

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMARI DE TIZI OUZOU
FACULTÉ DE GÉNIE ELECTRIQUE ET D'INFORMATIQUE
DÉPARTEMENT D'ELECTRONIQUE



Mémoire de fin d'études

En vue de l'obtention du diplôme d'Ingénieur d'Etat en
Electronique
Option : Communication & Contrôle

Thème

**Application de la logique floue à
la segmentation d'images couleur**

Proposé par :

Madame S. ALKAMA

Réalisé par:

M^r REZKI Farid
M^r SMAIL Mouloud
M^r HAHAD Idir

Promotion: 2007/2008

Remerciements

Au terme de ce mémoire, nous remercions Dieu le tout puissant de nous avoir donné le courage et la patience pour mener à bout ce modeste travail.

Egalement, nous tenons à exprimer nos vifs remerciements à notre promotrice M^{me} Alkama S qui nous a fait l'honneur de nous prendre sous sa responsabilité, ainsi que pour ces orientations et précieux conseils, qu'il trouve ici toutes les expressions de notre respect et gratitude.

Nos remerciements s'adressent également aux enseignants qui ont contribué à notre formation, ainsi que les membres du jury qui nous feront l'honneur de juger notre travail.

Nous tenons enfin à remercier tous qui nous ont portés d'aide de près ou de loin à l'élaboration de ce travail.



Je dédie ce modeste travail

A la mémoire de mon grand père

A mes plus chers parents qui m'ont toujours soutenu

A ma grande mère, à qui je souhaite une longue vie

A mes sœurs Chabha, Nadjia, Djouher

A mes frères Mourad, Samir, Karim

A tous mes amis(es)

A toute la promotion ELN 2008

Farid



Dédicaces

Je dédie ce modeste travail

A mes parents qui me sont très chers.

A mes frères.

A mes sœurs.

A toute la famille.

A mes amis(es).

Edir



Je dédie ce modeste travail

A mes parents qui me sont très chers.

A mes frères.

A mes sœurs.

A toute la famille.

A mes amis(es).

Mouloud

Introduction générale..... 1

Chapitre I Généralités sur le traitement d'images

I.1 Introduction 3
I.2 Notions d'images 3
 I.2.1 Image binaire 3
 I.2.2 Image en niveaux de gris 4
 I.2.3 Image couleur 4
I.3 Caractéristiques d'une image numérique 5
 I.3.1 Pixel 5
 I.3.2 Dynamique de l'image 6
 I.3.3 Dimension 6
 I.3.4 Résolution 6
 I.3.5 Bruit 6
 I.3.6 Contour 6
 I.3.7 Région 6
 I.3.8 Luminance 6
 I.3.9 Contraste 7
 I.3.10 Notion de voisinage 7
 I.3.11 Distance entre deux couleurs 7
I.4 Histogramme d'une image 8
I.5 Architecture d'un système de traitement d'image 9
 I.5.1 Acquisition 9
 I.5.2 Traitement 10
 I.5.3 Analyse 10
 I.5.4 Visualisation 10
 I.5.5 Stockage 10
I.6 Segmentation d'images couleur 10
 I.6.1 Définition mathématique de la segmentation 11
 I.6.2 L'approche contour 11
 I.6.2.1 Approches marginales 12
 I.6.2.2 Approche multi composante ou Vectorielle 13
 I.6.3 Méthodes de segmentation par analyse des propriétés spatiales 13
 I.6.3.1 Segmentation par division de régions 13
 I.6.3.2 Segmentation par croissance de régions 14
 I.6.3.3 Segmentation par fusion de régions 15
 I.6.3.4 Segmentation par division-fusion 15
 I.6.4 Méthodes de segmentation par analyse des composantes couleurs des
 pixels 17
Conclusion 19

Chapitre II Généralités sur la théorie des ensembles flous

II.1 Introduction 20
II.2 Notions fondamentales sur les ensembles flous 20

II.2.1 Définition d'un ensemble flou.....	20
II.2.2 Caractéristique d'un ensemble flou.....	21
II.2.2.1 Support.....	21
II.2.2.2 Noyau.....	21
II.2.2.3 Hauteur.....	22
II.2.2.4 Cardinalité.....	22
II.2.2.5 Sous ensemble convexe.....	22
II.3 Exemple de fonctions d'appartenance.....	23
II.3.1 Fonction d'appartenance trapézoïdale.....	23
II.3.2 Fonction d'appartenance exponentielle.....	24
II.4 Opérations sur les sous ensembles flous.....	24
II.4.1 Inclusion.....	25
II.4.2 Egalité.....	25
II.4.3 Réunion.....	25
II.4.4 Intersection.....	26
II.4.5 Complément d'un sous ensemble flou.....	26
II.5 Définition d'une α coupe et d'une α coupe stricte.....	26
II.6 Indice de comparaison.....	27
II.6.1 Indice de flou.....	27
II.6.2 Indices de chevauchement.....	27
II.6.3 Indices d'inclusion.....	27
II.6.4 Indices d'égalité.....	28
II.7 Modificateurs.....	28
II.7.1 Normalisation.....	28
II.7.2 Concentration.....	28
II.7.3 Dilatation.....	29
II.8 Produit cartésiens de sous ensembles flous.....	29
II.9 Projection.....	29
II.10 Principe d'extension.....	29
II.11 Application de la logique floue.....	30
II.11.1 Fuzzification.....	31
II.11.2 Traitement flou.....	31
II.11.3 Déffuzzification.....	31
II.12 Domaines d'application de la logique floue.....	31
Conclusion.....	31

Chapitre III Application de la logique floue à la segmentation d'images

III.1 Introduction.....	33
III.2 Segmentation d'images par classification flou Fuzzy c-means (Fcm).....	33
III.3 Variante de l'algorithme Fuzzy c-means adapté à la segmentation d'images.....	37
III.4 Déffuzzification.....	39
III.5 Décision par focalisation graduelle.....	40
III.5.1 Fonction de décision par coupe de niveau α	40
III.5.2 Fonction de décision par maximum d'appartenance.....	41

III.5.3 Fonction de décision par focalisation graduelle	42
III.6 Organigramme des différentes étapes de la défuzification.....	43
Conclusion	44

Chapitre IV Tests et Résultats

IV.1 Introduction	45
IV.2 Les images tests	45
IV.3 Comparaison entre la méthode de segmentation scalaire et la méthode de segmentation marginale	46
IV.4 Influence de la valeur du nombre de classes c sur la segmentation d'images couleurs	50
IV.5 Influence de la valeur du seuil de convergence ε	52
IV.6 Influence de la valeur du facteur flou m	54
IV.7 Analyse de la variation du Nombre d'itérations I_t et la fonction coût J_m	55
IV.8 Etude du temps de calcul nécessaire.....	58
Conclusion	60

Conclusion générale.....	61
---------------------------------	-----------

Bibliographie.....	62
---------------------------	-----------

Introduction générale

Le traitement d'images voit ses débuts dans les années 1920 dans la transmission de données par câble mais ne connaît de vrai essor que dans les années 1960 avec le développement des ordinateurs. Au départ, les techniques de traitement d'images sont essentiellement des méthodes de restauration et de compression d'images. Puis se développent avec les progrès de l'informatique.

Aujourd'hui il ne s'agit plus uniquement de traiter les images pour les améliorer mais aussi de les comprendre et de les interpréter. C'est dans ce contexte que la reconnaissance d'objets dans les images devient un sujet de recherche important. Pour reconnaître des objets afin d'interpréter les images, il faut souvent au préalable les segmenter, c'est-à-dire séparer les objets d'intérêt du fond de l'image.

Face à l'inadéquation de la logique classique à résoudre de nombreux cas de problèmes comme le traitement des connaissances imprécises et incertaines, une nouvelle logique a ainsi été proposée comme méthode rationnelle pour traiter ce type de problèmes, dite logique floue. Elle est considérée comme un outil mathématique rigoureux permettant de traiter d'une manière adéquate toutes les difficultés dont la logique binaire n'est pas venue à bout.

Dans ce mémoire, nous nous intéressons au traitement des images couleur. caractérisées par leurs composantes trichromatiques. Il s'agit plus précisément de la segmentation de ces images, traitement de bas niveau qui consiste à créer une partition de l'image en sous-ensembles appelés régions.

Pour illustrer notre travail, nous l'organisons en quatre chapitres.

Dans le **premier chapitre**, nous donnons quelques généralités sur l'imagerie, ainsi que certaines méthodes qui permettent de segmenter les images couleur.

Dans le **deuxième chapitre** nous présenterons les principes de base de la théorie des ensembles flous.

Le **troisième chapitre**, sera consacré aux méthodes utilisées dans notre travail, le **dernier chapitre**, portera sur les tests effectués et les résultats obtenus.

Nous terminons notre travail par une conclusion générale.

I.1 Introduction

L'image est considérée parmi les moyens les plus importants qu'utilise l'Homme pour communiquer avec autrui. C'est un outil de communication universel qui permet aux êtres humains de différents âges et de différentes cultures de se comprendre.

Le traitement d'image est l'ensemble des méthodes et techniques opérant sur celle-ci, en corrigeant les dégradations subies lors de son acquisition ou en extrayant les informations permettant une interprétation visuelle ou automatique, pour mieux exploiter les informations qu'elle contient.

I.2 Notions d'images

L'image est une Projection à 2 dimensions d'une scène à 3 dimensions. Elle est définie comme une matrice où la valeur de chaque élément (pixel) représente l'intensité discrète de la lumière en ce point. Chaque cellule élémentaire de l'image (ou pixel) a comme caractéristique un niveau de gris ou de couleur prélevé à l'emplacement correspondant dans l'image réelle [4].

Une image peut être aussi présentée sous forme d'un signal bidimensionnel auquel on peut appliquer tous les éléments du traitement du signal comme, à titre d'exemple, la transformée de Fourier. Il existe 3 types d'images : binaire, en niveau de gris et couleur.

I.2.1 Image binaire [3]

Une image binaire est une matrice rectangulaire dont le nombre de niveaux de gris est réduit aux deux éléments 0 et 1, où le niveau 0 représente le noir absolu et le niveau 1 représente le blanc. On n'utilise dans ce cas qu'un seul bit pour coder le niveau de gris.

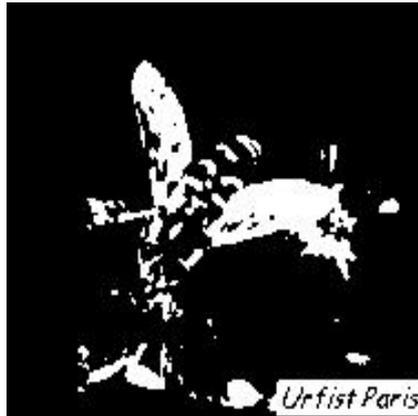


Figure I.1 Exemple d'une image binaire.

I.2.2 Image en niveau de gris [3]

Le niveau de gris est la valeur de l'intensité lumineuse en un point. Le pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter des images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière envoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet.



Figure I.2 Exemple d'une image en niveaux de gris.

I.2.3 Image couleur [3]

Ces images sont en général codées en utilisant le codage des trois couleurs fondamentales (rouge, vert, bleu), on parle alors d'images RVB. Une couleur contient

donc trois plans couleurs, rouge, vert et bleu (RVB). Chaque plan est codé comme une image en niveau de gris avec des valeurs allant de 0 à 225 (voir figure I.3). Lorsque $R=V=B$, la valeur associée est un niveau de gris. D'autre part, pour aller d'une image couleur à une image en niveau de gris on réalise la fonction suivante :

$$I(i, j) = \frac{R(i, j) + V(i, j) + B(i, j)}{3}$$

Avec :

$I(i, j)$: niveau de gris du pixel situé a la ligne i et la colonne j ,

$R(i, j)$: intensité de la couleur rouge du pixel (i, j) ,

$V(i, j)$: intensité de la couleur verte du pixel (i, j) ,

$B(i, j)$: intensité de la couleur bleu du pixel (i, j) .



Figure I.3 Exemple d'une image en couleur

I.3 Caractéristiques d'une image numérique [3]

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants :

I.3.1 Pixel

Contraction de l'expression anglaise " Picture éléments " qui signifie élément d'image, le pixel est le plus petit point de l'image. Dans une image couleur (RVB), un pixel peut être représenté sur trois octets, un octet pour chacune des couleurs, rouge (R), verte (V) et bleue (B).

I.3.2 Dynamique de l'image

C'est le nombre de niveaux de gris dans une image (c'est le nombre de niveaux de quantification).

I.3.3 Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses. Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

I.3.4 Résolution

La résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution.

I.3.5 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins.

I.3.6 Contour

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.

I.3.7 Région

C'est un ensemble connexe de pixels ayant une ou plusieurs propriétés communes, on parle de zones homogènes de l'image.

I.3.8 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface. Pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

I.3.9 Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2}$$

I.3.10 Notion de voisinage

Chaque pixel $P(x, y)$ a quatre voisins horizontaux-verticaux et quatre voisins diagonaux

$P(x-1,y-1)$	$P(x,y-1)$	$P(x+1,y-1)$
$P(x-1,y)$	$P(x,y)$	$P(x+1,y)$
$P(x-1,y+1)$	$P(x,y+1)$	$P(x+1,y+1)$

Figure I.4 Les huit voisins de $P(x, y)$

	$P(x,y-1)$	
$P(x-1,y)$	$P(x,y)$	$P(x+1,y)$
	$P(x,y+1)$	

Figure I.5 Les quatre voisins de $P(x, y)$

I.3.11 Distance entre deux couleurs

Pour les images en niveau de gris, la distance entre deux couleurs est la différence des niveaux de gris. Par contre, pour les images couleurs, cette distance peut être calculée différemment. Elle se traduit généralement par une distance euclidienne dans l'espace colorimétrique utilisée.

La distance euclidienne dans l'espace RVB est donnée par la formule suivante :

$$\Delta d = \sqrt{\Delta R^2 + \Delta V^2 + \Delta B^2} .$$

I.4 Histogramme d'une image [6]

L'histogramme d'une image représente le nombre (ou la proportion) de pixel en fonction de niveau de gris dans l'image, c'est à dire la distribution des valeurs de niveaux de gris dans une image.

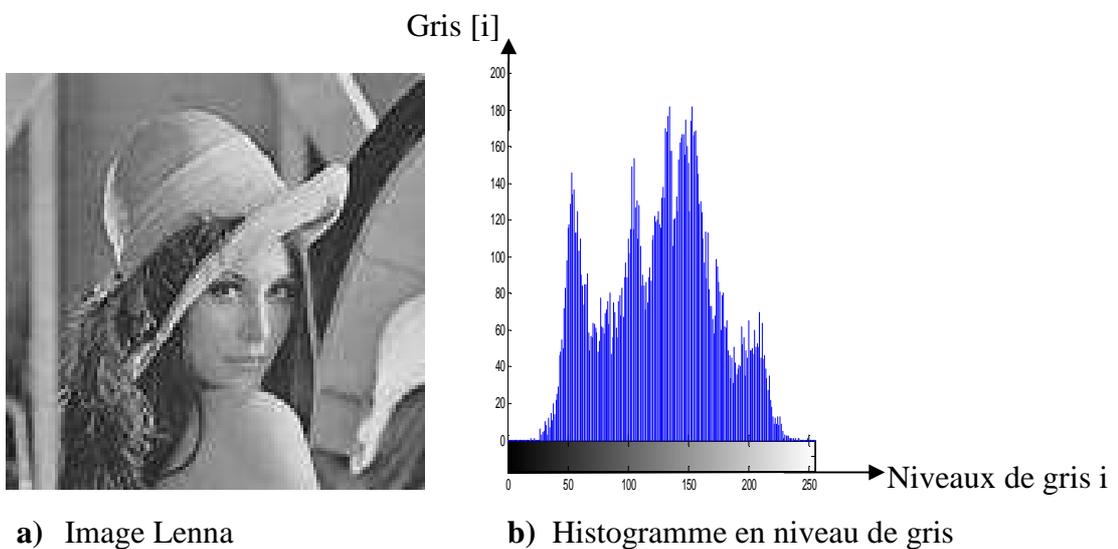


Figure I.6 Image Lenna en niveau de gris et son histogramme.

Pour les images en couleur plusieurs histogrammes sont nécessaires. Par exemple pour une image codée en RVB, trois histogrammes représentant respectivement la distribution des valeurs respectives des composantes rouges, bleues et vertes sont nécessaires.

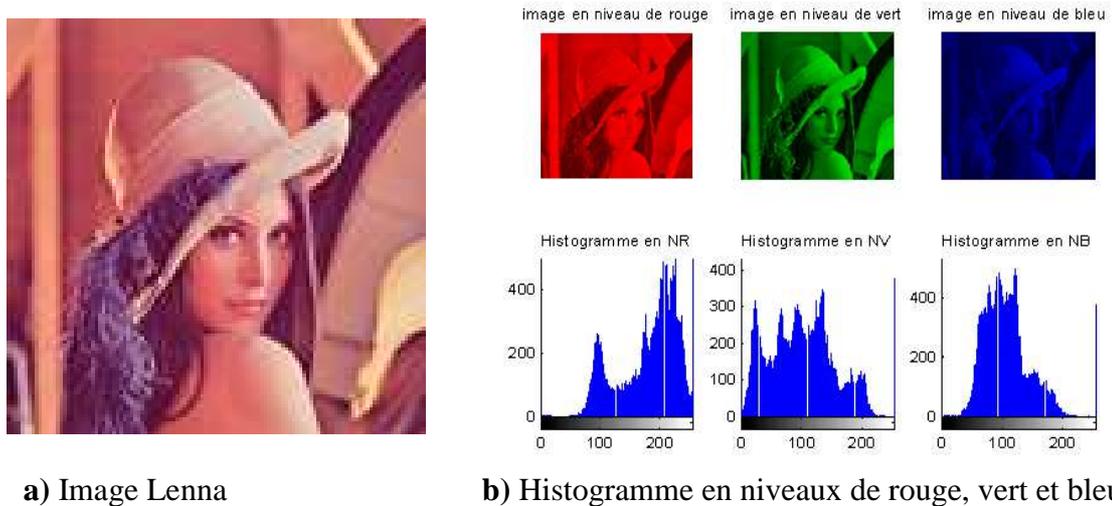


Figure I.7 Image Lenna en couleur et les histogrammes R, V et B

I.5 Architecture d'un système de traitement d'image [4]

Un système de traitement d'image numérique est composé de plusieurs parties à savoir acquisition, traitement, analyse, stockage et visualisation (voir figure I.8).

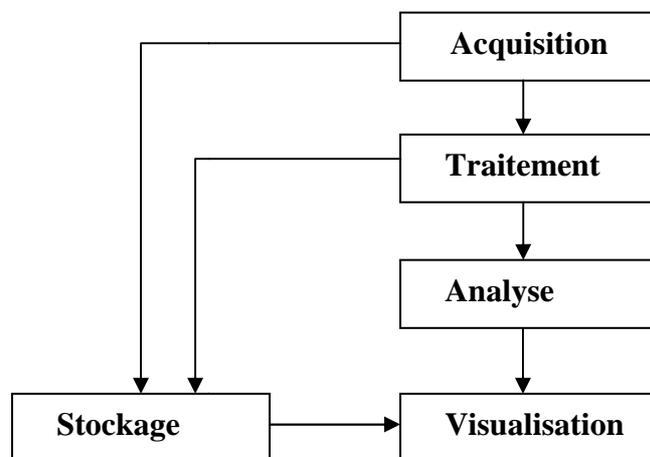


Figure I.8 Chaîne de traitement et d'analyse d'images

I.5.1 Acquisition

Cette phase est réalisée à l'aide de deux modules, le capteur et le numériseur.

Le capteur permet de capter des informations du monde extérieur et les transforme en signaux électriques capable d'être traité par un système électronique.

Le Numériseur transforme le signal analogique fourni par le capteur en une matrice de points appelés pixels afin d'être traité par des systèmes informatiques.

I.5.2 Traitement

Les traitements que peuvent subir les images sont principalement des opérations de convolution et de segmentation.

La convolution est le remplacement de la valeur d'un pixel par une combinaison linéaire de ses voisins. Elle consiste à faire balayer une fenêtre (masque) sur l'ensemble des points de l'image.

La segmentation est un traitement de bas niveau qui consiste à diviser l'ensemble des pixels en régions connexes, homogènes et différentes de ses voisines qui pourront par la suite être traitées de manières différentes.

I.5.3 Analyse

L'objectif de l'analyse est de permettre l'exploitation du contenu de l'image à des fins d'interprétation (tel l'aide au diagnostic en imagerie aérienne, satellitaire ou médicale), de localisation et de reconnaissance (par exemple en vidéosurveillance ou en control robotique) ou de mesure des évolutions (en contrôle qualité).

I.5.4 Visualisation

Tout système de traitement d'image est doté d'un dispositif de visualisation qui permet l'affichage des images. L'utilisation de différents types de reconstituteurs permet de transformer le signal numérique en un signal analogique.

I.5.5 Stockage

L'opération de stockage permet la mémorisation de l'image numérique sur des supports destinés à cet effet comme les bandes magnétiques, les disques durs,.....

I.6 Segmentation d'images couleur

La segmentation d'images couleur en régions est un traitement de bas niveau qui consiste à extraire de l'image des primitives, soit de type contour, soit de type régions. A la différence du prétraitement, qui transforme l'image couleur en une autre image couleur, la segmentation transforme l'image couleur en une autre image qui peut être binaire (contours et non contours), ou une image de région pour laquelle tous les pixels d'une même région sont affectés d'une même étiquette. Il y'a donc nécessairement passage de trois composante à une seul composante. Il existe trois approches possibles

pour le traitement des images couleur en vue de la segmentation à savoir l'approche scalaire, l'approche marginale et l'approche vectorielle : [8]

Dans l'approche scalaire, les composantes sont fusionnées permettant le passage d'une image couleur à une image monochrome. Cela peut se faire, par exemple, par une quantification.

Dans l'approche marginale, les composantes sont traitées séparément et les trois résultats sont ensuite fusionnés.

Dans l'approche vectorielle, chaque pixel est traité comme un vecteur dans l'espace couleur, les primitives sont extraites en tenant compte des trois composantes à la fois.

I.6.1 Définition mathématique de la segmentation [2]

ZUCKER a défini formellement la segmentation d'image en s'appuyant sur la partition d'un ensemble I en n sous ensembles R_1, R_2, \dots, R_n tels que :

- $\cup R_i = I$, pour $i=1, \dots, n$.
- $R_i \cap R_j = \emptyset$ avec $i \neq j$.
- Les sous ensembles $R_i, i=1, \dots, n$ sont connexes.
- Il existe un prédicat P tel que $P(R_i) = \text{vrai}, \forall i=1, \dots, n$.
- $P(R_i \cup R_j) = \text{faux}, \forall (i,j), i \neq j$ et R_i, R_j étant adjacent dans I .

Nous décrirons par la suite trois types méthodes de segmentation d'images couleurs à savoir l'approche contours, les méthodes de segmentation par analyse des propriétés spatiales (l'approche région) et les méthodes de segmentation par analyse des composantes couleurs des pixels.

I.6.2 L'approche contour [7]

Les techniques d'extraction de contours dans une image couleur se basent sur la recherche des discontinuités, par le moyen de méthodes dérivatives ou morphologique. Les discontinuités d'une image couleur s'obtiennent de manière marginale ou vectorielle.

Dans la manière marginale, l'analyse des discontinuités est réalisée sur chacune des composantes couleurs de façon indépendante, les résultats marginaux sont ensuite combinés avec des opérateurs logiques de manière à obtenir un résultat mono composant.

Dans la manière vectorielle, on exploitant des grandeurs calculées de manière conjointe à partir des trois composantes colorimétriques.

I.6.2.1 Approches marginales

Dans une image couleur, la transition entre deux objets adjacents peut être caractérisée par des changements brutaux de chrominance. Les dérivées premières (gradient) ou secondes (Laplacien) sont exploitées dans ce but.

La dérivée du plan image C_i , $i=1, \dots, 3$ ($C_1=R$, $C_2=V$ et $C_3=B$), ou gradient ∇C_i se caractérise par ses deux coordonnées $G_x^{C_i}$ et $G_y^{C_i}$, mais également par sa norme ainsi que par son orientation. Les contours sont détectés dans chaque plan image par la recherche des maxima du gradient. La figure I.9.b illustre l'obtention d'un gradient à partir d'un contour schématisé sur la figure I.9.a.

Le calcul des dérivées secondes, ou laplacien $\nabla_2 C_i(x, y)$ permet de détecter les contours ou d'améliorer une détection effectuée par analyse du gradient. La détection du passage par zéro du laplacien correspond à la fois aux maxima et aux minima du gradient comme le montre la figure I.9.c.

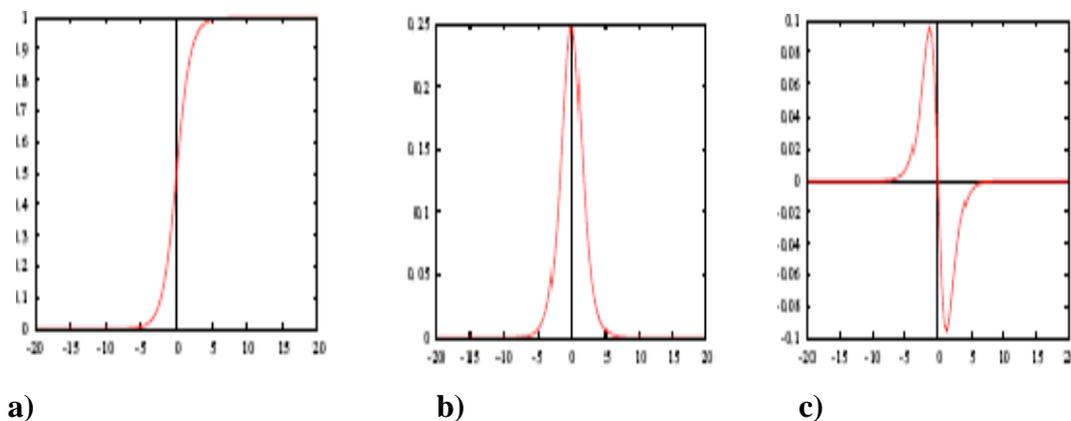


Figure I.9 Illustration de la détection de contours. (a) Transition entre deux régions, (b) Dérivée première associée et (c) Dérivée seconde.

I.6.2.2 Approche multi composante ou Vectorielle

Du fait des fortes corrélations spatiales et chromatiques entre chacun des plans image, il est généralement plus satisfaisant de détecter les contours de manière globale. Nous présentons à ce propos le gradient vectoriel.

Le gradient multi composante considère qu'une image exprimée dans l'espace (C_1, C_2, C_3) est représentée par un champ vectoriel C . La matrice du gradient spatial D de C est alors définie par la matrice Jacobienne de la fonction $(C_1(x, y), C_2(x, y), C_3(x, y))$ suivant les coordonnées spatiales x et y d'un point p de la manière suivante :

$$D(p) = \begin{pmatrix} \frac{\partial C_1}{\partial x} & \frac{\partial C_1}{\partial y} \\ \frac{\partial C_2}{\partial x} & \frac{\partial C_2}{\partial y} \\ \frac{\partial C_3}{\partial x} & \frac{\partial C_3}{\partial y} \end{pmatrix}$$

I.6.3 Méthodes de segmentation par analyse des propriétés spatiales [6]

Nous introduisons dans cette section quelques méthodes qui divisent directement l'image en régions distinctes, ces méthodes considèrent chaque région comme un ensemble de pixels connexes dont les couleurs sont proches les unes des autres. On procède par un balayage de l'image afin de construire les régions et on fait intervenir l'information colorimétrique pour la prise de décision. Nous pouvons distinguer quatre approches qui sont

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

I.6.3.1 Segmentation par division de régions

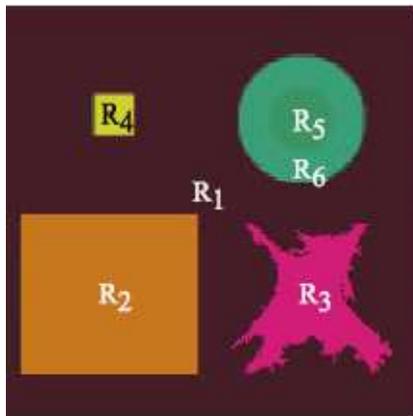
L'approche segmentation par division de régions consiste à diviser l'image originale en régions homogènes au sens d'un critère donné.

Ce processus est récursif et considère que la région initiale correspond à l'image à analyser. Si une région ne respecte pas un prédicat d'homogénéité, elle est divisée en

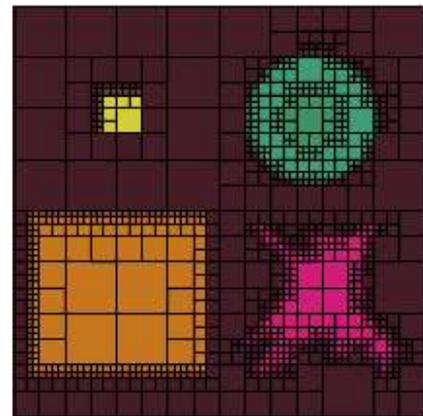
quatre sous-régions de taille égale. Chaque sous-région est ensuite analysée. L'algorithme récursif s'arrête lorsque toutes les régions respectent le prédicat d'homogénéité.

A cause des divisions en quatre des régions, cette méthode est plutôt adaptée à des images carrées ayant un nombre de lignes et de colonnes égal à une puissance de deux, et dans lesquelles les régions sont de forme rectangulaire. D'autre part, cette méthode a tendance à faire apparaître des effets de blocs.

A titre d'illustration, nous nous proposons de segmenter l'image de la figure I.10.a par division de régions. Chaque composante couleur de cette image de synthèse est codée sur 256 niveaux.



a) Image originale



b) Image segmentée par la méthode de division de régions

Figure I.10 Exemple de segmentation par division de régions.

I.6.3.2 Segmentation par croissance de régions

Ce type de segmentation 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 colorimétrique.

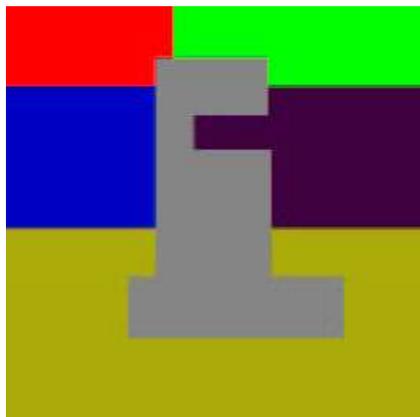
La plupart des algorithmes de croissance de régions sont régis par un parcours de l'image selon le balayage de gauche à droite et de haut en bas.

L'avantage de la croissance de régions est de préserver la forme de chaque région de l'image. Cependant, une mauvaise sélection des germes et/ou un choix du critère de similarité mal adapté peuvent entraîner des phénomènes de sous-segmentation ou de sur-segmentation.

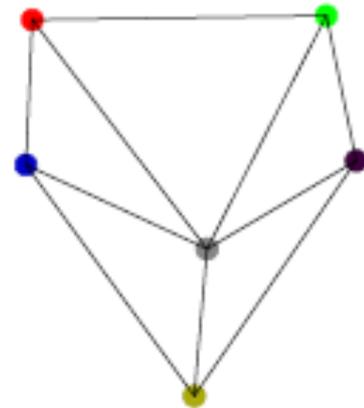
I.6.3.3 Segmentation par fusion de régions

L'approche de fusion de régions est principalement fondée sur l'analyse d'un graphe d'adjacence de régions qui analyse une image pré-segmentée, constituée d'un ensemble de régions.

C'est une structure de données constituée d'un graphe non-orienté dont chaque nœud représente une région et chaque arête représente une adjacence entre deux régions. La figure I.11 représente, à titre d'exemple, le graphe d'adjacence de régions obtenu à partir de l'image originale.



a) Image originale



b) Graphe d'adjacence de région de l'image originale.

Figure I.11 Exemple de graphe d'adjacence de régions

I.6.3.4 Segmentation par division-fusion

La segmentation par division-fusion regroupe les deux types d'approches précédents. Tout d'abord, l'image est divisée en régions homogènes qui respectent des critères globaux, puis les régions adjacentes qui répondent à des critères locaux sont fusionnées. Nous décrirons la méthode des quadtree qui est la plus utilisée dans la segmentation par division-fusion.

Cette méthode consiste à segmenter les images couleur en régions par l'analyse d'une structure appelée quadtree.

Le quadtree est un tétra-arbre dont chaque nœud possède exactement quatre nœuds fils, excepté les nœuds terminaux. Chaque nœud correspond à un bloc c'est-à-dire une zone de l'image de forme carrée, dont le nombre de lignes et de colonnes est une puissance de deux. La racine de l'arbre correspond à l'image entière qui doit être de forme carrée et dont le nombre de lignes et de colonnes est également une puissance de deux (figure. I.12).

Chaque bloc associé à un nœud du quadtree de la partition initiale est analysé de façon récursive afin de décider s'il doit être divisé en quatre sous-blocs. L'analyse récursive s'arrête lorsque chaque sous-bloc respecte un prédicat d'homogénéité.

A ce stade de l'analyse, certains blocs adjacents dans l'image présentent des caractéristiques colorimétriques identiques. C'est pourquoi ces couples de blocs sont fusionnés. L'analyse s'arrête lorsqu'il n'existe plus de couple qui respecte le prédicat de fusion.

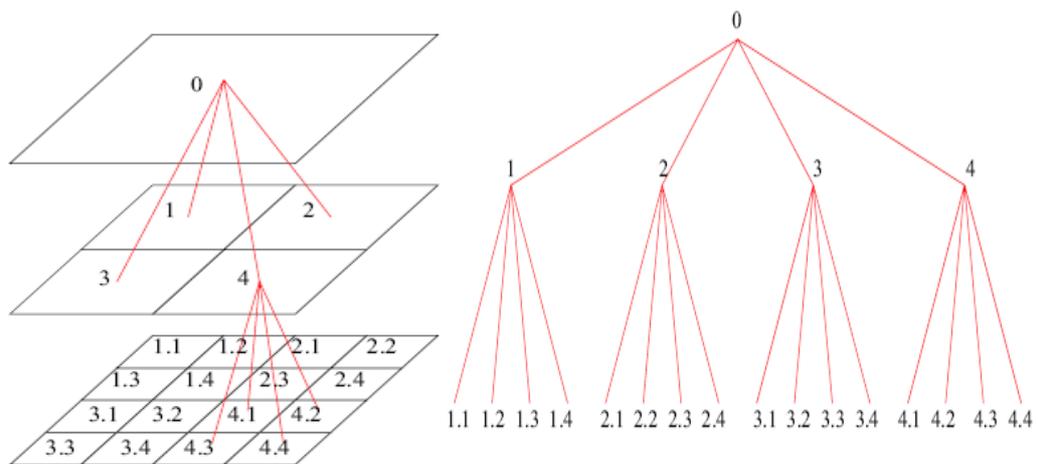
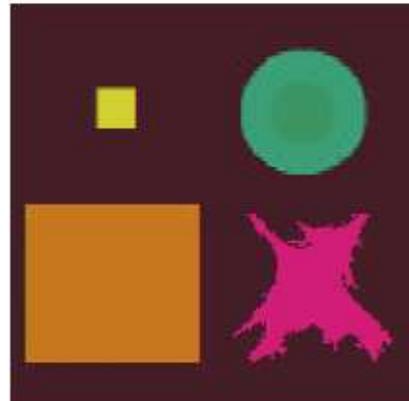


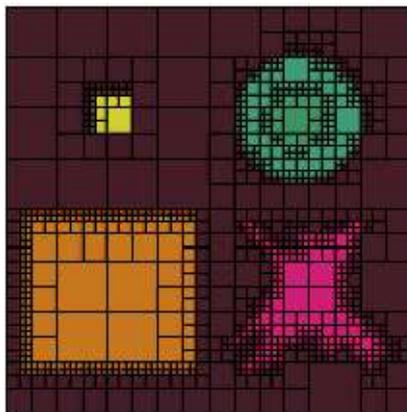
Figure I.12 Structure pyramidale du quadtree

L'inconvénient majeur du quadtree réside dans la rigidité du découpage carré qu'il impose. Il conduit à un partitionnement global de l'image qui ne respecte pas toujours la forme des régions présentes dans l'image. Par ailleurs, la phase de regroupement des blocs est sensible à l'ordre du parcours du quadtree.

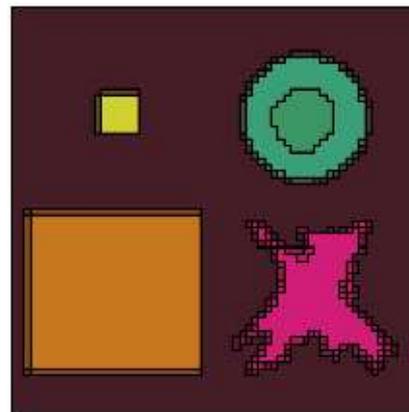
La figure I.13 illustre le problème du découpage carré avec le résultat obtenu après la phase de division-fusion de l'image originale.



a) Image originale



b) Image segmentée après la phase de division



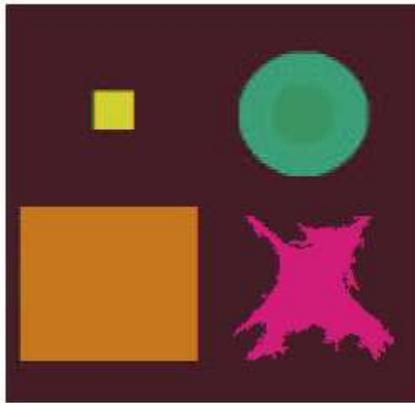
c) Image segmentée après la phase de fusion.

Figure I.13 Exemple de segmentation par division-fusion.

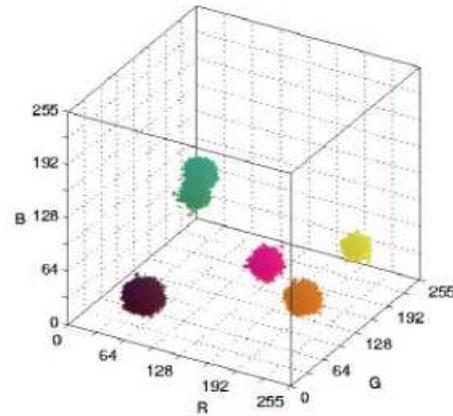
I.6.4 Méthodes de segmentation par analyse des composantes couleurs des pixels

Ces méthodes considèrent une région comme un ensemble de pixels connexes appartenant à une même classe. Elles supposent donc que les pixels qui appartiennent à une même région possèdent des couleurs similaires et forment un nuage de points bien identifiable dans l'espace de représentation des couleurs 3D (Figure I.14).

La classification consiste à retrouver ces nuages de points qui correspondent aux classes de pixels présentes dans l'image.



a) Image originale



b) Nuages de points dans l'espace des couleurs R, V et B

Figure I.14 Exemple de nuages de points dans l'espace RVB d'une image couleur

La classification est dite supervisée lorsque des informations a priori sont introduites dans le processus de construction des classes sous la forme d'un échantillon d'apprentissage. La classification est dite non supervisée lorsque aucune connaissance a priori n'est disponible. Toutes les méthodes de classification des données sont exploitables dans ce cas. Nous pouvons citer parmi les méthodes qui nécessitent la connaissance du nombre de classes l'algorithme des K-means et des Fuzzy c-means, et dans les parmi les algorithmes qui ne nécessitent pas la connaissance du nombre de classes l'algorithme Isodata.

L'algorithme des k-means consiste à partitionner l'ensemble des pixels de l'image en un nombre de classes prédéterminé par l'utilisateur. Pour chaque classe, un pixel est sélectionné de façon aléatoire parmi l'ensemble des pixels de l'image afin de constituer son centre de gravité initial et à chaque itération, chaque nouvelle affectation d'un pixel à une classe entraîne la remise à jour immédiate de son centre de gravité. Il est à noter que lorsque deux centres de gravité sont proches, les deux classes associées ne sont pas fusionnées. D'autre part, un centre de gravité isolé dans l'espace des couleurs peut être associé à une classe qui ne contient aucun pixel.

L'algorithme des fuzzy c-means est un algorithme de classification floue fondé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité. L'algorithme nécessite de connaître le nombre de classes au préalable et génère les classes de telle sorte que les sommes des écarts

quadratiques interclasses et intraclasse soient respectivement maximale et minimale. Cette approche est donnée en détaille dans le chapitre III.

L'algorithme Isodata Cet algorithme est issu de l'approche des k-means. Il permet de partitionner l'ensemble des pixels sans connaître a priori le nombre exact de classes présentes dans l'image. Il introduit de nouveaux paramètres afin de modifier le nombre de classes au cours des itérations.

Lorsque la somme des variances des composantes couleurs des pixels appartenant à une classe devient plus grande qu'un seuil fixé, la classe est divisée en deux,

Lorsque la distance entre les centres de gravité de deux classes devient inférieure à un autre seuil, les classes sont fusionnées.

Conclusion

Nous avons présenté d'une manière générale quelques notions essentielles sur l'image couleur et les méthodes conventionnelles de segmentation d'images. Leurs diversités sont dues au fait que le choix d'une méthode par rapport à une autre est dicté par les caractéristiques particulières du problème considéré. Notre intérêt se portera sur les méthodes de segmentation par analyse des composantes couleurs des pixels et particulièrement sur l'algorithme des Fuzzy c-means.

II.1 Introduction

La théorie des sous ensembles flous est une théorie mathématique dont l'objectif est de modaliser les notions vagues du langage naturel pour pallier à l'inadaptation de la théorie des ensembles classiques. Ce concept a été aussi introduit dans le but d'éviter les passages brusques d'une classe à une autre (par exemple : de la classe noire à la classe blanche) et autoriser les éléments à n'appartenir complètement ni à l'une ni à l'autre.

La définition d'un ensemble flou répond au besoin de représenter des connaissances imprécises, soit parce qu'elles sont exprimées en langage naturel par un observateur qui n'éprouve pas le besoin de fournir plus de précisions ou n'en est pas capable, soit parce qu'elles sont obtenues avec des instruments d'observation qui produisent des erreurs de mesure.

II.2 Notions fondamentales sur les ensembles flous [5]

II.2.1 Définition d'un ensemble flou

Un sous ensemble flou A de l'univers X est défini par une fonction d'appartenance μ_A qui associe à chaque élément x de X le degré $\mu_A(x)$, compris entre 0 et 1, avec lequel x appartient à A .

Exemple : Si X est l'ensemble des âges possibles d'une personne, on peut définir le sous ensemble flou A des personnes jeunes. Le degré d'appartenance de toute personne âgée de x au sous ensemble flou A est d'autant plus grand que son âge est petit (Figure II.1).

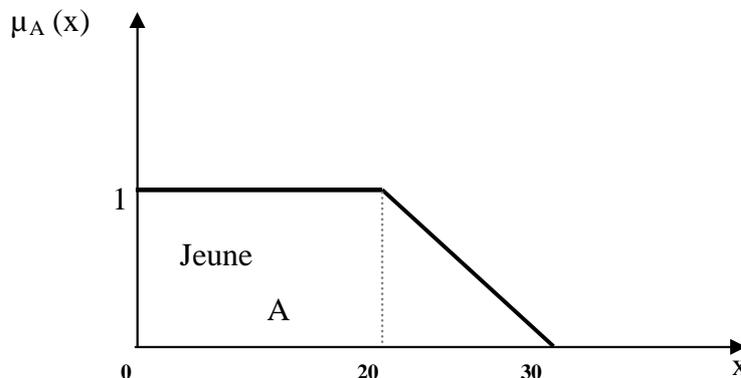


Figure II.1 Sous ensemble flou A représentant la population jeune, l'âge est représenté par la variable x .

Donc, soit un sous ensemble flou A de l'univers X définie par une fonction d'appartenance

$$\begin{aligned}\mu_A : X &\longrightarrow [0,1] \\ x &\longrightarrow \mu_A(x)\end{aligned}$$

Le sous ensemble flou A s'écrira alors :

$$A = \{(x, \mu_A(x)), x \in X\}.$$

μ_A est appelée fonction d'appartenance et $\mu_A(x)$ appelée degré d'appartenance de l'élément x au sous ensemble flou A.

Selon les valeurs de $\mu_A(x)$ nous distinguons trois cas :

$\mu_A(x) = 1$ correspond à l'appartenance absolue de x à A.

$\mu_A(x) = 0$ correspond à la non appartenance absolue de x à A.

$0 < \mu_A(x) < 1$ traduit l'appartenance nuancée de x à A.

Dans le cas où μ_A ne prend que des valeurs égales à 0 ou 1, le sous ensemble flou A n'est rien d'autre qu'un sous ensemble classique de X.

II.2.2 Caractéristique d'un ensemble flou

Divers notions complémentaires, d'utilisation courante, permettent de mieux distinguer les sous ensembles flous des ensembles classiques :

II.2.2.1 Support

Soit A un sous ensemble flou de l'univers X. Le support du sous ensemble flou A est un sous ensemble classique de X, dont chacun des éléments a un degré d'appartenance non nul par rapport à A (voir Figure II.2) :

$$\text{Support}(A) = \{x \in X / \mu_A(x) \neq 0\}.$$

II.2.2.2 Noyau

Le noyau d'un sous ensemble flou A de X est un sous ensemble ordinaire de X dont chaque élément a un degré d'appartenance égale à 1 (voir Figure II.2) :

$$\text{Noyau}(A) = \{x \in X / \mu_A(x) = 1\}.$$

Lorsque $\text{Noyau}(A) = \emptyset$, on dit que le sous ensemble flou A est normal ou normalisé, sinon on dit qu'il est sous normal.

II.2.2.3 Hauteur

La hauteur d'un sous ensemble flou A de X est la valeur maximale de la fonction d'appartenance, c'est le plus grand degré d'appartenance à A (voir Figure II.2)

$$h(A) = \max (\mu_A (x)).$$

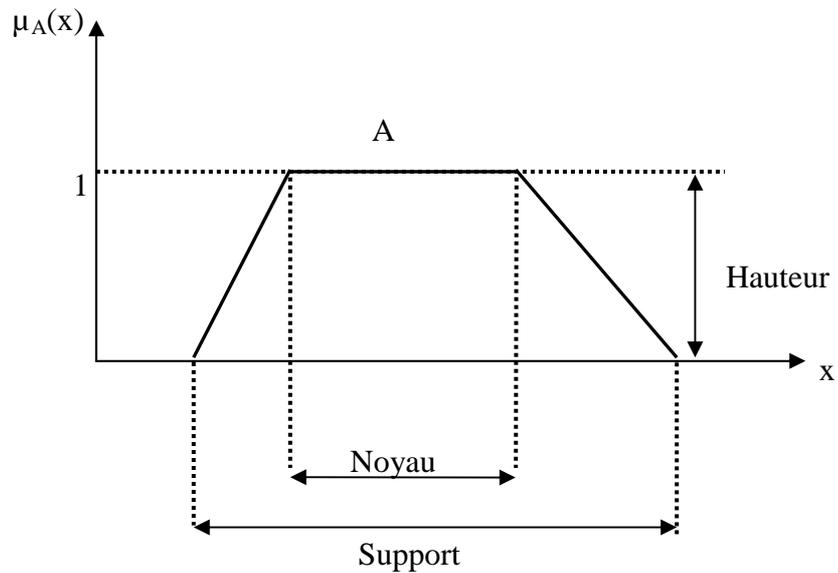


Figure II.2 Support, Noyau et Hauteur d'un sous ensemble flou

Un sous ensemble flou est normalisé si $h(A) = 1$.

II. 2.2.4 Cardinalité

La cardinalité d'un sous ensemble flou A de X est définie par :

$$|A| = \sum_{x \in X} \mu_A (x) \quad \text{si } X \text{ est fini}$$

qui indique le degré avec lequel les éléments de X appartiennent à A .

II.2.2.5 Sous ensemble convexe

Un sous ensemble flou A de X est convexe si, pour tout couple d'éléments a et b de X et pour tout nombre γ de $[0,1]$, la fonction d'appartenance de A vérifie :

$$\mu_A(\gamma a + (1 - \gamma)b) \geq \min (\mu_A(a), \mu_A(b)) .$$

Si A et B sont deux sous ensembles convexes flous de X, leur intersection est convexe.

Les sous ensembles flous les plus répandus sont ceux qui ont une fonction d'appartenance sans rupture brusque, c'est-à-dire régulière (sous ensemble convexe).

II.3 Exemple de fonctions d'appartenance [5]

Plusieurs fonctions d'appartenance utilisées dans la logique des ensembles flous, les plus répandues sont les fonctions d'appartenance trapézoïdale et exponentielle.

II.3.1 Fonction d'appartenance trapézoïdale

La fonction d'appartenance trapézoïdale est donnée par l'équation suivante :

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a - c \text{ ou } x \geq b + d \\ x/c + (1-a/c) & \text{si } a-c < x < a \\ 1 & \text{si } a \leq x \leq b \\ -x/d + (1+b/d) & \text{si } b < x < b + d \end{cases}$$

Sa courbe est représentée par la figure suivante :

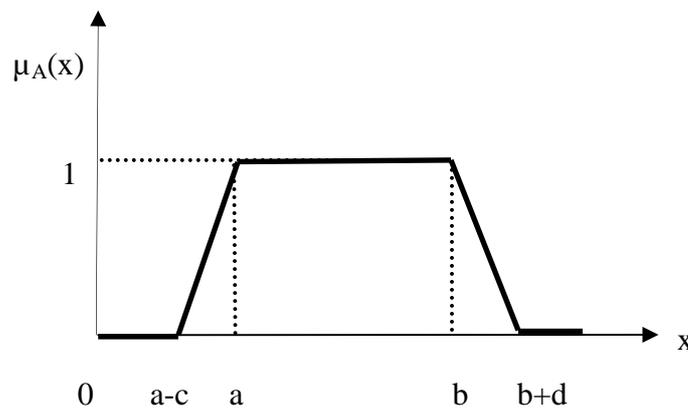


Figure II.3 Exemple de fonction d'appartenance trapézoïdale

Dans le cas où $a=b$ la fonction est dite triangulaire et sa courbe est représentée par la figure suivante :

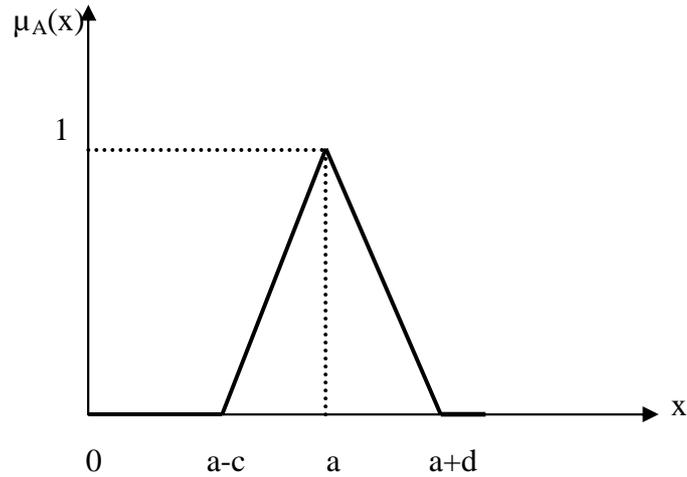


Figure II.4 Exemple de fonction d'appartenance triangulaire

II.3.2 Fonction d'appartenance exponentielle

La fonction exponentielle est sous la forme :

$$\mu_A(x) = \begin{cases} \exp(x-a) & \text{si } x < a \\ 1 & \text{si } a \leq x \leq b \\ \exp(-x+b) & \text{si } x > b \end{cases}$$

Sa courbe est donnée par la figure suivante

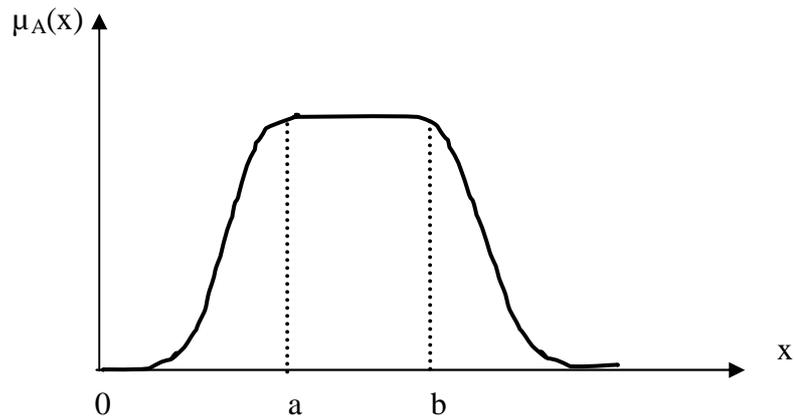


Figure II.5 Exemple de fonction d'appartenance exponentielle

II.4 Opérations sur les sous ensembles flous

Toutes les opérations ensemblistes définies pour les ensembles stricts ont été

généralisées de telles sortes qu'on puisse les utiliser pour les sous ensembles flous.

II.4.1 Inclusion

Un sous ensemble flou A est un sous ensemble d'un sous ensemble flou B si et seulement si pour tout éléments x de X, $\mu_A(x) \leq \mu_B(x)$.

$$A \subseteq B \text{ ssi } \forall x \in X ; \mu_A(x) \leq \mu_B(x).$$

II.4.2 Egalité

Soit deux sous ensembles flous A et B de l'univers X. On dit que A et B sont égaux si les fonctions d'appartenance ont la même valeur en tout point $x \in X$:

$$A = B \Leftrightarrow \forall x \in X ; \mu_A(x) = \mu_B(x).$$

II.4.3 Réunion

La réunion de deux sous ensembles flous A et B de même référentiel X est un sous ensemble flou noté $A \cup B$ dont la fonction d'appartenance est :

$$\forall x \in X ; \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)).$$

Tel que max désigne l'opérateur de maximisation.

La figure II.6 donne un exemple de graphe de la fonction d'appartenance correspondant à l'union de deux sous-ensembles flous A et B

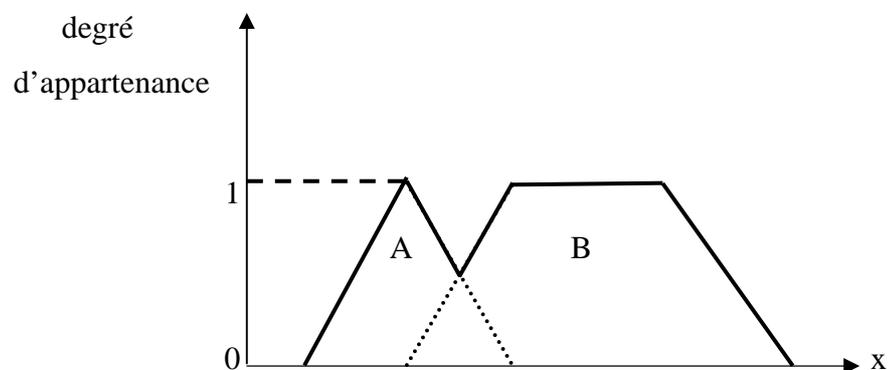


Figure II.6 Exemple de réunion de deux sous ensemble flous A et B

II.4.4 Intersection

L'intersection de deux sous ensembles flous A et B de même référentiel X est un sous ensemble flou noté $A \cap B$ dont la fonction d'appartenance est

$$\forall x \in X ; \mu_{A \cap B} (x) = \min (\mu_A (x), \mu_B (x)).$$

La figure II.6 donne un exemple de graphe de la fonction d'appartenance correspondant à l'intersection de deux sous-ensembles flous A et B

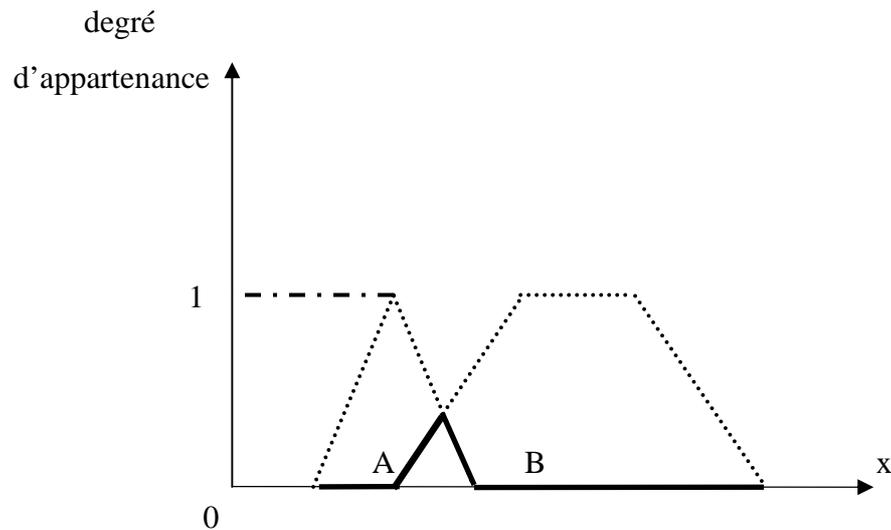


Figure II.7 Exemple d'intersection de deux sous ensembles flous A et B

II.4.5 Complément d'un sous ensemble flou

Le complément d'un sous ensemble flou A de référentiel X est un sous ensemble flou noté A^c dont la fonction d'appartenance est donnée par :

$$\forall x \in X, \mu_{A^c}(x) = 1 - \mu_A(x).$$

II.5 Définition d'une α coupe et d'une α coupe stricte

Pour un seuil donné α de $[0,1]$, nous définissons la α coupe de sous ensemble flou A de X comme le sous ensemble ordinaire A_α de X tel que

$$A_\alpha = \left\{ x \in X / \mu_A(x) \geq \alpha \right\}$$

Pour un seuil donné α de $[0,1[$, nous définissons la α coupe stricte de sous ensemble flou A de X comme le sous ensemble ordinaire $A_{\bar{\alpha}}$ de X tel que :

$$A_{\bar{\alpha}} = \left\{ x \in X / \mu_A(x) > \alpha \right\}$$

II.6 Indice de comparaison

Ce sont des paramètres utilisés pour permettre la comparaison des sous ensembles flous entre eux.

II.6.1 Indice de flou

Pour évaluer à quel point un sous ensemble flou A et son complémentaire se chevauchent, nous calculons l'indice de flou, noté par $\mu(A)$.

$$\mu(A) = 1 - \frac{\sum_{x \in X} |\mu_A(x) - \mu_{A^c}(x)|}{|A|}$$

Tel que $|\cdot|$ désigne le cardinal.

II.6.2 Indices de chevauchement

Ils mesurent le degré de chevauchement de deux sous ensemble flous, ils sont définis par:

$$a) I_1(A \cap B) = h(A \cap B).$$

$$b) I_2(A \cap B) = |(A \cap B)|.$$

Tel que h désigne la hauteur.

II.6.3 Indices d'inclusion

Ils permettent d'évaluer l'inclusion dans un sous ensemble flou A dans un autre sous ensemble flou B , ils sont au nombre de deux :

$$a) \text{INC}_1 (A \subseteq B) = \frac{|A \cap B|}{|A|} .$$

$$b) \text{INC}_2 (A \subseteq B) = 1 - h(A \cap B^c).$$

II.6.4 Indices d'égalité

Ils permettent d'évaluer l'égalité de deux sous ensembles flous. Ils sont donnés par :

$$a) \text{EG}_1 (A, B) = \min (A \subseteq B, B \subseteq A).$$

$$b) \text{EG}_2 (A, B) = \frac{|A \cap B|}{|A \cup B|} .$$

II.7 Modificateurs

Ils permettent de dériver de nouvelles fonctions d'appartenance, ce qui veut dire, définir de nouveaux sous ensembles flous à partir des fonctions d'appartenance initiales. Plusieurs modificateurs existent, les plus utilisés sont :

II.7.1 Normalisation

Elle est donnée par la formule suivante :

$$\mu_{\text{normale}(A)} (x) = \frac{\mu_A (x)}{h(A)} \quad \forall x \in X$$

Avec $h(A)$ qui désigne la hauteur de sous ensemble flou A .

II.7.2 Concentration

Elle est définie par :

$$\mu_{\text{con}(A)} (x) = (\mu_A(x))^2 \quad \forall x \in X .$$

II.7.3 Dilatation

Elle est définie par :

$$\mu_{\text{dil}(A)}(x) = \sqrt{\mu_A(x)} \quad \forall x \in X.$$

II.8 Produit cartésiens de sous ensembles flous

Il est parfois nécessaire de raisonner sur des éléments provenant de plusieurs ensembles de référence différents S_1, \dots, S_n .

On considère alors l'univers global S , produit cartésien de S_1, \dots, S_n :

$S_1 \times \dots \times S_n$ Ses éléments sont des n-uplets c-à-dire $x = (x_1, \dots, x_n)$.

Le produit cartésien de n sous-ensembles flous A_1, \dots, A_n définis respectivement sur les ensembles de référence S_1, \dots, S_n est alors un sous-ensemble flou A défini sur S par :

$$\forall x \in S; \mu_A(x) = \min(\mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n)).$$

II.9 Projection

A l'inverse, il peut être utile de se projeter sur les éléments constitutifs d'un univers correspondant à un produit cartésien, par exemple dans le contexte du raisonnement déductif flou.

La projection d'un sous-ensemble flou A de $S = S_1 \times \dots \times S_n$ sur le produit cartésien $U = S_a \times \dots \times S_k$, avec $\{a, \dots, k\} \subset \{1, \dots, n\}$ est le sous-ensemble flou noté $A \downarrow U$ de U de fonction d'apparence :

$$\forall u = (x_a, \dots, x_k) \in U, \mu_{A \downarrow U}(u) = \sup_{i/ 1 \leq i \leq n, i \neq a, \dots, i \neq k} \mu_A((\dots, x_i, \dots)).$$

II.10 Principe d'extension

Si X est l'ensemble des nombres réel, en observant ses éléments nous pouvons effectuer des opérations arithmétiques dessus, dont les résultats sont également des nombres réels. Quels résultats pouvons-nous obtenir lorsque les éléments de X ne sont qu'approximativement connus ?

C'est pour répondre à de telles questions que le principe d'extension a été introduit.

Ce principe est l'un des plus importants de la théorie des sous ensembles flous car il permet d'exploiter les connaissances classiques dans le cas de données floues. ce principe est énoncé comme suit :

Étant donné un sous ensemble flou A de X et une application φ de X vers Y, le principe d'extension permet de définir un sous ensemble flou B de Y associé à A par l'intermédiaire de φ .

$$\forall y \in Y \quad \mu_B(y) = \sup \mu_A(x) \quad \text{si } \varphi^{-1}(y) \neq \emptyset .$$

$$\text{Tel que } \{ x \in X; y = \varphi(x) \}$$

$$\mu_B(y) = 0 \quad \text{si } \varphi^{-1}(y) = \emptyset .$$

II.11 Application de la logique floue [2]

La résolution des problèmes avec la logique floue nécessite généralement, de suivre les trois étapes illustré sur la figure suivante :

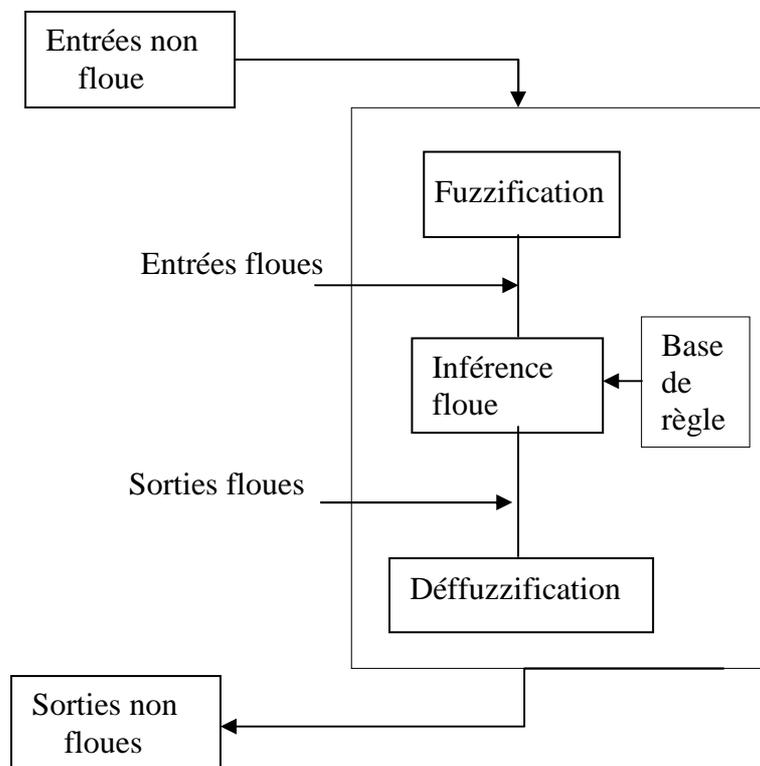


Figure II.8 Illustration du traitement flou.

II.11.1 Fuzzification

La fuzzification est une opération qui permet d'introduire des données floues. Une donnée floue est caractérisée par un terme linguistique (expression subjective), ce dernier est définie par une fonction d'appartenance dont la valeur appartient à l'intervalle $[0, 1]$.

II.11.2 Traitement flou

Le traitement flou constitue l'ensemble des opérations, que les quantités floues subissent.

II.11.3 Défuzzification

La défuzzification est la procédure inverse de la fuzzification. Elle consiste à convertir les valeurs floues obtenues à travers le traitement flou en valeur non floues.

II.12 Domaines d'application de la logique floue [2]

La logique floue a déjà fait ses preuves dans les domaines d'applications suivants :

- Gestion de projet,
- Sociologie,
- Détection de fraude à la protection sociale, identification de criminels,
- Gestion des aspirateurs, des systèmes de ventilation et de régulation thermique,
- Pilotage d'un système d'autofocus d'appareil photos,
- Système d'approche d'une station orbitale pour la navette spatiale américaine,
- Informatique,
- Traitements d'images et reconnaissance des formes.

Conclusion

Dans ce chapitre, nous avons décrit la théorie des sous ensembles flous qui permet de représenter des connaissances imprécises. Elle est particulièrement adaptée

pour raisonner à partir de connaissances vagues et approximatives et traiter des données appartenant à des catégories mal définies.

La logique floue est largement employée dans le domaine de la segmentation d'images où, elle peut être utilisée dans un but de classification, en n'exploitant que les fonctions d'appartenances à des classes bien déterminées. Le chapitre suivant détaillera la notion de segmentation par la logique floue

III.1 Introduction

Un ensemble flou est un ensemble dont les bords sont mal définis. Cela se traduit par une fonction d'appartenance à valeurs dans l'intervalle [0,1] tout entier, par opposition au cas classique où la fonction d'appartenance prend deux valeurs seulement 0 ou 1. Cela permet de définir des ensembles d'une manière plus souple, tolérant des informations imprécises, incomplètes et/ou incertaines.

Cette propriété est exploitée en traitement d'images, et plus précisément en classification où les classes, appelées aussi régions, sont représentées par des ensembles flous, cette notion est fort utile lorsque les régions ne peuvent pas être définies de manière nette et précise, la manipulation donc de ces ensembles, en gardant le caractère flou permet de traiter des données imprécises, incertaines et/ou redondantes d'une manière plus flexible.

III.2 Segmentation d'images par classification flou Fuzzy c-means (Fcm)

La méthode Fcm consiste à trouver l'appartenance exacte d'un pixel donné à une classe. Au départ, chaque pixel est assigné d'une valeur qui représente son degré d'appartenance dans chaque classe, Ce degré varie entre 0 et 1. Le principe consiste à trouver une partition de l'image caractérisée par une matrice de partition $U = \{[u_{ik}] / u_{ik} \in [0,1]\}$ qui caractérise le degré d'appartenance des pixels x_k à une classe c_i .

Soit la matrice U donnée par: [1]

$$U = \begin{pmatrix} u_{11} & \dots & u_{1n} \\ u_{21} & \dots & u_{2n} \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ u_{c1} & \dots & u_{cn} \end{pmatrix}$$

Où:

u_{ik} : degré d'appartenance du pixel x_k à la classe c_i .

n : nombre de pixels à classifier ($n =$ nombre de ligne x nombre de colonne dans l'image).

c : nombre de classes.

Avec : $0 \leq u_{ik} \leq 1$ et $\sum_i^n u_{ik} = 1 . \forall i \in \{1, 2, \dots, c\}, \forall k \in \{1, 2, \dots, n\}$.

Pour partitionner l'image, il faut minimiser le critère de la somme des distances intra-classes généralisée au cas flou et donné par la formule suivante: [1]

$$J_m(U, V, X) = \sum_{i=1}^c \sum_{k=1}^n (u_{ik})^m (d_{ik})^2 .$$

Avec : U est la matrice de partition floue d'éléments u_{ik} ,

$c = \{c_1, c_2, \dots, c_c\}$ est l'ensemble des vecteurs centres des classes,

$X = \{x_i / i=1, 2, \dots, n\}$ est l'ensembles des n pixels à classifier,

u_{ik} est le degré d'appartenance du pixel x_k à la classe c_i

$m \in]1, +\infty[$ est un facteur flou, qui règle le degré de flou entre les bords des classes. Si m tend vers 1, on tend vers une classification "nets", Par contre, plus m est grand, plus les classes des pixels sont "mous", c'est-à-dire que leurs fonctions d'appartenances est très étendues.

d_{ik} représente la distance euclidienne entre le pixel x_k et le centre de classe c_i . définie par

$$d_{ik}^2 = (x_k - c_i)^T I (x_k - c_i).$$

Tel que I est la matrice identité.

L'algorithme de partitionnement Fcm utilise donc l'ensemble des pixels $A = \{x_1, x_2, \dots, x_n\}$ où x_i est un vecteur mono composante dans le cas des images en niveaux de gris et à trois composantes lorsque qu'il s'agit d'une image couleur.

Les différentes étapes de cet algorithme sont : [9] [1]

Début

- Fixer le nombre de classes c tel que $2 \leq c \leq n$, n étant le nombre de pixels à classifier.
- Fixer une valeur pour m telle que $m > 1$.
- Fixer le seuil de convergence ϵ .

-1- Initialisation :

- Initialisation des centres de gravités des classes prédéfinis.

-2- Calculer I_k et I_g tel que :

$$I_k = \{i / 1 \leq i \leq c \text{ et } d_{ik}^2 = 0\}.$$

$$I_g = \{1, 2, \dots, n\} - I_k.$$

-3- Fuzzyfication :

C'est le calcul de la matrice de partition flou U, qui est donnée par les équations suivantes:

Si $I_k = \emptyset$

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}^2}{d_{jk}^2} \right)^{2/m-1}}$$

Si $I_k \neq \emptyset$

$$u_{ik} = 0 \quad \text{pour } i \in I_g.$$

$$u_{ik} = \frac{1}{|I_k|} \quad \text{pour } i \in I_k.$$

-4- Mise à jour des centres de classes v_i :

$$c_i = \frac{\sum_{k=1}^n (u_{ik}^{(b)})^m \cdot x_k}{\sum_{k=1}^n (u_{ik}^{(b)})^m} \quad \text{Tel que } 1 < i < c.$$

-5- Règle de décision :

Dans cette étape, on peut juger de la stabilité de la matrice de partition flou U en utilisant les deux cas suivants :

$$\|U_n - U_{n-1}\| > \varepsilon$$

$$\|U_n - U_{n-1}\| \leq \varepsilon. \quad \text{Tel que } U_{n-1} \text{ représente la matrice floue précédente.}$$

Dans le cas où:

Si $\|U_n - U_{n-1}\| > \varepsilon$ Retour à 4 pour la mise à jour de la matrice de partition par une nouvelle itération.

Si $\|U_n - U_{n-1}\| \leq \varepsilon$ La matrice floue converge, arrêter l'algorithme.

FIN.

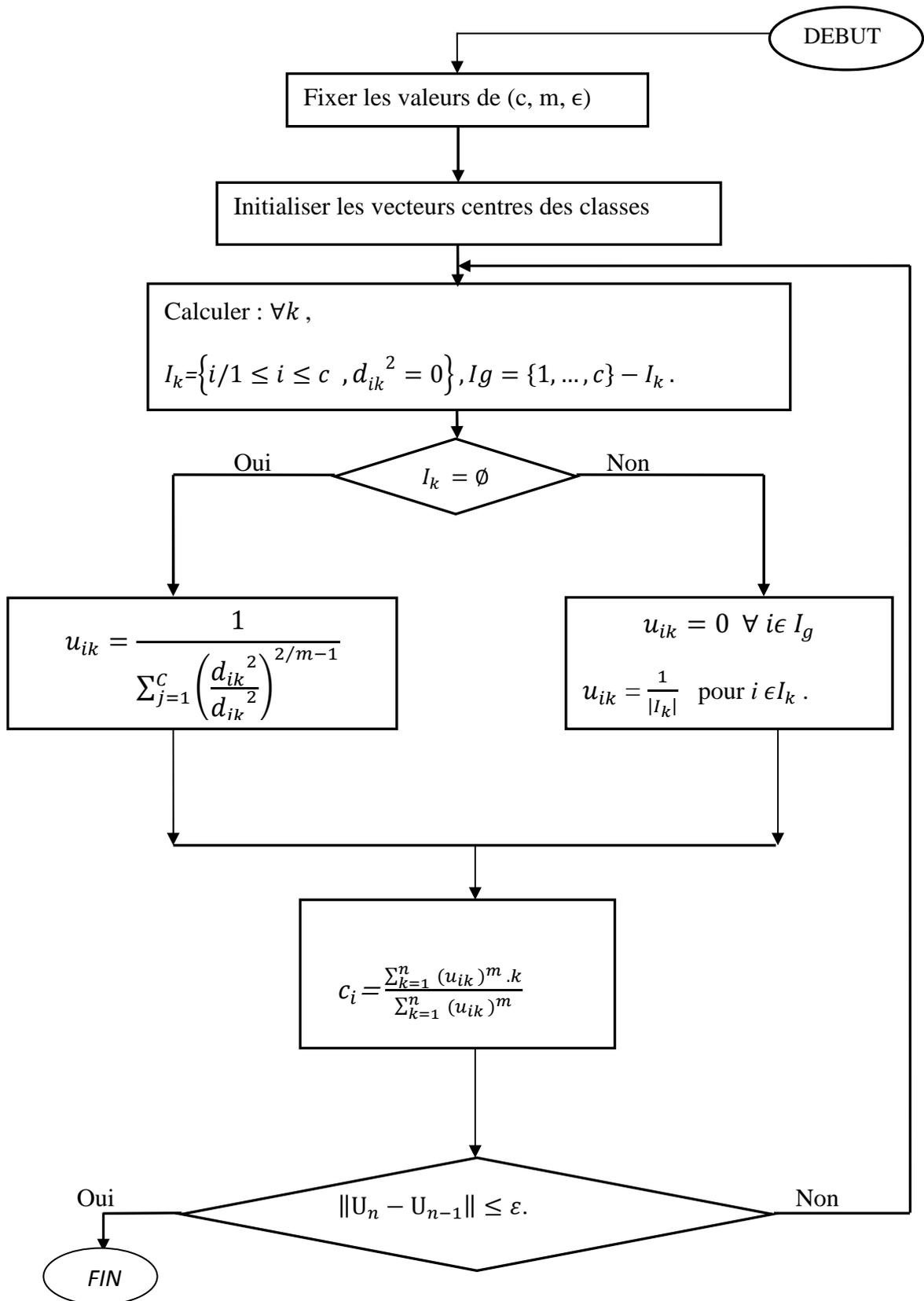


Figure III.1 Organigramme de partitionnement Fuzzy c-means

III.3 Variante de l'algorithme Fuzzy c-means adapté à la segmentation d'images

Dans le cas classique, la classification s'effectue sur le niveau de gris de chacun des pixels formant l'image, et comme les pixels ayant les mêmes niveaux de gris ont les mêmes degrés d'appartenance aux différentes classes, nous pouvons simplifier le nombre de formes à classer au nombre de niveau de gris présent dans l'image, sans altérer la qualité des résultats. Par exemple, pour une image de 512*512 pixels quantifiée en 256 niveaux de gris, nous passons 512*512 à 256 formes, en réduisant ainsi considérablement le temps de traitement. La seule différence se situe au niveau de l'équation donnant les centres des classes et donnée par la formule suivante :

$$c_i = \frac{\sum_{k=1}^n (u_{ik}^{(b)})^m \cdot x_k}{\sum_{k=1}^n (u_{ik}^{(b)})^m}$$

Cette équation devient comme suit

$$c_i = \frac{\sum_{k=N \min}^{N \max} (u_{ik}^m) \text{hist}(k) \cdot k}{\sum_{k=N \min}^{N \max} (u_{ik}^m) \text{hist}(k)}$$

Où:

N min : représente le pixel de niveaux du gris minimum présent dans l'image.

N max : représente le pixel de niveaux du gris maximum présent dans l'image.

k : représente un niveau de gris, tel que $N \min \leq k \leq N \max$.

hist(k): représente le nombre de pixels de niveau de gris k.

Les différentes étapes de l'algorithme de fuzzy c-means, adapté à la segmentation, sont décrites par l'organigramme suivant :

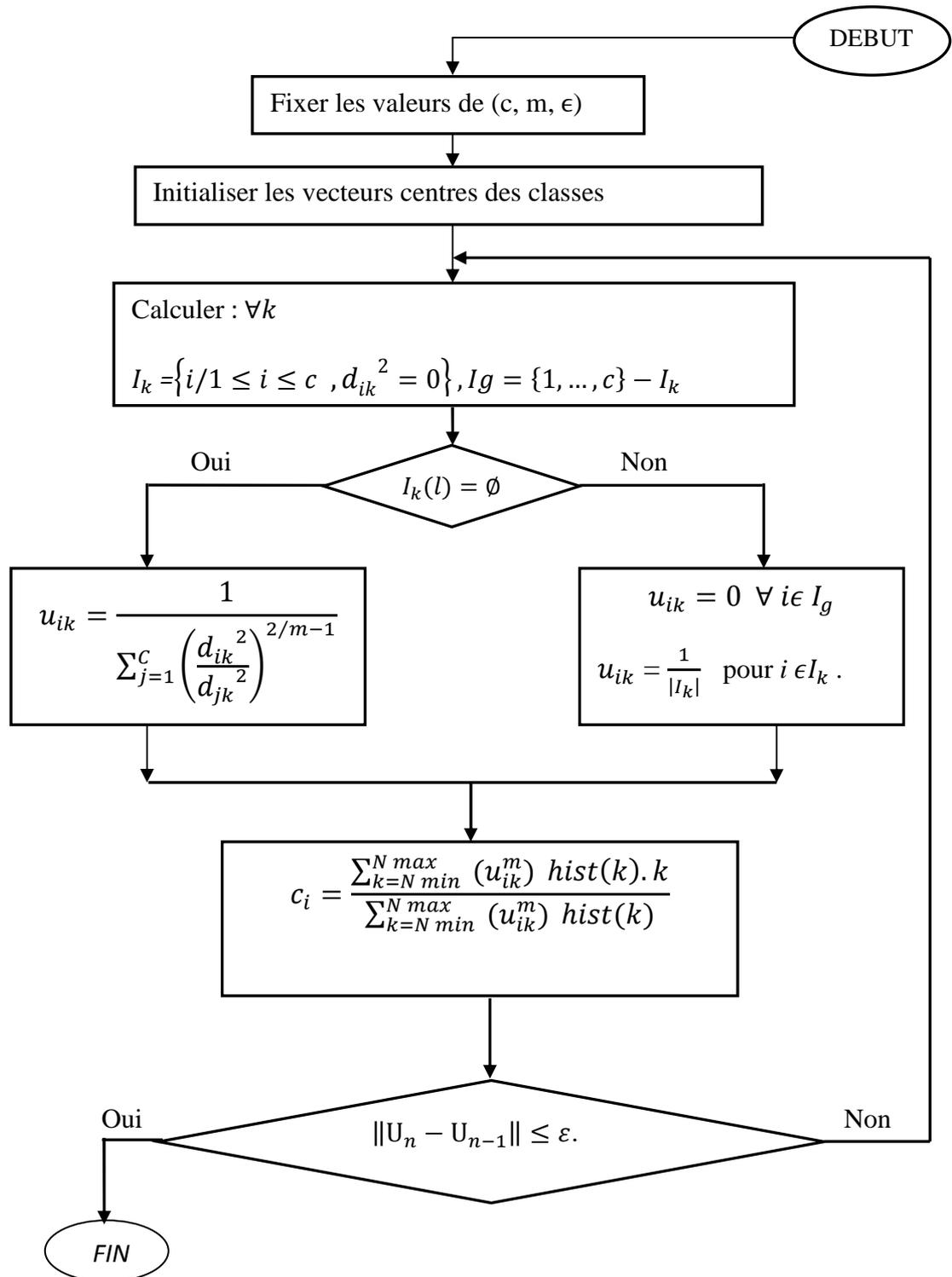


Figure III.2 Organigramme de Fuzzy c-means adapté à la segmentation d'images

Après convergence de l'algorithme, nous obtenons une partition floue de n niveaux de gris de l'image en c classes et un vecteurs des centres de classes.

les résultats obtenus ne nous permettent pas une segmentation complète de l'image, car à ce stade, chaque région est représentée par un sous ensemble flou auquel tout les pixels de l'image appartiennent avec des degrés d'appartenances différents. Alors que, chaque pixel doit appartenir à une seule classe.

Après la satisfaction de la condition de convergence, l'étape de défuzzification est donc impérative qui consiste à appliquer la règle floue choisie.

III.4 Défuzzification [2]

A l'issue de l'étape précédente, nous pouvons considérer chaque classe comme une image, dont chaque pixel est représenté non pas par son niveau de gris, mais par son degré d'appartenance à cette classe, ainsi, nous obtenons une représentation des degrés d'appartenance comme c'est illustré dans le schéma suivant :

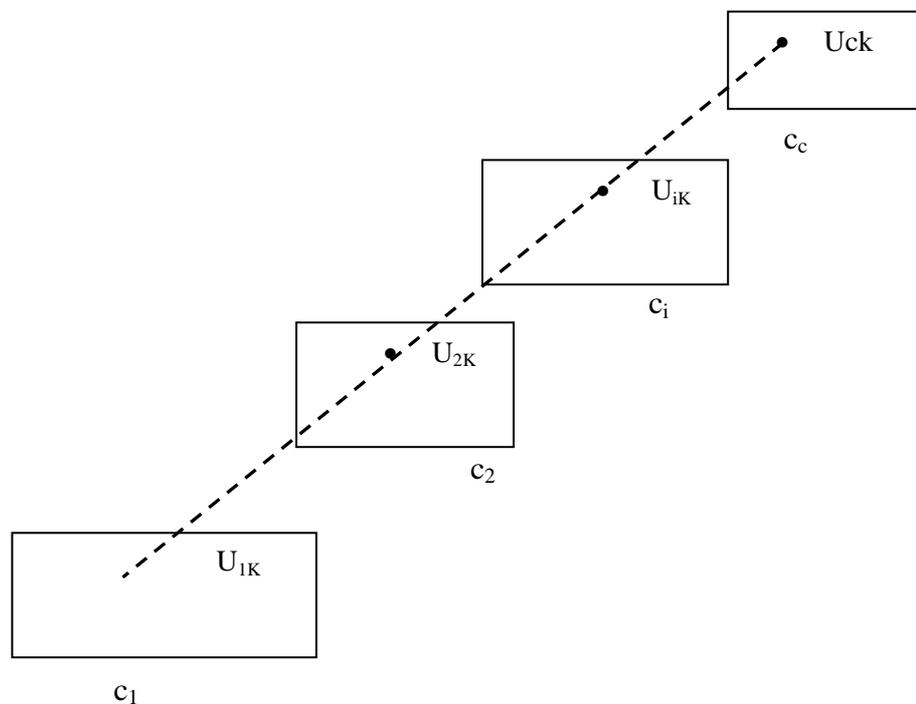


Figure III.3 Représentation spatial des degrés d'appartenance.

Pour aboutir à une image segmentée en c classes, une décision dépendante du contexte spatial dite décision par focalisation graduelle y généralement associée. Elle

nous permettra une défuzzification des résultats obtenus précédemment, cette fonction de décision n'est en fait que la combinaison de deux autres fonctions qui sont la fonction de décision par coupe de niveau α la fonction de décision par maximum d'appartenance.

III.5 Décision par focalisation graduelle

III.5.1 Fonction de décision par coupe de niveau α

Elle consiste à attribuer chaque pixel à une classe en ne prenant en considération que les différents degrés d'appartenance aux différentes classes :

Soient i une classe et k un niveau de gris

u_{ik} est le degré d'appartenance de k à i . telle que $1 \leq k \leq n$ et $1 \leq i \leq c$.

$$u_{ik} = \begin{cases} 1 & \text{Si } u_{ik} > \alpha, \alpha \text{ étant un seuil fixé par l'utilisateur} \\ 0 & \text{Sinon} \end{cases}$$

Ce type de fonction d'appartenance peut générer plusieurs difficultés dans le traitement. Pour illustre cela, supposons que nous ayons trois classes dans l'image : c_1 , c_2 , etc c_3 et supposons u_{c_1k} , u_{c_2k} et u_{c_3k} sont les trois degrés d'appartenance du pixel k respectivement aux trois classes suscitées.

Considérons par exemple $u_{c_1k}=0.3$, $u_{c_2k}=0.4$, $u_{c_3k}=0.3$ et $\alpha=0.5$. En appliquant la fonction de décision par coupe de niveau $\alpha=0.5$, le pixel x_k ne sera affecté à aucune des trois classes, car tout les degrés d'appartenance sont inférieur au seuil α .

Ce problème n'est pas le seule à se poser, pour illustrer cela, considérons l'exemple ou $u_{c_1k}=0.4$, $u_{c_2k}=0.2$, $u_{c_3k}=0.4$ avec $\alpha=0.3$.

Dans ce cas présent, nous constatons que le pixel est affecté à deux classes simultanément.

Pour ces deux raisons, cette fonction n'est jamais utilisée seule dans la défuzzyfication.

III.5.2 Fonction de décision par maximum d'appartenance

Dans cette fonction, un pixel est attribué à une classe si son degré d'appartenance à cette classe est le plus élevé.

Soit u_{ik} le degré d'appartenance du pixel k à la classe i .

$$u_{ik} = \begin{cases} 1 & \text{si } u_{ik} = \max (u_{jk}) \text{ avec } 1 \leq j \leq c, \\ 0 & \text{sinon.} \end{cases}$$

Cette fonction génère des difficultés. Pour illustrer cela considérons l'exemple de la matrice floue des degrés d'appartenance U donnée par :

	Pixel 1	pixel 2	pixel 3
	↓	↓	↓
Classe 1 →	0.3	0.47	0.25
Classe 2 →	0.4	0.46	0.35
Classe 3 →	0.3	0.07	0.4

La décision par maximum d'appartenance donnerait la partition suivante :

	Pixel 1	pixel 2	pixel 3
	↓	↓	↓
Classe 1 →	0	1	0
Classe 2 →	1	0	0
Classe 3 →	0	0	1

Les degrés d'appartenance du deuxième point à différentes classes sont (0.47, 0.46 et 0.07). Ce point a été affecté à la première classe vu que son degré d'appartenance à celle-ci est maximum, mais peut-on affirmer avec certitude son appartenance stricte à celle-ci, alors que son degré d'appartenance à la première est très proche de celui avec lequel il appartient à la deuxième classe ?

III.5.3 Fonction de décision par focalisation graduelle

La fonction de décision par focalisation graduelle évite les types de problèmes rencontrés sur les deux fonctions précédentes. En combinant donc ces deux fonctions de décision on obtient :

On pose $u_{ik} = \max (u_{jk})$ pour tout j tel que $1 \leq j \leq c$ et pour α fixé ((tel que α est appelé seuil d'ambiguïté), deux cas sont possibles :

$$\left\{ \begin{array}{l} \text{Si } u_{ik} \geq \alpha \quad \text{le pixel est dit fort} \\ \text{Sinon} \quad \quad \quad x_k \text{ est dit ambigu} \end{array} \right.$$

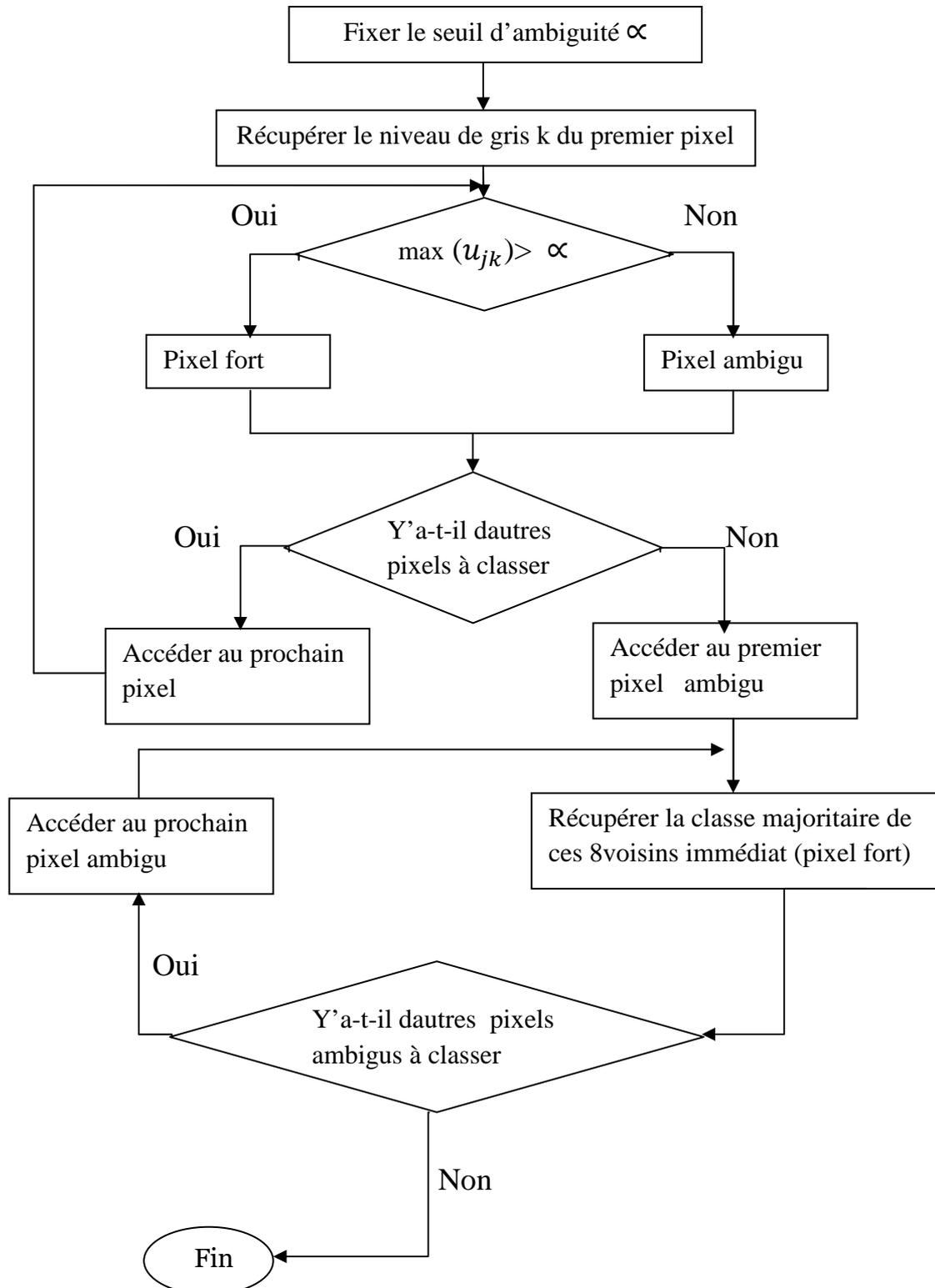
Pour éviter que les problèmes précédents se régénèrent nous devons fixer le seuil d'ambiguïté α à une valeur égale à 0.5, car s'il existe un degré d'appartenance supérieur à 0.5, il sera le seul. Les **pixels forts**, dont l'appartenance à une classe donnée est majoritaire, sont affectés de façon stricte à celle-ci, en remplaçant leurs niveaux de gris par le niveau de gris du centre de cette dernière.

Les **pixels ambigus**, sont traités en prenant en considération leurs voisinages spatiaux (8 voisins):

Pour tout pixel ambigu x_k de niveau de gris k , on cherche la classe majoritaire des pixels forts se situant dans le voisinage de x_k . Soit i cette classe majoritaire. Nous affectons alors le pixel x_k à la classe i .

Si plusieurs classes s'avèrent majoritaires en même temps, nous affectons le pixel x_k à l'une ou à l'autre indifféremment.

III.6 Organigramme des différentes étapes de la défuzification



Conclusion

Dans ce chapitre, nous avons décrit les détails de la méthode fondée sur la logique floue utilisée pour segmenter les images.

La première étape de cette méthode utilise l'algorithme de Fuzzy c-means qui peut être utilisé directement en considérant tous les pixels de l'image ou adapté à la segmentation d'image en considérant seulement les éléments de l'histogramme réduisant ainsi considérablement le temps de traitement.

La deuxième étape, permet la défuzification des résultats qui est obtenu à l'aide de la fonction de décision par focalisation graduelle, permettant ainsi d'obtenir une segmentation fine de l'image.

IV.1 Introduction

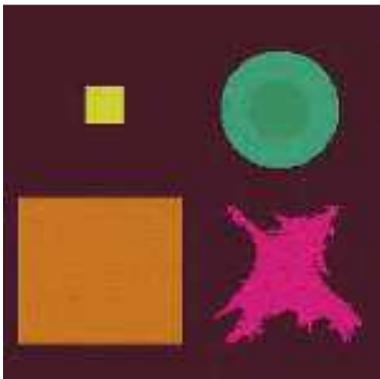
Nous avons implémenté sous Matlab 7.0 l'algorithme Fuzzy c-means sous ses deux versions à savoir la version qui traite un à un tous les pixels de l'image et la version qui utilise les histogrammes des différentes composantes spectrales de l'image couleur. Une comparaison des différents temps de calcul sera réalisée en fin de ce chapitre.

Nous commencerons par comparer les résultats de la segmentation de l'image couleur en utilisant la méthode de segmentation scalaire et la méthode de segmentation marginale.

Sachant que pour l'utilisation de l'algorithme Fuzzy c-means plusieurs paramètres doivent être fixés par l'utilisateur, nous évaluerons les résultats en faisant varier un à un ces différents paramètres qui sont en l'occurrence le seuil de convergence ϵ , le nombre de classe c , et le facteur floue m . La variation de ces éléments influe sur le nombre d'itération It de l'algorithme donc sur le temps de calcul. Nous donnerons à chaque fois ce nombre d'itération.

IV.2 Les images tests

Les tests sont effectués sur plusieurs images de types synthétiques et naturels représenté sur la figure IV.1. Ces images sont toutes de dimension 128×128 pixels² sauf pour l'image Lenna qui est sur 512×512 pixels².



a) Image Synthétique



b) Image Peppers



c) Image Jelly



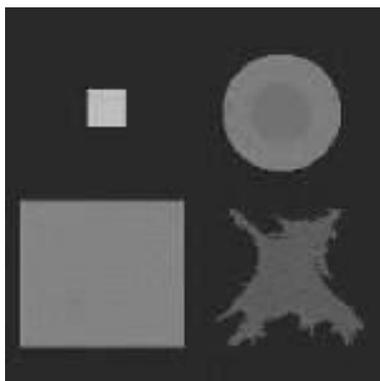
d) Image Lenna

Figure IV.1 Les images utilisées pour les tests.

IV.3 Comparaison entre la méthode de segmentation scalaire et la méthode de segmentation marginale

Dans cette partie, nous proposons de segmenter deux images à savoir l'image Synthétique et l'image Peppers. Pour réaliser la segmentation scalaire, nous devons fusionner les trois composantes couleurs. En ce qui nous concerne, nous les transformons simplement en images en niveaux de gris (voir figure IV.2).

La segmentation est alors réalisée en utilisant cette unique composante. Pour la visualisation des résultats de la classification, nous attribuerons à chaque étiquette, donc à chaque classe, la couleur moyenne des pixels qui composent l'image couleur.



a) Image Synthétique en niveaux de gris



b) Image Peppers en niveaux de gris

Figure IV.2 Exemple d'images en niveaux de gris

Pour la segmentation marginale, nous tenons compte des trois composantes chromatiques. Chaque composante est traitée indépendamment des autres comme une

image en niveaux de gris. A la fin les trois résultats sont fusionnés. Toujours pour la visualisation, nous attribuerons à chaque étiquette la couleur moyenne des pixels qui la composent.

Pour l'affectation de chaque pixel à une seule classe, la fusion des trois résultats obtenues c'est-à-dire les trois vecteurs des centres de classes est impérative. Pour se faire nous avons opté pour l'utilisation de la distance euclidienne des pixels par rapport aux centres des classes obtenus des trois plans rouge, vert et bleu.

La matrice des distances euclidiennes est donnée par :

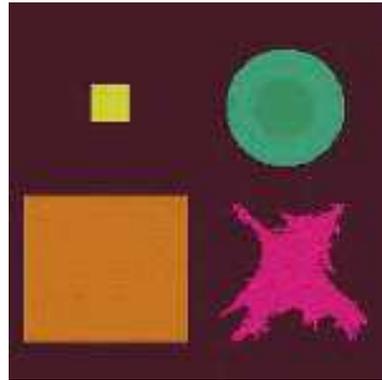
$$D = \begin{pmatrix} D_{11} & \dots & D_{1n} \\ D_{21} & \dots & D_{2n} \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ D_{c1} & \dots & D_{cn} \end{pmatrix}$$

Où :

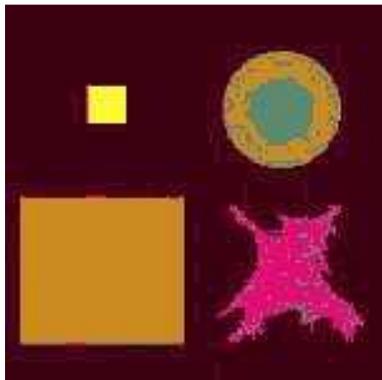
D_{ik} : est la distance en niveau de gris du pixel x_k à la classe c_i composée par R_i , V_i et B_i

L'affectation de chaque pixel de l'image à une classe se fait par la recherche de la distance minimale D_{ik} du pixel x_k par rapport aux différentes classes c_i .

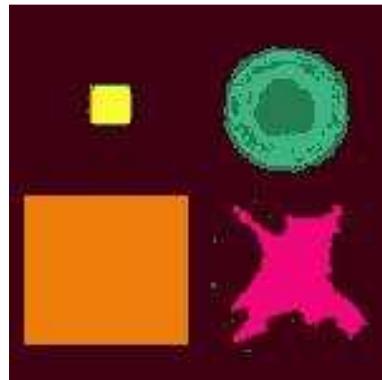
Les figures IV.7 et IV.8 illustrent les résultats obtenus par la méthode de segmentation scalaire et la méthode de segmentation marginale, les différents paramètres étant fixés.



a) Image originale



b) Segmentation scalaire

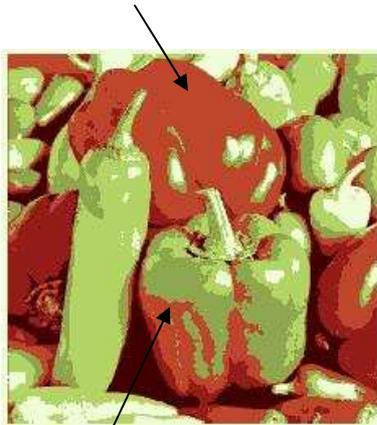


c) Segmentation marginale

Figure IV.7 Résultats de la segmentation de l'image Synthétique avec $c = 6$, $\varepsilon = 0.001$, $m = 1.5$, $\beta = 0.5$ (seuil d'ambiguïté).



a) Image originale



b) Segmentation scalaire



c) Segmentation marginale

Figure IV.8 Résultats de la segmentation de l'image Pepper
avec $c = 6$, $\varepsilon = 0.001$, $m = 1.5$, $\beta = 0.5$

Nous remarquons dans la figure IV.7, que dans le cas de la segmentation scalaire, les pixels de la couronne externe sont mal classés car ils sont attribués à la même classe que ceux du grand carrée. Les classes sont mieux différenciées dans le cas de la segmentation marginale et le résultat est bien meilleur.

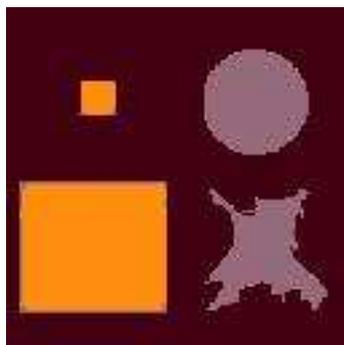
Dans l'image de la figure IV.8, une partie des pixels qui représentent le poivron vert foncé est assignée à la même classe que le poivron rouge dans le cas de la segmentation scalaire (voir les flèches). Cela signifie que la segmentation par la méthode scalaire ne permet pas de distinguer ces deux régions alors qu'elles sont bien séparées dans le cas de la segmentation marginale.

D'après ces résultats, et d'autres que nous n'avons pas affichés, nous constatons toujours que la segmentation marginale fournit un meilleur résultat, c'est pour cela, que pour toute la suite des testes, nous utilisons cette méthode.

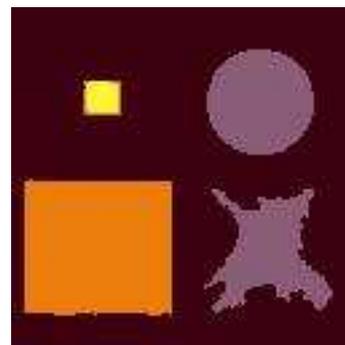
IV.4 Influence de la valeur du nombre de classes c sur la segmentation d'images couleurs

Dans cette partie, nous proposons d'étudier l'influence de c sur la segmentation des différentes images couleurs. Nous segmentons bien sur par la méthode marginale.

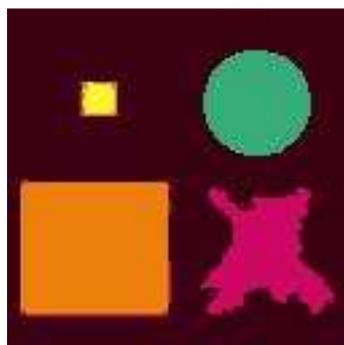
Les résultats de la segmentation des images Synthétique et Jelly sont présentés respectivement sur les figures IV.9 et IV.10, les paramètres ε et m étant fixe ($\varepsilon=0.01$, $m=1.5$).



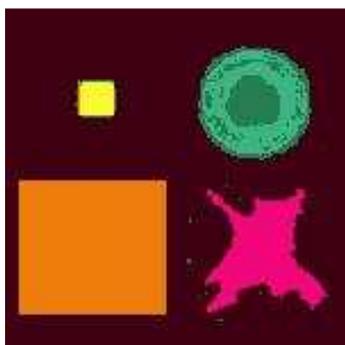
a) $c=3$, $It=66$



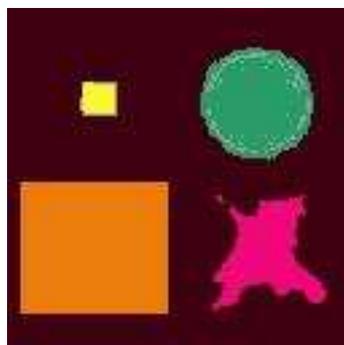
b) $c=4$, $It=115$



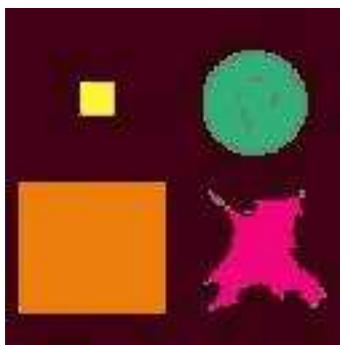
c) $c=5$, $It=142$



d) $c=6$, $It=189$



e) $c=7$, $It=207$



f) $c=8$, $It=308$

Figure IV.9 Résultats de la segmentation de l'image Synthétique en 3, 4, 5, 6, 7 et 8 classes

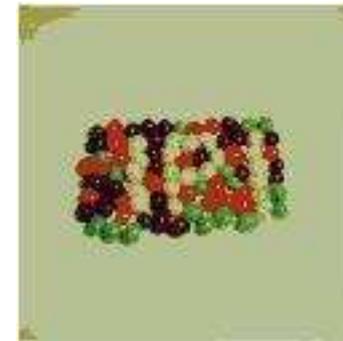
a) $c=3$, $It=41$ b) $c=5$, $It=163$ c) $c=7$, $It=269$ d) $c=8$, $It=383$ e) $c=9$, $It=420$ f) $c=10$, $It=336$

Figure IV.10 Résultats de la segmentation de l'image Jelly en 3, 5, 7, 8,9 et 10 classes.

Après une analyse globale des images de la figure IV.9 et IV.10, nous constatons que lorsque le nombre de classes est bien définie on a une bonne segmentation comme le montre les figures IV.9 d et IV.10 c. Par contre si le nombre de classes est mal choisi on a une mauvaise segmentation. Cela est dû à l'un des inconvénients de cet algorithme qui nécessite la connaissance préalable du nombre de classes.

IV.5 Influence de la valeur du seuil de convergence ϵ

Les figures IV.11 et IV.12 illustrent l'influence de la variation du seuil de convergence ϵ sur les résultats de la segmentation de l'image Lenna et de l'image Synthétique.



a) $\epsilon=0.0001$, It=581



b) $\epsilon=0.2$, It=119



c) $\epsilon=0.4$, It=83



d) $\epsilon=0.6$, It=65



e) $\epsilon=0.9$, It=48



f) $\epsilon=1$, It=44

Figure IV.11 Résultats de la segmentation de l'image Lenna en 7 classes avec différentes valeurs de ϵ .

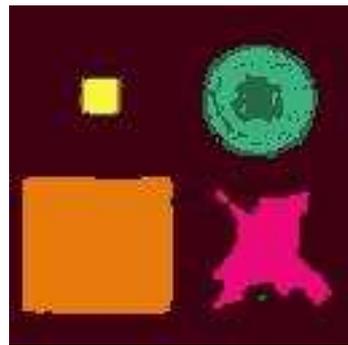
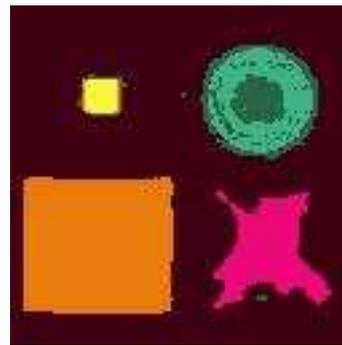
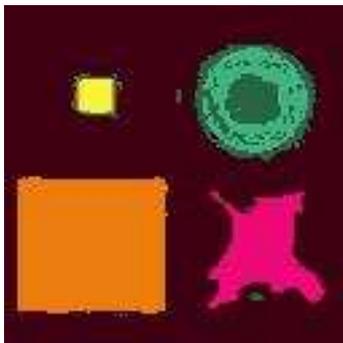
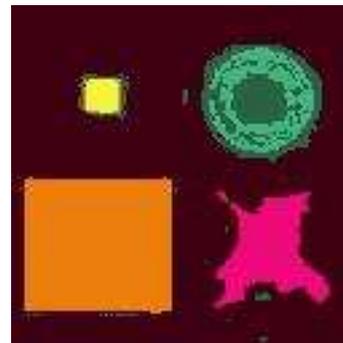
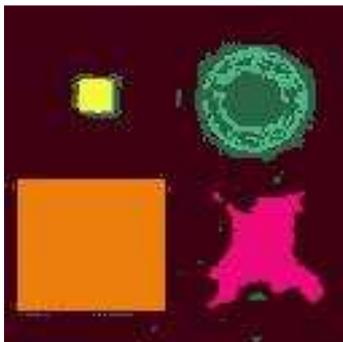
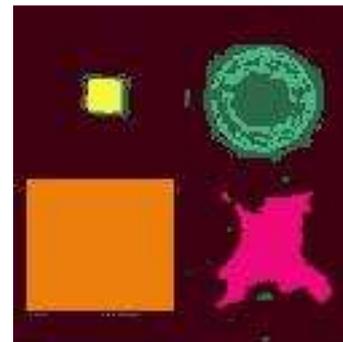
a) $\epsilon=0.0001$, It=367b) $\epsilon=0.2$, It=75c) $\epsilon=0.4$, It=48d) $\epsilon=0.6$, It=36e) $\epsilon=0.9$, It=27f) $\epsilon=1$, It=26

Figure IV.12 Résultats de la segmentation de l'image Synthétique en 6classes avec différentes valeurs de ϵ .

Nous remarquons que le nombre de classes construites est relativement stable pour différentes valeurs de ϵ , et que pour de très faibles valeurs de ϵ , nous avons une meilleure segmentation de point de vue qualité des résultats. En effet si nous observons les résultats de l'image synthétique de la figure IV.12, nous pouvons bien voir que dans le cas f, par exemple, le carrée jaune est entouré de pixels mal classés.

Ce résultat est tout à fait prévisible, car ce paramètre correspond à la valeur maximale de l'erreur autorisée. Plus cette valeur est faible, moins l'erreur est importante et meilleur sont les résultats.

IV.6 Influence de la valeur du facteur flou m

La figure IV.13 illustre l'influence du facteur flou m sur les résultats de la segmentation. Lorsqu'elle est appliquée à l'image de la figure IV.1 b.



a) $m=1.2$, $It=276$



b) $m=2$, $It=215$



c) $m=4$, $It=402$



d) $m=6$, $It=496$



e) $m=8$, $It=432$



f) $m=10$, $It=326$

Figure IV.13 Résultats de la segmentation de l'image Peppers en 6 classes avec différentes valeurs de m .

Dans la figure IV.13 nous constatons que l'augmentation de la valeur du facteur flou m conduit à une augmentation du nombre d'itérations d'une part, et d'autre part à une mauvaise classification comme le montre la figure IV.13 f. L'image classée avec la valeur $m = 2$ semble être le meilleur résultat car les classes sont assez bien définies avec un temps de calcul relativement faible (sachant que le temps de calcul est proportionnel au nombre d'itérations).

IV.7 Analyse de la variation du Nombre d'itérations It et la fonction coût J_m

Après avoir effectué tous ces tests sur l'influence et le choix des paramètres précédents, nous procédons à l'analyse d'autres paramètres, à savoir le nombre d'itérations It et la fonction coût J_m . Rappelons que la fonction coût J_m est donnée par la formule suivant

$$J_m = \sum_{i=1}^c \sum_{j=1}^n \mu_{ik}^m (d_{ik})^2.$$

avec :

μ_{ik} désigne le degré d'appartenance du pixel x_k à la classe i .

m est le facteur flou.

d_{ik} est la distance entre le pixel x_k et le centre de la classe i .

Pour étudier le changement du nombre d'itérations It trouvé à la fin de l'exécution du programme, nous faisons une initialisation fixe des centres des classes, car une initialisation aléatoire engendrera des résultats différents même si tous les autres paramètres sont fixés. Les tests sont réalisés sur l'image Lenna de la figure IV.1

Nous fixons le nombre de classes c à 3 et ε à 0.01, puis nous exécutons plusieurs fois le programme en changeant à chaque fois seulement la valeur du facteur flou m . La valeur de It est ensuite lu. Nous refaisons le même travail pour d'autres valeurs de c . Les résultats obtenus sont affichés sur le graphe de la figure IV.13.

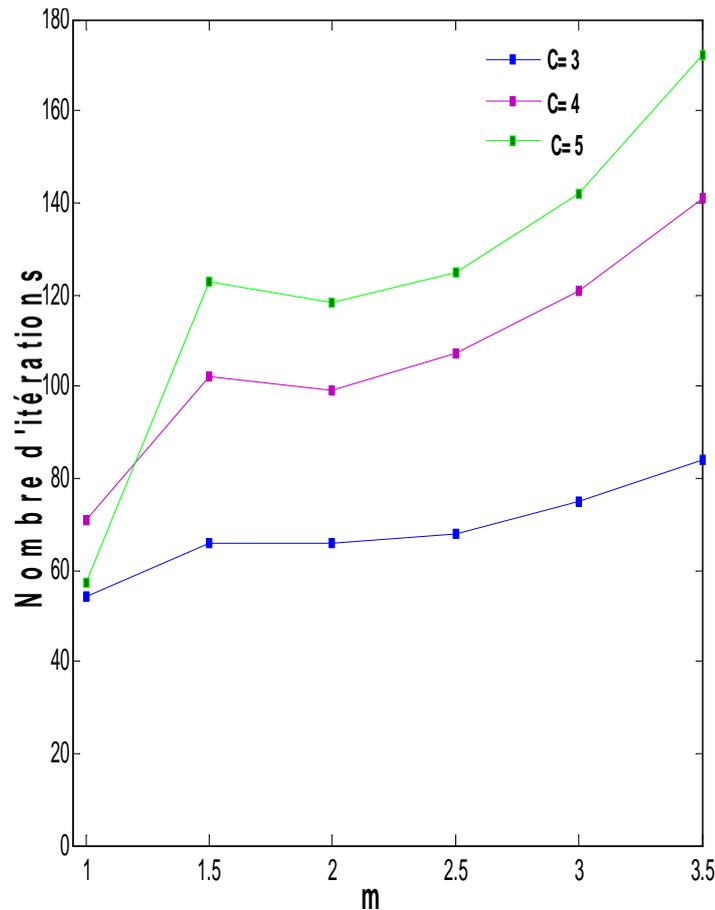


Figure IV.13 Influence de m et de c sur le nombre d'itérations It ($\epsilon=0.01$).

Nous remarquons que dans ce cas là, les courbes de It obtenues en fonction de m sont globalement croissantes. Cela signifie que pour des résultats assez semblables (puisque ϵ est fixé) le temps de calcul augmente avec m . Il est tout à fait prévisible de trouver que le temps de calcul (qui est proportionnel au nombre d'itérations), a augmenté en augmentant le nombre de classes, cela est bien relaté sur la figure IV.13

Pour étudier l'influence du seuil de convergence ϵ sur le nombre d'itérations It , nous avons fait varier c et ϵ avec m fixe ($m=1.5$), les résultats obtenus sont affichés sur le graphe de la figure IV.14.

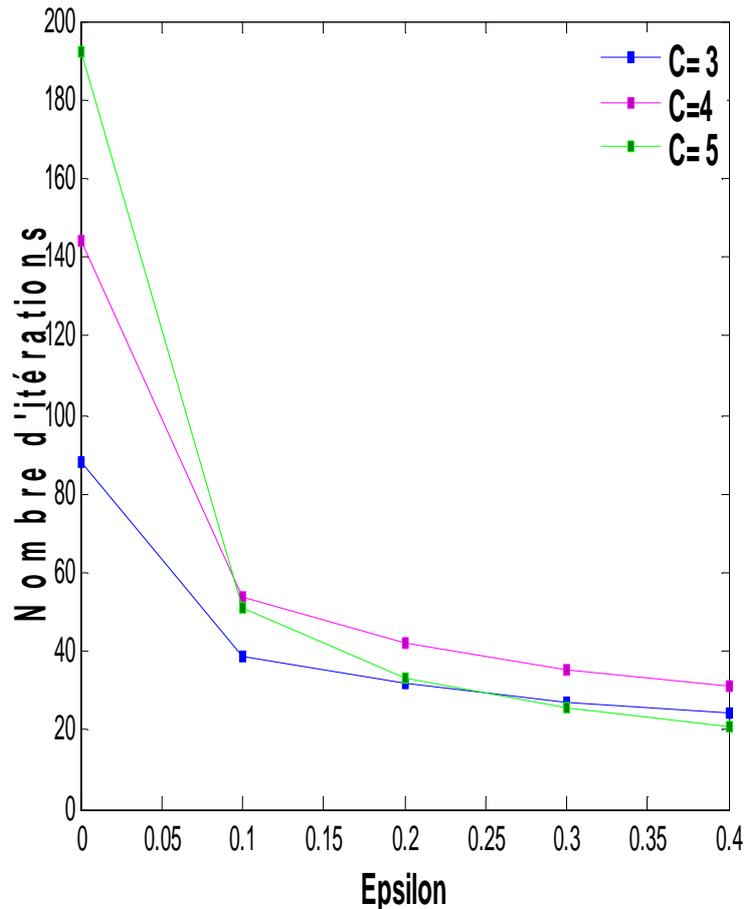


Figure IV.14 Influence de ε et de c sur le nombre d'itérations I_t ($m=1.5$).

D'après la figure IV.14, nous remarquons que les courbes de I_t obtenues en fonction de ε sont globalement décroissantes. Le nombre d'itérations diminue donc le temps de calcul en augmentant l'erreur autorisée (ε) d'une part, d'autre part la variation de c n'influe pas beaucoup sur le nombre d'itérations I_t ce qui est bien illustré sur la figure IV.14.

Nous constatons que le nombre d'itérations est lié beaucoup plus au seuil de convergence ε .

La figure IV.15 illustre le comportement de la fonction coût J_m en faisant varier le facteur flou m et le nombre de classes.

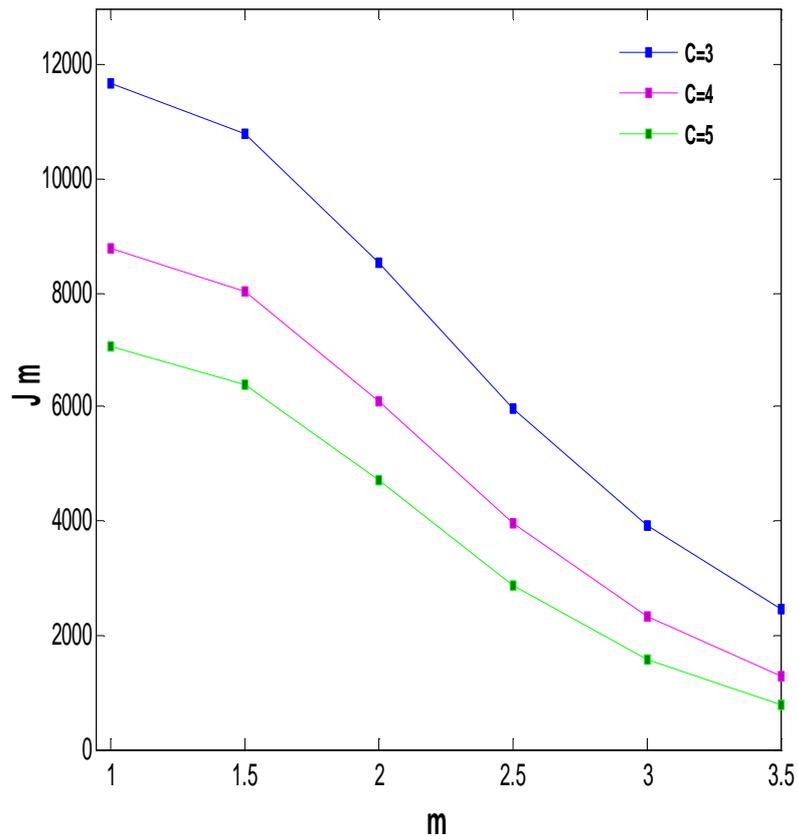


Figure IV.15 Influence de m et c sur la fonction coût J_m .

En observant la figure IV.15, nous constatons que la fonction coût J_m traduisant l'erreur quadratique intra-classe diminue lorsque m ou c augmente.

IV.8 Etude du temps de calcul nécessaire

Nous proposons d'aborder l'aspect des temps de calcul nécessaires pour les deux versions de la segmentation. Celles-ci ont été implantées sur une architecture du type PC avec un microprocesseur Intel Pentium 4 doté d'une mémoire de 128 MO cadencé à 2.26 GHz. Les tests sont effectués sur des images de taille 128×128 Pixels², et un cas particulier d'une image de taille 512×512 pixels². Nous fixons c à 6, ε à 0.01 et m à 1.5. Nous regroupons les résultats dans les deux tableaux ci-dessous. L'un fournit les résultats en utilisant une segmentation scalaire (donc dans notre cas une image en niveaux de gris) et l'autre en utilisant une segmentation marginale (donc en traitant indépendamment les trois composantes colorimétriques).

Image à segmenter	Temps d'analyse en utilisant l'algorithme de partitionnement FCM	Temps d'analyse en utilisant la méthode de partitionnement FCM basée sur l'analyse d'histogramme.
Image de Synthèse Taille : 128 x 128 Pixels ²	2 minutes 5 secondes	11 secondes
Image Lenna Taille : 128 x 128 Pixels ²	2 minutes 45 secondes	3 secondes
Image Lenna Taille : 512 x 512 pixels ²	20 minutes	1 minute 58 secondes

Figure IV.16 Temps de calcul avec les deux variantes de segmentation FCM en utilisant la méthode scalaire.

Image à segmenter	Temps d'analyse en utilisant l'algorithme de partitionnement FCM	Temps d'analyse en utilisant la méthode de partitionnement FCM basée sur l'analyse d'histogramme.
Image de Synthèse Taille : 128 x 128 Pixels ²	5 minutes 45 secondes	15 secondes
Image Lenna Taille : 128 x 128 Pixels ²	3 minutes 48 secondes	11 secondes
Image Lenna Taille : 512 x 512 pixels ²	45 minutes 50 secondes	4 minutes 55 secondes

Figure IV.17 Temps de calcul avec les deux variantes de segmentation FCM en utilisant la méthode marginale.

En analysant les figures IV.16 et IV.17, nous remarquons que l'algorithme de partitionnement FCM requiert un temps de calcul nettement plus élevé dans le cas où nous utilisons l'algorithme de partitionnement FCM classique. Ce temps est d'autant plus élevé lorsque la taille de l'image augmente et lorsque nous utilisons un traitement

marginal. En effet si nous observons le résultat de la segmentation marginal de l'image Lenna de taille 512 X 512 pixels², nous remarquons que ce temps est d'environ 1 heure pour le FCM classique alors qu'il ne dépasse pas 5 minute lorsqu'on utilise la méthode basée sur l'analyse d'histogramme.

Nous concluons donc que la méthode de segmentation en utilisant l'algorithme de partitionnement FCM basée sur l'analyse d'histogramme réduit considérablement le temps de calcul. En effet, au lieu d'effectuer l'analyse de tous les pixels de l'image un à un, cette analyse est réalisée seulement sur les 256 valeurs de l'histogramme de l'image dans le cas scalaire ou 256 x 3 valeurs des histogrammes des différentes composantes de l'image couleur dans le cas marginale.

Conclusion

Nous avons testé l'algorithme FCM sous ces deux versions et en utilisant les deux avec approches scalaire et marginale sur une série d'images en couleur. Toutes les différentes étapes des algorithmes ont été programmées. Nous avons étudié l'influence des différents paramètres sur les résultats de la segmentation.

Enfin nous avons mesuré le temps d'exécution de l'algorithme sous ces deux versions à savoir la méthode qui traite un à un tous les pixels de l'image et la méthode qui est basée sur l'analyse d'histogramme de l'image.

Conclusion générale

L'objectif du travail qui nous été proposé, consiste en la réalisation d'un programme permettant la segmentation non supervisée d'images couleur en régions, en incorporant les principes de la théorie des ensembles flous.

Ce travail nous a permis de nous familiariser avec trois domaines à savoir le traitement d'images, la programmation sous Matlab et la théorie des ensembles flous.

Nous avons implémenté l'algorithme Fuzzy c-means sous le logiciel Matlab version 7.0. En se référant aux résultats expérimentaux exposés au chapitre IV, nous avons pu observer que, de manière générale, que l'algorithme étudié permet une bonne segmentation d'une part, d'autre part d'un point de vue temps de calcul, s'avère plus efficace quel que soit la taille de l'image.

Nous avons réalisé une segmentation scalaire et marginale des images couleur. La segmentation marginale, bien que prenant plus de temps de calcul, fourni de meilleurs résultats. Il serait intéressant de comparer ces résultats avec une segmentation vectorielle.

L'inconvénient majeur de l'algorithme FCM de manière général est qu'il nécessite le choix préalable du nombre de classes. Si ce choix n'est approprié nous obtenons une mauvaise segmentation de l'image. Il serait judicieux de programmer au préalable une méthode qui permettra le choix automatique de ce nombre de classes.

- [1] YOUCEF SMARA, NADIA OUARAB. Technique de fusion et de classification d'images satellitaires multisources pour la caractérisation et le suivi de l'extraction du tissu urbain de la région d'Alger. Université d'Alger USTHB.
- [2] M^{lle} TASSADIT BELAICHE, M^r KAMAL MEHAOUED. Segmentation d'images en niveau de gris par la méthode de la logique floue. Mémoire d'ingénieur option Matériel & Logiciel, UMMTO 1999.
- [3] Kahina Lemikchi, Fatiha Ousmaal, Aldjia Rahali. Segmentation Markovienne des images multispectrales MSG. Mémoire d'ingénieur, département d'Electronique, faculté de Génie Electrique et Informatique, université Mouloud Mammeri de Tizi Ouzou, 2007 (Proposé et dirigé par Sadia Alkama)
- [4] J.P COQUEREZ, S. PHILIPP. Analyse d'images, filtrage et segmentation .Masson, PARIS 1995.
- [5] BERNADETTE BOUCHON MEUNIER, Addison-WESLY. Logique floue et ses applications. 1995.
- [6] MICHAEL FONTAINE. Segmentation non supervisée d'images couleur par analyse de la connexité des pixels. Thèse de doctorat, université de l'Lille1, décembre 2001.
- [7] MICHELE GOUIFFES. Apport de la couleur et des modèles de réflexion pour l'extraction et le suivi de primitives. Thèse de doctorat, université de POITIERS, décembre 2005.
- [8] LAURENT BUSIN. Sélection automatique d'espaces couleur pour la segmentation d'images. Thèse de doctorat, université de l'Lille1, décembre 2006.
- [9] M^{lle} NORA SI SEGHIR, M^r HICHEM HACHEROUF. Approche floue appliquée à la segmentation d'images. Mémoire d'ingénieur option Système Informatique (ML), département d'informatique UMMTO 1999.