Unsupervised cooperative partitioning approach of hyperspectral images for decision making

Akar Taher

ANNÉE 2014

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du Signal et Télécommunications

École doctorale MATISSE (Mathématiques Télécommunications Informatique Signal Systèmes Électronique)

présentée par

Akar TAHER

Préparée à l'unité de recherche 6164 IETR
(Institut d'Électronique et de Télécommunications de Rennes)

Équipe : Traitement des Signaux et Images Multicomposantes et Multimodales

---

Intitulé de la thèse :

Approche coopérative et non supervisée de partitionnement d'images hyperspectrales pour l'aide à la décision

Thèse soutenue à l'Université de Rennes 1
le 20, Octobre, 2014

Devant le jury composé de :

Jean-Marc BOUCHER
Professeur à Télécom Bretagne/ *rapporteur*

Olivier COLOT
Professeur à Université de Lille 1/ *rapporteur*

Philippe BOLON
Professeur à Polytech Annecy-Chambéry/ *examinateur*

Jean-Marc CHASSERY
Directeur de recherche CNRS+ GIPSA LAB/ *examinateur*

Claude CARIOU
Maître de conférences à l'université de Rennes 1 / *examinateur*

Kacem CHEHDI
Professeur à l'université de Rennes 1/ *directeur de thèse*

ACKNOWLEDGEMENTS

I would like first and before all to thank Professor Kacem CHEHDI, the leader of TSI2M team of Institut d'Électronique et de Telecommunications de Rennes (IETR), University of Rennes I, who supervised this study. I also wish to express proudly my deep gratitude for his absolute availability, scientific rigor, and support along the years of this research, which are rarely matched.

I would like to thank also the Ministry of Higher Education (MOHE) of the Kurdistan Regional Government (KRG), for their financial support. I would like to extend my thanks for KOYA University to give me the opportunity of making my PhD.

Thanks are due to Associate Professor Claude CARIOU, University of Rennes I, for his help and cooperation.

Appreciations go to Professor Jean-Marc BOUCHER, Telecom Bretagne and Professor Olivier COLOT, University of Lille 1, who helpfully accepted to review and validate this study as reporters.

I also, would like to thank, all members of the jury, for agreeing to review my work.

I finally, would like to thank all of my family members for their support.

Notations

I : *Image*

N : *Number of pixels in image I*

$X = \{x_1, ..., x_N\}$, *set of N pixels or elements to classify.*

$F = \{a_1, a_2, ..., a_{Nf}\}$, *set of Nf features.*

a_{ji} *the value of the feature* a_j *for the pixel* x_i

$F_i = \left[a_{1i}, a_{2i} ..., a_{Nfi}\right]$, *vector of Nf features representing the pixel* x_i

$F^j = \left[a_{j1}, a_{j2}, ..., a_{jN}\right]^T$, *vector of values of feature* a_j *for all N pixels*

C_i : *Class i*

NC : *Number of classes*

$\hat{NC}$: *Number of classes estimated*

NC_i : *Card* (C_i)

I_M : *Multithresholded image*

I_R : *Partitioned image*

$g(C_i)$: *Center of gravity of* C_i

N_{COL} : *Number of columns in an image*

N_{LIN} : *Number of lines or row in an image*

NG : *Number of gray levels in image I*

$g_l(C_i)$: *Average gray level of class* C_i

$g_l(x)$: *Gray level of pixel x in image I*

$\hat{g}_l(x)$: *Average gray level of pixels around x*

$d(.,.)$: *Euclidean distance*

$||.||$: *Euclidean norm*

ND : *Number of directions*

$P_{\theta d}(i, j)$: *The entries of cooccurence matrix*

Cov : *Covariance matrix*

W : *Analysis window*

Résumé Substantielle

Introduction Générale

Les nombreux travaux menés dans la littérature traitant le problème du partitionnement des images, montrent que ce problème est difficile et loin d'être résolu.

En général, l'application d'une technique simple n'est pas suffisante pour analyser correctement l'ensemble du contenu des images. En effet, de nombreuses études effectuées dans notre laboratoire montrent que l'utilisation d'une seule méthode dans des domaines d'application différents ne donne pas de résultats pertinents. Le problème majeur des méthodes de classification est leur incapacité à s'adapter au contenu local de l'image. En outre, avec l'avènement récent des systèmes d'acquisition d'image de pointe, la scène est mieux décrite et la taille des images devient de plus en plus grande (imagerie multispectrale et hyperspectrale). L'analyse et l'interprétation de ce type d'images est donc de plus en plus fastidieux et complexe.

Le problème du partitionnement des images est un problème mal posé, et aucune méthode générique ne peut prétendre donner un résultat de partitionnement optimal. Les erreurs de partitionnement sont inévitables (sur ou sous-partitionnement) ; le sur-partitionnement génère des régions qui ne correspondent pas aux objets réels de la scène, et le sous-partitionnement ne distingue pas tous les objets d'une scène.

Pour surmonter ce problème et trouver une solution, plusieurs chercheurs ont proposé l'utilisation de schémas coopératifs, qui combinent plusieurs méthodes pour partitionner une image. Ce processus de coopération exploite la redondance et la complémentarité du contenu de l'information dans l'image, permettant ainsi une meilleure compréhension du contenu informationnel d'une image. C'est pourquoi nous avons décidé de développer un système de coopération adaptatif pour le partitionnement des images hyperspectrales. La coopération est réalisée par plusieurs méthodes qui s'adaptent aux régions uniformes et texturées de l'image à partitionner.

L'objectif de cette thèse est donc de développer un système coopératif et adaptatif robuste pour classer les pixels des images hyperspectrales. Par système coopératif, nous entendons un système qui exploite plus d'une méthode de partitionnement. L'adaptativité concerne ici l'extraction des caractéristiques des pixels en fonction de la nature des régions.

Cette thèse est divisée en deux parties: la première est consacrée aux travaux de l'état de l'art, et la seconde à la présentation du système de classification développé.

Première Partie : État de l'art

La première partie est consacrée aux travaux de l'état de l'art portant sur le partitionnement des images et les critères d'évaluation. Les méthodes de partitionnement sont soit non coopératives, tels que les algorithmes génétiques, Fuzzy C-means (FCM), Linde-Buzo-Gray algorithme (LBG), Artificial Neural Network (ANN), *k*-means, et Affinity Propagation (AP), ou des approches coopératives qui combinent les méthodes non coopératifs ci-dessus. Des études expérimentales sont effectuées pour analyser les méthodes non-coopératives et montrer leurs limites. Dans cette partie, nous présentons aussi un examen des critères d'évaluation non supervisés.

Cette partie est organisée en deux chapitres. Le premier est consacré à la description des méthodes de classification de manière générale. Il présente de manière détaillée les méthodes semi-supervisées et les méthodes non supervisées, et d'autre part les méthodes coopératives de classification. Par ailleurs, le second chapitre comprend un état de l'art dédié à l'analyse des critères d'évaluation non supervisés pour évaluer les résultats de la classification.

Seconde Partie : Système développé

Dans la seconde partie de cette thèse, les différents modules du système de partitionnement développé sont présentés en détail. Chaque module est évalué et

validé séparément sur plusieurs images synthétiques et réelles mono et multi-composantes. Le système développé est également comparé à d'autres méthodes non coopératives et coopératives. Enfin, le système est également testé sur deux applications réelles relatives à la gestion du problème de l'environnement.

L'architecture du système développé pour le partitionnement d'images monocomposantes est présentée en Figure A. La Figure B présente son extension aux cas des images multicomposantes (multispectrales et hyperspectrales).

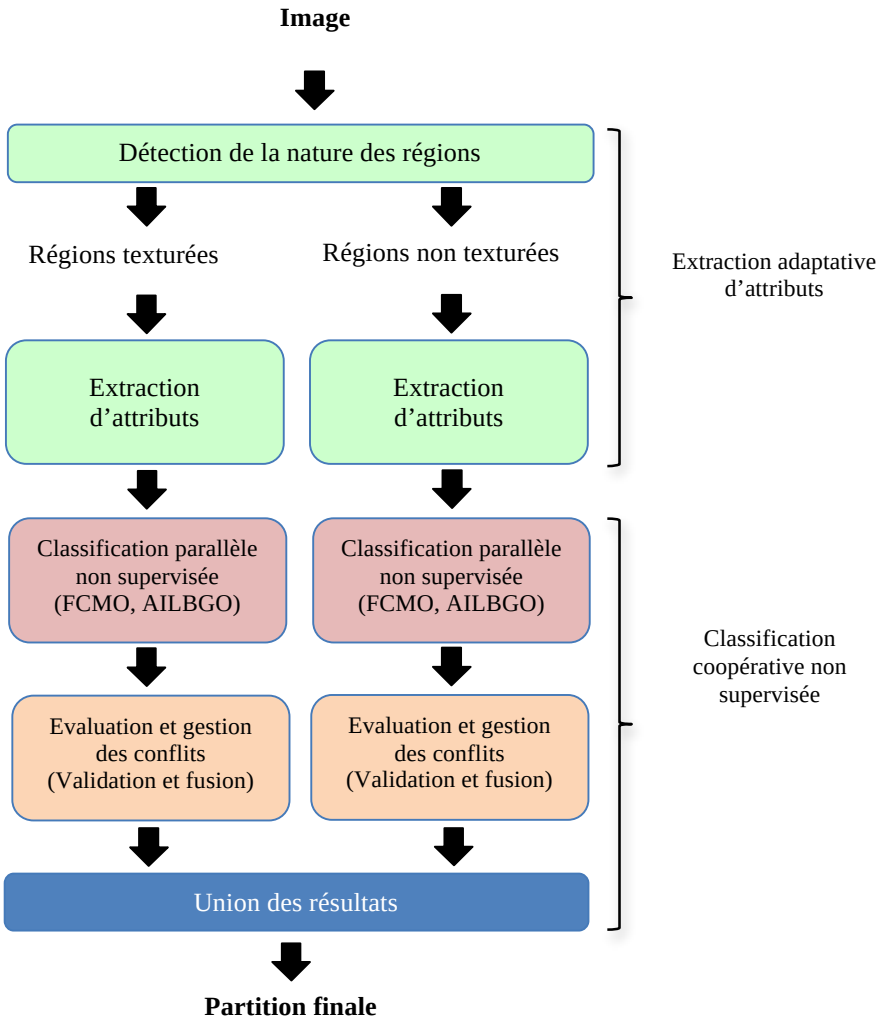


Figure A : Architecture du système coopératif de classification cas d'images monocomposantes

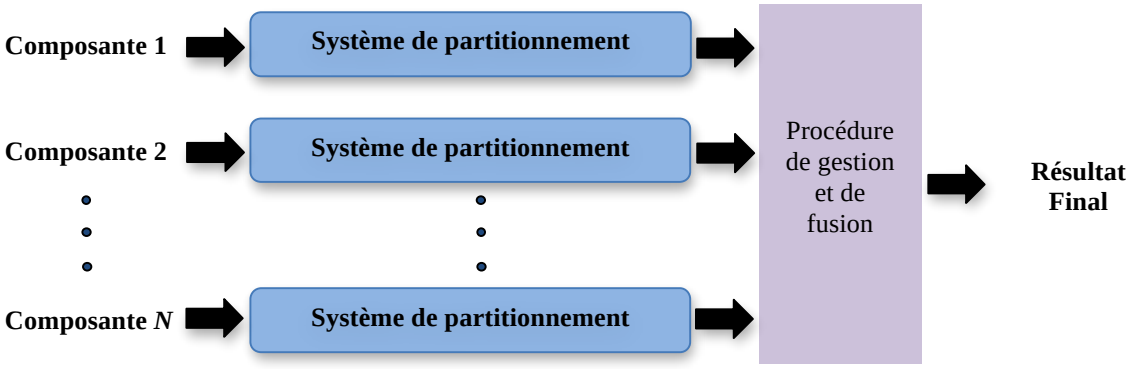


Figure B: Extension du système de la Figure A aux cas des images multicomposantes.

Cette partie est organisée également en deux chapitres. Le premier est consacré à la détection de la nature des régions et à l'extraction des attributs tenant compte de la nature de la région détectée (module 1). Il s'agit de partitionner l'image en deux types de régions texturées et non texturées, puis de caractériser les pixels en fonction de leur appartenance à ces régions. Plusieurs attributs de texture sont utilisés pour les pixels appartenant aux régions texturées, tandis que la moyenne locale est utilisée pour les pixels appartenant aux régions non texturées.

Le second chapitre présente les détails sur l'approche de classification coopérative, non supervisée, et non paramétrique. Ce chapitre inclut l'optimisation des algorithmes de classification FCM et AILBG (module 2), le processus d'évaluation et de gestion des conflits (module 3), et la réunion des résultats de partitionnement des régions texturées et non texturées (module 4). En outre, il inclut également l'évaluation du système développé sur deux applications réelles. Le descriptif de ces trois modules est présenté ci-dessous:

Le *deuxième module* fait coopérer parallèlement deux classifieurs optimisés : Fuzzy C-means (FCM), et l'algorithme Adaptatif Incrémental Linde-Buzo-Gray (AILBG) appelés respectivement FCMO et AILBGO, pour partitionner chaque composante. Pour rendre ces algorithmes non supervisés, le nombre de classes est estimé suivant un critère basé sur la dispersion moyenne pondérée des classes.

Le *troisième module* évalue et gère suivant deux niveaux les conflits des résultats de classification obtenus par les algorithmes FCMO et AILBGO. Le premier identifie les pixels classés dans la même classe par les deux algorithmes et les reportent directement dans le résultat final d'une composante. Le second niveau utilise un algorithme génétique (GA), pour gérer les conflits entre les pixels restant.

Le *quatrième module* consiste à reporter les résultats de classification des régions texturées et non texturées dans la même image.

Dans le cas des images multicomposantes, les trois premiers modules sont appliqués tout d'abord sur chaque composante indépendamment. Les composantes adjacentes ayant des résultats de classification fortement similaires sont regroupées dans un même sous-ensemble et les résultats des composantes de chaque sous-ensemble sont fusionnés en utilisant le même GA. Le résultat de partitionnement final est obtenu après évaluation et fusion par le même GA des différents résultats de chaque sous-ensemble.

Le système développé est testé avec succès sur une grande base de données d'images synthétiques (mono et multicomposantes) et également sur deux applications réelles : la classification des plantes invasives et la détection des Pins.

Table of contents

ABSTRACT

Hyperspectral and more generally multicomponent images are complex images, and cannot be successfully partitioned using a single classification method. The existing non-cooperative classification methods, parametric or nonparametric can be categorized into three types: supervised, semi-supervised and unsupervised. Supervised parametric methods require *a priori* information and also require making hypothesis on the data distribution model. Semi-supervised methods require some *a priori* knowledge (e.g. number of classes and/or iterations), while unsupervised nonparametric methods do not require any *a priori* knowledge.

Applying several non-cooperative methods on the same image is very unlikely to give identical partitions, and each result contains correct and incorrect information. For this reason, in this thesis an unsupervised cooperative and adaptive partitioning system for hyperspectral images is developed. Its originality relies on *i)* the adaptive nature of the feature extraction *ii)* the two-level evaluation and validation process to fuse the results, *iii)* the non requirement of training samples or the number of classes. This system is composed of four modules:

The *first module* classifies automatically the image pixels into textured and non-textured regions, and then different features of pixels are extracted according to the region types. Texture features are extracted for the pixels belonging to textured regions, and the local mean feature for pixels of non-textured regions.

The *second module* consists of an unsupervised cooperative partitioning of each component, in which pixels of the different region types are classified in parallel via the features extracted previously using optimized versions of Fuzzy C-Means (FCM) and Adaptive Incremental Linde-Buzo-Gray algorithm (AILBG) noted respectively as FCMO and AILBGO. For each algorithm the number of classes is estimated according to the weighted average dispersion of classes.

The *third module* is the evaluation and conflict management of the intermediate classification results for the same component obtained by the two classifiers. To obtain a final reliable result, a two-level evaluation is used; the first one identifies the pixels classified into the same class by both classifiers and report them directly to the final classification result of one component. In the second level, a genetic algorithm (GA) is used to remove the conflicts between the invalidated remaining pixels.

The *fourth module* unifies the results of textured and non-textured regions in the same labeled image.

In the case of a multicomponent image, the system handles all the components in parallel; where the above modules are applied on each component independently. The results of the different components are compared, and the adjacent components with highly similar results are grouped within a subset and fused using the same GA. To get the final partitioning result of the multicomponent image, the intermediate results of the subsets are evaluated and fused by GA.

The system is successfully tested on a large database of synthetic images (mono and multicomponent) and also tested on two real applications: the classification of invasive plants and the Pine trees detection.

Key words: *partitioning, classification, non-parametric, parallel cooperation, unsupervised, estimation, evaluation, validation, fusion, hyperspectral, image, validation, invasive vegetation, Pine trees.*

General Introduction

Images are one of the most important modes of communication. An image is a planar representation of a scene or an object in the space. There exist many types of images such as monochrome, color, multispectral and hyperspectral images depending on the number of components they contain.

In the recent years, the limitations of traditional RGB (color) imaging have become more and more evident as the requirements in terms of image quality are being raised, and new uses and applications are being conceived within the digital imaging field. At the same time, hyperspectral imaging has been emerging as a technology that allows the acquisition of up to several hundred of components of the same scene corresponding to its spectral decomposition. This large amount of information enables recognizing the content of the image with precision. Hyperspectral imagery can be used in many applicative domains; although it was originally developed for mining and geology [1], it has now spread into fields such as ecology [2], civil or military surveillance [3], agriculture [4], medicine [5], food safety and quality [6].

From a general viewpoint, the processing and analysis chain for mono and multicomponent images consists of the following steps:

- *Contrast enhancement*, to improve the dynamic of the image,
- *Filtering*, to remove the noise contained in the image that comes from disturbances during acquisition and digitization of the image,
- *Restoration* to remove the blur that can be generated by the source acquisition or the motion of the sensors,
- *Classification* to partition the image into a set of regions in order to analyze and interpret its content.

The general framework of this thesis concerns the classification process.

Problem position

If we refer to the literature of image partitioning, we can state that this problem is difficult and is far from being resolved.

Generally a single technique is not sufficient to grasp all the different contents of the image, many studies done in our laboratory show that the use of only one method in different domains does not give relevant results [7]–[12]. The major problem of classification methods is their inability to adapt to the local contents of the image. Moreover, with the recent advent of advanced image acquisition systems, the scene is better described and the size of images is getting larger and larger (multispectral and hyperspectral imagery). The analysis and interpretation of this type of images is therefore getting more and more tedious and complex.

The problem of image partitioning is an ill-posed problem [13], and no generic method can give the best partitioning result. Partitioning errors are inevitable (over or under-partitioning); over-partitioning generates regions that do not belong to any object in the scene, and under-partitioning does not distinguish all the objects in the scene.

To overcome this problem and find a solution, researchers have proposed to use a cooperative paradigm, which consist of combining several methods to partition an image. Cooperation uses redundancy and complementarity of the information content in the image. This paradigm allows better understanding of the information contained in the image. For this reason, we decided to develop a cooperative and adaptive system for hyperspectral image partitioning. The cooperation is realized by several methods that can be adapted to uniform and textured regions in the image.

Objective

The objective of this thesis is to develop a robust ***cooperative*** and ***adaptive*** system to classify the pixels of hyperspectral images. 'Cooperative system' means using more than one method and 'adaptive' means automatically extracting the features of pixels according to the nature of the regions.

This thesis is divided into two parts:

- In the first part the state of the art of image partitioning and evaluation criteria is presented. The partitioning methods are either non-cooperative, such as genetic algorithms, Fuzzy C-means (FCM), Linde-Buzo-Gray algorithm (LBG), Artificial Neural Network (ANN), *k*-means, and Affinity Propagation (AP), or cooperative approaches that combine the above non-cooperative methods. Some experiments are done to analyze the non-cooperative methods. In this part, we also present a review of unsupervised evaluation criteria.
- In the second part of this thesis, the different modules of the developed partitioning system are presented in details. Each module is evaluated and validated separately on several synthetic and real mono and multicomponent images. Our developed system is also compared with other non-cooperative and cooperative methods. Finally, the system is also tested on two real applications related to the management of the environment problem.

PART I: Literature Review

Introduction

The objective of this review is to help us to select the methods which will be incorporated in the partitioning system we wish to design. The literature in the general field of statistical data analysis and image vision is vast and represents very active research areas. Among the areas of investigation in statistical data analysis, classification is one of the most prominent.

Herein, the main advantages and drawbacks of the methodologies of classification and evaluation criteria found in the literature are analyzed. In the present thesis, data classification and its application to multivariate images will be the central objective. More precisely, we will concentrate on unsupervised techniques, for the main and important reason that this paradigm remains free of any subjectivity that can be brought by the user.

In this review, we will first focus on the different families of classification approaches, starting from the set of methods used in a standalone manner (which will be referred to as non-cooperative methods), and generalizing to cooperative approaches, i.e. approaches which can combine several individual methods. The methods described will be discussed regarding their advantages and drawbacks, and whenever possible, their performances compared to other methods.

Classification assessment is often performed using external information such as ground truths. However, this information is not always available to the user who still wants to evaluate a given classification result. Internal assessment criteria can be helpful for this purpose, but also to improve the unsupervised classification method itself. This is why we also present a short review of evaluation criteria, mainly focusing on unsupervised criteria, since the absence of any supervision or *a priori* information is a strong requirement throughout the present work.

This part is organized in two chapters. The first one is dedicated to the description of the classification approaches. This chapter first details semi-supervised and unsupervised methods and secondly, cooperative classification approaches. Besides, the second chapter includes a review of unsupervised evaluation criteria to assess classification results.

Chapter 1 Classification approaches

1.1 Introduction

Image partitioning is one of the most important operations in image analysis chain. Its goal is to simplify and/or change the representation of an image into another mode of representation that is more meaningful and easier to analyze [14]. Image partitioning is typically used to locate objects in images. More precisely, image partitioning is the process of assigning a label to every pixel in an image such that pixels with the same label share some characteristics. In the framework of this thesis, we are interested in developing an automatic cooperative and adaptive partitioning system for hyperspectral images, where no *a priori* information or knowledge is required.

The growth and the availability of hyperspectral images, which contain rich information, have opened new possibilities of applications in many domains. In order to interpret this richness of information, a large diversity of image classification approaches can be found in the literature. In [12], these approaches are classified into two groups: non-cooperative and cooperative approaches. Non-cooperative approaches use only one classification method and cooperative approaches use two or more methods.

The purpose of this review is to analyze several classification methods, showing their advantages and disadvantages. This review will help us to find the methods that are the most suitable for our proposed cooperative/adaptive paradigm. In addition, we also present a review of some cooperative approaches and reveal their advantages and disadvantages.

The remaining of the present chapter is organized as follows: the second section will review the non-cooperative classification methods (semi-supervised and unsupervised) in order to select the most relevant ones to integrate them in the system

to be developed. In the third section we analyze the cooperative classification approaches of the literature. The last section concludes this chapter.

1.2 Non-cooperative classification methods

As we have mentioned above image partitioning is an ill-posed problem [13], and this has greatly stimulated researchers to develop new methods. These methods can be generally divided into three categories: supervised, semi-supervised and unsupervised. These categories can be either parametric or nonparametric.

Parametric methods require some hypothesis on the data distribution model (e.g. Gaussian model), which often does not correctly match the observed distribution of complex images such as hyperspectral images, while nonparametric methods can be used when no assumption can be made about the characterizing features. Supervised/parametric methods like Maximum Likelihood (ML) [15], Support Vector Machines (SVM) [16], Expectation Maximization (EM) [17], are the most commonly used. Supervised methods need *a priori* knowledge (e.g. training samples) to accomplish the classification task. However this information is not available in all application cases. Because of the two above reasons, supervised and parametric methods cannot be integrated into an unsupervised partitioning system.

Besides the semi-supervised methods require minimal input from the operator (e.g. number of classes, threshold, number of iterations) like *k*-means [18], Linde-Buzo-Gray (LBG) [19], Self Organizing Map (SOM) and Fuzzy C-Means (FCM) [20]. These methods all require the number of classes to be known in advance.

Unsupervised classification is a kind of classification that does not need training samples or any other *a priori* knowledge. One of the most recent unsupervised methods is Affinity Propagation (AP) [21] which does not need any information about the dataset. Unfortunately, AP is found to highly overestimate the number of classes, and it is practically inapplicable to large image datasets because of its computational complexity, which is quadratic in the number of objects.

Taking into account the brief general analysis on the classification method types, in the following we merely give more details about semi-supervised and unsupervised methods.

1.2 .1 Semi-supervised methods

In this subsection the main semi-supervised classification algorithms are presented and analyzed.

- ***k*-means**

k-means [18] is one of the basic semi-supervised algorithms that classify each object according to their similarity/dissimilarity requiring the number of classes (*NC*) fixed *a priori* by the user. This algorithm aims at minimizing an objective function (sum of squared error):

$$J = \sum_{j=1}^{NC} \sum_{i=1}^{N} \left(F_i - g(C_j) \right)^2 \qquad (1.1)$$

where *NC* is the number of classes, *N* is the number of data points in the dataset, F_i is the vector of *Nf* features representing the pixel x_i, $g(C_j)$ is the center of gravity of class C_j.

In summary the algorithm is executed as follows [22]:

Step 1: Define the centroid of the classes randomly.

Step 2: Assign each object to the class that has the closest centroid.

Step 3: When all objects have been assigned, recalculate the positions of the *NC* centroids.

Step 4: Repeat Steps 2 and 3 until the centroids remain unchanged.

Although it is proven that this algorithm will always converge, the *k*-means algorithm does not necessarily find the most optimal configuration. This algorithm is significantly sensitive to the initial randomly selected cluster centroid [23]–[25] which makes it unstable, hence giving varying results on the same dataset from a run

to another. For these reasons this algorithm must be further optimized as shown below in this section.

- **Self-Organizing Map (SOM)**

One of the most famous semi-unsupervised neural networks is the self-organizing map (SOM), which was originally developed by Kohonen [26].

The SOM neural network consists of two layers, as shown in Figure 1.1. For every neuron in the input layer, there is a link to every neuron in the output layer. During the training process of SOM network, for each input vector one best matching neuron in the output layer is got. A competitive learning algorithm is used to adjust weight vectors in the neighborhood of best matching neuron. The adjustment decreases as the time and the range of neighborhood increases.

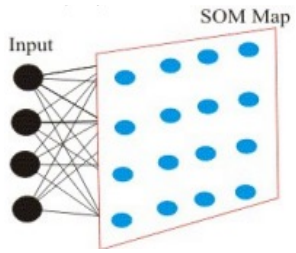


Figure 1.1: Self organizing map (SOM) structure

The SOM algorithm is described as follows:

Step 1: Randomly initialize the weights $Wt_i(0), i = 1, 2, ..., NC$, NC here is the number of the neurons in the output layer (number of classes). Set the maximum number of iterations as K.

Step 2: For iteration step k=1, 2... K: Get an input vector from the dataset X randomly or in predefined order.

Step 3: Find the best matching unit (BMU) i^* at iteration k, using the minimum distance:

$$i^*\left[X(n)\right] = \min_j d\left(X(n), wt_j(k)\right),\ j = 1, 2, ..., NC;\ 1 \le n \le N \qquad (1.2)$$

Step 4: Adjust the weight vectors of all neurons using:

$$wt_i(k+1) = wt_i(k) + \mu(k)\gamma(k, j, j^*\left[X(n)\right])\ (X(n) - wt_i(k)). \qquad (1.3)$$

where $\mu(k)$ is a learning rate parameter, $\gamma(j, j^*\left[X(n)\right])$ is a neighborhood function centered around the winning neuron. The size of the neighborhood is determined by a parameter $\sigma(k)$.

The parameters $\mu(k)$, $\gamma(j, j^*\left[X(n)\right])$ and $\sigma(k)$ are calculated as follows:

$$\mu(k) = \frac{1}{k}$$
$$\gamma(j, j^*\left[X(n)\right]) = \exp-(\frac{d(j^*, j)^2}{\sigma(k)}) \qquad (1.4)$$
$$\sigma(k) = \left(\frac{\sigma_\circ}{k}\right)^2$$

Step 5: Go to step 2 until no more changes in the weight space are observed or until the maximum iteration is achieved.

The performance of the SOM Neural Network depends on a lot of adjusting parameters:

- Number of output neurons: the ideal numbers of output neurons must be equal to the number of ground truth classes, associating exactly one neuron with one class.
- Weight initialization: the weights wt_i initially associated with each neuron contain random values.
- Choice of the neighborhood function (e.g. Gaussian).

This algorithm is often used to reduce the dimensionality of the datasets [27]–[29] rather than being used as a direct classifier.

- **Fuzzy C-Means (FCM)**

The Fuzzy C-means (FCM) is derived from the *k*-means algorithm by adding a fuzzification operation to solve ambiguous clustering problems [30]. FCM was originally developed by Dunn [31] and generalized by Bezdek [20]. This iterative algorithm assigns a class membership to a data point, depending on the similarity of the data point to a particular class relative to all other classes.

FCM seeks to minimize the following objective function:

$$J = \sum_{j=1}^{NC}\sum_{i=1}^{N} u_{ij}^{m}\left(F_i - g(C_j)\right)^2 \tag{1.5}$$

with the following constraint:

$$\sum_{j=1}^{NC} u_{ij}^{m} = 1 \quad \forall i \tag{1.6}$$

where NC is the number of classes, N is the number of pixels in the image, F_i is the vector of Nf features representing the pixel x_i, $g(C_j)$ is the center of gravity of class C_j, $m \in [1, \infty[$ is the fuzzification factor, and u_{ij} represents the entry (i,j) of the partition matrix, with $0 \le u_{ij} \le 1$.

The objective function is minimized when data points close to the centroid of their clusters are assigned high membership values, and low membership values are assigned to data points far from the centroid. The class centers and membership functions are updated by the following expressions:

$$g(c_j) = \sum_{i=1}^{N} \frac{u_{ij}^{m}}{\sum_{k=1}^{N} u_{ik}^{m}} F_i \tag{1.7}$$

$$u_{ij} = \frac{\left\| F_i - g(c_j) \right\|^{\frac{1}{m-1}}}{\left\| \sum_{j=1}^{NC} F_i - g(c_j) \right\|^{\frac{1}{m-1}}} \tag{1.8}$$

The four steps of FCM are:

Step 1: Initialize the membership matrix $U = \left[u_{ij} \right], \ 1 \le j \le NC, \ 1 \le i \le N$ with random values ranging between 0 and 1 satisfying the constraint in Equation (1.6).

Step 2: Calculate cluster centers $g(C_j)$ using Equation (1.7).

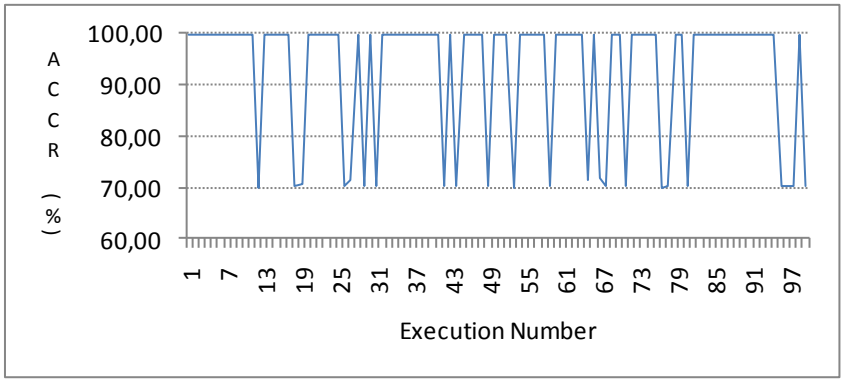
Step 3: Update the membership degree u_{ij} using Equation (1.8).

Step 4: Repeat steps 2 and 3 until the algorithm converges. This means that the difference between the current membership matrix and the previous membership matrix is below a specified tolerance value or the number of iteration reaches the maximum value specified by the user.

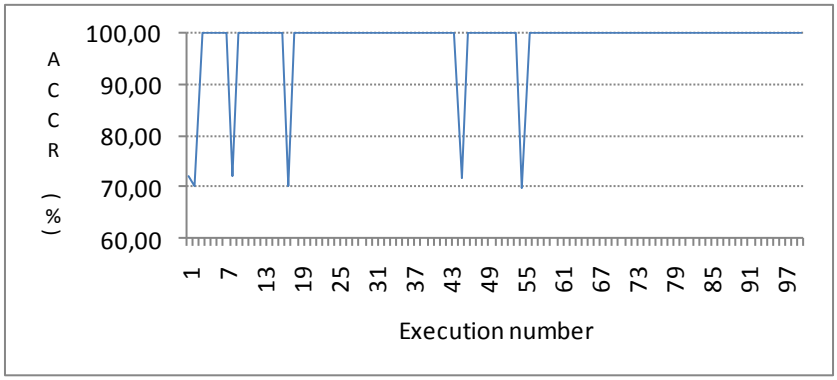
This algorithm is one of the most widely and successfully used methods and frequently applied in many domains such as: agricultural engineering, astronomy, chemistry, geology, medical diagnosis and pattern recognition [32], [33]. However, the choice of the fuzzification parameter *m* is very difficult because it influences the effectiveness of FCM and should be changed for each application type. Pal and Bezdek [34] suggested taking $m \in [1.5, 2.5]$, while in [35] the authors suggest implementing FCM with $m \in [1.5, 4]$.

To explore the impact of the fuzzification parameter on the performance of the FCM[1] we have tested it on synthetic images by changing the value of *m* (2, 4, 6 and 8) and fixed the number of classes (*NC*) to 5. The synthetic images includes images composed of five textured classes (textures are taken from the Brodatz album [36]). We also tested the impact of this parameter on the stability of this algorithm by executing it 100 times with a fixed value of *m*. Based on these experimental results (see Figure 1.2), the choice of the fuzzification parameter *m* affects the accuracy and the stability of the algorithms significantly.

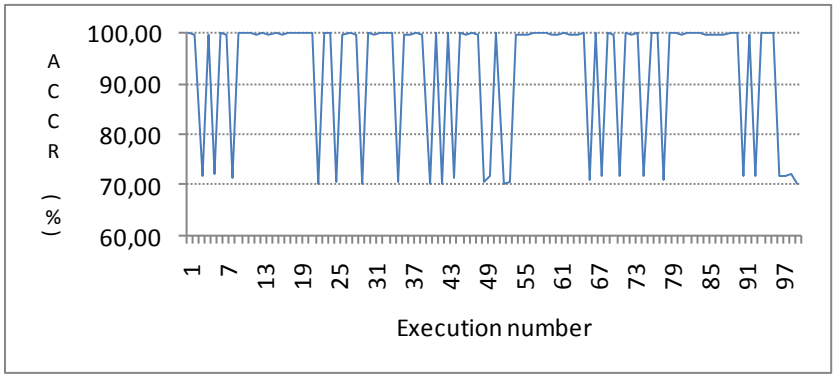
[1] The FCM algorithm used is the one provided by Matlab™ in release 7.11.0.



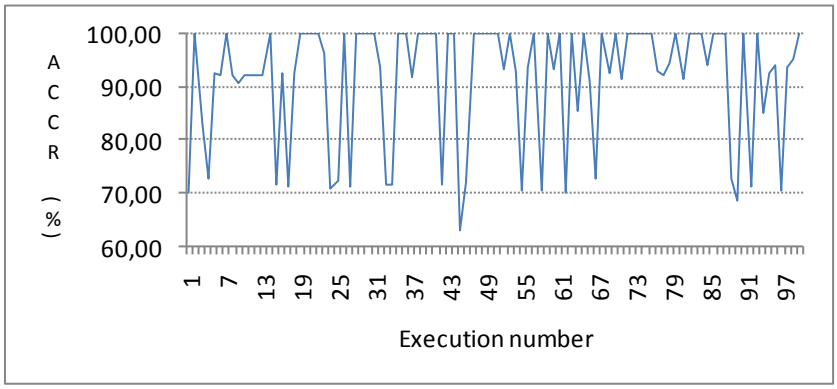
m=2



m=4



m=6



m=8

Figure 1.2: Effect of the fuzzification parameter (*m*) on the stability and the performance of FCM algorithm (*NC*= 5)

From the results shown in Figure 1.2, we observe that there is a high fluctuation of classification result accuracy, in the cases when *m*=2, 6 and 8. Contrarily when *m*=4 the results accuracy is more stable.

- **Modified versions FCM**

Many efforts have been done to enhance FCM algorithm especially in the field of image partitioning. In [37] an algorithm called FCM_S is proposed, in which the authors have changed the standard objective function of FCM by adding another term similar to the standard one as follows:

$$J = \sum_{j=1}^{NC}\sum_{i=1}^{N} u_{ij}^{m}\left(g_l(x_i) - g(C_j)\right)^2 + \frac{\alpha}{Card(W_i)}\sum_{j=1}^{NC}\sum_{i=1}^{N} u_{ij}^{m}\left(\sum_{x_r \in W_i}\left(g_l(x_r) - g(C_j)\right)^2\right) \qquad (1.9)$$

where $g_l(x_i)$ is the gray level of the pixel x_i, W_i stands for the set of pixel neighbors in the window around x_i, $Card(W_i)$ is the number of pixels in W_i and α is a regularizing parameter introduced to control the effect of the added term in the objective function. The added term takes into account the neighborhood of the pixel to be classified. The neighborhood effect biases the solution toward piecewise-homogeneous labeling. The value of α must be adjusted in regard to the type and level of the noise. Generally, the type of noise present in the image is unknown, so α is set empirically. The number of the neighbor pixels chosen also influences the result. Actually, the added term in the objective function (1.9) highly increases the calculation time. In order to reduce the computation time FCM_F algorithm [38] was developed, in which the neighborhood term added to the basic objective function in FCM_S is changed by only integrating the current pixel of a previously filtered version of the original image. The objective function of FCM_F algorithm is defined by:

$$J = \sum_{j=1}^{NC}\sum_{i=1}^{N} u_{ij}^{m}\left(g_l(x_i) - g(C_j)\right)^2 + \alpha\sum_{j=1}^{NC}\sum_{i=1}^{N} u_{ij}^{m}\left(\hat{g}_l(x_i) - g(C_j)\right)^2 \qquad (1.10)$$

where $\hat{g}_l(x_i)$ is the mean or median of neighboring pixels lying within a window around x_i. This algorithm processes the original image and the filtered image simultaneously. The Average Correct Classification Rate (ACCR) is practically the same as with FCM_S, but with much less computation time. FCM_F has the same disadvantages as FCM_S; in addition it requires a filtering operation in advance. The authors in [39] proposed EnFCM, also to enhance FCM_S. In EnFCM the image is segmented after a local linear transformation that takes into account the gray level of the current pixel and the local mean of its neighborhood as follows:

$$\xi_i = \frac{1}{1+\alpha}\left(g_l(x_i) + \frac{\alpha}{Card(W_i)} \sum_{r \in W_i} g_l(x_r) \right) \tag{1.11}$$

The partitioning is done by means of the histogram of the locally transformed image ξ. The objective function is the same as the standard FCM, except that a variable is added which is the frequency of occurrence of the gray values in the histogram of the filtered image. In EnFCM the computational load is highly reduced relative to FCM_S. Besides, the quality of the image partitioned by EnFCM is comparable to that of FCM_S. In [40] the same authors present a modified FCM based method that targets accurate and fast partitioning in the presence of mixed noise. This method extracts a scalar feature value from the neighborhood of each pixel, using a context dependent filtering technique that deals with both spatial and gray level distances. The authors in [41] propose FGFCM that changes only the way that the pre-filtered image is calculated in EnFCM [39]. For the transformation of the image, it defines coefficients for each window centered on the concerned pixel. These coefficients are calculated by taking into account spatial and gray level information, and then these coefficients are used to weight the neighbor pixels in the local transformation of the image. The partitioning is also done by means of the transformed image histogram. The pre-filtered image is influenced by the weighting spatial and gray level factors, which in turn are influenced by two other scale factors.

The presented algorithms above (EnFCM, FCM_S, FGFCM, FCM_F) are based on the standard FCM [20]. They either add a term to the objective function, or filter the image beforehand. In order to compare the classification results of the modified

algorithms with FCM, we have chosen the FCM_S[2] algorithm since these modified algorithms all give comparable results. Figure 1.3 shows the results of this comparison in Average Correct Classification Rates (ACCR)[3] by using only the gray level as feature. In this experiment the number of classes (*NC*) is set to 5.

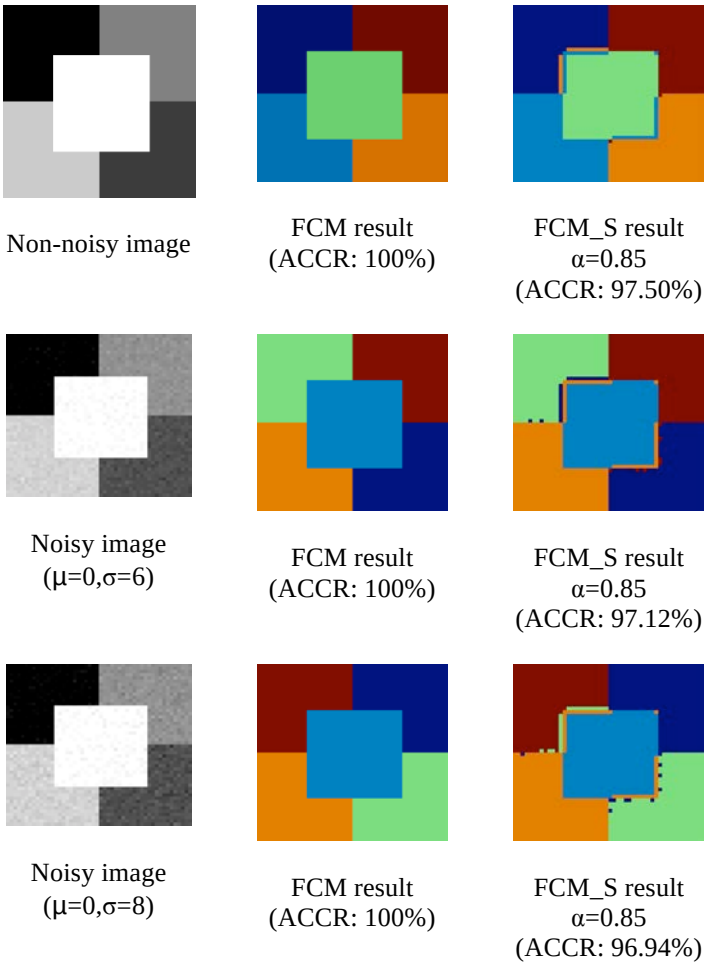


Figure 1.3: Classification results of FCM and FCM_S methods (*NC=5*)

In [42] another algorithm called FLICM is proposed which modifies the standard FCM objective function by adding a new fuzzy local neighborhood factor, which can automatically determine the spatial and gray level relationship. Contrarily to [37]–[41], FLICM algorithm is free of any empirically adjusted parameters.

These modified algorithms which were normally developed to partition noised images do not give the expected results. In the experiment shown in Figure 1.3, FCM

[2] FCM_S is programmed in Matlab™. The codes are available on the Matlab Central, File Exchange.
[3] Sum of correct classification rate of each class divided by number of classes.

gives better results whatever the noise level. In addition the problem of the choice of the fuzzification parameter is not resolved.

Another modified version of FCM is presented in [43]. This improved algorithm adds geometrical information during the classification process. The local neighborhood of each pixel determines the condition of each pixel, which guides the clustering process. In more details, the FCM objective function is modified to include two parameters: the membership degree of the labeled pixels and a Boolean variable to distinguish between labeled and unlabeled pixels. This algorithm outperforms the results of FCM but it requires *a priori* knowledge about the image to be partitioned. Another unsupervised version of FCM is presented below in the unsupervised methods subsection (see Section 1.2.2).

- **Linde-Buzo-Gray (LBG) Algorithm**

The LBG algorithm [20] is also a very well known algorithm especially in the domain of vector quantization. Since it is simple and easy to implement, it has been widely used in many other applications, such as pattern recognition, image segmentation, speech recognition and face detection [44], [45]. This algorithm is based on the idea of *k*-means clustering. The only difference between them is that the class centers in LBG are introduced incrementally. The algorithm of LBG can be briefly described by the following steps:

Step 1: Set number of clusters *K*. Set the dispersion $D_{t-1}=\infty$. Set the iteration counter *t=0, set k=1,* set the initial class center g_0 as the mean of all data points, set ε to a small positive value.

Step 2: Place new class centers near each class center in g_0 and then modify $k \rightarrow 2k$.

Step 3: Affect each data point in the dataset to the nearest class center including the new ones.

Step 4: *t=t+1*, calculate the overall distortion: $D_t = \frac{1}{N}\sum_{i=1}^{k}\sum_{j=1}^{NC_i} d(F_j^{(i)}, g(C_i))$

where *N* is the number of data points in the dataset, and NC_i is the number of objects in class *i*, $i=1\ldots k$. F_i is the feature set representing the pixel x_i

Step 5: calculate the centroids by $g(C_i) = \frac{1}{NC_i} \sum_{j=1}^{N_i} F_j^{(i)}$ $i=1\ldots k$.

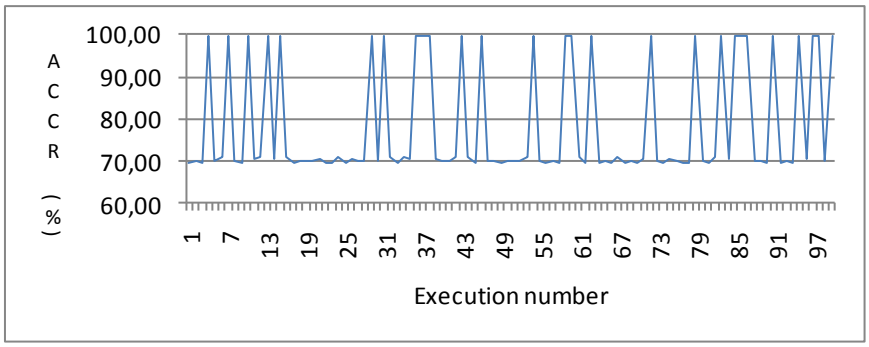
Step 6: if $(D_{t-1} - D_t)/D_t > \varepsilon$, go to step 3; otherwise go to Step 7,

Step 7: if $k=K$ stop, otherwise $D_{t-1}=D_t$, $g_0=g$, and then go to Step 2.

The drawback of this algorithm as well as the *k*-means is that the quality of the solution highly depends on the initial location of the class centers [46]. The result of the experiments that we have conducted confirms this drawback (see Figure 1.4(a)). Obviously, if the initial values are near an acceptable solution, a higher probability exists that the algorithm will find a better solution. If not, finding the optimal solution is not guaranteed and for these reasons many efforts have been done to overcome this problem of the algorithm. In [47] an algorithm called LBG with utility (LBG-U) is proposed which consists mainly of repeated runs of the standard LBG algorithm. Each time LBG converges, however, a utility measure is assigned to each class center. Thereafter, the center of the class with minimum utility is moved to a new location, LBG is run on the resulting modified class center until convergence, another center is moved, and so forth. This algorithm is more time consuming than LBG, and still affected by the initial choice of class centers. Another method is proposed in [48] called Enhanced LBG (ELBG). The basic idea of ELBG is the introduction of a utility measure as in LBG-U [47], but the technique of moving the class centers differs. Here the class center displacement is not validated unless it decreases the dispersion of the solution found. ELBG is less sensitive than LBG and LBG-U to the initial choice of class centers, and it is less time consuming than LBG-U. In [49], a method called Adaptive Incremental LBG (AILBG) is presented; in this method the class centers are inserted incrementally. New class centers are inserted in regions of the input vector space where the distortion error is highest until the desired number of centers is achieved. During the incremental process, a removal-insertion technique is used to fine-tune the class centers in order to make it independent of the initial conditions. Acutely we have confirmed the stability of AILBG by the experiments conducted on some synthetic images (see Figure 1.4(b)). The overhead time of AILBG to standard LBG is negligible. In [49] the authors also present a comparative study between the different versions of LBG algorithm, and found that AILBG outperforms all of them.

However AILBG like all other extensions is a semi-supervised method and requires the number of classes to be fixed in advance. In [50] an unsupervised version of LBG called modified LBG (MLBG) is proposed but its computing time is very high.

Figure 1.5 shows a comparison of classification results between LBG and AILBG[4] by fixing class number to 5. In the case of noisy images, LBG is trapped in a local minimum and gives an incorrect result, while AILBG is able to avoid this local minimum and finds a better result.



(a) LBG

(b) AILBG

Figure 1.4: Stability of LBG and AILBG algorithms (*NC* = 5)

[4] The algorithms of LBG and AILBG used are programmed in Matlab™ by our laboratory.

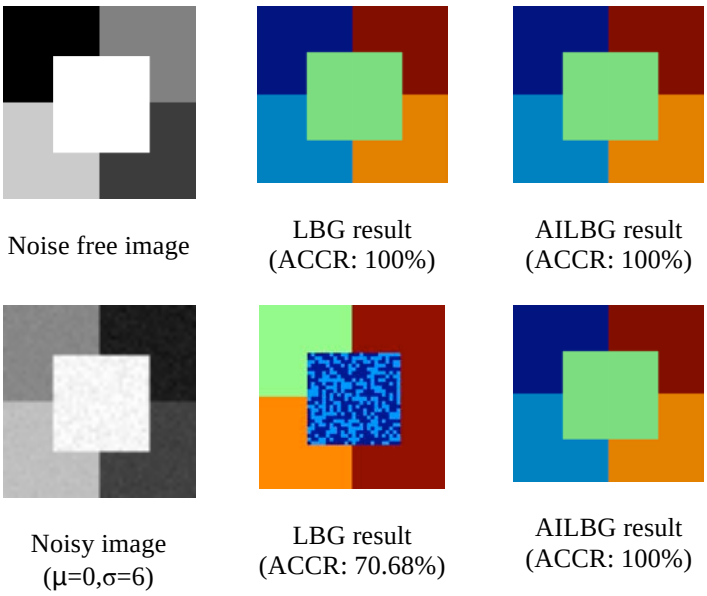


Figure 1.5: Classification results of LBG and AILBG methods (*NC* = 5)

- **Complementarities of FCM and AILBG**

FCM and AILBG are the most successful algorithms of semi-supervised classification [51], [52], and they have many common advantages: guaranteed convergence, fast execution time, easy implementation, and compatibility with different distance types. Besides, they differ from each other in many points; firstly the decision concept of the methods is completely different; FCM is based on fuzzy decision, while AILBG is based on hard decision. Secondly, in FCM all the class centers are initialized randomly once at the start of the algorithm, while in AILBG the class centers are introduced along the iterations. Lastly the objective function of FCM (see Equation 1.5) is composed of the mean squared error weighted by the membership value of each individual in the dataset, while the objective function of AILBG algorithm is the standard mean square error.

The above-mentioned differences between these two methods make them produce very different results for the same dataset. In the following we show some specific important cases where FCM, LBG and AILBG algorithms give different results:

- FCM is found to give better results than LBG and AILBG in the case of non-convex shaped clusters. Figure 1.6 shows an example of compared

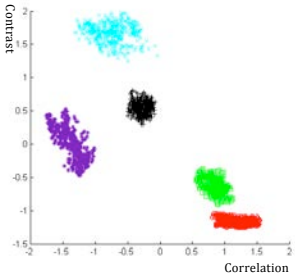
classification results between FCM, LBG and AILBG algorithms on an image by using contrast and correlation features. In this experiment, the number of classes is fixed to 5.

- FCM creates fuzzy intermediate classifications, rather than hard ones; this is useful when the boundaries between the clusters are ambiguous and not well separated [53]. In other words, thanks to the fuzzy membership, FCM is able to find uncertain boundaries that AILBG and all other hard decision clustering algorithms fail to obtain. Figure 1.7 shows an example of this case where contrast and sum average features are used. The comparison of classification results of FCM, LBG and AILBG algorithms is also given. The number of classes is fixed to 5.
- LBG and AILBG work much better than FCM in the case when the dataset contain small clusters (clusters with few objects); in such case FCM tends to locate centroids in the neighborhood of the larger clusters and misses the small clusters [54]. In Figure 1.8 we show the classification results of FCM, LBG, and AILBG on an example of six classes generated according to Gaussian models where two classes among them are low-populated. Therefore, in this experiment, the number of classes is fixed to 6.

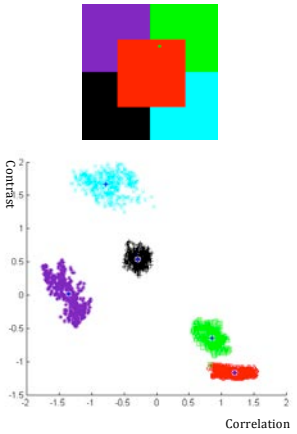
We precise in these experiments, the FCM fuzzification factor is set to 4.

In Figure 1.6, Figure 1.7 and Figure 1.8 we give two different results of LBG, because the result varies from a run to another on the same dataset, but for the FCM and AILBG the results remain unchanged in most cases. The tested cases show the difference between the results of FCM, AILBG and LBG. They also show that AILBG is much more stable than LBG and always gives much better results, with no additional parameters and a very negligible overtime. The cases mentioned in the previous paragraphs make FCM and AILBG good candidates to be put together in a cooperative classification approach.

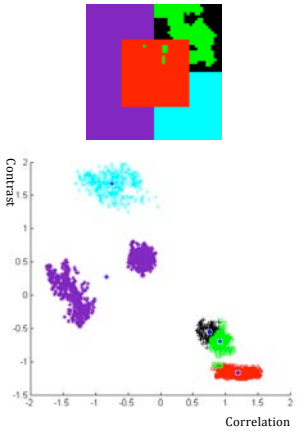
(a) Synthetic image



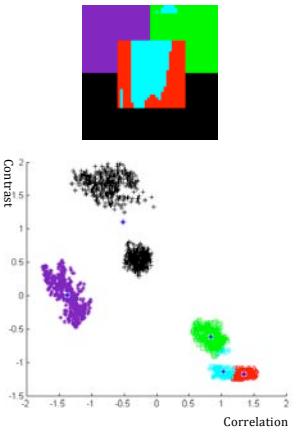
(b) Dataset with non-convex classes (contrast, correlation features)



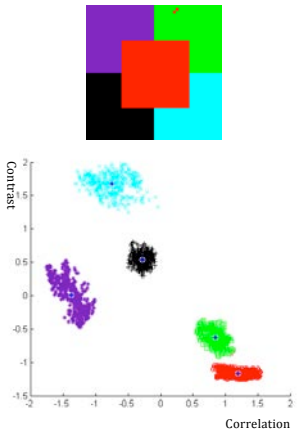
(c) FCM result
ACCR: **99.97%**



(d) LBG result trial 1
ACCR: 70.77%



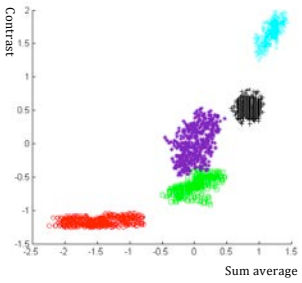
(e) LBG result trial 2
ACCR: 69.45%



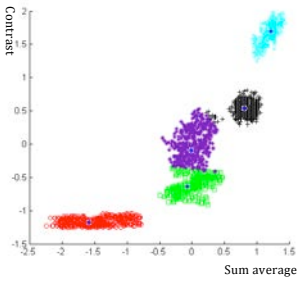
(f) AILBG result
ACCR: 99.91%

Figure 1.6: Classification results using FCM, LBG and AILBG methods on a dataset containing non-convex classes (*NC* = 5).

(a) Synthetic image

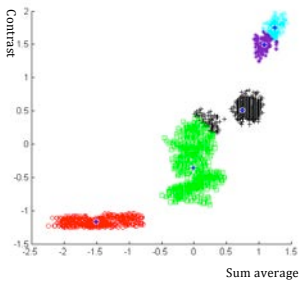


(b) Dataset with overlapping classes (contrast, sum average features)



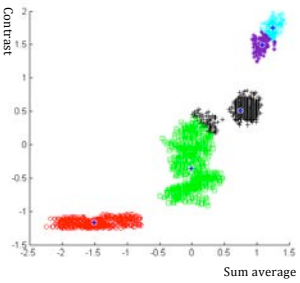
(c) FCM result,

ACCR: **98.08%**



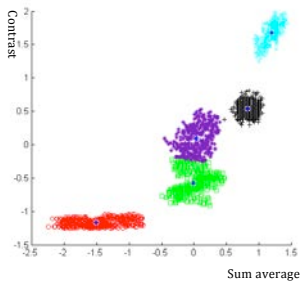
(d) LBG result trial 1,

ACCR: 74.43%



(e) LBG result trial 2

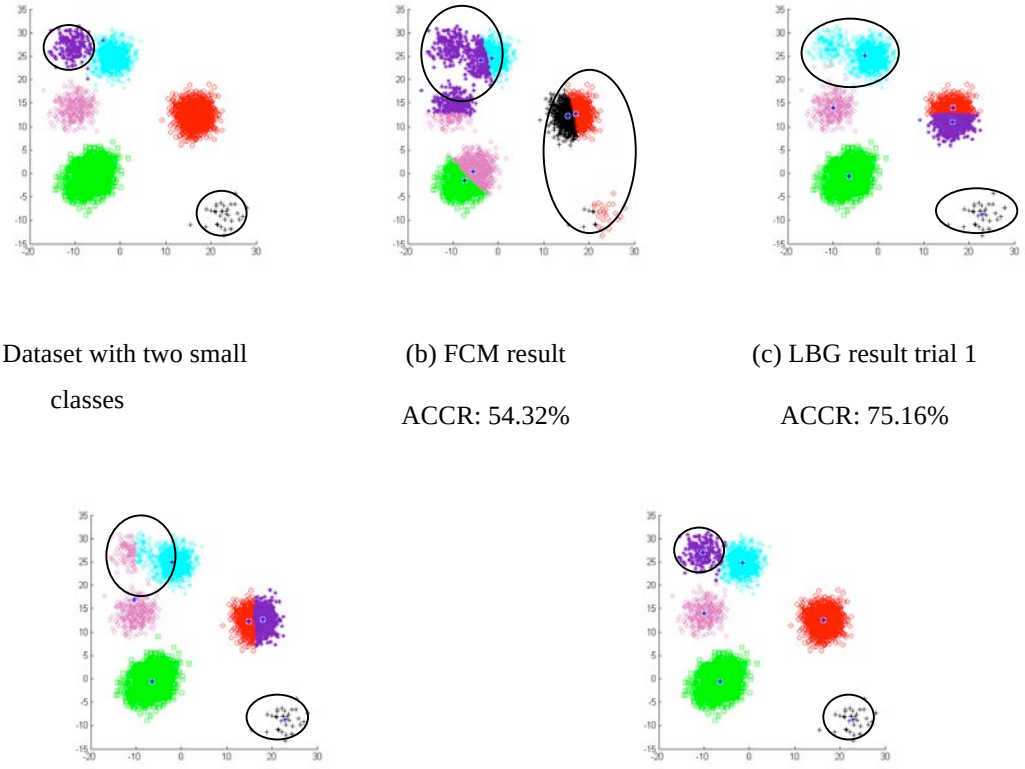
ACCR: 74.38%



(f) AILBG result

ACCR: 95.80%

Figure 1.7: Classification results using FCM, LBG and AILBG methods on a dataset containing overlapping classes (*NC* = 5).



(a) Dataset with two small classes

(b) FCM result

ACCR: 54.32%

(c) LBG result trial 1

ACCR: 75.16%

d) LBG result trial 2

ACCR: 75.05%

(e) AILBG result

ACCR: 99.50%

Figure 1.8: Classification results using FCM, LBG and AILBG algorithms on a dataset containing small classes (*NC* = 6).

1.2 .2 Unsupervised methods

In this subsection we present the main unsupervised and nonparametric methods. We recall that it is meant by unsupervised methods those which do not require any *a priori* knowledge (e.g. number of classes, training samples and threshold values).

- **Genetic algorithms**

A genetic algorithm is a search heuristic that mimics the process of natural evolution. It is a searching procedure based on the laws of natural selection and genetics. A genetic algorithm is composed of five parts [55]:

– Genotype (chromosome): it is the genetic makeup of an individual.

- Initial population: a group of individuals characterized by their genotypes.
- Objective (fitness) function: this function measures the adequacy of an individual to its environment by considering its genotype.
- Genetic operations: these operations are performed on genotypes in order to evolve the population during the generations. There are three types of genetic operations:
    - Mutation: the genes of an individual are modified in order to better adapt to the environment.
    - Selection: the individuals that best fit to the environment have bigger chance to be selected for reproduction.
    - Crossover: two or more individuals reproduce by combining their genotypes.
- Stopping criterion: this criterion allows stopping the evolution of the population.

The execution of the genetic algorithm is performed in five steps:

Step 1: Initial population definition and calculating the fitness of each individual.
Step 2: Selection and mutation of the current population.
Step 3: Apply crossover operation.
Step 4: Evaluate the individuals in the population.
Step 5: Go to the second step if the stopping criterion is not satisfied.

This algorithm has two main advantages: *i)* ability of solving problems with multiple solutions, *ii)* solving multi-dimensional, non-differential, non-continuous, and even non-parametrical problems. For these reasons this unsupervised algorithm is used for many applications, and widely used for data fusion.

However it presents some difficulties: *i)* the choice of the fitness function conditions the results; *ii)* it requires a large number of chromosomes to avoid premature convergence to local minima solution. In this case its computation time is a burden.

- **Hierarchical Genetic Algorithm:**

An HGA is presented in [56] that overcomes the difficulties encountered when using the conventional Genetic Algorithm (GA). The HGA simultaneously estimates the proper number of classes and then partitions the image into several homogeneous classes. The main difference between conventional GA and hierarchical GA is that the genetic structure of a chromosome is formed by a number of gene variations that are arranged in a hierarchical manner (see Figure 1.9). In HGA the chromosome consists of two types of genes, the control genes and the parametric genes. The purpose of control genes is to determine which parametric gene should be utilized and which one can be disabled during the evolution process.

The results of this method are compared to the ones of Dynamic Thresholding, and Contextual-Constraint based Hopfield neural cube. The accuracies of the obtained results are 100%, 85% and 90% respectively. The test is realized on a simple monocomponent image

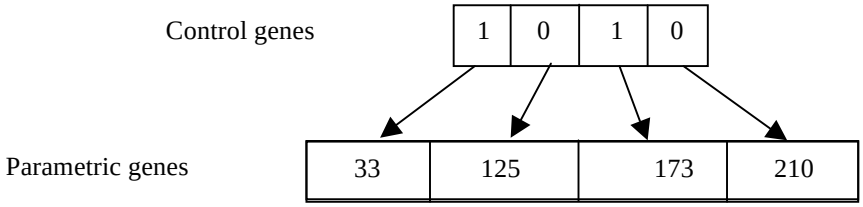


Figure 1.9: An example of chromosome representation [57]

- **Hybrid Genetic Algorithm**

In [57] a hybrid genetic algorithm incorporating the traditional genetic algorithm and *k*-means clustering method within a multiresolution framework is presented. It is another variant of GA where the crossover operator is replaced by *k*-means clustering method while the other operators are adopted. This replacement is done because the select-operator of genetic algorithm picks solutions according to the fitness values of

the chromosome (*global* information), instead of the local interaction among the genes.

In this method, first, a quad-tree structure is constructed and the input image is partitioned into blocks at different resolution levels. Texture features are then extracted from each block (mean gray value is the only employed as texture feature).
The whole image is then mated as a chromosome/solution and each block is seen as a gene. That is to say, instead of encoding a chromosome as a string of symbols, a chromosome is a two dimensional array of genes. Therefore, each gene will have four immediate neighbors except the ones along the image/chromosome borders.
The hybrid genetic algorithm is then performed to partition the image by assigning an optimal allele (texture class label) to each gene. Once the algorithm converges at a specific level, the allele of each gene (partitioning result) of the current level is propagated down to the next level as the initial alleles of its child genes in the lower level. The algorithm in each resolution level stops when a stop criterion is met.

The different steps of this algorithm for each resolution level can be summarized as:

Step 1: Extract texture features from each block
Step 2: Partition current level as follows:

- Initialize population
- Do
  - Perform *k*-means clustering.
  - Mutate.
  - Evaluate fitness.
  - Select chromosomes according to their fitness for next generation.

  While there is evolution in the population.

Step 3: Propagate partitioning result to the next level; if last level stop.
Step 4: Go to step 1.

The application of this algorithm on monocomponent image shows that the rate of misclassification is decreasing as the level of resolution increases. This method has the same drawback than the *k*-means algorithm.

- **Multi-objective variable length string GA:**

An important approach for unsupervised land-cover classification in remote sensing images is the clustering of pixels in the spectral domain into several fuzzy partitions. In [58] a multi-objective optimization algorithm is used to tackle the problem of fuzzy partitioning where a number of fuzzy cluster validity indexes are simultaneously optimized. This method uses a simultaneous optimization of two cluster validity measures: an index indicating the goodness of the obtained clustering and the fuzzy C-means (FCM) measure which calculates the global cluster variance. This method is compared with two other classification methods FCM and GA with one objective. The multi-objective GA shown higher classification efficiency compared to GA with one objective function and FCM.

The performance of this multi-objective clustering method depends highly on the choice of objectives, which should be as contradictory as possible. It also suffers from slow convergence.

- **Unsupervised modified FCM**

In [59] an unsupervised version of FCM which is adapted for high dimensional multiclass pattern recognition problems is presented. Its main objectives are to increase the accuracy and stability of the well-known FCM. It is based on two concepts: the unsupervised weighted mean and cluster centroids from nonparametric weighted feature extraction, and discriminant analysis feature extraction. The advantage of this method is its unsupervised nonparametric skills where the system finds the number of classes; besides, it is more robust than the FCM algorithm. However, it is not completely stable since for some complex images, the level of variability of the results is high.

1.2.3 Discussion

Several non-cooperative methods are presented above which are either semi-supervised or unsupervised but are all non parametric. Each method has its own advantages and drawbacks. A summarized analysis of these non-cooperative methods is shown in Table (A.1) in Appendix A.

Despite the existence of a vast number of algorithms [60]–[64] yet no single non-cooperative method or algorithm is able to identify all kinds of cluster shapes and structures that are encountered in practice. Each algorithm has its own approach, and imposes a structure on the data [65]–[67][68]–[70].

Since there is no general solution to the image segmentation/classification problem, using multiple methods allows to better interpret the data [71]; in this case, each method extracts an information, which is not always localized by the other methods. In fact, applying different methods or the same method with a tiny modification of initial data on the same image is very unlikely to give identical results, even when the number of classes is given in advance. Each result obtained using different methods contain correct and incorrect information, some pixels being correctly classified, while others are not. This situation thus makes the quality assessment difficult for the choice of a particular method. Therefore, cooperation between methods is highly recommended. However to produce reliable results, the choice of the methods to be put in cooperation should be operated carefully in such a way that the disadvantage of a method should be covered by the advantages of the others.

The following section presents some developed cooperative methods found in the literature.

1.3 Cooperative classification approaches

Since getting reliable results is difficult to obtain using one single classification method, many partitioning cooperative approaches have been developed in the literature in order to combine the advantages of the non-cooperative methods. Some of them are adapted for non-textured images using edge-region combination [72]–[76], and others dedicated for textured and/or non-textured images or other non-image data [7], [71], [77]–[80].

Voisine [11] defined the cooperation of the methods as two types: informal and formal cooperation. These methods are considered as agents according to the rules and the criteria that combine them. The informal cooperation takes place at a high level by the rules or the common criteria of the classification methods. In informal cooperation the image treatment takes place independently to each other but towards the same objective. The objective imposes the behavioral rules of the classification methods. In the informal approach different partitioning results or different classification methods might be exploited. Besides, formal cooperation defines the interaction between the classification methods. In this case all the rules of cooperation are defined by the user in advance, and share a common objective.

The cooperation between methods can be done using three different schemes [12]:

- Sequential [7], [8], [81], [82],
- Parallel [71], [77], [78], [80], [83]–[85],
- Hybrid [79], [86]–[88].

In the following section we present each of these structures.

1.3.1 Sequential cooperation

The principle of sequential cooperation scheme is to combine two or more methods in a way that the classification result of a method is the starting point of another one. In the following the main cooperative approaches are presented.

- ***k*-means and genetic algorithm**

In [89] a partitioning technique for multiband image that uses *k*-means and genetic algorithm in sequential manner is presented. Considering the edge ambiguity in the image, a novel fuzzy-set-based edge-boundary-coincidence measure is combined with a region heterogeneity measure to guide the GA and tune the partitioning process. The image partitioning is done by the following steps:

Step 1: The *k*-means clustering method is applied to generate an initial finely partitioned image. The *k*-means clustering result is used as the seed chromosome to generate the initial population for the GA.

Step 2: GA is used to control the splitting and merging of classes so as to optimize an evaluation function.

This image clustering technique has the following characteristics:

- The finely partitioned image from the *k*-means clustering is used as the input to the GA. This approach greatly reduces the search space of the GA. In addition, feature information conjuncts with spatial information globally.
- Evaluation criteria are used that incorporate both edge information and region information.

The results show that *k*-means and GA method provides better results than the other conventional edge detection methods. This method is dedicated to edge detection.

- **Split and merge using Fuzzy C-Means and Orthogonal arrays**

A generic splitting/merging partitioning method has been proposed in [90] that combines FCM and orthogonal arrays. The FCM algorithm is used to split the image into many small regions depending on the measurement of the contrast and the compactness between classes. To merge two adjacent classes C_1 and C_2 the difference between the average gray levels is considered, the smaller the difference, the higher the possibility of their mergence. In addition, the intensity distance along boundary

pixels among adjacent classes (C_i, C_j) is used to define the evaluation function of class discrimination for adjacent classes. A validity check is done to ensure the existence of common edges between regions and to determine the emergence probability based on the rank of significance of their contribution using orthogonal array (ISOA).

Let *IN* be the initial number of edges, *EN* be the new number of edges after the mergence operation. *PN* be the number of edges before the mergence operation, and *DE* be the number of deleted edges in one mergence operation.

The ISOA algorithm is presented as follows:

Step 1: Initially, let $EN = IN$ and $PN = IN$.

Step 2: Evaluate the values of edge.

Step 3: Select an orthogonal array $Ln(2N)$ where $n=2log2\ EN+1$ and $N=n-1$. Use the first *EN* columns of the orthogonal array.

Step 4: Compute the main effect of every edge using the objective function in the orthogonal array and rank the edges using the main effect values of level 1. The edges with large evaluation value have a higher rank.

Step 5: (Mergence operation) remove the worst *DE* edges having the lowest ranks.

Step 6: End the algorithm if the predefined region numbers is satisfied or some stopping condition is met or $EN - DE < 1$.

Step 7: *Let* $PN = EN$ *and* $EN = EN - DE$, go to step 3.

The method is tested on different generic and noisy artificial images; it is also tested on noisy, and blurred natural images. From the tests it is found that the method is fast and robust. This method is also dedicated to edge detection.

- **FCM and Hybrid Dynamic Genetic Algorithm (HDGA)**

In [7] a sequential cooperative approach is proposed. It puts in cooperation FCM and Hybrid Dynamic Genetic Algorithm (HDGA); FCM gets the cluster centers from HDGA in order to classify the content of different types of images. This approach is tested on two multicomponent satellite images (Ikonos and Landsat) to detect

different land covers in the images i.e. urban, bare land and agriculture. The classification rates are 90%, and 97% respectively. The spectral signatures are used as features in this approach.

- **Radial Basis Function Neural Network (RBFNN) and GA**

Another sequential approach for satellite image partitioning is presented in [82] that combines Radial Basis Function Neural Network (RBFNN) and GA. During the learning process of the RBFNN, GA is employed to automatically determine the hidden layer parameters. The image used to assess this approach is an RGB QuickBird satellite image taken over rural areas that contain vegetation and bare soil. Many features are extracted from the images like: entropy, second momentum, and dissimilarity. The best classification rate for this approach is 88%. The disadvantage of this approach relies in the fact that it uses a parametric method which requires making some hypothesis on the distribution of the image pixels.

- **Self-Organizing Map and Genetic algorithm**

A sequential approach presented in [81] combines SOM and GA. This approach divides the original image into many small rectangular regions and extracts texture features from the data using two-dimensional autoregressive model, and other features such as fractal dimension, mean, and variance. Various experiments were performed on a set of monocomponent synthetic images with 3 or 4 different real textures. Texture features are extracted to describe the textures in the image. The authors found that the combination of both methods is visually more accurate as compared to using both methods separately. The obtained results are shown without any Average Correct classification rate (ACCR). The authors conclude that the methods are effective for partitioning images that contain similar texture fields. The advantage of this sequential approach is that it uses two unsupervised methods.

In conclusion, the main drawback of a sequential cooperation scheme is that the quality of the classification results is strongly influenced by the sequencing order of the methods.

1.3.2 Parallel cooperation

Parallel cooperation schemes consist of combining or fusing classification results of different methods. These results are either fused in parallel or sequentially. A fusion stage is therefore required at the end of the process. Whatever is the type of the fusion, it is important to associate an index of confidence to the results of the classification or segmentation to be fused [91].

The data fusion approaches developed in the literature involve statistics theory, neural network, fuzzy logic, expert system, majority voting, weighted majority voting, weighted-linear opinion pool, minimum spanning forest, evidence commutation, fusion with background knowledge , GA based fusion, and mutual information [10], [71], [80], [84], [92]–[96].

In the following we first present the main fusion approaches and then we describe complete parallel partitioning approaches.

1.3.2.1 Fusion approaches

In the following we review some approaches from the literature that detail only the fusion process often used in cooperative parallel methods.

- **Evidence accumulation fusion**

In [80] a fusion approach called evidence accumulation is presented. The main idea of this approach is to produce a co-association matrix from the different initial results. This matrix gives the information of the number of times that two data objects have been put together in the same cluster. A hierarchical clustering is then used. This hierarchical method utilizes the co-association matrix as a distance matrix, to cluster the objects into the final partition. This approach is tested on 9 different synthetic and real datasets, for example: iris, breast cancer, three rings, and a synthetic textured monocomponent image represented by 19 texture features. The features extracted from the monocomponent images are not specified. The best correct classification rate for the monocomponent image is 92%.

- **Fusion with background knowledge**

Another parallel approach proposed in [71], in which different unsupervised methods are used to cluster the same dataset, and then the different results are combined. The disadvantage of this approach is that it requires some background knowledge while fusing the results of the different methods. In this work the methods used in parallel are not specified, the authors are focusing on the fusion process in particular. This approach is tested on many real non-image datasets (iris, wine, ionosphere, and segment). The best correct classification rate obtained is 95% for the segment dataset.

- **Fusion by conflict resolution**

In [84] three other iterative and a GA based approaches for conflict resolution are proposed. The iterative approaches are called: the worst conflict choice (WCC), stochastic conflict choice (SCC), and roulette-wheel conflict choice (R-WCC).

The advantage of these approaches is that they do not take into account any change unless they improve the global result, but unfortunately they could give suboptimal results. These approaches are tested on synthetic and real non-image datasets (iris, wine, and segment). The results obtained are assessed using different evaluation indices (Rand, Jaccard & Mallow, and F-measure). According to all the evaluation indices, GA gives the best results for all the datasets.

- **Fusion maximizing the mutual information**

In [96] another fusion approach is proposed that is based on maximizing the mutual information among the results. This information is measured through the Average Mutual Information (AMNI). This approach is tested on two real and two synthetic (non-image) datasets. The synthetic datasets contain Gaussian distributed clusters, while the two real datasets contain information for text recognition and clustering. The disadvantage of this approach is that it is sensitive to class sizes and seeks only for balanced-size classes (i.e. all the class should have approximately the same number of data objects).

- **Fusion by *k*-means**

Topchy et al. in [97] described how to create a new feature space from multiple results by interpreting them as a new set of categorical features. The *k*-means algorithm is applied on this new standardized feature space using a category utility function to evaluate the quality of the consensus. This approach is also tested on two real and two synthetic non-image datasets only. The correct classification rate obtained is 97% for the real dataset, and 100% for the synthetic ones.

1.3.2.2 Complete parallel cooperative approaches

Many complete parallel approaches where developed in the literature using different strategies. In the following we describe the main ones.

- **SVM +ISODATA and SVM + EM**

In [77] Tarabalka et al. propose two parallel cooperative approaches for hyperspectral images, each of them putting in cooperation two classification methods: the first one combines SVM and ISODATA methods, while the second one combines SVM and EM methods. In this approach the SVM associated to EM or ISODATA classifies image pixels in parallel, and the results obtained by the EM or ISODATA are used to create a dynamically shaped mask that is used to relax the SVM results afterwards. The partitioning of the images is done using the spectral signatures as features. Two hyperspectral images are used in the experiments, AVIRIS Indian Pine containing 16 types of vegetation and ROSIS Pavia University containing 9 land covers (buildings, asphalt, green area, trees, etc.). The obtained correct classification rates for the AVIRIS image are 80.60% and 71.90% for SVM+ISODATA and SVM+EM respectively. While for the ROSIS image the correct classification rates are 92.94% and 95.21% for SVM+ISODATA and SVM+EM respectively. SVM cooperating with ISODATA globally gives better classification results. In addition before partitioning the image using EM, a band selection process is required.

- **SVM, Watershed, EM and Recursive Hierarchical Segmentation (RHSEG)**

In [98] the same authors propose another parallel partitioning approach that uses SVM classifier, watershed segmentation, segmentation by EM and RHSEG segmentation. This approach is designed for hyperspectral images. The fusion process in this approach is based on majority voting and minimum spanning forest (MSF). The partitioning of the images is done using the spectral signatures only and no features are extracted. This system is tested on the same images used in [77]. The correct classification rates are: 94.28% for Indian Pine image, and 98.50% for Pavia University image.

The disadvantage of these approaches resides in using supervised and parametric methods like SVM, ISODATA, and EM.

- **Only ML (changing features)**

A recent parallel approach proposed in [78] to partition hyperspectral images that uses only ML classification method to partition the same image changing the features extracted from the image each time, and then the results are fused to get the final result. The fusion technique is based on weighted-linear opinion pool (WLOP), and weighted majority voting (WMV) to combine the class labels from this bank of classifiers. The partitioning of the images is done using the derivatives of the spectral signatures to detect variations in chemical stress on the corn crop from an airborne hyperspectral image. The obtained correct classification rate is 80%. The disadvantage of this method is that it uses a parametric method.

In conclusion parallel cooperation has many advantages: *i)* the classification methods are applied independently, *ii)* there is no imposed order of the applied methods that highly conditions the results, *iii)* applying the methods in parallel reduces computation time. The main overall difficulty of the parallel scheme is the fusion process that requires a robust decision rule to give optimal results.

1.3.3 Hybrid cooperation

The third cooperation scheme to partition an image is the hybrid cooperation that combines the two previous schemes simultaneously. In other words hybrid methods use elaborated concept such as adding intentionally intermediate results, context adaptation according to the obtained results. In the following we present some methods which cooperate in a hybrid way.

- **Neural network (NN) and ML**

A hybrid approach, presented in [79], involves the cooperation of Neural Networks (NN) and ML method. In this approach NN and ML methods classify the pixels of the image in parallel; then the results are fused and a set of validated and invalidated pixel results are obtained. The invalidated results are classified by another NN. The drawback of this approach is the use of supervised and parametric classification methods. This approach is designed for hyperspectral images. The partitioning of the images is done using the spectral signatures. This approach is tested on the hyperspectral AVIRIS Hekla (active volcano in Iceland) image that contains 16 different land covers. The best correct classification rate is 91%.

- **SOM, HGA, FCM, HDGA , NURB and GA**

In [7]–[9] a hybrid cooperative multicomponent image partitioning system using the minimum *a priori* knowledge is proposed by Awad et al. The partitioning methods used in this system are nonparametric. This system is composed of three subsystems; each of them is composed of more than one classification methods which cooperate in a sequential way. Then the results of the subsystems are fused to obtain the final partitioning result (see Figure 1.10).

The system works by analyzing the image in several hierarchical levels of complexity while integrating several methods in cooperation mechanisms. Three sequential cooperative approaches are created between different methods such as SOM (Self-Organizing Map)- HGA (Hybrid Genetic Algorithm), FCM (Fuzzy C-Means)-HDGA (Hybrid Dynamic Genetic Algorithm), and Non-Uniform Rational B-Spline (NURB-HDGA) [99], [100].

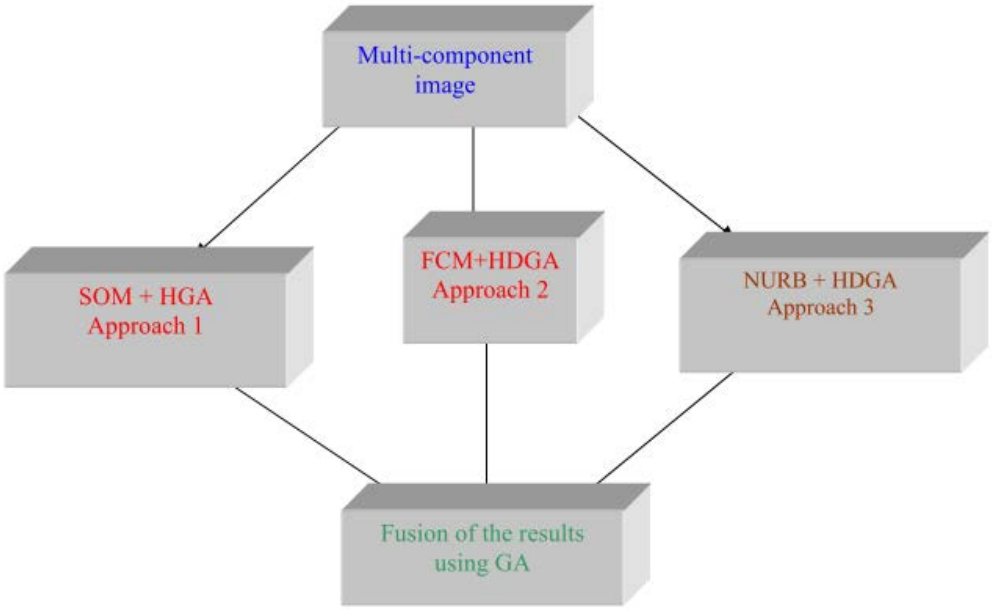


Figure 1.10: Hybrid cooperative partitioning system [9]

In order to combine the results of the above three approaches, a genetic fusion process is used. This approach is tested on three multicomponent satellite images Ikonos, Landsat and SPOT to detect land covers, i.e. urban, bare land and agriculture. The classification rates are: 97%, 88% and 97% respectively. These results are obtained by using only the pixels values, and no other features are extracted.

The disadvantage of this approach is that: *i)* some of the methods used require *a priori* knowledge, and *ii)* the evaluation technique used in the fusion process also needs some parameter initialization.

- **MLBG, GA and *k*-means**

Another hybrid cooperative and adaptive system is developed by Rosenberger et al. in [10], [101] for partitioning mono and multicomponent images. In this system the image to be partitioned is processed in many different steps, first the image is globally analyzed to determine different region types; then features of pixels are extracted according to the region types. A classifier called MLBG is used to partition the image via the features extracted. The author proposes using the same classifier on the same image to obtain different results. The results are then fused using a GA with an

unsupervised evaluation criterion. The evaluation criteria used is based on within-class and between-class disparities that are calculated in different ways for the textured and non-textured regions in the image. The evaluation of the developed system is done using different texture features extracted from synthetic images. The obtained correct classification rate for the monocomponent synthetic images is 93%.

The disadvantages of this system are: *i)* the classifier used (MLBG) is very time consuming, *ii)* same classifier is used several times, to get different results, *iii)* while calculating the within-class disparity for the textured regions a *k*-means is used in the process which is not a stable method and adds a very big overhead time to the evaluation process, *iv)* mutation is not used in the GA, this makes it easily get trapped in local minima, and finally *v)* the GA fusion used is very time consuming.

1.3.4 Discussion

The approaches presented in Section 1.3 show that cooperation between classification methods is a very interesting direction of research and worth further studying to resolve the different problems of image partitioning. Cooperative techniques appear to combine the advantages of non-cooperative methods, and to correct or to cover the disadvantage of a method by the advantages of the others.

Sequential cooperation approaches generally lead to robust algorithms but have the inconvenient of requiring a predefined order for sequencing the methods. Parallel cooperation approaches do not have the problem of sequencing and offers the advantage to produce redundant results which can be used as additional confirming information in the fusion process. Hybrid approaches combine the two above approaches at the same time; this cooperation type has the same disadvantages as sequential cooperation, and in addition the design of such methods is difficult to implement.

A summarized analysis of sequential, parallel and hybrid cooperative approaches are shown in Tables (B.1, B.2 and B.3) respectively in Appendix B.

1.4 Conclusion

From this literature review we can provide several conclusions. First of all, the problem of image partitioning is not yet satisfactorily resolved. The diversity of the information contained in images is the major problem of the partitioning systems developed until now. Non-cooperative partitioning methods give good results for some particular types of images and may not be applied to all types of images.

These methods use the same strategy for the whole image, however real images are rarely totally textured or totally non-textured. Applying one classification method to the entire image does not give reliable results because of the diversity of the information it contains. This is why adapting multiple strategies in a cooperative classification approach appears to be a good direction of research. Indeed it a fact that more and more researchers focus their effort on developing cooperative approaches.

As we have seen above most of the presented cooperative approaches partition an image without considering the types of regions it contains. Besides, most of the partitioning methods make implicit assumptions about the input images; however these assumptions are often not verified in practice. A relevant approach must identify the failure of a method and use this information to adjust the final classification result.

To achieve a reliable partitioning of an image, a suitable partitioning method must be applied locally to the data. While designing a partitioning system, the emphasis should be focused on the development of a local contextual correspondence between several appropriate partitioning methods.

In the sequel, we have chosen to adapt the partitioning of an image by detecting the presence or absence of texture. This choice seems justified since the features used to characterize these types of area are relatively distinct. Following this principle, the pixels in the different region types must be characterized by adapted sets of features. The average of pixel values is sufficient to classify the pixels of the non-textured regions, while classifying the pixels from the textured regions is more sensitive and requires more features.

In the overall strategy of the system to be defined, the difficulty relies in the selection of appropriate partitioning methods for each area type. Insofar as we take a special interest in unsupervised approaches, we have chosen to develop a general and unsupervised cooperative partitioning paradigm. This scheme is expected to present many advantages:

- Effectiveness for partitioning different region types (textured, non-textured).
- Natural adaptation to the local nature of the image by choosing appropriate features of pixels in the textured and non-textured regions.
- Automatic estimation of the number of classes.
- Fast convergence.

Chapter 2 Evaluation criteria

2.1 Introduction

The evaluation of classification results is an unavoidable process used to quantify the performance of existing partitioning or segmentation algorithms. In addition this process can also be used in designing classification methods or approaches.

The quality evaluation of a classification result is an active area of research and many criteria are being developed regularly. Unfortunately, the evaluation of a partitioning result always contains some elements of subjectivity and the criteria do not always give satisfactory evaluation. For this reason, it is impossible to define a universal criterion to evaluate the results produced by all the existing criteria. However, a number of criteria exist and are repeatedly used by many researchers to compare classification results. Since there are a large number of possible partitioning results for the same dataset, the objective is to assess whether any of these results is better than another. So to correctly evaluate the partitioning results, it can be necessary to use multiple evaluation criteria.

In this chapter, we provide generalities of the evaluation criteria, and then focus on the unsupervised ones.

2.2 Evaluation criteria types

Several types of evaluation methods have been proposed in the literature [102]–[104]. They are classified into three main groups. The first group contains unsupervised criteria that use only internal information of the data such as the distance between objects. These criteria are also called internal quality measures. The second group contains supervised criteria that calculate the degree of correspondence between the clustering produced by the algorithm and a known data partitioning. These criteria are also known as external quality measures. The last group is called relative criteria; this type of evaluation allows comparing the results obtained from

the same algorithm. These measures are simply the use of internal or external criteria to evaluate multiple results produced by the same algorithm and to choose the best one among them. As the partitioning system proposed in this thesis is unsupervised, we will review only internal quality criteria.

2.3 Unsupervised evaluation criteria

Unsupervised evaluation criteria [105] are based on internal information and do not need any *a priori* knowledge. This type of criteria generally computes statistical measures such as the standard deviation or the disparity of the classes. These measures are often based on the simplest definition of partitioning which says that objects from the same class should be as close as possible, and that objects from two distinct classes should be as far apart as possible [106]. To assess whether a classification result complies with this intuitive definition, the distances between the class centers and the class objects are calculated. These unsupervised measures assess the compactness and the separability of the classes. The evaluation of the quality of a partition is not formally defined, so there are many different criteria, which estimate the quality of the results differently. Some of these criteria can be directly used as the objective function of a classification algorithm. However others are very time-consuming, and therefore intended to be calculated after the application of the algorithm for the final evaluation process.

One of the most intuitive criteria able to quantify the quality of a partitioning result is the within-class uniformity. The simplest way to calculate this uniformity is the sum of the squared errors (SSE) which is calculated as follows:

$$SSE\left(I_R\right) = \sum_{i=1}^{NC} \sum_{x \in c_i} d\left(x - g(C_i)\right)^2 \qquad (2.1)$$

where $g(C_i)$ is the center of the class C_i and d is a distance measure.

Weszka and Rosenfeld [107] proposed such a criterion with thresholding that

measures the effect of noise to evaluate some thresholded images. Based on the same idea of within-class uniformity, Levine and Nazif [108] also defined a criterion that calculates the uniformity of a class as follows:

$$LEV1(I_R) = 1 - \frac{1}{N}\sum_{i=1}^{NC} \frac{\sum_{x \in C_i}\left[g_l(x) - \sum_{x \in C_i} g_l(x)\right]^2}{\left(max_{x \in C_i}\left(g_l(x)\right) - min_{x \in C_i}\left(g_l(x)\right)\right)^2} \qquad (2.2)$$

where

- I_R is the partitioning result of the image I into NC classes $C = \{C_1, ..., C_{NC}\}$,
- N is the number of pixels of the image I,
- $g_l(x)$ is the gray level of pixel x in the image I.

A standardized uniformity measure was proposed by Sezgin and Sankur [109] that is based on the Cochran homogeneity measurement [110]. However, this method requires a threshold that is often arbitrarily selected, thus limiting the usage of this criterion. Another criterion to measure the within-class uniformity was developed by Pal and Pal [111]. It is based on a thresholding that maximizes the local entropy of the classes in a partitioning result. In the case of slightly textured images, these criteria of within-class uniformity prove to be effective and very simple to use. However, the presence of textures in an image often generates improper results due to the over-influence of small regions.

Complementary to the within-class uniformity, Levine and Nazif [108] defined a disparity measure between two classes to evaluate the dissimilarity of different classes in a partitioning result. The formula of total between-class disparity is defined as follows:

$$LEV2(I_R) = \frac{\sum_{k=1}^{NC} w_{C_k} \sum_{j=1/C_j \in W(C_k)}^{NC} \left[p_{C_k \setminus C_j} \left(\left| g_l(C_k) - g_l(C_j) \right| / \left((g_l(C_k) + g_l(C_j)) \right) \right) \right]}{\sum_{k=1}^{NC} w_{R_k}} \qquad (2.3)$$

where w_{c_k} is a weight associated to C_k that can be dependent of its area, $g_l(C_k)$ is the average of the gray level of C_k and $p_{C_k \setminus C_j}$ is the length of the boundary of the class C_k common to the perimeter of the class C_j. This type of criterion has the advantage of penalizing over-segmentation.

Zeboudj [112] proposed a measure based on the combined principles of maximum between-class (external) disparity and minimal within (interior) class disparity measured at the pixel's neighborhood.

Let $c(x,z) = \frac{|g_l(x) - g_l(z)|}{L-1}$ be the disparity between two pixels x and z x and $z \in C_i$, and L be the maximum gray level.

The interior disparity $CI(C_i)$ of the class C_i is defined as follows:

$$CI(C_i) = \frac{1}{NC_i} \sum_{x \in C_i} \max\{c(x,z),\ z \in V_s,\ z \in C_i\} \tag{2.4}$$

where NC_i is the number of pixels in class C_i and V_s is the neighborhood of the pixel x.

The external disparity $CE(C_i)$ of the class C_i is defined as follows:

$$CE(C_i) = \frac{1}{p_i} \sum_{x \in C_i} \max\{c(x,z), z \in V_s,\ z \notin C_i\} \tag{2.5}$$

where p_i is the length of the boundary of class C_i.

Lastly, the disparity of the class C_i is defined by the measurement $D(C_i) \in [0, 1]$ expressed as follows:

$$D(C_i) = \begin{cases} 1 - \dfrac{CI(C_i)}{CE(C_i)} & if \;\; 0 < CI(C_i) < CE(C_i) \\ CE(C_i) & if \;\; CI(C_i) = 0 \\ 0 & otherwise \end{cases} \tag{2.6}$$

Zeboudj's criterion is defined by:

$$ZEB(I_R) = \frac{1}{N} \sum_{i=1}^{NC} NC_i x D(C_i) \tag{2.7}$$

where N is the number of pixels in the image.

This criterion has the disadvantage of not correctly taking into account strongly textured regions.

Another criterion that is based on the combination of the within-class and between-class disparities is the Davies-Bouldin index [113]. It estimates the within-class disparity based on the distance from the points in a class to its centroid and the between-class disparity based on the distance between centroids. It is defined as:

$$DB(I_R) = \frac{1}{NC} \sum_{i=1}^{NC} \max_{j, j \neq i} \left\{ \frac{\frac{1}{NC_i} \sum_{k=1}^{NC_i} d\left(F_k, g(C_i)\right) + \frac{1}{NC_j} \sum_{k=1}^{NC_j} d\left(F_k, g(C_j)\right)}{d\left(g(C_i), g(C_j)\right)} \right\} \tag{2.8}$$

where F_k is vector of Nf features representing the pixel x_k.

Another criterion of this type is the Silhouette index [114]. This index is a normalized summation-type index. The within-class is measured based on the distance between all the points in the same cluster and the separation is based on the nearest neighbor distance.

Let $d_1(x_j)$ be the average dissimilarity of x_j with all other pixels of its class C_i $d_1(x_j)$ indicates how well x_j is assigned to its class (the smaller the value, the better the assignment).

Let $d_2(x_j)$ be the lowest average dissimilarity of x_j to any other class C_l *with* $l = 1,2,...,K;\ l \neq i$.

The class with the lowest average dissimilarity is said to be the "neighboring cluster" of x_j because it is the next best-fit class for it; and then the size of the silhouette $Sil(x_j)$ is defined as:

$$Sil(x_j) = \frac{d_2(x_j) - d_1(x_j)}{max\left[d_2(x_j), d_1(x_j)\right]} \tag{2.9}$$

Basing on the definition of $Sil(x_j)$, the silhouette of the class C_i is defined as:

$$Sil(C_i) = \frac{1}{NC_i} \sum_{x_j \in C_i} Sil(x_j) \tag{2.10}$$

Finally the global silhouette for a partition is defined as:

$$Sil(I_R) = \frac{1}{NC} \sum_{i=1}^{NC} sil(C_i) \tag{2.11}$$

This criterion is very efficient but its time complexity makes it inapplicable to large datasets.

The Dunn's index (*Du*) [115] is another unsupervised criterion that measures the compactness of a class and the separateness between classes as follows:

$$Du\left(I_R\right)=\frac{\min\limits_{i=1:NC}\left(\min\limits_{j=1:NC,j\neq i}\left(d_{ij}\left(g(C_i),g(C_j)\right)\right)\right)}{\max\limits_{i=1:NC}\left(d_{ii}(g(C_i),g(C_i)\right)} \tag{2.12}$$

where $d_{ij}\left(g(C_i),g(C_j)\right)$ is the distance between the center of classes C_i *and* C_j, which is defined here as the minimum distance between the objects of different classes (see Equation (2.13)). $d_{ii}(g(C_i),g(C_i))$ is the maximum distance between two objects in the same class (see Equation (2.14)).

$$d_{ij}\left(g(C_i),g(C_j)\right)=\min_{x\in C_i,y\in C_j} d(x,y) \tag{2.13}$$

$$d_{ii}\left(g(C_i),g(C_i)\right)=\max_{x,y\in C_i} d(x,y) \tag{2.14}$$

This evaluation criterion has two disadvantages: firstly, it is very time consuming and secondly it is highly affected by the presence of noise in the dataset.

In [10], [101] Rosenberger and Chehdi presented a criterion that enables estimating the within-class homogeneity and the between-class disparity considering the types of regions (textured or non-textured) in the partitioning result. This criterion quantifies the quality of a partitioning result as follows:

$$ROS\left(I_R\right)=\frac{1+\bar{D}\left(I_R\right)-\underline{D}\left(I_R\right)}{2} \tag{2.15}$$

The global within-class disparity $\underline{D}\left(I_R\right)$ quantifies the homogeneity of each class obtained in the partitioning result I_R of image *I*. On the other hand, the global between-class disparity $\bar{D}\left(I_R\right)$ quantifies how well the classes obtained are separated from each other.

The global within-class disparity $\underline{D}(I_R)$ reflects the statistical stability of each class. It is calculated from the within-class disparity $\underline{D}(C_i)$ of the different classes in a partitioned image:

$$\underline{D}(I_R) = \frac{1}{NC}\sum_{i=1}^{NC}\frac{NC_i}{N}\underline{D}(C_i) \tag{2.16}$$

The weight of the within-class disparity of a class C_i in the global within-class disparity is proportional to the number of pixels for this class. The same principle is used to calculate the between-class disparity $\bar{D}(I_R)$ of the partitioned image I_R that measures the disparity of each class with the other classes:

$$\bar{D}(I_R) = \frac{1}{NC}\sum_{i=1}^{NC}\frac{NC_i}{N}\bar{D}(C_i) \tag{2.17}$$

This criterion is calculated using the between-class disparity $\bar{D}(C_i)$ and the within class disparity $\underline{D}(C_i)$ of each class C_i. The calculation of these two criteria is detailed in the following:

– ***Within-class disparity criterion***

This criterion evaluates the homogeneity of a class, i.e. the variation of the statistics in the interior of this class. In the calculation of the within-class disparity, the nature of the regions (i.e. textured and non-textured) is taken into account.

In the non-textured case, this criterion for class C_i is defined as:

$$\underline{D}(C_i) = \sqrt{\frac{1}{NC_i}\sum_{x\in C_i} g_l(x)^2 - \frac{1}{NC_i^2}\left(\sum_{x\in C_i} g_l(x)\right)^2} \tag{2.18}$$

This criterion is sufficient to characterize the within-class disparity of a non-textured region. However, in the textured case, each class is characterized by a set of

texture feature vectors. The dispersion of this set of vectors allows calculating the within-class disparity in the textured case.

– ***Between-class disparity criterion***

The evaluation process of between-class disparity of a class is similar to the with-in class disparity, but instead of estimating the homogeneity of a class, it is disparity with the other classes is calculated. The between-class disparity is also calculated according to the nature of the regions as follows:

- Between classes of the same region type:
  - The disparity between two classes belonging to uniform regions $\bar{D}\left(C_i, C_j\right)$ is defined as:

$$\bar{D}\left(C_i, C_j\right) = \frac{\left| g_l\left(C_i\right) - g_l\left(C_j\right) \right|}{NG} \tag{2.19}$$

    where *NG* is the number of the gray levels in the image
  - The disparity between two classes belonging to textured regions $\bar{D}\left(C_i, C_j\right)$ is defined as:

$$\bar{D}\left(C_i, C_j\right) = \frac{d\left(g\left(c_i\right), g\left(c_j\right)\right)}{\left\| g\left(c_i\right) \right\| + \left\| g\left(c_j\right) \right\|} \tag{2.20}$$

    where *d(.,.)* is the Euclidean distance, $g(C_i)$ is the centroid of class C_i, and $\| . \|$ denotes the Euclidean norm.

- Between classes of different region types: the disparity between classes of different region types is set as the maximum value, i.e. 1.

2.4 Conclusion

The evaluation of classification results is unavoidable to assess the quality of the results obtained.

In this chapter, we have presented various unsupervised evaluation criteria used to assess the quality of a partitioning result. These criteria are also called internal criteria because they do not use any external information in the evaluation process. Some of these criteria are effective in the case of non-textured or slightly textured images, while others give effective results in the case of textured images.

None of the evaluation methods can prove satisfactory in all the cases. Therefore, to correctly evaluate the algorithms and their results we have to use more than one evaluation technique and to combine their results.

Hyperspectral images are complex in their nature and contain different region types (i.e. textured and non-textured), so using an adaptive evaluation criterion that takes into account the nature of the regions of this type of images is recommended.

PART II: The developed system

Introduction

The research efforts in image partitioning have led to create a vast number of classification algorithms during the past decades. However, most of the developed non cooperative and cooperative approaches are dedicated to a specific application, and cannot be used in all general cases or applications.

To make these systems more efficient, it would be necessary that they have general capacity, flexibility and adaptability to classify the image content for a wide range of applicative domains. In this way, we propose in the framework of this thesis a partitioning system that adapts itself locally to the image content. We can expect better efficiency by adapting the cooperative process of partitioning to the data encountered, rather than applying a single non-cooperative classification method. In the case of using more than one partitioning method on the same data, the results of these methods are integrated in the final partitioning result using a genetic algorithm. In the case of multicomponent images, the classification results of all the components are fused to get the final result.

In the following, we first describe the principles of our proposed system, and secondly we give some details on the modules that compose it.

Scheme of the proposed system

To develop a partitioning system, it is necessary to adapt the classification process to the content of the image. This concept is inspired by the human visual system, where each component has a specific task, and these components cooperate to get the global correct vision of a scene.

▶ Monocomponent case

The basic system that we propose to partition a monocomponent image is composed of four modules: adaptive feature extraction, unsupervised parallel classification, evaluation and conflict management, and merging results. Figure 1

displays the overall layout of the proposed system and Figure 2 gives its extension to multicomponent evaluation and conflict management.

Module 1 (Adaptive feature extraction): in this module, the image is divided into two types of regions, i.e. textured and non-textured. The adaptive characterization of pixels, taking into account the textured or non-textured nature of the region to which they belong, is an essential step before the classification process. Indeed, the features dedicated to the description of regions with low variance do not have sufficient discriminating power for textured regions, and vice versa.

Module 2 (Unsupervised parallel classification): in this module, the image is partitioned using two different, unsupervised nonparametric classification methods (FCM and AILBG) selected after the analysis conducted in Part I, Chapter 1. These methods are optimized by estimating the number of classes in order to make them unsupervised. Moreover the problem of FCM instability is also resolved. We named these optimized algorithms FCMO and AILBGO. In this step, the pixels belonging to textured or non-textured regions are classified separately and in parallel, using appropriate feature sets.

Module 3 (Evaluation and conflict management): this module includes two validation processes. Firstly the pixels that are coherently classified by the two methods are validated, and secondly, the conflicting classification results are processed by using a GA. The objective function of the genetic algorithm is based on between-class and within-class disparities to evaluate and manage the conflicting pixels between the partitioning results.

Module 4 (Union of results): in this module, the results of textured and non textured regions are unified in the same labeled image.

Multicomponent case

In this case the four modules are first applied independently on each component, and the corresponding results are managed and fused by a dedicated process (see Figure 2). In this module, an evaluation and conflict management procedure which includes the identification of pixels belonging to identical classes of partitioning results for adjacent components is applied. In this step, the results from the different components are grouped into subsets, depending on the number of pixels that are classified to the same class in different components. Then these subsets are processed independently to get one classification result for each of them. The same process as in the third module of Figure 1 is used to evaluate and fuse the subsets results and then getting the final result of the multicomponent image.

The remaining of this part is organized in two chapters. The first one is dedicated to the description the region nature detection and the adaptive feature extraction (module 1). The second chapter presents the details about the unsupervised cooperative classification. This chapter includes the optimization of the FCM and AILBG algorithms (module 2), the process of evaluation and conflict management and merging results (modules 3 and 4). Besides, it includes also the assessment of the developed system on real applications.

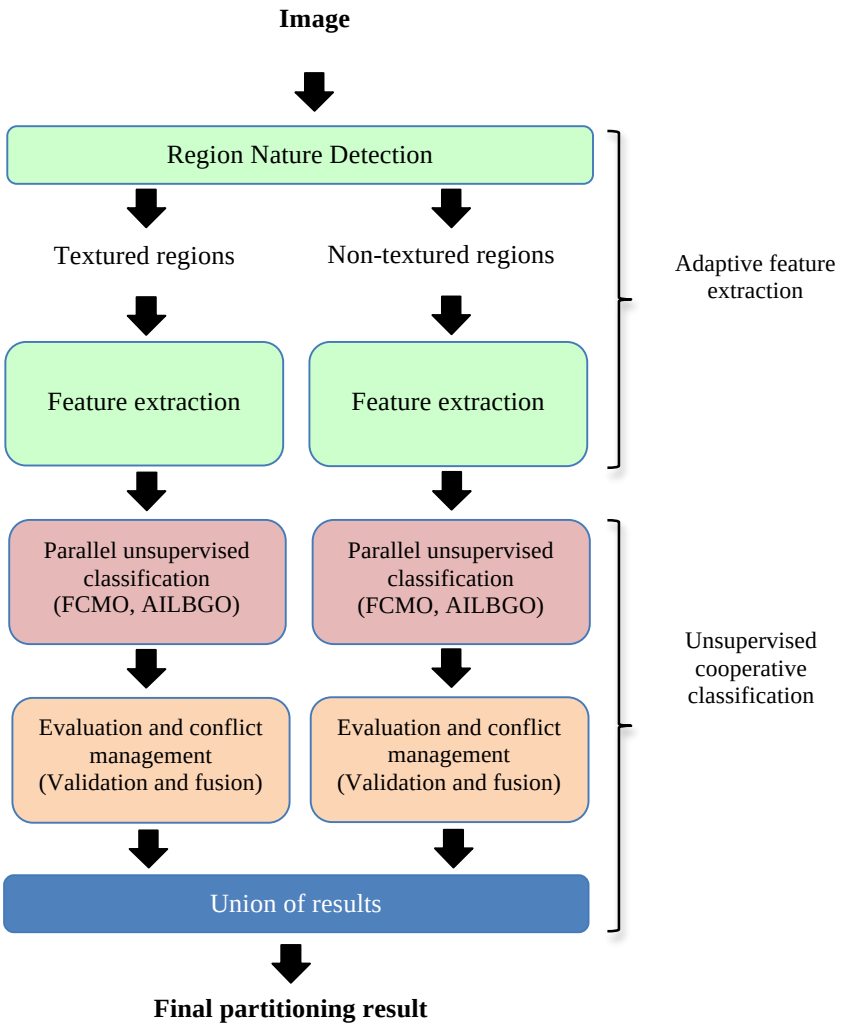


Figure 1: The general layout of the proposed basic partitioning system (case of a monocomponent image)

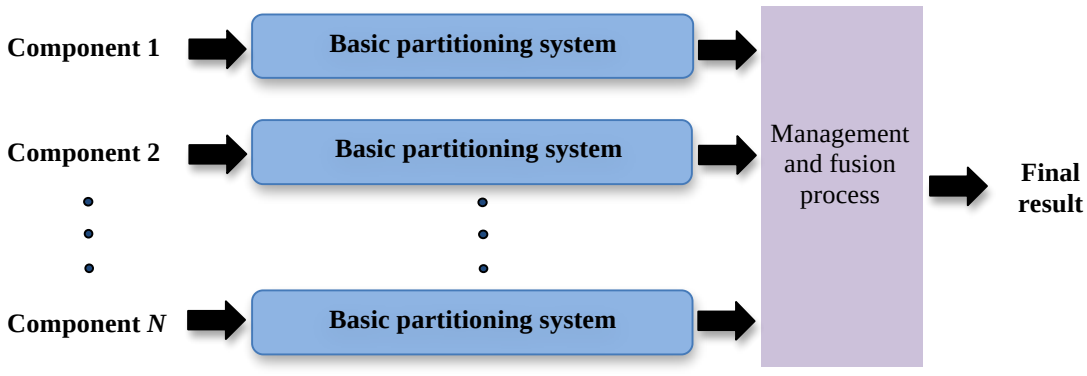


Figure 2: The general layout of the proposed partitioning approach (case of a multicomponent image)

Chapter 3 Region nature detection and adaptive feature extraction

3.1 Introduction

In this chapter, we present the proposed module of detection of the textured and non-textured regions in the image. We recall that the distinction between these two types of regions permits extracting appropriate feature sets of pixels for the different types of regions is introduced in order to obtain optimal partitioning results. The idea of dividing the image into different region types was introduced by Rosenberger and Chehdi [116].

This module is composed of two steps:

- ***Region nature detection***:

  To assign a pixel to one of the textured or non textured classes, we have used the uniformity feature, which is calculated from the co-occurrence matrices [117].

  - *Global detection*

    Here, the global nature of the image is identified by calculating the uniformity feature on the whole image.

  - *Local detection*

    To classify the pixels into two categories (textured and non-textured), the uniformity feature is calculated locally and in a multi-resolution framework by changing the size of the analysis window. The window sizes are chosen according to the global nature of the image.

- ***Adaptive feature extraction***: in this step the pixels in each region type detected in the previous step are characterized by a different set of features in order to be classified afterwards.

In the following section, first we recall the detection method developed in [116] used in the detection of the global nature of an image to adapt the window sizes and the features extracted. Secondly we describe in details its optimization for local region nature detection and provide some experimental results.

3.2 Region nature detection

If we refer to the literature, the characterization of textured and non-textured regions is usually done by considering the standard deviation [118]. In this case, a region in an image is considered as textured if its standard deviation is greater than a predefined threshold. However, the characterization of the region types using only the standard deviation is not sufficient. The notion of texture is more complex and it is related to the resolution of the observation; for example, among two images having the same standard deviation, one could be textured and the other non-textured [10].

For this reason, we have chosen a method based on the uniformity feature calculated from the co-occurrence matrix [117], which is well adapted to describe textures.

In the following sub-sections we describe in details the developed method.

3.2.1 Uniformity criterion to detect the global nature of an image

The developed scheme for the detection of the global nature of an image is given in Figure 3.1 [10], [116]. In this scheme, the uniformity criterion is used [117]. This feature characterizes the frequency of transitions between identical gray levels of a pixel and its connected neighborhood. If the co-occurrence matrix trace (uniformity feature) is greater than the sum of other elements of this matrix, this reflects homogeneity of transition, which is characteristic of uniform areas. Contrarily, disordered transitions indicate the presence of texture [117]. Unfortunately, the calculation time of this matrix is very high for an image in its original gray levels. In order to reduce the gray levels number of an image while preserving the significant information at the same time, we have used the multi-threshold method described in

[12]. This multi-threshold process selects the significant gray levels after a local analysis of the image. The approach consists of classifying the points of the image according to their gray levels using threshold values determined by analyzing the global histogram, this one being calculated from the significant modes issued by local histograms [12]. This procedure eliminates the irrelevant information and brings out the most important elements of the textures.

The co-occurrence matrix of the multi-threshold image has a lower dimension than the one of original image and is then easier to handle on one hand, and on the other hand it is more robust when the image is affected by noise, because only the significant transitions are preserved after the thresholding process.

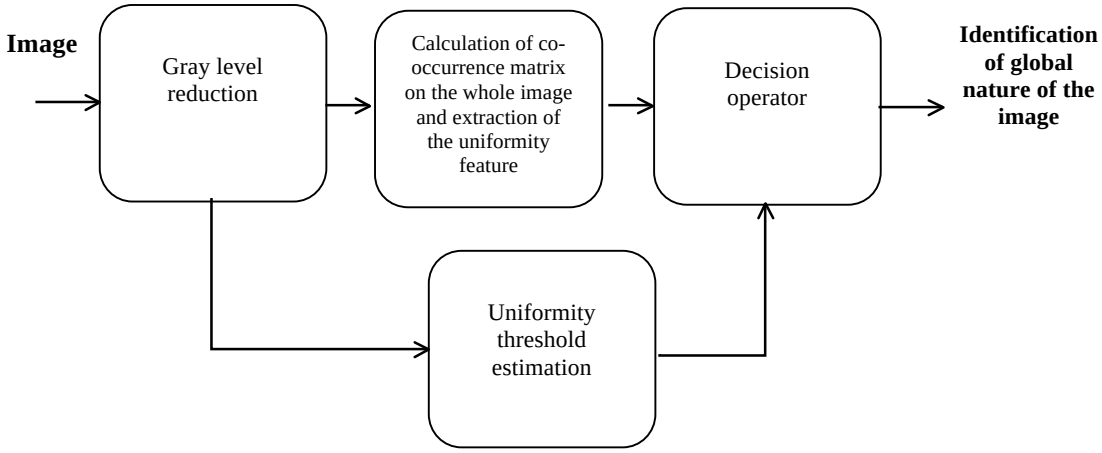


Figure 3.1: Diagram of region type detection by automatic thresholding

The uniformity parameter U calculated on the multi-thresholded image I_M is defined by:

$$U\left(I_M\right) = \frac{1}{ND}\sum_{j=1}^{ND}\sum_{i=1}^{NG} P_{d,\theta_j}\left(i,i\right) \tag{3.1}$$

where, NG is the number of gray levels in the thresholded image I_M, $P_{d,\theta}(.,.)$ are the entries of the co-occurrence matrix obtained with inter-pixel distance d ($d=1$and $\sqrt{2}$), and having orientation θ_j (0°,45°, 90° and 135°). The decision criterion is as follows:

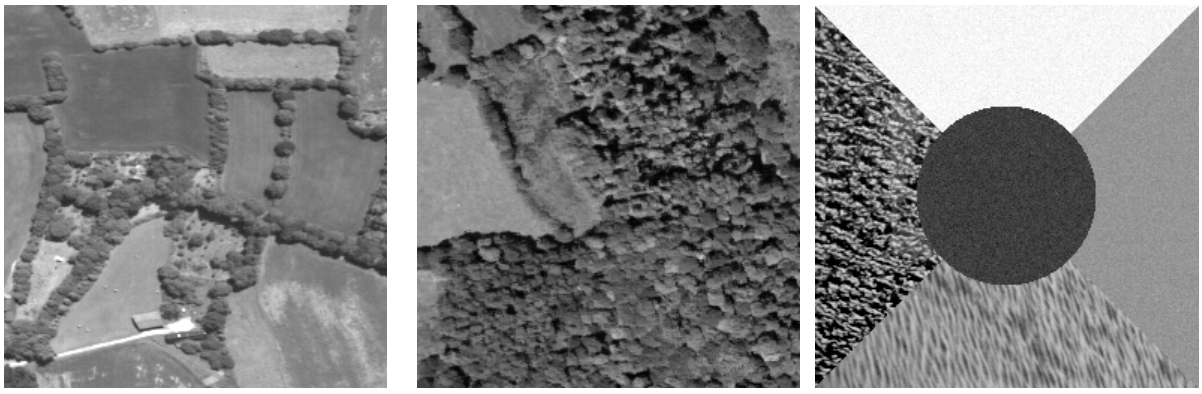
$$\begin{cases} if\ U(I_M) < \bar{U}(I_M) \text{ then } I \text{ contains more textured regions} \\ else\ I \text{ contains more non textured region} \end{cases} \quad (3.2)$$

where, $\bar{U}(I_M)$ is an adaptive threshold parameter also estimated from the thresholded image I_M as follows:

$$\bar{U}(I_M) = 1 - \left(\frac{NG-1}{NG} \right)^8 \quad (3.3)$$

The value of $\bar{U}(I_M)$ reflects the probability of transition of a pixel gray level with 8 connected neighborhood under the hypothesis of gray level independence.

To verify the validity of this criterion and show its insensitivity to gray level variation of image I_M, we have tested it on a large database of real and synthetic images. Figure 3.2 shows three example images and Table 3.1 shows the values of U *and* $\bar{U}$ for each image by varying the number of gray levels given by multi-thresholding method and also the detection results of the images global nature. The gray level variation is obtained by choosing different window sizes in the multi-thresholding method used. The results obtained show the independence of the criterion to the number of the gray levels in the multi-threshold image. The visual analysis confirms the efficiency of the criteria, because all the tested images are correctly classified (i.e. in majority textured or in majority non-textured).



(a) GSTB04 (b) GSTB06 (c) SYNTH01

Figure 3.2: Sample of monocomponent images used to validate the global uniformity criterion

Table 3.1: Detection of global image nature by varying the number of gray levels

Image	Number of gray levels	U	$\bar{U}$	Detection result of global nature of an image
GSTB04	20	0.44	0.33	Non-textured
	15	0.53	0.42	Non-textured
GSTB06	15	0.33	0.42	Textured
	12	0.41	0.50	Textured
SYNTH01	19	0.39	0.35	Non-textured
	14	0.53	0.45	Non-textured

This experiment confirms that the detection method is independent of the number of gray levels in the multi-thresholded images. Therefore, this step will be used in our system to better adapt the further processing to the content of images.

3.2.2 Detection of local textured and non-textured regions

In [10], [116], the previous detection step is applied locally and in multi-resolution on the image to identify if a pixel belongs to textured or non-textured regions in order to adapt the extraction of the features. However, its application is not optimal, since pixels belonging to the edge of a non-textured region near a textured region might be labeled as textured. In order to overcome this problem, we have used the same scheme, but instead of using the decision operator that compares U and $\overline{U}$, we have used a classifier after extraction of the multi-resolution uniformity feature as shown in Figure 3.3.

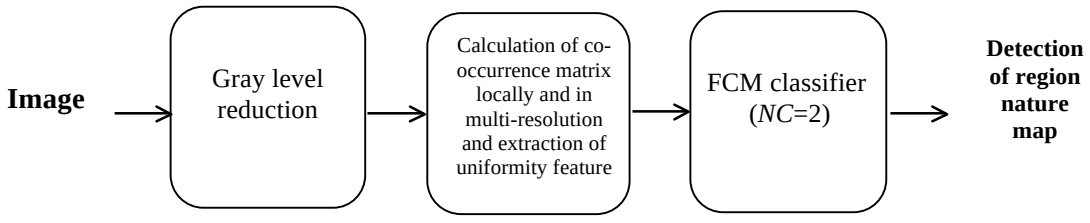


Figure 3.3: Local region type detection

The notion of textured and non-textured regions in the image is related to the resolution of observation, this is why a multi-resolution approach, calculating the uniformity feature *U* at different resolutions using several window sizes, is used in [10], [116]. The choice of the window sizes is done according to the global nature of the image identified in the previous step.

If the image is globally composed of uniform regions, the uniformity feature is calculated using different window sizes $N_{Wi} \times N_{W_i}$ for $i = 1,..,5$ where $N_{Wi} = 2^i + 1$ on image *I* with N_{COL} columns and N_{LIN} rows (see Figure 3.4). On the other hand, in the case where the image is globally composed of textured regions, the sizes of the windows must be greater to take into account all texture types. The uniformity parameter is calculated using windows of size: $N_{Wi} = 2^{i+2} + 1$.

The uniformity feature $U(W_i)$ is calculated from the co-occurrence matrix of the window W_i centered on the current pixel *x*. If *x* does not have sufficient neighborhood, which is the case when the pixels are on or near the borders of the image, we apply image mirroring for symmetry. The number of rows and columns mirrored depend on the size of W_i, which is equal to $\frac{N_{Wi} - 1}{2}$ rows and columns added to each edge.

After this step, each pixel is characterized by a set of five uniformity features (one for each resolution) which are extracted using different window sizes. These features are injected into a partitioning method to classify pixels into two region types.

To detect the nature of regions, by using the features extracted, three different semi-supervised classifiers, namely: *k*-means, AILBG, and FCM can be applied.

Here, we assessed the performance of the proposed detection approach by using FCM, AILBG or *k*-means algorithms as classifiers, and we have compared it with the approach based on the uniformity threshold estimation [116].

For this experimental study we have tested the proposed region nature detection approach on a database of 100 synthetic images and on a set of real images. The database of synthetic images includes images composed of two types of regions, i.e. textured regions (more than 300 textures are taken from the Brodatz album [36]) and

three non-textured regions. The average correct detection rates for the synthetic images using FCM, AILBG and *k*-means are respectively: 98.60%, 93.12%, 89.72%, whereas the detection by automatic thresholding provided a rate of 85.13%.

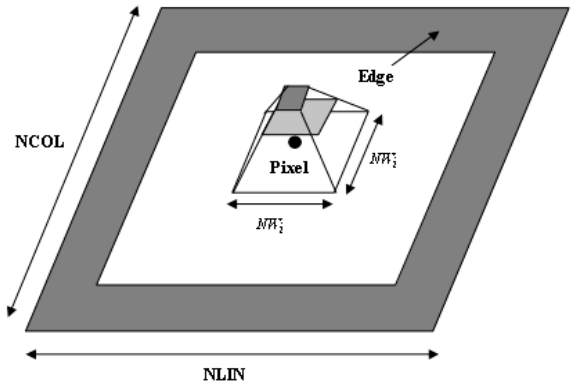


Figure 3.4: Multi-resolution feature extraction

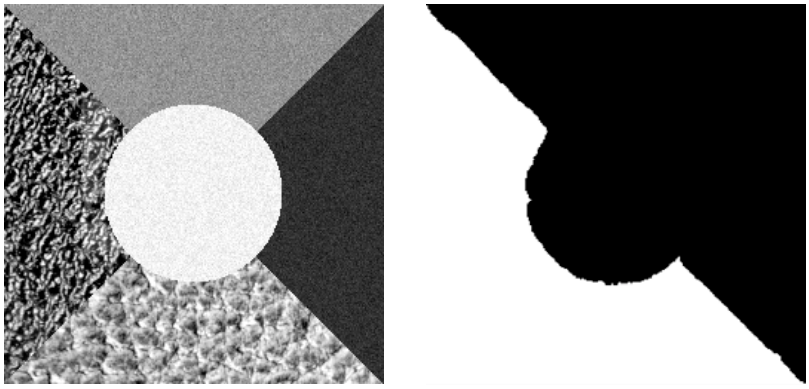
Figure 3.5 shows an example of the region nature detection using FCM classification on two synthetic images. We can observe that the results for the two images are relevant, where the boundary between the texture and non textured regions are accurate.

We have also tested this detection method on real gray level images. We have been observing that the detection using the FCM algorithm gives the best results, which is confirmed by visual inspection on Figure 3.6. This is the reason why we will be using this method in our partitioning system.

3.3 Adaptive feature extraction

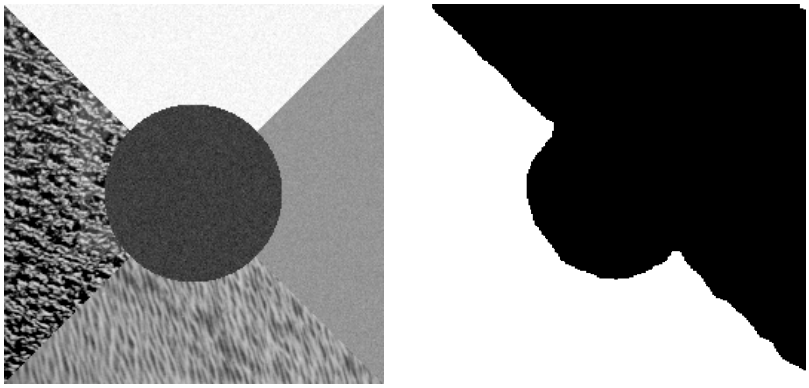
The choice of an appropriate feature extraction method for pixel characterization is a difficult task. There exist many methods in the literature, but each of them is adapted to some specific type of images, and gives reliable results for a limited type of applications. To be able to partition a large variety of images and give correct result for a wider range of applications, the choice of the features according to the content of the image (i.e. textured and non-textured regions) is very important, as we have discussed in previous sections. Identifying pixels belonging to one between two types of regions brings two advantages, i.e. a natural adaptation of feature extraction, and a time reduction of the feature extraction step.

In the case where pixels belong to a non-textured region, the local average of pixel values is a sufficient feature to characterize it. However, in the case of a textured region, there are many features extraction methods, where each of them is adapted to a certain type of textures. Therefore to make a good adequacy between the feature extraction methods and the textured regions detected in the previous step, it would be necessary to extract several texture descriptive features.



(a) Original image

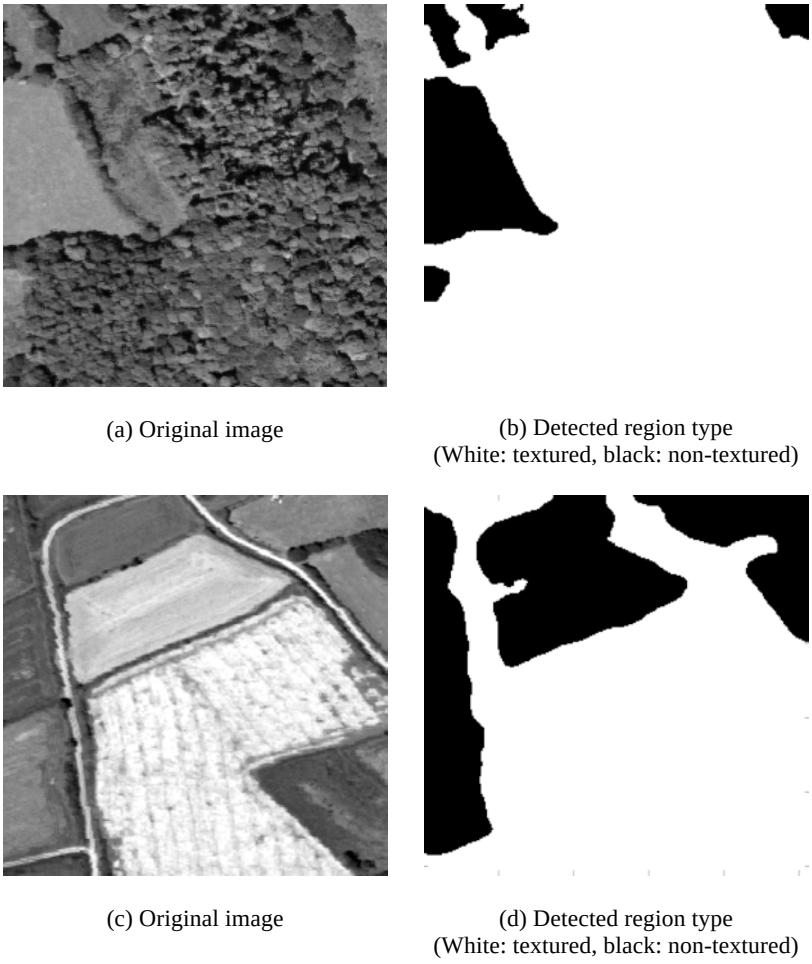
(b) Detected region type
(White: textured, black: non-textured)



(c) Original image

(d) Detected region type
(White: textured, black: non-textured)

Figure 3.5: Examples of region nature detection of synthetic images using FCM



(a) Original image

(b) Detected region type
(White: textured, black: non-textured)

(c) Original image

(d) Detected region type
(White: textured, black: non-textured)

Figure 3.6: Examples of region nature detection of real images by a classifier using FCM

3.3.1 Choice and analysis of the features

There are a large number of texture descriptive features in the literature; herein we analyze some of them to determine their redundancy and their discriminative power.

The features analyzed are the followings:

- Moments of order 1 to 4,
- 15 from co-occurrence matrix [117],
- 5 from run-length matrix [119],
- 2 from local histograms [120],
- 4 from local extrema [10],
- 4 from curvilinear integral [10].

These features are widely used in texture characterization. They are normally able to distinguish between different texture classes. The number of features is large (34 features in total), hence we have opted for the reduction method presented in [116], which calculates the correlation coefficients between the different features, and retains the less correlated ones. The correlation coefficient (ρ) between features F^i and F^j is calculated as follows:

$$\rho\left(F^i, F^j\right) = \frac{Cov\left(F^i, F^j\right)}{\sqrt{Cov\left(F^i, F^i\right) Cov\left(F^j, F^j\right)}} \tag{3.4}$$

where $Cov\ \left(F^i, F^j\right)$ is the covariance between two features F^i and F^j.

The correlation matrix between all features is symmetric and its elements have values in the interval $[-1,1]$. The correlation is high between two features F^i and F^j if $\left|\rho\left(F^i, F^j\right)\right|$ is close to 1. This will allow the identification of the features that do not give any additional information for the classification of the textured region. Therefore, two features are complementary and relevant if the absolute value of ρ is low. Otherwise, the features F^i and F^j are redundant if: $\left|\rho\left(F^i, F^j\right)\right| > \zeta$, where ζ is the maximum redundancy tolerated for two features. Practically ζ is set to a value very close to 1, to avoid losing any important information that describes the texture.

Once it is known that two features are redundant, we should discard one of them. To get this done we use the following criterion that takes into account the redundancy of one feature with respect to all other features:

$$\begin{cases} if \quad \sum_k \rho(F^k, F^i) > \sum_k \rho(F^k, F^j), \text{ then } F^i \text{ is discarded} \\ else \quad F^j \text{ is discarded} \end{cases} \tag{3.5}$$

By setting the threshold value ζ= 0.95, the features retained after the above procedure is 15 in total:

- The four statistical moments of order 1 to 4 (mean, variance, *skewness*, and *kurtosis*).
- 9 among the 15 standard features issued from the co-occurrence matrix, which are: contrast, correlation, inverse difference moment, sum average, sum entropy, entropy, first information measure of correlation, second information measure of correlation and contour information [10], [116], [117].

- 2 features from the curvilinear integral (using two angles) [10].

3.3.2 Adapting the feature extraction to the region types

After detecting the region type we use the local mean to characterize the pixels in the non-textured regions and the 15 features described in the previous section for the textured regions. The feature extraction process is done using an analysis window with maximum overlapping. All the features described before are extracted using an analysis window W of size $N_W \times N_W$ (N_W odd), which is centered on the pixel to be analyzed.

The choice of the local mean feature is sufficient to characterize the pixels in the non-textured regions using a window of size 3×3 pixels. In the case of textured regions, we extract the 15 texture features using an analysis window of size 9×9. In this case the window size is larger to take into account all texture types (random, deterministic, coarse, and smooth) that could be found in the image. After this step, the calculated features are normalized and centered.

3.4 Conclusion

This module permits detecting the region types and characterizing appropriately the pixels in function of the region types in monocomponent image. We have developed a region detection method that uses the uniformity feature issued from the co-occurrence matrix in multi-resolution, with which the pixels of the image are classified into two classes (i.e. textured and non-textured) using FCM. The sizes of

the analysis windows are set automatically according to the identification of the global nature of region types present in the image to be partitioned. To characterize pixels belonging to the non-textured areas of the image we have used the local mean feature, whereas for pixels in the textured regions we have used a set of 15 texture descriptive features.

In the following chapter, we will investigate and assess their discriminating power.

Chapter 4 Unsupervised cooperative classification

4.1 Introduction

The choice of a classification method is still a challenging problem in the field of image partitioning. Despite the existence of a vast number of classification methods, each of them is only adapted to some specific type of images and applications. In order to be able to partition a wide range of image types, and for different applications, the choice of the features and of the classification methods is a crucial issue.

In this chapter we present the last three modules (unsupervised parallel classification, monocomponent evaluation and conflict management and merging results) of the proposed basic cooperative and adaptive partitioning system, in three sections and its extension to multicomponent image. In the fourth section we assess the developed system on two real applications.

4.2 Parallel unsupervised classification

In general, unsupervised classification methods do not require any training or any other *a priori* knowledge, while semi-supervised methods require the number of classes, or other knowledge to be defined in advance as an input parameter (e.g. number of classes, iteration number, and minimum number of pixels in a class). Furthermore, unfortunately some of these methods are influenced by the position of the initial class centers as was shown in Part I, Chapter 1.

To make the approach more robust, we use two different classification methods (AILBG and FCM) in cooperation. Before putting the results of these methods in parallel cooperation, we have optimized each of them. Both AILBG and FCM require

the number of classes to be defined at the beginning. In addition, the FCM is sensitive to the initial class centers and to the value of the fuzzification parameter *m*. To overcome these problems in the following section we propose the optimization of FCM and AILBG (FCMO and AILBGO). These two new unsupervised classification methods will be used in parallel to produce two partition results of the same image that will be fused in the next module.

4.2.1 Optimization of FCM and AILBG algorithms

We recall that a robust classification method should have the following properties:

- Automatic class number estimation.
- Insensitivity to the choice of the initial class centers.
- Unsupervised evaluation of the intermediate results (without ground truth knowledge).

To determine the best partition of a dataset *X*, the following important steps are required to optimize either FCM or AILBG (see Figure 4.1):

Step 1: *Choice of the class to be subdivided*: at the beginning (*K=1*), the class to be divided is the whole dataset *X*. When *K>1*, choose the most expanded class.

Step 2: *Choice of the initial class centers*: after choosing the class to be subdivided into two classes, we need to identify two initial class centers. The first class center will be the center of gravity of the chosen class, and the other class center is chosen randomly.

Step 3: *Classification with class center fine-tuning*: we classify the dataset using FCM or AILBG. To make the approach independent of the initial class centers a removal-insertion fine-tuning process is used.

Step 4: *Evaluate the obtained intermediate partitioning using an unsupervised criterion*: if this criterion is satisfied the partitioning into *K+1* classes is validated, then go to step 1. If the criterion is not satisfied go to step 1 and change the class to be subdivided, choosing the next most expanded. The algorithm stops in case none of the classes satisfy the

criterion, in other words if any class of the current partition is not divisible, the current number of class is considered as optimal.

In the following we give details of the above four steps of the algorithm.

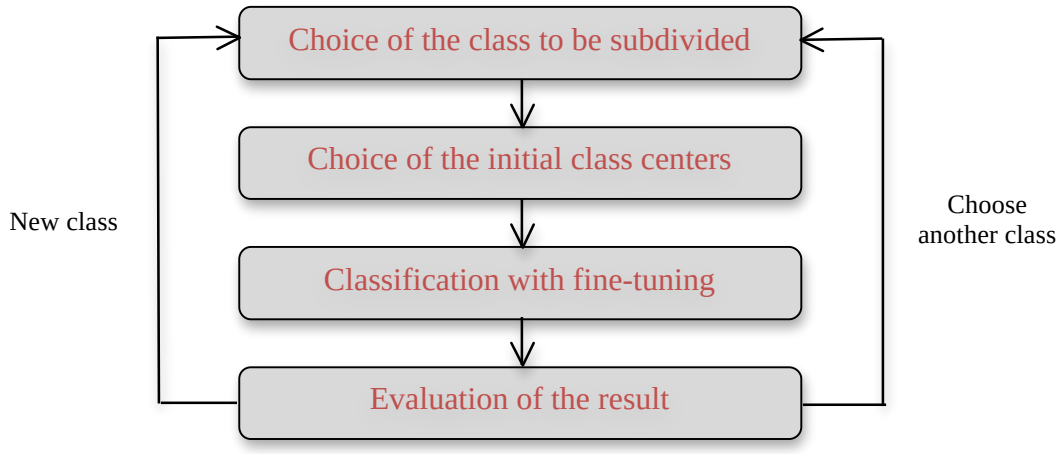


Figure 4.1: The general layout of the developed unsupervised classification approach

- ***Choice of the class to be subdivided***

We suppose that we have obtained so far K classes $C = \{C_1,..,C_K\}$, and we try to subdivide one of them to obtain $K+1$ classes. Intuitively the most expanded class will be the best candidate to be divided. So we calculate a dispersion measure for each class $C_i\ (i = 1,..,K)$ *as* follows:

$$Dispersion(C_i) = \frac{1}{NC_i}\sum_{j=1}^{NC_i} d(g(C_i), x_j^i) \qquad (4.1)$$

where NC_i is the number of elements in the class C_i with the center of gravity $g(C_i)$, $d(.,.)$ is the Euclidean distance, and x_j^i is the j^{th} element in the class C_i.

The dispersion values are calculated for each class $C_i\ (i = 1,..,K)$ and they are arranged in decreasing order. The most expanded class is chosen to be subdivided.

- ***Choice of the initial class centers***

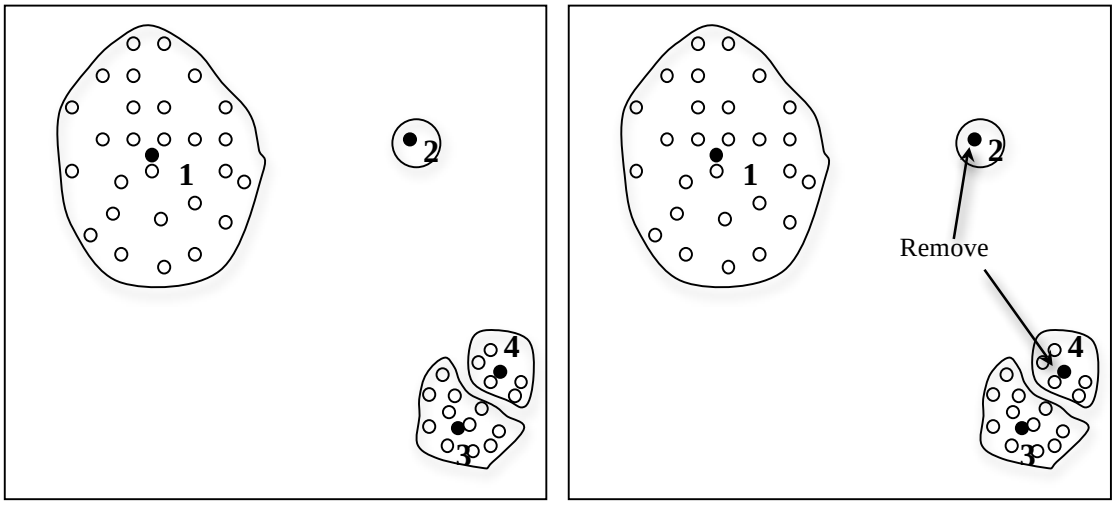
After choosing the class to be subdivided C_i, the initial class centers of the new class are determined. This class will be divided into two sub-classes. The most representative point in the class is its center of gravity $g(C_i)$, we keep this point as the first class center, and the center of the second class is chosen randomly (one of the data point in the class).

- ***Classification with class center fine-tuning***

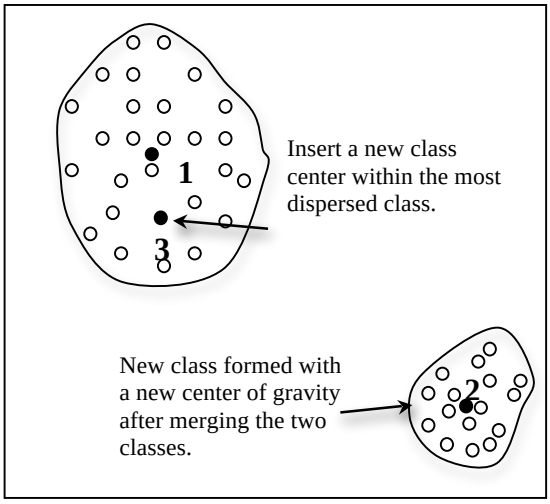
After choosing the initial class centers; the whole dataset is partitioned with *K+1* class centers. To make our method independent of the initial class centers, we have adopted a removal-insertion process proposed in [49] to fine-tune the class centers. This removal-insertion process is based on the assumption that the dispersion of each class will be mutually equal. According to this assumption, the empty classes and the class that has the lowest value of dispersion are removed. Then, a new class center is inserted within the class with the largest dispersion, and then a classification is applied on the dataset. This process is repeated until no decrease in the dispersion can be obtained. In the following we explain in details the removing and inserting criteria of this process:

- *Removing criterion*: if a class is empty it is removed directly, and then the class with the lowest dispersion (called the *loser* class) is removed also. The class adjacent to the *loser* class (called neighbor of *loser* class) and the *loser* class are merged and the center of gravity of this merged class (*loser* + neighbor of *loser*) is then recalculated. Figure 4.2 shows an example: the class center label with '2' is removed (empty class), and then the class center labeled with '4' (*loser* class) is also removed, and the individuals of this class are merged with class '3'.
- *Insertion criterion*: a new class center is inserted near the class with highest local dispersion (called *winner* class); this class is inserted by randomly choosing an individual from the *winner* class. Figure 4.2 (c) illustrates an example where a class center is inserted near class '1' (winner class).

- *Stopping condition*: when the removal-insertion process cannot generate a decrease in the dispersion, then the process stops.



(a) Classification result, four classes (b) Remove zero and lowest dispersion classes



(c) Insert a new class within the most expanded class

Figure 4.2: Class centers fine-tuning example

- ***Evaluation of partitioning (estimation of the optimal number of classes)***

This step validates or rejects the partitioning obtained with *K+1* classes. Here, it is verified that the partition obtained is coherent. In other words we here want to detect the invalid partitions obtained. The *K+1* classes partitioning is validated if:

$$ROS\left(I_R\right)^{(K+1)} - ROS\left(I_R\right)^{(K)} < \eta\, ROS\left(I_R\right)^{(K)} \qquad (4.2)$$

where $ROS\left(I_R\right)$ is the criterion defined in Part I, Chapter 2, Equation 2.15 and η is a low percentage value that guarantees the termination of the subdivision algorithm (in our experiments the value of η is set to 10^{-3}). Hence, the evaluation of the obtained partitioning is done by considering the coherency of the obtained classes.

After this evaluation process, two cases might be encountered:

- A valid partitioning is obtained, and then the algorithm is repeated, trying to subdivide the most expanded class among the ones created so far.
- If not, the algorithm is repeated, trying to subdivide the second most expanded class, and if none of the classes give valid subdivision, the current number of class *K* is considered as optimal.

4.2.2 Evaluation of FCMO and AILBGO algorithms

We have tested our proposed classification approaches FCMO and AILBGO regarding three aspects: firstly for its stability, secondly for the correct estimation of number of classes and lastly for its time complexity.

To check the stability of the two proposed algorithms (see Figure 4.1) we have executed each of them 100 times on 50 different synthetic datasets. The results were found to be 100% stable as they are identical from a run to another on the same dataset for FCMO (m=2, 4, 6, 8) (see Figure 4.3) and AILBGO. We have also tested the non optimized FCM, LBG and *k*-means for stability on the same datasets to compare their rates with our FCMO and AILBGO. The ACCRs obtained in function of the runs are: 87% for FCM (m=2), 94% for FCM (m=4), 74% for LBG and 59% for *k*-means stable.

In these experiments we observe that the FCMO gives better classification results when the fuzzification parameter m is set to 4 (see Figure 4.3).

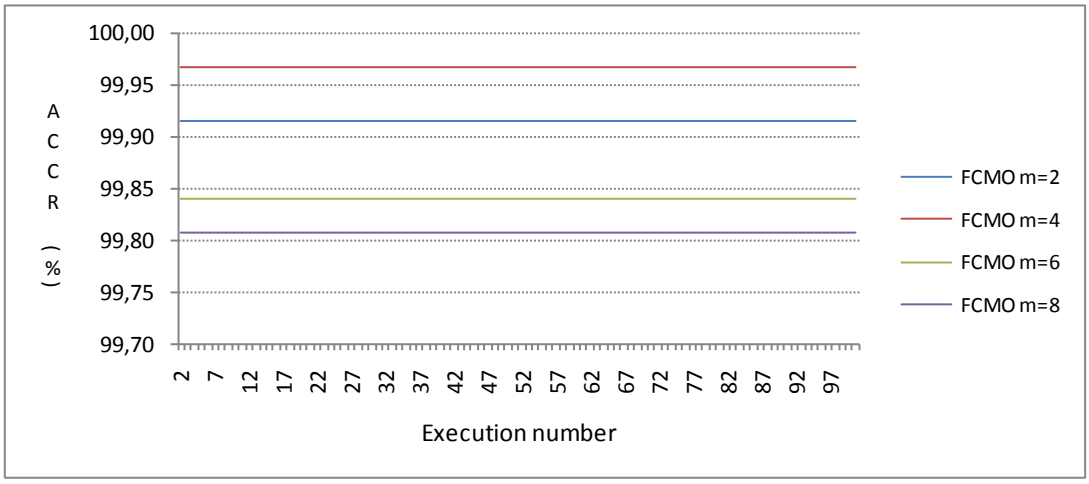


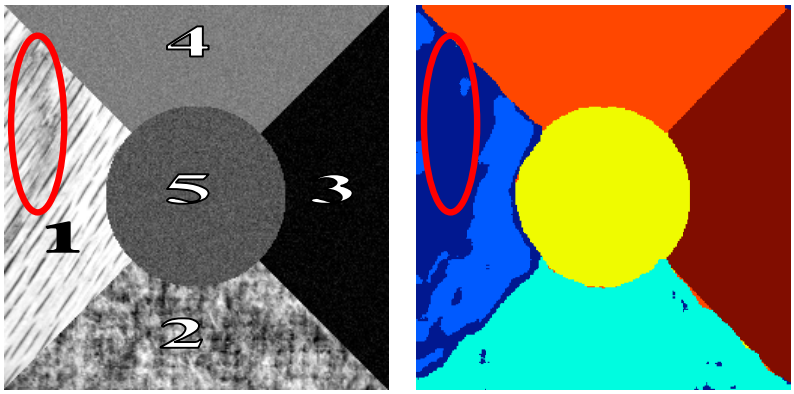
Figure 4.3: Assessments of accuracy and stability of FCMO in function of *m*

Concerning the correct estimation of the number of classes, we have tested FCMO (with *m*=4) and AILBGO on the image database described previously in Chapter 3, which contains 100 synthetic images. The average correct class number estimation over the tested image set is 90%. This rate is coherent because in some images in the defined database there are high fluctuations within same classes so that these latter is detected as more than one single class. For example, the class labeled as "1" in the image presented in Figure 4.4(a) is composed of wood, where a part of this class shows defects. It is clear from visual inspection that the area inside the highlighted red oval (wood defect) is not the same as the rest of the class and it is divided into two subclasses, which is actually true. Figure 4.5 shows an example of the evolution of the criterion described in Part I, Chapter 2 (Equation 2.15) used in Equation 4.2 to estimate the number of classes for the image in the Figure 4.4(a). We can observe that the maximum of this criterion gives three classes for textured regions and three classes for non-textured region. The number of classes estimated $\hat{NC}$ is 6.

We can point out that if this kind of information was accounted for in the ground truth of the tested image set, the correct class estimation rate would be greater than 90%.

For the time complexity aspect we have compared FCMO and AILBGO with MLBG by running them on the same datasets and inspecting the time elapsed to

accomplish the run. After the tests we have found that the optimized algorithms are more than three times faster than MLBG algorithm.



(a) Original image (b) Classification result ($\hat{NC} = 6$)

Figure 4.4: Example of estimation of the number of classes using FCMO (m=4)

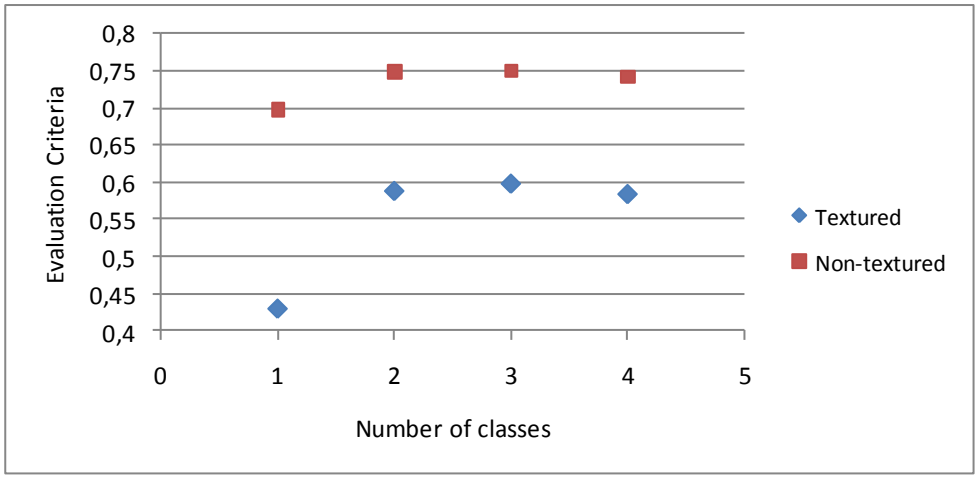


Figure 4.5: Evolution of the evaluation criteria $ROS(I_R)$ on synthetic image of Figure 4.4(a)

4.3 Conflict management by fusion and result merging

A process of conflict resolution or fusion is applied to improve the accuracy and get more reliable results by combining multiple results of the same image, or combining the results of different components in the case of multicomponent images.

We herein propose an approach for conflict resolution and fusion that is based on genetic algorithms (GA). A genetic algorithm solves problems of operational research, which cannot be solved by classic techniques. In the genetic algorithm

which is used hereafter, an unsupervised evaluation criterion is defined as the objective function. The unsupervised evaluation criterion used estimates the quality of the results at a global level without any *a prior* knowledge of the ground truth, and is able to adapt itself to the type of regions in the image [10], [121].

In this section we describe in details our proposed method for conflict resolution, between the partitioning results of the same image using different methods and also the fusion of different components results of a multicomponent image.

4.3.1 Monocomponent image case

To validate, and fuse the different classification results obtained by the application of each of the methods (FCMO and AILBGO) according to the diagram of Figure 4.1 and getting the best partition possible, a two-level evaluation process is applied [122], [123]. First, the pixels which are assigned to the same class by both methods are considered directly as valid pixels, and reported to the final partitioning result. Besides, the pixels that are classified differently by the two methods are considered as invalidated, and are subject to a second evaluation process using an objective function optimized by a GA. This two-level validation step reduces considerably the processing complexity.

The objective function used in the GA is the criterion of Rosenberger and Chehdi. presented in [10], [116], [124]. Using this criterion provides some advantages: *i)* it is unsupervised, i.e. no *a prior* knowledge is required [125]–[127]; *ii)* it adapts itself automatically to the nature of the regions (textured, non-textured) and works well in both cases [128], and *iii)* finally it controls efficiently the issue of under and over classification [121].

In our system, the implementation of the genetic algorithm presented in Part I, Chapter 1, Section 1.2.2 is performed as follows for the fusion process:

Step 1: Define the chromosomes of the initial population as the invalidated classification results (a chromosome for each result) and then

calculate the fitness of each chromosome using the above described criterion.

Step 2: Select chromosomes from the current population for reproduction using fitness proportionate selection [129], and mutate them by using single point mutation.

Step 3: Apply crossover operation on the selected chromosomes using uniform cross-over [130].

Step 4: Evaluate the chromosomes in the population.

Step 5: Stop if no better chromosomes are created, else go to step 2.

The selection operation used in this process allows weaker solutions to survive the selection process. Besides, the type of cross-over operation used enables the parent chromosomes to exchange at the gene level rather than at the segment level, and this allows better combination between the chromosomes.

At termination of the GA[5], the best-evaluated chromosome in the population is considered as the final result for the conflicting pixels. Eventually, these pixels are grouped with the valid pixels from the first level.

After the conflict management by fusion we unify the results of textured and non-textured regions by reporting them into the same labeled image.

To prove the reliability of the three modules, we have assessed them on the image database previously described in Chapter 3. In this experiment and the other experiments in this thesis, the fuzzification factor m is set to 4 [35] for the FCMO classifier, the tolerance threshold factor ε is set 10^{-10} for the GA, FCMO and AILBGO, the swipe probability of the uniform crossover operation in the GA is set to 0.5 [130].

Figure 4.6 and Table 4.1 show a sample image and the obtained results including the detected region natures, the results of AILBGO and FCMO, the result of our cooperative approach, and the corresponding average correct classification rates (ACCR). In this example, the result given by the AILBGO method only mixes up two

[5] The GA used is the one provided by Matlab™, in release 7.11.0.

regions of the image, yielding a low correct classification rate, while the result obtained by FCMO is clearly more robust. The number of classes estimated ($\hat{N}_C$) for each method is 5. Another remark is that some pixels from the AILBGO method result are classified in the correct class, which is not true with the FCMO method. The application of the cooperative approach has kept the pixels correctly classified and reassigns those which were not previously correctly classified. The final number of classes estimated ($\hat{N}_C$) after cooperation process is 5.

The global average correct classification rate (GACCR), for the set of 100 test images (described previously in Chapter 3) is: 94.71% for FCMO method, 88.31% for AILBGO method and 97.19% after fusion of both.

In order to assess the importance of the region nature detection and adapting the features extraction step, we have also tested our system without region nature detection step on the same set of synthetic images (see Figure 4.7). Figure 4.8 summarizes the ACCRs for the developed approach with and without the region nature detection. This confirms that adapting the feature extraction in function of the region types improves the quality of the classification results.

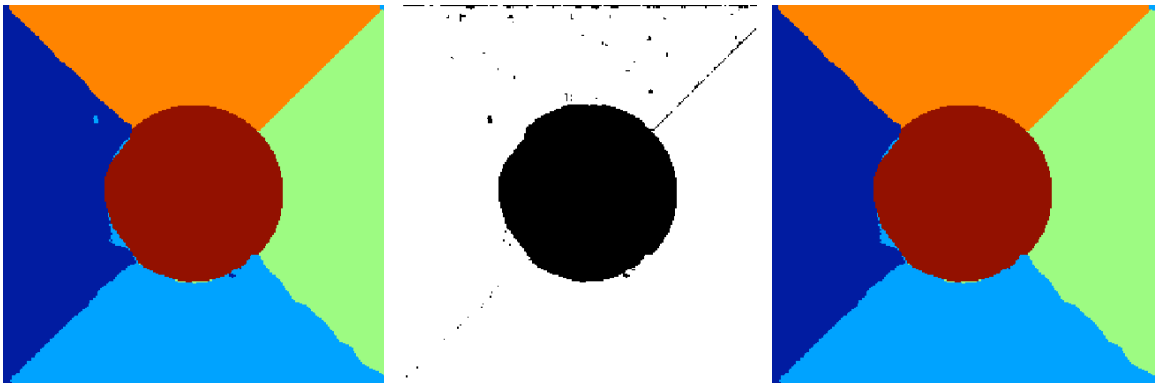
In addition we have compared the developed cooperative system with the one described in Section 1.3.2 (parallel cooperation) that cooperates SVM and ISODATA algorithms[6] [77]. Since SVM requires training data, 10% of the ground truth pixels were used to train it. The optimal parameter regularization for the SVM classifier was chosen by fivefold cross validation and the kernel function used is the Gaussian RBF. The parameters set for the ISODATA algorithm are: minimum and maximum numbers of classes, minimum number of pixels in a class, and change threshold. Their values are respectively: 4, 10, 2, and 5%. Figure 4.9 and Figure 4.10 show that the obtained results with our system are significantly improved (ACCR: +4.27%). Moreover the Figure 4.11 shows that the choice of the methods in a cooperative process is important to contribute in getting reliable results.

[6] The SVM and ISODATA algorithms used are the ones provided in the Envi™ software.

(a) Original image

(b) Region nature detection (white: textured, black: non-textured)

(c) AILBGO result ($\hat{NC}=5$, ACCR: 78.20%)



(d) FCMO result ($\hat{NC}=5$, ACCR: 98.58%)

(e) Validation map (white: valid, black: invalid)

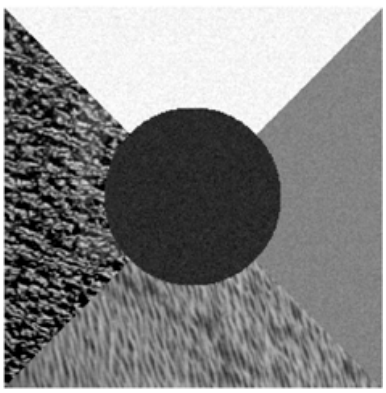
(f) System result ($\hat{NC}=5$, ACCR: 99.00%)

Figure 4.6: Classification results of a synthetic monocomponent image by the developed system

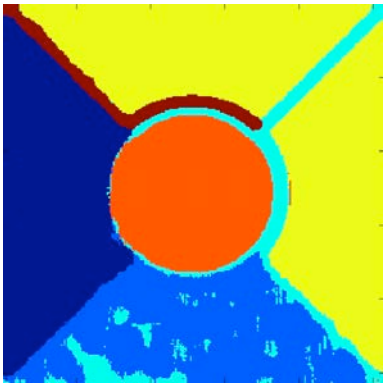
Table 4.1: Confusion matrix of classification result of a synthetic monocomponent image using the proposed cooperative approach (Figure 4.6 (f))

(CCR in %, number of pixels (.))

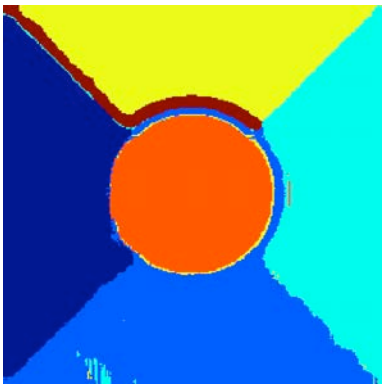
	Ground truth classes (number of pixels)				
Classes discriminated automatically by our approach	Class 1 *(13437)* (Textured)	Class 2 *(13607)* (Textured)	Class 3 *(13607)* *(Non-textured)*	Class 4 *(13608)* *(Non-textured)*	Class 5 *(11277)* *(Non-textured)*
Class 1	**99.73%** (13401)	1.20% (163)	0	1.08% (147)	*0.31% (35)*
Class 2	0.27% 36)	**98.33%** (13380)	0.26% (36)	0.11% (15)	*1.25% (142)*
Class 3	0	0.47% (64)	**99.74%** (13571)	0.03% (4)	*0*
Class 4	0	0	0	**98.75%** (13438)	*0*
Class 5	0	0	0	0.03% (4)	***98.44%*** ***(11100)***



(a) Original image

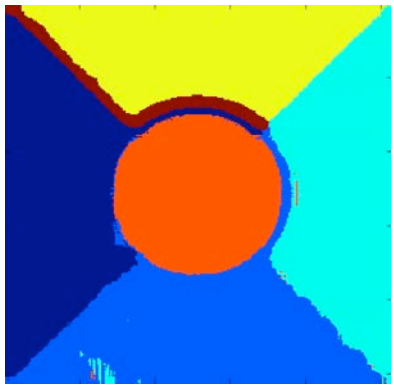


(b) AILBGO result
($\hat{NC}=6$, ACCR: 88.14%)



(c) FCMO result
($\hat{NC}=6$, ACCR: 92.69%)

(d) Validation map (white: valid, black: invalid)



(e) Fusion result
($\hat{NC}=6$, ACCR: 93.44%)

Figure 4.7: Classification results of a synthetic monocomponent image by the developed system (without region nature detection)

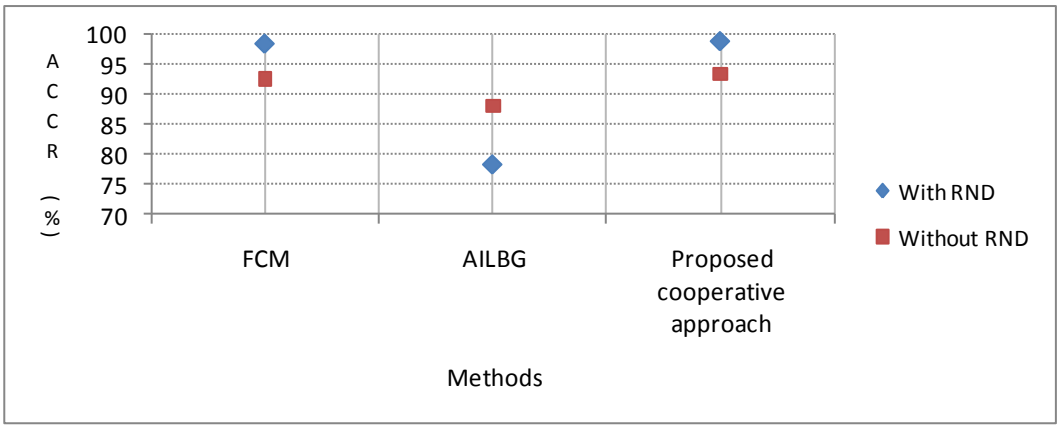


Figure 4.8: ACCR of the proposed approach with and without region nature detection

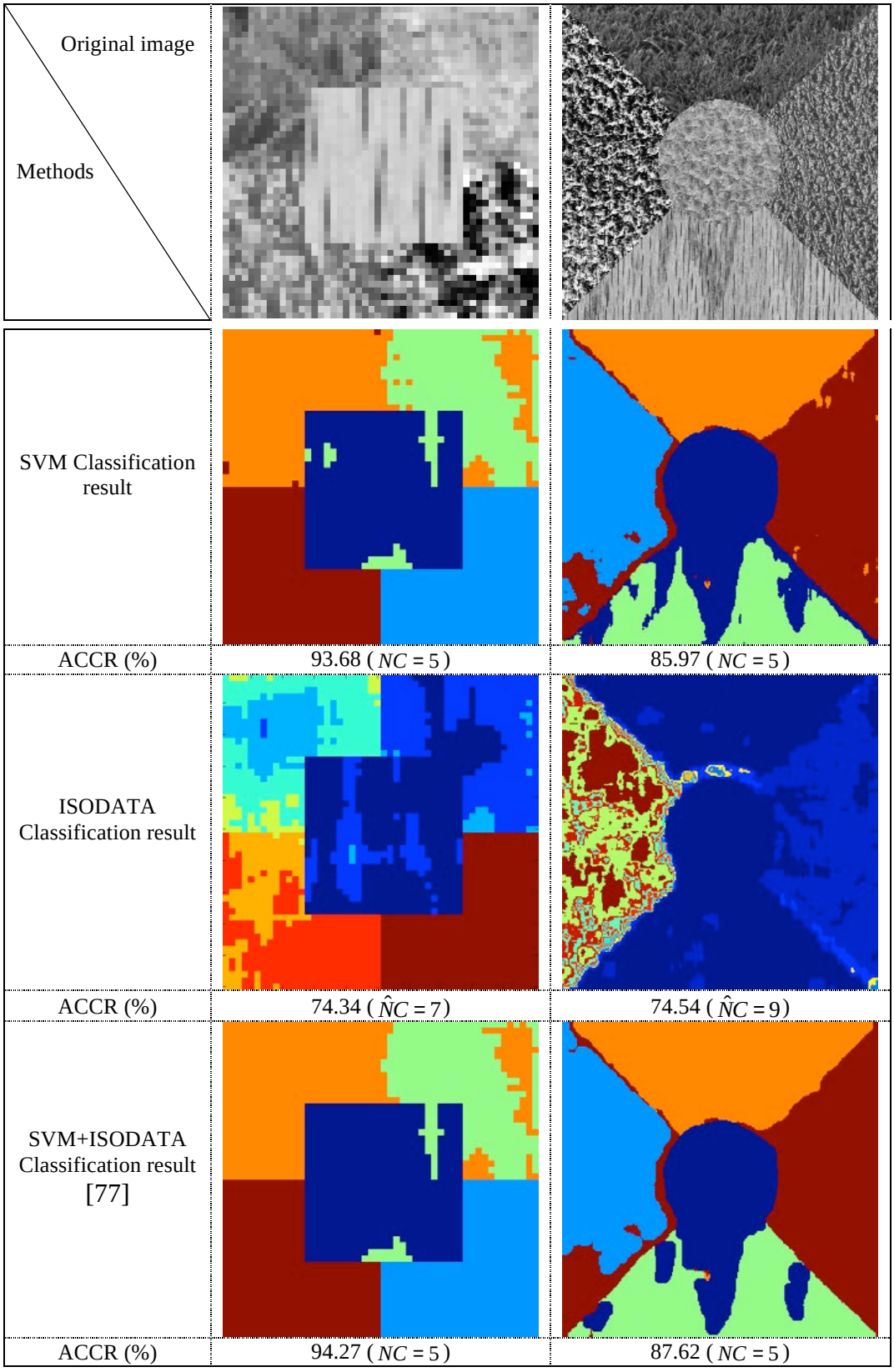


Figure 4.9: Partitioning results of the system that uses SVM and ISODATA [77]

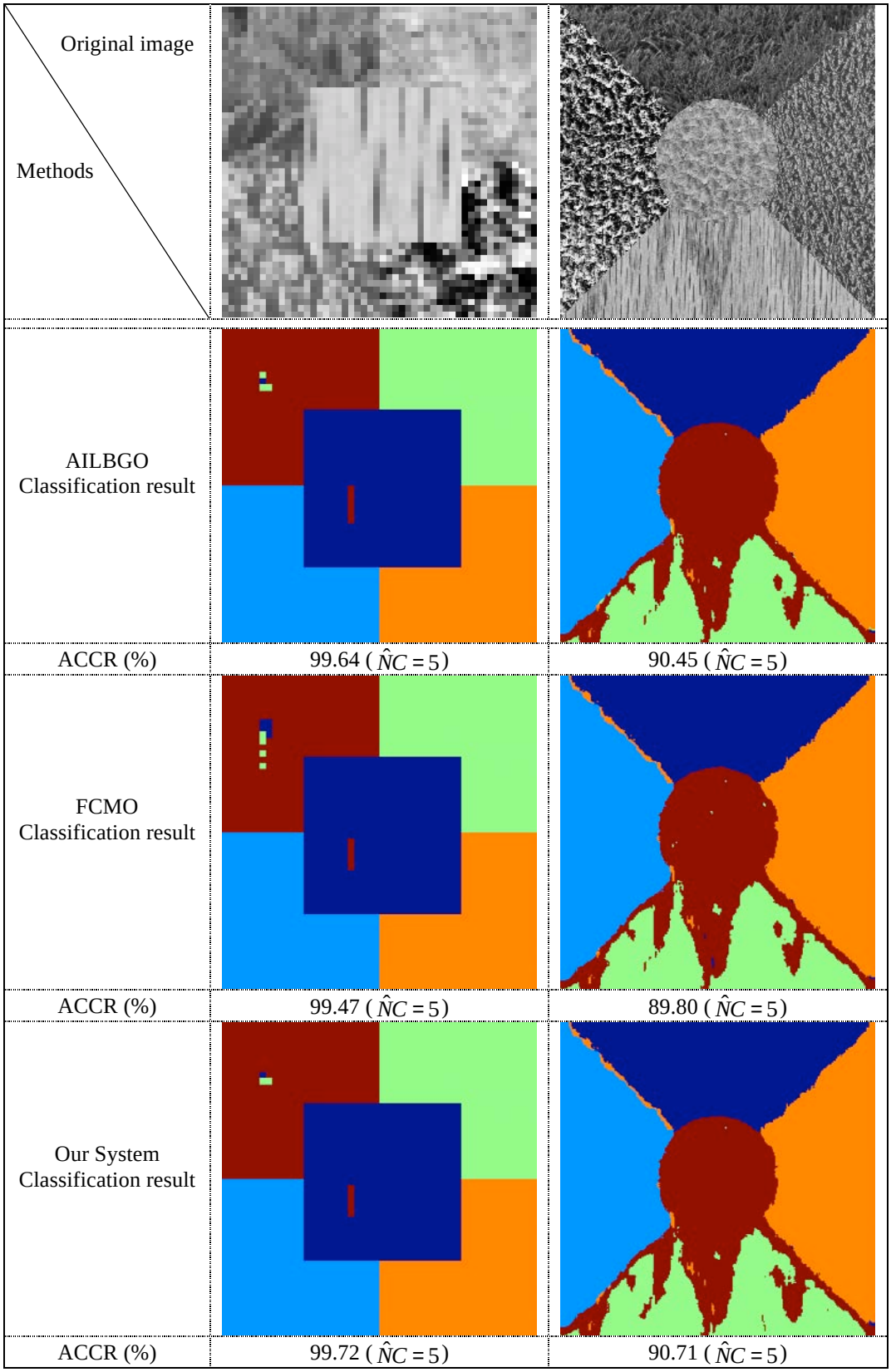


Figure 4.10: Partitioning results of the developed cooperative system

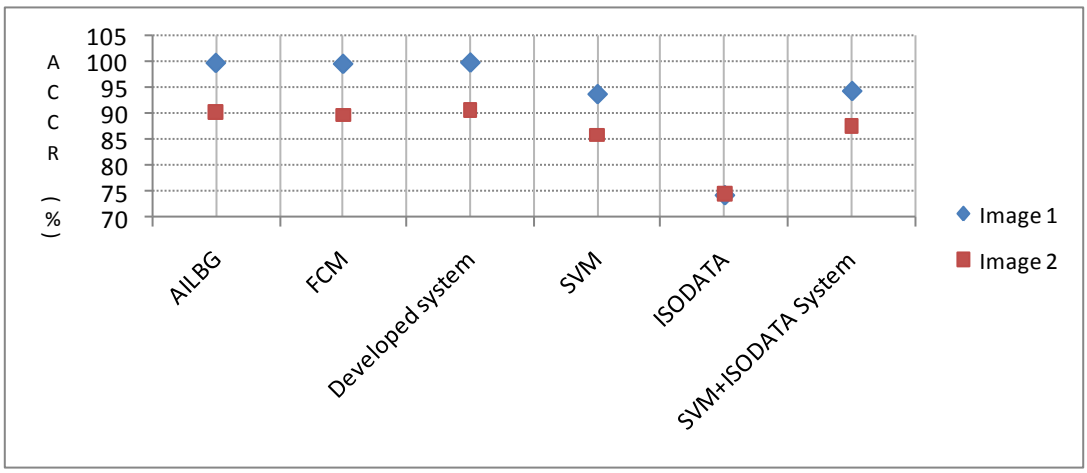


Figure 4.11: Performance of the compared approaches and the algorithms used in them (ACCR)

We have also tested our approach on real monocomponent images. Figure 4.12 shows an example. The values of the evaluation criterion Equation 2.15 are given to assess the results obtained because the image used does not have any associated ground truth data (see Figure 4.13).

4.3.2 Multicomponent image case

In this case, the results from the different components are evaluated and fused to get the final classification result. To do this, we propose a method [122], [123] in which the results of different components are compared, and adjacent components with high similar classification results are grouped within the same subset. The contents of each subset are fused independently. At the beginning the first component result is taken as reference and compared with the adjacent components result. The reference component is changed if the number of identical pixels decreases. For example, if the first component result is compared with the second component result and some percentage of the pixels where found to be classified identically, then the first component result is compared with the third component result, if the percentage of the identical classified pixels are greater than this percentage, the reference component remains unchanged and compared with a further component result; if not, the first and second component results are considered as one subset and the third component result becomes the reference component, and the same procedure is repeated until the last component is processed.

The component results in each subset are fused separately, and then the results of the subsets are fused to get the final partitioning result of the multicomponent image. GA is used in the fusion process where the objective function is the same as in the monocomponent case, but the fitness function is modified in order to evaluate a classification result by taking into account each concerned component. This parameter equals the average of the evaluation criteria calculated for the concerned components.

(a) Original image

(b) Region nature detection (white: textured, black: non-textured)

(c) AILBGO result ($\hat{NC} = 9$)
Ros (I_R) criterion for non-textured regions: 0.6220
Ros (I_R) criterion for textured regions: 0.8874

(d) FCMO result ($\hat{NC} = 9$)
Ros (I_R) criterion for non-textured regions: 0.6262
Ros (I_R) criterion for textured regions: 0.8874

(e) Validation map (white: valid, black: invalid)

(f) proposed approach result ($\hat{NC} = 9$)
Ros (I_R) criterion for non-textured regions: 0.6351
Ros (I_R) criterion for textured regions: 0.8874

Figure 4.12: Classification results of a real monocomponent image by the developed cooperative system

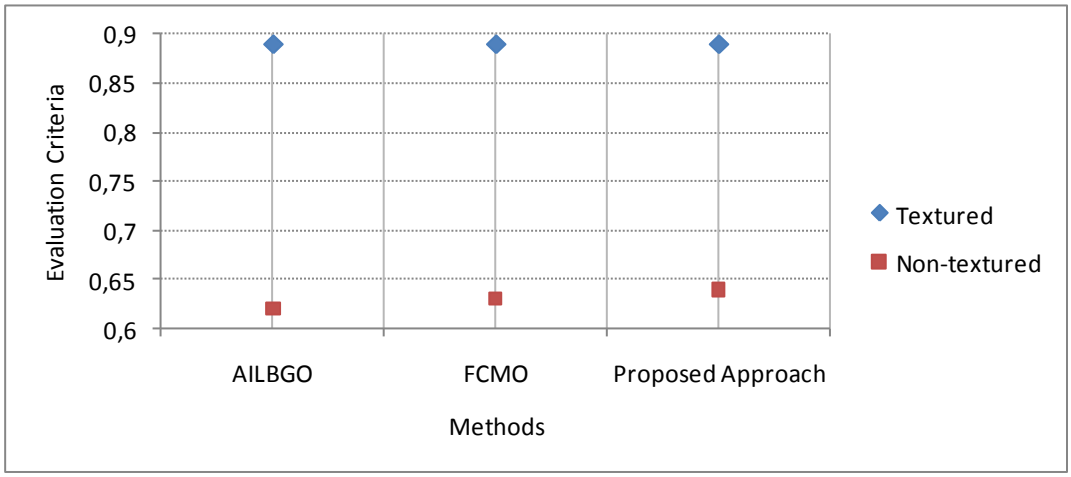


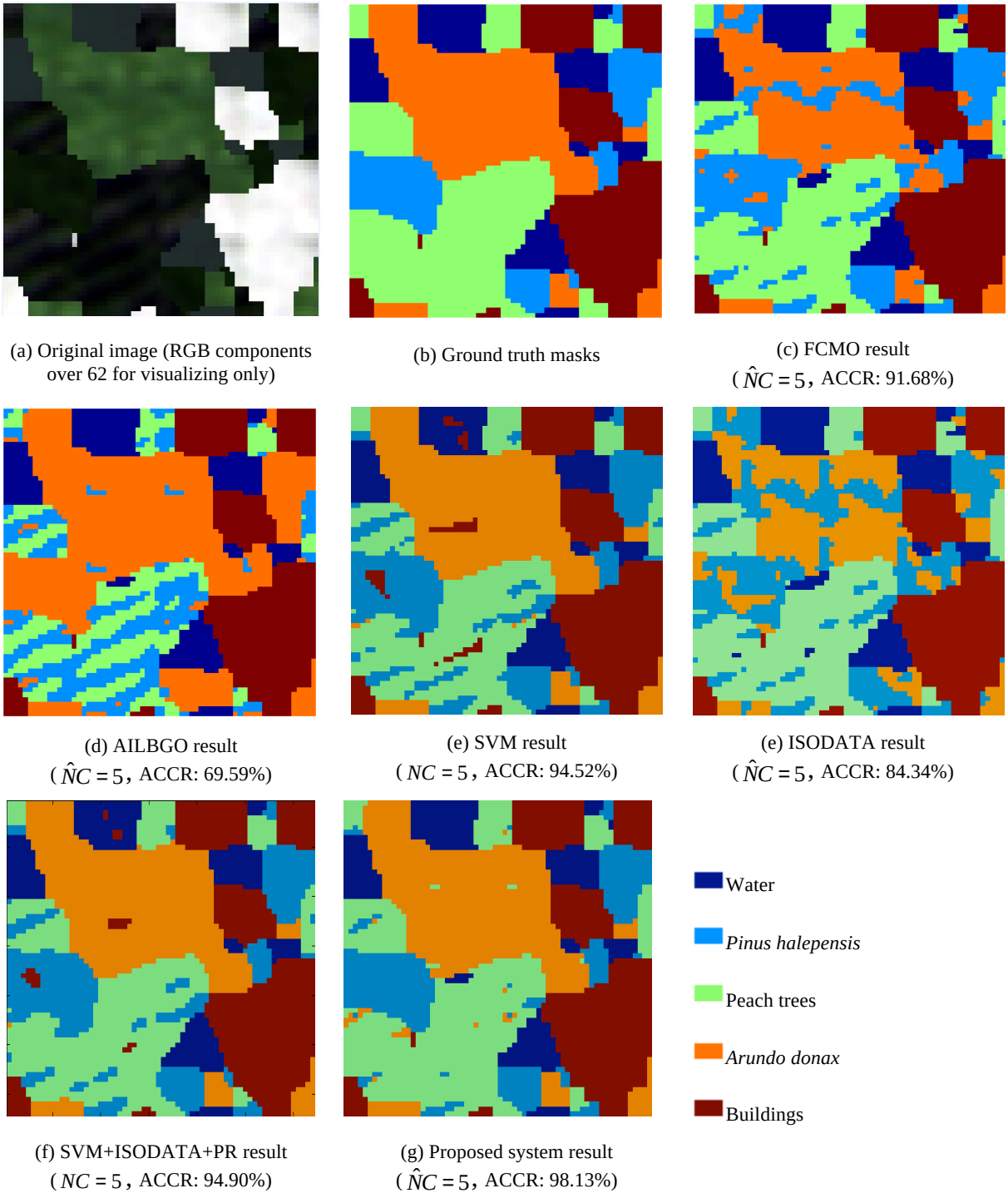
Figure 4.13: Evolution of the evaluation criteria *ROS(I_R)* on a monocomponent real image

To validate this approach of evaluation and fusion, it is applied on a synthetic hyperspectral image which is constructed from the ground truth regions of a real hyperspectral image collected by AISA Eagle sensor of our laboratory. This image is collected on October 1st 2010, over the region of Cieza in southeastern Spain. It is acquired at 0.5 meter spatial resolution in 62 spectral bands within the range [400, 970] nm. The data used to construct the test image are taken randomly from 5 different regions of the original image. The 5 land covers are: Water, *Pinus halepensis*, Peach trees, *Arundo donax*, and Buildings.

To assess the proposed system on this test image, we compare its result with each of the four non-cooperative methods (FCMO, AILBGO, ISODATA, SVM) and with the cooperative approach, which uses SVM and ISODATA [77]. Since SVM requires training data, 400 pixels over 4096 pixels of the ground truth were used to train it. The optimal parameter regularization for the SVM classifier was chosen by fivefold cross validation: $C = 100$, $\gamma = 0.16$ and the kernel function used is the Gaussian RBF. The parameters set for the ISODATA algorithm are: minimum and maximum numbers of classes, minimum number of pixels in a class, and change threshold. Their values are respectively: 4, 10, 2, and 5%.

The results for this test are shown in Figure 4.14. The average correct classification rates for all tested methods are: 91.68% for FCMO, 69.59% for AILBGO, 84.34% for ISODATA, 94.52% for SVM, 94.60% for SVM+ISODATA, and 98.13% for the proposed approach. Table 4.2 gives the confusion matrix for the

proposed approach and Table 4.3 provides details of the per-class correct classification rates for each method tested.



(a) Original image (RGB components over 62 for visualizing only)

(b) Ground truth masks

(c) FCMO result ($\hat{NC}=5$, ACCR: 91.68%)

(d) AILBGO result ($\hat{NC}=5$, ACCR: 69.59%)

(e) SVM result ($NC=5$, ACCR: 94.52%)

(e) ISODATA result ($\hat{NC}=5$, ACCR: 84.34%)

(f) SVM+ISODATA+PR result ($NC=5$, ACCR: 94.90%)

(g) Proposed system result ($\hat{NC}=5$, ACCR: 98.13%)

Figure 4.14: Comparison of synthetic hyperspectral image classification results between five methods and the proposed cooperative system

Table 4.2: Confusion matrix of classification result of the synthetic hyperspectral image using the proposed cooperative system
(CCR correct classification rate in %, number of pixels (.))

	Ground truth classes (number of pixels)				
Classes discriminated by our approach	Water *(452)*	*Pinus halepensis (500)*	*Peach trees (1189)*	*Arundo donax (1068)*	Buildings *(887)*
Water	**100%** (452)	0	0.93% (11)	0	0
Pinus halepensis	0	**100%** (500)	5.55% (66)	0	0
Peach trees	0	0	**91.50%** (1088)	0.85% (9)	0
Arundo donax	0	0	2.02% (24)	**99.15%** (1059)	0
Buildings	0	0	0	0	**100%** (887)

Table 4.3: Comparison of classification results of six methods on the synthetic hyperspectral image (CCRs and ACCRs in %)

	FCMO	AILBGO	ISODATA	SVM	SVM+ISODATA	Our cooperative approach
Water	**100%**	**100%**	**100%**	97.35%	96.90%	**100.00%**
Pinus halepensis	91.80%	0.12%	73.60%	97.00%	96.80%	**100.00%**
Peach trees	83.68%	48.6%	83.59%	79.73%	83.10%	**91.51%**
Arundo donax	82.96%	98.13%	64.51%	98.50%	98.69%	**99.16%**
Buildings	**100%**	**100%**	**100%**	**100%**	98.99%	**100.00%**
ACCR	91.69%	69.37%	84.34%	94.52%	94.90%	**98.13%**

4.4 Assessment on real applications

Our classification approach was also evaluated on two real applications. More precisely we have used a hyperspectral image for identification of invasive and non-invasive vegetation in the region of Cieza (Spain) as well as a multispectral image for the detection of Pine trees. The ground truths data were provided with these images. These data were collected in the framework of two projects, the first one with Infrastructure and Ecology SA, INFRAECO, Chile, and the second with the Lebanese National Remote Sensing Center.

An important key point to mention is that the proposed approach does neither require any training nor any other *a priori* knowledge about the data to be partitioned. The ground truths provided with the image data are only used for evaluating the obtained results by which the Average Correct Classification Rate (ACCR) is calculated.

4.4.1 Detection of invasive and non-invasive vegetation

Detection of invasive plants such as *Phragmites australis,* Arundo *donax* and *Tamarix,* at an early stage has a great interest in environmental and economical aspects. The goal of this early detection is to undertake appropriate management actions to limit their development in a given area.

For this application, the ground truth of the tested image described in Section 4.3.2 includes six different invasive and non-invasive vegetation classes, which consist of 9207 pixels, namely: *Phragmites australis*, *Arundo donax*, *Tamarix*, *Ulmus minor*, *Pinus halepensis,* and Peach trees. The spatial size of this image is 1000 lines by 1000 columns. This image is composed of 62 spectral bands.

To assess our unsupervised cooperative system, the correct classification rates are calculated using available ground truth areas.

In this section, the result of proposed system is compared to the results of five non-cooperative (SVM, SVM with post relaxation, ISODATA, AILBG, FCM) and two cooperative approaches (SVM+ISODATA, SVM+ISODATA with post relaxation). We give also the results of the proposed system by using only FCMO or AILBGO algorithm in the classification step. The optimal parameter regularization for the SVM classifier was chosen by fivefold cross validation: $C = 100$, $\gamma = 0.1$ and the kernel function used is the Gaussian RBF. The number of pixels used to train SVM is 3433 pixels over 9207 pixels of the ground truth.

We recall that 15 features are used for the textured regions, and the local mean for the non-textured regions.

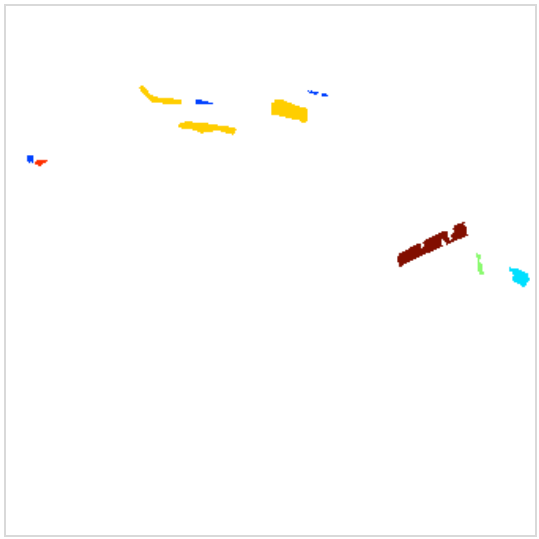
Figure 4.15 and Table 4.4 respectively show the result of our classification approach and the corresponding confusion matrix. This result shows that the proposed system provides a very good classification result with an ACCR 99.13%, for estimated number of classes $\hat{NC} = 6$.

Table 4.5 summarizes the results obtained by the different methods. We can observe from these results that our cooperative approach outperforms all the other tested methods. We can also observe that the paradigm of the proposed approach using only one classification method (i.e. without cooperation) gives similar results (-0.43% for AILBGO and -0.58% for FCMO), and that the results given by SVM with post relaxation (-3.1%), SVM+ISODATA with post relaxation (-5.46%), SVM (-5.93%), SVM+ISODATA (-8.02%) and ISODATA (-76%) are all lower than our approach paradigm with and without cooperation.

We also state that the results of the proposed approach scheme using either FCMO or AILBGO are very high compared to the results of FCM (-52.19%) and AILBG (-53.74%) when they are used directly on the hyperspectral image.

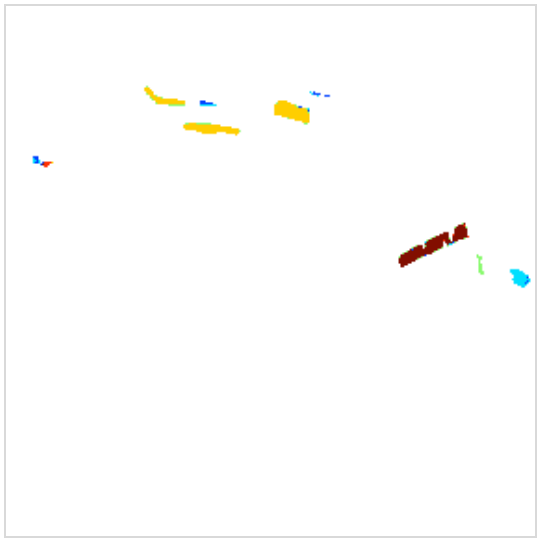
Overall, the present experimental study shows that analyzing a multicomponent/ hyperspectral image by our cooperative system outperforms all the other compared methods SVM, ISODATA, FCM, AILBG, SVM+ISODATA and SVM+ISODATA with post relaxation.

(a) Original Image (3 components over 62 for visualizing only)



(b) Ground truth masks

(c) Pixels of ground truth to classify



(d) Our cooperative approach classification result ($\hat{NC} = 6$, ACCR: 99.13%)

Phragmites australis
Arundo donax
Tamarix
Ulmus minor
Pinus halepensis
Peach trees

Figure 4.15: Detection of invasive and non-invasive vegetation results of hyperspectral image using cooperative approach.

Table 4.4: Confusion matrix of classification result using the proposed approach for detection of invasive and non-invasive vegetation
(CCR in %, (.): number of pixels)

	Ground truth classes (number of pixels)					
Classes discriminated by our approach	*Phragmites australis (544)*	*Arundo donax (4200)*	*Tamarix (162)*	*Ulmus minor (764)*	*Pinus halepensis (274)*	*Peach trees (3115)*
Phragmites australis	**99.08%** (539)	0.16% (7)	2.47% (4)	0.52% (4)	0.73% (2)	0.33% (10)
Arundo donax	0	**99.84%** (4193)	0	0	0	0
Tamarix	0	0	**97.53%** (158)	0	0	0
Ulmus minor	0,55% (3)	0	0	**99.48%** (760)	0	0.07% (2)
Pinus halepensis	0.37% (2)	0	0	0	**99.27%** (272)	0
Peach trees	0	0	0	0	0	**99.6%** (3103)

Table 4.5: Comparison of classification results of five non-cooperative and five cooperative approaches on the Cieza hyperspectral image
(CCRs and ACCRs in %)

	SVM	SVM+ PR[7]	ISODATA	SVM+ ISODATA	SVM+ ISODATA +PR[8]	AILBG	FCM	AILBGO by component	FCMO by component	Our cooperative system
Phragmentes australis	94.96%	**99.64%**	26.07%	98.56%	**99.64%**	62.41%	64.02%	98.95%	98.75%	99.10%
Arundo donax	97.67%	98.67%	31.24%	94.23%	97.77%	42.90%	45.13%	98.86%	98.66%	**99.83%**
Tamarix	82.71%	88.88%	4.93%	83.33%	87.65%	12.96%	14.19%	97.39%	97.43%	**97.53%**
Ulmus minor	97.23%	99.37%	29.18%	89.55%	90.94%	46.79%	47.80%	99.43%	99.15%	**99.49%**
Pinus helepensis	95.98%	97.81%	24.08%	91.24%	95.25%	52.18%	56.20%	98.97%	98.92%	**99.27%**
Peach trees	90.65%	91.81%	23.24%	89.75%	90.78%	55.12%	54.28%	98.58%	98.36%	**99.61%**
ACCR	93.20%	96.03%	23.13%	91.11%	93.67%	45.39%	46.94%	98.70%	98.55%	**99.14%**

[7] The process of Post Relaxation (PR) is programmed in Matlab™ by our laboratory.
[8] The system is programmed in our laboratory.

4.4.2 Detection of Pine trees

In the framework of this application, we seek to determine the discriminating power of multispectral images in the estimation of the land covered by Pine trees.

The multispectral image was acquired by the Earth observation satellite Ikonos on July 11, 2005, in the region of Baabdat (Lebanon) and it is used to detect the presence of Pine trees. The ground pixel size of this three bands (RGB) image is 0.8m. These data are provided by the Lebanese National Remote Sensing Center.

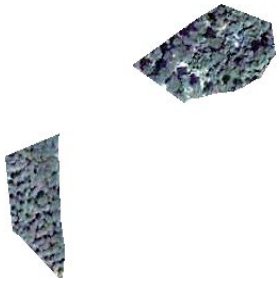
Here, the proposed approach is assessed by using available ground truth areas, in order to calculate the correct classification rate. The results of Pine trees detection are presented in Figure 4.16. This figure shows the detection result for each component and the final result issued from their fusion. The ground truth of the image contains 11736 pixels labeled as Pine trees and 11410 of these pixels are correctly detected as Pine trees, yielding 97.22% of correct detection. In this test the number of classes detected is two within the pixels of the ground truth. Using only FCMO or AILBGO in the paradigm of the proposed approach gives 96.52% and 96.73% of correct detection rates respectively.

4.5 Conclusion

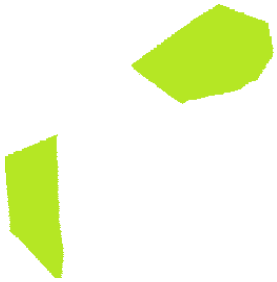
In this chapter, an original paradigm is proposed which makes use of the FCMO and AILBGO algorithms in cooperation. For each algorithm, the number of classes is estimated making them unsupervised. To fuse the results obtained by these two unsupervised proposed algorithms of the same component, an original two-level technique of validation is used. The pixels which are assigned to the same class by both methods are considered directly as valid pixels, while the pixels that are classified to different classes are subject to a second evaluation using GA. In case of multicomponent images; each component is partitioned independently and then these results are fused again using GA to get the final partition.

The experimental results show the efficiency of the proposed cooperative approach on different types of images, outperforming all the other non-cooperative and cooperative approaches tested. We state that the results of FCMO or AILBGO used in the proposed paradigm are 50% better compared to the results of direct application of FCM or AILBG on the multi/hyperspectral images.

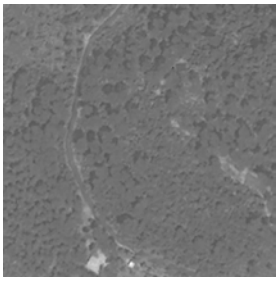
(a): Original Image RGB



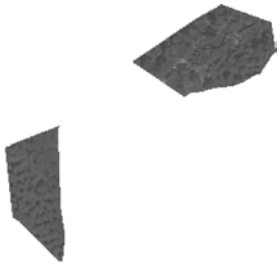
(b) Pixels of Ground Truth to classify RGB



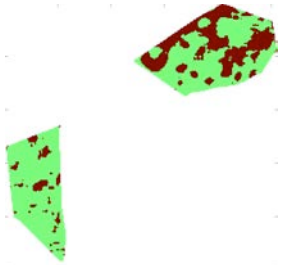
(c): Ground Truth mask of Pine trees



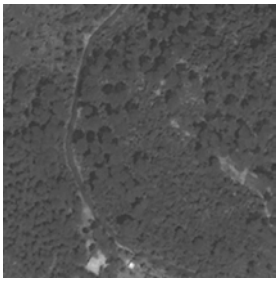
(d) R component



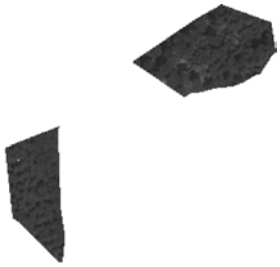
(e) Pixels of Ground Truth to classify R



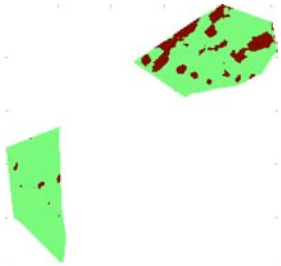
(f) Partitioning results of R component by fusion results of FCMO and AILBGO



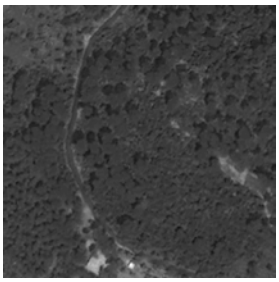
(g) G component



(h) Pixels of Ground Truth to classify G

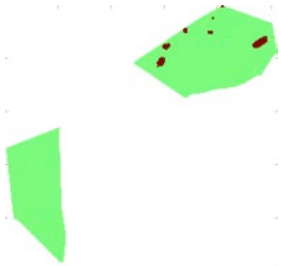


(i) Partitioning results of G component by fusion results of FCMO and AILBGO

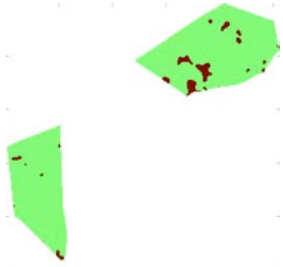


(j) B component

(k) Pixels of Ground Truth to classify B



(l) Partitioning results of B component by fusion results of FCMO and AILBGO



(m): System classification result (fusion of RGB components results): (CCR: 97.22%)
Pine trees, Other

Figure 4.16: Detection results of Pine trees from a multicomponent image

Conclusion

In the framework of this thesis, we have developed an unsupervised and adaptive partitioning system for hyperspectral images. The originality of the developed system relies *i)* on the adaptive nature of the feature extraction, since features of pixels are extracted taking into account different region types (i.e. textured and non-textured), and the fact that the region types which are detected automatically are treated independently from the feature extraction step to the final results, *ii)* on the introduction of several levels of evaluation and validation of intermediate partitioning results before obtaining the final result, *iii)* on the fact that it requires neither training samples nor the number of classes.

Each step of the proposed system is optimized to obtain the best possible intermediate and final results:

- To detect the regions type, we have proposed a new method to partition the image into two types of regions (i.e. textured and non-textured). After detecting the region type, we use the local mean to characterize the pixels detected in non-textured regions, and a set of 15 texture features for the ones in textured regions.
- The features extracted are used to classify the pixels by using FCMO and AILBGO algorithms. We have improved these two classifiers on two aspects: automatic class number estimation for both and insensitivity to the initial class centers for FCM.
- The results obtained from the two classifiers are fused to get one reliable result. We have used a GA for the process of conflict resolution or fusion. GAs are in general time consuming, but we have reduced this computation time by directly validating the pixels assigned to the same class by the two classifiers. In other words, the GA processes only the conflicting pixels. An

unsupervised evaluation criterion based on within and between class disparities is used as the objective function of the GA.

- In the case of multicomponent images, each component is partitioned separately and independently using the above steps. Then the results of the different components are fused to get the final result of the multicomponent image. The same evaluation criterion is used, but it is modified so as to take into account all the components of the image while evaluating the partitioning results.

The architecture of the developed system permits the parallel execution of the different steps, thereby reducing the calculation time.

The experimental results obtained on different images (multispectral/ hyperspectral) have shown the efficiency of the developed system.

It is expected that this system will be improved in the following aspects:

- The features extracted could be extended to take into account the spectral dimension also. Indeed, recent advances in the use of higher order statistic features following also spectral information of hyperspectral images show that they can significantly improve the classification of pixels.
- Each classifier could be used more specifically according to the type of regions in order to better partition the images and reduce the computing time.

References

[1] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, “Imaging Spectrometry for Earth Remote Sensing,” *Science*, vol. 228, no. 4704, pp. 1147–1153, Jun. 1985.

[2] J. P. Ryan, C. O. Davis, N. B. Tufillaro, R. M. Kudela, and B.-C. Gao, “Application of the Hyperspectral Imager for the Coastal Ocean to Phytoplankton Ecology Studies in Monterey Bay, CA, USA,” *Remote Sens.*, vol. 6, no. 2, pp. 1007–1025, Jan. 2014.

[3] P. Lagueux, E. Puckrin, C. S. Turcotte, M.-A. Gagnon, J. Bastedo, V. Farley, and M. Chamberland, “Airborne infrared hyperspectral imager for intelligence, surveillance and reconnaissance applications,” *Proc. Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications IX*, Baltimore, Maryland, USA, vol. 8542, pp. 854226–854226–10, 2012.

[4] F. M. Lacar, M. M. Lewis, and I. T. Grierson, “Use of hyperspectral imagery for mapping grape varieties in the Barossa Valley, South Australia,” *Proc. IEEE International Geoscience and Remote Sensing Symposium, IGARSS*, vol. 6, pp. 2875–2877 vol.6, 2001.

[5] H. Akbari, Y. Kosugi, K. Kojima, and N. Tanaka, “Detection and Analysis of the Intestinal Ischemia Using Visible and Invisible Hyperspectral Imaging,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 8, pp. 2011–2017, Aug. 2010.

[6] Y.-Z. Feng and D.-W. Sun, “Application of hyperspectral imaging in food safety inspection and control: a review,” *Crit. Rev. Food Sci. Nutr.*, vol. 52, no. 11, pp. 1039–1058, 2012.

[7] M. Awad, K. Chehdi, and A. Nasri, “Multi-component image segmentation using a hybrid dynamic genetic algorithm and fuzzy C-means,” *IET Image Process.*, vol. 3, no. 2, pp. 52–62, Apr. 2009.

[8] M. Awad, K. Chehdi, and A. Nasri, “Multicomponent Image Segmentation Using a Genetic Algorithm and Artificial Neural Network,” *Geosci. Remote Sens. Lett. IEEE*, no. 4, pp. 571 – 575, 2007.

[9] M. M. Awad, *Mise en Oeuvre D’un Système Coopératif Adaptatif de Segmentation D’images Multicomposantes*, Phd Thesis, University of Rennes 1, 2008.

[10] C. Rosenberger, *Mise en oeuvre d’un systeme adaptatif de segmentation d’images*, Phd Thesis, University of Rennes 1, 1999.

[11] N. Voisine, *Approche adaptative de coopération hiérarchique de méthodes de segmentation: application aux images multicomposantes*, Phd Thesis, University of Rennes 1, 2002.

[12] C. D. Kermad and K. Chehdi, “Automatic image segmentation system through iterative edge-region co-operation,” *Image Vis. Comput. Ed. Elsevier*, vol. 20, pp. 541–555, 2002.

[13] T. Poggio, V. Torre, and C. Koch, “Computational vision and regularization theory,” *Nature*, vol. 317, no. 6035, pp. 314–319, Sep. 1985.

[14] G. Stockman and L. G. Shapiro, *Computer Vision*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. R. Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[16] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[17] T. K. Moon, “The expectation-maximization algorithm,” *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[18] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967.

[19] Y. Linde, A. Buzo, and R. M. Gray, “An Algorithm for Vector Quantizer Design,” *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, 1980.

[20] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.

[21] B. J. Frey and D. Dueck, “Clustering by Passing Messages Between Data Points,” *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[22] V. K. Dehariya, S. K. Shrivastava, and R. C. Jain, “Clustering of Image Data Set Using K-Means and Fuzzy K-Means Algorithms,” *Proc. International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 386–391, 2010.

[23] J. M. Peña, J. A. Lozano, and P. Larrañaga, “An empirical comparison of four initialization methods for the K-Means algorithm,” *Pattern Recognit. Lett.*, vol. 20, no. 10, pp. 1027–1040, Oct. 1999.

[24] S. Bubeck, M. Meilă, and U. von Luxburg, “How the initialization affects the stability of the κ-means algorithm,” *ESAIM Probab. Stat.*, vol. 16, pp. 436–452, 2012.

[25] R. Xu and I. Wunsch, D., “Survey of clustering algorithms,” *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, 2005.

[26] T. Kohonen, “The self-organizing map,” *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[27] E. L. Bohez, “Two level cluster analysis based on fractal dimension and iterated function systems (IFS) for speech signal recognition,” *Proc. IEEE*

Asia-Pacific Conference on Circuits and Systems (APCCAS), pp. 291–294, 1998.

[28] M. F. Hussin, M. S. Kamel, and M. H. Nagi, “An Efficient Two-Level SOMART Document Clustering Through Dimensionality Reduction,” in *Neural Information Processing*, N. R. Pal, N. Kasabov, R. K. Mudi, S. Pal, and S. K. Parui, Eds. Springer Berlin Heidelberg, pp. 158–165, 2004.

[29] *Selection of Clusters Number and Features Subset During a Two-Levels Clustering Task*. .

[30] X. Li, X. Lu, J. Tian, P. Gao, H. Kong, and G. Xu, “Application of Fuzzy c-Means Clustering in Data Analysis of Metabolomics,” *Anal. Chem.*, vol. 81, no. 11, pp. 4468–4475, Jun. 2009.

[31] J. C. Dunn, “A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters,” *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.

[32] Y. Yong, Z. Chongxun, and L. Pan, “A Novel Fuzzy C-Means Clustering Algorithm for Image Thresholding,” *Meas. Sci. Rev.*, vol. 4, pp. 11–19, 2004.

[33] T. Kim, H. Adeli, C. Ramos, and B.-H. Kang, *Signal Processing, Image Processing and Pattern Recognition*. Springer, 2011.

[34] N. R. Pal and J. C. Bezdek, “On cluster validity for the fuzzy c-means model,” *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.

[35] K.-L. Wu, “Analysis of parameter selections for fuzzy c-means,” *Pattern Recognit.*, vol. 45, no. 1, pp. 407–415, Jan. 2012.

[36] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. Dover Publications, Incorporated, 1999.

[37] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriarty, “A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data,” *IEEE Trans. Med. Imaging*, vol. 21, no. 3, pp. 193–199, Mar. 2002.

[38] S. Chen and D. Zhang, “Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure,” *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 34, no. 4, pp. 1907–1916, Aug. 2004.

[39] L. Szilagyi, Z. Benyo, S. M. Szilagyi, and H. S. Adam, “MR brain image segmentation using an enhanced fuzzy C-means algorithm,” *Proc. IEEE 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, pp. 724–726 Vol.1, 2003.

[40] L. Szilágyi, S. M. Szilágyi, and Z. Benyó, “A Modified Fuzzy C-Means Algorithm for MR Brain Image Segmentation,” in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds. Springer Berlin Heidelberg, pp. 866–877, 2007.

[41] W. Cai, S. Chen, and D. Zhang, “Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation,” *Pattern Recognit.*, vol. 40, no. 3, pp. 825–838, Mar. 2007.

[42] S. Krinidis and V. Chatzis, “A Robust Fuzzy Local Information C-Means Clustering Algorithm,” *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1328–1337, May 2010.

[43] J. C. Noordam, W. H. A. M. Van den Broek, and L. M. C. Buydens, “Geometrically guided fuzzy C-means clustering for multivariate image segmentation,” *Proc. IEEE 15th International Conference on Pattern Recognition*, vol. 1, pp. 462–465 vol.1, 2000.

[44] D. H. B. Kekre, S. M. Gharge, and T. K. Sarode, “Tumor Demarcation in Mammography Images using LBG on Probability Image,” *Int. J. Comput. Appl.*, vol. 3, no. 8, pp. 47–53, Jun. 2010.

[45] C.-W. Tsai, C.-Y. Lee, M.-C. Chiang, and C.-S. Yang, “A fast VQ codebook generation algorithm via pattern reduction,” *Pattern Recognit. Lett.*, vol. 30, no. 7, pp. 653–660, May 2009.

[46] B. Huang and L. Xie, “An improved LBG algorithm for image vector quantization,” in *2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, vol. 6, pp. 467–471, 2010.

[47] B. Fritzke, “The LBG-U Method for Vector Quantization – an Improvement over LBG Inspired from Neural Networks,” *Neural Process. Lett.*, vol. 5, no. 1, pp. 35–45, Feb. 1997.

[48] G. Patané and M. Russo, “The enhanced LBG algorithm,” *Neural Netw.*, vol. 14, no. 9, pp. 1219–1237, Nov. 2001.

[49] F. Shen and O. Hasegawa, “An adaptive incremental LBG for vector quantization,” *Neural Netw.*, vol. 19, no. 5, pp. 694–704, Jun. 2006.

[50] C. Rosenberger and K. Chehdi, “Unsupervised clustering method with optimal estimation of the number of clusters: application to image segmentation,” *Proc. 15th International Conference on Pattern Recognition*, vol. 1, pp. 656–659 vol.1, 2000.

[51] S. Içer, “Automatic Segmentation of Corpus Collasum Using Gaussian Mixture Modeling and Fuzzy C Means Methods,” *Comput Methods Prog Biomed*, vol. 112, no. 1, pp. 38–46, Oct. 2013.

[52] L. Yanxia, Y. Jiawei, and L. Ye, “One Effective Method to Design LBG Initial Codebook,” *Proc. International Conference on Intelligent Computation Technology and Automation (ICICTA)*, vol. 2, pp. 628–631, 2011.

[53] R. Xu and D. Wunsch, *Clustering*. John Wiley & Sons, 2008.

[54] J. C. Noordam, W. H. A. M. van den Broek, and L. M. C. Buydens, “Multivariate image segmentation with cluster size insensitive Fuzzy C-means,” *Chemom. Intell. Lab. Syst.*, vol. 64, no. 1, pp. 65–78, Oct. 2002.

[55] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA, USA: MIT Press, 1992.

[56] B.-C. Kuo and D. . Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, May 2004.

[57] C.-T. Li and R. Chiao, "Unsupervised texture segmentation using multiresolution hybrid genetic algorithm," *Proc. International Conference on Image Processing (ICIP)*, vol. 2, pp. II–1033–6 vol.3, 2003.

[58] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective Genetic Clustering for Pixel Classification in Remote Sensing Imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1506–1511, May 2007.

[59] C.-C. Hung, S. Kulkarni, and B.-C. Kuo, "A New Weighted Fuzzy C-Means Clustering Algorithm for Remotely Sensed Image Classification," *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 3, pp. 543–553, Jun. 2011.

[60] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Comput Surv*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[61] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. New York: Wiley, 2001.

[62] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. Hoboken, N.J.: Wiley, 2005.

[63] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, 2008.

[64] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*. John Wiley & Sons, 2011.

[65] A. K. Jain and J. V. Moreau, "Bootstrap technique in cluster analysis," *Pattern Recognit.*, vol. 20, no. 5, pp. 547–568, 1987.

[66] R. Kothari and D. Pitts, "On finding the number of clusters," *Pattern Recognit. Lett.*, vol. 20, no. 4, pp. 405–416, Apr. 1999.

[67] L. Wang, C. Leckie, K. Ramamohanarao, and J. Bezdek, "Automatically Determining the Number of Clusters in Unlabeled Data Sets," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 3, pp. 335–350, Mar. 2009.

[68] J. M. Buhmann and M. Held, *Unsupervised Learning Without Overfitting: Empirical Risk Approximation As An Induction Principle For Reliable Clustering*. 1998.

[69] D. Stanford and A. E. Raftery, "Principal Curve Clustering With Noise," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1997.

[70] Y. H. Man and I. Gath, "Detection and separation of ring-shaped clusters using fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 8, pp. 855–861, Aug. 1994.

[71] G. Forestier, P. Gançarski, and C. Wemmert, “Collaborative clustering with background knowledge,” *Data Knowl. Eng.*, vol. 69, no. 2, pp. 211–228, Feb. 2010.

[72] Y. Tian, F. Duan, M. Zhou, and Z. Wu, “Active contour model combining region and edge information,” *Mach. Vis. Appl.*, vol. 24, no. 1, pp. 47–61, Jan. 2013.

[73] A. Bhalerao and R. Wilson, “Unsupervised Image Segmentation Combining Region and Boundary Estimation,” *Image Vis. Comput.*, vol. 19, pp. 353–386, 2000.

[74] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, “Yet another survey on image segmentation: Region and boundary information integration,” *ECCV*, p. 408–, 2002.

[75] N. Jamil, H. C. Soh, T. M. T. Sembok, and Z. A. Bakar, “A Modified Edge-Based Region Growing Segmentation of Geometric Objects,” in *Visual Informatics: Sustaining Research and Innovations*, H. B. Zaman, P. Robinson, M. Petrou, P. Olivier, T. K. Shih, S. Velastin, and I. Nyström, Eds. Springer Berlin Heidelberg, pp. 99–112, 2011.

[76] I. Sebari and D.-C. He, “Approach to nonparametric cooperative multiband segmentation with adaptive threshold,” *Appl. Opt.*, vol. 48, no. 20, pp. 3967–3978, Jul. 2009.

[77] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, “Spectral-Spatial Classification of Hyperspectral Imagery Based on Partitional Clustering Techniques,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2973–2987, 2009.

[78] H. R. Kalluri, S. Prasad, and L. M. Bruce, “Decision-Level Fusion of Spectral Reflectance and Derivative Information for Robust Hyperspectral Land Cover Classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4047–4058, 2010.

[79] J. A. Benediktsson and I. Kanellopoulos, “Classification of multisource and hyperspectral data based on decision fusion,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1367–1377, 1999.

[80] A. L. N. Fred and A. K. Jain, “Combining multiple clusterings using evidence accumulation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.

[81] M. Yoshimura and S. Oe, “Evolutionary segmentation of texture image using genetic algorithms towards automatic decision of optimum number of segmentation areas,” *Pattern Recognit.*, vol. 32, no. 12, pp. 2041–2054, Dec. 1999.

[82] H. Mahi and H. F. Izabatene, “Segmentation of Satellite Imagery using RBF Neural Network and Genetic Algorithm,” *Asian J. Appl. Sci.*, vol. 4, no. 2, pp. 186–194, Feb. 2011.

[83] T. Rohlfing and C. R. Maurer Jr., “Multi-classifier framework for atlas-based image segmentation,” *Pattern Recognit. Lett.*, vol. 26, no. 13, pp. 2070–2079, Oct. 2005.

[84] G. Forestier, C. Wemmert, and P. Gançarski, “Towards conflict resolution in collaborative clustering,” *Proc. Intelligent Systems (IS), 5th IEEE International Conference*, pp. 361–366, 2010.

[85] A. Lourenço, S. R. Bulò, N. Rebagliati, A. L. N. Fred, M. A. T. Figueiredo, and M. Pelillo, “Probabilistic consensus clustering using evidence accumulation,” *Mach. Learn.*, pp. 1–27.

[86] N. Benamrane and S. Nassane, “Medical Image Segmentation by a Multi-Agent System Approach,” in *Multiagent System Technologies*, P. Petta, J. P. Müller, M. Klusch, and M. Georgeff, Eds. Springer Berlin Heidelberg, pp. 49–60, 2007.

[87] P. Paclík, R. P. W. Duin, G. M. P. van Kempen, and R. Kohlus, “Segmentation of multi-spectral images using the combined classifier approach,” *Image Vis. Comput.*, vol. 21, no. 6, pp. 473–482, Jun. 2003.

[88] P. Gançarski and C. Wemmert, “Collaborative multi-step mono-level multi-strategy classification,” *Multimed. Tools Appl.*, vol. 35, no. 1, pp. 1–27, Oct. 2007.

[89] X. Jin and C. H. Davis, “A genetic image segmentation algorithm with a fuzzy-based evaluation function,” *Proc. The 12th IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 938–943 vol.2, 2003.

[90] S.-Y. Ho and K.-Z. Lee, “Efficient image segmentation using a generic and non-parametric approach,” *Proc. The Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region*, vol. 2, pp. 777–781 vol.2, 2000.

[91] M. Moghrani, *Segmentation coopérative et adaptative d’images multicomposantes application aux images CASI*, Phd Thesis, University of Rennes 1, 2007.

[92] W. Chao, D. Q. Jishuang, D. Student, and L. Zhi, “Data Fusion, the Core Technology for Future on-Board Data Processing System,” *Proc. Pecora 15/Land Satellite Information IV/ISPRS Commission I/FIEOS*, Denver,USA.

[93] S. G. Nikolov, D. R. Bull, C. N. Canagarajah, M. Halliwell, and P. N. T. Wells, “Image fusion using a 3-D wavelet transform,” *Proc. Seventh International Conference on Image Processing And Its Applications*, vol. 1, pp. 235–239, 1999.

[94] G. Piella, “A general framework for multiresolution image fusion: from pixels to regions,” *Inf. Fusion*, vol. 4, no. 4, pp. 259–280, Dec. 2003.

[95] O. Rockinger, T. Fechner, and D. B. Ag, *Pixel-Level Image Fusion: The Case of Image Sequences*. 1998.

[96] A. Strehl, J. Ghosh, and C. Cardie, “Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions,” *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2002.

[97] A. Topchy, A. K. Jain, and W. Punch, “Combining Multiple Weak Clusterings,” *Proc. Third IEEE International Conference onData Mining (ICDM).*, Washington, DC, USA, pp. 331–338, 2003.

[98] Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, “Multiple Spectral-Spatial Classification Approach for Hyperspectral Data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4122–4132, Nov. 2010.

[99] M. Awad, K. Chehdi, and A. Nasri, “Multicomponent image segmentation: a comparative analysis between a hybrid genetic algorithm and self-organizing maps,” *Int. J. Remote Sens.*, vol. 30, no. 3, pp. 595–610, 2009.

[100] M. Awad, K. Chehdi, and A. Nasri, “Enhancement of the segmentation process of multi-component images using fusion with Genetic Algorithm,” *Proc. IEEE 5th International Multi-Conference on Systems, Signals and Devices*, pp. 1–6, 2008.

[101] C. Rosenberger, K. Chehdi, and C. Kermad, “Adaptive segmentation system,” *Proc. 5th International Conference on Signal Processing, WCCC-ICSP*, vol. 2, pp. 918–921 vol.2, 2000.

[102] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “Clustering Validity Checking Methods: Part II,” *SIGMOD Rec*, vol. 31, no. 3, pp. 19–27, Sep. 2002.

[103] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[104] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 1st ed. Boston: Addison-Wesley, 2005.

[105] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, “Validity index for crisp and fuzzy clusters,” *Pattern Recognit.*, vol. 37, no. 3, pp. 487–501, Mar. 2004.

[106] R.M. Haralick and L. G. Shaprio, “Image Segmentation Techniques,” *Proc. Computer Vision Graphics and Image Processing*, Arlington, vol. 29, pp. 100–132, 1985.

[107] J. S. Weszka and A. Rosenfeld, “Threshold Evaluation Techniques,” *IEEE Trans. Syst. Man Cybern.*, vol. 8, no. 8, pp. 622–629, Aug. 1978.

[108] M. D. Levine and A. M. Nazif, “Dynamic Measurement of Computer Generated Image Segmentations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-7, no. 2, pp. 155–164, Mar. 1985.

[109] B. S. Mehmet Sezgin, “Survey over image thresholding techniques and quantitative performance evaluation,” *J. Electron. Imaging*, vol. 13, pp. 146–168, 2004.

[110] W. G. Cochran, “Some Methods for Strengthening the Common X 2 Tests,” *Biometrics*, vol. 10, no. 4, p. 417, Dec. 1954.

[111] N. R. Pal and S. K. Pal, “Entropic thresholding,” *Signal Process.*, vol. 16, no. 2, pp. 97–108, Feb. 1989.

[112] R. Zéboudj, *Filtrage, seuillage automatique, contraste et contours: du pré-traitement à l’analyse d’image*. Saint-Etienne, 1988.

[113] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[114] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[115] J. C. Dunn†, “Well-Separated Clusters and Optimal Fuzzy Partitions,” *J. Cybern.*, vol. 4, no. 1, pp. 95–104, Jan. 1974.

[116] C. Rosenberger and K. Chehdi, “Toward a complete adaptive analysis of an image,” *J. Electron. Imaging*, vol. 12, no. 2, pp. 292–298, 2003.

[117] R. M. Haralick, “Statistical and structural approaches to texture,” *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[118] C. S. Won and Y. Choe, “Image block classification using stochastic image segmentation,” *Electron. Lett.*, vol. 32, no. 16, pp. 1462–1463, Aug. 1996.

[119] M. M. Galloway, “Texture analysis using gray level run lengths,” *Comput. Graph. Image Process.*, vol. 4, no. 2, pp. 172–179, Jun. 1975.

[120] J. Ronsin, D. Barba, and S. Raboisson, “Comparison Between Cooccurrence Matrices, Local Histograms And Curvilinear Integration For Texture Characterization,” *Proc. Architectures and Algorithms for Digital Image Processing III*, vol. 0596, pp. 98–104, 1986.

[121] S. Chabrier, B. Emile, C. Rosenberger, and H. Laurent, “Unsupervised Performance Evaluation of Image Segmentation,” *EURASIP J. Adv. Signal Process.*, vol. 2006, no. 1, p. 096306, Mar. 2006.

[122] A. Taher, K. Chehdi, and C. Cariou, “An Unsupervised Nonparametric and Cooperative Approach for Classification of Multicomponent Image Contents,” *Proc. International Conference on Pattern Recognition Applications and Methods, ICPRAM*, Angers-France, pp. 263–270, 2014.

[123] A. Taher, K. Chehdi, and C. Cariou, “Hyperspectral image segmentation using a cooperative nonparametric approach,” *Proc. Image and signal processing for remote sensing XIX*, Dresden-Germany, vol. 8892, p. 88920J–88920J–8, 2013.

[124] C. Rosenberger and K. Chehdi, “Genetic fusion: application to multi-components image segmentation,” *Proc. IEEE Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 2223–2226, 2000.

[125] B. Karasulu and S. Korukoglu, “A simulated annealing-based optimal threshold determining method in edge-based segmentation of grayscale images,” *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2246–2259, Mar. 2011.

[126] M. H. Karimi and D. Asemani, “Surface defect detection in tiling Industries using digital image processing methods: Analysis and evaluation,” *ISA Trans.*

[127] A. Martínez-Usó, F. Pla, and P. García-Sevilla, “Unsupervised colour image segmentation by low-level perceptual grouping,” *Pattern Anal. Appl.*, vol. 16, no. 4, pp. 581–594, Nov. 2013.

[128] S. Chabrier, B. Emile, H. Laurent, C. Rosenberger, and P. Marche, “Unsupervised evaluation of image segmentation application to multi-spectral images,” *Proc. 17th International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 576–579 Vol.1, 2004.

[129] F. Neumann, P. S. Oliveto, and C. Witt, “Theoretical Analysis of Fitness-proportional Selection: Landscapes and Efficiency,” *Proc. 11th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA, pp. 835–842, 2009.

[130] G. Syswerda, “Uniform Crossover in Genetic Algorithms,” *Proc. 3rd International Conference on Genetic Algorithms*, San Francisco, CA, USA, pp. 2–9, 1989.

List of figures

List of tables

Appendices

Appendix A: Summary of non-cooperative methods

Table A.1: Unsupervised and semi-supervised non-cooperative and non-parametric methods

Unsupervised methods				
Method	**Input used**	**Input parameters**	**Remarks**	**Applicable to multicomponent**
Genetic Algorithms [55]	Depends on the application	None	Requires large amount of data to give optimal results	Yes
Hierarchical Genetic Algorithm [56]	Gray level value of pixels	Predefined threshold value.	Developed to overcome the difficulties of GA	Yes
Hybrid Genetic Algorithm [57]	Mean gray value	Population size, iteration number.	Crossover replaced by *k*-means	Yes with modifications
Multi-objective variable length string GA [58]	Gray level value of the pixels	Population size, iteration number, crossover probability.	Evaluation method used does not require any knowledge	Already used
Modified Fuzzy C-means clustering [59]	Grey level value of pixels	Fuzzification value, iteration number	Developed to improve the FCM (stability, and accuracy). Slight improvement	Already used

Semi-supervised methods				
Method	**Input used**	**Input parameters**	**Remarks**	**Applicable to multicomponent**
k-means [18]	Depends on the application	Number of classes and iterations	Unstable	Yes
Kohonen Neural Network Self-Organizing Maps (SOM) [26]	Hyperspectral signatures of the pixels	Number of iterations, neighborhood starting value	Number of iterations, neighborhood starting value effect the final result	Already used
FCM [20]	Depends on the application	Number of classes and iterations, fuzzification factor *m*	Unstable, choice of *m* effect the results	Yes
LBG [46]	Grey level value of pixels	Number of classes and iteration	Unstable, developed for vector quantization	Yes
AILBG [49]	Grey level value of pixels	Number of classes and iterations	Stable, developed for vector quantization	Yes
Geometric Guided Fuzzy C-Means Clustering [43]	Grey level value of pixels	Prior geometric knowledge	Adds geometrical information during clustering	Yes

Appendix B: Summary of cooperative approaches

Table B.1: Sequential cooperative approaches

Method	Input used	Input parameters	Remarks	Applicable to multicomponent
k-means and GA [89]	Gray level values of pixels	Population size, iteration number, crossover probability.	Dedicated to edge detection	Already used
FCM and orthogonal array [90]	Gray level value of pixels	None	Can be used in two modes (supervised and unsupervised), dedicated to edge detection	Yes
Radial Basis Function Neural Network (RBFNN) and genetic algorithm [82]	Texture features	None	One of the methods used in cooperation is parametric	Already used
Self-Organizing Maps (SOMs) and Genetic algorithm (GA) [81]	Texture Features	Grid size and topology of the SOM map	Effective for partitioning images that contain similar texture fields	Already used

Table B.2: Parallel cooperative approaches

Method	Input used	Input parameters	Remarks	Applicable to multicomponent
SVM and ISODATA/ SVM and EM [77]	Spectral values of the image pixels	Training samples, number of classes, ISODATA adjusting parameters	The ISODATA or EM results are used as a dynamic mask to relax SVM results	Already used
SVM, Watershed, EM and Recursive Hierarchical Segmentation (RHSEG) [98]	Spectral values of the image pixels	Training samples, number of classes, ISODATA adjusting parameters	Uses minimum spanning forest for the fusion process	Already used
Only ML (Changing features) [78]	Spectral values of the image	ML adjusting parameters and type of function used in it	The fusion technique is based on weighted-linear opinion pool	Already used

Table B.3: Hybrid cooperative approaches

Method	Input used	Input parameters	Remarks	Applicable to multicomponent
Neural network and ML[79]	Spectral values of the image pixels	Training samples, number of classes	Invalidated classification results are called into question	Already used
SOM+Hybrid GA, FCM+ Hybrid Dynamic GA, NURBS+ Hybrid Dynamic GA [9]	Gray level value of the pixels	Interactive data for the NURBS	The system is composed of three sequentially cooperative subsytems then the results are fused using GA. The fusion process needs also some parameter initialization	Already used
MLBG and GA [10]	Texture features	Major number of classes	MLBG is very time consuming. Adaptive to the content of the image	Already used

Appendix C: Summary of unsupervised evaluation criteria

Table C.1: Summary of main internal (unsupervised) evaluation criteria

Evaluation Criteria	Remarks
Sum of squared errors	Measures within-class disparity.
Levine and Nazif (*LEV1*) [108]	Measures within-class disparity.
Levine and Nazif (*LEV2*) [108]	Measures within class and between class disparities.
Zeboudj index [112]	Measures within class and between class disparities.
Davies-Bouldin index [113]	Measures within class and between classes, time consuming.
Silhouette index [114]	Measures with-in class and between classes, very time consuming.
Dunn index [115]	Measures with-in class and between class disparities, not effective in case of noisy images
Rosenberger and Chehdi [10], [101]	Measures with-in class and between class disparities, takes into account the region type (textured and non textured)