

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOU MAMMERI, TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT AUTOMATIQUE

MEMOIRE DE MAGISTER

en Automatique

Option : **Traitement d'Images et Reconnaissance de Formes**

présenté par

HALICHE Zohra

ingénieur U.M.M.T.O.

Classification et segmentation d'images texturées basée sur la théorie des ensembles

Mémoire soutenu le

devant le jury d'examen composé de :

DIAF Moussai

Professeur à l'U.M.M.T.O.

Président

HAMMOUCHE Kamal

M.C.A. à l'U.M.M.T.O.

Rapporteur

DJENNOUNE Saïd

Professeur à l'U.M.M.T.O.

Examineur

AHMED-OUAMAR Rachid

M.C.A. à l'U..M.M.T.O.

Examineur

Remerciements

Ce présent travail a été effectué au sein du Laboratoire Robotique et Vision de la faculté Génie Electrique et Informatique de l'université Mouloud Mammeri de Tizi-Ouzou.

J'exprime mes sincères remerciements à mon promoteur, Monsieur Hammouche Kamal pour m'avoir guidée et encouragée le long de ce travail. Je voudrais aussi le remercier pour sa disponibilité permanente, ses discussions fructueuses, sa patience, mais aussi pour l'aide qu'il m'a offert pendant la réalisation de ce travail. Je pense que je ne le remercierai jamais assez.

Ma profonde gratitude s'adresse à Monsieur Diaf Moussa professeur à l'UMMTO pour m'avoir donné la chance de travailler dans son équipe de recherche et d'avoir mis à ma disposition tous les moyens nécessaires. Je le remercie aussi d'avoir accepté de présider le jury de ce mémoire.

Je tiens aussi à remercier les membres de jury, Monsieur Djenoune Saïd Professeur à l'UMMTO et Monsieur Ahmed Ouamar Rachid Maître de conférences A à l'UMMTO pour l'honneur qu'ils me font en participant au jury, et qui ont pris la peine de lire avec soin ce mémoire pour juger son contenu.

Mes remerciements s'adressent aussi à tous les membres du Laboratoire Robotique et Vision.

Je ne saurais terminer, sans remercier les membres de ma très chère famille qui m'ont toujours soutenu et cru en moi, ainsi que tous mes amis qui étaient là pour moi quand j'avais réellement besoin d'eux.

SOMMAIRE

Introduction générale	1
Chapitre 1 : Analyse et segmentation d'images texturées.	
1.1 Introduction	4
1.2 Définition de la texture	4
1.3 Méthodes d'analyse de la texture	6
1.3.1 Approche structurelle	6
1.3.2 Approche fréquentielle.....	6
1.3.2.1 Transformée de Fourier.....	7
1.3.2.2 Les masques de Laws.....	8
1.3.2.3 Filtres de Gabor.....	9
1.3.2.4 Transformée en ondelettes.....	9
1.3.3 Approche statistique.....	10
1.3.3.1 Méthode statistique du 1 ^{er} Ordre.....	11
1.3.3.2 Méthode des matrices de cooccurrence.....	11
1.3.3.3 Méthode des matrices de cooccurrence généralisées.....	15
1.3.3.4 Méthode des histogrammes somme et différence des niveaux de gris.....	16
1.3.3.5 Méthode LBP.....	18
1.3.3.6 Fonction d'Auto-corrélation.....	18
1.3.4 Méthodes basées sur un modèle.....	19
a)Modélisation linéaire de la texture.....	19
b) Champs de Markov.....	20
- Champs de Gibbs	21
- Modèle Markovien binomial.....	23
1.3.5 Morphologie mathématique.....	25
1.3.5.1 Transformations morphologiques binaires de base	25
- Erosion binaire.....	25
- Dilatation binaire.....	25
- Ouverture et fermeture binaires.....	26
- Gradient morphologique binaires.....	26
1.3.5.2 Transformations morphologiques des images en niveaux de gris	27
- Erosion.....	27

- Dilatation.....	28
- Gradient	28
- Ouverture et fermeture des images au niveau de gris.....	28
1.3.5.3 Analyse de la texture par morphologie mathématique.....	29
- Granulométrie et distribution de taille.....	29
1.4 Classification des images	30
1.4.1 Classification par la méthode bayésienne.....	31
1.4.2 Méthode des k plus proches voisins	32
1.5 Segmentation d'images texturées.....	33
1.5.1 Segmentation par analyse des propriétés spatiales	34
- Croissance des régions.....	35
- Division-Fusion	35
1.5.2 Segmentation par classification des pixels.....	36
- Algorithme des <i>K</i> -means.....	36
- Algorithme de Fuzzy C-means.....	37
1.6 Conclusion.....	39

Chapitre 2 : Etude théorique du concept Aura

2.1 Introduction	40
2.2 Ensemble Aura	40
2.2.1 Propriétés des ensembles Aura.....	41
2.3 Notion de voisinage.....	43
2.3.1 Système de voisinage symétrique et asymétrique	45
2.4 Mesure Aura	45
2.4.1 Propriétés de la mesure Aura.....	46
2.5 Intérêt pratiques des mesures Aura.....	47
2.6 Mesures Aura sur les sous-voisinages.....	48
2.7 Matrice Aura.....	48
2.7.1 Propriétés de la Matrice Aura.....	49
2.8 Matrice Aura des niveaux de gris.....	49
2.9 Relation entre les ensembles Aura et la morphologie mathématique.....	50
2.10 Relation entre la matrice Aura et la matrice de cooccurrence.....	52
2.11 Relation entre la matrice Aura et la matrice de covariance	53
2.12 Relation entre la matrice Aura et les champs aléatoires de Markov.....	54

2.13 Mesure de la similarité de deux images à partir des GLAMs.....	56
2.14 Matrices Aura de niveaux de gris symétriques (SGLAMs).....	57
2.15 Matrices Aura de niveaux de gris de base.....	57
2.16 Mesure de similarité d'images à partir de leurs BGLAM	59
2.17 Conclusion	59

Chapitre 3 : Segmentation d'images à base des matrices Aura

3.1 Introduction.....	60
3.2 Méthode classique de calcul des matrices Aura d'une image.....	60
3.3 Extraction des attributs locaux de texture basée sur des matrices Aura	62
3.3.1 Calcul des matrices Aura à base des listes chaînées	64
3.3.1.1 Listes chaînées.....	64
3.3.1.1.1 Listes chaînées simples.....	64
3.3.1.1.2 Listes chaînées doubles.....	65
3.3.1.2 Application des listes chaînées au calcul des matrices Aura.....	66
3.3.1.3 Attributs de texture calculés à partir de la liste chaînée.....	69
3.3.2 Calcul des matrices Aura à base d'une structure hybride.....	71
3.4 Choix d'une structure de voisinage.....	73
3.4.1 Utilisation d'une seule structure de voisinage.....	75
3.4.2 Utilisation de plusieurs structures de voisinage de même forme et de tailles variables.	76
3.4.3 Utilisation des matrices Aura de base.....	76
3.5 Conclusion.....	77

Chapitre 4 : Tests et résultats

4.1 Introduction.....	78
4.2 Classification des images texturées.....	79
4.3 Segmentation d'images texturées basée sur la matrice Aura de niveaux de gris.....	80
4.3.1 Critères d'évaluation.....	80
4.3.2 Attributs texturaux extraits à partir d'une matrice Aura.....	81
4.3.2.1 Utilisation d'un seul élément structurant.....	81
4.3.2.2 Utilisation de plusieurs structures de voisinage de même forme et de tailles variables.	93
4.4 Utilisation des matrices Aura de base.....	98

4.5 Comparaison avec les attributs texturaux extraits à partir d'une matrice de cooccurrence.....	100
4.6 Images synthétiques et réelles.....	103
4.7 Conclusion.....	111
Conclusion générale.....	112

Références

Annexes

Introduction générale

Introduction générale

Le traitement d'images est une discipline des mathématiques appliquées qui connaît un fort essor depuis quelques décennies grâce notamment à l'amélioration, la miniaturisation et le faible coût des systèmes d'acquisition d'images. Il couvre des applications extrêmement variées comme le traitement des images de télédétection afin de discerner un champ, d'une forêt, d'une ville. Un autre type de traitement permet de classifier les différents types de nuages à partir d'images satellitaires, des mers de glace à partir des images radar ou encore des fonds marins à partir des images sonar. En robotique le traitement d'images permet le tri automatique d'objets, l'inspection et le contrôle de qualité. Dans le domaine médical, on fait appel à l'analyse d'images de modalité différentes (IRM, scanner, échographie, scintigraphie) pour la détection de lésions, le dépistage de pathologies, la reconnaissance des différentes cellules entre autres. En biométrie, on utilise l'image du visage, de l'iris ou de l'empreinte pour l'identification des personnes. D'autres domaines aussi variés que ceux qui ont trait à l'astronomie, la pharmacologie ou encore à la géographie et à l'analyse des documents utilisent l'imagerie comme un moyen d'analyse rapide, sure, fiable et économique.

La plus part des applications requièrent l'identification ou l'extraction préalable des différents objets qui composent l'image à traiter. De ce besoin est née la segmentation d'images qui est sans doute la tâche la plus cruciale de l'analyse d'images. Elle demeure encore l'un des principaux thèmes de recherche en vision artificielle. Formellement, la segmentation est définie comme l'opération de partitionnement de l'image en régions homogènes. Certaines méthodes de segmentation exploitent uniquement l'information spectrale (niveaux de gris). Cependant, vue la nature et la complexité des images, l'utilisation de cette seule information peut s'avérer insuffisante dans plusieurs cas. On a alors recours à une autre information plus riche qui est la texture.

En traitement d'image, la texture est considérée comme une caractéristique très importante, due à sa présence dans la quasi-totalité des images naturelles. Elle apporte une information sur l'agencement spatial des niveaux de gris. La segmentation des images nécessite par conséquent une analyse de la texture de l'image en question afin d'en extraire ses caractéristiques. A cet effet, plusieurs techniques d'analyse d'images texturées, basées sur des concepts très variées, ont été proposées. Parmi elles, on

retrouve les matrices de cooccurrence, les champs aléatoires de Markov, la morphologie mathématique et les matrices Aura. Notre intérêt dans ce mémoire est l'analyse de la texture basée sur les matrices Aura des niveaux de gris. Elles offrent un cadre théorique général pour l'analyse des images texturées, elles ont l'avantage d'être liées aux matrices de cooccurrence, aux matrices de covariance, aux champs aléatoires de Markov et à la morphologie mathématique. Ces matrices décrivent les relations spatiales des niveaux de gris d'un pixel et ses pixels voisins. La notion de voisinage prend alors une importance capitale, elle est définie par un système de voisinage qui peut être représenté à l'aide des éléments structurant noté E .

Historiquement, le concept Aura a été introduit pour la première fois en 1990 par Elfadel et Picard dans le cadre de la synthèse des textures [7][9][12][16]. Cette approche, qui s'inspire de la miscibilité des fluides, est basée sur la théorie des ensembles et définit de nouvelles notions telles que l'ensemble Aura, mesure Aura et matrice Aura ou matrice de miscibilité. On assiste actuellement à un regain d'intérêt de cette approche. En effet, les matrices aura ont été récemment utilisées comme une mesure de similarité dans le cadre de la recherche d'images [5], pour synthétiser des textures bidimensionnelles [6][10] et tridimensionnelles [11] ou pour la classification des images de texture [15][18][19].

Notre objectif principal dans ce mémoire est d'exploiter les matrices Aura des niveaux de gris en vue de la classification et de la segmentation d'images texturées. La classification des images se fait d'une manière supervisée, elle est souvent sollicitée pour résoudre des problèmes de reconnaissance des formes (visages, empreintes, etc...). Elle consiste à identifier une image dans une base d'images où l'appartenance de chaque image de cette base à une classe est connue à priori.

Notre démarche en segmentation consiste en première lieu, à caractériser chaque pixel de l'image par un ensemble d'attributs à partir des matrices Aura des niveaux de gris. Ensuite, les pixels ayant des attributs similaires sont regroupés en classes de textures en utilisant une des méthodes de classification non supervisée. Le calcul de la matrice Aura dépend de la forme, du type (symétrique ou non) et de la taille des éléments structurants E . Plusieurs stratégies sont proposées pour utiliser un ou plusieurs éléments structurants.

Le premier chapitre dresse un panorama sur les méthodes d'analyse de la texture. Une attention particulière est apportée aux méthodes d'analyse basées sur les matrices de cooccurrence, les champs aléatoires de Markov et la morphologie mathématique à

cause de leur lien avec les matrices Aura. Un bref aperçu sur les méthodes de segmentation d'images texturées est également présenté. La fin du chapitre introduit deux algorithmes de classification non supervisée à savoir K-means et Fuzzy C-means qui sont utilisés lors de la phase de regroupement des pixels.

Le deuxième chapitre est consacré à l'étude théorique, d'une manière exhaustive, du concept Aura. Celui-ci s'articule autour des notions d'ensemble Aura, mesure Aura et matrice Aura. Les définitions et les propriétés de ces trois notions sont particulièrement énoncées. Le lien entre les matrices Aura et les matrices de cooccurrence, les champs aléatoires de Markov, les matrices de covariance et la morphologie mathématique est établi. Leurs intérêts en analyse d'images, notamment pour mesurer la similarité entre deux images, seront également présentés.

Le troisième chapitre est dédié à l'application du concept Aura dans le cadre de la segmentation d'images texturées. Le calcul classique d'attributs pour chaque pixel à partir des matrices Aura des niveaux de gris exige un temps de calcul assez long. Pour remédier à ce problème, nous avons fait appel à deux techniques de programmation. La première est la technique des listes chaînées et la deuxième dite structure hybride qui combine à la fois des listes chaînées et des tables de hachage. Plusieurs stratégies de choix des éléments structurants pour le calcul des matrices Aura est également abordé à la fin de ce chapitre.

Le chapitre quatre expose les principaux résultats obtenus sur des images synthétiques et réelles par l'approche proposée, accompagnés de leurs interprétations. L'influence des différents éléments structurants et stratégies de calcul sont évaluées.

La conclusion générale synthétise notre travail et donne les perspectives qui en découlent.

Deux annexes clôturent ce mémoire. La première contient des rappels succincts sur la théorie des ensembles et la deuxième présente les preuves de quelques propriétés et théorèmes énoncés dans le deuxième chapitre.

Chapitre 1

Analyse et segmentation d'images texturées

1.1 Introduction

La texture est souvent employée pour décrire l'état d'une surface d'un objet. Elle joue ainsi un rôle prépondérant dans l'analyse et l'interprétation d'une grande variété d'images [38]. Malgré cette importance, il n'existe pas de définition universelle car la texture est une notion subjective, relative à la perception visuelle de l'être humain et pouvant être différente d'un individu à l'autre. Nous rappellerons par la suite les différentes approches de l'analyse de la texture. Quelques méthodes d'analyse seront brièvement exposées. Quelques méthodes de classification et de segmentation d'images texturées seront également présentées.

1.2 Définition de la texture

La texture est une caractéristique intuitive facile à reconnaître mais difficile à définir. Une littérature riche et abondante montre qu'il existe plusieurs définitions de la texture. Toutefois, trois types d'orientations de la texture se dégagent.

L'orientation qualitative est celle qui respecte le mieux les origines linguistiques du mot texture. Le dictionnaire Robert définit le mot texture comme venant du latin *textura* pour décrire la disposition des files d'une chose tissée. Dans cette orientation, on décrit les textures selon leur apparence fine ou grossière, bosselée ou striée, régulière ou irrégulière, ridée ou granuleuse, isotrope ou anisotrope ..., autant de termes qualificatifs qui sont proches de nos concepts perceptifs.

La deuxième orientation définit la texture en fonction des formes ou d'objets élémentaires qui la constitue. Elle considère la texture comme un phénomène à deux dimensions à partir desquels elle peut être décrite. La première dimension concerne la description de primitives (motifs de base) constituant la texture, la deuxième dimension est relative à la description des relations spatiales ou à l'interaction entre ces primitives. Les primitives correspondent à un pixel ou à des régions connexes de pixels ayant des particularités de niveaux de gris. Par contre, les relations entre ces primitives peuvent être structurales, probabilistes ou fonctionnelles. Les textures qui obéissent à cette orientation sont souvent qualifiées de macro-textures ou structurelles (Fig.1.1).

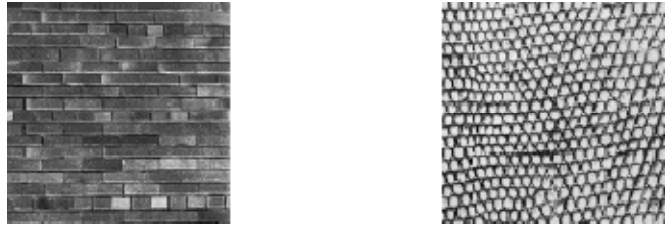


Fig.1.1 : Exemples de textures structurales.

La troisième orientation considère la texture comme un phénomène stochastique ayant un aspect anarchique et désordonné mais homogène. Elle se prête bien à la description des textures à caractère aléatoire où il est difficile d'isoler un motif de base. Les textures répondant à cette orientation sont généralement qualifiées de micro-textures ou aléatoires. La figure (1.2) montre deux exemples de textures aléatoires.

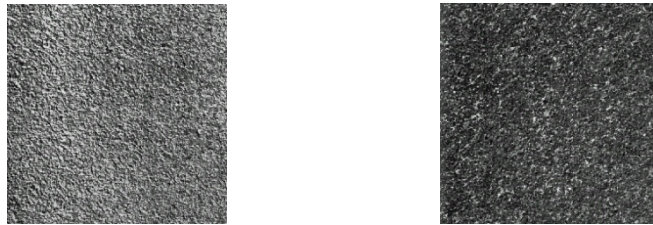


Fig.1.2 : Exemples de textures aléatoires

Hormis ces deux catégories de textures, il en existe d'autres comme, par exemple, les textures directionnelles (voir Fig. 1.3). Ces textures ne sont pas totalement aléatoires et ne présentent pas d'éléments structurants de base.

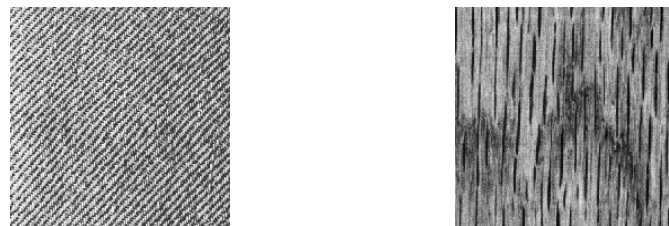


Fig.1.3 : Exemples de textures directionnelles

Afin de mieux appréhender notre travail de segmentation, nous avons retenu principalement les définitions suivantes :

«Une texture est une région de l'image pour laquelle on peut définir une fenêtre de dimension minimale à travers laquelle l'observation se traduit par une impression visuelle identique pour toutes translations possibles de cette fenêtre à l'intérieur de la région considérée».

Aussi, on définit la texture d'un pixel comme étant l'information apportée par l'ensemble des pixels appartenant à son voisinage, alors que, la texture d'une région donnée désigne les caractéristiques de la répartition spatiale des niveaux de gris des pixels dans cette région.

Ces définitions mettent en valeur l'aspect spatial qui doit être pris en compte pour caractériser une texture et l'aspect homogène de la texture dans une région qui la constitue.

1.3 Méthodes d'analyse de la texture

La diversité des images, ainsi que la complexité de donner une définition précise de la texture a permis l'émergence de plusieurs méthodes d'analyse de la texture [38][39][40][41][42]. Ces méthodes ont pour but d'extraire un ensemble d'attributs ou paramètres pouvant décrire les caractéristiques de la texture. Ces attributs doivent être représentatifs, pertinents et discriminants de façon qu'on puisse discerner une texture parmi d'autres. Leur extraction peut être classée essentiellement en cinq approches, l'approche structurelle, l'approche fréquentielle, l'approche statistique, l'approche basée sur les modèles et celle basée sur la morphologie mathématique.

1.3.1 Approche structurelle

L'approche structurelle cherche à extraire et à localiser des primitives des textures [37]. Principalement, ces méthodes utilisent des techniques d'auto-corrélation pour retrouver le placement dans la texture de la primitive initialement extraite, afin d'en déduire une règle de placement. Cette approche ne sera pas plus commentée, étant donné la diversité des techniques employées très liées à la nature des images. Le trait caractéristique de ces méthodes est que toutes se déroulent en 2 étapes, l'extraction de la primitive puis la recherche de la règle de placement.

1.3.2 Approche fréquentielle

Dans ce type d'approche on cherche à décrire une texture à partir de ses caractéristiques fréquentielles.

Une texture douce, pour laquelle on ne décèle pas des variations significatives du niveau de gris sera présentée par un spectre de puissance caractérisé par de fortes amplitudes dans le domaine des basses fréquences. Par contre pour une texture grossière

et granuleuse, dans laquelle les variations du niveau de gris sont brutales et rapprochées, le spectre de puissance présente les plus fortes amplitudes dans le domaine des hautes fréquences.

1.3.2.1 Transformée de Fourier

Cette méthode consiste à extraire les caractéristiques fréquentielles de la texture. Pour une image numérique I de dimension $M \times N$, on définit la Transformée de Fourier Discrète (TFD) par l'expression suivante [37]:

$$F(\mu, \nu) = \frac{1}{M \times N} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} I(k, l) e^{2\pi i \left(\frac{k}{N} \mu + \frac{l}{M} \nu \right)} \quad \text{avec } i^2 = -1 \quad (1.1)$$

L'expression de la transformée de Fourier inverse est :

$$I(k, l) = \frac{1}{M \times N} \sum_{\mu=0}^{N-1} \sum_{\nu=0}^{M-1} I(\mu, \nu) e^{2\pi i \left(\frac{\mu}{N} k + \frac{\nu}{M} l \right)} \quad (1.2)$$

Cette transformation permet de passer d'une représentation de l'image dans le domaine spatiale (coordonnées k et l) à une représentation dans le domaine fréquentielle (coordonnées μ et ν). u et v étant les fréquences spatiales des variations des niveaux de gris. Les termes de fréquences basses de cette expression désignent les variations douces des niveaux de gris de l'image, les termes de hautes fréquences représentent les variations brutales comme dans le cas des contours d'objets.

On peut extraire des attributs de texture à l'aide de la transformée de Fourier comme par exemple l'énergie calculée dans une couronne ou bien en fonction de certaines directions [37]. Le domaine des fréquences est alors divisé en anneaux ou en secteurs angulaires et l'énergie calculée dans ces régions définit alors une caractéristique de la texture (Fig.1.4).

$$f_{r_1, r_2} = \int_0^{2\pi} \int_{r_2}^{r_1} |F(u, v)|^2 dr d\theta \quad \text{avec } r = \sqrt{u^2 + v^2} \quad \text{et } \theta = \arctan\left(\frac{v}{u}\right) \quad (1.3)$$

$$f_{\theta_1, \theta_2} = \int_{\theta_1}^{\theta_2} \int_0^{\infty} |F(u, v)|^2 dr d\theta \quad \text{avec } r = \sqrt{u^2 + v^2} \quad \text{et } \theta = \arctan\left(\frac{v}{u}\right) \quad (1.4)$$

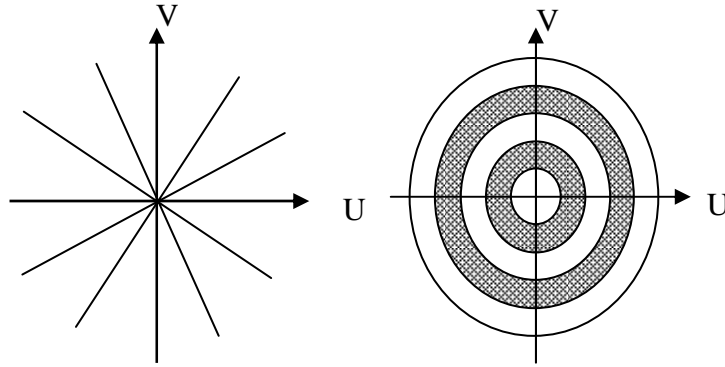


Fig.1.4 : Partitionnements radiaux et angulaires de l'espace de représentation de la Transformée de Fourier.

1.3.2.2 Les masques de Laws

Laws a étudié un ensemble de masques de détection de contours, bords, taches pour caractériser les textures [28]. Ces masques de trois vecteurs de longueur égale à trois, $L_3 = (1, 2, 1)$, $E_3 = (-1, 0, 1)$ et $S_3 = (-1, 2, -1)$, représentent respectivement une moyenne centrale pondérée, une différence symétrique du premier ordre et du deuxième ordre. Des masques de taille (3×3) ou (5×5) ont été générés par convolution à partir des masques précédents. Les filtres obtenus sont de type passe-bas, passe-bande ou coupe-bande, avec ou sans caractéristiques d'orientations.

Sur les images filtrées $I_f(x, y)$, obtenue après convolution de l'image $I(x, y)$ avec les masques, est estimée une énergie locale pour chaque pixel sur une fenêtre de voisinage de largeur $(2w + 1)$. Les caractéristiques spectrales détectées sont en haute fréquence et les orientations relevées sont les directions horizontales, verticales et obliques. Les propriétés statistiques locales (moyenne et variance) de l'énergie sont estimées pour caractériser les différentes textures.

Moyenne

$$V_{moy}(x, y) = \frac{1}{(2w + 1)^2} \sum_{i=x-w}^{x+w} \sum_{j=y-w}^{y+w} I_f(i, j) \quad (1.5)$$

Variance

$$Var(x, y) = \frac{1}{(2w + 1)^2} \sum_{i=x-w}^{x+w} \sum_{j=y-w}^{y+w} |I_f(i, j) - V_{moy}(x, y)| \quad (1.6)$$

1.3.2.3 Filtres de Gabor

Les filtres de Gabor sont des filtres orientés passe-bande qui présentent d'excellentes propriétés de localisation fréquentielles et spatiales. Leur efficacité dans le domaine de l'analyse de la texture a été démontrée dans plusieurs travaux [43] [44][45].

Un filtre de Gabor h , à deux dimensions, peut être représenté comme une fonction gaussienne modulée par une onde plane sinusoïdale [34]:

$$h(x, y) = \exp\left(-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right) \cos(2\pi(u_0x + v_0y) + \Phi) \quad (1.7)$$

où u_0, v_0 et Φ sont respectivement les fréquences et la phase de l'onde plane sinusoïdale. σ_x^2 et σ_y^2 caractérisent l'étendue spatiale du filtre.

Un filtre de Gabor d'orientation arbitraire θ peut être ensuite obtenu en faisant subir une rotation au système d'axes (x, y) .

$$x' = x \cos \theta + y \sin \theta \quad (1.8)$$

$$y' = -x \sin \theta + y \cos \theta \quad (1.9)$$

En faisant varier les différents paramètres ($\theta, \sigma_x, \sigma_y, \mu_0$ et v_0), plusieurs filtres de réponse impulsionnelle h_j sont obtenus. Pour une image I , le résultat I_j est déterminé à la sortie de chacun de ces filtres. Plusieurs caractéristiques peuvent ensuite être extraites de ces images comme, par exemple, la moyenne absolue de la déviation :

$$m_j = 1/n^2 \sum_{x,y}^{n-1} |I_j(x, y)| \quad (1.10)$$

1.3.2.4 Transformée en ondelettes

Les ondelettes sont des familles de fonctions réelles ou complexes. Ce sont des fonctions oscillantes qui sont construites par translation et dilatation d'une ondelette de base $\Psi(t)$ appelée « ondelette mère ». De plus ces fonctions sont de moyenne nulle [33] :

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \quad (1.11)$$

et de norme unitaire :

$$\int_{-\infty}^{+\infty} |\Psi(t)|^2 dt = 1 \quad (1.12)$$

La transformée en ondelettes d'un signal $f(t)$ est une fonction du temps et de l'échelle obtenue par projection du signal sur la famille de fonctions ondelettes :

$$Wf(u, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t) \Psi^* \left(\frac{t-u}{s} \right) dt \quad (1.13)$$

où s est le paramètre d'échelle ou de dilatation, u le paramètre de translation et $\frac{1}{\sqrt{s}}$ est appelé le facteur de normalisation. On note $\Psi_s(u) = \frac{1}{\sqrt{s}} \Psi \left(\frac{t-u}{s} \right)$ est la version translatée et dilatée de l'ondelette mère $\Psi(u)$. « * » dénoté le complexe conjugué.

La transformée en ondelettes est souvent citée dans le cadre des méthodes d'analyse par filtrage linéaire, puisqu'on peut la réécrire comme une convolution :

$$Wf(u, s) = f * \overline{\Psi}_s(\mu) \quad (1.14)$$

$$\overline{\Psi}_s(\mu) = \frac{1}{\sqrt{s}} \Psi^* \left(\frac{-t}{s} \right) \quad (1.15)$$

La transformée en ondelettes peut être étendue au cas bidimensionnel (image). Son application en analyse de la texture se traduit par une décomposition itérative de l'image en quatre sous bandes de résolution inférieure. Ces sous bandes sont obtenues par convolution avec des filtres passe bas et passe haut. On associe ensuite à chaque pixel un vecteur d'attributs de texture correspondant aux coefficients d'ondelettes obtenus dans chaque sous bande [46]. Les coefficients d'ondelettes ne sont pas directement discriminants car leur réponse n'est pas uniforme pour une fréquence donnée. L'information texturale est alors extraite à l'aide d'une estimation locale dans une fenêtre de voisinage.

1.3.3 Approche statistique

Dans ce type d'approche, la nature statistique de la texture est prise en compte dans les procédures d'analyse, car on considère que la texture ne comporte pas des contours forts et ne possède pas un motif de base isolable, mais qu'elle a au contraire une structure aléatoire. Cette approche utilise le domaine spatial et regroupe une multitude de méthodes dont certaines seront présentées en détail dans la suite de ce chapitre.

1.3.3.1 Méthode statistique du 1^{er} Ordre

Les estimateurs statistiques du premier ordre caractérisent la texture au niveau d'un pixel isolé en termes de niveaux de gris et de leur distribution [47] [48]. Par opposition aux estimateurs d'ordre supérieur qui sont basés sur la corrélation des niveaux de gris de deux ou plusieurs pixels en fonction de leur position dans l'image. Les paramètres des estimateurs du premier ordre sont généralement calculés à partir de l'histogramme de l'image.

Soit $g(x,y)$ le niveau de gris du pixel de coordonnées (x,y) , l'histogramme de l'image $p(i)$ associé à $g(x,y)$ est définis par :

$$p(i) = p(g(x,y) = i) = \frac{h(i)}{N}, \quad 0 \leq i \leq G - 1 \quad (1.16)$$

où N est le nombre de pixels de l'image, $h(i)$ la fréquence d'apparition du niveau de gris i et G le nombre de niveaux de gris.

Les attributs de texture calculés à partir de l'histogramme de l'image sont consignés dans le tableau 2. Ils décrivent l'allure de l'histogramme.

<i>Moyenne</i>	$S_M = \frac{1}{N} \sum_{i=0}^{G-1} ih(i)$
<i>Variance</i>	$\sigma^2 = \frac{1}{N} \sum_{i=0}^{G-1} (i - S_M)^2 h(i)$
<i>Energie</i>	$ASM = \frac{1}{N} \sum_{i=0}^{G-1} h(i)^2$
<i>Entropie</i>	$ENT = -\frac{1}{N} \sum_{i=0}^{G-1} h(i) \log [h(i)]$
<i>Skewness</i>	$SKW = \frac{1}{N\sigma^3} \sum_{i=0}^{G-1} (i - S_M)^3 h(i)$
<i>Kurtosis</i>	$Kur = \frac{1}{N\sigma^4} \sum_{i=0}^{G-1} (i - S_M)^4 h(i)$

Tableau 2 : Attributs statistiques du 1^{er} ordre.

1.3.3.2 Méthode des matrices de cooccurrence

La matrice de cooccurrence de niveaux de gris GLCM (Grey Level Cooccurrence Matrix) a été initialement introduite par Haralick [49]. C'est une méthode de second

ordre puisqu'elle permet d'analyser la relation entre les niveaux de gris des pixels pris deux à deux. Elle est considérée comme la méthode de référence dans le domaine de l'analyse de la texture. Une matrice de cooccurrence de niveaux de gris définie comme suit :

Soit $I(x, y)$ la fonction image définie sur un domaine D telle que $I(x, y) = 0, 1, 2, \dots, G - 1$, où G est le nombre de niveaux de gris dans l'image.

Soit R une relation spatiale liant deux pixels. La matrice de cooccurrence du domaine D relative à la relation R est définie par le terme général suivant :

$$t(i, j) = \text{card} \left\{ \begin{array}{l} ((x, y), (x', y')) \in D, \text{ vérifiant } R \\ I(x, y) = i, I(x', y') = j \end{array} \right. \quad (1.17)$$

avec $i = 0, \dots, G - 1$ et $j = 0, \dots, G - 1$.

Chaque élément $t(i, j)$ représente le nombre de couples de points du domaine D liés par la relation R , l'un ayant le niveau de gris i et l'autre le niveau de gris j .

On peut également définir la matrice de cooccurrence normalisée en divisant chaque élément $t(i, j)$ par le nombre total de couples du domaine D vérifiant R .

$$T(i, j, d, \theta) = \frac{t(i, j, d, \theta)}{N} \quad (1.18)$$

tel que

$$N = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T(i, j, d, \theta) \quad (1.19)$$

$T(i, j, d, \theta)$ n'est plus un nombre mais une fréquence relative. C'est la probabilité de trouver un couple de points du domaine D , l'un ayant le niveau de gris i et l'autre de niveau j , vérifiant R . Cette relation spatiale R liant les deux points est définie par Haralick par la distance d entre ces deux points et leur orientation θ par rapport à l'horizontale.

Les matrices de cooccurrence usuelles sont définies par la distance d entre deux points et les quatre directions horizontale ($\theta = 0$) verticale ($\theta = \frac{\pi}{2}$), diagonale gauche ($\theta = \frac{\pi}{4}$) et diagonale droite ($\theta = \frac{3\pi}{4}$) (Fig. 1.5).

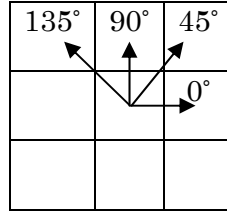


Fig. 1.5 : Directions des matrices de cooccurrence

La figure 1.6 illustre le principe de calcul d'une matrice de cooccurrence sur une région de 25 pixels quantifiée sur 4 niveaux de gris.

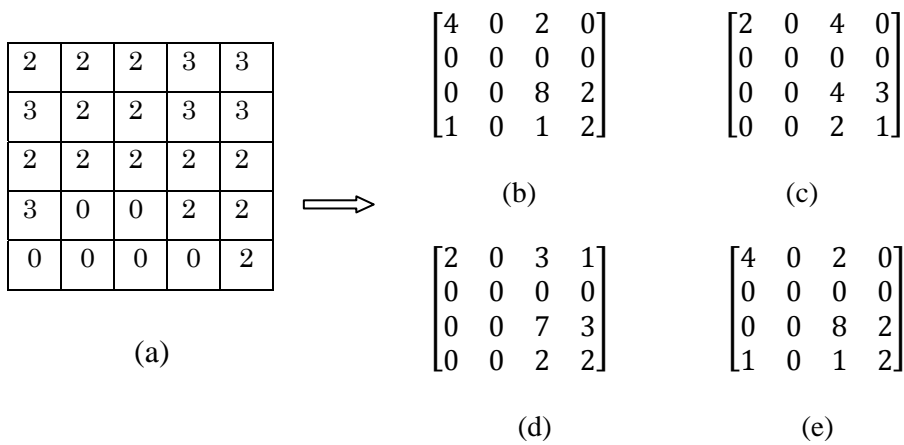


Fig. 1.6 : Exemple de calcul d'une matrice de cooccurrence.

(a) Image originale, (b) Matrice $t(0^\circ, 1)$, (c) Matrice $t(45^\circ, 1)$, (d) Matrice $t(90^\circ, 1)$, (e) Matrice $t(135^\circ, 1)$.

Les matrices obtenues ne sont pas symétriques, le calcul des attributs de texture est beaucoup plus performant dans une matrice symétrique.

$$t_s(0^\circ, 1) = \frac{1}{2}(t(0^\circ, 1) + t^T(0^\circ, 1)) \quad (1.20)$$

$$t_s(0^\circ, 1) = \begin{bmatrix} 4 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 8 & 1.5 \\ 0.5 & 0 & 1.5 & 2 \end{bmatrix} \quad (1.21)$$

La matrice normalisée est obtenue en divisons $t_s(0^\circ, 1)$ par $N=20$:

$$T_s(0^\circ, 1) = \begin{bmatrix} 0.2 & 0 & 0.05 & 0.025 \\ 0 & 0 & 0 & 0 \\ 0.05 & 0 & 0.4 & 0.075 \\ 0.025 & 0 & 0.075 & 0.1 \end{bmatrix} \quad (1.22)$$

La matrice de cooccurrence est riche en information mais reste difficilement

exploitable dans son intégralité. Une solution consiste à réduire le nombre de niveaux de gris afin d'utiliser les éléments de la matrice de cooccurrence comme attributs de texture [50]. L'autre démarche qui est la plus courante consiste à extraire plusieurs attributs afin de résumer toute l'information apportée par la matrice de cooccurrence [49]. Les six attributs du tableau 3 sont ceux qui sont les plus utilisés.

<i>Contraste</i>	$CON = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 T^2(i, j, d, \theta)$
<i>Variance</i>	$VAR = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 T(i, j, d, \theta)$
<i>Second moment angulaire</i>	$SMA = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T^2(i, j, d, \theta)$
<i>Entropie</i>	$ENT = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} T(i, j, d, \theta) \log T(i, j, d, \theta)$
<i>Corrélation</i>	$COR = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{(i - \mu_x)(j - \mu_y) T(i, j, d, \theta)}{\sigma_x \sigma_y}$
<i>Homogénéité locale ou moment angulaire inverse</i>	$IDM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1 + (i-j)^2} T(i, j, d, \theta)$

Tableau 3 : Attributs de texture extraits par les matrices de cooccurrence.

avec

$$\sigma_x^2 = \sum_{i=0}^{G-1} (i - \mu_x)^2 T_x(i) \quad \text{et} \quad \sigma_y^2 = \sum_{j=0}^{G-1} (j - \mu_y)^2 T_y(j) \quad (1.23)$$

tels que

$$T_x = \sum_{i=0}^{G-1} T(i, j), \quad T_y = \sum_{j=0}^{G-1} T(i, j), \quad \mu_x = \sum_{i=0}^{G-1} i T_x(i) \quad \text{et} \quad \mu_y = \sum_{j=0}^{G-1} j T_y(j) \quad (1.24)$$

Dans le premier indice, chaque terme de la matrice T est pondéré par sa distance à la diagonale. On obtient ainsi un indice correspondant à la notion usuelle du contraste, il est élevé quand les termes éloignés de la diagonale de la matrice sont élevés, c'est-à-dire quand on passe souvent d'un pixel très clair à un pixel très foncé ou inversement. Le SMA est d'autant plus élevé que l'on retrouve souvent le même couple de pixels, ce qui est le cas quand le niveau de gris est uniforme (texture lisse) ou quand il y a périodicité dans le cas de translation. L'entropie est faible si on a souvent le même nombre de

pixels, forte si chaque couple est peu représenté. Elle fournit un indicateur de désordre que peut présenter une texture. La corrélation quantifie la dépendance directionnelle des niveaux de gris, elle atteint ses plus grandes valeurs lorsque θ est proche de l'orientation des lignes de la texture. IDM reflète l'existence de plages uniformes de texture et affecte un poids de plus en plus faible au fur et à mesure qu'on s'éloigne de la diagonale principale. Plus la valeur de l'IDM est élevée plus la texture est grossière. La variance mesure la somme des fréquences pour différents écarts de niveaux de gris.

1.3.3.3 Méthode des matrices de cooccurrence généralisées

La matrice de cooccurrence a été généralisée par Davis et al. [35]. Elle consiste à calculer la matrice de cooccurrence non pas de l'image texturée originale, mais d'une image reflétant certaines propriétés locales de l'image texturée telle que l'amplitude et l'orientation du gradient dans le cas des images contours. Cette dernière est habituellement obtenue en faisant un produit de convolution de l'image avec un ensemble de masques. Une matrice de cooccurrence généralisée peut être définie de la manière suivante:

Soit $P = \{p_i = (x_i, y_i, d_i)\}_{i=1}^n$ où (x_i, y_i) est la position de pixel i , d_i est la description du $i^{\text{ème}}$ pixel. Soit F un prédicat de contrainte spatial. Par exemple, $F(p_i, p_j, k) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < k$ est vrai si la distance entre p_i et p_j est inférieure à k . Ainsi, la matrice de cooccurrence généralisée (GCM), notée $G_F(d, d')$, est le nombre de paires (p_k, p_l) tels que d et d' sont respectivement les descripteurs des pixels p_k et p_l .

Nous pouvons utiliser d'autres prédicats de contraintes spatiales, telle que:

$F_2(p_i, p_j)$ est vrai si p_j est le plus proche voisin de p_i ou combiner plusieurs prédicats:

$$F_1(p_i, p_j, k) \wedge F_2(p_i, p_j), \text{ et } F_2(p_i, p_j) \wedge F_2(p_j, p_i).$$

Représentant les orientations des maximum locaux obtenus par l'application des opérateurs de gradient (H-horizontal, V-vertical, L- diagonale gauche, R- diagonale droite, blanc- aucun maximum local) (Fig. 1.7(a)).

La figure 1.7(b) montre une matrice de cooccurrence généralisée de l'image, le prédicat utilisé est $F = F_1(p_i, p_j, 2)$.

						V	
	H	H	H			V	
	V		V				L
	H	H	H				
					R		
	R			R		L	
R		V		V		R	
H	H			H	H		

	H	V	L	R
H	22	11	0	5
V	11	6	1	4
L	0	1	0	3
R	5	4	3	4

(a)
(b)

Fig.1.7 : Exemple de calcul de la matrice de cooccurrence généralisée.
 (a) Image codée, (b) GCM en utilisant $F = F_1(p_i, p_j, 2)$

Une autre manière de définir une matrice de cooccurrence généralisée (Generalized Co-occurrence Matrix) est proposée dans [36]. Celle-ci décrit la distribution statistique entre le niveau de gris de l'image originale et celui de l'image lissée.

Soient $I(x, y)$ la fonction image de taille $M \times N$, $f(x, y)$ est une image lissée, obtenue par un lissage de l'image $I(x, y)$ en utilisant un ensemble de matrices, d'ordre (2×2) . Ainsi, le terme de la matrice de cooccurrence généralisée est donné par:

$$H(i, j) = \#\{(x, y) \mid I(x, y) = i \ \& \ f(x, y) = j\} \quad (1.25)$$

1.3.3.4 Méthode des histogrammes somme et différence des niveaux de gris

L'idée de cette méthode repose sur l'étude statistique basée sur les histogrammes somme et différence (SDH) des niveaux de gris entre paires de pixels [51].

Soient deux pixels de position relative fixée par d_x , et d_y . La probabilité d'observer les niveaux de gris i et j à une position fixée par (d_x, d_y) est donnée par :

$$P[g(x, y) = i, g(x + d_x, y + d_y) = j] = P(i, j, d_x, d_y) = P(i, j) \quad (1.26)$$

Cette probabilité ne dépend pas de la position (x, y) du pixel considéré, on peut alors définir la somme et la différence des niveaux de gris de pixels séparés par la position (d_x, d_y) ainsi que leurs histogrammes associés de la manière suivante:

$$\begin{aligned} \text{Somme : } S(x, y) &= g(x, y) + g(x + d_x, y + d_y) \\ \text{Différence : } D(x, y) &= g(x, y) - g(x + d_x, y + d_y) \end{aligned} \quad (1.27)$$

$$\text{Histogramme de la somme : } h_s(i, d_x, d_y) = h_s(i) = \text{Card}\{(x, y) / S(x, y) = i\}$$

$$\text{Histogramme de la différence : } h_D(j, d_x, d_y) = h_D(j) = \text{Card}\{(x, y) / D(x, y) = j\}$$

Les histogrammes sommes et différences normalisées sont données par les équations suivantes :

$$P_S(i) = P(S(x, y) = i) = \frac{h_S(i)}{N} \quad (i = 0, \dots, 2G - 2) \quad (1.28)$$

$$P_D(j) = P(D(x, y) = j) = \frac{h_D(j)}{N} \quad (j = -G + 1, \dots, G - 1)$$

avec :

$$N = \sum_{i=0}^{2G-2} h_S(i) = \sum_{j=-G+1}^{G-1} h_D(j) \quad (1.29)$$

Les attributs calculés à partir de ces histogrammes sont consignés dans le tableau 6.

<i>Contraste</i>	$Con = \sum_{j=-G+1}^{2G} j^2 P_D(i)$
<i>Second Moment Angulaire</i>	$ASM = \sum_{i=0}^{G-1} P_S(i) \sum_{j=-G+1}^{G-1} P_D(j)$
<i>Entropie</i>	$Ent = - \sum_{i=0}^{2G} P_S(i) \log[P_S(i)] - \sum_{j=-G+1}^{G-1} P_D(i) \log [P_D(j)]$
<i>Variance</i>	$Var = \sum_{i=0}^{2G} (i - 2\mu)^2 P_S(i) \quad \mu = \frac{1}{2} \sum_{i=0}^{2G} iP_S(i)$
<i>Corrélation</i>	$Cor = \frac{1}{2} \sum_{i=0}^{2G} (i - 2\mu)^2 P_S(i) - \sum_{j=-G+1}^{G-1} j^2 P_D(j)$
<i>Moment Inverse</i>	$IDM = \sum_{j=-G+1}^{G-1} j^2 P_D(j)$
<i>Cluster Shade</i>	$CS = \sum_{i=0}^{2G} (i - 2\mu)^3 P_S(i)$
<i>Cluster Prominence</i>	$CP = \sum_{i=0}^{2G} (i - 2\mu)^4 P_S(i)$

Tableau 6: Attributs extraits à partir de SDH

Les attributs issus de la méthode SDH sont, en général, d'une performance proche de ceux évalués à partir des matrices de cooccurrences avec une réduction considérable de l'espace mémoire et du temps de calcul [52]. Cependant, ces attributs n'ont pas tous la même signification que ceux extraits par les matrices de cooccurrence [53].

1.3.3.5 Méthode LBP

Cette méthode initialement proposée par Ojala et al. [54], considère un nombre de pixels voisins p et une distance d reliant le pixel central avec les pixels voisins, centré sur chaque pixel de l'image. Pour chaque pixel, on définit une matrice de taille $(p \times p)$. En codant ces matrices, on obtient un nombre appelé Local Binary Patterns ou (LBP) qui peut prendre 2^p valeurs et il est défini par l'équation suivante [18]:

$$\text{LBP}(p, d) = \sum_{i=0}^{p-1} u(g_i - g_c) 2^i \quad (1.30)$$

où g_c est le niveau de gris du pixel central, g_i ($i = 0, 1, \dots, p - 1$) est le niveau de gris du pixel voisin et $u(x)$ est la fonction définie comme suit :

$$u(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (1.31)$$

Ce nombre est généralement combiné avec une mesure de contraste C qui évalue la différence des niveaux de gris des pixels voisins ayant la valeur binaire 1 et ceux ayant la valeur 0. La distribution bidimensionnelle reliant les valeurs de LBP et celles de C permet de caractériser la texture. La figure (1.8) illustre le principe de calcul du nombre LBP et du contraste C d'un pixel ayant $p=8$ pixels voisins.

6	5	2	1	0	0	1	2	4	1	0	0
7	6	1	1		0	8		16	8		0
9	3	7	1	0	1	32	64	128	32	0	128
(a)			(b)			(c)			(d)		

Fig. 1.8 : Calcul du nombre LBP et du contraste C .

Dans cet exemple, nous avons : $\text{LBP} = 1 + 8 + 32 + 128 = 169$ et $C = (6 + 7 + 9 + 7)4 - (5 + 2 + 1 + 3)4 = 4.5$.

1.3.3.6 Fonction d'Auto-corrélation

La fonction d'auto-corrélation mesure la ressemblance entre une image $I(x, y)$ définie sur une fenêtre rectangulaire de taille $N \times M$ et sa translatée $I(x + dx, y + dy)$. A partir de cette mesure, on peut estimer la taille des éléments d'une texture présente dans une image [55]. La fonction d'auto-corrélation de l'image est définie de la manière suivante:

$$C(d_x, d_y, x, y) = \frac{1}{(N - |dx|)(M - |dy|)} \frac{\sum_{x=1}^N \sum_{y=1}^M I(x, y) I(x + d_x, y + d_y)}{\sum_{x=1}^N \sum_{y=1}^M I^2(x, y)} \quad (1.32)$$

avec $|dx| < N$ et $|dy| < M$.

Une texture fine a une fonction d'auto-corrélation qui décroît rapidement comparativement à celle d'une texture grossière. Si les primitives qui constituent la texture sont grandes, alors les petites translations ne produisent pas une grande différence dans la comparaison. Par conséquent, la fonction d'auto-corrélation va montrer une décroissance lente. Une répétitivité des primitives entraîne également une périodicité de la fonction d'auto-corrélation.

Les attributs de la texture de chaque pixel sont déterminés de la manière suivante :

$$T(x, y) = \sum_{d_x=-\alpha}^{\alpha} \sum_{d_y=-\alpha}^{\alpha} d_x^2 d_y^2 C(d_x, d_y, x, y) \quad (1.33)$$

α étant un entier positif.

Plus $T(x, y)$ augmente, plus la texture est grossière. L'inconvénient majeur de cette méthode est que la fonction d'auto-corrélation est presque la même pour des textures naturelles.

1.3.4 Méthodes basées sur un modèle

Le but de ces méthodes est d'obtenir un modèle générateur de la texture. Les paramètres de ce modèle permettent de caractériser ou de synthétiser une texture. On distingue en général trois types de modèles. Les modèles linéaires, particulièrement, les modèles autorégressifs, les modèles fractals et les modèles Markoviens.

a) Modélisation linéaire de la texture

Dans ce type de modélisation, la valeur du niveau de gris de chaque pixel est une combinaison linéaire de celle des pixels voisins [48].

Considérant une image $I = \{I(i, j) / 0 \leq i, j \leq M - 1\}$ modélisée par :

$$I(i, j) = \sum_{(k, l) \in R} \alpha(k, l) I(i - k, j - l) + \sum_{(k', l') \in R} \beta(k', l') e(i - k', j - l') \quad (1.34)$$

$e(i, j)$ est un élément d'un champ aléatoire, R est la région de prédiction. M représente les tailles horizontale et verticale de l'image.

Parmi les modélisations linéaires de la texture, on trouve le modèle autorégressif où

$$\begin{cases} \beta(0, 0) = 1 \\ \beta(k', l') = 0 \end{cases} \quad \forall (k', l') \neq (0, 0) \quad (1.35)$$

C'est-à-dire

$$I(i, j) = \sum_{(k, l) \in R} \alpha(k, l) I(i - k, j - l) + e(i, j) = \hat{I}(i, j) + e(i, j) \quad (1.36)$$

$\hat{I}(i, j)$ correspond à l'estimer de $I(i, j)$ et e représente l'erreur de prédiction.

Les coefficients $\alpha(k, l)$, considérés comme attributs de texture, peuvent être obtenus en minimisant l'espérance de carré de l'erreur entre l'estimation et la valeur réelle du niveau de gris en scrutant point par point la texture et en modifiant les valeurs des coefficients en fonction de l'écart entre $I(i, j)$ et $\hat{I}(i, j)$.

L'ensemble d'équations linéaire à résoudre est de la forme :

$$\gamma(m, n) = \sum_{(k, l) \in R} \alpha(k, l) \gamma(m - k, n - l) \quad \text{pour } (0 \leq i, j \leq M - 1) \quad (1.37)$$

$$\gamma(m, n) \equiv E[I(i, j) \cdot I(i - m, j - n)] \quad (1.38)$$

La résolution de ce système permet alors d'obtenir les $\alpha(k, l)$.

b) Champs de Markov

Les champs de Markov sont largement utilisés en vision par ordinateur, notamment dans le domaine de l'analyse et la synthèse des textures. Ils permettent de modéliser des propriétés globales en utilisant des contraintes locales. Les modèles les plus utilisés sont le champ de Markov Gaussien ou GMRF (Gaussian Markov Random Field) [56] et le modèle binomial [57]. Les paramètres de ces modèles sont considérés comme attributs de texture. Dans le cas du GMRF classique, le niveau de gris d'un pixel est égale à la somme pondérée par les paramètres du modèle, des niveaux de gris des voisins, à laquelle on ajoute un bruit Gaussien de moyenne nulle. La variance de ce bruit constitue également un des paramètres du modèle. Dans le deuxième modèle, la

probabilité de chaque pixel de la texture conditionnellement à ses voisins est binomiale dépendant du voisinage et du nombre de niveaux de gris de l'image.

Considérons un champ aléatoire X constitué de variables aléatoires X_s en chaque site $s, s \in S$ (ici s est le pixel et S l'image). N_s désigne le voisinage de s tel que :

$$N_s \subset S \text{ et } (s, r) \subset S: r \in N_s \Rightarrow s \in N_r \quad (1.39)$$

On notera par la lettre P les diverses lois de probabilité liées à X . On dit qu'un champ X est un champ de Markov relativement à N_s , si et seulement si les deux propriétés suivantes sont vérifiées:

- Positivité

$$\forall x \in \Omega^{\text{card}(S)}, P(x) > 0 \quad (1.40)$$

- Markovianité

$$P(X_s = x_s / X_r = x_r, r \in S) = P(X_s = x_s / X_r = x_r, r \in N_s) \quad (1.41)$$

où les lettres majuscules, $X_s \in X$ désignent les variables et les lettres minuscules, $x_s \in \Omega$, désignent les valeurs des réalisations respectives. Ω est l'ensemble des réalisations possibles du champ X .

Cette propriété signifie que la probabilité $P(./.)$ en un site s conditionnellement au reste du champ est égale à la probabilité en ce site connaissant uniquement ces voisins.

L'utilisation des champs de Markov en traitement d'images repose principalement sur le théorème d'Hammerslay-Clifford et l'utilisation de la mesure de Gibbs [2].

Champs de Gibbs :

Soit S un ensemble de pixels muni d'un système de voisinage N , et soit C l'ensemble des cliques c associées à N . Les interactions locales entre les niveaux de gris des sites voisins s'expriment alors comme un potentiel de clique. Pour $c \in C$, un potentiel V_c est associé à c , dont la valeur dépend des niveaux de gris des pixels constituant la clique. De même, L'énergie $U(X = x)$ s'exprime généralement sous forme d'une somme de fonctions V_c qui est défini comme la somme des potentiels de toutes les cliques présentées dans le système de voisinage considéré :

$$U(X = x) = \sum_{c \in \mathcal{C}} V_c(x) \quad (1.42)$$

Le champ X est un champ de Gibbs (CG), si et seulement si sa loi est définie par:

$$P(X = x) = \frac{1}{Z} \exp[-U(X = x)] \quad (1.43)$$

avec :

$$Z = \sum_{x \in \Omega} \exp[-U(X = x)] \quad (1.44)$$

Z est une constante de normalisation, également appelée « fonction de partition » qui fait de $P(x)$ une probabilité. Etant donné que cette constante est une somme sur l'ensemble de toutes les configurations possibles du champ X , son calcul est dans la pratique généralement impossible. Par exemple, pour une image binaire ($\text{Card}(\Omega) = 2$) de taille 256×256 pixels, le calcul de Z nécessite une somme sur $2^{256 \times 256}$ configurations possibles, ce qui est impossible.

L'énergie totale d'un champ de Gibbs se décompose donc sous forme d'une somme d'énergies locales. Le lien entre champ de Gibbs et processus Markovien repose principalement sur le théorème d'Hammersley-Clifford. Ce théorème établit l'équivalence entre champ de Markov et champ de Gibbs, en effet si X est un champ de Markov, sa distribution de probabilité est une distribution de Gibbs. Ainsi, la probabilité conditionnelle $P(X_s = x_s / X_r = x_r, r \in N_s)$ s'écrit :

$$P(X_s = x_s / X_r = x_r, r \in N_s) = \frac{P(X = x)}{P(X_r = x_r, r \in N_s)} = \frac{e^{-U(x_s, x_r, r \in N_s)}}{\sum e^{-U(x_s, x_r, r \in N_s)}} \quad (1.45)$$

On peut définir pour N_s toute forme de système de voisinage. Le voisinage le plus souvent utilisé pour les champs de Markov en traitement d'images est le voisinage en 4 ou en 8 connexités. A partir de ce voisinage, on peut définir des sous ensembles \mathcal{C} de voisins connexes appelés cliques. L'ordre de la clique représente le nombre de voisins contenu dans le sous ensemble, le pixel s fait partie du sous ensemble (Fig. 1.9).

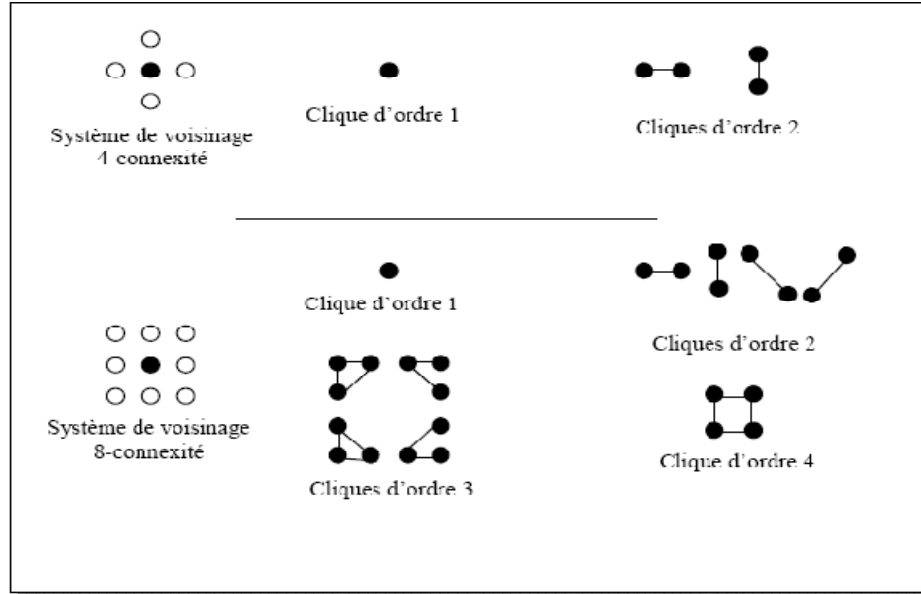


Fig. 1.9: Voisinage bidimensionnel d'ordre 1 et 2 associée à la distance euclidienne et les cliques associées.

Pour une image composée d'une seule texture homogène, on a :

$$U(X = x) = \frac{1}{2} \sum_{s=1}^N V(x_s, N_s)^T \theta \quad (1.46)$$

avec $V(x_s, N_s)$ est le potentiel d'interaction du site s avec ses sites voisins définis par N_s .

Dans le cas des quatre plus proches voisins, on a :

$$V(x_s, N_s) = [v_1(x_s), v_2(x_s), v_3(x_s), v_4(x_s)]^T \quad (1.47)$$

et

$$\theta = [\theta_1, \theta_2, \theta_3, \theta_4]^T \quad (1.48)$$

Les éléments du vecteur θ correspondent aux attributs de texture, ils peuvent être estimés en utilisant la méthode des moindres carrés.

Modèle Markovien binomial

On emploie la formulation bien connue de la loi binomiale, avec la probabilité P qui dépend du voisinage, et un nombre de tirages égale à $G-1$ (où G est le nombre total des niveaux de gris). Ainsi on peut écrire que la probabilité conditionnelle d'avoir un niveau de gris x_s pour le site s par :

$$P(X_s = x_s / X_r, \forall r \in N_s) = \frac{(G-1)!}{x_s! (G-1-x_s)!} \theta_s^{x_s} (1-\theta_s)^{G-1-x_s} \quad (1.49)$$

où $\frac{(G-1)!}{g!(G-1-g)!}$ correspond au nombre de fois de choisir x_s parmi $G-1$ niveaux de gris, $\theta_s^{x_s}$ la probabilité d'obtenir x_s succès et $(1 - \theta_s)^{G-1-x_s}$ la probabilité d'obtenir $G - 1 - x_s$ échecs.

θ_s est une fonction du voisinage N_s telle que:

$$\theta_s = \frac{\exp(\Psi)}{1 + \exp(\Psi)} \quad (1.50)$$

Ψ est une fonction (linéaire ou non) calculée sur les niveaux de gris des pixels appartenant au voisinage du site considéré et qui prend des expressions différentes selon l'ordre du modèle.

$$\Psi = \frac{1}{T} (\alpha_s + \sum_{r \in N_s} \beta_{sr} x_r) \quad (1.51)$$

Ainsi pour un modèle d'ordre 1, on pose :

$$\Psi = a + b_1(x_{i,j-1} + x_{i,j+1}) + b_2(x_{i-1,j} + x_{i+1,j}) + b_3(x_{i-1,j-1} + x_{i+1,j+1}) + b_4(x_{i+1,j-1} + x_{i-1,j+1}) \quad (1.52)$$

Une texture est donc caractérisée par le vecteur de paramètres $\beta = (a, b_1, b_2, b_3, b_4)^T$. Les coefficients a, b_1, b_2, b_3, b_4 de β peuvent être assimilés à des attributs texturaux. Il est possible de leur associer une signification perceptuelle. Le paramètre a est lié au niveau de gris moyen de la texture, les coefficients b_1, b_2, b_3, b_4 décrivent respectivement l'homogénéité horizontale (b_1), verticale (b_2) et diagonale (b_3, b_4) de la texture.

La notation matricielle donne :

$$\Psi = Q_s^T \beta \quad (1.53)$$

avec

$$Q_s = \begin{bmatrix} x_{i,j-1} + x_{i,j+1} \\ x_{i-1,j} + x_{i+1,j} \\ x_{i-1,j-1} + x_{i+1,j+1} \\ x_{i+1,j-1} + x_{i-1,j+1} \end{bmatrix} \quad (1.54)$$

Leurs estimations par les moindres carrés donnent :

$$\beta = \left[\sum_{s=1}^N Q_s Q_s^T \right]^{-1} \left[\sum_{s=1}^N Q_s \Psi \right] \quad (1.55)$$

1.3.5 Morphologie mathématique

La morphologie mathématique est une théorie d'analyse d'images élaborée par G. Matheron et de J. Serra en 1964. Elle est fondée sur les concepts de géométrie, topologie, théorie des probabilités et surtout sur la théorie des ensembles. La morphologie mathématique constitue une approche d'analyse d'images à part entière et peut être utilisée pour résoudre un grand nombre de problèmes de traitement d'images tels que le filtrage, l'analyse de textures et la segmentation d'images [8]:

L'idée de base de la morphologie mathématique est de comparer les objets présents dans l'image à analyser à l'aide d'un autre objet de forme connue appelé élément structurant [58]. Ce dernier est une forme géométrique qui peut être un carré, un cercle, une croix, ou un losange... . L'élément structurant est ainsi déplacé de façon à ce que son centre O passe par toutes les positions dans l'image à analyser. Pour chacune des positions de O , on se pose une question relative à l'union ou à l'intersection de l'élément structurant avec les objets de l'image. L'ensemble des points correspondant à une réponse positive permet de construire une nouvelle image à partir de laquelle, on peut extraire une information.

1.3.5.1 Transformations morphologiques binaires de base

Une image binaire peut être considérée comme une partie d'un espace $D = R^d$ dont l'origine est notée O .

Erosion binaire

L'érosion d'un ensemble A par un élément structurant E est définie à partir de la soustraction ensembliste de Minkowski, notée \ominus , sous la forme :

$$\varepsilon_E(A) = A \ominus E = \{s/N_s \subseteq A\} \quad (1.56)$$

où N_s désigne la translation de l'élément structurant E centré au point $s / s \in A$.

Dilatation binaire

La dilatation d'un ensemble A par un élément structurant E est définie à partir de l'addition de Minkowski, notée \oplus , telle que :

$$\delta_E(A) = A \oplus E = \{s/N_s \cap A \neq \emptyset\} \quad (1.57)$$

La dilatation s'exprime donc comme l'union des éléments structurants dont l'origine est déplacée à l'intérieur de A .

En pratique, l'érosion et la dilatation binaires sont rarement utilisées seules. Leurs combinaisons donnent naissance à deux autres opérations morphologiques que sont l'ouverture, la fermeture et le gradient morphologique.

Ouverture et fermeture binaires

L'ouverture d'un ensemble A par E , notée A_E est l'ensemble résultant d'une érosion de A par E , suivie d'une dilatation de l'ensemble érodé par le même élément structurant tel que :

$$A_E = (A \ominus E) \oplus E \quad (1.58)$$

L'opération d'ouverture a pour effet de supprimer les petits détails se trouvant à la périphérie des sous ensembles convexes.

D'une manière duale à l'ouverture, la fermeture, notée A^E , est le résultat d'une dilatation suivie d'une érosion en utilisant le même élément structurant :

$$A^E = (A \oplus E) \ominus E \quad (1.59)$$

La fermeture morphologique agit de manière duale à l'ouverture, c'est à dire qu'elle bouche les parties du fond de l'image ne contenant pas l'élément structurant.

Gradient morphologique binaires

Le gradient morphologique est défini à partir de l'érosion et de la dilatation. On considère :

- Le gradient morphologique ou gradient de Beucher :

$$\rho_E(A) = \delta_E(A) - \varepsilon_E(A) \quad (1.60)$$

- Le gradient morphologique interne :

$$\rho_{E^-}(A) = A - \varepsilon_E(A) \quad (1.61)$$

- Le gradient morphologique externe :

$$\rho_{E^+}(A) = \delta_E(A) - A \quad (1.62)$$

1.3.5.2 Transformations morphologiques des images en niveaux de gris

La morphologie mathématique en niveaux de gris peut être vue comme l'extension de la morphologie binaire (dans \mathbb{Z}^2) à l'espace \mathbb{Z}^3 à trois dimensions et ceci, en passant par l'intermédiaire de la notion d'ombre d'une image. En effet, une image en niveaux de gris, est caractérisée par une fonction I définie sur le support $D \subset \mathbb{Z}^2$ à valeur dans l'intervalle discret $[0, \dots, G-1]$ représentant l'ensemble des niveaux de gris possibles. La fonction I définit ainsi une surface de l'espace \mathbb{Z}^3 ; l'ensemble de tous les points situés sous cette surface correspond à l'ombre de l'image I (Fig. 1.10). On note cet ensemble $U(I)$ tel que :

$$U(I) = \{(s, m) \in \mathbb{Z}^3 / s \in D \text{ et } m \leq I(s)\} \quad (1.63)$$

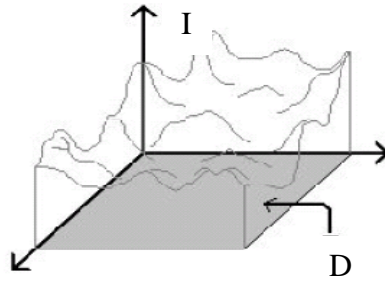


Fig. 1.10: Ombre d'une image à niveau de gris I de support D .

On considère des éléments structurants fonctionnels : c'est-à-dire qu'un élément structurant est défini, comme étant une petite image, par un support $E \subset \mathbb{Z}^2$ et une fonction de gris h_E définie sur E , on le note (E, h_E) (Fig.1.11). C'est un ensemble de \mathbb{Z}^3 qui en contient toujours l'origine.

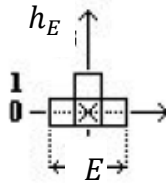


Fig. 1.11 : Exemple d'élément structurant fonctionnel (E, h_E)

Erosion

L'érosion de l'image I par l'élément structurant (E, h_E) , notée $\varepsilon_{(E, h_E)}(I)$, est identique à l'érosion, $\varepsilon_{(E, h_E)}(U(I))$, de l'ombre d'une fonction I , $U(I)$, par un élément structurant (E, h_E) .

$$\varepsilon_{(E, h_E)}(I)(s) = \varepsilon_{(E, h_E)}(U(I)) = U(I) \ominus \check{E} = \inf \{I(y) - h_E(y) / y \in E(s)\} \quad (1.64)$$

où \check{E} est le transposé de l'élément structurant E , il est défini par la symétrie centrale.

Si l'élément structurant est non fonctionnel ($h_E = 0$), on écrira :

$$\varepsilon^E I(s) = \inf\{I(u); u \in N_s\} = I(s) \ominus \check{E} \quad (1.65)$$

Dilatation

De la même façon que pour l'érosion, on peut définir la dilatation d'une image I par l'élément structurant (E, h_E) comme la dilatation de son ombre $U(f)$:

$$\delta_{(E, h_E)}(I)(s) = \delta_{(E, h_E)}(U(I)) = U(I) \oplus \check{E} = \sup \{f(y) + h_E(y) / y \in E(s)\} \quad (1.66)$$

Si l'élément structurant est non fonctionnel, on a :

$$\delta^B I(s) = \text{Sup}\{f(u); u \in N_s\} = (f \oplus E)(s) \quad (1.67)$$

Gradient

Un filtre Gradient permet de mettre en évidence les variations de niveaux de gris suivant un axe variable, ce qui aura pour effet de mettre en évidence les fronts et de révéler les textures. Pour une direction donnée, un filtre Gradient est utilisé pour augmenter ou bien réduire les fronts sur cette direction. Il est utilisé en particulier pour mettre en évidence des contours.

$$g = (I \oplus E) - (I \ominus E) \quad (1.68)$$

Ouverture et fermeture des images au niveau de gris

L'ouverture et la fermeture des images sont, comme dans le cas de la morphologie binaire, déduites des transformations élémentaires.

$$\text{- l'ouverture : } O^E I(A) = \delta^E(\varepsilon^E I(A)) \quad (1.69)$$

$$\text{- la fermeture : } F^E I(A) = \varepsilon^E(\delta^E I(A)) \quad (1.70)$$

1.3.5.3 Analyse de la texture par morphologie mathématique

L'utilisation de la morphologie mathématique en analyse de la texture d'images est basée en générale sur une série d'ouvertures par un élément structurant de taille croissante [30] [31] [32] [59].

Granulométrie et distribution de taille

La granulométrie permet d'analyser des objets ou de filtrer des images suivant des critères de forme. L'analyse granulométrie ou distribution de taille, introduite par Matheron, consiste à itérer une ouverture (ou une fermeture) à l'aide des éléments structurants de tailles croissantes, appliquée aux images binaires. Son application pour les images de niveaux de gris est faite par Serra. Il s'agit plus exactement d'appliquer une suite d'ouvertures morphologiques. D.Wang [59] construit un ensemble de n éléments structurants $(E_0, E_1, \dots, E_{n-1})$ définis sur \mathbb{Z}^2 , construit à partir d'un élément E comme suit (Fig. 1.12) :

- $E_0 = \{o\}$ (E_0 se restreint à l'origine \mathbb{Z}^2).
- $E_{i+1} = E_i \oplus E$ (E_{i+1} est le dilaté de E_i par E).

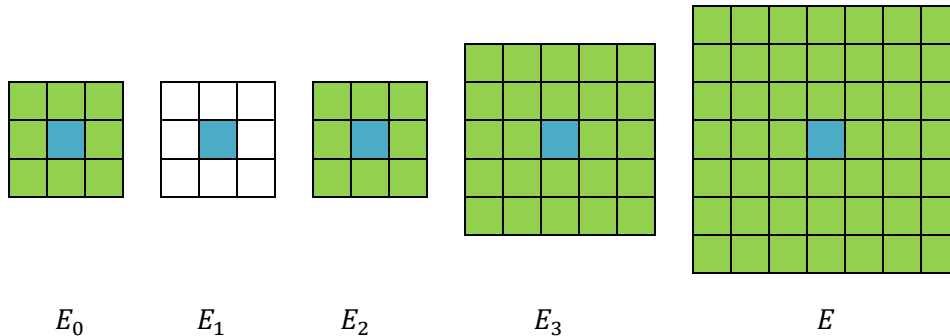


Fig.1.12: Exemple des quatre premiers éléments structurants obtenus à partir de l'élément E des huit voisins.

Il décompose ensuite une image I en un ensemble d'images composantes I_j ($j = 0, 1, \dots, n$) contenant chacune des éléments textuels de même taille selon l'algorithme suivant :

Algorithme
 $F_0 = I$;
 Pour $j=0$ à n faire
 {
 $I_j = O_{E_{n-j}}(F_j)$;
 $F_{j+1} = F_j - I_j$;
 }

Algorithme 1: Décomposition morphologique multi-taille

L'algorithme commence par extraire les primitives de grandes tailles dans I_0 en utilisant l'élément structurant le plus grand (E_n). Ensuite, on réitère l'extraction sur l'image source privée de ses plus grands éléments à l'aide de E_{n-1} . Et, de la même façon, les primitives de moins en moins larges sont extraites en utilisant des éléments structurants de plus en plus petits.

Une fois cette décomposition effectuée, un ensemble de paramètres de chaque image composante est calculé (moyenne des niveaux de gris m_i). Il en résulte ainsi, une texture caractérisée par un vecteur à $n + 1$ composantes (m_0, \dots, m_n) .

Chaque image composante I_j conserve les primitives de texture d'une certaine taille correspondant à la taille de l'élément structurant E_{n-j} . La somme des images composantes redonne l'image originale :

$$\forall x, \sum_{i=0}^n I_i(x) = I(x) \quad (1.71)$$

Cette méthode d'analyse de textures peut être considérée comme appartenant à l'approche structurelle car elle permet d'extraire les primitives composant la structure des textures.

1.4 Classification des images

La classification permet de partitionner un ensemble d'observations comme les images en différentes classes, en regroupant dans une même classe les observations (images) présentant les mêmes caractéristiques. La classification des images intervient comme une étape très importante dans des applications de reconnaissance des formes telles que la reconnaissance des visages, des empreintes, etc.... ainsi qu'en indexation d'images (rechercher une image dans une base de données). Dans ce type d'application chaque image peut être caractérisée par un ensemble d'attributs qui correspondent généralement aux attributs de texture. Les méthodes de classification se distinguent par leur caractère supervisé et non supervisé. Lorsque les différentes classes des observations sont connues au préalable, la classification est dite supervisée. En revanche, dans le cas où l'on n'a aucune connaissance *a priori* sur l'appartenance des observations aux classes, on parle de classification non-supervisée ou *clustering*, celle-ci est plus utilisée pour la segmentation des images. Dans un contexte supervisé, on dispose d'échantillons (ensemble d'observations) ou chaque observation est caractérisée par un ensemble d'attributs. L'appartenance de chaque observation de cet échantillon à une classe est connue a priori. Dans ce cas ces observations sont appelées prototypes de la classe. La classification a pour but de définir à partir des prototypes des modèles mathématiques pour chaque classe. Ces modèles peuvent être un simple centre de gravité, une fonction de densité de probabilité ou une surface de séparation entre des classes. Les étapes de caractérisation et de classification des observations prototypes des

classes constituent ce qu'on appelle la phase d'apprentissage. Celle-ci se distingue de la phase de décision qui consiste à affecter une nouvelle observation à l'une des classes définie lors de la phase d'apprentissage. Il existe une multitude de méthodes de classification supervisée. Les plus classiques sont la classification optimale de Bayes et la règle des K plus proches voisins.

1.4.1 Classification par la méthode bayésienne

Cette méthode de nature statistique, suppose que l'échantillon à analyser est composé de N observations tirées, indépendamment, de K classes C_k où $k = 1, 2, \dots, K$ selon les lois de probabilité conditionnelles $P(X/C_k)$ connues. Par application du théorème de Bayes, il est possible de déterminer la probabilité pour qu'une observation X_q appartienne à la classe C_k . Cette probabilité, appelée aussi probabilité *a posteriori* de la classe C_k , est donnée par la relation [64] :

$$P(C_k/X_q) = \frac{P(X_q/C_k)P(C_k)}{P(X_q)} \quad (1.72)$$

où $P(X_q)$ désigne la valeur de la fonction de densité sous-jacente à la distribution des observations provenant de l'échantillon au point X_q , et $P(C_k)$ désigne la probabilité *a priori* de la classe C_k de telle sorte que :

$$\sum_{k=1}^K P(C_k) = 1 \quad (1.73)$$

$P(X_q/C_k)$ désigne la probabilité conditionnelle de la classe C_k , c'est-à-dire, la fonction de densité de probabilité sous-jacente à la distribution des observations provenant de la classe C_k . Dans ces conditions, le risque d'erreur lié à l'affectation de l'observation X_q à la classe C_k est donné par [65]:

$$R(C_k/X_q) = 1 - P(C_k/X_q) \quad (1.74)$$

Pour obtenir un classement optimal, il faut affecter toute observation X_q à la classe C_k qui minimise ce risque, donc qui maximise la probabilité *a posteriori*. Le problème de classification d'une observation X_q peut être résolu en adoptant la règle de décision suivante :

$$\text{Décider } C_s \text{ si : } P(C_s/X_q) > P(C_k/X_q), \forall k = 1, 2, \dots, K, (k \neq s) \quad (1.75)$$

L'application de la règle de Bayes revient à trouver des fonctions de décision $d_k(X_q)$, $k = 1, 2, \dots, K$, telles que l'observation X_q soit affectée à la classe C_s si et seulement si:

$$d_s(X_q) > d_k(X_q) \quad \forall k = 1, 2, \dots, K, \quad s \neq k \quad (1.76)$$

$$\text{où } d_k(X_q) = P(X_q/C_k)P(C_k) \quad (1.77)$$

Dans la plus part des cas, la probabilité conditionnelle de la classe C_k est de type gaussienne telle que [64][65]:

$$P(X_q/C_k) = \frac{\exp(-\frac{1}{2}(X_q - \mu_k)^T \Sigma_k^{-1}(X_q - \mu_k))}{(2\pi)^{\frac{N}{2}} |\Sigma_k|^{\frac{1}{2}}} \quad (1.78)$$

Dans cette expression, μ_k est le vecteur moyenne de la classe C_k et Σ_k la matrice de covariance de la classe C_k .

Ainsi, les fonctions de décision d_k peuvent être écrites sous la forme :

$$d_k(X_q) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (X_q - \mu_k)^T \Sigma_k^{-1} (X_q - \mu_k) + \log[P(C_k)] \quad (1.79)$$

Les probabilités $P(C_k)$ peuvent être estimées par :

$$P(C_k) = \frac{N_{C_k}}{N} \quad (1.80)$$

où N_{C_k} est le nombre d'observations appartenant à la classe C_k et N le nombre d'observations totales. μ_k et Σ_k sont estimés par la moyenne et la variance empirique à partir des prototypes de la classe C_k .

1.4.2. Méthode des k plus proches voisins

Contrairement à la méthode précédente où la probabilité conditionnelle de la classe C_k est choisie a priori (généralement gaussienne), dans la méthode des k plus proches voisins (kppv), cette probabilité est supposée inconnue, elle est alors estimée de la manière suivante :

Un volume $V(X_q)$ centré sur chaque point X_q des échantillons est déterminé de façon à ce qu'il englobe k observations de l'ensemble d'apprentissage. On dénombre alors le

nombre k_i de voisins de X_q appartenant à chaque classe C_k . La densité de probabilité de la classe C_k contenant N_{C_k} points prototypes est alors estimée comme suit [64] :

$$\hat{P}(X_q/C_k) = \frac{k_i}{N_{C_k} V(X_q)} \quad (1.81)$$

En remplaçant $P(C_k)$ par $\frac{N_{C_k}}{N}$ et $P(X_q/C_k)$ par $\hat{P}(X_q/C_k)$ dans la règle de Bayes (Equation (1.71)), on aboutit à la règle suivante :

$$X_q \in C_k \quad \text{si } P(C_k/X_q) = \max_{i=1,\dots,K} (k_i) \quad k = \sum_{i=1}^K k_i \quad (1.82)$$

Cette règle peut être énoncée comme suit : si les k plus proches voisins d'un point X_q appartiennent en majorité à la classe C_k , alors X_q appartiendra à la classe C_k .

1.5 Segmentation d'images texturées

Segmenter une image texturée revient à décomposer celle-ci en un ensemble de régions ayant des caractéristiques texturales homogène. Mathématiquement la segmentation est le partitionnement d'une image I en N régions R_i (ensembles de pixels) disjointes et homogènes selon un prédicat ou critère P déterminé (couleur, intensité, texture, niveau de gris, indice, ...) [1].

Formellement, l'ensemble des régions $\{R_1, R_2, \dots, R_N\}$ est une segmentation de l'image I si :

- 1- $\bigcup_{i=1}^N R_i = I, \quad i \in [1, N]$
 - 2- $R_i \cap R_j = \emptyset \quad \forall i \neq j$
 - 3- R_i est connexe, $\forall i \in [1, N]$
 - 4- $P(R_i) = \text{vrai} \quad \forall i \in [1, N]$
 - 5- $P(R_i \cup R_j) = \text{faux}, \quad \forall i \neq j$ et R_i adjacente à R_j
- (1.83)

La première condition implique que chaque pixel de l'image doit appartenir à une région R_i et l'union de toutes les régions correspond à l'image entière. La deuxième condition stipule l'intersection de deux régions distinctes égale à l'ensemble vide. La troisième condition est relative à la structure des régions. Elle définit une région comme un ensemble de pixels qui doivent être connexes. La quatrième condition exprime le fait que

chaque région doit respecter un prédicat d'uniformité P . La dernière condition implique le non réalisation de ce même prédicat pour la réunion de deux régions adjacentes.

La segmentation d'images texturées est basée sur les mêmes approches de la segmentation basée seulement sur le niveau de gris ou couleur. Elle consiste soit à extraire les contours des régions (type contour), soit les régions elles-mêmes (type région). Les méthodes de type contour détectent les zones de fortes discontinuités des attributs de texture. Contrairement à l'approche frontière qui recherche les dissimilarités, l'approche région recherche plutôt la similarité. Elle consiste à chercher les ensembles de pixels partageant des propriétés (attributs de texture) communes. Ces deux approches sont duales même si elles utilisent des formalismes différents. On s'intéressera dans ce qui suit qu'à l'approche de type région. Les étapes de la segmentation d'images texturées de type région sont résumées dans le schéma de la figure (1.13).

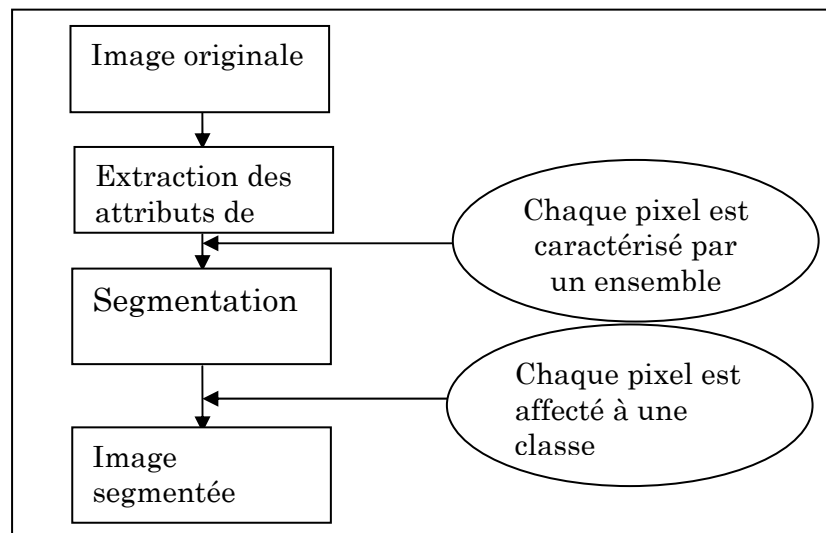


Fig. 1.13 : Etape de segmentation d'images texturées.

Les méthodes proposées dans ce cadre de la segmentation de type région peuvent être classées en deux catégories :

- Segmentation par analyse des propriétés spatiales.
- Segmentation par classification des pixels.

1.5.1 Segmentation par analyse des propriétés spatiales

Ces méthodes considèrent une région comme un ensemble de pixels connexes dont les attributs de texture sont proches les uns des autres. Elles procèdent par un balayage

de l'image afin de construire les régions et font intervenir l'information texturale pour la prise de décision. On distingue deux types d'approches :

- la segmentation par croissance de régions.
- la segmentation par division-fusion de régions.

Croissance des régions

La segmentation par croissance de régions consiste à faire croître des régions en y ajoutant successivement les pixels adjacents qui satisfont un critère d'homogénéité. La croissance s'arrête lorsque tous les pixels ont été traités. L'étape initiale consiste à sélectionner les germes des régions qui correspondent généralement à un pixel. Puis, les régions sont construites en y ajoutant successivement les pixels qui leur sont connexes et qui vérifient un critère de similarité. L'avantage de la croissance de régions est de préserver la forme de chaque région de l'image. Cependant, une mauvaise sélection des germes ou un choix du critère de similarité mal adapté peut entraîner des phénomènes de sous-segmentation ou de sur-segmentation. Ce type d'approche a été notamment utilisé par Reed et al pour la segmentation d'images texturées [60].

Division-Fusion

La segmentation par division-fusion procède quant à elle en deux étapes. Dans la première étape, l'image est divisée en quatre régions de taille plus petite. Chaque région non homogène est elle-même divisée en quatre sous régions de taille encore plus petite. Ce processus de division est répété jusqu'à ce que toutes les sous régions respectent le critère d'homogénéité. Dans la deuxième étape, les régions adjacentes issues de l'étape de division présentant des caractéristiques de texture identiques sont fusionnées. Cette approche a été utilisée par Chen et Pavlidis [61] en utilisant principalement les paramètres extraits de la matrice de cooccurrence comme attributs de texture. Doherty et al [62] ont proposé une méthode constituée de deux phases. Dans la première phase, l'algorithme division-fusion est appliqué en tenant compte que des niveaux de gris seulement. Dans la 2^{ème} phase, les régions formées lors de la première phase sont fusionnées si leurs attributs de texture sont proches. A noter que les attributs utilisés sont ceux extraits à partir de la matrice de cooccurrence. Les méthodes de segmentation par région exploitent correctement les informations contextuelles de l'image. L'inconvénient de ces méthodes est qu'elles produisent un nombre élevé de régions par rapport au nombre d'objets présents dans l'image.

1.5.2 Segmentation par classification des pixels

Des méthodes de classification non supervisées ont été abondamment utilisées pour la segmentation d'images texturées. Ces méthodes considèrent une région comme un ensemble de pixels connexes ayant des caractéristiques texturales similaires et forment un nuage de point dans l'espace des attributs. La classification consiste à retrouver des groupes de points qui forment des classes [3]. Pour atteindre cet objectif plusieurs techniques ont été proposées [48] [63]. Parmi elles, on peut citer les algorithmes K-means et Fuzzy C-means.

Algorithme des *K-means*

La méthode *K-means* fait partie d'un ensemble de techniques métriques de partition dont le principe consiste à choisir initialement K représentants $R_1, R_2, \dots, R_k, \dots, R_K$ de K classes $C_1, C_2, \dots, C_k, \dots, C_K$, définissant ainsi une représentation $R = \{R_1, R_2, \dots, R_k, \dots, R_K\}$ et une partition $C = \{C_1, C_2, \dots, C_k, \dots, C_K\}$. Chacune des observations à classer est affectée à l'une des C_k selon une mesure de similarité entre chaque représentant R_k et l'observation considérée. La partition C ainsi obtenue est utilisée pour redéfinir une nouvelle représentation R . Ce processus itératif, qui inclut les deux phases de définition de la partition C et la réactualisation de la représentation R , prend fin lorsqu'un critère mesurant l'adéquation entre la représentation R et la partition C est optimisé.

La méthode des *K-means*, connue sous le nom de procédure de minimisation de l'erreur quadratique [4], fonctionne suivant l'algorithme 2. La représentation R_k de chaque classe C_k est définie par son centre \bar{X}_k . La mesure de similarité utilisée est la distance euclidienne $d_k(X_q) = d(X_q, \bar{X}_k)$ de chacune des observations X_q aux différents centres \bar{X}_k choisis initialement. Le critère à optimiser est défini par :

$$J(C) = \sum_{l=1}^K \sum_{q=1}^{Q_K} d(X_q, \bar{X}_k)^2 = \sum_{l=1}^K J_k(C) \quad (1.84)$$

où Q_K représente le nombre d'observations de la classe C_k .

La partition C^* qui optimise ce critère est définie par :

$$J(C^*) = \min_C J(C) \quad (1.85)$$

Dans cet algorithme, les fonctions de décision d_k permettant l'affectation des observations aux différentes classes sont définies par :

$$d_k(X_q) = \frac{Q_k}{Q_k - 1} \|X_q - \bar{X}_k\|^2, k \neq s \text{ et } d_s(X_q) = \frac{Q_s}{Q_s - 1} \|X_q - \bar{X}_s\|^2, k = s \quad (1.86)$$

L'algorithme *K-means* nécessite, dans sa version de base, la spécification de nombreux paramètres, tels que le nombre de classes K et les partitions de départ. Les performances de cette procédure dépendent fortement des valeurs de ces paramètres prédéfinis par l'opérateur. En ce qui nous concerne, nous avons utilisé cette technique dans une version simplifiée. Nous avons donc donné au paramètre K la valeur correspondant au nombre exact de classes constituant chacun une texture. Les K centres sont générés d'une manière aléatoire.

Etape 1 : Initialisation

1. Lire la taille de la fenêtre de voisinage (w).
2. Lire le nombre de classes de texture (n_c).
3. Calculer les attributs de texture pour chaque pixel (analyse local).

Etape 2 : Initialisation des centres de classes

Initialiser au hasard (n_c) centre de gravité qui correspond à (n_c) classes.

Etape 3 : Affectation

1. Affecter chaque observation X_q à une classe C_k , Dont le centre est \bar{X}_k . L'observation X_q appartient à la classe C_k du centre \bar{X}_k si et seulement si $\|X_q - \bar{X}_k\|$ est minimum pour tout $k=1,2,\dots, n_c$.
2. Mettre à jour le centre de gravité \bar{X}_k .

$$\bar{X}_{kj} = \frac{1}{N_k} \sum_{i \in C_k} T_{ij} \text{ où } N_k \text{ le nombre de pixel de la classe } C_k.$$
3. Répéter à chaque fois l'étape 3 si les nouveaux centres ont changés.

Algorithme 2: Algorithme *K-means*

Algorithme de Fuzzy C-means

L'algorithme *Fuzzy C-means* (FCM) découle de l'algorithme *K-means*. L'idée de base est qu'un individu peut appartenir à plusieurs classes avec des degrés d'appartenance différents, contrairement à la classification exclusive où un individu ne peut appartenir qu'à une seule classe. Les degrés d'appartenance des observations aux différentes classes sont présentés dans une matrice de dimension ($Q \times K$) où Q est le nombre d'observations et K est le nombre de classes. Ces degrés d'appartenance notés μ_{qk} ($q = 1, \dots, Q$ et $k = 1, \dots, K$) doivent satisfaire les conditions suivantes :

$$\left\{ \begin{array}{l} \mu_{qk} \in [0, 1] \quad \forall q, k \\ \sum_{k=1}^K \mu_{qk} = 1 \quad \forall q \\ 0 < \sum_{q=1}^Q \mu_{qk} < Q \quad \forall k \end{array} \right. \quad (1.87)$$

Le principe de la méthode FCM consiste à minimiser la fonction objective suivante :

$$J_m(C) = \sum_{q=1}^Q \sum_{k=1}^K \mu_{qk}^m d_{qk}^2 \quad 1 < m < \infty \quad (1.88)$$

où m est le facteur flou et d_{qk} , la distance séparant l'objet X_q du centre \bar{X}_k de la classe k . Cette distance exprime la similarité entre l'observation et le centre d'une classe telle que :

$$d_{qk} = \|X_q - \bar{X}_k\| > 0, \quad \forall q \text{ et } k \quad (1.89)$$

Comme pour la méthode *K-means*, la méthode FCM fonctionne d'une manière itérative. A chaque itération, une mise à jour du degré d'appartenance μ_{qk} et des centres \bar{X}_k des classes est effectuée. La mise à jour des centres des classes est donnée par l'expression suivante :

$$\bar{X}_k = \frac{\sum_{q=1}^Q \mu_{qk}^m X_q}{\sum_{q=1}^Q \mu_{qk}^m} \quad (1.90)$$

La mise à jour des centres des classes modifie la répartition des observations, par conséquent une mise à jour de la matrice d'appartenance est nécessaire. Elle est effectuée en utilisant l'expression suivante :

$$\mu_{qk} = \frac{1}{\sum_{j=1}^K \left(\frac{d_{qk}}{d_{qj}} \right)^{2/m-1}} \quad \text{avec } \mu_{qk} = 1 \text{ si } d_{qk} = 0 \text{ et } \mu_{qk} = 0 \text{ pour } j \neq K / d_{qj} = 0 \quad (1.91)$$

Algorithme 3 présente l'algorithme *Fuzzy C-means* où la variable t indique le numéro de l'itération. Le critère d'arrêt est indiqué par la dernière étape de l'algorithme. Il consiste à vérifier si la répartition entre deux itérations est la même ou non. ε est un nombre réel dont la valeur est comprise entre 0 et 1.

- 1- Fixer le nombre K de classes et le facteur flou m .
- 2- Initialisation des centres des classes, $t = 0$.
- 3- Déterminer la matrice d'appartenance $U^{(0)}$ en utilisant les équations 1.88 et 1.89.
- 4- $t = t + 1$ et déterminer les nouveaux centres des classes en utilisant l'équation 1.89.
- 5- Déterminer la matrice d'appartenance $U^{(t)}$ en utilisant les équations 1.88 et 1.90.
- 6- Si $\|U^{(t+1)} - U^{(t)}\| < \varepsilon$ s'arrêter; sinon aller à l'étape 4.

Algorithme 3: Algorithme *Fuzzy C-means*

L'algorithme FCM possède les mêmes inconvénients que l'algorithme *K-means* à savoir la sensibilité à la répartition initiale et le choix du nombre de classes. Comme pour l'algorithme *K-means*, nous avons également utilisé cette technique dans sa version simplifiée en donnant à K la valeur correspondant au nombre exact de classes et en initialisant les centres des classes très près des véritables centres.

1.6 Conclusion

L'analyse de la texture est un domaine du traitement d'images très riche qui englobe plusieurs méthodes dont les concepts sont très variés. Il existe également d'autres techniques telles que la méthode de longueurs de plages, les modèles fractales.

L'analyse de la texture est souvent associée à la classification des données pour résoudre des problèmes de reconnaissance des formes ou de segmentation. En reconnaissance des formes, la classification des images texturées est généralement de type supervisée. Par contre, la segmentation des images texturées est basée sur la classification non supervisée. Ce type de méthodes sont plus intéressantes et également très nombreuses. Les algorithmes *K-means* et *Fuzzy C-means* sont parmi ces méthodes appartenant à cette catégorie les plus connus. Ils sont utilisés dans le chapitre 3 pour segmenter une image à base des matrices Aura qui fera objet du prochain chapitre. D'autres méthodes de classification non supervisée utilisées en segmentation existent tels que l'algorithme d'estimation-maximisation (EM), la méthode *mean-shift*, les réseaux de neurones (Kohonen, ART), ect... .

Chapitre 2

Etude théorique du concept Aura

2.1 Introduction

Les modèles des champs aléatoires de Markov et les matrices de cooccurrences de niveau de gris sont souvent considérés séparément comme des outils très populaires pour l'analyse de la texture. Cependant, les matrices de cooccurrence prennent en compte seulement l'interaction entre deux pixels et les champs aléatoires de Gibbs utilisent des cliques de petites tailles (ordre 2 en générale) afin de prendre en compte les relations de voisinage entre deux pixels. Ces deux méthodes ne permettent pas en général de modéliser toutes les textures naturelles. Pour remédier à ce problème, le concept Aura a été introduit afin de prendre en compte l'interaction entre plusieurs pixels [12].

Nous allons présenter dans ce chapitre les notions de base sur les ensembles Aura, mesures Aura et matrices Aura, leurs liens avec la morphologie mathématique, les matrices de cooccurrence, les matrices covariance et les champs aléatoires de Gibbs ainsi que leurs intérêts en analyse d'images.

2.2 Ensemble Aura

On considère une grille finie S constituée d'un ensemble de sites s et munie d'un système de voisinage $N = \{N_s, s \in S\}$, où N_s est l'ensemble des sites voisins du site s . Soient deux sous ensembles A et B de S tel que $A, B \subseteq S$. L'Aura de A par rapport à B dénotée par $\vartheta_B(A, N)$, pour un système de voisinage N est défini comme suit [5][6][10][17]:

$$\vartheta_B(A, N) = \bigcup_{s \in A} (N_s \cap B) \quad (2.1)$$

L'Aura de A par rapport à B est un sous-ensemble qui contient des éléments de B appartenant au voisinage du sous ensemble A . Autrement dit, l'ensemble Aura $\vartheta_B(A, N)$ décrit la présence d'un sous-ensemble B dans le voisinage du sous-ensemble A (Fig. 2.2). Dans le cas où $B = A^c$ (ensemble complémentaire de A) et N_s contient les plus proches voisins de s , l'ensemble Aura $\vartheta_B(A, N)$ forme un anneau autour de A . Et c'est justement de là que le mot « Aura » a été tiré. En effet, le mot Aura est utilisé pour décrire la forme du contour décrivant un ensemble comme l'a définit Elfadel et Picard dans [7][13] : « *The name aura comes from the « surrounding shape » of the set it describes* ».

C'est aussi un mot qui désigne le "Halo", c'est-à-dire un cercle ou un anneau brillant qu'on aperçoit autour du soleil, de la lune ou d'une source lumineuse (Fig. 2.1).



Fig. 2. 1 : Exemple d'un « Halo lunaire »

On peut voir à travers cette définition que l'Aura fait appel à la notion d'ensembles dont quelques définitions sont rappelées en Annexe A.

2.2.1 Propriétés des ensembles Aura

Soit une grille G et soient A et B avec un système de voisinage donné N . Pour simplifier les notations, on écrira l'Aura de A par rapport à N pour un système de voisinage N :

- 1) L'ensemble Aura par rapport à l'ensemble vide est un ensemble vide:
- 2) L'Aura de A par rapport à N est incluse dans l'ensemble A :
- 3) L'auto Aura de A est incluse dans A :

En général, $A \cap B$, à moins que la structure de voisinage soit définie tels que chaque site de la grille est dans son propre voisinage.

- 4) L'Aura de $A \cup B$ par rapport à N est égale à l'union de leurs Aura par rapport à N :

5) L'Aura de $A \cap B$ par rapport à C est égale à l'intersection de leurs Aura par rapport à C : $\vartheta_C(A \cap B) \subseteq \vartheta_C(A) \cap \vartheta_C(B)$ (2.6)

6) Si $A \subseteq B$ alors, $\vartheta_C(A) \subseteq \vartheta_C(B)$ (2.7)

7) L'Aura de A par rapport à $B \cup C$ est l'union de l'Aura de A par rapport à B et à C : $\vartheta_{B \cup C}(A) = \vartheta_B(A) \cup \vartheta_C(A)$ (2.8)

8) L'Aura de A par rapport à $B \cap C$ est l'intersection de l'Aura de A par rapport à B et à C : $\vartheta_{B \cap C}(A) = \vartheta_B(A) \cap \vartheta_C(A)$ (2.9)

9) Généralement, l'opération Aura n'est pas symétrique : $\vartheta_B(A) \neq \vartheta_A(B)$ (2.10)

Ces propriétés peuvent être facilement démontrées à partir de la définition de l'ensemble Aura $\vartheta_B(A, N) = \bigcup_{s \in A} (N_s \cap B)$ et des propriétés sur les ensembles présentées en Annexe A.

La grille S peut être une image dont les sites sont régulièrement espacés et correspondent aux pixels de l'image. On utilisera ainsi par la suite le terme « site » pour désigner soit un point d'une grille quelconque soit un pixel d'une image.

Un exemple d'un ensemble Aura est illustré sur la figure (2.2). La figure (2.2(a)) montre une grille de sites binaires S de taille (5×5) . Soit A le sous ensemble de sites ayant des valeurs 1 tel que $A = \{a/ a \in S, a = 1\}$ et soit B le sous ensemble de sites de valeurs 0 tel que $B = \{b/ b \in S, b = 0\}$. Soit N_s une structure de voisinage à 4-connexes $N_s = \{n_i, i = 0, 1, 2, 3\}$ schématisée sur la figure (2.2(b)). L'Aura de A par rapport à B , est constitué des éléments encadrés dans la figure (2.2(c)).

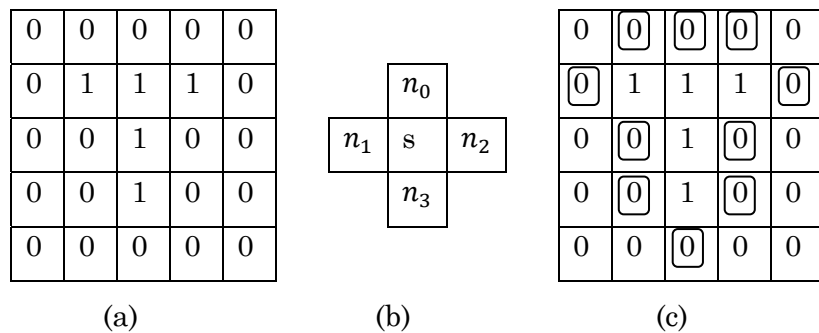


Fig. 2. 2 : Exemple d'un ensemble Aura.

(a) : Image binaire, (b) : Structure de voisinage, (c) : Ensemble Aura

Nous pouvons remarquer sur la figure (2.2) que l'Aura de A par rapport à B forme la frontière entre les ensembles de sites ayant respectivement la valeur 1 et la valeur 0. Il est également très important de noter que l'ensemble Aura dépend de la structure de voisinage choisie sur une grille.

2.3 Notion de voisinage

Les sites de la grille S sont considérés comme des sources indépendantes d'information. Pour analyser les relations spatiales entre ces sites, on fait appel à la notion de voisinage. Soit $N = \{N_s, s \in S\}$ un système de voisinage défini sur une grille rectangulaire de sites S . N_s étant un ensemble de sites voisins du site s . La disposition des sites voisins dans N_s définit une structure de voisinage caractérisée par sa dimension d qu'on appelle ordre.

Pour un système de voisinage d'ordre d , l'ensemble N_s de sites voisins d'un site s de coordonnées (i, j) est:

$$N_{s=(i,j)} = \{r = (k, l) \in S \mid 0 < d(s, r) \leq d\} \quad (2.11)$$

où $d(s, r)$ est la distance qui sépare les sites s et r .

Les distances les plus couramment utilisées sont :

$$\text{Distance de Manhattan (City block): } d_1(s, r) = |k - i| + |l - j| \quad (2.12)$$

$$\text{Distance euclidienne : } d_2(s, r) = [(k - i)^2 + (l - j)^2]^{1/2} \quad (2.13)$$

$$\text{Distance de l'échiquier (Chessboard): } d_{inf}(s, r) = \text{Max}(|k - i|, |l - j|) \quad (2.14)$$

$$\text{Ces distances sont reliées par la propriété : } d_{inf}(s, r) \leq d_2(s, r) \leq d_1(s, r) \quad (2.15)$$

Selon le type de distance utilisé $d_{inf}(s, r)$ ou $d_1(s, r)$, on peut définir deux types de connexité : 4 et 8. Ce nombre correspond à la taille du plus petit voisinage non vide d'un site (voisinage d'ordre $d = 1$) selon que l'on choisit la distance d_1 ou d_{inf} . Un site a 4 voisins directs si l'on choisit la distance d_1 (Fig. 2.3.a), il en a 8 avec la distance d_{inf} (Fig. 2.3.e).

La figure (2.3) montre quelques configurations usuelles de voisinage d'ordre d d'un site s de coordonnées (i, j) en connexités 4 et 8.

Les sites voisins dans les structures de voisinage de la figure (2.3) sont adjacents au site central $s(i,j)$ et forment des ensembles connexes. D'autres structures de voisinages exotiques, où les sites voisins ne sont pas adjacents au site central, peuvent être définies (Fig. 2.4).

En principe N_s ne dépend pas de la position du site s . Dans ce cas, il est considéré comme une version translaté d'un élément de base notée E et qu'on appelle élément structurant en termes de morphologie mathématique. Par conséquent $|N_s| = |E|, \forall s$.

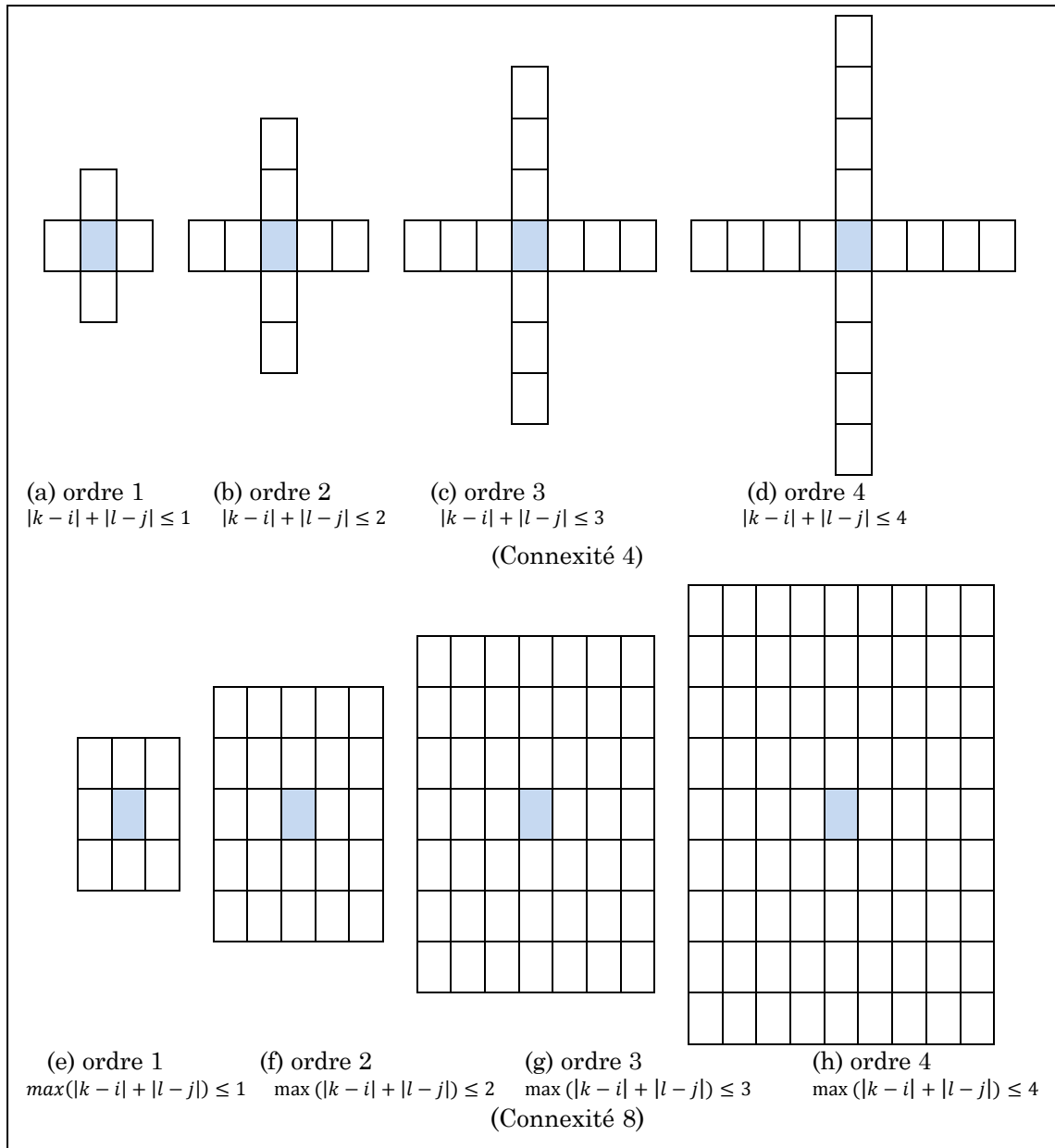


Fig. 2.3: Configurations de voisinage usuels d'ordre $d \leq 4$ en connexité 4 et 8

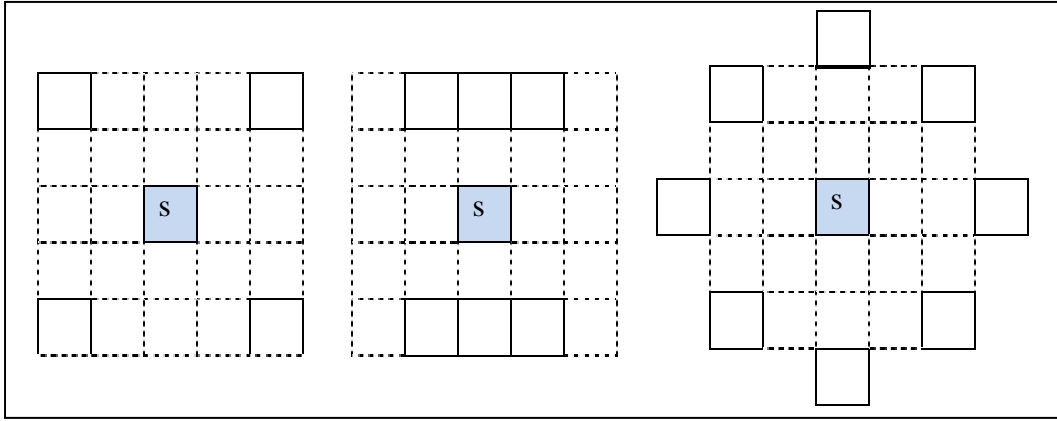


Fig. 2. 4: Configurations de voisinages exotiques.

2.3.1 Système de voisinage symétrique et asymétrique

Un système de voisinage $N = \{N_s, s \in S\}$ est symétrique si : $\forall s, r \in S, s \in N_r$ si et seulement si $r \in N_s$ [5][6].

Une autre définition proposée dans [6][17] stipule qu'un système de voisinage N est symétrique (asymétrique) si son élément structurant correspondant E est symétrique (asymétrique). La figure (2.5) donne quelques exemples de structures de voisinage symétriques et asymétriques.

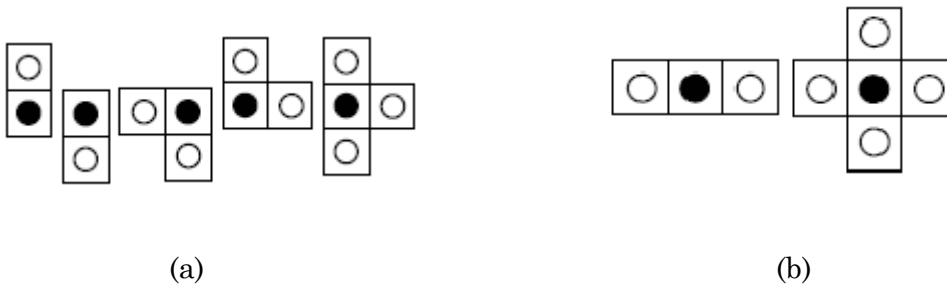


Fig. 2.5: Exemples de structures de voisinage (a) Asymétriques, (b) Symétriques.

2.4 Mesure Aura

La mesure Aura notée $m(A, B)$ est définie comme suit [7][10][13][17] :

$$\begin{aligned}
 m(A, B) &= m(A, B, N) \\
 &= \sum_{s \in A} |N_s \cap B| \\
 &\geq \vartheta_B(A, N)
 \end{aligned} \tag{2.16}$$

Notons que pour un sous-ensemble donné $A \subseteq S$, $|A|$ représente le cardinal de A .

A titre d'exemple, la mesure Aura de l'ensemble A par rapport à B de la figure (2.2) vaut $m(A, B) = 12$, et reste supérieur à la taille de l'ensemble Aura $|\vartheta_B(A)| = 10$.

La mesure Aura $m(A, B)$ donne une information sur le mélange de deux sous ensembles. Elle évalue la quantité de sites qui sépare ou qui unit les deux sous ensembles A et B . Une valeur élevée de $m(A, B)$ indique que A et B sont fortement mélangés alors qu'une faible valeur indique que A et B sont séparés.

2.4.1 Propriétés de la mesure Aura

Soit une grille S muni d'une structure de voisinage donnée et soient A et B deux ensembles tels que $A, B \subseteq S$ [7][13] :

$$1) m(A, B) \text{ est une fonction non négative : } \forall A \subseteq S, m(A, B) \geq 0. \quad (2.18)$$

$$2) m(A, B) \text{ est monotone : } A \subseteq A' \Rightarrow m(A, B) \leq m(A', B). \quad (2.19)$$

$$3) m(A, B) \text{ est sub-additif: } m(\bigcup_{i=1}^n A_i, B) \leq \sum_{i=1}^n m(A_i, B) \quad (2.20)$$

$$4) \text{ En générale, } m(A, B) \neq m(B, A). \text{ Mais, si } N \text{ est un système de voisinage symétrique, alors : } m(A, B) = m(B, A) \quad (2.21)$$

$$5) \text{ Si } N_s = \{s\}, \text{ alors } m(A, B) = m(B, A) = |A \cap B| \quad (2.22)$$

$$6) \text{ Si } N_s = S, \text{ alors } m(A, B) = m(B, A) = |A||B| \quad (2.23)$$

7) Si $A \cup B = S$ et $A \cap B = \emptyset$ on a:

$$m(A, A) + m(A, B) = |A||N| \quad (2.24)$$

$$m(B, A) + m(B, B) = |B||N| \quad (2.25)$$

8) Soient A, B, C trois sous-ensembles disjoints dans S . Soient $A', B', A_s \subseteq A$, et $B_s \subseteq B$, tel que $|A_s| = |B_s|$, $A' = (A - A_s) \cup B_s$ et $B' = (B - B_s) \cup A_s$, on a [13]:

$$\begin{aligned} m(A', A') + m(B', B') + m(A', B') + m(B', A') \\ = m(A, A) + m(B, B) + m(A, B) + m(B, A) \end{aligned} \quad (2.26)$$

et

$$m(A', C) + m(B', C) = m(A, C) + m(B, C) \quad (2.27)$$

Nous présenterons en annexe B les preuves des propriétés 4, 7 et 8. Les autres propriétés peuvent être facilement démontrées à partir de la définition de $m(A,B)$ (Eq. 2.16).

2.5 Intérêt pratiques des mesures Aura

Pour une grille munie d'une structure de voisinage 4-connexes, la mesure Aura de deux sous ensembles A et B est égale à la longueur de la frontière (calculée en nombre de sites) entre deux sous ensemble A et B . Cette propriété présente un grand intérêt dans différents domaines comme la mécanique des fluides et la théorie des graphes [12].

Dans le cas de la mécanique des fluides, on peut assimiler les deux sous ensembles A et B à deux fluides. La mesure Aura $m(A,B)$ caractérise alors le caractère miscible ou non de ces deux fluides : $m(A,B)$ est maximale si A et B sont miscibles et minimale dans le cas contraire. La miscibilité désigne usuellement la capacité de divers liquides à se mélanger. Si le mélange obtenu est homogène, les liquides sont qualifiés de miscibles. Inversement, les liquides sont dits non-miscibles s'ils ne peuvent pas se mélanger et forment un mélange hétérogène : on observe alors plusieurs phases. Le liquide de densité plus faible sera alors placé au-dessus de l'autre. C'est le cas par exemple de l'eau et de l'huile (Fig. 2.6). Notant qu'en chimie et en physique, une phase désigne une région de l'espace dont les paramètres ou propriétés sont uniformes, ou du moins varient peu entre deux points voisins. Le contact entre deux phases est la surface de séparation.

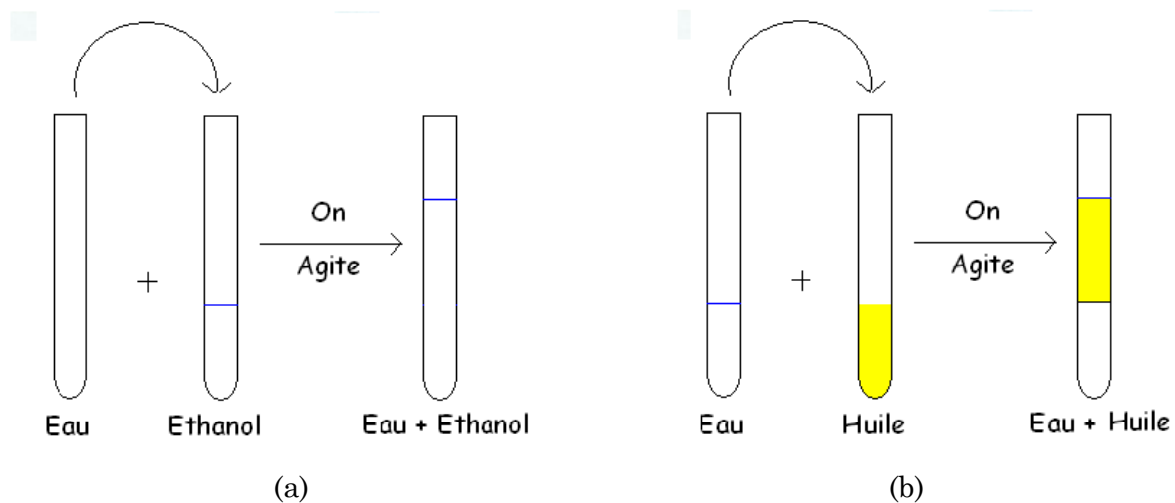


Fig. 2.6 : Exemple d'un mélange de deux fluides
(a)Fluide miscible, (b) Fluide non miscible.

2.6 Mesures Aura sur les sous-voisinages

La plupart des textures naturelles sont anisotrope, il est alors préférable de décomposer chaque voisinage N_s en un ensemble de sous-voisinages isotropes $N_s^k \subseteq N_s$, tel que [7][13]:

$$N_s = \bigcup_{k=1}^K N_s^k, \quad \forall s \in S \quad \text{Où} \quad N_s^k \cap N_s^l = \emptyset, \forall l \neq k \quad (2.28)$$

L'Aura de A par rapport à B pour la structure de $k^{\text{ème}}$ sous-voisinage est :

$$\vartheta_B^k(A) = \bigcup_{s \in A} (B \cap N_s^k) \quad (2.29)$$

L'Aura de A par rapport à B est par conséquent:

$$\vartheta_B(A) = \bigcup_{k=1}^K \vartheta_B^k(A) \quad (2.30)$$

La mesure Aura avec le $k^{\text{ème}}$ sous-voisinage est :

$$m^k(A, B) = \sum_{s \in A} |N_s^k \cap B| \quad (2.31)$$

Puisque le $k^{\text{ème}}$ sous-voisinage N_s^k est une partition de N , nous pouvons facilement déduire que :

$$m(A, B) = \sum_{k=1}^K m^k(A, B) \quad (2.32)$$

La mesure Aura de deux ensembles A et B pour une structure de voisinage est égale à la somme des mesures Aura de ces deux ensembles déterminées en utilisant tous les sous-voisinages de la structure.

2.7 Matrice Aura

La matrice Aura est une matrice qui englobe les mesures Aura de tous les sous-ensembles considérés sur une grille donnée [6][7].

Soit N un système de voisinage sur une grille S , et soit $\{S_i, 0 \leq i \leq G-1\}$ une partition de S , telle que $\bigcup_{i=0}^{G-1} S_i = S$ et $S_i \cap S_j = \emptyset$ pour $i \neq j$, G étant le nombre maximal de partitions. La matrice Aura notée A de dimension $(G \times G)$ est définie par [5][6][10]:

$$A = A(N) = [a_{ij}] = [m(S_i, S_j)] \quad (2.33)$$

2.7.1 Propriétés de la Matrice Aura

Soit $A = [a_{ij}]$ une matrice Aura.

1) Chaque somme de lignes satisfait

$$\sum_{j=0}^{G-1} a_{ij} = |S_i||N|, \quad \forall i = 0, \dots, G-1; \quad (2.34)$$

2) Chaque somme de colonnes satisfait

$$\sum_{i=0}^{G-1} a_{ij} = |S_j||N|, \quad \forall j = 0, \dots, G-1; \quad (2.35)$$

3) Si N est symétrique, alors A est symétrique, et

$$\sum_{j=0}^{G-1} a_{kj} = \sum_{i=0}^{G-1} a_{ik} = |S_k||N|, \quad \forall k = 0, \dots, G-1. \quad (2.36)$$

2.8 Matrice Aura des niveaux de gris

On se place maintenant dans le cadre du traitement d'images. Soit I une image en niveaux de gris de taille $(H \times L)$ représentée sur une grille S telle que $S = \{s = (i, j); 0 \leq i \leq H-1 \text{ et } 0 \leq j \leq L-1\}$. Soit x_s le niveau de gris du site s telle que $x_s \in \Lambda = \{0, 1, \dots, G-1\}$. G étant le niveau de gris maximal. Soient S_g l'ensemble des sites ayant le niveau de gris g définie sur S tel que $S_g = \{s \in S | x_s = g\}$, $\forall g \in \Lambda$. Il est clair que les sous ensembles S_g constituent des partitions de S . Le vecteur $h(g) = |S_g|$ tel que $g = 0, 1, \dots, G-1$ correspond à l'histogramme de l'image.

En considérant un système de voisinage $N = \{N_s, s \in S\}$ où N_s est l'ensemble des sites voisins de site s , on peut définir l'Aura de l'ensemble S_g par rapport à l'ensemble $S_{g'}$ par :

$$\vartheta_{S_{g'}}(S_g) = \vartheta_{S_{g'}}(S_g, N) = \bigcup_{s \in S_g} (N_s \cap S_{g'}) \quad (2.37)$$

La mesure Aura $m(S_g, S_{g'})$ est :

$$m(S_g, S_{g'}) = \sum_{s \in S_g} |N_s \cap S_{g'}| \quad (2.38)$$

Par la suite on notera $m(g, g')$ au lieu de $m(S_g, S_{g'})$.

La matrice Aura des niveaux de gris est définie comme suit :

$$A = [m(g, g')] \quad (2.39)$$

La mesure Aura $m(g, g')$ représente le nombre de fois où un site s de niveau de gris g a dans son voisinage N_s un site de niveau de gris g' . La dimension de la matrice est évidemment de $(G \times G)$. En considérant l'exemple de la figure (2.2), on a $\Lambda = \{0,1\}$ et la matrice Aura est alors:

$$A = \begin{bmatrix} m(0,0) & m(0,1) \\ m(1,0) & m(1,1) \end{bmatrix} = \begin{bmatrix} 48 & 12 \\ 12 & 8 \end{bmatrix}.$$

Remarque

Une matrice Aura de miscibilité A_n est définie comme une matrice Aura de niveaux de gris normalisée. Dans le cas d'une structure de voisinage symétrique et pour un histogramme uniforme ($|S_g| = \frac{|S|}{G}$), la matrice Aura de miscibilité est définie dans [13] par:

$$A_n = [m_n(g, g')] = \frac{G}{|N_s||S|} A \quad (2.40)$$

D'une manière plus générale, la matrice Aura de miscibilité est :

$$A_n = \frac{A}{|A|}, \quad (2.41)$$

telle que

$$\sum_{g=0}^{G-1} \sum_{g'=0}^{G-1} m_n(g, g') = 1 \quad (2.42)$$

Par abus de langage, une matrice Aura est aussi appelée matrice de Miscibilité [12].

$m_n(g, g')$ est une mesure de miscibilité qui peut être interprétée comme la probabilité qu'un site s de niveau de gris g ait dans son voisinage N_s un site de niveau de gris g' [13] :

$$m_n(g, g') = P(\exists r \in N_s, x_r = g' \mid \exists s \in S, x_s = g) \quad (2.43)$$

2.9 Relation entre les ensembles Aura et la morphologie mathématique

L'analyse d'image par la morphologie mathématique consiste à déplacer un élément structurant E sur l'image et à étudier les interactions entre les objets de l'image et E . L'un des intérêts de l'ensemble Aura est son lien à la morphologie

mathématique et plus exactement à l'opérateur de dilatation. Rappelons que la dilatation d'un ensemble A par un élément structurant E est :

$$A \oplus E = \{s \mid N_s \cap A \neq \emptyset\} = \bigcup_{t \in E} A_t \quad (2.44)$$

où N_s correspond à l'élément structurant E centré sur le site s . A_t est le translaté de l'ensemble A par le vecteur t .

L'ensemble Aura de A par rapport à B peut être exprimé en termes de dilatation morphologique comme suit [7][13] :

$$\vartheta_\beta(A, N) = \bigcup_{s \in A} (N_s \cap B) = \left(\bigcup_{s \in A} N_s \right) \cap B = (E \oplus A) \cap B \quad (2.45)$$

Si E est l'élément structurant, N_s désigne un élément structurant centré sur le site s . C'est donc une version translatée de E . L'Aura de A par rapport à B est ainsi équivalent à la dilatation de A avec E suivie de l'intersection avec B .

La figure (2.6) montre un exemple de calcul d'un ensemble Aura en utilisant la dilatation. La figure (2.6(a)) montre une image de taille (5x5) quantifiée sur 6 niveaux de gris. On définit l'ensemble A par les sites ayant le niveau de gris 0 et l'ensemble B par les sites de niveau de gris 2. $A = \{a / a \in S, a = 0\}$ et $B = \{b / b \in S, b = 2\}$. L'élément structurant E est présenté sur la figure (2.6 (b)). La dilatation $A \oplus E$ est donnée dans la figure (2.6(c)). L'intersection de cet ensemble avec B donne l'Aura de A par rapport à B qui est montré dans la figure (2.6(d)).

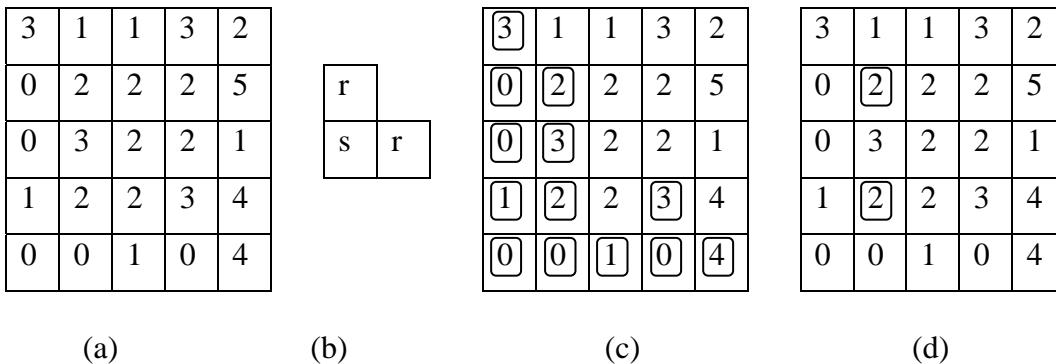


Fig. 2.6 : Exemple de calcul d'un ensemble Aura en utilisant la dilatation morphologique. (a) Image en niveaux de gris, (b) Elément structurant E , (c) Dilatation $A \oplus E$, (d) Ensemble Aura $\vartheta_\beta(A, E) = (E \oplus A) \cap B$.

2.10 Relation entre la matrice Aura et la matrice de cooccurrence

L'analyse de la texture basée sur les matrices de cooccurrence est sans doute la méthode la plus populaire, d'où l'intérêt d'établir une relation entre les matrices de cooccurrence et les matrices Aura.

Un élément d'une matrice de cooccurrence est défini par [13]:

$$C_{gg'}(d) = \{Card(s, r) \in S \times S \text{ tel que } s - r = d, x_s = g, x_r = g'\} \quad (2.46)$$

$C_{gg'}(d)$ représente le nombre de couples de sites séparés par un vecteur d dont l'un a un niveau de gris g et l'autre a un niveau de gris g' .

La matrice de cooccurrence est notée $C(d) = [C_{gg'}(d)]$ est de dimension $(G \times G)$. Or un élément de la matrice Aura $m(S_g, S_{g'})$ est défini par le nombre de sites de niveau de gris g ayant dans leurs voisinages définis par N_s des sites de niveaux de gris g' . Compte tenue de ces deux définitions, si N_s contient un seul site tel que $N_s = \{s + d\}$, alors on a :

$$m(S_g, S_{g'}) = m(g, g') = C_{gg'}(d) \quad (2.47)$$

La cooccurrence du couple (g, g') peut être interprétée comme la longueur de la frontière de deux ensembles de niveaux de gris S_g et $S_{g'}$.

Dans le cas général où un système de voisinage composé de k sous voisinages N_s tel que $N_s = \bigcup_{k=1}^K N_s^k$ où N_s^k contient un seul site séparé du site s par le vecteur d_k tel que:

$$N_s^k = \{s + d_k\} \text{ et } K = |N_s|, \quad (2.48)$$

on peut décrire la matrice Aura comme la somme des matrices de cooccurrence obtenues en utilisant des vecteurs de déplacement d_k contenus dans l'élément structurant N_s [13]:

$$m(g, g') = \sum_{k=1}^K m^k(g, g') = \sum_{k=1}^K C_{gg'}(d_k) \quad (2.49)$$

Une matrice Aura de niveaux de gris est donc une généralisation des matrices de cooccurrences des niveaux de gris mais différente de celle proposée par Davis (voir Chap1). Cependant, il y'a lieu d'indiquer qu'une matrice de cooccurrence permet de

caractériser les relations spatiales entre deux sites seulement alors que les matrices Aura permettent de modéliser les relations spatiales entre plusieurs sites.

2.11 Relation entre la matrice Aura et la matrice de covariance

La fonction de covariance ou d'auto-corrélation permet de mesurer la ressemblance entre une image I et sa version translatée I^t . Elle est souvent utilisée en analyse de la texture (voir Chap.1). Sa relation avec les matrices Aura est décrite dans ce qui suit [13]:

Soit d un vecteur de déplacement, l'auto-corrélation est alors donnée par la relation suivante :

$$C(d) = \sum_{s \in S} x_s x_{s+d} \quad (2.50)$$

qui peut être reformulée comme suit:

$$C(d) = \sum_{g \in \Lambda} \sum_{s \in S_g} g x_{s+d} = \sum_{g \in \Lambda} \sum_{s \in S_g} g \sum_{g' \in \Lambda} g' \sum_{(s+d) \in S_{g'}} 1 \quad (2.51)$$

Comme $\sum_{(s+d) \in S_{g'}} 1 = \sum_{s \in S_g} |(s+d) \cap S_{g'}| = C_{gg'}(d)$, l'auto-corrélation s'écrira:

$$C(d) = \sum_{g, g' \in \Lambda} g g' C_{gg'}(d) \quad (2.52)$$

où $C_{gg'}(d)$ est l'expression de l'élément de la matrice de cooccurrence.

La fonction d'auto-corrélation peut être également définie pour une structure de voisinage donnée comme suit :

$$C(d) = \sum_{s \in S} \sum_{r \in N_s} x_s x_r \quad (2.53)$$

En utilisant les mêmes transformations que précédemment, on aura :

$$C(d) = \sum_{g \in \Lambda} \sum_{s \in S_g} \sum_{r \in N_s} g x_r$$

$$C(d) = \sum_{g \in \Lambda} \sum_{s \in S_g} \sum_{g' \in \Lambda} \sum_{r \in N_s \cap S_{g'}} g g'$$

$$\begin{aligned}
 C(d) &= \sum_{g \in \Lambda} \sum_{g' \in \Lambda} g g' \sum_{s \in S_g} \sum_{r \in N_s \cap S_{g'}} 1 \\
 C(d) &= \sum_{g, g' \in \Lambda} g g' \sum_{s \in S_g} |N_s \cap S_{g'}| \\
 C(d) &= \sum_{g, g' \in \Lambda} g g' m(g, g') \tag{2.54}
 \end{aligned}$$

Ainsi, en égalisant les équations (2.53) et (2.54), on obtient :

$$C(d) = \sum_{g, g' \in \Lambda} g g' C_{gg'}(d) = \sum_{g, g' \in \Lambda} g g' m(g, g') \tag{2.55}$$

Cette équation montre que les mesures Aura et les matrices de cooccurrences ont une relation avec les matrices d'auto-corrélation même si elles sont différentes. Cependant, la dimension ($G \times G$) des matrices de cooccurrences et des matrices aura est différente de celle des matrices d'auto-corrélation qui dépend de la taille de l'image.

2.12 Relation entre la matrice Aura et les champs aléatoires de Markov

Les champs aléatoires de Markov jouent un rôle important en traitement d'images [9][16]. Quelques notions sur les champs de Markov ont été particulièrement présentées dans le chapitre précédent. Etant donné, l'équivalence entre un champ de Markov et un champ de Gibbs, nous allons reformuler dans cette section, l'énergie de Gibbs en fonction des mesures Aura. Notons qu'un champ de Gibbs possède une distribution de probabilité définie comme suit :

$$P(x) = \frac{1}{Z} \exp\left(-\frac{1}{T} U(x)\right) \tag{2.56}$$

où T est « la température » du champ et Z une constante de normalisation. $U(x)$ est la fonction d'énergie qui peut être formulée de la manière suivante:

$$U(x) = \sum_{s \in S} V_s(x_s) + \sum_{s \in S} \sum_{r \in N_s} V_{sr}(x_s, x_r) \tag{2.57}$$

V_s est le potentiel d'un site unique s et V_{sr} est le potentiel d'interaction de deux sites s et r . V_s et V_{sr} sont aussi appelés respectivement champ externe et champ interne.

Plusieurs modèles de potentiel sont proposés. Les plus connus sont le modèle auto-binomial et le modèle de Potts [9].

L'énergie de Gibbs d'un modèle auto-binomial est :

$$U(x) = - \sum_{s \in S} (\alpha_s x_s + T \ln \binom{G-1}{x_s}) + \sum_{r \in N_s} \beta_{sr} x_s x_r \quad (2.58)$$

où α_s et β_{sr} sont les paramètres du modèle et $\binom{G-1}{x_s} = \frac{(G-1)!}{x_s!(G-1-x_s)!}$ est le coefficient binomial.

Dans le cas où le champ de l'image est homogène (α_s est indépendant de la position du site s), $\alpha_s = \alpha$, et $\beta_{sr} = \beta_r$. Si de plus le champ est isotrope, on a alors, $\beta_r = \beta \forall r \in N_s$ et l'énergie de Gibbs d'un modèle auto binomial homogène prend la forme suivante [9][16] :

$$U(x) = - \sum_{s \in S} \left(\alpha x_s + \sum_{r \in N_s} \beta_r x_s x_r \right) \quad (2.59)$$

L'énergie de modèle de Potts est donnée par la formule suivante [9]:

$$U(x) = \sum_{s \in S} \sum_{r \in N_s} \beta_r (2\delta_{x_s x_r} - 1) \quad (2.60)$$

avec

$$\delta_{x_s x_r} = \begin{cases} 1 & \text{si } x_s = x_r \\ 0 & \text{sinon} \end{cases} \quad (2.61)$$

Soit $\{S_g, g \in \Lambda\}$ une partition de S tel que $S_g = \{s \in S \mid x_s = g\}$. En négligeant le champ externe ($\alpha_s = 0$), l'énergie de Gibbs prend la forme suivante [9]:

$$U(x) = \sum_{s \in S} \sum_{r \in N_s} V_{rs}(x_s, x_r), \quad (2.62)$$

celle-ci peut encore s'écrire :

$$U(x) = \sum_{g \in \Lambda} \sum_{s \in S_g} \sum_{r \in N_s} V_{r,s}(g, x_r) \quad (2.63)$$

En décomposant le système de voisinage en k sous voisinages tel que $N_s = \bigcup_{k=1}^K N_s^k$, on aura :

$$U(x) = \sum_{g \in \Lambda} \sum_{s \in S_g} \sum_{k=1}^K \sum_{r \in N_s^k} V^k(g, x_r) \text{ pour } r \in S_{g'}, \text{ et } x_r = g' \quad (2.64)$$

ou encore :

$$U(x) = \sum_{g \in \Lambda} \sum_{s \in S_g} \sum_{k=1}^K \sum_{g' \in \Lambda} \sum_{r \in N_s^k \cap S_{g'}} V^k(g, g') \quad (2.65)$$

cette expression est équivalente à :

$$U(x) = \sum_{k=1}^K \sum_{g, g' \in \Lambda} V^k(g, g') \sum_{s \in S_g} |N_s^k \cap S_{g'}| \quad (2.66)$$

Finalement l'énergie de Gibbs prend la forme générale suivante :

$$U(x) = \sum_{k=1}^K \sum_{g, g' \in \Lambda} V^k(g, g') m^k(g, g') \quad (2.67)$$

où $m^k(g, g')$ est une mesure Aura en considérant le voisinage N_s^k .

L'énergie de Gibbs est ainsi décrite comme une combinaison linéaire des mesures Aura.

Dans le cas du modèle auto binomial, cette énergie devient :

$$U(x) = \sum_{k=1}^K \beta_k \sum_{g, g' \in \Lambda} g g' m^k(g, g') \quad (2.68)$$

Pour le modèle de Potts, cette énergie de Gibbs prend la forme suivante :

$$U(x) = \sum_{k=1}^K \beta_k \sum_{g, g' \in \Lambda} (2\delta_{gg'} - 1) m^k(g, g') \quad (2.69)$$

2.13 Mesure de la similarité de deux images à partir des GLAMs

Les matrices Aura de niveau de gris sont définies selon le système de voisinage N . Par la suite, nous emploierons les termes GLAM (Gray Level Aura Matrix) pour désigner une matrice Aura de niveau de gris en utilisant les systèmes de voisinage arbitraires (c.-à-d. symétrique ou asymétrique) [5][10]; SGLAM (Symmetric Gray Level Aura Matrix) pour désigner une matrice Aura de niveau de gris qui utilise un système de voisinage symétrique [17]; AGLAM (Asymmetric Gray Level Aura Matrix) pour décrire une matrice Aura de niveaux de gris avec un système de voisinage asymétrique [17]; et BGLAM (Basic Gray Level Aura Matrix) pour désigner une matrice Aura de niveau de gris de base [6][11][15].

Les GLAMs ont été particulièrement utilisées pour mesurer la similarité de deux images [5]. Cette similarité est clairement indiquée par le Lemme 1 et le théorème 1 proposés dans [6][17].

Lemme 1 : Soit I_1 une image définie sur S , s un site donné sur S , et N un système de voisinage. Supposant que l'image I_2 est obtenue à partir de l'image I_1 , en changeant la valeur de niveau de gris du pixel s de g_1 en g_2 tel que $g_2 \neq g_1$ (les

valeurs des niveaux de gris des autres sites demeurent inchangées). Soient A_1 et A_2 les matrices Aura de niveau de gris des images I_1 et I_2 , respectivement sur N , alors $A_1 \neq A_2$.

Théorème 1: Deux images de même taille sont identiques si et seulement si leurs GLAMs correspondantes sur tous les systèmes de voisinage possibles sont identiques.

Les démonstrations du Lemme 1 et du théorème 1 sont données en annexe B.

Le théorème 1 indique que deux images sont différentes s'il existe au moins un système de voisinage pour lequel les GLAMs de ces deux images sont différentes. Cependant, il est pratiquement impossible de tester tous les systèmes de voisinage car leur nombre varie d'une manière exponentielle avec la taille de l'élément structurant, soit $(2^{|N_s|} - 1) = \sum_{i=1}^{|N_s|} C_{|N_s|}^i$ [6].

2.14 Matrices Aura de niveaux de gris symétriques (SGLAMs)

Lorsque le système de voisinage est symétrique, la GLAM devient une SGLAM. Cependant, l'utilisation des SGLAMs pour différencier des textures s'est avérée peu efficace [6]. En effet, deux images avec différentes textures peuvent avoir la même SGLAM comme le montre la figure (2.7). L'image à droite est obtenue par la rotation de l'image gauche de 90 degrés. Ces deux images de taille 128x128 sont binaires et le système de voisinage 4-connexes est employé pour calculer la SGLAM.

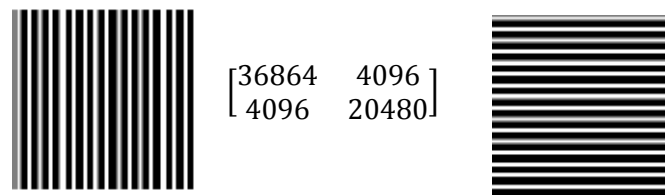


Fig. 2.7 : Exemple de l'inefficacité de SGLAMs pour la différenciation des textures.

Pour éviter le problème de la non invariance à la rotation des SGLAMs, Qin et Yang ont proposés soit de combiner les SGLAMs avec la transformée en ondelettes [5], soit d'utiliser des BGLAMs [6] [15].

2.15 Matrices Aura de niveaux de gris de base

Introduites par Qin et Yang dans le cadre de la synthèse des textures [6] et de la classification des images texturées [15], les matrices Aura de niveaux de gris de base (BGLAMs) se sont avérées comme un outil très performant [6] [19]. Une matrice Aura de niveaux de gris de base est une matrice Aura de niveaux de gris calculée à partir d'un système de voisinage comportant un seul site voisin. Par conséquent, elle correspond tout simplement à une matrice de cooccurrence. Cependant, en pratique, on considère toujours un ensemble de BGLAMs correspondants à des sites voisins appartenant à un élément structurant spécifique. Un exemple de calcul d'un ensemble de matrices Aura de base est illustré sur la figure (2.8).

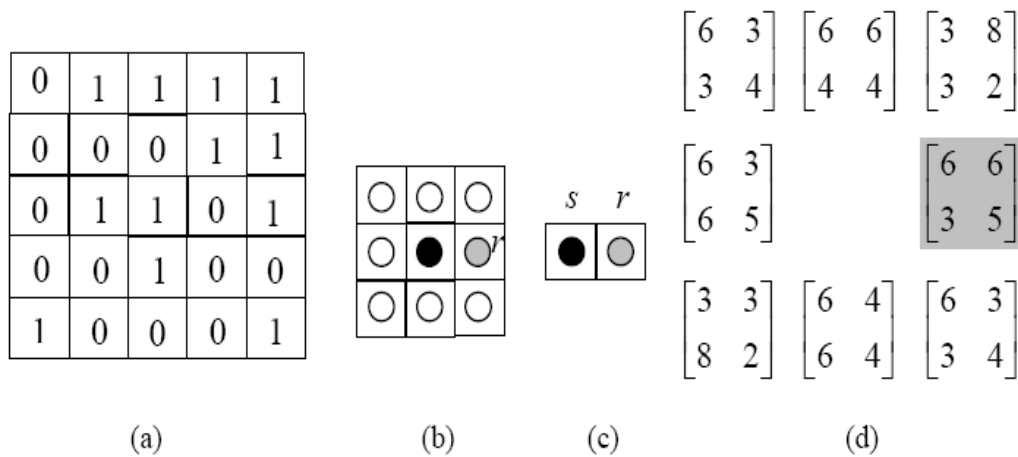


Fig. 2.8 : Calcul des BGLAMs d'une image binaire de taille 5×5. (a) Image binaire, (b) Élément structurant, (c) Configuration de déplacement du Pixel voisin r , (d) Les BGLAMs correspondantes à huit configurations de déplacement.

Deux théorèmes ont été proposés dans [6] pour montrer la capacité des BGLAMs à fournir toute l'information nécessaire et suffisante pour différencier deux images.

Théorème 2: Une GLAM peut être représentée comme la somme de BGLAMs.

On peut en effet vérifier sur l'exemple de la figure (2.8) que la somme des BGLAMs est égale à la GLAM $\begin{bmatrix} 42 & 36 \\ 36 & 30 \end{bmatrix}$ obtenue avec l'élément structurant de la figure (2.8 (b)).

Théorème 3: Deux images de même taille sont identiques si et seulement si leurs BGLAMs correspondantes sont identiques [15].

Les preuves de ces deux précédents théorèmes sont données en annexe B.

Le théorème 3 stipule qu'on peut représenter une image uniquement par ses BGLAMs. Pour un élément structurant N_s , le nombre de BGLAMs est $(|N_s| - 1)$. Par conséquent, l'utilisation des BGLAMs pour différencier deux images est plus avantageuse que celle des GLAMs.

2.16 Mesure de similarité d'images à partir de leurs BGLAM

Soient deux images X et Y définies sur S . Soit $A(X) = \{A_i | 0 \leq i \leq m\}$ et $A(Y) = \{B_i | 0 \leq i \leq m\}$ leurs BGLAMs normalisées correspondantes, m étant le nombre de sites de l'élément structurant ($m = |N_s|$). La similarité entre les images X et Y est donnée par la distance de leurs BGLAMs [6][11]:

$$d(X, Y) = d(A(X), A(Y)) = \sum_{i=0}^m \|A_i - B_i\| \quad (2.70)$$

Notons qu'une matrice aura $A = [a_{ij}]_{0 \leq i, j \leq G}$ est normalisée si $\sum_{i, j=0}^G a_{ij} = 1$.

$\|A\| = \sum_{i, j=0}^G |a_{ij}|$ est la norme de la matrice A .

Cette propriété garantit que plus la valeur de la distance est petite plus les deux images de texture sont semblables.

2.17 Conclusion

Dans ce chapitre nous avons présenté d'une manière assez exhaustive des notions de base sur les ensembles Aura, les mesures Aura et les matrices Aura. Le concept Aura, est axé principalement sur la théorie des ensembles, ainsi l'ensemble Aura peut être interprété comme la frontière entre deux ensembles, une mesure Aura peut être assimilée à la longueur de la frontière de deux ensembles et la matrice Aura ou de miscibilité est une matrice qui englobe les mesures Aura des différents ensembles pris deux à deux. La généralisation de ce concept dans le cadre d'une image nous a permis de définir les matrices Aura de niveau de gris et d'établir d'une part ses liens avec des théories aussi variées que sont les champs aléatoires de Gibbs, les matrices de cooccurrences, les matrices de covariance et la dilatation morphologique ; et de fournir un moyen efficace afin de mesurer la similarité entre deux images d'autre part. Différentes méthodes de calcul des matrices Aura seront abordées dans le prochain chapitre en vue de la segmentation d'images texturées.

Chapitre 3

Segmentation d'images à base des matrices Aura

3.1 Introduction

Nous avons vu dans le premier chapitre que la segmentation d'images texturées peut être réalisée en deux étapes. Dans la première étape, on cherche à caractériser chaque pixel de l'image par un ensemble d'attributs obtenus par une méthode d'analyse local de texture. La deuxième étape consiste à regrouper les pixels ayant des attributs similaires en classes en utilisant une des méthodes de classification non supervisées. Dans cette thèse, nous poursuivons la même démarche en déterminant les attributs de texture de chaque pixel à partir des matrices Aura des niveaux de gris (GLAMs) dans la première étape et les méthodes de classification des données à l'instar des algorithmes K-means ou Fuzzy C-means que nous avons déjà présenté dans le 1^{er} chapitre lors de la deuxième étape. On s'intéressera, ainsi, dans ce chapitre qu' à la première étape. Cependant, nous verrons que l'utilisation de la méthode de caractérisation locale de texture basée sur la GLAM nécessite un temps de calcul assez long. Pour remédier à ce problème, nous présenterons deux techniques pour la réduction du temps de calcul des attributs de texture et la réduction de l'espace mémoire. La première est la technique des listes chaînée et la deuxième dite structure hybride qui combine à la fois la notion des listes doublement chaînées et les tables de hachage. Le choix des éléments structurants pour le calcul des matrices Aura sera également abordé à la fin de ce chapitre.

3.2 Méthode classique de calcul des matrices Aura d'une image

L'Aura de A par rapport à B détermine comment un sous-ensemble B est lié au voisinage d'un sous-ensemble A . La matrice Aura de niveau de gris d'une image mesure la quantité de mixage entre deux sous-ensembles de sites ayant des niveaux de gris différents.

Soit I une image donnée sur une grille S avec un système de voisinage $N = \{N_s, s \in S\}$. Supposant qu'un espace d'état commun $\Lambda = \{0, 1, \dots, G - 1\}$ est utilisé pour tous les sites $s \in S$ et que la même structure de voisinage est utilisée pour tous les sites $s \in S$, c.-à-d. $\forall s \in S$, le voisinage N_s sera la translation d'un voisinage de base E [9]. Une manière directe de calculer la GLAM est de déterminer d'abord tous les ensembles de niveau de gris $\{S_g | g \in \Lambda\}$ avec $S_g = \{s \in S | x_s = g\}$. Ensuite, on consulte chaque paire d'ensembles S_g et $S_{g'}$, où g et $g' \in \Lambda$ pour calculer la mesure Aura $m(S_g, S_{g'})$ en utilisant l'équation (2.3) vue au chapitre précédent. Cette méthode de calcul de $m(S_g, S_{g'})$ possède

une complexité algorithmique égale à $O(|E||S_g||S_{g'}|)$. Sachant que $\sum_{g=0}^{G-1} |S_g| = |S|$, la complexité algorithmique pour calculer toute la matrice Aura A est égale à $O(|E| * |S|^2)$. Si l'élément structurant est symétrique, cette complexité peut être divisée par 2. En se basant sur la propriété de la GLAM donnée en chapitre 2, $\sum_{j=0}^{G-1} m(S_g, S_{g'}) = |S_g||N|$ et sachant que $N_s = \bigcup_{g \in \Lambda} (N_s \cap S_g)$, il est possible de calculer une matrice Aura de la même manière qu'une matrice de co-occurrence. L'algorithme de calcul de la matrice Aura de niveau de gris peut être alors donné comme suit :

- 1- Initialisez chaque élément de la GLAM, dénotée par $A = [m(S_g, S_{g'})]$, à zéro, c.-à-d. $m(S_g, S_{g'}) = 0$ pour $0 \leq g, g' \leq G - 1$.
- 2- Pour chaque site s de niveau de gris g , on prélève le niveau de gris g' de chaque site r situé dans le voisinage N_s du site s . Incrémenter la valeur de $m(S_g, S_{g'})$ par 1.
- 3- L'algorithme s'arrête quand tous les sites $s \in S$ ont été traités.

Algorithme 1 : Calcul classique d'une GLAM

Cette méthode est plus rapide que la précédente puisque sa complexité est réduite à $O(|E||S|)$.

D'après la propriété 3 de la matrice Aura (voir chapitre 2), si le système de voisinage est symétrique, alors, la matrice Aura de niveau de gris est aussi symétrique (Fig. 3.1). Si par contre le système de voisinage est asymétrique, alors la matrice Aura de niveau de gris sera asymétrique (Fig. 3.2).

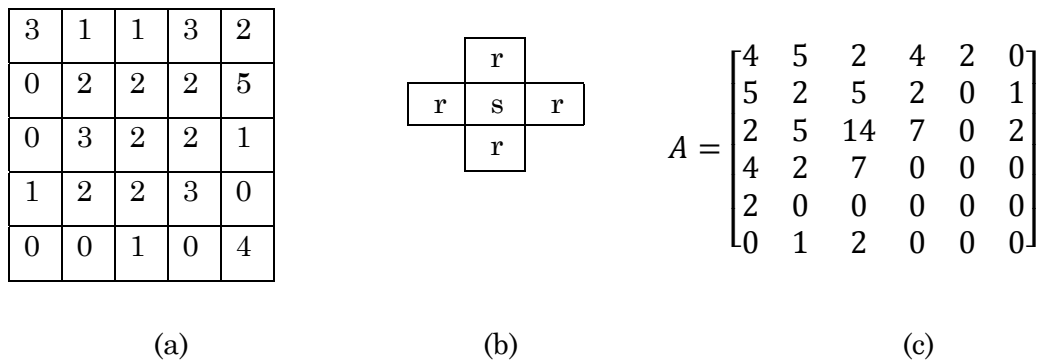


Fig. 3.1 : Exemple de calcul d'une matrice Aura de niveau de gris avec un système de voisinage symétrique. (a) Image de taille 5*5 avec 6 niveaux de gris (0-5), (b) Élément structurant symétrique et (c) SGLAM correspondante.

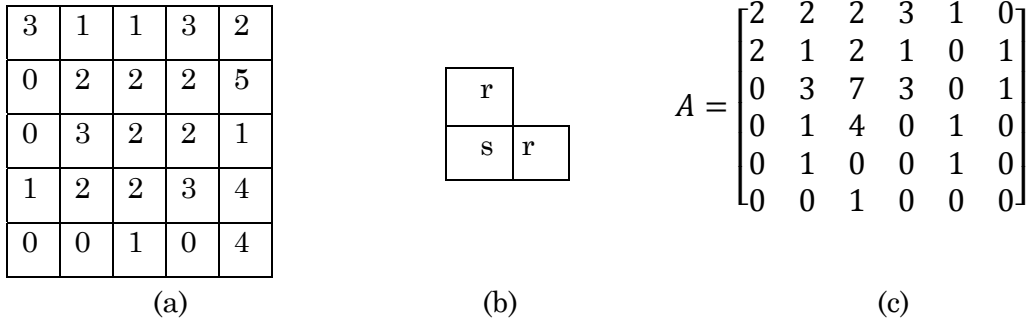


Fig. 3.2 : Exemple de calcul d'une matrice Aura de niveau de gris avec un élément structurant asymétrique. (a) Image de taille 5*5 avec 6 niveaux de gris (0-5), (b) Élément structurant asymétrique et (c) AGLAM correspondante.

3.3 Extraction des attributs locaux de texture basée sur des matrices Aura

En segmentation, il est nécessaire de caractériser chaque site par un ensemble d'attributs de texture extraits à partir des matrices Aura des niveaux de gris. Pour cela, nous commençons par scruter chaque site s , $s = 1, 2, \dots, |S|$. Pour chaque site, une matrice Aura est ensuite construite à partir de tous les sites situés dans une fenêtre de voisinage de taille $(2w + 1)^2$. L'élément structurant E correspondant au système de voisinage N choisi est déplacé de façon à ce que son centre passe par tous les sites de la fenêtre de voisinage à analyser. La matrice Aura est alors déterminée en utilisant la même procédure décrite précédemment. Cette matrice de taille $(G \times G)$ est finalement parcourue une ou deux fois selon les attributs à calculer. Cette procédure classique sera appelée *Grey Level Aura Matrice* (GLAM), elle peut engendrer des temps de calcul prohibitifs surtout lorsque G est élevé. En effet, le nombre d'opérations nécessaires pour construire une GLAM pour chaque site s est estimé à $(2w + 1)^2|E|$ et le calcul des attributs de texture à partir d'une GLAM est approximativement égal à G^2 (si on suppose que chaque attribut nécessite un seul parcourt de la GLAM). Par conséquent, la complexité algorithmique global de cette méthode est évaluée à $O[|S|((2w + 1)^2|E| + |G|^2)]$. Il est donc nécessaire de réduire G pour réduire la complexité algorithmique de cette méthode. L'augmentation de la taille de la fenêtre de voisinage ou de l'élément structurant à peu d'effet. Pour réduire ces temps, nous nous sommes inspirés des techniques utilisées dans le cas des matrices de co-occurrence.

Une façon simple de réduire la taille de la matrice de co-occurrence consiste à réduire, au préalable, le nombre de niveaux de gris [20].

Une autre solution consiste à réduire le nombre de sites sur lesquels les calculs sont effectués [21].

Certains auteurs préfèrent découper l'image en un ensemble de fenêtres non chevauchantes [22]. Dans chaque fenêtre, une matrice de co-occurrence à partir de laquelle on extrait les attributs de texture est déterminée. Ainsi, chaque centre de la fenêtre sera caractérisé par un vecteur d'attributs. Les sites se trouvant dans une même fenêtre héritent, automatiquement, des attributs de son centre. Cette méthode peut, cependant, fausser les résultats de la segmentation car les sites situés dans une même fenêtre peuvent provenir de textures différentes entraînant, ainsi, un effet mosaïque notamment lors d'une segmentation.

Un algorithme récursif capable de réduire le temps de calcul des attributs de texture à partir d'une matrice de co-occurrence a été également proposé par Argenti *et al.* [23]. L'idée de cette technique provient du fait que les fenêtres de voisinages utilisées pour calculer les attributs de texture pour deux sites successifs voisins situés sur une même ligne, se chevauchent. Par conséquent, la majorité, des sites intervenant dans le calcul des matrices de co-occurrence de deux sites adjacents sont communs. Pour éviter de scruter inutilement, à chaque fois, ces sites, Argenti *et al.* proposent de déterminer les attributs correspondant à un site courant en fonction des valeurs des attributs du site adjacent précédemment traité. A ces valeurs, on ajoute un terme qui prend en compte les nouvelles paires de sites n'ayant pas contribué au calcul des attributs du site précédent et on soustrait un deuxième terme, qui rend compte de la contribution des paires de sites qui ont contribué au calcul des attributs du site précédent mais pas pour celui du site courant. Cette méthode est très efficace lorsque la taille des fenêtres de voisinage est grande. Elle reste cependant difficilement exploitable dans le cas des matrices Aura car il faut prendre en compte non pas une paire de sites mais un ensemble de sites.

Dans une autre approche complètement différente, il a été constaté que les matrices de co-occurrence sont souvent creuses ou éparées, c'est-à-dire, elles contiennent beaucoup d'éléments nuls qui ne doivent pas être pris en compte dans le calcul des attributs de texture. L'idée serait alors de ne sauvegarder sous forme d'une liste que les éléments non nuls de la matrice. Ainsi, la taille de cette liste, généralement, inférieure à celle de la matrice, est parcourue plus rapidement. Deux techniques exploitant cette idée ont été proposées dans la littérature. La première est basée sur l'utilisation d'une technique de programmation dynamique faisant appel à la notion de pointeurs et, plus

précisément, au concept de listes doublement chaînées [24]. La seconde technique combine, à la fois, la notion de listes doublement chaînées et les tables de hachage [25].

Dans ce mémoire, nous avons opté pour cette dernière solution, afin d'accélérer les calculs de la procédure d'extraction des attributs locaux de texture à partir des GLAMs.

3.3.1 Calcul des matrices Aura à base des listes chaînées

Initialement proposée par Clausi et Jernigan dans le cadre des matrices de co-occurrence, cette technique, appelée *Grey Level Co-occurrence Linked List* (GLCLL), est basée sur l'utilisation de la notion de structure de listes chaînées [24]. Etant donné la similitude entre les matrices de co-occurrence et les matrices Aura, nous avons utilisé cette technique pour déterminer une matrice Aura. Par analogie, cette technique sera appelée *Grey Level Aura Linked List* (GLALL).

3.3.1.1 Listes chaînées

Une liste chaînée est une structure de données constituée d'un ensemble d'éléments appelés nœuds. Ces éléments sont organisés de façon linéaire facilitant l'ajout ou l'extraction d'un ou de plusieurs éléments. Ainsi, pour pouvoir atteindre le $i^{ème}$ élément d'une liste, il faut parcourir tous les éléments le précédant (de 1 à $i-1$). D'un point de vue informatique, le moyen le plus naturel pour rendre un élément capable de fournir un accès à l'élément suivant est certainement de doter chaque élément d'un pointeur sur l'élément suivant.

3.3.1.1.1 Listes chaînées simples

Une liste chaînée simple contient des informations caractéristiques de l'application (le niveau de gris d'un pixel d'une image par exemple), d'un pointeur (Next) vers un autre élément ou une marque de fin (Nul) s'il y a pas d'éléments successeur. La structure ci-dessous définit un type `ListeNode` qui possède trois membres de données, deux entiers `x` et `y` et un pointeur `Next` vers `ListeNode`.

```
Struct ListeNode
```

```
{
```

```
/*Variable x de type entier*/
```

```

int x ;
/*Variable y de type entier*/
int y ;
/*Variable de type pointeur*/
Struct ListNode *Next ;
};

```

La figure (3.3) montre une représentation simplifiée d'une liste chaînée simple :

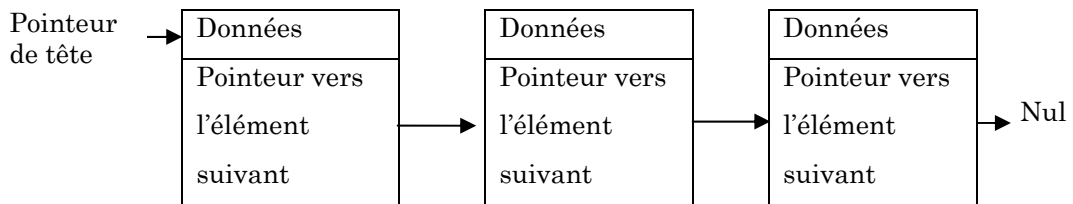


Fig. 3.3 : Représentation graphique d'une liste chaînée simple.

3.3.1.1.2 Listes chaînées doubles

Une liste chaînée double est basée sur le même principe que la liste chaînée simple, à la différence près qu'elle contient non seulement un pointeur vers le nœud suivant mais aussi un pointeur vers le nœud précédent lui permettant ainsi le parcours de la liste dans les deux sens. La structure ci-dessous définit un type ListNode qui possède quatre membres de données, deux entiers x et y et deux pointeurs Next, Prec qui pointent respectivement vers le nœud suivant et le nœud précédent dans la liste.

```

Struct ListNode
{
/*Variable x de type entier*/
int x ;
/*Variable y de type entier*/
int y ;
/*Variable de type pointeur*/
Struct ListNode *Next ;
/*Variable de type pointeur*/
Struct ListNode *Prec ;
};

```

Une représentation d'une liste chaînée double est donnée sur la figure (3.4).

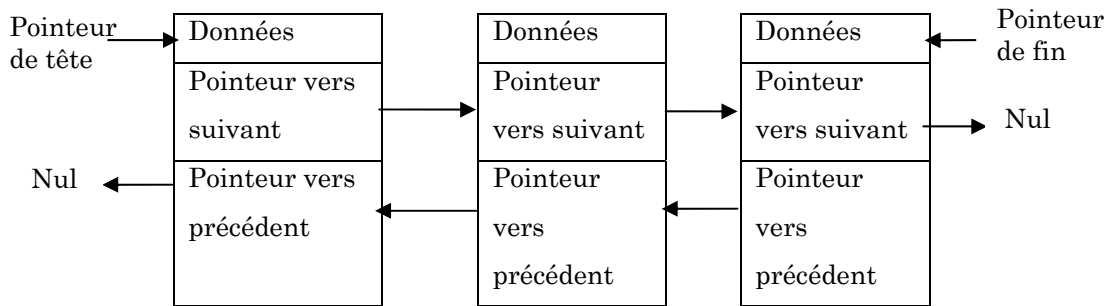


Fig. 3.4 : Représentation graphique d'une liste chaînée double.

Les listes chaînées ont l'avantage d'être souples car elle peuvent facilement s'allonger et se rétrécir. On n'a pas besoin de lui réserver à l'avance de la place mémoire. On demande de l'espace au fur et à mesure des besoins, contrairement aux tableaux dynamiques où un bloc mémoire doit être initialement alloué. De plus, l'ajout d'une donnée dans une liste ordonnée ne nécessite pas de déplacement des autres données de la liste. Cependant, l'accès à un élément ne peut se faire qu'en examinant séquentiellement les nœuds précédents.

3.3.1.2 Application des listes chaînées au calcul des matrices Aura

Au lieu d'utiliser une matrice pour ranger les mesures Aura $m(S_g, S_{g'})$ calculées pour deux ensembles de niveaux de gris S_g et $S_{g'}$, une structure de liste doublement chaînée peut être utilisée pour sauvegarder que les mesures Aura non nulles. En effet, une matrice Aura contient souvent plusieurs éléments nuls qui ne doivent pas être pris en compte dans le calcul des attributs de texture. Partant de cette constatation, il est nécessaire de sauvegarder que les éléments non nuls de la matrice Aura en faisant appel au concept des listes chaînées.

Chaque élément de la liste est composé des champs d'information, un champ noté x_1 contenant le niveau de gris g , un champ noté x_2 qui contient le niveau de gris g' , un compteur noté M contenant la mesure Aura $m(S_g, S_{g'})$ qui désigne le nombre de fois où un site de niveau de gris g possède dans son voisinage, défini par l'élément structurant E , un site ayant un niveau de gris g' et deux pointeurs Next et Prec (Fig. 3.5). Le premier pointant vers le nœud suivant et l'autre vers le nœud précédent. Pour chaque site à traiter, on parcourt les sites voisins situés dans une fenêtre de voisinage de taille $(2w + 1)^2$. Au lieu d'utiliser une matrice pour ranger les mesures Aura, une structure de

liste doublement chaînée est utilisée pour sauvegarder que les éléments de la matrice Aura ayant des valeurs non nulles de la manière suivante. En parcourant les sites s de la fenêtre de voisinage, on récupère le niveau de gris g du site s et les valeurs des niveaux de gris g' des sites voisins $r \in N_s$. Ensuite, on parcourt la liste chaînée pour chercher si un nœud contient chaque paire (g, g') . La recherche commence en tête de la liste en vérifiant la première instance de la valeur x_1 , si $x_1 = g$ on cherche la deuxième instance de valeur x_2 si $x_2 = g'$ alors la paire (g, g') est présente dans la liste et sa mesure Aura $m(S_g, S_{g'})$, rangée à l'intérieur de ce nœud, est incrémentée. Si le nœud de la paire (g, g') n'est pas trouvé à l'emplacement recherché, on consulte séquentiellement les autres nœuds. Si aucun nœud de la liste ne contient pas la paire (g, g') , alors un nouveau nœud est créé à la fin de cette liste où sera rangée la nouvelle paire (g, g') avec une mesure Aura $m(S_g, S_{g'}) = 1$. Une fois, l'ensemble des sites de la fenêtre de voisinage examiné, la liste résultante ne contiendra que les paires des valeurs des niveaux de gris ayant une mesure Aura non nulle. Le calcul des attributs de texture est ainsi effectué en parcourant la liste du début vers la fin. Dans le cas où la longueur « L » de la liste est courte, ce calcul devient rapide. La même procédure se répète pour tous les autres sites.

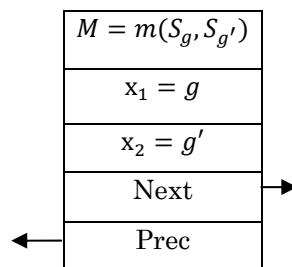


Fig. 3.5 : Représentation Graphique d'un élément de la liste.

Remarque

Si le système de voisinage est symétrique, on retient dans la liste le couple (g, g') si $g \geq g'$ et (g', g) sinon.

Pour illustrer le principe de calcul des éléments non nuls d'une matrice Aura et leurs sauvegarde dans une liste chaînée, on reprend l'exemple de la figure (3.1). La matrice Aura A contient 21 éléments non nuls qu'on peut représenter dans une liste à 12 nœuds. Pour construire la liste chaînée, on suit les étapes suivantes :

On parcourt la fenêtre de gauche à droite et de haut en bas. Le premier site de la fenêtre est de niveau de gris $g = 3$, ses voisins appartenant à l'élément structurant E sont les deux sites de niveaux de gris $g' = 1$ et $g' = 0$. Comme $g \geq g'$ et la liste est initialement vide, on crée alors les deux nœuds suivants :

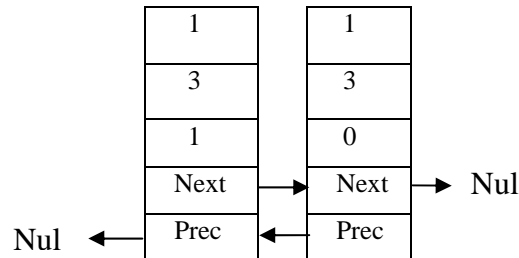


Fig. 3.6 : Liste chaînée après traitement du 1^{er} pixel .

Le deuxième site de la fenêtre est de niveau de gris $g = 1$ et ses voisins sont de niveau de gris $g' = 1$, $g' = 3$ et $g' = 2$. Les deux paires de niveau de gris $(1, 1)$ et $(1, 2)$ n'existent pas sur la liste chaînée alors deux nœuds sont alors créés. Et comme $1 < 2$ pour la paire $(1,2)$ on prend la paire $(2,1)$ et $1 < 3$ pour la paire $(1,3)$ on prend la paire $(3,1)$. Comme elle existe déjà la mesure Aura correspondante est incrémentée $m(S_g, S_{g'})=2$.

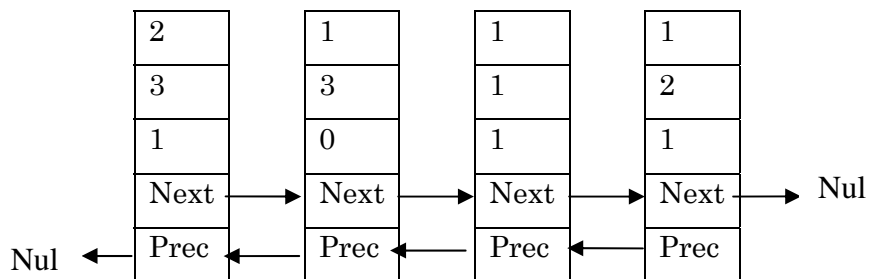
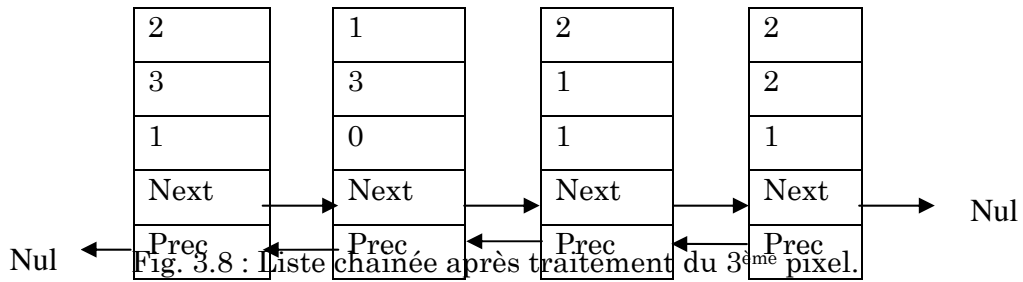
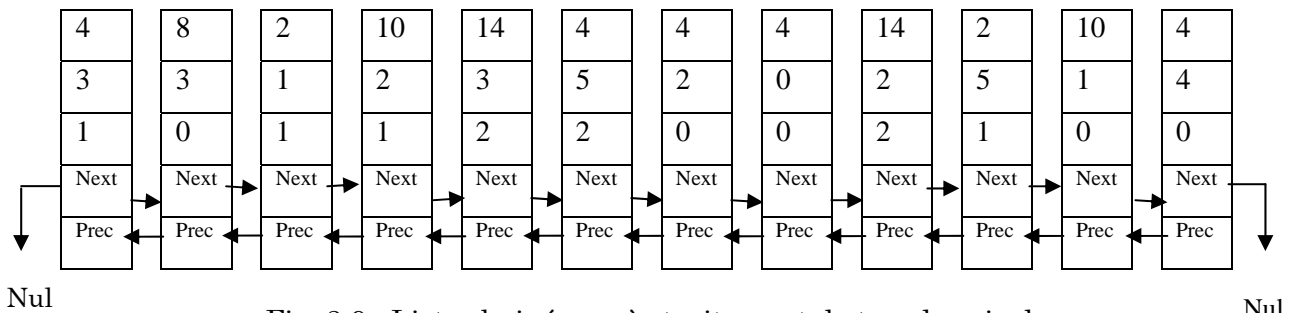


Fig. 3.7 : Liste chaînée après traitement du 2^{ème} pixel .

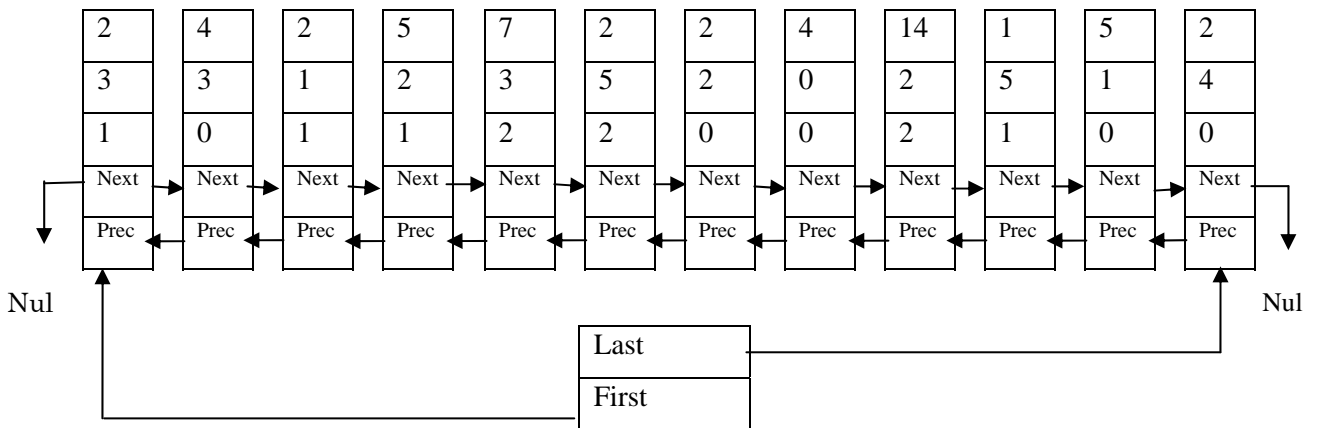
On continue à parcourir la fenêtre pour récupérer le niveau de gris $g' = 1$ du troisième site ainsi que les niveaux de gris de ses voisins $g' = 1$, $g' = 2$ et $g' = 3$. Le parcourt de la liste précédente, montre que les paires $(1,1)$, $(1,2)$ et $(1,3)$ existent déjà, donc la mesure Aura sera incrémentée dans les trois nœuds correspondants.



En procédant de la même manière pour les autres sites de la fenêtre on aboutit à la liste chaînée suivante qui regroupe les 12 éléments non nuls de la matrice Aura A de la figure (3.1(c)).



A la fin on divise sur 2 les mesures Aura dont les niveaux de gris $i \neq j$.



3.3.1.3 Attributs de texture calculés à partir de la liste chaînée

Pour calculer les attributs de texture, on parcourt la liste de début vers la fin. Si la longueur de la liste L est petite, alors le calcul devient rapide. Si sa longueur est grande, le calcul devient lent. Quelques attributs extraits à partir d'une liste chaînée sont calculés de la manière suivante :

1) Second moment angulaire

$$ASM = \sum_{\substack{k=1 \\ x_1=x_2}}^L \left(\frac{P_k \rightarrow M}{N_L} \right)^2 + \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L 2 * \left(\frac{P_k \rightarrow M}{N_L} \right)^2 \quad (3.1)$$

2) Contraste

$$CON = \sum_{\substack{k=1 \\ x_1=x_2}}^L (P_k \rightarrow x_1 - P_k \rightarrow x_2)^2 \left(\frac{P_k \rightarrow M}{N_L} \right)^2 + \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L 2 * (P_k \rightarrow x_1 - P_k \rightarrow x_2)^2 \left(\frac{P_k \rightarrow M}{N_L} \right)^2 \quad (3.2)$$

3) Entropie

$$ENT = - \sum_{\substack{k=1 \\ x_1=x_2}}^L \left(\frac{P_k \rightarrow M}{N_L} \right) \log \left(\frac{P_k \rightarrow M}{N_L} \right) - \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L 2 * \left(\frac{P_k \rightarrow M}{N_L} \right) \log \left(\frac{P_k \rightarrow M}{N_L} \right) \quad (3.3)$$

4) Moment de différence inverse

$$IDM = \sum_{\substack{k=1 \\ x_1=x_2}}^L \frac{1}{1 + (P_k \rightarrow x_1 - P_k \rightarrow x_2)^2} \left(\frac{P_k \rightarrow M}{N_L} \right) + \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L 2 * \left[\frac{1}{1 + (P_k \rightarrow x_1 - P_k \rightarrow x_2)^2} \left(\frac{P_k \rightarrow M}{N_L} \right) \right] \quad (3.4)$$

5) Variance

$$VAR = \sum_{\substack{k=1 \\ x_1=x_2}}^L (P_k \rightarrow x_1 - P_k \rightarrow x_2)^2 \left(\frac{P_k \rightarrow M}{N_L} \right) + \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L 2 * (P_k \rightarrow x_1 - P_k \rightarrow x_2)^2 \left(\frac{P_k \rightarrow M}{N_L} \right) \quad (3.5)$$

6) Corrélation

$$COR = \sum_{\substack{k=1 \\ x_1=x_2}}^L (P_k \rightarrow x_1 - \mu_{x_1})(P_k \rightarrow x_2 - \mu_{x_2}) \left(\frac{P_k \rightarrow M}{N_L} \right) + \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L (P_k \rightarrow x_1 - \mu_{x_1})(P_k \rightarrow x_2 - \mu_{x_2}) \left(\frac{P_k \rightarrow M}{N_L} \right) \quad (3.6)$$

avec :

$$\mu_{x_1} = \sum_{\substack{k=1 \\ x_1=x_2}}^L (P_k \rightarrow x_1) \left(\frac{P_k \rightarrow M}{N_L} \right) + \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L (P_k \rightarrow x_2) \left(\frac{P_k \rightarrow M}{N_L} \right) \quad (3.7)$$

$$\mu_{x_2} = \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L (P_k \rightarrow x_2) \left(\frac{P_k \rightarrow M}{N_L} \right) + \sum_{\substack{k=1 \\ x_1=x_2}}^L (P_k \rightarrow x_1) \left(\frac{P_k \rightarrow M}{N_L} \right) \quad (3.8)$$

P_k est un pointeur vers le nœud de la liste à la position k , avec $k = 1, \dots, L$.

L Longueur de la liste après construction.

N_L Nombre total des mesures Aura de niveaux de gris.

$$N_L = \sum_{\substack{k=1 \\ x_1=x_2}}^L (P_k \rightarrow M) + \sum_{\substack{k=1 \\ x_1 \neq x_2}}^L 2 * (P_k \rightarrow M) \quad (3.9)$$

En évaluant approximativement le nombre d'opérations nécessaires pour déterminer une liste chaînée pour un site s à $(2w + 1)^2|E|L$ (cas extrême) et le nombre d'opérations pour déterminer les attributs de textures à L , la complexité algorithmique totale de la méthode GLALL est alors $O[|S|((2w + 1)^2|E|L + L)]$. Comme $(2w + 1)^2|E|L \gg L$, la complexité algorithmique devient $O[|S|(2w + 1)^2|E|L]$. L'augmentation de w ou de $|E|$ fait croire L car le nombre de pairs de niveaux de gris distinct augmente. D'un autre côté $L \ll G^2$ et que si G diminue L diminue aussi. Par conséquent, la GLALL devrait rester plus rapide que la GLAM. Cependant, en pratique si la longueur L de la liste est grande, la GLALL peut devenir plus lente que la GLAM car la recherche d'un couple de niveaux de gris dans la liste peut prendre un temps de calcul excessif.

3.3.2 Calcul des matrices Aura à base d'une structure hybride

L'utilisation des listes chaînées présente, toutefois, l'inconvénient de consommer beaucoup du temps lors de la recherche du couple de valeurs de niveaux de gris (g, g') , dans la liste. En effet, si ce couple est positionné dans un nœud situé à la fin de la liste, la liste est parcourue, séquentiellement, du début à la fin. Ceci induit un temps de calcul élevé, surtout, si la longueur de la liste est grande. Pour remédier à ce problème, Clausi et Zhao [25] ont suggéré l'utilisation d'une structure de données hybride, composée d'une liste doublement chaînée ne permettant que la sauvegarde des co-occurrences des niveaux de gris non nuls et une table de hachage pour permettre un accès direct à cette liste lors de la recherche d'un couple de valeurs de niveaux de gris [25]. Cette structure de données est appelé structure hybride de co-occurrence des niveaux de gris ou *Gray Level Co-occurrence Hybrid Structure* (GLCHS).

Une table de hachage (*hashtable* en anglais) est une technique algorithmique utilisée pour stocker ou retrouver efficacement des données en mémoire [26]. La table de hachage que nous avons utilisée est une matrice à deux dimensions où chaque élément de cette matrice contient un pointeur vers un nœud d'une liste doublement chaînée identique à celle décrite précédemment. Elle permet un accès direct à tous les éléments

de la liste. Cet accès est rapide, puisque aucun calcul n'est exigé. Chaque nœud de la liste doublement chaînée est une structure dans laquelle sont stockées les deux niveaux de gris (g, g') , leur mesure Aura $m(S_g, S_{g'})$ et les deux pointeurs Next et Prev qui permettent le passage d'un nœud vers un autre nœud voisin dans les deux sens. Pour chaque site s à traiter de niveau de gris g , on récupère les valeurs des niveaux de gris g' des sites voisins r situés dans le voisinage N_s . Pour chaque paire des niveaux de gris (g, g') , on consulte l'élément (g, g') de la table de hachage. Si sa valeur est nulle alors, le nœud (g, g') n'existe pas dans la liste et sera, par conséquent, créé à la fin de la liste en sauvegardant la nouvelle paire (g, g') et en mettant sa valeur de la mesure Aura $m(S_g, S_{g'})$ à 1. Afin de retrouver, ultérieurement et facilement ce nœud, son adresse mémoire est mentionnée à la position (g, g') de la table de hache. Sinon, le nœud (g, g') existe dans la liste, l'élément (g, g') de la table hache, qui n'est pas nulle, fournit alors l'adresse du nœud correspondant dans la liste afin d'y accéder pour incrémenter la valeur de sa mesure Aura. La structure hybride correspondant à l'exemple précédent (Fig. 3.1) est illustrée sur la figure (3.10). Cette technique est désignée sous l'appellation de *Gray Level Aura Hybrid Structure* (GLAHS).

Le calcul des attributs de textures se fait de la même façon que ceux calculés par la méthode GLALL, c'est-à-dire, en parcourant la liste chaînée pour récupérer les valeurs des niveaux de gris (g, g') et leurs mesures Aura $m(S_g, S_{g'})$. La complexité algorithmique global de la méthode GLAHS est $O[|S|(2w + 1)^2|E|]$. Par conséquent cette méthode est plus rapide que la GLAM et la GLALL.

Le tableau (3.1) donne un résumé sur le nombre d'opérations d'extraction des attributs de texture pour un pixel à partir des 3 méthodes GLAM, GLALL et GLAHS.

Méthode	Détermination des mesures Aura	Calcul des attributs
GLAM	$O[S ((2w + 1)^2 E + G ^2)]$	G^2
GLALL	$O[S (2w + 1)^2 E L]$	L
GLAHS	$O[S (2w + 1)^2 E]$	L

Tableau 3.1 : Complexité algorithmique global des trois méthodes proposées

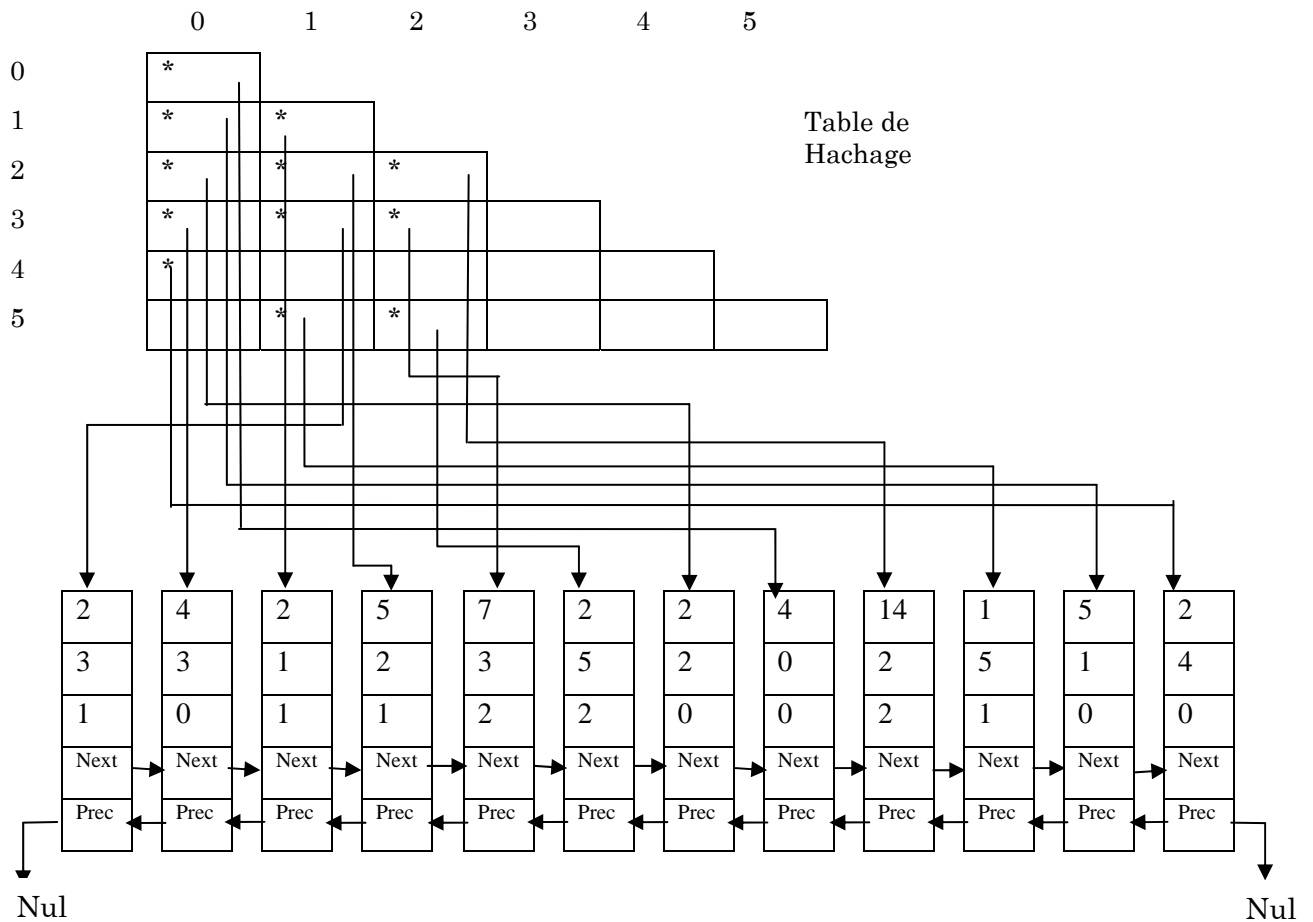


Fig. 3.11 : Calcul d'une matrice Aura de niveau de gris correspondante à l'image de la figure 3.5 en utilisant une structure hybride.

3.4 Choix d'une structure de voisinage

Une matrice Aura est définie en fonction du système de voisinage N , qui peut être défini par un élément structurant $E = N_s$. Dans notre cas, on a considéré des éléments structurants E de formes géométriques et de tailles différentes. La figure (3.12) montre quelques éléments structurants symétriques ayant des formes classiques (carré, losange, croix, cercle, X, ...). Quelques exemples d'éléments structurants asymétriques sont représentés sur la figure (3.13). Plusieurs stratégies peuvent être employées pour utiliser un ou plusieurs éléments structurants.

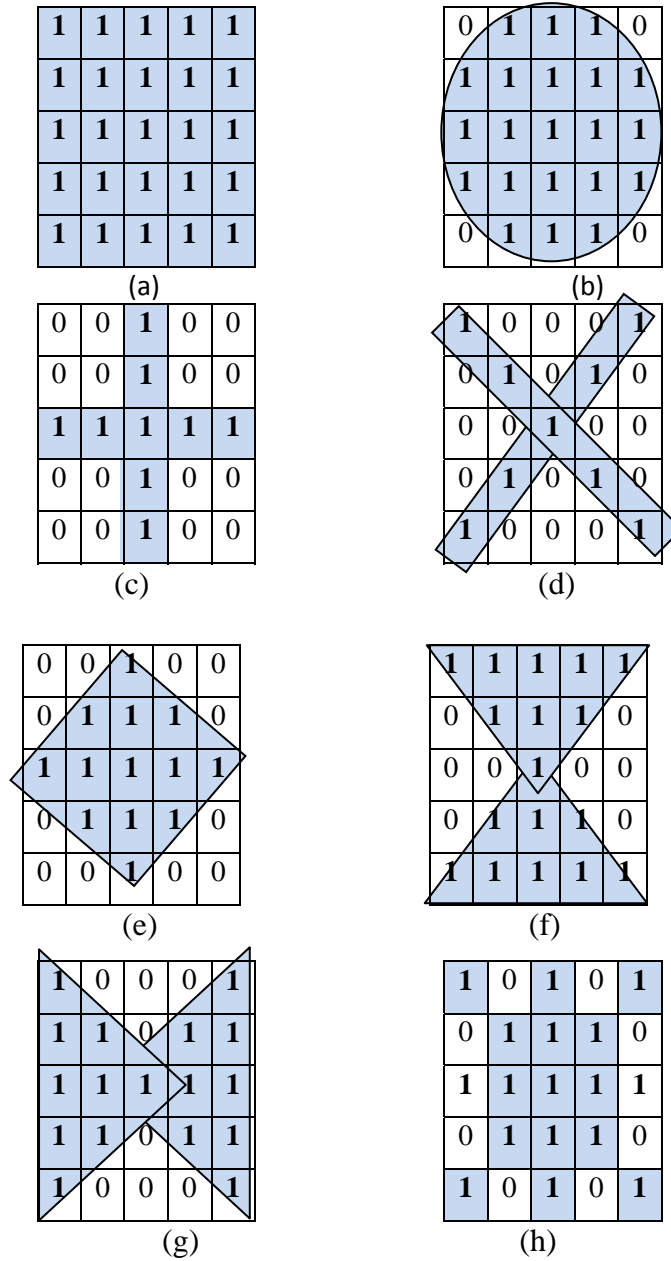


Fig. 3.12 : Eléments structurants symétriques de taille 5*5.
 (a) Carré, (b) Cercle, (c) Croix, (d) Élément de forme X, (e) Losange,
 (f) Sablier, (g) Sablier horizontal, (h) Quelconque.

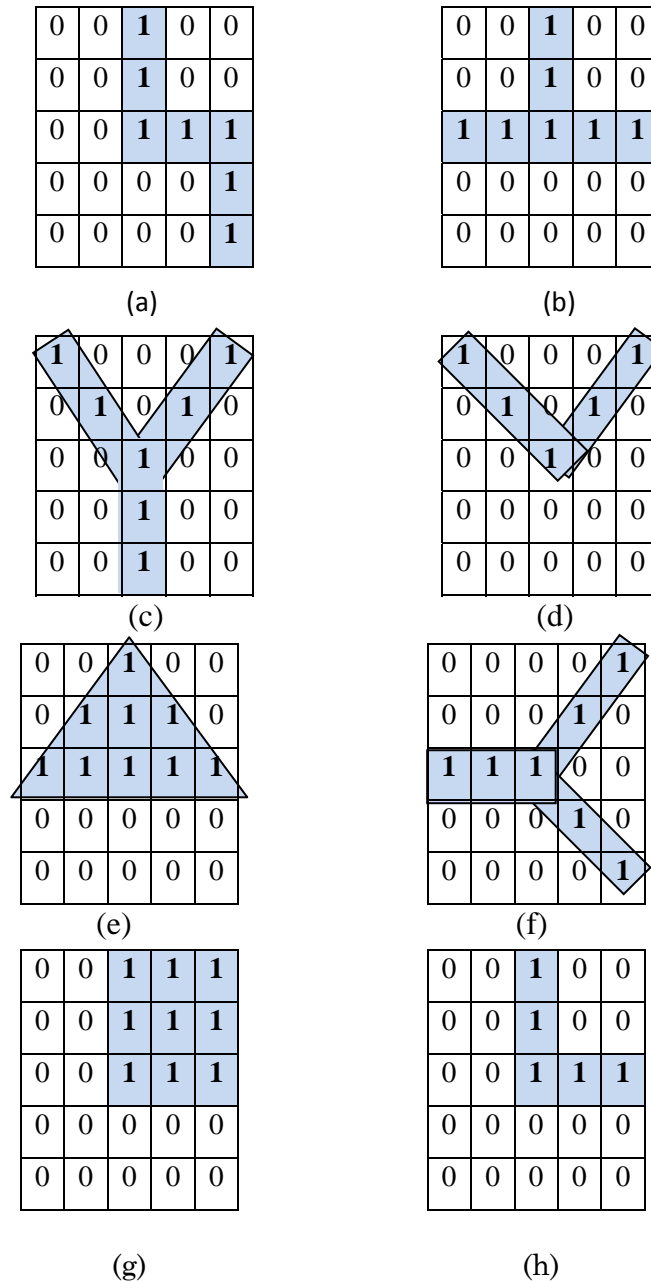


Fig. 3.13 : Eléments structurants asymétriques de taille 5*5

3.4.1 Utilisation d'une seule structure de voisinage

Dans ce cas, un seul élément structurant est choisi parmi les 5 premiers éléments symétriques de formes géométriques (carré, losange, croix, cercle et X) de la figure (3.12) où les 5 premiers éléments asymétriques de la figure (3.13). Le nombre d'attributs est alors égal à 6. La taille de l'élément structurant est choisie à priori par l'utilisateur.

3.4.2 Utilisation de plusieurs structures de voisinage de même forme et de tailles variables.

Les éléments structurants utilisés dans les méthodes précédentes ont une taille fixe qui doit être choisie aussi à priori. Sans aucune connaissance sur les textures présentes dans l'image, le choix de cette taille sera délicat. De plus, les images à segmenter peuvent avoir plusieurs textures ayant des structures de tailles différentes. Afin de pouvoir capturer toute l'information texturale, nous proposons d'utiliser en même temps un élément structurant avec des tailles variables. Autant de matrices Aura ou de structures hybrides que de tailles sont alors utilisées. La figure (3.14) montre quelques éléments structurants avec des tailles variables. Pour 5 tailles différentes (3x3), (5x5), (7x7) et (9x9), le nombre d'attributs de texture sera alors égal à $5 \times 6 = 30$. Cette technique peut s'apparenter à une analyse multi résolution.

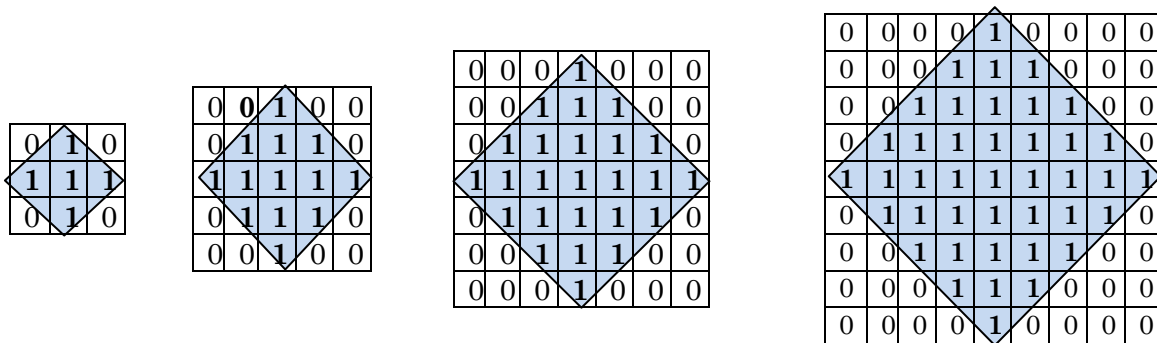


Fig. 3.14 : Eléments structurants de taille différentes
(a) 3x3, (b) 5x5, (c) 7x7, (d) 9x9.

3.4.3 Utilisation des matrices Aura de base

Les matrices Aura de niveaux de gris de base (BGLAM) se sont avérés très efficaces en classification des images textures texturées [15][19][14]. En choisissant un seul élément structurant carré de taille (3x3) (Fig. 3.12(a)), on peut soustraire 8 éléments structurants contenant chacun un seul élément. On pourra ainsi extraire 8 BGLAMs chacune de taille ($G \times G$) et prendra chaque élément de la BGLAM comme attribut [14]. Le nombre d'attributs sera malheureusement très élevé même si G est faible. On peut alors envisager d'utiliser une analyse en composantes principales (ACP) [29] afin de réduire le nombre d'attributs. En ce qui nous concerne, nous proposons d'adapter les méthodes des listes chaînées GLALL et celle de la structure hybrides GLAHS pour déterminer des attributs de texture à partir des BGLAM. Ces deux techniques seront respectivement appelées *Base Grey Level Aura Linked List* (BGLALL) et *Base Grey Level*

Aura Hybrid Structure (BGLAHS). 8 listes chaînées seront ainsi créées et le nombre d'attributs sera égal à $8 \times 6 = 48$.

3.5 Conclusion

Nous avons présenté dans ce chapitre deux méthodes rapides pour le calcul de la matrice Aura de niveau de gris (GLAM) dans le cadre de la segmentation d'images. La première appelée GLALL est basée sur une liste doublement chaînée et la deuxième dite GLAHS est basée sur la combinaison entre la liste doublement chaînée et la table de hachage. Dans les deux cas, la liste chaînée permet de sauvegarder que les éléments non nuls de la GLAM. Cependant, dans le cas de la GLALL, cette liste doit être parcourue à chaque nouvelle mise à jour, ce qui provoque un temps de recherche plus important. La GLAHS est venue remédier à cet inconvénient en proposant un accès direct à la liste. Ces deux méthodes nous permettent d'envisager l'utilisation de plusieurs éléments structurants en même temps. Plusieurs stratégies pour combiner plusieurs éléments structurants de formes et de tailles variables ont été également proposées afin de segmenter une image texturée. Ces deux techniques peuvent également servir pour extraire rapidement des attributs de textures à partir des matrices Aura de niveaux de gris de base (BGLAM). Celles-ci ont été nommées respectivement BGLALL et BGLAHS. Les performances de ces stratégies seront évaluées dans le chapitre suivant.

Chapitre 4

Tests et résultats

4.1 Introduction

Nous allons présenter dans ce chapitre les résultats de la segmentation d'images texturées basée sur les matrices Aura que nous avons proposée dans le chapitre précédent.

Rappelons que pour segmenter une image, nous avons déterminé un ensemble d'attributs pour chaque pixel. Ensuite, nous avons appliqué la méthode de classification FCM ou K-means afin de regrouper les pixels en zones homogènes. Les attributs de texture de chaque pixel sont extraits à partir des matrices Aura des niveaux de gris des pixels situés à l'intérieure d'une fenêtre de voisinage de taille $(2w + 1)^2$, centrée sur le pixel en question. Le calcul de la matrice Aura dépend de la forme, du type (symétrique ou non) et de la taille des éléments structurants E . Plusieurs stratégies ont été ainsi proposées pour utiliser un ou plusieurs éléments structurants à savoir :

- Un seul élément structurant de forme et de taille fixe.
- Plusieurs éléments structurants à la fois de même forme mais de tailles variables (analyse multi-résolution).
- Des matrices Aura de niveaux de gris de base.

De plus, parallèlement à la techniques classique de programmation du calcul des matrices Aura de niveaux de gris, que nous avons nommé (GLAM), nous avons proposé deux techniques de programmation, l'une est basée sur les listes chaînées (GLALL), l'autre à structure hybride qui combine la notion des listes chaînées et des tables de hachage (GLAHS).

Par conséquent, nous allons analyser dans ce chapitre les performances de ces différentes stratégies et techniques de programmation en fonction de la largeur de la fenêtre de voisinage $(2w + 1)$, du type, de la forme et de la taille de l'élément structurant E . Les résultats de ces méthodes seront également confrontés à ceux obtenus avec les matrices de cooccurrence.

Notons que toutes les méthodes présentées dans notre travail ont été implémentées sous l'environnement Borland C++ version 6 sur un micro ordinateur Acer, ayant une fréquence de 1.66 GHZ, une mémoire vive (RAM) de 1 Go et un disque dur de 64 Go.

Cependant avant de donner les résultats de la segmentation, il est utile d'étudier la capacité des attributs de texture, extraits à partir des matrices Aura à discriminer des textures différentes.

4.2 Classification des images texturées

Afin d'évaluer le pouvoir discriminatoire des attributs de texture extraits à partir des GLAMs, nous avons créé une base d'images de texture à partir des images de l'album de Brodatz. Cette base contient 10 textures différentes (10 classes). Il s'agit des textures D112, D74, D54, D35, D26, D24, D20, D11, D5 et D4 (Fig. 4.1). Celles-ci se différencient par leurs aspects microscopique et macroscopique ayant des propriétés différentes. Chaque classe de texture contient 30 images de taille (64×64), ce qui nous donne au total une base de N=300 échantillons notée \mathcal{I} . Chaque image est ensuite caractérisée par des attributs de texture extraits par la méthode des GLAMs.

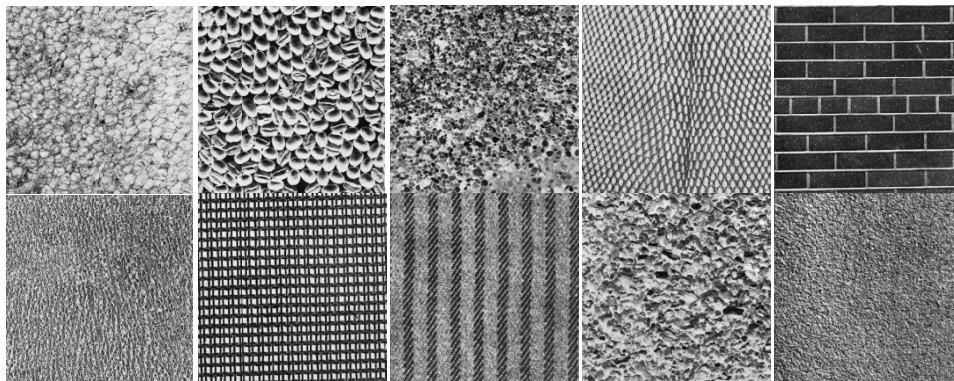


Fig. 4.1 : Base d'images. De gauche à droite et de haut en bas (D112, D74, D54, D35, D26, D24, D20, D11, D5, D4).

L'ensemble des échantillons \mathcal{I} est ensuite divisé en deux sous ensembles : un sous ensemble noté \mathcal{I}_{train} réservée à l'apprentissage et un autre noté \mathcal{I}_{test} réservée au test. Ces deux ensembles sont disjoints et ont la même taille, soit $|\mathcal{I}_{train}| = |\mathcal{I}_{test}| = 150$. L'ensemble \mathcal{I}_{train} constitue la base d'apprentissage, il est utilisé pour construire la règle de décision, tandis que l'ensemble test \mathcal{I}_{test} est utilisé pour évaluer les performances de la règle de décision. Les deux règles de décision employées sont Bayes et K- plus proches voisins. Pour évaluer les performances de ces deux classifieurs, nous avons déterminé le taux de classification à partir des observations de la base de test :

Quelques résultats de la classification des images de la base test sont répertoriés dans le tableau 4.1. Ces résultats sont obtenus en utilisant des éléments structurants de forme et de taille variables. Ils montrent bien que les attributs de texture extrait à partir des matrices Aura des niveaux de gris sont capables de discriminer des textures différentes puisque que le taux de classification est égale ou proche de 100%. Ces résultats montrent également que la forme et la taille des éléments structurants influents légèrement sur les performances du classifieur.

	Bayes	KPPV	Bayes	KPPV	Bayes	KPPV	Bayes	KPPV	Bayes	KPPV
Élément structurant	Carré		Boule		Croix		X		Losange	
5 × 5	100%	100%	97,33%	100%	98%	99,33%	100%	100%	98%	98,66
7 × 7	98,66%	98,66%	100%	100%	96%	98%	100%	96,66%	98,66	99,33%
9 × 9	99,33%	98,66%	98,66%	98,66%	98%	98,66%	98,66%	97,33%	98,66	98%

Tableau 4.1 : Résultats de la classification des images texturées caractérisées à partir des GLAMs

4.3 Segmentation d’images texturées basée sur la matrice Aura de niveaux de gris

Nous allons présenter dans cette section les principaux résultats de segmentation.

4.3.1 Critères d’évaluation

Afin d’évaluer d’une manière objective la performance des différentes méthodes de segmentation proposées, nous avons utilisé le taux de classification et le temps d’exécution comme critères d’évaluation.

Le taux de classification est basé sur des connaissances a priori disponibles sur l’image à savoir le nombre de régions de textures contenues dans l’image et la classe d’appartenance de chaque pixel à une classe de texture. Il est déterminé par le rapport entre le nombre de pixels bien classés et le nombre total des pixels de l’image. Pour cela, nous avons utilisé une image synthétique composée des trois textures tirées de l’album de BRODATZ à savoir D24 représentant le Cuire de veau pressé, D92 : peau de porc et D34 : filet. La taille de cette image est de (256*256) codée en 8 bits (Fig.4.2). Les textures D92 et D24 sont plus au moins fines, par contre, la texture D34 présente une macro- texture. Notons que les deux textures D24 et D92 sont difficilement discernables à l’œil nu.

Le temps d'exécution est utilisé comme critère de comparaison des procédures d'extraction des attributs de texture utilisant la technique classique (GLAM) ainsi que les deux techniques de programmation GLALL et GLAHS.

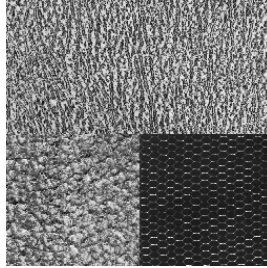


Fig. 4.2 :Image Test.

4.3.2 Attributs texturaux extraits à partir d'une matrice Aura

4.3.2.1 Utilisation d'un seul élément structurant

Les tableaux 4.2, 4.3, 4.4, 4.5 et 4.6 regroupent les valeurs du taux de classification et du temps de calcul des attributs de texture de l'image test obtenus à partir des matrices Aura en fonction de la taille de voisinage et en fonction des techniques de programmation utilisées pour le calcul des attributs. Dans ce cas, les résultats des tableaux sont obtenus en utilisant un seul élément structurant choisis parmi les 5 premiers éléments de la figure 3.12 (Chap. 3) ayant la forme d'un carré, d'une boule, d'une croix, d'une forme X et d'un losange.

Interprétation

Concernant le temps de calcul, les résultats montrent que quelque soit l'élément structurant, la méthode GLAHS est plus rapide par rapport aux méthodes GLAM et GLALL. La méthode GLAM consomme un temps assez important pour le calcul des attributs locaux de textures. Notons que pour la GLAM, ce temps dépend de la taille de l'élément structurant ainsi que la taille de la fenêtre de voisinage, mais aussi du nombre de pixels de l'image.

La taille de la fenêtre de voisinage et la taille de l'élément structurant jouent un rôle important dans l'estimation du temps du calcul. Dans la méthode GLALL, le temps de calcul augmente avec l'augmentation de la taille de la fenêtre de voisinage et de la taille de l'élément structurant. En effet, plus la taille de la fenêtre de voisinage et la taille de l'élément structurant augmentent, plus le nombre de couple différent de niveau de gris augmente. Par conséquent, la taille de la liste devient plus grande. Dans

ce cas, la GLALL présente l'inconvénient de consommer beaucoup du temps lors de la recherche du couple de valeurs de niveaux de gris (g, g') , dans la liste. Elle consomme un temps plus grand que celui de la GLAM.

Dans la méthode GLAHS, l'augmentation de la taille de la fenêtre de voisinage et la taille de l'élément structurant n'influence pas beaucoup sur le temps du calcul car la recherche de couple de niveaux de gris est directe.

L'avantage de la méthode GLAHS par rapport à la méthode GLALL apparaît clairement avec l'augmentation de la taille des fenêtres de voisinage et la taille de l'élément structurant.

Le temps du calcul des attributs de texture est pratiquement différent pour les différents éléments structurants utilisés. Pour les différentes méthodes de programmation, les éléments structurants de forme carré, boule et losange ont pratiquement le même temps du calcul qui reste plus grand à celui obtenu avec les éléments structurants ayant la forme d'une croix ou d'un X. Plus le nombre de pixels d'un élément structurant est grand, plus le temps de calcul augmente.

En ce qui concerne la qualité de la segmentation Les figures 4.3, 4.4, 4.5, 4.6 et 4.7 montrent les courbes de variation des valeurs de taux de classification obtenus avec des éléments structurants de forme carré, croix, boule, X et losange, respectivement. Dans chaque figure, les courbes sont données en fonction de la taille de la fenêtre de voisinage variant de $W=2$ à $W=9$ et en fonction de la taille de l'élément structurant.

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de classification
		GLAM	GLALL	GLAHS	
5×5	W=2	27min27s	2min40s	2min10s	93,45%
	W=3	28min14s	3min3s	2min12s	94,67%
	W=4	38min44s	5min50s	3min42s	96,66%
	W=5	40min34s	9min45s	4min32s	96,96%
	W=6	43min12s	11min36s	5min38s	97,49%
	W=7	48min15s	1h34min7s	6min42s	97,12%
	W=8	53min20s	1h40min	8min19s	96,73%
	W=9	57min22s	1h56min5s	10min4s	96,70%
7×7	W=2	28min34s	5min07s	3min11s	92,76%
	W=3	30min32s	6min14s	4mi4s	94,75%
	W=4	40min12s	12min3s	5min33s	96,35%
	W=5	42min25s	38min3s	7min2s	96,51%
	W=6	46min39s	1h34min23s	8min58s	96,66%
	W=7	48min11s	1h40min42s	11min2s	96,76%
	W=8	54min45s	2h44min32s	13min48s	96,62%
	W=9	59min31s	3h25min7s	15min34s	96,54%
9×9	W=2	29min35s	10min12s	3min46s	92,84%
	W=3	36min55s	19min35s	5min34s	94,29%
	W=4	45min3s	20min4s	7min44s	96,27%
	W=5	47min45s	1h21min4s	10min13s	96,34%
	W=6	50min40s	1h45min34s	13min13s	96,40%
	W=7	52min4s	2h43min42S	16min36s	96,63%
	W=8	57min18s	2h57min47s	20min37s	96,74%
	W=9	1h10min	3h34min38s	22min45s	96,73%
11×11	W=2	16min06s	10min17s	4min44s	92,39%
	W=3	18min45s	15min15s	7min13s	94,32%
	W=4	20min40s	25min04s	10min21s	95,66%
	W=5	56min7s	1h44min5s	13min55s	95,94%
	W=6	57min35s	2h56min28s	18min18s	96,45%
	W=7	1h2min9s	3h12min12s	22min3s	96,42%
	W=8	1h15min3s	3h24min38s	25min13s	96,38%
	W=9	1h30min	3h45min5s	28min40s	95,24%

Tableau 4.2 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme d'un carré.

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de classification
		GLAM	GLALL	GLAHS	
5×5	W=2	14min56s	2min10s	2min3s	93,37%
	W=3	15min14s	3min	2min23s	95,46%
	W=4	15min36s	4min23s	2min47s	96,75%
	W=5	16min16s	7min18s	3min36s	97,24%
	W=6	29min52s	13min35s	5min47s	97,12%
	W=7	34min39s	42min58s	6min40s	97,54%
	W=8	37min45s	56min7s	8min23s	96,87%
	W=9	42min56s	1h10min34s	10min45s	96,73%
7×7	W=2	14min23s	4min30s	3min03s	92,75%
	W=3	15min55s	6min	3min9s	95,60%
	W=4	16min44s	11min37s	5min4s	96,71%
	W=5	17min50s	21min14s	5min12s	97,04%
	W=6	38min24s	35min7s	9min	97,08%
	W=7	40min3s	1h35min	10min27s	96,96%
	W=8	44min16s	1h47min34s	11min45s	96,70%
	W=9	47min34s	2h13min45s	13min25s	96,54%
9×9	W=2	15min35s	11min12s	4min05s	92,84%
	W=3	16min55s	24min15s	5min30s	95,03%
	W=4	17min52s	24min45s	5min33s	96,40%
	W=5	19min35s	46min	7min24s	96,48%
	W=6	37min5s	1h15min10s	13min	96,68%
	W=7	39min42s	2h1min8s	17min24s	96,87%
	W=8	42min12s	2h22min45s	20min41s	97,34%
	W=9	46min38s	2h42min26s	25min23s	96,67%
11×11	W=2	14min55s	8min12s	6min13s	93,11%
	W=3	16min15s	12min44s	7min12s	94,50%
	W=4	17min50s	21min14s	10min17s	95,71%
	W=5	34min32s	1h32min23s	10min35s	96,53%
	W=6	41min56s	1h20min4s	18min21s	97,11%
	W=7	44min39s	1h58min	23min	96,53%
	W=8	48min45s	2h34min5s	25min12s	96,40%
	W=9	56min43s	2h47min23s	28min37s	95,96%

Tableau 4.3 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme d'une boule.

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de Classification
		GLAM	GLALL	GLAHS	
5×5	W=2	13min22s	1min56s	1min56s	92,41 %
	W=3	15min35s	2min26s	2min15s	95,51 %
	W=4	15min26s	3min	2min25s	96,26%
	W=5	16min26s	4min32s	2min55s	97,39 %
	W=6	16min34s	7min38s	3min20s	97,41%
	W=7	17min11s	8min2s	3min40s	97,50%
	W=8	19min45s	10min23s	5min13s	97,37%
	W=9	23min43s	12min57s	6min56s	97,11%
7×7	W=2	13min33s	2min45s	2min10s	91,65%
	W=3	15min43s	3min	2min27s	95,32 %
	W=4	17min55s	5min	2min49s	95,67%
	W=5	16min32s	7min01s	3min30s	97,25%
	W=6	18min50s	9min40s	3min52s	97,32 %
	W=7	19min58s	14min30s	4min32s	97,40%
	W=8	21min34s	17min14s	5min21s	97,21%
	W=9	24min23s	19min57s	7min4s	97,01%
9×9	W=2	14min07s	2min25s	2min08s	92,28%
	W=3	15min36s	2min45s	2min43s	95,06%
	W=4	16min29s	5min48s	3min17s	95,98%
	W=5	17min36s	10min1s	4min	96,83%
	W=6	18min01s	14min40s	4min35s	97,04 %
	W=7	19min08s	22min20s	6min	97,18%
	W=8	21min12s	32min27s	6min28s	96,70%
	W=9	23min54s	37min43s	7min10s	96,36%
11×11	W=2	14min47s	2min55s	2min15s	92,08%
	W=3	15min56s	4min38s	3min07s	93,47%
	W=4	16min34s	10min45s	4min02s	95,55%
	W=5	17min36s	12min46s	4min22s	96,83%
	W=6	18min42s	20min45s	5min42s	96,95%
	W=7	33min31s	34min19s	6min21s	97,08%
	W=8	34min55s	41min34s	6min28s	96,99%
	W=9	37min02s	46min10s	6min54s	96,78%

Tableau 4.4: Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme d'une croix

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de classification
		GLAM	GLALL	GLAHS	
5×5	W=2	14min23s	2min30s	1min50s	91,37%
	W=3	16min3s	2min54s	2min11s	95,11%
	W=4	18min30s	3min29s	2min27s	95,68%
	W=5	19min30s	5min2s	3min	96,04%
	W=6	20min46s	5min12s	3min14s	97,12%
	W=7	29min6s	6min34s	3min44s	97,05%
	W=8	32min9s	10min45s	5min15s	96,73%
	W=9	35min45s	20min5s	6min54s	96,58%
7×7	W=2	13min39s	2min15s	2min10s	92,25%
	W=3	15min52s	3min24s	2min27s	94,83%
	W=4	17min22s	4min59s	2min52s	96,33%
	W=5	18min52s	6min32s	3min33s	96,78%
	W=6	19min56s	9min45s	4min11s	96,82%
	W=7	33min18s	9min56s	4min26s	96,88%
	W=8	37min42s	10min47s	4min34s	96,99%
	W=9	40min50s	13min23s	4min51s	96,84%
9×9	W=2	15min12s	2min45s	2min18s	93,20%
	W=3	16min46s	3min55s	2min57s	95,26%
	W=4	29min21s	5min2s	3min19s	95,87%
	W=5	29min29s	9min45s	4min1s	96,45%
	W=6	34min45s	12min54s	4min46s	96,60%
	W=7	35min37s	25min37s	5min50s	96,74%
	W=8	38min3s	33min32s	6min38s	96,83%
	W=9	42min45s	36min28s	7min12s	96,78%
11×11	W=2	16min57s	3min35s	2min45s	92,48%
	W=3	17min36s	4min58s	3min34s	93,57%
	W=4	20min45s	8min45s	4min04s	95,45%
	W=5	32min34s	10min23s	4min31s	96,11%
	W=6	34min6s	15min33s	5min32s	96,45%
	W=7	35min47s	36min47s	8min6s	96,75%
	W=8	39min53s	43min51s	12min4s	96,62%
	W=9	46min37s	51min03s	14min39s	96,44%

Tableau 4.5 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme X.

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de classification
		GLAM	GLALL	GLAHS	
5×5	W=2	14min56s	2min10s	2mn2s	93,30%
	W=3	15min14s	3min	2mn23s	95,46%
	W=4	15min36s	4min2s	2mn27s	96,75%
	W=5	16min16s	7min10s	3mn30s	97,24%
	W=6	28min51s	9min25s	3min50s	97,39%
	W=7	29min39s	14min29s	5min	97,43%
	W=8	32min18s	16min13s	6min56s	97,34%
	W=9	35min34s	20min56s	8min12s	96,75%
7×7	W=2	17min36s	3min02s	2min20s	93,48%
	W=3	26min02s	5min09s	3min	95,64%
	W=4	26min42s	9min03s	3min49s	96,69%
	W=5	29min29s	21min03s	4min40s	97,10%
	W=6	30min30s	26min08s	5min47s	97,61%
	W=7	44min30s	34min29s	6min57s	97,14%
	W=8	47min24s	39min34s	8min45s	96,95%
	W=9	53min14s	57min35s	11min4s	96,75%
9×9	W=2	18min15s	5min32s	2min48s	92,22%
	W=3	23min05s	9min25s	4min04s	95,31%
	W=4	29min41	18min28s	5min	96,47%
	W=5	31min22s	33min37s	6min25s	96,76%
	W=6	32min41s	55min31s	8min12s	97,08%
	W=7	54min34s	1h12min4s	10min	96,86%
	W=8	58min45s	1h34min15s	12min17s	96,53%
	W=9	1h3min38s	2h43min49s	15min28s	96,37%
11×11	W=2	20min12s	7min22s	3min32s	92,86%
	W=3	26min45s	11min34s	5min02s	94,64%
	W=4	29min55s	20min44s	6min40s	95,90%
	W=5	32min45s	1h14min	8min32s	96,57%
	W=6	40min51s	1h36min	11min2s	96,57%
	W=7	59min14s	1h45min45s	13min37	96,36%
	W=8	1h14min6s	2h15min19s	15min23s	96,24%
	W=9	1h20min	2h34min15s	17min29s	95,76%

Tableau 4.6 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme d'un losange.

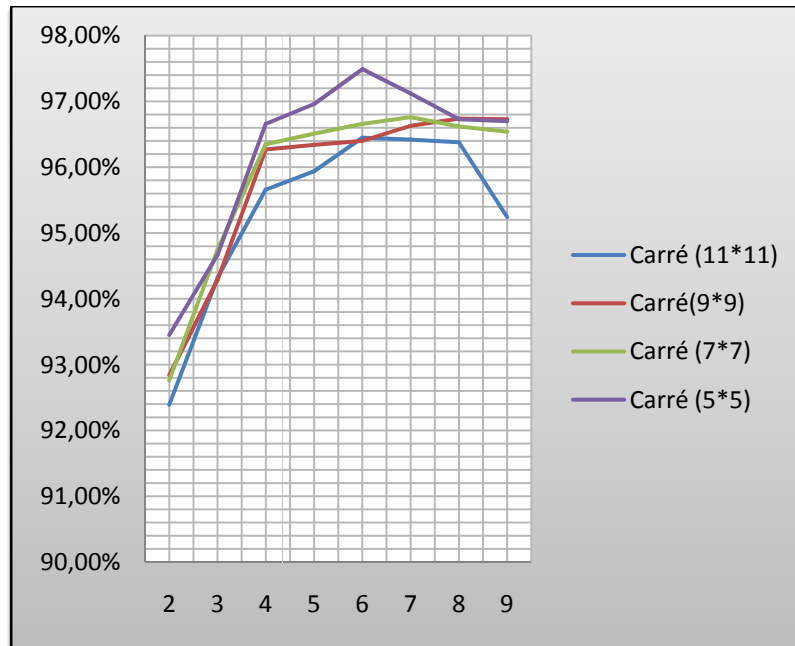


Fig. 4.3: Taux de classification obtenu en utilisant l'élément structurant ayant la forme d'un carré

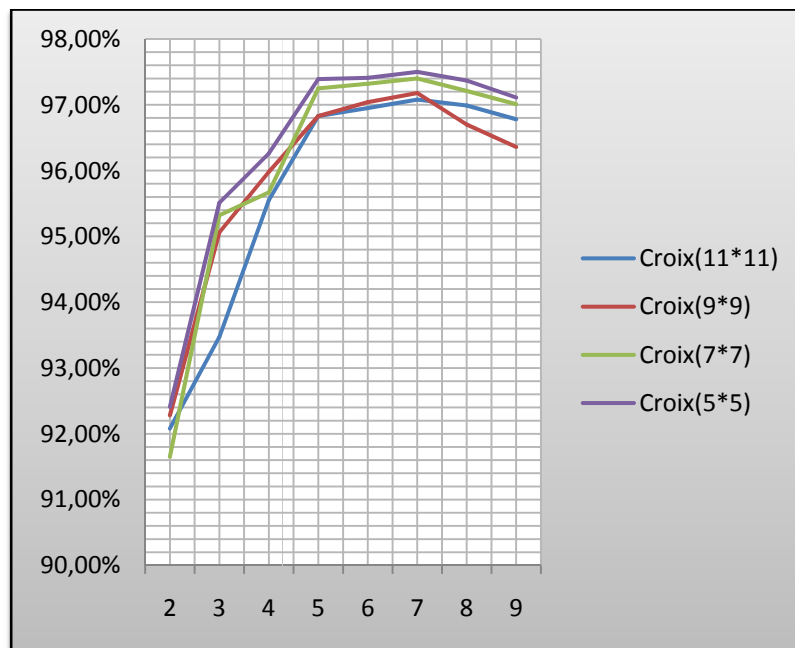


Fig. 4.4 : Taux de classification obtenu en utilisant l'élément structurant ayant la forme d'une croix

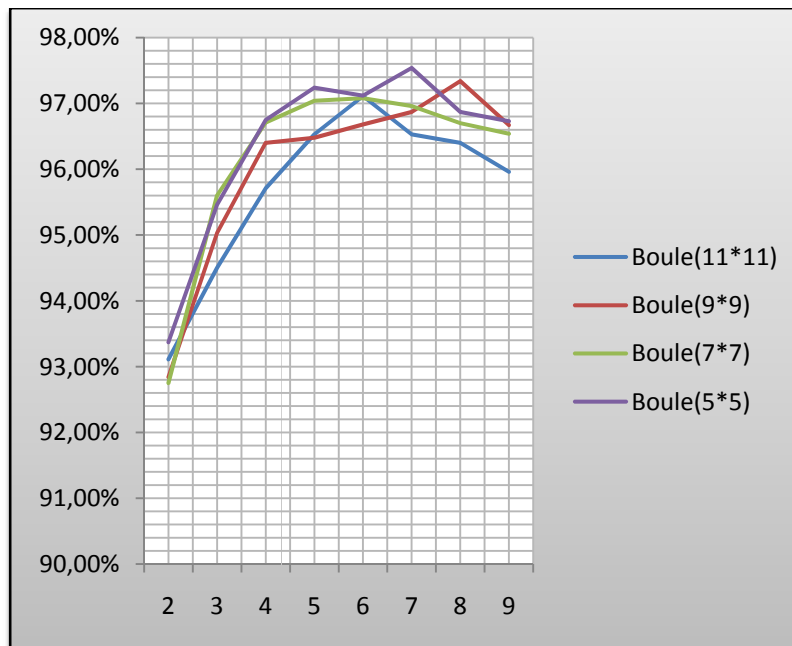


Fig. 4.5 : Taux de classification obtenu en utilisant l'élément structurant ayant la forme d'une boule

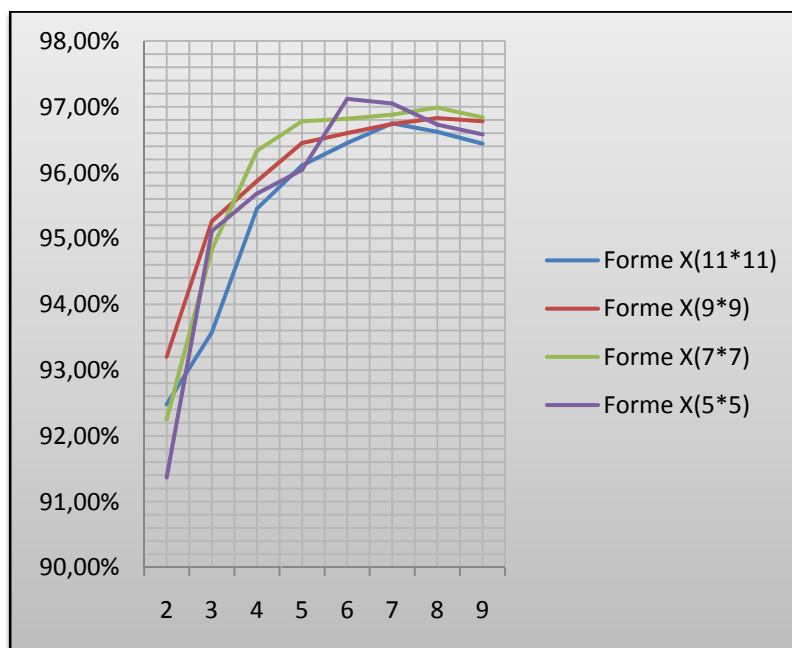


Fig. 4.6: Taux de classification obtenu en utilisant l'élément structurant ayant la forme X

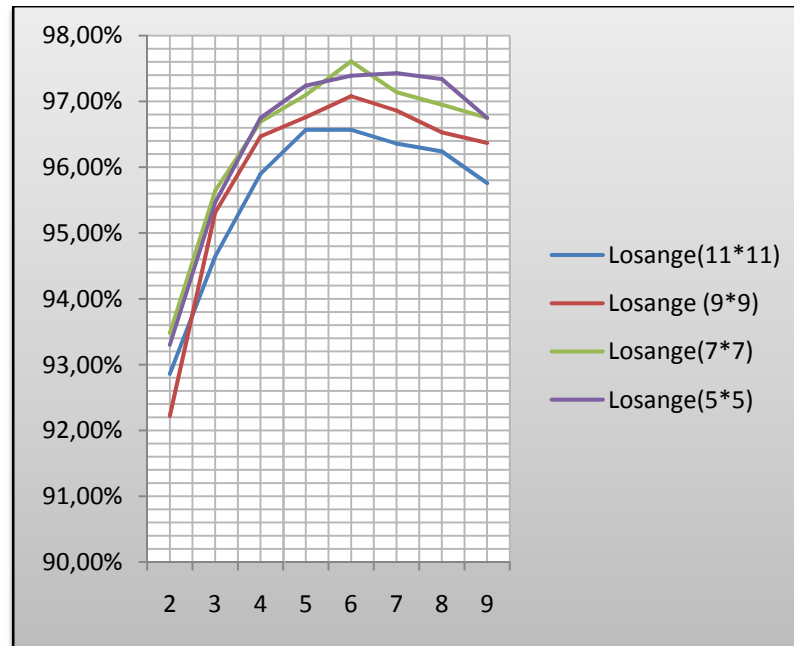


Fig. 4.7 : Taux de classification obtenu en utilisant l'élément structurant ayant la forme d'un Losange

Interprétation

A travers les figures précédentes, nous pouvons remarquer que les courbes de variation du taux de classification varient quasiment de la même manière en fonction de la taille de la fenêtre de voisinage et ce quelque soit la forme et la taille de l'élément structurant utilisé. Plus la taille de la fenêtre de voisinage augmente, plus le taux de classification augmente pour atteindre une valeur maximale. Cependant, à partir d'une certaine taille, le taux de classification diminue, ceci est dû à une mauvaise localisation des frontières entre les différentes textures. Ainsi, pour l'élément structurant de la forme d'un carré (Fig.4.3) le taux maximal 97,49% est obtenu avec une taille (5 et $W=6$. Avec la croix (Fig.4.4) le taux de classification maximal 97,50% est obtenu avec un élément de taille (5 et $W=7$. Pour l'élément structurant ayant la forme d'une boule (Fig.4.5), le taux maximal est 97,54%, il est obtenu avec une taille de (5 et $W=7$. L'élément structurant de la forme X et de taille (5 donne un taux maximal de 97,12%, pour une fenêtre de voisinage ayant $W=6$ (Fig.4.6). Un taux maximal de 97,61% est obtenu avec l'élément structurant ayant la forme d'un losange et une taille de (7 avec une fenêtre de voisinage caractérisée par $W=6$ (Fig.4.7).

Les figures 4.8, 4.9, 4.10 et 4.11 montrent par contre les courbes de variation du taux de classification en fonction de la taille de la fenêtre de voisinage W et en fonction

de l'élément structurant. Dans chaque figure, les courbes correspondent à une seule taille de l'élément structurant.

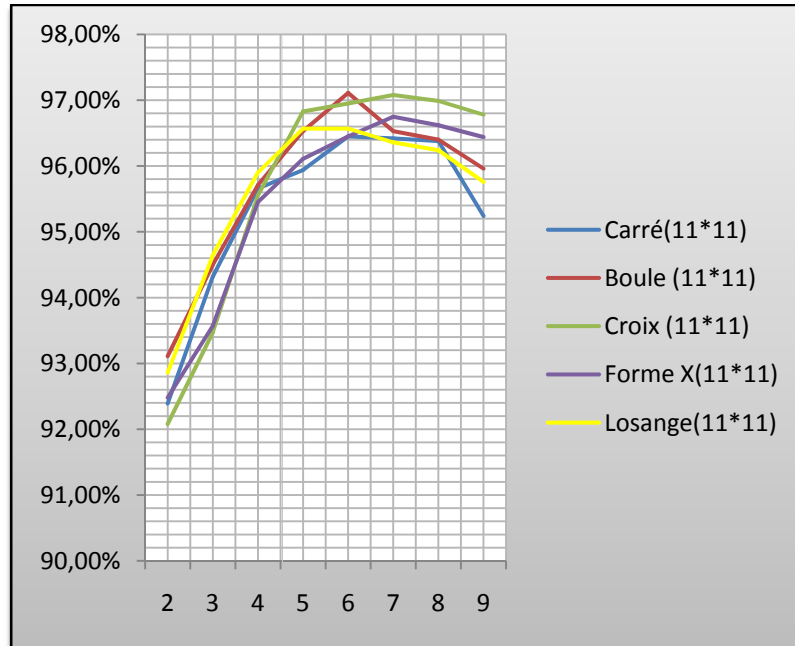


Fig. 4.8 : Taux de classification en fonction de l'élément structurant de taille (11*11)

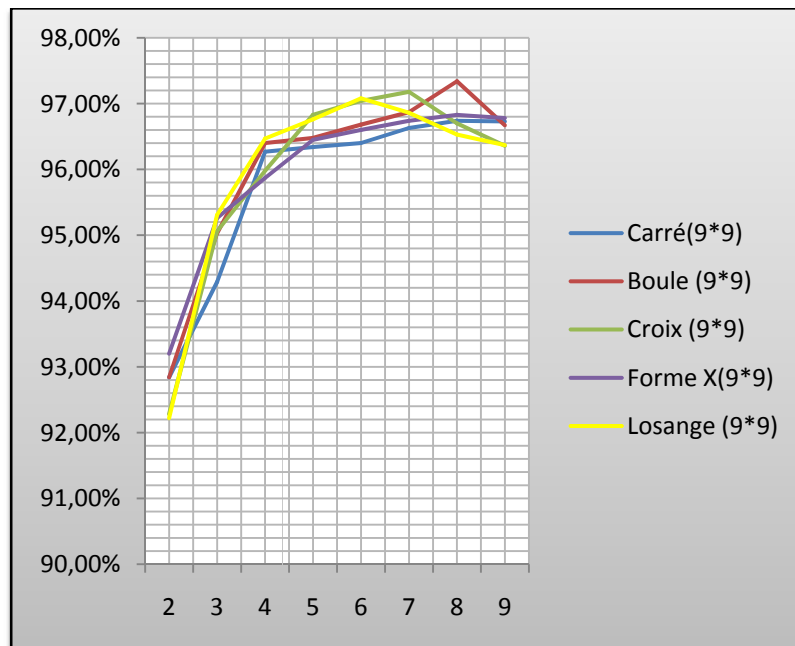


Fig. 4.9: Taux de classification en fonction de l'élément structurant de taille (9*9)

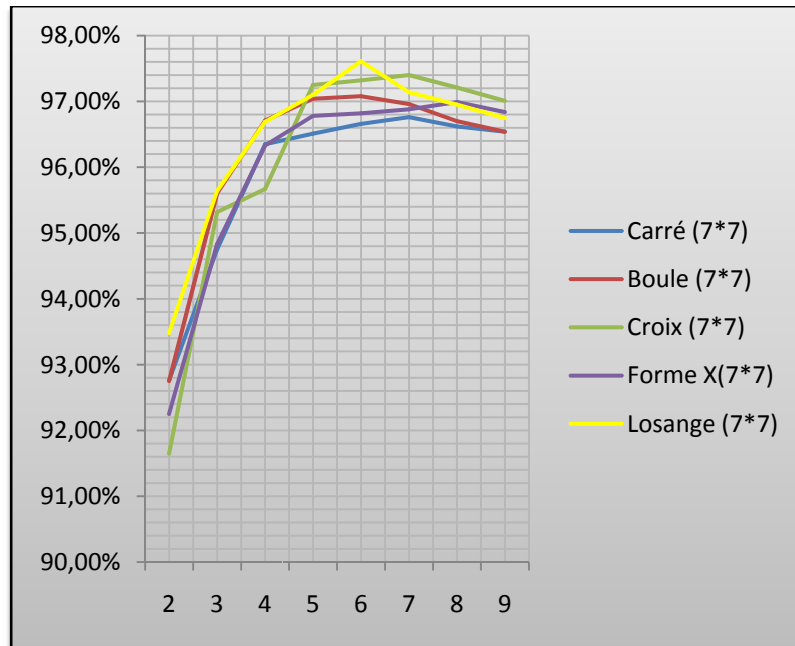


Fig. 4.10 : Taux de classification en fonction de l'élément structurant de taille (7*7)

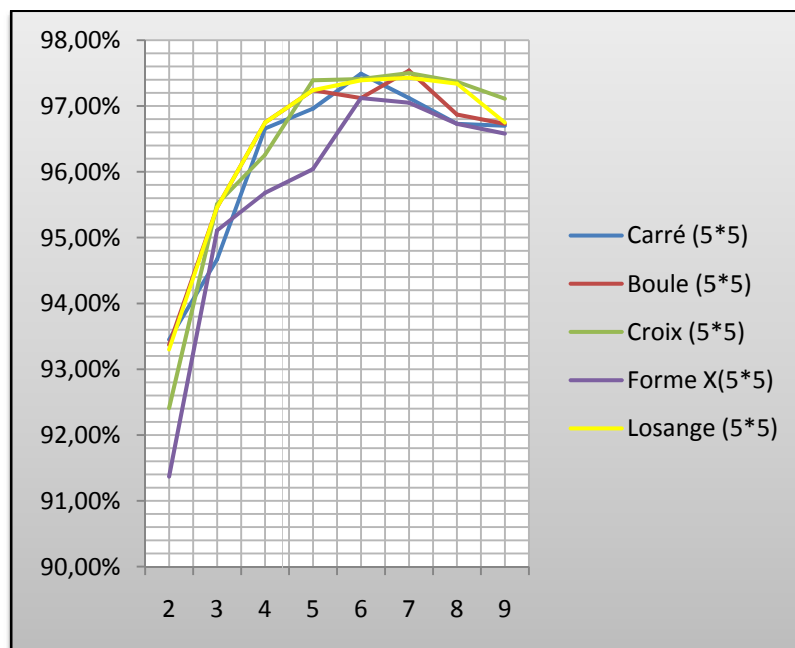


Fig. 4.11 : Taux de classification en fonction de l'élément structurant de taille (5*5)

Ces courbes montrent également que le taux de classification varie quasiment de la même manière en fonction de la taille de la fenêtre de voisinage et ce quelque soit la forme et la taille de l'élément structurant utilisé. Ainsi pour l'image test de la figure 4.2, le meilleur résultat est obtenu en utilisant l'élément structurant de forme losange de taille pour une fenêtre de voisinage . Ces paramètres constituent un bon compromis entre la segmentation et la localisation des frontières.

Tous les résultats donnés jusqu'à présent sont obtenus en utilisant des éléments structurants symétriques. Les mêmes conclusions ont été obtenues avec des éléments structurants non symétriques. Ceux-ci n'ont pas été incorporés par soucis d'espace. Cependant, on peut voir sur la figure 4.12 quelques résultats de la segmentation de l'image test en fonction des différents éléments structurant symétriques et asymétriques.

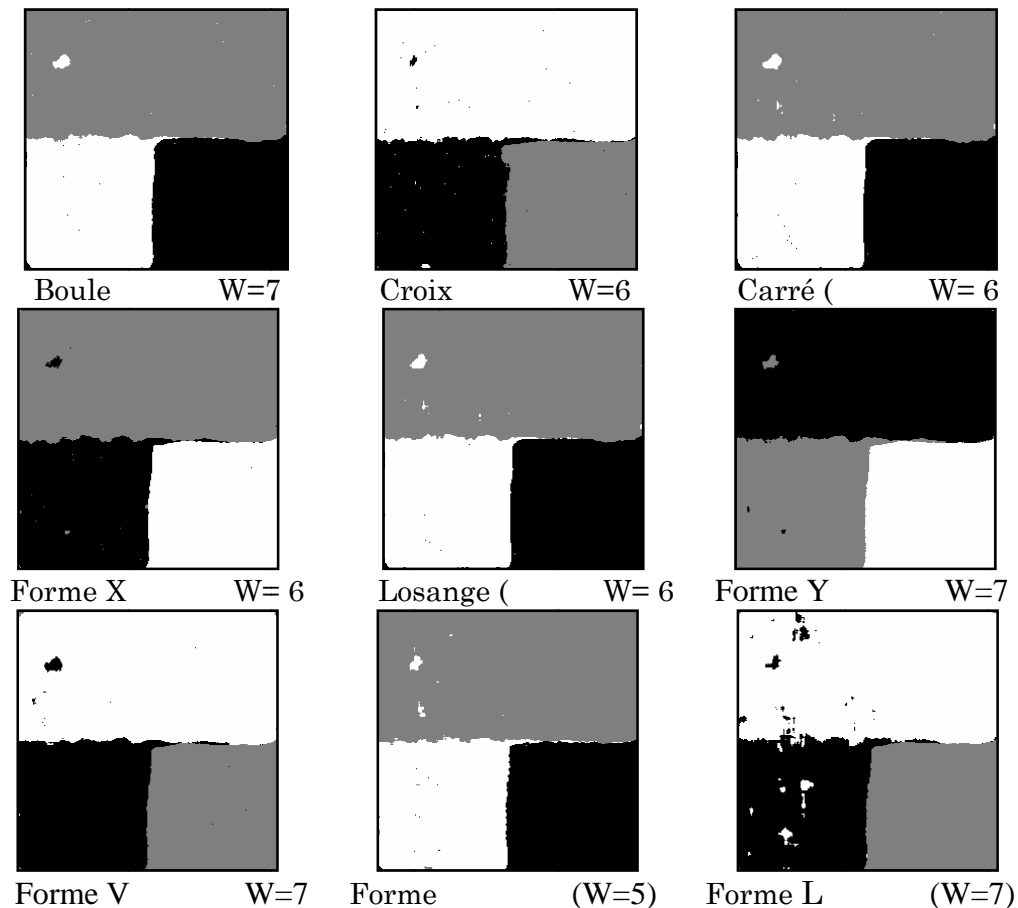


Fig. 4.12 : Images segmentées en utilisant les attributs de texture extraits à partir des matrices Aura

4.3.2.2 Utilisation de plusieurs structures de voisinage de même forme et de tailles variables.

Les tableaux 4.7, 4.8, 4.9, 4.10 et 4.11 donnent les valeurs du taux de classification et du temps de calcul des attributs de texture de l'image test obtenus avec les attributs de textures extraits à partir des matrices Aura en fonction de la taille de voisinage. Ces résultats sont obtenus en utilisant plusieurs éléments structurants à la fois ayant la même forme avec des tailles variables (analyse multi résolution).

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	38min7s	93,43%
W=3	40min38s	96,11%
W=4	49min13s	96,66%
W=5	52min1s	96,93%
W=6	1h10min02s	97,50%
W=7	1h21min30s	97,58%
W=8	1h40min48s	96,90%
W=9	1h57min5s	96,87%
W=10	2h23min22s	96,70%
W=11	2h36min16s	96,58%

Tableau 4.7 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme d'un carré mais des tailles variables.

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	34min12s	93,46%
W=3	35min21s	95,90%
W=4	49min56s	97,15%
W=5	56min2s	97,48%
W=6	58min43s	97,58%
W=7	1h4min44s	97,72%
W=8	1h42min52s	97,69%
W=9	1h50min5s	97,61%
W=10	2h7min6s	97,44%
W=11	2h26min45s	96,55%

Tableau 4.8 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme d'une boule mais des tailles variables.

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	26min51s	93,21%
W=3	30min40s	95,97%
W=4	32min57s	96,97%
W=5	36min	97,56%
W=6	38min9s	97,62%
W=7	41min45s	97,64%
W=8	49min14s	97,64%
W=9	49min58s	97,68%
W=10	53min22s	97%
W=11	58min42s	96,85%

Tableau 4.9 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme d'une croix mais des tailles variables.

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	29min27s	92,91%
W=3	31min28s	94,71%
W=4	36min33s	96,59%
W=5	38min02s	96,70%
W=6	39min16s	97,16%
W=7	42min34s	97,09%
W=8	50min43s	96,92%
W=9	51min12s	96,89%
W=10	54min25s	96,73%
W=11	59min3s	96,59%

Tableau 4.10 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme X mais des tailles variables.

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	31min25s	93,96%
W=3	34min13s	95,71%
W=4	39min12s	96,73%
W=5	44min07s	96,97%
W=6	52min47s	97,48%
W=7	57min15s	97,54%
W=8	1h07min31s	97,90%
W=9	1h18min30s	97,61%
W=10	1h26min14s	97,64%
W=11	1h32min42s	97,64%

Tableau 4.11 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme d'un losange mais des tailles variables.

Interprétation

Comme nous l'avons évoqué précédemment, les deux méthodes GLAM et GLALL consomment un temps assez long lors du calcul des attributs de texture. Ce temps devient excessivement long lorsqu'on utilise en même temps plusieurs éléments structurants ayant tous la même forme mais des tailles variables. A titre d'exemple, pour un élément structurant de forme du losange de taille variable et $W=8$, la GLAM consomme 2h52min 33s et la GLALL 7h23min25s. C'est la raison pour laquelle nous avons relevé les temps de calcul uniquement pour la méthode GLAHS. Ces temps de calcul restent très courts par rapport à ceux obtenus par les deux autres méthodes GLAM et GLALL. Cependant, il y'a lieu d'indiquer que contrairement à l'utilisation d'un seul élément structurant de taille fixe, la taille de la fenêtre de voisinage influe sur le temps du calcul car les pixels de la fenêtre de voisinage sont parcourus autant de fois que le nombre de taille de l'élément considéré.

La figure (4.13) montre la variation du taux de classification en fonction de l'élément structurant de la taille de la fenêtre de voisinage.

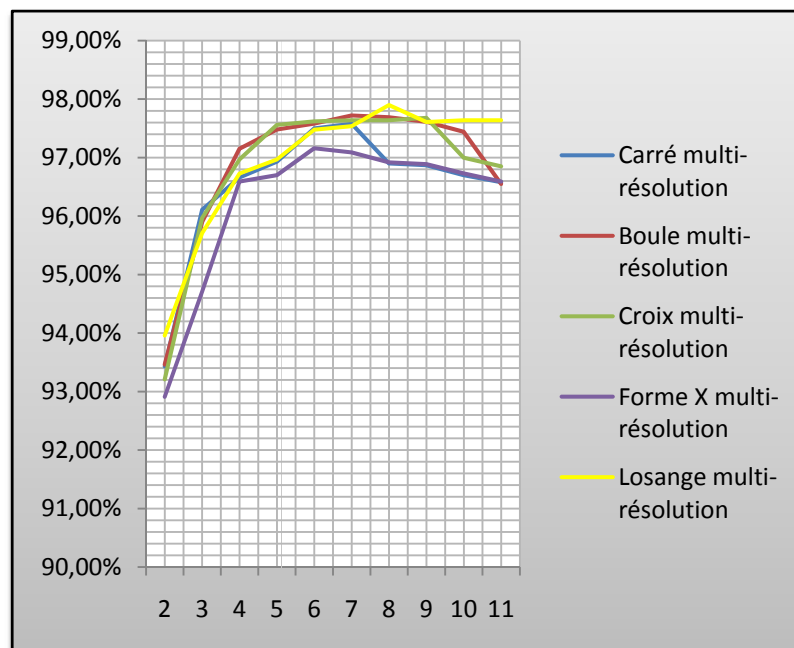


Fig. 4.13 : Taux de classification en fonction de l'élément structurant de taille variable

On remarque que comme pour un seul élément structurant, dans l'analyse multi-résolution quelque soit la forme de l'élément structurant, plus la taille de la fenêtre de voisinage augmente, plus le taux de classification augmente pour atteindre un taux maximal ($W=7$ pour le carré et la boule, $W=9$ pour la croix, $W=6$ pour la forme X et $W=8$ pour le losange). Cependant, à partir de cette taille le taux de classification diminue à cause d'une mauvaise localisation des frontières entre les différentes textures. Cependant, ce taux de classification (97.90%) reste supérieur au meilleur résultat (97.61%) obtenu avec un seul élément structurant de taille (7×7) qui a la même forme losange avec $W=6$. Il y'a lieu également de noter que, pour cette image test, la différence entre ces deux taux n'est pas vraiment significative. Elle est toute de même très expressive pour d'autres images comme nous le verrons par la suite. Comme, il est également remarqué que ce n'est pas toujours le même élément structurant qui donne le meilleur résultat.

La figure 4.14 montre les meilleurs résultats de la segmentation de l'image test en utilisant plusieurs éléments structurants de la même forme et des tailles variables.

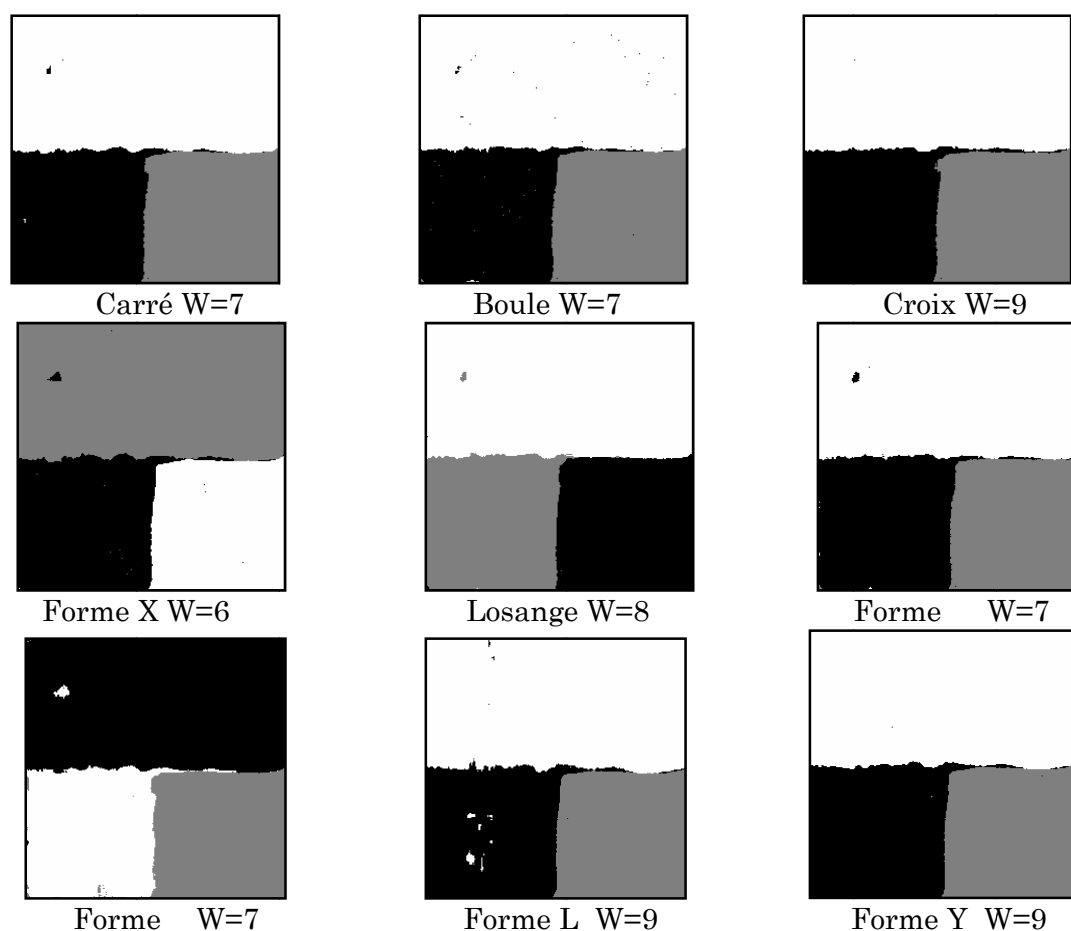


Fig. 4.14 : Images segmentées en utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la même forme mais des tailles variables.

Interprétation

En comparant ces résultats par rapport à ceux obtenus par un seul élément structurant, nous remarquons que les taches isolées ont pratiquement disparues avec plusieurs éléments structurants de la même forme et des tailles variables (analyse multi résolution).

4.4 Utilisation des matrices Aura de base

Le tableau 4.12 donne les valeurs du taux de classification et du temps de calcul des attributs de texture de l'image test obtenus avec les attributs de textures extraits à partir des matrices Aura de base BGLAHS en fonction de la taille de voisinage.

La figure 4.15 illustre les résultats de la segmentation de l'image test en utilisant les matrices Aura de base.

Taille de la fenêtre de voisinage	Temps de calcul de la BGLAHS	Taux de classification
W=3	1h11min56s	93,31%
W=5	1h14min34s	96,09%
W=7	1h14min40s	96,30%
W=9	1h14min56s	96,11%

Tableau 4.12 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura de base.

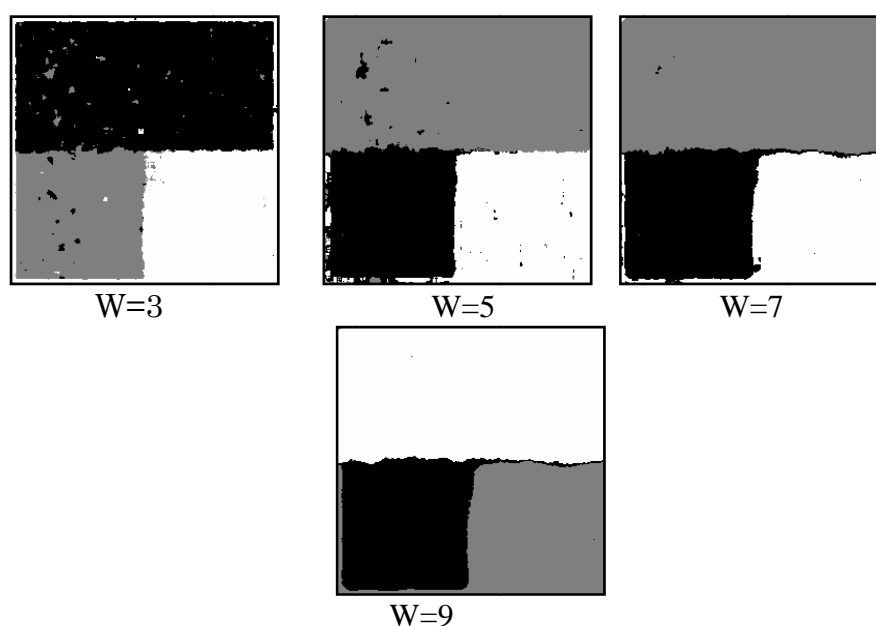


Fig. 4.15 : Images segmentées en utilisant les attributs de textures extraits à partir des BGLAMs.

Interprétation

On constate également, que la taille de la fenêtre de voisinage joue un rôle important dans la segmentation d'image par les attributs de texture issus des matrices Aura de base. Le tableau 4.12 montre que, plus la taille de voisinage augmente plus le taux de classification augmente pour atteindre un taux maximal pour W=7. Cependant, pour cette image test, le taux de classification obtenu avec les BGLAMs reste légèrement inférieur à celui obtenu avec les GLAMs que ce soit pour un seul élément structurant ou pour l'analyse multi résolution.

4.5 Comparaison avec les attributs texturaux extraits à partir d'une matrice de cooccurrence

Afin de comparer les résultats de la segmentation obtenus en utilisant les attributs de textures extraits à partir des matrices Aura de niveaux de gris et ceux obtenus par les attributs extraits à partir des matrices de cooccurrences, nous avons utilisé une image synthétique (1) donnée par la figure 4.16. Elle est constituée de quatre textures différentes.

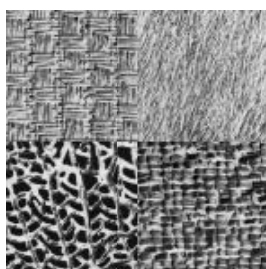


Fig. 4.16 : Image synthétique 1

Le tableau 4.13 regroupe les valeurs du taux de classification de l'image de la figure 4.16 obtenus avec les attributs de textures extraits à partir des matrices de cooccurrence (direction 0°) et des matrices Aura (élément structurant de forme carré de taille W) en fonction de la taille de voisinage.

Taille de la fenêtre de voisinage	Taux de classification	
	GLCM	GLAM
W=5	56,43%	90,12%
W=6	56,75%	90,99%
W=7	58,54%	90,94%
W=8	54,23%	90,42%
W=9	45,32%	87,62%

Tableau 4.12 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des différentes méthodes.

La figure 4.17 montre les résultats de la segmentation de l'image de la figure 4.16 en utilisant les six attributs locaux extraits à partir des matrices de cooccurrences pour les différentes tailles de la fenêtre de voisinage en prenant en compte une seule direction.

La figure 4.18 montre les résultats de la segmentation de l'image de la figure 4.16 en utilisant les attributs de textures extraits à partir des matrices Aura en fonction de l'élément structurant.

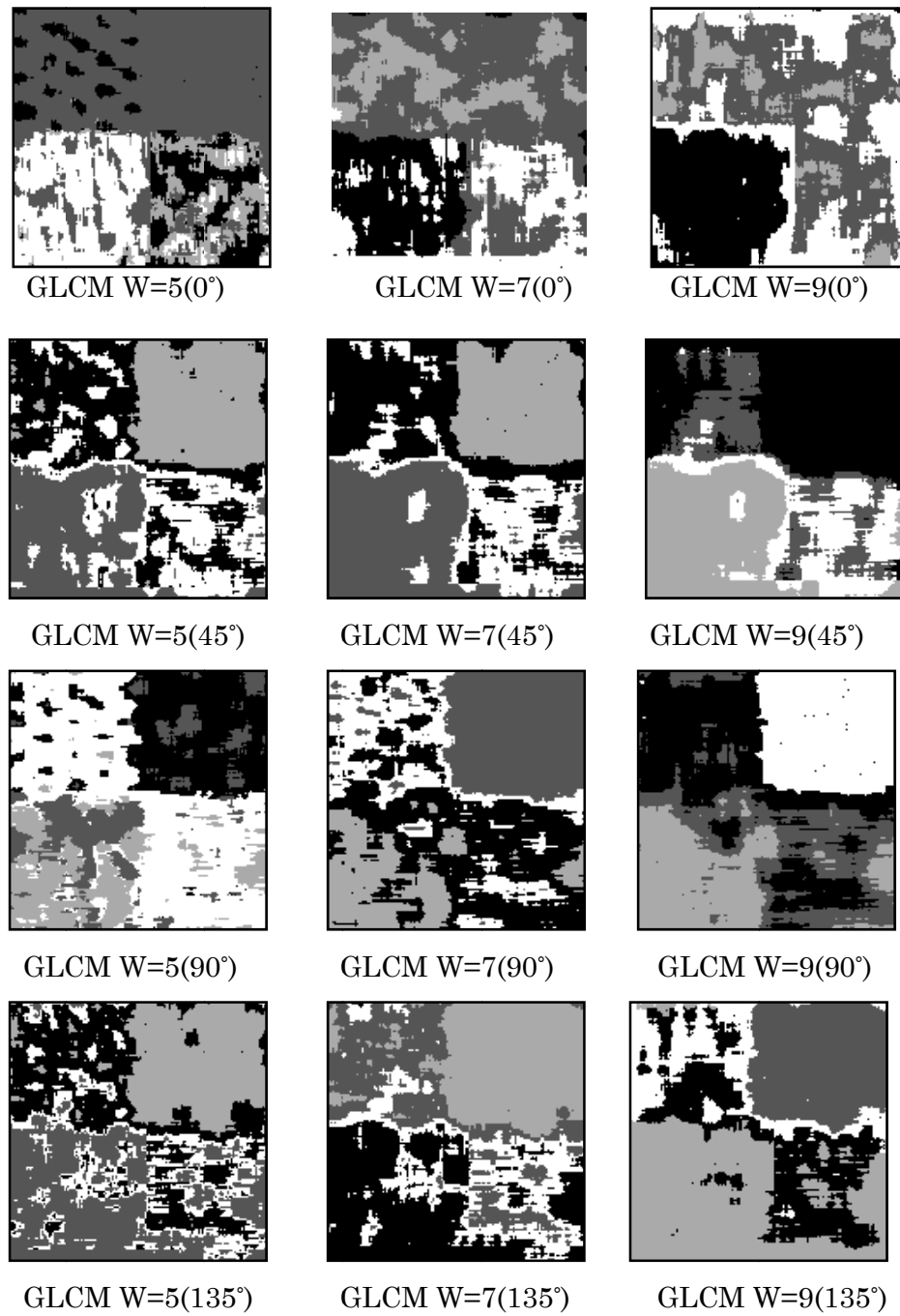


Fig. 4.17 : Images segmentées en utilisant les attributs de textures extraits à partir des matrices de cooccurrence pour les différentes directions

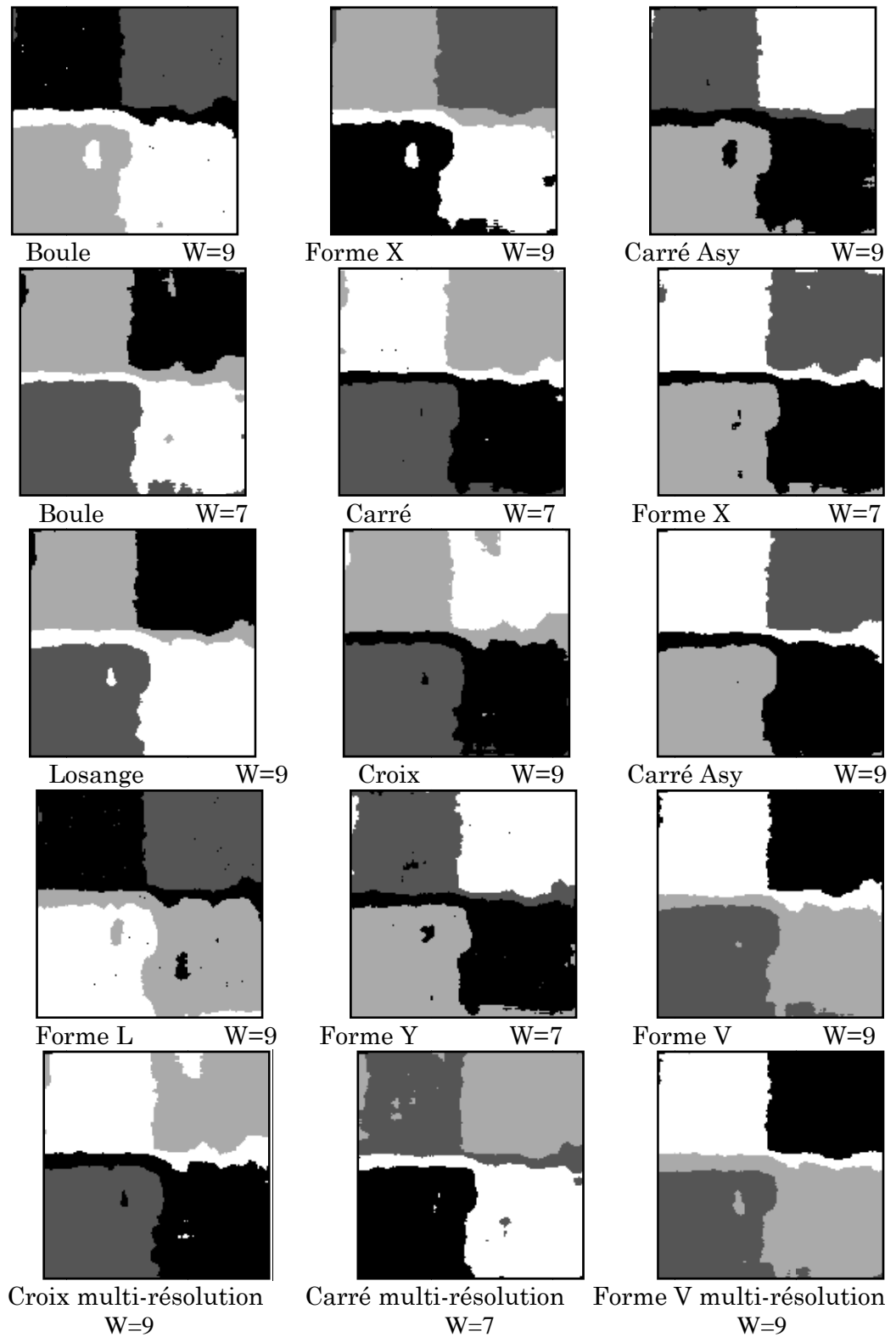


Fig. 4.18 : Images segmentées en utilisant les attributs de textures extraits à partir des matrices Aura pour les différents éléments structurants.

Interprétation

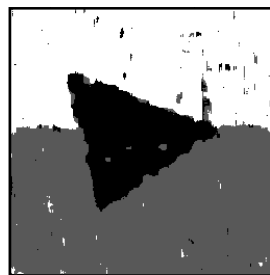
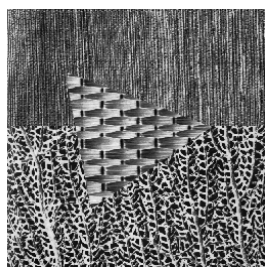
Quelque soit l'élément structurant utilisé, les résultats de la segmentation à base des matrices Aura sont beaucoup meilleurs que ceux obtenus avec les matrices des cooccurrences même si des problèmes aux frontières des textures persistent.

4.6 Images synthétiques et réelles

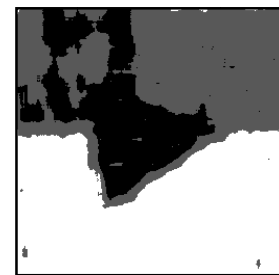
Nous avons également appliqué la méthode proposée sur d'autres images synthétiques et sur des images réelles.

Image synthétique 2

La figure (4.19) montre les différentes images segmentées obtenus sur une autre image synthétique par la méthode des matrices de cooccurrence et les matrices Aura



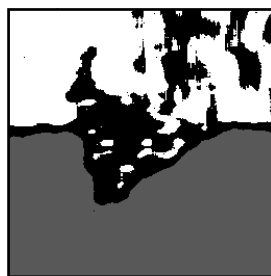
GLCM $w=5(0^\circ)$



Croix de taille
W=9



Boule de taille
W= 9



Carré de taille
W=9



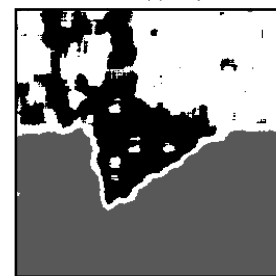
Forme L
W=7



Losange de taille
W=9



Forme X de taille
W= 9



Forme de taille
W=7

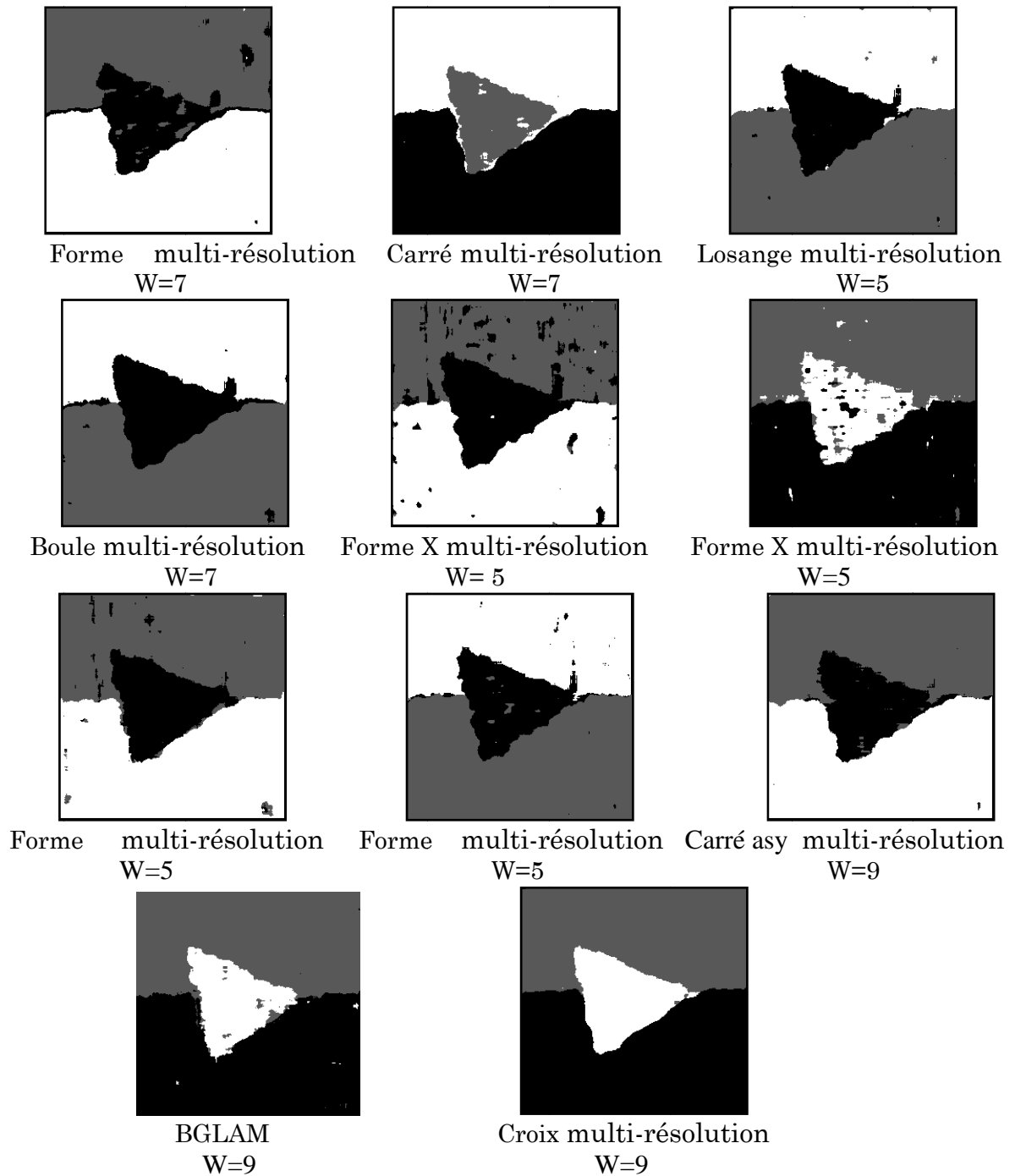


Fig.4.19 : Image synthétique 2 segmentée par les différentes méthodes.

Interprétation

Pour cette image, nous pouvons remarquer que l'utilisation d'un seul élément structurant peut s'avérer insuffisant (Fig. 4.19). En effet, les trois zones texturées sont mises en évidence d'une certaine manière. Mais ces résultats restent moins bons par rapport à ceux obtenus par l'utilisation de plusieurs éléments structurants de la même forme et de tailles différentes (analyse multi-résolution) car plus le nombre d'éléments

est élevé plus l'information texturale sera riche. Contrairement à l'image précédente, les résultats obtenus sur cette image synthétique montre que les BGLAMs, la GLAM avec un seul élément non symétrique et la GLCM sont plus performantes par rapport à la GLAM avec un seul élément symétrique.

Image synthétique 3

La figure (4.20) montre les différentes images segmentées obtenus sur une image synthétique composée de 5 textures différentes par la méthode des matrices de cooccurrence et les matrices Aura

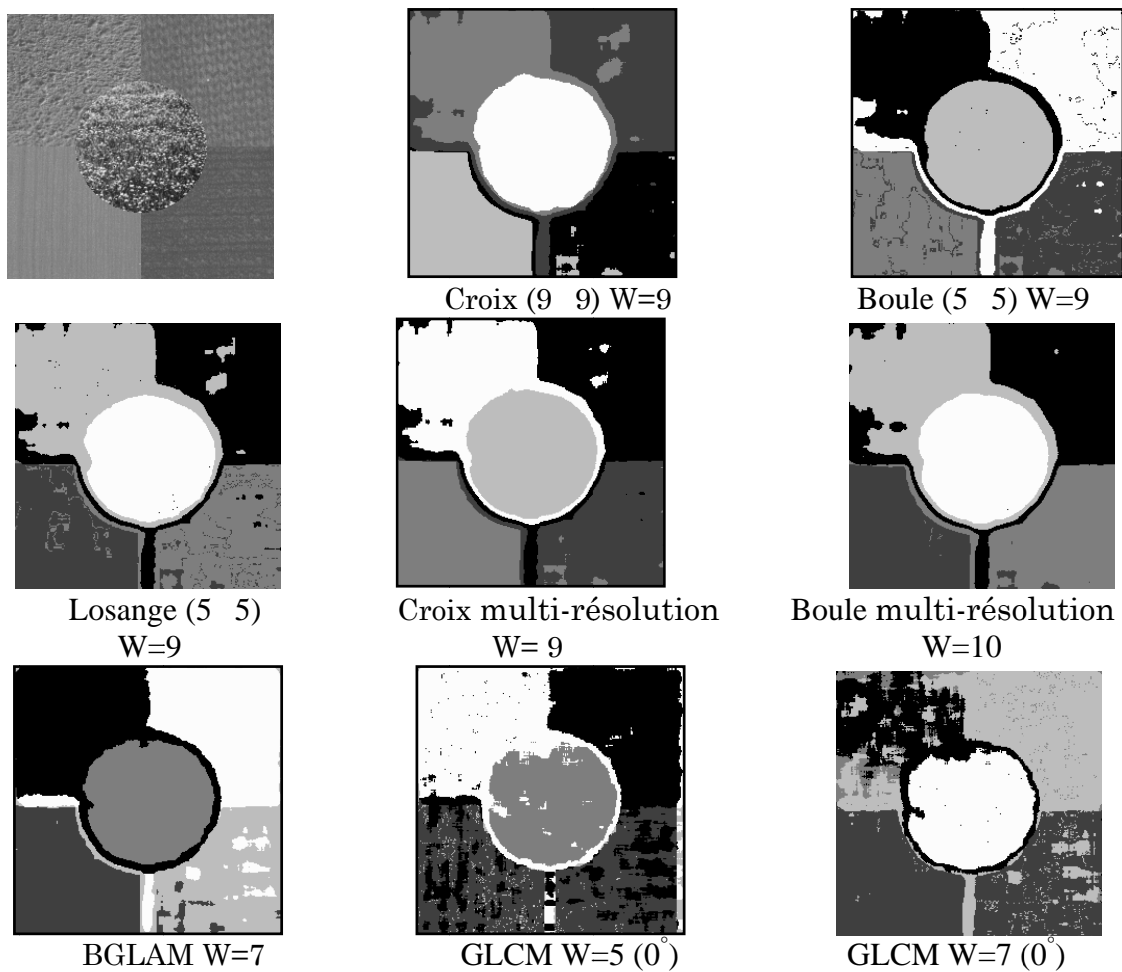


Fig.4.20 : Image synthétique 3 segmentée par les différentes méthodes.

Image synthétique 4

La figure (4.21) montre les différentes images segmentées obtenus sur une autre image synthétique composée de 4 autres textures par la méthode des matrices de cooccurrence et les matrices Aura.

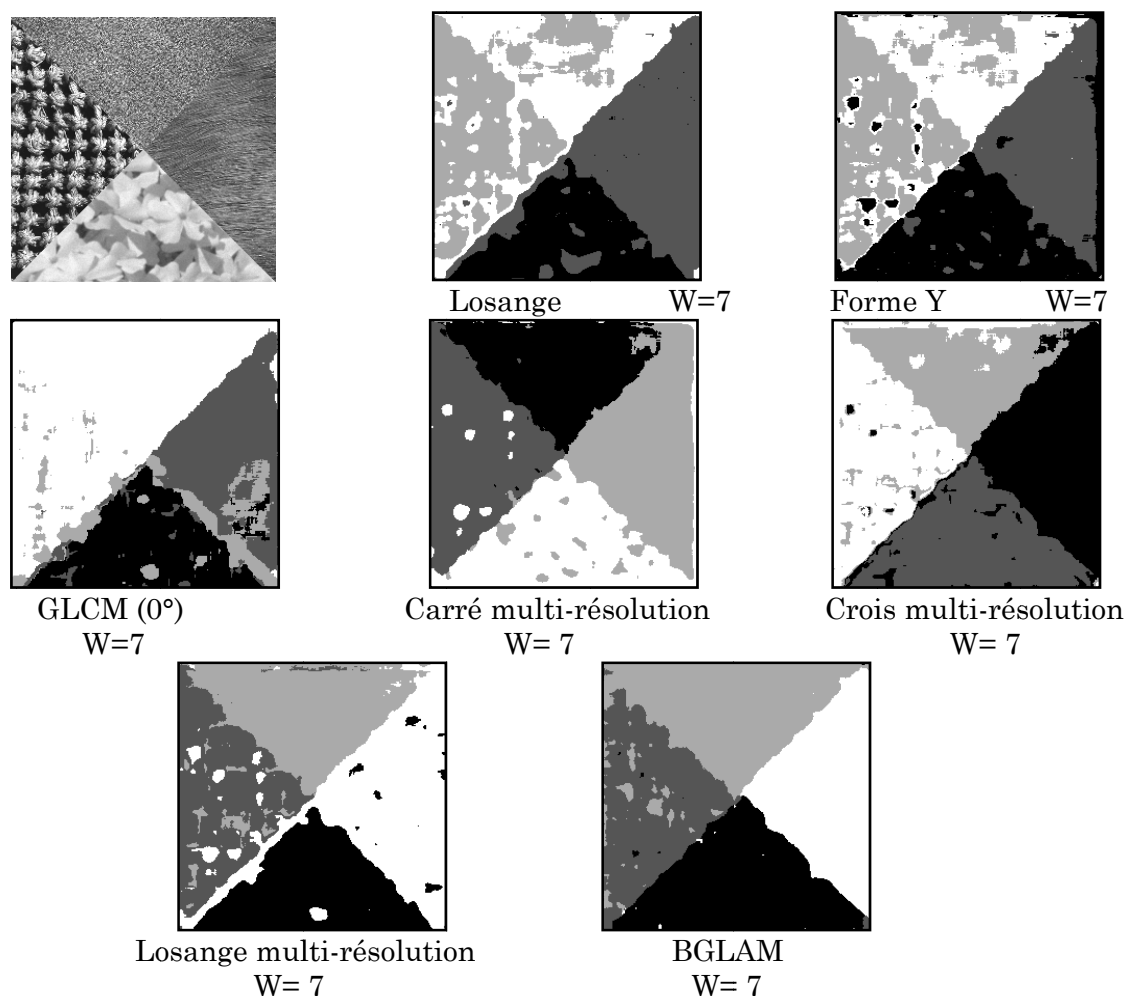


Fig.4.21 : Image synthétique 4 segmentée par les différentes méthodes.

Image synthétique 5

La figure (4.22) montre les différentes images segmentées obtenus sur une autre image synthétique composée de 3 textures différentes par la méthode des matrices de cooccurrence et les matrices Aura.

Les figures 4.19, 4.20, 4.21 et 4.22 montrent que, les résultats de segmentation obtenus par la méthode de la matrice Aura de niveau de gris quelque soit la stratégie utilisée (1 seul élément structurant symétrique ou non symétrique, BGLAMs ou analyse multi-résolution) sont meilleurs par rapport aux résultats obtenus avec la matrice de cooccurrence.

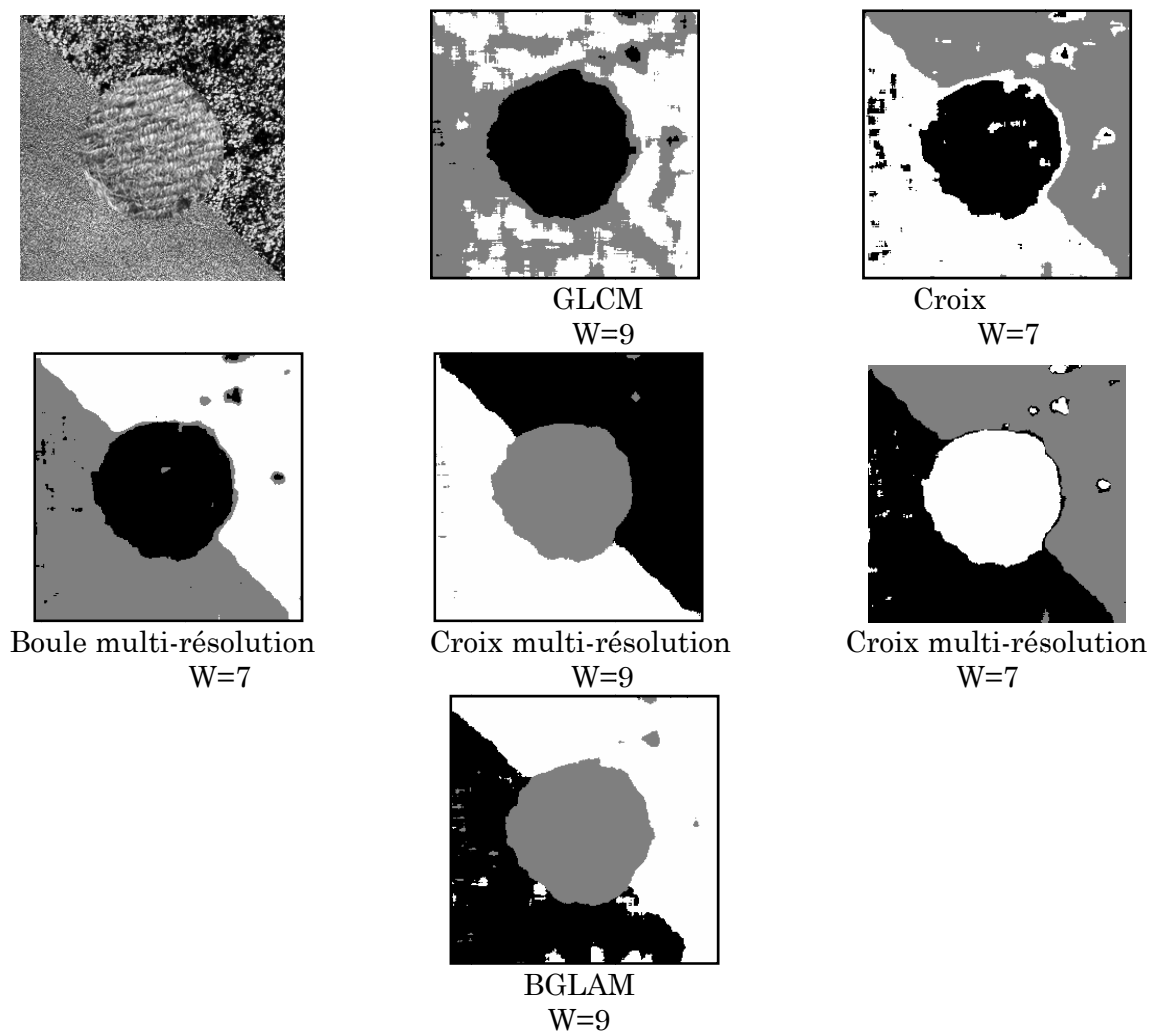


Fig.4.22 : Image synthétique 5 segmentée par les différentes méthodes.

Image réelle 1

La figure (4.23) montre les différentes images segmentées obtenus sur une image réelle décrivant un zèbre par la méthode des matrices de cooccurrence et les matrices Aura. Nous avons considérée que cette image est composée de deux zones texturées. La première correspond au fond de l'image et la deuxième à la robe du zèbre.

Image réelle 2

La figure (4.24) montre les différentes images segmentées obtenus sur une image réelle décrivant un léopard par la méthode des matrices de cooccurrence et les matrices Aura. Nous avons considérée que cette image est composée de trois zones texturées. La première correspond à l'herbe, la deuxième aux arbres et la troisième à la robe du léopard.

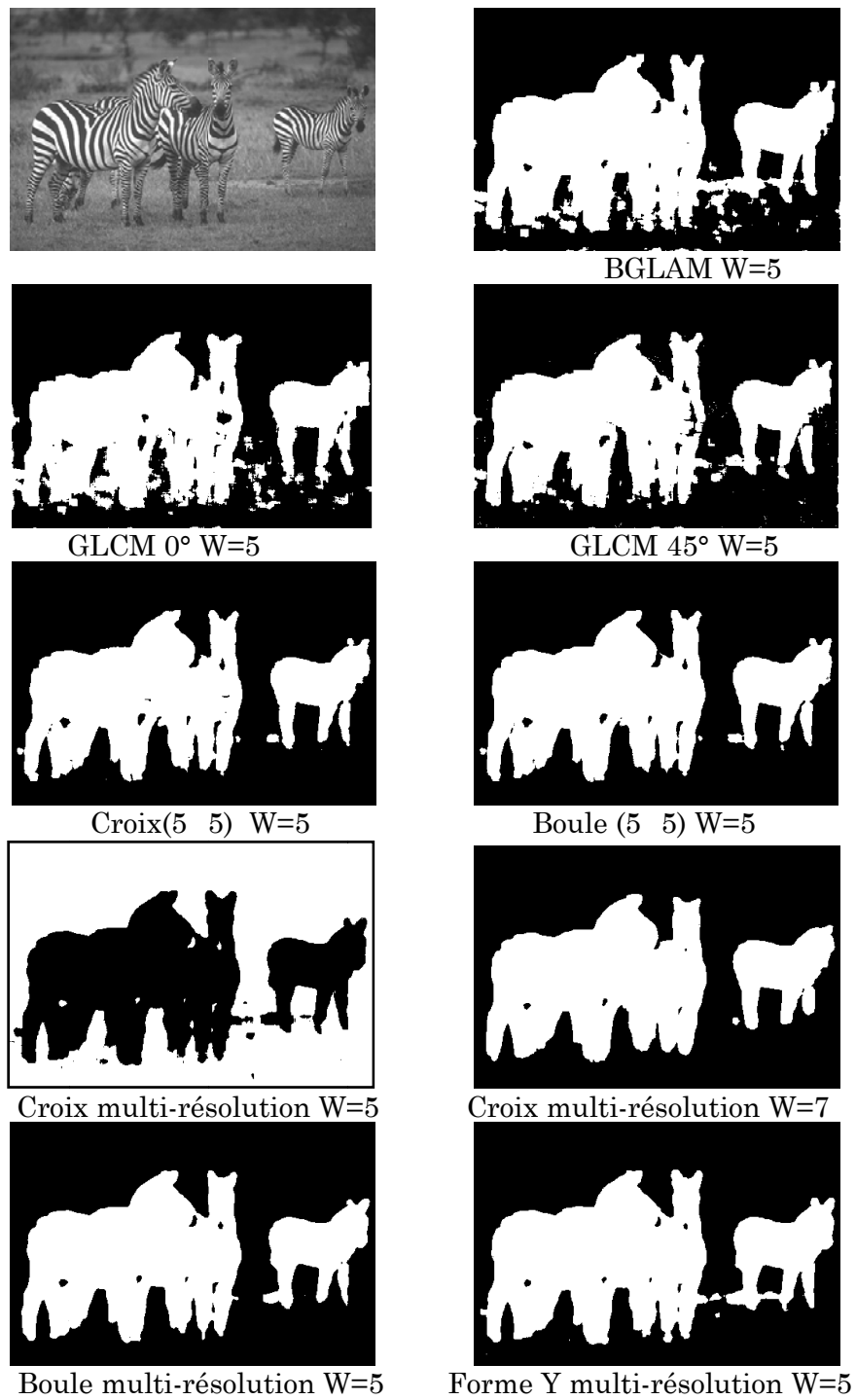


Fig.4.23 : Image réelle 1 segmentée par les différentes méthodes.

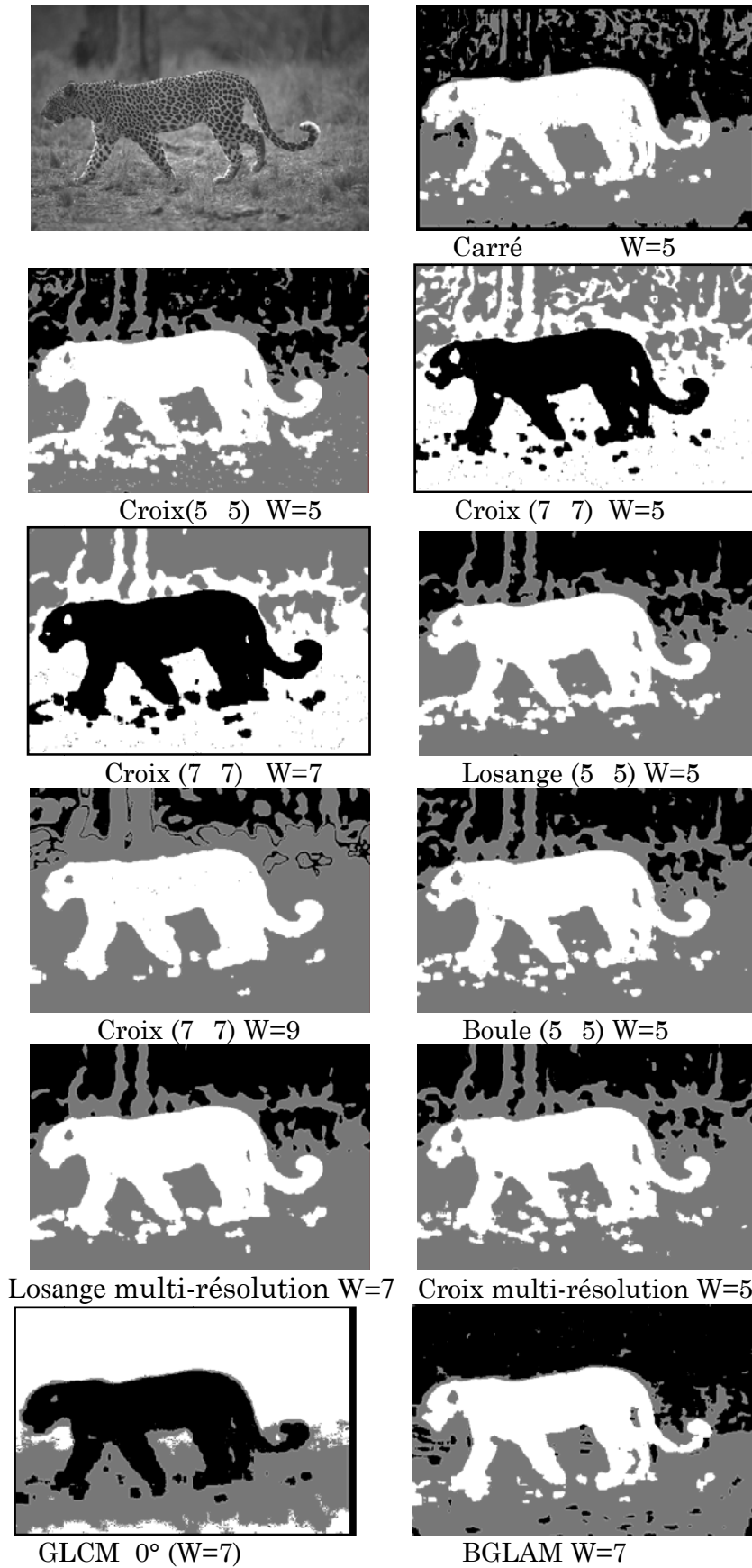


Fig.4.24 : Image réelle 2 segmentée par les différentes méthodes.

Image réelle 3

La figure (4.25) montre les différentes images segmentées obtenus sur une image médicale composée principalement de trois zones texturées.

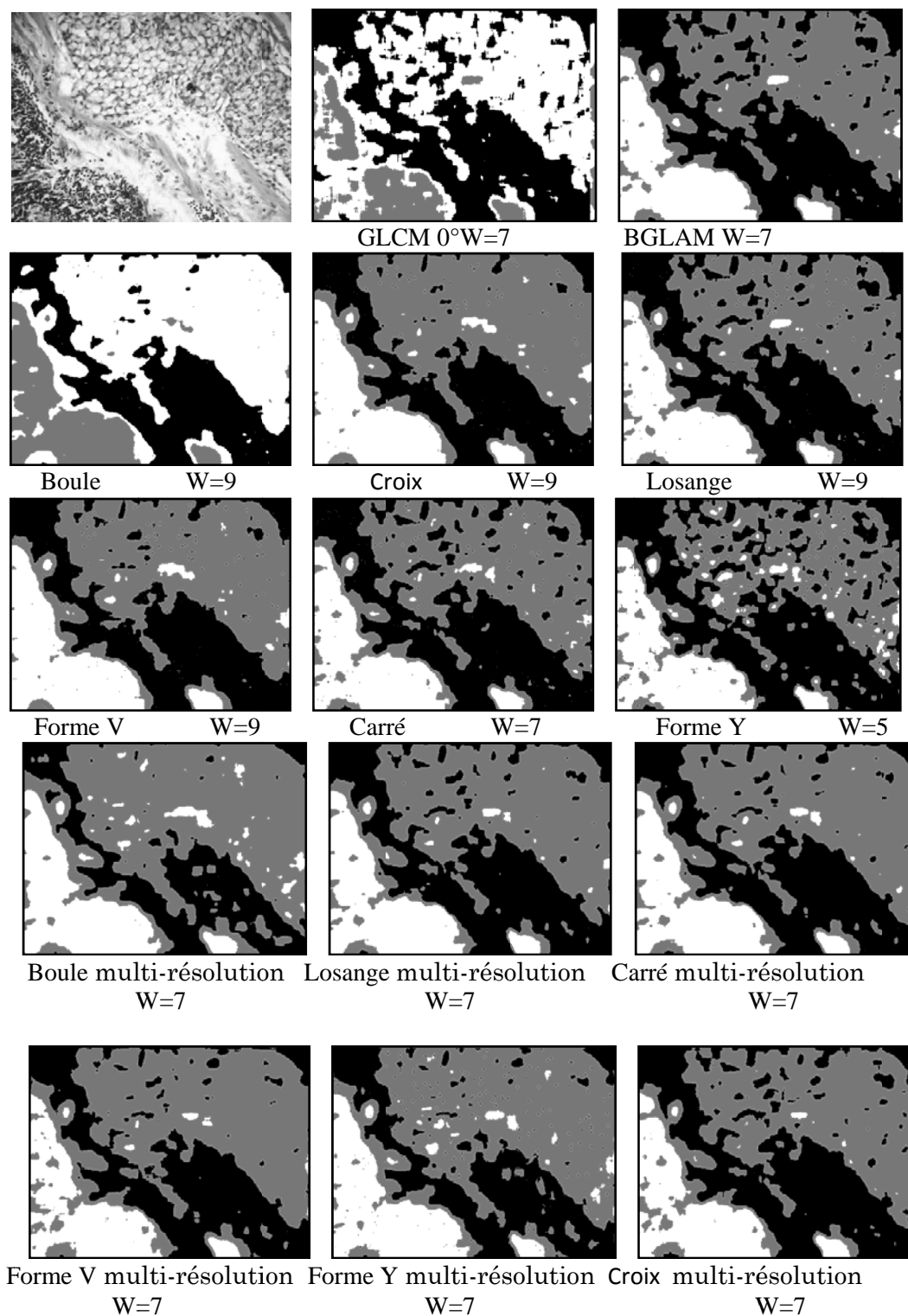


Fig.4.25 : Image réelle 3 segmentée par les différentes méthodes.

Interprétation

Les figures 4.23, 4.24 et 4.25 montrent que, quelque soit l'élément structurant utilisé, les résultats de segmentation obtenus sur des images réelles par la méthode de la matrice Aura de niveau de gris sont très appréciables et meilleures par rapport aux résultats obtenus par la matrice de cooccurrence.

4.7 Conclusion

Nous avons présenté dans ce chapitre quelques résultats de classification des textures à base des matrices Aura. Ces résultats nous a démontré le pouvoir discriminatoire des attributs de texture extraits à partir des matrices Aura. Nous avons également présenté les tests des différentes méthodes implémentées pour le calcul de ces attributs dans le cadre de la segmentation d'images texturées. Une comparaison par rapport au temps d'exécution entre les différentes méthodes de programmation a été menée. Les résultats montrent que la méthode utilisant les structures hybride est plus rapide que la méthode des listes chaînées et beaucoup plus rapide que la méthode traditionnelle.

Quelque soit l'élément structurant utilisé, les résultats de la segmentation à base des matrices Aura en utilisant plusieurs éléments structurants ayant une même forme et des tailles variables (analyse multi résolution) sont beaucoup meilleurs que ceux obtenus avec les matrices des cooccurrences, mais ils sont proches ou légèrement supérieurs à ceux obtenus avec un seul élément structurant ayant une seule forme et une seule taille et avec les BGLAMs.

La taille de la fenêtre de voisinage a une influence directe sur le résultat de la segmentation par les matrices de cooccurrences, les matrices Aura des niveaux de gris et les matrices Aura de base.

Notons également que les résultats de la segmentation peuvent être tributaires des attributs de texture. L'idéal serait d'utiliser une méthode de sélection d'attributs qui permettra une meilleure discrimination entre les différentes textures.

4.1 Introduction

Nous allons présenter dans ce chapitre les résultats de la segmentation d'images texturées basée sur les matrices Aura que nous avons proposée dans le chapitre précédent.

Rappelons que pour segmenter une image, nous avons déterminé un ensemble d'attributs pour chaque pixel. Ensuite, nous avons appliqué la méthode de classification FCM ou K-means afin de regrouper les pixels en zones homogènes. Les attributs de texture de chaque pixel sont extraits à partir des matrices Aura des niveaux de gris des pixels situés à l'intérieure d'une fenêtre de voisinage de taille $(2w + 1)^2$, centrée sur le pixel en question. Le calcul de la matrice Aura dépend de la forme, du type (symétrique ou non) et de la taille des éléments structurants E . Plusieurs stratégies ont été ainsi proposées pour utiliser un ou plusieurs éléments structurants à savoir :

- Un seul élément structurant de forme et de taille fixe.
- Plusieurs éléments structurants à la fois de même forme mais de tailles variables (analyse multi-résolution).
- Des matrices Aura de niveaux de gris de base.

De plus, parallèlement à la techniques classique de programmation du calcul des matrices Aura de niveaux de gris, que nous avons nommé (GLAM), nous avons proposé deux techniques de programmation, l'une est basée sur les listes chaînées (GLALL), l'autre à structure hybride qui combine la notion des listes chaînées et des tables de hachage (GLAHS).

Par conséquent, nous allons analyser dans ce chapitre les performances de ces différentes stratégies et techniques de programmation en fonction de la largeur de la fenêtre de voisinage $(2w + 1)$, du type, de la forme et de la taille de l'élément structurant E . Les résultats de ces méthodes seront également confrontés à ceux obtenus avec les matrices de cooccurrence.

Notons que toutes les méthodes présentées dans notre travail ont été implémentées sous l'environnement Borland C++ version 6 sur un micro ordinateur Acer, ayant une fréquence de 1.66 GHZ, une mémoire vive (RAM) de 1 Go et un disque dur de 64 Go.

Cependant avant de donner les résultats de la segmentation, il est utile d'étudier la capacité des attributs de texture, extraits à partir des matrices Aura à discriminer des textures différentes.

4.2 Classification des images texturées

Afin d'évaluer le pouvoir discriminatoire des attributs de texture extraits à partir des GLAMs, nous avons créé une base d'images de texture à partir des images de l'album de Brodatz. Cette base contient 10 textures différentes (10 classes). Il s'agit des textures D112, D74, D54, D35, D26, D24, D20, D11, D5 et D4 (Fig. 4.1). Celles-ci se différencient par leurs aspects microscopique et macroscopique ayant des propriétés différentes. Chaque classe de texture contient 30 images de taille (64×64), ce qui nous donne au total une base de N=300 échantillons notée \mathcal{I} . Chaque image est ensuite caractérisée par des attributs de texture extraits par la méthode des GLAMs.

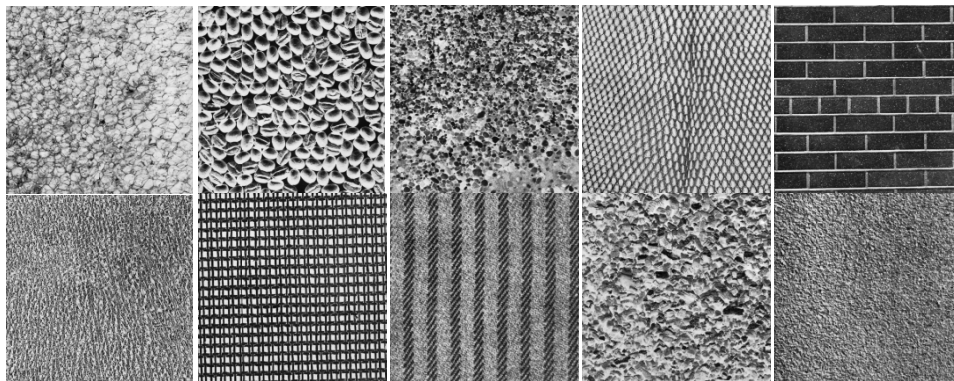


Fig. 4.1 : Base d'images. De gauche à droite et de haut en bas (D112, D74, D54, D35, D26, D24, D20, D11, D5, D4).

L'ensemble des échantillons \mathcal{I} est ensuite divisé en deux sous ensembles : un sous ensemble noté \mathcal{I}_{train} réservée à l'apprentissage et un autre noté \mathcal{I}_{test} réservée au test. Ces deux ensembles sont disjoints et ont la même taille, soit $|\mathcal{I}_{train}| = |\mathcal{I}_{test}| = 150$. L'ensemble \mathcal{I}_{train} constitue la base d'apprentissage, il est utilisé pour construire la règle de décision, tandis que l'ensemble test \mathcal{I}_{test} est utilisé pour évaluer les performances de la règle de décision. Les deux règles de décision employées sont Bayes et K- plus proches voisins. Pour évaluer les performances de ces deux classifieurs, nous avons déterminé le taux de classification à partir des observations de la base de test :

Quelques résultats de la classification des images de la base test sont répertoriés dans le tableau 4.1. Ces résultats sont obtenus en utilisant des éléments structurants de forme et de taille variables. Ils montrent bien que les attributs de texture extrait à partir des matrices Aura des niveaux de gris sont capables de discriminer des textures différentes puisque que le taux de classification est égale ou proche de 100%. Ces résultats montrent également que la forme et la taille des éléments structurants influents légèrement sur les performances du classifieur.

	Bayes	KPPV	Bayes	KPPV	Bayes	KPPV	Bayes	KPPV	Bayes	KPPV
Élément structurant	Carré		Boule		Croix		X		Losange	
5 × 5	100%	100%	97,33%	100%	98%	99,33%	100%	100%	98%	98,66
7 × 7	98,66%	98,66%	100%	100%	96%	98%	100%	96,66%	98,66	99,33%
9 × 9	99,33%	98,66%	98,66%	98,66%	98%	98,66%	98,66%	97,33%	98,66	98%

Tableau 4.1 : Résultats de la classification des images texturées caractérisées à partir des GLAMs

4.3 Segmentation d’images texturées basée sur la matrice Aura de niveaux de gris

Nous allons présenter dans cette section les principaux résultats de segmentation.

4.3.1 Critères d’évaluation

Afin d’évaluer d’une manière objective la performance des différentes méthodes de segmentation proposées, nous avons utilisé le taux de classification et le temps d’exécution comme critères d’évaluation.

Le taux de classification est basé sur des connaissances a priori disponibles sur l’image à savoir le nombre de régions de textures contenues dans l’image et la classe d’appartenance de chaque pixel à une classe de texture. Il est déterminé par le rapport entre le nombre de pixels bien classés et le nombre total des pixels de l’image. Pour cela, nous avons utilisé une image synthétique composée des trois textures tirées de l’album de BRODATZ à savoir D24 représentant le Cuire de veau pressé, D92 : peau de porc et D34 : filet. La taille de cette image est de (256*256) codée en 8 bits (Fig.4.2). Les textures D92 et D24 sont plus au moins fines, par contre, la texture D34 présente une macro- texture. Notons que les deux textures D24 et D92 sont difficilement discernables à l’œil nu.

Le temps d'exécution est utilisé comme critère de comparaison des procédures d'extraction des attributs de texture utilisant la technique classique (GLAM) ainsi que les deux techniques de programmation GLALL et GLAHS.

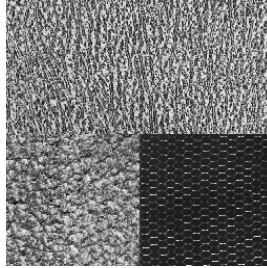


Fig. 4.2 :Image Test.

4.3.2 Attributs texturaux extraits à partir d'une matrice Aura

4.3.2.1 Utilisation d'un seul élément structurant

Les tableaux 4.2, 4.3, 4.4, 4.5 et 4.6 regroupent les valeurs du taux de classification et du temps de calcul des attributs de texture de l'image test obtenus à partir des matrices Aura en fonction de la taille de voisinage et en fonction des techniques de programmation utilisées pour le calcul des attributs. Dans ce cas, les résultats des tableaux sont obtenus en utilisant un seul élément structurant choisis parmi les 5 premiers éléments de la figure 3.12 (Chap. 3) ayant la forme d'un carré, d'une boule, d'une croix, d'une forme X et d'un losange.

Interprétation

Concernant le temps de calcul, les résultats montrent que quelque soit l'élément structurant, la méthode GLAHS est plus rapide par rapport aux méthodes GLAM et GLALL. La méthode GLAM consomme un temps assez important pour le calcul des attributs locaux de textures. Notons que pour la GLAM, ce temps dépend de la taille de l'élément structurant ainsi que la taille de la fenêtre de voisinage, mais aussi du nombre de pixels de l'image.

La taille de la fenêtre de voisinage et la taille de l'élément structurant jouent un rôle important dans l'estimation du temps du calcul. Dans la méthode GLALL, le temps de calcul augmente avec l'augmentation de la taille de la fenêtre de voisinage et de la taille de l'élément structurant. En effet, plus la taille de la fenêtre de voisinage et la taille de l'élément structurant augmentent, plus le nombre de couple différent de niveau de gris augmente. Par conséquent, la taille de la liste devient plus grande. Dans

ce cas, la GLALL présente l'inconvénient de consommer beaucoup du temps lors de la recherche du couple de valeurs de niveaux de gris (g, g') , dans la liste. Elle consomme un temps plus grand que celui de la GLAM.

Dans la méthode GLAHS, l'augmentation de la taille de la fenêtre de voisinage et la taille de l'élément structurant n'influence pas beaucoup sur le temps du calcul car la recherche de couple de niveaux de gris est directe.

L'avantage de la méthode GLAHS par rapport à la méthode GLALL apparaît clairement avec l'augmentation de la taille des fenêtres de voisinage et la taille de l'élément structurant.

Le temps du calcul des attributs de texture est pratiquement différent pour les différents éléments structurants utilisés. Pour les différentes méthodes de programmation, les éléments structurants de forme carré, boule et losange ont pratiquement le même temps du calcul qui reste plus grand à celui obtenu avec les éléments structurants ayant la forme d'une croix ou d'un X. Plus le nombre de pixels d'un élément structurant est grand, plus le temps de calcul augmente.

En ce qui concerne la qualité de la segmentation Les figures 4.3, 4.4, 4.5, 4.6 et 4.7 montrent les courbes de variation des valeurs de taux de classification obtenus avec des éléments structurants de forme carré, croix, boule, X et losange, respectivement. Dans chaque figure, les courbes sont données en fonction de la taille de la fenêtre de voisinage variant de $W=2$ à $W=9$ et en fonction de la taille de l'élément structurant.

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de classification
		GLAM	GLALL	GLAHS	
5×5	W=2	27min27s	2min40s	2min10s	93,45%
	W=3	28min14s	3min3s	2min12s	94,67%
	W=4	38min44s	5min50s	3min42s	96,66%
	W=5	40min34s	9min45s	4min32s	96,96%
	W=6	43min12s	11min36s	5min38s	97,49%
	W=7	48min15s	1h34min7s	6min42s	97,12%
	W=8	53min20s	1h40min	8min19s	96,73%
	W=9	57min22s	1h56min5s	10min4s	96,70%
7×7	W=2	28min34s	5min07s	3min11s	92,76%
	W=3	30min32s	6min14s	4mi4s	94,75%
	W=4	40min12s	12min3s	5min33s	96,35%
	W=5	42min25s	38min3s	7min2s	96,51%
	W=6	46min39s	1h34min23s	8min58s	96,66%
	W=7	48min11s	1h40min42s	11min2s	96,76%
	W=8	54min45s	2h44min32s	13min48s	96,62%
	W=9	59min31s	3h25min7s	15min34s	96,54%
9×9	W=2	29min35s	10min12s	3min46s	92,84%
	W=3	36min55s	19min35s	5min34s	94,29%
	W=4	45min3s	20min4s	7min44s	96,27%
	W=5	47min45s	1h21min4s	10min13s	96,34%
	W=6	50min40s	1h45min34s	13min13s	96,40%
	W=7	52min4s	2h43min42S	16min36s	96,63%
	W=8	57min18s	2h57min47s	20min37s	96,74%
	W=9	1h10min	3h34min38s	22min45s	96,73%
11×11	W=2	16min06s	10min17s	4min44s	92,39%
	W=3	18min45s	15min15s	7min13s	94,32%
	W=4	20min40s	25min04s	10min21s	95,66%
	W=5	56min7s	1h44min5s	13min55s	95,94%
	W=6	57min35s	2h56min28s	18min18s	96,45%
	W=7	1h2min9s	3h12min12s	22min3s	96,42%
	W=8	1h15min3s	3h24min38s	25min13s	96,38%
	W=9	1h30min	3h45min5s	28min40s	95,24%

Tableau 4.2 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme d'un carré.

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de classification
		GLAM	GLALL	GLAHS	
5×5	W=2	14min56s	2min10s	2min3s	93,37%
	W=3	15min14s	3min	2min23s	95,46%
	W=4	15min36s	4min23s	2min47s	96,75%
	W=5	16min16s	7min18s	3min36s	97,24%
	W=6	29min52s	13min35s	5min47s	97,12%
	W=7	34min39s	42min58s	6min40s	97,54%
	W=8	37min45s	56min7s	8min23s	96,87%
	W=9	42min56s	1h10min34s	10min45s	96,73%
7×7	W=2	14min23s	4min30s	3min03s	92,75%
	W=3	15min55s	6min	3min9s	95,60%
	W=4	16min44s	11min37s	5min4s	96,71%
	W=5	17min50s	21min14s	5min12s	97,04%
	W=6	38min24s	35min7s	9min	97,08%
	W=7	40min3s	1h35min	10min27s	96,96%
	W=8	44min16s	1h47min34s	11min45s	96,70%
	W=9	47min34s	2h13min45s	13min25s	96,54%
9×9	W=2	15min35s	11min12s	4min05s	92,84%
	W=3	16min55s	24min15s	5min30s	95,03%
	W=4	17min52s	24min45s	5min33s	96,40%
	W=5	19min35s	46min	7min24s	96,48%
	W=6	37min5s	1h15min10s	13min	96,68%
	W=7	39min42s	2h1min8s	17min24s	96,87%
	W=8	42min12s	2h22min45s	20min41s	97,34%
	W=9	46min38s	2h42min26s	25min23s	96,67%
11×11	W=2	14min55s	8min12s	6min13s	93,11%
	W=3	16min15s	12min44s	7min12s	94,50%
	W=4	17min50s	21min14s	10min17s	95,71%
	W=5	34min32s	1h32min23s	10min35s	96,53%
	W=6	41min56s	1h20min4s	18min21s	97,11%
	W=7	44min39s	1h58min	23min	96,53%
	W=8	48min45s	2h34min5s	25min12s	96,40%
	W=9	56min43s	2h47min23s	28min37s	95,96%

Tableau 4.3 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme d'une boule.

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de Classification
		GLAM	GLALL	GLAHS	
5×5	W=2	13min22s	1min56s	1min56s	92,41 %
	W=3	15min35s	2min26s	2min15s	95,51 %
	W=4	15min26s	3min	2min25s	96,26%
	W=5	16min26s	4min32s	2min55s	97,39 %
	W=6	16min34s	7min38s	3min20s	97,41%
	W=7	17min11s	8min2s	3min40s	97,50%
	W=8	19min45s	10min23s	5min13s	97,37%
	W=9	23min43s	12min57s	6min56s	97,11%
7×7	W=2	13min33s	2min45s	2min10s	91,65%
	W=3	15min43s	3min	2min27s	95,32 %
	W=4	17min55s	5min	2min49s	95,67%
	W=5	16min32s	7min01s	3min30s	97,25%
	W=6	18min50s	9min40s	3min52s	97,32 %
	W=7	19min58s	14min30s	4min32s	97,40%
	W=8	21min34s	17min14s	5min21s	97,21%
	W=9	24min23s	19min57s	7min4s	97,01%
9×9	W=2	14min07s	2min25s	2min08s	92,28%
	W=3	15min36s	2min45s	2min43s	95,06%
	W=4	16min29s	5min48s	3min17s	95,98%
	W=5	17min36s	10min1s	4min	96,83%
	W=6	18min01s	14min40s	4min35s	97,04 %
	W=7	19min08s	22min20s	6min	97,18%
	W=8	21min12s	32min27s	6min28s	96,70%
	W=9	23min54s	37min43s	7min10s	96,36%
11×11	W=2	14min47s	2min55s	2min15s	92,08%
	W=3	15min56s	4min38s	3min07s	93,47%
	W=4	16min34s	10min45s	4min02s	95,55%
	W=5	17min36s	12min46s	4min22s	96,83%
	W=6	18min42s	20min45s	5min42s	96,95%
	W=7	33min31s	34min19s	6min21s	97,08%
	W=8	34min55s	41min34s	6min28s	96,99%
	W=9	37min02s	46min10s	6min54s	96,78%

Tableau 4.4: Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme d'une croix

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de classification
		GLAM	GLALL	GLAHS	
5×5	W=2	14min23s	2min30s	1min50s	91,37%
	W=3	16min3s	2min54s	2min11s	95,11%
	W=4	18min30s	3min29s	2min27s	95,68%
	W=5	19min30s	5min2s	3min	96,04%
	W=6	20min46s	5min12s	3min14s	97,12%
	W=7	29min6s	6min34s	3min44s	97,05%
	W=8	32min9s	10min45s	5min15s	96,73%
	W=9	35min45s	20min5s	6min54s	96,58%
7×7	W=2	13min39s	2min15s	2min10s	92,25%
	W=3	15min52s	3min24s	2min27s	94,83%
	W=4	17min22s	4min59s	2min52s	96,33%
	W=5	18min52s	6min32s	3min33s	96,78%
	W=6	19min56s	9min45s	4min11s	96,82%
	W=7	33min18s	9min56s	4min26s	96,88%
	W=8	37min42s	10min47s	4min34s	96,99%
	W=9	40min50s	13min23s	4min51s	96,84%
9×9	W=2	15min12s	2min45s	2min18s	93,20%
	W=3	16min46s	3min55s	2min57s	95,26%
	W=4	29min21s	5min2s	3min19s	95,87%
	W=5	29min29s	9min45s	4min1s	96,45%
	W=6	34min45s	12min54s	4min46s	96,60%
	W=7	35min37s	25min37s	5min50s	96,74%
	W=8	38min3s	33min32s	6min38s	96,83%
	W=9	42min45s	36min28s	7min12s	96,78%
11×11	W=2	16min57s	3min35s	2min45s	92,48%
	W=3	17min36s	4min58s	3min34s	93,57%
	W=4	20min45s	8min45s	4min04s	95,45%
	W=5	32min34s	10min23s	4min31s	96,11%
	W=6	34min6s	15min33s	5min32s	96,45%
	W=7	35min47s	36min47s	8min6s	96,75%
	W=8	39min53s	43min51s	12min4s	96,62%
	W=9	46min37s	51min03s	14min39s	96,44%

Tableau 4.5 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme X.

Taille de l'élément structurant	Taille de la fenêtre de voisinage	Temps de calcul des attributs de texture			Taux de classification
		GLAM	GLALL	GLAHS	
5×5	W=2	14min56s	2min10s	2mn2s	93,30%
	W=3	15min14s	3min	2mn23s	95,46%
	W=4	15min36s	4min2s	2mn27s	96,75%
	W=5	16min16s	7min10s	3mn30s	97,24%
	W=6	28min51s	9min25s	3min50s	97,39%
	W=7	29min39s	14min29s	5min	97,43%
	W=8	32min18s	16min13s	6min56s	97,34%
7×7	W=9	35min34s	20min56s	8min12s	96,75%
	W=2	17min36s	3min02s	2min20s	93,48%
	W=3	26min02s	5min09s	3min	95,64%
	W=4	26min42s	9min03s	3min49s	96,69%
	W=5	29min29s	21min03s	4min40s	97,10%
	W=6	30min30s	26min08s	5min47s	97,61%
	W=7	44min30s	34min29s	6min57s	97,14%
9×9	W=8	47min24s	39min34s	8min45s	96,95%
	W=9	53min14s	57min35s	11min4s	96,75%
	W=2	18min15s	5min32s	2min48s	92,22%
	W=3	23min05s	9min25s	4min04s	95,31%
	W=4	29min41	18min28s	5min	96,47%
	W=5	31min22s	33min37s	6min25s	96,76%
	W=6	32min41s	55min31s	8min12s	97,08%
11×11	W=7	54min34s	1h12min4s	10min	96,86%
	W=8	58min45s	1h34min15s	12min17s	96,53%
	W=9	1h3min38s	2h43min49s	15min28s	96,37%
	W=2	20min12s	7min22s	3min32s	92,86%
	W=3	26min45s	11min34s	5min02s	94,64%
	W=4	29min55s	20min44s	6min40s	95,90%
	W=5	32min45s	1h14min	8min32s	96,57%
11×11	W=6	40min51s	1h36min	11min2s	96,57%
	W=7	59min14s	1h45min45s	13min37	96,36%
	W=8	1h14min6s	2h15min19s	15min23s	96,24%
	W=9	1h20min	2h34min15s	17min29s	95,76%

Tableau 4.6 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec un élément structurant ayant la forme d'un losange.

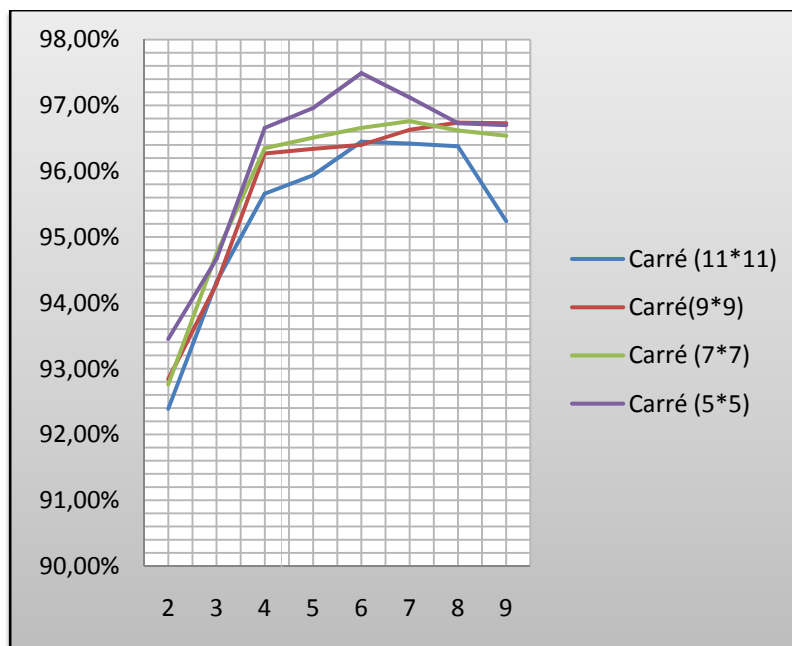


Fig. 4.3: Taux de classification obtenu en utilisant l'élément structurant ayant la forme d'un carré

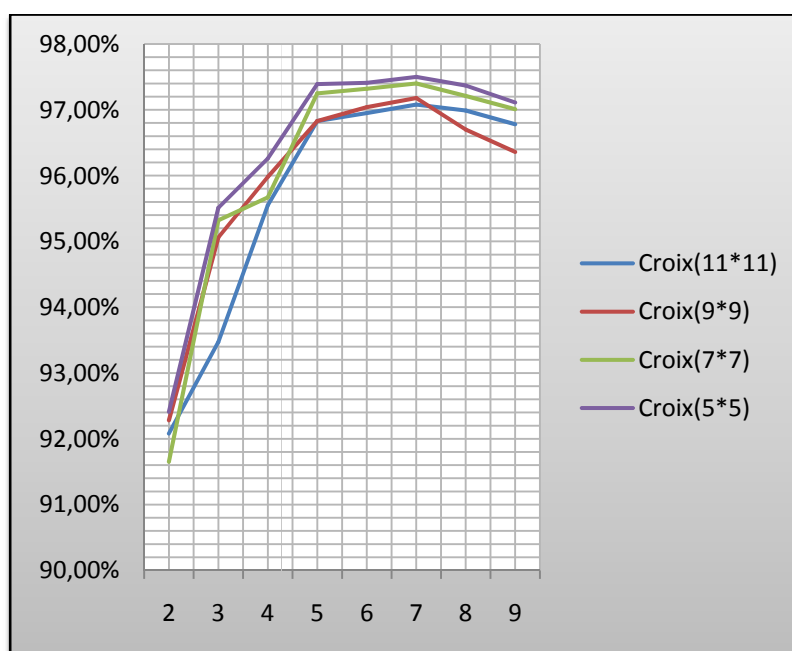


Fig. 4.4 : Taux de classification obtenu en utilisant l'élément structurant ayant la forme d'une croix

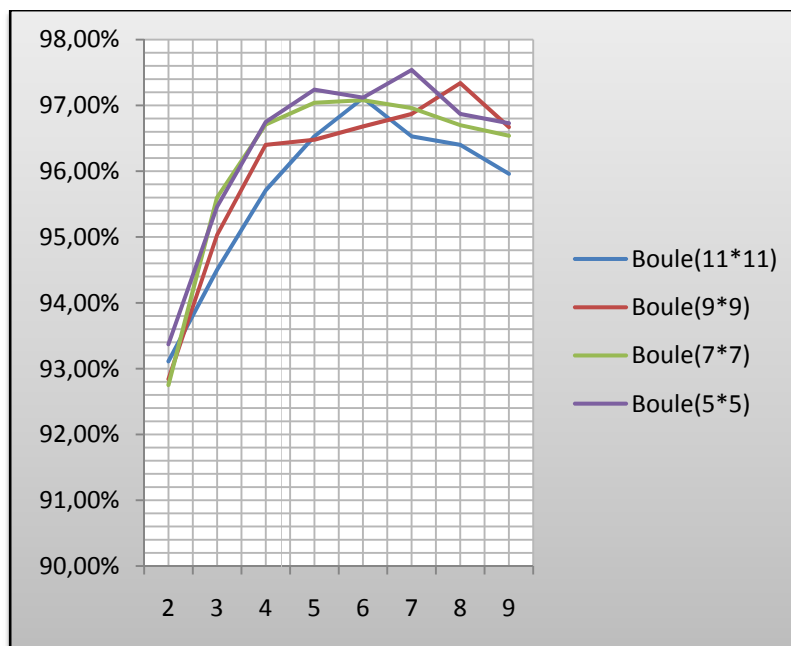


Fig. 4.5 : Taux de classification obtenu en utilisant l'élément structurant ayant la forme d'une boule

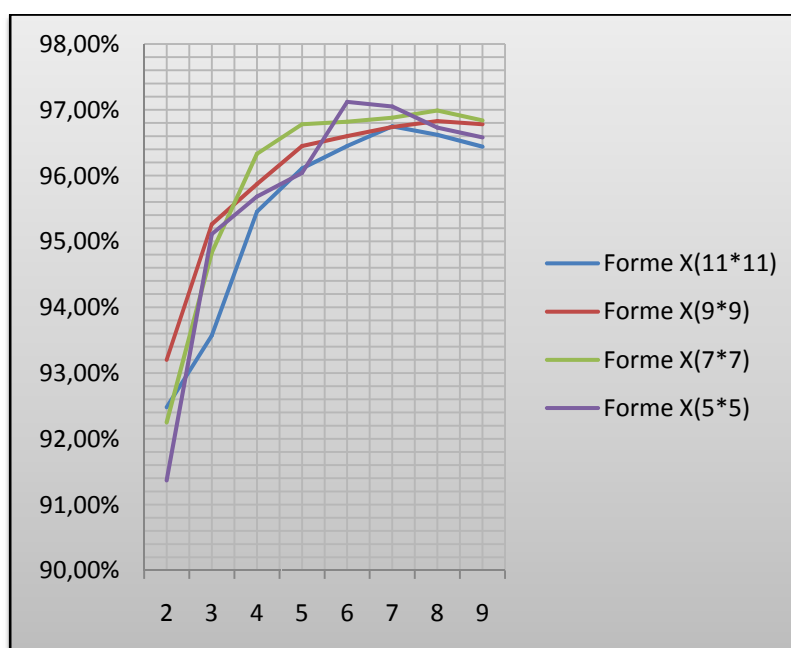


Fig. 4.6: Taux de classification obtenu en utilisant l'élément structurant ayant la forme X

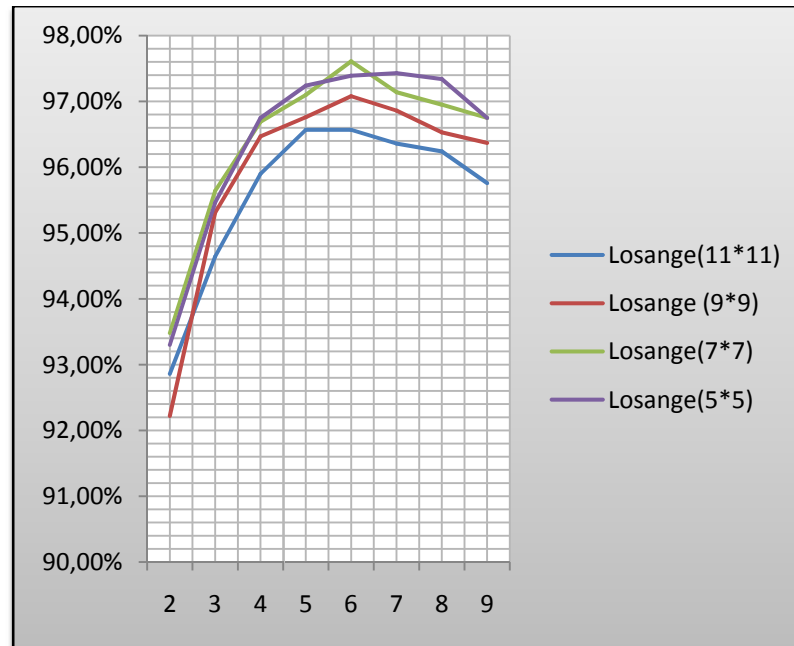


Fig. 4.7 : Taux de classification obtenu en utilisant l'élément structurant ayant la forme d'un Losange

Interprétation

A travers les figures précédentes, nous pouvons remarquer que les courbes de variation du taux de classification varient quasiment de la même manière en fonction de la taille de la fenêtre de voisinage et ce quelque soit la forme et la taille de l'élément structurant utilisé. Plus la taille de la fenêtre de voisinage augmente, plus le taux de classification augmente pour atteindre une valeur maximale. Cependant, à partir d'une certaine taille, le taux de classification diminue, ceci est dû à une mauvaise localisation des frontières entre les différentes textures. Ainsi, pour l'élément structurant de la forme d'un carré (Fig.4.3) le taux maximal 97,49% est obtenu avec une taille (5 et $W=6$. Avec la croix (Fig.4.4) le taux de classification maximal 97,50% est obtenu avec un élément de taille (5 et $W=7$. Pour l'élément structurant ayant la forme d'une boule (Fig.4.5), le taux maximal est 97,54%, il est obtenu avec une taille de (5 et $W=7$. L'élément structurant de la forme X et de taille (5 donne un taux maximal de 97,12%, pour une fenêtre de voisinage ayant $W=6$ (Fig.4.6). Un taux maximal de 97,61% est obtenu avec l'élément structurant ayant la forme d'un losange et une taille de (7 avec une fenêtre de voisinage caractérisée par $W=6$ (Fig.4.7).

Les figures 4.8, 4.9, 4.10 et 4.11 montrent par contre les courbes de variation du taux de classification en fonction de la taille de la fenêtre de voisinage W et en fonction

de l'élément structurant. Dans chaque figure, les courbes correspondent à une seule taille de l'élément structurant.

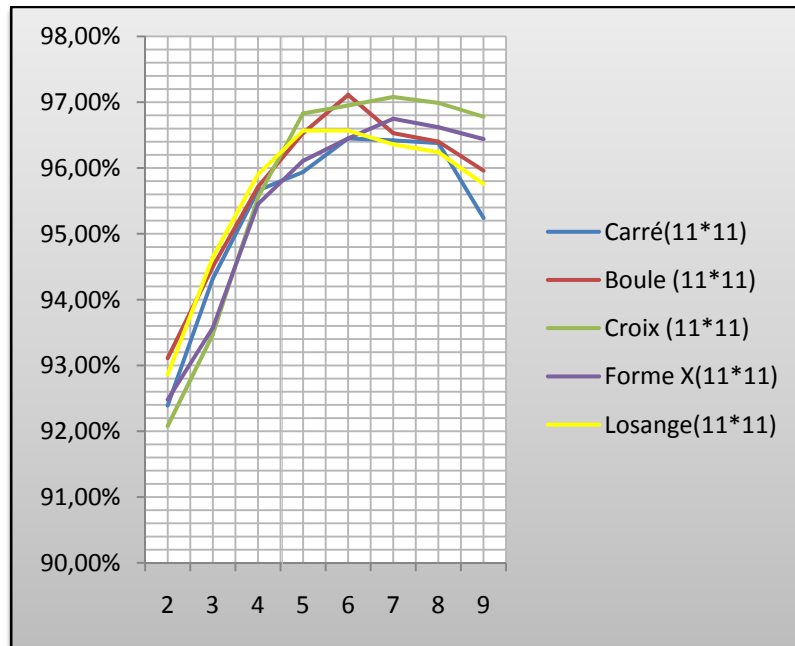


Fig. 4.8 : Taux de classification en fonction de l'élément structurant de taille (11*11)

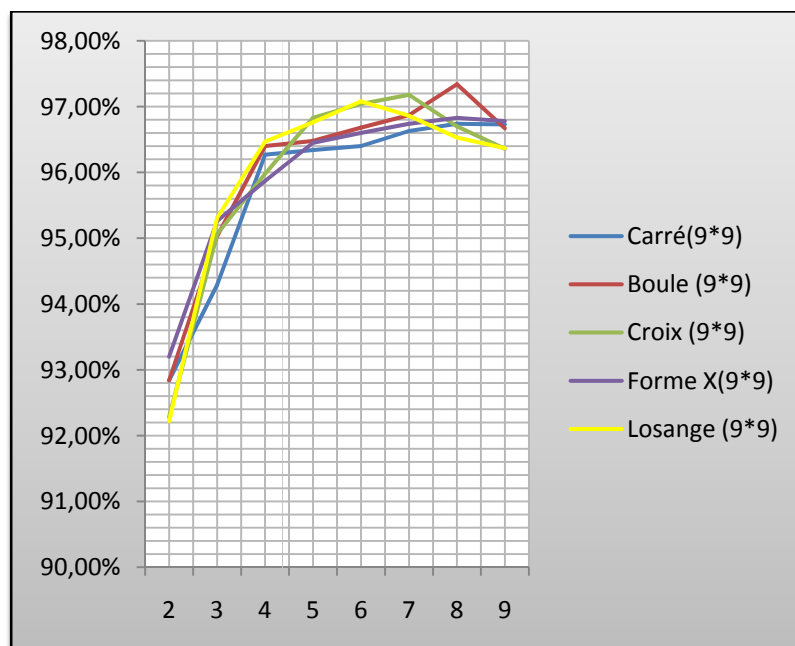


Fig. 4.9: Taux de classification en fonction de l'élément structurant de taille (9*9)

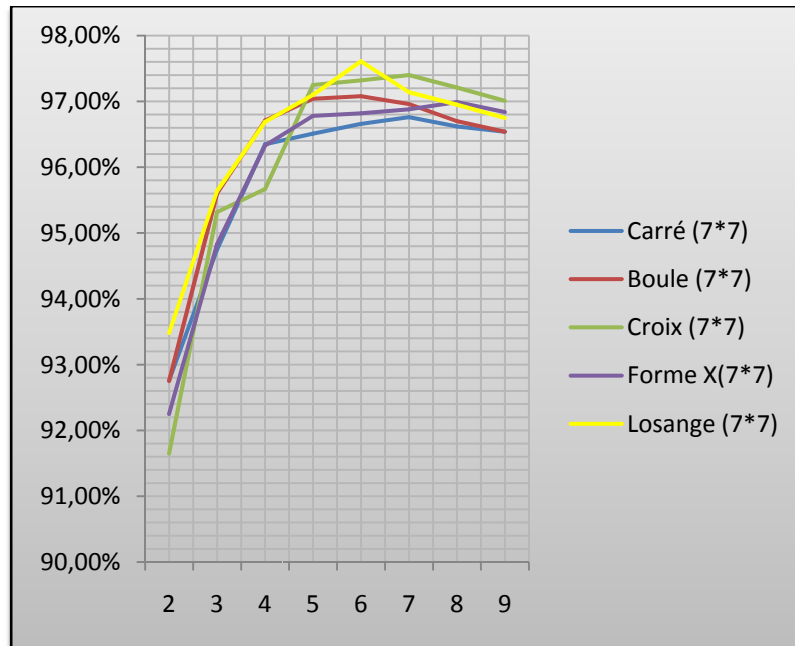


Fig. 4.10 : Taux de classification en fonction de l'élément structurant de taille (7*7)

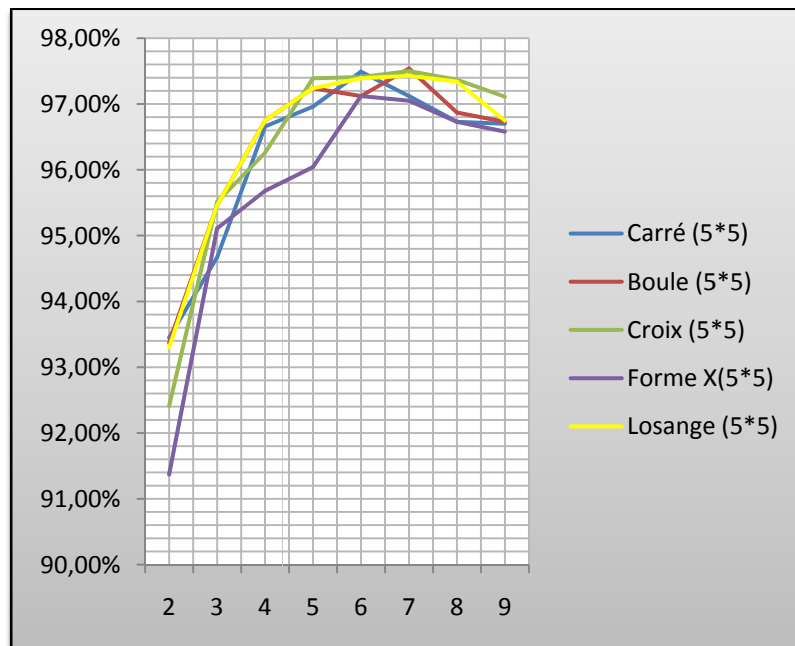


Fig. 4.11 : Taux de classification en fonction de l'élément structurant de taille (5*5)

Ces courbes montrent également que le taux de classification varie quasiment de la même manière en fonction de la taille de la fenêtre de voisinage et ce quelque soit la forme et la taille de l'élément structurant utilisé. Ainsi pour l'image test de la figure 4.2, le meilleur résultat est obtenu en utilisant l'élément structurant de forme losange de taille pour une fenêtre de voisinage . Ces paramètres constituent un bon compromis entre la segmentation et la localisation des frontières.

Tous les résultats donnés jusqu'à présent sont obtenus en utilisant des éléments structurants symétriques. Les mêmes conclusions ont été obtenues avec des éléments structurants non symétriques. Ceux-ci n'ont pas été incorporés par soucis d'espace. Cependant, on peut voir sur la figure 4.12 quelques résultats de la segmentation de l'image test en fonction des différents éléments structurant symétriques et asymétriques.

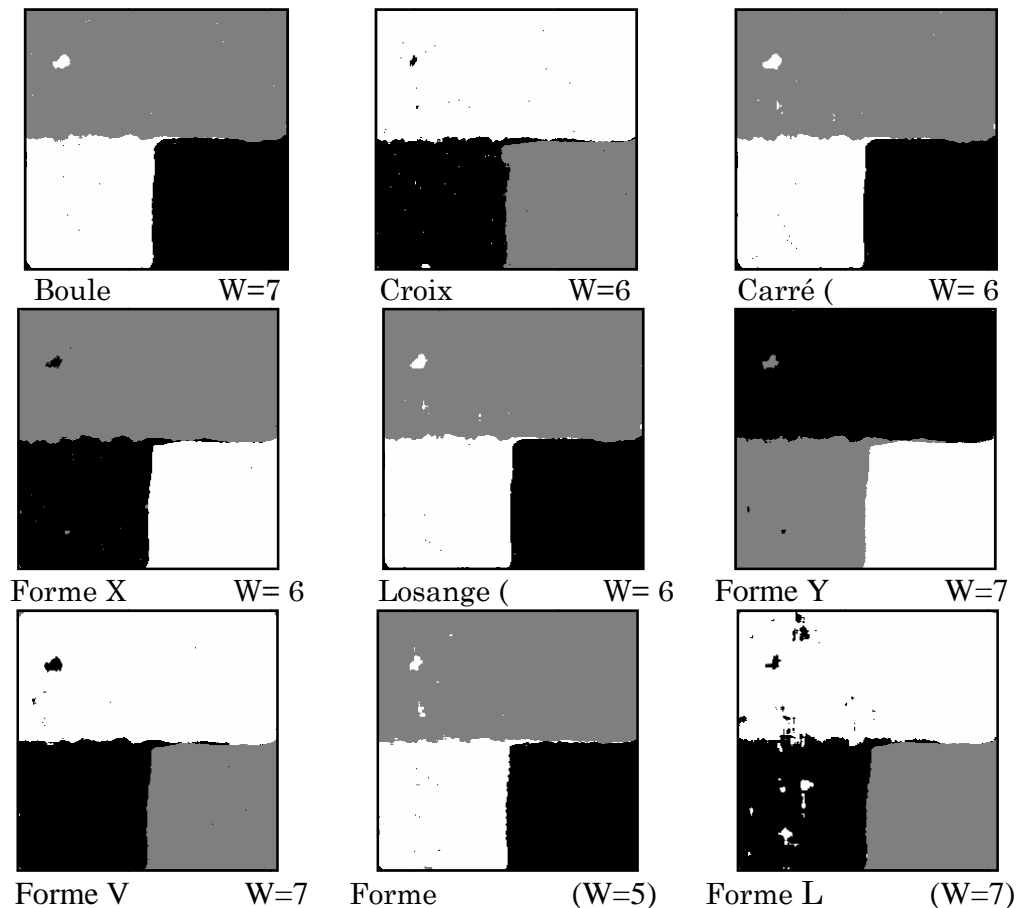


Fig. 4.12 : Images segmentées en utilisant les attributs de texture extraits à partir des matrices Aura

4.3.2.2 Utilisation de plusieurs structures de voisinage de même forme et de tailles variables.

Les tableaux 4.7, 4.8, 4.9, 4.10 et 4.11 donnent les valeurs du taux de classification et du temps de calcul des attributs de texture de l'image test obtenus avec les attributs de textures extraits à partir des matrices Aura en fonction de la taille de voisinage. Ces résultats sont obtenus en utilisant plusieurs éléments structurants à la fois ayant la même forme avec des tailles variables (analyse multi résolution).

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	38min7s	93,43%
W=3	40min38s	96,11%
W=4	49min13s	96,66%
W=5	52min1s	96,93%
W=6	1h10min02s	97,50%
W=7	1h21min30s	97,58%
W=8	1h40min48s	96,90%
W=9	1h57min5s	96,87%
W=10	2h23min22s	96,70%
W=11	2h36min16s	96,58%

Tableau 4.7 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme d'un carré mais des tailles variables.

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	34min12s	93,46%
W=3	35min21s	95,90%
W=4	49min56s	97,15%
W=5	56min2s	97,48%
W=6	58min43s	97,58%
W=7	1h4min44s	97,72%
W=8	1h42min52s	97,69%
W=9	1h50min5s	97,61%
W=10	2h7min6s	97,44%
W=11	2h26min45s	96,55%

Tableau 4.8 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme d'une boule mais des tailles variables.

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	26min51s	93,21%
W=3	30min40s	95,97%
W=4	32min57s	96,97%
W=5	36min	97,56%
W=6	38min9s	97,62%
W=7	41min45s	97,64%
W=8	49min14s	97,64%
W=9	49min58s	97,68%
W=10	53min22s	97%
W=11	58min42s	96,85%

Tableau 4.9 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme d'une croix mais des tailles variables.

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	29min27s	92,91%
W=3	31min28s	94,71%
W=4	36min33s	96,59%
W=5	38min02s	96,70%
W=6	39min16s	97,16%
W=7	42min34s	97,09%
W=8	50min43s	96,92%
W=9	51min12s	96,89%
W=10	54min25s	96,73%
W=11	59min3s	96,59%

Tableau 4.10 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme X mais des tailles variables.

Taille de la fenêtre de voisinage	Temps de calcul de la GLAHS	Taux de classification
W=2	31min25s	93,96%
W=3	34min13s	95,71%
W=4	39min12s	96,73%
W=5	44min07s	96,97%
W=6	52min47s	97,48%
W=7	57min15s	97,54%
W=8	1h07min31s	97,90%
W=9	1h18min30s	97,61%
W=10	1h26min14s	97,64%
W=11	1h32min42s	97,64%

Tableau 4.11 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la forme d'un losange mais des tailles variables.

Interprétation

Comme nous l'avons évoqué précédemment, les deux méthodes GLAM et GLALL consomment un temps assez long lors du calcul des attributs de texture. Ce temps devient excessivement long lorsqu'on utilise en même temps plusieurs éléments structurants ayant tous la même forme mais des tailles variables. A titre d'exemple, pour un élément structurant de forme du losange de taille variable et $W=8$, la GLAM consomme 2h52min 33s et la GLALL 7h23min25s. C'est la raison pour laquelle nous avons relevé les temps de calcul uniquement pour la méthode GLAHS. Ces temps de calcul restent très courts par rapport à ceux obtenus par les deux autres méthodes GLAM et GLALL. Cependant, il y a lieu d'indiquer que contrairement à l'utilisation d'un seul élément structurant de taille fixe, la taille de la fenêtre de voisinage influe sur le temps du calcul car les pixels de la fenêtre de voisinage sont parcourus autant de fois que le nombre de taille de l'élément considéré.

La figure (4.13) montre la variation du taux de classification en fonction de l'élément structurant de la taille de la fenêtre de voisinage.

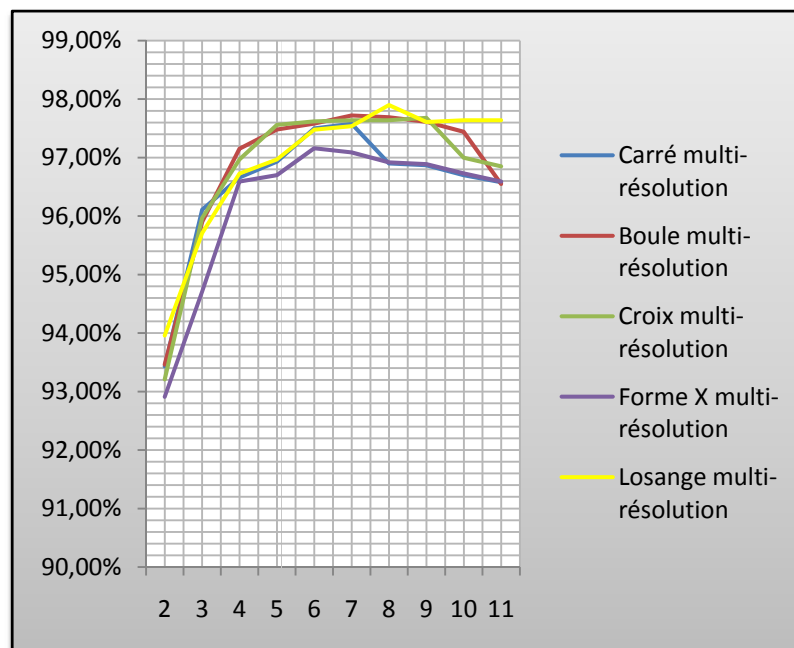


Fig. 4.13 : Taux de classification en fonction de l'élément structurant de taille variable

On remarque que comme pour un seul élément structurant, dans l'analyse multi-résolution quelque soit la forme de l'élément structurant, plus la taille de la fenêtre de voisinage augmente, plus le taux de classification augmente pour atteindre un taux maximal ($W=7$ pour le carré et la boule, $W=9$ pour la croix, $W=6$ pour la forme X et $W=8$ pour le losange). Cependant, à partir de cette taille le taux de classification diminue à cause d'une mauvaise localisation des frontières entre les différentes textures. Cependant, ce taux de classification (97.90%) reste supérieur au meilleur résultat (97.61%) obtenu avec un seul élément structurant de taille (7×7) qui a la même forme losange avec $W=6$. Il y'a lieu également de noter que, pour cette image test, la différence entre ces deux taux n'est pas vraiment significative. Elle est toute de même très expressive pour d'autres images comme nous le verrons par la suite. Comme, il est également remarqué que ce n'est pas toujours le même élément structurant qui donne le meilleur résultat.

La figure 4.14 montre les meilleurs résultats de la segmentation de l'image test en utilisant plusieurs éléments structurants de la même forme et des tailles variables.

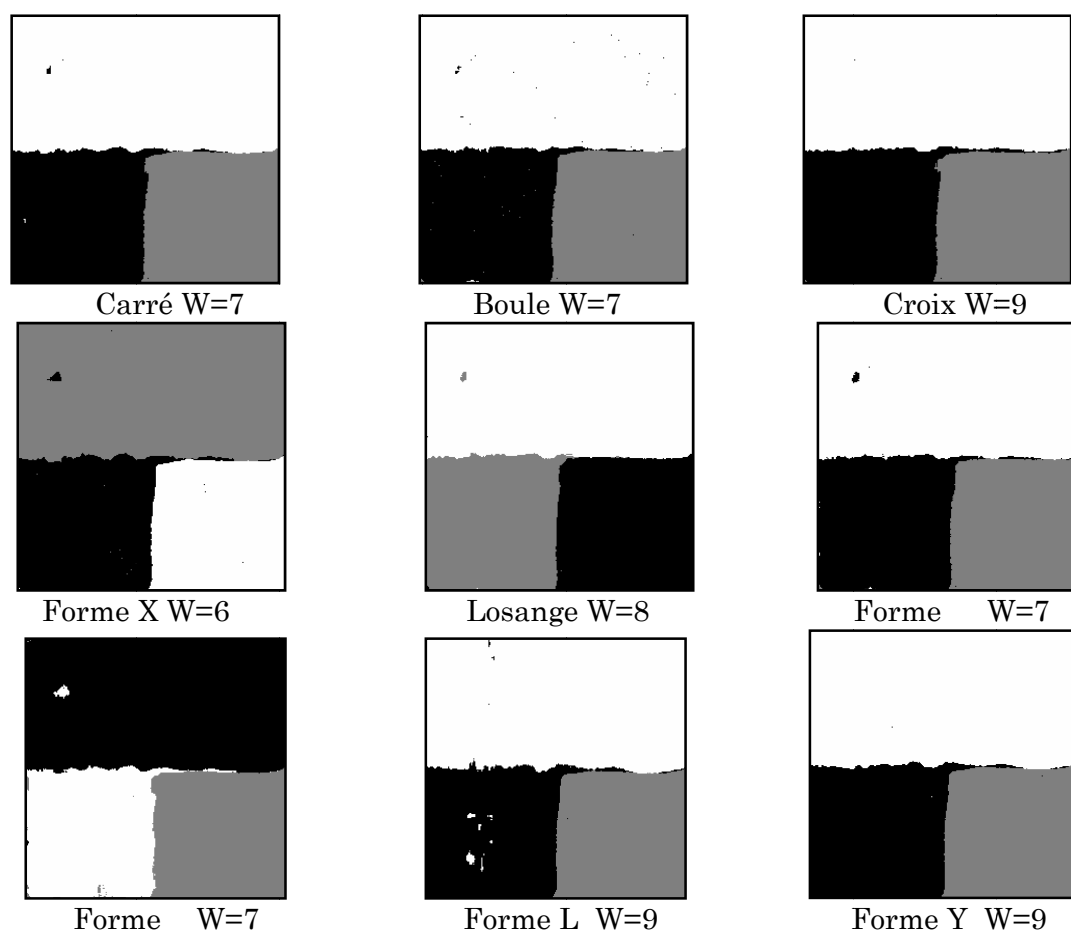


Fig. 4.14 : Images segmentées en utilisant les attributs de textures extraits à partir des matrices Aura avec plusieurs éléments structurants ayant tous la même forme mais des tailles variables.

Interprétation

En comparant ces résultats par rapport à ceux obtenus par un seul élément structurant, nous remarquons que les taches isolées ont pratiquement disparues avec plusieurs éléments structurants de la même forme et des tailles variables (analyse multi résolution).

4.4 Utilisation des matrices Aura de base

Le tableau 4.12 donne les valeurs du taux de classification et du temps de calcul des attributs de texture de l'image test obtenus avec les attributs de textures extraits à partir des matrices Aura de base BGLAHS en fonction de la taille de voisinage.

La figure 4.15 illustre les résultats de la segmentation de l'image test en utilisant les matrices Aura de base.

Taille de la fenêtre de voisinage	Temps de calcul de la BGLAHS	Taux de classification
W=3	1h11min56s	93,31%
W=5	1h14min34s	96,09%
W=7	1h14min40s	96,30%
W=9	1h14min56s	96,11%

Tableau 4.12 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des matrices Aura de base.

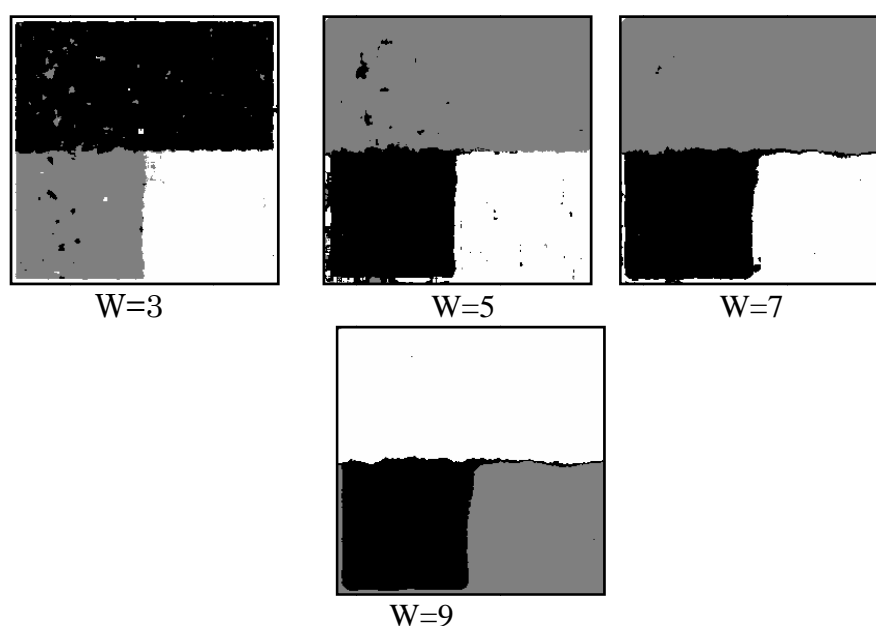


Fig. 4.15 : Images segmentées en utilisant les attributs de textures extraits à partir des BGLAMs.

Interprétation

On constate également, que la taille de la fenêtre de voisinage joue un rôle important dans la segmentation d'image par les attributs de texture issus des matrices Aura de base. Le tableau 4.12 montre que, plus la taille de voisinage augmente plus le taux de classification augmente pour atteindre un taux maximal pour W=7. Cependant, pour cette image test, le taux de classification obtenu avec les BGLAMs reste légèrement inférieur à celui obtenu avec les GLAMs que ce soit pour un seul élément structurant ou pour l'analyse multi résolution.

4.5 Comparaison avec les attributs texturaux extraits à partir d'une matrice de cooccurrence

Afin de comparer les résultats de la segmentation obtenus en utilisant les attributs de textures extraits à partir des matrices Aura de niveaux de gris et ceux obtenus par les attributs extraits à partir des matrices de cooccurrences, nous avons utilisé une image synthétique (1) donnée par la figure 4.16. Elle est constituée de quatre textures différentes.

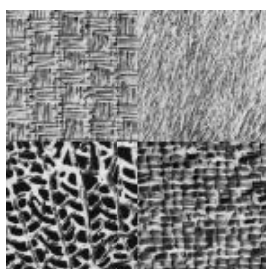


Fig. 4.16 : Image synthétique 1

Le tableau 4.13 regroupe les valeurs du taux de classification de l'image de la figure 4.16 obtenus avec les attributs de textures extraits à partir des matrices de cooccurrence (direction 0°) et des matrices Aura (élément structurant de forme carré de taille W) en fonction de la taille de voisinage.

Taille de la fenêtre de voisinage	Taux de classification	
	GLCM	GLAM
W=5	56,43%	90,12%
W=6	56,75%	90,99%
W=7	58,54%	90,94%
W=8	54,23%	90,42%
W=9	45,32%	87,62%

Tableau 4.12 : Résultats de la segmentation de l'image test utilisant les attributs de textures extraits à partir des différentes méthodes.

La figure 4.17 montre les résultats de la segmentation de l'image de la figure 4.16 en utilisant les six attributs locaux extraits à partir des matrices de cooccurrences pour les différentes tailles de la fenêtre de voisinage en prenant en compte une seule direction.

La figure 4.18 montre les résultats de la segmentation de l'image de la figure 4.16 en utilisant les attributs de textures extraits à partir des matrices Aura en fonction de l'élément structurant.

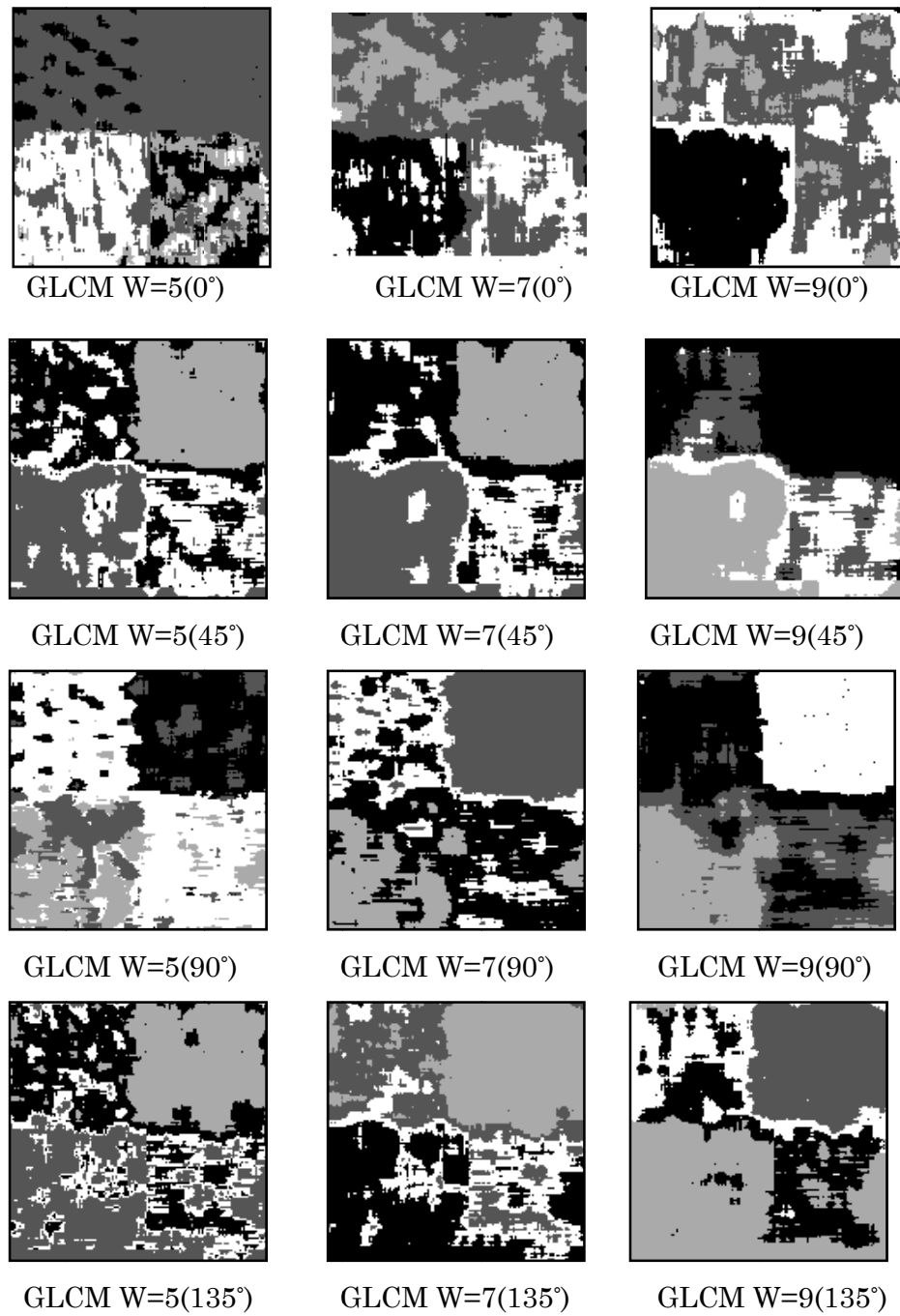


Fig. 4.17 : Images segmentées en utilisant les attributs de textures extraits à partir des matrices de cooccurrence pour les différentes directions

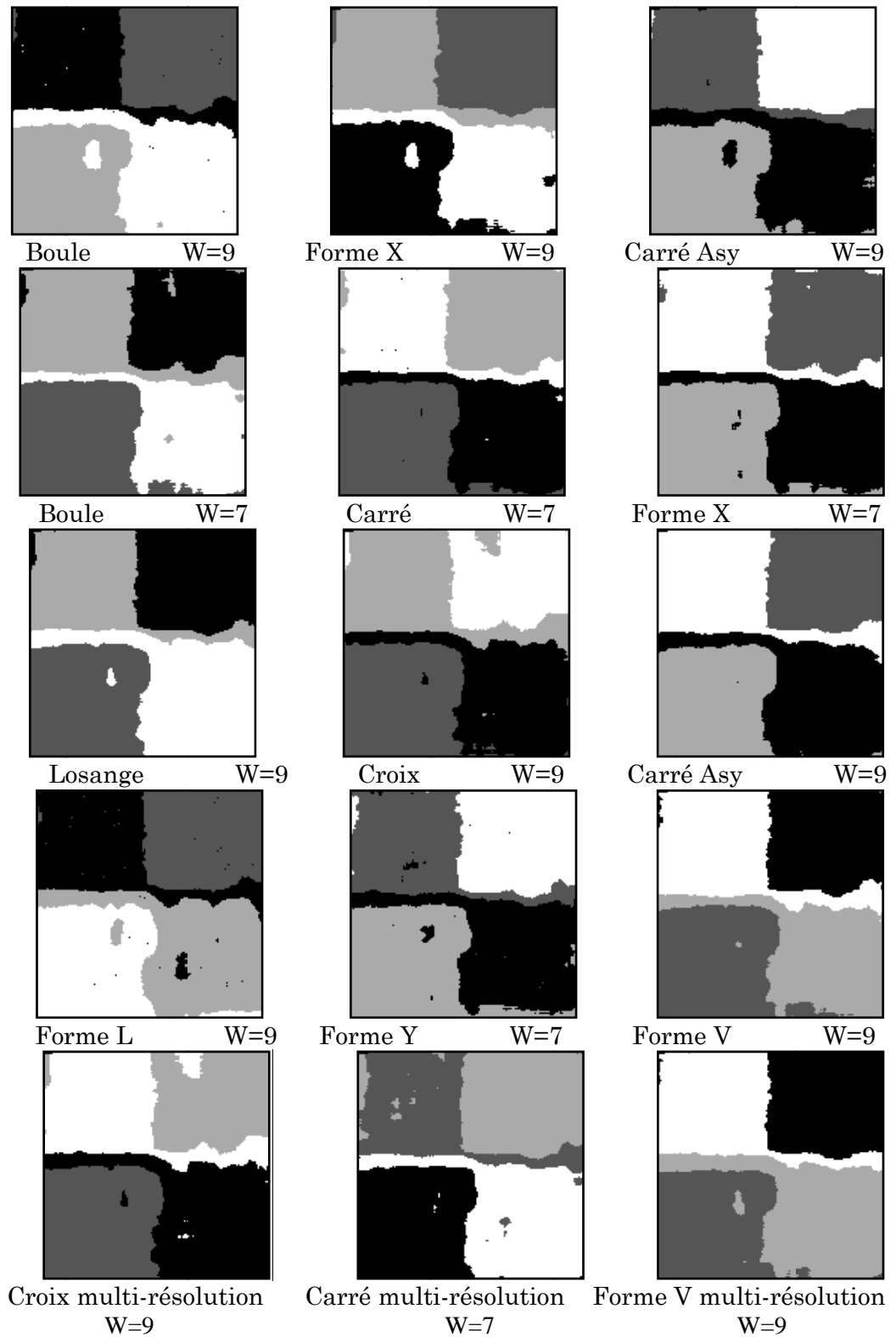


Fig. 4.18 : Images segmentées en utilisant les attributs de textures extraits à partir des matrices Aura pour les différents éléments structurants.

Interprétation

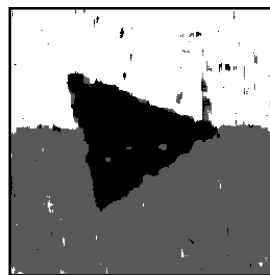
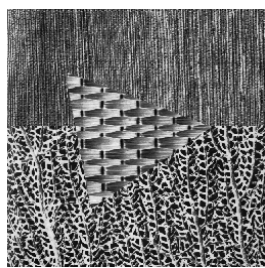
Quelque soit l'élément structurant utilisé, les résultats de la segmentation à base des matrices Aura sont beaucoup meilleurs que ceux obtenus avec les matrices des cooccurrences même si des problèmes aux frontières des textures persistent.

4.6 Images synthétiques et réelles

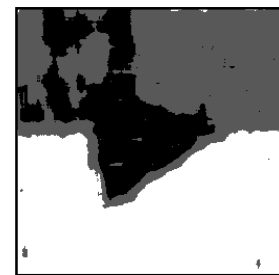
Nous avons également appliqué la méthode proposée sur d'autres images synthétiques et sur des images réelles.

Image synthétique 2

La figure (4.19) montre les différentes images segmentées obtenus sur une autre image synthétique par la méthode des matrices de cooccurrence et les matrices Aura



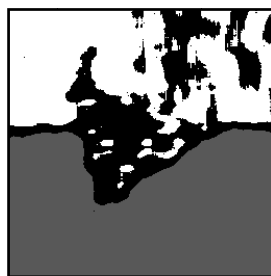
GLCM $w=5(0^\circ)$



Croix de taille
W=9



Boule de taille
W= 9



Carré de taille
W=9



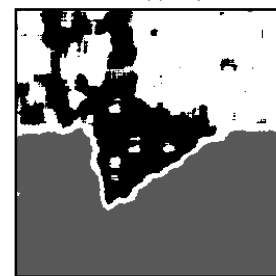
Forme L
W=7



Losange de taille
W=9



Forme X de taille
W= 9



Forme de taille
W=7

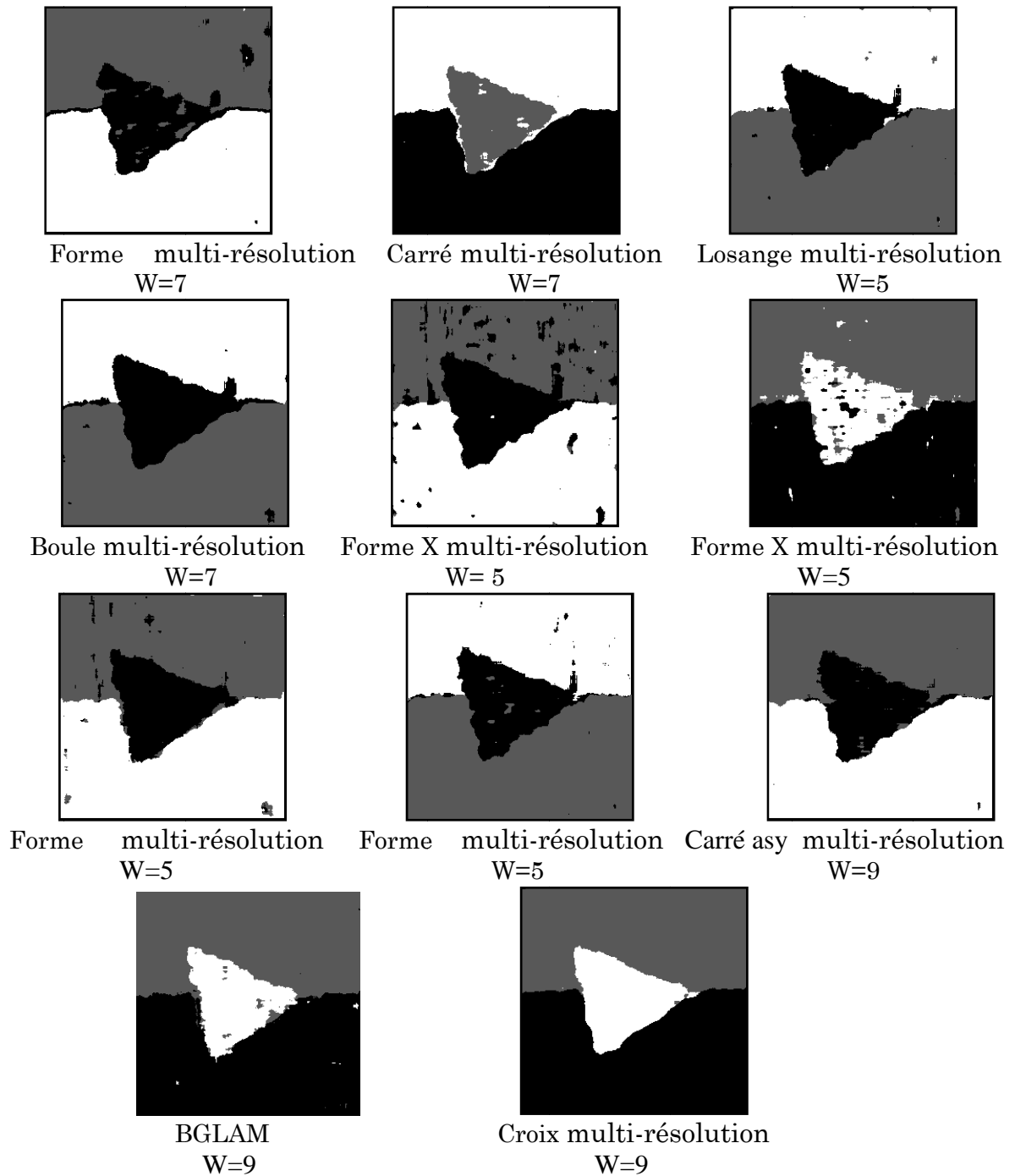


Fig.4.19 : Image synthétique 2 segmentée par les différentes méthodes.

Interprétation

Pour cette image, nous pouvons remarquer que l'utilisation d'un seul élément structurant peut s'avérer insuffisant (Fig. 4.19). En effet, les trois zones texturées sont mises en évidence d'une certaine manière. Mais ces résultats restent moins bons par rapport à ceux obtenus par l'utilisation de plusieurs éléments structurants de la même forme et de tailles différentes (analyse multi-résolution) car plus le nombre d'éléments

est élevé plus l'information texturale sera riche. Contrairement à l'image précédente, les résultats obtenus sur cette image synthétique montre que les BGLAMs, la GLAM avec un seul élément non symétrique et la GLCM sont plus performantes par rapport à la GLAM avec un seul élément symétrique.

Image synthétique 3

La figure (4.20) montre les différentes images segmentées obtenus sur une image synthétique composée de 5 textures différentes par la méthode des matrices de cooccurrence et les matrices Aura

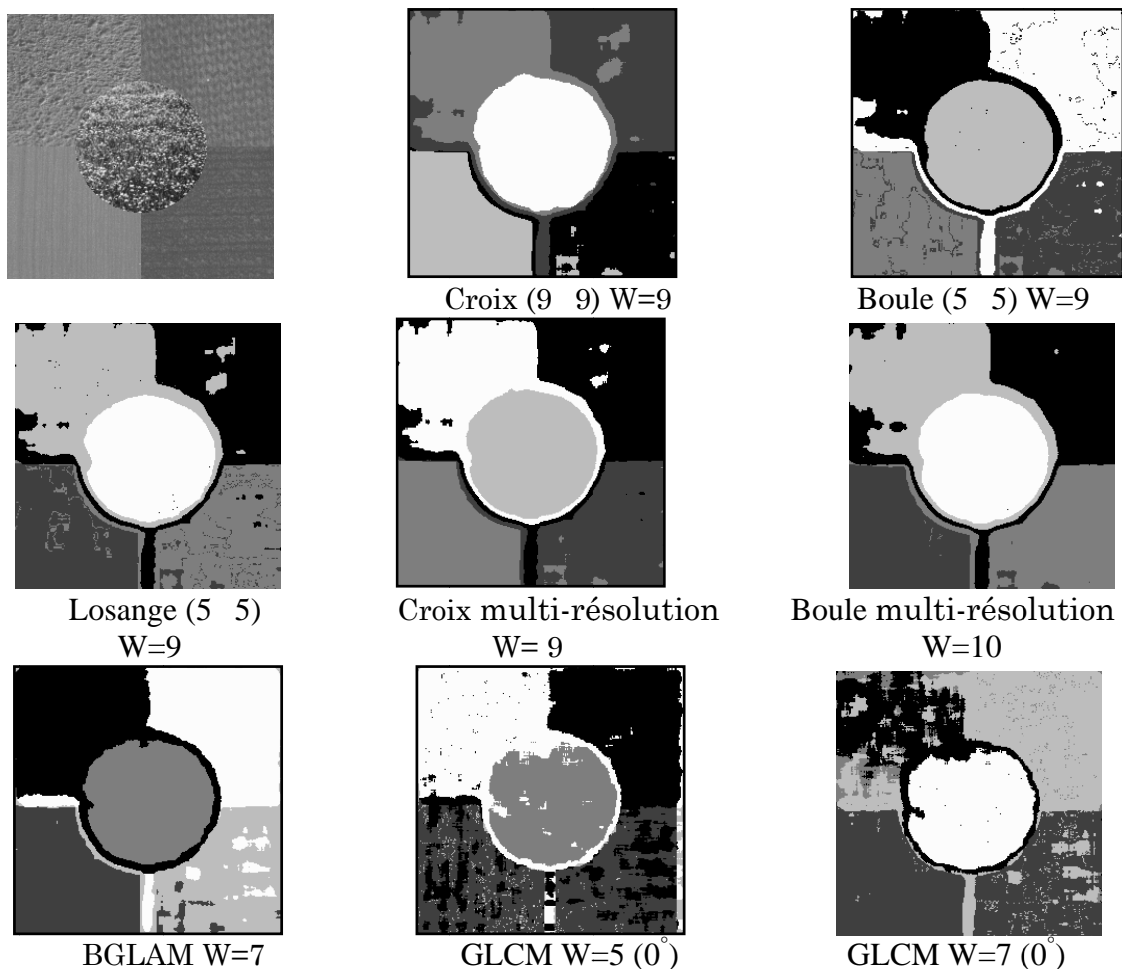


Fig.4.20 : Image synthétique 3 segmentée par les différentes méthodes.

Image synthétique 4

La figure (4.21) montre les différentes images segmentées obtenus sur une autre image synthétique composée de 4 autres textures par la méthode des matrices de cooccurrence et les matrices Aura.

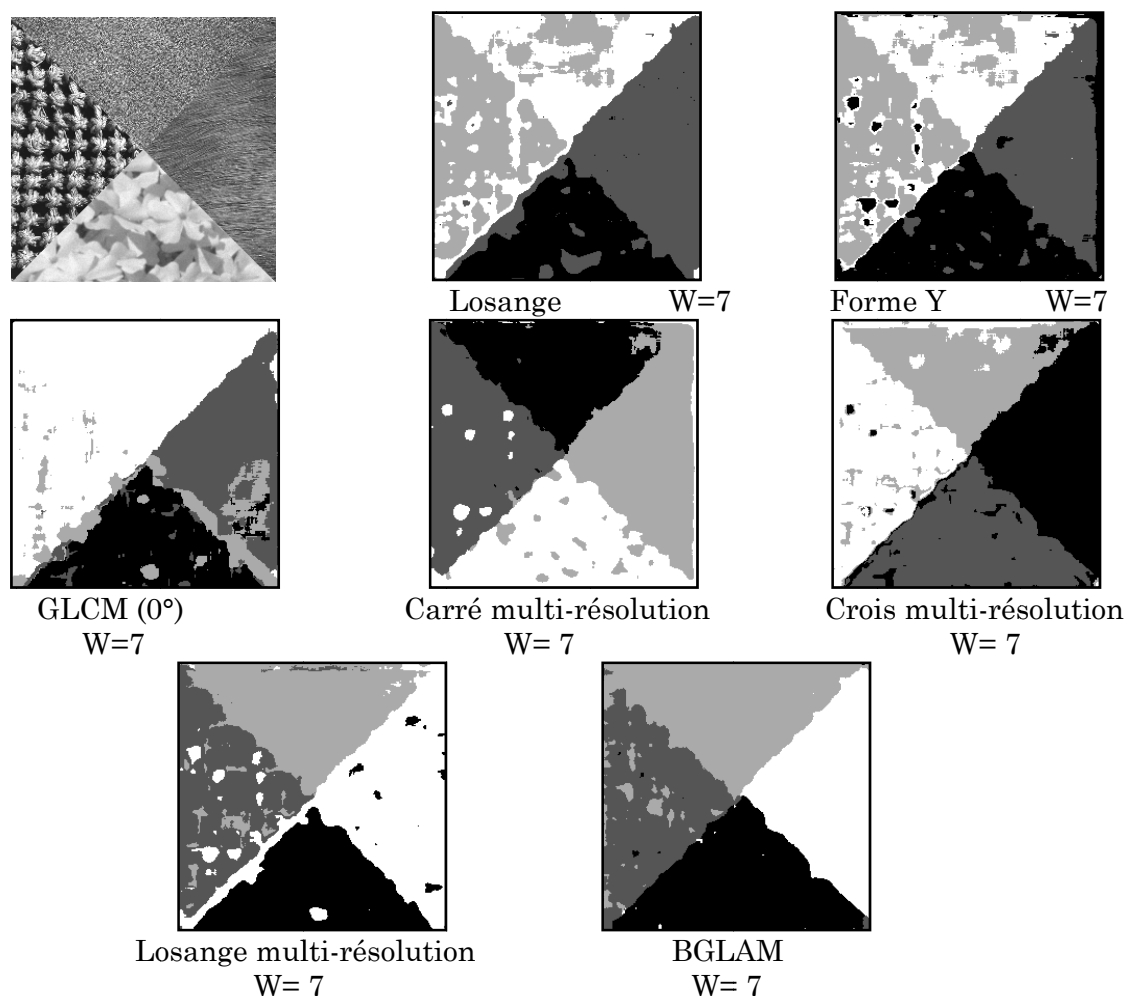


Fig.4.21 : Image synthétique 4 segmentée par les différentes méthodes.

Image synthétique 5

La figure (4.22) montre les différentes images segmentées obtenus sur une autre image synthétique composée de 3 textures différentes par la méthode des matrices de cooccurrence et les matrices Aura.

Les figures 4.19, 4.20, 4.21 et 4.22 montrent que, les résultats de segmentation obtenus par la méthode de la matrice Aura de niveau de gris quelque soit la stratégie utilisée (1 seul élément structurant symétrique ou non symétrique, BGLAMs ou analyse multi-résolution) sont meilleurs par rapport aux résultats obtenus avec la matrice de cooccurrence.

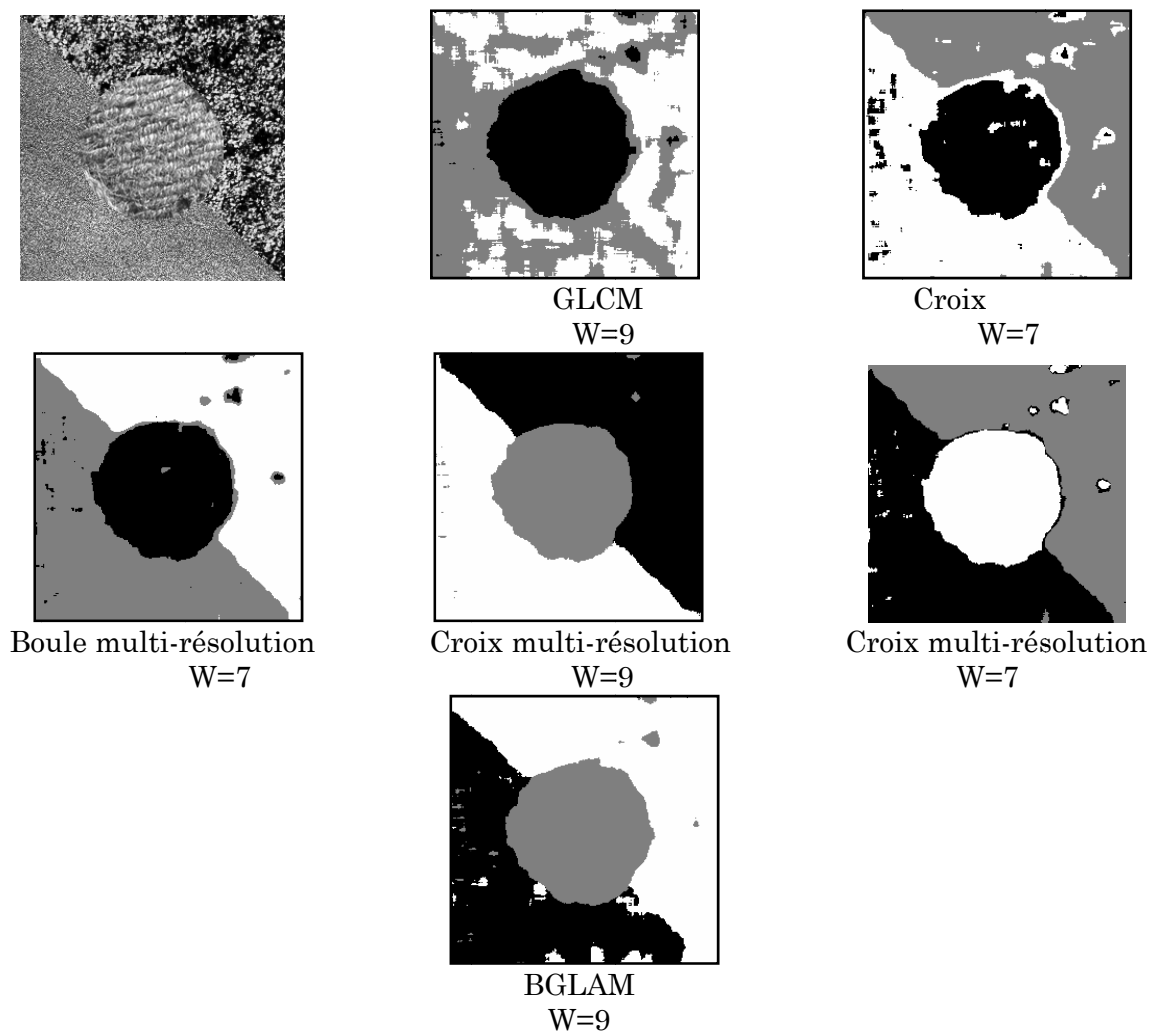


Fig.4.22 : Image synthétique 5 segmentée par les différentes méthodes.

Image réelle 1

La figure (4.23) montre les différentes images segmentées obtenus sur une image réelle décrivant un zèbre par la méthode des matrices de cooccurrence et les matrices Aura. Nous avons considérée que cette image est composée de deux zones texturées. La première correspond au fond de l'image et la deuxième à la robe du zèbre.

Image réelle 2

La figure (4.24) montre les différentes images segmentées obtenus sur une image réelle décrivant un léopard par la méthode des matrices de cooccurrence et les matrices Aura. Nous avons considérée que cette image est composée de trois zones texturées. La première correspond à l'herbe, la deuxième aux arbres et la troisième à la robe du léopard.

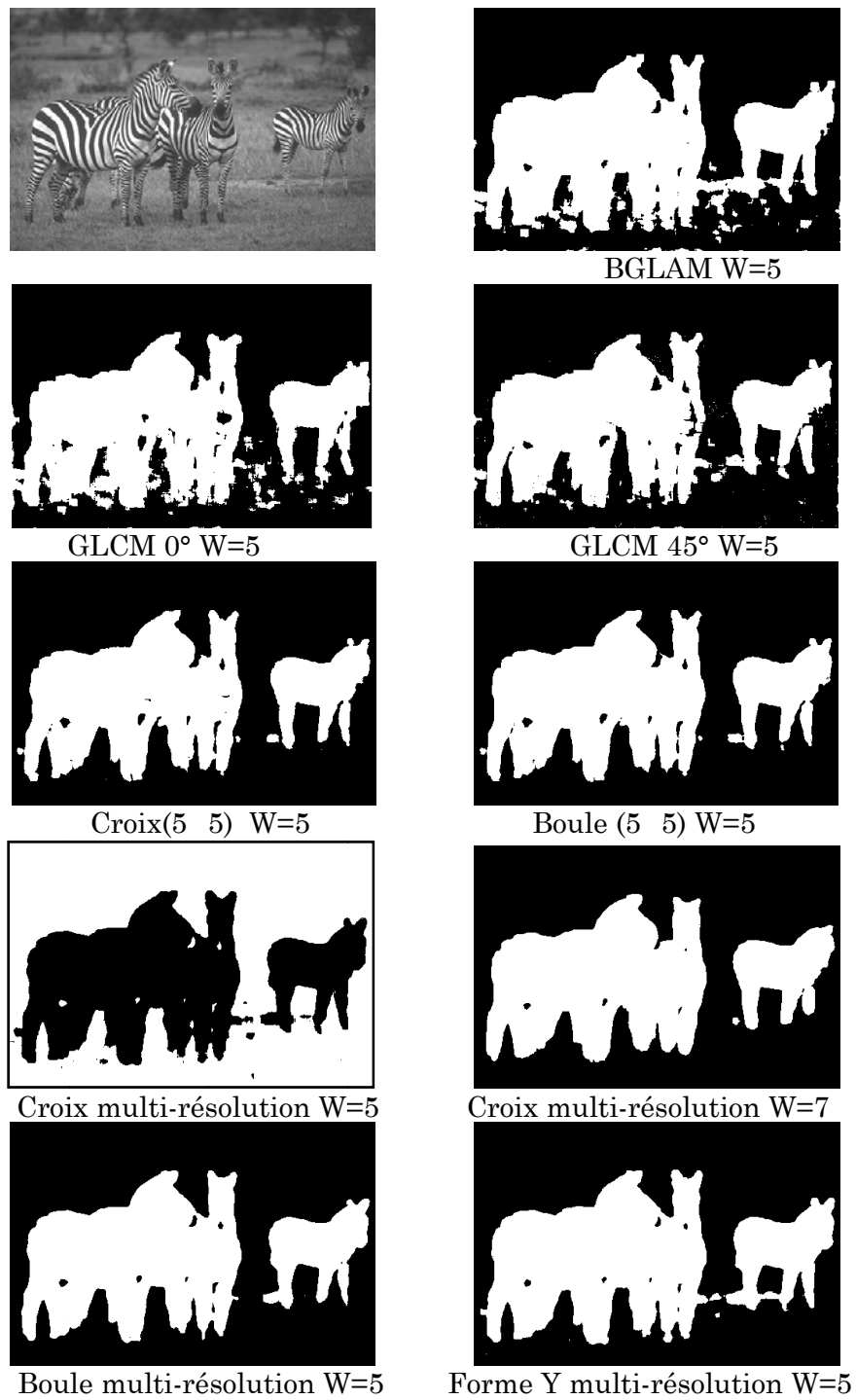


Fig.4.23 : Image réelle 1 segmentée par les différentes méthodes.

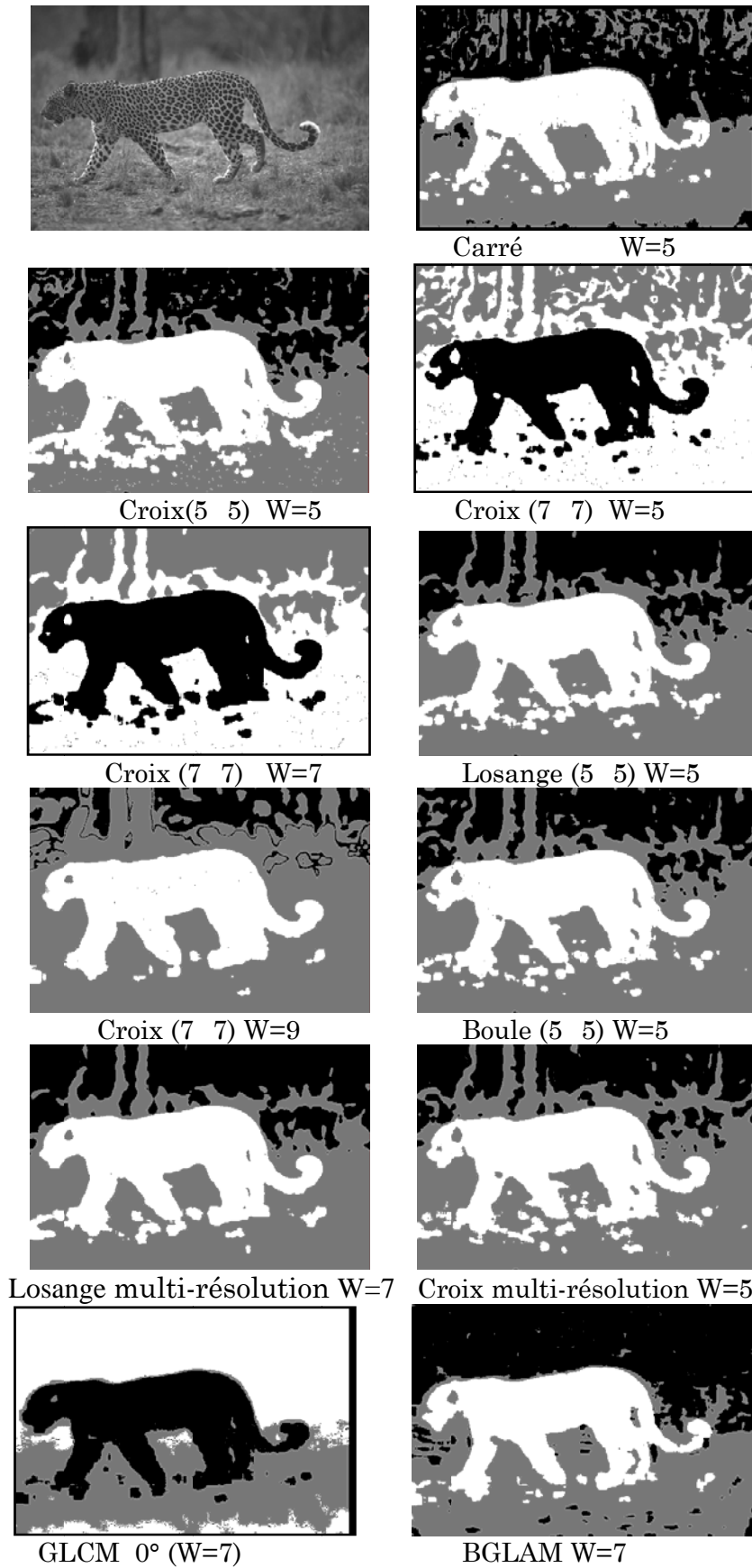


Fig.4.24 : Image réelle 2 segmentée par les différentes méthodes.

Image réelle 3

La figure (4.25) montre les différentes images segmentées obtenus sur une image médicale composée principalement de trois zones texturées.

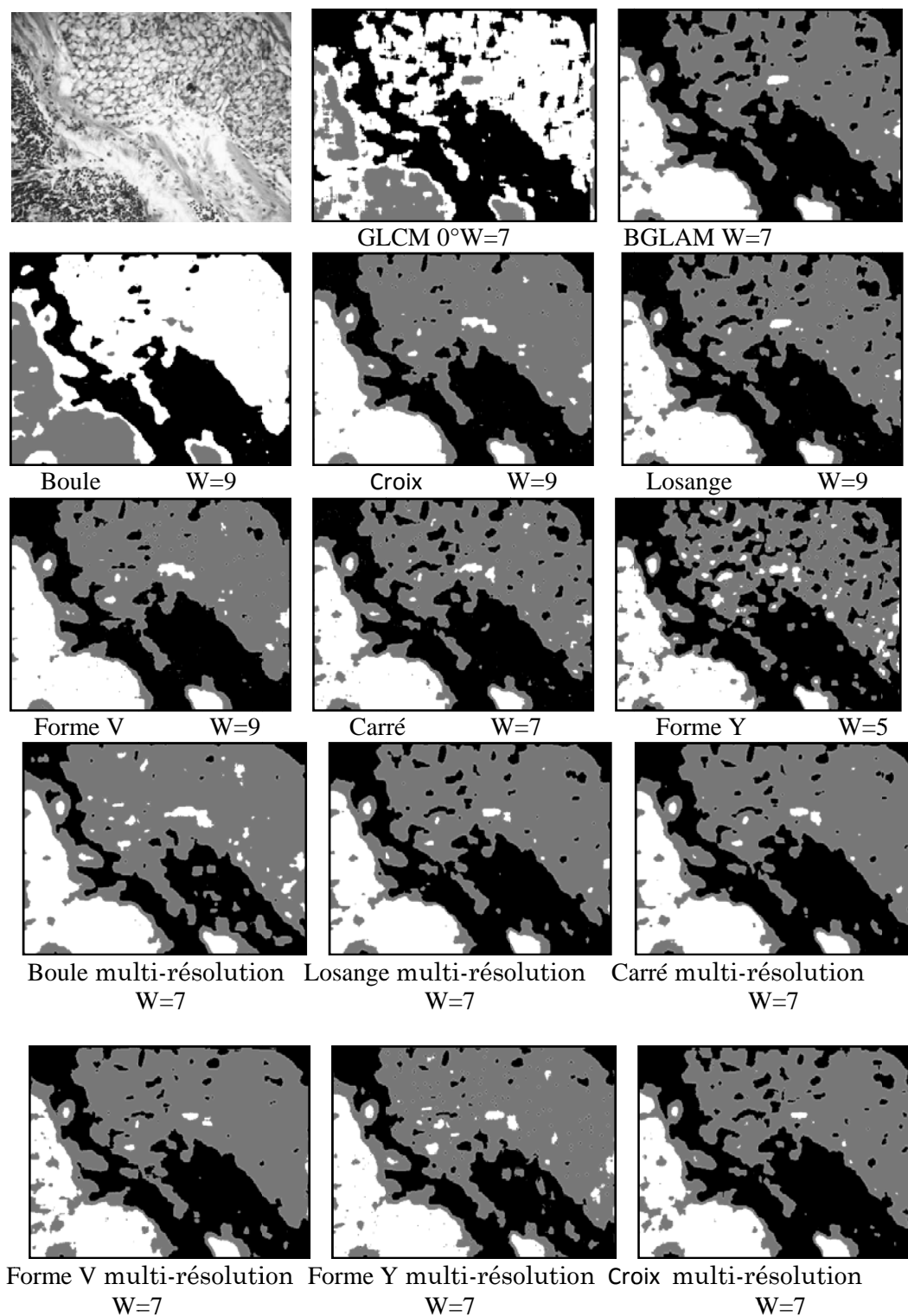


Fig.4.25 : Image réelle 3 segmentée par les différentes méthodes.

Interprétation

Les figures 4.23, 4.24 et 4.25 montrent que, quelque soit l'élément structurant utilisé, les résultats de segmentation obtenus sur des images réelles par la méthode de la matrice Aura de niveau de gris sont très appréciables et meilleures par rapport aux résultats obtenus par la matrice de cooccurrence.

4.7 Conclusion

Nous avons présenté dans ce chapitre quelques résultats de classification des textures à base des matrices Aura. Ces résultats nous a démontré le pouvoir discriminatoire des attributs de texture extraits à partir des matrices Aura. Nous avons également présenté les tests des différentes méthodes implémentées pour le calcul de ces attributs dans le cadre de la segmentation d'images texturées. Une comparaison par rapport au temps d'exécution entre les différentes méthodes de programmation a été menée. Les résultats montrent que la méthode utilisant les structures hybride est plus rapide que la méthode des listes chaînées et beaucoup plus rapide que la méthode traditionnelle.

Quelque soit l'élément structurant utilisé, les résultats de la segmentation à base des matrices Aura en utilisant plusieurs éléments structurants ayant une même forme et des tailles variables (analyse multi résolution) sont beaucoup meilleurs que ceux obtenus avec les matrices des cooccurrences, mais ils sont proches ou légèrement supérieurs à ceux obtenus avec un seul élément structurant ayant une seule forme et une seule taille et avec les BGLAMs.

La taille de la fenêtre de voisinage a une influence directe sur le résultat de la segmentation par les matrices de cooccurrences, les matrices Aura des niveaux de gris et les matrices Aura de base.

Notons également que les résultats de la segmentation peuvent être tributaires des attributs de texture. L'idéal serait d'utiliser une méthode de sélection d'attributs qui permettra une meilleure discrimination entre les différentes textures.

Conclusion générale

Conclusion générale

Dans cette thèse, nous avons proposé une nouvelle méthode de segmentation d'images texturées basée sur les matrices Aura des niveaux de gris.

Nous avons vu en premier lieu que le concept Aura s'inspire de la miscibilité des fluides, il est basée sur la théorie des ensembles et contient des notions telles que l'ensemble Aura qui peut être interprété comme la frontière entre deux ensembles, une mesure Aura qui donne une information sur le caractère miscible de deux sous ensembles, elle peut être assimilée à la longueur de la frontière de deux ensembles et la matrice Aura ou matrice de miscibilité qui englobe les mesures Aura des différents ensembles pris deux à deux. La généralisation de ce concept dans le cadre d'une image a permis de définir les matrices Aura de niveau de gris. Par la suite nous avons montré d'un point de vue théorique le lien entre les matrices Aura avec les matrices de cooccurrence, les champs aléatoires de Markov, les matrices de covariance et la morphologie mathématique. Contrairement aux matrices de cooccurrence et aux champs aléatoires de Gibbs qui prennent en compte seulement l'interaction entre deux pixels, les matrices Aura de niveau de gris ont l'avantage de décrire les relations spatiales des niveaux de gris entre plusieurs pixels en général ou plus exactement entre un pixel et un ensemble de ses pixels voisins définis par un élément structurant.

La méthodologie employée pour segmenter une image à base des matrices Aura des niveaux de gris consiste à caractériser chaque pixel de l'image par un ensemble d'attributs extraits des matrices Aura des niveaux de gris. Ces matrices sont déterminées localement en utilisant les pixels situées dans une fenêtre de voisinage. Sur la base de ces attributs, les pixels sont regroupés en classes de textures.

Les performances de l'approche proposée dépendent de la forme, du type (symétrique ou non), de la taille des éléments structurants et de la taille de la fenêtre de voisinage. Plusieurs stratégies ont été ainsi proposées pour utiliser un ou plusieurs éléments structurants à savoir :

- Un seul élément structurant de forme et de taille fixe à la fois.
- Plusieurs éléments structurants en même temps ayant tous la même forme mais de tailles variables (analyse multi résolution).
- Des matrices Aura de niveaux de gris de base (BGLAMs).

De plus, parallèlement à la techniques classique de programmation du calcul des matrices Aura de niveaux de gris (GLAM), nous avons proposé deux techniques de programmation, l'une est basée sur les listes chaînées (GLALL), l'autre à structure hybride qui combine la notion des listes chaînées et des tables de hachage (GLAHS).

Les performances de ces différentes stratégies et techniques de programmation en fonction de la largeur de la fenêtre de voisinage, du type, de la forme et de la taille l'élément structurant ont été évaluées sur la base du taux de classification et le temps d'exécution pour différentes images synthétiques et réelles. Quelque soit l'élément structurant utilisé et pour toutes les images, les résultats de la segmentation à base des matrices Aura utilisant plusieurs éléments structurants ayant une même forme et des tailles variables (analyse multi résolution) sont meilleurs ou proches que ceux obtenus avec un seul élément structurant ayant une seule forme et une seule taille et avec les BGLAMs. Cependant les résultats obtenus avec les matrices Aura sont beaucoup meilleurs que ceux obtenus avec les matrices de cooccurrence quelque soit la stratégie adoptée. Concernant le temps de calcul, les résultats montrent que dans tous les cas la GLAHS est plus rapide que la GLALL et la GLAM. La GLALL reste plus rapide que la GLAM que lorsque la taille de la fenêtre de voisinage est petite. Le calcul de la GLAHS avec plusieurs éléments structurants reste malheureusement plus lent par rapport au calcul de la GLAHS avec un seul élément structurant.

Parallèlement à la segmentation, nous avons abordé la classification des textures à base des matrices Aura. Le taux de classification obtenue sur une base de test frôle les 100% quelque soit l'élément structurant. Cependant, ces résultats doivent être confirmés sur d'autres bases d'images.

Finalement, les matrices Aura offrent un outil puissant non seulement pour la synthèse et la classification de la texture, mais également pour la segmentation d'images texturées.

Comme perspectives, nous préconisons de déterminer automatiquement le type de l'élément structurant, d'accélérer encore plus les calculs lors d'une analyse multi résolution, d'introduire une phase de sélection des attributs de texture (ACP ou ACI), d'utiliser une méthode de classification non supervisée qui serait insensible à l'initialisation et capable de déterminer automatiquement le nombre de classes.

Références

Références

- [1] T. Zouagui. Approche fonctionnelle générique des méthodes de segmentation d'images. Thèse de Doctorat en Images et systèmes, L'institut National des Sciences Appliquées de Lyon, 2004.
- [2] D. Benboudjema. Champs de Markov Triplets et segmentation Bayesienne non supervisée d'images. Thèse de Doctorat, Université de technologie de Troyes, 2005.
- [3] S. Chabrier. Contribution à l'évaluation de performances en segmentation d'images, Thèse de Doctorat, Université d'Orléans, 2005.
- [4] J. Macqueen. Some methods for classification and analysis of multivariate observations. Proc. 5th Symp. Math. Stat. Prob., pp. 281-297, 1967.
- [5] X. Qin and Y.-H. Yang. Similarity measure and learning with gray level Aura matrices (GLAM) for texture image retrieval. IEEE Computer Vision and Pattern Recognition, vol. 1, pp. 326-333, 2004.
- [6] X. Qin and Y.-H. Yang. Basic gray level Aura matrices: theory and its application to texture synthesis. IEEE International Conference on Computer Vision, pp. 128-135, 2005.
- [7] I.-M. Elfadel and R.-W. Picard. Gibbs random fields, co-occurrences, and texture modeling. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 16, no. 1, 1994.
- [8] B. Naegel. Outils pour le traitement et l'analyse d'images, Introduction à la morphologie mathématique. Ecole des Mines de Nancy, 2008.
- [9] I.-M. Elfadel, R.-W. Picard and A.-P. Pentland. Markov/Gibbs texture modeling: Aura matrices and temperature effects. IEEE Conference on Computer Vision and Pattern Recognition, pp. 371-377, 1991.
- [10] X. Qin and Y.-H Yang. Aura texture.
- [11] X. Qin and Y.-H Yang. Aura 3D textures. IEEE Transaction on Visualization and Computer Graphics, vol. 13, no. 2, 2007.
- [12] I.-M. Elfadel and R.-W. Picard. Miscibility matrices explain the behavior of grayscale fields. Intelligent Robots and Computer Vision, pp. 524-535, 1990.
- [13] I.-M. Elfadel and R.-W. Picard. Gibbs random fields, co-occurrences and texture modeling. M.I.T. Media Lab., Perceptual Computing Group Technical Report, 1993.
- [14] M. Heydari, R. Greiner and A.Murtha. Segmenting brain tumours using texture with BGLAMs. University of Alberta, 2007.

- [15] X. Qin and Y.-H. Yang. Texture image classification using basic gray level Aura matrices.
- [16] R.-W. Picard. Gibbs random fields: temperature and parameter analysis. IEEE, vol.3, no. 177, pp. 45-48, 1992.
- [17] X. Qin and Y.-H. Yang. Representing texture images using asymmetric gray level Aura matrices.
- [18] L. Shu and A.C.S.Chung. Texture classification by using advanced local binary patterns and spatial distribution of dominant patterns. Lo Kwee-Seong Medical Image Analysis Laboratory, 2007.
- [19] M.T. Syrjäsoo, E.F. Donova, X. Qin, and Y.-H. Yang. Automatic classification of auroral images in substorm studies. Int. Conf. Substorms-8, pp.309-313, 2006.
- [20] D.A. Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. Canadian Journal Remote Sensing, vol. 28, no. 1, pp. 45-62, 2002.
- [21] U. Kandaswamy, D.A. Adjeroh and M.C. Lee. Efficient texture analysis of SAR imagery. IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no.9, pp. 2075-2083, 2005.
- [22] A. Baraldi and F. Parmiggiani. An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters. IEEE Transactions on Geoscience and Remote Sensing, vol. 33, no. 2, pp. 293-304, 1995.
- [23] F. Argenti, L. Alparone and G. Benelli. Fast algorithms for texture analysis using cooccurrence matrices. IEE Proc., vol. 137, no. 6, pp.443-448, 1990.
- [24] D.A. Clausi and M.E. Jernigan. A fast method to determine co-occurrence texture features. IEEE International on Geoscience and Remote Sensing, vol.36, no 1, pp. 298-300, 1998.
- [25] D.A. Clausi and Y. Zhao. Rapid co-occurrence texture features extraction using a hybrid data structure. Computers and Geosciences, vol. 28, no. 6, pp. 763-774, 2002.
- [26] M. Divay, Algorithmes et structures de données génériques, Dunod, 1996.
- [27] F. Angot. Segmentation d'images 2D et 3D ; application à la quantification d'images histologiques et cytologiques obtenues par microscopie. Thèse de Doctorat, Université de Caen, 1999.
- [28] K. I. Laws. Textured image segmentation. Tech. Rep. 940, USC, P, Technical Report, 1980.
- [29] J. Shlens. A tutorial on principal component analysis. 2002.

- [30] E. R. Dougherty, Euclidean grey-scale granulometries : representation and umbra inducement, *J. Math. Imaging and Vision*, vol. 1, pp. 7-21, 1992.
- [31] Y. Chen and E. R. Dougherty, Grey-scale morphological granulometric texture classification, *Opt. Eng.*, vol. 33, no. 8, pp. 2713-2722, 1994.
- [32] X.Q. Zhou and B.Z. Yuan, Shape description and recognition using the high order morphological pattern spectrum, *Patt. Recog.*, vol. 28, no. 9, pp. 1333-1340, 1995.
- [33] M.N. Shirazi, H. Noda et N. Takao, Texture classification based on Markov modeling in wavelet feature space, *Image and Vision Computing*, vol. 18, pp. 967-973, 2.
- [34] A.Hafiane. Caractérisation de texture et segmentation pour la recherche d'images par le contenu. Thèse de Doctorat, Université de Paris-Sud XI, Faculté des Science d'Orsay, 2005.
- [35] L.-S.Davis, S.-A.Johns and J.K.Aggarwal. Texture analysis using generalized co-occurrence matrices. *IEEE*, vol.PAMI-1, no.2, pp.251-259, 1979.
- [36] W.Pianpian, W.Beizhan, Q.Cheng and C.Layan. Improved generalized co-occurrence matrix for image retrieval using MPEG-7 texture edge. *IEEE, International Symposium on Information Science and Engineering*, 2008.
- [37] S. Mavromatis. Analyse de texture et visualisation scientifique. Thèse de Doctorat en Informatique, Université de la Méditerranée, 2001.
- [38] A. Materka and M. Strzelecki. Texture analysis methods- A review. COST B11 report, Brussels, 1998.
- [39] M. Tuceryan and A.K. Jain. Texture analysis. Chapter 2.1, *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), pp. 207-248, Word Scientific Publishing Co., 1998.
- [40] T. Randen, J.H. Husoy. Filtering for texture classification: a comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 291-310, 1999.
- [41] M. Nixon and A. Agnado. Feature extraction and image processing. Newnes Edition, 2002.
- [42] I. Christodoulou, C.S. Pattichis, M. Pantziaris and A. Nicolaidis. Texture-based classification of atherosclerotic Carotid Plaques. *IEEE Transactions on Medical Imaging*, vol. 22, no. 7, pp. 902-912, 2003.
- [43] M.R. Turner. Texture discrimination by Gabor function. *Biological Cybernetics*, vol. 55, pp. 71-82, 1986.
- [44] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern. Recognition*, vol. 24, n° 12, pp. 1167-1186, 1991.

- [45] M. Idrissa, M. Acheroy. Texture classification using Gabor filters. *Pattern Recognition*, vol. 23, pp. 1095-1102, 2002.
- [46] M. Acharyya and M.K. Kundu. An adaptative approach to unsupervised texture segmentation using M-band wavelet transform. *Signal Processing*, vol. 81, pp. 1337, 1356, 2001.
- [47] K. Pratt. *Digital image processing*. Willey Interscience, New York, 1991.
- [48] S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Elsevier Academic Press, 2003.
- [49] R.M. Haralick, K. Shanmugam and I.H. Dinstein. Textural features for image classification. *IEEE Transactions on Systems Man and cybernetics*, Vol. SMC-3, n°6, 1973.
- [50] Desachy. *Contribution à l'étude de la texture en télédétection*. Thèse de Doctorat, Université Paul-Sabatier, Toulouse, France, 1980.
- [51] M. Unser. Sum and difference histograms for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 118-125, 1986.
- [52] P. Garcia-Sevilla and M. Petrou, Analysis of irregularly shaped texture regions, *Computer Vision and Image Understanding*, vol. 84, pp. 62-76, 2001.
- [53] D.A. Clausi, Y. Zhao. Grey level co-occurrence integrated algorithm (GLCIA): A superior computation method to rapidly determine co-occurrence probability texture features. *Computer and Geosciences*, vol. 29, pp. 837-850, 2003.
- [54] T. Ojala and M. Pietikäinen. Unsupervised texture segmentation using features distributions. *Pattern Recognition*, vol. 32, pp. 477-786, 1998.
- [55] O. D. Faugeras and W. K. Pratt. Decorrelation methods of texture feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 4, 1980.
- [56] R. Chellappa. Two-dimensional discrete gaussian Markov random field models for image processing. In Kanal, L. N. and Rosenfeld, A., editors, *Progress in Pattern Recognition* vol. 2, pp. 79-112, 1985.
- [57] G.C. Cross and A.K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 25-39, 1983.
- [58] J.Serra. *Image analysis and mathematical morphology*. Academic Press, London, 1982.
- [59] D.Wang. *Décomposition morphologique multi-tailles et ségmentation adaptative de textures*. PhD thesis, INSA de Rennes, decembre 1991.

- [60] T. R. Reed and H. Wechsler, Texture segmentation using diffusion region growing technique, *Pattern Recognition*, vol. 23, n° 9, pp. 953-960, 1990.
- [61] P. C. Chen and T. Pavlidis, Segmentation by texture using a cocurrence matrix and split and merge, *Comput. Graphics Image Process.* 10, pp. 172-182, 1979.
- [62] M. F. Doherty, C.M. Bjorklund and M. T. Noga, Split-merge-merge. An enhanced segmentation compatibility, in proceedings, *IEEE Computer Society Conference on computer Vision and Pattern Recognition*, Miami Beach, Florida, pp. 325-330, 1986.
- [63] Z. Wang and al, Comparison of several approaches for the segmentation of texture images, *pattern Recognition*, No 17, pp. 509-521, 1996.
- [64] M.S. Bouguelid. Contribution à l'application de la reconnaissance des formes et la théorie des probabilités au diagnostic adaptatif et prédictif des systèmes dynamiques. Thèse de Doctorat, Université de Reims Champagne-Ardenne, 2007.
- [65] C. Bouveyron. Modélisation et classification des données de grande dimension, Application à l'analyse d'images. Thèse de doctorat, Université Joseph Fourier-Grenoble 1, 2006.

Annexes

Annexe A : Introduction à la théorie des ensembles.

A.1 Notations ensemblistes :

Soient V et V' deux ensembles finis.

- Le cardinal de V , noté $|V|$, est le nombre d'éléments de V .
- V' est inclus dans V , noté $V' \subseteq V$, si tout élément de V' est aussi un élément de V .
Dans ce cas, on dit que V' est une partie ou un sous-ensemble de V .
- Si $V' \subseteq V$ et $V' \neq V$, on dit que V' est un sous-ensemble strict de V et on écrit $V' \subset V$.
- On désigne par 2^V , l'ensemble de toutes les parties de V .
- L'union de V et V' , notée $V \cup V'$, est l'ensemble formé des éléments de V et de V' .
- L'intersection de V et V' , notée $V \cap V'$, est l'ensemble des éléments qui sont à la fois dans V et dans V' .
- La différence (ensembliste) de V et V' , notée $V \setminus V'$, est l'ensemble des éléments de V qui n'appartiennent pas à V' .
- Si $V' \subseteq V$, le complémentaire de V' dans V , noté \bar{V}'^V est l'ensemble $V \setminus V'$.

A.2 Propriétés des opérations sur les ensembles

Quelques propriétés sur les ensembles sont données ci-dessous :

1) Idempotence

$$V \cap V = V \tag{A.1}$$

$$V \cup V = V \tag{A.2}$$

2) Commutativité

L'intersection et la réunion sont des opérations commutatives. Quels que soient les ensembles V et V' :

$$V \cap V' = V' \cap V \tag{A.3}$$

$$V \cup V' = V' \cup V \tag{A.4}$$

3) Associativité

L'intersection et la réunion sont associatives. Quels que soient les ensembles V , V' et V'' :

$$V \cap (V' \cap V'') = (V \cap V') \cap V'' \tag{A.5}$$

$$V \cup (V' \cup V'') = (V \cup V') \cup V'' \quad (\text{A.6})$$

4) Complémentation

Deux sous-ensembles V et V' de l'ensemble E sont dits complémentaires si leur réunion est l'ensemble E et leur intersection est l'ensemble vide :

$$V \cap V' = \emptyset \quad (\text{A.7})$$

$$V \cup V' = E \quad (\text{A.8})$$

On note alors :

$$V' = C_E^V \quad (\text{A.9})$$

Ou encore :

$$V' = V^c \quad (\text{A.10})$$

5) Elément neutre

$$V \cap E = V \quad (\text{A.11})$$

$$V \cup \emptyset = V \quad (\text{A.12})$$

6) Elément absorbants

$$V \cap \emptyset = \emptyset \quad (\text{A.13})$$

$$V \cup E = V \quad (\text{A.14})$$

7) Distributivité :

L'intersection et la réunion sont distributives l'une par rapport à l'autre :

$$V \cap (V' \cup V'') = (V \cap V') \cup (V \cap V'') \quad (\text{A.15})$$

$$V \cup (V' \cap V'') = (V \cup V') \cap (V \cup V'') \quad (\text{A.16})$$

8) Lois de De Morgan

Quels que soient les sous-ensembles V et V' de E :

$$(V \cup V')^c = V^c \cap V'^c \quad (\text{A.17})$$

$$(V \cap V')^c = V^c \cup V'^c \quad (\text{A.18})$$

Annexe B : Preuves de quelques propriétés et théorèmes sur les mesures Aura et les matrices Aura.

B.1 Preuves des propriétés des mesures Aura

Propriété 4

En générale, $m(A, B) \neq m(B, A)$. Mais, si N est un système de voisinage symétrique, alors : $m(A, B) = m(B, A)$ (B.1)

Preuve de la propriété 4

Soit N est un système de voisinage symétrique. Par définition on a :

$$m(A, B) = \sum_{s \in A} |N_s \cap B| \quad (B.2)$$

$$m(B, A) = \sum_{s \in B} |N_s \cap A| \quad (B.3)$$

Considérons b l'élément unique de l'ensemble B . Montrons que, $m(A, B) = m(B, A)$ quand $|B| = 1$.

On a:

$$m(B, A) = |N_b \cap A| \quad \forall |B| = 1 \quad (B.4)$$

D'autre part, on a :

$$m(A, B) = \sum_{s \in A} |N_s \cap \{b\}| \quad (B.5)$$

Quand $b \in N_s$ ou $s \in N_b$ (système de voisinage symétrique) on a alors:

$$m(A, B) = \sum_{s \in N_b \cap A} 1 = |N_b \cap A| = m(B, A) \quad (B.6)$$

Supposant maintenant, que $m(A, B) = m(B, A)$ est satisfaite quelque soit le cardinal de l'ensemble B tel que $|B| = k \in \{1, \dots, n-1\}$ et considérons un ensemble B tels que $|B| = n$. En considérant $b \in B$, nous pouvons écrire $B = C \cup \{b\}$ avec $C = B - \{b\}$ et $|C| = n-1$.

On a, alors:

$$m(A, B) = m(A, C \cup \{b\})$$

$$\begin{aligned}
&= \sum_{s \in A} |N_s \cap (C \cup \{b\})| \\
&= \sum_{s \in A} |(N_s \cap C) \cup (N_s \cap \{b\})| \\
&= \sum_{s \in A} (|N_s \cap C| + |N_s \cap \{b\}|) \\
&= \sum_{s \in C} (|N_s \cap A| + |N_b \cap A|) \quad \text{car } C \cap \{b\} = \emptyset \\
&= \sum_{s \in C \cup \{b\}} |N_s \cap A| \\
&= \sum_{s \in B} |N_s \cap A| \\
&= m(B, A) \tag{B.7}
\end{aligned}$$

Maintenant, on considère $m(A, B) = m(B, A)$ est vraie, Montrons que, N est un système de voisinage symétrique.

Soient s, r deux pixels tel que $r \in N_s$. Nous allons prouver que $s \in N_r$. Soient $A = \{s\}$ et $B = \{r\}$. On a:

$$m(A, B) = |N_s \cap \{r\}| = |N_r \cap \{s\}| = m(B, A) \tag{B.8}$$

$$|N_r \cap \{s\}| = 1 \Rightarrow s \in N_r \tag{B.9}$$

Propriété 7

Si $A \cup B = S$ et $A \cap B = \emptyset$ on a:

$$m(A, A) + m(A, B) = |A||N| \tag{B.10}$$

$$m(B, A) + m(B, B) = |B||N| \tag{B.11}$$

Preuve de la propriété 7

Par définition on a :

$$\begin{aligned}
m(A, A) + m(A, B) &= \sum_{s \in A} |N_s \cap A| + \sum_{s \in A} |N_s \cap B| \\
&= \sum_{s \in A} |N_s \cap (A \cup B)| \quad (\text{car } A, B \text{ sont disjoints}) \\
&= \sum_{s \in A} |N_s \cap S| \\
&= \sum_{s \in A} |N_s|
\end{aligned}$$

$$= |A||N| \quad (B.12)$$

La deuxième égalité de la propriété 7 peut être démontrée de la même façon que la première.

$$\begin{aligned} m(B, A) + m(B, B) &= \sum_{s \in B} |N_s \cap A| + \sum_{s \in B} |N_s \cap B| \\ &= \sum_{s \in B} |N_s \cap (A \cup B)| \quad (\text{car } A, B \text{ sont disjoints}) \\ &= \sum_{s \in B} |N_s \cap S| \\ &= \sum_{s \in B} |N_s| \\ &= |B||N| \end{aligned} \quad (B.13)$$

Propriété 8

Soient A, B, C trois sous-ensembles disjoints dans S . Soient $A', B', A_s \subseteq A$, et $B_s \subseteq B$, tel que $|A_s| = |B_s|$, $A' = (A - A_s) \cup B_s$ et $B' = (B - B_s) \cup A_s$, on a [13]:

$$\begin{aligned} m(A', A') + m(B', B') + m(A', B') + m(B', A') \\ = m(A, A) + m(B, B) + m(A, B) + m(B, A) \end{aligned} \quad (B.14)$$

et

$$m(A', C) + m(B', C) = m(A, C) + m(B, C) \quad (B.15)$$

Preuve de la propriété 8

Cette propriété est prouvée quand les deux sous-ensembles A et B échangés par une paire de sous-ensembles de taille égale.

Nous avons $A \cup B = A' \cup B'$.

L'égalité (B.14) résulte du fait que: $m(A \cup B, A \cup B) = m(A' \cup B', A' \cup B')$

En effet on a :

$$\begin{aligned} m(A \cup B, A \cup B) &= \sum_{s \in A \cup B} |N_s \cap (A \cup B)| \\ &= \sum_{s \in A} |N_s \cap (A \cup B)| + \sum_{s \in B} |N_s \cap (A \cup B)| \end{aligned}$$

$$\begin{aligned}
&= \sum_{s \in A} |N_s \cap A| + \sum_{s \in A} |N_s \cap B| + \sum_{s \in B} |N_s \cap A| + \sum_{s \in B} |N_s \cap B| \\
&= m(A, A) + m(A, B) + m(B, A) + m(B, B)
\end{aligned} \tag{B.16}$$

Nous avons une égalité semblable pour $m(A' \cup B', A' \cup B')$:

$$\begin{aligned}
m(A' \cup B', A' \cup B') &= \sum_{s \in A' \cup B'} |N_s \cap (A' \cup B')| \\
&= \sum_{s \in A'} |N_s \cap (A' \cup B')| + \sum_{s \in B'} |N_s \cap (A' \cup B')| \\
&= \sum_{s \in A'} |N_s \cap A'| + \sum_{s \in A'} |N_s \cap B'| + \sum_{s \in B'} |N_s \cap A'| + \sum_{s \in B'} |N_s \cap B'| \\
&= m(A', A') + m(A', B') + m(B', A') + m(B', B')
\end{aligned} \tag{B.17}$$

La deuxième égalité (B.15) de la propriété 8 résulte de fait que $m(A \cup B, C) = m(A' \cup B', C)$, spécifiquement :

$$\begin{aligned}
m(A \cup B, C) &= \sum_{s \in A \cup B} |N_s \cap C| \\
&= \sum_{s \in A} |N_s \cap C| + \sum_{s \in B} |N_s \cap C| \\
&= m(A, C) + m(B, C)
\end{aligned} \tag{B.18}$$

$$\begin{aligned}
m(A' \cup B', C) &= \sum_{s \in A' \cup B'} |N_s \cap C| \\
&= \sum_{s \in A'} |N_s \cap C| + \sum_{s \in B'} |N_s \cap C| \\
&= m(A', C) + m(B', C)
\end{aligned} \tag{B.19}$$

La première égalité dans la propriété peut être interprétée comme conservation principale d'Aura: quand deux ensembles disjoints échangent des éléments, la somme de leurs mesures Aura est conservée. En outre, considérant le cas spécial où $A_s = A, B_s = B$, et $|A|=|B|$. Dans ce cas-ci, les deux ensembles ont complètement échangé des endroits, et la conservation d'Aura se tient toujours.

B.2 Preuve du lemme 1

Lemme 1 : Soit I_1 une image définie sur S , s un site donné sur S , et N un système de voisinage. Supposant que l'image I_2 est obtenue à partir de l'image I_1 , en changeant la valeur de niveau de gris du pixel s de g_1 en g_2 tel que $g_2 \neq g_1$ (les valeurs des niveaux de gris des autres sites demeurent inchangées). Soient A_1 et A_2 les

matrices Aura de niveau de gris des images I_1 et I_2 , respectivement sur N , alors $A_1 \neq A_2$.

Preuve :

Soient $A_1 = [a_1(i, j)]$ et $A_2 = [a_2(i, j)]$. Soit r le pixel voisin unique de s (N est supposé être un système de voisinage contenant un seul site), et soit r' le site symétrique de r par rapport à s comme montré par la figure suivante :

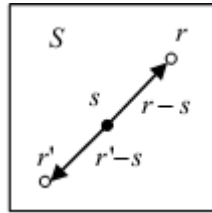


Fig. B.1 : Système de voisinage contenant un seul site.

Soient $g = I_1(r)$ et $g' = I_1(r')$ les niveaux de gris des sites r et r' , respectivement, dans l'image I_1 . On peut montrer que A_2 peut être obtenu à partir de A_1 dans les deux étapes suivantes:

- 1) Initialisé A_2 comme A_1 , c.-à-d. $A_2 \leftarrow A_1$.
- 2) Mis à jour de A_2 :

$$a_2(g_1, g) \leftarrow a_2(g_1, g) - 1$$

$$a_2(g_2, g) \leftarrow a_2(g_2, g) + 1 \tag{B.20}$$

$$a_2(g', g_1) \leftarrow a_2(g', g_1) - 1$$

$$a_2(g', g_2) \leftarrow a_2(g', g_2) + 1$$

Pour toutes les valeurs possibles de g et de g' après vérification de chaque étape dans l'équation (B.18) et en notant $g_1 \neq g_2$, on conclut que $A_1 \neq A_2$.

B.3 Preuve du théorème 1

Théorème 1: Deux images de même taille sont identiques si et seulement si leurs GLAMs correspondantes sur tous les systèmes de voisinage possibles sont identiques.

Preuve

Supposant que deux images identiques I_1 et I_2 sont définies sur une grille S . Il est évident que leurs GLAMs correspondantes sur tous les systèmes de voisinage possibles soient identiques.

On suppose maintenant, que les matrices Aura de niveau de gris correspondantes à deux images I_1 et I_2 sur tous les systèmes de voisinage possibles sont identiques, et nous allons montrer que $I_1 = I_2$. Ceci est équivalent à prouver que si $I_1 \neq I_2$, alors il existe un système de voisinage N tel que les matrices Aura de niveau de gris A_1 et A_2 ne sont pas égales ($A_1 \neq A_2$).

Supposons que $I_1 \neq I_2$. Soit $S = C \cup D$, C et D sont des partitions de S , où $C = \{s \in S \mid I_1(s) = I_2(s)\}$ et $D = \{s \in S \mid I_1(s) \neq I_2(s)\}$. Soit $|D| = n$ et $D = \{r_i \in S \mid 1 \leq i \leq n\}$. Choisissons un système de voisinage N tel que son élément structurant E est assez grand pour contenir D , c.-à-d. $E \supseteq D$.

Soient A_1 et A_2 les matrices Aura de niveau de gris de deux images I_1 et I_2 , respectivement, sur N . En utilisant le lemme 1, on peut montrer que $A_1 \neq A_2$.

B.4 Preuve du théorème 2

Théorème 2: Une GLAM peut être représentée comme la somme de BGLAMs.

Preuve

Soit S une grille, A une GLAM définie sur un système de voisinage arbitraire N , $\{N_r, r \in E\}$ est la décomposition de système de voisinage N et A_r la matrice Aura de niveau de gris calculée à partir d'un système de voisinage N_r . On a par définition de la matrice Aura et la mesure Aura :

$$A = [m(S_i, S_j, N)]_{0 \leq i, j \leq G-1} \quad (B.21)$$

où

$$\begin{aligned} m(S_i, S_j, N) &= \sum_{s \in S_j} |N_s \cap S_i| \\ &= \sum_{s \in S_j} |(\bigcup_{r \in E} N_s^r) \cap S_i| \end{aligned}$$

$$\begin{aligned}
&= \sum_{s \in S_j} |\cup_{r \in E} (N_s^r \cap S_j)| \\
&= \sum_{s \in S_j} \sum_{r \in E} |N_s^r \cap S_j| \\
&= \sum_{r \in E} \sum_{s \in S_j} |N_s^r \cap S_j| \tag{B.22}
\end{aligned}$$

Pour chaque $r \in E$, on a:

$$A_r = [m(S_i, S_j, N_r)]_{0 \leq i, j \leq G-1} \tag{B.23}$$

$$\text{Avec : } m(S_i, S_j, N_r) = \sum_{s \in S_j} |N_s^r \cap S_j| \tag{B.24}$$

A partir des deux équations (B.20) et (B.22), nous aurons $A = \sum_{r \in E} A_r$. Puisque N_r est un système de voisinage contenant un seul site, alors chaque A_r est une BGLAM.

B.5 Preuve de théorème 3

Théorème 3: Deux images de même taille sont identiques si et seulement si leurs BGLAMs correspondantes sont identiques [15].

Preuve

Soient I_1 et I_2 sont deux images définies sur S , nous devons montrer que si $I_1 \neq I_2$, alors il existe un système de voisinage simple N_r tel que $A_1 \neq A_2$, où A_1 et A_2 sont des BGLAMs.

Supposons que $I_1 \neq I_2$, alors par le théorème 1, il existe un système de voisinage N tel que $A_1(N) \neq A_2(N)$. Soit $\{N_r\}$ la décomposition de système de voisinage N , par le théorème 1 on a :

$$A_1(N) = \sum_{r \in E} A_1(N_r) \text{ et } A_2(N) = \sum_{r \in E} A_2(N_r) \tag{B.25}$$

Comme $A_1(N) \neq A_2(N)$, il existe $r \in E$ tel que $A_1(N_r) \neq A_2(N_r)$, autrement nous avons $A_1(N) = A_2(N)$ compte tenu de l'équation (B.18).