Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERI de Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique
Département Automatique

MEMOIRE DE MAGISTER

En Automatique

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

Présenté par : **M^{elle} SBILI Lila**

Segmentation d'images par classification floue

Mémoire soutenu le : 08/03/2015

Devant le jury d'examen composé de :

Président	**DIAF Moussa**	Professeur	UMMTO
Rapporteur	**HAMMOUCHE Kamal**	Professeur	UMMTO
Examinateur	**MELLAH Rabah**	Maitre de conférences A	UMMTO
Examinateur	**HADDAB Salah**	Maitre de conférences A	UMMTO
Examinateur	**AICHE Farid**	Maitre de conférences B	UMMTO

A mon défunt frère Kamal, qu'Allah l'accueille dans

son vaste paradis, à ma très chère famille.

Remerciements

Cette thèse a été préparée au sein du laboratoire Robotique et Vision, département Automatique de l'université Mouloud Mammeri de Tizi-Ouzou.

Tout d'abord je tiens à exprimer ma grande reconnaissance pour mon encadreur Mr. HAMMOUCHE Kamal, professeur à l'université Mouloud Mammeri de Tizi-Ouzou. Je lui adresse un grand merci pour sa grande disponibilité, ses remarques et ses conseils tout au long de ce travail.

Je tiens à remercier également les membres de mon jury, monsieur DIAF Moussa, professeur à l'UMMTO, monsieur MELLAH Rabah, maître de conférence A à l'UMMTO, monsieur HADDAB Salah, maître de conférence A à l'UMMTO et monsieur AICHE Farid, maître de conférence B à l'UMMTO d'avoir accepté d'examiner ce présent travail.

J'adresse également ma profonde gratitude et ma reconnaissance à toute ma famille, mes très chers parents qu'Allah les bénisse, mes chers frères et mes chères sœurs qui, sans leur soutien et encouragement, ce travail n'aurait pas vu le jour.

Je remercie aussi mes amis et toutes les personnes qui me sont très chères et qui ont contribué de prés ou de loin à l'élaboration de ce travail.

Enfin, j'adresse un grand merci à Melle. DORBANE pour son aide et son temps qu'elle m'a accordé.

Sommaire

Introduction générale 01

Chapitre 1 : Notions générales sur la logique floue

1.1 Introduction 05

1.2 Notions sur les sous ensembles flous 05

1.2.1 Introduction 05

1.2.2 Définition d'un sous ensemble flou 05

1.2.3 Fonction d'appartenance 06

1.2.4 Caractéristiques des sous flous 09

1.2.4.1 La hauteur 09

1.2.4.2 Le support 09

1.2.4.3 Le noyau 09

1.2.4.4 La cardinalité 09

1.2.4.5 La α-coupe 10

1.2.5 Opérations ensemblistes sur les ensembles flous 10

1.2.5.1 Complémentarité d'un sous ensemble flou 11

1.2.5.2 Egalité et inclusion de deux sous ensembles flous 11

1.2.5.3 Union et intersection de deux sous ensembles flous 11

1.2.5.4 T-norme et t-conorme triangulaire 12

1.2.6 Produit cartésien de deux sous ensembles flous 13

1.2.7 Structure générale d'un traitement flou 14

1.2.8 Ensembles flous type-2 15

1.2.8.1 Définition d'un sous ensemble flou type-2 16

1.2.8.2 Type d'ensembles flous de type-2 18

1.2.8.2.1 Gaussien 18

1.2.8.2.2 Intervalle 18

1.2.8.2.3 Triangulaire 18

1.3 Logique floue en traitement d'images 19

1.3.1 Représentation de l'information spatiale 19

1.3.1.1 Objets flous spatiaux 19

1.3.1.2 Ensembles flous géométriques 20

1.3.1.3 Mesures géométriques d'objets flous 20

1.3.1.4 Transformations géométriques d'objets flous 21
1.3.2 Indice de flou d'une image 22
1.3.2.1 Indice de flou basé sur une distance 22
1.3.2.2 Indice de flou basé sur l'entropie 23
1.3.3 Seuillage flou 24
1.3.3.1 Degré d'appartenance en seuillage flou 24
1.3.4 Histogramme flou et matrices de cooccurrence floues 28
1.3.5 Morphologie mathématique floue 28
1.4 Conclusion 30
Chapitre 2 : L'algorithme Fuzzy C-Means et ses variantes
2.1 Introduction 31
2.2 Algorithme Fuzzy C-Means standard 31
2.2.1 Principe 31
2.2.2 Inconvénients de l'algorithme FCM 34
2.3 Variantes de l'algorithme FCM 34
2.3.1 Variantes selon la distance 35
2.3.1.1 Algorithme de Gustafson et Kessel 35
2.3.1.2 Algorithme de Gath et Geva 36
2.3.1.3 Algorithme de Shen et al 36
2.3.1.4 Algorithme de Wang et al 37
2.3.1.5 Algorithme de Tsai et Lin 38
2.3.1.6 Algorithme FCM basé sur des fonctions noyau 39
2.3.1.6.1 Fonction noyau 39
2.3.1.6.2 La version noyau de l'algorithme FCM 40
2.3.2 Variantes selon la fonction objectif 41

2.3.2.1 Algorithme de Frigui et Krishnapuram 41

2.3.2.2 Algorithme de Pham 42

2.3.2.3 Algorithme de C-moyennes possibiliste (PCM) 43

2.3.2.4 Algorithme de Wu et al 44

2.3.2.5 Algorithme d'Ahmed et al 44

2.3.2.6 Algorithme de Chen et Zhang 45

2.3.2.7 Algorithme de Szilágyi et al 46

2.3.2.8 Algorithme de Kouhi et al 46

2.3.2.9 Algorithme de Kang et al 47

2.3.2.10 Algorithme de Cai et al 48

2.3.2.11 Algorithme de Li et Mukaidono 49

2.3.2.12 Algorithme de Yang et al 50

2.3.2.13 Algorithme de Tang et al 50

2.3.2.14 Algorithme de Krinidis et Chatzis 51

2.3.2.15 Algorithme de Gauge et Sasi 52

2.3.2.16 Algorithme de Zhu et al 52

2.3.2.17 Algorithme de Zhao et Jiao 53

2.3.2.18 Algorithme de Ji et al 54

2.3.2.19 Algorithme de Benaichouche et al 55

2.3.3 Variantes selon la fuzzification 56

2.3.3.1 Algorithme de Rousseeuw et al 56

2.3.2.19 Algorithme de Klawonn et Hoppner 57

2.4 Conclusion 57

Chapitre 3 : Tests et résultats

3.1 Introduction 59

3.2 Stratégie de segmentation 59

3.3 Extraction des attributs de texture 60

3.3.1 Analyse de texture 61

3.4 Résultats 62

3.5 Images bruitées 66

3.6 Interprétation des résultats 71

3.7 Conclusion 72

Conclusion générale 73

Bibliographie

Introduction générale

L'information visuelle est sans doute la plus riche des différentes sources d'information existantes dans notre vie quotidienne. L'extraction de cette information par des systèmes de traitement et d'analyse d'images, a suscité un intérêt sans cesse croissant. Le traitement d'images qui est un processus comprenant plusieurs étapes, est né de la nécessité de remplacer l'observateur humain par la machine. Il intervient dans bon nombre de domaines tels que la médecine, l'agriculture et l'industrie.

Une étape incontournable dans tout processus d'analyse d'images est celle de la segmentation. En fournissant une description compacte de l'image, plus exploitable que l'ensemble des pixels, la segmentation d'images permet de faciliter l'interprétation automatique d'une image de façon similaire à une interprétation humaine. En effet, elle a été inspirée du système de perception visuel humain qui utilise la notion de similarité et de différence afin de localiser et de délimiter les objets d'une scène. Elle peut être définie ainsi : ″la segmentation d'images est un traitement de bas niveau qui consiste à créer une partition de l'image I en sous-ensembles R_i, appelés régions tel qu'aucune région ne soit vide, l'intersection entre deux régions soit vide et l'ensemble des régions recouvre toute l'image. Une région est un ensemble de pixels connexes ayant des propriétés communes qui les différencient des pixels des régions voisines″ [8], [9].

Cette tâche bien que relevée avec facilité par le système visuel humain, est en réalité complexe et reste un véritable défi pour la communauté du traitement d'images malgré plusieurs décennies de recherche. Ainsi, plusieurs méthodes de segmentation ont été proposées dans la littérature, et qui peuvent être classées selon trois grandes approches :

Approche région : cherche à regrouper les pixels ayant des propriétés similaires selon un certain critère d'homogénéité (niveau de gris, couleur ou texture) en régions connexes. Les techniques les plus connues de cette approche sont : la croissance de régions qui consiste à sélectionner au début du traitement des germes de régions qu'on fait croitre en leurs associant les pixels de leur voisinage qui vérifient un critère d'homogénéité. La segmentation par cette technique exploite correctement les informations topologique et contextuelle de l'image, mais son résultat dépend fortement de la sélection des germes et du critère de similarité [1], [2]. La division-fusion est une autre technique de l'approche région [5]. Elle permet de produire automatiquement une partition initiale de l'image en petites régions, qui vont ensuite, se croitre en se regroupant. La partition initiale (Split) est réalisée en divisant récursivement

l'image en régions de tailles identiques si un certain critère d'homogénéité n'est pas vérifié. La phase de regroupement (Merge) consiste à regrouper les régions ainsi produites en régions plus importantes et qui satisfont un autre critère d'homogénéité. La dernière technique à citer dans cette approche, est la classification. La classification qu'elle soit supervisée ou non, a toujours été un outil fort employé dans le domaine de l'analyse et du traitement des images numériques, en particulier à des fins de segmentation et d'interprétation [10]. En classification supervisée, on cherche à affecter une observation inconnue à une classe. Chaque classe est définie par quelques observations préalablement étiquetées (leur appartenance à la classe connue). Parmi ces méthodes, on peut citer la méthode des K plus proches voisins, les supports à vaste marge (SVM), etc.… La classification non supervisée consiste à partitionner un ensemble d'observations en différentes classes, en regroupant dans une même classe les observations ayant les mêmes caractéristiques. Dans le cas de la segmentation, une observation est assimilée à un pixel caractérisé par un ensemble d'attributs (couleur, texture,…).

Approche contour : Les contours se manifestent dans une image par une forte transition des valeurs d'intensité. La segmentation par l'approche contour, permet de détecter ces transitions en cherchant les points de discontinuité. Les méthodes dérivatives et les méthodes fondées sur les modèles déformables sont parmi les plus utilisées dans cette approche [4], [13]. Les méthodes dérivatives consistent à détecter les transitions par différentiation numérique, ce sont des méthodes locales qui balayent l'image avec un masque (convolution) et le résultat est une image binaire constituée de points contour et de points non contour. Parmi ces méthodes, les algorithmes les plus connus sont ceux basés sur le Gradient et ceux basés sur le Laplacien. Les méthodes fondées sur les modèles déformables, quant à elles, fournissent des contours fermés. Elles sont connues, beaucoup plus, sous le nom de contours actifs. Un contour actif est défini comme une courbe minimisant une énergie et évoluant de manière itérative à partir d'une position initiale proche de la zone d'intérêt, jusqu'à convergence (contour détecté).

Approche coopérative : La segmentation par coopération régions-contours consiste à combiner la segmentation par région et la segmentation par contour afin d'exploiter les avantages de ces deux approches et aboutir à un résultat de segmentation plus précis et meilleur que celui obtenu par une seule approche [6]. Ainsi, une segmentation par coopération région-contour peut être exprimée comme une entraide entre ces deux concepts afin d'améliorer le résultat final de la segmentation.

Nous nous intéressons dans ce mémoire à l'approche région de la segmentation d'images par classification non supervisée. Cependant, comme le processus d'acquisition de l'image, l'éclairage et la mesure d'attributs peuvent introduire des erreurs, il en résulte des régions homogènes floues avec des frontières ambigües. Pour tenir compte de ces incertitudes, on s'est intéressé à la classification floue dans laquelle un pixel appartient à une classe avec un certain degré. Ce degré d'appartenance appartient à l'intervalle [0, 1] et les classes obtenues ne sont pas forcément disjointes. Les algorithmes de classification floue les plus répandus tendent à minimiser une fonction cout, dépendant de la distance de chaque pixel aux prototypes (centres) des classes. Le prototype d'une classe étant un point connu dont l'appartenance à la classe est garantie.

Parmi ces algorithmes, l'algorithme Fuzzy C-Means (FCM), proposé par Bezdek [27], est le plus connu et a été utilisé dans bon nombre de domaines à cause de sa simplicité. Cependant, cet algorithme qui nécessite la connaissance du nombre de classes est jugé moins robuste face au bruit et son efficacité dépend fortement de l'étape d'initialisation des centres des classes car le processus itératif peut facilement fournir une solution localement optimale. De plus, il utilise généralement la distance euclidienne pour mesurer la similarité entre une observation et le centre d'une classe ce qui le rend utilisable que pour détecter des classes de forme sphérique [54]. Afin d'éviter ces inconvénients et donc améliorer les résultats de la classification, plusieurs modifications ont été apportées à l'algorithme standard et qui sont présentées comme des variantes du FCM. Les différentes variantes, auxquelles nous nous sommes intéressés dans ce mémoire, font face au problème du bruit. Nous les avons classées, dans notre travail, en trois catégories. Dans la première, la distance euclidienne utilisée par le FCM standard, est remplacée par un autre type de distance afin de détecter des classes de formes quelconques. Dans la deuxième catégorie, l'information spatiale a été introduite dans la fonction objectif sous forme d'un terme supplémentaire dans le but de prendre le bruit en considération et la troisième catégorie modifie l'influence du facteur flou.

La plupart de ces méthodes ont été développées dans le cadre de la segmentation d'images en niveaux de gris ou en couleur. Leurs applications pour la segmentation d'images texturées n'ont pas fait l'objet d'études. Ainsi, afin de tester l'apport de ces méthodes par rapport à l'algorithme FCM standard, nous avons effectué, dans ce mémoire, une étude comparative entre ce dernier avec certaines de ses variantes dans le cadre de la segmentation d'images texturées.

Le mémoire est principalement scindé en trois chapitres.

Dans le premier chapitre, nous avons abordé les notions générales sur les ensembles flous, en décrivant leurs caractéristiques, les opérations de base sur les ensembles flous et les propriétés de chacune des deux classes d'ensembles flous à savoir, les ensembles flous de type-1 et les ensembles flous de type-2. Pour clore ce chapitre, nous avons donné un bref aperçu sur l'utilisation de la logique floue en traitement d'images.

Le deuxième chapitre est dédié à l'algorithme FCM standard, à la présentation de ses inconvénients et à la description des différentes méthodes développées pour faire face à ses inconvénients. Le problème du nombre de classes et celui de l'initialisation des centres des classes ne sera pas abordé. Seules les méthodes permettant d'éviter le problème du bruit et celui de la distance sont présentées.

Le troisième chapitre est consacré à la description de la méthode de segmentation d'images texturées basée sur l'algorithme FCM et ses variantes. Les attributs de texture utilisés pour caractériser chaque pixel ainsi que les résultats de segmentation sont présentés.

1.1 Introduction

La logique floue apparaît comme une logique graduelle qui se veut être très proche de notre perception nuancée du monde. Elle a pour objectif de proposer des concepts, des techniques et des méthodes formellement rigoureuses pour représenter et traiter des connaissances et des données floues c'est-à-dire contenant de l'imprécision sur le contenu de l'information, de l'incertitude sur l'aspect vérité de l'information et de la subjectivité. D'autres synonymes tels que connaissance mal spécifiée, mal décrite, imparfaite, vague, qualitative, linguistique, partielle, incomplète, approximative ou approchée, recouvrent cette même acception du flou.

Dans ce chapitre, nous donnons un aperçu sur la théorie des sous ensembles flous en présentant ses concepts de base ainsi que les opérations les plus couramment utilisées. Nous donnons également un aperçu sur l'utilisation de la logique floue en traitement d'images.

1.2 Notions sur les sous ensembles flous

1.2.1 Introduction

La théorie des sous ensembles flous est en fait selon Zadeh [14], un pas vers un rapprochement entre la précision des mathématiques classiques et la subtile imprécision du monde réel. Le concept d'un sous ensemble flou, contrairement à l'approche classique, permet des graduations dans l'appartenance d'un élément à un sous ensemble (classe). Plusieurs types d'ensembles flous peuvent être considérés, lorsque les valeurs prises par les fonctions d'appartenance sont des nombres, de tels ensembles sont appelés de type 1. Cependant, les valeurs d'appartenance peuvent aussi être des ensembles flous, dans ce cas on parle d'un ensemble flou de type 2, dont les valeurs d'appartenance sont des ensembles flous de type 1.

D'une manière générale, un ensemble flou de type m est un ensemble flou dont les valeurs d'appartenance sont des ensembles flous de type $m-1$ (pour $m>1$). De telles extensions des ensembles flous sont utiles lorsque la valeur d'appartenance que l'on peut affecter à un élément, ne peut pas être définie de manière précise.

1.2.2 Définition d'un sous ensemble flou

Soit X un référentiel ou un espace de points également appelé univers de discours et soit x un élément de X. Un sous ensemble flou A de X est caractérisé par une fonction

d'appartenance $\mu_A(x)$ qui associe à chaque élément x de X un nombre réel dans l'intervalle [0,1] et qui représente le degré d'appartenance de cet élément à A.

Ainsi, le sous ensemble flou A de X est défini par :

$$A = \{(x, \mu_A(x)), x \in X\} \text{ avec } \mu_A : X \to [0,1] \ , \forall x \in X \quad (1.1)$$

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

- $\mu_A(x) = 0 \rightarrow$ non appartenance absolue de x à A.
- $\mu_A(x) = 1 \rightarrow$ appartenance absolue de x à A.
- $0 < \mu_A(x) < 1 \rightarrow$ appartenance graduée ou partielle de x à A..

Si A est un sous ensemble classique ou net, la fonction d'appartenance qui lui est associée ne prendra que deux valeurs extrêmes c'est-à-dire :

$$\mu_A(x) = \begin{cases} 0 \ si \ x \notin A \\ 1 \ si \ x \in A \end{cases} \quad (1.2)$$

1.2.3 Fonction d'appartenance

Dans un domaine discret $X = \{x_i / i = 1,2, \dots, n\}$ ou continu X, un sous ensemble flou A de X peut être défini par un ensemble de paires : degré d'appartenance/élément.

$$A = \mu_A(x_1) / x_1 + \mu_A(x_2) / x_2 + \cdots + \mu_A(x_n) / x_n \quad \text{cas discret}$$

Les fonctions d'appartenance les plus usuelles sont [15] :

- Trapézoïdale représentée sur la figure (1.1), elle est définie par :

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a-\alpha \text{ ou } x \geq b+\beta \\ x/\alpha + (1 - a/\alpha) & \text{si } a-\alpha < x < a \\ 1 & \text{si } a \leq x \leq b \\ -x/\beta + (1 + b/\beta) & \text{si } b < x < b+\beta \end{cases} \quad (1.3)$$

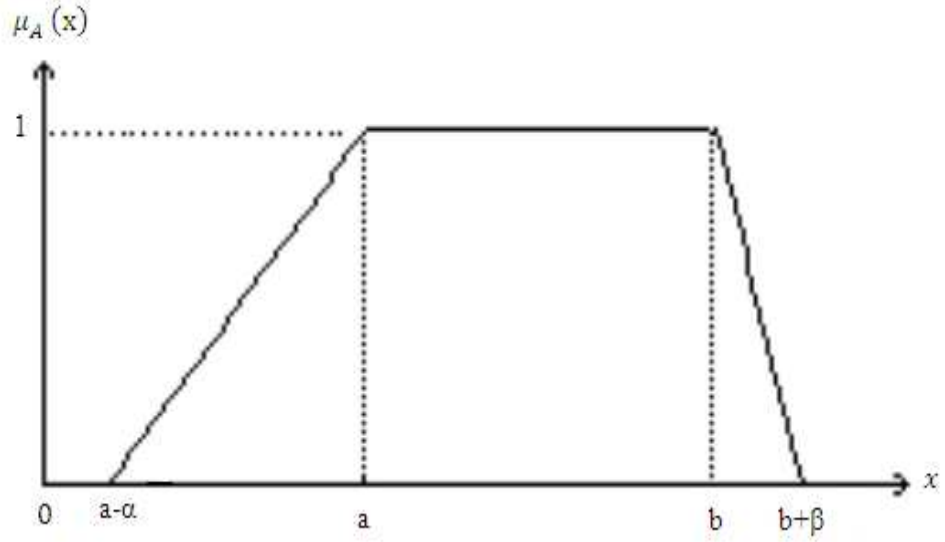

Figure (1.1): Fonction d'appartenance trapézoïdale.

- Triangulaire : elle constitue un cas particulier de la fonction trapézoïdale (Fig. 1.2)

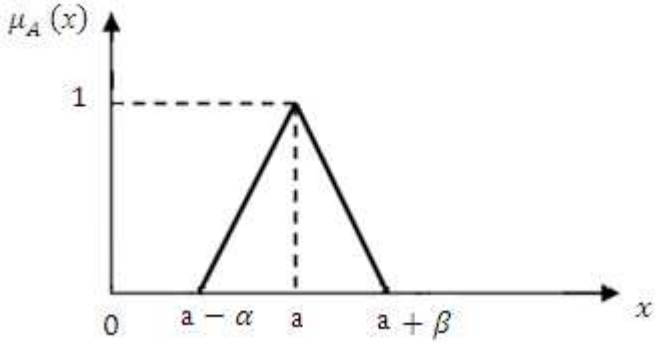

Figure (1.2) : Fonction d'appartenance triangulaire

- Sigmoïde : elle est définie par (Fig. 1.3) :

$$\mu_A(\mathrm{x}) = \begin{cases} 0 & \text{si } x < \mathrm{a} - \alpha \text{ ou } x > \mathrm{b}+\beta \\ 1 & \text{si } \mathrm{a} < x < \mathrm{b} \\ 2(x - \mathrm{a} + \alpha)^2/\alpha^2 & \text{si } \mathrm{a} - \alpha < x < \mathrm{a} - \alpha/2 \\ 1 + 2(x - \mathrm{a})^2/\alpha^2 & \text{si } \mathrm{a} - \alpha/2 < x < \mathrm{a} \\ 1 - 2(\mathrm{b} - x)^2/\beta^2 & \text{si } \mathrm{b} < x < \mathrm{b} + \beta/2 \\ 2(x - \mathrm{b} + \beta)^2/\beta^2 & \text{si } \mathrm{a} + \beta/2 < x < \mathrm{b} \end{cases} \quad (1.4)$$

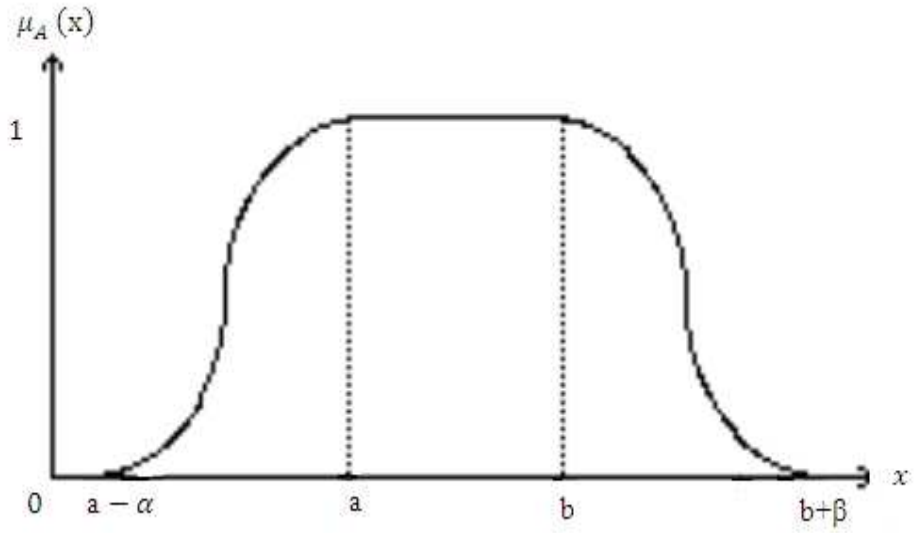

Figure (1.3) : Fonction d'appartenance sigmoïde

- Gaussienne : cette fonction d'appartenance est définie par (Fig. 1.4)

$$\mu_A(x) = \exp[-(x-m)^2 / 2\sigma^2] \quad (1.5)$$

où m est la moyenne de la gaussienne et σ sa variance.

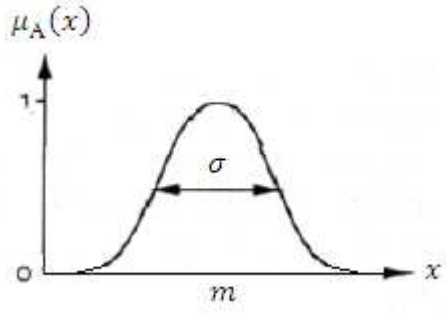

Figure (1.4) : Fonction d'appartenance gaussienne

- Singleton : elle est définie comme suit

$$\mu_A(x) = \begin{cases} 1 \text{ si } x = x_0 \\ 0 \quad \text{ailleurs} \end{cases} \quad (1.6)$$

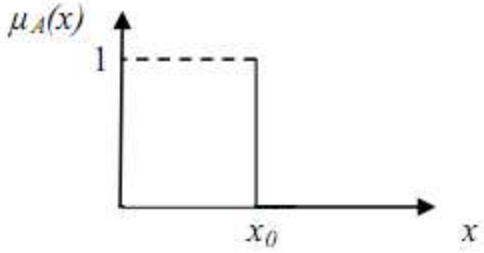

Figure (1.5) : Fonction d'appartenance singleton

1.2.4 Caractéristiques des sous ensembles flous

Un sous ensemble flou est complètement défini par la donnée de sa fonction d'appartenance. A partir d'une telle fonction, un certain nombre de caractéristiques du sous ensemble flou peuvent être définies [16].

1.2.4.1 La hauteur

Soit A un sous ensemble flou du référentiel X, la hauteur du sous ensemble flou A de X, notée $h(A)$ est le plus fort degré avec lequel un élément de X appartient à A. Formellement :

$$h(A) = sup_{x \in X} \left(\mu_A (x)\right) \tag{1.7}$$

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

1.2.4.2 Le support

Le support d'un sous ensemble flou A de X, noté par $Supp(A)$ est l'ensemble de tout les éléments qui lui appartiennent au moins un petit peu.

$$Supp(A) = \{x \in X / \mu_A (x) \neq 0\} \tag{1.8}$$

1.2.4.3 Le noyau

Le noyau d'un sous ensemble flou A de X, noté $N(A)$ est l'ensemble des éléments qui lui appartiennent avec un degré égal à 1.

$$N(A) = \{x \in X / \mu_A (x) = 1\} \tag{1.9}$$

Pour les fonctions d'appartenance triangulaires, le noyau est un singleton qui est appelé aussi valeur modale.

1.2.4.4 La cardinalité

La cardinalité du sous ensemble flou A de X, notée par $|A|$ est le nombre d'éléments appartenant à A pondéré par leur degré d'appartenance.

$$|A| = \sum_{x \in A} \mu_A (x) \tag{1.10}$$

Pour un sous ensemble classique, sa cardinalité est le nombre d'éléments qui le composent.

1.2.4.5 La α-coupe

Le sous ensemble ordinaire ou net A_α de X associé à A pour le seuil $\alpha \in [0,1]$, est l'ensemble des éléments qui appartiennent à A avec un degré au moins égal à α. On dit que A_α est la α-coupe de A.

$$A_\alpha = \{x \in X / \mu_A(x) \geq \alpha\} \quad (1.11)$$

$\mu_{A_\alpha}(x)$ prend ses valeurs dans {0,1}.

La figure (1.6) montre les différentes caractéristiques (noyau, support, α-coupe et la hauteur) d'un sous ensemble flou *A*

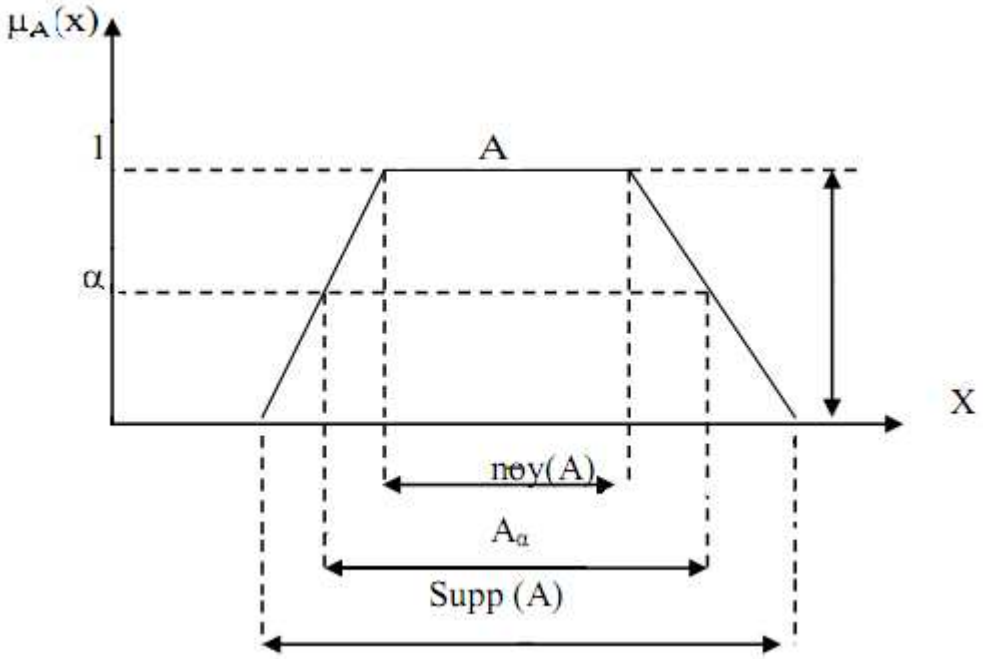

Figure (1.6) : Illustration des caractéristiques noyau, support, α-coupe et hauteur d'un sous ensemble flou.

1.2.5 Opérations ensemblistes sur les ensembles flous

Etant donné que le concept de sous ensemble flous est une extension du concept des ensembles classiques, on est amené à définir des opérations sur les sous ensembles flous qui sont équivalentes aux opérations classiques. Nous présenterons ici les opérations les plus couramment utilisées.

1.2.5.1 Complémentarité d'un sous ensemble flou

Le complémentaire d'un sous ensemble flou A noté $\bar{A}$ est défini par :

$$\forall x \in X,\ \mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{1.12}$$

Contrairement aux sous ensembles classiques, les propriétés de non contradiction $(A \cap \bar{A} = \emptyset)$ et du tiers exclu $(A \cup \bar{A} = X)$ ne sont pas respectées ici, c'est-à-dire qu'un fait et son complémentaire peuvent coexister dans la logique floue. Cependant d'autres propriétés sont conservées telle que $\overline{\bar{A}} = A, \overline{\emptyset} = X,\ \overline{X} = \emptyset$

1.2.5.2 Egalité et inclusion de deux sous ensembles flous

Soit A, B deux sous ensembles flous de X définis respectivement par leurs fonctions d'appartenance μ_A et μ_B. Les deux sous ensembles flous A et B sont égaux $(A = B)$ si et seulement si :

$$\forall x \in X, \mu_A(x) = \mu_B(x) \tag{1.13}$$

De même, le sous ensemble flous A est inclus dans le sous ensemble flous B, si la fonction d'appartenance de A prend des valeurs inferieures ou égales à la fonction d'appartenance de B, ceci pour tous les points x de X :

$$A \subseteq B \text{ si et seulement si } \forall x \in X, \mu_A(x) \leq \mu_B(x) \tag{1.14}$$

1.2.5.3 Union et Intersection de deux sous ensembles flous

L'union de deux sous ensembles flous A et B de X est le sous ensemble flou constitué des éléments de X affectés du plus grand des degrés avec lesquels ils appartiennent à A et B (Fig. 1.7). Formellement $A \cup B$ est défini par

$$\forall x \in X, \mu_{A \cup B}(x) = max\left(\mu_A(x), \mu_B(x)\right) \tag{1.15}$$

L'intersection de deux sous ensembles flous A et B de X est le sous ensemble flou $A \cap B$ défini par la fonction d'appartenance :

$$\mu_{A \cap B}(x) = min\,(\mu_A(x), \mu_B(x)), \forall x \in X \tag{1.16}$$

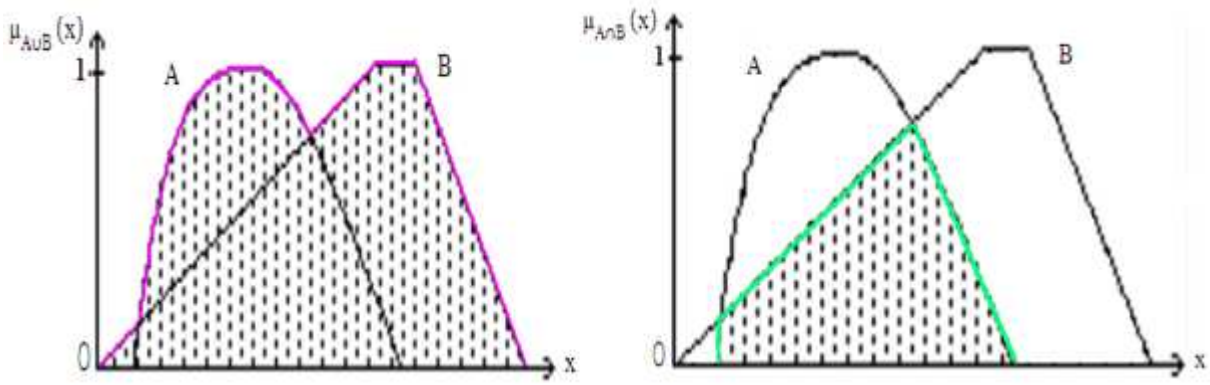

Figure (1.7) : Union et Intersection de deux sous ensembles flous.

Certaines propriétés de la théorie des ensembles classiques sont vérifiées, telles que :

- Commutativité : $A \cap B = B \cap A$ et $A \cup B = B \cup A$
- Associativité : $A \cup (B \cup C) = (A \cup B) \cup C$ et $A \cap (B \cap C) = (A \cap B) \cap C$
- Idempotence : $A \cup A = A$ et $A \cap A = A$
- Distributivité : $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$et $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- Relations de Morgan : $\overline{A \cup B} = \overline{A} \cap \overline{B}$ et $\overline{A \cap B} = \overline{A} \cup \overline{B}$
- Lois d'absorption : $A \cup (A \cap B) = A \cap (A \cup B) = A$
- $A \cap \emptyset = \emptyset$ et $A \cup X = X$
- Identité : $A \cup \emptyset = A$ et $A \cap X = X$
- Cardinalité : $|A| + |B| = |A + B|$
- Formule d'équivalence : $(\overline{A} \cup B) \cap (A \cup \overline{B}) = (\overline{A} \cap \overline{B}) \cup (A \cap B)$
- Formule de la différence symétrique : $(\overline{A} \cap B) \cup (A \cap \overline{B}) = (\overline{A} \cup \overline{B}) \cap (A \cup B)$

Les opérateurs min et max sont généralement choisis pour définir l'intersection et l'union des sous ensembles flous, car ils préservent presque tous les résultats issus de la théorie classique des ensembles. Cependant, il existe d'autres opérateurs pour traduire l'intersection et l'union floues. Il s'agit de deux opérateurs les plus connus : la norme triangulaire (**t-norme**) pour l'intersection et la conorme triangulaire (**t-conorme**) pour l'union.

1.2.5.4 t-norme et t-conorme triangulaires

Une norme triangulaire (t-norme T) est une fonction à deux variables x et y définies dans l'intervalle [0,1] et qui vérifie les propriétés suivantes :

T : $[0,1] \times [0,1] \rightarrow [0,1]$ telle que :

- T est commutative : $\forall(x,y) \in [0,1]^2, T(x,y)=T(y,x)$
- T est associative : $\forall(x,y,z) \in [0,1]^3, T\big(x,T(y,z)\big) = T(T(x,y),z)$
- 1 est un élément neutre : $T(x,1) = T(1,x) = x$

Comme exemples de fonctions t-norme, on peut citer :

- Intersection standard (Zadeh) : $T(x,y) = \min(x,y)$
- Produit algébrique (Intersection probabilistique) : $T(x,y) = xy$
- Intersection Lukasiewicz : $T(x,y) = \max(x+y-1,0)$

De façon analogique**,** la t-conorme $\perp$ est définie par :

$\perp : [0,1] \times [0,1] \to [0,1]$ tel que $\forall x,y,z \in [0,1]$

- $\perp(x,y) = \perp(y,x)$ (commutativité)
- $\perp\big(x,\perp(y,z)\big) = \perp(\perp(x,y),z)$ (associativité)
- $\perp(x,0) = x$ (l'élément neutre)

Les fonctions t-conorme les plus utilisées sont :

- Zadeh : $\perp(x,y) = \max(x,y)$
- Probabiliste : $\perp(x,y) = x+y-xy$
- Lukasiewicz : $\perp(x,y) = \min(x+y,1)$

1.2.6 Produit cartésien de deux sous ensembles flous

Il est utilisé pour combiner l'information provenant de plusieurs ensembles de référence.

Soit X_1, X_2 deux univers de référence et X leur produit cartésien, $X = X_1 \times X_2$ dont les éléments sont les couples (x_1,x_2) avec $x_1 \in X_1, x_2 \in X_2$.

Soit A_1, A_2 deux sous ensembles flous définis respectivement sur X_1 et X_2, le produit cartésien $A = A_1 \times A_2$ est un sous ensemble flou de X tel que :

$$\forall x \in X,\ x = (x_1,x_2),\ \mu_A(x) = \min\left(\mu_{A_1}(x_1), \mu_{A_2}(x_2)\right) \tag{1.17}$$

1.2.7 Structure générale d'un traitement flou

Un système flou est un système à base de connaissances, sa structure générale est illustrée dans la figure (1.8). Il se compose essentiellement de quatre modules à savoir :

- Fuzzification : elle consiste à attribuer à la valeur réelle (valeur numérique) de chaque entrée, au temps t, sa fonction d'appartenance à chacune des classes préalablement définies donc transformer l'entrée réelle en un sous ensemble flou. C'est l'interface ‹‹ numérique-flou (linguistique) ››
- Base de règles : c'est une base de connaissances qui contient les règles floues décrivant le comportement d'un système. L'ensemble des règles se présente comme suit :

  R=***Si*** condition 1et/ou condition 2 et/ou …***Alors*** action sur les sorties

  Exemple de règle utilisée quotidiennement :

  Si feu rouge et ***Si*** vitesse-véhicule élevée et ***Si*** feu proche ***Alors*** freinage fort
- Mécanisme d'inférence : il transforme à l'aide des opérateurs flous, la partie floue issue de la fuzzification en une nouvelle partie floue conformément aux règles floues et une méthode d'inférence choisie (Mamdani, Larsen, Takagi-Sugeno).

  La conclusion finale du système d'inférence est le résultat de la combinaison des résultats des différentes règles en utilisant les normes triangulaires, afin de déterminer la fonction d'appartenance résultante globale.
- Défuzzification : c'est une interface ‹‹ linguistique-numérique ›› qui transforme la partie floue issue du raisonnement flou en valeurs numériques.

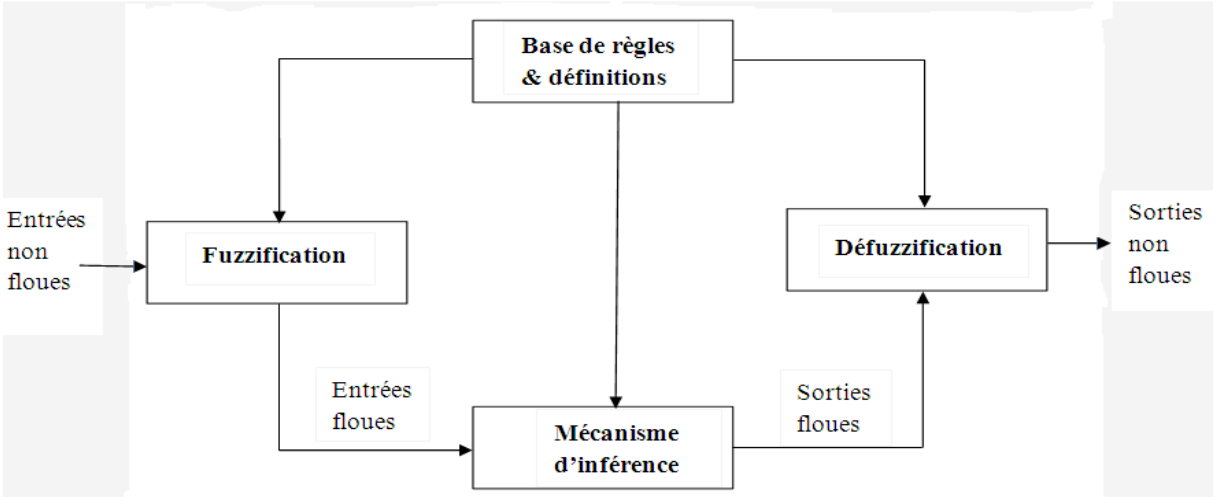

Figure (1.8) **:** Structure générale d'un traitement flou

1.2.8 Ensembles flous type-2

Les systèmes flous employant les ensembles flous type-1 sont utilisés pour traiter et manipuler les informations imprécises et incertaines, en les représentant par des nombres dans l'intervalle [0,1]. Quand quelque chose est incertaine, comme la mesure, il est difficile de déterminer sa valeur exacte, alors il n'est pas raisonnable d'utiliser une fonction d'appartenance précise pour quelque chose d'incertain. Dans ce cas, ce que dont nous avons besoin est un autre type d'ensembles flous capables de manipuler et de prendre en compte ces incertitudes [17].

Ainsi une nouvelle classe d'ensembles flous, appelée ensembles flous type-2, a été introduite par Zadeh en 1975. Dans ce nouveau type d'ensembles flous, les degrés d'appartenance sont considérés eux même comme des ensembles flous type-1. Ils sont très efficaces dans les circonstances où il nous est difficile de déterminer exactement les fonctions d'appartenance pour les ensembles flous.

Actuellement, on considère que la sortie d'un système flou de type-1 correspond à la valeur moyenne d'une fonction de densité de probabilité, c'est-à-dire que le calcul de la défuzzification pour un système flou de type-1 est équivalent au calcul de la moyenne d'une fonction densité de probabilité.

La théorie des probabilités est utilisée pour modéliser l'incertitude aléatoire, dans laquelle la fonction de densité de probabilité interprète la totalité des informations concernant les incertitudes aléatoires. Dans la plupart des applications pratiques, il est impossible de connaitre ou de déterminer la fonction de densité de probabilité, ainsi on est obligé d'admettre le fait que cette dernière sera complètement caractérisée par tous ses moments. Il n'est pas très pratique et efficace d'utiliser uniquement le moment d'ordre 1, car les incertitudes aléatoires exigent la connaissance de la dispersion autour de la valeur moyenne, et cette information est fournie par la variance. Par conséquent, nous devons utiliser au moins ces deux premiers moments de la fonction de densité de probabilité.

La variance nous fournit une mesure de dispersion autour de la valeur moyenne, et elle est généralement utilisée pour capturer plus d'informations sur les incertitudes statistiques.

Par conséquent, les systèmes flous ont aussi besoin d'une certaine mesure de dispersion pour leur permettre de capturer plus d'incertitude concernant les règles. Les ensembles flous de type-2 permettent de prendre en charge cette mesure de dispersion.

1.2.8.1 Définition d'un sous ensemble flou de type-2

Si nous prenons l'exemple d'une fonction d'appartenance de type-1 triangulaire, (Fig. 1.9), et imaginons que nous la brouillons en faisant décaler les points du triangle à gauche ou à droite (pas nécessairement de la même quantité) comme il est illustré dans la figure (1.10). Alors, pour une valeur spécifique x′, la fonction d'appartenance prend différentes valeurs, partout où la ligne verticale coupe le nuage de points. Ces valeurs n'ont pas la même pondération, d'où nous pouvons attribuer une distribution d'amplitude pour tous ces points [17].

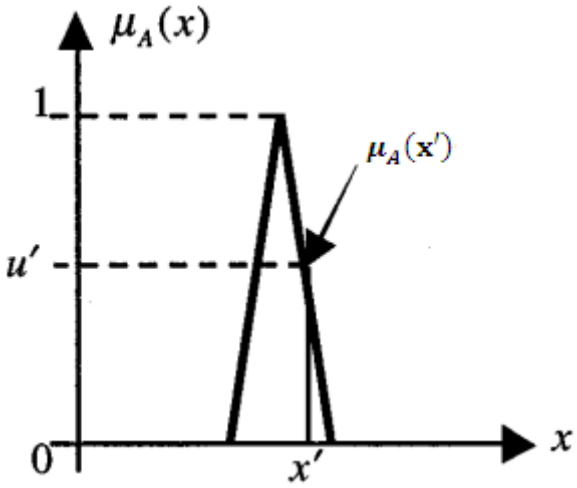

Figure (1.9) **:** Fonction d'appartenance de type-1

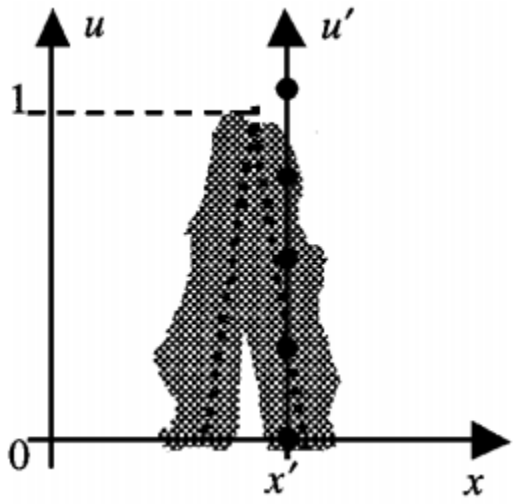

Figure (1.10) **:** Fonction d'appartenance type-1 brouillée (atténuée)

Faisons ça pour tout x ∈ X, nous créons ainsi, une fonction d'appartenance tridimensionnelle (la fonction d'appartenance de type-2) qui caractérise un sous ensemble flou de type-2.

Exemple

La figure suivante, montre une fonction d'appartenance de type-2 pour x et u discrets. $X = \{1, 2, 3, 4, 5\}$ et $U = \{0, 0.2, 0.4, 0.6, 0.8\}$

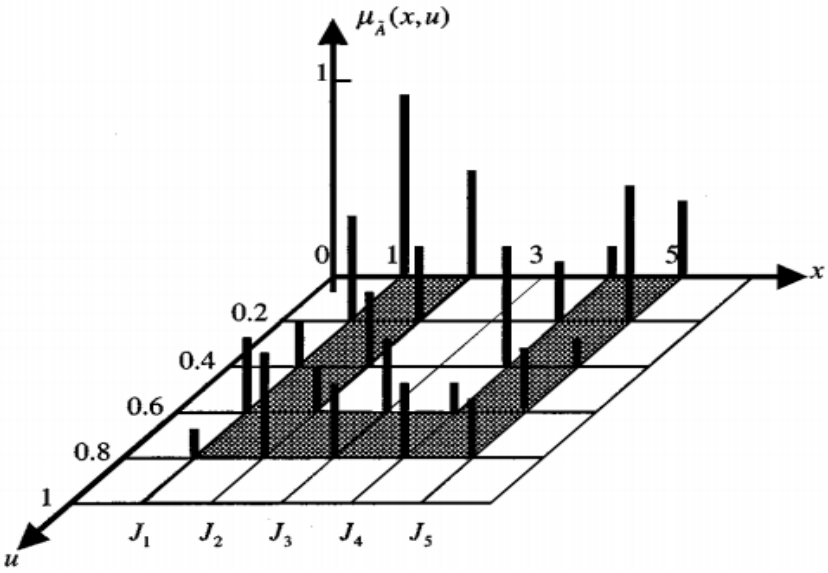

Figure (1.11) **:** Fonction d'appartenance de type-2

Définition 1 :

Un ensemble flou de type-2 est caractérisé par une fonction d'appartenance de type-2 $\mu_{\tilde{A}}(x, u)$, où $x \in X$ et $u \in J_x \subseteq [0,1]$ tel que :

$$\tilde{A} = \{((x, u),\ \mu_{\tilde{A}}(x, u))/\forall x \in X, \forall u \in J_x \subseteq [0,1]\} \quad (1.18)$$

avec $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$

$\tilde{A}$ peut être exprimé aussi de la manière suivante :

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u), \quad J_x \subseteq [0,1] \quad (1.19)$$

Pour un univers discret $\int$ est remplacé par $\sum$

J_x représente les degrés d'appartenance primaires de x

Définition 2 :

Pour chaque valeur de x, $x = x'$ correspond un plan 2-D dont les axes sont u et $\mu_{\tilde{A}}(x', u)$, cette dernière représente la fonction d'appartenance secondaire (contient les degrés d'appartenance des degrés d'appartenance primaires dans $\mu_{\tilde{A}}(x)$, de x à $\tilde{A}$), tel que :

$$\mu_{\tilde{A}}(x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) / u \,, J_{x'} \subseteq [0,1] \quad (1.20)$$

où $0 \leq f_{x'}(u) \leq 1$

Prenons l'exemple de la figure (1.11), la fonction d'appartenance secondaire pour $x' = 2$ est $\mu_{\tilde{A}}(2) = 0.5/0 + 0.35/0.2 + 0.35/0.4 + 0.2/0.6 + 0.5/0.8$

1.2.8.2 Types d'ensembles flous de types-2

1.2.8.2.1 Gaussien

Dans ce type d'ensembles, le degré d'appartenance de chaque point est un ensemble flou de type-1 gaussien dont le domaine de définition est inclus dans l'intervalle [0,1]. Notons qu'il n'est pas nécessaire que la fonction d'appartenance principale soit aussi gaussienne.

1.2.8.2.2 Intervalle

Dans ce type d'ensembles, toutes les appartenances secondaires sont égales à 1. La figure (1.12) montre l'exemple d'un tel ensemble.

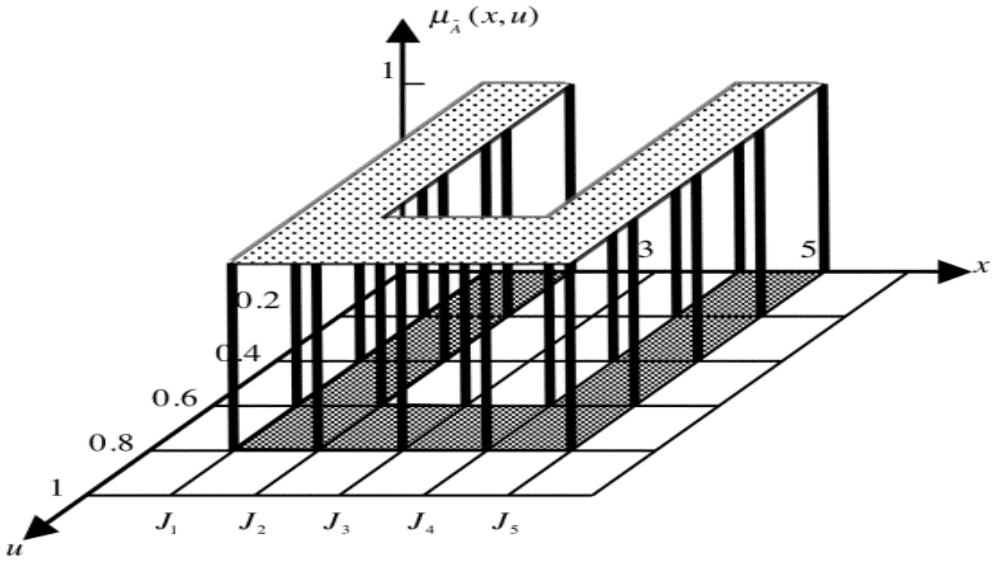

Figure (1.12) **:** Exemple d'une fonction d'appartenance type-2 intervalle

1.2.8.2.3 Triangulaire

Dans ce type d'ensembles, le degré d'appartenance de chaque point est un ensemble flou de type-1 triangulaire dont le domaine de définition est inclus dans l'intervalle [0,1].

1.3 Logique floue en traitement d'images

Imprécision et incertitude sont inhérentes aux images, et trouvent leur origine à différents niveaux : phénomènes observés, capteurs et acquisitions de données, processus numériques de reconstruction, nature des images et modes de représentation de leurs éléments constitutifs, etc.

L'intérêt des ensembles flous pour le traitement de l'information en image et vision peut se décliner en particulier selon les quatre aspects suivants :

- La capacité des ensembles flous à représenter l'information spatiale dans les images ainsi que son imprécision, à différents niveaux (local, régional ou global), et sous différentes formes (numérique, symbolique, qualitative, quantitative).
- La possibilité de représenter des informations très hétérogènes, extraites directement des images ou issues de connaissances externes, comme des connaissances expertes ou génériques sur un domaine ou un problème.
- La possibilité de généraliser aux ensembles flous des opérations pour manipuler l'information spatiale.
- La souplesse des opérateurs de combinaison permettant de fusionner des informations de multiples natures dans des situations très variées.

Ce paragraphe se propose d'illustrer quelques uns de ces aspects. Nous définissons des objets flous spatiaux permettant de manipuler des objets avec leur imprécision spatiale dans les images. Des objets géométriques sont définis, ainsi que des mesures et des transformations géométriques [19].

1.3.1 Représentation de l'information spatiale

1.3.1.1 Objets flous spatiaux

Un objet flou spatial est un ensemble flou défini dans le domaine de l'image. Sa fonction d'appartenance μ représente l'imprécision sur la définition spatiale de l'objet (position, taille, forme, limites, etc.). Pour tout pixel x de l'image, $\mu(x)$ représente le degré avec lequel x appartient à l'objet flou.

1.3.1.2 Ensembles flous géométriques

Des objets flous particuliers peuvent être définis, tels que des points, droites, rectangles et disques flous.

- Points flous : les points flous sont des extensions dans l'espace à n dimensions de nombres flous considérés comme des points sur la droite réelle.
- Droites floues : des droites floues ont été définies dans un espace bidimensionnel par Buckley et Eslami [24]. Des extensions dans des espaces de dimension supérieure sont immédiates en suivant le même principe qui considère que les paramètres d'une forme analytique de droite classique sont des nombres flous et non pas des nombres classiques. La fonction d'appartenance d'une droite floue est alors calculée de manière similaire au principe d'extension. Ainsi, si une droite du plan est définie analytiquement sous la forme $ax + by = c$, on définit une droite floue par sa fonction d'appartenance μ_D comme suit :

  $$\mu_D(x,y) = sup\{\alpha \in [0,1] | ax + by = c, a \in \mu_{A_\alpha}, b \in \mu_{B_\alpha}, c \in \mu_{C_\alpha}\} \quad (1.21)$$

  où μ_A, μ_B et μ_C sont les fonctions d'appartenance des trois nombres flous.
- Rectangle flou : un rectangle flou est un ensemble flou séparable et convexe. La séparabilité signifie que la fonction d'appartenance s'exprime comme la conjonction de fonctions sur chaque axe.

1.3.1.3 Mesures géométriques d'objets flous

- Surface ou volume **:** s'exprime par le cardinal de l'ensemble flou.
- Périmètre **:** le périmètre d'un objet flou de fonction d'appartenance μ est défini par

  $$P(\mu) = \int_{x \in S} |\nabla \mu(x)| \, dx \quad (1.22)$$

Dans le cas discret 2D, cette expression revient à calculer la somme pondérée des longueurs des courbes délimitant les α-coupes, les poids représentent la différence des valeurs d'appartenance entre deux α-coupes successives.

- Compacité : dans le cas 2D, la compacité est donnée par

  $$C(\mu) = \frac{V(\mu)}{P(\mu)^2} \quad (1.23)$$

  V désigne le volume (dans le cas 2D, surface) et P le périmètre.

Les mesures géométriques données dans ce paragraphe, sont définies comme nombres nets. Or, si les objets sont imprécis, leurs mesures seront également imprécises et donc définies comme des nombres flous pouvant être calculés selon le principe d'extension. Pour une mesure M (surface, périmètre, etc.) sa forme floue s'exprime par

$$\forall \lambda \in \mathbb{R}^+, M(\mu)(\lambda) = sup_{\,M(\mu_\alpha)=\lambda} \alpha \tag{1.24}$$

1.3.1.4 Transformations géométriques d'objets flous

Des transformations géométriques d'ensembles flous spatiaux, telles que des translations ou des rotations, peuvent être définies.

- Transformations nettes : dans ce cas on suppose que seul l'ensemble à transformer est flou. Soit $\mathcal{T}$ une transformation géométrique quelconque (translation, rotation symétrie, etc.) dont les paramètres sont des nombres usuels et soit μ un ensemble flou dans S, la transformation de μ par $\mathcal{T}$ est alors :

  $$\forall x \in S, \mathcal{T}(\mu)[\,\mathcal{T}(x)] = \mu(x) \tag{1.25}$$

  ou encore

  $$\forall x \in S, \mathcal{T}(\mu)(x) = \mu[\,\mathcal{T}^{-1}(x)] \tag{1.26}$$

- Transformations floues : on considère maintenant que la transformation est également imprécise et que ses paramètres sont des nombres flous. Soit $\mathcal{T}$ une transformation floue dépendant des paramètres flous $p_1, \dots, p_2$. Soit μ_{p_i} les fonctions d'appartenance définissant les paramètres p_i. La transformation d'un point y de S est un ensemble flou défini par la fonction d'appartenance suivante :

  $$\forall z \in S, \mu_{\mathcal{T}(y)}(z) = \sup_{p_1,\dots p_n | \mathcal{T}_{p_1,\dots p_n}(y)=z} t\big[\mu_{p_1}(p_1), \dots \mu_{p_n}(p_n)\,\big] \tag{1.27}$$

  où t est une t-norme et $\mathcal{T}_{p_1,\dots p_n}$ est une transformation nette obtenue pour des valeurs précises des paramètres.

  En étendant cette définition au cas d'un point appartenant à un ensemble flou μ, on obtient la transformation de μ définie par :

  $$\forall x \in S, \mathcal{T}(\mu)(x) = \sup_{y \in S} t\big[\mu(y), \mu_{\mathcal{T}(y)}(x)\,\big] \tag{1.28}$$

La logique floue a été largement utilisée en traitement d'images. Outre la classification floue, à travers l'algorithme FCM, qui sera abordée en détail dans le prochain chapitre, elle a été notamment très utilisée en segmentation d'images et plus particulièrement dans le seuillage d'images [4]. Les notions d'histogramme flou et de matrices de cooccurrence floues

ont été introduites pour définir une texture floue. La morphologie mathématique a également su profiter de la logique floue où des opérations d'érosion et de dilatation floues par exemple ont été définies [19].

1.3.2 Indice de flou d'une image

Une image en niveaux de gris X, de dimension $M \times N$, est définie dans le cas flou par :

$$X \equiv \mu_X(x_{mn}) = \sum_{m \epsilon M} \sum_{n \in N} \mu_{mn} / x_{mn} \tag{1.29}$$

où $\mu_X(x_{mn})$ représente le degré de brillance du pixel (m, n) d'intensité x_{mn}.

Pour évaluer l'ambigüité associée aux niveaux de gris présents dans l'image X, on utilise souvent une mesure de la valeur d'un indice de flou. Cette mesure prend en compte la distance entre l'ensemble X et l'ensemble ordinaire $\tilde{X}$ le plus proche.

Cet ensemble ordinaire est donné sous cette forme :

$$\mu_{\tilde{X}}(x_{mn}) = \begin{cases} 1 \; si \; \mu_X(x_{mn}) \geq 0.5 \\ 0 \; si \; \mu_X(x_{mn}) < 0.5 \end{cases} \tag{1.30}$$

1.3.2.1 Indice de flou basé sur une distance

L'indice de flou d'un ensemble A ayant n points est défini par :

$$\gamma(A) = \frac{2}{n^k} d\left(A, \tilde{A}\right) \tag{1.31}$$

avec $d\left(A, \tilde{A}\right)$, la distance entre A et le sous ensemble ordinaire $\tilde{A}$ le plus proche. $\tilde{A}$ est donné par l'équation (1.30) et k est une constante positive de normalisation dont la valeur dépend du type de la distance utilisée.

- **Distance de Hamming ou linéaire**

  La distance de Hamming ou linéaire entre deux sous ensembles flous A_1 et A_2 sur X est définie par :

  $$d_l(A_1, A_2) = \sum_{x \in X} |A_1(x) - A_2(x)| \tag{1.32}$$

- **Distance euclidienne ou quadratique**

La distance quadratique entre deux sous ensembles flous A_1 et A_2 sur X est définie par :

$$d_q(A_1, A_2) = \left(\sum_{x\in X}\left(A_1(x) - A_2(x)\right)^2\right)^{1/2} \quad (1.33)$$

Plusieurs indices de flou ont été développés selon plusieurs distances existantes. La distance euclidienne équivaut à utiliser $k = 0.5$ dans l'équation (1.31) et la distance de Hamming équivaut à utiliser $k = 1$. Les indices de flou d'un ensemble A correspondant à ces distances sont :

$$\gamma_l(A) = \frac{2}{n} d_l\left(A, \tilde{A}\right)$$

$$= \frac{2}{n}\sum_{x\in X}\left|\mu_A(x) - \mu_{\tilde{A}}(x)\right| \quad (1.34)$$

$$= \frac{2}{n}\sum_{x\in X}\mu_{A\cap\tilde{A}}(x)$$

$$\gamma_q(A) = \frac{2}{n^{1/2}} d_q\left(A, \tilde{A}\right)$$

$$= \frac{2}{n^{1/2}}\left(\sum_{x\in X}\left(\mu_A(x) - \mu_{\tilde{A}}(x)\right)^2\right)^{1/2} \quad (1.35)$$

Dans le cas d'une image X de dimension $M \times N$, ces fonctions se traduisent par :

$$\gamma_l(X) = \frac{2}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\mu_{X\cap\tilde{X}}(x_{ij}) = \frac{2}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|\mu_X(x_{ij}) - \mu_{\tilde{X}}(x_{ij})\right| \quad (1.36)$$

$$\gamma_q(X) = \frac{2}{(M\times N)^{1/2}}\left(\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(\mu_X(x_{ij}) - \mu_{\tilde{X}}(x_{ij})\right)^2\right)^{1/2} \quad (1.37)$$

1.3.2.2 Indice de flou basé sur l'entropie

Pour évaluer l'ambiguïté associée à un ensemble flou, il est possible d'utiliser la mesure d'information classique d'entropie adaptée au cas flou. L'incertitude de la nature floue dans un ensemble flou A peut être mesurée par :

$$HF(A) = \frac{1}{n\ ln2}\sum_{x\in X} S\left(\mu_A(x)\right)$$

$$= \frac{1}{n\ ln2} \sum_{x \in X} \left[\mu_A(x)\ ln \mu_A(x) + \left(1 - \mu_A(x)\right)\ ln\left(1 - \mu_A(x)\right) \right] \quad (1.38)$$

où S est appelée fonction de Shannon.

Dans le cas d'une image X de dimension $M \times N$, cette fonction s'écrit par :

$$HF(X) = \frac{1}{MN\ ln2} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} S\left(\mu_X\left(x_{ij}\right)\right) \quad (1.39)$$

1.3.3 Seuillage flou

Dans les techniques de segmentation par seuillage, l'objectif est de déterminer le meilleur seuil (s) permettant d'extraire les objets de l'image en affectant chaque pixel de l'image à la classe fond ou à la classe objet. En raisonnant avec la logique floue, le but est de trouver la valeur seuil qui minimise l'incertitude associés à une image. Cette incertitude peut être déterminée par l'un des indices de flou présentés ci-dessus. Pour effectuer un seuillage flou il faut évidemment définir le degré d'appartenance d'un niveau de gris à la classe fond (b) ou objet (f).

1.3.3.1 Degré d'appartenance en seuillage flou

Les degrés d'appartenance d'un niveau de gris aux classes fond ou objet peuvent être définis de plusieurs manières. La formule qui revient le plus souvent dans la littérature est (Fig. 1.13) :

$$\mu_b(g) = \begin{cases} 1 & si\ g \leq a \\ 1 - \frac{(g-a)^2}{(c-a)(b-a)} & si\ a \leq g \leq b \\ \frac{(g-c)^2}{(c-a)(c-b)} & si\ b < g \leq c \\ 0 & si\ g \geq c \end{cases} \quad (1.40)$$

$$\mu_f(g) = \begin{cases} 0 & si\ g \leq a \\ \frac{(g-a)^2}{(c-a)(b-a)} & si\ a < g \leq b \\ 1 - \frac{(g-c)^2}{(c-a)(c-b)} & si\ b < g \leq c \\ 1 & si\ g > c \end{cases} \quad (1.41)$$

avec bien sur $\mu_b(g) + \mu_f(g) = 1$

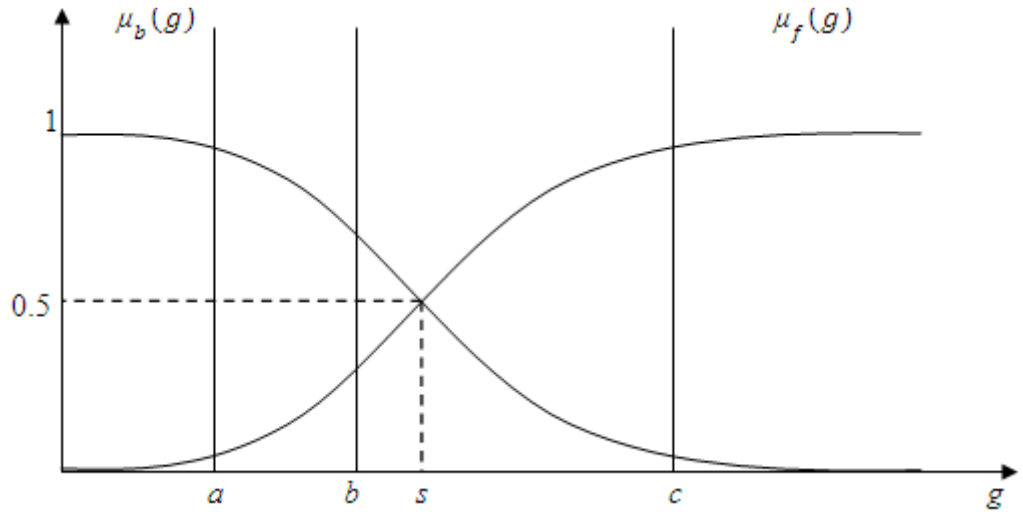

Figure (1.13) : Courbe des degrés d'appartenance

Le seuil (s) est généralement compris entre a et c et correspond à la valeur de g qui vérifie $\mu_b(g) = \mu_f(g) = 0.5$.

Cheng et al [21] utilisent la fonction suivante (Fig. 1.14) :

$$\mu_b(g) = \begin{cases} 0 & si \quad g \leq a \\ \frac{g-a}{c-a} & si\ a < g < c \\ 1 & si \quad c \leq g \end{cases} \tag{1.42}$$

$$\mu_f(g) = \begin{cases} 1 & si \quad g \leq a \\ \frac{g-c}{a-c} & si\ a < g < c \\ 0 & si \quad c \leq g \end{cases} \tag{1.43}$$

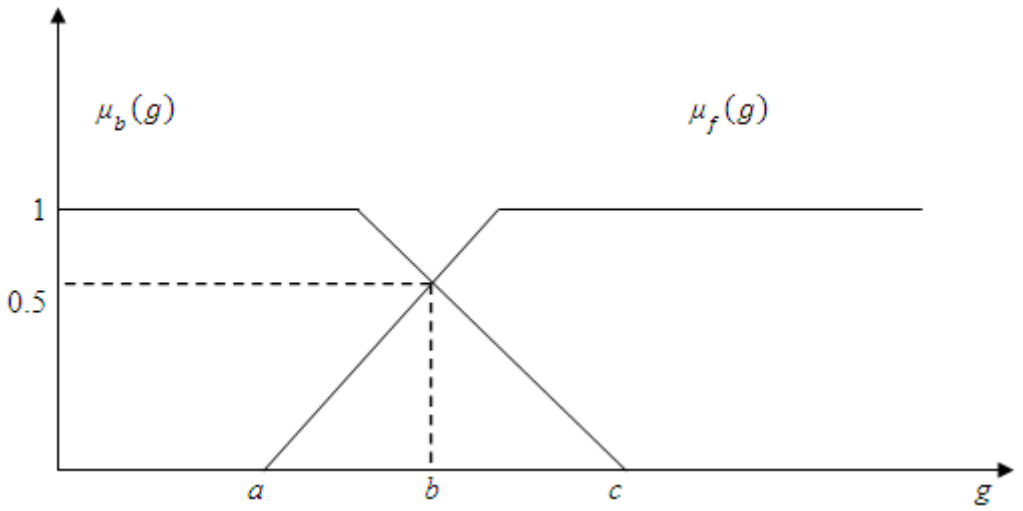

Figure (1.14) : Courbe des degrés d'appartenance

Le point de croisement b qui correspond à $\mu_b(g) = \mu_f(g)$ donne le seuil, $s = b = \frac{c+a}{2}$. Le problème revient alors à déterminer les valeurs a et c afin de pouvoir déduire s.

Les degrés d'appartenance d'un niveau de gris aux classes fond et objet peuvent être également définis à partir de fonction S_z de *Zadeh.* Cette fonction est définie de manière que, pour tout élément g dans un intervalle $[a, b]$ inclus dans $[0, N_G]$, la valeur de S_z soit dans l'intervalle [0,1]

$$S_z(g) = \begin{cases} 0 & si\ g \le a \\ 2\left[\frac{g-a}{c-a}\right]^2 & si\ a \le g \le b \\ 1 - \left[\frac{g-c}{c-a}\right]^2 & si\ b \le g \le c \\ 1 & si\ g \ge c \end{cases} \tag{1.44}$$

Le point $b = (c + a)/2$ pour lequel S_z possède la valeur 0.5 est appelé point de croisement et la valeur de$(c - a)$est dite largeur de bande. En notant par s le point de croisement et en posant $|c - a| = 2k$ pour la largeur de bande, la fonction de Zadeh S_z peut alors s'écrire en fonction du paramètre k comme suit (Fig. 1.15) :

$$S_z(g) = \begin{cases} 0 & si\ g \le s - k \\ \left[\frac{g-(s-k)}{2k^2}\right]^2 & si\ s - k \le g \le s \\ 1 - S_z(2s - g) & si\ s \le g \le s + k \\ 1 & si\ \ g \ge s + k \end{cases} \tag{1.45}$$

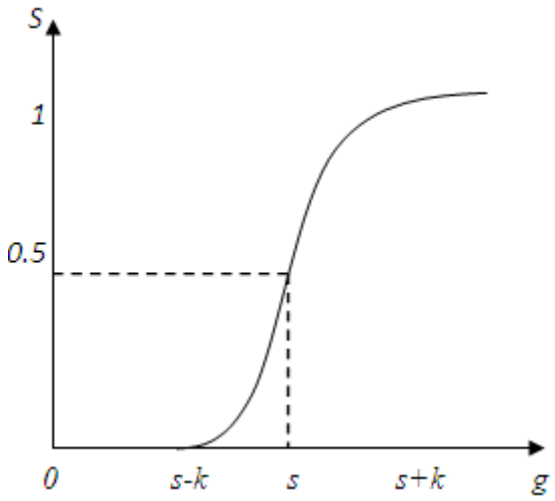

Figure (1.15): Fonction S_z de Zadeh

Les degrés d'appartenance de tous les niveaux de gris aux deux classes (fond et objet) sont :

$$\mu_b(g) = \begin{cases} 0 & si\ g < s-k \\ S_z(s,g) & si\ s-k \le g \le s+k \\ 1 & si\ g > s+k \end{cases} \quad (1.46)$$

$$\mu_f(g) = \begin{cases} 1 & si\ g < s-k \\ 1 - S_z(s,g) & si\ s-k \le g \le s+k \\ 0 & si\ g > s+k \end{cases} \quad (1.47)$$

Le seuil, s correspond à $\mu_b(g) = \mu_f(g) = 0.5$ est situé entre $s-k$ et $s+k$. Le choix de k est primordial, si l'intervalle $[s-k, s+k]$ est trop petit, il est possible que des seuils non représentatifs soient détectés, par contre si cet intervalle est trop grand, des bons seuils peuvent être éliminés. On peut alors procéder au calcul de ce paramètre comme pour a, b et c précédents puis déduire le seuil s ou le fixer puis déterminer la valeur de ce dernier.

Une autre manière de déterminer le degré d'appartenance d'un niveau de gris a été proposée par Huang et Wang [22]

$$\mu_f(g) = \begin{cases} \frac{1}{1+\frac{|g-m_1(s)|}{D}} & si\ g \le s \\ \frac{1}{1+\frac{|g-m_2(s)|}{D}} & si\ g > s \end{cases} \quad (1.48)$$

avec

$$m_1(s) = \frac{\sum_{g=0}^{s} g.h(g)}{\sum_{g=0}^{s} h(g)}, m_2(s) = \frac{\sum_{g=s+1}^{N_G-1} g.h(g)}{\sum_{g=s+1}^{N_G-1} h(g)} \quad (1.49)$$

et

$$D = g_{max} - g_{min} = N_G - 1 \text{ si } g_{max} = N_G - 1 \text{ et } g_{min} = 0.$$

g_{max} et g_{min} sont respectivement les niveaux de gris max et min.

Dans [20] et [23] les degrés d'appartenance d'un niveau de gris g sont :

$$\mu(g) = \begin{cases} e^{-c|g-m_1|} & si\ g \leq s \\ e^{-c|g-m_2|} & si\ g > s \end{cases} \quad (1.50)$$

avec m_1 et m_2 sont définis dans l'équation (1.49), et c est une constante de normalisation telle que : $c = \frac{1}{g_{max} - g_{min}}$

1.3.4 Histogramme flou et Matrices de cooccurrence floues

En considérant les valeurs des niveaux de gris comme des nombres flous, des statistiques floues d'ordre un et deux (histogramme et matrices de cooccurrence) ont été proposées [18] dans le but de représenter au mieux la distribution spatiale des niveaux de gris dans l'image.

L'histogramme flou est une séquence de nombres réels q_n $(n \in \{0,1, \dots, L-1\})$, où q_n désigne le nombre de pixels ayant des valeurs autour du niveau de gris n et L le nombre de niveaux de gris. En considérant la valeur du niveau de gris $I(x,y)$ comme un nombre flou $\tilde{I}(x,y)$, l'histogramme flou peut être calculé comme suit :

$$q_k \leftarrow q_k + \sum_x \sum_y \mu_{\tilde{I}(x,y)k}\,, \qquad k \in [a,b] \quad (1.51)$$

avec $a = \min\left(k|\mu_{\tilde{I}(x,y)k} > 0\right)$ et $b = \max\left(k|\mu_{\tilde{I}(x,y)k} > 0\right)$

La matrice de cooccurrence floue $F\left[f_{ij}^d\right]$, qui représente la fréquence d'occurrence d'une valeur autour du niveau de gris i suivie d'une valeur autour du niveau de gris j, peut être calculée par l'équation :

$$f_{ij}^d \leftarrow f_{ij}^d + \sum_{\varphi_{mn}} \min\left(\mu_{\tilde{m}i}, \mu_{\tilde{n}j}\right), \quad i \in [a_1,b_1],\ j \in [a_2,b_2],\ \forall \varphi_{mn} \quad (1.52)$$

avec $a_1 = \min(x|\mu_{\tilde{m}x} > 0)$, $b_1 = \max(x|\mu_{\tilde{m}x} > 0)$, $a_2 = \min(x|\mu_{\tilde{n}x} > 0)$, $b_2 = \max(x|\mu_{\tilde{n}x} > 0)$ et φ_{mn} est l'ensemble de toutes les paires de pixels avec des niveaux de gris $\tilde{m}$ et $\tilde{n}$ séparés d'une distance d.

1.3.5 Morphologie mathématique floue

La morphologie est une théorie largement employée en traitement d'images binaires et numériques. Elle propose des transformations non linéaires utilisées en filtrage et amélioration d'images, segmentation, mesures et reconnaissance de formes. Elle se base sur

les concepts de la théorie des ensembles, de topologie et de géométrie. Un autre intérêt de l'extension de la morphologie mathématique au cas flou est que d'autres opérateurs peuvent en être déduits (adjacence ou distance par exemple).

Nous allons tout d'abord nous rappeler les définitions des opérations de base de la morphologie mathématique à savoir la dilatation, l'érosion, l'ouverture et la fermeture. Soit X (ou f) l'ensemble (ou la fonction) à transformer et B (ou g) l'élément structurant ensembliste (ou fonctionnel).

La dilatation dans l'espace S est définie par :

$$D_B(X) = X \oplus B = \{x \in S | B_x \cap X \neq \emptyset\} \tag{1.53}$$

où B_x désigne le translaté de B au point x

$$D_B(f)(x) = sup\{f(y) | y \in B_x\} \tag{1.54}$$

$$D_g(f)(x) = sup\{f(y) + g(y-x) | y \in S\} \tag{1.55}$$

L'érosion est définie par :

$$E_B(X) = X \ominus B = \{x \in S | B_x \subset X\} \tag{1.56}$$

$$E_B(f)(x) = inf\{f(y) | y \in B_x\} \tag{1.57}$$

$$E_g(f)(x) = inf\{f(y) - g(y-x) | y \in S\} \tag{1.58}$$

L'ouverture et la fermeture sont données par :

$$X_B = D_{\check{B}}[E_B(X)] \tag{1.59}$$

$$X^B = E_{\check{B}}[D_B(X)] \tag{1.60}$$

où $\check{B}$ est le symétrique de B par rapport à l'origine.

Les constructions de la morphologie mathématique floue reposent sur des principes assez classiques d'extension d'opérations binaires au cas flou, soit à partir du principe d'extension ou par combinaison d'opérations appliquées sur les α-coupes, soit par traduction directe d'équations ensemblistes en équations floues.

- A partit des α-coupes : soient μ et v deux ensemble flous définis sur S, la dilatation et l'érosion de μ par v sont données par les équations suivantes :

$$\forall x \in S, D_v(\mu)(x) = \int_0^1 sup_{y\in(v_\alpha)_x}\mu(y)d\alpha \quad (1.61)$$

$$\forall x \in S, E_v(\mu)(x) = \int_0^1 inf_{y\in(v_\alpha)_x}\mu(y)d\alpha \quad (1.62)$$

où $(v_\alpha)_x$ est la translation de v_α(α-coupe de v). Ces définitions assurent que $D_v(\mu)$ et $E_v(\mu)$ sont les fonctions d'appartenance d'ensembles flous (prenant leurs valeurs dans $[0,1]$).

- Par traduction formelle à partir de t-normes et de t-conormes : cette méthode consiste à remplacer formellement, dans les équations définissant les opérations, les symboles binaires ou logiques par leurs équivalents flous, telle que :

$$\forall x \in S, D_v(\mu)(x) = sup_{y\in S}t[v(y-x), \mu(x)] \quad (1.63)$$

$$\forall x \in S, E_v(\mu)(x) = inf_{y\in S}T\big[c\big(v(y-x)\big), \mu(x)\big] \quad (1.64)$$

où t est la t-norme et T est la t-conorme duale de t par rapport à la complémentation.

1.4 Conclusion

Nous avons présenté dans la première partie de ce chapitre, quelques notions de base sur la logique floue, en introduisant les concepts de base des sous ensembles flous de type-1 et en expliquant le rôle de chaque bloc constituant les systèmes flous de type-1.

Nous avons présenté dans la deuxième partie de ce chapitre un concept fondamental sur les sous ensemble flous de type-2 caractérisés par des fonctions d'appartenance dont les degrés d'appartenance sont eux même flous.

A la fin de ce chapitre, quelques exemples d'application de la logique floue en traitement d'images ont été présentés.

2.1 Introduction

L'algorithme FCM est une méthode de classification non supervisée très populaire et très performante par rapport aux autres méthodes comme l'algorithme K-means ou les réseaux de neurones [26].

Il a été abondamment utilisé pour la segmentation d'images [25].

Malgré cela, l'algorithme FCM souffre de certains inconvénients comme sa sensibilité au bruit, la forme des classes qui est sphérique, la nécessité de connaitre le nombre de classes ainsi que la dépendance de ses résultats à l'initialisation. Pour surmonter ces inconvénients plusieurs variantes du FCM, que nous décrirons dans ce chapitre, ont été proposées

2.2 Algorithme Fuzzy C-means standard

Le Fuzzy C-means est un algorithme de classification non supervisée, issu de l'algorithme K-means. Son apport par rapport à ce dernier est l'introduction de la notion du flou, afin de prendre en compte l'imprécision des données [28]. Développé par Bezdek en 1981 à la suite des travaux de Dunn [27], l'algorithme FCM est un algorithme de réallocation floue, dans lequel les classes sont représentées par des prototypes (centres de gravité). Son application fournit donc pour chaque observation à classifier un degré d'appartenance (compris entre 0 et 1) à chaque classe, produisant ainsi une partition floue. Comme pour la plus part des autres algorithmes de classification par partition, le FCM est basé sur la minimisation d'un critère en suivant un processus itératif.

2.2.1 Principe

Soit $X = \{x_1, x_2, \dots, x_i, \dots, x_N\}$ un ensemble de N observations à classifier en K classes où chaque observation x_i $(i = 1,2, \dots, N)$ peut être représentée par un ensemble de D attributs : $x_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iD})^T$.

Les K classes sont représentées par un vecteur des centres des classes $V = \{v_1, v_2, \dots, v_k, \dots, v_K\}$ où $v_k = (v_{k1}, v_{k2}, \dots, v_{kD})^T$ est le centre de la classe k.

Chaque observation x_i est définie par son degré d'appartenance μ_{ki} à la classe k tel que $\mu_{ki} \in [0\ 1]$. On peut alors définir une matrice de partition $U = [\mu_{ki}]$ de dimension$(K \times N)$.

L'algorithme FCM consiste à partitionner les N observations en K classes de manière à minimiser la similarité des observations à l'intérieur de chaque classe. Il se traduit par la minimisation de la fonction objectif suivante :

$$J_{FCM}(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N} {\mu_{ki}}^{m} d^2(\mathrm{x}_i, v_k) \tag{2.1}$$

Sous les contraintes suivantes:

$$0 < \sum_{i=1}^{N} \mu_{ki} < N \tag{2.2}$$

$$\sum_{k=1}^{K} \mu_{ki} = 1 \tag{2.3}$$

m : est le facteur de fuzzification ou facteur flou tel que $1 < m < \infty$.

$d(\mathrm{x}_i, v_k) = \|\mathrm{x}_i - v_k\|$ est la distance séparant l'observation i et le centre de la classe k.

Généralement la distance utilisée est euclidienne

$$d(\mathrm{x}_i, v_k) = \sqrt{\sum_{j}^{D} \left(\mathrm{x}_{ij} - v_{kj}\right)^2} \tag{2.4}$$

La première contrainte assure qu'aucune classe ne doit être vide et la seconde est une contrainte de normalisation qui assure que la somme des degrés d'appartenance de chaque observation à toutes les classes soit égale à 1.

Le partitionnement flou est réalisé par une optimisation itérative de la fonction objectif donnée par l'équation (2.1), avec une mise à jour des degrés d'appartenance μ_{ki} et des centres des classes v_k comme dans le cas de l'algorithme K-means.

Les formules de mise à jour sont obtenues par l'introduction d'un multiplicateur de Lagrange λ associé à la contrainte de normalisation donnée dans l'équation (2.3).

En appliquant le Lagrangien par rapport à x_i, on aura :

$$L(\mathrm{x}_i) = \sum_{k=1}^{K} {\mu_{ki}}^{m} \|\mathrm{x}_i - v_k\|^2 + \lambda\left(\sum_{k=1}^{K} \mu_{ki} - 1\right), \qquad \lambda > 0 \tag{2.5}$$

La minimisation du Lagrangien par rapport aux degrés d'appartenance μ_{ki} et au coefficient de Lagrange λ nous donne :

$$\frac{\partial L(\mathrm{x}_i)}{\partial \mu_{ki}} = 0 \implies m({\mu_{ki}}^{m-1})\|\mathrm{x}_i - v_k\|^2 + \lambda = 0 \tag{2.6}$$

$$\frac{\partial L(\mathrm{x}_i)}{\partial \lambda} = 0 \implies \sum_{k=1}^{K} \mu_{ki} - 1 = 0 \implies \sum_{k=1}^{K} \mu_{ki} = 1 \tag{2.7}$$

A partir de l'équation (2.6), on déduit :

$$\mu_{ki} = \left(\frac{-\lambda}{m}\right)^{\frac{1}{m-1}} \left(\frac{1}{\|\mathrm{x}_i - v_k\|^2}\right)^{\frac{1}{m-1}} \tag{2.8}$$

et à partir de l'équation (2.7) et (2.8), on obtient :

$$\left(\frac{-\lambda}{m}\right)^{\frac{1}{m-1}} \sum_{l=1}^{K} \left(\frac{1}{\|\mathrm{x}_i - v_l\|^2}\right)^{\frac{1}{m-1}} = 1$$

$$\Leftrightarrow \left(\frac{-\lambda}{m}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{l=1}^{K} \left(\frac{1}{\|\mathrm{x}_i - v_l\|^2}\right)^{\frac{1}{m-1}}}$$

$$\Leftrightarrow \frac{-\lambda}{m} = \frac{1}{\sum_{l=1}^{K} \left(\frac{1}{\|\mathrm{x}_i - v_l\|^2}\right)} \tag{2.9}$$

En remplaçant l'équation (2.9) dans l'équation (2.8), on aboutit à :

$$\mu_{ki} = \left(\frac{1}{\sum_{l=1}^{K} \left(\frac{1}{\|\mathrm{x}_i - v_l\|^2}\right)}\right)^{\frac{1}{m-1}} \left(\frac{1}{\|\mathrm{x}_i - v_k\|^2}\right)^{\frac{1}{m-1}} \Leftrightarrow \mu_{ki} = \frac{1}{\sum_{l=1}^{K} \left(\frac{\|\mathrm{x}_i - v_k\|^2}{\|\mathrm{x}_i - v_l\|^2}\right)^{\frac{1}{m-1}}}^{2}$$

D'où la formule de mise à jour des degrés d'appartenance :

$$\mu_{ki} = \sum_{l=1}^{K} \left(\frac{\|\mathrm{x}_i - v_k\|}{\|\mathrm{x}_i - v_l\|}\right)^{\frac{-2}{m-1}} \tag{2.10}$$

La minimisation du Lagrangien par rapport à la variable v_k représentant le centre de la classe k, s'écrit comme suit :

$$\frac{\partial L(\mathrm{x}_i)}{\partial v_k} = 0 \Leftrightarrow \sum_{i=1}^{N} 2\|\mathrm{x}_i - v_k\| {\mu_{ki}}^m = 0$$

$$\Leftrightarrow \sum_{i=1}^{N} {\mu_{ki}}^m \mathrm{x}_i - \sum_{i=1}^{N} {\mu_{ki}}^m v_k = 0$$

D'où la formule de mise à jour des centres de classes :

$$v_k = \frac{\sum_{i=1}^{N} {\mu_{ki}}^m \mathrm{x}_i}{\sum_{i=1}^{N} {\mu_{ki}}^m} \tag{2.11}$$

L'algorithme FCM peut être résumé comme suit :

1. Fixer les paramètres :
    a. Le nombre de classes K
    b. Le seuil ε représentant l'erreur de convergence
    c. Le degré flou m, généralement pris égal à 2
2. Initialiser les centres des K classes de manière aléatoire.
3. Mettre à jour la matrice U des degrés d'appartenance par la relation (2.10)
4. Mettre à jour le vecteur V des centres des classes par la formule (2.11)
5. Répéter les étapes 3 et 4 jusqu'à satisfaction du critère d'arrêt : $\left\|V^{(t)} - V^{(t+1)}\right\| < \varepsilon$, t étant la $t^{ième}$ itération.

2.2.2 Inconvénients de l'algorithme FCM

L'algorithme FCM a été largement étudié et a été utilisé dans bon nombre de domaines (segmentation d'images médicales, géologiques et satellitaires). Cependant cet algorithme qui nécessite la connaissance du nombre de classes, n'est pas robuste face aux bruits introduits par l'imprécision des attributs et son efficacité dépend fortement de l'étape d'initialisation des centres des classes car le processus itératif peut facilement fournir une solution localement optimale. De plus, il est basé sur la distance euclidienne lors de la mesure de similarité entre une observation et le centre d'une classe ce qui le rend utilisable que pour détecter des classes de forme sphérique. Afin d'éviter ces inconvénients donc améliorer les résultats de la classification, plusieurs modifications ont été apportées à l'algorithme standard et qui sont présentées comme des variantes du FCM. Celles que nous présentons à la suite prennent en compte le problème du bruit et celui de la distance.

2.3 Variantes de l'algorithme FCM

De nombreuses variantes ont été ainsi proposées, celles-ci consistent soit à changer la fonctionnelle à minimiser, soit à définir une autre distance ou soit encore à modifier l'influence du facteur flou. Nous les avons ainsi organisées en trois catégories :

- Variante selon la distance
- Variante selon la fonction objectif

- Variante selon la fuzzification

2.3.1 Variantes selon la distance

La version standard de FCM décrite précédemment utilise la distance euclidienne. Cependant cette distance donne de bons résultats que lorsque les classes sont de forme sphérique et ayant la même taille, ou lorsqu'elles sont bien séparées. Or en réalité les classes peuvent avoir des formes géométriques quelconques et des tailles différentes. Ainsi, plusieurs variantes du FCM utilisant d'autres types de distances ont été proposées.

2.3.1.1 Algorithme de Gustafson et Kessel

Gustafson et Kessel [29] ont proposé une variante du FCM dans laquelle la distance de Mahalanobis est utilisée à la place de la distance euclidienne. Dans cet algorithme, chaque classe est représentée, non seulement par son centre de gravité v_k mais aussi par sa matrice de covariance, notée A_k de dimension $(D \times D)$. Chaque matrice A_k est carrée, définie positive dont le déterminant $\rho_k > 0$ est fixé a priori.

Lorsque les tailles des classes sont connues à l'avance, elles peuvent être contrôlées en utilisant la constante $\rho_k > 0$ exigeant que $det(A_k) = \rho_k$. Généralement, on suppose que les classes ont des tailles égales, en mettant $det(A_k) = 1$.

La distance entre une observation x_i et le centre d'une classe v_k prend la forme suivante :

$$d_{ki}^2 = \|\mathrm{x}_i - v_k\|_{A_k}^2 = (\mathrm{x}_i - v_k)^T A_k (\mathrm{x}_i - v_k) \quad (2.12)$$

La fonction objectif utilisée par Gustafson et Kessel est donc la suivante :

$$J_{GK}(U,V) = \sum_{k=1}^{K} \sum_{i=1}^{N} {\mu_{ki}}^m \|\mathrm{x}_i - v_k\|_{A_k}^2 \quad (2.13)$$

La formule permettant le calcul des differents centres reste inchangée (équation (2.11)). Celle relative aux degrés d'appartenance (équation (2.10)) est légèrement modifiée en faisant intervenir les matrices A_k pour le calcul des distances et non plus la matrice unité correspondant à la distance euclidienne. Ces matrices sont également remises à jour à chaque itération par la formule suivante :

$$A_k = \left[\rho_k \left(det(S_k)\right)\right]^{\frac{1}{D}} S_k^{-1} \quad (2.14)$$

S_k représente la matrice de covariance floue de la $k^{iéme}$ classe, définie par :

$$S_k = \frac{\sum_{i=1}^{N} \mu_{ki}{}^m (\mathrm{x}_i - v_k)(\mathrm{x}_i - v_k)^T}{\sum_{i=1}^{N} \mu_{ki}{}^m} \quad (2.15)$$

2.3.1.2 Algorithme de Gath et Geva

Gath et Geva ont proposé une variante de FCM dans le but de détecter des classes hyper-ellipsoïdales qui peuvent avoir différentes formes, tailles et densités [30]. Ils proposent pour cela d'utiliser une distance exponentielle $d_e(\mathrm{x}_i, v_k)$ entre chaque observation x_i et le centre v_k de chaque classe au lieu de la distance euclidienne.

Cette distance est définie par la relation suivante :

$$d_e^2(x_i, v_k) = \frac{[det\, S_k]^{1/2}}{P_k} exp[(x_i - v_k)^T S_k^{-1} (x_i - v_k)/2] \quad (2.16)$$

$P_k = \frac{1}{N}\sum_{i=1}^{N} \mu_{ki}$ représente la probabilité a priori de la $k^{ième}$ classe, et S_k la matrice floue de covariance de la $k^{ième}$classe identique à celle définie dans l'équation (2.15).

Le calcul des degrés d'appartenance se fait de la manière suivante :

$$\mu_{ki} = \frac{1/d_e^2(x_i, v_k)}{\sum_{l=1}^{K} 1/d_e^2(x_i, v_l)} \quad (2.17)$$

L'algorithme de Gath et Geva est semblable à celui du FCM standard mais comporte deux étapes supplémentaires qui sont le calcul de probabilité à priori P_k et des matrices floues de covariance S_k.

2.3.1.3 Algorithme de Shen et al

Dans le cadre de la segmentation d'images en niveaux de gris, Shen et al [31] ont utilisé une version de l'algorithme FCM dont la fonction distance est :

$$d^2(x_i, v_k) = \|x_i - v_k\|^2 (1 - \lambda H_{ki} - \xi F_{ki}) \quad (2.18)$$

H_{ki} représente un paramètre d'attraction et F_{ki} une distance d'attraction. Ces deux paramètres sont calculés en tenant en compte du voisinage de chaque pixel de la manière suivante :

$$H_{ki} = \frac{\sum_{r=1}^{N_i} \mu_{kr} g_{ir}}{\sum_{r=1}^{N_i} g_{ir}} \quad (2.19)$$

$$F_{ki} = \frac{\sum_{r=1}^{N_i} \mu_{kr}^2 q_{ir}^2}{\sum_{r=1}^{N_i} g_{ir}} \quad (2.20)$$

N_i étant le nombre de pixels voisins du $i^{ième}$ pixel.

$g_{ir} = |x_i - x_r|$ est la différence absolue des niveaux de gris des pixels i et r.

$q_{ir} = \sqrt{(a_i - a_r)^2 + (b_i - b_r)^2}$ est la distance spatiale entre les pixels i et r.

(a_i, b_i), (a_r, b_r) étant les coordonnées des pixels i et r respectivement.

λ et ξ sont des nombres réels qui permettent d'ajuster la contribution de chaque terme d'attraction : $0 < \lambda < 1$ et $0 < \xi < 1$. Forouzanfar et al [32] ont récemment proposé trois algorithmes d'optimisation métaheuristique basés sur les algorithmes génétiques (GA), l'essaim des particules (PSO) et l'essaim d'abeilles (BS) pour déterminer d'une manière optimale les valeurs de ces paramètres.

La mise à jour des centres de classes et des degrés d'appartenance se fait de la même manière que dans l'algorithme FCM mais en utilisant la distance définie par la relation (2.18) au lieu de la distance euclidienne lors du calcul des degrés d'appartenance.

2.3.1.4 Algorithme de Wang et al

Wang et al [33] ont utilisé une distance entre l'observation i et le centre de la classe k dans laquelle la contribution de chaque attribut j $(j = 1, \dots, D)$ est prise en compte :

$$d_{in}^{(w)} = \sqrt{\sum_{j=1}^{D} w_j^2 \, (x_{ij} - x_{nj})^2} \tag{2.21}$$

w_j est un poids qui pondère la contribution de chaque attribut j.

Pour déterminer le vecteur W, il suffit de minimiser l'expression suivante :

$$E(w) = \frac{2}{W(W-1)} \sum_{i=1}^{N} \sum_{\substack{n=1 \\ i \neq n}}^{N} \frac{1}{2} \left[\rho_{in}^{(w)} (1 - \rho_{in}) + \rho_{in} \left(1 - \rho_{in}^{(w)} \right) \right] \tag{2.22}$$

où $\rho_{in}^{(w)} = \frac{1}{1 + \beta d_{in}^{(w)}}$ est une mesure de similarité entre deux observations i et n.

β étant une constante positive.

Pour $w_j = 1 \; \forall j = 1, \dots, D, \rho_{ij}^{(w)} = \rho_{ij}$

La minimisation de $E(w)$ peut se faire par la méthode du gradient :

$$w_j^{(t)} = w_j^{(t-1)} + \Delta w_j \quad (2.23)$$

avec $\Delta w_j = -\eta \frac{\partial E(W)}{\partial w_j}$ (2.24)

La fonction objectif devient donc :

$$J_W(U,V) = \sum_{k=1}^{K} \sum_{i=1}^{N} {\mu_{ki}}^m \left(\sqrt{\sum_{j=1}^{D} w_j^2 \, (v_{kj} - x_{ij})^2} \right)^2 \quad (2.25)$$

En minimisant J_W sous la contrainte (2.3), nous obtenons les formules de mise à jour des μ_{ki} et v_k comme suit :

$$\mu_{ki} = \frac{1}{\sum_{l=1}^{K} \left(\frac{d_{ki}^{(w)}}{d_{li}^{(w)}} \right)^{2/(m-1)}}, m \neq 1 \quad (2.26)$$

$$v_k = \frac{\sum_{i=1}^{N} {\mu_{ki}}^m \mathrm{x}_i}{\sum_{i=1}^{N} {\mu_{ki}}^m}, \forall k \quad (2.27)$$

2.3.1.5 Algorithme de Tsai et Lin

Afin d'améliorer l'efficacité de l'algorithme FCM, une nouvelle mesure de distance différente de celle de Mahalanobis et qui prend en compte la variation des distances dans chaque classe a été proposée par Tsai et Lin [34]. Cette distance s'exprime comme :

$$\hat{d}_{ik}^2 = \frac{\|x_i - v_k\|^2}{\sigma_k}, i = 1, \dots, N \text{ et } k = 1, \dots, K \quad (2.28)$$

où σ_k est la moyenne pondérée des distances dans la classe k, définie par :

$$\sigma_k = \left[\frac{\sum_{i=1}^{N} \mu_{ki}^m \|x_i - v_k\|^2}{\sum_{i=1}^{N} \mu_{ki}^m} \right]^{1/2} \quad (2.29)$$

La fonction objectif ainsi proposée est alors :

$$J_{TL}(U,V) = \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki}^m \frac{\|x_i - v_k\|^2}{\sigma_k} \quad (2.30)$$

Les équations des degrés d'appartenance et des centres des classes sont :

$$\mu_{ki} = \frac{1}{\sum_{l=1}^{K} \left(\frac{\hat{d}_{ik}}{\hat{d}_{il}} \right)^{2/(m-1)}}, m \neq 1 \quad (2.31)$$

$$v_k = \frac{\sum_{i=1}^{N} {\mu_{ki}}^m x_i}{\sum_{i=1}^{N} {\mu_{ki}}^m} \ , \forall k \tag{2.32}$$

Cette distance a été appliquée sur l'algorithme FCM classique pour la classification dans l'espace de données et l'algorithme FCM basé sur les fonctions noyau (voir paragraphe ci-dessous). Les deux versions de FCM utilisant cette distance ont été appliquées à la segmentation d'images couleurs.

2.3.1.6 Algorithme FCM basé sur des fonctions noyau

2.3.1.6.1 Fonction noyau

Les fonctions noyaux sont devenues, durant ces dernières années, très populaires et largement utilisées en classification de données. L'idée générale, est de supposer que l'espace des données peut être plongé dans un espace de plus haute dimension (via une transformation non linéaire φ) dans lequel les données peuvent être séparées linéairement. La fonction noyau s'applique à tout algorithme pouvant s'exprimer sous forme d'une expression basée sur des produits scalaires entre les observations à classer.

L'ensemble d'observations $X = \{x_1, x_2, \dots, x_i, \dots, x_N\}$ à classer peuvent être présentées dans un espace de D dimensions $\mathcal{R}^D$, tel que $x_i = \{x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iD}\}, i = 1, \dots, N$. Soit φ une fonction de transformation non linéaire permettant de projeter les observations initiales à classer vers un espace de plus grande dimension $\mathcal{H}$ dit espace d'attributs, tel que :

$$\varphi : \ \mathcal{R}^D \rightarrow \mathcal{H}$$

$$x_i \rightarrow \varphi(x_i)$$

L'idée est de supposer qu'on peut calculer le produit scalaire $< \varphi(x_i), \varphi(x_n) >$ directement sans jamais avoir à calculer explicitement $\varphi(x)$, en choisissant une fonction noyau O, tel que : $O(x_i, x_n) =< \varphi(x_i), \varphi(x_n) >$. Plusieurs fonctions noyaux sont proposées [35]. Parmi elles on peut citer :

Noyau linéaire $O(x_i, x_n) = x_i . x_n$

Noyau polynomial $O(x_i, x_n) = (x_i . x_n + c)^d$ où $c \geq 0, d \in \mathbb{N}$

Noyau Gaussien $O(x_i, x_n) = exp(-\|x_i - x_n\|^2 / 2\sigma^2)$ où $\sigma > 0$

Noyau de base radiale $O(x_i, x_n) = exp(\sum|x_i^a - x_n^a|^b/\sigma^2)$où $\sigma, a, b > 0$. Pour $a = 1$et $b = 2$, ce noyau devient un noyau Gaussien.

Noyau Hyper Tangent $O(x_i, x_n) = 1 - tanh(-\|x_i - x_n\|^2/\sigma^2)$ où $\sigma > 0$

2.3.1.6.2 La version noyau de l'algorithme FCM

L'exploitation de la fonction noyau dans l'algorithme de Fuzzy C-means, a donné naissance à une version dite Kernel FCM [36]. Il s'agit dans cet algorithme d'appliquer la transformation non linéaire sur les données à classer afin de les projeter dans un espace de plus grande dimension dans lequel les classes deviennent linéairement séparables.

L'algorithme KFCM modifie la fonction objectif du FCM via la transformation φ, comme suit :

$$J_{KFCM}(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N}\mu_{ki}^m\|\varphi(x_i) - \varphi(v_k)\|^2 \quad (2.33)$$

Où $\|\varphi(x_i) - \varphi(v_k)\|^2$ est le carré de la distance entre $\varphi(x_i)$ et $\varphi(v_k)$

La distance dans l'espace d'attributs entre les transformées d'une observation $\varphi(x_i)$ et d'un centre d'une classe $\varphi(v_k)$ est donnée par :

$$\varphi_{d_{ik}^2} = \|\varphi(x_i) - \varphi(v_k)\|^2 = \big(\varphi(x_i) - \varphi(v_k)\big)\big(\varphi(x_i) - \varphi(v_k)\big)$$

$$= \varphi(x_i)\varphi(x_i) - 2\varphi(x_i)\varphi(v_k) + \varphi(v_k)\varphi(v_k)$$

$$= O(x_i, x_i) - 2O(x_i, v_k) + O(v_k, v_k)$$

En utilisant le noyau gaussien à base radiale, c'est-à-dire $O(x,y) = exp(-\|x-y\|^2/\sigma^2)$, on en déduit que $O(x,x) = 1$ et $\varphi_{d_{ik}^2} = 2\big(1 - O(x_i, v_k)\big)$

La fonction objectif de l'équation (2.33) devient donc

$$J_{KFCM}(U,V) = 2\sum_{k=1}^{K}\sum_{i=1}^{N}\mu_{ki}^m\big(1 - O(x_i, v_k)\big) \quad (2.34)$$

D'une manière similaire à l'algorithme FCM, la fonction objectif J_{KFCM} peut être minimisée sous la contrainte (2.3). Plus précisément en prenant les dérivées premières du Lagrangien de J_{KFCM} par rapport à μ_{ki} et v_k égales à zéros, on aboutit à La matrice de partition floue U définie par son élément général :

$$\mu_{ki} = \sum_{l=1}^{K} \frac{\left(1-O(x_i,v_l)\right)^{1/m-1}}{\left(1-O(x_i,v_k)\right)^{1/m-1}} \quad (2.35)$$

et aux centres des classes v_k définissant le vecteur V dont le $k^{ième}$ élément est :

$$v_k = \frac{\sum_{i=1}^{N} \mu_{ki}^m O(x_i,v_k) x_i}{\sum_{i=1}^{N} \mu_{ki}^m O(x_i,v_k)} \quad (2.36)$$

2.3.2 Variantes selon la fonction objectif

D'autres variantes du FCM agissent directement sur la fonction objectif afin d'améliorer les résultats de la classification notamment en présence du bruit. Celles-ci introduisent généralement un autre terme dans la fonction objectif de l'algorithme de FCM standard.

2.3.2.1 Algorithme de Frigui et Krishnapuram

Frigui et Krishnapuram [37], ont proposé un algorithme d'agrégation compétitive dit compétitive agglomeration (CA) qui combine les avantages des approches de classification hiérarchique et de partitionnement. L'algorithme CA est un algorithme itératif, il partitionne les observations dans un premier temps en un grand nombre de classes de petite taille, qui rentrent en compétition puis, progressivement les classes qui perdent sont fusionnées. L'algorithme CA minimise la fonction objectif suivante :

$$J_{CA}(U,V) = \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki}^2 \, d_{ik}^2 - \alpha \sum_{k=1}^{K} [\sum_{i=1}^{N} \mu_{ki}]^2 \quad (2.37)$$

$d_{ik}^2 = d^2(x_i - v_k)$: représente la distance entre le pixel x_i et le centre de la classe k. Cette distance peut être de type euclidienne ou autre.

La fonction objectif décrite dans la relation (2.37) est composée de deux termes. Le premier terme est similaire à celui de l'équation (2.1) de l'algorithme FCM standard (avec un facteur flou m égal à 2), il permet de contrôler la forme et la taille des classes et obtenir des classes compactes (condensées). Le deuxième terme de cette équation nous permet de contrôler le nombre de classes.

La minimisation de la fonction objectif (2.37) en appliquant les multiplicateurs de Lagrange, nous conduit aux formules de mise à jour des degrés d'appartenance μ_{ki} et des centres de classes v_k suivantes :

$$\mu_{ki} = \alpha \frac{N_k}{d^2(x_i,v_k)} + \frac{1-\alpha \sum_{l=1}^{K} [N_l / d^2(x_i,v_l)]}{\sum_{l=1}^{K} [d^2(x_i,v_k)/d^2(x_i,v_l)]} \quad (2.38)$$

$$v_k = \frac{\sum_{i=1}^{N} \mu_{ki}^2 \, x_i}{\sum_{i=1}^{N} \mu_{ki}^2} \tag{2.39}$$

$N_k = \sum_{i=1}^{N} \mu_{ki}$ est le cardinal de la classe k

α est un coefficient dont la valeur doit être choisie de telle sorte que les deux termes de l'équation (2.37) soient de même ordre de grandeur. Pour cela, α doit être proportionnel à β et égal au rapport de ces deux termes :

$$\beta = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki}^2 d^2(x_i, v_k)}{\sum_{k=1}^{K} [\sum_{i=1}^{N} \mu_{ki}]^2} \tag{2.40}$$

Dans [49], α est aussi fonction du nombre d'itérations q, il prend la forme suivante :

$$\alpha(q) = \eta_0 exp(-q/\tau).\beta \tag{2.41}$$

où η_0 est la valeur initiale et τ une constante.

L'algorithme CA est itératif, initialement le nombre de classes est surestimé puis à chaque itération ce nombre décroit jusqu'à atteindre le nombre optimal. A chaque itération, les classes ayant un cardinal faible sont éliminées.

2.3.2.2 Algorithme de Pham

Dans le cadre de la segmentation d'images IRM et en partant de l'idée que l'algorithme FCM est très sensible au bruit, Pham [38] a proposé la fonction objectif suivante :

$$J_P(U, V) = \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki}^m \, d^2(x_i, v_k) + \frac{\beta}{2} \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki}^m \sum_{r \in N_i} \sum_{n \in M_k} \mu_{nr}^m \tag{2.42}$$

β est un paramètre qui pondère l'influence du deuxième terme de la fonction objectif.

N_i: représente l'ensemble des pixels voisins du pixel x_i

$M_k = \{1, \dots, K\} \backslash \{k\}$ représente l'ensemble des classes à l'exception de la classe k.

Dans cet algorithme, l'appartenance d'un pixel à une classe est conditionnée par l'appartenance de ses voisins aux autres classes. En d'autres termes si l'appartenance des pixels voisins aux autres classes est faible alors cela augmente l'appartenance du pixel x_i à la classe k. Cette méthode dépend donc des valeurs des autres classes et non pas uniquement de la classe envisagée. Comme il n'y a pas de contraintes sur les centres des classes, l'équation

qui calcule les v_k, reste identique à celle de FCM. En revanche la fonction de partition U est différente, elle est obtenue par optimisation Lagrangienne :

$$\mu_{ki} = \frac{[d^2(x_i,v_k)+\beta \sum_{r\in N_i}\sum_{n\in M_k}\mu_{nr}^m]^{\frac{-1}{(m-1)}}}{\sum_{l=1}^{K}[d^2(x_i,v_l)+\beta \sum_{r\in N_i}\sum_{n\in M_l}\mu_{nr}^m]^{\frac{-1}{(m-1)}}} \quad (2.43)$$

Sinon, l'algorithme procède selon les mêmes étapes que l'algorithme FCM.

2.3.2.3 Algorithme C-moyennes possibiliste (PCM)

L'algorithme PCM proposé par Krishnapuram et Keller [39] introduit la notion de possibilité très voisine du flou. Le degré d'appartenance d'une observation à une classe est considéré comme une possibilité d'appartenir à une classe. Contrairement à l'algorithme FCM, la somme des degrés d'appartenance d'un pixel à toutes les classes n'est pas égale à 1.

La fonction objectif ainsi modifiée est :

$$J_{PCM}(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N}\mu_{ki}^m\, d_{ik}^2 + \sum_{k=1}^{K}\eta_k \sum_{i=1}^{N}(1-\mu_{ki})^m \quad (2.44)$$

$$\sum_{i=1}^{N}\mu_{ki} > 0 \quad \forall\, k \in \{1,\ldots,K\}$$

η_k est un nombre positif à déterminer.

Le premier terme de l'équation (2.44) est identique à celui de FCM, le second force μ_{ki} à la plus grande valeur possible.

En dérivant J_{PCM} par rapport à μ_{ki} et en annulant cette dérivée, on obtient l'équation de mise à jour des degrés d'appartenance :

$$\mu_{ki} = \frac{1}{1+[d^2(x_i,v_k)/\eta_k]^{1/m-1}} \quad (2.45)$$

Quant au paramètre η_k, il est estimé à chaque itération en utilisant l'équation suivante

$$\eta_k = \frac{\sum_{i=1}^{N}\mu_{ki}^m\, {d_{ik}}^2}{\sum_{i=1}^{N}\mu_{ki}^m} \quad (2.46)$$

Les centres de classes sont mis à jour comme dans le FCM. Dans cet algorithme, les auteurs ont fixé le facteur flou m à 1,5 et non pas 2 comme dans la plus part des cas.

2.3.2.4 Algorithme de Wu et al

Wu et al [40] proposent de modifier la fonction objectif de l'algorithme FCM standard en rajoutant un terme lié à la variance inter-classes :

$$J_W(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N} \mu_{ki}^{m} \parallel x_i - v_k \parallel^2 - \sum_{k=1}^{K} \sum_{i=1}^{N} \eta_k \mu_{ki}^{m} \parallel v_k - \bar{x} \parallel^2 \quad (2.47)$$

où η_k est un coefficient réel positif $(\eta_k \geq 0)$ à déterminer et $\bar{x}$ est le centre de gravité de l'ensemble des observations à classer tel que :

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (2.48)$$

La minimisation de J_W par rapport à μ_{ki}, v_k et η_k nous permet de déterminer les formules de mise à jour ces variables inconnues comme suit :

$$\mu_{ki} = \frac{(\|x_i - v_k\|^2 - \eta_k \|v_k - \bar{x}\|^2)^{-1/m-1}}{(\sum_{l=1}^{K} \|x_i - v_l\|^2 - \eta_l \|v_l - \bar{x}\|^2)^{-1/m-1}} \quad (2.49)$$

$$v_k = \frac{\sum_{i=1}^{N} \mu_{ki}^{m} x_i - \eta_k \sum_{i=1}^{N} \mu_{ki}^{m} \bar{x}}{\sum_{i=1}^{N} \mu_{ki}^{m} - \eta_k \sum_{i=1}^{N} \mu_{ki}^{m}} \quad (2.50)$$

$$\eta_k = \frac{(\beta/4) min_{k' \neq k} \|v_k - v_{k'}\|^2}{max_l \|v_l - \bar{x}\|^2} \ , 0 \leq \beta \leq 1 \quad (2.51)$$

β est un paramètre qui contrôle la taille d'une classe, les auteurs préconisent une valeur égale à 0.01.

2.3.2.5 Algorithme d'Ahmed et al

Dans le cadre de la segmentation d'images en niveaux de gris, Ahmed et al [41] ont proposé une modification de l'algorithme de FCM standard en introduisant un terme qui fait intervenir l'influence des pixels voisins. Ce terme représente une contrainte spatiale exercée par les pixels voisins sur chaque pixel courant en tenant compte de leurs distances au même centre de la classe. Si un pixel est proche d'un centre d'une classe et ses voisins affichent également une proximité pour ce même centre, alors son degré d'appartenance à cette classe va augmenter. Donc un pixel est contraint de suivre son voisinage, ce qui produit un effet de régularisation. La fonction objectif proposée est :

$$J_A(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N} {\mu_{ki}}^{m} \|\mathrm{x}_i - v_k\|^2 + \frac{\alpha}{card(N_i)} \sum_{k=1}^{K}\sum_{i=1}^{N} {\mu_{ki}}^{m} \left(\sum_{r \in N_i} \|\mathrm{x}_r - v_k\|^2\right) \quad (2.52)$$

N_i représente l'ensemble des pixels spatialement voisins du pixel i et α un paramètre réel qui contrôle l'influence du terme de voisinage

Chaque pixel i respecte la contrainte $\sum_{k=1}^{K} \mu_{ij} = 1$. La minimisation de la fonction objectif J_A est résolue en appliquant également le principe du Lagrangien et en annulant ses dérivées par rapport aux inconnues μ_{ki} et v_k. Cette minimisation nous conduit à :

$$\mu_{ki} = \frac{\left(\|x_i - v_k\|^2 + \frac{\alpha}{card(N_i)} \sum_{r \in N_i} \|x_r - v_k\|^2\right)^{-1/m-1}}{\sum_{l=1}^{K} \left(\|x_i - v_l\|^2 + \frac{\alpha}{card(N_i)} \sum_{r \in N_i} \|x_r - v_l\|^2\right)^{-1/m-1}} \quad (2.53)$$

$$v_k = \frac{\sum_{i=1}^{N} \mu_{ki}^m \left(x_i + \frac{\alpha}{card(N_i)} \sum_{r \in N_i} x_r\right)}{(1+\alpha) \sum_{i=1}^{N} \mu_{ki}^m} \quad (2.54)$$

2.3.2.6 Algorithme de Chen et Zhang

Chen et Zhang [42] ont proposé une extension de la technique d'Ahmed et al, en simplifiant le terme de voisinage afin de réduire le temps de calcul. Leur fonction objectif s'exprime comme :

$$J_{CZ}(U,V) = \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki}^m \|x_i - v_k\|^2 + \alpha \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki}^m \|\bar{x}_i - v_k\|^2 \quad (2.55)$$

$\bar{x}_j$ représente la moyenne des pixels voisins qui se trouve à l'intérieur d'une fenêtre de voisinage centré sur i. La minimisation de la fonction objectif J_{CZ} suivant μ_{ki} et v_k peut être tiré selon le même principe de la technique des multiplicateurs de Lagrange, en utilisant les équations suivantes :

$$\mu_{ki} = \frac{\left(\|x_i - v_k\|^2 + \alpha \|\bar{x}_i - v_k\|^2\right)^{-1/m-1}}{\sum_{l=1}^{K} (\|x_i - v_l\|^2 + \alpha \|\bar{x}_i - v_l\|^2)^{-1/m-1}} \quad (2.56)$$

$$v_k = \frac{\sum_{i=1}^{N} \mu_{ki}^m (x_i + \alpha \bar{x}_i)}{(1+\alpha) \sum_{i=1}^{N} \mu_{ki}^m} \quad (2.57)$$

Cet algorithme permet non seulement de réduire le temps d'exécution mais aussi d'être très performante pour segmenter des images contaminées par un bruit gaussien [42]. Cependant, il s'est avéré moins robuste face à la présence du bruit de type poivre et sel. Afin de résoudre ce problème, les auteurs ont introduit une modification dans laquelle ils ont remplacé la moyenne des pixels voisins par leur valeur médiane [42].

2.3.2.7 Algorithme de Szilágyi et al

Pour segmenter une image en niveaux de gris, Szilágyi et al [43] ont proposé d'appliquer l'algorithme FCM non pas sur l'image d'origine mais sur l'image filtrée ξ générée à partir des pixels de l'image originale et de la moyenne de leurs voisins locaux.

$$\xi_i = \frac{1}{1+\alpha}\left(x_i + \frac{\alpha}{card(N_i)}\sum_{r\in N_i} x_r\right), i = 1, \dots, N \quad (2.58)$$

N_i représente l'ensemble des pixels voisins à l'intérieur d'une fenêtre de voisinage centrée sur le $i^{ième}$ pixel.

La fonction objectif utilisée pour segmenter l'image nouvellement générée ξ est définie par :

$$J_S(U,V) = \sum_{k=1}^{K}\sum_{l=1}^{Q} h_l \mu_{il}^m \, (\xi_l - v_k)^2 \quad (2.59)$$

Q désigne le nombre de niveaux de gris dans l'image, avec $Q \ll N$

h_l représente le nombre de pixels ayant le niveau de gris l avec $l = 1, \dots, Q$, et $\sum_{l=1}^{Q} h_l = N$

Sous la contrainte $\sum_{k=1}^{K} \mu_{kl} = 1 \; (\forall \, l = 1, \dots, Q)$, on minimise J_S en annulant ses dérivées premières par rapport à μ_{kl} et v_k respectivement pour en déduire :

$$\mu_{kl} = \left[\sum_{c=1}^{K}\left(\frac{\xi_l - v_k}{\xi_l - v_c}\right)^{2/m-1}\right]^{-1} \quad (2.60)$$

$$v_k = \frac{\sum_{l=1}^{Q} h_l \mu_{kl}^m \xi_l}{\sum_{l=1}^{Q} h_l \mu_{kl}^m} \quad (2.61)$$

Dans cet algorithme, les auteurs notent qu'une valeur optimale de α comprise entre 0.5 et 1.2, permet à la fois d'éliminer le bruit dans l'image et de préserver son contenu.

2.3.2.8 Algorithme de Kouhi et al

L'une des caractéristiques les plus importantes dans l'image est que les pixels voisins sont très corrélés. En d'autres termes, ces pixels voisins possèdent des caractéristiques similaires et la probabilité qu'ils appartiennent à la même classe est grande. Cette liaison spatiale est très importante dans la classification, malheureusement, elle n'est pas prise en compte dans l'algorithme FCM standard. Cependant, certaines versions de l'algorithme FCM telles que

l'algorithme d'Ahmed et al [41], l'algorithme de Chen et Zhang [42] et l'algorithme de Szilágyi et al [43] cités précédemment, ont su profiter de cette caractéristique spatiale. Ils l'ont introduit dans leurs algorithmes tout en partageant un certain paramètre α qui contrôle l'influence du terme de voisinage. Mais, le choix d'une valeur optimale de ce paramètre est aussi important que difficile, car d'une part, une bonne sélection du paramètre α permet d'assurer un équilibre entre la robustesse face au bruit et l'efficacité à préserver les détails de l'image et d'autre part, sa détermination dépend du bruit dont le type et l'intensité sont généralement inconnus a priori.

Afin d'améliorer l'exploitation de l'information spatiale et surmonter l'inconvénient de la sélection du paramètre α, Kouhi et al [44] ont défini une nouvelle fonction objectif :

$$J_K(U,V) = \sum_{i=1}^{N}\sum_{k=1}^{K}\mu_{ki}^{m}\left\{\|x_i - v_k\|^2 + \frac{1}{d_{ir}+1}\sum_{\substack{r\in N_i \\ i\neq r}}\left(\sum_{\substack{l=1 \\ l\neq k}}^{K}\mu_{lr}\right)^{m}\|x_r - v_k\|^2\right\} \tag{2.62}$$

N_i représente l'ensemble des pixels voisins autour du pixel i et d_{ir} est la distance spatiale entre les pixels i et r.

La fonction objectif J_K est minimisée de la même façon que l'algorithme FCM standard en dérivant par rapport à μ_{ki} et v_k et en annulant les premières dérivées du Lagrangien :

$$v_k = \frac{\sum_{i=1}^{N}\mu_{ki}^{m}x_i}{\sum_{i=1}^{N}\mu_{ki}^{m}} \tag{2.63}$$

$$\mu_{ki} = \frac{\left[\|x_i - v_k\|^2 + \frac{1}{d_{ir}+1}\sum_{\substack{r\in N_i \\ i\neq r}}\left(\sum_{\substack{l=1 \\ l\neq k}}^{K}\mu_{lr}\right)^{m}\|x_r - v_k\|^2\right]^{-1/m-1}}{\sum_{s=1}^{K}\left[\|x_i - v_s\|^2 + \frac{1}{d_{ir}+1}\sum_{\substack{r\in N_i \\ i\neq r}}\left(\sum_{\substack{l=1 \\ l\neq k}}^{K}\mu_{lr}\right)^{m}\|x_r - v_s\|^2\right]^{-1/m-1}} \tag{3.64}$$

2.3.2.9 Algorithme de Kang et al

Dans le cadre de la segmentation d'images en niveaux de gris Kang et al [45] ont apporté une légère modification à la fonction objectif proposée par Ahmed et al (paragraphe 2.3.2.5) afin de réduire le temps de calcul. Celle ci consiste à remplacer le terme $\frac{1}{card(N_i)}\sum_{r\in N_i}\| x_r - v_i \|^2$ par $\| \overline{x^*}_i - v_k \|^2$ où $\overline{x^*}_i$ est le niveau de gris moyen calculé pour chaque pixel i de coordonnées (p,q) dans le centre de la fenêtre de voisinage de taille $(2n+1)(2n+1)$.

$$\overline{x^*}_i = \frac{\sum_{s=-n}^{n}\sum_{t=-n}^{n}w_{s,t}x_{p+s,q+t}}{\sum_{s=-n}^{n}\sum_{t=-n}^{n}w_{s,t}} \tag{2.65}$$

La matrice poids W de taille $(2n+1)(2n+1)$ est définie par la fonction non linéaire :

$$w_{s,t}(x_{p+s,q+t}) = \begin{cases} k_0, & si \mid x_{p,q} - x_{p+s,q+t} \mid > k_1 \\ 1, & sinon \end{cases} \quad (2.66)$$

Notons que $w_{00} = 1$

où $x_{p+s,q+t}$ représente le niveau de gris du pixel de coordonnées $(p+s, q+t)$. k_1, k_0 sont des constantes positives telles que : $0 < k_0 < 0.5$ et $0.5 < k_1 < 1$

La fonction objectif ainsi modifiée est:

$$J_K(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N}\mu_{ki}^m \|x_i - v_k\|^2 + \alpha \sum_{k=1}^{K}\sum_{i=1}^{N}\mu_{ki}^m \|\overline{x^*}_i - v_k\|^2 \quad (2.67)$$

Sa minimisation conduit aux formules de mise à jour des centres des classes et des degrés d'appartenances :

$$v_k = \frac{\sum_{i=1}^{N}\mu_{ki}^m (x_i + \alpha\overline{x^*}_i)}{(1+\alpha)\sum_{i=1}^{N}\mu_{ki}^m} \quad (2.68)$$

$$\mu_{ki} = \frac{1}{\sum_{l=1}^{K}\left(\frac{\|x_i - v_k\|^2 + \alpha\|\overline{x^*}_i - v_k\|^2}{\|x_i - v_l\|^2 + \alpha\|\overline{x^*}_i - v_l\|^2}\right)^{1/m-1}} \quad (2.69)$$

Cette méthode a été particulièrement appliquée à la détection des plaques dentaires [45].

2.3.2.10 Algorithme de Cai et al

Cai et al [46] proposent une autre variante de l'algorithme FCM, dans le but de segmenter une image en niveaux de gris. Cette méthode définit un facteur S_{ir} comme une mesure de similarité locale (spatiale et en niveaux de gris) entre un pixel i et son voisin r appartenant à une fenêtre de voisinage centrée sur le pixel i.

$$S_{ir} = \begin{cases} S_{s_ir} \times S_{g_ir}, & i \neq r \\ 0 & i = r \end{cases} \quad (2.70)$$

avec

$$S_{s_ir} = exp\left(\frac{-max(|p_r - p_i|, |q_r - q_i|)}{\lambda_s}\right) \quad (2.71)$$

où (p_i, q_i) représente les coordonnées spatiales du pixel i et λ_s est un facteur d'échelle.

S_{s_ir} introduit l'information spatiale locale

$$S_{g_ir} = exp\left(\frac{-\|x_i - x_r\|^2}{\lambda_g \times \sigma_{g_i}^2}\right) \quad (2.72)$$

x_i est le niveau de gris du pixel central et x_r désigne le niveau de gris des pixels voisins dans la même fenêtre et σ_{g_i} est défini par :

$$\sigma_{g_i} = \sqrt{\frac{\sum_{r \in N_i} \|x_r - x_i\|^2}{card(N_i)}} \quad (2.73)$$

Sa valeur reflète le degré d'homogénéité en niveau de gris des pixels situés dans la fenêtre de voisinage, c'est-à-dire plus sa valeur est petite, plus la région locale est homogène. Ce paramètre peut être calculé à l'avance et sa valeur diffère d'un pixel à un autre.

A partir des facteurs S_{ir} une nouvelle image ξ est générée de la manière suivante :

$$\xi_i = \frac{\sum_{r \in N_i} S_{ir} x_r}{\sum_{r \in N_i} S_{ir}} \quad (2.74)$$

La méthode de Cai et al revient à appliquer la version de Szilágyi et al (paragraphe 2.3.2.7) sur l'image ξ.

2.3.2.11 Algorithme de Li et Mukaidono

Li et Mukaidono [47] proposent de rajouter à la fonction objectif de l'algorithme FCM un deuxième terme caractérisé par l'entropie du degré d'appartenance comme suit :

$$J_{LM}(U,V) = \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki} \|x_i - v_k\|^2 + \delta \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki} log \mu_{ki} \quad (2.75)$$

δ est un coefficient réel strictement positif ($\delta > 0$)

La mise à jour des degrés d'appartenance est réalisée par la formule définie ci-dessous :

$$\mu_{ki} = \frac{exp\left(-\frac{\|x_i - v_k\|^2}{\delta}\right)}{\sum_{l=1}^{C} \left(-\frac{\|x_i - v_l\|^2}{\delta}\right)} \quad (2.76)$$

et les centres des classes sont déterminés comme dans l'algorithme FCM classique mais avec un facteur flou $m = 1$:

$$v_k = \frac{\sum_{i=1}^{N} \mu_{ki} x_i}{\sum_{j=1}^{N} \mu_{ki}} \quad (2.77)$$

2.3.2.12 Algorithme de Yang et al

La fonction objectif proposée par Yang et al dans le cadre de la segmentation d'images en niveaux de gris [48] est donnée par la relation suivante :

$$J_Y(U,V) = \sum_{k=1}^{K}\sum_{i}^{N} \mu_{ki}^2 \ \|x_i - v_k\|^2 + \sum_{i=1}^{N} d_i \sum_{k=1}^{K} \mu_{ki}(1-\mu_{ki}) \quad (2.78)$$

d_i désigne un paramètre de pondération de chaque pixel. Il peut être déterminé par la relation suivante :

$$d_i = min\left\{d_{ik}^2 \ / \ k \in \{1,2,\dots,K\}\right\} - \eta \quad (2.79)$$

η est un paramètre strictement positif ($\eta > 0$). Une valeur égale à $0{,}01$ a été utilisée par les auteurs.

La minimisation de J_Y par rapport à μ_{ki} conduit à :

$$\mu_{ki} = \frac{1}{\sum_{l=1}^{K}(d_{ik}^2 - d_i)/(d_{il}^2 - d_i)} \quad (2.80)$$

La formule de mise à jour des centres de classes reste identique à celle de l'algorithme FCM classique (équation (2.11)) avec $m = 2$.

2.3.2.13 Algorithme de Tang et al

Tang et al [49] proposent de minimiser la fonction objectif suivante :

$$J_T(U,V) = \sum_{i=1}^{N} e^{t_i} \sum_{k=1}^{K} \mu_{ki}^m d_{ik}^2 \quad (2.81)$$

Le terme e^{t_i} est le facteur d'impact de la $i^{ième}$ observation et t_i son facteur poids tel que :

$$\prod_{i=1}^{N} e^{t_i} = 1 \quad (2.82)$$

La minimisation de la fonction J_T représente un problème d'optimisation non linéaire qui peut être résolu par la méthode des multiplicateurs de Lagrange. Le résultat de cette minimisation est :

$$\mu_{ki} = \left(\frac{d_{ik}}{\sum_{l=1}^{K} d_{il}}\right)^{-2/m-1} \quad (2.83)$$

et

$$e^{t_i} = \frac{\prod_{n=1}^{N}[(\sum_{k=1}^{K}\mu_{kn}d_{nk}^2)^{\frac{1}{N}}]}{\sum_{k=1}^{K}\mu_{ki}^m d_{ik}^2} \quad (2.84)$$

$$v_k = \frac{\sum_{i=1}^{N} e^{\frac{t_i}{s}}\mu_{ki}^m x_i}{\sum_{i=1}^{N} e^{\frac{t_i}{s}}\mu_{ki}^m} \quad (2.85)$$

s est un paramètre réel qui joue le même rôle que le facteur flou m. Une valeur égale à -1 est préconisée par les auteurs.

2.3.2.14 Algorithme de Krinidis et Chatzis

Krinidis et Chatzis [50] ont proposé un algorithme appelé Fuzzy Local Information C-Means (FLICM) pour segmenter une image en niveaux de gris. Cet algorithme introduit un nouveau paramètre G_{ki} dans la fonction objectif du FCM. La caractéristique principale de l'algorithme FLICM est l'utilisation d'une mesure de similarité locale floue (à la fois spatiale et en niveau de gris), visant à garantir l'insensibilité face aux bruits et à la préservation des détails de l'image.

Le calcul du facteur G_{ki} est obtenu en utilisant une fenêtre glissante de dimension prédéfinie tel que :

$$G_{ki} = \sum_{\substack{r\in w_i \\ r\neq i}} \frac{1}{d_{ir}+1}(1-\mu_{kr})^m \|x_r - v_k\|^2 \quad (2.86)$$

où i désigne le pixel situé au centre de la fenêtre choisie, r représente les pixels appartenant à la fenêtre w_i autour du pixel i, d_{ir} est la distance euclidienne spatiale entre les pixels i et r et μ_{kr} est le degré d'appartenance du pixel r à la classe k.

La fonction objectif devient donc :

$$J_{KC}(U,V) = \sum_{i=1}^{N}\sum_{k=1}^{K}[\mu_{ki}^m \|x_i - v_k\|^2 + G_{ki}] \quad (2.87)$$

Les degrés d'appartenance sont calculés par l'équation suivante :

$$\mu_{ki} = \frac{1}{\sum_{l=1}^{K}\left(\frac{\|x_i - v_k\|^2 + G_{ki}}{\|x_i - v_l\|^2 + G_{li}}\right)^{1/m-1}} \quad (2.88)$$

La formule (2.11) permettant le calcul des centres de classes reste inchangée.

2.3.2.15 Algorithme de Gauge et Sasi

Gauge et Sasi [51] ont proposé une variante de l'algorithme FCM pour la segmentation des images couleurs. Cette méthode comprend à la fois l'information spatiale locale des pixels voisins et la distance euclidienne spatiale au centre de gravité de chaque classe. Les auteurs ont inclus un nouveau facteur H_{ik} dans l'équation de calcul du degré d'appartenance de chaque pixel à toutes les classes. Il est basé sur la distance euclidienne spatiale au centre de gravité de chaque classe tel que :

$$H_{ik} = \frac{d_{ik}}{1+\sum_{l=1}^{K} d_{il}} \tag{2.89}$$

d_{ik} : est la distance euclidienne spatiale entre le pixel i et le centre de gravité de la classe k

L'équation de calcul des degrés d'appartenance incluant ce nouveau facteur devient :

$$\mu_{ki} = \frac{1}{\sum_{l=1}^{K}\left(\frac{|p_i-v_k|^2}{|p_i-v_l|^2}\right)^{2/m-1}} - H_{ik} \tag{2.90}$$

Afin d'accélérer la convergence de cet algorithme, ce nouveau facteur H_{ik} est combiné avec le facteur G_{ki} proposé par Krinidis et Chatzis (paragraphe 2.3.2.14) et la formule de mise à jour des degrés d'appartenance devient :

$$\mu_{ki} = \frac{1}{\sum_{l=1}^{K}\left(\frac{|p_i-v_k|^2+G_{ki}}{|p_i-v_l|^2+G_{li}}\right)^{2/m-1}} - H_{ik} \tag{2.91}$$

2.3.2.16 Algorithme de Zhu et al

L'algorithme de Zhu et al [52] constitue une autre version modifiée de l'algorithme de FCM. Un nouveau terme a été introduit à la fonction objectif de FCM standard et qui lui permet de converger plus rapidement.

Cette fonction objectif est définie par :

$$J_Z(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N} {\mu_{ki}}^m \|x_i - v_k\|^2 + \sum_{i=1}^{N} a_i \sum_{k=1}^{K} \mu_{ki}(1 - {\mu_{ki}}^{m-1}) \tag{2.92}$$

En minimisant J_Z par la méthode des multiplicateurs de Lagrange, les nouveaux degrés d'appartenance sont donnés par :

$$\mu_{ki} = \left[\sum_{l=1}^{K}\left(\frac{\|x_i - v_k\|^2 - a_i}{\|x_i - v_l\|^2 - a_i}\right)\right]^{\frac{-1}{m-1}} \quad (2.93)$$

et les centres des classes par :

$$v_k = \frac{\sum_{i=1}^{N} {\mu_{ki}}^m x_i}{\sum_{i=1}^{N} {\mu_{ki}}^m} \quad (2.94)$$

Pour satisfaire la condition $0 \leq \mu_{ki} \leq 1$, a_i prend la forme suivante $a_i = \alpha.min\{\|x_i - v_k\|^2 / k \in \{1, \dots, K\}\}$. Le paramètre α $(0 < \alpha < 1)$ contrôle la vitesse de convergence de l'algorithme. Cet algorithme converge plus rapidement que l'algorithme FCM standard, mais il souffre toujours du même inconvénient que ce dernier et qui est sa sensibilité au bruit.

2.3.2.17 Algorithme de Zhao et Jiao

Afin de surmonter le problème de la sensibilité face aux bruits de l'algorithme proposé par Zhu et al, Zhao et Jiao [53] ont proposé un nouvel algorithme de segmentation d'images qui consiste à introduire une contrainte sur l'information spatiale dans la fonction objectif J_Z de Zhu et al (paragraphe 2.3.2.16). La nouvelle fonction objectif proposée s'écrit :

$$J_{ZJ}(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N} {\mu_{ki}}^m \|x_i - v_k\|^2 + \sum_{i=1}^{N} a_i \sum_{k=1}^{K} \mu_{ki}(1 - {\mu_{ki}}^{m-1}) +$$

$$\beta \sum_{k=1}^{K}\sum_{i=1}^{N} {\mu_{ki}}^m \|\bar{x}_i - v_k\|^2 \quad (2.95)$$

Le troisième terme désigne la contrainte spatiale, dans lequel $\bar{x}_i$ résume l'information spatiale du pixel i et qui est représentée soit par la moyenne ou la valeur médiane des pixels à l'intérieur de la fenêtre de voisinage w autour du pixel i. Le paramètre β contrôle l'effet de la contrainte spatiale.

En minimisant la fonction objectif J_{ZJ} par les multiplicateurs de Lagrange, on aboutit au formules de mise à jour suivantes :

$$v_k = \frac{\sum_{i=1}^{N} \mu_{ki}^m (x_i + \beta \bar{x}_i)}{(1+\beta)\sum_{i=1}^{N} \mu_{ki}^m} \quad (2.96)$$

$$\mu_{ki} = \left[\sum_{l=1}^{K}\left(\frac{\|x_i - v_k\|^2 - a_i + \beta\|\bar{x}_i - v_k\|^2}{\|x_i - v_l\|^2 - a_i + \beta\|\bar{x}_i - v_l\|^2}\right)\right]^{-1/m-1} \quad (2.97)$$

Pour que la condition $0 \leq \mu_{ki} \leq 1$ soit satisfaite, il faut que $a_i = \alpha.min\{\|x_i - v_k\|^2 + \beta\|\bar{x}_i - v_k\|^2 / k \in \{1, \dots, K\}\}$

Plus récemment une version de cet algorithme basé sur des fonctions noyaux a été proposée dans [58]. Cette nouvelle méthode converge rapidement et se comporte mieux lors de la segmentation d'images en niveaux de gris bruitées.

2.3.2.18 Algorithme de Ji et al

Ji et al [55] abandonnent l'idée d'ajouter un terme de contrainte spatiale dans la fonction objectif de l'algorithme FCM classique. Au lieu de cela, ils intègrent l'information spatiale locale directement dans le processus de segmentation en remplaçant les pixels par des patchs.

Pour chaque pixel i de l'image, on définit un patch centré sur ce pixel par un vecteur P_i contenant les pixels voisins appartenant à une fenêtre de voisinage de taille N_i.

$P_i = (x_{ir}, r \in N_i)$ et x_{ir} est la valeur de son $r^{ième}$ pixel voisin. Pour chaque pixel r appartenant au patch P_i on lui associe un poids noté w_{ir} tel que :

$$w_{ir} = \frac{\xi_r}{\sum_{r \in N_i} \xi_r} \tag{2.98}$$

où $\xi_r = exp\left[-\left(\sigma_r - \frac{\sum_{r \in N_i} \sigma_r}{N_i}\right)\right]$ (2.99)

et σ_r l'écart quadratique moyen tel que : $\sigma_r = \left[\frac{\sum_{s \in N_i \setminus \{r\}} (x_s - x_r)^2}{N_i - 1}\right]^{1/2}$ (2.100)

La fonction objectif de cet algorithme est :

$$J_J(U,V) = \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{ki}^m \sum_{r \in N_i} w_{ir} \|x_{ir} - v_k\|^2 \tag{2.101}$$

Sous la contrainte $\sum_{k=1}^{K} \mu_{ki} = 1$ et de la même manière que l'algorithme FCM classique, la minimisation de cette fonction conduit à la matrices des degrés d'appartenance U définie par :

$$\mu_{ki} = \left[\sum_{l=1}^{K} \left(\frac{\sum_{r \in N_i} w_{ir} \|x_{ir} - v_k\|^2}{\sum_{r \in N_i} w_{ir} \|x_{ir} - v_l\|^2}\right)^{1/(m-1)}\right]^{-1} \tag{2.102}$$

et aux centres des classes :

$$v_k = \frac{\sum_{i=1}^{N} \mu_{ki}^m w_{ir} x_{ir}}{\sum_{i=1}^{N} \mu_{ki}^m w_{ir}} \tag{2.103}$$

2.3.2.19 Algorithme de Benaichouche et al

Une méthode de segmentation d'images en niveaux de gris a été proposée récemment par Benaichouche et al [54]. Elle se déroule en trois étapes :

1. Initialisation des centres des classes en utilisant l'algorithme de PSO
2. La segmentation d'image en utilisant l'algorithme de FCM amélioré en introduisant l'information spatiale et la distance de Mahalanobis
3. Reclassification des pixels éventuellement mal classés (post-segmentation)

Afin d'optimiser l'étape d'initialisation de l'algorithme de FCM, une méta-heuristique PSO est utilisée pour trouver les meilleures positions initiales des centres des classes.

Dans le processus de segmentation basée sur l'optimisation d'un critère, le choix de ce dernier est crucial pour le résultat final. Dans leur méthode proposée les auteurs ont pris la fonction objectif de Chen et Zhang [42] et afin de l'améliorer, ils ont introduit la distance de Mahalanobis comme mesure de similarité au lieu de la distance euclidienne, tel que

$$J_{CZ}(U,V) = \sum_{i=1}^{N}\sum_{k=1}^{K}\mu_{ki}^{m}d^2(x_i,v_k) + \alpha\sum_{i=1}^{N}\sum_{k=1}^{K}\mu_{ki}^{m}d^2(\bar{x}_i,v_k) \quad (2.104)$$

Les formules de mise à jour sont :

$$\mu_{ki} = \frac{\left(\|x_i-v_k\|^2+\alpha\|\bar{x}_i-v_k\|^2\right)^{-1/m-1}}{\sum_{l=1}^{K}(\|x_i-v_l\|^2+\alpha\|\bar{x}_i-v_l\|^2)^{-1/m-1}} \quad (2.105)$$

$$v_k = \frac{\sum_{i=1}^{N}\mu_{ki}^{m}(x_i+\alpha\bar{x}_i)}{(1+\alpha)\sum_{i=1}^{N}\mu_{ki}^{m}} \quad (2.106)$$

avec

$$d^2(x_i,v_k) = (x_i-v_k)^T A_k (x_i-v_k) \quad (2.107)$$

$$A_k = [\rho_k \det(S_k)]^{\frac{1}{p}} {S_k}^{-1} \quad (2.108)$$

et $S_k = \frac{\sum_{i=1}^{N}\mu_{ki}^{m}(x_i-v_k)(x_i-v_k)^T}{\sum_{i=1}^{N}\mu_{ki}^{m}}$ (2.109)

La correction des pixels éventuellement mal classés, se déroule en deux étapes. La première consiste à détecter ces pixels de l'image segmentée par extraction de tous les pixels qui n'ont pas la même étiquette dans leur voisinage, et la deuxième étape consiste à reclassifier ces pixels extraits en minimisant le critère suivant :

$$J_i = \frac{\alpha\sigma_l{}^{(i)} + \|x_i - u_l{}^{(i)}\|}{\beta N_l} + \sum_{\substack{k=1 \\ k\neq l}}^{N_{pi}} \sigma_k{}^{(i)} \quad (2.110)$$

où N_{pi} est le nombre des différentes étiquettes présentes dans le voisinage du pixel i, N_l est le nombre de pixels appartenant à la classe l dans le voisinage du pixel i, x_i est le pixel extrait, $u_l{}^{(i)}$ est la moyenne locale de la classe l dans le voisinage du pixel i, $\sigma_l{}^{(i)}$ est la variance locale de la classe l dans le voisinage du pixel i après sa réallocation à cette classe et $\sigma_k{}^{(i)}$ est la variance locale de la classe k dans le voisinage du pixel i.

Le paramètre α permet de régler l'effet de la variance locale sur la redistribution des pixels et β est un paramètre fixe mis égal à $\frac{1}{N_{voi}}$.

2.3.3 Variantes selon la fuzzification

Une autre classe de variantes du FCM concerne la modification de l'influence du facteur flou m.

2.3.3.1 Algorithme de Rousseeuw et al

Rousseeuw et al [56] ont proposé de modifier la fonction objectif du FCM en remplaçant μ_{ki} par une fonction $f(\mu_{ki})$, telle que :

$$f(\mu_{ki}) = \alpha\mu_{ki} + (1-\alpha)\mu_{ki}^2 \quad (2.111)$$

telle que :

$f(\mu_{ki}) \to \mu_{ki}$ lorsque $\alpha \to 1$

$f(\mu_{ki}) \to \mu_{ki}^2$ lorsque $\alpha \to 0$

α : étant un nombre réel dans l'intervalle [0,1], il permet de moduler le contraste entre les degrés d'appartenance élevés et ceux qui sont faibles.

La fonction objectif ainsi modifiée est :

$$J_R(U,V) = \sum_{k=1}^{K}\sum_{i=1}^{N}[\alpha\mu_{ki} + (1-\alpha)\mu_{ki}^2]\, d_{ik}^2 \quad (2.112)$$

La minimisation de cette fonction par la technique des multiplicateurs de Lagrange aboutit aux centres des classes :

$$v_k = \frac{\sum_{i=1}^{N} f(\mu_{ki}) x_i}{\sum_{i=1}^{N} f(\mu_{ki})} \quad (2.113)$$

Tandis que le degré d'appartenance μ_{ki} reste inchangé.

2.3.3.2 Algorithme de Klawonn et Hoppner

Partant du fait que les degrés d'appartenance ne prennent jamais les valeurs extrêmes 0 et 1, Klawonn et Hoppner [57] proposent une fonction $g(\mu_{ki})$ à la place de ${\mu_{ki}}^m$

$$g(\mu_{ki}) = \alpha\mu_{ki}^2 + (1-\alpha)\mu_{ki}, 0 < \alpha < 1 \quad (2.114)$$

La fonction objectif devient :

$$J_{KH}(U,V) = \sum_{k=1}^{K} \sum_{i=1}^{N} g(\mu_{ki}) d_{ik}^2 \quad (2.115)$$

La minimisation de cette fonction conduit à :

$$\mu_{ki} = \frac{1}{1-\beta} \left[\frac{1+(\widehat{K}-1)\beta}{\sum_{\substack{l=1 \\ \mu_{li}\neq 0}}^{K} \left(\frac{d_{ik}}{d_{il}}\right)} - \beta \right] \quad (2.116)$$

$\widehat{K}$ représente le nombre de classes pour une observation x_i ayant des degrés d'appartenance non nuls.

Le paramètre β est défini par :

$$\beta = \frac{g'(0)}{g'(1)} = \frac{1-\alpha}{1+\alpha} \quad (2.117)$$

La mise à jour des centres des classes reste la même que dans le cas du FCM.

2.4 Conclusion

Nous avons présenté dans ce chapitre l'algorithme standard FCM ainsi qu'un nombre exhaustif de ses variantes. Ces variantes ont été proposées afin d'améliorer les performances du FCM et éviter ses inconvénients notamment sa sensibilité au bruit et les formes sphériques des classes.

Nous avons classé ces variantes en trois catégories. Dans la première, la distance euclidienne utilisée par le FCM standard, est remplacée par un autre type de distance afin de détecter des classes de formes quelconques. Dans la deuxième catégorie, l'information

spatiale a été introduite dans la fonction objectif sous forme d’un terme supplémentaire dans le but de prendre le bruit en considération. La troisième catégorie modifie l’influence du facteur flou.

Certaines de ces méthodes ont été conçues dans le cadre général, c'est-à-dire, pour classer des données multidimensionnelles, d’autres ont été développées spécifiquement pour la segmentation d’images en niveaux de gris ou couleur.

Nous allons appliquer certaines de ces méthodes dans le prochain chapitre à la segmentation d’images texturées.

3.1 Introduction

Nous allons présenter dans ce chapitre les résultats de la segmentation d'images texturées par l'algorithme FCM et ses variantes. Afin de tester l'apport de ces variantes par rapport à l'algorithme FCM standard, nous avons effectué une étude comparative entre ce dernier avec huit variantes qui appartiennent à la deuxième catégorie à savoir les méthodes de Krinidis et Chatzis, Kouhi et al, Yang et al, Zhao et Jiao, Ji et al, Keller et Krishnapuram (PCM), Wu et al ainsi que celle de Tang et al. Toutes ces méthodes ont été implémentées sous Matlab. Ces méthodes seront appliquées à la segmentation d'images texturées et synthétiques. L'utilisation des images synthétiques nous permet d'effectuer une évaluation objective des résultats. Cette évaluation est basée sur le taux de classification exprimé par le rapport entre le nombre de pixels bien classés et le nombre total des pixels de l'image, ainsi que sur le temps de calcul.

Afin d'effectuer une comparaison équitable entre toutes les méthodes implémentées, nous avons fixé le nombre de classes (textures) pour chaque image et nous avons procédé à la même initialisation des centres des classes pour chaque image. Notons que cette initialisation est effectuée d'une manière manuelle afin que les résultats soient indépendants de l'initialisation et que le nombre d'itérations pour chaque algorithme est fixé à 10.

3.2 Stratégie de segmentation

La méthode de segmentation que nous proposons appartient à l'approche "région", elle se déroule en deux phases. Dans la première, chaque pixel de l'image est caractérisé par un ensemble d'attributs de texture. Ces attributs sont calculés pour chaque pixel de l'image à partir de l'histogramme et de la matrice de cooccurrence en tenant compte des pixels voisins situés à l'intérieur d'une fenêtre de voisinage centrée sur le pixel à traiter. La taille de cette fenêtre est de $(2W+1)^2$. Pour accélérer les calculs, l'image à traiter est quantifiée sur Q niveaux de gris. Dans la deuxième phase, on regroupe les pixels ayant des attributs identiques en un ensemble de K classes en utilisant l'algorithme FCM ou l'une de ses variantes. Le nombre de classes K est fixé a priori pour chaque image. Il correspond au nombre de texture présentes dans l'image. La segmentation est donc conditionnée d'une part par la taille de la fenêtre de voisinage (paramètre W), par le nombre de niveaux de gris (Q) et par les paramètres propres à chaque algorithme de classification à savoir:

- Le facteur flou m pour l'algorithme FCM
- Le facteur flou m pour l'algorithme PCM

- Le facteur flou m et le paramètre β pour l'algorithme de Wu.
- Le facteur flou m et le paramètre s pour l'algorithme de Tang.
- Le facteur flou m et le paramètre Nr pour l'algorithme de Kouhi.
- Le facteur flou m et le paramètre Nr pour l'algorithme de Krinidis.
- Le facteur flou m et le paramètre Nr pour l'algorithme de Ji.
- Le facteur flou m et le paramètre η pour l'algorithme de Yang.
- Le facteur flou m et les paramètres α, β et Nr pour l'algorithme de Zhao.

Dans notre travail, nous avons utilisé la texture pour caractériser les pixels de l'image, et avant d'exposer les résultats de nos tests, nous allons donner quelques définitions de la texture ainsi que les différentes méthodes d'analyse de texture.

3.3 Extraction des attributs de texture

La texture est une composante visuelle riche d'information pour la description et la reconnaissance des objets de scènes naturelles. La texture est définie de plusieurs manières [7], [11]. Celle que nous avons retenue est donnée par Unser [3] :

« Une texture est une région d'une image pour laquelle il est possible de définir une fenêtre de dimension minimale, telle qu'une observation au travers de celle-ci se traduit par une perception (impression) visuelle identique pour toutes les translations possibles de cette fenêtre à l'intérieure de la région considérée».

Dans la littérature, trois grandes familles de textures sont recensées: les textures déterministes, les textures stochastiques et les textures observables (hybrides). Les textures déterministes présentent une structure spatiale régulière (répétition spatiale d'un motif de base appelé texton). Notons que les textures parfaitement régulières existent rarement dans la réalité, elles sont généralement artificielles. Les textures stochastiques, aussi connues sous le nom de textures aléatoires ou micro-textures, se caractérisent par un contenu irrégulier dont il est parfois difficile d'en extraire une primitive de base (motif non localisable). Ces textures sont bien adaptées aux textures naturelles, elles sont décrites par des lois ou des modèles statistiques et sont considérées comme une réalisation d'un champ aléatoire bidimensionnel homogène. Les textures observables se décrivent par un mélange entre les deux textures précédentes.

3.3.1 Analyse de la texture

Les techniques d'analyse de la texture existantes sont utilisées pour formaliser des descriptifs de cette dernière par des attributs mathématiques qui serviront à l'identifier.

Il est possible de classer les méthodes d'analyse de la texture en quatre grandes catégories : géométrique, statistique, fréquentielle et celle basée sur le modèle.

Pour notre application, nous avons utilisé deux méthodes statistiques à savoir celle basée sur l'histogramme et celle basée sur les matrices de cooccurrence. Les attributs extraits à partir de l'histogramme $h(i)$ sont :

- Moyenne : $M = \frac{1}{N}\sum_{i=1}^{Q} ih(i)$ (3.1)
- Variance : $Var = \left[\sum_{i=1}^{Q}(i-M)^2 h(i)\right]^{1/2}$ (3.2)
- Skewness : $Ske = \frac{1}{Var^3}\sum_{i=1}^{Q}(i-M)^3 h(i)$ (3.3)
- Kurtosis : $Kur = \frac{1}{Var^4}\sum_{i=1}^{Q}(i-M)^4 h(i)$ (3.4)

Les matrices de cooccurrence, développées par Haralick et al en 1973, consistent à estimer les fréquences d'apparition de couples de niveaux de gris de pixels séparés par une certaine distance d (généralement prise égale à un pixel) dans une orientation particulière θ ($0^\circ, 45^\circ, 90^\circ$ et 135°). Pour une orientation θ et une distance d, la matrice de cooccurrence est définie par la relation suivante [12] :

$$P(i,j;d,\theta) = card\{((k,l),(m,n)) \in (H \times L) \times (H \times L) | I(k,l) = i, I(m,n) = j\} \quad (3.5)$$

où (k,l) et (m,n) les coordonnées de deux pixels voisins et (i,j) leurs niveaux de gris.

Haralick définit quatorze attributs de texture à partir de la matrice de cooccurrence. Parmi eux, nous avons utilisé le contraste, l'homogénéité (moment différentiel inverse), l'énergie (second moment angulaire) et la corrélation dont les équations sont les suivantes :

$$CONT = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{\substack{i=1 \\ |i-j|=n}}^{N_g} \sum_{j=1}^{N_g} \frac{P(i,j;d,\theta)}{N} \right\} \quad (3.6)$$

$$IDM = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} \frac{1}{1+(i-j)^2}\frac{P(i,j;d,\theta)}{N} \quad (3.7)$$

$$ASM = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} \left\{\frac{P(i,j;d,\theta)}{N}\right\}^2 \quad (3.8)$$

$$COR = \frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Ng}(ij)(P(i,j;d,\theta)/N)-\mu_x\mu_y}{\sigma_x\sigma_y} \quad (3.9)$$

3.4 Résultats

Nous allons présenter les résultats de la segmentation sur trois images synthétiques (Fig.3.1). Leurs images de reference correspondantes sont également, données sur la figure (3.1). Ces images contiennent des textures différentes, 3 pour la première, 4 pour la deuxième et 5 pour la troisième. Les régions texturées ont des formes et des tailles également différentes.

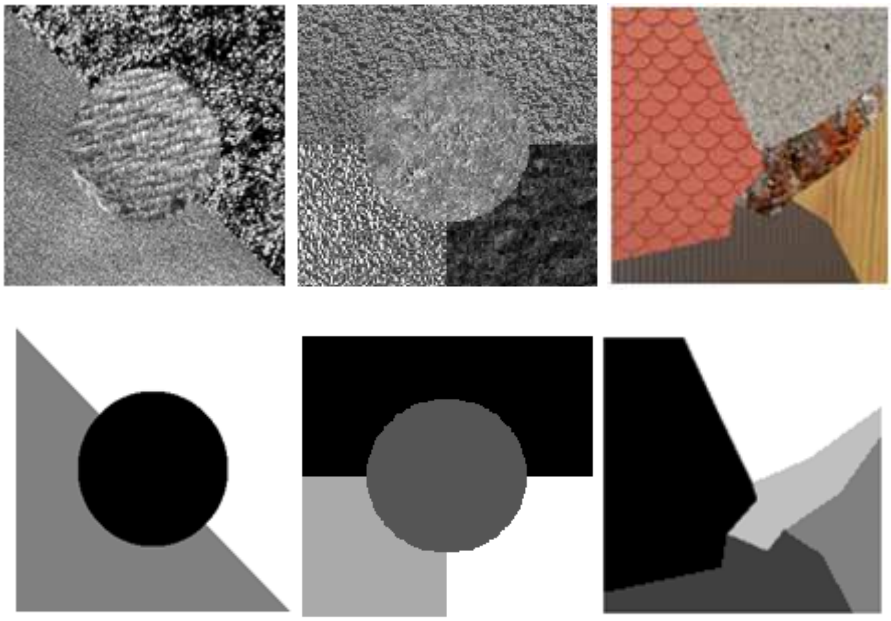

Figure (3.1) : Images tests et leurs images de référence correspondantes

Pour chaque méthode, nous avons varié le nombre de niveaux de gris Q à 8, 16, 32, 64, 128 et 256 ainsi que la taille de la fenêtre de voisinage $(2W+1)(2W+1)$ à (7×7), (9×9) et (11×11) puis nous avons retenu la paire (Q, W) qui a donné le meilleur taux de classification.

Le tableau (3.1) montre à titre d'exemple, la variation du taux de classification (Tc) en fonction de W et Q obtenus par les différentes méthodes sur l'image test 1 et le tableau (3.2) regroupe les valeurs des meilleurs taux de classification obtenus sur les trois images tests. Ce tableau contient également le temps de calcul (Tps) consommé par chaque méthode.

W	Q	FCM	PCM	Wu et al.	Tang et al	Yang et al	Zhao et Jiao	Krinidis et Chatzis	Kouhi et al	Ji et al
3	8	**95.54**	**88.90**	**95.53**	**94.68**	95.80	96.83	95.54	96.29	96.72
	16	95.12	84.70	95.12	93.59	95.66	96.86	95.11	96.13	96.74
	32	94.88	84.01	94.88	93.48	95.32	96.80	94.87	96.93	96.64
	64	94.70	82.87	94.70	92.67	95.20	96.77	94.70	95.78	96.54
	128	94.58	82.49	94.59	91.93	95.07	96.67	94.58	95.67	96.47
	256	94.37	82.36	91.97	94.63	94.86	96.61	94.37	95.54	96.36
4	8	95.97	95.53	95.97	95.75	96.13	96.32	95.97	96.35	96.42
	16	**96.03**	**95.53**	**96.03**	**95.78**	**96.13**	**96.47**	**96.03**	**96.44**	**96.50**
	32	95.91	94.54	95.92	95.45	96.07	96.37	95.91	96.39	96.38
	64	95.72	92.04	95.72	95.07	95.94	96.29	95.72	96.23	96.31
	128	95.68	90.58	95.68	94.80	95.85	96.23	95.68	96.15	96.20
	256	95.53	89.27	95.53	94.66	95.70	96.18	95.53	96.04	96.12
5	8	95.65	95.15	95.62	95.61	95.65	95.61	95.63	95.85	95.66
	16	**95.67**	**95.23**	**95.69**	**95.71**	**95.66**	**95.72**	**95.70**	**95.93**	**95.76**
	32	95.65	94.94	95.65	95.45	95.63	95.62	95.65	95.93	95.70
	64	95.61	94.55	95.53	95.45	95.53	95.55	95.62	95.89	95.62
	128	95.46	93.59	95.46	95.32	95.45	95.53	95.46	95.79	95.53
	256	95.34	92.75	95.34	95.13	95.33	95.41	95.34	95.67	95.38

Tableau (3.1) : Variation du taux de classification (Tc) en fonction de W et Q obtenus par l'algorithme FCM et ses variantes sur l'image test 1.

		FCM	PCM	Wu et al	Tang et al	Yang et al	Zhao et Jiao	Krinidis et Chatzis	Kouhi et al	Ji et al
Image test 1	Tc (%)	96.03	95.56	96.03	95.83	96.13	**96.86**	96.03	96.44	96.74
	Tps (s)	0.30	0.24	0.24	0.54	2.51	4.66	23.89	183.01	148.10
Image test 2	Tc (%)	97.17	93.88	97.16	97.15	97.07	97.24	97.17	96.97	**97.95**
	Tps (s)	0.43	0.18	0.49	0.67	3.24	4.23	48.84	294.91	372.98
Image test 3	Tc (%)	92.16	83.00	92.61	95.18	92.70	**96.58**	91.64	95.31	92.20
	Tps (s)	0.47	0.72	0.62	0.90	0.53	4.30	59.66	638.74	462.36

Tableau (3.2) : Taux de classification et temps d'exécution obtenus par chaque algorithme FCM et ses variantes sur les trois images tests.

Les figures (3.2) à (3.4) montrent les images segmentées obtenues par chaque méthode sur chaque image test, respectivement.

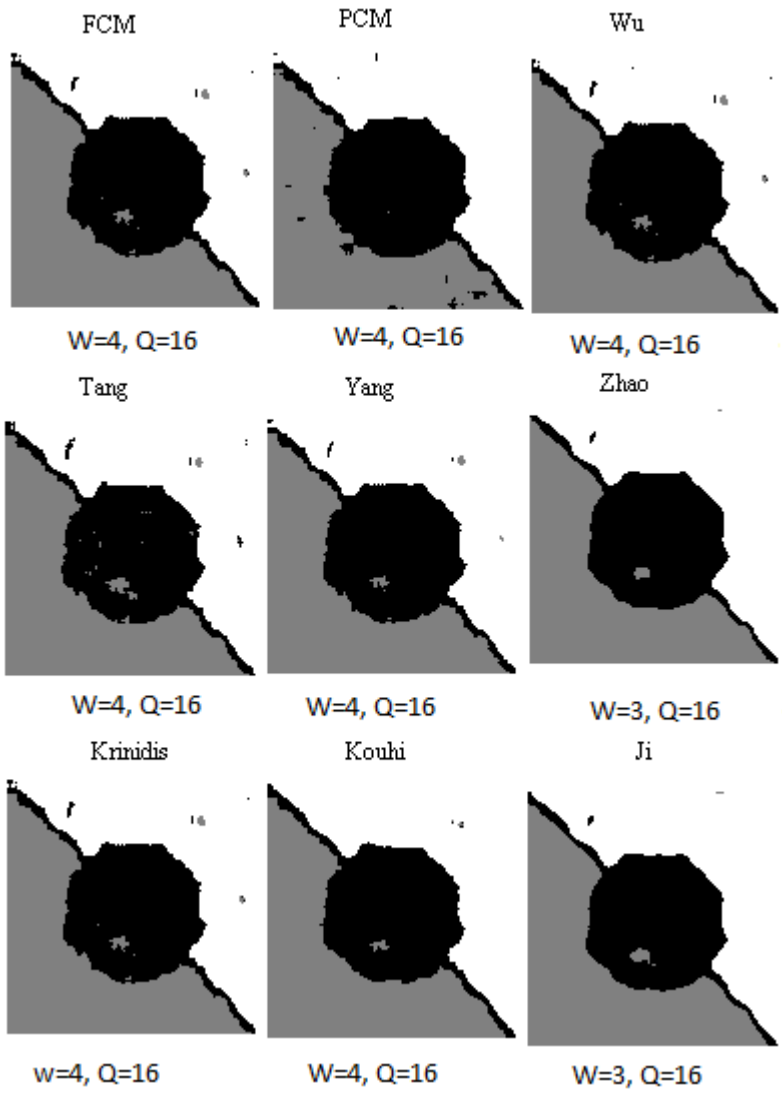

Figure (3.2) : Image test 1 segmentée par l'algorithme FCM et ses variantes.

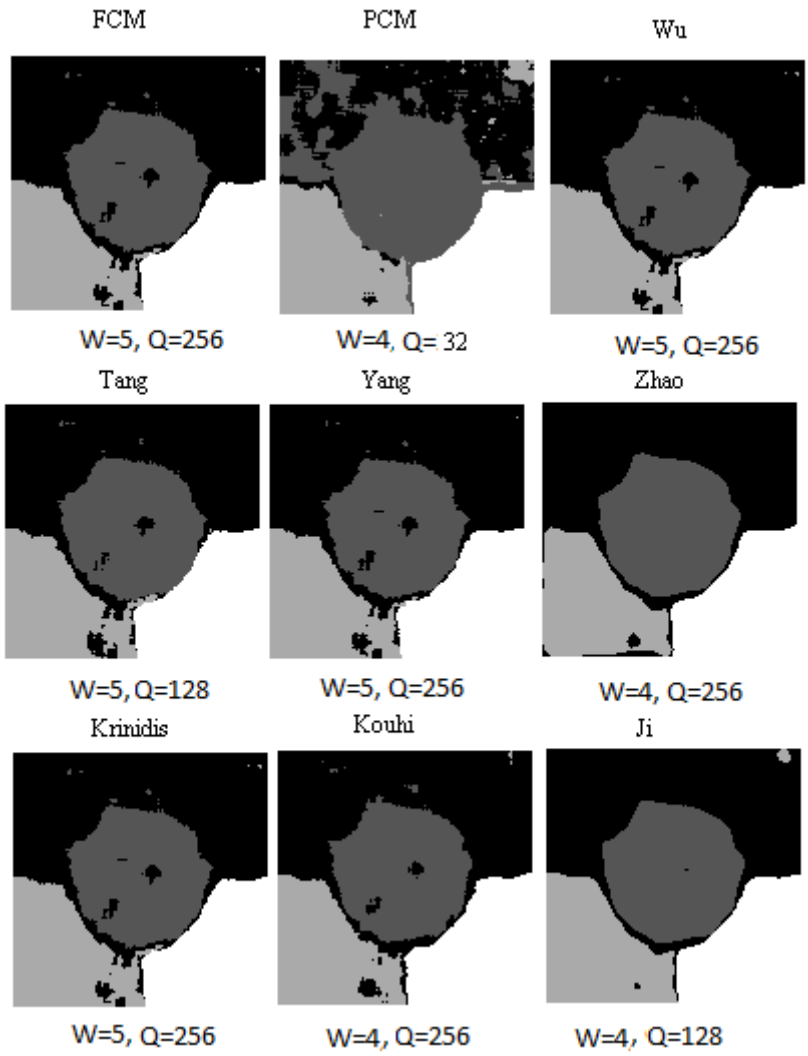

Figure (3.3) : Image test 2 segmentée par l'algorithme FCM et ses variantes.

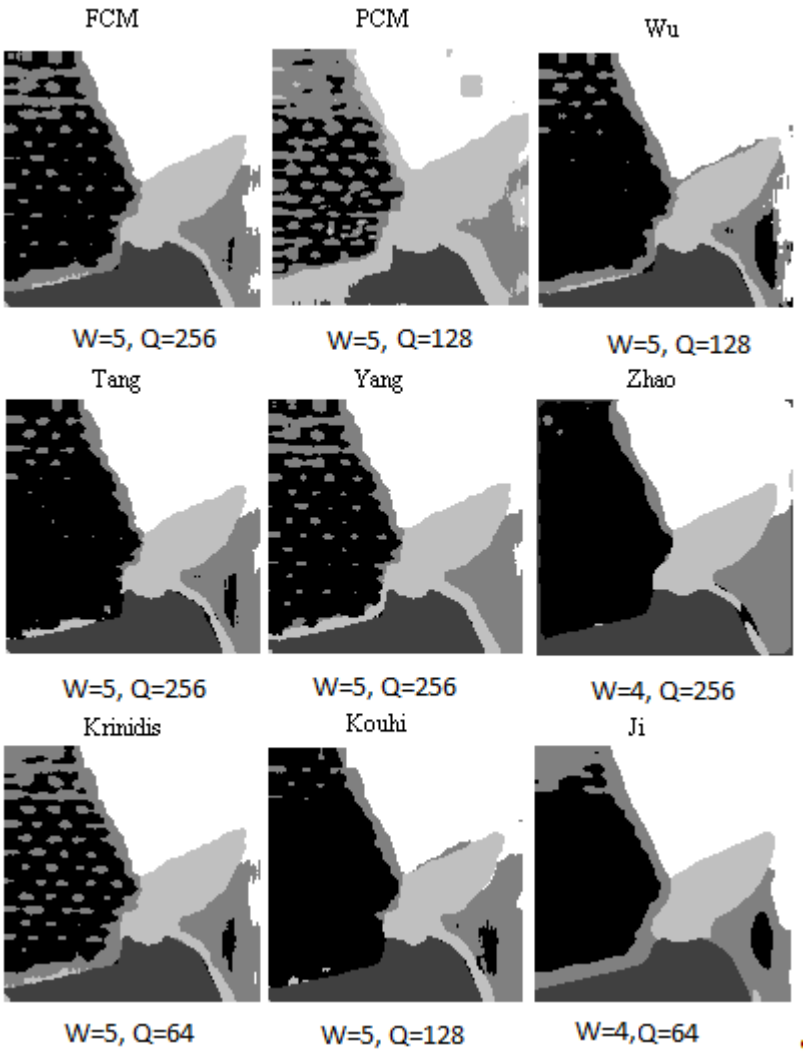

Figure (3.4) : Image test 3 segmentée par l'algorithme FCM et ses variantes.

3.5 Images bruitées

Afin de tester la sensibilité au bruit de l'algorithme FCM et ses variantes, nous avons introduit un bruit gaussien de variance $\sigma^2 = 0.005$ et $\sigma^2 = 0.01$, aux trois images tests.

Le tableau (3.3) montre la variation du taux de classification (Tc) en fonction de W et Q obtenus par les différentes méthodes sur l'image test 1 bruitée avec $\sigma^2 = 0.01$.

Le tableau (3.4) regroupe les valeurs des taux de classification obtenus sur les trois images bruitées ($\sigma^2 = 0.005$ et $\sigma^2 = 0.01$).

Les figures (3.5) à (3.10) montrent les images segmentées obtenues par chaque méthode sur chaque image test bruitée avec $\sigma^2 = 0.005$ et $\sigma^2 = 0.01$, respectivement.

W	Q	FCM	PCM	Wu et al	Tang et al	Yang et al	Zhao et Jiao	Krinidis et Chatzis	Kouhi et al	Ji et al
3	8	87.95	84.30	88.37	86.12	92.70	96.02	88.47	89.39	94.62
	16	90.47	84.87	90.47	**88.27**	**92.92**	95.47	90.39	91.55	94.63
	32	**91.82**	83.49	**91.83**	87.83	92.49	95.58	**91.70**	93.66	**95.49**
	64	90.20	**85.64**	90.10	85.46	91.51	**96.14**	90.54	92.30	94.51
	128	89.31	84.13	89.22	84.91	91.27	95.74	89.12	89.93	92.35
	256	88.15	85.02	89.91	76.90	89.06	93.93	89.87	91.28	92.92
4	8	**94.52**	**90.95**	**94.54**	91.46	**94. 96**	96.28	**94.52**	95.38	**96.15**
	16	94.01	88.95	94.01	**91.96**	94.77	**96.87**	94.01	95.27	95.62
	32	92.96	90.54	93.01	90.59	93.92	95.40	92.96	94.12	95.26
	64	93.04	87.99	93.08	90.38	94.23	95.70	93.04	94.66	95.58
	128	93.75	88.73	93.76	90.23	94.55	95.58	93.75	94.72	95.41
	256	93.12	89.81	93.05	90.86	94.02	95.76	93.13	94.51	95.48
5	8	**94.82**	92.83	**94.81**	**93.88**	94.96	95.17	**94.81**	95.28	95.03
	16	94.69	93.28	94.69	92.25	94.95	95.55	94.69	95.35	95.39
	32	94.65	92.80	94.65	91.79	**95.96**	**95.60**	94.66	95.00	95.40
	64	94.67	**93.90**	94.68	92.93	94.91	95.34	94.67	94.97	95.13
	128	94.05	91.18	94.07	91.38	95.56	95.27	94.08	94.50	95.11
	256	94.79	90.90	**94.81**	91.70	95.04	95.58	94.79	**95.43**	**95.43**

Tableau (3.3) : Variation du taux de classification (Tc) en fonction de W et Q obtenus par l'algorithme FCM et ses variantes sur l'image test 1 bruitée ($\sigma^2 = 0.01$).

Image test	σ^2	FCM	PCM	Wu et al	Tang et al	Yang et al	Zhao et Jiao	Krinidis et Chatzis	Kouhi et al	Ji et al
1	0.005	95.56	91.19	95.56	94.68	95.98	**96.46**	95.56	96.24	96.03
	0.01	94.01	88.95	94.01	91.96	94.77	**95.47**	94.01	95.27	94.63
2	0.005	95.93	91.19	95.93	95.35	96.02	95.67	95.93	95.01	**96.12**
	0.01	94.88	86.40	94.89	92.86	95.04	94.69	94.89	92.92	**95.55**
3	0.005	90.31	72.51	**91.66**	91.23	91.08	89.72	90.23	87.43	86.01
	0.01	85.47	67.46	85.32	82.14	**90.40**	88.97	85.92	86.67	84.91

Tableau (3.4) : Taux de classification obtenus par chaque algorithme FCM et ses variantes sur les trois images tests bruitées.

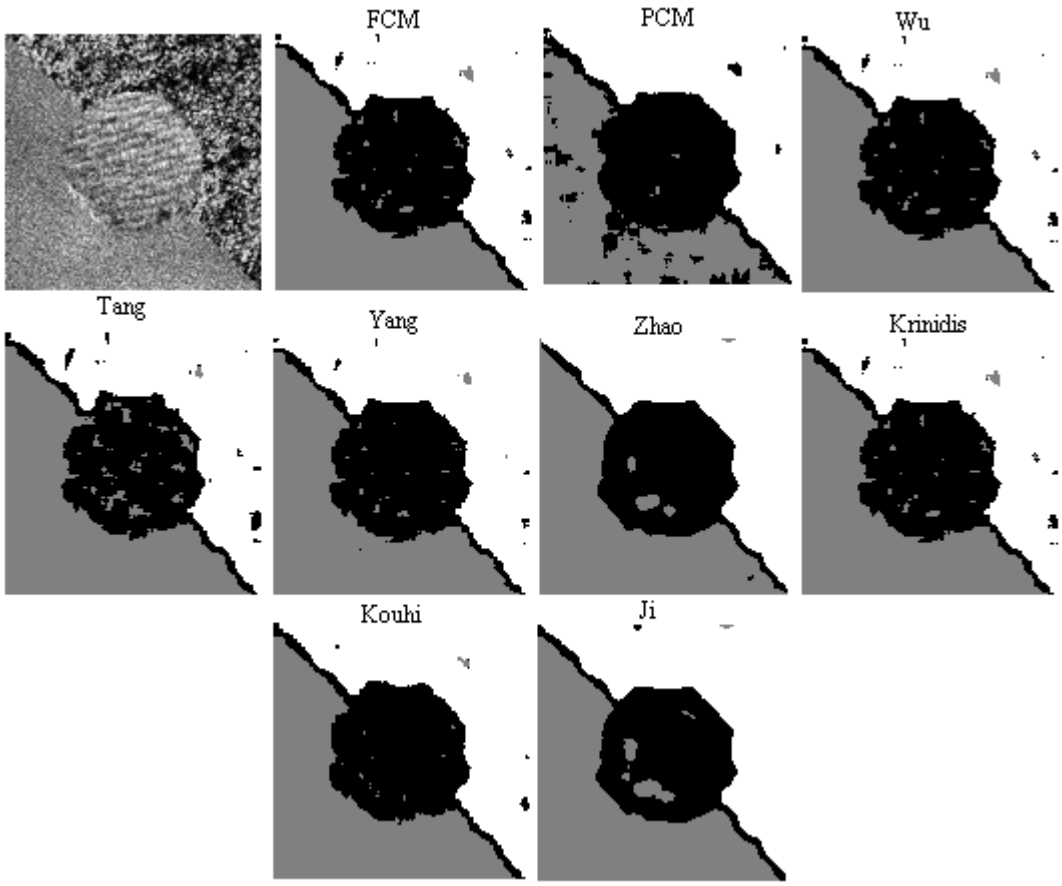

Figure (3.5) : Résultats de segmentation de l'image test 1 bruitée avec $\sigma^2 = 0.005$.

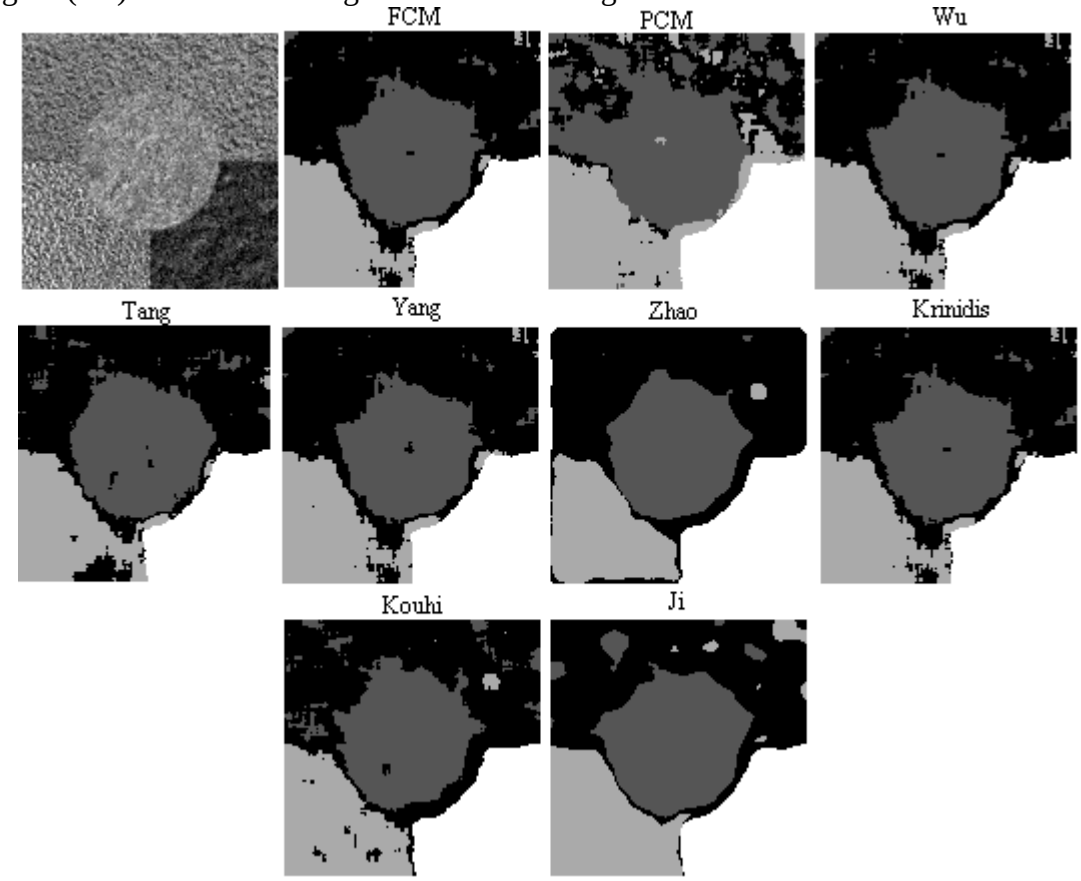

Figure (3.6) : Résultats de segmentation de l'image test 2 bruitée avec $\sigma^2 = 0.005$.

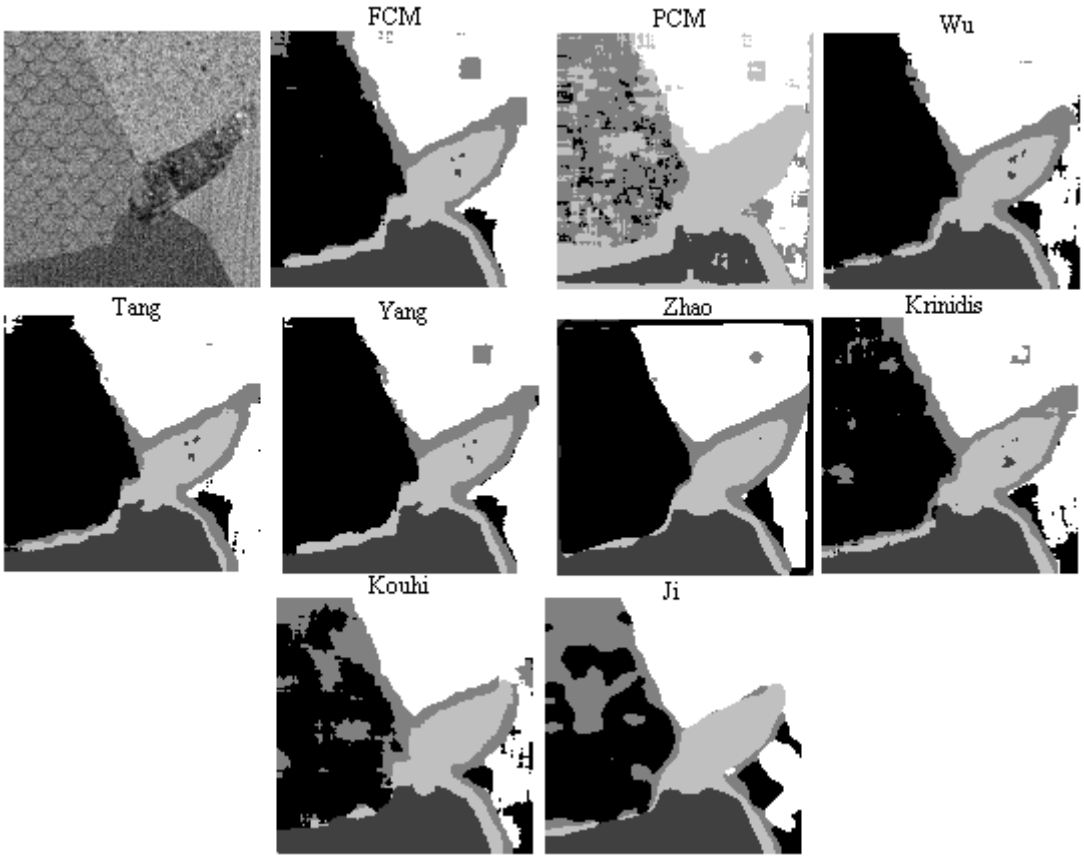

Figure (3.7) : Résultats de segmentation de l'image test 3 bruitée avec $\sigma^2 = 0.005$.

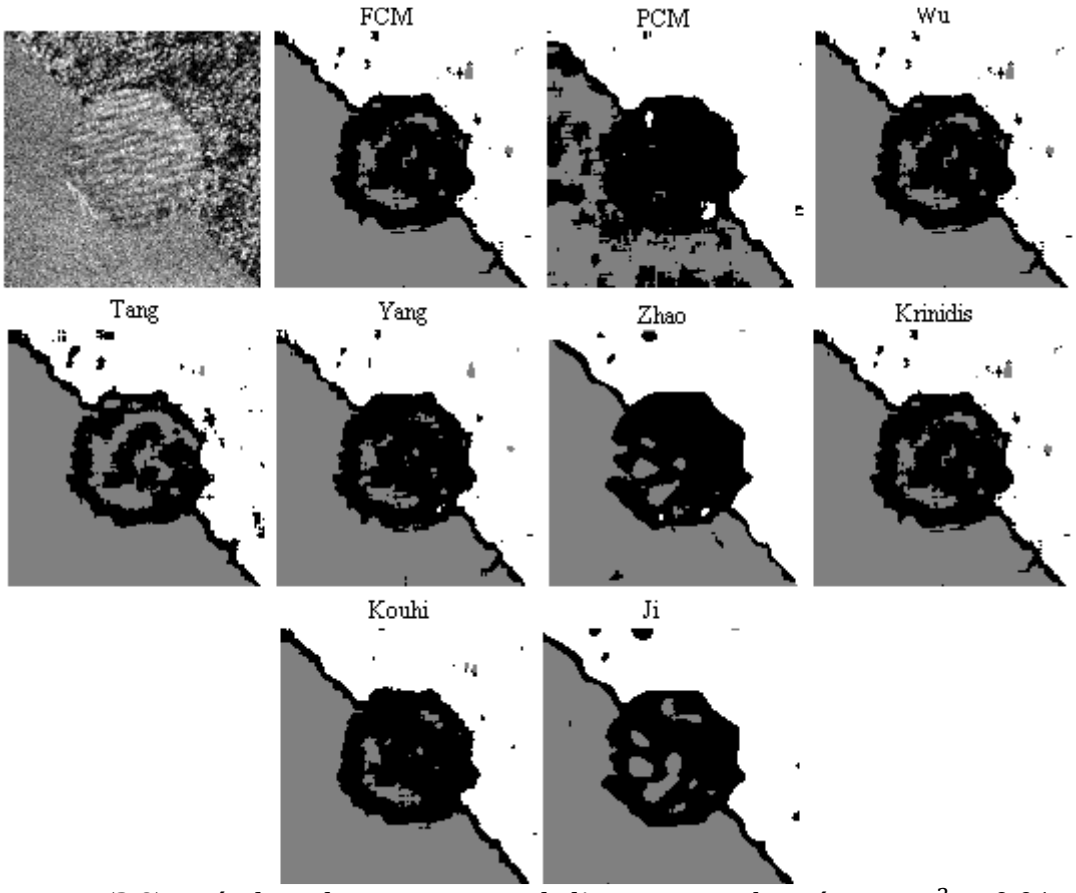

Figure (3.8) : Résultats de segmentation de l'image test 1 bruitée avec $\sigma^2 = 0.01$.

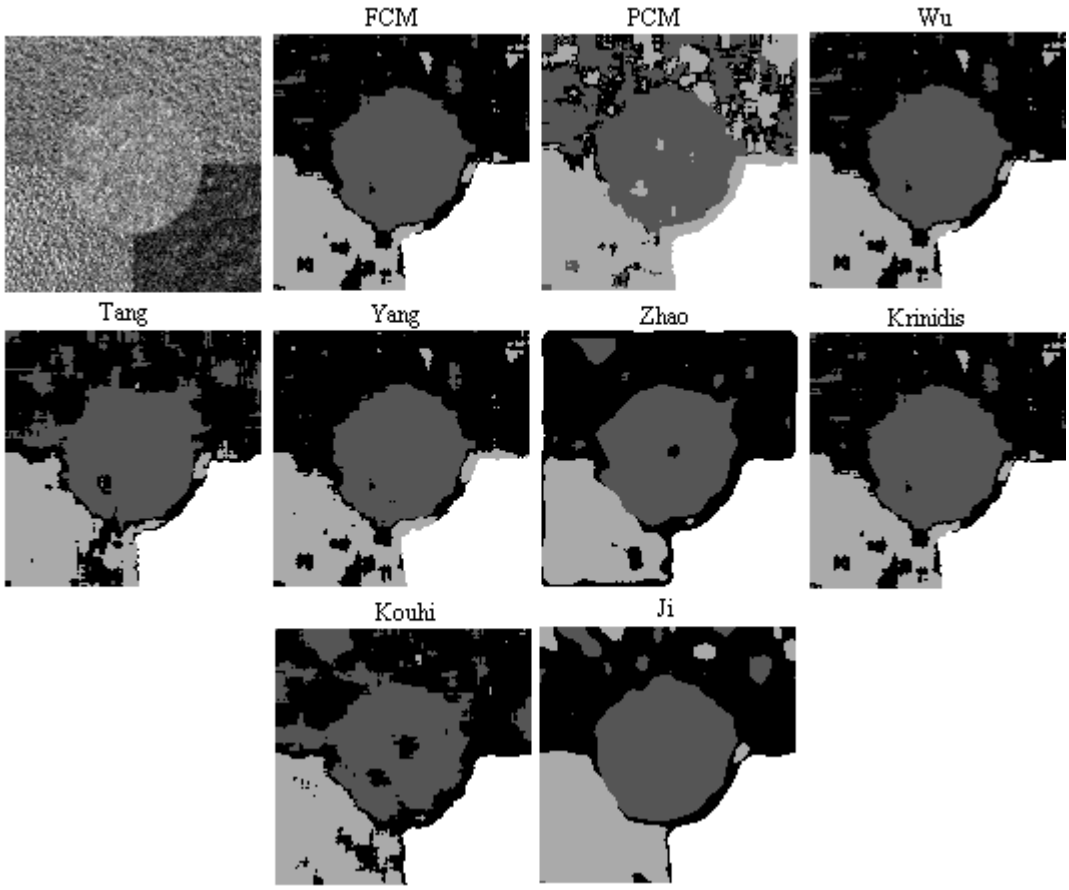

Figure (3.9) : Résultats de segmentation de l'image test 2 bruitée avec $\sigma^2 = 0.01$.

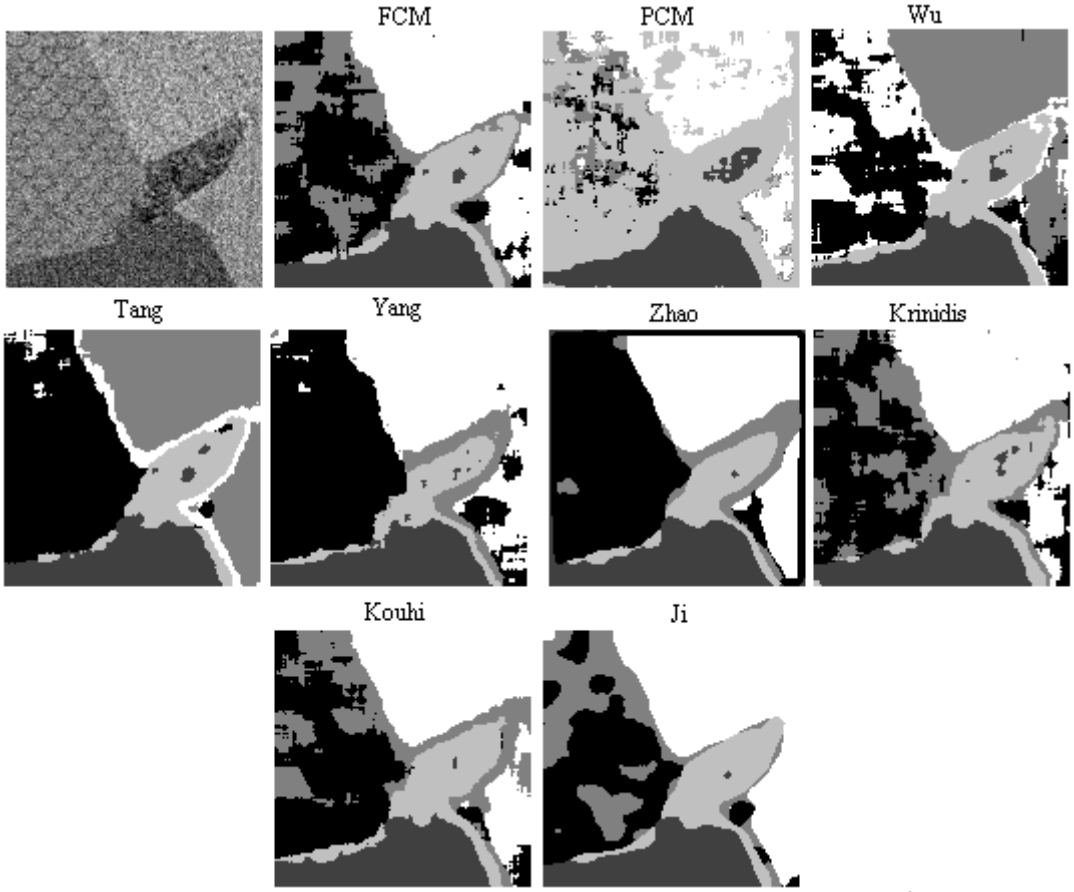

Figure (3.10) : Résultats de segmentation de l'image test 3 bruitée avec $\sigma^2 = 0.01$.

3.6 Interprétation des résultats

Les résultats de la segmentation des trois images synthétiques montrent que certaines méthodes FCM modifiées ont amélioré les performances du FCM standard alors que d'autres ont été moins efficaces comme l'algorithme PCM qui a donné un taux de classification inférieur à celui donné par FCM pour les trois images et les algorithmes de Krinidis et Chatzis et de Kouhi et al dans le cas de la segmentation de la troisième et la deuxième image, respectivement.

La méthode de Zhao et Jiao fournit les meilleurs taux de classification pour les trois images tests. Cette méthode affiche le meilleur taux de classification de 96.86% pour la première image et 96.58% pour la troisième image et un taux de 97.24% proche du meilleur taux (97.95%) délivré par la méthode de Ji et al pour la deuxième image.

Certaines méthodes comme celle de Kouhi et al ainsi que celle de Ji et al peuvent être performantes pour certaines images et médiocres pour d'autres images.

Les résultats fournis par les autres méthodes sont en moyenne médiocres. Ceci peut s'expliquer par le fait que ces méthodes confondent la texture et le bruit.

Concernant la sensibilité par rapport au bruit, la méthode de Zhao et Jiao ainsi que celle de Ji et al se sont montrées plus robuste en délivrant les meilleurs taux de classification pour la première et la deuxième image, respectivement. Cependant, dans le cas de la troisième image, la méthode de Wu et al a affiché le meilleur taux de classification de 91.66% pour $\sigma^2 = 0.005$ alors que c'est la méthode de Yang et al qui donne le meilleur taux (90.40%) pour $\sigma^2 = 0.01$.

Nous avons constaté également qu'un meilleur résultat de la segmentation d'une image donnée (avec ou sans bruit) pour des valeurs précises de W et Q ne peut être obtenu dans la segmentation d'une autre image pour ces mêmes valeurs des paramètres W et Q. Autrement dit, chaque méthode nécessite le réglage de ces deux paramètres.

D'un point de vue temps de calcul, les méthodes de FCM, PCM, Wu et al, Tang et al et celle de Yang et al sont les plus rapides car, celles-ci contrairement aux autres méthodes n'ont pas introduit dans leur fonction objectif un terme qui tient compte de l'information spatiale et des pixels voisins.

3.7 Conclusion

Nous avons présenté dans ce chapitre les résultats de la segmentation d'images texturées par classification floue, basée sur l'algorithme Fuzzy C-means et ses variantes.

Une série de tests a été effectuée sur trois images synthétiques afin de montrer l'apport de chacune des méthodes par rapport à l'algorithme FCM classique. Pour tester la sensibilité au bruit de tous ces algorithmes, nous avons bruité les trois images tests avec un bruit gaussien de variance égale à 1% et à 10% du nombre de niveaux de gris Q.

La méthode de Zhao et Jiao et celle de Ji et al fournissent généralement les meilleurs taux de classification pour les trois images tests et les méthodes de Zhao et Jiao, Ji et al, Wu et al et celle de Yang et al sont les moins sensibles au bruit.

Les résultats fournis par les autres méthodes sont en moyenne médiocres. Ceci peut s'expliquer par le fait que ces méthodes confondent la texture et le bruit.

Conclusion générale

Le travail présenté dans ce mémoire concerne le domaine du traitement d'images et plus précisément celui de la segmentation d'images texturées par classification floue basée sur l'algorithme Fuzzy C-Means et ses variantes.

Nous avons étudié dans ce mémoire un nombre exhaustif de ces variantes, qui ont été proposées dans la littérature pour remédier au problème de la sensibilité au bruit de l'algorithme FCM classique. Celles-ci tentent soit à changer la fonctionnelle à minimiser, soit à définir une autre métrique ou soit encore à modifier l'influence du facteur flou m. Parmi ces variantes, nous avons implémenté huit d'entre elles à savoir les algorithmes Krinidis et Chatzis, Kouhi et al, Yang et al, Zhao et Jiao, Ji et al, Keller et Krishnapuram (PCM), Wu et al et enfin Tang et al.

Certaines de ces méthodes ont été conçues dans le cade général, c'est-à-dire pour classer des données multidimensionnelles et d'autres ont été développées spécifiquement pour la segmentation d'images couleur ou en niveaux de gris. L'application de ces méthodes dans le cadre de la segmentation d'images texturées a montré que certaines méthodes ont amélioré la qualité de la segmentation par rapport à celle obtenue par FCM, cependant d'autres ont été moins efficaces.

Sur tous les tests effectués, la méthodes de Zhao et Jiao demeure la plus performante alors que celle de Ji et al ainsi que celle de Kouhi et al donnent de bons résultats pour certaines images et des résultats médiocres pour d'autres.

D'un autre côté, les méthodes de Zhao et Jiao, Ji et al, Wu et al et celle de Yang et al se sont avérées les moins sensibles au bruit.

Si ces méthodes ont montré leur efficacité pour des applications auxquelles, elles ont été conçues, c'est-à-dire en segmentation d'images en niveaux de gris ou en couleur, elles restent cependant moins efficaces dans la segmentation d'images texturées car elles ont tendance à confondre texture et bruit.

Nous avons relevé dans la littérature plusieurs méthodes modifiant l'algorithme FCM. Huit d'entre elles seulement ont été implémentées et testées. Il serait donc souhaitable d'étendre notre étude à d'autres variantes comme celles qui agissent sur le facteur flou ou celles qui

utilisent un autre type de distance ou encore combiner des méthodes appartenant à des catégories différentes.

Nous avons abordé dans ce travail le problème du bruit, mais pas celui de l'initialisation automatique des centres des classes ni celui du nombre de classes. Comme perspectives à ce travail, nous préconisons d'appliquer une des méthodes d'initialisation automatique des centres des classes.

Nous avons utilisé dans notre étude les attributs de texture extraits à partir de l'histogramme et des matrices de cooccurrence. L'utilisation d'autres attributs de texture pour effectuer une segmentation avec l'algorithme FCM et ses variantes doit être envisagée afin d'affirmer les résultats obtenus.

Bibliographie

[1] B. Hadjira et M. Soumia, Segmentation d'images par croissance de régions, Ecole Nationale de l'Informatique, 2010.

[2] Michèle Gouiffés, Segmentation d'images.

[3] M. Unser, Description statistique de la texture. PhD thesis, EPFL, Lausanne, 1984

[4] M. Kerchit et M. Kechir, Segmentation d'images basée sur le seuillage flou, université Mouloud Mammeri de Tizi-Ouzou, 2010.

[5] S. Horowitz and S. Pavlidis, Picture segmentation by a directed split and merge procedure, second international conference on pattern recognition, pp. 424-433, 1974.

[6] I. Sebari et D-C. HE, Les approches de segmentation d'images par coopération régions-contours, Revue télédétection, 2007, vol.7, pp.499-506.

[7] L. Paulhac, Outils et méthodes d'analyse d'images 3D texturées : application à la segmentation d'images échographiques, université François RABELAI de Tours

[8] S. Ghandour, Segmentation d'images couleurs par morphologie mathématique : application aux images microscopiques, université de Toulouse, 2010

[9] H. Benoit-Cattin, T. Zouagui et C. Odet, Une vision fonctionnelle de la segmentation d'images.

[10] C. Murie, Segmentation d'images couleurs par classification pixellaire et hiérarchie de partitions, université de CAEN, 2005.

[11] N-E. Lasmar, Modélisation stochastique pour l'analyse d'images texturées : Approches Bayésiennes pour la caractérisation dans le domaine des transformées, université de Bordeaux I, 2012.

[12] R. Haralick, K. Shanmugan et I. Dinstein, Textural features for images classification. IEEE transaction on systems, Man and Cybernetics, 610-621, 1973.

[13] Segmentation 2

[14] L. A. Zadeh, Fuzzy sets, Information and Control 8, 338-353, Berkeley California, 1965.

[15] S. Elkosantini, Introduction à la logique floue : les concepts fondamentaux et applications.

[16] S. Ambapour, Théorie des ensembles flous : application à la mesure de la pauvreté au Congo.

[17] J. M. Mendel et R. I. Bob John, Type-2 fuzzy sets made simple, IEEE transaction on fuzzy systems, vol. 10, No. 2, April 2002.

[18] C. V. Jawahar et A. K. Ray, fuzzy statistics of digital images, IEEE signal processing letters, vol. 3, No. 8, August 1999.

[19] I. Bloch et H. Maitre, les méthodes de raisonnement dans les images, Ecole nationale supérieure des télécommunications, Paris. 2004

[20] W. B. Tao, J. W. Trian et J. Liw, Image segmentation by level thresholding based on maximum Fuzzy entropy and genetic algorithm, Pattern recognition, vol. 24, pp. 3069-3078, 2003.

[21] H. D. Cheng, J. R. Chen et J. Li, Threshold selection based on Fuzzy C-partition entropy approach, Pattern recognition, vol. 31, no. 7, pp. 857-870, 1998.

[22] L. K. Huang et M. J. J. Wang, Image thresholding by minimizing the measure of fuzziness, Pattern recognition, vol. 38, no. 1, pp. 41-51, 1995.

[23] T. Chaira et A. K. Pay, Threshold selection using fuzzy set theory, Pattern recognition letters, vol. 25, no. 1, pp. 865-874, 2004.

[24] J. J. Buckley et E. Eslami, Fuzzy plane geometry I : points and lines. Fuzzy sets and systems, 86: 179-187, 1997.

[25] C-Y. Yu, Y. Li, A-L. Liu et J-H. Liu, A novel modified Kernel Fuzzy C-Means clustering algorithm on image segmentation, IEEE Computer society, 2011.

[26] S. A. Mingoti et J. O. Lima, comparing SOM neural net work with fuzzy c-means, k-means and traditional hierarchical clustering algorithms. European Journal of Optimal Research. Vol. 174, pp. 1742-1759, 2006.

[27] J. C. Bezdek, pattern recognition with fuzzy objective function. Plenum press, New York, 1981

[28] M. F. Saad, F. Karem et M. A. Alimi, Classification des images IRM.

[29] D. E. Gustafson et W. C. Kessel, Fuzzy clustering with a fuzzy covariance matrix. IEEE, Conference of Decision and Control, San Diego California, pp. 761-766, 1979.

[30] J. Gath et A. B. Geva, Unsupervised Optimal fuzzy clustering. IEEE Trans. Pattern anal. Mach. Intel. 11, 773-781, 1989.

[31] S. Shen, W. Sandhan, M. Granat et A. Steer, MRI fuzzy segmentation of brain tissue using neighborhood attraction with neural net work optimization. IEEE Trans. Intel. technol. vol. 9, pp. 459-467, 2005.

[32] M. Forouzanfar, N. Forghani et M. Teshnehleb, Parameter optimization of improved fuzzy c-means clustering algorithm for MR image segmentation. Engineering App. of Art. Intelligence 2010.

[33] X. Wang, Y. Wang et L. Yang, Improving fuzzy c-means algorithm based on feature-weight learning. Pattern recognition, vol.25, pp. 1123-1132, 2004.

[34] D. M. Tsai et C. C. Lin, fuzzy c-means based clustering for linearly and non linearly separable data. Pattern recognition 44, 1750-1760, 2011.

[35] http: //www.iro.umontreal.ca/vincentp/ift3390 (Fondement de l'apprentissage machine).

[36] P. Kaur, A. k. Soni et A. Gosain, Robust kernalized approach to clustering by incorporating new distance measure. Engineering App. of art. Int. Elsevier 2012. pp. 833-847.

[37] H. Frigui, et R. Krishnapuram, clustering by competitive agglomeration, Pattern recognition, Vol. 30, No. 7, pp. 1109-1119

[38] D. L. Pham, Fuzzy clustering with spatial constraints. IEEE ICIP (2002).

[39] R. Krishnapuram et J. Keller, A possibilistic approach to clustering. IEEE Trans. Fuzzy Syst. 1 (2), 98-110.

[40] K-L. Wu, J. Yu et M-S. Yang, A novel fuzzy clustering algorithm based on fuzzy scatter matrix with optimality sets. Pattern Recognition letters, vol. 26, pp. 639-652, 2005.

[41] M. N. Ahmed, S. M. Yamani, N. Mohammed, A. Farag et T. Moriarty, A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. IEEE Trans. On medical imaging, 21(3), pp. 193-199, 2002

[42] S. Chen et D. Zhang, Robust image segmentation using fuzzy c-means with spatial constraints based on new kernel-induced distance measure. IEEE Trans. Trans. Syst. Man. Cybern, vol. 34, pp. 1907-1916, 2004.

[43] L. Szilagyi, Z. Benyo, S. Szilagyi et H. Adam, MR brain image segmentation using an Enhanced fuzzy c-means algorithm. In proc. 25th Annu. Int. Conf. IEEE, pp. 17-21, 2003.

[44] A. Kouhi, H. Sayedarabi et A. Aghagolzadeh, A modified fuzzy c-means algorithm for MRI brain image segmentation. IEEE, 2011.

[45] J. Kang, L. Min, Q. Luan, X. L. Li et J. Liu, Novel modified fuzzy c-means algorithm with applications. Digital Image Processing, 2008.

[46] W. Cai, S. Chen et D. Zhang, Fast and robust fuzzy c-means clustering algorithm incorporating local information for image segmentation. Pattern recognition 40, 825-838, 2007.

[47] R. Li et M. Mukaidono, A maximum entropy approach to fuzzy clustering estimation. IEEE, International Conference on Fuzzy Systems, 1995

[48] Z. Yang, F-L. Chung et W. Shitong, Robust fuzzy clustering based image segmentation. Applied soft computing, 2008.

[49] C. Tang, S. Wang et W. Xu. New fuzzy c-means clustering model based on the data weighted approach. Data & Knowledge Engineering. Vol. 69, pp. 881–900, 2010.

[50] S. Krinidis et V. Chatzis, A robust fuzzy local information c-means clustering algorithm. IEEE, Trans. On image processing. Vol.19. No. 5, 2010.

[51] C. Gauge et S. Sasi, Modified fuzzy c-means clustering algorithm with spatial distance to cluster center of gravity. IEEE, Inter. Symposium on Multimedia, 2010.

[52] L. Zhu, F-L. Chung et S. T. Wang, Generalized fuzzy c-means clustering algorithm with improved fuzzy partition. IEEE, Trans. Syst. Man and Cybern. Vol. 39, pp. 578-591, 2009.

[53] F. Zhao et L. Jiao, Spatial improved fuzzy c-means clustering algorithm for image segmentation. IEEE, Inter. Conf. on Electr. Inf. Tech. 2011

[54] A. Benaichouche, H. Oulhadj et P. Siarry, Improved spatial fuzzy c-means clustering for image segmentation using PSO initialization, Mahalanobis distance and and post segmentation correction. Digital signal processing 23, 1390-1400. 2013**.**

[55] Z. Ji, Y. Xia, Q. Chen, Q. Sun, D. Xia et D D. Feng, Fuzzy c-means clustering with weighted image patch for image segmentation. Applied Soft Computing 12, 1659–1667, 2013.

[56] P. Rousseeuw, E. Trauwaert et L. Kaufman, Fuzzy clustering with high contrast. Journal of Computational and Applied Mathematics, 64, 81-90, 1995.

[57] F. Klawonn et F. Hoppner, An alternative approach to the fuzzifier in fuzzy clustering to obtain better clustering results. Proceedings 3rd Eusflat, pp730-734

[58] F. Zhao, L. Jiao et H. Liu, Kernalized fuzzy c-means clustering with spatial information for image segmentation. Digital Signal Processing 23, 184-199, 2013.