



N° d'ordre :

UNIVERSITE DE M'SILA
FACULTE DES MATHÉMATIQUES ET DE L'INFORMATIQUE
Département d'Informatique

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Systèmes d'Informations Avancés

Par: ZAHRA YAHIAOUI

SUJET

**Etude et implémentation de l'algorithme c moyenne floue
et ses variantes**

Soutenu publiquement le : / /2013 devant le jury composé de :

.....	Université de M'sila	Président
Mme. O.Assas	Université de M'sila	Rapporteur
.....	Université de M'sila	Rapporteur
.....	Université de M'sila	Examineur
.....	Université de M'sila	Examineur

Promotion : 2012 /20 13

Dédicaces

Je dédie ce modeste travail,

À ma mère avec toute mon affection,

À mon père avec toute ma reconnaissance,

À mes sœurs,

À mes frères,

À ma famille,

Et à tous mes amis.

Zahra yahiaoui

Remerciement

Je remercie tout d'abord le bon dieu pour m'avoir donnée le courage et la santé pour accomplir ce travail.

Ce travail n'aurait pas pu aboutir à des résultats sans l'aide et les encouragements de plusieurs personnes que je remercie.

J'adresse mes plus vifs remerciements à Mme ASSAS Ouarda qui m'a encadrée durant ce projet. Elle m'a fait connaître le domaine du traitement d'images, Je lui exprime toute ma gratitude, car elle a su me conseiller et me diriger avec ses remarques pertinentes.

Je tiens à exprimer aussi mes sincères remerciements à :

*Mes parents, pour leur encouragement, leur soutien et pour les sacrifices qu'ils ont enduré.
Je remercie très vivement les membres de jury pour avoir accepté de juger ce modeste travail.
Enfin, j'exprime mes remerciements à tous les professeurs de département d'Informatique.*

TABLE DES MATIERES

Liste des figures.....	v
Liste des tableaux.....	vi
Introduction générale	1
<i>Chapitre 1 Traitement d'images</i>	
1.1. Introduction.....	2
1.2. Définition de l'image	2
1.3. Image numérique.....	3
1.4. Type des images.....	3
1.4.1. Les images vectorielles.....	3
1.4.2. Une image matricielle (ou bitmap)	4
1.5. Conception de base d'un Images bitmap.....	4
1.5.1. Le pixel et le niveau de gris.....	4
1.5.2. Echantillonnage et quantification.....	5
1.5.3. Le codage de pixel.....	5
1.5.4. La taille d'un images.....	5
1.5.5. Résolution d'une images.....	6
1.5.6. La luminance.....	6
1.5.7. Le bruit.....	6
1.5.8. Histogramme.....	7
1.6. Le codage des couleurs.....	7
1.7. Les différents formats d'images.....	10
1.8. La structure d'un fichier BMP.....	11
1.9. Quelques aspects du Traitement d'Image.....	13
1.9.1. Binarisation.....	13
1.9.2. Segmentation	14

1.9.3. Squelettisation	14
1.9.4. Convolution	15
1.9.5. Filtrage	15
1.9.5.1. Le filtre de Moyenne	16
1.9.5.2. Le filtre Médian.....	16
1.10. Morphologie mathématique	17
1.11. La détection des contours.....	17
1.12. Conclusion.....	19

Chapitre 2 Segmentation d'images

2.1. Introduction	20
2.2. Définition de la segmentation	20
2.3. Méthodes de la segmentation	21
2.3.1. Méthodes basées sur l'approches Région	22
2.3.1.1. Méthodes de type division-fusion "split-merge"	22
2.3.1.2. Méthodes de type croissance des régions.....	23
2.3.2. Méthodes basé sur l'approche Contours.....	24
2.3.3. Méthodes basé sur la classification.....	24
2.3.3.1 Méthode basé sur la classification supervisée.....	24
2.3.3.1.1. Méthodes bayésiennes.....	25
2.3.3.1.2. Réseaux de Neurones.....	25
2.3.3.1.3. Segmentation par les champs de Markov.....	27
2.3.3.2. Méthodes basé sur la classification non-Supervisées.....	27
2.3.4.2.1. L'algorithme K-Maens.....	28
2.3.4.2.2. L'algorithme Fuzzy C-Maens.....	29
2.3.4.2.3. L'algorithme Possibilités C-Maens	30
2.4. Exemples d'applications de la segmentation.....	31
2.5. Conclusion.....	32

Chapitre 3 La segmentation floue d'images

3.1. Introduction.....	33
3.2. La logique floue	34
3.3. La théorie des sous ensembles flous.....	34
3.4. K- Means.....	34
3.4.1. Principe et utilisation.....	35
3.4.2. Description de l'algorithme.....	35
3.4.3. Les principales étapes.....	36
3.4.4. Complexité	36
3.5. Fuzzy C-Means (FCM).....	36
3.5.1. Présentation de l'algorithme FCM	36
3.5.2. Principe et utilisation.....	36
3.5.3. Description de l'algorithme.....	37
3.5.4. Les principales étapes	37
3.5.5. Description de la méthode	37
3.6. Les inconvénients du FCM.....	40
3.7. Les variantes des C-moyennes floues.....	40
3.7.1. L'approche de Kamel et Selim.....	40
3.7.2. L'approche de Kehtarnavaz.....	41
3.7.3. Une approche due à Pedrycz : FCM conditionnelles.....	42
3.7.4. Les c-moyennes possibilistes.....	43
3.7.5. Extension aux contours courbés.....	44
3.7.6. Approximations.....	44
3.8. Les limite du FCM	45
3.9. Conclusion.....	45

Chapitre 4 Réalisation et évaluation des résultats

4.1. Introduction.....	46
4.2. Plateforme et outils de développement.....	46
4.2.1. Système d'exploitation.....	46
4.2.2. Utilisation de l'environnement de développement Visual C#.....	46
4.2.3. Langage de programmation.....	47
4.3. Présentation et implémentation du système	47
4.4. Evaluation	47
4.4.1. Critères de validité.....	48
4.4.1.1. Mesure de dissimilarité de LIU ET YANG	48
4.4.1.2. Critère de Borsotti.....	48
4.4.1.3. Contraste de Zeboudj.....	49
4.4.1.4. Critère d'uniformité de mesure.....	50
4.4.2. Les images utilisées.....	50
4.5. Présentation et Evaluation des résultats obtenus.....	51
4.5.1. L'algorithme FCM.....	51
4.5.2. L'algorithme TFCM.....	53
4.5.3. L'algorithme CTFCM.....	56
4.6. Quelque interface de l'application.....	60
4.7. Conclusion.....	62
Conclusion générale.....	63
Bibliographie.....	64

LISTE DES FIGUREURES

Figure.1.1. Représentation d'une image numérique dans le plan cartésien.....	2
Figure.1.2. Images vectorielles	3
Figure.1.3. Images matricielle.....	4
Figure.1.4. Image numérique.....	4
Figure.1.5. Représentation de Echantillonnage et quantification.....	5
Figure.1.6. La résolution d'une image.....	6
Figure.1.7. L'histogramme et la palette associés à l'image.....	7
Figure.1.8. Exemples de squelettes pour des formes simples.....	15
Figure.1.9. Mise en œuvre du filtre Médian.....	17
Figure.1.10. Détection des contours (Laplacien).....	18
Figure.1.11. Détection des contours (Sobel).....	18
Figure.2.1. Algorithme division/ fusion par les quadtree.....	23
Figure.4.1. L'interface de Microsoft Visuel Studio 2010.....	47
Figure.4.2. L' images standard	50
Figure.4.3. L'interface principale.....	60
Figure.4.4. Le menu de Fichier.....	61
Figure.4.5. L'interface select image file	61
Figure.4.6. La résultat de la segmentation.....	62

LISTE DES TABLEAUX

Table.1.1. L'information couleur de pixel est codée sur 3 octets.....	8
Table.1.2. Les principaux modes de codage.....	9
Table.1.3. Exemples de poids d'images avec différentes profondeurs de codage.....	10
Table.1.4. Entête d'un fichier BMP.....	12
Table.1.5. Entête de l'image BMP.....	12
Table.1.6. Les différents types de masques de convolution.....	19
Table.4.1. exemple du les résultats de segmentation d'image par FCM.....	52
Table.4.2. Résultats expérimentaux de segmentation d'image par FCM.....	53
Table.4.3. Exemple du les résultats de segmentation d'image par TFCM.....	55
Table.4.4. Résultats expérimentaux de segmentation d'image par TFCM.....	56
Table.4.5. Exemple du les résultats de segmentation d'image par CTFCM.....	58
Table.4.6. Résultats expérimentaux de segmentation d'image par CTFCM.....	59

INTRODUCTION GENERALE

Au cours de la dernière décennie, le domaine de traitement d'images s'est énormément développé et un grand nombre de travaux ont été effectués dans différents domaines d'applications tels que le domaine médical, la télédétection, etc.

Dans un système de traitement d'images, la segmentation d'images est l'opération la plus importante car elle conditionne la qualité de l'interprétation d'une image. Un bon résultat de segmentation ne permet pas forcément une bonne interprétation, mais nous ne pouvons pas obtenir une bonne interprétation à partir d'un mauvais résultat de segmentation.

Dans la méthode des C-moyennes dur (HCM), un élément de X est attribué à une classe et une seule parmi les C proposées. Dans une méthode comme HCM, les éléments sont classés de façon certaine comme appartenant à une classe et une seule. Quelle que soit la modalité d'imagerie, cette assertion ne reflète pas la réalité physique de l'échantillon étudié (bruit, volume partiel, hétérogénéité de champ (voir annexe)). Les méthodes de segmentation floue permettent d'obtenir une partition floue qui prend en compte ces aspects imprécis et incertains.

L'objectif de ce mémoire est de présenter un système de segmentation d'images en utilisant la logique floue. L'algorithme des c-moyens flous (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 par un processus itératif en minimisant une fonction objectif. Ainsi, il permet d'obtenir une partition floue de l'image en donnant à chaque pixel un degré d'appartenance à une région donnée.

Notre mémoire comporter les parties suivantes :

- _ *Le premier chapitre*, introduit des notions générales sur le traitement d'images.
- _ *Le second chapitre*, expose un état de l'art de la segmentation des images.
- _ *Le troisième chapitre*, décrit le principe de la segmentation flou et l'algorithme c-moyennes flou. Puis, il énonce les différentes variantes de cet algorithme.
- _ *Le quatrième chapitre*, a pour but de décrire les protocoles d'évaluation de la segmentation et les résultats que nous avons obtenus. Cette démarche va nous permettre d'évaluer les performances de nos méthodes adoptées.

CHAPITRE 1

TRAITEMENT D'IMAGES

1.1. Introduction

Aujourd'hui, l'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre. C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

On désigne par "technique de traitement d'images" toutes les techniques ayant pour but la modification des caractéristiques chromatiques des pixels des images bitmap.

Traitement d'images est souvent synonyme d'amélioration des images avec pour but d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes qu'on va utiliser dans différentes applications par exemple la reconnaissance, la segmentation.

Dans ce chapitre, on présente quelques principaux concepts de traitements d'images qui sont en relation avec notre sujet d'étude.

1.2. Définition de l'image

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc. C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain.

Une image est une fonction de deux variables, par exemple $I(x,y)$ représentera une fonction d'amplitude (l'intensité) de deux variables réelles de position (x,y) dans le plan cartésien.

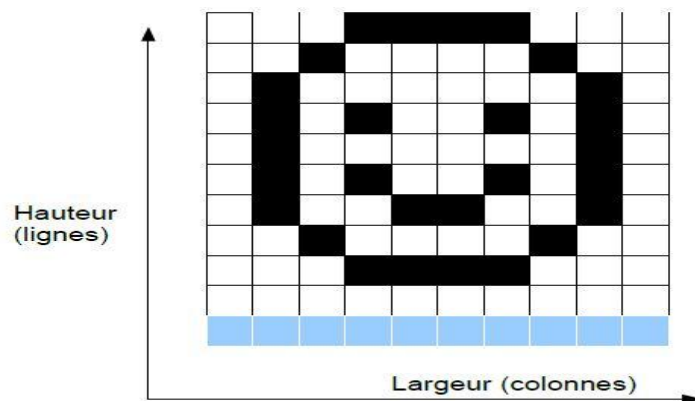


Figure1.1. représentation d'une image numérique dans le plan cartésien.

1.3. Image numérique [49]

Le terme d'image numérique désigne, dans son sens le plus général, toute image qui a été acquise, traitée et sauvegardée sous une forme codée représentable par des nombres (valeurs numériques).

La numérisation est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal qu'elle représente (une infinité de valeur dans l'intensité lumineuse par exemple), à l'état d'image numérique qui est caractérisée par l'aspect discret (l'intensité lumineuse ne peut prendre que des valeurs quantifiées en un nombre fini de points distincts). C'est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur.

1.4. Type des images

Il existe deux types d'images en informatique. Les images vectorielles et les images Bitmap.

1.4.1. Les images vectorielles

Les images vectorielles sont composées de formes géométriques qui vont pouvoir être décrites d'un point de vue mathématique. Par exemple une droite sera définie par 2 points, un cercle par un centre et un rayon. Le processeur est chargé de "traduire" ces formes en informations interprétables par la carte graphique (images Word, Publisher, CorelDraw-format WMF, CGM, etc.) (voir **Figure1.2**)



Figure1.2. images vectorielles

1.4.2. Une image matricielle (ou bitmap)

Une image matricielle (ou bitmap) est une image constituée d'un ensemble de points: les pixels. Chaque point porte des informations de position et de couleur.

Les photos numériques et les images scannées sont de ce type (voir Figure1.3).

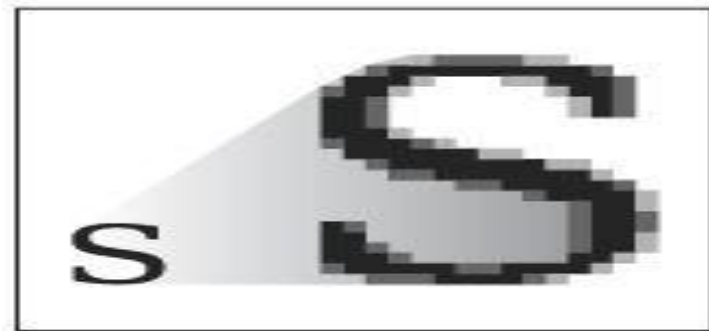


Figure1.3. images matricielle.

1.5. Concepts de base d'une image bitmap

1.5.1. Le pixel

Le pixel représente le plus petit élément constitutif d'une image matricielle. Le mot pixel provient d'une abréviation de l'expression britannique PICTUREELEMENT. La valeur numérique d'un pixel représente une intensité lumineuse.

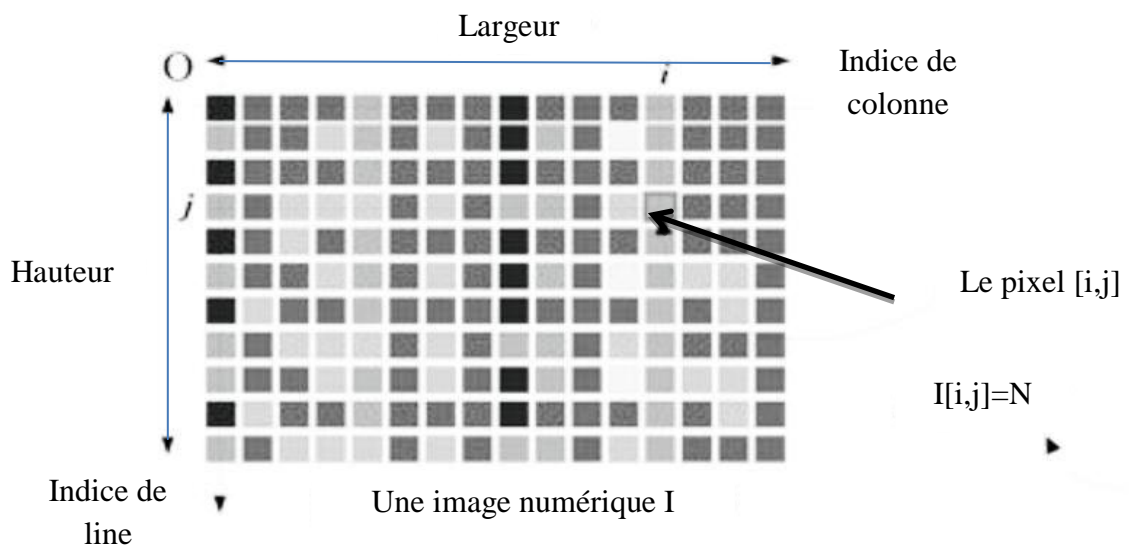


Figure1.4. Image numérique.

1.5.2. Echantillonnage et quantification [16]

L' **échantillonnage** est le procédé de discrétisation spatiale d'une image consistant à associer à chaque zone rectangulaire $R(x, y)$ d'une image continue une unique valeur $I(x, y)$. On parle de sous l'échantillonnage lorsque l'image est déjà discrétisée et qu'on diminue le nombre d'échantillons.

Une image numérique est une image échantillonnée et quantifiée. La **quantification** désigne la limitation du nombre de valeurs différentes que peut prendre $I(x, y)$.

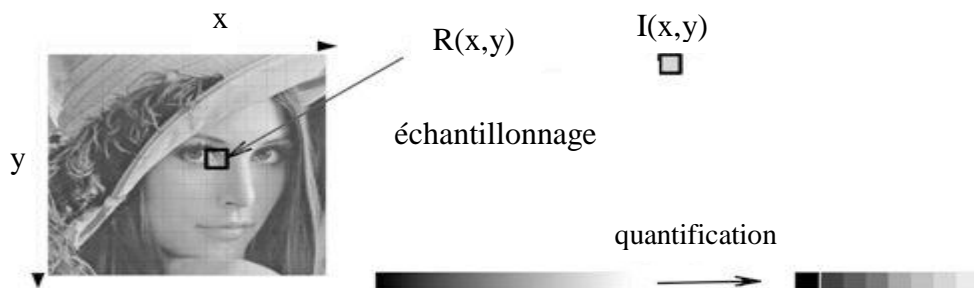


Figure1.5. représentation d'Echantillonnage et quantification

1.5.3. Le codage de pixel

Une image est divisée en points ou pixels. Considérons une image de 10 cm sur 10 cm avec une résolution très faible de 10 pixels par cm. Elle est codée sur $100 \times 100 = 10000$ pixels.

Avec une résolution de 100 pixels par cm (un pixel mesure 0,1 mm), elle est codée sur $1000 \times 1000 = 1000000$ pixels = 1 Million pixels.

Pratiquement toujours, la valeur d'un pixel est un mot binaire de longueur K bits, par conséquent un pixel peut prendre l'une des valeurs de l'intervalle $[0 \dots 2^K - 1]$. La valeur K est appelée profondeur de l'image.

L'ensemble de ces pixels est contenu dans un tableau à deux dimensions (une matrice) constituant l'image finalement obtenue.

1.5.4. La taille d'une image

La taille d'une image est le nombre de pixels de cette image, la taille d'une image qui est représenté par (328×456) dont 328 est le nombre de lignes, et 456 est le nombre de colonnes, est égale à: $328 \times 456 = 149568$ pixels.

1.5.5. Résolution d'une image:

La résolution d'une image est définie par le nombre de pixels par unité de longueur. Plus on a des points (ou pixels) par pouce (**ppp**) ou dots per inch (**dpi**) en anglais, meilleure est la qualité de l'images.

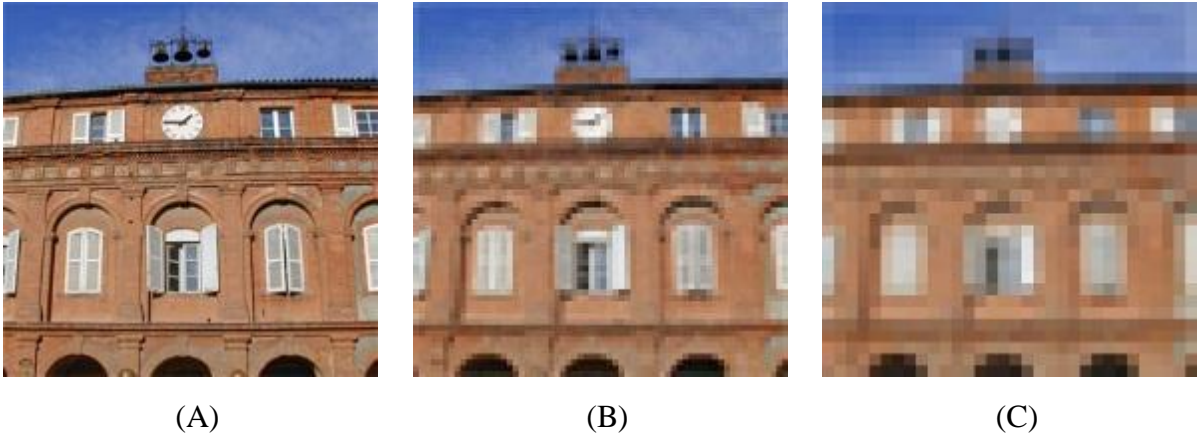


Figure1.6. La résolution d'une image : (A) image acquise à 256 dpi, (B) image acquise à 64 dpi, (C) image acquise à 32 dpi.

1.5.6 La luminance [53]

Le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. La luminance est le degré de luminosité des pixels de l'image. Elle est définie aussi comme l'intensité d'une source étendue dans une direction donnée, divisée par l'aire apparente de cette source dans cette même direction

1.5.7. Le bruit

Un système d'enregistrement d'images ne restitue pas l'image de manière parfaite. En effet des informations parasites s'ajoutent de manière aléatoire aux détails de la scène d'origine, c'est ce qu'on appelle: **bruit**.

Le bruit n'a pas les mêmes origines, mais provoque des effets assez semblables comme la perte de netteté sur les détails ou l'apparition de grains.

1.5.8. Histogramme [48]

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé au plus clair.

Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représenté par un graphique possédant 256 valeurs en abscisses, et le nombre de pixels de l'image en ordonnées. Prenons par exemple l'image suivante composée de niveaux de gris :

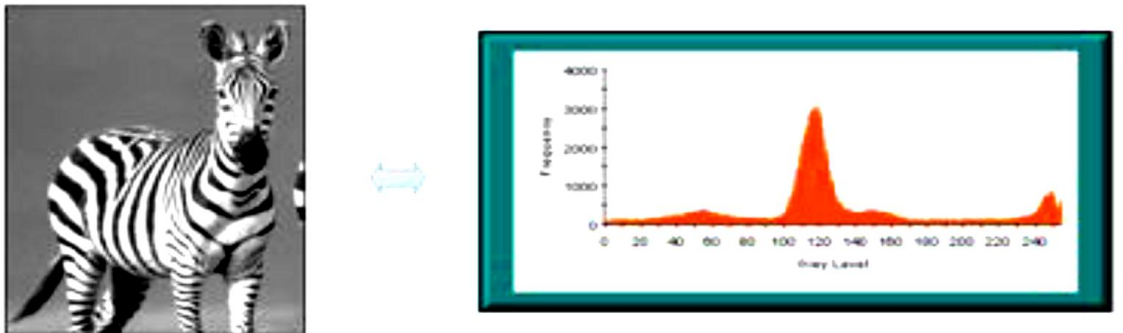


Figure 1.7. L'histogramme et la palette associés à cette image.

Pour les images en couleur plusieurs histogrammes sont nécessaires. Par exemple pour une image codée en RGB(Red Green Bleu):

- un histogramme représentant la distribution de la luminance,
- trois histogrammes représentant respectivement la distribution des valeurs respectives des composantes rouges, bleues et vertes.

1.6. Le codage des couleurs [17]

Une image est représentée par un tableau à deux dimensions dont chaque case est un pixel. Pour représenter informatiquement une image monocanal, il suffit donc de créer un tableau de pixels dont chaque case contient une valeur. La valeur stockée dans une case est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur). Alors la profondeur d'image représente le nombre de bits associés à chacun des pixels d'une image afin de coder les couleurs. Il existe plusieurs standards de codage de la profondeur. Cette information de couleur va dépendre du nombre de valeurs numérique que l'on veut attribuer à chaque pixel voir (**Table 1.1**).

- **Pour les images les plus simples (noir et blanc):** Le stockage d'un bit dans chaque cas permet de définir 2 couleurs (noir ou blanc).
- **Les images 16 couleurs ou 16 niveaux de gris.** En stockant 4 bits dans chaque case, il est possible de définir pour chaque pixel 16 dégradés de gris, du noir au blanc, ou bien 16 couleurs différentes.
- **Les images 256 couleurs ou 256 niveaux de gris.** En stockant un octet dans chaque case, il est possible de définir 256 dégradés de gris allant du noir au blanc ou bien 256 couleurs différentes.
- **Les images en couleurs:** Un pixel doit pouvoir prendre beaucoup plus de valeurs différentes. La valeur de chacun d'entre-deux est habituellement enregistrée sous forme de trois nombres binaires à 8 bits (d'une valeur de 0 à 255), correspondant respectivement à chacune des couleurs primaires: **Rouge, Vert et Bleu (RVB)**. Cela donne une gamme de $256 \times 256 \times 256$ couleurs possibles (soit 16777216 couleurs). On obtient une image de couleur dit à 24 bits ou encore en couleurs variés.


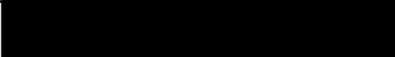




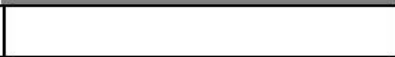
Valeur R	Valeur V	Valeur B	Couleur correspondante	Commentaire
0	0	0		noir
0	0	1		un peu moins noir (nuance impossible de détecter à l'oeil par rapport au noir)
...
0	0	255		bleu
...
0	255	0		vert
...
255	0	0		rouge
...
128	128	128		couleur intermédiaire correspondant à un gris
255	255	255		blanc

Table 1.1. L'information couleur de pixel est codée sur 3 octets. Ce qui entraîne des fichiers volumineux.

Il existe plusieurs modèles de codage des couleurs:

- Le modèle RGB (ou RVB): système additif. Le modèle additif se prête aux couleurs qui seront restituées sur des périphériques de sortie tels que les que les moniteurs, les téléviseurs, les projecteurs.
- Le modèle CMYK (Cyan, Magenta, Yellow, Black) ou CMJN (Cyan, Magenta, Jaune, Noire): système soustractif. Ce modèle est utilisé dans le domaine de l'impression.
- Le modèle HSL (Hue, Saturation, Luminance) ou TSL (Teinte, Saturation, Luminosité): Ce modèle se fonde sur la perception des couleurs par l'œil humain.

Type de codage	Nombre des bits par pixel	Valeurs possible	Interprétation de ces valeurs
Treil	1 bit	0 ou 1	0=noire, 1=blanc
Nuances de gris	8 bits	De 0 à 255	256 niveaux de gris, entre le noire (0) et le blanc (1)
Couleurs réelles (true color) RVB	24 bits (3×8)	3 valeurs de 0 à 255	Chaque valeurs représenter une intensité de rouge, de vert ou de bleu, 0=pas de couleur, 255=intensité maximale. On obtient ainsi 16 million de couleurs.
Couleurs indexées par une palette	8 bits	De 0 à 255	Une table appelée palette établit la correspondance entre ces 256 numéro de couleur et les couleurs réelles qu'ils représenter, codées dans le mode RVB

Table 1.2. Les principaux modes de codage.

Pour connaître le poids (en octets) d'une image, il est nécessaire de compter le nombre de pixels que contient l'image, cela revient à calculer le nombre de cases du tableau, soit la hauteur de celui-ci que multiplie sa largeur. Le poids de l'image est alors égal à son nombre de pixels que multiplie le poids de chacun de ces éléments [47].

Voici le calcul pour une image 640x480 en True color voire (**Table 1.3.**):

- Nombre de pixels: $640 \times 480 = 307200$
- Poids de chaque pixel: $24 \text{ bits} / 8 = 3 \text{ octets}$
- Le poids de l'image est ainsi égal à:

$$307200 * 3 = 921600 \text{ octets} \text{ -----} 921600 / 1024 = 900 \text{ Ko}$$

Définition de l'image	Noire et Blanc	256 couleurs (8 bits)	65000 couleurs (16 bits)	True color (24 bits)
320×200	7.8 Ko	62.5 Ko	125 Ko	187.5 Ko
640×480	37.5 Ko	300 Ko	600 Ko	900 Ko
320×200	58.6 Ko	468.7 Ko	987.5 Ko	1.4 Mo
320×200	96 Ko	768 Ko	1.5 Ko	2.3 Mo

Table 1.3. Exemples de poids d'images avec différentes profondeurs de codage.

1.7. Les différents formats d'images

Un format d'image est une représentation informatique de l'image, incluant des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler.

La plupart des formats sont composés d'un en-tête contenant des attributs (dimension de l'image, type de codage, etc.), suivi des données (l'image proprement dite).

La structuration des attributs et des données diffère d'un format à un autre. Il existe plusieurs formats d'images, nous allons mentionner quelques-uns:

1.7.1. Windows BitMaP (BMP)

Le format BMP est l'un des formats les plus simples. Il a été développé conjointement par Microsoft et IBM. Cette technologie a pour principal avantage la qualité des images fournies pas de compression (pas de perte de qualité). Cela fait de lui un format d'image très lourd, pas ou peu utilisé sur Internet.

1.7.2. Le JPEG (Joint Photographic Experts Group)

C'est le format le plus courant, on le rencontre sur Internet. Il occupe peu d'espace disque. C'est le format développé par les photographes pour transmettre des images de qualité photographique professionnelle. Il gère des millions de couleurs mais il ne possède pas de palette de couleurs associée et donc les couleurs peuvent être différentes sur des machines et des systèmes différents.

1.7.3. GIF (Graphical Interchange Format)

Ce format est l'autre standard d'Internet. Les fichiers GIF sont de petites tailles, ce qui est dû au fait que ces images ne peuvent enregistrer que 256 couleurs: le plus gros avantage du format est lié à son plus gros inconvénient. Le format GIF permet également la création d'animations et de détournement.

1.7.4. Le PNG (Portable Network Graphics)

Le format PNG reprend le principe de codage du format GIF mais n'est pas limité à 256 couleurs, et offre une compression généralement plus efficace. Il permet donc contrairement à GIF d'enregistrer des photographies sans perte de qualité, mais avec un gain d'espace de stockage moindre comparativement au format JPEG.

1.7.5. Le PSD (PhotoShop Document)

Le PSD est le format utilisé par le logiciel Adobe Photoshop. Sa particularité est qu'il utilise des calques qui, à la manière de feuilles transparentes superposées, permet de travailler sur l'une ou l'autre des couches sans apporter de modifications aux autres.

1.7.6. Le format d'image TIFF (Tagged Image File Format) [52]

Le TIF / TIFF est le format de fichier le plus couramment utilisé pour stocker des images, des photographies. TIF est plus couramment utilisé dans les environnements professionnels et pour l'impression commerciale. Le format TIF est le format le plus largement pris en charge sur toutes les plateformes. Il est le format standard pour les images de haute qualité. Bien que de grande taille, le format TIF est considéré comme étant le format le plus fiable pour des impressions de haute qualité comme pour le textile, les tissus, etc.

1.7.7. WMF (Window Meta File)

Le format Window Meta File est la version de Microsoft du format PICT de Macintosh et peut contenir des objets comme des lignes, des cercles et ainsi de suite. Les WMF ont un entête spécial défini par Aldus Corp. Pour permettre aux applications de lire, mettre à l'échelle et dessiner les fichiers WMF.

1.8. La structure d'un fichier BMP [1]

Le fichier se décompose en 4 parties dans l'ordre:

- L'entête du fichier (FileHeader).
- L'entête du Bitmap (BitMapHeader).
- Une palette de couleur (pour les images d'au plus 256 couleurs soit 8 bits par pixels).
- Les données du BitMap.

1.8.1. En-tête du fichier [30]

L'en-tête est constitué d'une série d'entiers, codés sur 16 ou 32 bits (respectivement 2 ou 4 octets). Ces entiers sont disposés dans un ordre précis:

Nom du champ	Longueur en octets	Signification
Identifiant	2 octets	Contient toujours l'octet 'B' suivi de l'octet 'M'.
FileSize	4 octets	Taille totale du fichier en octets.
Reserved	4 octets	Champ réservé, doit être égal à 0.
DataOffset	4 octets	Nombre d'octets séparé.

Tab.1.4. Entête d'un fichier BMP.

1.8.2. L'en-tête du Bitmap (BitmapHeader)

L'en-tête fournit des informations sur l'image, comme le montre de tableau suivant:

Nom du champ	Longueur en octets	Signification
HeaderSize	4 octets	La taille de l'entête de l'image en octets
Width	4 octets	Largeur de l'image en pixels
Height	4 octets	Hauteur de l'image en pixels
Planes	2 octets	Le nombre de plans. Cette valeur vaut toujours 1
BitsPerPixels	2 octets	Nombre de bits nécessaires pour représenter un pixel
Compression	4 octets	Type de compression
BitmapDataSize	4 octets	Taille en octets des données de l'image
HResolution	4 octets	Résolution horizontale de l'image en pixels par mètre
VResolution	4 octets	Résolution verticale de l'image en pixels par mètre
Colors	4 octets	Nombre de couleurs dans l'image
ImportantColors	4 octets	Nombre de couleurs importantes

Table.1.5. Entête de l'image BMP.

1.8.3. La palette [1]

Les couleurs sont codées sur 4 octets: Bleu, Vert, Rouge et alpha (sur 8 octets, ordre inversé par rapport au RGB). La palette n'est présente que pour les images d'au plus 256 couleurs. En gamme de gris, les valeurs sont de 00 00 00 00 à FF FF FF 00 (ou 256 valeurs, de 0 à 255, pour le RGB). La palette a donc 256 fois 4 octets soit 1024 octets.

1.8.4. Le BitMap [1]

L'image est codée de la dernière ligne à la première, de gauche à droite. La largeur de l'image en octets est toujours un multiple de 4. Si ce n'est pas le cas, la fin de la ligne est complétée avec de zéros. Les pixels sont stockés bits à bits: Pour un 256 couleur ou Niveau de gris, un octet décrit un pixel et un octet est représenté par un caractère ayant la valeur ascii du pixel.

1.8.5. Les données du BitMap

L'image est codée de la dernière ligne à la première, de gauche à droite. La largeur de l'image en octets est toujours un multiple de 4. Si ce n'est pas le cas, la fin de la ligne est complétée avec de zéros. Les pixels sont stockés bits à bits: Pour un 256 couleur ou grisés, un octet décrit un pixel et un octet est représenté par un caractère ayant la valeur Ascii du pixel.

1.9. Quelques aspects du traitement d'Image

1.9.1. Binarisation [44]

La binarisation (le seuillage) est la technique de classification la plus simple où les pixels de l'image sont partagés par un seul seuil s en deux classes: ceux qui appartiennent au fond et ceux qui appartiennent à la scène (l'objet). L'image est alors séparée en deux classes de façon à ce que l'information comprise entre 0 et s est retenue et l'autre non, ou vice-versa.

Soit l'image I ($M \times N$), supposons que $f(x, y)$ représente le niveau de gris du pixel aux coordonnées (x, y) , $0 \leq x \leq M$, $0 \leq y \leq N$ et s est le seuil choisi, les pixels de l'objet sont ceux ayant le niveau de gris inférieur à s et les autres ayant le niveau de gris supérieur à s sont des pixels du fond. Alors, l'image binarisée G est déterminée par les pixels (x, y) dont la valeur est:

$$g(x, y) = \begin{cases} 1 & \text{Si } f(x, y) > S \\ 0 & \text{Si } f(x, y) \leq S \end{cases} \quad (1.1)$$

Il existe trois grandes techniques de sélection du seuil s : global, local et dynamique. Comme il y a des différentes façons de déterminer le seuil s , il peut être considéré comme une fonction sous forme de $s = t(x, y, p(x, y), f(x, y))$ où $p(x, y)$ représente des propriétés locales du point (x, y) . Si s ne dépend que de la valeur $f(x, y)$ du point, le seuil est global, s'il dépend en plus de $p(x, y)$, s est un seuil local. Et si s dépend à la fois de (x, y) , de $p(x, y)$ et de $f(x, y)$, on dit le seuil dynamique ou bien adaptatif.

Dans la méthode de binarisation **globale** un seuil unique est calculé à partir d'une mesure globale sur toute l'image. Il nous permet de décider l'appartenance d'un pixel à l'objet ou au fond. Pour la binarisation **locale**, la classification d'un pixel dépend non seulement du pixel soi-même mais aussi de ses informations locales.

1.9.2. Segmentation [35]

La segmentation d'images ainsi définie est un domaine vaste où l'on retrouve de très nombreuses approches.

- La détection de frontières.
- La recherche de régions.
- L'approche de la classification.

Toutes ces approches visent à l'extraction des indices visuels. Après de nombreuses années passées à rechercher la méthode optimale, les chercheurs ont compris que la segmentation idéale n'existait pas. On peut même montrer que le problème de la segmentation est le plus souvent un problème mal posé. Etant donnée une image, il existe toujours plusieurs segmentations possibles. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation. Elle devra donc avoir simplifié l'image sans pour autant en avoir trop réduit le contenu.

1.9.3. Squelettisation [11]

La squelettisation donne une représentation compacte des objets, même dans le cas de formes et de topologies complexes. En dimension 2, les squelettes représentent des lignes inter connectées au centre d'un objet, comme le montre la figure (**Figure1.8.**).

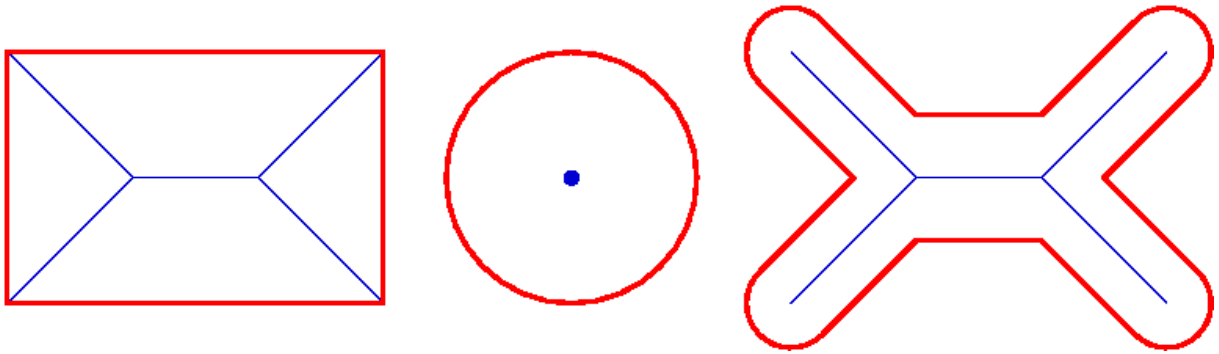


Figure 1.8. Exemples de squelettes pour des formes simples.

Ils sont très utilisés en analyse d'image et reconnaissance de forme, car ils permettent de décrire synthétiquement non seulement la forme, mais aussi certaines propriétés mathématiques des objets, comme par exemple la longueur ou la surface. Ils sont par ailleurs utilisés en compression d'image car ils décrivent les objets de façon compacte.

1.9.4. Convolution [41]

La convolution est le remplacement de la valeur d'un pixel par une combinaison de ses avoisinants. Elle consiste à faire balayer une fenêtre d'analyse (masque) sur l'ensemble des pixels de l'image. L'opération de convolution est calculée en tout point de l'image source en 3 étapes:

1. On centre le masque (par exemple un masque carré 3*3) sur le pixel courant.
2. On calcule les 9 produits entre la valeur de l'image et la valeur du masque superposé.
3. Puis on somme les 9 produits pour obtenir la valeur du pixel de l'image filtrée.

On peut résumer l'opération de convolution par la formule suivante:

$$S(i, j) = \sum_{k=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} \sum_{l=-\frac{(N-1)}{2}}^{\frac{(N-1)}{2}} E_i + k, j + h_{k,l} \quad (1.2)$$

Où E est l'image d'entrée, h le masque de convolution avec h (0, 0) le centre du masque et N la taille du filtre (dans notre exemple N=3).

1.9.5. Filtrage [54]

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine.

1.9.5.1. Le filtre de Moyenne

Le filtrage des bruits blancs conduit à des filtres de moyenne (ou de lissage) qui remplacent le pixel courant par la valeur moyenne des pixels sur une fenêtre 3x3 ou 5x5.

Ce filtre est bien compatible avec notre observation de départ selon laquelle le bruit est caractérisé par d'importantes hautes fréquences. Mais certaines fréquences (assimilées à de fortes variations dans les niveaux de gris) sont définitivement détruites, le lissage agissant comme un filtre passe-bas. L'image est non seulement lissée mais elle est aussi fortement altérée, les contours étant particulièrement dégradés par les filtres de moyenne.

Le filtre de moyenne est une version d'une gamme plus vaste de filtres de lissages dits linéaires.

Ici, le masque représentant ce filtre est le suivant:

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}$$

Ce filtre, normalisé par un facteur 1/9 pour le ramener dans les valeurs normales, génère un lissage fort. Un lissage moyen peut être réalisé par le biais du filtre suivant:

$$\begin{array}{ccc} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{array}$$

Le facteur de normalisation est sans surprise 1/16 et le coefficient plus élevé du pixel central permet d'atténuer l'effet de lissage.

1.9.5.2. Le filtre Médian

Les inconvénients du filtre de Moyenne induisent une orientation vers le filtre dit Médian, qui préserve les contours. Ce filtre vise à remplacer la valeur du pixel central par la valeur médiane de la répartition (on trie les luminances dans l'ordre croissant) des niveaux de gris des pixels situés à l'intérieur de cette fenêtre (figure 1.9)

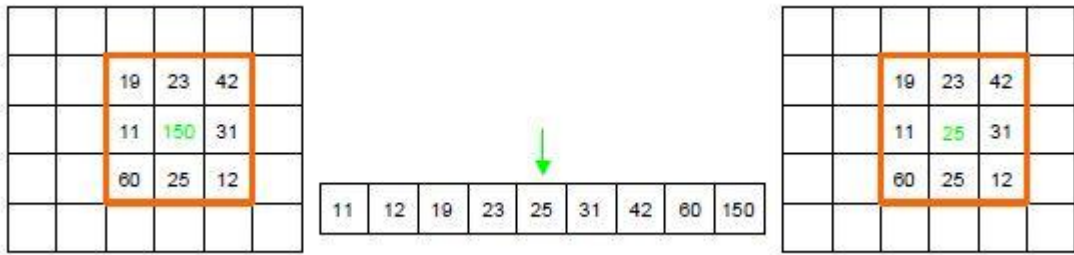


Figure1.9. Mise en œuvre du filtre Médian.

Ce filtre - non linéaire - permet d'éliminer des pixels isolés ou les traînées fines noyées dans des zones assez homogènes.

1.10. Morphologie mathématique [50]

La morphologie mathématique est une théorie d'analyse d'images née dans les années soixante lorsque Matheron et Serra effectuèrent des travaux sur les milieux poreux. A la base, cette technique s'appliquait à des images binaires, qui étaient considérées comme des ensembles.

Parmi les outils les plus importants de la morphologie mathématique on cite: érosion et dilatation.

- **Érosion:** qui consiste à enlever un pixel sur les pourtours de la forme, agrandir les trous.
- **Dilatation:** consiste à ajouter un pixel sur les pourtours de la forme, à boucher les trous.

1.11. La détection des contours [45]

Le but de la détection de contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse. Ces changements de propriétés de l'image traduisent en général des événements importants ou des changements dans les propriétés du monde. Ils incluent des discontinuités dans la profondeur, dans l'orientation d'une surface, dans les propriétés d'un matériau et dans l'éclairage d'une scène. La détection de contour est un champ de la recherche qui appartient au traitement d'image et à la vision par ordinateur, particulièrement dans le domaine de l'extraction de caractéristiques.

La détection des contours d'une image réduit de manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les

propriétés structurelles importantes de l'image. Il existe un grand nombre de méthodes de détection de l'image mais la plupart d'entre elles peuvent être regroupées en deux catégories. La première recherche les extremums de la dérivée première, en général les maximums locaux de l'intensité du gradient. La seconde recherche les annulations de la dérivée seconde, en général les annulations du laplacien ou d'une expression différentielle non-linéaire. Où laplacien est l'opérateur différentiel défini par l'application de l'opérateur gradient suivie de l'application de l'opérateur divergence.

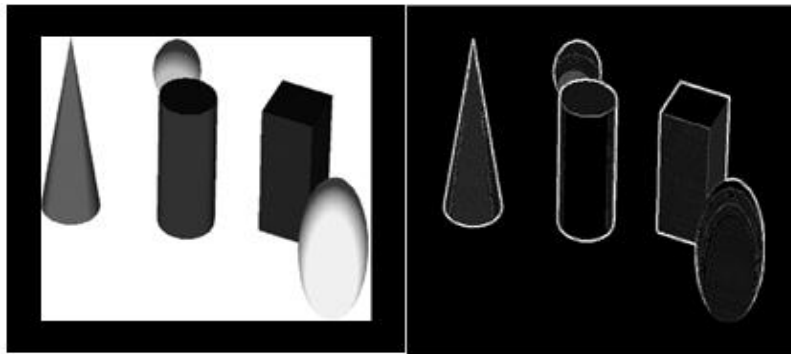


Figure1.10. détection des contours (Laplacien)

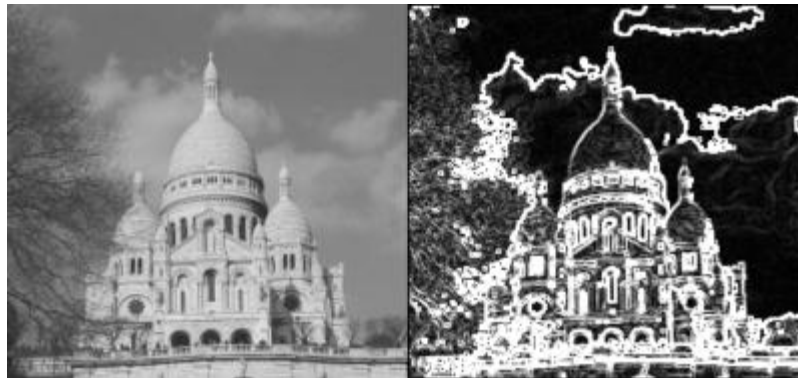


Figure1.11. détection des contours(Sobel).

Le tableau (**Table 1.6**) illustre les différents types de masques de convolution utilisables pour l'obtention du contour d'une image [41].

Type de masque	Gradient partiel	Extraction de l'amplitude	Extraction de la direction																								
Masque de Roberts <table border="1"> <tr><td colspan="2">Gradient selon x</td></tr> <tr><td>-1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table> <table border="1"> <tr><td colspan="2">Gradient selon y</td></tr> <tr><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	Gradient selon x		-1	0	0	1	Gradient selon y		0	-1	1	0	$G_x G_y$ Substitution du pixel supérieur gauche	$A = \sqrt{G_x^2 + G_y^2}$	$\theta = \frac{\pi}{4} + \arctan\left(\frac{G_x}{G_y}\right)$												
Gradient selon x																											
-1	0																										
0	1																										
Gradient selon y																											
0	-1																										
1	0																										
Masque de Sobel <table border="1"> <tr><td colspan="3">Gradient selon x</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>2</td><td>0</td><td>-2</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table> <table border="1"> <tr><td colspan="3">Gradient selon y</td></tr> <tr><td>1</td><td>2</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-2</td><td>-1</td></tr> </table>	Gradient selon x			1	0	-1	2	0	-2	1	0	-1	Gradient selon y			1	2	1	0	0	0	-1	-2	-1	$G_x G_y$	$A = \sqrt{G_x^2 + G_y^2}$	$\theta = \arctan\left(\frac{G_x}{G_y}\right)$
Gradient selon x																											
1	0	-1																									
2	0	-2																									
1	0	-1																									
Gradient selon y																											
1	2	1																									
0	0	0																									
-1	-2	-1																									
Masque de Prewitt <table border="1"> <tr><td colspan="3">Gradient selon x</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table> <table border="1"> <tr><td colspan="3">Gradient selon y</td></tr> <tr><td>1</td><td>1</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	Gradient selon x			1	0	-1	1	0	-1	1	0	-1	Gradient selon y			1	1	-1	0	0	0	-1	-1	-1	$G_x G_y$	$A = \sqrt{G_x^2 + G_y^2}$	$\theta = \arctan\left(\frac{G_x}{G_y}\right)$
Gradient selon x																											
1	0	-1																									
1	0	-1																									
1	0	-1																									
Gradient selon y																											
1	1	-1																									
0	0	0																									
-1	-1	-1																									

Table.1.6. les différents types de masques de convolution utilisée pour la détection du contour d'une image.

1.12. Conclusion

Le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dont le but est de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations utiles.

Dans ce chapitre nous avons présenté les différents concepts liés à l'image numérique, on a parlé sur les types d'une image, ces caractéristiques et quelques formats de fichiers d'images, on a expliqué aussi la structure d'un fichier bitmap et finalement nous avons présenté quelques traitements d'images usuels.

CHAPITRE 2

SEGMENTATION D'IMAGES

2. 1. Introduction

Une image est une représentation planaire d'une scène ou d'un objet situé en général dans un espace tridimensionnel. L'information élémentaire associée à chaque point de l'image est transcrite en niveau de gris ou en couleur. L'analyse pertinentes de l'image en regard de l'application concernée, les traiter et les interpréter. La segmentation est un traitement de bas-niveau qui consiste à créer une partition de l'image en sous-ensembles appelés régions. Une région est un ensemble connexe de points images (pixel) ayant des propriétés communes (intensité, texture, . . .) qui les différencient des pixels des régions voisines. Les connaissances utilisées sont le plus souvent des domaines de l'image numérique et du traitement du signal, donc sémantiquement assez pauvres. Il n'y a pas de méthode unique de segmentation d'une image, le choix d'une technique est lié à plusieurs facteurs. Du fait de cette diversité, il est difficile de définir de manière absolu une « bonne » segmentation.

La segmentation est une des étapes critiques de l'analyse l'images qui conditionne la qualité des mesures effectuées ultérieurement. Elle permet d'isoler dans l'image les objets sur lesquels doit porter analyse, de séparer, dans une image, les régions d'intérêt du fond.

2. 2. Définition de la segmentation

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image [46].

La segmentation est un processus de synthèse, visant à extraire les caractéristiques géométriques des images en faisant abstraction des nuances de couleur, des reflets, des transparences, des ombres, des textures et autres éléments qui constituent l'incroyable richesse de l'information visuelle [43] .

Formellement, Soit X le domaine de l'image et f la fonction qui associe à chaque pixel une valeur $f(x,y)$. 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 C sous-ensemble $\{R_1, \dots, R_c\}$ tels que:

1. $X = \sqcup_{i=1}^c R_i$
2. $\forall i \in \{1, \dots, c\}$ R_i est connexe
3. $\forall i \in \{1, \dots, c\}$ $P(R_i) = \text{vrai}$

4. $\forall i, j \in \{1, \dots, c\}^2$ R_i est adjacent à R_j et $i \neq j \Rightarrow p(R_i \cup R_j) = \text{faux}$ Où \sqcup représente une union d'ensemble disjoints.

Le prédicat P est utilisé pour tester l'homogénéité des ensembles R_i . Ces sous-ensembles constituent les régions de l'image. Une segmentation de l'image est donc sa décomposition en un ensemble de régions homogènes, le critère d'homogénéité P restant à déterminer. La première condition implique que tout pixel de l'image appartienne à une région et une seule. Cela signifie que l'algorithme de segmentation ne doit pas se terminer avant d'avoir traité tous les points. La seconde condition implique que toute région doit être connexe. La connexité des régions étant induite par le voisinage défini sur l'image. La troisième condition implique que chaque région doit être homogène. Enfin, la quatrième condition est une condition de maximalité indiquant que la fusion de deux régions ne doit pas être homogène. Il est important de remarquer que le nombre n de régions formant la partition de l'image reste indéterminé. Il peut donc exister plusieurs segmentations possibles pour un prédicat P donné.

2. 3. Méthodes de la segmentation

Après de nombreuses années passées à rechercher la méthode optimale, les chercheurs ont compris que la segmentation idéale n'existait pas. On peut même montrer que le problème de la segmentation est le plus souvent un problème mal posé. Etant donnée une image, il existe toujours plusieurs segmentations possibles. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation. Elle devra donc avoir simplifié l'image sans pour autant en avoir trop réduit le contenu.

La segmentation d'images ainsi définie est un domaine vaste où l'on retrouve de très nombreuses méthodes de segmentation, que l'on peut regrouper en trois principales classes:

- Segmentation fondée sur les régions (en anglais: region-based segmentation).
On y trouve par exemple: la croissance de région (en anglais: region-growing),
décomposition/fusion (en anglais: split and merge).
- Segmentation fondée sur les contours (en anglais: edge-based segmentation).
- Segmentation fondée sur classification ou le seuillage des pixels en fonction de leur intensité (en anglais: classification ou thresholding).

2. 3. 1. Méthodes basées sur l'approche Région

Les approches de segmentation régions visent à créer une partition de l'image en un ensemble de régions homogènes au sens d'un ou plusieurs critères. On cherche donc à rassembler les pixels par critères de similarité.

La segmentation d'une image vis à vis d'un critère d'homogénéité H (par exemple "tous les pixels sont du même niveau de gris" ou "la variation de niveau de gris n'excède pas n niveaux") peut s'exprimer de la manière suivante:

La segmentation d'une image I en regard du critère H est une partition de l'image I en n régions homogènes X_1, \dots, X_n telles que:

- $U_{i=1}^n X_i = I$.
- Pour tout i , X_i est connexe.
- Pour tout i , $H[X_i]$ est vrai.
- Pour tout couple (X_i, X_j) de régions voisines, $H[X_i, X_j]$ est faux.

Cette définition conduit à deux remarques très importantes. Tout d'abord, une segmentation dépend du critère employé. Le choix du critère est donc primordial. Ensuite, la décomposition obtenue n'est pas unique. Pour un critère donné, il existe plusieurs solutions. [35].

2. 3. 1. 1. Méthodes de type division-fusion "split-marge"

Le processus de segmentation est itératif et alterne deux phases: une phase de division de toutes les régions non homogènes et une phase de fusion de toutes les régions adjacentes de sorte que la région résultante respecte toujours le critère d'homogénéité. En d'autres termes, cette approche consiste à utiliser des propriétés globales pour diviser l'image en zones homogènes puis à fusionner certaines de ces régions grâce à des caractéristiques locales.

L'algorithme de division-fusion consiste, dans un premier temps, à découper l'image en zones de taille moyenne (le plus souvent en quatre quadrants), puis à parcourir chacune des régions en cherchant si elle vérifie un critère d'homogénéité. Les régions non homogènes sont à leurs tours divisées en sous-blocs. Ce processus de découpage de régions non homogène est pour suivit jusqu'à obtenir que des régions homogènes. Une étape de fusion

permet par la suite de regrouper les zones contigües correspondant à la même région. Si deux régions adjacentes peuvent être fusionnées pour former une zone homogène, elles le sont.

Ces méthodes font généralement appel à la théorie des graphes, ainsi elles peuvent être classées selon la structure du graphe utilisé. La figure (2. 1) illustre le principe de la segmentation basée sur les quadtree.

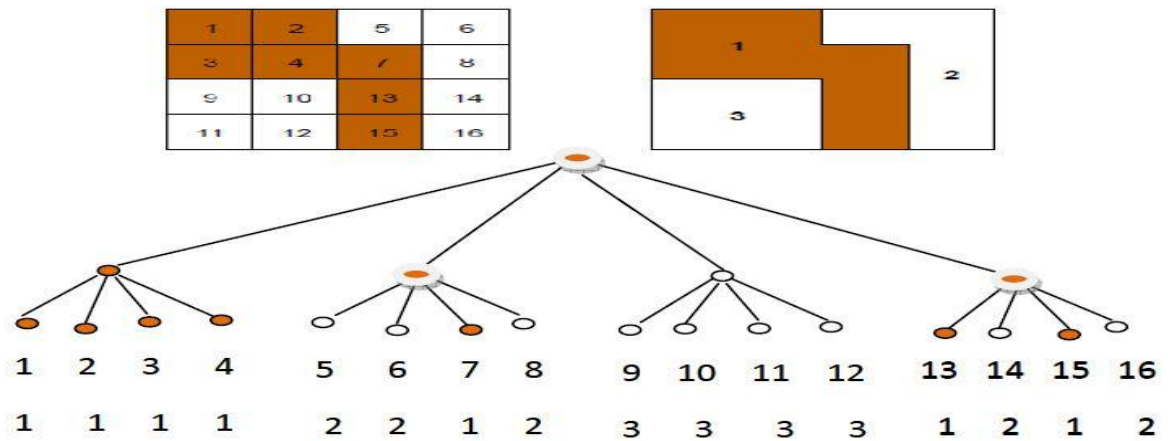


Figure 2. 1 Algorithme division/ fusion par les quadtree.

2. 3. 1. 2. Méthodes de type croissance des régions

Également appelées agrégation de pixels (les pixels forment la base du processus). Ces méthodes intègrent implicitement l'information spatiale dans le processus de segmentation. Les régions sont créées les unes après les autres (critère d'adjacence) avec pour chaque région, une phase d'initialisation et une phase itérative. La phase d'initialisation est la phase du choix d'un nouveau germe (point de départ d'une nouvelle région). La phase itérative est la phase d'agrégation des pixels voisins au germe, autrement dit, une région est construite à partir d'un germe par agrégation de pixels autour de ce germe selon un critère d'homogénéité. Ce germe doit être considéré comme une région, c'est-à-dire satisfaire au critère d'homogénéité (on peut choisir un pixel unique comme germe) [10], Un agrégat est un assemblage hétérogène de substances ou d'éléments qui adhèrent solidement entre eux.

Un prédicat classique d'homogénéité pour une région peut être par exemple la variance $\delta^2(R)$ des niveaux de gris de l'image associés aux points de cette région. Un point de l'image appartient à R si sa variance est inférieure à un seuil fixé. Le processus est initialisé par un germe choisi par seuillage sur un attribut simple (comme le niveau de gris). Il s'arrête lorsqu'il n'y a plus de région qui vérifie le prédicat.

La qualité de la segmentation peut également être un critère de segmentation. Si l'on possède une référence permettant de construire un indicateur global de la qualité de la segmentation, cette mesure peut servir localement pour savoir si une fusion l'accroît significativement [32].

Une telle mesure peut être la probabilité de bonne interprétation des régions et des contours. Il est également possible d'évaluer la qualité de fusion en considérant une région comme homogène si aucune fusion locale ne peut l'améliorer. Ces méthodes peuvent intégrer des informations locales (histogramme local pour être plus robuste à la présence du bruit). L'agrégation de pixels donne d'assez bons résultats même si la localisation des contours reste moins précise que celle obtenue par les techniques de détection de contours. Notons aussi que les seuils d'agrégation influent beaucoup sur la qualité du résultat.

2. 3. 2. Méthodes basé sur l'approche Contours [22]

Contrairement aux approches régions, qui cherchent à former des zones homogènes, les approches contours se basent sur les discontinuités des images pour déterminer les contours des régions. On peut distinguer les méthodes dérivatives et les méthodes fondées sur

Les modèles déformables. Les méthodes dérivatives permettent de détecter de manière simple les ruptures dans une image ou un volume. Les contours (ou surfaces en 3D) sont assimilés à des points de forts gradients ou de dérivées secondes nulles.

2. 3. 3. Méthodes basé sur la classification

Cette méthode consiste à regrouper et à classer les pixels d'une image en classes en fonction de leurs propriétés. A chaque point de l'image est associé un vecteur d'attributs. La classification est alors effectuée sur ces vecteurs d'attributs de façon à aboutir à un nombre restreint de régions homogènes au sein de l'image.

Donc la classification est définie comme une procédure dans laquelle les pixels similaire d'une image sont identifiés et regroupés dans une même classe. Il existe deux grandes tendances: *La classification non supervisée* et *La classification supervisée*.

2. 3. 3. 1. Méthodes basé sur la classification Supervisées

Les méthodes supervisées nécessitent un ensemble d'images ou de régions d'intérêt dans l'image déjà étiquetée. Soit, les données à analyser $X = \{X_j \in R^p, 1 \leq j \leq N\}$, où X_j est un vecteur forme représentant le pixel j . Cet ensemble de données est décomposé en deux sous-

ensembles: un ensemble d'apprentissage et un ensemble de test. L'ensemble d'apprentissage est construit par étiquetage manuel des images ou des régions d'intérêt en C classes de tissus (C_1, \dots, C_i) par un ou plusieurs experts (radiologues). Chaque classe C_i se voit donc affecter un ensemble d'apprentissage E_i , et les données de l'ensemble de test sont segmentées en fonction des E_j . Parmi ces méthodes on peut citer: la segmentation Bayésienne, la segmentation par les champs de Markov, réseaux de neurones, segmentation bayésiennes, etc.

a. Méthodes bayésiennes

La segmentation bayésienne consiste à calculer, pour chaque vecteur forme X_j , la probabilité conditionnelle $P(C_i|X_j)$ pour chacune des C classes (C_1, \dots, C_C) à l'aide de la règle Bayes:

$$\forall i \in \{1..C\} P(C_i|X_j) = \frac{P(C_i)P\left(\frac{X_j}{C_i}\right)}{\sum_{k=1}^C P(C_k) \cdot P\left(\frac{X_j}{C_k}\right)} \quad (2.1)$$

La segmentation bayésienne admet:

- Soit une approche paramétrique, dans laquelle l'intensité d'un voxel (est un point à trois dimensions) est considérée comme la combinaison linéaire de probabilités d'appartenance attachées à chaque class (en effectuant l'hypothèse que les données sont conformes à des distributions paramétrique). La probabilité conditionnelle $P(X_j/C_i)$
- Est modélisée par une fonction dépendant d'un vecteur de paramètre θ . Le problème est alors pour chaque classe C_i d'estimer le meilleur paramètre θ connaissant l'ensemble d'apprentissage E_i de cette classe.
- Soit une approche non paramétrique, dans ce type d'approche, les probabilités conditionnelles $P(X_j/C_i)$ sont supposées quelconques. La forme non paramétrique doit permettre de rendre compte le plus fidèlement possible de la réelle distribution statistique des niveaux de gris dans l'image.

b. Réseaux de Neurones [38]

Un réseau de neurones est un réseau d'unité élémentaire (les nœuds) interconnectés, à fonction d'activation linéaires ou non linéaire. Ces nœuds sont regroupés pour les réseaux multicouches en sortie et éventuellement un ensemble de neurones cachés. De nombreux modèles de réseaux existent (réseaux de Hopfield, perceptrons multicouche, etc.), les

différents nœuds étant complètement ou partiellement interconnectés aux autres. L'ensemble des liens convergeant vers un nœud constitue les connexions. Ceux qui divergent vers d'autres nœuds sont les connexions sortantes. A chaque connexion, entre des nœuds i et j , est associé un poids W_{ij} représentant la force de l'influence du nœud i sur le nœud j . L'ensemble des poids est regroupé dans un vecteur de poids synaptiques W . Un vecteur de scalaires a présenté à tous les nœuds d'entrée est appelé exemple. A cet exemple sont aussi associées les valeurs Y (le vecteur de sortie) que l'on désire apprendre. Les poids des connexions sont éventuellement modifiés au cours d'un cycle d'apprentissage.

Modifier la sortie des nœuds à partir de leurs entrées consiste tout d'abord à calculer l'activation présente à l'entrée du nœud puis à calculer la sortie du nœud suivant la fonction d'activation qu'elle possède. Un réseau de neurones peut ainsi être défini pour chaque nœud par quatre éléments:

- La nature de ses entrées, qui peuvent être binaires ou réelles.
- La fonction d'entrée totale e , qui définit le pré-traitement $e(a)$ effectué sur les entrées. Généralement, e est une combinaison linéaire des entrées pondérées par les poids synaptiques des connexions entrées pondérées par les poids synaptiques des connexions entrantes.
- La fonction d'activation f du nœud qui définit son état de sortie en fonction de la valeur de e . Toute fonction croissante et impaire convient et la fonction sigmoïde est souvent utilisée. La valeur de f en $e(a)$ est redirigée vers l'extérieure ou vers d'autres nœuds ou elle contribue à calculer leur état d'activation.
- La nature de ses sorties, qui peuvent être binaires ou réelles.

Les réseaux de neurones sont également utilisés pour obtenir une classification de l'image en tissus cérébraux. Ils sont organisés autour d'un ensemble de cellules (ou neurones) interconnectées. On dispose d'une base de connaissances constituée de couples (entrées, Sorties) et on utilise cette base pour entraîner une « mémoire » informatique à raisonner en prenant comme référence cette base empirique. Ainsi, en fonction de la base d'apprentissage, le réseau de neurones détermine la sortie pour chaque nouvelle entrée ; une mesure d'erreur est calculée pour chaque sortie obtenue, on cherche donc la meilleure valeur de sortie (c'est-à-dire minimiser l'erreur). Ici encore, la base d'entraînement est d'images segmentées par un expert, ou d'un échantillon de l'image à segmenter.

La méthode de segmentation supervisée offre l'avantage d'être plus rapides et plus reproductibles que les méthodes manuelles. Toutefois, elles ont le désavantage de rester très dépendantes de la base d'apprentissage. C'est pourquoi il est intéressant de développer des algorithmes entièrement automatiques «non supervisée », qui présenteront l'avantage d'offrir un résultat reproductible et indépendant des actions de l'opérateur.

c. segmentation par les champs de Markov [4]

Les champs de Markov sont largement utilisés en analyse d'images. Nous les avons utilisés dans la troisième étape pour régulariser notre image par l'introduction de l'information contextuelle. Les champs de Markov permettent de modéliser les propriétés globales en utilisant des contraintes locales.

Nous considérons une image (ou un volume) comme un ensemble S de n sites. L'ensemble S est défini par: $S = \{S_i, 1 \leq i \leq n\}$.

À chaque site S est associée une variable aléatoire X_s . L'image est alors considérée comme la réalisation d'un vecteur aléatoire $X = \{ X_s, s \in S \}$, appelé champ des observables. La classification markovienne a alors pour objectif d'estimer le champ des étiquettes $Y = \{ Y_s, s \in S \}$.

Les champs X et Y sont respectivement associés aux réalisations x et y définies par:

$$x = \{x_s, s \in S\} \quad \text{et} \quad y = \{y_s, s \in S\} \tag{2.2}$$

Si Y est un processus aléatoire sur un ensemble S , alors on dira que Y est un champ de Markov relativement au système de voisinage V si et seulement si: la probabilité conditionnelle locale en un site n n'est fonction que de la configuration du voisinage du site considéré.

Ce qui s'exprime de la façon suivante:

$$P(Y_s = y_s) > 0, \forall s \in S \tag{2.3}$$

$$P(Y_s = y_s | Y_t = y_t, t \neq s) = P(Y_s = y_s | Y_t = y_t, t \in V_s) \tag{2.4}$$

2. 3. 3. 2. Méthodes basé sur la classification non-Supervisées

L'intérêt des méthodes non supervisées est qu'elles ne nécessitent aucune base d'apprentissage et par là même aucune tâche préalable d'étiquetage manuel n'est requise.

Les algorithmes non supervisés les plus répandus tendent à minimiser une fonction coût, dépendant de la distance de chaque pixel aux prototypes (ou noyaux) des classes. Le prototype d'une classe étant un point connu dont l'appartenance à la classe est garantie et où chaque pixel est assigné à la classe qui lui est la plus proche. Selon la certitude de la classification que nous voulons obtenir, et la relation entre les classes, nous pourrions distinguer plusieurs méthodes de classification.

a. L'algorithme K-Means

L'algorithme k-means est l'algorithme de clustering le plus connu et le plus utilisé, du fait de sa simplicité de mise en œuvre. Il partitionne les données d'une image en K clusters. Contrairement à d'autres méthodes dites hiérarchiques, qui créent une structure en « arbre de clusters » pour décrire les groupements, k-means ne crée qu'un seul niveau de clusters. L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde[3].

Soit $X = \{x_k / k=1, \dots, n\}$ un sous-ensemble d'un espace n-dimensionnel. Une partition de X en C classes peut être représentée par des sous-groupes mutuellement disjoints $C_1 \dots C_c$, tel que $C_1 \cup \dots \cup C_c = X$.

Chaque vecteur x_k est assigné à une et seulement une classe qui lui est la plus proche.

La fonction objective correspondante s'exprime sous la forme:

$$J = \sum_{i=1}^c \sum_{k=1}^n U_{ik} d^2(x_k, v_i) \tag{2.5}$$

Où:

- v_i désigne le prototype (centroïde) de cette classe.
- d mesure la similarité entre le prototype et le vecteur de données.
- U_{ik} est un indicateur binaire $\in \{0, 1\}$ désignant l'appartenance ou non du vecteur x_k à la classe C_i ($U_{ik} = 1$ si $x_k \in C_i$ et $U_{ik} = 0$ sinon, $\forall i=1 \dots C$ et $x_k \in X$).

L'indicateur d'appartenance U_{ik} s'exprime comme suit:

$$U(x_k, v_i) = \{1 \text{ si } d^2(x_k, v_i) = \min C(x_k, v_c) \text{ ou } 0 \text{ sinon}\}.$$

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

- 1- Initialiser l'ensemble des centroides avec des éléments, soit pris au hasard de l'ensemble de données, soit d'après des connaissances a priori.
- 2- Assigner chaque point au centroïde le plus proche.
- 3- Calculer les nouveaux centroides des classes construites. Les centroïdes étant les centres des classes, 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 centroïde le plus proche.
- 5- S'il y a changement dans la partition, réitérer depuis l'étape 3 [36].

b. L'algorithme Fuzzy C-Means [5]

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 de 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 B 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 $U_{C \times N} [U_{ij}]$ est appelée matrice de C-partitions floues si et seulement si elle satisfait aux conditions:

$$\forall i \in \{1..C\}, \forall j \in \{1 \dots N\} \left\{ \begin{array}{l} u_{ij} \in [0,1] \\ 0 < \sum_{j=1}^N u_{ij} < N \end{array} \right. \quad (2.6)$$

$$\forall j \in \{1 \dots N\} \sum_{i=1}^c u_{ij} = 1 \quad (2.7)$$

La fonctionnelle à minimiser (2.8), et les solutions (2.9), (2.10), au problème du FCM sont décrites par les formules suivantes:

$$J(B, U, X) = \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^m d^2(x_j, b_i) \quad (2.8)$$

$$b_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \quad (2.9)$$

$$u_{ij} = \left[\sum_{k=1}^c \left(\frac{d^2(x_j, b_i)}{d^2(x_j, b_k)} \right)^{\frac{2}{m-1}} \right]^{-1} \quad (2.10)$$

c. L'algorithme Possibilités C-Maens [5][4]

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, qui est dû à la contrainte probabiliste que la somme de tous les degrés d'appartenance d'un objet aux différentes classes soit égale à 1. Une condition nécessaire pour passer à des degrés d'appartenance "absolus" est donc de lever cette contrainte, ou, plutôt, de l'assouplir afin que les degrés d'appartenance ne soient plus dépendants les uns des autres. Cependant, bien qu'incarnant cette notion de typicalité, l'algorithme PCM présente des performances plutôt médiocres comparées à celles de l'algorithme FCM. Dans le cas général, l'algorithme PCM a tendance à produire des classes concentriques, i. e les classes obtenues ont des centres très proches, voire confondus.

L'algorithme PCM

L'algorithme PCM permet de partitionner un ensemble $X = \{x_1; x_2, K, x_n\} \subset R^p$ en C classes, en minimisant la fonction objective:

$$J(B, U, X) = \sum_{i=1}^c \sum_{j=1}^N (u_{ij})^m d^2(x_j, b_i) + \sum_{i=1}^c \eta_i \sum_{k=1}^N (1 - u_{ik})^m \quad (2.11)$$

$$u_{ij} \in [0,1], 1 \leq i \leq c, 1 \leq k \leq N, \quad (2.12)$$

$$0 < \sum_{k=1}^N u_{ik} < N, 1 \leq i \leq c \quad (2.13)$$

- U_{ik} : C'est la distance euclidienne entre l'élément k et le centre de la classe i.
- $V = \{V_1, K, V_c\}$ est l'ensemble des prototypes des classes C_k dont la distance aux x_i est notée d_{ik} .
- $\eta = \{\eta_1, K, \eta_c\}$ est l'ensemble des termes de pénalité des données atypiques associés à chacune des classes. w_i Est à déterminer par l'utilisateur, et représente une évaluation de l'extension de la classe k. La minimisation de $J(U, V, \eta)$ consiste à minimiser pour toutes les classes:

$$J(B, U, \eta) = (u_{ij})^m d^2(x_j, b_i) + \eta_i(1 - u_{ik})^m \quad (2.14)$$

En dérivant la formule (1) par rapport à U_{ik} et en annulant cette dérivée on obtient:

$$u_{ij} = \left[1 + \left(\frac{d_{ik}^2}{\eta_i} \right)^{\frac{1}{m-1}} \right]^{-1} \quad (2.15)$$

Où le paramètre w_i représente le diamètre de la classe et peut être défini a priori par initialisation. Les points aberrants (les points très éloignés de toutes les classes) se voient alors dotés de petits poids pour toutes les classes et dont l'influence dans la phase de classification est réduite. Le PCM souffre, cependant, du problème de coïncidence de classes. On peut avoir, dans certains cas, des classes confondues tandis que des sous-groupes de données normaux fonction objectif est obtenue quand toutes les classes sont coïncidentes.

Des résultats plus ou moins satisfaisants ont été obtenus avec le PCM parce que l'optimisation mène à un minimum local mais pas à un minimum global. Cette propriété n'est pas toujours satisfaite, d'un point de vue théorique.

2. 4. Exemples d'applications de la segmentation

La segmentation est utilisée dans de nombreuses applications dont certaines concernent:

- La reconnaissance de la parole: Pal et Majumder [12] ont été les premiers à introduire le concept de sous-ensemble flou dans ce domaine. Ils décrivent des techniques permettant la discrimination de voyelles et de différentes voix d'orateurs appliquées à la langue Telugu (une des nombreuses langues en usage en Inde).
- La reconnaissance des caractères écrits peut avoir de nombreuses applications, telles que, par exemple, le traitement automatique des documents.
- Le traitement des eaux usées.
- La segmentation d'images RMN (Résonance Magnétique Nucléaire).

2. 5. Conclusion

Dans ce chapitre nous nous intéressons aux techniques de la classification automatique, ou segmentation (clustering). Nous avons vu que la segmentation (clustering) permet de regrouper des objets (individus ou variables) en un nombre limité de groupe ou de classes (segmentes, ou cluster).

La segmentation d'images est l'un des problèmes phares du traitement d'images. Elle consiste à partitionner l'image en un ensemble de régions connexes. L'intérêt de ces régions est de pouvoir être manipulées ensuite via des traitements de haut niveau pour extraire des caractéristiques de forme, de position, de taille, etc.

Le problème est évidemment très mal posé, car on ne sait jamais dire quelle est la segmentation idéale. L'idée est bien sûr que la région se rapproche de la notion d'objet, au sens courant du terme. Néanmoins, on peut dégager des propriétés plus raisonnables qu'on cherche à obtenir dans un algorithme de segmentation, en particulier:

- Stabilité: la segmentation obtenue ne doit pas varier beaucoup lorsque les conditions d'acquisition varie légèrement (bruit, illumination, point de vue, . . .).
- Régularité: les régions obtenues doivent être simples à manipuler (taille suffisante, forme régulière, . . .).

CHAPITRE 3

LA SEGMENTATION FLOUE D'IMAGES

3.1. Introduction

La segmentation est une tâche très importante dans le processus de traitement d'images. C'est une classification de pixels en des clusters homogènes et bien séparés. Une image est une projection d'une réalité tridimensionnelle sur un plan. Le traitement d'image est né du désir de vouloir déléguer l'observation humaine aux machines. Il a considérablement évolué avec la prise en compte des images non visibles, pour un observateur humain normal, telles que les images acoustiques, ultrasonores et les images radar. Le traitement d'image se résume à l'analyse d'image. Celle-ci consiste à extraire d'une image numérique des informations pertinentes.

Les techniques de segmentation cherchent à associer chaque pixel d'une image à une des primitives recherchées dans cette image, en fonction de certains critères. Dans certains cas, cette décision peut s'avérer difficile. Les résultats de logique floue permettent alors d'aborder le problème avec plus de souplesse.

Fuzzy C-Means (FCM) est un algorithme de classification non-supervisée floue. Issu de l'algorithme des C-moyennes (C-means), il introduit la notion d'ensemble flou dans la définition des classes : chaque point dans l'ensemble des données appartient à chaque cluster avec un certain degré, et tous les clusters sont caractérisés par leur centre de gravité.

Comme les autres algorithmes de classification non supervisée, il utilise un critère de minimisation des distances intra-classe et de maximisation des distances inter-classe, mais en donnant un certain degré d'appartenance à chaque classe pour chaque pixel. Cet algorithme nécessite la connaissance préalable du nombre de clusters 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 une classe donnée. Le cluster auquel est associé un pixel est celui dont le degré d'appartenance sera le plus élevé.

3.2. La logique floue

Dans la vie de tous les jours, nous nous trouvons dans des situations où les informations dont nous disposons ne sont pas toujours précises. Dans le domaine de la science aussi, des imprécisions peuvent être vues ; par exemple lorsqu'on veut établir une valeur qui caractérise le degré de similarité de deux formes.

L'être humain est habitué à utiliser des informations entachées d'incertitudes et d'imprécision dans la vie de tous les jours, il utilise ces informations incomplètes, raisonne avec elles et prend des décisions. Dans le domaine scientifique, il a été nécessaire de créer une logique qui admette des valeurs de vérité en dehors de l'ensemble {vrai, faux} pour pouvoir tenir compte et manipuler ce genre d'information incomplète.

Lukasiewicz propose en 1920 une logique ayant les trois valeurs de vérité suivantes : "vrai", "faux" et "doute". Ces valeurs ont été représentées par l'ensemble {0, 1, 0.5}, et ont été ensuite étendues à l'intervalle [0,1].

Néanmoins c'est Zadeh [14], qui à partir de l'idée d'appartenance partielle d'un élément à plusieurs classes, a formellement introduit la logique floue.

3.3. La théorie des sous ensembles flous

La théorie des sous ensembles flous et les outils de raisonnement qui en découlent, proposent un cadre formel qui permet de modéliser le langage naturel et de gérer l'imprécis et l'incertain. Cette théorie est basée sur la notion de degré d'appartenance. Ces degrés sont des valeurs qui expriment l'appartenance incertaine d'un pixel à une région. 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 de degrés d'appartenance u_{ij} du vecteur x_j à la classe i . [36]

3.4. K- Means

La méthode des k-means est un outil de classification classique qui permet de répartir un ensemble de données en k classes homogènes. La plupart des images (photos, dessins vectoriels 2D, synthèses 3D, ...) vérifient localement des propriétés d'homogénéité,

notamment en terme d'intensité lumineuse. L'algorithme des k-means permet donc d'apporter une solution à la segmentation d'images. [51]

L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde.

Le k-means est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centroïdes conditionne le résultat final, de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme. K-means change les objets de cluster jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur K du nombre de clusters. [3]

3.4.1. Principe et utilisation

Probablement l'algorithme de clustering le plus répandu. Il se base sur une initialisation préalable des centres de chaque cluster puis les affine au fil d'itérations. L'initialisation est une étape particulièrement importante car tout cluster ignoré par l'initialisation sera absorbé par le cluster le plus proche. Cet algorithme est simple à mettre en œuvre et présente une faible complexité en termes de calcul.

3.4.2. Description de l'algorithme

Le déroulement de l'algorithme est le suivant :

- Association de chaque vecteur au centre du cluster le plus proche
- Calcul des nouvelles coordonnées du centre par rapport aux vecteurs qui y sont associés

Les itérations se poursuivent jusqu'à ce qu'on atteigne une condition terminale, qui peut être un nombre fixe "i" d'itérations, descente du nombre de vecteurs assignés à un autre cluster sous un nombre minimal ou encore des conditions sur l'erreur quadratique entre un centre et les vecteurs qui y sont associées (oscillations, arrivée à un minimum, ...). [33]

3.4.3. Les principales étapes

Les principales étapes de l'algorithme k-means sont :

1. Choix aléatoire de la position initiale des K clusters.
2. (Ré-)Affecter les objets à un cluster suivant un critère de minimisation des distances (généralement selon une mesure de distance euclidienne).
3. Une fois tous les objets placés, recalculer les K centroïdes.
4. Répéter les étapes 2 et 3 jusqu'à ce que plus aucune ré-affectation ne soit faite. [3]

3.4.4. Complexité

Le nombre de calculs de distance effectués ici est très simple puisqu'à chaque itération on va calculer la distance de chaque point à chaque centre [33]

3.5. Fuzzy C-Means (FCM)

3.5.1. Présentation de l'algorithme FCM

Cet algorithme est une extension de l'algorithme de K-Mean en introduisant une notion de floue dans la définition du degré d'appartenance. Son principe est de regrouper des individus dans k classes qui soient le plus homogènes et naturelles possibles. Les groupes obtenus doivent contenir des pixels les plus semblables, et entre groupes différents les individus doivent être le plus différents possible. En fait, 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 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 degré d'appartenance. Le but des algorithmes de classification est non seulement de calculer les centres de classe B mais aussi l'ensemble des degrés d'appartenances des vecteurs aux classes. [40]

3.5.2. Principe et utilisation

Evolution du K-Means, il fait intervenir l'appartenance floue des points à un cluster. La complexité de l'algorithme reste plus faible mais il permet un clustering plus fin puisqu'on détermine pour chaque point un degré d'appartenance aux différents clusters. Cette information plus nuancée que l'appartenance définitive à une classe peut se révéler utile pour

des traitements ultérieurs de l'information. Cet algorithme garde, par contre, une forte sensibilité à l'initialisation tout comme le K-Means.

3.5.3. Description de l'algorithme

La phase d'initialisation est identique, mais va s'accompagner de l'initialisation de valeurs d'appartenance u_{ij} du vecteur x_i au cluster de centre c_j . Ces valeurs sont comprises entre 0 et 1.

Après cette phase d'initialisation, la résolution du problème de clustering se fait par la minimisation de la fonction objective définie par (3.1), m étant un paramètre constant supérieur à 1 (généralement pris égal à 2). [33]

$$j \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i) \quad (3.1)$$

3.5.4. Les principales étapes [3]

Les principales étapes de l'algorithme Fuzzy C-means sont :

1. La fixation arbitraire d'une matrice d'appartenance.
2. Le calcul des centroïdes des classes.
3. Le réajustement de la matrice d'appartenance suivant la position des centroïdes.
4. Calcul du critère de minimisation et retour à l'étape 2 s'il y a non convergence de critère.

3.5.5. Description de la méthode

On considère l'espace de n points de dimension p suivant :

$$X: \begin{bmatrix} x_1^1 & \dots & x_1^j & \dots & x_1^p \\ \dots & \dots & \dots & \dots & \dots \\ x_i^1 & \dots & x_i^j & \dots & x_i^p \\ \dots & \dots & \dots & \dots & \dots \\ x_n^1 & \dots & x_n^j & \dots & x_n^p \end{bmatrix} \quad (3.2)$$

- On suppose que les n points peuvent être groupés en c clusters $c < n$
- Les clusters sont décrits par leurs centres :

$$V = [v_i^1, v_i^2, \dots, v_i^j, \dots, v_i^p], 1 \leq i \leq c \quad (3.3)$$

On considère la matrice de proximité suivante : u_{11}

$$U = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ u_{i1} & \dots & u_{ik} & \dots & u_{in} \\ \dots & \dots & \dots & \dots & \dots \\ u_{c1} & \dots & u_{ck} & \dots & u_{cn} \end{bmatrix}, \text{ avec } k = 1, \dots, n \text{ et } i = 1, \dots, c. \quad (3.4)$$

u_{ik} représente le degré d'appartenance du point X_k au centre V_i

$$u_{ik} = \left[\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}} \right]^{-1} \quad (3.5)$$

d_{ik} représente la distance entre V_i et V_k

$$u_{ik} = \left[\sum_{j=1}^c \left(\frac{\|X_k - V_i\|}{\|X_k - V_j\|} \right)^{\frac{2}{m-1}} \right]^{-1} \quad (3.6)$$

,si on choisit la distance euclidienne [42]

L'algorithme utilise dans notre approche [38]

Etape 1 : fixer les paramètres.

Les entrées : $X = (x_j, j = 1..N)$ l'ensemble des vecteurs forme. C : nombre de classes.

ε : Seuil représentant l'erreur de convergence (par exemple $\varepsilon = 0.001$)

m : degré de flou, $m \in (1.5,3)$

Etape2 : initialiser la matrice degrés d'appartenance U par valeur aléatoire dans l'intervalle $[0,1]$

Etape 3 : mettre à jour la matrice prototype B par la relation.

$$b_i \leftarrow \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \quad (3.7)$$

$$j^{ancien} \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i) \quad (3.8)$$

Etape 4 : mettre à jour la matrice des degrés d'appartenance par la relation

$$u_{ij} \leftarrow \left[\sum_{k=1}^c \left(\frac{d^2(x_j, b_i)}{d^2(x_j, b_k)} \right)^{2/(m-1)} \right]^{-1} \quad (3.9)$$

$$j^{nouveau} \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i)$$

Etape 5 : répéter les étapes 3 a 4 jusqu'à satisfaction du critère d'arrêt qui s'écrit :

$$\|J^{ancien} - J^{nouveau}\| \leq \varepsilon \quad (3.10)$$

Les sorties : la matrice d'appartenance U et les centres des classe B .

3.6. Les inconvénients du FCM [55]

L'algorithme floue c-moyens possède plusieurs inconvénients qui affectent à la performance de cet algorithme

- la spécification a priori du nombre de clusters.
- Pour de petites valeurs de ϵ nous obtenons le meilleur résultat, mais au détriment de plus nombre d'itérations.
- les mesures de distance euclidienne peuvent inégalement pondérer les facteurs sous-jacents.
- Un temps de calcul énorme.
- Sensibilité à la partition initiale (vitesse, minima locaux)
- La sensibilité au bruit
- On s'attend à faible degré d'appartenance (ou même pas) pour les valeurs aberrantes (les points bruyants)

3.7. Les variantes des C-moyennes floues

De nombreuses variations de l'algorithme FCM sont possibles. Nous présentons ici quelques variantes.

3.7.1. L'approche de Kamel et Selim [38]

Selon Khodja[39], Kamel et Selim. [18] ont proposé une généralisation de l'algorithme qui réduit le degré de flou de la partition finale. Lorsqu'un degré d'appartenance u_{ik} du vecteur x_k à la classe i est inférieur à une valeur seuil τ , il est forcé à zéro et les autres degrés d'appartenance sont ensuite normalisés de telle sorte que leur somme soit égale à 1.

Pratiquement, cette approche est mise en œuvre en ajoutant les étapes suivantes à la fin de l'algorithme de Bezdek[7]:

- calcul du seuil (global)

$$\min(\min_i \max_k u_{ik} , \min_k \max_i u_{ik}) \quad (3.11)$$

- Seuillage:

$$(\text{si}(u_{ik} < \tau) \text{ alors } u_{ik} = 0) \quad (3.12)$$

- normalisation, calcul de nouveaux degrés d'appartenance u_{ik}

$$U_{ik} * = \frac{U_{ik}}{\sum_{j=1}^c U_{ik}} \quad (3.13)$$

Cette approche est connue dans la littérature sous le nom de C-moyennes floues à seuil (TFCM, pour Thresholded FCM). Ces étapes supplémentaires peuvent être vues comme une tâche de pré-défuzzification.

3.7.2. L'approche de Kehtarnavaz [39]

Kehtarnavaz et al. [20] ont mis au point une version qui prend en compte le voisinage d'un point donné dans le calcul de ses degrés d'appartenance. Leur approche, connue sous le nom de c-moyennes floues contextuelles (CTFCM, pour Contextual TFCM), reprend l'idée de seuillage de Kamel et Selim mais la valeur du seuil qu'ils adoptent est déterminée localement pour chaque point et non pas globalement comme dans les TFCM. Développant leur méthode pour la segmentation d'images RMN(Résonance Magnétique Nucléaire)., les auteurs définissent le voisinage d'un pixel situé en (x,y) , $V(x,y)$, comme étant une fenêtre centrée sur ce pixel. Pour toute classe i , un degré d'appartenance (de groupe) est défini par :

$$U_i = \sum_{V(x,y)} u_{ik} \quad (3.14)$$

U_i est donc la somme des degrés d'appartenance à la classe i de tous les pixels appartenant au voisinage du pixel situé en (x,y) . U_i dépend de l'environnement du pixel étudié, on peut donc le qualifier de contextuel. A partir de la partition de Bezdek, on effectue alors les étapes suivantes :

- calculer :

$$U_a = \max_i U_i \quad \text{et} \quad U_b = \min_i U_i \quad (3.15)$$

- calculer de nouveaux degrés d'appartenance

$$\text{Si } \frac{U_a}{\sum_{i=1}^c U_i - U_a} \rightarrow 1 \text{ alors } U_{ak} * = 1 \text{ et } U_{ik} * = 0 \text{ pour } i \neq a \quad (3.16)$$

$$\text{Si } \frac{\sum_{i=1}^c U_i - U_b}{U_b} \rightarrow 1 \text{ alors } U_{bk} * = 0 \text{ et } U_{ik} * = \frac{U_{ik}}{\sum_{j=1}^c U_{ij}} \text{ pour } i \neq b \quad (3.17)$$

Ces deux conditions augmentent le contraste entre les différents degrés d'appartenance en prenant en compte l'importance relative des degrés d'appartenance de groupe. La première met en valeur le groupe ayant le plus grand degré d'appartenance (de groupe) contextuel U_a , tandis que le deuxième élimine le groupe au plus faible degré d'appartenance de groupe U_b . Les classes a et b ne correspondent pas forcément au maximum et au minimum des degrés d'appartenance "individuels" du pixel. Par ailleurs, le choix de la taille du voisinage est crucial. En effet, ce dernier doit être à la fois assez petit pour rendre compte du contexte local et assez grand pour traduire le degré d'homogénéité autour du pixel étudié. Cette prise en compte de l'environnement à une échelle locale peut conduire à attribuer un pixel à un groupe ne correspondant pas au plus élevé des degrés d'appartenance de ce pixel. La configuration locale autour d'un pixel suggère dans ce cas que l'appartenance du pixel à un tel groupe n'est pas plausible. Dans le cas des TFCM, le seuil, calculé globalement, ne peut pas rendre compte d'une telle situation.

3.7.3. Une approche due à Pedrycz : FCM conditionnelles

Pedrycz [31] a proposé une version des FCM qui fait intervenir des variables auxiliaires. Les vecteurs sont groupés en tenant compte de leurs coordonnées dans l'espace des données mais aussi de la valeur prise par la variable auxiliaire, et qui traduit une certaine condition fixée par l'utilisateur, d'où le nom donné par Pedrycz à son approche : FCM conditionnelles. Si f est la variable auxiliaire utilisée et f_1, f_2, \dots, f_n sont les valeurs qu'elle prend pour les différents vecteurs, alors f_k décrit le degré d'engagement du vecteur x_k dans la construction de la partition. Cette version est particulièrement attrayante dans le domaine de la requête d'information à partir de bases de données. Il ne s'agit pas simplement de trouver la structure ou les groupes d'un ensemble de données X , mais d'en identifier les groupes sous une certaine condition, ou dans un certain contexte défini par les variables auxiliaires. Une telle condition peut éventuellement être énoncée linguistiquement. Il est à noter que la partition ainsi obtenue peut s'éloigner considérablement de celle qui serait obtenue sans la prise en compte des variables auxiliaires. Ces dernières peuvent priori intervenir de différentes façons dans le calcul des degrés d'appartenance. Pedrycz propose les deux options suivantes :

$$\sum_{i=1}^c u_{ik} = f_k \quad (3.18)$$

$$\max_i u_{ik} = f_k \quad (3.19)$$

et se limite à l'étude de la première, particulièrement simple, qui correspond à des partitions

$$u_{ik} \in [0,1] \quad (3.20)$$

$$\sum_{i=1}^c u_{ik} = f_k \quad (3.21)$$

$$0 < \sum_{k=1}^n u_{ik} < n \quad (3.22)$$

Afin de minimiser la même fonction objective que dans l'algorithme de Bezdek, les degrés d'appartenance sont mis à jour à l'aide de la formule suivante :

$$u_{ik} = f_k \frac{1/(d(x_k, v_i))^{2/(m-1)}}{\sum_{i=1}^c 1/(d(x_k, v_i))^{2/(m-1)}} \quad (3.23)$$

3.7.4. - Les c-moyennes possibilistes [5]

Krishnapuram et Keller [24] ont suivi les idées de Zadeh [13] selon lesquelles (3.24), ne doit pas contraindre les appartenances d'un vecteur quelconque j x si les classes représentées par les nuages sont considérées comme ces sous-ensembles flous sur le domaine $X = \{x_j, j = 1..N\}$ Les degrés d'appartenance doivent seulement appartenir à l'intervalle $[0,1]$.

Ainsi, un nouvel ensemble de contraintes est défini :

$$\forall j \in \{1..N\} \sum_{i=1}^c u_{ij} = 1 \quad (3.24)$$

$$\forall i \in \{1..C\}, \forall j \in \{1..N\} u_{ij} \in [0,1] \quad (3.25)$$

$$\forall j \in \{1..N\} \left\{ \begin{array}{l} 0 < \sum_{j=1}^n u_{ij} < n \\ \max_i u_{ij} > 0 \end{array} \right. \quad (3.26)$$

La condition (3.26), assure simplement que la partition floue résultante de l'algorithme recouvre le domaine X. En toute rigueur, la matrice U résultante n'est plus une C-partition floue, puisque la contrainte (3.24), n'est plus satisfaite.

La fonctionnelle à minimiser (3.27), et les solutions (3.28), (3.29), de l'algorithme PCM deviennent :

$$J_m(U, v) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d(x_j, b_i)^2 + \sum_{i=1}^c \eta_i \sum_{j=1}^n (1 - u_{ij})^m \quad (3.27)$$

$$\forall i \in \{1..C\} \quad b_i \leftarrow \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \quad (3.28)$$

$$\forall i \in \{1..C\}, \forall j \in \{1..N\} \quad u_{ij} = \frac{1}{1 + \left(\frac{d^2(x_j, v_i)}{\eta_i} \right)^{\frac{1}{m-1}}} \quad (3.29)$$

Où η_i est un réel positif déterminant la distance à laquelle le degré d'appartenance d'un vecteur à la classe i est égal à 0,5.

3.7.5. Extension aux contours courbés

Davé [27], [26] a proposé une variante des FCM pour la détection de contours courbés (circulaires, elliptiques) et particulièrement utile en traitement d'image. Cette approche est connue sous le nom de "fuzzy c-shells" ou FCS. D'autres travaux ont été menés sur le thème des "fuzzy c-shells" par des auteurs tels que Davé et Bhaswan [25], Bezdek et Hathaway [6] qui en ont étudié la convergence, Krishnapuram, Nasraoui et Frigui [23] qui ont proposé une autre approche des FCS.

3.7.6. - Approximations

Signalons également que Cannon et al. [2] ont proposé une implémentation des FCM qui fait intervenir des calculs sur des quantités approchées, entières ou réelles. Cette approche, baptisée AFCM (Approximate FCM), serait 6 fois plus rapide que l'implémentation faisant appel au calcul exact des degrés d'appartenance et des coordonnées des centres. La rapidité de la méthode est accrue grâce à l'utilisation de tables pour le calcul approché de puissances et de distances. Elle ne possède cependant pas les mêmes propriétés de convergence que les FCM. Cheng, Goldgof et Hall [29] ont proposé un algorithme baptisé MRFCM M(ultistage Random Sampling FCM) plus rapide que l'algorithme de base et ayant les mêmes propriétés de convergence. [39]

3.8. Les limite du FCM

Un grand nombre d'algorithmes est dérivé du FCM, utilisant des métriques différentes ou des prototypes de dimensions supérieures. Le FCM et ses algorithmes dérivés souffrent 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 construites sont donc interdépendantes. Aussi, les 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 l'hésitation ou l'ambiguïté entre classes à partir de règles de décisions floues fixées à priori. Dans la phase de classification, les points ambigus ou non représentatifs des classes peuvent alors affecter la position des centres.

[19]

3.9. Conclusion

Plusieurs algorithmes de segmentation d'images ont été développés ces dernières années chacun possédant ses propres avantages et ses limites d'utilisation. Dans ce chapitre, Nous avons décrit brièvement, quelques propriétés fondamentales d'algorithmes de classification floue. Nous avons rappelé d'abord l'algorithme dit FCM et nous avons présenté par la suite quelques variantes de ce même algorithme.

CHAPITRE 4

REALISATION ET EVALUATION DES RESULTATS

4.1. Introduction

La segmentation des images consiste à créer une partition de l'image en sous ensembles homogènes appelés régions. Dans ce dernier chapitre et après l'aperçu théorique des chapitres précédents, nous présentons le côté pratique de notre application .Notre but est la réalisation d'un système flexible et fiable qui segmente les images en utilisant l'algorithme FCM et quelques variantes de celui-ci.

4.2. Plateforme et outils de développement

4.2.1. Système d'exploitation

Pour que notre travail atteigne l'objectif qu'on visait, on a pris l'initiative d'exploiter et d'implémenter notre algorithme sur la version : Windows 7.

Ce choix se traduit par l'efficacité de ce L'environnement en ce qui concerne la structure d'interaction événementielle qu'elle dispose pour communiquer avec des applications actives , ainsi que les ressources de la machines qu'il offre aux différentes applications , enfin , son système d'allocation de mémoire qui est un des meilleurs présents dans ce domaine.

4.2.2. Utilisation de l'environnement de développement Visual C#

Pour l'implémentation de notre application, nous avons opté pour le langage C# en utilisant l'EDI (Environnement de développement intégrée) Microsoft visual studio 2010 sous Windows 7.

Microsoft visual studio 2010 Garantisiez des résultats de qualité de la conception au déploiement et Simplifiez votre processus de développement dans un environnement intégré au système de développement Microsoft Visual Studio. Accélérez le processus de codage en travaillant à partir d'un nombre croissant de plateformes, notamment la technologie Microsoft SharePoint et les applications informatiques en nuage.

Les développeurs et les testeurs peuvent travailler de manière plus efficace avec des outils de test et de débogage intégrés pour corriger rapidement et aisément des bogues. La figure (Figure.4.1) présente l'interface de Microsoft visual studio 2010.

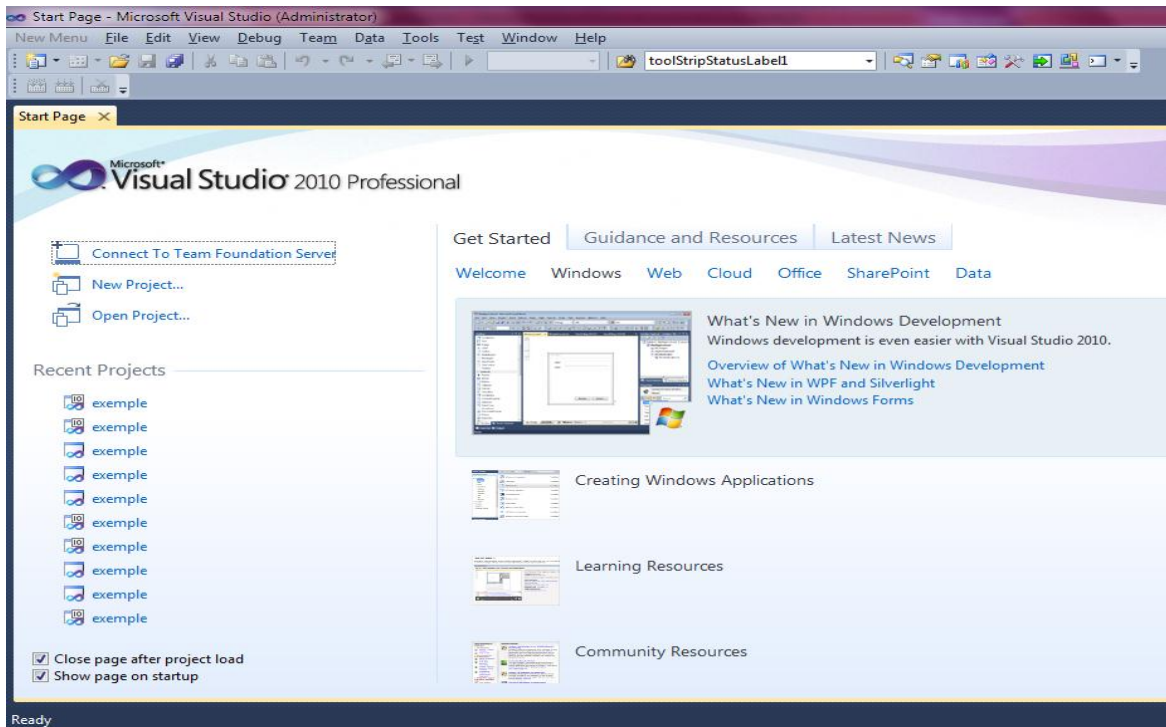


Figure.4.1. L'interface de Microsoft Visual Studio 2010.

4.2.3. Langage de programmation

Pour l'implémentation de notre application, nous avons opté pour le langage C#

4.3. Présentation et implémentation du système

Le système qu'on a réalisé consiste en un système de segmentation d'image utilisant l'algorithme flou FCM et quelques variantes de celui-ci. Nous appliquons l'ensemble de ces algorithmes à des bases de données d'images numériques de format BMP afin de pouvoir évaluer la performance notre système de segmentation d'images.

4.4. Evaluation

L'importance du problème de la segmentation en vision par ordinateur peut se mesurer par le nombre de travaux qui lui ont été dédiés. Une étude datée de 1995 recensait plus d'un millier d'algorithmes proposés [9]. Aujourd'hui, étant donnée l'expansion du domaine, on peut aisément imaginer que ce nombre a doublé.

Face à cette multitude de méthodes, l'évaluation quantitative des résultats est devenue une nécessité. Il s'agit cependant d'une tâche délicate qui a traditionnellement constitué un point faible des approches de segmentation. Souvent, les auteurs se contentent d'illustrer leurs publications avec quelques exemples d'images fréquemment utilisés ("Lenna", "cameraman",

"mandrill", etc.) ou de présenter des images synthétiques qui mettent en évidence les points forts de leur méthode.

Il n'existe pas de "bonne" segmentation. Seule l'appréciation de l'utilisateur (qualité visuelle) et le but recherché permettent de définir une bonne segmentation pour un type de données. C'est pour cette raison que nous avons mesuré et quantifié les performances de notre segmentation de l'ensemble d'image.

4.4.1. Critères de validité

Pour tester les méthodes développées de façon pertinente, nous jugeons la qualité de la segmentation obtenue quantitativement et qualitativement à partir des différents critères d'évaluation.

4.4.1.1. Mesure de dissimilarité de LIU ET YANG

La mesure de dissimilarité de LIU ET YANG [8] notée C_{LY} devant être faible, intègre trois paramètres qui sont le nombre de régions, l'aire des régions et la couleur moyenne de chaque région. Cette mesure pénalise les segmentations ayant un nombre de régions trop importants ou ayant des régions non homogènes en couleur.

$$C_{LY} = \frac{1}{1000 \times A} \sqrt{N} \sum_{i=1}^N \left(\frac{e_i^2}{\sqrt{A_i}} \right) \quad (4.1)$$

Où

- A_i est le nombre de pixels de la région R_i
- A le nombre total de pixels de l'image $A = \sum A_i$
- N représente le nombre de régions de l'image segmentée.
- e_i^2 correspond à l'erreur quadratique portant sur la couleur de la région R_i . Elle est calculée à partir de la somme des distances de norme L_2 entre le vecteur couleur des pixels de la région R_i et le vecteur couleur attribué à la région R_i dans l'image segmentée égal en général à la moyenne des couleurs de la région [34]

4.4.1.2. Critère de Borsotti

La mesure de dissimilarité de Liu et Yang pénalise les segmentations contenant trop de régions ou avec des régions non homogènes en couleur. Borsotti et al ont proposé de l'améliorer par :

$$C_{LY} = \frac{1}{10000 \times A} \sqrt{N} \sum_{i=1}^N \left(\frac{e_i^2}{1 + \log A_i} + \frac{N(A_i)^2}{A_i^2} \right) \quad (4.2)$$

Où $N(A_i)$ est le nombre de régions ayant une aire égale à A_i .

Ce critère doit aussi être faible.

- Le premier terme de la somme favorise les régions homogènes, comme le critère de Liu et Yang.
- Le deuxième terme a une valeur élevée quand il y a beaucoup de petites régions, ce qui pénalise les images sur-segmentées en beaucoup de régions de même taille [15]

4.4.1.3. Contraste de Zeboudj [28]

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

Soit $c(s, t) = \frac{|f(s) - f(t)|}{L-1}$ le contraste entre deux pixels s et t , avec

- f représentant l'intensité
- L représentant le maximum des intensités.

Le contraste intérieur d'une région R_i est :

$$I_i = \frac{1}{A_i} \sum_{s \in R_i} \max\{c(s, t), t \in W(s) \cap R_i\} \quad (4.3)$$

Le contraste extérieur d'une région R_i est :

$$E_i = \frac{1}{l_i} \sum_{s \in F_i} \max\{c(s, t), t \in W(s), t \notin R_i\} \quad (4.4)$$

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

Le contraste de R_i est :

$$C(R_i) = \begin{cases} 1 - \frac{I_i}{E_i} & \text{si } 0 < I_i < E_i \\ E_i & \text{si } I_i = 0 \\ 0 & \text{sinon} \end{cases} \quad (4.5)$$

Le contraste global est enfin :

$$\frac{1}{A} \sum_i A_i \cdot C(R_i) \quad (4.6)$$

4.4.1.4. Critère d'uniformité de mesure [21]

Ce critère simple est basé sur la somme des variances des régions.

$$u = 1 - 2 * c * \frac{\sum_{j=0}^c \sum_{i \in R_j} (f_i - \mu_j)^2}{N * (f_{\max} - f_{\min})} \quad (4.7)$$

Où

- c est le nombre de classes ;
- R_j la région de chaque classe j ;
- f_i la valeur dans chaque pixel i ;
- μ_j la moyennes de division la valeur de pixel i sur la somme de pixel i ;
- N est le nombre totale des pixels dans un image donnée ;
- f_{\min} et f_{\max} représenter la valeur minimale et maximale en respective dans une image donnée.

La valeur d'uniformité va entre 0 et 1. La maximum valeur d'uniformité de la segmentation d'image est la meilleure.

Dans notre cas nous avons utilisé le critère d'uniformité de mesure. L'avantage de ce critère est d'être facilement mis à jour dans les opérations de division des régions.

4.4.2. Les images utilisées

L'utilisation des images standards permet de comparer quantitativement la segmentation obtenue d'une chaque algorithmes par rapport les autres.

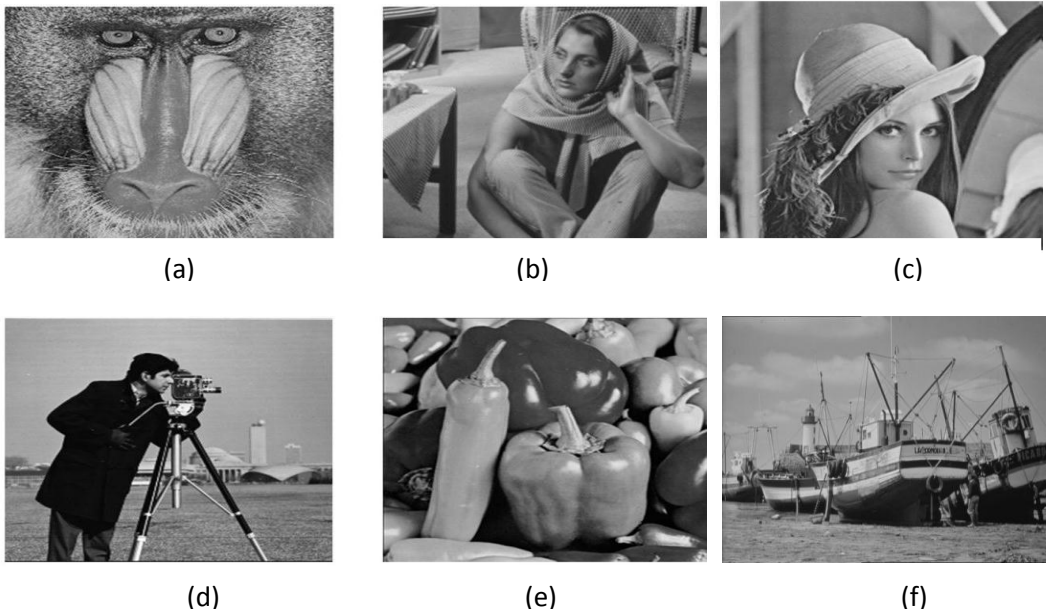


Figure.4.2. images standard. (a) Baboon, (b) Barbara, (c) Lena, (d) Cameraman, (e) Peppers, (f) Boats.

4.5. Présentation et Evaluation des résultats obtenus

L'évaluation objective et quantitative des résultats joue un rôle important dans la segmentation d'image. Il est possible pour un même système de segmentation de tester quantitativement l'influence du choix de certains paramètres sur les résultats de la segmentation.

4.5.1. L'algorithme FCM

Fuzzy C-Means (FCM) est un algorithme de classification non-supervisée floue. Issu de l'algorithme des C-moyennes (C-means), il introduit la notion d'ensemble flou dans la définition des classes : chaque point dans l'ensemble des données appartient à chaque cluster avec un certain degré, et tous les clusters sont caractérisés par leur centre de gravité.

Etape 1 : fixer les paramètres.

Les entrées : $X = (x_j, j = 1..N)$ l'ensemble des vecteurs forme. C : nombre de classes.

ε : Seuil représentant l'erreur de convergence (par exemple $\varepsilon = 0.001$)

m : degré de flou, $m \in (1.5,3)$

Etape2 : initialiser la matrice degrés d'appartenance U par valeur aléatoire dans l'intervalle $[0,1]$.

Etape 3 : mettre à jour la matrice prototype B par la relation.

$$b_i \leftarrow \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \quad (4.8)$$

$$j^{\text{ancien}} \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i) \quad (4.9)$$

Etape 4 : mettre à jour la matrice des degrés d'appartenance par la relation

$$u_{ij} \leftarrow \left[\sum_{k=1}^c \left(\frac{d^2(x_j, b_i)}{d^2(x_j, b_k)} \right)^{2/(m-1)} \right]^{-1} \quad (4.10)$$

$$j^{\text{nouveau}} \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i)$$

Etape 5 : répéter les étapes 3 a 4 jusqu'à satisfaction du critère d'arrêt qui s'écrit :

$$\|j^{\text{ancien}} - j^{\text{nouveau}}\| \leq \varepsilon \quad (4.11)$$

Les sorties : la matrice d'appartenance U et les centres des classe B .

Les résultats obtenus par FCM :


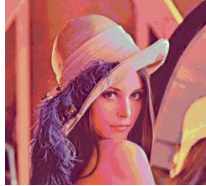


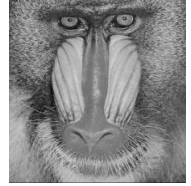



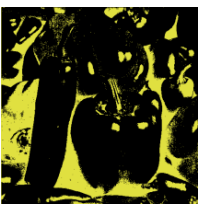



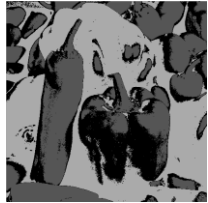
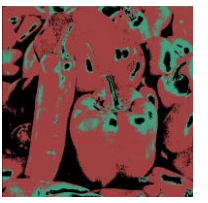



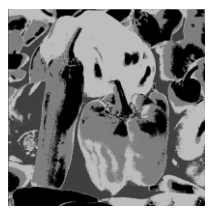
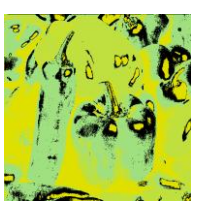



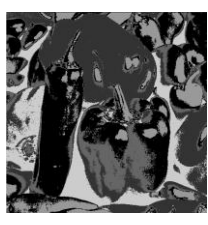
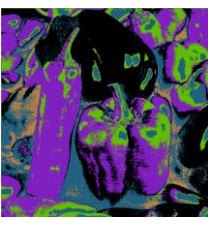




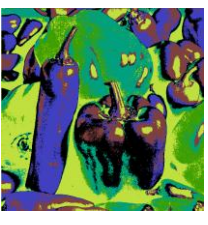

Image Nbr de classe					
	Lenna niveau de gris	Lenna couleur	Pepper niveau de gris	Pepper couleur	Baboon niveau de gris
2					
3					
4					
5					
6					

Table.4.1. exemple du les résultats de segmentation d'image par FCM.

image	Nombre de classe	Les centres	L'itération	temps	uniformité	Fonction objectif
Baboon niveau de gris	2	164 ;91	8	10,97	0,9568762	4139442
	3	175 ;124 ;69	7	19,08	0,9726257	2459992
	4	182 ;111 ;145 ;63	14	1 :01,03	0,9888366	1799726
	5	156 ;93 ;125 ;185 ;55	21	2 :06,51	0,9813491	1418992
	6	87 ;52 ;187 ;164 ;117 ;140	15	2 :08,51	0,9924669	1168771
Pepper couleur	2	114 ;14 ;	4	6,58	0,9335725	4665659
	3	9 ;99 ;183 ;	8	24,67	0,9825	2373577
	4	183 ;9 ;84 ;113	19	01 :18,07	0,9958156	1171823
	5	85 ;183 ;1 ;11 ;113	4	24,52	0,9906875	1029333
	6	107 ;78 ;183 ;9 ;88 ;118	8	01 :05,32	0,9993585	9.39566.1
Pepper niveau de gris	2	51 ;155 ;	6	9,93	0,9464869	4651778
	3	119 ;169 ;45 ;	16	045,43	0,950572	3263112
	4	178 ;18 ;137 ;66	21	01 :34,08	0,986064	2213929
	5	151 ;61 ;110 ;190 ;15 ;	26	02 :49,89	0,9724575	1792890
	6	47 ;70 ;120 ;156 ;11 ;196	28	04 :06,66	0,9928228	1468037
Lenna couleur	2	78 ;171	3	5,7	0,9680148	2952213
	3	67 ;170 ;87	2	5,09	0,9594411	2030282
	4	169 ;72 ;75 ;89	6	29,75	0,9580289	1946528
	5	89 ;166 ;237 ;7 ;75	12	01 :20,89	0,9916845	1375332
	6	74 ;177 ;238 ;89 ;162 ;7	6	53,62	0,9990263	738454.1
Lenna niveau de gris	2	81 ;159 :	6	2,87	0,9426587	1101968
	3	133 ;68 ;186 ;	14	9,05	0,9590787	685089.4
	4	60 ;107 ;194 ;146	12	13,25	0,9887851	460007.4
	5	85 ;129 ;99 ;156 ;199	9	17,36	0,97896	345581.6
	6	58 ;121 ;143 ;202 ;166 ;96	13	30,88	0,9934313	290683.9

Table.4.2.Résultats expérimentaux de segmentation d'image par FCM.

4.5.2. L'algorithme TFCM :

Cette approche est connue dans la littérature sous le nom de C-moyennes floues à seuil (TFCM, pour Thresholded FCM). Ces étapes supplémentaires peuvent être vues comme une tâche de pré-défuzzification.

Etape 1 : fixer les paramètres.

Les entrées : $X = (x_j, j = 1..N)$ l'ensemble des vecteurs forme. C : nombre de classes.

ε : Seuil représentant l'erreur de convergence (par exemple $\varepsilon = 0.001$)

m : degré de flou, $m \in (1.5,3)$

Etape2 : initialiser la matrice degrés d'appartenance U par valeur aléatoire dans l'intervalle $[0,1]$.

Etape 3 : mettre à jour la matrice prototype B par la relation.

$$b_i \leftarrow \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m}$$

$$j^{ancien} \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i)$$

Etape 4 : mettre à jour la matrice des degrés d'appartenance par la relation

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

Etape 5 : calcul du seuil (global)

$$\min(\min_i \max_k u_{ik} , \min_k \max_i u_{ik}) \quad (4.12)$$

Etape 6 : Seuillage:

$$(si(u_{ik} < \tau) alors u_{ik} = 0) \quad (4.13)$$

Etape 7 : normalisation, calcul de nouveaux degrés d'appartenance u_{ik}

$$U_{ik} * = \frac{U_{ik}}{\sum_{j=1}^c U_{ik}} \quad (4.14)$$

$$j^{nouveau} \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i)$$

Etape 8: répéter les étapes 3 a 7 jusqu'à satisfaction du critère d'arrêt qui s'écrit :

$$\|j^{ancien} - j^{nouveau}\| \leq \varepsilon$$

Les sorties : la matrice d'appartenance U et les centres des classe B .

Les résultats obtenus par TFCM :





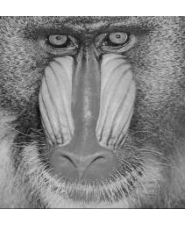












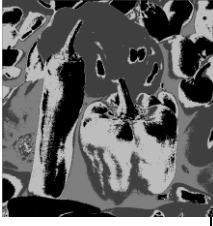




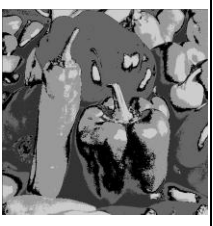
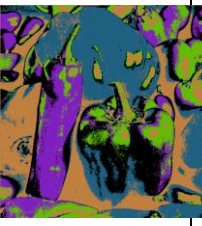



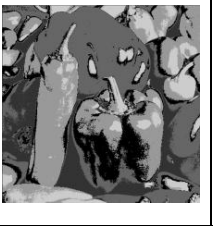


Nbr de classe	Image					
		Lenna niveau de gris	Lenna couleur	Pepper niveau de gris	Pepper couleur	Baboon niveau de gris
2						
3						
4						
5						
6						

Table.4.3. exemple du les résultats de segmentation d'image par TFCM.

image	Nombre de classe	Les centres	L'itération	temps	uniformité	Fonction objectif
Baboon niveau de gris	2	163 ;87	8	11,31	0,9567012	4138921
	3	66 ;175 ;122 ;	6	16,64	0,9725712	2447352
	4	173 ;67 ;173 ;120 ;	10	48,82	0,9757606	1845910
	5	182 ;118 ;85 ;148 ;50 ;	6	47,22	0,9814002	1415781
	6	159 ;51 ;84 ;132 ;112 ;185	12	01 :37,04	0,993564101	1168205
Pepper couleur	2	115 ;15 ;	7	11,72	0,9335725	4621098
	3	99 ;183 ;9 ;	6	11,06	0,9825	2369712
	4	85 ;113 ;9 ;183 ;	6	24,32	0,9962633	1157689
	5	105 ;183 ;9 ;83 ;118	5	34,52	0,9926574	1041706
	6	183 ;83 ;103 ;1 ;117 ;11	10	01 :22,51	0,9933564	810969.1
Pepper niveau de gris	2	155 ;51 ;	7	9,98	0,9465133	4652650
	3	21 ;73 ;159	6	16,4	0,9509172	3325624
	4	174 ;64 ;16 ;132 ;	18	1 :14,78	0,9777165	2217660
	5	110 ;15 ;61 ;151 ;190 ;	23	02 :39,39	0,9725159	1792048
	6	200 ;59 ;161 ;14 ;94 ;133	10	1 :30,44	0,9917442	1466837
Lenna couleur	2	78 ;171 ;	3	4,39	0,9680148	2954458
	3	79 ;236 ;166 ;	5	27,39	0,9739603	2028985
	4	8 ;77 ;170 ;94 ;	3	13,87	0,9987323	1632620
	5	7 ;166 ;89 ;237 ;75 ;	10	1 :0,9	0,9916845	1128438
	6	167 ;95 ;7 ;237 ;74 ;85	6	45,64	0,9916618	780048
Lenna niveau de gris	2	79 ;160 ;	6	3,45	0,9426979	1096635
	3	186 ;133 ;67 ;	16	12,47	0,9590189	687434.4
	4	149 ;197 ;61 ;110 ;	13	17,95	0,9860032	456306.2
	5	156 ;58 ;129 ;199 ;98 ;	10	16,68	0,97896	345952.1
	6	58 ;99 ;128 ;204 ;174 ;150 ;	14	35,88	0,9904739	291125.1

Table.4.4. Résultats expérimentaux de segmentation d'image par TFCM.

4.5.3. L'algorithme CTFCM

Cet algorithme connue sous le nom de c-moyennes floues contextuelles (CTFCM, pour Contextual TFCM), il reprend l'idée de seuillage de Kamel et Selim mais la valeur du seuil qu'ils adoptent est déterminée localement pour chaque point et non pas globalement comme dans les TFCM.

Etape 1 : fixer les paramètres.

Les entrées : $X = (x_j, j = 1..N)$ l'ensemble des vecteurs forme. C : nombre de classes.

ε : Seuil représentant l'erreur de convergence (par exemple $\varepsilon = 0.001$)

m : degré de flou, $m \in (1.5,3)$

Etape2 : initialiser la matrice degrés d'appartenance U par valeur aléatoire dans l'intervalle $[0,1]$.

Etape 3 : mettre à jour la matrice prototype B par la relation.

$$b_i \leftarrow \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m}$$

$$j^{ancien} \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i)$$

Etape 4 : mettre à jour la matrice des degrés d'appartenance par la relation

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

Etape 5 : calculer un degré d'appartenance (de groupe)

$$U_i = \sum_{V(x,y)} u_{ik} \quad (3.15)$$

Etape 6 : calculer :

$$U_a = \max_i U_i \quad \text{et} \quad U_b = \min_i U_i \quad (3.16)$$

Etape 7 : calculer de nouveaux degrés d'appartenance

$$\text{Si } \frac{U_a}{\sum_{i=1}^c U_i - U_a} \rightarrow 1 \text{ alors } U_{ak} = 1 \text{ et } U_{ik} = 0 \text{ pour } i \neq a \quad (3.17)$$

$$\text{Si } \frac{\sum_{i=1}^c U_i - U_b}{U_b} \rightarrow 1 \text{ alors } U_{bk} = 0 \text{ et } U_{ik} = \frac{U_{ik}}{\sum_{j=1}^c U_{ij}} \text{ pour } i \neq b \quad (3.18)$$


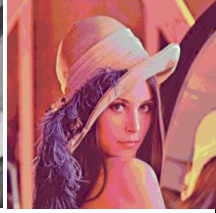


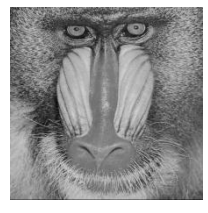

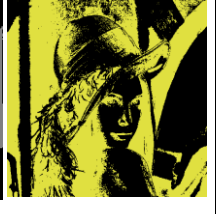


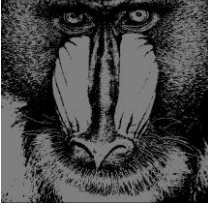



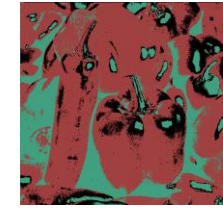


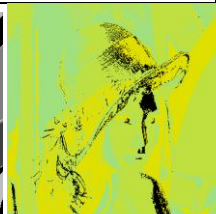
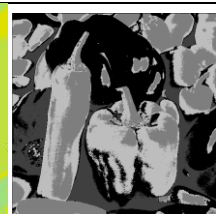
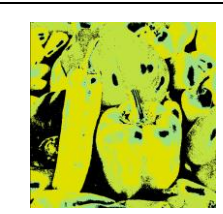
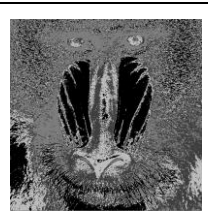


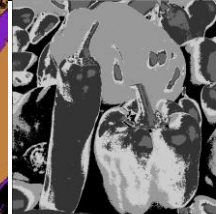
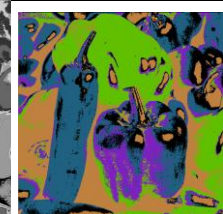




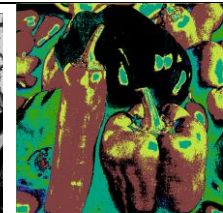

$$j^{nouveau} \leftarrow \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d^2(x_j, b_i)$$

Etape 8: répéter les étapes 3 a 7 jusqu'à satisfaction du critère d'arrêt qui s'écrit :

$$\|j^{ancien} - j^{nouveau}\| \leq \varepsilon$$

Les sorties : la matrice d'appartenance U et les centres des classe B .

Les résultats obtenus par CTFCM :

Image Nbr de classe					
	Lenna niveau de gris	Lenna couleur	Pepper niveau de gris	Pepper couleur	Baboon niveau de gris
2					
3					
4					
5					
6					

Tab.4.5. exemple du les résultats de segmentation d'image par CTFCM.

image	Nombre de classe	Les centres	L'itération	temps	uniformité	Fonction objectif
Baboon niveau de gris	2	89 ;163 ;	8	21,0	0,9568514	4114456
	3	124 ;176 ;69 ;	8	31,70	0,9725713	2449086
	4	181 ;109 ;62 ;142	8	51,32	0,8990808	1802395
	5	156 ;125 ;185 ;55 ;93 ;	28	4 :31,87	0,9813658	1415453
	6	52 ;138 ;186 ;162 ;116 ;86 ;	12	2 :43,89	0,993659	1158824
Pepper couleur	2	14 ;115 ;	2	4,07	0,9335725	4673524
	3	183 ;99 ;9	7	25,55	0,9825	2373577
	5	9 ;113 ;84 ;183	9	49,76	0,9968799	1170838
	5	183 ;82 ;9 ;116 ;100	9	1 :19,00	0,9922965	872856.6
	6	85 ;1 ;113 ;183 ;10 ;21 ;	11	2 :8,40	0,9972381	80327.9
Pepper niveau de gris	2	51 ;155 ;	2	4,07	0,9464869	4652669
	3	112 ;43 ;166 ;	5	17,23	0,9504395	3262802
	4	65 ;17 ;174 ;133 ;	11	1 :09,71	0,9868397	2208078
	5	16 ;157 ;197 ;63 ;119	19	2 :43,09	0,9725373	1792444
	6	60 ;137 ;15 ;101 ;201 ;163	12	2 :25,04	0,9898622	1473080
Lenna couleur	2	171 ;78 ;	4	10,71	0,9680148	2954644
	3	171 ;9 ;83	4	17,18	0,970614	2029382
	4	237 ;70 ;166 ;87 ;	4	28,08	0,9814491	1414479
	5	87 ;238 ;166 ;166 ;71 ;	6	55,34	0,9689605	1077590
	6	89 ;7 ;157 ;238 ;75 ;170 ;	10	2 :11,6	0,9988454	1003543
Lenna niveau de gris	2	79 ;160 ;	11	6,57	0,9426979	1214116
	3	67 ;133 ;186 ;	9	7,74	0,9590787	686375.3
	4	91 ;197 ;149 ;109 ;	14	19,49	9,9893935	456458.1
	5	157 ;100 ;59 ;130 ;200 ;	9	19,09	0,9790581	346312.1
	6	126 ;171 ;204 ;148 ;98 ;58 ;	9	23,66	0,9952553	293709.2

Table.4.6.Résultats expérimentaux de segmentation d'image par CTFCM.

Discussion:

A partir des tableaux précédents nous avons remarqué que :

- Les résultats obtenus de la segmentation par FCM et ces variantes sont satisfaisants .
- les valeurs du critère d'uniformité montrent clairement la performance des variantes proposées par rapport à l'algorithme FCM.
- les trois approches fournissent des centres qui possèdent des valeurs très proches.
- lors que le nombre des centres augmente la valeur de fonction objective diminue.
- La performance des variantes proposées change par rapport à l'image.

4.6. Quelques interfaces de l'application

L'interface principale de notre logiciel est présentée par la figure suivante

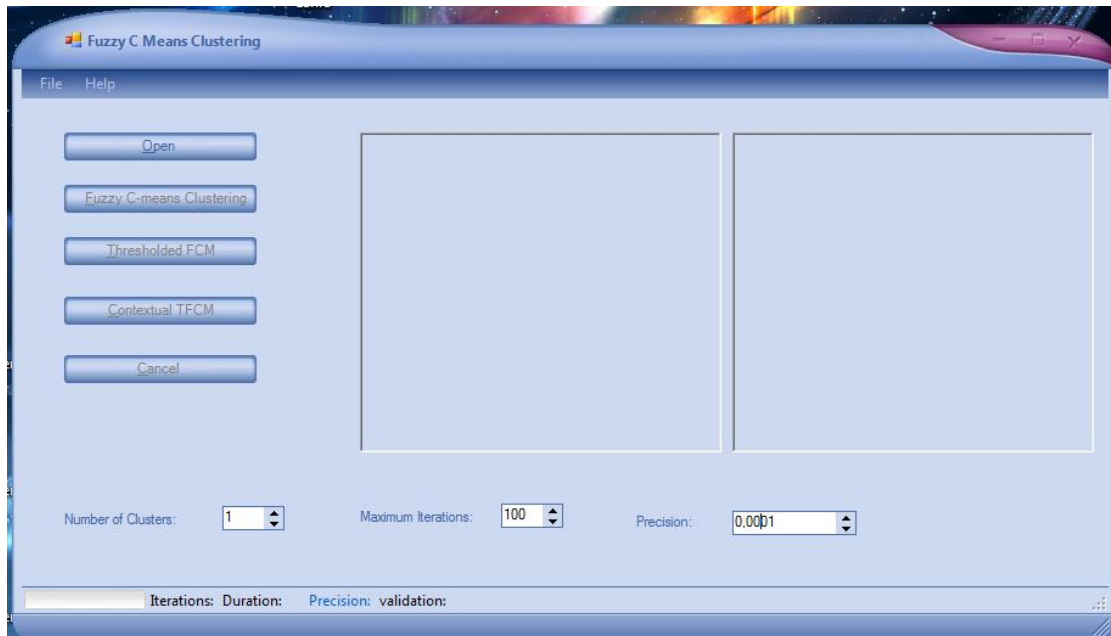


Figure.4.3. L'interface principale.

Dans le menu fichier on peut charger un image, et on peut également applique les algorithmme FCM(fuzzy c means),TFCM(Thresholded Fuzzy C Means),CTFCM(Contextuel Thresholded Fuzzy C Means) (voir Figure. 4.4).

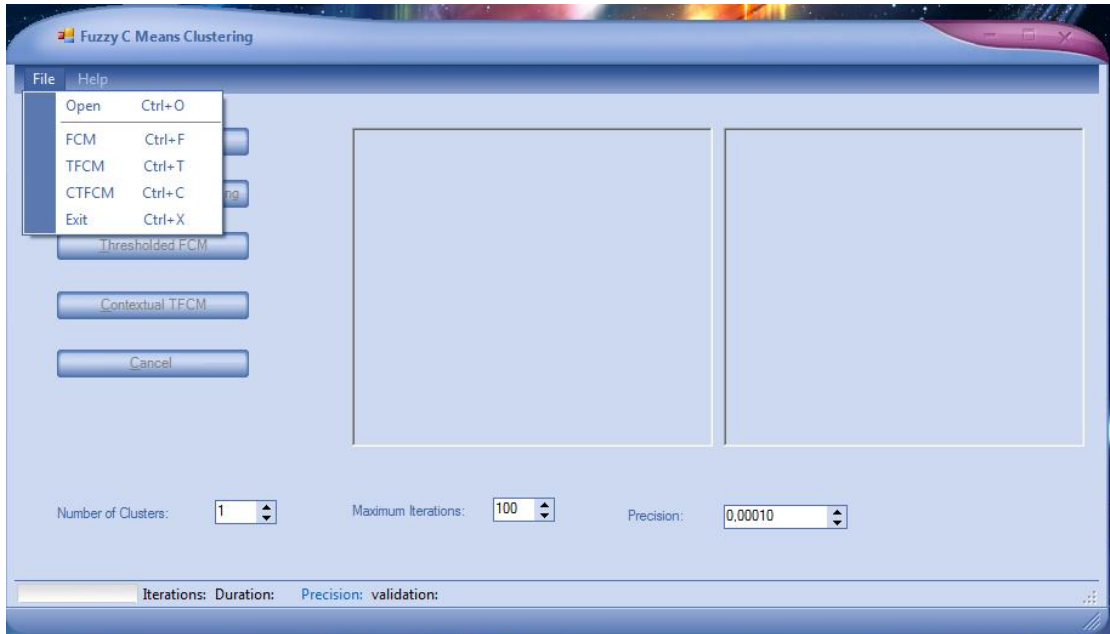


Figure.4.4. Le menu de Fichier.

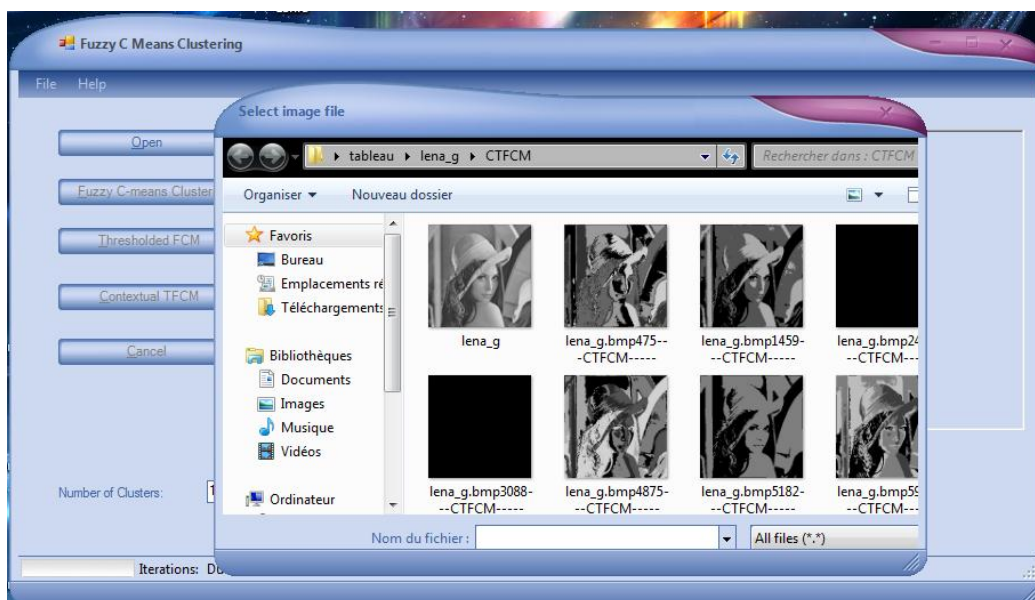


Figure.4.5. L'interface select image file



Figure.4.6. La résultat de la segmentation.

4.7. Conclusion

Nous avons présenté dans ce chapitre les résultats de segmentation pour l'algorithme flou FCM et quelque variante de celui-ci .et pour évaluer la qualité de notre segmentation, nous avons utilisé la méthode d'évaluation proposée par Levine and Nazif.

Les résultats expérimentaux portant sur différents images ont montré l'efficacité de l'algorithme FCM et ces variantes et que l'amélioration apporter par ces derniers sont satisfaisante.

CONCLUSION GENERALE

La segmentation d'images est l'une des opérations les plus importantes en traitement d'images. Elle permet d'extraire des informations pertinentes de l'image. Il existe une grande variété de techniques de segmentation, mais aucune n'est efficace sur tous les types d'images. D'où ce domaine de recherche reste ouvert pour trouver des méthodes plus évoluées et plus efficaces

Dans ce mémoire, nous avons abordé le problème de la segmentation d'images. Plus précisément, nous nous sommes focalisés à la segmentation d'images basée sur la logique floue. Et son l'implémentation avec une comparaison du ces différent variantes.

Notre système a été testés avec plusieurs d'images. Pour l'évaluer, nous avons calculé le critère d'uniformité pour chaque image pour les trois méthodes adoptées. Nos résultats ne sont pas toujours excellents avec toutes les images mais ils sont cependant très prometteurs. Tout fois, on peut dire qu'ont est satisfait des résultats obtenu que le travail soit bien valorisé.

A l'issue des travaux menés dans le cadre de ce mémoire, nous dégageons quelques perspectives. Les perspectives de ce travail peuvent être résumées dans les points suivants :

- * Ce travail peut être enrichi par d'autres méthodes de segmentation
- * Ce travail peut être également enrichi par d'autres variantes de FCM
- * Développer un système base sur la coopération de plusieurs algorithmes de différentes méthodes par exemple (algorithme génétique et FCM)
- * Testes les méthodes implémentées sur d'autres types d'images (IRM).

BIBLIOGRAPHIE

- [1] C. Parisel, Structure du format BMP : sa lecture sa construction et son écriture, Université de Montréal, Canada, 2003.
- [2] E. Charnik, D. McDermott, Introduction to Artificial Intelligence, Addison-Wesley, Reading, Mass.1986.
- [3] F. D'Hondt, B. El Khayati, Etude de méthodes de Clustering pour la segmentation d'images en couleurs, Faculté Polytechnique de Mons.
- [4] H. Chellakh, A. Moussaoui, La segmentation des IRM cérébrales pathologiques par une combinaison Floue Possibiliste Markovienne, Université Farhet Abbas de Sétif, Algérie.
- [5] H. Zouaoui, A. Moussaoui, Clustering par fusion floue de données appliqué à la segmentation d'images IRM cérébrales, Faculté Polytechnique de Mons,.
- [6] J. Bezdek ,R. J. Hathaway, Numerical convergence and interpretation of the fuzzy c-shells clustering algorithm, IEEE Trans. on Neural Networks, vol. 3, n. 5, pp. 787-793, 1992.
- [7] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
- [8] J. LIU, Y. YANG. Multiresolution color image segmentation. IEEE Trans on PAMI, 16(7), pages 689–700, 1994.
- [9] J. M. Morel, S. Solimini, Variational Methods in Image Segmentation. Birkhauser, 1995.
- [10] J. P. Cocquerez, S. Philipp, Analyse d'images: filtrage et segmentation, Masson. 1995.
- [11] J. P. Gastellu , Acquisition et traitement d'image numerique, Université de Paul Sabatier, France, 2008.
- [12] K. Sankar, D. Majumder, Fuzzy Sets and Decisionmaking Approaches in Vowel and Speaker Recognition, IEEE Trans. Syst. Man Cybern., pp. 625-629, August, 1977.
- [13] L. Zadeh, Fuzzy Sets as a Basis for Theory of Possibility, International Journal of Fuzzy Sets and Systems, 1:3-28. 1978
- [14] L. Zadeh, Fuzzy sets, Information Control, 13(8) :338 353, 1965.
- [15] M .Borsotti, P .Campadelli, R .Schettini, Quantitative evaluation of color image segmentation results, Pattern Recognition Letters, 1998, 19, p. 741-747.

- [16] M. Bergounioux, Quelques méthodes mathématiques pour le traitement d'image, Université d'Orléans, France, 2009.
- [17] M. Colinet, Le traitement des images, Université de Namur, Belgique, 2004.
- [18] M. S. Kamel, S.Z.Selim, A threshold fuzzy c-means algorithm for semi- fuzzy clustering, Pattern Recognition 24, 1991, pp. 825-833
- [19] M. Semchedine, L.Toumi, A. Moussaoui, Classification Adaptative Robuste pour la Segmentation d'Images IRM Cérébrales, Université Ferhat Abbas de Sétif, Algérie , 2007.
- [20] N. Kehtarnavaz, M. Chung, Magnetic resonance image segmentation by contextual fuzzy clustering, Journal of Intelligent and Fuzzy Systems, Vol. 1, 1994, pp. 295-305.
- [21] P .Y. Yin, L.H.Chenb, A fast iterative scheme for multilevel thresholding methods, Signal Processing 60 (1997) 305-3 13
- [22] R. Deriche, International journal of computer vision, vol. 98, pp. 167-187. 1987.
- [23] R. Krishna, O. Nasraoui , H. Frigui, The fuzzy c spherical shells algorithm : A new approach, IEEE Trans. on Neural Networks, vol. 3, n. 5, pp. 663-671, 1992.
- [24] R. Krishna, The Possibilistic C-Means Algorithm: Insights and Recommendations, IEEE transactions on Fuzzy Systems 1996
- [25] R. N. Davé , K. Bhaswan, Adaptive fuzzy c-shells clustering and detection of ellipses, IEEE Trans. on Neural Networks, vol. 3, n. 5, pp. 643-662, 1992.
- [26] R. N. Davé, Generalized fuzzy c-shells clustering and detection of circular and elliptical boudaries, Pattern Recognition, Vol. 25, n. 7, pp. 713-721, 1992.
- [27] R.N. Davé, Fuzzy c-shells clustering and applications to circle detection in digital images, General Systems, vol. 16, n. 4, pp. 343-355, 1990.
- [28] S, Philipp, L.Guigues. Évaluation de la segmentation d'images : état de l'art, nouveaux indices et comparaison. Traitement du signal 2006, volume 23, numéro 2
- [29] T. W. Cheng, D.B. Goldgof, L.O. Hall, Fast Fuzzy Clustering with Application to Fuzzy Rule Generation, in the Proceedings of the FUZZ-IEEE 95 Conference, Tokyo, Japan, 1995, pp. 2289-2295.
- [30] V. Bénony, A.Sedoglavic, Stéganographie élémentaire avec un fichier BMP, Université des Sciences et Technologies de Lille, France, 2004.
- [31] W. Pedrycz, Conditional fuzzy c-means, Pattern Recognition Letters 17, pp. 625-631, 1996.

- [32] Y. Yakimovsky, Boundary and objet detection in real world images, ACM, vol.23, no. 4, pp.599-618, 1976.
- [33] A. PUJOL, Contributions à la Classification Sémantique d'Images, Thèse de Doctorat, Ecole Centrale de Lyon,France, 2009.
- [34] C.Meurie, Segmentation d'images couleur par classification pixellaire et hiérarchie de partitions". Thèse de Doctorat, Université de CAEN, France, 2005
- [35] H. Elghazel, Analyse et identifications de bactéries dans les images microscopiques couleur de biofilms , Mémoire de Licence, Université de Sfax, Tunisie, 2003.
- [36] H. Laguel, Déploiement sur une plateforme de visualisation, d'un algorithme coopératif pour la segmentation d'images IRM basé sur les systèmes multi-agents, Projet de Fin d'Étude pour l'obtention du diplôme d'ingénieur d'état, Université de Houari Boumediene, Algérie ,2010.
- [37] H. Laguel, Déploiement sur une plateforme de visualisation, d'un algorithme coopératif pour la segmentation d'images IRM basé sur les systèmes multi-agents, Projet de Fin d'Étude pour l'obtention du diplôme d'ingénieur d'état, Université de Houari Boumediene, Algérie ,2010.
- [38] H. Zouaoui , Clustering par fusion floue de données appliqué à la segmentation d'images IRM, Mémoire de Magister , Université M'hamed Bougara de Boumerdes, Algérie,2008.
- [39] L. Khodja, Contribution à la Classification Floue non Supervisée, Thèse de Doctorat,Université de Savoie, France, 1997
- [40] M. Lehamel, Segmentation d'images texturées à partir des attributs fractals, Mémoire de Magister, Université de Mouloud Mammeri de Tizi-Ouzou, Algérie, 2007.
- [41] M.El hassani, Segmentation basée-région pour l'amélioration de séquences vidéo : algorithme-architecture dans un contexte temps réel, Thèse de Doctorat, Université de Caen, France, 2006.
- [42] N. Passalacqua , Méthodes de clustering flou appliquées au « filtrage parental » d'Internet, Mémoire de stage de Master , Université de Montpellier 2,France, 2006.
- [43] P. Arbeláez, Une approche métrique pour la segmentation d'images, Thèse de Doctorat, Université de PARIS, France, 2005.
- [44] T. Nguyen, Binarisation d'images de documents graphiques, rapport de stage de fin d'études université de Hanoi, Vietnam, 2004.

WEBGRAPHIE

- [45] http://fr.wikipedia.org/wiki/D%C3%A9tection_de_contours
- [46] http://fr.wikipedia.org/wiki/Segmentation_d'image
- [47] <http://www.commentcamarche.net/contents/1191-infographie>
- [48] <http://www.commentcamarche.net/contents/1216-traitement-d-images>
- [49] <http://www.map.toulouse.archi.fr/works/panoformation/imagenum/imagenum.htm>
- [50] <http://www.misfu.com/cours/tutoriel/morphologie-mathematique-642.html>
- [51] <http://www.tsi.telecom-paristech.fr/pages/enseignement/ressources/>
- [52] <http://www.webmarketing-com.com/2012/11/06/16580-quels-sont-les-9-formats-differents-pour-une-image>
- [53] <http://www.wikipedia.org/>
- [54] http://www.xphilipp.developpez.com/articles/filtres/?page=page_3
- [55] <https://www.sites.google.com/site/dataclusteringalgorithms/fuzzy-c-means-clustering-algorithm>

Résumé :

Ce mémoire s'inscrit dans le domaine de traitement d'images, le domaine qui est parmi les plus étudiés depuis l'avènement de l'informatique. Le but principal du mémoire est l'étude des algorithmes de segmentation floue pour les images BMP, plus précisément, nous nous sommes focalisés à la segmentation d'images avec FCM (Fuzzy C Means) et ces variantes: TFCM (Thresholded Fuzzy C Means) et CTFCM (Contextual Thresholded Fuzzy C Means)

Mot clés : FCM, TFCM, CTFCM, Segmentation, Traitement d'images, Logique floue, BMP.

Abstract

Image processing is the field which is among fields that are been the most studied since the advent of data processing. The principal goal of this thesis is the study of the fuzzy images segmentation algorithms for BMP images; we focus on segmentation with FCM (fuzzy c means) and its variant: TFCM (Thresholded Fuzzy C Means) and CTFCM (Contextual Thresholded Fuzzy C Means).

Keywords: FCM, TFCM, CTFCM, Segmentation, images processing, Fuzzy logic BMP.

ملخص

تندرج هذه المذكرة في مجال معالجة الصور ،هذا المجال الذي يعتبر من أكثر المجالات دراسة منذ ظهور معالجة البيانات ،الهدف الرئيسي من هذه المذكرة هو دراسة الخوارزميات الضبابية لتقسيم الصور من أجل الصور ذات الامتداد BMP مع التركيز على تقسيم الصور باستخدام الخوارزمية الضبابية and TFCM (Thresholded Fuzzy C Means) و الخوارزميات المحسنة FCM (Fuzzy C Means) and CTFCM (Contextual Thresholded Fuzzy C Means)

كلمات مفتاحية: FCM, TFCM, CTFCM, التقسيم, معالجة الصور, BMP, المنطق الضبابي.