

17/004.482

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

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

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Informatique Académique

14/870

Thème :

Classification floue des images texturées

Encadré Par :

Mr. KOUAHLA Mohamed
Nadjib

Présenté par :

DEBBABI Nourddine
GUERNINE Akram

Juin 2014



Remerciements

Nous remercions Dieu de nous avoir accordé des connaissances de la science et de nous avoir aidés à réaliser ce travail.

Au terme de ce modeste travail nous tenons à remercier chaleureusement et respectivement tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste projet de fin d'étude.

*Nos vifs remerciements vont tous d'abord à notre encadreur **Mr. Kouahla Med Nadjibe.***

Tout notre respect et nos remerciements vont vers les membres du jury qui vont pleinement consacrer leur temps et leur attention afin d'évaluer notre travail, qui espérons le sera à la hauteur de leur attente.

Enfin, nos remerciements les plus sincères sont adressés à tous les professeurs, l'administration et le personnel de l'informatique qui ont contribué à forger nos connaissances et à assister notre formation.



DEDICACE

A ma famille

Mes amis

A mon oncle

akram

DEDICACE

*A mes parents, aux être qui sont les plus chères au monde et auxquels je ne
Saurais jamais exprimer ma gratitude et ma reconnaissance en quelques lignes, Je*

Les dédié ce modeste travail, que dieu le tout puissant les protège.

Pour ton amour, ton affection et ton soutien, pour ton courage et ton sacrifice, je

Te dédié, pour la deuxième et mille fois, ma très chère mère, un résultat modeste

De la bienveillance et tes longues années de patience.

A ma petite famille, mes frères et sœurs.

A tous mes amis sans exception.

Et à tous ceux qui j'aime et qu'ils m'aiment... où qu'ils soient.

Nourddine

Liste des figures

Figure 1.1 : schéma représentant les techniques de segmentation.....	13
Figure 1.2 : illustration de détection de contour (a) image lenna original (b) détection de contour par filtre de Sobel. (c) détection de contour par filtre de Prewitt. (d) détection de contour par filtre de Roberts. (e) détection de contour par filtre de Canny.....	14
Figure 1.3 : Principe de la segmentation par division des régions (Structure quaternaire).....	15
Figure 1.4 : Principe de la segmentation par division-fusion.....	16
Figure 2.1 : Texture structurelles.....	24
Figure 2.2: Texture aléatoire.....	25
Figure 2.3: Texture directionnelles.....	25
Figure 2.4: calcul la matrice de cooccurrence d'une image.....	27
Figure 2.5: Exemple de support.....	34
Figure 3.3 : schéma bloc de la méthode.....	37
Figure 3.4 : image couleur RVB de taille 4*4.....	39
Figure 3.5 : Image couleur $R'V'B'$ de taille 4×4 décorrélée.....	41
Figure 3.6: Image résultante L'ACP avec $q = 1$	42
Figure 3.6 : Étiquetage en régions connexes.....	46
Figure 4.1 : test 1.....	49
Figure 4.2 : test 2.....	50
Figure 4.3 : test 3.....	50
Figure 4.4: test 4.....	50
Figure 4.5 : Fenêtre principale de l'application.....	51
Figure 4.6 : La barre de menu images.....	52
Figure 4.7 : La barre de menu traitement.....	52
Figure 4.8: Application de l'ACP.....	53
Figure 4.9 : Paramètres de filtre de Gabor.....	53
Figure 4.10 : application de Gabor.....	54
Figure 4.12 : Extraction des régions texturées.....	54

Figure 4.12 : Extraction des régions texturées.....	55
Figure 4.13 : Paramètres de FCM.....	55
Figure 4.14 : Résultat de FCM.....	56
Figure 4.15 : Appartenance de chaque pixel.....	56
Figure 4.16 : Étiquetage en régions connexes.....	57
Figure 4.17 : Barre de menu Aide (?).....	57
Figure 4.18 : Information sur le logiciel.....	58
Figure 4.19 : image original(1), résultat de ACP(2), résultat de Gabor (3), résultat de l'extraction(4), résultat de FCM(5), résultat finale(6).....	59
Figure 4.20 : résultat de test 1.....	59
Figure 4.21 : résultat de test 2.....	59
Figure 4.22 : résultat de test 3.....	60
Figure 4.23 : résultat de test.....	61
Figure 4.24 : comparaison entre notre approche et FCM.....	61
Figure 4.25 : l'influence des paramètres de filtre de Gabor.....	62

Résumé

Les concepts d'analyse et de segmentation d'images peuvent se dédier à plusieurs thématiques, allant de la photogrammétrie à la segmentation d'objets, Nous nous sommes intéressés à la caractérisation de points et à la segmentation d'images sur un fond de complexité.

Nous avons réalisé un logiciel en java, pouvant traiter toute image en couleur, permettant de classifier les pixels en utilisant une technique floue, de les diviser en régions et de leur affecter une étiquette chacune. La méthode de segmentation d'image texturées couleur par banque de filtres de Gabor qu'on a adapté aux images de texture couleur, a donné des résultats très satisfaisants. De plus cette méthode est performante et simple à mettre en œuvre.

Nous espérons à l'avenir enrichir notre outil en ajoutant de nouvelles fonctionnalités comme et à titre non exhaustif : la reconnaissance des régions, image couleur dynamique, hybridation de méthodes ou complètement changer de stratégie en raisonnant directement sur la couleur.

Mots clés : *Image texturée couleur, ACP, classification floue, Filtre de Gabor*

Table des matières

Introduction générale

Chapitre 01 : Analyse et segmentation

1.1	Introduction.....	10
1.2	Analyse de l'image	10
1.2.1	Analyse statistique	10
1.2.1.1	Modélisation Markovienne	10
1.2.1.2	Analyse par extraction des paramètres.....	10
1.2.2	Analyse par transformation.....	11
1.2.3	Analyses fractales	11
1.2.4	Analyse en composantes principales <ACP>	11
1.2.5	Analyse de l'homogénéité	11
1.3	Définition de la segmentation	11
1.4	Les approches de segmentation d'images	12
1.4.1	Approche contour	13
1.4.2	Segmentation par seuillage	14
1.4.3	Approche région	15
1.4.3.1	Méthode Par Fusion (Bottom-Up)	15
1.4.3.2	Méthode Par Division (Top-Down)	15
1.4.3.3	Méthode par division/fusion (Split And Merge).....	16
1.4.4	Segmentation par classification	16
1.4.4.1	La Classification	16
1.4.4.2	Méthodes de Classification	17
1.5	Domaine d'application de la segmentation	22
1.6	Conclusion	23

Chapitre 02 : Généralités sur la texture couleur

2.1	Introduction.....	24
2.2	Définition de la texture	24
2.3	Modèle de la texture.....	24
2.3.1	Texture structurelles	24
2.3.2	Texture aléatoire	25
2.3.3	Texture directionnelles	25
2.4	Analyse de texture.....	25

2.5	Méthodes d'analyse de texture.....	26
2.5.1	Méthodes structurelles.....	26
2.5.2	Méthodes statistiques.....	26
2.5.2.1	Matrices de cooccurrences.....	27
2.5.2.2	Matrices de longueurs de plages.....	28
2.5.2.3	Méthodes basées sur la Transformée de Fourier.....	29
2.5.3	Méthodes géométriques.....	30
2.5.4	Méthodes spatio-fréquentielles.....	30
2.5.4.1	Transformation de Gabor.....	30
2.5.4.2	Transformée en ondelettes.....	31
2.5.5	Méthodes basées sur la modalisation spatiale des textures.....	31
2.5.5.1	Modélisation par champ de Markov.....	32
2.5.5.2	Modélisation autorégressive.....	33
2.5.6	Méthodes fractales.....	34
2.6	Conclusion.....	35

Chapitre 03 : Segmentation d'image texturée couleur par un banque de filtre de Gabor

3.1	Introduction.....	36
3.2	Schéma bloc de la méthode.....	37
3.2.1	Réduction de dimensions des données par l'ACP.....	38
3.2.2	Application de la banque de filtres de Gabor.....	42
3.2.3	Extraction des régions texturées.....	43
3.2.4	La classification par méthodes de fuzzy c-means.....	44
3.3	Conclusion.....	48

Chapitre 04: Implémentation

4.1	Présentation des données.....	49
4.2	Images de la base VisTex.....	49
4.3	Environnement matériel et logiciel de programmation.....	51
4.5	Présentation de quelques vues.....	51
4.6	Résultats et discussions.....	58

Conclusion général

Bibliographie

Introduction générale

La couleur d'un objet, telle que nous le percevons, est une notion très complexe qui obéit à des lois physiques, physiologiques et psychologiques particulières. Elle peut être considérée comme le résultat d'une combinaison des quatre éléments suivants : source lumineuse, le matériau, le récepteur et le système d'interprétation.

Le concept d'analyse d'images peut se décliner en diverses thématiques, allant de la photogrammétrie à la segmentation d'objets, Nous nous intéresserons à la caractérisation de points et à la segmentation d'images sur un fond de complexité.

L'analyse d'images a pour finalité l'extraction de l'information caractéristique résidant dans une image. Cette analyse donne une spécification structurelle de l'image pouvant se ramener à une image ou à toute structure de données ('niveau de gris, couleur, forme, texture, ...). L'analyse d'image est liée à la segmentation attachant à chaque pixel une étiquette. La segmentation consiste à diviser une image en plusieurs parties et est typiquement utilisée pour identifier les objets ou des informations importantes dans l'image numérique.

La littérature abonde d'approches différentes ayant trait à des techniques d'analyse et de segmentation. Ces dernières ne prennent pas en charge les images texturées.

Dans ce contexte, nous adoptons une démarche spécifique qui essaie de pallier à ce problème.

Ainsi notre travail s'intitule : « Classification floue d'images texturées couleurs ».

Notre mémoire est organisé en 4 chapitres comme suit :

Le premier chapitre dresse un panorama des différentes approches courantes d'analyse et de segmentation d'image couleur et statique.

Dans le deuxième, sont introduits le concept de texture de l'image couleur, les différentes techniques d'analyse associées.

Le troisième présente l'approche adoptée, en mettant en avant le filtre de Gabor et la classification par une technique floue.

Quant au quatrième chapitre, il décrit les détails d'implantation de l'approche, en présentant les différentes étapes sous forme d'écrans.

Une conclusion suivie de perspectives termine notre mémoire.

Chapitre 1

1.1 Introduction

Une bonne segmentation de l'image vient d'une bonne analyse qui est une étape très importante dans le traitement d'images, car c'est à partir de l'image segmentée que les mesures sont effectuées pour l'extraction des paramètres discriminants en vue de la classification ou de l'interprétation. La tâche de segmentation peut se résumer de la manière suivante : étant donnée une image, l'objectif de la segmentation est d'établir une description compacte et représentative de son contenu informationnel, plus exploitable que l'ensemble de ses points.

Dans ce chapitre, nous allons dresser un panorama des méthodes d'analyse et de segmentation d'images courantes, nécessaires à la compréhension du domaine puis de faire quelques commentaires pour justifier nos choix ultérieurs.

1.2 Analyse de l'image

Dans le traitement d'images, la phase de segmentation n'est pas être une partition aléatoire parce qu'elle doit être guidée et répondre à des critères de qualité. Par contre, l'analyse de l'image est une étape décisive sur laquelle va dépendre le résultat de la segmentation.

1.2.1 Analyse statistique

Actuellement, les approches par modélisation sont les techniques les plus utilisées en analyse d'images, comme les modèles par champ de Markov. Mais il existe aussi des méthodes qui utilisent les histogrammes ou les matrices pour faire l'extraction des paramètres.

1.2.1.1 Modélisation Markovienne

Les approches à base de modélisation Markovienne sont les techniques les plus utilisées en segmentation d'images, généralement leur intérêt principal est de faire une modélisation des interactions entre les pixels à travers un modèle probabiliste.

1.2.1.2 Analyse par extraction des paramètres

Ces approches donnent de bons résultats pour des images mono-composantes, mais leur application aux images multispectrales paraît irréalisable.

Ils sont basés sur l'extraction des paramètres, tels que les paramètres de la matrice de cooccurrence, les paramètres des histogrammes locaux ainsi que les paramètres de la matrice de longueurs de plages.

1.2.2 Analyse par transformation

La transformation de l'image elle est la base de ces techniques. La transformée de Fourier, filtrage par un banc de filtres de Gabor sont les plus utilisées. Ces approches sont basées sur l'analyse fréquentielle, elles sont largement utilisées dans la caractérisation de la texture dans les images en niveaux de gris. Ces méthodes donnent une représentation spectrale de l'image, c'est-à-dire une bijection entre l'espace spatial et son espace fréquentiel.

1.2.3 Analyses fractales

Le but général de ces techniques d'analyse c'est l'estimation des propriétés sur la surface de l'image.

Une droite a une dimension de un et une surface à une dimension de deux. Il existe des objets ayant des dimensions fractionnaires, que l'on appelle fractales.

1.2.4 Analyse en composantes principales <ACP>

L'analyse en composantes principales est fondée sur la réduction de l'espace des attributs. Elle calcule dans un premier temps la matrice de corrélation T des N_a attributs. Dans un second temps, elle recherche les valeurs propres λ_i et les vecteurs propres v_i de T . Pour finir, elle sélectionne parmi les N attributs les N_{ACP} qui ont la plus grande valeur propre.

1.2.5 Analyse de l'homogénéité

Elle est très utilisée en segmentation d'images. L'homogénéité est apparentée à l'information locale extraite d'une image et reflète le degré d'uniformité d'une région donnée. Cette technique joue un rôle très important dans la segmentation d'images du moment que le résultat de la segmentation serait plusieurs régions homogènes.[3]

1.3 Définition de la segmentation

Il n'existe pas une seule ou une définition précise de la segmentation parce qu'elle souvent confondue avec la classification.

Segmenter une image signifie, trouver ses régions homogènes et ses contours, c'est-à-dire que les régions doivent correspondre aux parties significatives des objets du monde réel, et les contours à leurs frontières apparentes.

Soit 'X' le domaine de l'image et 'f' la fonction qui associe à chaque pixel une valeur. Si nous définissons un prédicat 'P' sur l'ensemble des parties de 'x', la segmentation de 'x' est définie comme une partition de 'x' en 'x' sous-ensembles : $\{R_1, R_2, \dots, R_n\}$ tels que :

- $X = \bigcup_i R_i$
- $\forall i, j = \overline{1, n} R_i$ est connexes
- $\forall i, j ; i \neq j R_i \cap R_j = \emptyset$
- $\forall i, j = \overline{1, n} P(R_i) = \text{vrai}$
- $\forall i, j ; i = \overline{1, n, j} = P(R_i \cap R_j) = \text{faux}$

R_j et R_i étant deux régions voisines et 'P' est un prédicat d'homogénéité.

Ce qui se traduit par :

- L'union de toutes les régions nous donne l'image initiale.
- Les pixels appartenant à une même région sont connexes.
- Deux régions différentes sont disjointes.
- Dans une région donnée, chaque pixel satisfait le critère d'homogénéité.
- Deux régions différentes sont hétérogènes.[8]

1.4 Les approches de segmentation d'images

Il existe plusieurs techniques de segmentation et on peut les décomposer en quatre grandes catégories :

- Approche contour
- Segmentation par seuillage
- Approche région
- Segmentation par classification

Le schéma de la **Figure 1.1** représente les différentes méthodes de segmentation

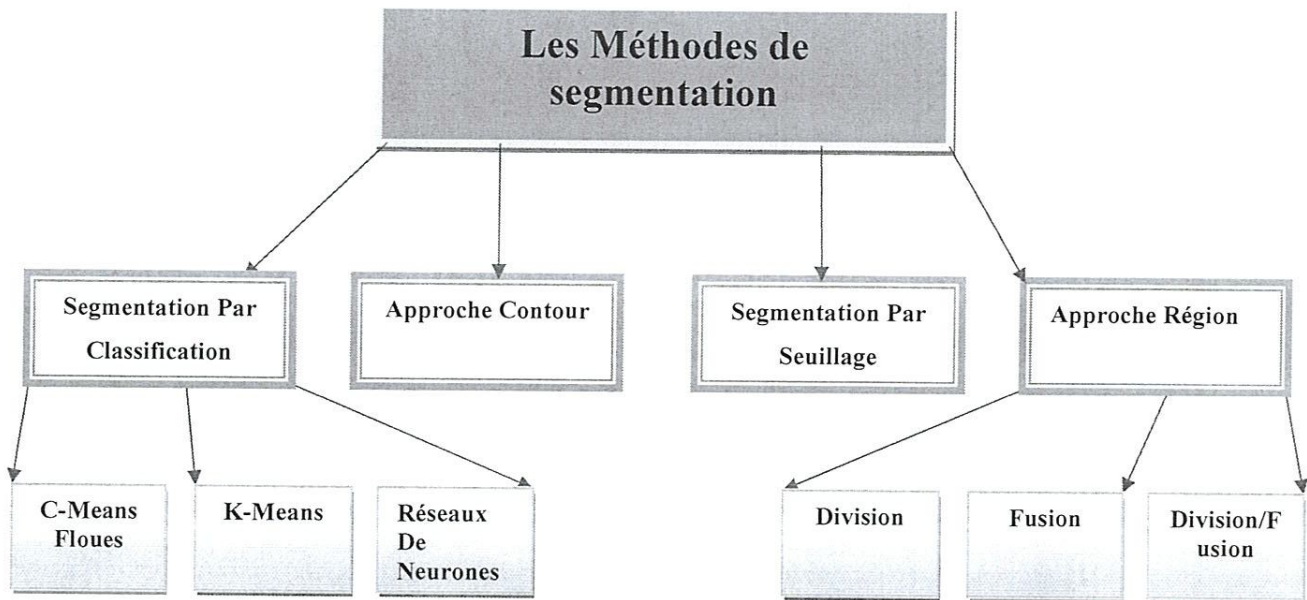


Figure 1.1 : Schéma représentant les techniques de segmentation.

1.4.1 Approche contour

Généralement on peut définir un contour comme étant la frontière entre deux régions, la détection du contour est équivalente à trouver les zones de variations significatives d'intensité lumineuse (niveaux de gris) ou de couleur dans l'image. Nous pouvons citer les approches se basant sur les différences finies comme l'opérateur de gradient, l'opérateur Lapacien et les différents filtres à savoir : le filtre de Sobel, Prewitt et Roberts ou bien le filtre de Canny. Mais ces techniques est peu exploitable parce'elles donnent des contours bruités ou non détectés ; une utilisation des propriétés des régions comprise entre ces contours pourrait nettement améliorer la détection de ce dernier. La **Figure 1.2** illustre les résultats obtenus avec quelques filtres.[7]

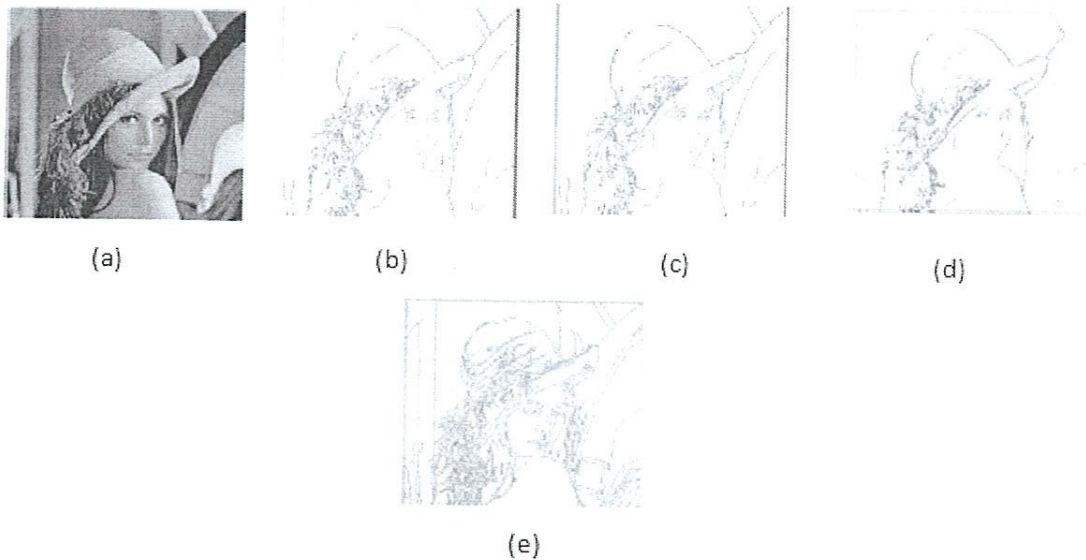


Figure 1.2 : Illustration de détection de contours : a-image lenna originale

b- détection de contour par filtre de Sobel. c- détection de contour par filtre de Prewitt .d- détection de contour par filtre de Roberts. e- détection de contour par filtre de Canny.

1.4.2 Segmentation par seuillage

Le seuillage est une technique simple et efficace au but de séparer les objets du fond. Toutefois, la difficulté apparaît dans le choix des seuils optimaux pour une séparation optimale. Soit 'I' l'image originale et 'Is' l'image segmentée. Segmenter I en N classes revient à chercher N-1 seuils. La classification de chaque pixel est définie comme suit :

$$I_s(x, y) = k \text{ si } k \leq I(x, y) < S_{k+1} \text{ pour } k = 0, \dots, n$$

Où x, y sont les coordonnées du pixel ; S_0, \dots, S_N sont les différents seuils de segmentation.

A titre d'exemple, nous allons présenter un seuillage manuel qui comporte quatre étapes :

1. Observation de l'histogramme de l'image
2. Choix des seuils dans les vallées
3. Définition des classes de régions par intervalle de couleurs
4. Classification des pixels

1.4.3 Approche région

Les méthodes de segmentation base région peuvent être classées en trois catégories que nous allons détaillées ci-dessous :

1.4.3.1 Méthode Par Fusion (Bottom-Up)

L'algorithme commence à partir de petits éléments de l'image qu'il va essayer de regrouper en éléments plus importants. Le critère d'homogénéité locale qui exprimé la similarité entre deux régions adjacentes, c'est la base de la décision de fusion.

Les inconvénients de cette méthode se situent à deux niveaux :

- Cette méthode dépend du critère de fusion qui peut influencer sur le résultat final de la segmentation.
- Il peut introduire l'effet de sous-segmentation.

1.4.3.2 Méthode Par Division (Top-Down)

Ils consistent à diviser l'image d'une manière récursive tant qu'un critère d'homogénéité sur les régions n'est pas vérifié. Initialement, l'image est divisée en zones. Chaque zone ainsi créée est elle-même divisée si elle ne vérifie pas le critère d'homogénéité. Ce processus de divisions se poursuit pour toutes les zones nouvellement créées et ne s'arrête que lorsque toutes les zones sont homogènes ou bien leurs tailles sont en dessous d'un seuil fixé préalablement généralement. La divisions de l'image est représentée par une structure à arbre quaternaire « quadtree » ou la structure de Voronoi. La **Figure 1.3** montre le principe de la segmentation par division suivant la structure d'arbre quaternaire.[2]

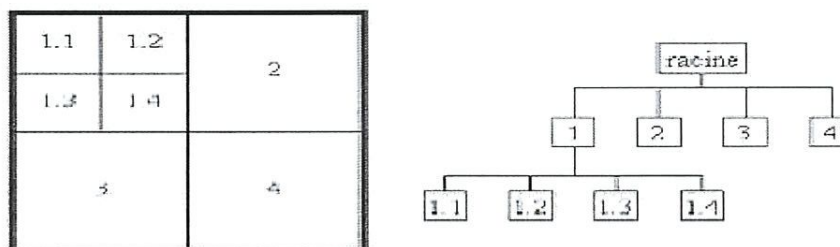


Figure 1.3 : Principe de la segmentation par division des régions (Structure quaternaire)

1.4.3.3 Méthode par division/fusion (Split And Merge)

Ces méthodes regroupent les deux méthodes décrites précédemment, la division de l'image en de petites régions homogènes, puis la fusion des régions connexes et similaires au sens d'un prédicat de regroupement. On part du principe que chaque pixel représente à lui seul une région. Deux régions seront fusionnées si elles répondent aux critères de similarité des niveaux de gris et d'adjacence de régions. On s'arrête quand le critère de fusion n'est plus vérifié. Les inconvénients de cette méthode se situent à trois niveaux :

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

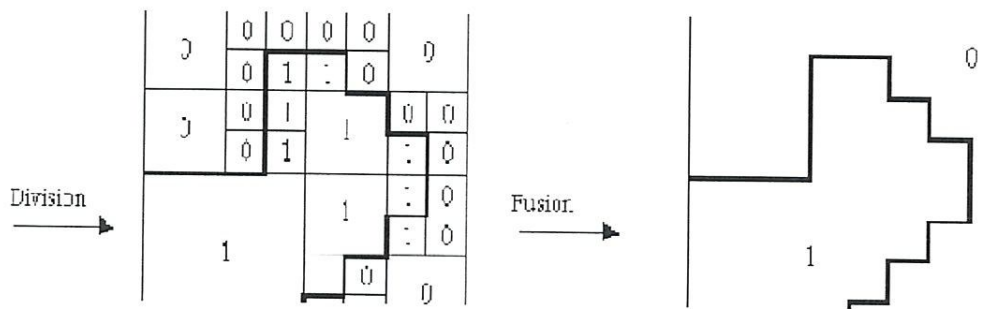


Figure 1.4 : Principe de la segmentation par division-fusion.

1.4.4 Segmentation par classification

1.4.4.1 La Classification

Classifier une image peut être défini comme attribuer à chaque pixel une classe (étiquette). La classification peut être représentée comme une fonction, qui associe à chaque pixel p l'indice k de la classe C_k à laquelle il appartient. De même que pour les régions R_k , le contenu des classes C_k doit être homogène (par exemple, les pixels qui composent une classe doivent avoir des valeurs f similaires).

Notons que les problèmes de segmentation et classification sont très liés et peuvent être utilisés pour signifier la même chose : Un classifieur implicitement segmente une image : L'ensemble des pixels ayant la même classe forme une région de l'image. Une segmentation implique une classification (les régions sont étiquetées selon leurs appartenances).[1]

1.4.4.2 Méthodes de Classification

Les méthodes de classification ont pour but de regrouper les éléments d'un ensemble $X = \{X_1, \dots, X_n, \dots, X_N\}$ en un nombre C optimal de classes selon leurs ressemblances.

De manière générale, les problèmes de classification s'attachent à déterminer des procédures permettant d'associer un objet (individu) à une classe. Ces problèmes se déclinent essentiellement en deux variantes selon Bezdek : la classification dite " supervisée " et la classification dite " non supervisée ".

La classification, supervisée ou non, en tant que discipline scientifique, n'a été automatisée et massivement appliquée que relativement récemment. Comme la plupart des activités scientifiques, l'essor des différentes techniques de classification a largement bénéficié de l'avènement et du perfectionnement des outils informatiques. De nos jours, la classification est une démarche qui est appliquée dans d'innombrables domaines. Un autre nom possible pour cette branche de la recherche est la typologie, et la science qui lui est associée est la taxinomie. Les méthodes de classification ont pour but de regrouper les éléments d'un ensemble X , de nature quelconque, en un nombre restreint de classes. La qualité de la classification peut être jugée sur la base des deux critères suivants :

- Les classes générées doivent être les plus différentes possibles les unes des autres vis-à-vis de certaines caractéristiques
- Chaque classe doit être la plus homogène possible vis-à-vis de ces caractéristiques.

Si l'on se place dans un espace métrique, en termes d'inertie, le premier critère peut être traduit par le fait que l'inertie interclasse doit être la plus élevée possible, et le deuxième critère traduit la préoccupation d'obtenir une inertie intra-classe la plus petite possible.

1.4.4.2.1 Méthodes supervisées

Ces sont des méthodes dans lesquelles les classes sont connues a priori avant l'opération d'identification des éléments de l'image. Elles demandent une **phase d'apprentissage** sur

l'échantillon représentatif dans le but d'apprendre les caractéristiques de chaque classe et une autre phase pour décider l'appartenance d'un individu à telle ou telle classe.

Nous pouvons donc citer des algorithmes de classification pixels supervisés comme celui de Bayes , Les K plus proches voisins (K-PPV) ou les réseaux de neurones multicouches.[]

1.4.4.2.1.1 Algorithmes de Bayes

Ce sont des méthodes qui reposent sur la théorie de Bayes, représentant une référence théorique pour les approches statistiques de résolution des problèmes de classification. Le principe de cette théorie est le suivant : Soit X_j le vecteur forme dont la classe est inconnue et qu'on veut la déterminer, et soit C_i une hypothèse (X_j appartient à la classe C_i par exemple). On cherche à déterminer $P(C_i/X_j)$.

La probabilité de vérification de C_i -après l'observation de X_j . $P(C_i/X_j)$ est la probabilité postérieure (après la connaissance de X_j) tandis que $P(C_i)$ est la probabilité a priori représentant la probabilité de vérification de C_i pour n'importe quel vecteur forme. Le théorème de Bayes propose une méthode de calcul de $P(C_i/X_j)$ en utilisant les probabilités $P(C_i)$, $P(X_j)$ et $P(X_j/C_i)$ $P(C_i/X_j)$, est donc la probabilité d'appartenance de X_j à la classe C_i , $P(C_i)$ est la probabilité d'apparition de la classe C_i (peut être calculée comme le rapport entre le nombre d'échantillons appartenant à la classe C_i et le nombre total d'échantillons). $P(X_j/C_i)$ peut être considéré comme la probabilité d'apparence de chaque valeur des attributs de X_j dans les attributs des échantillons appartenant à la classe C_i . Alors l'algorithme cherche à déterminer pour chaque élément X_j la classe C_i qui maximise la probabilité de contenir cet élément.[6]

1.4.4.2.1.2 Algorithme K plus proches voisins (K-PPV)

L'algorithme des k plus proches voisins (KPPV) (k-Nearest-Neighbors (kNN) en anglais) est une méthode non paramétrique et supervisée de classification introduite dans .

Elle est largement utilisée en classification d'une manière générale et en segmentation d'images en particulier. Elle repose sur un principe simple et intuitif de regroupement d'individus en fonction de leur voisinage.

L'algorithme des K Plus Proches Voisins se base essentiellement sur les deux éléments principaux suivants :

1. le nombre de cas les plus proches (K) à utiliser et une métrique pour mesurer le plus proche voisin.

2. La valeur de K est spécifiée à chaque utilisation de l'algorithme puisqu'il détermine le nombre de cas existants qui sont considérés pour prédire un nouveau cas.

Le KNN est basé sur le concept de distances. Une métrique est nécessaire pour déterminer les distances, cette dernière est à la fois importante car le choix de métrique influe beaucoup sur la qualité des prédictions et arbitraire du fait qu'il n'existe pas de définition préalable sur ce qui constitue une bonne métrique.

La méthode des k plus proches voisins repose sur le regroupement des pixels en fonction de leur voisinage : chaque point est affecté à la classe la plus représentée parmi ses k plus proches voisins. Cette méthode requiert l'établissement d'une règle de distance et la détermination du nombre de voisins à prendre en considération, ainsi qu'un ensemble d'apprentissage représentant les différentes classes.

1.4.4.2.1.3 Méthodes basées sur les réseaux de neurones artificiels

Un réseau de neurones est constitué d'un certain nombre d'unités élémentaires (neurones) fortement interconnectées (neurones) fonctionnant en parallèle et organisées en couches avec un flux d'informations à anticipation. L'architecture du perceptron multicouche est organisée de la manière suivante : les signaux flux successivement dans les différentes couches de l'entrée à la couche de sortie. Les couches intermédiaires sont appelées couches cachées. Pour chaque couche, chaque unité élémentaire calcule un produit scalaire entre un vecteur de poids et le vecteur de sortie donné par la couche précédente. Une fonction de transfert est alors réalisée à la suite de faire une entrée pour la couche suivante. Les poids des connexions sont ajustés au cours du processus d'apprentissage pour obtenir la relation d'entrée/sortie désirée du réseau. De nombreux modèles de réseaux existent (réseaux de Hopfield, perceptrons multicouche, etc.),

Les méthodes de segmentation supervisée offrent l'avantage d'être plus rapides et plus reproductibles que les méthodes non supervisées. Toutefois, leur inconvénient est de rester très dépendant de la base d'apprentissage. C'est pourquoi il est intéressant de développer des algorithmes entièrement automatiques "non supervisés", qui présenteront l'avantage d'offrir un résultat reproductible et indépendant des actions de l'opérateur.[5]

1.4.4.2.2 Méthodes non supervisées

L'intérêt des méthodes non supervisées est qu'elles ne nécessitent **aucun apprentissage** et par là même aucune tâche préalable d'étiquetage manuel. Elles ont pour but de découper

l'espace d'individus (pixels) en zones homogènes selon un critère de ressemblance (critère de proximité de leurs vecteurs d'attributs dans l'espace de représentation entre les individus).

Nous pouvons donc citer des algorithmes de classification pixels non-supervisés comme les C-moyennes proposées par Mac Queen, C-moyennes floues ou algorithmes des C-moyennes possibilistes

1.4.4.2.1 Algorithme des C-moyennes

L'algorithme des C-moyennes (C-means) (centres mobiles) est l'un des algorithmes de clustering les plus connus, il a été mis au point par Mac Queen à la fin des années soixante. L'algorithme des C-moyennes (C-means) réalise un partitionnement automatique des pixels de l'image en C classes homogènes. Soit $X = \{X_j : j=1, \dots, N\}$ un sous-ensemble d'un espace N-dimensionnel. Une partition de X en C classes peut être représentée par des sous-groupes mutuellement disjoints $C_1 \dots C_k$, tels que : $C_1 \cup \dots \cup C_k = X$. Chaque vecteur X_j est assigné à une et seulement une classe qui lui est la plus proche. La fonction objective correspondante s'exprime sous la forme

$$J = \sum_{i=1}^C \sum_{j=1}^n U_{ij} d^2(x_j, b_i)$$

Où: b_i désigne le prototype (centre) de cette classe, d : mesure l'inverse de la similarité entre le prototype et le vecteur de données, U_{ij} : est un indicateur binaire désignant l'appartenance ou non du vecteur x_j à la classe C_i tel que $U_{ij} \in \{0, 1\}$ ($U_{ij} = 1$ si $x_j \in C_i$ et $U_{ij} = 0$ sinon pour $i = 1 \dots C$). L'indicateur d'appartenance U_{ij} s'exprime comme suit :

$$U_{ij} = \begin{cases} 1 & \text{si } d^2(x_j, b_i) = \min\{d^2(x_j, b_k) \mid k = 1 \dots C\} \\ 0 & \text{sinon} \end{cases}$$

L'optimisation se fait de manière itérative. Les étapes de cette classification pouvant être résumées comme suit :

- 1- Initialiser l'ensemble des centres avec des éléments, soit pris au hasard de l'ensemble de données, soit d'après des connaissances a priori
- 2- Assigner chaque point au centre le plus proche.
- 3- Calculer les nouveaux centres des classes construites. Ils sont calculés par la moyenne des éléments de la classe.
- 4- Générer une nouvelle partition de l'ensemble de données en assignant chaque point au centre le plus proche.
- 5- S'il y a un changement dans la partition, réitérer depuis l'étape 3.

1.4.4.2.2 Algorithmes C-moyennes floues ("Fuzzy C-Means" ou FCM)

La modélisation de l'imprécision s'effectue en considérant des frontières graduelles au lieu de frontières nettes entre les classes. L'incertitude s'exprime par le fait qu'un pixel possède aussi bien des attributs qui l'assignent à une classe qu'à une autre. La classification floue assigne donc, non pas à un pixel une étiquette relative à une classe unique, mais son degré d'appartenance à chacune des classes. Ces valeurs expriment l'appartenance incertaine d'un pixel à une région et sont appelées degrés d'appartenance. Le degré d'appartenance se situe dans l'intervalle $[0, 1]$ et les classes obtenues ne sont pas forcément disjointes. Dans ce cas, les données X_j ne sont plus assignées à une classe unique, mais à plusieurs par l'intermédiaire des degrés d'appartenance U_{ij} du vecteur X_j à la classe i . Le but des algorithmes de classification est non seulement de calculer les centres de classe v mais aussi l'ensemble des degrés d'appartenance des vecteurs aux classes.

Si U_{ij} est le degré d'appartenance de X_j à la classe i , la matrice U $C \times N$ est appelée matrice de C-partitions floues si et seulement si elle satisfait aux conditions :

$$\forall j \in [1, N] \left\{ \begin{array}{l} u_{ij} \in [0, 1] \\ 0 < \sum_{j=1}^n u_{ij} < N \end{array} \right.$$

$$\forall i \in [1, c] \sum_{i=1}^c u_{ij} = 1$$

La fonctionnelle à minimiser J (5) et les solutions b_i (6), u_{ij} (7), au problème du FCM sont décrites par les formules suivantes :

$$J(B, U, X) = \sum_{i=1}^c \sum_{j=1}^N (U_{ij})^m d^2(x_j, b_i)$$

$$b_i = \frac{\sum_{j=1}^N (U_{ij})^m \cdot X_j}{\sum_{j=1}^N (u_{ij})^m}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d^2(X_j, b_i)}{d^2(X_j, b_k)} \right)^{\frac{2}{m-1}}}$$

Avec la variable m et le coefficient de fuzzification qui prend ses valeurs dans l'intervalle $[0, +\infty[$. Lorsque m vaut 1, on retrouve l'algorithme classique et lorsque m tend vers l'infini, on obtient une partition infiniment floue où les coefficients U_{ij} tendent vers $1/C$.

L'algorithme de FCM s'arrête lorsque la partition devient stable, c'est-à-dire lorsqu'elle n'évolue plus entre deux itérations successives.

Le FCM souffre de plusieurs inconvénients :

- Les degrés d'appartenance sont des degrés relatifs. Autrement dit, l'appartenance d'un individu à une classe dépend de l'appartenance de cet individu aux autres classes.
- Les fonctions d'appartenance estimations des centres des classes ne correspondent pas aux centres réels ou typiques.
- Les points aberrants (éloignés) peuvent avoir des valeurs d'appartenance élevées et ils peuvent affecter de façon significative l'estimation des centres des classes.

Ces algorithmes modélisent dans la phase de classement de l'hésitation ou de l'ambiguïté entre classes à partir de règles de décisions floues fixées a priori. Dans la phase de classification, les points ambigus ou non représentatifs des classes peuvent alors affecter la position des centres.

1.4.4.2.2.3 Algorithmes les C-moyennes possibilistes

Krishnapuram et Keller ont proposé une approche possibiliste des c-moyennes appelée Possibilistic C-Means, ou PCM. Leur approche est censée conduire à une meilleure performance en présence de bruit. Mais leur travail est motivé essentiellement par le désir de remédier au caractère relatif des degrés d'appartenance générés par les FCM. En effet, ces derniers sont interprétés en tant que degrés de vérité relatifs décrivant l'appartenance d'un vecteur quelconque à chacune des classes possibles. Un élément à classer est donc, en quelque sorte, partagé entre ces différentes classes. A cette idée de partage, Krishnapuram et Keller préfèrent substituer la notion de typicalité. En effet, le résultat d'un regroupement devrait décrire la parenté absolue entre un objet et chacune des C classes possibles, indépendamment du lien entre cet objet et les (C-1) classes restantes.[3]

1.5 Domaine d'application de la segmentation

Le système de traitement d'images présenté possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que : Médecine (Radiographies, tomographies, ...), Biologie, Météorologie, Astronomie, Géologie, Physique (spectroscopie, physique des

plasmas,...), Applications militaires, Applications industrielles (Robotique, surveillance de qualité, ...), Photographies, Publicité, etc.

1.6 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art des méthodes d'analyses d'images couleurs en vue de réduire leurs complexités. De même, nous avons présenté les approches directes de la segmentation d'images les plus connues et le domaine d'application de la segmentation. L'ensemble des techniques décrites dans les sections précédentes montre que le problème de la segmentation est bien un problème difficile. Ce qui a nous a poussé à introduire l'analyse des textures afin d'extraire des attributs significatifs

Le chapitre suivant traite des différentes techniques d'analyse d'images texturées couleurs.

Chapitre 2

2.1 Introduction

La texture est un élément important dans le domaine du traitement d'image, c'est une caractéristique déterminante conduisant à la reconnaissance des objets observables. Il existe une multitude de définitions et méthodes d'analyse de texture qu'on va aborder dans ce chapitre.

2.2 Définition de la texture

Il n'existe pas de définition précise de la texture, mais la définition littéraire, dit que la texture est l'agencement spatiale d'un même motif différentes directions de l'espace. Cette définition est limitative car elle ne tient pas compte des informations visuelles qui permettent de décrire la texture qualitativement à l'aide d'adjectifs tels que contractée, grossière, fine, lisse, tachetée, granuleuse, marbrée, régulière ou irrégulière.

2.3 Modèle de la texture

Il existe plusieurs catégories de la texture. Citons les trois principales familles [] :

- Texture structurelles.
- Texture aléatoires.
- Texture directionnelles.

2.3.1 Texture structurelles

On les appelle ainsi car on peut les considérer comme étant la répartition spatiale de motifs élémentaire de base dans différents directions de l'espace suivant une certaine règle de placement

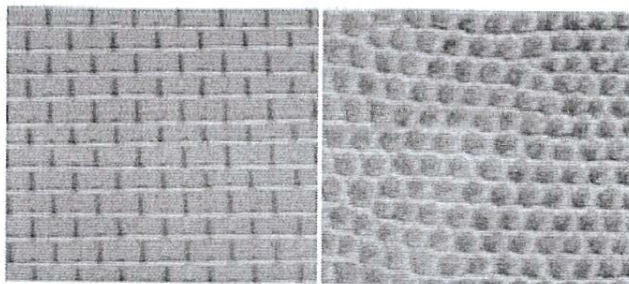


Figure 2.1 : Texture structurelles

2.3.2 Texture aléatoire

Ce type de texture a un aspect anarchique tout en restant globalement homogénéisons. On ne peut pas en extraire de motif de base se répétant spatialement. Cette catégorie a fourni d'autres travaux de recherche plutôt fondés sur des méthodes d'analyse statistique. Dans ces cas-la l'image texturée est considérée comme étant un processus aléatoire bidimensionnel.[2]

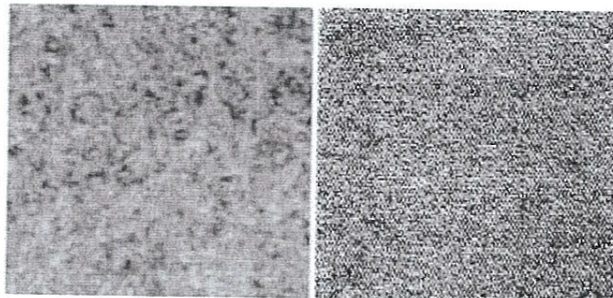


Figure 2.2: Texture aléatoire

2.3.3 Texture directionnelles

Ces texture ne sont pas totalement aléatoires et ne présentent pas d'éléments structurants de base. Néanmoins, elles se caractérisent par certaines orientations.

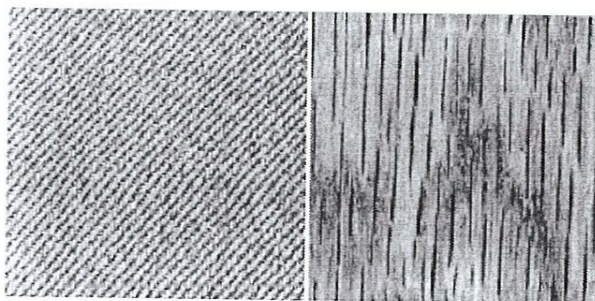


Figure 2.31 Texture directionnelles

La texture de gauche laisse apparaître des lignes obliques, tandis que cette droite des lignes verticales.

2.4 Analyse de texture

L'analyse de texture regroupe un ensemble de techniques mathématiques permettant de quantifier les différents niveaux de gris présents dans une image en termes d'intensité ou de

rugosité et leur distribution. La diversité des images ainsi que la difficulté de donner une définition précise de la texture ont aussi permis l'émergence de plusieurs méthodes d'analyse de la texture.

2.5 Méthodes d'analyse de texture

On distingue cinq types de méthodes :

- Méthodes structurelles
- Méthodes statistiques
- Méthodes géométriques
- Méthodes spatio-fréquentielles
- Méthodes basées sur la modalisation spatiale des textures
- Méthodes fractales

2.5.1 Méthodes structurelles

Les méthodes structurelles cherchent à extraire des primitives des textures et leurs localisations. Ces méthodes s'avèrent donc adaptées aux macrotextures, où une structure spatiale forte se dégage par l'intermédiaire d'un motif plus ou moins répétitif. Principalement, ces méthodes utilisent des techniques d'autocorrélation pour retrouver le placement des primitives des textures initialement extraites, afin d'en déduire une règle de placement. Le trait caractéristique de ces méthodes est que toutes se déroulent en deux étapes, l'extraction de la primitive puis la recherche de la règle de placement.[11]

2.5.2 Méthodes statistiques

Ces méthodes se basent aussi sur les relations entre les pixels et leurs voisins en termes de variation de couleur, les attributs extraits de ces méthodes permettent de caractériser tout type de texture, même les textures fines et sans régularité apparente.

De nombreux descripteurs statistiques sont utilisés dans la discipline de traitement d'images texturées couleur. On trouve entre autre,

- Les statistiques d'image.
- Les histogrammes d'image.
- Les matrices de cooccurrences chromatiques.
- Les histogrammes des sommes et des différences.
- Les matrices de longueurs de plages.

Ces descripteurs peuvent être divisés en plusieurs catégories selon leur ordre. L'ordre du descripteur dépend du type d'interaction spatiale entre les pixels considérés. Par exemple, pour les catégories des descripteurs statistiques du premier ordre.

Les matrices de cooccurrences sont quant-à-elles des descripteurs d'ordre 2 car leur calcul considère des couples de pixels.

2.5.2.1 Matrices de cooccurrences

La matrice de cooccurrence est largement utilisée dans l'analyse de texture. Elle est très facile à mettre en œuvre et donne de bons résultats sur plusieurs types de texture. Dans la plupart des applications, les images utilisées sont représentées en niveaux de gris et la matrice de cooccurrence est connue sous le nom GLCM (Gray Level Co-occurrence Matrix).

La matrice de cooccurrence C d'un vecteur de déplacement (Δ_x, Δ_y) d'une image I de la taille $(m \times n)$ est définie par la formule :

$$c_{\Delta_x, \Delta_y}(x, y) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & \text{si } I(p, q) = i \text{ et } \text{sin on } I(p + \Delta_x, q + \Delta_y) = j \\ 0 & \text{sin on} \end{cases}$$

Pour améliorer la performance de la méthode, nous pouvons réduire la taille de la matrice. Pour le faire, nous pouvons grouper des niveaux gris de l'image par les techniques de partitionnement de données (clustering).

La Figure 2.4 montre la façon dont est calculée la matrice de cooccurrence du vecteur de déplacement $(1,0)$.

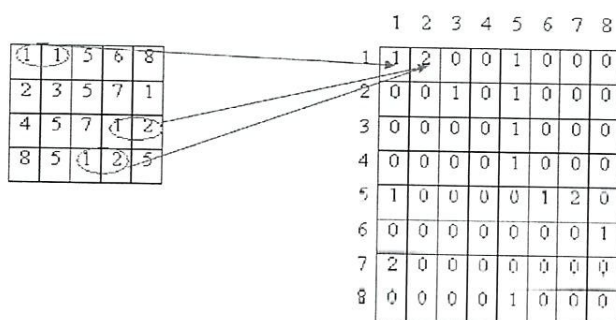


Figure 2.4: calcul la matrice de cooccurrence d'une image

2.5.2.2 Matrices de longueurs de plages

Les longueurs de plages sont une manière de représenter une texture par un ensemble de segments de droites de même luminance et de taille maximale. Cette méthode estime la probabilité d'observer des plages à niveau de gris constant dans une certaine direction.[12]

Sur cette matrice, on calcule également quelques attributs, comme :

- le nombre de longueurs de plages :

$$SLP = \sum_{g=0}^{G-1} \sum_{l=1}^L P_{\theta}(g, l)$$

- la proportion de petites et grandes plages :

$$RF\ 1 = \frac{1}{SLP} \sum_{g=0}^{G-1} \sum_{l=1}^L \frac{p_{\theta}(g, l)}{l^2}$$

$$RF\ 2 = \frac{1}{SLP} \sum_{g=0}^{G-1} \sum_{l=1}^L l^2 P_{\theta}(g, l)$$

- l'hétérogénéité des niveaux de gris:

$$RF\ 3 = \frac{1}{SLP} \sum_{l=1}^L \left(\sum_{g=0}^{G-1} P_{\theta}(g, l) \right)^2$$

Qui mesure la dispersion des plages entre les niveaux de gris,

- l'hétérogénéité des longueurs de plages :

$$RF\ 4 = \frac{1}{SLP} \sum_{l=1}^L \left(\sum_{g=0}^{G-1} P_{\theta}(g, l) \right)^2$$

Qui mesure la dispersion des plages entre les longueurs,

- et le pourcentage de plages:

$$RF\ 5 = SLP/K$$

(Avec K le nombre total de pixels de l'image)

Cette méthode est également utilisée pour le codage d'images et la réduction du nombre de niveaux de gris d'une image.

2.5.2.3 Méthodes basées sur la Transformée de Fourier

La Transformée de Fourier permet, dans le domaine de l'image, d'obtenir une représentation fréquentielle spatiale des données.

$$TDF_I(u, v) = \frac{1}{IJ} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} y(i, j) \exp\left(-2j\pi\left(\frac{ui}{I} + \frac{vj}{J}\right)\right)$$

avec $0 \leq u \leq I - 1$ et $0 \leq v \leq J - 1$.

Cette TFD bidimensionnelle n'est pas toujours facile à interpréter; on s'intéresse principalement à son module, appelé spectre de Fourier. De façon très générale, on peut dire que :

- Si la texture est directionnelle, le spectre de puissance contient des pics sur une ligne perpendiculaire à la direction principale de la texture.
- Si la texture est formée de grains circulaires, le spectre contient des pics sur un anneau circulaire centré sur l'origine.
- Si la texture est périodique, on retrouvera cette périodicité dans le spectre.
- Si la texture est fine, le spectre sera maximum dans les hautes fréquences, et, inversement, si elle est plutôt grossière, le maximum sera proche des basses fréquences; en effet, le spectre de Fourier rend compte de la rugosité d'une texture.

On peut également calculer des attributs de texture sur le spectre de Fourier, comme l'énergie et l'étendue du pic principal, les énergies et entropies relatives de différentes régions carrées centrées sur l'origine du spectre, les moments d'inertie de chaque quadrant, etc.

L'intérêt de ces méthodes travaillant dans le domaine fréquentiel, est surtout lié à la représentation quasi-uniforme du bruit dans ce domaine, alors qu'il peut altérer, de façon dramatique, les variations locales d'intensité dans la représentation spatiale. De même, les attributs statistiques spatiaux sont plus sensibles au bruit que les attributs tirés du spectre de Fourier.

Depuis peu, l'utilisation des représentations temps-fréquence en analyse d'images s'est développée. Le problème crucial reste l'interprétation des données obtenues; les 4 dimensions ne peuvent être visualisées, même si ces représentations peuvent être vues comme une manière d'associer à chaque point de l'image, un « spectre local ». De manière à réduire la quantité de

données, des attributs ou caractéristiques sont calculés sur ces spectres locaux (composantes les plus énergétiques, moments des spectres en coordonnées polaires, amplitude et orientation du pic principal, etc.). Ces méthodes ont un champ d'applications assez restreint pour le moment (textures synthétiques directionnelles, images réelles directionnelles); cependant, leur investigation est loin d'être terminée et, de ce fait, les résultats sont prometteurs.

2.5.3 Méthodes géométriques

Ce type de méthodes tient compte de l'information structurelle et contextuelle de l'image. Ces méthodes sont particulièrement bien adaptées aux textures macroscopiques (structurelles). En effet la description de la texture est faite par une extraction explicite de primitives (primitives : ensemble connexe de pixels qui partagent des appelés attributs géométriques. Cependant, comme ces attributs géométriques sont sensibles à la régularité des motifs texturés présents dans l'image, ils ne peuvent pas caractériser des textures irrégulières comme celles présentes dans la majorité des images naturelles. Par conséquent, ce type de méthodes n'est pas efficace ; on s'intéresse plutôt aux méthodes qui permettent la caractérisation de maximum de type de textures.

2.5.4 Méthodes spatio-fréquentielles

Ces méthodes, préservent à la fois les informations globales et locales. Elles sont bien adaptées aux signaux quasi périodiques, en effet, les textures sont des signaux quasi périodiques qui ont une énergie fréquentielle localisée. Ces méthodes permettent de caractériser la texture à différentes échelles.[13]

2.5.4.1 Transformation de Gabor

La transformation de fourrier d'une image permettra de mettre en évidence les régularités de la texture en examinant le domaine fréquentiel. Le problème posé par cette opération qui agit globalement sur l'image, est qu'elle ne tient pas compte de la localisation spatiale. Une solution est d'utiliser une transformation alternative appelée transformation de fourrier à fenêtre glissante, où le principe est d'appliquer la transformation de fourrier dans une fenêtre d'observation de dimensions fixes qui se déplace dans l'image. Le choix de la taille de la fenêtre et du pas de déplacement dépendent plus exactement des caractéristiques spatiales des textures à analyser.

Il existe différentes fenêtres d'observation. Quand la fenêtre de Gausse est utilisée, on parle alors de la transformation de Gabor [12]

L'inconvénient principal de cette méthode est le réglage des paramètres des filtres. De plus, pour caractériser certaines textures, il est nécessaire d'avoir une fenêtre d'analyse assez grande, ce qui peut impliquer un temps de calcul assez élevé [14].

2.5.4.2 Transformée en ondelettes

La transformation de Gabor se base sur une fenêtre d'observation de dimension fixe, ce qui peut être un handicap car certaines textures peuvent être caractérisées selon différentes échelles (ex : texture grossières). Pour pallier cette limite, la transformée en ondelettes à différentes tailles sont utilisées [13].

Pour caractériser la texture couleur, de nombreux attributs statistiques ou de cooccurrences peuvent être extraits des transformées en ondelettes couleur d'une image et également par le calcul des énergies concernant chaque filtre.[11]

L'avantage de cette transformée est qu'elle permet une caractérisation multi échelles de la texture en considérant à la fois l'information globale et locale contenues dans l'image. De plus, les attributs extraits des ondelettes se rapprochent du processus de la vision humaine qui opère en une décomposition fréquentielle systématique des images lorsque celles-ci parviennent sur la rétine.

Finalement, la sélection des attributs extraits des ondelettes est diverse selon les auteurs. Certains auteurs suggèrent que ces attributs ne sont pas toujours les plus adéquats pour caractériser les textures couleur, les autres, au contraire, montrent que ces attributs sont plus efficaces quand ils sont judicieusement exploités [6].

2.5.5 Méthodes basées sur la modalisation spatiale des textures

Les méthodes basées modèle supposent que la texture est formée par un modèle et on considère que les paramètres de ce modèle sont des caractéristiques de la texture.

Un bon nombre de modèles sont issus de la théorie des processus stochastiques. Une texture est, dans ce cas, la réalisation d'un processus aléatoire stationnaire et ergodique.

On peut citer :

- les modèles markoviens,

- les modèles utilisant un filtre linéaire excité par un bruit blanc, avec, comme cas particulier, les modèles autorégressifs (AR ou ARMA), dits « paramétriques »

2.5.51 Modélisation par champ de Markov

Utilisés depuis de nombreuses années en mécanique statistique pour modéliser, par exemple, des phénomènes ferromagnétiques, les champs aléatoires de Markov sont appliqués au traitement d'images. L'idée de base de cette modélisation est que la connaissance du voisinage d'un pixel s est suffisante pour calculer sa loi conditionnelle.

Leur utilisation, en analyse d'images, repose, essentiellement, sur le théorème d'Hammersley-Clifford qui donne une équivalence entre champ de Markov et distribution de Gibbs.

Le choix de la forme de U détermine le modèle paramétrique; on peut citer :

- le modèle d'Ising, un des plus simples qui est utilisé pour des images binaires,
- le modèle de Graffigne, dont les potentiels sont de la forme :

$$V_c(x_s) = \frac{\theta_c}{1 + (x_s - x_t)^2 / 8}$$

- le modèle des filtres propres,
- le modèle gaussien pour lequel la densité de probabilité conditionnelle est gaussienne.

Un des problèmes est lié à la simulation des champs de Markov. On utilise des algorithmes stochastiques permettant de visiter les sites une infinité de fois et de converger vers la probabilité $P(x)$ choisie; les plus courants sont l'algorithme de Métropolis et l'échantillonneur de Gibbs; le recuit-simulé (Simulated Annealing) étant plus précisément un algorithme de restauration. Quelques algorithmes déterministes, comme l'algorithme ICM (iterated Conditional Mode), sont, de plus en plus utilisés, en optimisation, de manière à pallier l'inconvénient majeur de l'utilisation des champs markoviens : le coût prohibitif de calcul.

A noter, qu'en segmentation, une modélisation doublement stochastique est utilisée : un au champ des étiquettes des régions. L'énergie de Gibbs contient alors des termes d'interaction entre les 2 processus dont le poids relatif peut varier au cours des itérations.

Le problème délicat de la complexité calculatoire de cette méthode de modélisation est important; comme nous venons de le voir, l'utilisation d'algorithmes déterministes et leur coopération avec

des méthodes statistiques dans le cadre de modélisations stochastiques permettent de pallier, partiellement, cet inconvénient.

2.5.5.2 Modélisation autorégressive

La modélisation autorégressive joue un grand rôle en traitement du signal. Du fait de leur bon compromis complexité-adéquation, les méthodes d'identification et d'optimisation de ces modèles ont été particulièrement développées et testées. A noter que cette identification est limitée au second ordre.

Considérer une image de texture comme la réalisation d'un modèle autorégressif revient à considérer la variable aléatoire attachée à un site de l'image comme linéairement dépendante des variables aléatoires attachées aux sites voisins. Soit Y une image et S l'ensemble des sites s de Y :

$$s = \{s = (i, j); 1 \leq i \leq I, 1 \leq j \leq J\}$$

Y est la réalisation d'un modèle AR 2D si :

$$y_s = y(i, j) = \sum_{(m, n) \in D} a(m, n, i, j) y(i - m, j - n) + b e(i, j)$$

Avec :

- e : variable aléatoire qui définit le type du modèle; si e est un bruit blanc, le modèle est un modèle AR (Autoregressive) et si e est un bruit corrélé, le modèle est un modèle
- ARMA (Autoregressive Moving Average)
- b : écart-type de e
- D : ensemble de prédiction du modèle dont la forme dépend du type de causalité choisie
- a : paramètres du modèle; si $a(m, n, i, j) = a(m, n)$, le modèle est stationnaire

Un des points cruciaux pour l'utilisation de ces modèles est, ici, mis en évidence : le choix du voisinage et, par conséquent, le type de causalité du modèle.

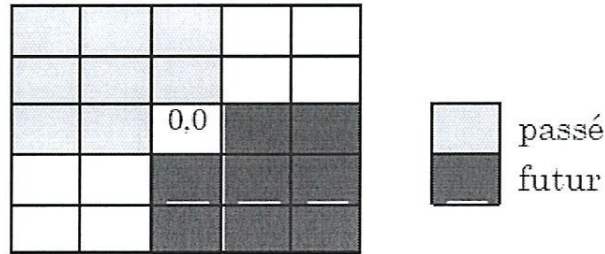


Figure 2.5: Exemple de support

Le nombre de voisins, donc l'étendue spatiale du voisinage, détermine le nombre de coefficients du modèle. Ceux-ci doivent être estimés sur la texture et la méthode d'estimation choisie joue un rôle dans la qualité de représentativité du modèle AR. La méthode du maximum de vraisemblance (MLE) donne généralement de bons résultats, mais est très coûteuse en temps de calcul et nécessite certaines hypothèses de travail. Dans le cas d'un signal 1D, l'hypothèse gaussienne suffit pour obtenir une équivalence entre estimation par moindres carrés (LSE) et par MLE. Pour des signaux 2D, il faut, en outre, que le signal soit causal. Dans la pratique, l'estimation par LSE est assez largement préférée pour son bon compromis complexité/performance.

De manière plus précise, on peut rappeler que la résolution des équations normales bidimensionnelles (c'est à dire l'algorithme de moindres carrés) a été le premier algorithme d'estimation des paramètres d'un modèle AR 2D; il est, encore, très utilisé car il reste l'algorithme le plus facile à implémenter et donnant de bons résultats.

2.5.6 Méthodes fractales

Ces méthodes permettent de synthétiser des images très proches de la réalité. En analyse de texture, la dimension fractale, qui est une mesure du degré d'irrégularité d'un objet, décrit une certaine propriété de la texture. Le modèle fractal est basé essentiellement sur l'estimation par des méthodes spatiales de la dimension fractale de la surface représentant les niveaux de gris de l'image.

2.6 Conclusion

Dans ce chapitre, nous avons présenté des notions sur quelques méthodes d'analyse de textures d'image en couleurs. Ces dernières sont très nombreuses mais il n'existe pas de méthode parfaite. Parmi toutes ces méthodes, nous nous sommes intéressés aux méthodes spatio-fréquentielles, particulièrement la transformation de Gabor. Le chapitre 3 présente notre approche.

Chapitre 3

3.1 Introduction

Notre travail, est axé sur les images en couleurs demandant un traitement sur plusieurs composantes, est réduit par une fusion des trois composantes (R, V, B) pour obtenir une image mono composante, c'est-à-dire qu'on minimise la taille des images via la méthode ACP. Le processus continue en appliquant une banque du filtre de Gabor sur l'image résultante de l'étape précédente et on termine par la classification floue via la méthode FCM. Nous allons décrire dans les paragraphes suivants chacune des étapes.

3.2 Schéma bloc de la méthode

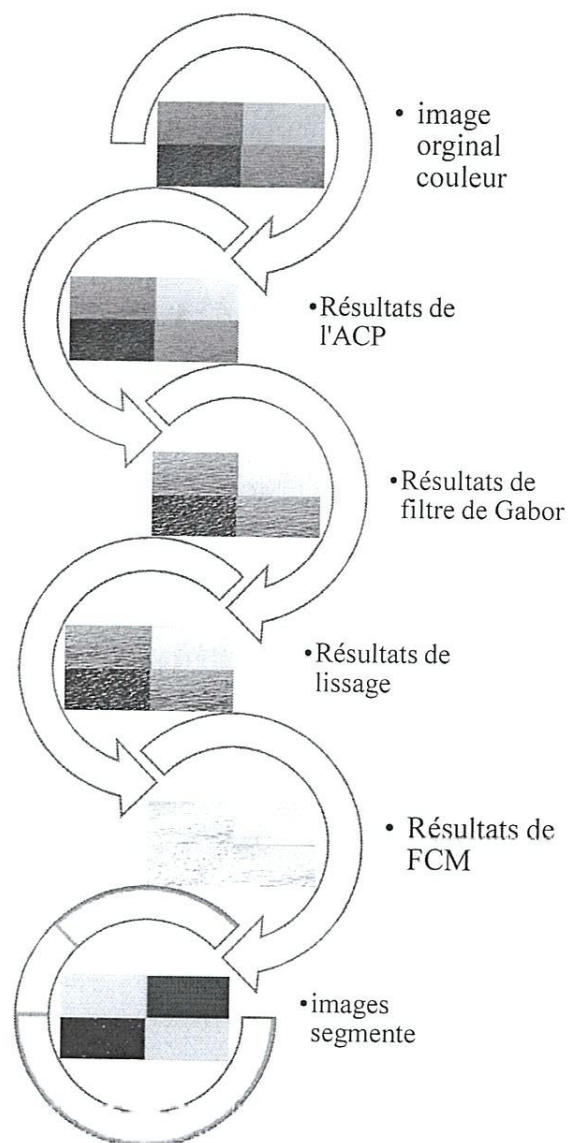


Figure 3.3 : schéma bloc de la méthode

3.2.1 Réduction de dimensions des données par l'ACP

L'ACP est un outil qui devient rapidement indispensable en traitement d'image dès lors que l'on travaille sur des images multi-spéctrales. Il permet de réduire d'une manière significative le nombre d'images avec très peu de perte d'informations.

La procédure de calcul des composantes principales de l'image multi-composantes nécessite le calcul de la moyenne simple de chaque image composante et de la matrice de covariance comme suit :

- **Calcul de la moyenne**

On calcule la moyenne m_x de chaque image composante par cette expression de la moyenne simple :

$$m_x = \frac{1}{MN} \sum_{k=1}^{MN} x_k$$

Dans notre cas d'images couleur, on aura trois moyennes (m_r, m_v, m_b)

- **Calcul de la matrice de covariance**

On calcule également la matrice de covariance c_x de taille $n * n$ de l'image originale à l'aide de cette expression :

$$c_x = \frac{1}{MN - 1} \sum_{k=1}^{MN} (x_k - m_x)(x_k - m_x)^t$$

Concernant les images couleur, la taille de la matrice de covariance est de 3*3.

Notons que la matrice de covariance est réelle et symétrique. Elle peut contenir jusqu'à n vecteurs propres orthogonaux et les éléments de sa diagonale représentent les variances des images composante.

- **Calcul des composantes principales**

On définit la matrice A de taille maximale $n \times n$ des vecteurs propres normalisés de la matrice de covariance c_x . Après la réorganisation de la matrice A , de vecteur (ligne) propre ordonné par ordre décroissant (du plus important au moins important). On aboutit à la composante principale y à l'aide de l'expression suivante :

$$y = A(x - m_x)$$

Les éléments de la matrice y de taille maximale $M \times N \times n$ sont décorrélés, par conséquent la matrice de covariance c_y de la matrice des composantes principales y sera diagonale et les valeurs de cette diagonale représentent les valeurs propres correspondantes aux vecteurs propres contenus dans la matrice A . Les autres éléments représentent les corrélations partielles qui sont nulles.

L'importance de l'analyse en composante principale c'est que nous qui va choisir le nombre de vecteurs propre q à utiliser soit pour décorréler les n composantes alors dans ce cas on utilise tous les vecteurs propre c'est-à-dire $q = n$ et on aura un nouveau espace de représentation de la couleur appelé espace d'HOTELING, ou de réduire les n composantes à un nombre de composantes inférieur, alors dans ce cas $1 \leq q < n$. La matrice A des vecteurs propres devient A_q de taille $q \times n$.

On a l'exemple suivant pour bien illustrer toute cette théorie, on a une image couleur (RVB) de taille 4×4 et $n = 3$.

R:118 G: 96 B: 54	R:129 G: 92 B: 54	R:198 G: 91 B: 50	R:170 G: 90 B: 45
R:145 G: 83 B: 41	R:139 G: 79 B: 33	R:164 G: 84 B: 45	R:157 G: 89 B: 56
R:147 G: 83 B: 41	R:214 G: 79 B: 33	R:117 G: 84 B: 45	R:134 G: 89 B: 56
R:133 G: 85 B: 44	R:158 G: 87 B: 40	R:143 G: 86 B: 45	R:122 G: 86 B: 50

Figure 3.4 : image couleur RVB de taille 4*4

La matrice de covariance c_x :

$$c_x = \begin{bmatrix} 699.1875 & -25.2813 & -68.3750 \\ -25.2813 & 19.9844 & 25.8438 \\ -68.3750 & 25.8438 & 49.4375 \end{bmatrix}$$

On voit bien qu'elle est réelle et symétrique, maintenant le calcul de la matrice des vecteurs propres A .

$$A = \begin{bmatrix} -0.9937 & 0.0405 & 0.1048 \\ -0.1105 & -0.5224 & -0.8455 \\ 0.0205 & -0.8518 & 0.5235 \end{bmatrix}$$

Dont :

Le premier vecteur propre est le vecteur ligne en haut de la matrice A , le deuxième et le troisième se suivent dans cet ordre.

Pour faire de la décorrélation, alors on les utilise tous, c'est-à-dire $q = 3$. on aura la matrice des composantes principale suivante :

$$y = \begin{bmatrix} -0.9937 & 0.0405 & 0.1048 \\ -0.1105 & -0.5224 & -0.8455 \\ 0.0205 & -0.8518 & 0.5235 \end{bmatrix} \begin{bmatrix} -31.2500 & -2.2500 & \dots - 27.2500 \\ 9.6250 & -3.3750 & \dots - 1.3750 \\ 8.2500 & -4.7500 & \dots 4.2500 \end{bmatrix}$$

$$= \begin{bmatrix} 32.3066 & 3.5884 & \dots 27.4673 \\ -8.5504 & 6.2489 & \dots 0.1358 \\ -4.5208 & 0.3006 & \dots 2.8365 \end{bmatrix}$$

La taille de la matrice Y est $3 \times 16 (n \times MN)$. Après sa normalisation entre $[0,1]$ puis entre $[0,255]$ on retrouve la nouvelle image qui est constituée de trois nouvelles images composantes (représentation des couleurs dans un nouveau espace $(R'V'B')$ suivante :

R:255 G:149 B:159	R:226 G:151 B:169	R: 47 G:142 B:169	R:118 G:162 B:163
R:180 G:187 B:172	R:193 G:212 B:170	R:133 G:172 B:176	R:154 G:143 B:180
R:175 G:187 B:172	R: 0 G:191 B:174	R:254 G:185 B:174	R:214 G:150 B:178
R:212 G:182 B:171	R:147 G:180 B:162	R:187 G:175 B:171	R:242 G:172 B:179

Figure 3.5 : Image couleur $R'V'B'$ de taille 4×4 décorrelée.

Sa matrice de covariance c_x est :

$$c_x = \begin{bmatrix} 707.4321 & 0.0000 & 0.0000 \\ 0.0000 & 56.4683 & 0.0000 \\ 0.0000 & 0.0000 & 4.7091 \end{bmatrix}$$

Maintenant pour faire la réduction de dimension comme dans notre cas, on utilise le vecteur propre le plus important qui se trouve en première ligne de matrice Λ .

Cette fois ci on aura le vecteur Y (pas la matrice Y) des composantes principale suivant :

$$Y = [-0.9937 \quad 0.0405 \quad 0.1048] \begin{bmatrix} -31.2500 & -4.2500 & \dots & -27.2500 \\ 9.6250 & -3.3750 & \dots & -1.3750 \\ 8.2500 & -4.7500 & \dots & 4.2500 \end{bmatrix} =$$

$$[32.3066 \quad 3.5884 \quad \dots \quad 27.4673]$$

Après sa normalisation entre $[0,1]$ puis entre $[0,255]$ on aura l'image monochromatique suivante :

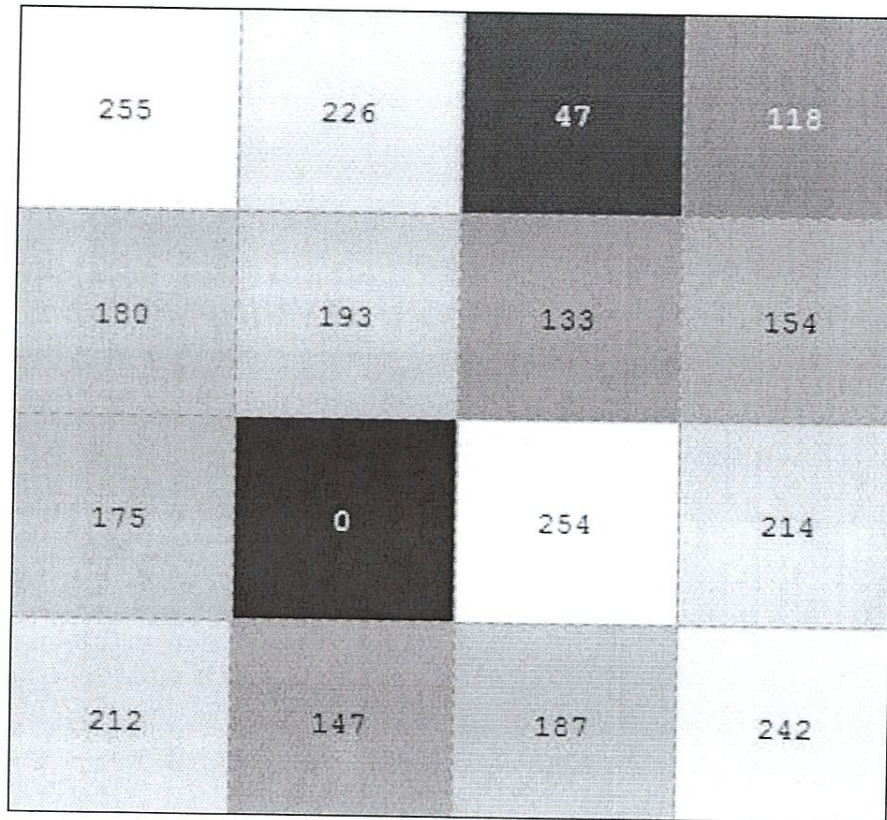


Figure 3.6: Image résultante L'ACP avec $q = 1$.

3.2.2 Application de la banque de filtres de Gabor

Cette étape est la plus importante, car la configuration des quatre paramètres (f, γ, b, θ) (φ est toujours nulle dans notre méthode) des filtres se fait à ce niveau.

La conception de la banque de filtre de Gabor qu'on utilise dépend du choix optimal du paramètre suivant :

- **Fréquences centrales f :**

Les différentes fréquences centrales de la banque de filtres de Gabor sont déterminées

- **Orientations θ :**

On a choisi les orientations recommandées $\theta = \{0^0, 45^0, 90^0, 135^0\}$ pour deux raison :

- Diminution du nombre de filtre de Gabor, ce qui entrainera un temps de réponse plus petit et une classification de pixel plus exacte.
- Résultats obtenus en utilisant les images tests avec ces orientations sont pratiquement identiques par rapport aux résultats obtenus avec ces orientations.
- b, γ :

La valeur de la largeur b de la bande de fréquences de chaque filtre est souvent 1 octave ($1 \text{ octave} = \log \frac{f_2}{f_1}$, f_1 et f_2 sont successives), cela revient aux expériences qui ont été faites sur l'œil humain.

On aura avec $b = 1, \sigma = 0.56 \times \lambda$.

On choisit souvent la valeur de l'allongement spatial $\gamma, 2$, pour construire une banque de filtre de Gabor a arrangement tangentiel car ce type de banque nous donne des résultats satisfaisants.

3.2.3 Extraction des régions texturées

L'extraction des régions texturées se fait après la non linéaire a l'aide de la fonction non linéaire $\tan(ax) = \frac{1 - e^{-2ax}}{1 + e^{-2ax}}$

Ou a est une constante (la valeur $a = 0.25$ a été choisie).

Afin d'extraction les attributs de la texture, on a appliqué une fonction de lissage Gaussienne sur chaque image filtrée.

La fonction est donnée par la relation :

$$g(x, y) = \exp \left\{ - \frac{x^2 + y^2}{2\sigma^2} \right\}$$

Ou σ est l'écart type qui détermine la taille de la fenêtre

3.2.4 La classification par méthodes de fuzzy c-means

- **L'algorithme du Fuzzy C-Means**

L'algorithme du Fuzzy C-Means est une technique de classification non supervisée, qui a été introduite par J. C. Bezdek et qui a été souvent utilisée dans différents domaines, notamment dans le traitement d'images. Dans notre approche nous avons adapté cet algorithme à la classification des images à niveau de gris. Notre démarche vise à assigner à chaque pixel de l'image un degré d'appartenance flou (compris entre 0 et 1) pour chacune des classes de centre de gravité C_i . Le calcul du degré d'appartenance se base sur le Niveau de gris des différents pixels.

L'algorithme FCM adapté au traitement d'images cherche à classer au mieux les pixels de l'image aux différentes classes et cela en se basant sur la logique floue. Issu de l'algorithme Des C-moyennes (C-means), FCM introduit la notion d'ensembles flous dans la formation des classes : chaque point de l'ensemble des données appartient à chacune des classes avec un certain degré, et toutes les classes sont caractérisées par leur centre de gravité. Comme les autres algorithmes de classification non supervisée, il utilise un critère de Minimisation de la distance intraclasse et de maximisation de la distance interclasse. Cet algorithme nécessite la connaissance préalable du nombre de classes et génère les Classes par un processus itératif en minimisant une fonction objective. Ainsi, il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance, compris entre 0 et 1, à chacune des classes. Ce degré d'appartenance est calculé à partir d'une formule qui utilise un paramètre appelé l'exposant flou dit aussi l'indice flou m . La classe à laquelle est associé un pixel est celle dont le degré d'appartenance sera le plus élevé.

En peut dire que le résultat de cet étape est de décomposé l'image en classe et associe à chaque classe des pixels par la méthode FCM.

Le principe de cette dernière est :

1. La fixation arbitraire d'une matrice d'appartenance des pixels de l'image aux C classes, de telle façon que

$$\sum_{i=1}^c u_{ij} = 1, \forall j = 1, \dots, n$$

2. Le calcul des centroïdes des classes selon la formule suivante

$$C_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m}$$

3. Le réajustement de la matrice d'appartenance suivant la position des centroïdes avec

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}}$$

4. Le calcul du critère de minimisation de la fonction de dissimilarité J donnée

Par

$$J(U, c_1, c_2, \dots, c_c) = \sum_i^c J_i = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}^2$$

Les étapes de 2 à 4 sont répétées jusqu'à convergence du critère J, où u_{ij} ($u_{ij} \in [0, 1]$), le degré d'appartenance du pixel i à la classe j , c_j le centroïde de la classe i , d_{ij} la distance Euclidienne entre le $i^{\text{ème}}$ centroïde et le $j^{\text{ème}}$ pixel et m l'exposant flou.

Par conséquent nous avons deux paramètres d'action, qui ont une influence directe sur le traitement, il s'agit du nombre de classes N -classes et de l'exposant flou m . Ces deux paramètres agissent sur le résultat de la segmentation de la manière suivante : un nombre trop grand de classes donne une sur-segmentation, et un nombre trop petit donne une sous-segmentation.

De même, m est proportionnel au nombre de régions issues de la segmentation.

- **Étiquetage en régions connexes**

Le résultat de l'algorithme FCM qui est une division de l'image en différentes régions, peut donner deux régions non connexes qui sont de la même classe. Ceci peut prêter à confusion défaut. C'est pour cette raison que nous avons procédé à une étape d'étiquetage afin de différencier tous les objets existant dans l'image pouvant avoir les mêmes propriétés. A cet effet, nous avons utilisé la méthode de l'étiquetage en composantes connexes, qui fait attribuer à tous les pixels d'une composante connexe une même étiquette.

L'algorithme d'étiquetage en composantes connexes détecte les adjacences entre pixels et définit l'étiquette du pixel en question en fonction de celles des pixels voisins. La méthode la plus classique repose sur un balayage séquentiel ligne par ligne de l'image. On considère pour chaque pixel x ses voisins déjà traités : deux pixels pour la 4-connexité et quatre pixels pour la 8-

connexité. Ces pixels sont nommés les prédécesseurs de x . Dans notre cas nous avons pris la 8-connexité.

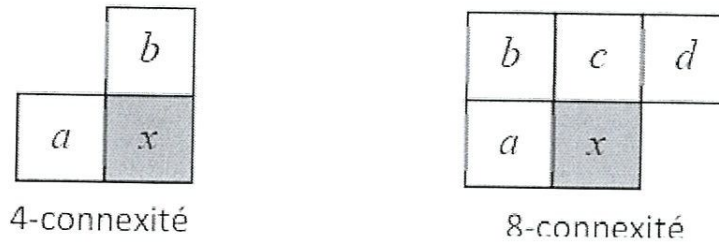


Figure 3.7 : Étiquetage en régions connexes

Pour réduire le nombre de balayage de l'image à deux, il est nécessaire de construire une table d'équivalence T qui permet de gérer les équivalences d'étiquettes qui apparaissent lors du parcours séquentiel. L'algorithme d'étiquetage en composantes connexes est donné ci-dessous,

```

Début
k:=1;
pour j:=1 à M faire
pour i := 1 à N faire
début
si(tous les prédécesseurs de A[s]=0) alors
E[s] :=k ; k :=k+1 ;
sinon
si(tous les prédécesseurs de A[s] ont la même étiquette)alors
E[s] :=  $\alpha$  ;
sinon
début
 $\alpha := \min\{T[E[sp]] \mid (sp \in \{\text{ensemble des prédécesseurs de } s\} \text{ et } (E(sp) \neq 0))\}$ 
E[s] := $\alpha$  ;
pour chaque prédécesseur d'étiquette  $\beta$  telle que  $T[\beta] \neq \alpha$  faire
début
tant que  $T[\beta] \neq \alpha$  faire
début
var := T[ $\beta$ ] ; T[ $\beta$ ]= $\alpha$  ;  $\beta :=$ var ;

```

```
fin  
fin  
fin  
fin  
fin  
fin
```

L'algorithme d'étiquetage en composantes connexes

où se représente un pixel de coordonnées (i,j) , sachant que la taille de l'image est $N \times M$, N étant le nombre de ligne et M le nombre de colonnes.

A la fin du balayage de l'image, on actualise la table T de manière à ce qu'à tout indice correspond l'étiquette définitive de l'objet, en suivant l'algorithme ci-dessous,

```
pour k :=1 jusqu'à un nombre maximum d'étiquettes faire  
début  
var :=k ;  
tant que T[var]≠k faire  
var :=T[var] ;  
T[k] :=var ;  
fin  
fin
```

L'algorithme de l'étiquette définitive

Le second balayage de l'image exploite la table d'équivalence pour affecter à chaque pixel la plus grande étiquette de l'ensemble des étiquettes équivalentes. Il est à préciser qu'à la

fin de cette étape, nous avons représenté le résultat par une image couleur en faisant correspondre à chaque étiquette une couleur de représentation.

3.3 Conclusion

Nous avons présenté dans ce chapitre dans un premier temps les outils nécessaires que notre approche exploite comme ACP. Dans un second temps, nous avons détaillé les principales étapes en suivant le séquençement de notre approche. Le chapitre 4 sera consacré à l'implémentation en mettant en relief les résultats obtenus de notre expérimentation sur différentes images.

Chapitre 4

4.1 Présentation des données

Pour les différents tests, on a utilisé des images pouvant être classées en trois catégories :

- **Images de la base VisTex**
- **Images de scènes naturelles et artificielles**

4.2 Images de la base VisTex

Cette base de référence est constituée de 168 images acquises dans des conditions non contrôlées et réparties en 19 catégories de texture couleur. Ces images sont disponibles en différentes tailles (128×128, 128×192, 256×256, 512×512, ...), ce qui correspond à différentes résolutions spatiales.

La base VisTex est une collection d'images couleur texturées qui sont représentatives de ce qu'on peut trouver dans le monde réel. Elle a été créée dans le but de fournir un large panel de textures, de haute qualité, pour les applications de traitement d'images.

Lors de nos travaux de recherche, c'est cette base d'images qui a été utilisée afin de mettre en place et de tester la méthode de classification d'images couleur texturées.

Dans la base VisTex on trouve de haut en bas :

- La classe **Bark**, représentant des écorces d'arbres,
- La classe **Fabric**, où on trouve divers textiles,
- La classe **Food**, où sont présentes différents types d'aliments,
- La classe **Metal**, représentant la texture de différents métaux,
- Et enfin la classe **Water**, où diverses images d'eau sont présentes.

On a utilisé des images de test forme de plusieurs textures extraites de cette base.

➤ Voici les images a testé, composé de deux textures:



Figure 4.1 : test 1 (a)

(b)

(c)

➤ Voici l'image a testé, composé de trois textures:

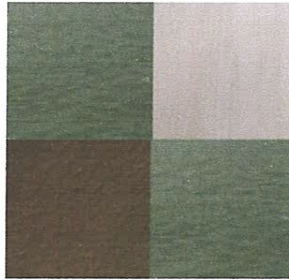
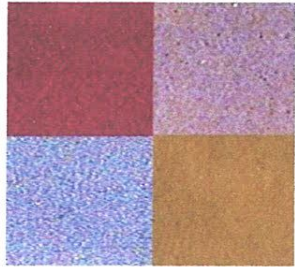
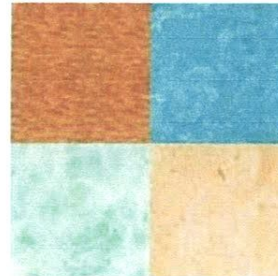


Figure 4.2 : test 2 (d)

➤ voici les images a testé, composé de quatre textures:



(e)



(f)

Figure 4.3: test 3

- **Images de scènes naturelles et artificielles**

De même en a choisi des images dans littérature pour montrer l'efficacité de l'approche mise en œuvre.

Nous avons sélectionné c'est images la :



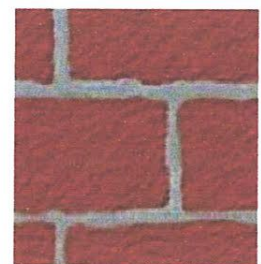
figure 4.4: test 4 (g)



(h)



(i)



(j)

4.3 Environnement matériel et logiciel de programmation

Notre application a été réalisée sur un PC portable doté d'un processeur AMD E-300 APU de fréquence 1.3 GHz et d'une RAM de 4.00 GB sous Windows 8.

L'environnement de programmation utilisé est celui de langage de programmation JAVA (jdk 7) et l'environnement Eclipse standard Kepler.

4.5 Présentation de quelques vues

L'application est composée d'une fenêtre principale, elle contient des boutons permettant d'accéder aux différentes fonctionnalités.

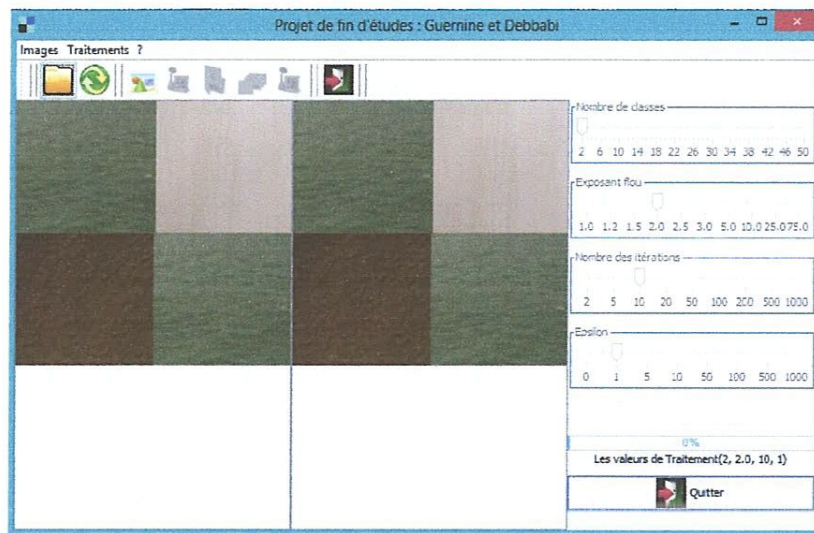


Figure 4.5 : Fenêtre principale de l'application

La barre de menu **images** permet de faire les opérations suivantes :

- Charger une image
- Initialiser l'application
- Quitter l'application

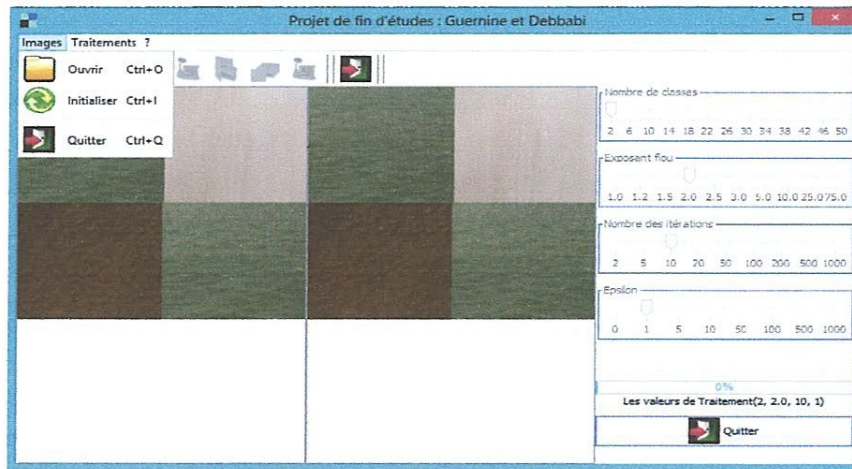


Figure 4.6 : La barre de menu images

La barre de menu **traitement** prend en charge les fonctions suivantes :

- Application de l'ACP sur l'image chargée
- Filtrage de l'image résultat de l'ACP par un banc de filtre de Gabor
- Extraction des régions texturées
- classification par méthodes de fuzzy c-means
- Étiquetage en régions connexes

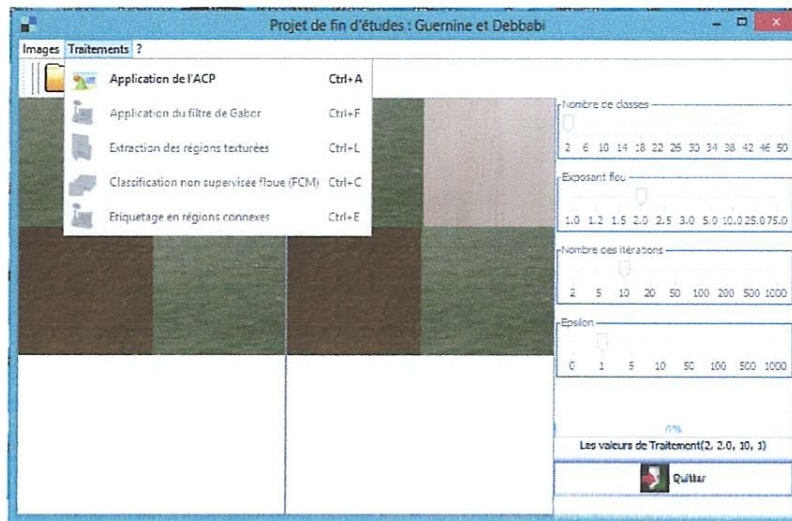


Figure 4.7 : La barre de menu traitement

Quand on a exécuté l'application de l'ACP, le logiciel a produit le résultat de cette étape dans une nouvelle fenêtre

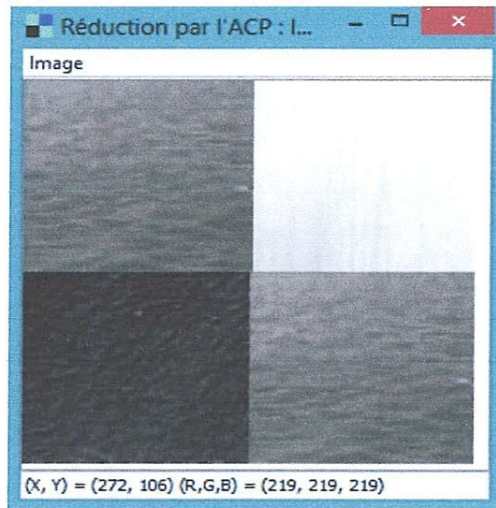


Figure 4.8: Application de l'ACP

Ensuite on filtre l'image résultat de l'ACP par un banc de filtre de Gabor mais il faut avant ajuster les paramètres du filtre.

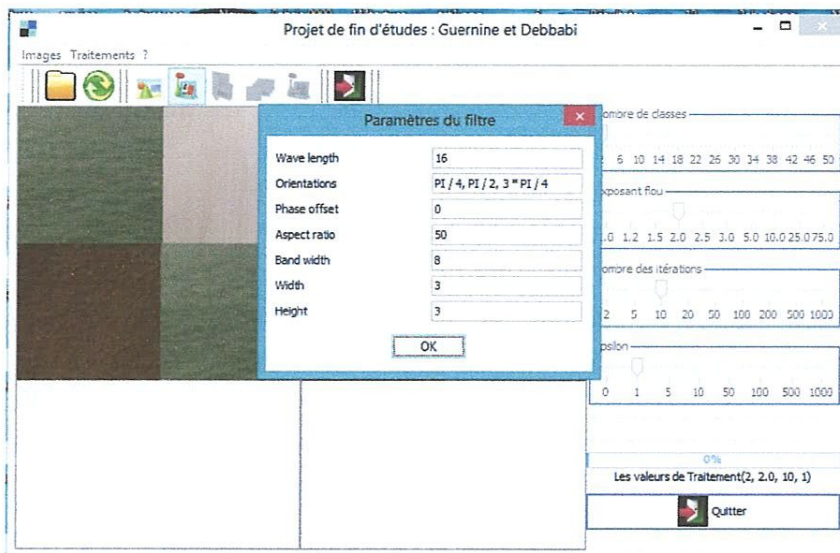


Figure 4.9 : Paramètres de filtre de Gabor

Est voici le résultat obtenu par cette étape

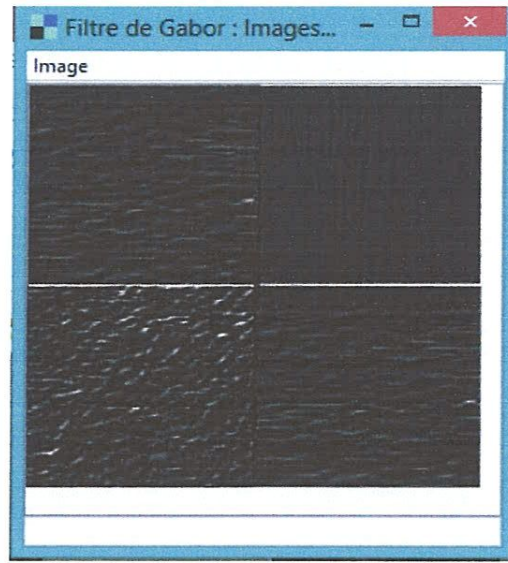


Figure 4.10 : application de Gabor

L'étape suivante du traitement est celle de l'extraction régions texturées. Un ajustement préalable des paramètres du filtre de Gabor est requis.

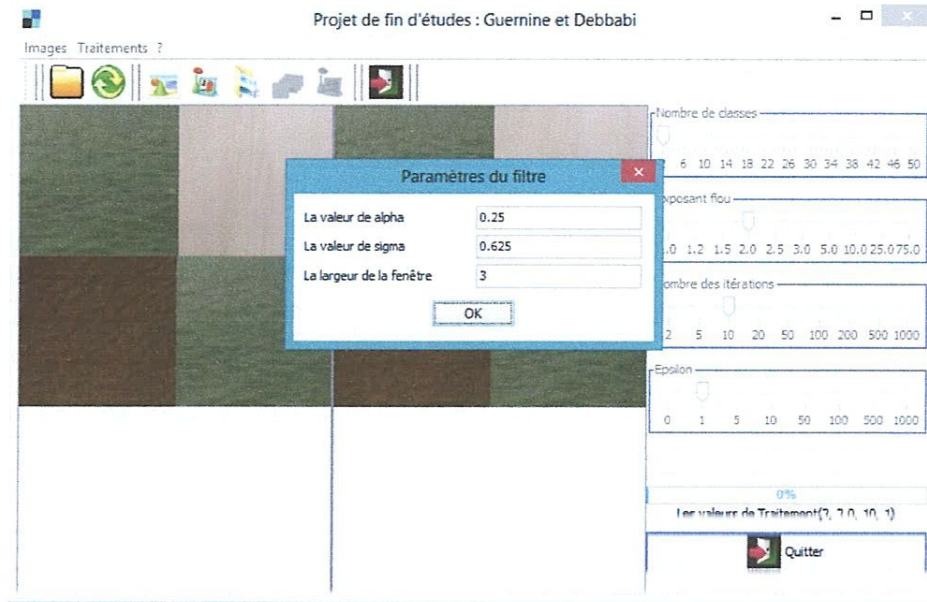


Figure 111 : extraction des régions texturées

Voici le résultat obtenu au niveau de cette étape (Figure 4.12).

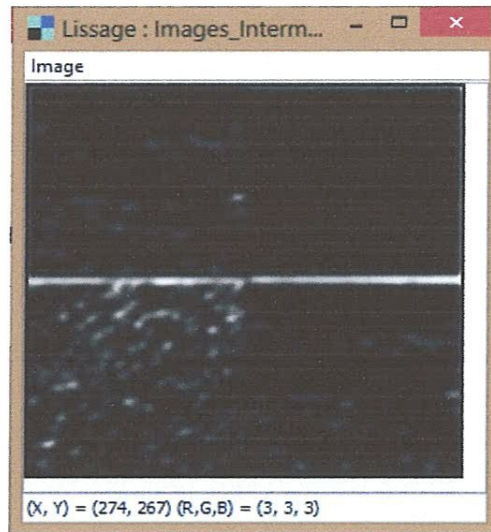


Figure 4.12 : Extraction des régions texturées

L'étape suivante de notre processus est celle de la classification FCM. On agit sur les quatre paramètres pour obtenir les valeurs optimales (Figure 4.13) :

- Le nombre de classes
- L'exposant flou m
- Nombre d'itérations
- Epsilon

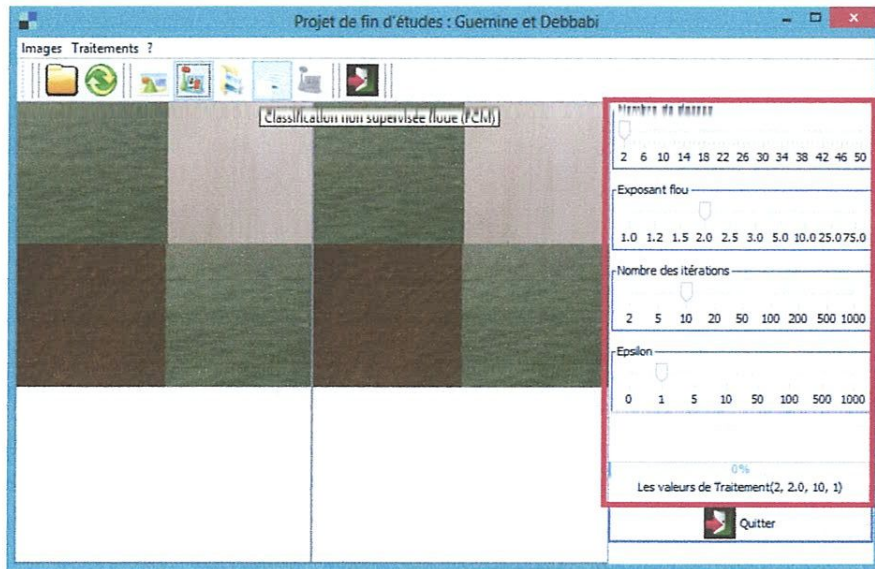


Figure 4.13 : Paramètres de FCM

Le résultat obtenu de l'exécution du programme FCM est indiqué en Figure 4.14.

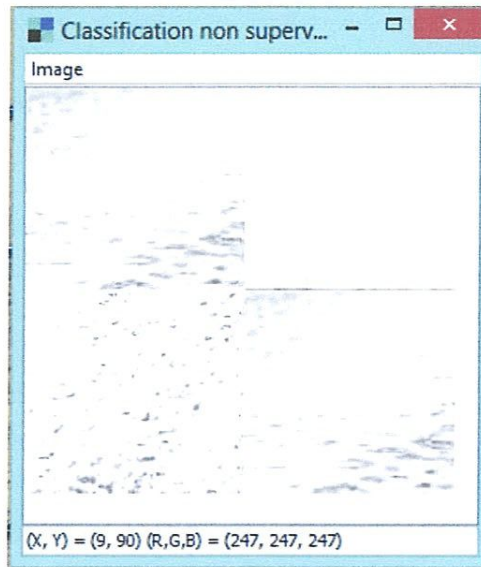


Figure 4.14 : Résultat de FCM

L'excusions de FCM produit aussi un tableau montrant l'appartenance de chaque pixel à sa classe, (voir Figure 4.15).

Résultat de la classification FCM					
i	J	C1	C2	C3	Classe
0	0	0.0040380973	0.9900517	0.0059103044	C1
0	1	0.009333654	0.9787385	0.011927937	C1
0	2	0.014281567	0.9690401	0.016678374	C1
0	3	0.011658378	0.9740995	0.014242061	C1
0	4	0.020468451	0.95768434	0.021847174	C1
0	5	0.03691907	0.93016	0.032920953	C1
0	6	0.047359917	0.9140449	0.03859512	C1
0	7	0.03691907	0.93016	0.032920953	C1
0	8	0.06602844	0.8869799	0.046991643	C1
0	9	0.032268908	0.93763334	0.030097788	C1
0	10	0.017214496	0.9635664	0.01921907	C1
0	11	0.024054691	0.9513988	0.024546495	C1
0	12	0.041945852	0.92229646	0.035757676	C1
0	13	0.053171698	0.9054073	0.041421015	C1
0	14	0.0593913	0.89638513	0.044223388	C1
0	15	0.0805918	0.86702657	0.05238161	C1
0	16	0.088534884	0.85648173	0.054983452	C1
0	17	0.07309238	0.8771933	0.049714405	C1
0	18	0.047359917	0.9140449	0.03859512	C1
0	19	0.027984465	0.9447139	0.027301619	C1
0	20	0.020468451	0.95768434	0.021847174	C1
0	21	0.014281567	0.9690401	0.016678374	C1
0	22	0.014281567	0.9690401	0.016678374	C1
0	23	0.017214496	0.9635664	0.01921907	C1
0	24	0.0027953081	0.99292326	0.0042819485	C1
0	25	0.0055346848	0.98672336	0.007741955	C1
0	26	0.011658378	0.9740995	0.014242061	C1
0	27	0.011658378	0.9740995	0.014242061	C1

Figure 4.15 : Appartenance de chaque pixel

A ce stade en arrive à la dernière étape de notre approche : étiquetage des régions connexes, la figure 4.12 illustre cette étape.

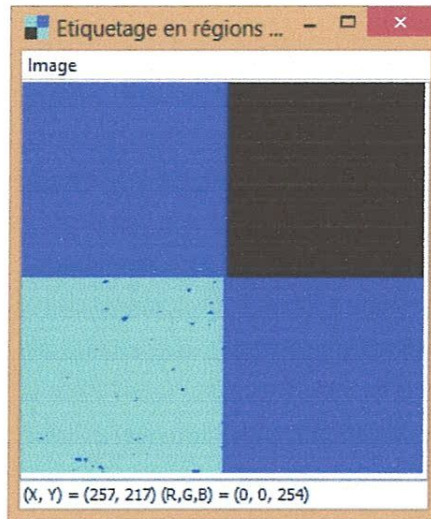


Figure 4.16 : Étiquetage en régions connexes

Nous avons inséré à notre outil une Aide (?) pour faciliter son exploitation. La figure 4.13 montre cette aide, composée de deux sous options :

- Aide sur le logiciel
- A propos donne des informations sur le projet

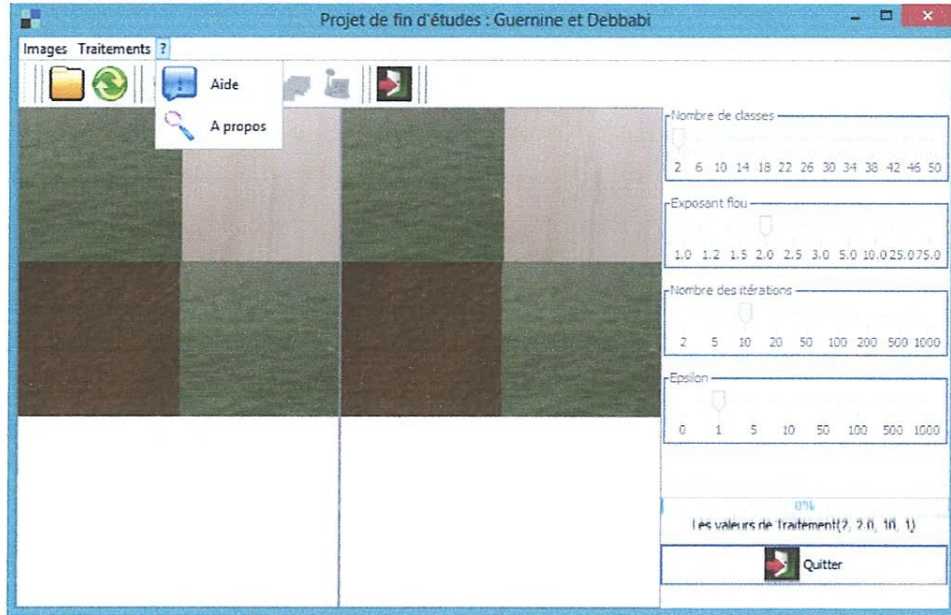


Figure 4.17 : Barre de menu Aide (?)

Le Bouton « A propos » donne des informations sur le projet (Figure 4.14).

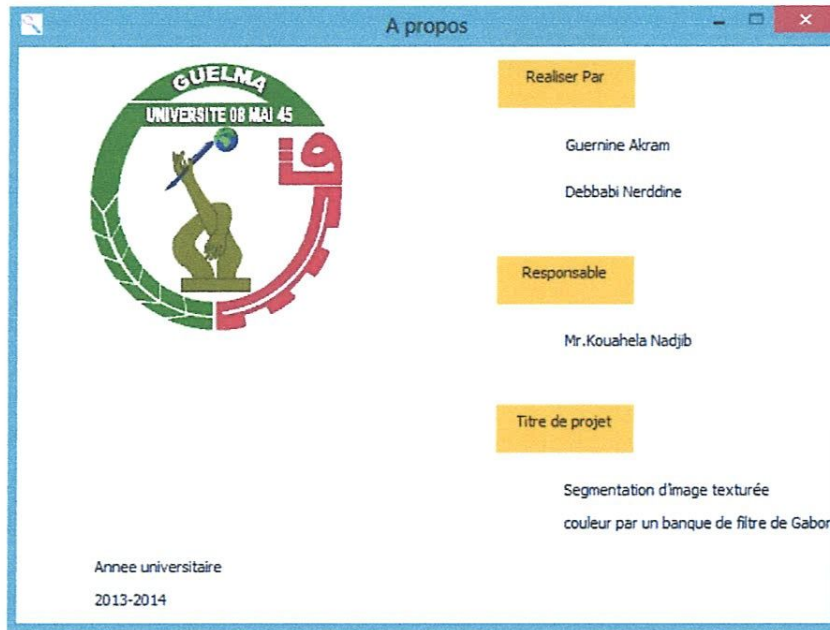


Figure 4.18 : Information sur le logiciel

4.6 Résultats et discussions

Dans notre travail, on a utilisé des images de taille 256*256 pixels codes en RVB.

Afin de montrer l'efficacité de la méthode de classification floue mise en œuvre, on va examiner les résultats obtenus sur quelques images avec un nombre de textures différent : 2, 3 et 4.

Les résultats sont représentés comme suit :

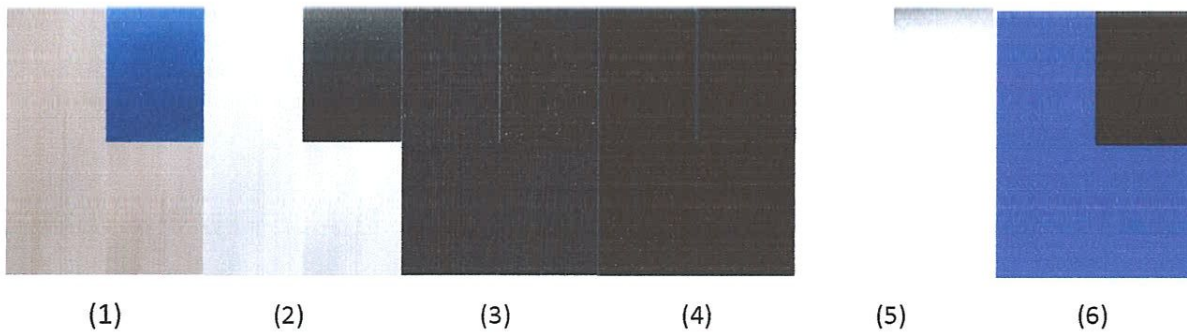


Figure 4.19 : image original(1), résultat de ACP(2), résultat de Gabor (3), résultat de l'extraction(4), résultat de FCM(5), résultat finale(6).

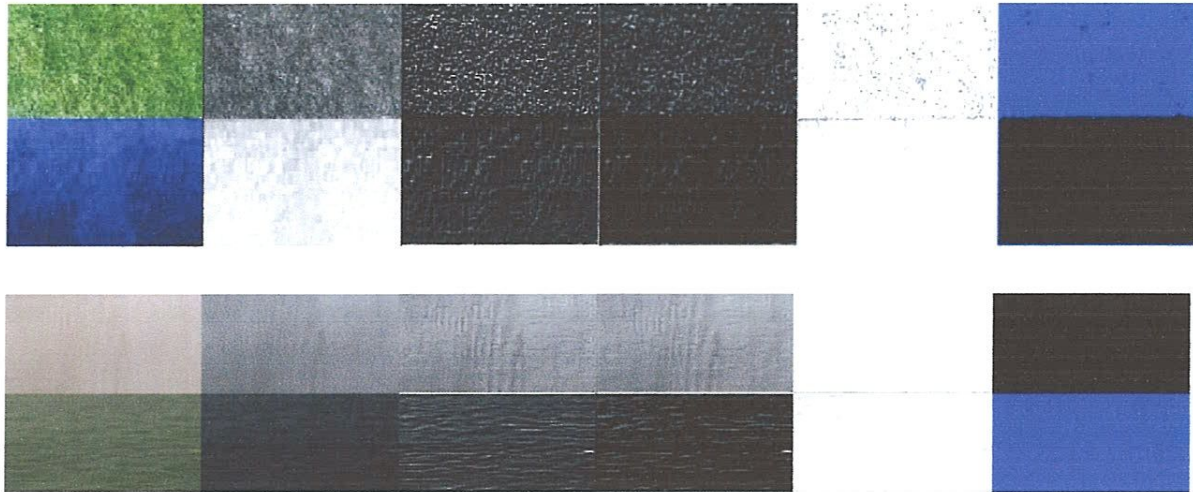


Figure 4.20 : résultat de test 1

Image avec trois textures :

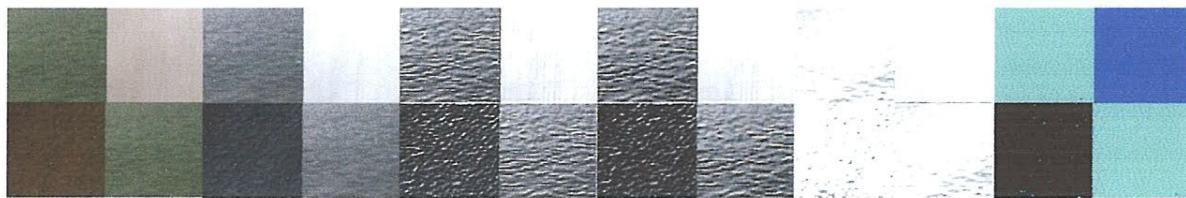


Figure 4.21 : résultat de test 2

Image avec quatre textures :

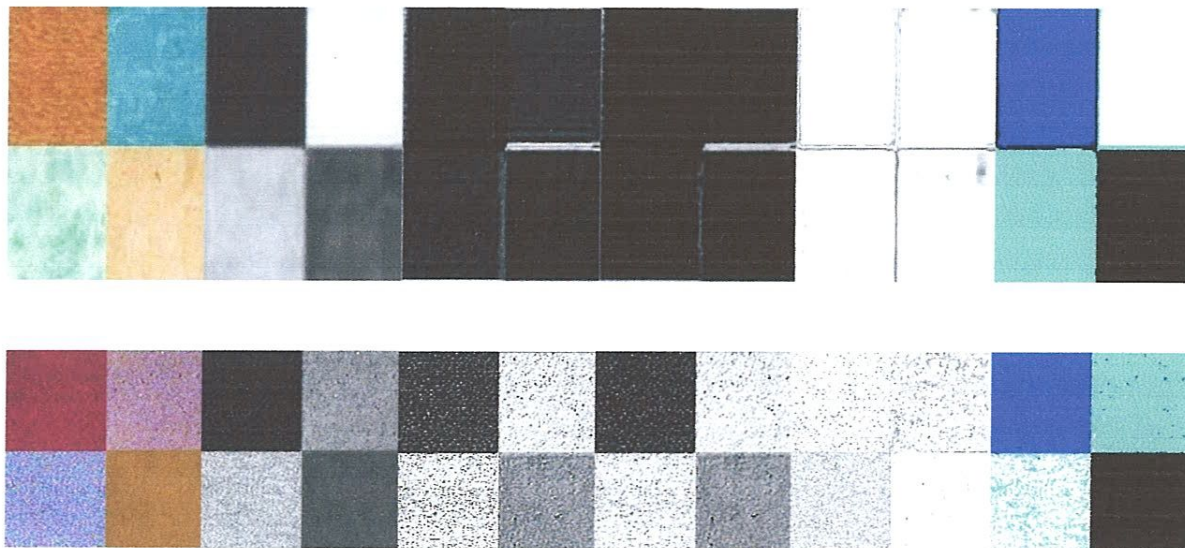


Figure 4.22 : résultat de test 3

Les résultats des Images de scènes naturelles et artificielles



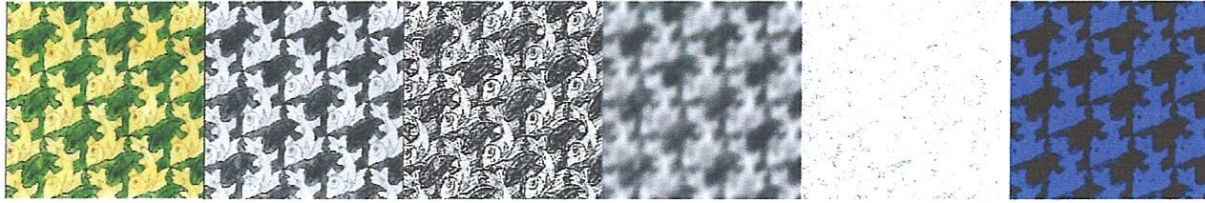


Figure 4.23 : résultat de test 4

Notre outil permet de retrouver les régions texturés fidèlement pour chacune des images expérimenté.

Notre approche laisse apparaitre qu'elle est valide lorsque nous considérons des textures toutes différentes dans l'image.

De même, nous avons constaté qu'elle reste valide pour un nombre de textures qui ne sont pas toutes différentes c'est à dire que certaines sont identiques

En plus nous avons confronté sur une même image notre approche utilisant FCM est le filtre de Gabor est une autre approche utilisant FCM seulement, le résultat est montré sur les figures suivante :

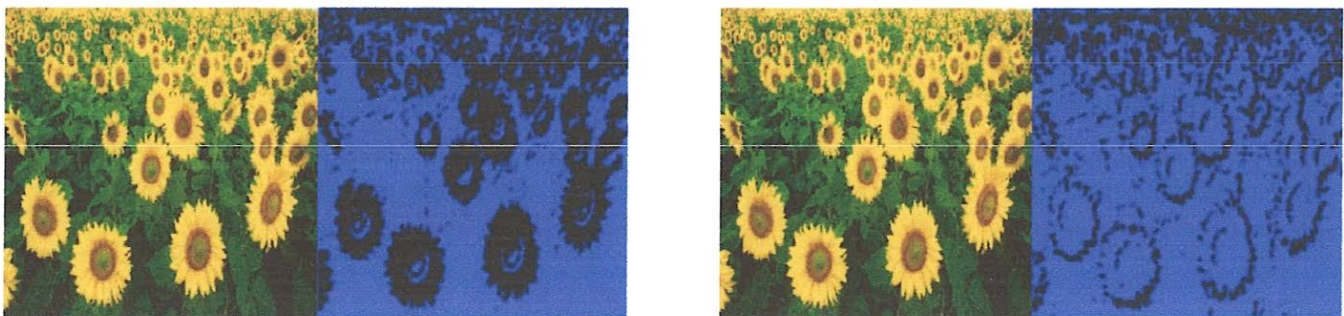


Figure 4.24 : comparaison entre notre approche et FCM

L'approche sans le filtre de Gabor présente des résultats non satisfaisants
La nôtre a produit les résultats attendus. Ainsi nous pouvons dire que notre méthode présente des
bons résultats dans la plupart des cas.

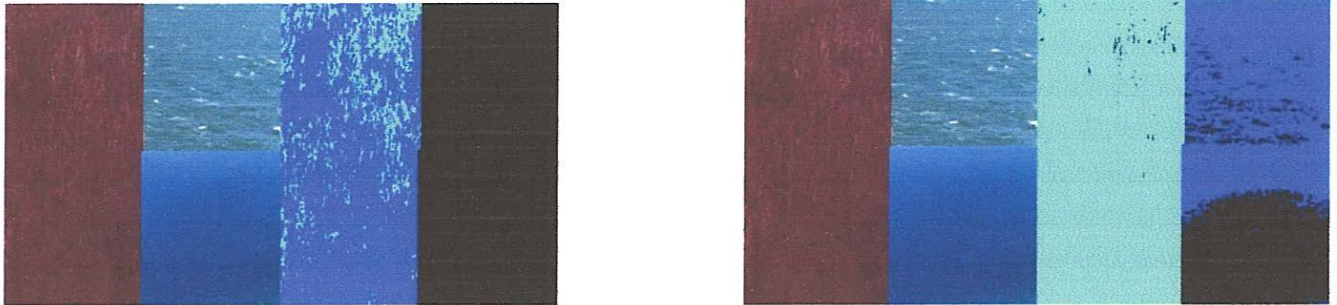


Figure 4.25 : l'influence des paramètres de filtre de Gabor

Néanmoins, nous remarquons qu'elle présente une difficulté dans le réglage des
paramètres optimaux au niveau de filtre de Gabor.

Conclusion générale et perspectives

Dans ce mémoire, nous avons brassé toutes les principales méthodes de segmentation rencontrées dans la littérature, connexes aux images texturées couleur. Nous avons également fait un état de l'art sur les méthodes de segmentation basées sur les régions. Malgré le nombre et la diversité des méthodes de segmentation existantes, la problématique de segmentation reste un problème non résolu et loin de l'être, pour cela la recherche se focalise sur cette problématique en optant davantage sur de nouvelles méthodes telle que par exemple l'utilisation des nombres hypercomplexes dans les filtres de Gabor et l'intégration d'algorithmes aux filtres de Gabor, etc. On a constaté que le processus de segmentation d'images texturées couleur a deux grandes parties essentielles : la partie d'analyse de texture couleur et la partie segmentation proprement dite. On ne peut pas faire de la segmentation d'images texturées que ce soit couleurs ou non, sans passer par l'analyse de texture car c'est au niveau de cette phase que se fait l'extraction d'attributs texturaux.

Dans ce sillage, nous avons réalisé un outil en java, applicable à toute image en couleur qui permet de classifier les pixels de manière floue, de les catégoriser en régions et de leur attribuer une étiquette chacune. La méthode de segmentation d'image texturées couleur par banque de filtres de Gabor qu'on a adapté aux images de texture couleur, a donné des résultats très satisfaisants. De plus cette méthode est performante et simple à mettre en œuvre.

Les difficultés essentielles résident d'une part dans le lissage qui nous fait apparaître dans certains cas des résultats de sur-segmentation et d'autre part dans le réglage des paramètres de Gabor en particulier dans le choix des fréquences et des orientations.

Les résultats expérimentaux portant sur les différents types d'images, ont montré l'efficacité et la flexibilité de la méthode développée.

Nous espérons à l'avenir enrichir notre outil en ajoutant de nouvelles fonctionnalités comme à titre non exhaustif : la reconnaissance des régions, image dynamique, hybridation de méthodes ou complètement changer de stratégie raisonner directement sur la couleur.

Bibliographie

- [1] Hadjer LAGUEL,< Déploiement sur une plateforme de visualisation, d'un algorithme coopératif pour la segmentation d'images IRM basé sur les systèmes multi-agents>,USTHB,2010.
- [2] Rabia Nacim,<revue des méthodes de segmentation d'image texturées : cas des images couleur>,université Mouloud Mammer,2011.
- [3] Bitam Abdelmadjid,<Analyse et segmentation d'images multispectrales :application aux images MSG>,université Mouloud Mammer,2013.
- [4] Ouarda Assas,<classification floue des images>,universite de batna,2013.
- [5] D'Hondt Frédéric, El Khayati Brahim<Etude de méthodes de Clustering pour la segmentation d'images en couleurs>, Faculté Polytechnique de Mons, 5^{ème} Electricité, Certificat Applicatifs Multimédia.
- [6] S.Ameur, Z.Ameur,< REVUE DES APPROCHES DE SEGMENTATION D'IMAGES TEXTUREES : EXEMPLE DES IMAGES METEOROLOGIQUE>, 3rd International Conference: Sciences of Electronic,Technologies of Information and Telecommunications, TUNISIA, 2005
- [7] Hocini lotfi,<conception de metaheuristique d'optimisation pour la segmentation des images de teledetection>, université Mouloud Mammer,2012.
- [8] HOUASSINE Charif< Segmentation d'images par une approche biomimétique hybride>, UNIVERSITE M'HAMED BOUGARA- BOUMERDES,2012.
- [9] N. Mhamda,N. Nacereddine< Logique Floue et Traitement des Images de Radiographie Industrielles >, Laboratoire Signal et Communications Ecole Nationale Polytechnique Alger, Algérie.

[10] Oumar DIOP,< Détection de nuages de poussière dans les images Météosat à l'aide des attributs de textures et de la fusion de segmentations : application à la zones ahélienne du continent africain >, L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES ,2008

[11] Lehamel Malha,<segmentation d'images texturées à partir des attributs fractals>,Université Mouloud Mammeri.

[12] Pierre Martin,<segmentation d'images de texture par modèles multirésolutions>,Université laval,2000

[13] Arab Dalila, Ameer Soltane, Ameer Zohra< Segmentation d'images textures par une approche statistique >, 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications,TUNISIA,2007.

[14] hamza chkkaf,Ibrahim touil<segmentation des masses dans les images ammographiques>, Université Abou bakr belkaid.