Hyperspectral Image Segmentation through Evolved Cellular Automata

Blanca Priego, Daniel Souto, Francisco Bellas, Richard J. Duro

Integrated Group for Engineering Research
Universidade da Coruña, Spain
{blanca.priego, dsouto, fran, richard}@udc.es
Tel/fax: +34981337400/+34981337419

Abstract. Segmenting multidimensional images, in particular hyperspectral images, is still an open subject. Two are the most important issues in this field. On one hand, most methods do not preserve the multidimensional character of the signals throughout the segmentation process. They usually perform an early projection of the hyperspectral information to a two dimensional representation with the consequent loss of the large amount of spectral information these images provide. On the other hand, there is usually very little and dubious ground truth available, making it very hard to train and tune appropriate segmentation and classification strategies. This paper describes an approach to the problem of segmenting and classifying regions in multidimensional images that performs a joint two-step process. The first step is based on the application of cellular automata (CA) and their emergent behavior over the hyperspectral cube in order to produce homogeneous regions. The second step employs a more traditional SVM in order to provide labels for these regions to classify them. The use of cellular automata for segmentation in hyperspectral images is not new, but most approaches to this problem involve hand designing the rules for the automata and, in general, average out the spectral information present. The main contribution of this paper is the study of the application of evolutionary methods to produce the CA rule sets that result in the best possible segmentation properties under different circumstances without resorting to any form of projection until the information is presented to the user. In addition, we show that the evolution process we propose to obtain the rules can be carried out over RGB images and then the resulting automata can be used to process multidimensional

hyperspectral images successfully, thus avoiding the problem of lack of appropriately labeled ground truth images. The procedure has been tested over synthetic and real hyperspectral images and the results are very competitive.

Keywords: Hyperspectral imaging, Evolution, Cellular Automata, Segmentation.

1 Introduction

Multidimensional image processing is becoming a field of great interest. Specifically, hyperspectral imaging is now a very important source of remote sensing and industrial processing information. It implies the acquisition of imaging data with a very high level of spectral detail. Unlike more traditional imaging techniques which are based on a three value representation of the color of each pixel (RGB), hyperspectral images provide hundreds of values that correspond to the intensity of a large number of bands in the visible and near infrared spectrum. Obviously, this leads to a very accurate and information intensive definition of the color they capture, making the discrimination of different elements within the images easier. To provide some order of magnitude values, a hyperspectral image is made up of two spatial dimensions each comprising anywhere from 250 pixels to a few of thousand pixels and a spectral dimension of up to a thousand bands per pixel.

Originally hyperspectrometers were applied to remote sensing and the images were obtained from high flying planes or satellites [1]. Initially, the processing was strongly biased towards determining the content of each pixel in the image. Thus, classification methods were developed to assign the spectrum of each pixel to one of the possible classes [2]. These types of applications implied that each pixel could usually correspond to a large area (100 square meters or more) containing different types of covers (endmembers), and thus, it became a necessity to develop analysis methods aimed at providing the ratio of endmembers present in every pixel [3][4]. In fact, in all of these applications the emphasis was placed on the processing of the spectral characteristics found in the different pixels, mostly without considering spatial or geometric relationships. This is currently not the case for two main reasons, on one hand, hyperspectrometers have improved their spatial definition drastically and, on the other, there has been a great drive towards ground based applications [5][6]. In this second case, the images are taken close enough to the subject to obtain a relatively detailed view allowing for the use of the geometric layout of the pixel sets in order to provide an adequate identification or classification of the elements present. That is, it is becoming necessary to

perform combined spatial-spectral processing in order to identify morphological features present in the image and this obviously implies segmenting patterns from the background and classifying them.

To integrate spatial and spectral information within a classification process contemplates using the information on the spectrum present in the pixel and those of the pixels around it in order to decide to which class this pixel belongs. To this end, different approaches have arisen in the last decade. Some, such as morphological levels [7] or Markov random fields [8][9], use fixed window based neighborhoods and thus are not as flexible as would be required for complex images. Others resort to mathematical morphology based techniques [10][11] or watershed based algorithms [12] [13] to segment the images and try to extend them to this domain although it is not clear how to modify the operators involved due to the ordering problem that one finds in high dimensional spaces. Other segmentation techniques have been proposed [14], mostly extensions of approaches developed for lower dimensionalities, such as region-merging [15] or hierarchical segmentation [16]. One can find more advanced techniques that apply intelligent learning algorithms, such as those presented in [17] [18] or [19]. Finally, some authors [20] even resort to lattice computing techniques. These approaches usually require complex algorithms that lead to weighty processing stages that are very hard to implement in limited computing resources, making them inadequate for real time processing.

In this sense, multidimensional images usually involve very large data sets due to the spectral detail they contain for each spatial pixel. The size of these data sets becomes even larger when high spatial resolutions are involved. This represents a severe processing problem that requires the development of specialized techniques and the use of very high levels of parallelism, especially when real time processing is desired. Thus, it is often the case that one would like to have an algorithm that iteratively performs simple operations over pixels and their immediate surroundings so that the current state of the art distributed architectures, such as those based on FPGAs [21] or GPUs [22], can be used. This is not easy with most segmentation algorithms commented above, as their algorithmic structure makes them rather cumbersome in this type of architectures and very specialized ad-hoc implementations have to be developed in order to benefit from their architecture. It is in this context that techniques that are

intrinsically extremely distributed, such as Cellular Automata (CA), stand out and open up a wealth of opportunities for the development of specialized algorithms.

Cellular automata were first proposed by Von Neumann and Ulam [23] as a biologically inspired distributed computing paradigm. They demonstrated that using these systems a universal computing machine could be constructed. Cellular automata are a spatially extended decentralized system made up of a number of cells distributed throughout a portion of space, usually in some type of grid fashion, and which communicate only with their neighbors. Each cell defines an automaton (usually a finite state automaton) and, like any other automaton, the state of a cell each instant of time depends on the state of its neighbors and on its own previous state through a number of transition rules. The secret to harnessing the power of CAs in order to achieve a particular objective is to adequately choose the rule set so that by iterating the state transition process in time this objective is achieved.

Thus, the main challenge in order to use CAs for a particular application, especially when complex functions need to be achieved, is to determine the appropriate rules for the automata. That is, it is necessary to determine what rules implemented in each cell will produce, as the CA iterates, the desired global behavior for the CA based system. This is the so called inverse problem which presents a very high level of difficulty. The inverse problem has been addressed from many points of view by researchers in the CA community (see, for example, [24]). However, the most popular approach has been to use evolutionary techniques in order to evolve the rule set [25][26][27][28].

This paper deals with a particular realm, namely, hyperspectral imaging. Within this topic, as series of authors in the field of segmenting low dimensional images such as [29] [30] [31], and some in the high dimensional image processing area, such as [32] in the case of edge detection, or [13] in the case of watershed based segmentation, have started to address these problems using CAs. Most of the CAs they have proposed, however, have usually been hand created ad hoc and, even though some do consider multidimensional images, most are still far from the dimensions of hyperspectral images. In fact, when multidimensional images are considered, they are usually projected onto a lower dimension and it is over this dimension that the whole the segmentation process takes place.

Therefore, the field of the application of CAs to multidimensional image processing is still wide open. On one hand, there is a need to explore algorithms and techniques that permit obtaining the rule sets of the CAs so that a particular task over a given type of images can be achieved in a repeatable manner. In order to preserve the representational power of these high dimensional images it is important that these algorithms work with the complete spectral wealth of the images and do not project them onto lower dimensionality spaces. On the other, it is necessary to provide methodologies that allow this process to be achieved without having to resort to large training or labeled sets of images, as in the case of multidimensional hyperspectral images reliably labeled sets of images are very hard to come by.

These are the problems that will be addressed in this paper for the case of the segmentation of hyperspectral images. To this end a two-step algorithm is presented in which the first step makes use of Cellular Automata in order to create spectrally representative relatively homogeneous regions which are then classified using a more traditional SVM algorithm. This algorithm has been called ECAS (Evolutionary Cellular Automata based Segmentation) and its main objective here is to explore the possibilities of evolving a CA based system for the first step of the hyperspectral image segmentation and classification process using spectrally relevant information. That is, without irreversibly projecting the images onto lower dimensional spaces.

As the evolutionary process requires an evaluation phase and this evaluation needs some type of ground truth, instead of resorting to complex labeled hyperspectral images, which are usually not very trustworthy, we will generalize the segmentation function so that the CA can be trained using synthetic RGB images and then directly applied to hyperspectral images. To this end we consider a particular type of CA, that is, the eight cell neighborhood CA (usually called a Moore neighborhood) as it has been shown to be capable of universal computation [33] together with a spectral distance measure that is independent from the dimensionality of the data.

The remainder of the paper is structured as follows. In the next section we will provide an overview of the procedure we are following to evolve the cellular automata as well as a description of their characteristics and how the data are processed. Section 3 is devoted to the application of the CA once it has been evolved, that is, how the

segmentation and classification are performed using the information provided by the CA. Section 4 presents the experiments that were carried out over real hyperspectral images and some comparative results of this approach. Finally, Section 5 provides some conclusions and indications of future avenues of work.

2 Cellular Automata and their evolution

As indicated in the introduction, this paper deals with the evolution of cellular automata that can be used to segment hyperspectral images benefiting from the wealth of information the spectrum associated to each pixel provides. Consequently, in this section we will first present the cellular automaton we will be using and then we will dwell on how it will be evolved so as to simplify the process of obtaining appropriate CAs.

2.1 Definition and operation of the Cellular Automaton

In this work, a cell of the automaton is placed over each pixel of the hyperspectral image and to preserve the spectral character of the information, the state of the cell (S_i) will be given by an N-band spectrum, initially that of the pixel over which it is placed. Consequently, the state space is continuous and corresponds to the positive vector space $\mathbb{R}^N$. Due to the continuity of the state space, we will speak of state modifications rather than state changes. During execution, the CA is iteratively applied modifying each iteration the state (S_i) of each one of the cells on the image, that is, modifying the N-band vector that corresponds to the spectrum in that location. This modification depends on the information of the spectra (state) of the 8 closest neighboring cells (S_{i1}, S_{i2}, …., S_{i8}), on the set of rules that control the automaton and, finally, on the state of the cell over which the automaton is applied (S_i).

The modification that a cell state undergoes is given by a set of transition rules that control the operation of the CA. These rules can perform different operations over the spectra of the cells. Considering that a spectrum is a vector of size *N* (number of bands), these operations may be complex and costly in computational terms. As a

consequence, most approaches found in the literature usually project these spectra onto spaces of lower dimensionality and then operate over them. Here we are going to do something different, we are going to preserve the spectral information, that is, work with the whole N-band state vector throughout the segmentation process. Thus, the operations the transition rules of the CA will have to carry out will somehow modify the spectra that define the states of the cells based on comparisons of spectra. In order to be able to compare the spectra, a distance measure must be defined and here we have considered the *spectral angle*. For a cell *i*, the *spectral angle* α_{ij} with respect to cell *j* is defined as:

$$\alpha_{ij} = \frac{2}{\pi} arc\cos\left(\frac{\sum \mathbf{s_{ij}} \mathbf{s_i}}{\sqrt{\sum \mathbf{s_{ij}^2}} \sqrt{\sum \mathbf{s_i^2}}} \right), \quad \text{for j} = 1,2,...,8$$

This value is normalized between 0 and 1. Thus, for each cell of the hyperspectral image, a vector A_i made up of the eight angles with respect to the eight neighboring cells, can be computed as:

$$A_i = \left(\alpha_{i1}, \alpha_{i2}, ..., \alpha_{i8} \right)$$

In terms of the rule set of the cellular automaton, it consists of two *threshold angles*, ε and θ, and a set of M *rules*, each one of them made up of 9 parameters r_{xj}:

$$CA = \left(\varepsilon, \theta, r_{11}, r_{12}, ..., r_{19}, r_{21}, r_{22}, ..., r_{29}, ..., r_{M1}, r_{M2}, ..., r_{M9} \right), \begin{cases} 0 < \varepsilon < 1 \\ 0 < \theta < 1 \\ r_{ij} = \{-1, 0, 1\}, j = 1,2..,8 \\ r_{i9} = \{1,2,..8\} \end{cases}$$

Every iteration of the CA, only one of these M rules is applied in each cell, that is, over each pixel. To decide which one is selected, a very simple procedure involving the comparison of spectral angles has been developed. Thus, to run the CA rule set in a cell over a given pixel *i*, the first step is to compute vector A_i for this cell. Then,

and to simplify the rule search space, vector A_i is transformed into a new vector A^*_i through the transformation of angles α_{ij} into a set of *discretized angles* (α_{cij}) using the *threshold angles* of the automaton, ε and θ, and the following rule:

$$\alpha_{c_{ij}} = \begin{cases} -1 & if \quad \alpha_{ij} < \varepsilon \\ 0 & if \quad \varepsilon \le \alpha_{ij} \le \theta \\ 1 & if \quad \alpha_{ij} > \theta \end{cases}$$

Finally, the automaton selects the rule whose first eight r values ($r_{x1}, r_{x2}, \ldots, r_{x8}$) are most similar to A^*_i and this rule is applied in order to change the state of cell i. As similarity measure we have chosen the Euclidean distance.

The ninth parameter of each rule can take discrete values from 1 to 8 representing the cell neighbor that is selected. Specifically, 1 corresponds to the north-western cell, 2 to the northern one, 3 to the north-eastern one, 4 to eastern one, 5 to south-eastern one, 6 to southern one, 7 to south-western one and 8 to western one. Thus, for example, for a given cell (pixel) i, let's assume that the CA selects rule j for application and its ninth parameter is 4. This means that the spectrum of cell i will be modified towards that of the cell on its right (eastern direction). In other words, the state of a cell will tend to become more similar to that of the cell to which the rule points. Thus, the change of state of each cell corresponds to a modification of the spectrum of the cell. As a first approximation, we will consider as modification operator a spectrum averaging operator that implies that the resulting spectrum is the band by band average of the two spectra.

Finally, it must be pointed out that, in order to make rule selection invariant to rotations and reflections, four possible rotations and the vertical and horizontal flips of each one of them are taken into account during the rule selection process.

Following this procedure, the cellular automaton is iteratively applied to all the cells of the image. We can thus say, by assimilating each cell to an image pixel, that an updated hyperspectral cube is produced every iteration.

2.2 Evolutionary process

The selection of the set of rules that make up the cellular automaton is a complex problem that is difficult to solve manually. As stated above, this problem is addressed here in an automatic manner through the application of an evolutionary algorithm (EA). Consequently, it has been necessary to establish the individual encoding, the fitness function and specific algorithm to be applied.

Regarding the encoding of the individuals, the automata that make up the EA population are encoded as a vector with $9 \cdot M + 2$ floating point values in the [0, 1] interval. They are a direct representation of the *threshold angles* and *rules* of the automata.

On the other hand, to obtain a cellular automaton that works for a given range of desired image features, it is necessary to provide a set of sample images over which the prospective automata created by the EA can be evaluated during evolution in order to provide a fitness value. In the field of hyperspectral imaging this is usually a problem since there are not that many hyperspectral images appropriately and reliably labeled ground truth, and, it seems obvious that if the ground truth is not reliable, we might not obtain automata with the desired functionality. In order to avoid this problem, and as the operations and distance measures the CA described above is considering are independent of the dimensionality of the images, we decided to make use of synthetic RGB images as samples for the evolutionary process. This way it is a lot simpler to produce the variability range we wish the CA to generalize and one can even generate segmentable images and their ground truth in an automatic process.

Consequently, to determine its fitness, the prospective CA is run over each RGB image of the evaluation set for a given number of iterations and the result is compared to a desired ground truth. To this end, the spectral angles that must be computed for N-band vectors in a hyperspectral images are obtained here from a 3-band vector, the R, G and B components of each pixel. These evaluation images are generated in an automatic manner incorporating a predefined set of features we want to include in them. To improve the evaluation process a different image is generated at execution time for each evaluation.

There are several different possibilities to measure the quality of the segmentation obtained after applying the automaton in order to provide a fitness value for it. Here, we have chosen an error measure that seeks a balance between the homogeneity of the segmented regions and the discrimination of the different regions within the hyperspectral image. Specifically, we have defined fitness function to be minimized as:

$$e = \max\left(\text{intra-error, inter-error}\right)$$

$$e = \max\left(\frac{\sum_{i=1}^{K}\max(std(\mathbf{p}_{\mathbf{i,1}}),std(\mathbf{p}_{\mathbf{i,2}}),...,std(\mathbf{p}_{\mathbf{i},n})...,std(\mathbf{p}_{\mathbf{i,N}}))}{K\cdot 0.5}, 1-\frac{\sum_{i=1}^{K}\left(\sum_{l=1}^{nr_i} f(\alpha_{l,m_i},\alpha_{th})/nr_i\right)}{K}\right)$$

being

$$f(\alpha_{l,m_i},\alpha_{th}) = \begin{cases} 1 & if \quad \alpha_{l_i,m} > \alpha_{th} \\ 0 & f \quad \alpha_{l,m_i} \le \alpha_{th} \end{cases}$$

where K is the number of different regions in the hyperspectral image, N is the number of bands in the image, $\mathbf{p}_{i,n}$ is the reflectance value of the n^{th} band of the pixels belonging to region i, $std(\mathbf{p}_{i,n})$ is the standard deviation of $\mathbf{p}_{i,n}$, nr_i is the number of pixels of region i, $\alpha_{l,m}$ is the spectral angle between pixel l of region i and the average spectrum of region i and α_{th} is a threshold angle value.

Thus, the selected fitness function is an evaluation error e, which is defined as the maximum between two error measures: the intra-region error and the inter-region error (separated by a comma in the formula). The first one provides a measure of the homogeneity of the regions in the image, being a region a set of pixels that should have the same label after segmentation. To this end, the maximum standard deviation is computed band by band for all the pixels in the region and the intra-region error is obtained as a sum of these values. Therefore, if all the pixels in a

region have the same spectrum, such a region would be completely homogenous and the maximum band by band standard deviation would be zero.

Regarding the inter-region error, it provides a measure of the dissimilarity of two different regions. To compute it, for each region, the following steps are carried out:

1. The average spectrum is calculated.
2. The spectral angle α_{l,m_i} between the average spectrum and the spectrum of the remaining pixels in the image is calculated.
3. The number of angles above a threshold value α_{th} is computed and normalized with respect to the total number of pixels out of the region under consideration.
4. Steps 1 t o3 are repeated for all the regions in the image, being their sum the inter-region error.

Thus, if all the pixels in the image have the same spectrum, this inter-region error is maximum because

$$\sum_{l=1}^{nr_i} f(\alpha_{l,m_i}, \alpha_{th}) = 0$$

Finally, the particular evolutionary algorithm employed in this work is a standard Real Valued Genetic Algorithm (RVGA) with a configuration of parameters that will be presented in section 4. Once the evolutionary process finishes, the individual with the highest fitness (minimum error function *e*) is selected as the resulting CA that will be applied over the image or images to be segmented.

3 Cellular Automata and their execution: segmentation and classification

As stated before, the main purpose of this work is to propose a particular type of CAs that contemplate a continuous high dimensional spectral state space and whose rule sets have been evolved using for their evaluation synthetic images of much lower dimensionality and analyzing their performance in hyperspectral image

segmentation tasks. This type of CA is quite proficient in fusing spectral and spatial information to detect spectrally homogenous areas in this type of images. But the success of the application of the CA is not easily measurable unless one performs a classification process over benchmark cases and compares the results using standard metrics. To this end, this section describes how the complete segmentation and classification processes have been carried out in this paper in order to evaluate the performance of the CAs, which will be analyzed in section 4. It must be pointed out that that the whole segmentation and classification algorithms are not the main topic here but rather the application of CAs within the segmentation process, or the use of the information they provide in order to determine borders. For the other parts of the algorithms we have chosen to apply well known and published techniques (or modifications of them) so that the results could be compared to those of other authors.

3.1 Segmentation

Segmentation involves two basic processes, on one hand the homogenization of areas that must be differentiated from other areas and, on the other, the determination of the closed borders that delimit each area so that they can then be classified in a later process. The homogenization and differentiation of the areas is the process the CAs described in the previous section carry out. Their final output is an image where each pixel within an area should be represented by a spectrum that is very similar to that of its neighbors within the area and different from pixels in other areas. The next step in any segmentation algorithm involves determining where the borders of these areas are. In order to do this, we propose an edge detector that takes into account how the cell over each pixel behaved during the CA execution stage together with some other clues. This segmentation structure is what we have called the ECAS (Evolutionary Cellular Automata based Segmentation) algorithm.

A lot of information can be gathered from the way the CA operates when executed over an image in terms of what cells seem to be more closely related. Most authors using CAs do not make use of this information in later steps. However, we believe it can improve the border detection processes and, as a consequence, that we should

keep track of it. To this end, an eight-element vector that is updated each iteration has been associated to each cell of the CA. This vector corresponds to a set of eight counters ($c_{i1}, c_{i1}, ..., c_{i8}$) that allow counting the number of times the state of the cell (or spectrum of the pixel) has been averaged with each of its eight neighbors. Then, after n_{iter} iterations of the CA over the hyperspectral image, a transformed image is obtained which, together with the information stored in these counter vectors, can be segmented. The segmentation algorithm used here is based on a simple edge detection process with the following steps (for a better understanding, Figure 1 displays the resulting images after applying these steps over a real hyperspectral image):

1. An edge intensity image is obtained (Fig. 1a). To do it, for each pixel:
    a. The spectral angle to its eight neighbors $\alpha_{i1}, \alpha_{i1}, ..., \alpha_{i8}$ is calculated
    b. The edge intensity value is calculated from the spectral distances and the counter vector $c_{i1}, c_{i1}, ..., c_{i8}$ using the following relationship (Fig. 1b):

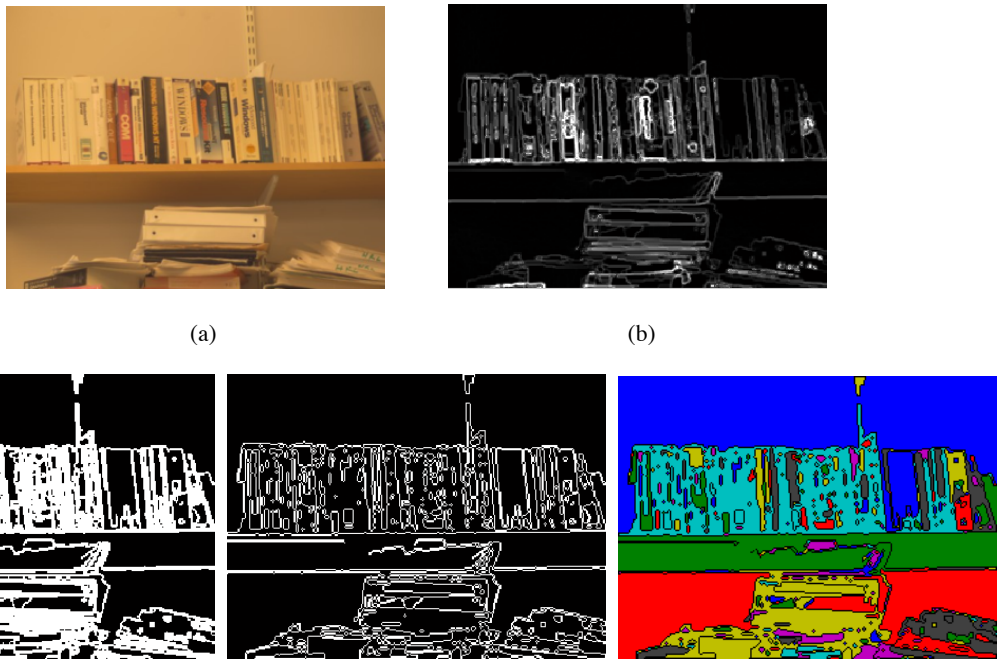
$$I_i = \frac{1}{8} \sum_{j=1}^{8} \alpha_{ij} \cdot \left(1 - \frac{c_{ij}}{n_{iter}} \right)$$

2. The edge intensity image is binarized using a threshold value, which in the cases presented here is (Fig. 1c):

$$threshold = \frac{\max\left(I_i\right)}{50}$$

3. This binary image is “cleaned”, using a morphological operator that removes isolated pixels (those with a value of 1 surrounded by 0 in the eight neighboring positions)
4. A second morphological operator, *remove*, is applied over the resulting image. This operator changes a cell value to 0 when its top, left, right and bottom neighbors present a value of 1. This way, only the boundary pixels are preserved (Fig. 1d)

5. Finally, all the pixels that are connected in this final image are arbitrarily labeled (Fig. 1e), that is, each self-contained area is given a label that differentiates it from its neighbors. Note that this label does not correspond to any particular class, i.e. areas corresponding to the same class may be labeled differently depending on where they are located on the image.



(a) (b)

(c) (d) (e)

Fig. 1. Different steps performed in the segmentation process over a real hyperspectral image (a) extracted from a database of the Harvard University [35]: (a): RGB representation of the hyperspectral image (b) edge intensity (c) binarized edge intensity (d) clean & remove operators (e) labeled image

3.2 Classification

Classification must combine both spatial and spectral information. To do it, a pixel-wise classification process is performed and this result is combined with the information obtained from the segmentation process described above. Specifically, the three following steps are carried out:

1. *SVM pixel-wise classification*: a standard SVM has been applied following the same training procedure as in the work of Tarabalka et al. [12]. Therefore, a multi-class pairwise (one versus one) SVM classification of the hyperspectral image is performed using a Gaussian Radial Basis Function (RBF) kernel. The specific implementation that has been used was that of the LIBSVM library [34], with the C and γ parameters optimally determined in [12] to C = 128 and γ =0.125.
2. *Spatial-spectral classification*: the resulting segmentation map is combined with the pixel-wise classification map. Specifically, all the pixels in a region of the segmentation map are assigned to the most frequent label in the classification map for that region (majority vote approach).
3. *Boundary labeling*: all the pixels in the resulting boundaries are assigned the most frequent label of its eight neighbors.

Figure 2 displays a classification result over the same real hyperspectral image used in Figure 1. Image (a) corresponds to the result after performing the pixel-wise classification (step 1 in the previously explained procedure), that is, considering only spectral information. It can be seen that the spectrum mixture is complex, and the different areas are not clearly defined. Image (b) corresponds to the result obtained after step 2, when combining spectral and spatial information. Now, it can be clearly observed how the areas are uniformly defined and the points corresponding to different spectra have been removed. Finally, in order to remove the boundaries, step 3 is performed over the image and the result is that of image (c).

(a) (b) (c)

Fig. 2. Spatial (a) and spatial-spectral (b and c) hyperspectral classification over a real hyperspectral image after applying steps 1, 2 and 3 of the algorithm described in section 3.2

Considering the segmentation and classification processes described in this section, the CA performance can now be compared to other techniques. This is the objective of the following section, where the results obtained after applying this two-step algorithm (CA+segmentation/classification) over real hyperspectral images are presented.

4 Experiments and Results

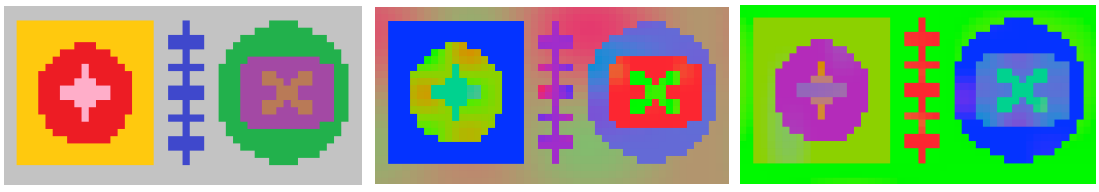
In this section we present some results of ECAS for segmenting hyperspectral images in an unsupervised manner. The automata were obtained using the algorithm described in section 2 and the classification results after applying the segmentation and classification algorithms described in section 3 over the image resulting from the application of the CA. For the sake of clarity, this section has been organized into three sub-sections. The first one is devoted to the specific details of the CA evolution process that has been carried out and a preliminary analysis of the practical segmentation that this type of CA performs depending on its parameterization. The second contains the classification results the two-step algorithm provides in perfectly labeled synthetic images. Finally, in the third sub-section, a benchmark real image is used and the results are compared to other methods.

In order to provide standard metrics for the quality of the classification results and be able to compare them to the results of others in the literature, in all the cases where classification was involved the following measures have been computed [12]:

1. Overall Accuracy (OA): is the percentage of correctly classified pixels
2. Average Accuracy (AA): is the mean of class-specific accuracies, that is, the mean of the percentage of correctly classified pixels for each class
3. Kappa coefficient ($\varkappa$): is the percentage of agreement (correctly classified pixels) corrected by the number of agreements that would be expected purely by chance.
4. Class-specific accuracies: is the percentage of agreement of each class (correctly classified pixels).

4.1 CA evolution and parameterization

As indicated in section 2.2, to evaluate the individuals of the genetic algorithm during the evolution of the CAs, simple RGB synthetic images that contain a predefined set of features similar to those one wants to segment from the hyperspectral images were generated. Specifically, for this particular evolutionary process image (a) of Figure 3 has been used as ground truth. It is a 25x80 pixel image with 8 classes. Color and noise variations of this image are generated each generation of the evolutionary process to evaluate the individuals in a more general and robust manner. For example, images (b) and (c) of Figure 3 are training images created from the ground truth.



(a) (b) (c)

Fig. 3. (a): Ground truth image. (b) and (c): Training images

The automata that have been evolved are made up of 20 rules. As each rule has 9 components and we have two additional parameters corresponding to the threshold angles, we end up with 182 real valued parameters for each automaton. A summary of the parameters used in the RVGA used for their evolution is displayed in Table I. It can be observed that the configuration of the algorithm is very simple and standard.

TABLE I. PARAMETERS OF THE RVGA USED IN THE EXPERIMENTS

GENETIC ALGORITHM PARAMETERS	
Type of parameters	Real
Number of parameters	182
Population size	182
Number of generations	50
Crossover method	Scattered
Crossover fraction	0.8
Elitism (number of individuals)	2
Mutation method	Gaussian
Selection method	Tournament (size = 4)
Stopping criterium	Maximum number of generations

Figure 4 displays the evolution of the fitness of the best individual and the average fitness of the population for a typical evolutionary process. The fluctuations that can be observed in the fitness curves are due to the fact that the evaluation of the individuals is carried out over different images each generation. However, the global decreasing trend of the error followed by the curves is clear.

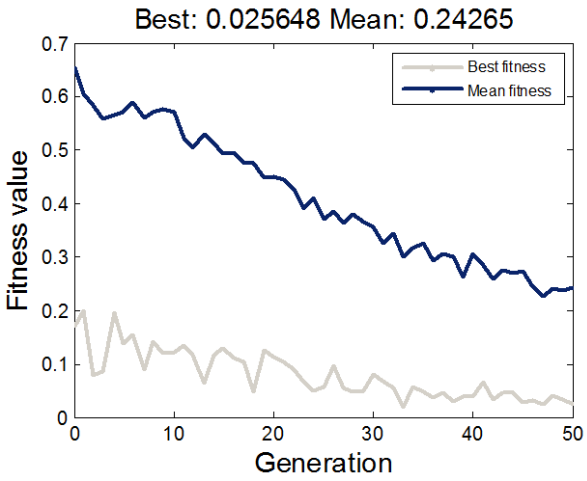
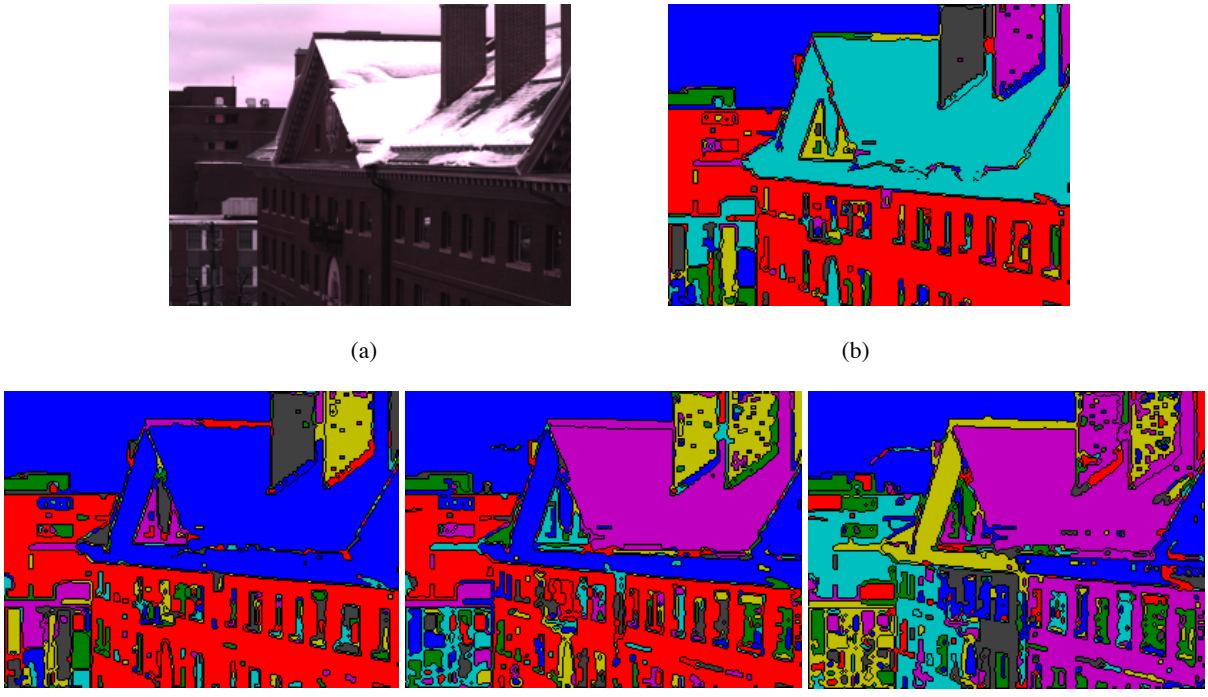


Fig. 4. Fitness function value (minimized) during a typical CA evolutionary process with the parameters of Table I.

The best individual after evolution has been chosen as the CA that will be used within the image segmentation processes that follow.

First of all, it must be pointed out that the two threshold parameters evolved together with the rules for the CA, **ε** and **Θ**, are very relevant parameters in terms of adjusting the segmentation detail level (number of areas that are segmented). This segmentation detail level is something that completely depends on the application and the user and cannot really be finely adjusted during the evolutionary process. Obviously, the way the training images are constructed, that is, the size of the features they contain and the types of borders found in them provide a coarse dimensionality for the segmentation detail level. Small details and many types of features lead to automata that segment smaller and more complex regions, larger primitives and a smaller number of them lead to CAs that produce larger segmented areas. However, to achieve a finer control over the segmentation detail level once the automata have been evolved in order to be able to adjust them to the particular images and desires of the user a scaling factor (S_f) parameter has been defined. This factor is introduced to scale the threshold angles of the CA, **ε**

and **Θ**, which are those provided by the evolutionary process based on the characteristics of the RGB training images.



(a) (b)

(c) (d) (e)

Fig. 5. Hyperspectral image (a) and the resulting images provided by the CA after evolution with different values of the scaling factor: (b) $S_f = 8$, (c) $S_f = 9$, (d) $S_f = 12$ and (e) $S_f = 14$

Thus, and just to show how the segmentation process works over hyperspectral images and the effects of changing S_f, in what follows we present the results of the application of the CA over a quite simple and recognizable real 31 band spectral image. Each band corresponds to a bandwidth of approximately 10nm from 420nm to 720nm [35]. Figure 5a displays one band of this image. It has been chosen due to the clear boundaries it contains, which allows the visual evaluation of the CA response. Figure 5b displays the result of applying the CA and the segmentation algorithm described in section 3.1 over image (a). Note that the image has been segmented but no

classification process has been carried out. This implies that the colors assigned as labels for each segmented area do not mean anything, they just help to visually discriminate the areas. That is, two areas with the same color do not necessarily contain the same material. Anyway, it can be observed that the result quite good, with most of the areas and boundaries of the original image clearly defined and the window areas with reflections generating many more segmented areas. This is a very interesting result because, as displayed in Figure 4, only 50 generations of evolution were required to obtain a CA that is able to operate with images whose dimensionality is much higher than that of those used in evolution.

The result shown in Figure 5 (b) was obtained with $S_f = 8$ in the CA. The next three images of Figure 5 display the segmentation results obtained after applying the same CA with $S_f = 9$ (c), $S_f = 12$ (d) and $S_f = 14$ (e). It can be clearly observed that, as the value of S_f increases, the number of areas segmented in the image increases too. Consequently, S_f is a very important parameter because it allows regulating the segmentation detail level of the process using the same CA. Thus, depending on the application or depending on the definition level of the ground truth, the CA response can be easily adjusted.

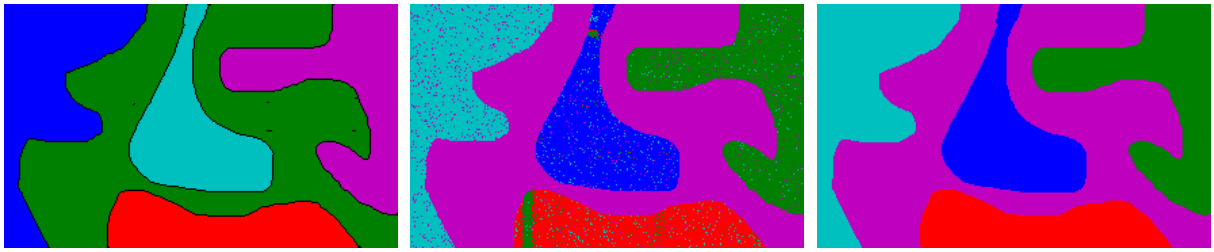
4.2 CA classification over synthetic images

To analyze the CA response in perfectly labeled and controlled cases, a set of synthetic images have been generated. They were constructed using five 64-band base spectra that were spatially corrupted by noise and artifacts. Figure 6 displays the result of applying the resulting CA and the segmentation and classification algorithms described in section 3 over one of these synthetic hyperspectral images containing 5 different classes. Image (a) shows the ground truth for the image. Image (b) corresponds to a 2D transformation of the synthetic 64-band hyperspectral image where each pixel corresponds to the angle between the spectrum of the pixel and a reference spectrum with all of its bands at the maximum value. Image (c) displays the result provided by the segmentation algorithm, that is, the segmented image with the areas arbitrarily labeled. Image (d) contains the output provided by

425 the SVM pixel-wise classification, that is, considering only spectral information. Finally, image (e) shows the final
426 classification obtained with the classification algorithm described in section 3.2.

427

428 (a) (b)



429

430 (c) (d) (e)

431 **Fig. 6.** (a): Ground truth. (b): 2D transformation of a synthetic 64 band hyperspectral image. Each pixel displays the angle
432 between the spectrum of the pixel and a reference spectrum with all of its bands at the maximum value. c): Output provided by
433 the segmentation algorithm. (d): SVM pixel-wise classification (e) Final classification

434

435 A simple visual analysis of these images shows that the final classification (e) and the ground truth (a) are very
436 similar, with the same areas clearly defined. Moreover, in the SVM pixel-wise classification image (d), one can
437 observe the spectral mixtures that are really in the image, and that must be processed considering the spatial
438 information. After iterating the CA 100 times over the synthetic image, the following results (Figure 6c) were
439 obtained:

440 • OA: 98,8 %

441 • AA: 99,3 %

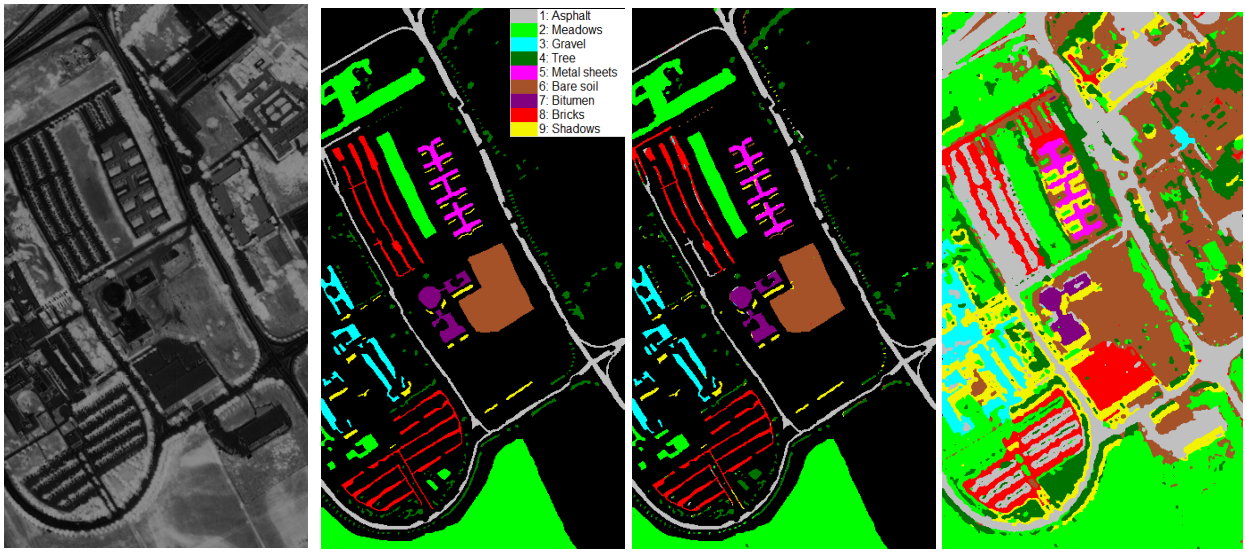
- Kappa: 98,4 %
- Class specific accuracies:
    - Class 1: 99,8 %
    - Class 2: 100,0 %
    - Class 3: 99,9 %
    - Class 4: 99,9 %
    - Class 5: 98,4 %

From these data, it can be concluded that the classification accuracy is very high and the preliminary successful results obtained in the previous section have been confirmed now using an image whose classification results are perfectly known. In fact, most of the small classification error present stems from pixels right on the borders, which is something that will have to be addressed in further iterations of the segmentation algorithm. Anyway, the most relevant result here is that from a spectrally quite inhomogeneous hyperspectral image as the one shown in Figure 6b the CA that was evolved using RGB training samples, is able to provide a very homogeneous pre-segmented image that is really easy to segment and classify.

4.3 CA classification over benchmark images

To compare the two-step algorithm presented in this work with other existing techniques, a benchmark hyperspectral real image has been selected. Specifically, that of the University of Pavia displayed in Figure 7. It is a 610x340 pixel image that contains 103 spectral bands ranging from 430 to 860 nm and where 9 different classes haven been defined (they are shown in Table II). Image (a) of Figure 7 displays a 2D angular transformation of the original hyperspectral image where each pixel corresponds to the angle between the spectrum of the pixel and a reference spectrum with all of its bands at the maximum value. Image (b) shows the ground truth with the 9 classes

that are considered. This image is only partially labeled and, even though many authors use it in order to perform comparisons; the labeling is at least dubious in some cases, especially in the areas corresponding to shadows.



(a) (b) (c) (d)

Fig. 7. (a): 2D angular transformation of the PAVIA image. (b): ground truth. (c): final image provided by the two-step algorithm considered here showing only the labeled areas. (d): final classification result provided by the algorithm for the whole image.

The resulting classification provided by the algorithm is that of images (c) and (d) of Figure 7. Image (c) corresponds to the final image where only the labeled areas are displayed while image (d) contains the classification of all the areas in the original image. Again, a simple visual comparison of images (b) and (c) shows the high degree of success that has been obtained, which is quantitatively confirmed by the class-specific accuracy contained in Table II. The main differences between the images are in very specific areas. For instance, it must be pointed out that the lower accuracy obtained for class 9 (shadows) is mainly because it is quite a confusing label, which does not really correspond to a specific spectrum but to several different ones with a lower definition.

To compare the overall classification results obtained with this image, three reference algorithms have been selected. First, and to provide reference values, we have we have included in the comparison a purely spectral

algorithm based on a pixel-wise SVM (data extracted from [12]). In addition, we have considered two spatial-spectral algorithms. The first one is the EMP (extended morphological profiles), originally presented by Plaza et al. in [36] and the second one is a watershed transformation-based algorithm presented by Tarabalka et al in [12] which has been labeled as W-RCMG in Table III. This algorithm considers a SVM based classification with majority vote where the watershed pixels are assigned to the regions with the closest median before the majority vote is performed (With a *WHEDS* approach) and which uses a Robust Color Morphological Gradient (RCMG) to obtain the gradient image. It is this second algorithm the one that provided the best results in the comparison the authors performed over the same image in [12].

TABLE II. CLASS-SPECIFIC ACCURACY (%) FOR THE UNIVERSITY OF PAVIA IMAGE

Area	**Class**	**OA(%)**
1	Asphalt	94,63
2	Meadows	98,92
3	Gravel	98,57
4	Tree	86,29
5	Metal sheets	95,17
6	Bare soil	99,66
7	Bitumen	96,62
8	Bricks	91,01
9	Shadows	84,16

The comparative overall classification results over the University of Pavia image are those of Table III. The ECAS approach proposed here using a SVM for classification obtains the best overall results in the three measures considered.

TABLE III. OVERALL ACCURACY (%) FOR DIFFERENT ALGORITHMS

	ECAS+SVM	Pixel-wise SVM	EMP	W-RCMG
Kappa	**94,99**	75,86	80,86	81,30
OA	**96,22**	81,01	85,22	85,42
AA	**93,89**	88,25	90,76	91,31

It is important to note here that, in addition to the good results the algorithm provides, since it is mostly a CA based algorithm its distribution over high speed GPU based platforms is quite simple thus allowing it to be used in real time applications which is not the case of the other algorithms.

4 Conclusions

Segmentation and classification is basic task that must be performed when processing hyperspectral images. Consequently, this task should be as efficient as possible in order to be able to reliably execute it in real-time when in the field. In this line, the use of Cellular Automata (CA) based structures for performing spatial-spectral operations over hyperspectral images is quite a promising approach given the fact that these structures are well adapted to their implementation in very efficient high performance processing hardware such as GPUs. Nonetheless, this approach poses the problem of how to determine the rule set for the CA to perform the task it has been assigned. In this paper we have shown that through evolution it can be quite simple to obtain these rule sets. In fact, the main problem in evolving the CAs within this realm is to provide appropriately labeled samples over which to test the candidate CAs during evolution since reliably labeled hyperspectral images are quite hard to find. This problem has been addressed in the approach presented here by adopting a dimension independent distance metric within the

operation of the CAs that permits evolving the automata using perfectly labeled synthetic images of much lower dimensionality. In addition, and with the objective of preserving the large amount of information provided by the spectra, the CAs work and modify the spectral information and do not project it onto a lower dimension. Thus, their output is really a hyperspectral image with different spectrally homogeneous regions.

The automata obtained this way were run over a typical hyperspectral benchmark image and the results they provided compared to other approaches presented in the literature. These results have shown that the approach is very competitive and, more importantly, that it permits adapting better to the way in which the user wants the segmentation to be performed than other more traditional approaches. This is achieved by choosing what types of features one includes in the training sets as well as through the use of a scale parameter that permits a finer tuning of the degree of segmentation desired by the user.

ECAS still has room for improvement, especially in terms of the classification of pixels on the borders between regions. At the same time an in depth study of the way the RGB training sets must be constructed in order to better adapt to the desires of the users in terms of segmentation detail would be of great interest. These are the areas where our work is currently focused.

Acknowledgments. This work was partially funded by the Spanish MICINN through project TIN2011-28753-C02-01 and the Xunta de Galicia and European Regional Development Funds through projects 09DPI012166PR and 10DPI005CT.

References

[1] D. L. Glackin and G. R. Peltzer, *Civil, Commercial and International Remote Sensing Systems and Geoprocessing*. American Institute of Aeronautics and Astronautics, 1999.

[2] D. Landgrebe, *Hyperspectral image data analysis*, vol. 19, no. 1. IEEE, 2002, pp. 17–28.

[3] L. Liu, B. Wang, and L. Zhang, "An approach based on self-organizing map and fuzzy membership for decomposition of mixed pixels in hyperspectral imagery," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1388–1395, 2010.

[4] F. L. Pena, J. L. Crespo, and R. J. Duro, "Unmixing low ratio endmembers through Gaussian synapse ANNs in hyperspectral images," in *2004 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications 2004 CIMSA*, 2004, vol. 59, no. 7, pp. 1834–1840.

[5] Z. Pan, G. E. Healey, M. Prasad, and B. J. Tromberg, "Hyperspectral face recognition for homeland security," in *Infrared Technology and Applications XXIX, Proceedings SPIE, V 5074*, 2003, pp. 767–776.

[6] A. De Juan, R. Tauler, R. Dyson, C. Marcolli, M. Rault, and M. Maeder, "Spectroscopic imaging and chemometrics: a powerful combination for global and local sample analysis," *TrAC Trends in Analytical Chemistry*, vol. 23, no. 1, pp. 70–79, 2004.

[7] M. Pesaresi and J. A. Benediktsson, *A new approach for the morphological segmentation of high-resolution satellite imagery*, vol. 39, no. 2. IEEE, 2001, pp. 309–320.

[8] A. A. Farag, R. M. Mohamed, and A. El-Baz, *A unified framework for MAP estimation in remote sensing image segmentation*, vol. 43, no. 7. 2005, pp. 1617–1634.

[9] O. Eches, N. Dobigeon, and J. Y. Tourneret, *Markov random fields for joint unmixing and segmentation of hyperspectral images*. 2010, pp. 0–3.

[10] G. Flouzat, O. Amram, and S. Cherchali, *Spatial and spectral segmentation of satellite remote sensing imagery using processing graphs by mathematical morphology*, vol. 4, no. c. Ieee, 1998, pp. 1–3.

[11] P. L. P. Li and X. X. X. Xiao, *Evaluation of multiscale morphological segmentation of multispectral imagery for land cover classification*, vol. 4, no. C. Ieee, 2004, pp. 0–3.

[12] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," *Pattern Recognition*, vol. 43, no. 7, pp. 2367–2379, 2010.

[13] P. Quesada-Barriuso, F. Argüello, and D. B. Heras, "Efficient segmentation of hyperspectral images on commodity GPUs," *Advances in KnowledgeBased and Intelligent Information and Engineering Systems*, vol. 243, pp. 2130–2139, 2012.

[14] N. Gorretta, J. M. Roger, G. Rabatel, V. Bellon-Maurel, C. Fiorio, and C. Lelong, *Hyperspectral image segmentation: The butterfly approach*. Ieee, 2009, pp. 1–4.

[15] A. Darwish, K. Leukert, and W. Reinhardt, *Image segmentation for the purpose of object-based classification*, vol. 3, no. C. Ieee, 2003, pp. 2039–2041.

[16] J. C. Tilton, *Analysis of hierarchically related image segmentations*, vol. 00, no. C. Ieee, 2003, pp. 60–69.

[17] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Hyperspectral Image Segmentation Using a New Bayesian Approach With Active Learning," *October*, vol. 49, no. 10, pp. 3947–3960, 2011.

[18] T. Veracini, S. Matteoli, M. Diani, and G. Corsini, "Robust Hyperspectral Image Segmentation Based on a Non-Gaussian Model," *Distribution*, pp. 192–197, 2010.

[19] R. Duro, F. Lopez-Pena, and J. Crespo, "Using Gaussian Synapse ANNs for Hyperspectral Image Segmentation and Endmember Extraction," in *Computational Intelligence for Remote Sensing*, Springer, 2008, pp. 341–362.

[20] M. Graña, I. Villaverde, J. O. Maldonado, and C. Hernandez, "Two lattice computing approaches for the unsupervised segmentation of hyperspectral images," *Neurocomputing*, vol. 72, no. 10–12, pp. 2111–2120, 2009.

[21] Z. Chuanwu, "Performance Analysis of the CPLD/FPGA Implementation of Cellular Automata," in *IEEE Trans. of The 2008 International Conference on Embedded Software and Systems Symposia (ICESS2008)*, 2008, pp. 308–311.

[22] D. B. Heras, F. Arguello, J. L. Gomez, J. A. Becerra, and R. J. Duro, "Towards real-time hyperspectral image processing, a GP-GPU implementation of target identification," in *Intelligent Data Acquisition and Advanced Computing Systems IDAACS 2011 IEEE 6th International Conference on*, 2011, vol. 1, pp. 316–321.

[23] V. N. J and B. Aw, "Theory of self-reproducing automata. Urbana: University of," *Illinois Press*, 1966.

[24] N. Ganguly, B. K. Sikdar, A. Deutsch, G. Canright, and P. P. Chaudhuri, "A survey on cellular automata," *Engineering*, pp. 1–30, 2003.

[25] N. H. Packard, "Adaptation toward the edge of chaos," in *Dynamic Patterns in Complex Systems*, vol. 212, J A S Kelso, A. J. Mandell, and M. F. Shlesinger, Eds. World Scientific, 1988, pp. 293–301.

[26] M. Mitchell, P. T. Hraber, and J. P. Crutchfield, "Revisiting the Edge of Chaos□: Evolving Cellular Automata to Perform Computations," *Complex Systems*, vol. 7, no. 2, pp. 89–130, 1993.

[27] R. Subrata and A. Y. Zomaya, *Evolving cellular automata for location management in mobile computing networks*, vol. 14, no. 1. 2003, pp. 13–26.

[28] W. Elmenreich and I. Fehérvári, "Evolving Self-organizing Cellular Automata Based on Neural Network Genotypes," *Network*, vol. LNCS 6557, pp. 16–25, 2011.

[29] D. Wang, N. M. Kwok, X. Jia, and G. Fang, "A Cellular Automata Approach for Superpixel Segmentation," in *Proc. Image and Signal Processing (CISP), 2011 4th International Congress on*, 2011, pp. 1108–1112.

[30] C. Kauffmann and N. Piché, "Seeded ND medical image segmentation by cellular automaton on GPU.," *International journal of computer assisted radiology and surgery*, vol. 5, no. 3, pp. 251–262, 2010.

[31] J. Gallego, C. Hernandez, and M. Grana, “A morphological cellular automata based on morphological independence,” *Logic Journal of IGPL*, Feb. 2011.

[32] M. A. Lee and L. M. Bruce, “Applying cellular automata to hyperspectral edge detection,” in *Proceedings 2010 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2010, pp. 2202–2205.

[33] A. R. Smith III, “Introduction to and Survey of Polyautomata Theory,” in *Automata Languages Development*, A. Lindenmayer and G. Rozenberg, Eds. North-Holland, 1976, pp. 405–422.

[34] C. Chang and C. Lin, “LIBSVM – a library for support vector machines, http://www.csie.ntu.edu.tw/%E2%88%BCcjlin/libsvm.” 2008.

[35] A. Chakrabarti and T. Zickler, *Statistics of real-world hyperspectral images*. IEEE, 2011, pp. 193–200.

[36] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, “Recent Advances in Techniques for Hyperspectral Image Processing,” *Remote Sensing of Environment*, vol. 113, no. July 2007, pp. S110–S122, 2009.