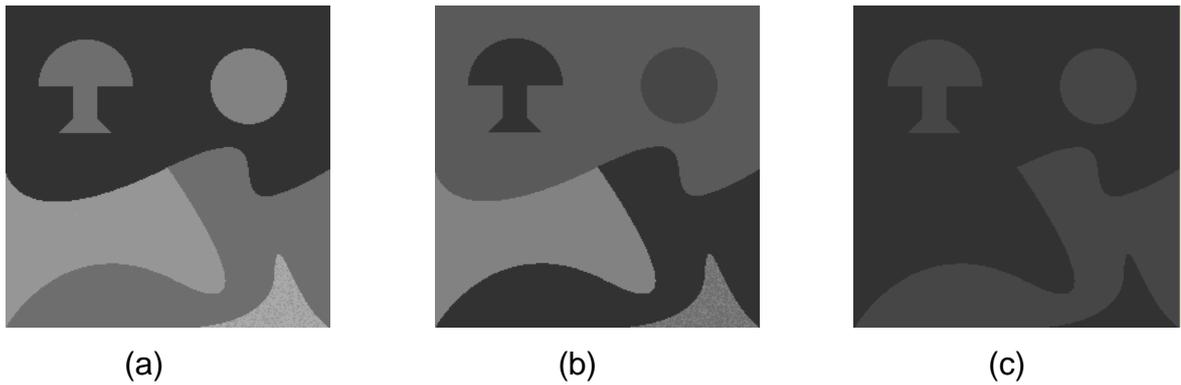


Figure 3.10: Images segmentées avec **valeur fixe de hr** , (a) « Image classifiée à 5 Ngs, $hs=13$, $hr=11$ », (b) « Image classifiée à 5 Ngs, $hs=49$, $hr=11$ », (c) « Image classifiée à 2 Ngs, $hs=75$, $hr=11$ »

2) Cas d'une image parfaitement uniforme

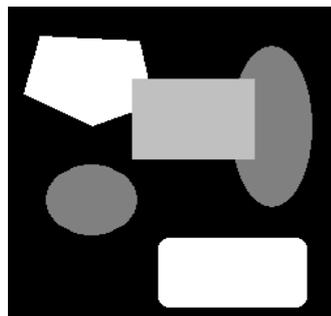


Figure 3.11: Image originale parfaitement uniforme à 6 classes

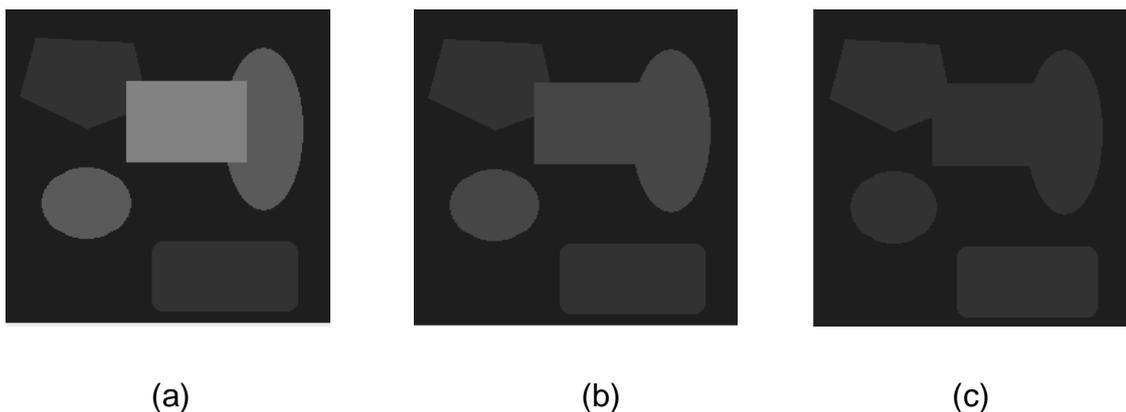


Figure 3.12: Images segmentées avec **valeur fixe de hs** , (a) « Image segmentée à 4 Ngs, $hs=9$, $hr=21$ », (b) « Image segmentée à 4 Ngs, $hs=9$, $hr=67$ », (c) « Image segmentée à 2 Ngs, $hs=9$, $hr=100$ »

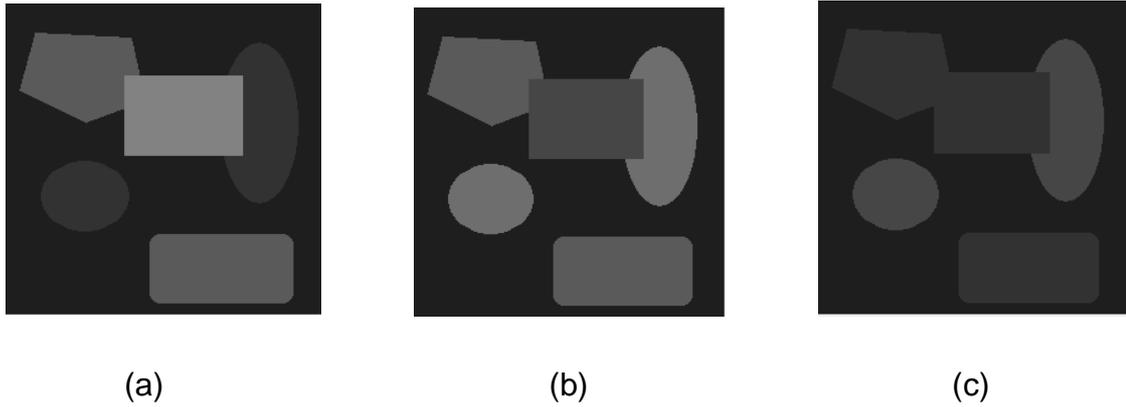


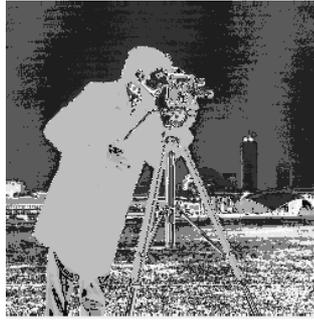
Figure 3.13: Images segmentées avec **valeur fixe de hr** , (a) « Image segmentée à 4 Ngs, $hs=21$, $hr=6$ », (b) « Image segmentée à 4 Ngs, $hs=55$, $hr=6$ », (c) « Image segmentée à 3 Ngs, $hs=75$, $hr=6$ »

Analysons ces différents résultats, on constate que pour chaque changement de hs ou hr ou les deux à la fois, on obtient en général un résultat différent. D'après nos tests, nous avons déduit qu'un changement modéré du hs n'altère pas significativement la segmentation des images uniformes. En outre, un choix adéquat de l'une des largeurs de bande peut stabiliser le résultat malgré l'augmentation de l'autre largeur jusqu'à une borne précise. L'influence de hr est remarquable et agit directement sur la qualité de la segmentation.

Pour mettre en évidence ces conclusions, nous avons segmenté d'autres images classées comme faiblement texturées par le descripteur 'niveau de gris' (Figures 3.14 - 3.15) :



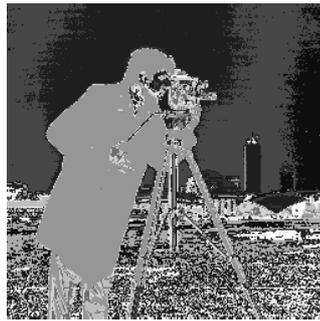
Figure 3.14 : Image cameraman originale



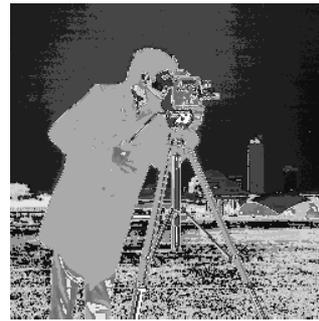
hs=11, hr=13 (14 modes)



hs=11, hr=28 (7 modes)



hs= 15, hr=13 (15 modes)



hs=29, hr=13 (13 modes)

Figure 3.15: Images segmentées avec différentes valeurs de largeurs de bande

Nous donnons ci-après, autres résultats (Figures 3.16 - 3.19), en utilisant l'attribut d'auto-corrélation que nos tests expérimentaux ont révélé son pouvoir discriminatoire notable.

Avec l'image « cameraman », on obtient :



Figure 3.16 : Image originale à droite et image attribut à gauche (normalisée de 0 à 255)



hs= 7, hr=6 (13 modes)



hs=7, hr=13 (7 modes)



hs=7, hr=16 (6 modes)



hs= 5, hr=6 (14 modes)



hs=11, hr=6 (12 modes)



hs=17, hr=6 (12 modes)

Figure 3.17 : Images segmentées avec différentes valeurs des largeurs de bande

Avec une image médicale réelle, on a :

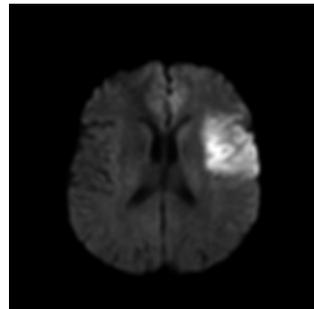
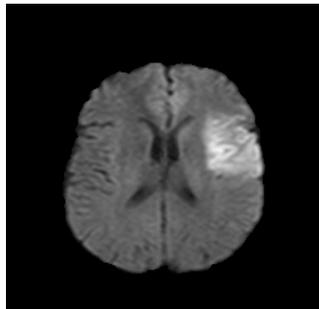
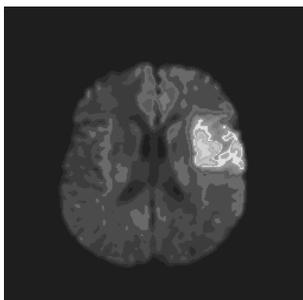
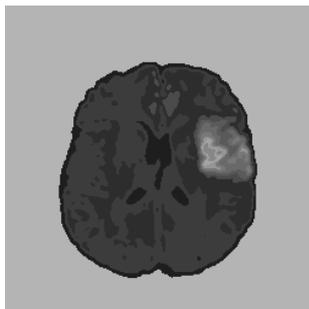


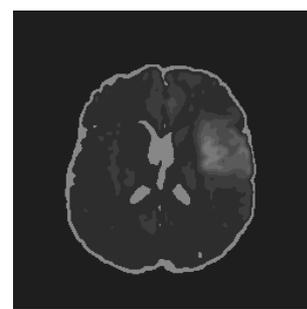
Figure 3.18 : Image originale et image attribut (normalisée de 0 à 255)



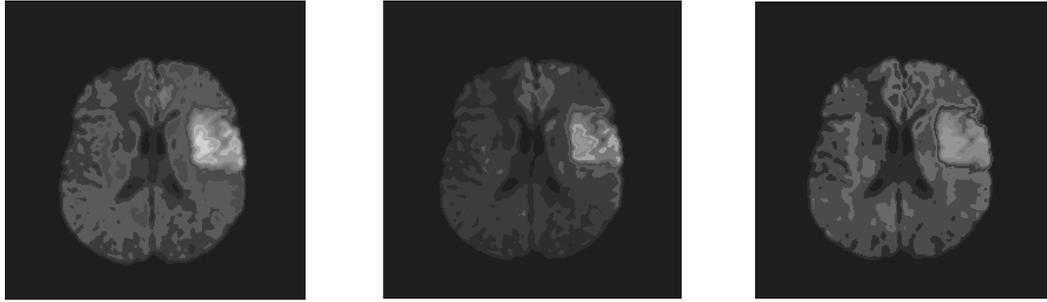
hs= 5, hr=6 (14 modes)



hs=5, hr=9 (11 modes)



hs=5, hr=14 (8 modes)



$hs=7, hr=6$ (13 modes) $hs=11, hr=6$ (10 modes) $hs=17, hr=6$ (9 modes)

Figure 3.19 : Images segmentées avec différentes valeurs des largeurs de bande

Tous ces résultats confirment que la valeur de hr agit sur la finesse du résultat. Le nombre des régions dans les images segmentées diminue quand la valeur de hr ou hs s'élève.

b. Effet du seuil de fusion

L'implémentation de l'algorithme du Mean-shift peut être parallèle ou itérative. Le processus de recherche des modes peut être lancé pour tous les candidats à la fois ou pour un seul candidat par fois. Dans les deux cas, les modes trouvés doivent subir une analyse de ressemblance. En effet, deux itérations différentes ou deux candidats séparés peuvent conduire au même mode. La décision de fusionner ces deux modes selon un critère de similitude est prise si ce critère est inférieur à certain seuil.

Certes, le seuil en question joue un rôle primordial dans l'estimation du nombre de modes. En effet, dans notre implémentation de l'algorithme de mean-shift, deux modes qui ont une distance inférieure à hs dans l'espace coordonnées et inférieure à hr dans l'espace paramètres seront fusionnés. Donc, hr pour ce cas est considéré comme le seuil de fusion. Son effet est déjà présenté dans la section précédente. Un seuil de fusion peut être choisi empiriquement est pris en général inférieur à hr .

3.4. Conclusion

L'implémentation de l'algorithme du meanshift a révélé son utilité malgré ses insuffisances. Le processus assure la non-superviséité de notre approche dont

le nombre de classe est obtenu automatiquement. L'efficacité et la qualité des résultats obtenus dépendent directement du choix des largeurs de bande de l'espace coordonnés hs et l'espace attributs hr .

La variation de hs n'influe pas beaucoup sur le résultat. En revanche, cette valeur affecte le temps de convergence de la procédure, car elle est liée directement au nombre de points couvert par la fenêtre d'analyse et le nombre de fois que l'algorithme doit être appliqué. La valeur de hr , par contre, a une contribution considérable pour avoir une bonne segmentation.

Lors d'établissement de cette phase, la segmentation des images classées comme uniformes à l'aide du meanshift est jugé satisfaisante. En revanche, la segmentation des images faiblement texturées par le descripteur « niveau de gris » a révélé l'obligation de recherche d'autres attributs ou une amélioration au processus de segmentation. Cette déduction met en évidence l'utilisation des autres attributs pour le cas des images texturées et justifie notre choix des matrices de cooccurrence et/ou les filtres de Gabor qui font l'objet du chapitre suivant.

CHAPITRE 4

SEGMENTATION DES IMAGES TEXTUREES

4.1. Introduction

Pour analyser des images réelles riches en détails et contenant une grande variété de régions de nature uniforme et/ou texturée, la caractérisation de texture à l'aide de paramètres pertinents est un outil capital.

Dans un tel cas, les méthodes globales à utiliser sont simples et rapides. Par contre, les méthodes d'analyse locale sont plus précises et plus riches en information. Les méthodes d'analyse globales permettant de décrire globalement la texture et, utilisées en première approximation, permettent d'orienter l'application des méthodes locales en économisant du temps car ces dernières sont coûteuses en temps de calcul.

Dans le chapitre précédent, nous avons retenu qu'il faut avoir des attributs qui caractérisent les différentes textures. En prenant en compte leurs résolutions, nous avons choisi les attributs issus des matrices de cooccurrence et ceux des filtres de Gabor.

Dans ce qui suit, nous donnons et commentons après quelques rappels, les différents résultats obtenus lors de la segmentation des images classées par notre système comme faiblement ou fortement texturées.

4.2. Analyse de texture

Les principales informations dans l'interprétation du message visuel pour un observateur humain sont **les contours et/ou les textures**. De ce fait, l'analyse de texture est utilisée dans des domaines de plus en plus variés. La caractérisation, la segmentation des images ou la reconnaissance des formes représentent ses plus importantes applications.

4.2.1. Notion de texture

Bien que la notion de texture soit naturelle pour l'être humain, elle résiste depuis longtemps à toute tentative de définition. Elle peut être considérée comme une zone de l'image ayant certaines caractéristiques d'homogénéité qui la rende unique. Nous pouvons, aussi, la décrire comme étant un ensemble de primitives de taille et de forme variables, présentant une organisation spatiale particulière [35] [68].

Il existe deux types de textures: les textures périodiques définies par la répétition d'un motif de base et les textures dites aléatoires qui ont un aspect anarchiques mais qui apparaissent homogènes vis-à-vis de leur environnement.

Brodatz a établi un catalogue de texture à des fins de synthèse d'images texturées [69]. Ce catalogue (Figure 4.1) constitue aujourd'hui une référence en matière de base de test pour les chercheurs travaillant sur la texture.

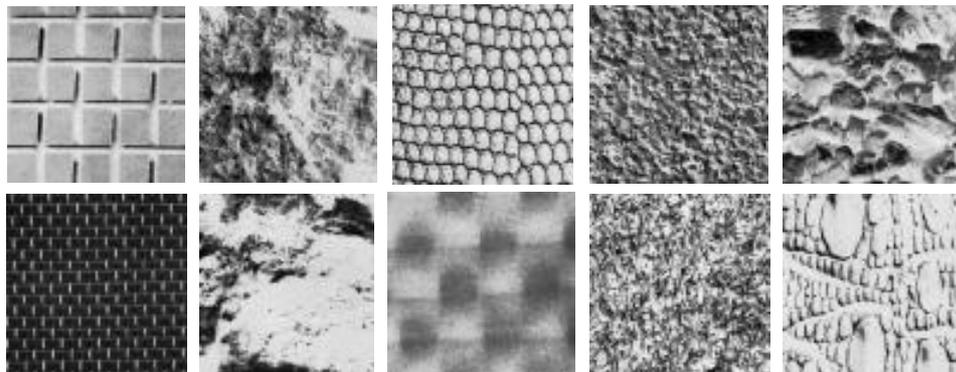


Figure 4.1 : Exemples de textures (10 premières extraites de l'Album de Brodazt)

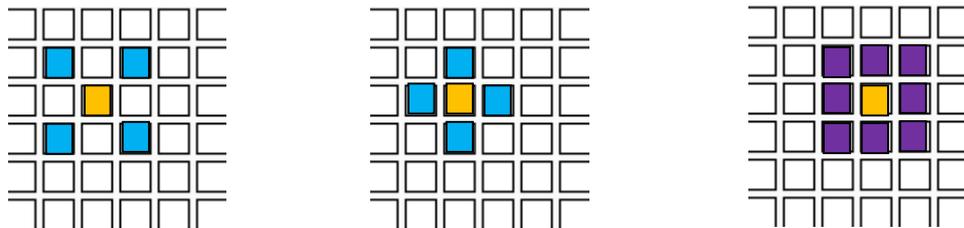
4.2.2. Notion de voisinage

Le traitement d'image s'appuie fondamentalement sur des traitements à l'aide de voisinages. Cela signifie que les traitements effectués en un endroit donné correspondant à un pixel dépendent non seulement de ce pixel mais aussi de pixels appartenant à son voisinage.

Considérons un pixel P dont la position dans l'image est donnée par les coordonnées (x, y) . Un voisinage de P , noté $V(P)$, se définit comme un ensemble de pixels P' connectés à P .

Plutôt que de s'étendre sur la notion de connexité, nous présentons ici des schémas (Figure 4.2) illustrant les deux cas les plus couramment choisis :

- ✓ un voisinage de « 4-connexité » : le pixel central n'a que quatre pixels voisins, chacun d'eux étant à distance unité ;
- ✓ un voisinage de « 8-connexité » : le pixel au centre a huit pixels voisins, chacun d'eux étant également à distance unité.



Voisinage 4-connexité

Voisinage 8-connexité

Figure 4.2: Illustration des connexités

4.2.3. Effet d'échelle

Parmi les particularités des textures sont leurs apparences locales [69] et leurs organisations des détails que le système visuel humain est capable d'identifier et de différencier d'une façon précise. Pour bien identifier les textures, on a souvent besoin de travailler à différentes échelle [36]. Le lien entre la texture et sa résolution est par conséquent très important, direct et ne peut pas être négligé.

4.2.4. Techniques d'analyse de texture

L'analyse de texture regroupe un ensemble de méthodes permettant de quantifier les différents niveaux de gris présents dans une image en termes d'intensités et de distribution dans le but de calculer un certain nombre de paramètres caractéristiques de la texture à étudier. Cette analyse contient deux groupes de méthodes :

➤ Les méthodes structurelles

Permettent de décrire la texture en définissant les primitives et les

"règles" d'arrangement qui les relient. Elles sont donc applicables aux textures aléatoires.

➤ Les méthodes statistiques

Visent à étudier les relations entre un pixel et ses voisins et définissent des paramètres discriminants de la texture en se basant sur des outils statistiques. Généralement, elles sont utilisées pour caractériser des structures fines, sans régularité apparente. Plus l'ordre de la statistique est élevé et plus le nombre de pixels mis en jeu est important.

4.3. Descripteurs de texture

Depuis les travaux de Haralick en 1973 sur la texture, plusieurs approches ont été proposées [36] [70]. Cependant, la recherche de paramètres discriminants caractérisant la texture et l'utilisation de ces paramètres pour la segmentation des images texturées restent encore un problème délicat, sans solution universelle.

Les descripteurs recherchés sont exprimés, en générale, sous forme paramétrique. Il est évident que le choix des paramètres dépend surtout de l'application considérée. Les attributs souvent utilisés pour caractériser une texture, sont issus de l'histogramme de l'image, de l'image gradient, de la matrices de cooccurrence, de la matrice des longueurs de plage [71], des coefficients des filtres de Gabor [70] ou des coefficients de la transformée en ondelettes.

Parmi les méthodes d'extraction de paramètres, nous avons utilisé les matrices de cooccurrence et les filtres de Gabor. La méthode de matrice de cooccurrence est largement utilisée dans le monde du traitement d'image [26]. Elle est connue par sa richesse en information, son popularité et son efficacité prouvée dans plusieurs travaux. Nous l'avons confronté avec la méthode d'analyse multi-résolution basée sur les filtres de Gabor analysant la structure :

- ✓ à différentes échelles (fréquences)
- ✓ à différentes orientations (angles)

Ces filtres sont bien célèbres par leur simplicité d'implémentation et leur capacité de discrimination.

4.3.1. Matrice de cooccurrence

De nombreuses études psycho-visuelles ont été faites sur la discrimination de texture par le système visuel humain. Une conjecture importante et valide dans beaucoup de cas est que l'œil humain ne peut discerner instantanément deux textures dont les statistiques du second ordre sont identiques. Cependant, il existe des cas où des textures ayant les mêmes statistiques du second ordre sont néanmoins discriminables sur la base de propriétés locales.

La matrice de cooccurrence est un outil de description d'images qui a été introduit par Haralick [26]. Cette matrice a été retenue parmi diverses méthodes de segmentation pour sa capacité d'analyse sélective de l'image suivant une direction d'observation. Elle présente une grande simplicité de mise en œuvre et donne de bons résultats sur la plupart des types d'images.

Vue la masse importante d'informations difficile à manipuler, la matrice de cooccurrence n'est pas utilisée directement mais à travers des mesures dites *indices de texture*. En 1973, Haralick *et al* [26] ont proposé quatorze indices, on va présenter les plus utilisés après une formulation mathématique de cet outil.

a. Formulation mathématique

Les matrices de cooccurrence sont analogues à des histogrammes bidimensionnels. Elles représentent le nombre d'occurrence de couples de pixels particuliers dans l'image. Leur principale caractéristique est de s'intéresser à des couples de pixels qui sont, par définition, séparés par une distance d selon une orientation θ .

Une image discrète de dimension $M \times N$, est considérée comme une fonction à deux variables sur un domaine discret D . La fonction p prend ses valeurs dans l'ensemble discret E de K éléments, $E = \{0, 1, \dots, K - 1\}$ et s'écrit :

$p: D \rightarrow p(x, y) \in E$ où les valeurs $p(x, y)$ sont les intensités des pixels de l'image.

La matrice de cooccurrence $M_{d,\theta}(i, j)$ est une fonction paramétrée de $E \times E$ dans N , i.e. de taille $K \times K$.

$$M_{d,\theta}(i,j) = \text{card}(p(x,y) = i \text{ et } p'(x,y) = j) \text{ dont } \text{distance}(p,p') = d \text{ et } \arg(p,p') = \theta \quad (4.1)$$

$M_{d,\theta}(i,j)$ traduit la fréquence d'apparition des niveaux de gris i et j séparés par d selon l'orientation θ . L'image de telle matrice est illustrée dans ce qui suit (Figures 4.3 et 4.4) :

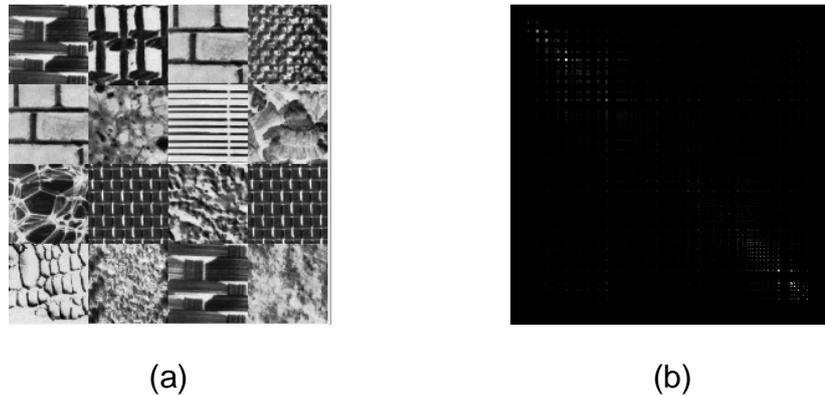


Figure 4.3 : Matrice de cooccurrence ($d=2, \theta=90^\circ$), (a) « Image originale texturée à **125** Ngs », (b) « son image de cooccurrence »

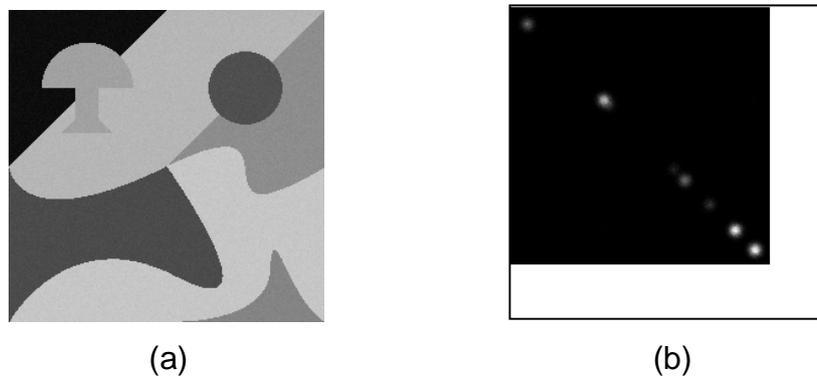


Figure 4.4 : Matrice de cooccurrence ($d=1, \theta=45^\circ$), (a) « Image originale uniforme à **142** Ngs », (b) « son image de cooccurrence »

b. Choix des paramètres (d et θ)

La difficulté à surmonter lors de l'application de cette technique réside dans le choix du vecteur de déplacement et de la taille de la fenêtre du voisinage sur laquelle s'effectue la mesure. Plusieurs études ont été menées pour déterminer une distance ou une orientation optimale. En 1986, Weszka [72] a conclu, qu'en pratique, une distance courte donne généralement de bons résultats.

En ce qui concerne la taille de la fenêtre d'analyse, elle, doit satisfaire deux critères contradictoires à savoir, la plus petite possible pour réduire le risque de mélange de textures différentes et, en même temps, la plus grande possible pour pouvoir extraire des statistiques assez robustes et significatives.

Ce genre de compromis peut être obtenu par une analyse empirique des résultats obtenus pour différentes fenêtres.

Nos nombreux tests (Figure 4.3 et 4.4) ont montré que la distance $d=1$ est largement adéquate. Pour la taille de la fenêtre d'analyse, nous avons adopté une dimension de **7 X 7 pixels** qui nous paraît suffisamment large pour contenir des informations sur la texture en question, et qui de plus, nous donne un temps raisonnable lors de calcul des matrices de cooccurrences et leurs caractéristiques.

On présente quelques résultats (Figures 4.5 à 4.8) de notre système pour ces matrices :

- Pour une image uniforme

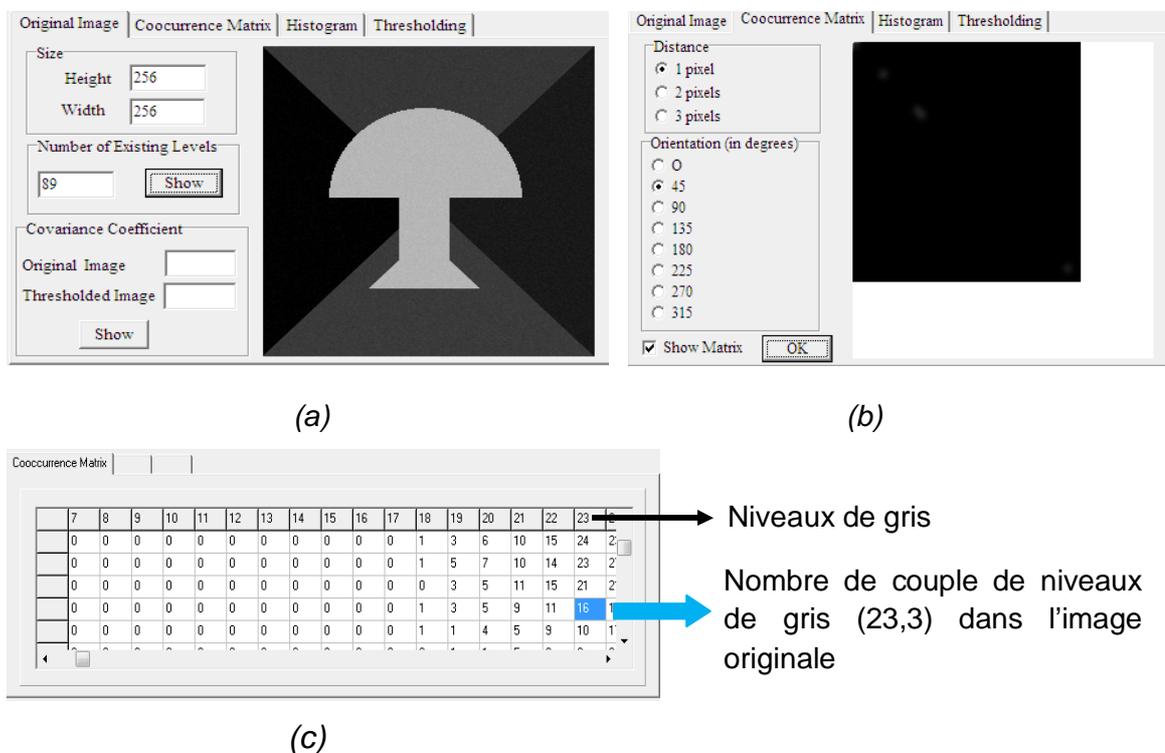


Figure 4.5 : Matrice de cooccurrence ($d=1$, $\theta = 45^\circ$), (a) « Image originale », (b) « Image de cooccurrence », (c) « valeurs numériques de la matrice »

NB : l'affichage numérique des valeurs calculées pour les matrices de cooccurrence est illustré dans des tableaux (Figure 4.5 (c)) dont le nombre de lignes et colonnes égale à la valeur maximale des intensités dans l'image en question.

- Pour une image faiblement texturée

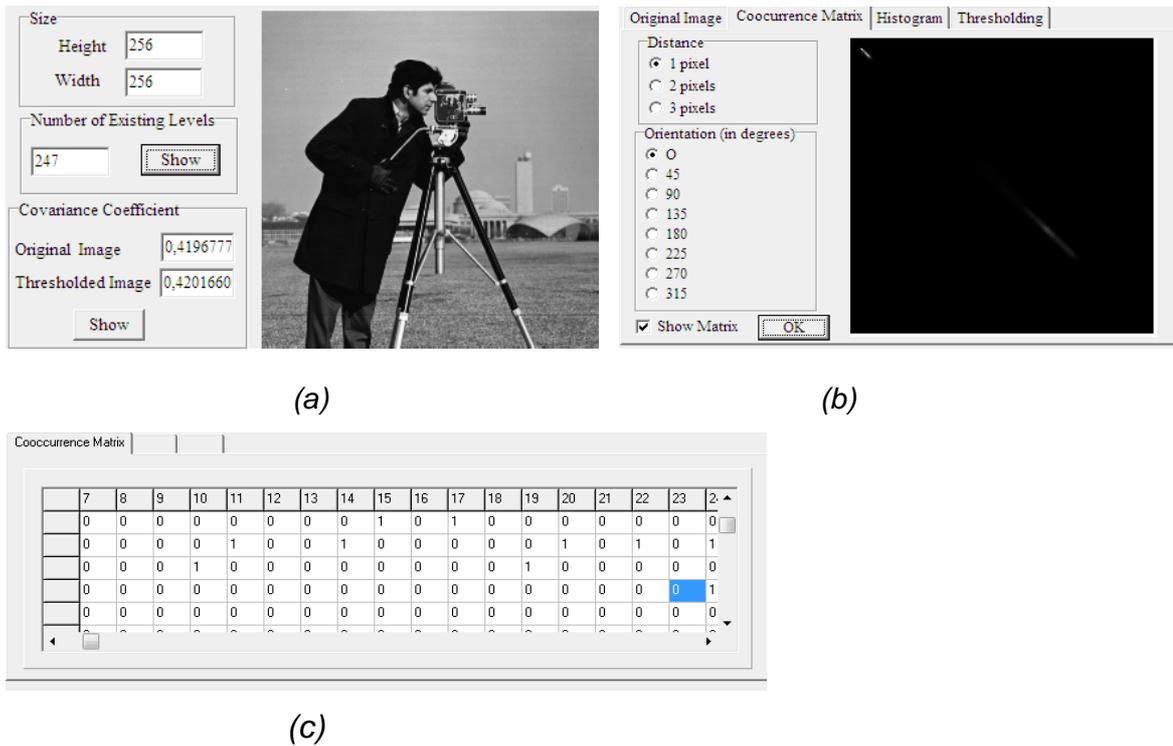


Figure 4.6 : Matrice de cooccurrence ($d=1$, $\theta = 0^\circ$), (a) « Image originale », (b) « Image de cooccurrence », (c) « valeurs numériques de la matrice »

Avec la même image du cameraman (Figure 4.7) et une autre direction :

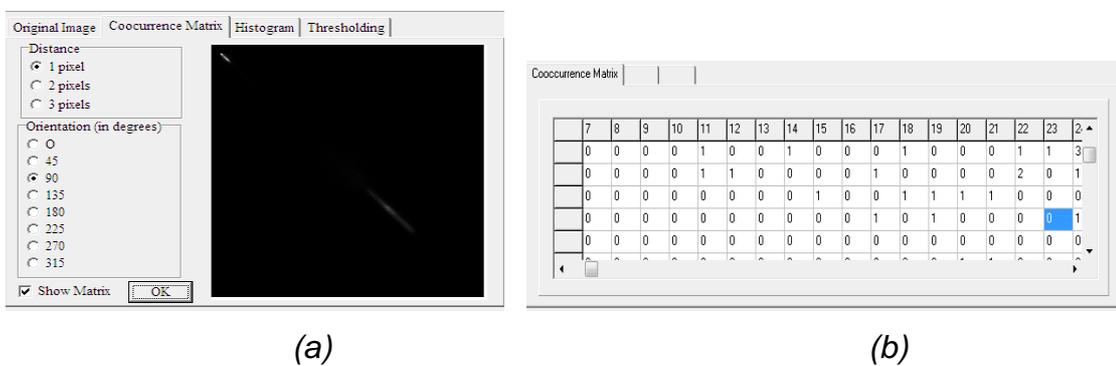


Figure 4.7 : Matrice de cooccurrence ($d=1$, $\theta = 90^\circ$), (a) « Image de cooccurrence », (b) « valeurs numériques de la matrice »

- Pour une image fortement texturée

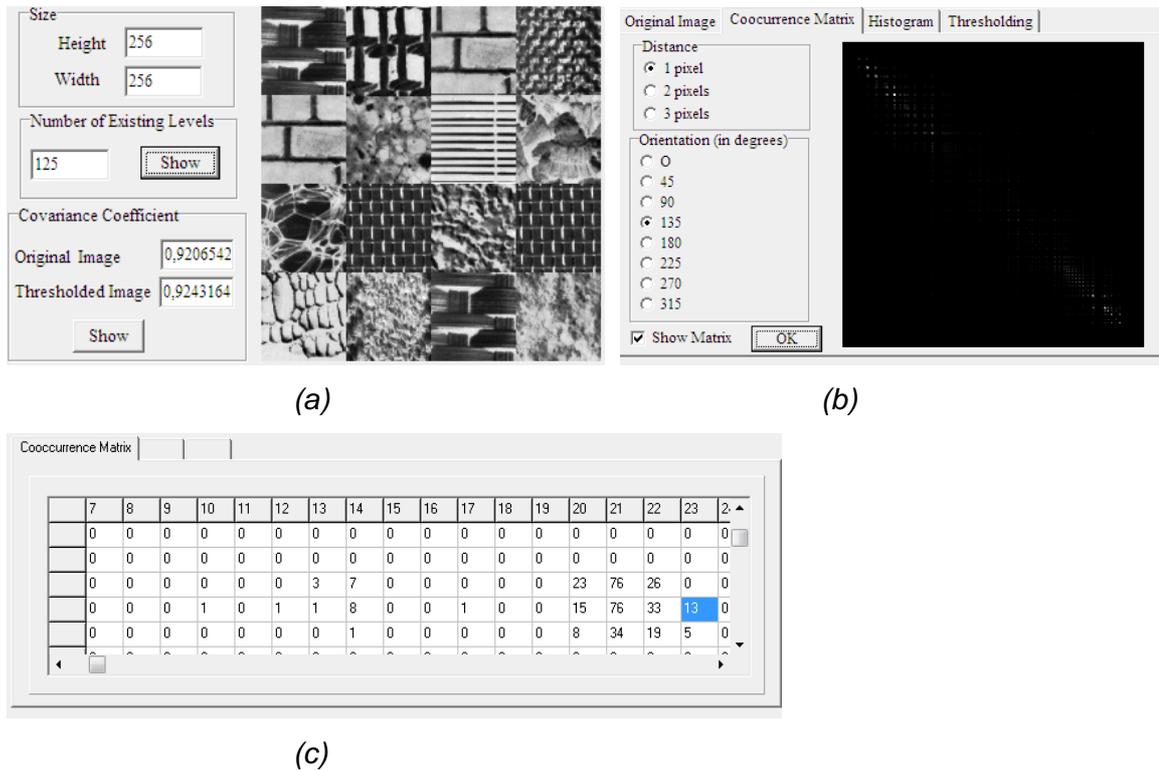


Figure 4.8 : Matrice de cooccurrence ($d=1$, $\theta = 135^\circ$), (a) « Image de cooccurrence », (b) « valeurs numériques de la matrice »

c. Indices de Haralick

Les matrices de cooccurrences contiennent toute l'information texturale, ce qui représente une masse d'informations très importante et difficilement manipulable dans son intégralité. Pour pallier cet inconvénient, des indices extraits de cette matrice ont été définis. Bien que corrélés entre eux, ces indices réduisent l'information contenue dans une telle matrice et permettent une meilleure séparation entre les différents types de textures. Parmi ces indices, on présente les plus utilisés :

- **Contraste** : $Contr = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} |i - j|^2 \cdot M_{d,\theta}(i, j)$ (4.2)

- **Entropie** : $Entrp = - \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} M_{d,\theta}(i, j) \cdot \log(M_{d,\theta}(i, j))$ (4.3)

- **Energie** : $Energ = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} (M_{d,\theta}(i, j))^2$ (4.4)

- Moment des différences inverses:

$$Momdiff = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} \frac{M_{d,\theta}(i,j)}{1+(i-j)^2} \quad (4.5)$$

Dans l'appendice A on se réfère aux autres indices.

A la fin de l'implémentation de son algorithme, la matrice de cooccurrence nous a paru assez coûteuse en temps de calcul. Quand l'image a une marge large en niveaux de gris ou quand l'image a une valeur maximale importante, qui donne une taille considérable de cette matrice, le délai de calcul devient énorme.

Après cette constatation, nous avons cherché une amélioration ou transformation à la définition brute de cette matrice pour réduire ce temps.

4.3.2. Matrice de cooccurrence indexée

a. Fondement

Afin d'améliorer la rapidité lors du calcul des matrices de cooccurrence, nous avons opté pour leur compression. Pour formuler notre idée, on va analyser le cas suivant :

Soit A une image telle que : $A = \begin{bmatrix} 0 & 0 & 2 & 0 \\ 2 & 2 & 7 & 4 \\ 7 & 4 & 4 & 4 \end{bmatrix}$

On remarque que cette image contient une valeur maximale est égale à 7. Le calcul direct de la matrice de cooccurrence, pour une distance et une orientation précises, conduit à une matrice de 8 x 8. Si $d=1$ et $\theta = 45^\circ$, on a :

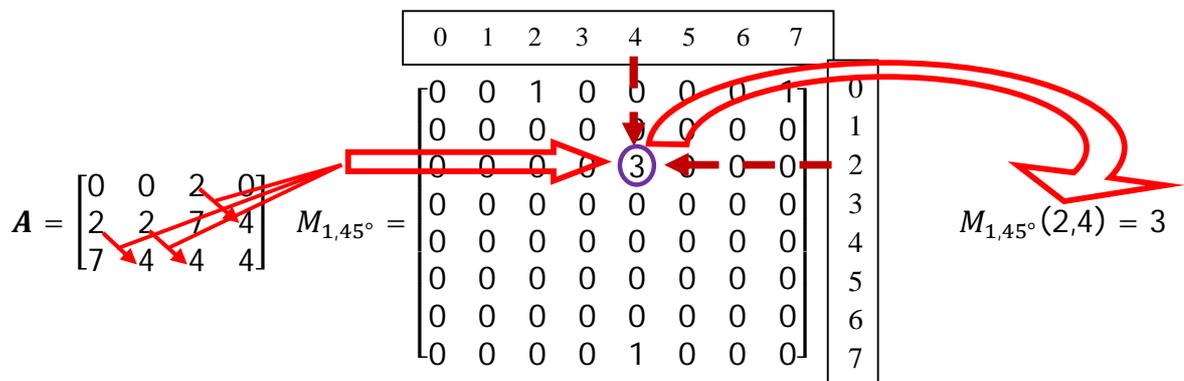


Figure 4.9 : Illustration du calcul de la matrice de cooccurrence

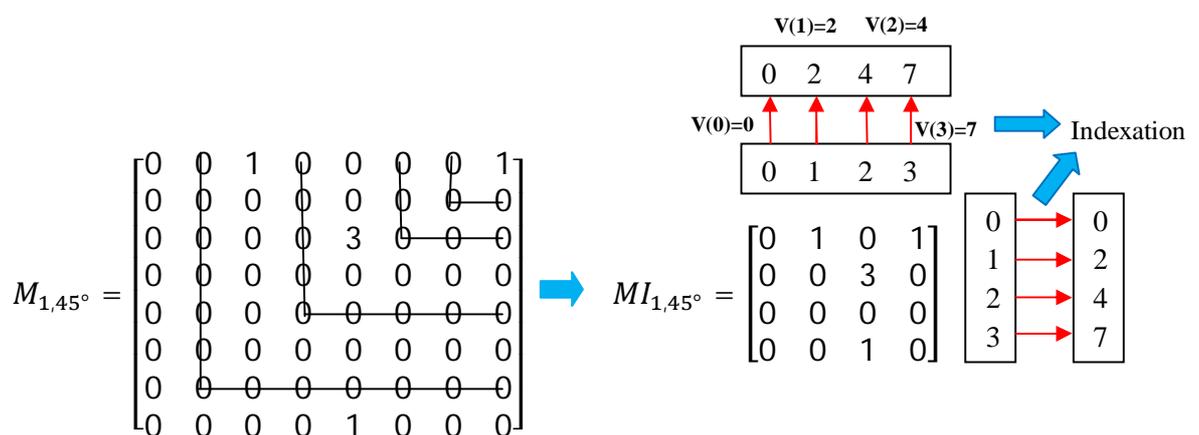
En analysant cette matrice, nous avons déduit qu'une étape de compression peut être réalisée en éliminant les zéros correspondent aux niveaux de gris qui n'existent pas réellement dans l'image (correspondent aux valeurs : 1, 3, 5 et 6).

Pour arriver à cette nouvelle matrice que nous avons appelé « matrice de cooccurrence indexée 'MI' », une première étape d'extraction des différents niveaux de gris existant réellement dans l'image est essentielle. **Le nombre de ces niveaux de gris construit la taille de la nouvelle matrice proposée.**

En continuant avec l'exemple précédent, on constate que l'image « **A** » n'a que **4 niveaux différents** de valeurs 0, 2, 4 et 7. La question qui se pose :

Pourquoi parcourir toutes les valeurs depuis la minimale jusqu'à la maximale si le résultat est sûrement nulle pour certaines ?

La réponse à cette question nous a guidé à déterminer un vecteur V qui ne contient que les valeurs existantes dans l'image. Dans notre exemple, le vecteur V est tel que: $V = [0, 2, 4, 7]$. Notre nouvelle matrice de cooccurrence devient une matrice de taille **4 x 4** :



d'où : $M_{1,45^\circ}(2,4) = MI_{1,45^\circ}(1,2) = 3$

Figure 4.10 : Illustration du calcul de la matrice de cooccurrence indexée

Pour mettre en évidence cette étape, on illustre dans ce qui suit quelques résultats démonstratifs :

- Pour une image purement uniforme:

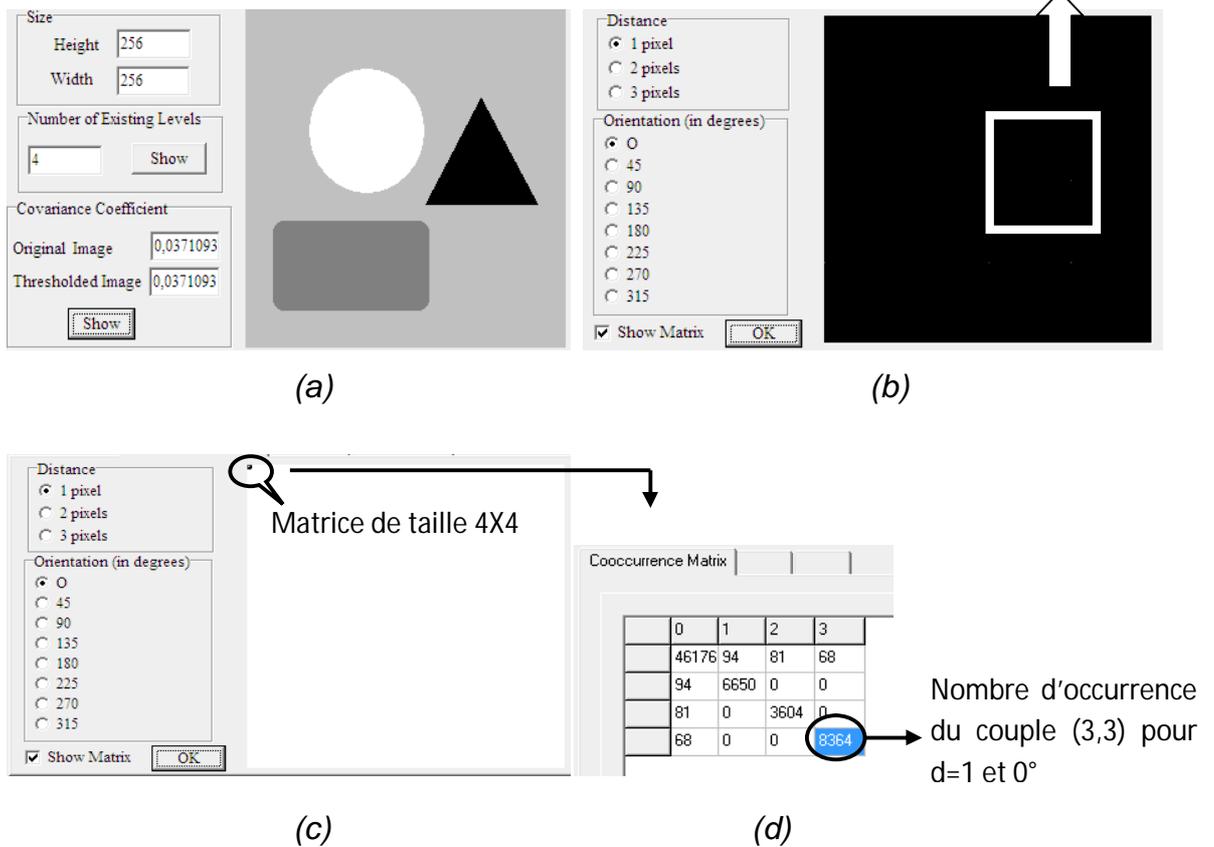


Figure 4.11 : Matrice de cooccurrence indexée ($d=1$, $\theta=0^\circ$), (a) « Image originale », (b) « Image de cooccurrence initiale », (c) « Image de cooccurrence indexée », (d) « Valeurs numériques de la matrice indexée »

NB : le nombre de lignes et de colonnes des tableaux d'affichage numérique (Figure 4.11 (d)) est égale cette fois-ci au nombre des intensités existants dans l'image en question.

Dans cet exemple (Figure 4.11), on obtient une très petite matrice de taille **4 x 4**. Cela montre la portée énorme des modifications apportées à la matrice de cooccurrence initiale.

Nous présentons ci-après quelques résultats (Figures 4.12 - 4.15) pour les différents types d'images :

- Image uniforme bruitée

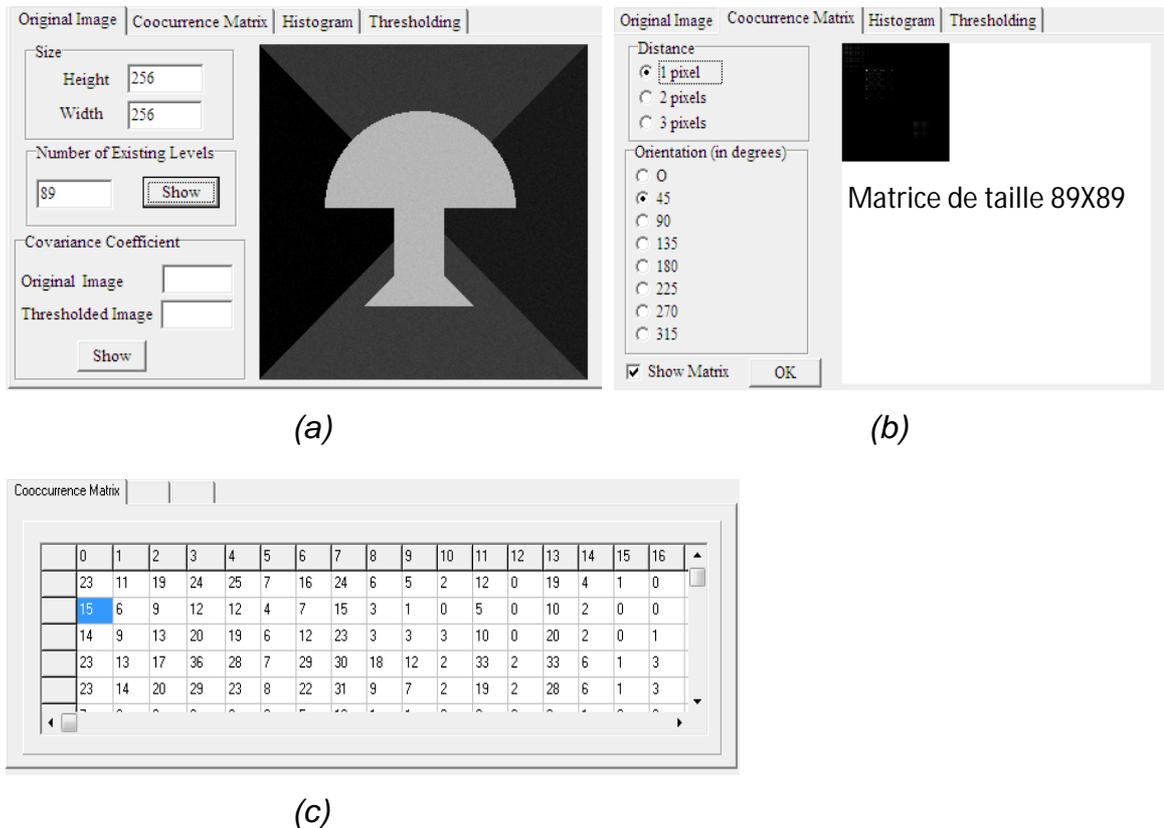
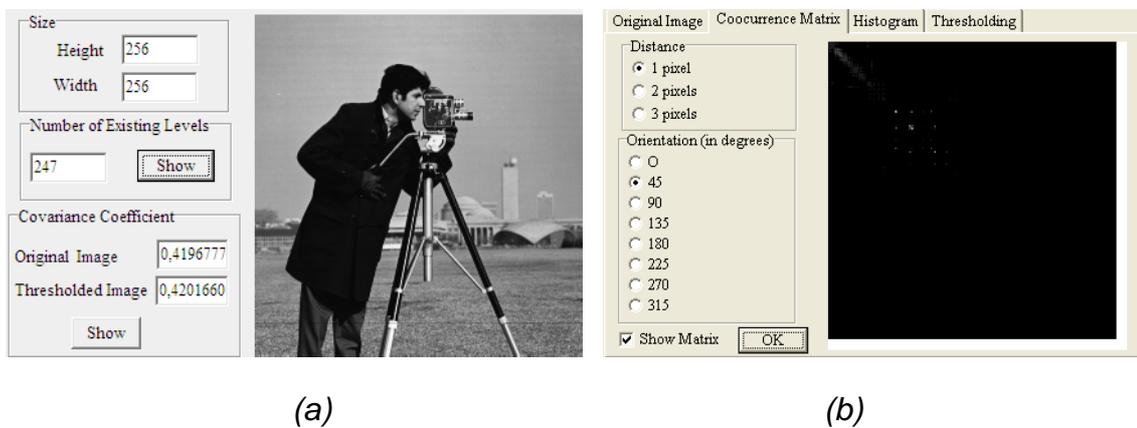


Figure 4.12 : Matrice de cooccurrence indexée ($d=1$, $\theta=45^\circ$), (a) « Image originale uniforme bruitée », (b) « Image de cooccurrence », (c) « Valeurs numériques de la matrice indexée »

- Image faiblement texturée



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	28	15	26	25	27	11	3	5	3	0	1	0	0	0	1	0	0
1	15	39	42	19	25	33	15	29	25	6	2	3	0	0	0	0	1
2	30	36	34	19	37	28	7	17	17	3	3	1	0	0	0	0	1
3	31	9	17	35	27	7	0	4	1	0	0	1	0	0	0	0	0
4	24	24	35	29	32	21	7	7	5	2	1	1	0	0	0	0	0

(c)

Figure 4.13 : Matrice de cooccurrence indexée ($d=1, \theta=45^\circ$), (a) « Image originale faiblement texturée », (b) « Image de cooccurrence », (c) « Valeurs numériques de la matrice indexée »

Avec la même image du cameraman et une autre direction :

Size

Height

Width

Number of Existing Levels

Covariance Coefficient

Original Image

Thresholded Image



Original Image | Cooccurrence Matrix | Histogram | Thresholding

Distance

1 pixel

2 pixels

3 pixels

Orientation (in degrees)

0

45

90

135

180

225

270

315

Show Matrix



(a)

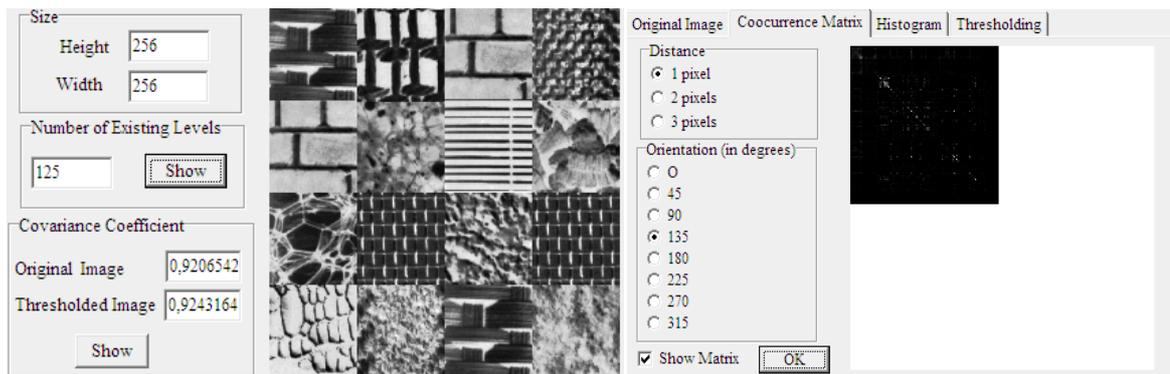
(b)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	23	13	28	26	24	10	1	4	3	1	0	0	0	0	0	0	0
1	17	38	28	12	23	36	15	21	27	4	2	1	0	0	0	0	0
2	19	31	38	17	31	30	5	20	14	2	3	1	0	0	0	0	0
3	29	9	19	30	21	8	0	1	2	0	1	0	1	0	0	0	0
4	26	21	32	23	30	17	2	12	8	1	2	0	0	0	0	0	0

(c)

Figure 4.14 : Matrice de cooccurrence indexée ($d=1, \theta=90^\circ$), (a) « Image originale faiblement texturée », (b) « Image de cooccurrence », (c) « Valeurs numériques de la matrice indexée »

- Image fortement texturée



(a)

(b)

Figure 4.15(c) displays the numerical values of the indexed cooccurrence matrix. The matrix is a 17x17 grid with columns and rows indexed from 0 to 16. The value 3 is highlighted in the cell at row 3, column 0.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	5	3	1	1	0	3	7	12	6	14	1	1	0	5	0	0	0
1	3	20	25	17	20	17	28	21	26	24	4	0	1	7	0	1	3
2	3	25	42	11	23	21	46	27	21	31	9	3	2	7	3	3	1
3	1	13	19	7	11	10	17	5	6	4	0	1	2	3	0	1	0
4	3	15	32	8	53	24	24	9	21	13	2	1	3	7	1	0	2
5	3	15	32	8	53	24	24	9	21	13	2	1	3	7	1	0	2
6	7	28	46	27	46	27	21	31	9	3	2	7	3	3	1	3	1
7	12	21	27	21	17	5	6	4	0	1	2	3	0	1	0	0	0
8	6	26	21	21	10	17	5	6	4	0	1	2	3	0	1	0	0
9	14	24	31	9	4	0	1	2	3	0	1	2	3	0	1	0	0
10	1	4	9	3	0	1	2	3	0	1	2	3	0	1	0	0	0
11	1	0	1	7	0	1	2	3	0	1	2	3	0	1	0	0	0
12	0	5	2	7	3	2	7	3	0	1	2	3	0	1	0	0	0
13	5	7	7	3	3	7	3	0	1	2	3	0	1	0	0	0	0
14	0	0	3	3	1	3	1	0	1	2	3	0	1	0	0	0	0
15	0	1	3	3	1	3	1	0	1	2	3	0	1	0	0	0	0
16	0	3	1	1	0	3	7	12	6	14	1	1	0	5	0	0	0

(c)

Figure 4.15 : Matrice de cooccurrence indexée ($d=1$, $\theta=135^\circ$), (a) « Image originale faiblement texturée », (b) « Image de cooccurrence », (c) « Valeurs numériques de la matrice indexée »

La matrice de cooccurrence indexée revient à compresser la matrice initiale en éliminant une valeur si sa colonne et sa ligne s'annulent en même temps.

b. Apports obtenus

Vue que la taille de la matrice de cooccurrence indexée est généralement inférieure de la matrice initiale, cette diminution influe sur le nombre des niveaux de gris à manipuler et par la suite, le temps nécessaire pour leur traitement.

La quantification d'efficacité de la nouvelle matrice est mesurée par le *gain immense en temps de calcul et en mémoire* qui reste relative et liée à l'image en cours. L'avantage de cette matrice est plus accentué quand les images sont multi-seuillées. On peut illustrer cela par comparaison des deux cas pour une même image (Figures 4.16 - 4.17):

- Matrice de cooccurrence indexée d'une image uniforme bruitée

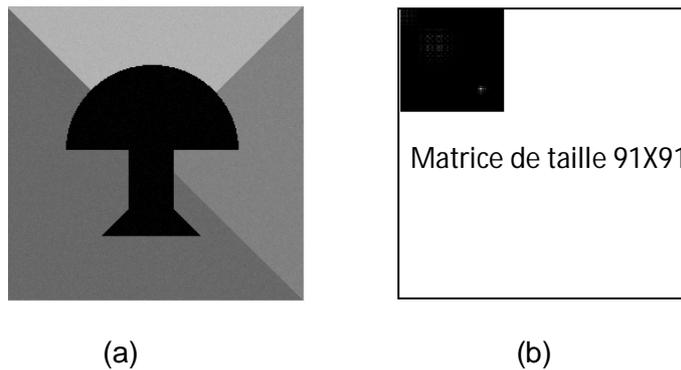


Figure 4.16 : Matrice de cooccurrence ($d=1, \theta = 45^\circ$), (a) « Image originale à 91Ngs », (b) « son image de cooccurrence »

- Matrice de cooccurrence indexée d'une image uniforme bruitée multi-seuillée

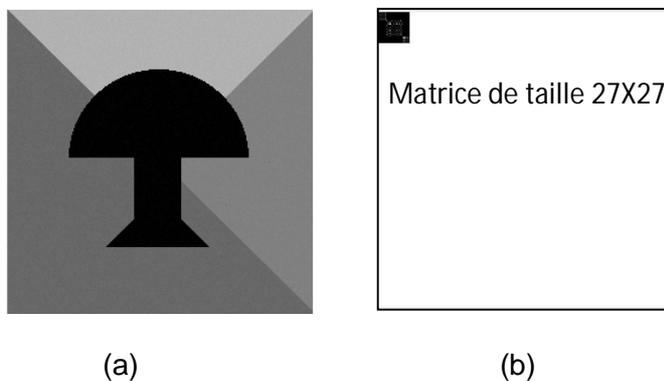


Figure 4.17 : Matrice de cooccurrence ($d=1, \theta = 45^\circ$), (a) « Image multiseuillée à 27 Ngs », (b) « son Image de cooccurrence indexée »

L'association du multi-seuillage avec l'indexation de la matrice de cooccurrence rapporte un gain considérable en temps et en mémoire.

c. Attributs de texture de la matrice de cooccurrence indexée :

Le changement de la matrice de cooccurrence indexée donne de nouvelles formules pour les attributs qu'utilisent la valeur directe des niveaux de gris. Le vecteur d'indexation V qui contient les niveaux de gris existants dans l'image originale est utilisé dans ce cas:

- Contraste : $Cont = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} [V(i) - V(j)]^2 \cdot M_{d,\theta}(i, j)$ (4.6)

- Variance : $Var = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} [V(i) + 1 - \sum_{k=0}^{NG-1} \sum_{l=0}^{NG-1} V(k) \cdot M_{d,\theta}(i, j)]^2$

$$M_{d,\theta}(i, j) \quad (4.7)$$

- Moment des différences inverses :

$$Momdiff = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} \frac{M_{d,\theta}(i, j)}{1 + [V(i) - V(j)]^2} \quad (4.8)$$

4.3.3. Filtres de Gabor

Généralement les méthodes du premier et du second ordre prennent en considération l'arrangement spatial des pixels. Elles demandent donc un temps de calcul relativement long et manquent de précision lorsque l'ordre des attributs statistiques devient assez grand. La difficulté principale des méthodes présentées est le manque d'outils adéquats capables de caractériser efficacement la texture sur différentes échelles.

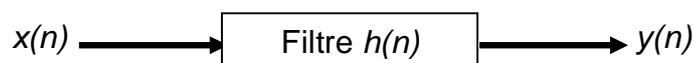
Le changement de la résolution induit un changement de l'apparence locale de l'image. Il sera donc très judicieux d'exploiter ce changement d'aspect de la texture pour une meilleure discrimination.

Des recherches intéressantes ont montré que les ondelettes ont la capacité de modéliser le système visuel humain qui prouve son sensibilité à l'orientation spécifique et les fréquences spatiales [2]. Elles ont fourni un nouvel ensemble d'outils analytiques multi-résolution.

Les ondelettes de Gabor sont identifiées par plusieurs recherches [38] [70] comme une méthode adéquate, récente et prometteuse dans la segmentation de texture.

a. Formulation mathématique

Un filtre numérique est un système qui fait une modification contrôlée d'une séquence d'entrée $x(n)$ en une séquence de sortie $y(n)$.



$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{+\infty} h(k) \cdot x(n - k) \quad (4.9)$$

où (*) indique le produit de convolution, $h(n)$ est la réponse impulsionnelle du filtre (réponse à une entrée égale à une impulsion).

La réponse impulsionnelle d'un filtre de Gabor est le produit d'une gaussienne et une exponentielle complexe. Dans le domaine temporel (1D), l'expression d'un filtre de Gabor est donnée par :

$$g(n) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{n^2}{2\sigma^2}} \cdot e^{2\pi j \cdot u_0 \cdot n} \quad (4.10)$$

où u_0 représente la fréquence centrale du filtre et σ l'écart type d'enveloppe gaussien.

En deux dimensions (2D), la formule (4.10) devient:

$$g(x, y) = \frac{1}{\sigma_x \cdot \sigma_y \cdot 2\pi} \cdot e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)} \cdot e^{j \cdot 2\pi \cdot (f_{x0} \cdot x - f_{y0} \cdot y)} \quad (4.11)$$

où σ_x et σ_y sont les écart-types respectivement le long des axes x et y, f_{x0} et f_{y0} les fréquences centrales respectivement suivant x et y. Le filtrage d'une image $I(x, y)$ par le filtre bidimensionnelle $g(x, y)$ est une image $I_f(x, y)$ donné par :

$$I_f(x, y) = \sum_{a,b} g(a, b) \cdot I(x + a, y + b) \quad (4.12)$$

En passant du repère cartésien à un repère polaire avec :

$f_0 = \sqrt{f_{x0}^2 + f_{y0}^2}$, la fréquence centrale du filtre et $\theta = \arctang\left(\frac{f_{y0}}{f_{x0}}\right)$ l'angle entre l'axe des abscisses et le module f_0 , la formule (4.11) s'écrit sous la forme suivante :

$$g(x, y) = \frac{1}{\sigma_x \cdot \sigma_y \cdot 2\pi} \cdot e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)} \cdot e^{j \cdot 2\pi f_0 \cdot (x \cos\theta - y \sin\theta)} \quad (4.13)$$

La gaussienne est généralement orientée avec le même angle θ , ce qui donne l'expression connu :

$$g(x, y) = \frac{1}{\sigma_x \cdot \sigma_y \cdot 2\pi} \cdot e^{-\left(\frac{x'^2}{2\sigma_x^2} + \frac{y'^2}{2\sigma_y^2}\right)} \cdot e^{j \cdot 2\pi f_0 \cdot x'} \quad (4.14)$$

Avec :

$$\begin{cases} x' = x \cdot \cos\theta + y \cdot \sin\theta \\ y' = -x \cdot \sin\theta + y \cdot \cos\theta \end{cases} \quad (4.15)$$

b. Fondement d'un banc de filtres de Gabor

Il est possible de créer ce qu'on appelle un banc de filtres de Gabor ou 'famille de Gabor' en utilisant plusieurs filtres de caractéristiques fréquentielles différentes grâce aux quatre paramètres (f_0, σ_x, σ_y et θ). L'équation (4.13) montre que la fonction $g(x,y)$ a une forme similaire à une ondelette [73] mère $\psi(t)$ dont la famille d'ondelettes $\psi_{s,\tau}(t)$ est donné par (4.16) :

$$\forall t \in \mathbb{R}, \psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t - \tau}{s}\right) \quad (4.16)$$

où s un facteur d'échelle et τ un paramètre de translation.

La formule de la fonction génératrice du Gabor devient :

$$g_{mn}(x,y) = a^{-m} \cdot g(x',y'), a > 1 \text{ et } m,n \text{ entiers} \quad (4.17)$$

où :

$$\begin{cases} x' = a^{-m} \cdot (x \cdot \cos\theta + y \cdot \sin\theta) \\ y' = a^{-m} \cdot (-x \cdot \sin\theta + y \cdot \cos\theta) \end{cases} \quad (4.18)$$

avec $\theta = \frac{n \cdot \pi}{K}$, K le nombre total des orientations, a^{-m} est le facteur d'échelle

c. Choix des paramètres

Le choix pour fixer les paramètres du filtre, notamment la largeur de bande de fréquence et l'orientation, doit être minutieux.

Les fonctions de Gabor forment une base complète mais non-orthogonales d'ondelettes. Cet aspect engendre des informations redondantes dans les images filtrées. Pour réduire cette répétition, une stratégie de transformation des paramètres est appliquée [74].

On note U_l et U_h les fréquences (supérieur et inférieur) du centre d'intérêt, S le nombre des résolutions dans la décomposition multi-résolution.

Pour manipuler que des multiplications au lieu des produits de convolution dans le domaine temporel, on passe au domaine fréquentielle. La transformée de Fourier discrète bidimensionnelle donnée par (4.19) a abouti à $G(u,v)$, la transformée de Fourier de $g(x,y)$ dans (4.20) :

$$F(u, v) = \frac{1}{\sqrt{N_y N_x}} \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} f(i, j) \cdot e^{-\frac{j2\pi i u}{N_x}} e^{-\frac{j2\pi j v}{N_y}} \quad (4.19)$$

$$G(u, v) = e^{\left[-\frac{1}{2} \left(\frac{(u-v)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right) \right]} \quad (4.20)$$

avec :

$$\sigma_u = \frac{1}{2\pi\sigma_x}, \sigma_v = \frac{1}{2\pi\sigma_y}, a = \left(\frac{U_h}{U_l} \right)^{\frac{-1}{S-1}}, \text{ et } m=0, 1, \dots, S-1. \quad (4.21)$$

Cette représentation est jugée la plus optimale [74].

d. Attributs de Gabor

Si $I(x,y)$ est l'image en question, sa transformée en ondelettes de Gabor à l'échelle m et l'orientation n est définie par l'équation suivante :

$$W_{mn}(x, y) = \int I(x_1, y_1) * g_{mn}(x - x_1, y - y_1) \cdot dx_1 dy_1 \quad (4.22)$$

Les attributs calculés à partir de cette représentation sont la moyenne μ_{mn} et l'écart type σ_{mn} :

$$\mu_{mn} = \int \int |W_{mn}(x, y)| \cdot dx dy \quad (4.23)$$

$$\sigma_{mn} = \sqrt{\int \int (|W_{mn}(x, y)| - \mu_{mn})^2 dx dy} \quad (4.24)$$

Si le nombre d'échelles est $S=3$ avec 4 orientations, le vecteur d'attributs est composé de 24 composantes :

$$V_{attrib} = [\mu_{00}\sigma_{00}, \dots, \mu_{23}\sigma_{23}] \quad (4.25)$$

Vu l'existence du terme complexe, l'équation (4.17) peut se décomposer en deux parties l'une réelle et l'autre imaginaire.

$$g_{mn(R)}(x, y) = a^{-m} \cdot g_{Réelle}(x', y') \quad (4.26)$$

$$g_{mn(I)}(x, y) = a^{-m} \cdot g_{Imag}(x', y') \quad (4.27)$$

On représente un exemple (Figures 4.19 – 4.20) de filtrage avec des masks de 5×5 (Figure 4.18) générés par les formules (4.26) et (4.27) pour $\theta = \frac{7\pi}{8}$, $S=3$, $U_l = 0.05, U_h = 0.4$:

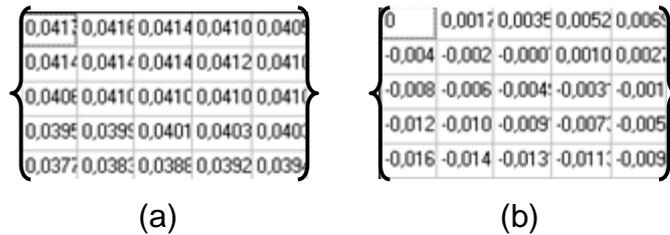


Figure 4.18 : Exemple d'un Filtre de Gabor, (a) « partie réelle », (b) « partie imaginaire »

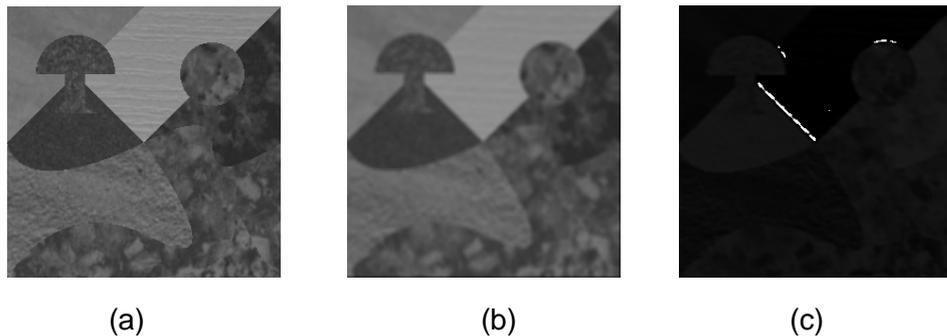


Figure 4.19 : Application d'un filtre de Gabor à une image texturée, (a) « Image originale », (b) « Image filtrée 'partie réelle' », (c) « Image filtrée 'partie imaginaire' »

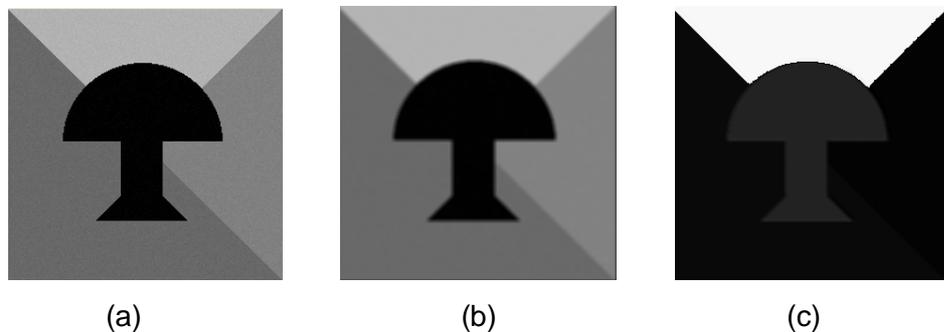


Figure 4.20 : Application d'un filtre de Gabor à une image uniforme bruitée, (a) « Image originale », (b) « Image filtrée 'partie réelle' », (c) « Image filtrée 'partie imaginaire' »

Après le choix des attributs issus de la matrice de cooccurrence indexée et les filtres de Gabor, des centres de classes sont déterminées par le meanshift et une dernière étape dans l'approche proposée sert à classifier les différents pixels par rapport à ces centroides.

4.4. Classification par Fuzzy C-Means

Notre dernière étape dans la segmentation proposée adopte l'algorithme de classification floue « Fuzzy C-Means » (**FCM**) [27] [75]. Avant de procéder à la représentation des résultats obtenus après l'implémentation de tel algorithme, on présente quelques rappels :

4.4.1. Notion de classification

Une classification ou système de classification est un système organisé et hiérarchisé de catégorisation d'objets. L'objectif de la classification est d'identifier les classes auxquelles appartiennent ces objets à partir de traits descriptifs.

4.4.2. Types de classification

La classification peut être :

- ✦ **Supervisée** dont les classes sont connues à priori et ont en général une sémantique associée.
- ✦ **Non-supervisée** (en anglais Clustering) dont les classes sont fondées sur la structure des objets. La sémantique associée aux classes est plus difficile à déterminer. A la fin du processus de cette classification, les individus doivent appartenir à l'une des classes générées par la classification.

Pour la classification non-supervisée, on distingue deux catégories: hiérarchiques et non-hiérarchiques. Dans la **classification hiérarchique (CH)**, les sous-ensembles créés sont emboîtés de manière hiérarchique les uns dans les autres. On distingue la CH *descendante (ou divisive)* qui part de l'ensemble de tous les individus et les fractionne en un certain nombre de sous-ensembles. Et la CH *ascendante (ou agglomérative)* qui part des individus seuls que l'on regroupe

en sous-ensembles. Pour déterminer quelles classes on va fusionner, on utilise un critère d'agrégation [1] [4].

Dans la **classification non-hiérarchique**, on parle de *partition* si chaque individu ne fait partie que d'un sous-ensemble. Si chaque individu peut appartenir à plusieurs groupes, avec probabilité p_i d'appartenir au groupe i , alors on parle de *recouvrement*.

Dans les deux cas (supervisé ou non), on a besoin de définir la notion de distance des classes entre eux et entre un individu et une classe avec la définition du critère d'agrégation.

4.4.3. Critère d'agrégation

Le critère d'agrégation permet de comparer les classes deux à deux pour sélectionner les classes les plus similaires. Les critères les plus classiques sont le plus proche voisin, le diamètre maximum, la distance moyenne et la distance entre les centres de gravités.

4.4.4. Méthode de clustering adoptée

L'algorithme choisi est un algorithme de classification floue [28][76] fondé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité [77]. Il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance (compris entre 0 et 1) à une classe donnée. Le cluster auquel est associé un pixel est celui dont le degré d'appartenance sera le plus élevé.

L'algorithme développé par Dunn [78] en 1973 et amélioré par Bezdek [79] en 1981, nécessite la connaissance a priori du nombre de classes. Il génère des classes par un processus itératif en minimisant une fonction objective.

Si u_{ik} désigne le degré d'appartenance du pixel i à la classe k , $k=1, \dots, C$ (C le nombre totale des classes), ces valeurs sont regroupées dans une matrice $\mathbf{U} = [u_{ik}]$.

Pour avoir une bonne partition, on impose les contraintes suivantes :

- a. $u_{ik} \in [0,1]$
- b. $\sum_k u_{ik} = 1, \forall i$

On note v_i le vecteur descriptif du pixel i , l'algorithme du FCM fait évoluer la partition (Matrice U) en minimisant la fonction objective suivante :

$$J_U(U, C) = \sum_i \sum_k u_{ik}^m \cdot \|v_i - c_k\|^2 \quad (4.28)$$

où: $m \geq 1$ est un paramètre contrôlant le degré de flou (généralement $m = 2$), c_k le centre de la classe k .

Algorithme

1. Fixer le paramètre floue m
2. Initialiser la matrice de partition U et les centres de classes c_k .
3. Fait évoluer U et c_k selon les deux équations suivantes :

$$- u_{ik} = \frac{1}{\sum_{j=1, \dots, c} \left(\frac{D_{ik}}{D_{ij}} \right)^{2/(m-1)}}, \text{ avec}$$

$$D_{ij} = \|v_i - c_j\| \text{ et } \| \| \text{ une métrique donnée} \quad (4.29)$$

$$- c_k = \frac{\sum_i (u_{ik}^m \cdot v_i)}{\sum_i (u_{ik}^m)} \quad (4.30)$$

4. Répéter 3. Jusqu'à convergence (tester que $\|J^{t+1} - J^t\| < \text{seuil}$, t numéro d'itération).

Après la présentation de différents outils employés pour mettre en œuvre notre processus de segmentation, on illustre ci-après nos aboutissements.

4.5. Segmentation des images texturées

Dans la méthode développée, nous avons segmenté ce type d'images selon leur contenu texturale (peu ou fortement texturées).

4.5.1. Choix d'attributs

Après la formulation de la matrice de cooccurrence indexée et choix des paramètres, nous avons opté pour une étude comparative de performance des différents attributs pour retenir les plus discriminants. Le détail de cette étude est présenté dans *l'appendice B*.

Les résultats obtenus illustrent que, pour la majorité des attributs, le pouvoir de discrimination est limité et ils ne peuvent pas être utilisés seuls. Les descripteurs retenus après plusieurs expérimentations sont comme suit :

- Cas des images faiblement texturées :

Nous avons choisi l'attribut de cooccurrence « **Variance** ». Ce descripteur semble efficace pour discriminer les textures fines à faibles variations.

La segmentation de ces images se fait à travers la fenêtre suivante (Figure 4.21) :

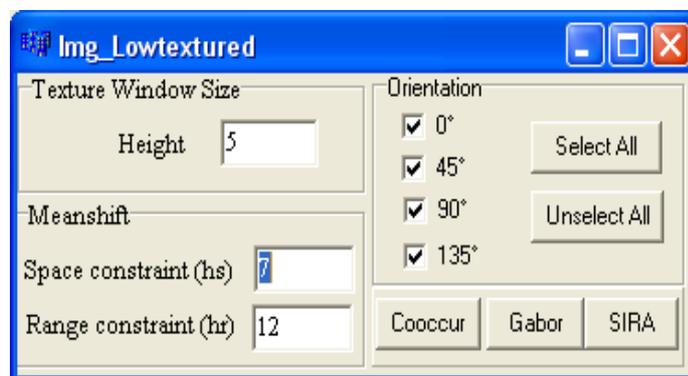


Figure 4.21 : Fenêtre de segmentation des images faiblement texturées

- Cas des images fortement texturées :

Dans le cas où on dispose de multiples textures ayant des caractéristiques proches difficilement discernables ou des textures grossières, il est ardu de trouver le couple ou l'ensemble des paramètres qui aboutit au meilleur résultat.

Nous avons opté pour l'ajout d'une étape intermédiaire d'étude de corrélation entre plusieurs attributs retenus (variance selon x, variance selon y, homogénéité et covariance).

Le système choisit les descripteurs ayant un taux de corrélation inférieure à un seuil préfixé à 0.8. Le calcul de la corrélation est basé sur la formule (4.31) en supposant que r_{XY} est le coefficient de corrélation entre les deux vecteurs caractéristiques $X(x_1, \dots, x_n)$ et $Y(y_1, \dots, y_n)$.

$$r_p = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.31)$$

Dans les deux cas de texture, les attributs sont calculés selon les quatre premières orientations (0° , 45° , 90° et 135°) et coopérés avec ceux issus des filtres de Gabor.

La figure suivante (Figure 4.22) montre la fenêtre utilisée pour la segmentation de ce type d'images :

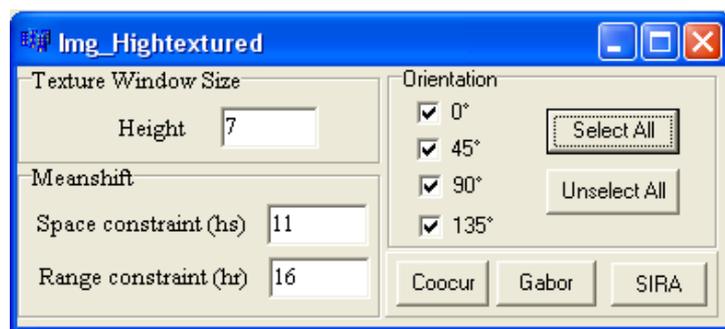


Figure 4.22 : Fenêtre de segmentation des images fortement texturées

4.5.2. Application du meanshift

Une fois l'espace attributs est déterminé, la procédure du meanshift présentée dans le chapitre précédent est appliquée. L'algorithme fournit un certain nombre de modes que nous les avons exploités pour surmonter la phase d'initialisation de l'algorithme de classification choisi.

a. Estimation du nombre de modes

Nous avons présenté dans une section précédente l'effet des largeurs de bandes. Le nombre de modes a été ambigu par rapport au nombre réel des niveaux de gris existants dans le cas des images uniformes.

Pour le cas des images texturées, le meanshift est appliqué à un espace multidimensionnel. Le problème majeur d'utilisation de tel algorithme est le temps de calcul élevé nécessaire si l'on applique à tous les pixels de l'image à segmenter. A cet effet, nous l'avons appliqué à un certain nombre limité des candidats. Cette limitation peut affecter l'estimation correcte des modes pour deux raisons :

- Les pixels non couverts par les fenêtres utilisées n'appartiennent à aucun cluster. Dans ce cas la segmentation est incomplète.
- La convergence peut être atteinte en arrivant au nombre maximal des itérations. Le mode détecté dans cette situation n'est qu'une valeur approchée.

En conséquence, la valorisation du pouvoir d'estimation reste relative. Ces raisons justifient notre utilisation de l'algorithme **FCM** pour classifier tous les pixels de l'image en question.

Dans notre travail, nous avons gardé les deux paramètres *hr* et *hs* variables pour les modifier lors de la segmentation. Un choix adéquat aboutit nécessairement à une estimation très satisfaisante.

4.5.3. Application du FCM

L'adoption du FCM améliore largement les résultats issus de la classification directe par rapport aux centroides déterminés par le meanshift surtout dans le cas des images fortement texturées.

L'association du FCM avec le meanshift présente une complémentarité intéressante dont l'un comble les lacunes de l'autre. Le FCM sert à classifier tous les pixels de l'image même ceux non parcouru par le meanshift. Ce dernier à son tour assure la phase d'initialisation du FCM.

Le résultat final de segmentation après l'application de ces méthodes, s'affiche dans une fenêtre (Figure 4.23) illustrant l'image segmentée avec les différents paramètres obtenus.

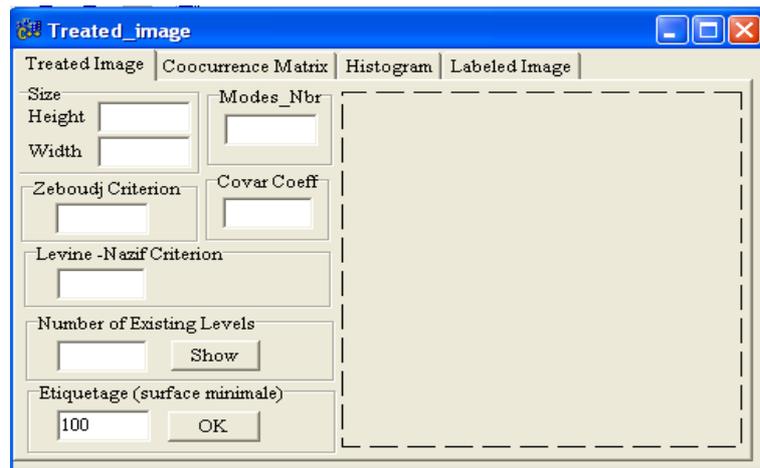


Figure 4.23 : Fenêtre de résultat segmentation

4.5.4. Apports de coopération adoptée

Il est connu dans la plupart des méthodes de segmentation que nous avons besoin de régler un certain nombre de paramètres. L'avantage majeur de notre système réside dans le nombre faible des paramètres à gérer.

Pour mettre en évidence nos choix en justifiant la coopération implémentée, nous montrons dans ce qui suit quelques résultats.

a. Illustrations démonstratives

Arrivons à notre but en mettant en œuvre une méthode adaptatif de segmentation coopérative, on montre par quelques exemples le résultat et l'apport du système développé en comparant le résultat issu d'une seule méthode avec celui issu de la coopération. Les démonstrations quantifiées avec quelques critères d'évaluation, font l'objet du prochain chapitre.

Pour les exemples suivants, la taille de fenêtre d'analyse est de 7×7 , le nombre d'échelles est $S=3$, le nombre d'orientations est $K=4$ avec des masks 7×7 pour les filtres de Gabor en itérant le FCM 15 fois ($m= 1,37$):

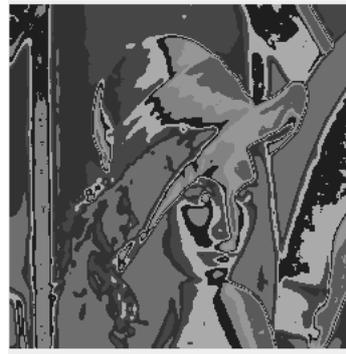
1) Cas des images faiblement texturées

Pour $hs= 7$, $hr= 12$:

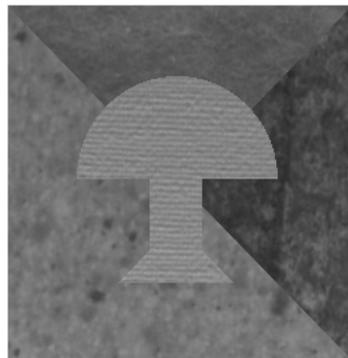
- En utilisant les attributs de cooccurrence



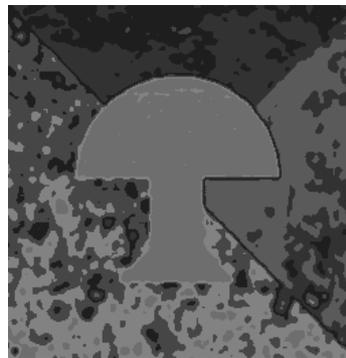
(a)



(b)



(a)



(b)

Figure 4.24 : Segmentation par matrices de cooccurrence, (a) « Image originale faiblement texturée », (b) « Image segmentée »

- En utilisant les attributs de Gabor



(a)



(b)

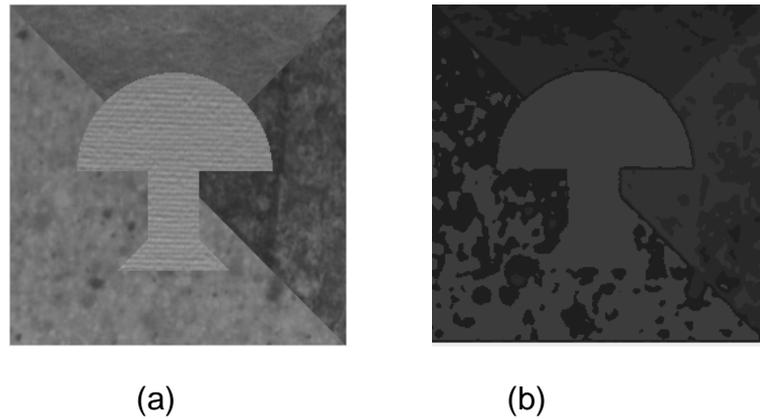


Figure 4.25 : Segmentation par filtres de Gabor, (a) « Image originale faiblement texturée », (b) « Image segmentée »

- Par coopération

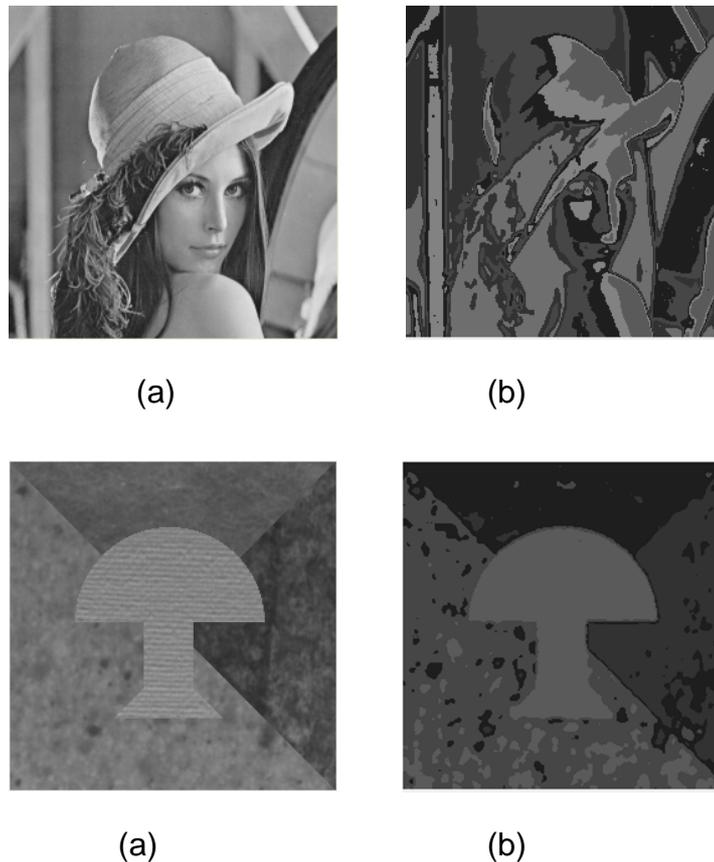


Figure 4.26 : Segmentation par coopération, (a) « Image originale faiblement texturée », (b) « Image segmentée »

Nous sommes capables en examinant visuellement ces images de constater que la coopération achève un meilleur résultat.

2) Cas des images fortement texturées

Pour $hs= 11$, $hr= 14$

- En utilisant les attributs de cooccurrence

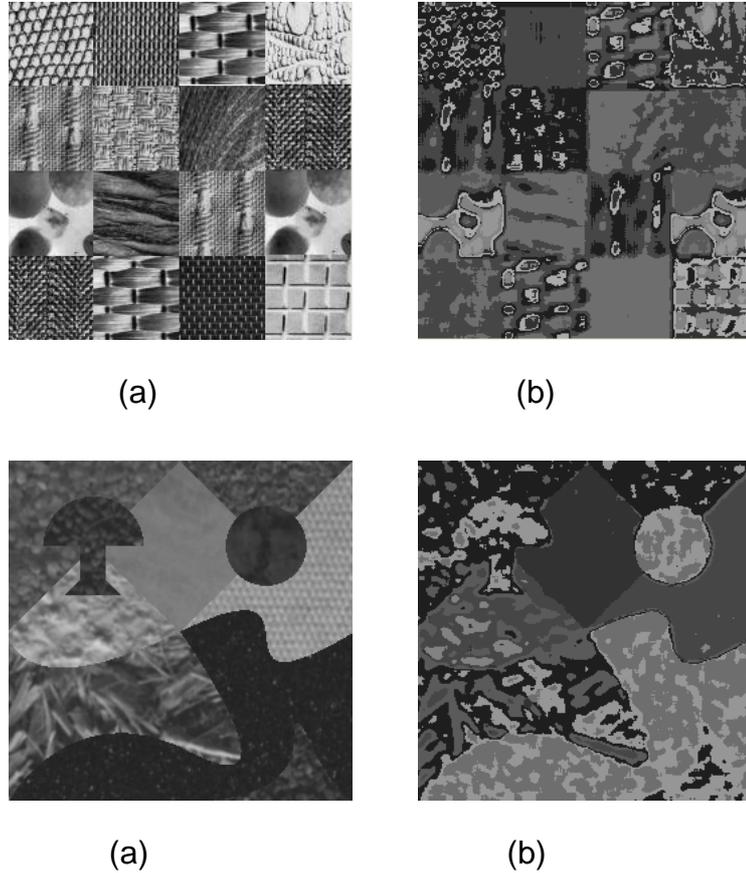
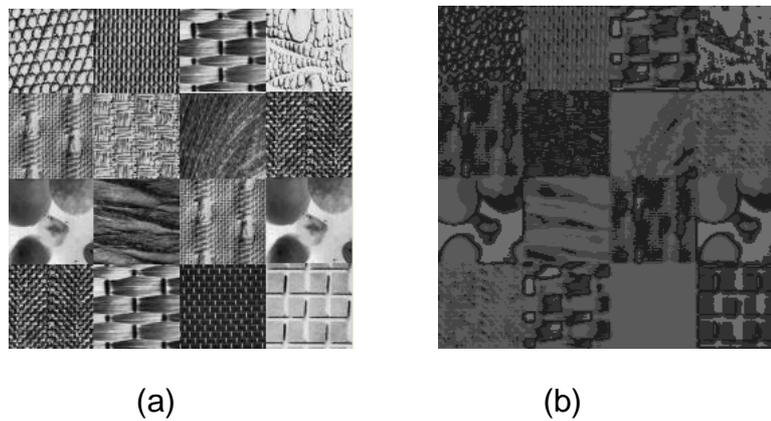


Figure 4.27 : Segmentation par matrices de cooccurrence, (a) « Image originale fortement texturées », (b) « Image segmentée »

- En utilisant les attributs de Gabor



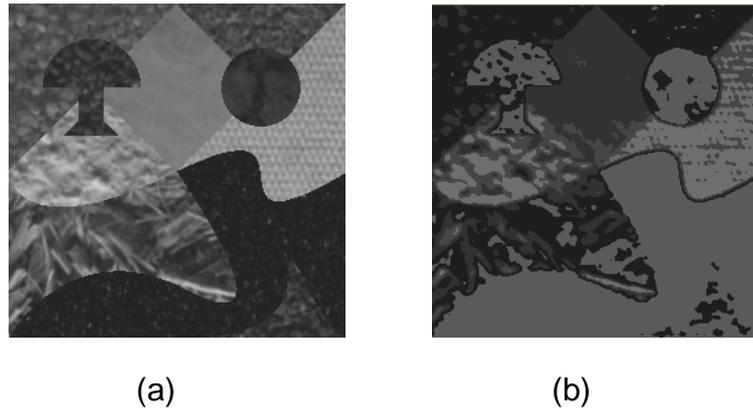


Figure 4.28 : Segmentation par filtres de Gabor, (a) « Image originale fortement texturées », (b) « Image segmentée »

- Par coopération

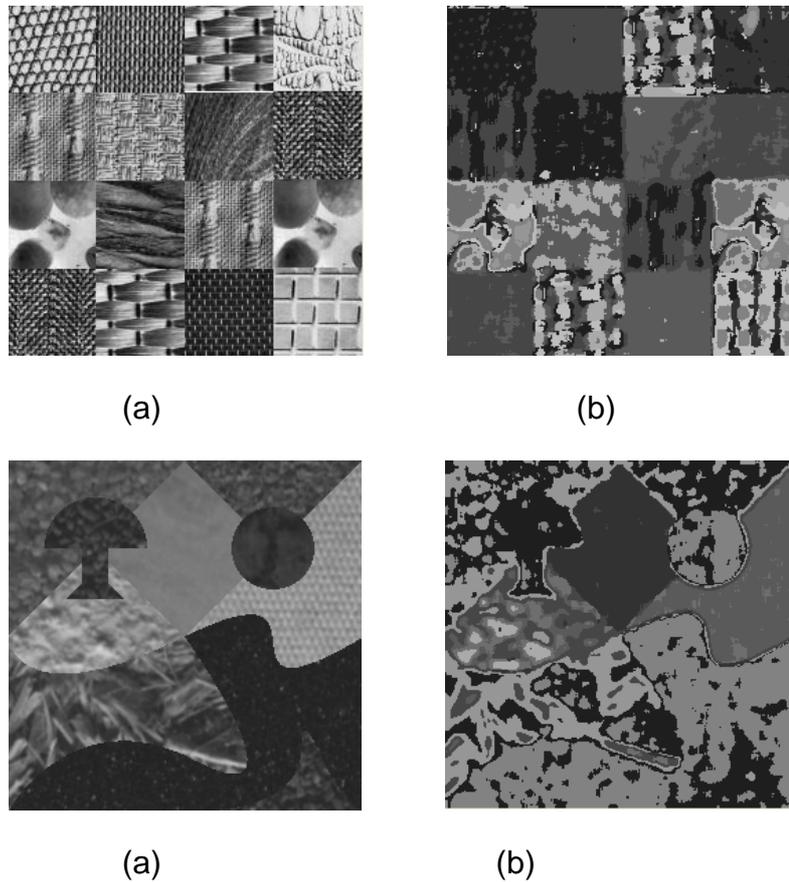


Figure 4.29 : Segmentation par coopération, (a) « Image originale fortement texturées », (b) « Image segmentée »

En analysant les résultats précédents (Figures 4.24 à 4.28), on peut clairement déduire que la qualité de segmentation issue de la coopération suggérée

montre l'apport et l'avantage de celle-ci. Les régions obtenues sont plus précises et convaincues. La coopération avec un espace plus élevé des attributs donne un aspect meilleur de discrimination dont les centroïdes issus du meanshift devenus plus assurés et fiables.

4.6. Conclusion

L'élaboration d'un système automatique qui mène à segmenter différents types d'images mono-spectrales est notre principale finalité. Néanmoins, les expériences ont montré qu'une telle tâche est très difficile à implémenter vu la diversité des images et leurs particularités. Un grand nombre d'expérimentations est alors incontournable pour arriver à ce but en faisant le bon choix soit des méthodes à appliquer soit des compromis à faire.

Notre méthode devrait montrer son adaptabilité au contexte d'image en question et sa performance en termes de segmentation en régions. La partie suivante illustre les résultats obtenus et leur évaluation.

CHAPITRE 5

EVALUATION DES RESULTATS OBTENUS

5.1. Introduction

Devant la multitude des méthodes développées depuis plusieurs décennies pour la segmentation des images, l'efficacité de ces dernières reste difficile à évaluer. Disposer de méthodes d'évaluation de résultats est nécessaire afin de comparer un nouvel algorithme à ceux préexistants et même pour régler ses paramètres en fonction du problème à résoudre.

L'objectif principal d'un critère d'évaluation est d'être capable de comparer différents résultats de segmentation afin de trouver la meilleure. Malgré le nombre élevé des critères développés, leur aptitude reste assez méconnue surtout dans le cas où l'image à segmenter contient des zones texturées, comme dans la plupart des images naturelles.

5.2. Méthodes d'évaluation

Lorsqu'on se réfère à la littérature, on trouve une multitude de critères [80] [81] développée pour évaluer un résultat de segmentation dont aucun à notre connaissance, n'est générale. Les critères d'évaluation peuvent être divisés, selon que l'on possède ou non une « *vérité-terrain* » qui constitue une *segmentation de référence*, en deux catégories principales : critères d'évaluation supervisée et critères d'évaluation non supervisée. Le premier type évalue la qualité d'un résultat de segmentation en mesurant sa similarité avec une vérité terrain. Le second estime la qualité d'un résultat de segmentation à partir de statistiques calculées sur chaque région détectée.

Ces critères sont groupés, selon la disponibilité d'une vérité terrain, en deux catégories :

5.2.1. Evaluation supervisée

Il y a plusieurs critères de ce type. Il s'agit d'obtenir le meilleur résultat de segmentation choisi par des juges, ou de calculer la distance entre un résultat de segmentation et une référence connue. Dans le cas d'utilisation d'une vérité terrain, on peut trouver deux groupes : les critères évaluant un résultat de détection de frontières [82] et ceux évaluant un résultat de segmentation en régions.

Il y a moins de critères d'évaluation d'un résultat de segmentation en régions/classes que ceux de détection de frontières.

5.2.2. Evaluation non supervisée

Nous pouvons évaluer la qualité du résultat de segmentation d'une image sans aucune information. Ce sont les critères d'évaluation non supervisée qui nous permettent de créer un système d'évaluation automatique. On peut classer ces derniers en deux grandes catégories: les critères de «contraste» et les critères d'«adéquation à un modèle». Les premiers recherchent une variabilité inter-région, alors que les seconds recherchent une uniformité en intensité ou en couleur à l'intérieur des régions.

5.3. Evaluation des résultats du système proposé

Dans le cas général, aucune vérité terrain n'est disponible pour évaluer un résultat de segmentation. Pour choisir un critère d'évaluation non supervisé qui nous convient mieux vu que nous ne disposons aucune image référence, nous avons préféré à se référer aux études comparatives de performance de plusieurs critères d'évaluation.

Parmi les critères étudiés, le contraste de Zeboudj et le critère d'uniformité de Levine et Nasif sont considérés comme les meilleures mesures testées pour évaluer la segmentation des images uniformes et texturées [83].

5.3.1. Formulation mathématique

Soit I une image définie sur un ensemble X de sites $s(i,j)$ et f une fonction à valeurs dans un ensemble Z . f pourra être l'intensité pour les images à niveaux de gris.

On notera R une segmentation à évaluer, c'est donc une partition de X en régions notées $R_i, i = 1, \dots, N$ vérifiant :

- $R_i \cap R_j = \emptyset$
- $\bigcup_{i=1}^N R_i = X$

On note A le nombre de pixels de l'image et A_i le nombre de pixels de la région R_i . La segmentation référence est notée V dont les régions sont $V_i, i=1, \dots, M$, μ_i et σ_i désignent respectivement la moyenne et l'écart type de f sur R_i .

a. Critère d'uniformité intra-région de Levine et Nazif

Ce critère simple [84] est basé sur la somme des variances des régions. Il doit donc être faible.

$$C_{intra} = \sum_i \sum_{s \in R_i} \left[f(s) - \frac{1}{A_i} \sum_{s \in R_i} f(s) \right]^2 = \sum_i \frac{\sigma_i^2}{C} \quad (5.1)$$

C est un facteur de normalisation, égal à la variance maximale

$$\sigma_{max}^2 = \frac{(f_{max} - f_{min})^2}{2} \quad (5.2)$$

Les régions peuvent être également pondérées par leur nombre de pixels.

b. Contraste de Zeboudj

Cet indice [85] prend en compte le contraste intérieur et le contraste extérieur aux régions, mesurés sur un voisinage $W(s)$ du pixel s .

Soit : $C(s, t) = \frac{|f(s) - f(t)|}{L-1}$, le contraste entre deux pixels s et t , avec L le maximum des intensités.

Le contraste intérieur d'une région R_i est :

$$I_i = \frac{1}{A_i} \cdot \sum_{s \in R_i} \max\{C(s, t), t \in W(s) \cap R_i\} \quad (5.3)$$

Le contraste extérieur d'une région R_i est :

$$E_i = \frac{1}{l_i} \cdot \sum_{s \in F_i} \max\{C(s, t), t \notin W(s) \cap R_i\} \quad (5.4)$$

où F_i est la frontière de R_i et l_i la longueur de F_i .

Le contraste de R_i est :

$$C(R_i) = \begin{cases} 1 - \frac{I_i}{E_i} & \text{si } 0 < I_i < E_i \\ E_i & \text{si } I_i = 0 \\ 0 & \text{sinon} \end{cases} \quad (5.5)$$

Le contraste global est enfin :

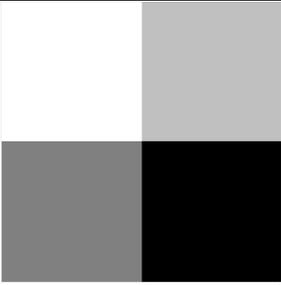
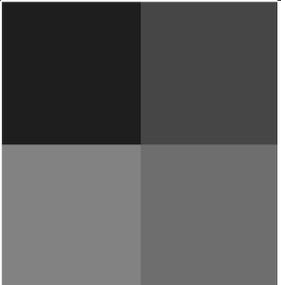
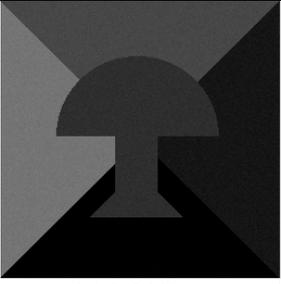
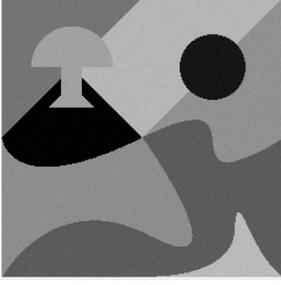
$$C_Z = \frac{1}{A} \sum_i A_i \cdot C(R_i) \quad (5.6)$$

5.3.2. Résultats et évaluation

L'évaluation de la segmentation issue de notre approche pour différentes images et différents descripteurs est présentée ci-après (Tableaux N° 5.1 à 5.8) en rappelant qu'une bonne segmentation doit avoir des valeurs proches de 1 pour le contraste de Zeboudj et des valeurs faibles du critère de Levine et Nazif.

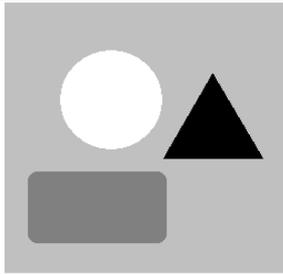
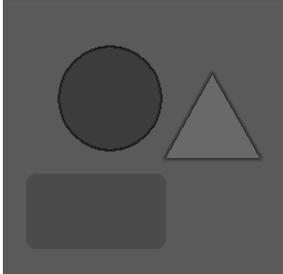
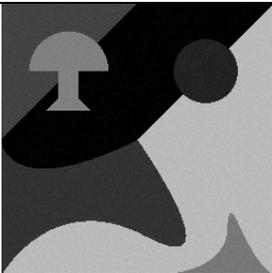
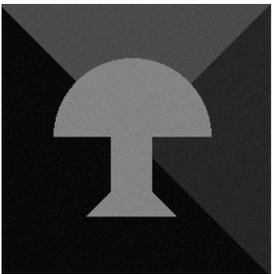
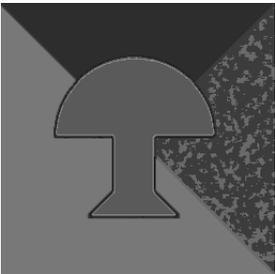
Pour tous les résultats des images faiblement et fortement texturées, l'algorithme de FCM est itéré 15 fois. Le facteur m est fixé à 1,37.

Tableau 5.1 : Résultats et évaluations de segmentation des images uniformes par l'attribut « niveau de gris »

Images originales	Images segmentées	P%	Evaluation	
			Lev & Naz	Zebo udj
 <p>4 NGs</p>	 <p>$hs= 7, hr= 4$ (6 modes obtenus)</p>	0%	0	<u>0,002</u>
 <p>112 NGs</p>	 <p>$hs= 13, hr= 15$ (7 modes obtenus)</p>	4,9%	0,31	0,49
 <p>151 NGs</p>	 <p>$hs= 13, hr= 17$ (8 modes obtenus)</p>	9,6%	0,30	0,58
 <p>(82 NGs)</p>	 <p>$hs= 11, hr= 21$ (5 modes obtenus)</p>	9,3%	0,035	0,63

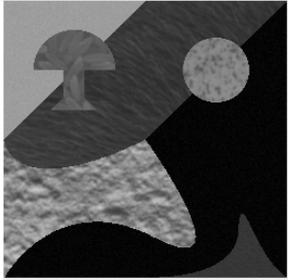
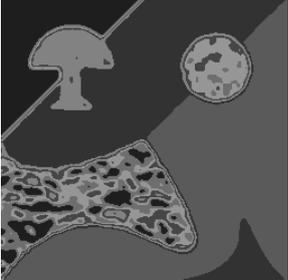
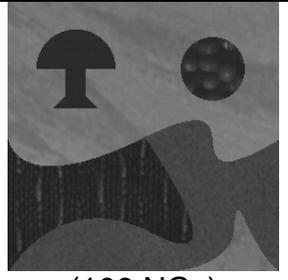
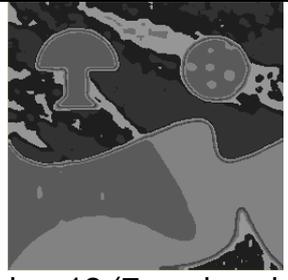
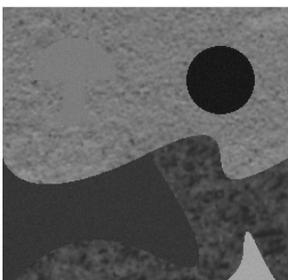
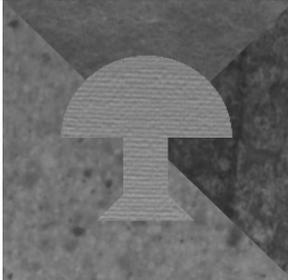
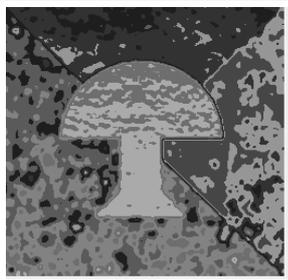
La segmentation des images bruitées confirme l'efficacité du meanshift. Les valeurs des critères montrent la bonne qualité de la segmentation obtenue. Pour les images purement uniformes, le critère de Zeboudj peut donner une appréciation incorrecte.

Tableau 5.2 : Résultats et évaluations de segmentation des images uniformes par l'attribut « Autocorrélation »

Images originales	Images segmentées	P%	Evaluation	
			Lev & Naz	Zebo udj
 (4 NGs)	 $hs= 11, hr= 21$ (6 modes obtenus)	3,7%	0,16	0,066
 (125 NGs)	 $hs= 13, hr= 9$ (10 modes obtenus)	8,2%	0,044	0,92
 (145 NGs)	 $hs= 11, hr= 15$ (8 modes obtenus)	6,1%	0,03	0,84
 (82 NGs)	 $hs= 9, hr= 17$ (6 modes obtenus)	9,3%	0,022	0,71

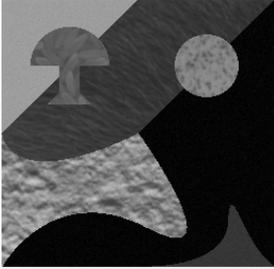
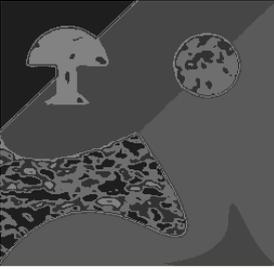
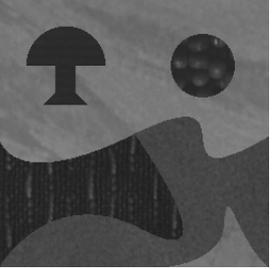
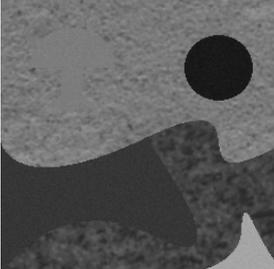
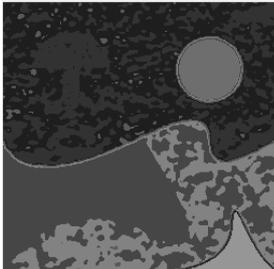
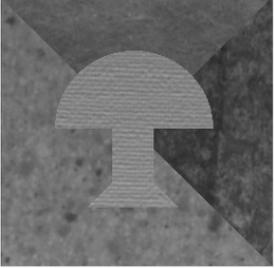
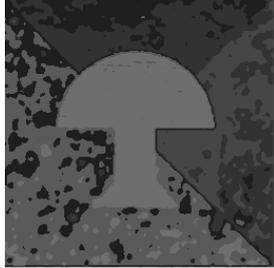
L'utilisation de l'auto-corrélation pour segmenter ce type d'images donne des valeurs plus encourageantes mais cet attribut donne des contours gênants.

Tableau 5.3 : Résultats et évaluations de segmentation des images faiblement texturées par l'attribut « Variance »

Images originales	Images segmentées	P%	Evaluation	
			Lev & Naz	Zebo udj
 <p>(178 NGs)</p>	 <p>$hs= 7, hr= 12$ (7 modes obtenus)</p>	34,9%	0,079	0,72
 <p>(106 NGs)</p>	 <p>$hs= 7, hr= 12$ (7 modes obtenus)</p>	17,8%	0,03	0,72
 <p>(158 NGs)</p>	 <p>$hs= 11, hr= 14$ (5 modes obtenus)</p>	54%	0,024	0,62
 <p>(158 NGs)</p>	 <p>$hs= 7, hr= 11$ 8 modes obtenus)</p>	49%	0,014	0,56

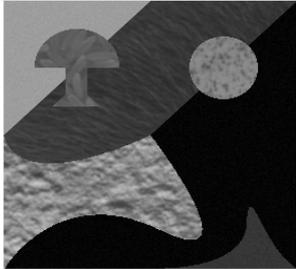
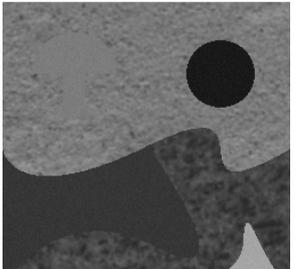
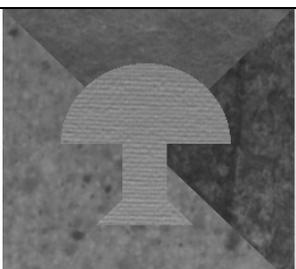
Malgré l'utilisation d'un seul attribut, l'appréciation des critères assure que les résultats sont satisfaisants et l'approche élaborée est prometteuse.

Tableau 5.4 : Résultats et évaluations de segmentation des images faiblement texturées par «Filtres de Gabor »

Images originales	Images segmentées	P%	Evaluation	
			Lev & Naz	Zebo udj
 (178 NGs)	 $hs= 7, hr=12$ (6 modes obtenus)	34,9%	0,038	0,70
 (106 NGs)	 $hs= 7, hr= 12$ (5 modes obtenus)	17,8%	0,052	0,79
 (158 NGs)	 $hs= 11, hr=9$ (7 modes obtenus)	54%	0,020	0,59
 (158 NGs)	 $hs= 7, hr=11$ (5 modes obtenus)	49%	0,11	0,61

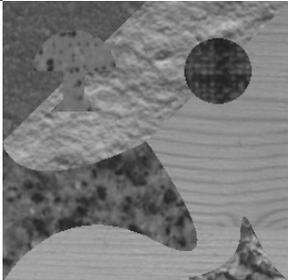
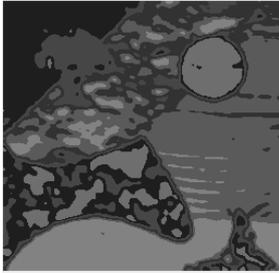
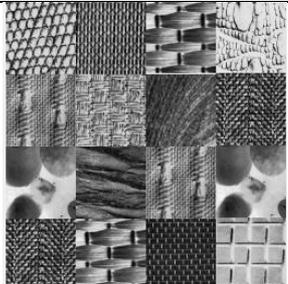
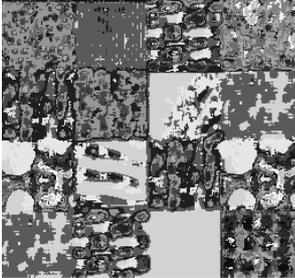
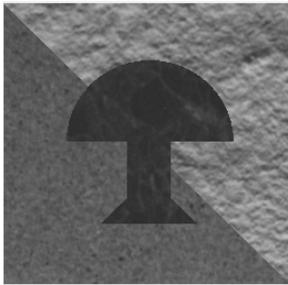
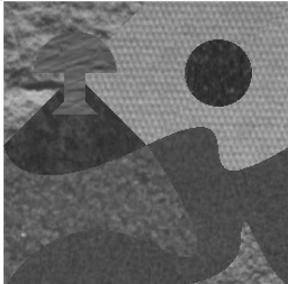
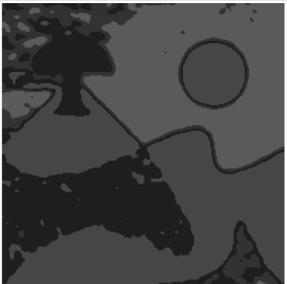
Ces filtres ont donné dans quelques cas des valeurs d'évaluation meilleurs. Le résultat de la segmentation par ces filtres justifie leur popularité. Cependant, ces filtres montrent leur insuffisance pour la localisation des textures grossières.

Tableau 5.5 : Résultats et évaluations de segmentation des images faiblement texturées par le système proposé « Coopération »

Images originales	Images segmentées	P%	Evaluation	
			Lev & Naz	Zebo udj
 (178 NGs)	 $hs= 7, hr= 12$ (9 modes obtenus)	34,9%	0,070	0,71
 (106 NGs)	 $hs= 7, hr= 12$ (5 modes obtenus)	17,8%	0,056	0,75
 (158 NGs)	 $hs= 13, hr= 12$ (6 modes obtenus)	54%	0,011	0,61
 (158 NGs)	 $hs= 7, hr= 14$ (5 modes obtenus)	49%	0,15	0,65

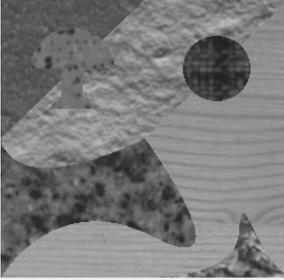
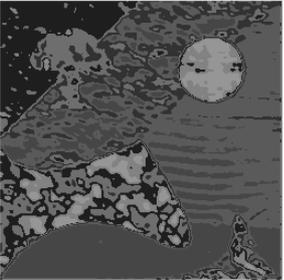
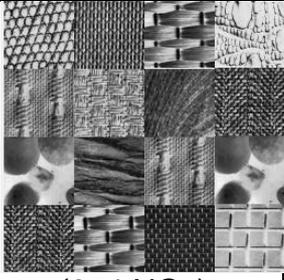
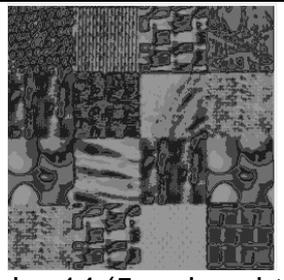
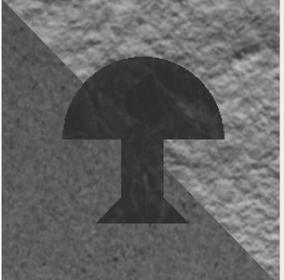
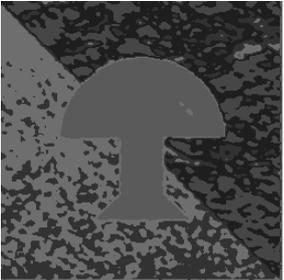
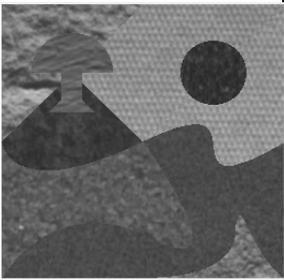
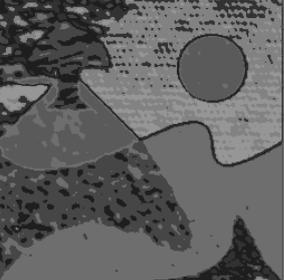
L'évaluation du système proposé a donné des valeurs meilleures par rapport aux deux cas précédents. Ces aboutissements prouvent la coopération adoptée.

Tableau 5.6 : Résultats et évaluations de segmentation des images fortement texturées par « Attributs de cooccurrence »

Images originales	Images segmentées	P%	Evaluation	
			Lev & Naz	Zebo udj
 <p>(165 NGs)</p>	 <p>$hs= 7, hr=9$ (6 modes obtenus)</p>	71%	0,11	0,61
 <p>(254 NGs)</p>	 <p>$hs= 7, hr=14$ (5modes obtenus)</p>	94%	0,032	0,31
 <p>(158 NGs)</p>	 <p>$hs= 11, hr=14$ (4modes obtenus)</p>	64%	0,31	0,65
 <p>(160 NGs)</p>	 <p>$hs= 11, hr=10$ (4 modes obtenus)</p>	69%	0,18	0,70

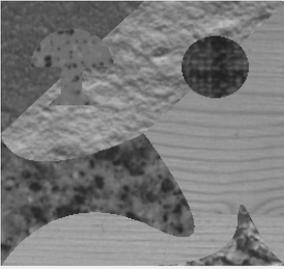
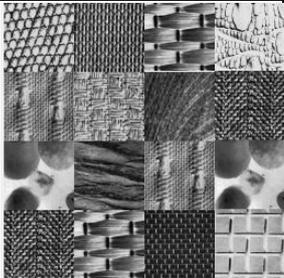
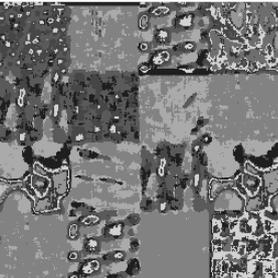
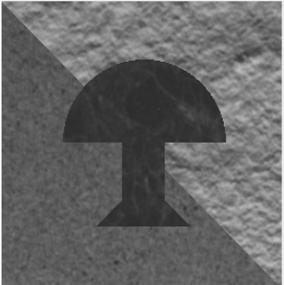
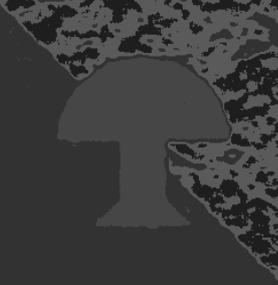
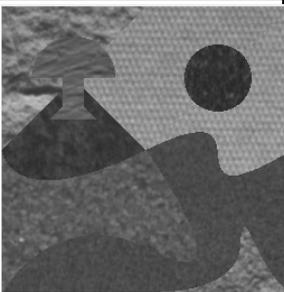
La segmentation à l'aide d'un ensemble faible d'attributs pour ce type d'images a donné des résultats satisfaisants malgré la taille petite de la fenêtre d'analyse comparée avec ceux utilisé habituellement. Le système reste limité pour la localisation des régions à texture grossières.

Tableau 5.7 : Résultats et évaluations de segmentation des images fortement texturées par « Filtres de Gabor »

Images originales	Images segmentées	P%	Evaluation	
			Lev & Naz	Zebo udj
 (165 NGs)	 $hs= 11, hr=9$ (7modes obtenus)	71%	0,033	0,58
 (254 NGs)	 $hs= 7, hr=14$ (5modes obtenus)	94%	0,08	0,36
 (148 NGs)	 $hs= 11, hr=12$ (7 modes obtenus)	64%	0,018	0,55
 (160 NGs)	 $hs= 11, hr=10$ (4modes obtenus)	69%	0,02	0,60

Les filtres de Gabor montrent leur besoin d'amélioration (fenêtre plus élevée ou coopération avec autres attributs) pour segmenter ce type d'images. Malgré que les valeurs d'évaluation soient acceptables, la difficulté de segmenter les macro-textures reste apparente.

Tableau 5.8 : Résultats et évaluations de segmentation des images fortement texturées par le système proposé « Coopération »

Images originales	Images segmentées	P%	Evaluation	
			Lev & Naz	Zebo udj
 <p>(165 NGs)</p>	 <p>$hs= 11, hr=9$ (5modes obtenus)</p>		0,11	0,62
 <p>(254 NGs)</p>	 <p>$hs= 11, hr=10$ (7modes obtenus)</p>	94%	0,071	0,34
 <p>(148 NGs)</p>	 <p>$hs=11, hr=14$ (4modes obtenus)</p>	64%	0,15	0,66
 <p>(160 NGs)</p>	 <p>$hs= 11, hr=14$ (5modes obtenus)</p>	69%	0,048	0,67

L'évaluation de la coopération proposée prouve son efficacité. Dans la plupart des cas, nous avons obtenu des valeurs plus élevés.

5.4. Conclusion

Dans tous les tableaux précédemment présentés, nous avons exposé les diverses images segmentées dans la deuxième colonne. A chacune d'image est associé deux valeurs d'évaluation concernant les critères choisis présentés auparavant. L'analyse de ces résultats peut aboutir à plusieurs déductions :

- Les images de résultats reflètent visuellement l'efficacité du système développé.
- L'appréciation du critère de Zeboudj montre l'avantage apporté par la coopération suggérée. Sa valeur est généralement supérieure à celles concernant l'utilisation d'une seule méthode (cooccurrence ou Gabor).
- Le critère de Levine & Nazif confirme nos conclusions.

En récapitulant nos tests et analyses, on affirme que malgré la taille faible (5x5 ou 7x7) de la fenêtre d'analyse, l'approche proposée a donné des résultats satisfaisants et encourageants. Notre système est très performant pour les régions uniformes et texturées à texture fine ou microscopique. Quand le choix des largeurs de bandes est adéquat, le résultat est très bon.

L'évaluation via critère de Zeboudj est parait adéquate et donne des appréciations convenables. Ce dernier peut aider à choisir les bonnes valeurs des largeurs *hs* et *hr* en réajustant leurs valeurs selon son évaluation.

CONCLUSION ET PERSPECTIVES

Il existe essentiellement deux approches au problème de segmentation. L'une est dite par régions et l'autre par frontières. La thématique de ce mémoire s'appartient au premier type.

La philosophie de segmentation des approches régions est généralement basée sur l'hypothèse d'homogénéité des objets, souvent en termes de niveau de gris et/ou textures. Les approches région utilisent également l'hypothèse que deux objets adjacents ne peuvent pas avoir la même distribution de niveaux de gris ou pas la même texture. Néanmoins, une segmentation région offre une certaine souplesse pour le choix des critères d'homogénéité et peut intégrer plusieurs informations si le modèle de segmentation est approprié. Vu la diversité des images et leurs particularités, les techniques dites adaptatives, permettent d'aboutir en partie des bons résultats de segmentation. Le travail présenté dans ce document s'inscrit dans cette optique.

Nous avons commencé par présenter un aperçu global sur les différentes méthodes utilisées en segmentation d'images dans le but de définir, de présenter et de mettre en évidence les différents outils qu'offre ces approches et introduire les principaux liés au cadre de notre étude. Notre algorithme de segmentation prend en considération l'hétérogénéité de l'image en cours. Selon le type d'image : uniforme, faiblement texturée ou fortement texturée, on choisit les attributs à calculer pour discriminer les différents objets dans cette image. Pour atteindre ce but nous avons procédé à différents tests pour le réglage des paramètres. Sa mise en œuvre s'articule autour des points suivants :

- La simplification d'image tous en minimisant le nombre de niveaux de gris à manipuler.
- La détermination du type d'image en question afin de mieux adaptés les traitements à porter en aval
- L'adaptation de l'espace attribut selon le type d'image.

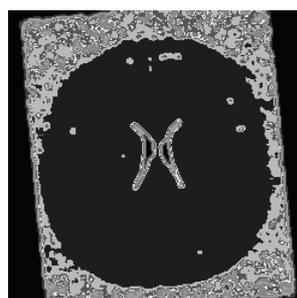
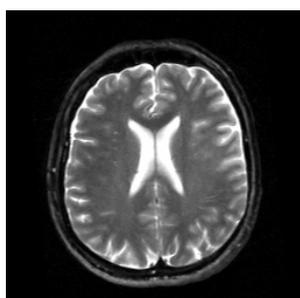
- La détermination du nombre de classes par l'algorithme de Mean-shift. L'aspect automatique été notre première motivation lors du choix de tel algorithme.
- Une classification par l'algorithme de Fuzzy-C-Mean est réalisée en dernière étape.

Des tests ont été effectués, sur des images de synthèse et d'autres réelles de différents types, pour aboutir aux meilleurs paramètres à utiliser. Les résultats obtenus ont permis de définir les meilleurs compromis à faire pour multi-seuiller l'image et pour préciser son type. La prise en compte d'attributs autres que le niveau de gris s'est avéré nécessaire pour segmenter des images ayant des régions texturées. Une amélioration significative, en mémoire et en temps de calcul, a été obtenue par la prise en compte du nombre réel des niveaux de gris existant dans l'image avec l'introduction de la matrice de cooccurrence indexée. Nous avons illustré la contribution de telle matrice proposée dans ce mémoire avec de nombreux résultats expérimentaux sur des images de différentes natures, attestant la validité et l'intérêt de cette dernière. L'introduction du multi-seuillage et la matrice de cooccurrence indexée rendent les temps de traitement raisonnables.

Nous avons introduit l'estimation automatique du nombre de classes par le Mean-shift dont l'avantage réside au nombre faible des paramètres à contrôler.

L'adaptabilité au contexte est un avantage qui renforce la robustesse du système proposé. Nos résultats de segmentation des différentes images synthétiques et des images réelles médicales sont très encourageants.

Divers aspects de ce travail méritent sans doute d'être approfondis, offrant ainsi de nouvelles perspectives pour arriver à un système complet de segmentation adaptative en régions.



Des améliorations peuvent être apportées portant notamment sur :

- Le choix des attributs à utiliser qui peut être fondé sur une étude plus approfondie du pouvoir discriminatoire de ces derniers.
- L'aspect échelle peut être considéré non seulement via un banc de filtres mais aussi avec les mêmes attributs calculés sur différentes tailles de fenêtre d'analyse.
- Une analyse pour une meilleure sélection des distances et des mesures de similarité peut avantager le système.
- Une segmentation multi-agents peut être une étape très efficace pour accélérer l'obtention du résultat final
- Une boucle de rectification interne peut être un outil performant d'autocorrection dans le système. Une évaluation et un re-calcul se fait automatiquement jusqu'à l'atteinte du meilleur résultat.

APPENDICE A

STATISTIQUES DU PREMIER ET DEUXIEME ORDRE

A.1. Statistiques du premier ordre

Les statistiques du premier ordre sont estimées sur les niveaux de gris sans prendre en compte leur distribution relative. Beaucoup d'attributs sont proposés : tels que les moments et moyennes d'espace ainsi que les attributs qu'on peut extraire de l'histogramme de niveau de gris. On présente quelques paramètres :

- Energie : $W = \sum_i |p(i)|^2$ (A.1)

- Entropie : $E = -\sum_i p(i) \cdot \log(p(i))$ (A.2)

- Moment d'ordre k : $\mu_k = \sum_i i^k \cdot p(i)$ (A.3)

-Contraste: $C = (\max(i) - \min(i)) / (\max(i) + \min(i))$

- Moment centré d'ordre k : $\eta_k = \sum_i (i - \mu_k)^k \cdot p(i)$ (A.4)

En particulier :

➤ Variance : $\sigma^2 = \eta_2$ (A.5)

➤ Biais : $\gamma_1 = \eta_3 / \sigma^3$ (A.6)

➤ Kurtosis (aplatissement) : $\gamma_2 = \eta_4 / \sigma^4 - 3$ (A.7)

A.2. Statistiques du deuxième ordre (matrice de cooccurrence):

• Inertie : $inert = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} |i - j|^2 \cdot M_{d,\theta}(i, j)$ (A.8)

• Contraste : $cont = \sum_{i=0}^{NG-1} n^2 \cdot \sum_{|i-j|=n} M_{d,\theta}(i, j)$ (A.9)

• Moyenne : $Moy = \frac{1}{NG^2} \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} M_{d,\theta}(i, j)$ (A.10)

- Variance : $\text{var} = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} (i - \text{Moy})^2 \cdot M_{d,\theta}(i, j)$ (A.11)

- Moyenne des lignes : $\text{Moy}_i = \sum_{i=0}^{NG-1} i \cdot M_{d,\theta}(i, j)$ (A.12)

- Variance : $\text{var} = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} \left[i + 1 - \sum_{k=0}^{NG-1} \sum_{l=0}^{NG-1} k \cdot M_{d,\theta}(k, l) \right]^2 \cdot M_{d,\theta}(i, j)$ (A.13)

- Entropie : $\text{entrp} = - \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} M_{d,\theta}(i, j) \cdot \log(M_{d,\theta}(i, j))$ (A.14)

- Moment des différences inverses : $\text{momdiff} = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} \frac{M_{d,\theta}(i, j)}{1 + (i - j)^2}$ (A.15)

- Moment angulaire du second ordre: $\text{energ} = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} (M_{d,\theta}(i, j))^2$ (A.16)

- Homogénéité : $\text{Homog} = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} \frac{1}{1 + |i - j|} \cdot M_{d,\theta}(i, j)$ (A.17)

- Uniformité : $\text{unif} = \sum_{i=0}^{NG-1} M_{d,\theta}(i, i)$ (A.18)

- Corrélation : $\text{corr} = \frac{\sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} i \cdot j \cdot M_{d,\theta}(i, j) - \left(\sum_{i=0}^{NG-1} i \cdot P_x(j) \right) \left(\sum_{j=0}^{NG-1} j \cdot P_y(i) \right)}{\sqrt{|\chi| \cdot |\lambda|}}$ (A.19)

Où : $P_x(i) = \sum_{j=0}^{NG-1} M_{d,\theta}(i, j)$, $P_y(j) = \sum_{i=0}^{NG-1} M_{d,\theta}(i, j)$

$$\chi = \sum_{i=0}^{NG-1} i^2 \cdot P_x(i) - \left(\sum_{i=0}^{NG-1} i \cdot P_x(i) \right)^2, \quad \lambda = \sum_{j=0}^{NG-1} j^2 \cdot P_x(j) - \left(\sum_{j=0}^{NG-1} j \cdot P_x(j) \right)^2$$

Ou bien
$$\text{corr} = \frac{\sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} (i - j) \cdot M_{d,\theta}(i, j) - \mu_x \cdot \mu_y}{\sigma_x \cdot \sigma_y}$$
 (A.20)

- Auto-corrélation : $\text{Autocorr} = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} i \cdot j \cdot M_{d,\theta}(i, j)$ (A.21)

- Covariance : $\text{cov ar} = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} (i - \text{moy}_i) \cdot (j - \text{moy}_j) \cdot M_{d,\theta}(i, j)$ (A.22)

APPENDICE B

Etude comparative des attributs de cooccurrence

Dans une première étape, nous avons comparé les différentes images attributs pour différents types d'images à segmenter. Le résultat est comme suit:

- Pour une orientation des matrices de cooccurrence égale à 45°

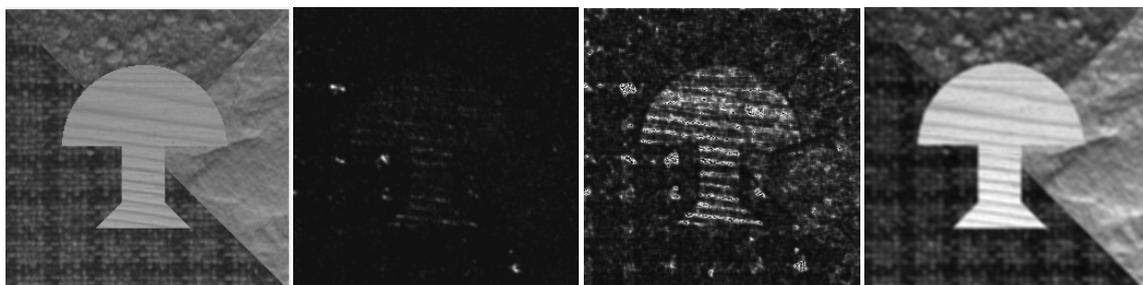
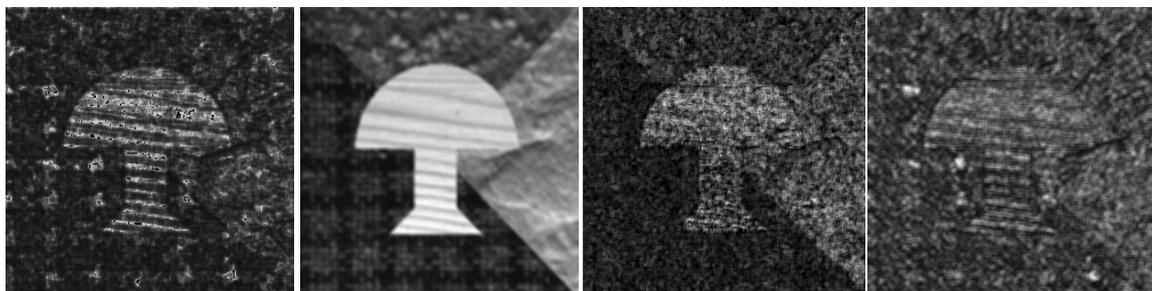


Image originale

Energie

Variance

Covariance



Moyenne

Auto-corrélation

Somme des moyennes

Homogénéité

Figure B.1 : Images attributs pour différentes type d'images

- Pour une orientation des matrices de cooccurrence égale à 135°

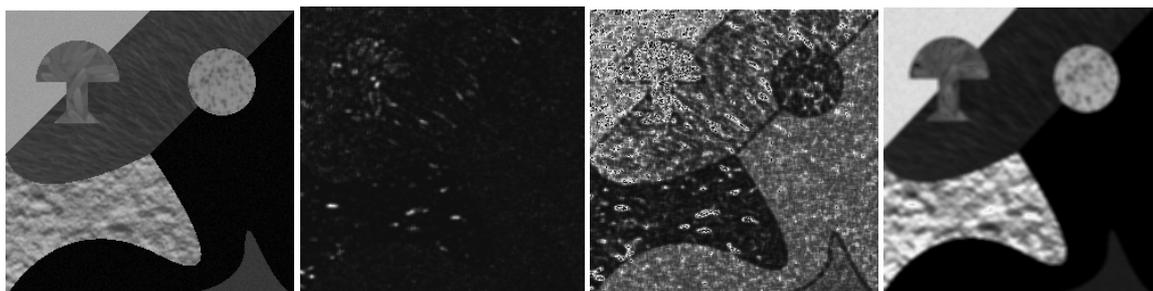
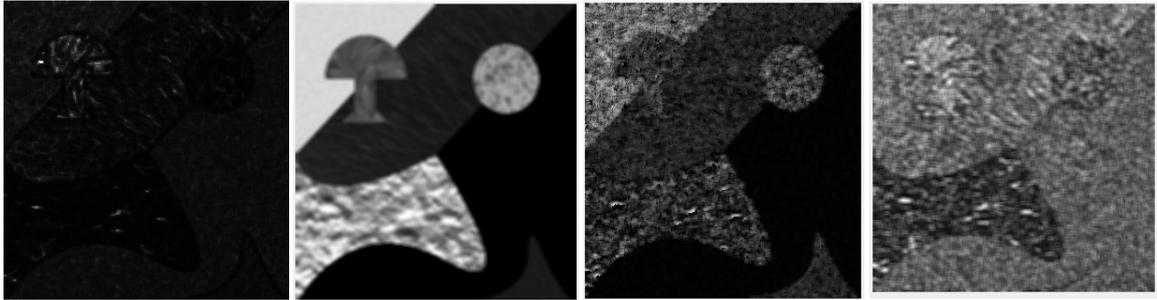


Image originale

Energie

Variance

Covariance



Moyenne Auto-corrélation Somme des moyennes Homogénéité

Figure B.2 : Images attributs pour différentes type d'images

NB : Les valeurs des attributs ont été normalisés et multiplier par 255 pour les visualiser.

Dans la deuxième étape, nous avons opté pour la segmentation de différentes images en utilisant un attribut à la fois en appliquons le mean-shift et le classificateur Fuzzy C-Means.

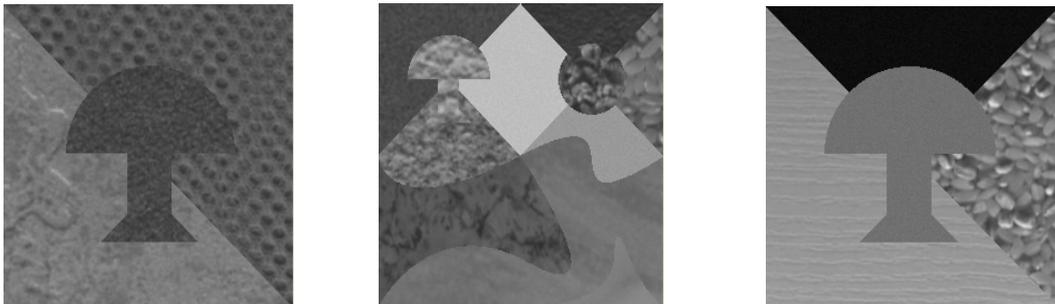
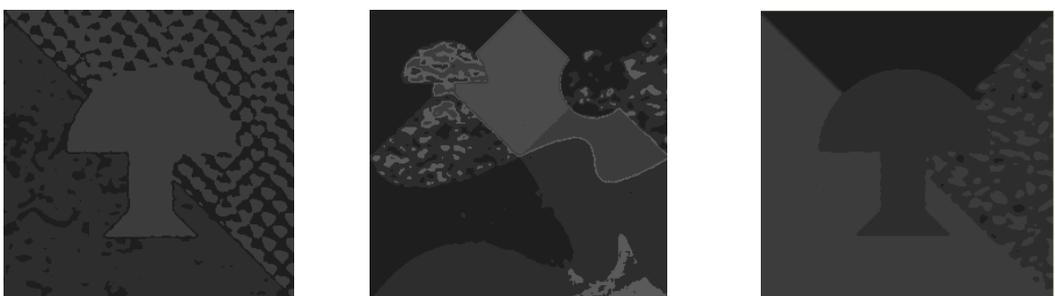


Figure B.3 : Images originales

❖ Segmentation par Variance :



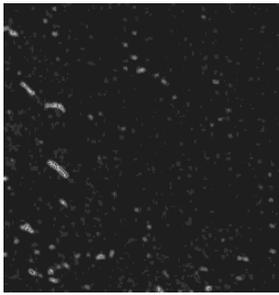
(hs= 13, hr=27)

(hs= 13, hr=14)

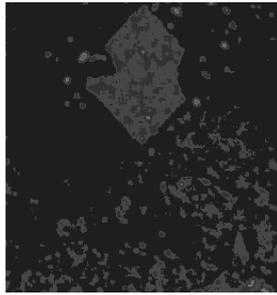
(hs= 13, hr=21)

Figure B.4 : Segmentation par calcul de variance

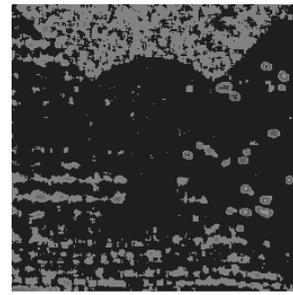
❖ Segmentation par Energie :



($hs= 5, hr=6$)



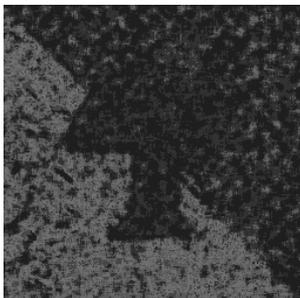
($hs= 5, hr=6$)



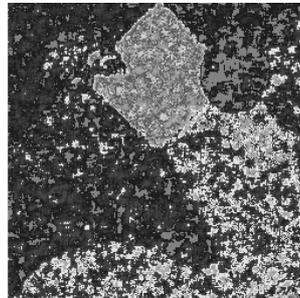
($hs= 13, hr=21$)

Figure B.5 : Segmentation par calcul d'énergie

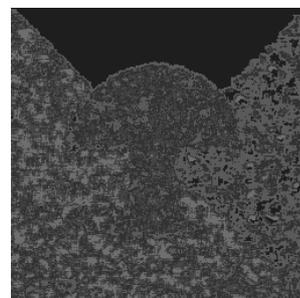
❖ Segmentation par Somme des moyennes :



($hs= 7, hr=9$)



($hs= 5, hr=6$)



($hs= 7, hr=9$)

Figure B.6 : Segmentation par calcul de somme des moyennes

❖ Segmentation par Covariance :



($hs=11, hr=11$)



($hs=15, hr=14$)($hs=15, hr=14$)

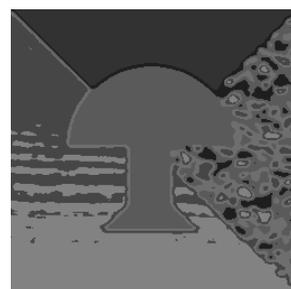


Figure B.7 : Segmentation par calcul de covariance

A partir de ces résultats, nous avons retenu que quelques attributs comme : la variance, la covariance et l'auto-corrélation peuvent être utilisés pour la segmentation des images faiblement texturées. En revanche, pour le cas des images fortement texturées, il est nécessaire d'utiliser plusieurs attributs.

REFERENCES

- [1] Kermad, C.D., "Segmentation d'images: Recherche d'une mise en œuvre automatique par coopération de méthodes", LASTI-Groupe Image, ENSSAT, Lannion, Université de Rennes I, France,(1997).
- [2] Carolina, A., Claudia, A., "Gabor Filter Applied in Supervised Classification of Remote Sensing Images", Université de Sao Paulo, Brazil (2005).
- [3] Alexandra, B., Denis, L. and Christian, M., "Tumour Detection in MR Liver Images by Integrating Edge and Region Information", ESAIM Proceedings, vol 12, (Novembre 2002), 17-24.
- [4] Christophe, R., "Mise en œuvre d'un système adaptatif de segmentation d'images", ENSSAT, Université de Rennes I, France, (1999).
- [5] Zucker, S.W., "Region growing: Childhood and adolescence", Graphics and Image Processing, vol.5, n°3, (1976), 382–399.
- [6] David, A. Clausi and Deng, H., "Fusion of Gabor Filter and Co-occurrence probability Features for Texture Recognition", Systems Design Engineering, University of Waterloo, (2003).
- [7] Laurienne, S., "Fusion d'approches non-supervisées et génériques pour la segmentation thématique", Université d'Avignon, (2000).
- [8] Deriche, R., "Using Canny's criteria to derive a recursively implemented optimal edge detector", Int. Journal of Computer Vision, vol.1, n°2, (1987) , 167-187.
- [9] Prewitt, J.M.S., "Object enhancement and extraction", In Picture Processing and Psychopictorics, Academic Press, New York, (1970), 76-149.
- [10] Roberts, L.G., "Machine perception of three-dimensional solids", In Optical and Electro-Optical Information Processing, MIT press, Cambridge, Massachusetts, (1965), 159-197.
- [11] Kirsch, R., "Computer determination of the constituent structure of biological images", Computer biomedical research, (1971), 315-328.
- [12] Marr, D. and Hildreth, E., "Theory of edge detection", In Proceedings of the Royal Society of London, vol.207, (1980), 187-217.
- [13] Canny, J., "A computational approach to edge detection", IEEE Trans, On PAMI, (Nov 1986), 679-698.
- [14] Deriche, R., "Techniques d'extraction de contours", INRIA Sophia-Antipolis, (2004).

- [15] Castan, S. and Shen, J., "An optimal linear operator for step edge detection", CVGIP, (1992), 112-133.
- [16] Huertas, A. and Medioni, G., "Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks", T-PAMI, vol.8, n°5, (September 1986), 651-664.
- [17] Peterfreund, N., "Robust tracking of position and velocity with Kalman snakes. IEEE Trans", Pattern Anal, Machine Intell, vol.21, n° 6, (1999), 564-569.
- [18] Adams, R. and Bischof, L., "Seeded region growing", IEEE Trans, on PAMI, vol.16, n°6, (June 1994), 641 -647.
- [19] Haifeng, C. and Peter, M., "Robust Fusion of Uncertain Information", IEEE transactions on systems, man and Cybernetics-Part B, vol. 35, n°3, (2005).
- [20] Xiaogang, W., Weiliang, Q. and Ruben, H., "An Iterative Non-parametric Clustering Algorithm Based on Local Shrinking", Department of Mathematics and Statistics, York University, Department of Statistics, University of British Columbia.
- [21] Hansen, W. and Higgins, W.E., "Relaxation methods for supervised image segmentation", IEEE trans, Pat.An.Ma.In, (September 1997), 949-962.
- [22]. Busin, L. Vandenbroucke, N., Macaire, L. et Postaire, J.G., "Color space selection for unsupervised color image segmentation by histogram multithresholding", Proceedings of the IEEE International Conference on Image Processing ICIP, (2004).
- [23] Bernsen, J., "Dynamic thresholding of grey-level images", In Proc, Eighth Int 'l Conf on Pattern Recognition, (1986), 1251-1255.
- [24] Kapur, J.N., Sahoo, P.K. and Wong, A.K.C., "A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram", Computer Vision, Graphics and Image Processing, vol. 29, (1985), 273-285.
- [25] Otsu, N., "A threshold selection method from gray-level histogram", IEEE Trans. Syst, Man Cybern, vol.9, (1979), 62-66.
- [26] Haralick, R., Shanmugan, K. and Dinstein, I., "Textural features for image classification", IEEE Transactions on SMC, vol. 3, n°6, (1973), 610-621.
- [27] Ameer, A., Karmkar, C. and Laurence, S., "Image segmentation using Fuzzy Clustering Incorporating Spatial Information", Gippsland School of Computing & Information Technology, Monash University, Australia.
- [28] Dzung, L. and Jerry, L., "Adaptive Fuzzy Segmentation of Magnetic Resonance Images", IEEE transactions on medical imaging, vol.18, n° 9, (1999).
- [29] Zadeh, L.A., "Fuzzy sets", Inf and control, (1965), 338-353.

- [30] Lei, J. And Wenhui, Y., "A Modified Fuzzy C-means algorithm for Segmentation of magnetic resonance images", Biomedical Engineering Division, Institute of Electrical Engineering Chinese Academy of Sciences, Beijing, P.R.China.
- [31] Chehikian, A., "Recursive filters for gradient estimation and contour extraction using B-spline interpolation", *Traitement du signal*, vol.4,29, (1997).
- [32] Wemmert, C., Puissant, A., Forestier, G., and Gançarski, P., "Multiresolution remote sensing image clustering", *Geosciences and Remote Sensing Letters*, (In Press), (2009).
- [33] Guillaume, R., Xavier, D., Frédéric, F. et Josiane, Z., "La Poursuite de Projection pour la Classification d'Image Hyperspectrale Texturée", *Rapport de recherche-INRIA*, (2001).
- [34] Jain, A.K. and Tuceryan, M., "Texture analysis", Chapter 11 in the *Handbook of pattern recognition and computer vision* by C.H.Chen (1992).
- [35] Haralick, R., "Statistical and structural approaches to texture", *Proc. of IEEE*, vol. 67, n°5, (May 1979), 786-804.
- [36] Bovic, A.C., Clark, M. and Geisler, W.S., "Multichannel texture analysis using localized spatial filter", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.12, (January 1990).
- [37] Assogba, K., "Analyse et reconnaissance des formes des vaisseaux rétiniens pour la quantification de leurs modifications", *Thèse de Docteur de l'Université Paris XII*, (Mars1999).
- [38] Dengsheng, Z., Aylwin, W., Maria, I. and Guojun, L., "Content-based Image Retrieval using Gabor Texture Features", *Gippsland School of Computing and Information technology, Monach University, Australia*.
- [39] Kaneko, H., "A Generalized Fractal Dimension and Its Application to Texture Analysis", *IEEE*, (1989), 1711-1714.
- [40] Spinu, C., Garbay, C. and Chassery, J.M., "Une approche coopérative et adaptative pour la segmentation d'images", In *GRESTI*, (Septembre 1995), 609-612.
- [41] Germond, L., "Trois Principes de Coopération pour la Segmentation en imagerie de résonance magnétique cérébrale", *Université Joseph Fourier, Grenoble 1*, (1999).
- [42] Kermad, C.D. and Chehdi, K., "Automatic image segmentation system through iterative edge-region cooperation", *Image and Vision Computing*, vol. 20, (2002), 541-555.
- [43] Monga, O. et Wrobel, B., "Segmentation d'images : vers une méthodologie", *Traitement du signal*, vol.4, n° 3, (1987), 169-193.

- [44] Bertolino, P. et Montanvert, A., "Coopération régions contours multirésolution en segmentation d'image", dans Actes du 10e Congrès AFCET/Reconnaissance des formes et intelligence artificielle, Rennes, (janvier 1996), 299-307.
- [45] Gambotto, J.P., "A new approach to combining region growing and edge detection", Pattern Recognition Letters, vol.14, n°11, (1993), 869-875.
- [46] Pavlidis, T. and Liow, Y.T., "Integrating Region Growing and Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.12, n°3, (1990), 225-233.
- [47] Kermad, C.D., Chehdi, K. and Cariou, C. "Segmentation d'images par multi-seuillage et fusion des régions labélisées minimisant un critère de similarité", In processing of GRETSI 95, (Septembre 1995), 641-644.
- [48] Jain, A.K and Farrokhnia, F., "Unsupervised texture segmentation using Gabor filters", Pattern Recognition, vol. 24, (1991), 1167-1186.
- [49] Bonnin, P., Hayot, J.C. and Zavidovique, B., "A new edge point/region cooperative segmentation deduced from a 3D scene reconstruction application".
- [50] Haddon, J.F. and Boyce, J.F., "Texture classification of segmented regions of flir images using neural networks", In IEEE International Conference on Image Processing, vol.69, (May 1994), 786-804,
- [66] Kara Falah. R. and Bolon, Ph., "Une technique d'intégration des résultants en segmentation d'images", In GRETSI, vol.2, (Septembre 1995), 573-576.
- [52] Chabbi, H. et Massini, G., "Construction de facettes 3D à partir des contours et des régions, en utilisant les invariants projectifs", dans Actes du 9e Congrès AFCET/Reconnaissance des Formes et Intelligence Artificielle, Paris, vol.1, (1994), 303-314.
- [53] Bajcsy, R., "Active perception", Proceedings of the IEEE, (August 1988), 996-1005.
- [54] Baujard, O. et Garbay, C., "Un système de vision multi-agents", Actes du 8ème congrès de Reconnaissance des Formes et Intelligence Artificielle, Lyon-Villeurbanne, (Novembre 1991), 89-98.
- [55] Salzenstein, F. and Pieczynski, W., "Choix automatique de la meilleure méthode de segmentation statistique non supervisée d'images", In GRETSI, vol.2, (Septembre 1995), 625-628.
- [56] <http://pages.upf.pf/Sebastien.Chabrier/ressources.php>.
- [57] Brodatz, P., "Textures: A Photographic Album for Artists and Designers", Dover Publications, USC-SIPI, (1966).
- [58] Ping-Sung, L., Tse-Sheng, C. and Pau-Choo, C., "A Fast Algorithm for Multilevel Thresholding", J. Inf. Sci. Eng., vol.17, n° 5, (2001), 713-727.

[59] http://urfist.enc.sorbonne.fr/anciensite/image_numerique/segmentation.htm

[60] http://fr.wikipedia.org/wiki/Méthode_d'Otsu

[61] Bommana, R., Ramasubba, R., Swarnamani, S. and Suesh, S., “Study on Ultrasound Kidney Images for Implementing Content Based Image Retrieval System using Regional Gray-Level Distribution”, Proc. of Intl. Conference, Italy, (2003), 1-4.

[62] Fukunaga, K. and Hostetler, L.D., “The estimation of the gradient of a density function, with applications in pattern recognition”, IEEE Trans. Information Theory, vol.21, (1975), 32-40.

[63] Cheng, Y., “Mean shift, mode seeking, and clustering”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.17, n°8, (1995), 790–799.

[64] Comaniciu, D. and Meer, P., “Mean shift: A robust approach toward feature space analysis”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, n°5, (2002), 603–619.

[65] Keselman, Y. et Micheli-Tzanakou, E., “Extraction and characterization of regions of interest in biomedical images”, Dans Information Technology Applications in Biomedicine, ITAB98, Proceedings, IEEE International Conference, (May 1998), 87–90.

[66] Mayer, A. and Greenspan, H., “Segmentation of brain MRI by adaptive mean shift”, In International Symposium on Biomedical Imaging : Macro to Nano, (Avril 2006), 319–322.

[67] Bogdan, G., Ilan.S. and Peter, M., “Mean Shift based Clustering in high dimensions”, Rutgers University.

[68] Cocquerez, J.P. et Philip, S., “Analyse d’images : filtrage et segmentation”, Edition Dunod, (1995).

[69] Pierre-Martin, T., “Segmentation d’images de texture par des modèles multirésolutions”, La Faculté des études supérieures de l’Université Laval, Département de génie électrique et de génie informatique, Québec, (Juin 2000).

[70] Levesque, V., “Texture segmentation using Gabor filters”, Center For Intelligent Machines, McGill university, (December 2000).

[71] Xiaogang, W., Weiliang, Q. et Ruben, H., “An Iterative Non-parametric Clustering Algorithm Based on Local Shrinking”, Department of Mathematics and Statistics, York University, Department of Statistics, University of British Columbia.

[72] Karthanassi, V., Iossifidis, C. and Rokos, D., “A texture based classification method for classifying build areas according to their density”, International Journal of Remote Sensing, vol.21, n°9, (2000), 1807-1823.

- [73] Berkner, K. and Wells, R.O., "A new hierarchical scheme for approximating the continuous wavelet transform with applications to edge detection", *IEEE Signal Processing, Letters*, vol.6, n°8, (Aout 1999), 148-153
- [74] Wai-Man P., "GPU-friendly Design for Computation of Per-pixel Gabor Wavelet Texture Features", Spatial Media Group, Computer Arts Lab, University of Aizu, Aizu-wakamatsu, Japan, The 3rd International Conference on Machine Vision (ICMV), (2010).
- [75] Chaozhe.Z, Tianzi.J. Multicontext fuzzy clustering for separation of brain tissues in magnetic resonance images. Department of Computer Science, University of Houston, USA, 2002.
- [76] Benmahammed, K., Moussaoui, A., Ferahta, N. et Chen, V., "A New MR Brain Image Segmentation using An Optimal Semi-Supervised Fuzzy C-means and pdf Estimation", Laboratoire systèmes intelligents, Université Ferhat Abbas, Algérie, Laboratoire d'automatique, Université de Reims, France, (2005).
- [77] Cheng, H.D., Jiang, X.H., Sun, Y. et Wang, J., "Color image segmentation: advances and projects", *Pattern Recognition*, vol.34, (2001), n°9, 2259-2281.
- [78] Dunn, J.C.A., "Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics*, (1973), 32-57.
- [79] Bezdek, J.C., "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, (1981).
- [80] Y.J. Zhang, «A survey on evaluation methods for images segmentation", *Pattern Recognition*, Vol. 29, No. 8.
- [81] Chabriei, S., "Évaluation de la segmentation", Laboratoire de Vision et Robotique, Université d'Orléans, Bourges.
- [82] Chalana, V. and Kim, Y., "A methodology for evaluation of boundary detection algorithms on medical images", *IEEE Trans, On Medical Imaging*, (October 1997).
- [83] Laurent, H., Chabrier, S., Rosenberger, C., et Marché, E.P., "Etude comparative de critères d'évaluation de la segmentation", Laboratoire Vision et Robotique ENSI de Bourges, Université d'Orléans, France
- [84] Levine, M. and Nazif, D., "Dynamic measurement of computer generated images segmentation", *PAMI*, (March 1985), 155-164.
- [85] Zeboudj, R., "Filtrage, seuillage automatique, contraste et contours : du pré-traitement à l'analyse d'image", PhD thesis, Université de Saint-Etienne, (1988).
- [86] Jimnez-Alaniz, J., Medina-Bauelos, M. et Yez-Surez, O., "Data-driven brain MRI segmentation supported on edge confidence and a priori tissue information", *IEE Transactions on Medical Imaging*, (Janury 2006), 74–83.

- [87] Bonnin, P., Hoeltzener, B. and Pissaloux, E., "A new way of image data fusion: the multi-spectral cooperative segmentation", IEEE International Conference on Image Processing, vol.3, Washington, (October 1995), 572-575.
- [88] Carstensen, J.M., "Cooccurrence feature performance in texture classification", In Scandinavian conference on image analysis, vol.1, (1993), 831-838.
- [89] Daniel, F., "An analysis of recent work on Clustering Algorithms", Department of Computer Science & Engineering, University of Washington, (1999).
- [90] Dorin, C. And Peter, M., "Mean shift: A robust approach towards feature space analysis", IEEE transactions on pattern analysis and machine intelligence, vol.24, n°.5, (2002).
- [91] Hofmann, T., "An optimization approach to unsupervised hierarchical texture segmentation", IEEE in Conf on Image Processing, Santa Barbara, (October 1997)
- [92] Jean-Loïc, R., "Croissance de région variationnelle et contraintes géométriques tridimensionnelles pour la segmentation d'image", L'Institut National des Sciences Appliquées de Lyon, (Décembre 2008).
- [93] Leena, L., Iivari, K. Jorma, A. and Ari, V., "Classification Method for Colored Natural Textures using Gabor Filtering", Tampere University of Technology.
- [94] Robert, J. and Torbjorn, E., "ICA Filter Bank for Segmentation of Textured Images", Université de Tromsø, Norway.
- [95] Vincent, L., "Texture Segmentation using Gabor Filter", Center For Intelligent Machines, McGill University, (2000).
- [96] <http://users.polytech.unice.fr>
- [97] <http://docs.gimp.org/fr/filters.html>.
- [98] <http://perso.telecom-paristech.fr/~maitre/BETI/croissance/index.html>.