ECAS-II: A Hybrid Algorithm for the Construction of Multidimensional Image Segmenters

B. Priego, F. Bellas, R. J. Duro
Integrated Group for Engineering Research
Universidade da Coruña, Spain
{blanca.priego, fran, richard}@udc.es

***Abstract*—In this paper we describe a hybrid evolutionary-cellular automata based algorithm for the segmentation of multidimensional images, in particular hyperspectral images. This algorithm permits automatically generating the cellular automata transition rule set using as training set a group of appropriately generated synthetic RGB images, which greatly simplifies the process given the lack of adequately labeled hyperspectral images. In addition, different types of high dimensional segmentations can be obtained through the regulation of the parameters of the RGB images in the training set. The algorithm has been tested over synthetic and real hyperspectral images and the segmentation results it produces are very competitive when compared to other approaches found in the literature.**

Keywords—Hyperspectral image segmentation; cellular automata; evolutionary algorithm.

I. INTRODUCTION

Many different approaches have been proposed for the segmentation of multidimensional images, mainly medical and remote sensing images. However, most of the algorithms developed ignore the fact that an image can be segmented in different ways depending on the requirements of the user. One could, for instance, want to segment a remote sensing hyperspectral image into areas corresponding to fields, roads and water (rivers, sea) or into areas corresponding to the different products within the fields or differentiate between rivers and the sea. Most algorithms cannot be regulated to adapt to the user desires and contemplate a single way of segmenting the images. This is because it is a very complex and nonlinear process based on where the decision boundaries that determine what is spectrally similar and what is not are set within images. Take into account that multidimensional similarity is not a straightforward concept.

In this paper we are going to use the case of hyperspectral imaging as a typical example of these types of multidimensional imaging applications. Hyperspectral images provide hundreds of values for each pixel that correspond to the intensity of a large number of bands within the visible and near infrared spectrum. They are typically made up of two spatial dimensions of anywhere from 250 pixels to a few of thousand pixels and a spectral dimension of up to a thousand bands per pixel. This usually leads to extremely large data sets due to the very high levels of spectral and spatial detail they contain.

Most approaches in the literature that consider the hyperspectral image segmentation problem are basically extensions of more traditional image processing techniques, that were developed for images of much lower dimensionality (gray level or RGB images) such as region-merging [1] or hierarchical segmentation [2]. Some use fixed window based neighborhoods, which reduces their flexibility. This is the case of morphological levels [3] or Markov random fields [4][5]. Others make use of mathematical morphology based techniques [6][7] or watershed based algorithms [8][9] and try to extend them to this domain although the ordering problem that one finds in high dimensional spaces makes the modification of the operators involved quite a complex and still not fully solved problem. All of these techniques, as indicated above, provide very little adaptability to changes in the way the user wants to segment the image.

Additionally, due to their origin, most of these methods do not preserve the multidimensional character of the signals throughout the segmentation process. They usually perform an early projection of the multidimensional information onto a two dimensional representation with the consequent loss of the large amount of spectral information these images provide.

From another point of view, algorithms that require training such as those based on Artificial Neural Networks, or evolved automata, that is, which use training samples in order to learn to segment images, intrinsically have the capacity of adapting to what the user requires through an appropriate selection of the training samples. This way the system can learn to perform the segmentation, presumably in an appropriate way, from sets of sample labeled images. Examples of this approach are those presented in [10], [11] or [12]. However, they usually require complex algorithms that lead to computationally intensive processing stages that are very hard to implement in limited computing resources, which is usually the case in many applications where hyperspectrometers are used in the field. Additionally, there is the perennial problem of multidimensional and, in particular, hyperspectral imaging: Obtaining and labeling the images is quite costly leading to very few labeled images being available and they are not always too reliable in terms of their labels. Thus, it is quite hard to construct a training set that really reflects what the user wants.

Consequently, it is necessary to provide methodologies that allow this processing to be achieved without having to resort to large training or labeled sets of real images. In addition, and 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. Thus, instead of resorting to complex labeled real hyperspectral images, which

are usually costly and not very trustworthy, we will generalize the segmentation function so that the algorithms can be trained using synthetic RGB images and then directly applied to hyperspectral images. This will be achieved through the use of a spectral distance measure that is independent from the dimensionality of the data.

Additionally, and to provide a way to address the high computational cost issues that working with such high dimensional datasets induces, we have considered the creation of algorithms based on structures that can be easily processed in a concurrent fashion within hardware such as GPUs. To this end, the concept of Cellular Automata (CAs) seems very promising. A series of authors in the field of segmenting low dimensional images such as [13] [14] [15], and some in the high dimensional image processing area, such as [16] who dealt with edge detection, 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, as we have already indicated, when multidimensional images are considered, they are usually projected onto a lower dimension and it is over this dimension that the whole segmentation process takes place.

Here we propose using a hybrid two-step algorithm in which the first step makes use of Evolved Cellular Automata in order to create spectrally representative homogeneous regions, that is, segment the images, which are then classified using a more traditional SVM algorithm. This algorithm has been called ECAS-II (Evolutionary Cellular Automata based Segmentation) and it is an evolution of ECAS [17], which was a first prospective version of this approach with very simple rule encodings that only considered a direct neighborhood. ECAS-II is a much more powerful and efficient algorithm.

To address the issues related to the training sets, and as a first approximation, we have established a methodology for the production of synthetic low dimensional training sets that can be used to train structures for the segmentation of high dimensional images. Thus, on one hand, the training cost is reduced by using low dimensional images, on the other, the problem of the number of training images is addressed by providing a method to automatically produce synthetic images that can be used to train segmentation algorithms to provide a particular type of segmentation.

The rest of the paper is structured as follows. In section 2 we introduce the operation and training of the ECAS-II modules. Section 3 is devoted to the presentation of a series of experiments of the application of ECAS-II over synthetic and real images and its comparison to other approaches. Finally, section 4 presents some conclusions.

II. SEGMENTATION ALGORITHM

A. Cellular Automata

For this CA, the cells are located over each pixel of the hyperspectral image and the states of the cells ($\boldsymbol{s_i}$) are given by an N band spectrum that takes values in [0 1]. Consequently, the state space is continuous and corresponds to $\mathbb{R}^N$. The automata is applied iteratively to the whole hyperspectral cube, gradually modifying the state or spectrum of each pixel of the image. This way, there is no projection of the multidimensional spectral information onto lower dimensionalities.

How a cell is modified each iteration depends on its spectrum, that of the $N_{Smax} \times N_{Smax}$ neighboring cells, where N_{Smax} is the maximum size considered for the spatial window centered over each cell, and on the set of transition rules that control the behavior of the CA. These rules will be automatically obtained to reflect the type of segmentation a user wants through an evolutionary process.

As a first step, in order to be able to compare the state or spectrum of neighboring cells it is necessary to establish a spectral distance measure. Here we have chosen the well-known spectral angle (normalized between 0 and 1). This measure is independent of the number of spectral components of the image and it is invariant to changes in illumination. This permits using the same CA over images of different dimensionality. Thus, for a cell i, the normalized spectral angle, α_{ij} with respect to its neighboring cell j is defined as:

$$\alpha_{i,j} = \frac{2}{\pi}\cos^{-1}\left(\frac{\sum \boldsymbol{s_{ij}} \boldsymbol{s_i}}{\sqrt{\sum \boldsymbol{s_{ij}^2}}\sqrt{\sum \boldsymbol{s_i^2}}}\right) \qquad (1)$$

To decide on the application of a particular transition rule, an automata needs information on its neighborhood. The spectral angle is a rather low-level information source and here we propose using it for this by calculating spatial gradients from the spectral angles of the cell and its neighbors. More specifically, these gradient vectors are calculated taking into account the pixels contained in three different two dimensional windows of size $N_S \times N_S$, where $N_S = \{3, 5, 7\}$. This implies that neighborhood information is obtained at three different levels using the closest 8, 24 and 48 neighboring cells. These vector gradients provide information on the intensity and direction in which spectral changes in the image occur and these spectral variations determine the differentiation among regions. These gradient vectors are obtained using two two-dimensional masks $M_{X_{NS}}$ and $M_{Y_{NS}}$ which are applied to each cell. Thus, the horizontal and vertical components of the gradient vectors corresponding to cell i, denoted as $G_{X_{NS_i}}$ y $G_{Y_{NS_i}}$, are:

$$G_{X_{NS_i}} = \sum_{j=1}^{N_S \cdot N_S} \alpha_{i,j} \cdot M_{X_{NS_j}} \qquad (2)$$

$$G_{Y_{NS_i}} = \sum_{j=1}^{N_S \cdot N_S} \alpha_{i,j} \cdot M_{Y_{NS_j}} \qquad (3)$$

where $M_{X_{NS_j}}$ and $M_{Y_{NS_j}}$ represent the j^{th} elements of the gradient masks $M_{X_{NS}}$ and $M_{Y_{NS}}$.

These three gradient vectors, $\boldsymbol{G_{NS_i}}$, where $N_S = \{3, 5, 7\}$ can then be composed and expressed as a modulus value, $\left|\boldsymbol{G_{N_{S_i}}}\right|$, and an angle value, $\phi_{N_{S_i}}$,:

$$\left|\boldsymbol{G_{N_{S_i}}}\right| = \sqrt{G_{X_{NS_i}}^2 + G_{Y_{NS_i}}^2} \qquad (4)$$

$$\phi_{N_{S_i}} = \tan^{-1}\left(\frac{G_{Y_{NS_i}}}{G_{X_{NS_i}}}\right) \qquad (5)$$

The modulus, $\left|\boldsymbol{G}_{N_{S_i}}\right|$, is related to the intensity of the spectral change following the direction given by the angle, $\phi_{N_{S_i}}$, for window size N_{S_i}. In this algorithm, we consider the values of the three modules and angles of the gradient vector as all the information the CA needs to operate over a cell of the hyperspectral cube.

As a result, the set of transition rules that control the behavior of the CA consists of M rules, each one of them made up of 6 parameters:

$$CA = \begin{Bmatrix} \left|\boldsymbol{G}_{r3_1}\right| & \left|\boldsymbol{G}_{r5_1}\right| & \left|\boldsymbol{G}_{r7_1}\right| & \phi_{r5_1} & \phi_{r7_1} & \theta_{r_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \left|\boldsymbol{G}_{r3_k}\right| & \left|\boldsymbol{G}_{r3_k}\right| & \left|\boldsymbol{G}_{r3_k}\right| & \phi_{r5_k} & \phi_{r7_k} & \theta_{r_k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \left|\boldsymbol{G}_{r3_M}\right| & \left|\boldsymbol{G}_{r3_M}\right| & \left|\boldsymbol{G}_{r3_M}\right| & \phi_{r5_M} & \phi_{r7_M} & \theta_{r_M} \end{Bmatrix} \quad (6)$$

Each iteration of the CA, only one of these M rules is applied over each cell. To decide which, we compare the neighborhood information of the pixel ($\left|\boldsymbol{G}_{3_i}\right|, \left|\boldsymbol{G}_{5_i}\right|, \left|\boldsymbol{G}_{7_i}\right|, \phi_{3_i}, \phi_{5_i}, \phi_{7_i}$) to the first 5 parameters of each of the M rules ($\left|\boldsymbol{G}_{r3_k}\right|, \left|\boldsymbol{G}_{r5_k}\right|, \left|\boldsymbol{G}_{r7_k}\right|, \phi_{r5_k}, \phi_{r7_k}, \phi_{r_k}$) and choose the rule that is closest. Only 5 parameters are used because the representation has been chosen so that the three gradient vectors as a group are independent from rotations and reflections. For this reason, parameter ϕ_{r3_k} is taken as zero and is not included as a parameter of the rule.

The distance between the neighborhood vector and the vectors that represent a rule, which is calculated as the sum of the L^2 norms of the vector differences once the vectors have been rotated and reflected will be denoted as d_{ik}:

$$d_{ik} = \left\|\boldsymbol{G}_{3_i} - \boldsymbol{G'}_{r3_k}\right\| + \left\|\boldsymbol{G}_{5_i} - \boldsymbol{G'}_{r5_k}\right\| + \left\|\boldsymbol{G}_{7_i} - \boldsymbol{G'}_{r7_k}\right\| \quad (7)$$

where $\{\boldsymbol{G}_{3_i}, \boldsymbol{G}_{5_i}, \boldsymbol{G}_{7_i}\}$ are the gradient vectors of the cell under evaluation and $\{\boldsymbol{G'}_{r3_k}, \boldsymbol{G'}_{r5_k}, \boldsymbol{G'}_{r7_k}\}$ those of the gradient vectors representing rule k after rotating them an angle ϕ and, if necessary, reflecting them.

Summarizing, the process of selecting a rule consists in calculating the minimum distance d_{ik} for each rule and selecting the rule, s, that provides the lowest value for the distance, $s = \underset{k\in\{1,2,\dots,M\}}{\operatorname{argmin}}\ d_{ik}$.

The last parameter of the selected rule contains the information required to update the state of cell i. The state of the cell will be modified by performing a weighted average of its state or spectrum and that of some of its neighbors. If we follow the direction given by $\phi + \theta_{rs}$ for a distance of 1 pixel, where ϕ is the rotation angle of the selected rule that was used to calculate the distance d_{is}, and θ_{rs} is the last parameter of rule s, we will be at a point of the image, P_i. The neighboring cells used to modify the spectrum of cell i will be located at a maximum distance of one pixel from P_i .

Thus, the update of the state of cell i, s'_i, will be given by the weighted average of the spectrum of those cells and that of i:

$$\boldsymbol{s'_i} = \sum_{j=1}^{n} w_{ij} \cdot \boldsymbol{s_{ij}} + w_i \cdot \boldsymbol{s_i} \quad (8)$$

$$w_{ij} = \frac{f_r(r_j)}{\sum_{j=1}^{n} f_r(r_j) + f_{th}};\ w_i = \frac{f_{th}}{\sum_{j=1}^{n} f_r(r_j) + f_{th}} \quad (9)$$

$$f_r(r_j) = \begin{cases} f_{th}, & \frac{1}{r_j} > f_{th} \\ \frac{1}{r_j}, & \frac{1}{r_j} \le f_{th} \end{cases} \quad (10)$$

$\boldsymbol{s'_i}$ is the updated spectrum of cell i; $\boldsymbol{s_i}$ represents the original spectrum of cell i; $\boldsymbol{s_{ij}}$ is the spectrum of cell j which is a neighbor of cell i; n is the number of neighboring cells that will participate in the update of the state of cell i; r_j is the distance between the neighboring cell j and point P_i; w_{ij} is the weight associated to the spectrum of cell j; w_i is the weight associated to the spectrum of cell i; $f_r(r_j)$ is a function that assigns weights as a function of distance r_j.

Following this procedure, the CA is applied iteratively to the whole image producing a new hyperspectral cube every iteration, preventing this way any projection onto lower dimensions. With the appropriate rules, the final hyperspectral cube will be a segmented version of the original, where each region will be represented by a narrow range of spectra.

B. Evolving and training the CAs

To select the rules that will control the behavior of the cellular automata, we have chosen an automatic optimization process in the form of an Evolutionary Algorithm (EA). In particular, for this implementation the algorithm we have chosen is Differential Evolution (DE) [18].

Regarding the encoding of the individuals, in this case, the set of transition rules that make up the CA is encoded as a vector of $6 \cdot M$ floating point values. The values corresponding to $\left|\boldsymbol{G}_{r3_k}\right|, \left|\boldsymbol{G}_{r5_k}\right|$ and $\left|\boldsymbol{G}_{r7_k}\right|$ belong to $[0, \sqrt{2}]$ and those corresponding to ϕ_{r5_k}, ϕ_{r7_k} and θ_{rk} are in the interval $[0, 2\pi]$.

The fitness function to be minimized by DE is defined as the maximum value of two error measurements: intra-region (e_{intra}) error and inter-region (e_{inter}) error:

$$e = \max(e_{intra}, e_{inter}) \quad (11)$$

Let us introduce some notation in order to describe these error measurements. Let $\mathcal{I} = \{x_p\}_{p=1}^{P\times Q}$ be a synthetic RGB image of size $P \times Q$ where x_p is a 3-dimensional vector. Let $\mathcal{L} = \{l_p\}_{p=1}^{P\times Q}$ be the ground truth of this synthetic RGB image, with $l_p \in S$, being $S = \{1, 2, \dots, s_k, \dots, M\}$ the set of different labels in $\mathcal{L}$. Let $\mathcal{R} = \{r_p\}_{p=1}^{P\times Q}$ with $r_p \in \{0,1\}$ be a binary image that characterizes each pixel of $\mathcal{L}$ as a border pixel, $\{0\}$, or an internal one, $\{1\}$. We can then also consider $\mathcal{I}$ the merging of regions with the same associated label, $\mathcal{I} = \bigcup_{k=1}^{M} B_k$, where $B_k = \{x_p \mid l_p = s_k\}$. B_k can also be divided into two subsets, one that only contains interior pixels and another that only

contains border pixels, taking into account the labeling in $\mathcal{L}$. This way, $B_k = I_k \cup F_k$, where $I_k = \{x_p \mid l_p = s_k \wedge r_p = 1\}$ and $F_k = \{x_p \mid l_p = s_k \wedge r_p = 0\}$. F_k may be decomposed into $M-1$ subsets, $F_k = \bigcup_{k'=1}^{M} F'_{kk'}$ with $k' \neq k$, where $F'_{kk'}$ contains the pixels located in the border between B_k and $B_{k'}$.

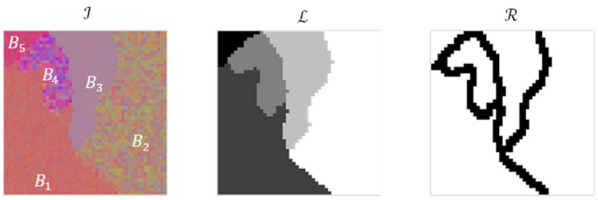

Fig.1 Synthetic RGB image (left), ground truth (center) and border image (right)

With this notation in mind, we can define the two types of errors. The intra-region error provides a measure of the homogeneity of the regions in the image, being a region a set of pixels that share the same label in the ground truth associated to the training image. This error is calculated as the maximum of two errors: the local intra-region error ($e_{local-intra}$) and the non-local intra-region error ($e_{nonlocal-intra}$):

$$e_{intra} = \max(e_{local-intra}, e_{nonlocal-intra}) \quad (12)$$

Being the local intra-region error defined as:

$$e_{local-intra} = \frac{\sum_{k=1}^{M}\left[\sum_{x_p \in I_k}\left(\sum_{j=1}^{8}\frac{\alpha_{pj}}{8}\right)\right]}{\sum_{k=1}^{M}\#I_k} \quad (13)$$

where α_{pj} refers to the normalized spectral angle between a pixel x_p and its neighbor x_j and $\#I_k$ is the cardinality of subset I_k. $e_{local-intra}$ measures the local homogeneity of neighboring pixels that share the same label in $\mathcal{L}$. The local homogeneity for each I_k is defined as:

$$H_k = \frac{\sum_{x_p \in I_k}\left(\sum_{j=1}^{8}\frac{\alpha_{pj}}{8}\right)}{\#I_k} \quad (14)$$

On the other hand, to define the $e_{nonlocal-intra}$, it is necessary to randomly select a set of V pairs of pixels belonging to each subset I_k, $\{x_{kv}, y_{kv}\} \in I_k$ with $v = \{1,2,\ldots V\}$. Thus, $e_{nonlocal-intra}$ is defined as:

$$e_{nonlocal-intra} = \frac{\sum_{k=1}^{M}\left[\sum_{v=1}^{V}\frac{\alpha_{kv}}{V}\right]\cdot \#I_k}{\sum_{k=1}^{M}\#I_k} \quad (15)$$

where α_{kv} denotes the normalized spectral angle between x_{kv} and y_{kv}. $e_{nonlocal-intra}$ measures the spectral homogeneity between pixels that belong to the same region but that are not necessarily neighbors.

Finally, the inter-region error e_{inter}, provides a measure of the dissimilarity of the different regions that have been segmented in the image. In order to calculate e_{inter}, we randomly select a set of U pairs of pixels for each $F'_{kk'} \cup F'_{k'k}$, $\{x_{ku}, y_{k'u}\}$, with $x_{ku} \in F'_{kk'}$, $x_{k'u} \in F'_{k'k}$, y $v = \{1,2,\ldots V\}$. The inter-region error, e_{inter}, is thus defined as:

$$e_{inter} = \frac{\sum_{k=1}^{M}\left[\sum_{k'=1}^{M}\frac{\sum_{u=1}^{U} f\left(\alpha_{kk'u},\ H_k,\ H_{k'}\right)\cdot \#B_{k'}}{U\cdot\sum_{k'=1}^{M}\#B_{k'}}\right]\cdot \#B_k}{\sum_{k=1}^{M}\#B_k} \quad (16)$$

$$f(\alpha_{kk'u}, H_k, H_{k'}) = \begin{cases} 1 & if \quad \alpha_{kk'u} \leq H_k + H_{k'} \\ 0 & if \quad otherwise \end{cases} \quad (17)$$

where $\alpha_{kk'u}$ denotes the normalized spectral angle between x_{ku} and $y_{k'u}$.

Every time the CA is evaluated, it is executed over a training RGB image for a given number of iterations and the result is compared to the ground truth. In order to improve the evaluation process maximizing the generalization capabilities of the CA and at the same time preventing circularity problems, each individual is evaluated using different images that can be either generated online or randomly selected from a pre-constructed RGB image collection.

As indicated above, the application of the CA over an image is completely independent from its dimensionality due to the use of the spectral angle as the similarity measure between adjacent cells. We have found it interesting to exploit this feature in order to accelerate and simplify the evolutionary process. Thus, due to the similarity measure, the CA will be valid to segment low or high dimensional images (for instance RGB and hyperspectral images). Therefore, a CA whose rule set is adjusted using low dimensional images (RGB) will be completely effective when executed over multi/hyperspectral images as long as the problem presents the same type of general features. In this case, we can say that as long as the user desires the same type of segmentation.

C. *Final classification*

To produce a provisional classification image, a SVM based pixel wise classification is applied to the ECAS-II segmentation result following the work of Tarabalka et al. [9]. 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 used was the one found in the LIBSVM library [19], using the C and γ parameter values of [9]: C = 128 and γ =0.125. These parameters will be considered constant for all the experiments presented in the application and results section.

III. Some Experiments

This section is organized as follows: in the first part, the capabilities of the CA are demonstrated with regards to how appropriately evolved CAs can adapt to different levels of segmentation, employing for this purpose synthetic hyperspectral images. Afterwards, a benchmark hyperspectral real image has been selected to compare the segmentation performance of the ECAS-II algorithm to that of other existing techniques.

A. *CA evolution and adaptability to segmentation casuistry*

The CAs that have been evolved following the description in section II are made up of 30 rules. Each rule has 6 components, making a total of 180 real valued parameters for each automaton.

A summary of the DE used for their evolution is displayed in Table I.

TABLE I. DIFFERENTIAL EVOLUTION (DE) PARAMETERS

DE parameters	
Type of parameters	Real
Number of parameters	180
Population Size	100
Number of generations	50
Crossover	0.7
Mutation	0.8
Stopping criterion	Max. number of generations or Min. fitness error
Minimum fitness error	1e-6

For this experiment, four different CAs, denoted as CA #1, CA #2, CA #3 and CA #4, have been evolved. Each evolution differs from the others in the set of RGB images used in the evaluation process. We will show below that the proper selection of the dataset for the evolutionary procedure will directly affect the kind of segmentation obtained when the CA is applied over RGB or hyperspectral images.

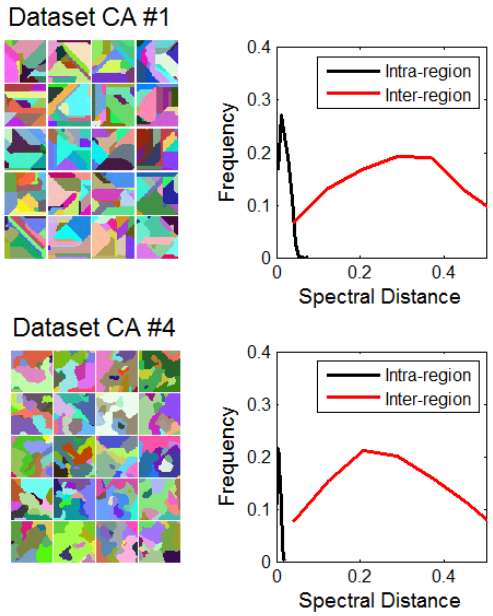

Fig. 2. 20th band of a synthetic 64-band image (top left). Histogram of spectral distances between neighboring pixels in the original 64-band synthetic image (top right). Samples of the RGB data set used to evolve CA #1 (middle left). Intra-region (black line) and inter-region spectral distances (red line) histograms from samples belonging to the evolution of CA #1 (middle right). The same for CA #4 in the bottom images.

In order to easily control the creation of the RGB dataset used to evolve the CA, we have developed a synthetic image generation tool that uses some parametrized spectral and spatial information as inputs and produces an RGB image that fulfills those features. This tool can also be used to create synthetic hyperspectral images based on real spectra obtained from public hyperspectral repositories. The spatial features are based on measurements of the density of regions existing in the image and of the nature of the borders that separate contiguous areas in terms of their linearity. The spectral information is related to the spectral homogeneity between pixels that belong to the same region (intra-region spectral distances) and between pixels from different regions (inter-region spectral distances). The middle and bottom left images of Fig. 2 present some samples of the RGB image dataset employed for the evolutionary processes to obtain the evolved CA #1 and CA #4. On the other hand, the middle and bottom right images of Fig. 2 show information about the spectral features, specifically, the histograms of the spectral distances (intra and inter-region) of the RGB dataset used for evolving CA#1 and CA#4 respectively.

We will show below that the distribution of the intra-region spectral distances affects the segmentation result. That is, a RGB image dataset with a wider range of intra-region spectral distances will produce a CA that once evolved and applied to a RGB or hyperspectral image will result in a segmentation that contemplates less regions. On the other hand, a dataset that exhibits a narrow intra-region spectral distance distribution will result in an evolved CA that produces more finely segmented images. Also, the proximity between the distributions of intra-region and inter-region spectral distances in the RGB dataset also conditions the segmentation result after the application of the CA. The closer these distributions are, the less detailed that segmentation of the resulting image after the application of the evolved CA.

As experimental dataset in this subsection, we have created a synthetic hyperspectral image set where the images contain 64 spectral bands (Fig. 3). A total of 12 different spectral endmembers have been noised and blended taking into account spatial considerations. As a representation of the spectral relation among neighboring pixels, the right graph of Fig. 3 shows the histogram of the local spectral distance between pixels belonging to the same region.

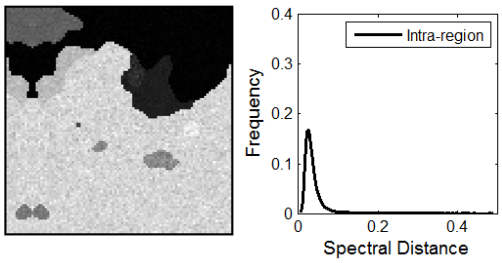

Fig. 3. 20th band of a synthetic 64-band image (left). Histogram of spectral distances between neighboring pixels.

In the first place, to better understand how the CA operates over the images, Fig. 4 shows the evolution of the pixel spectra in the application of CA #2 to the synthetic hyperspectral image shown in Fig. 3. As indicated in the introduction, the CA-based segmentation process consists in the gradual modification of the cell spectra until all pixels converge to some spectrum clusters. The top left graph of Fig. 4 shows the original pixel spectra before CA #2 is applied. The other graphs of Fig. 4 represent the pixel spectra for different iterations in the CA application process. On the 150th iteration, it is clear that the pixel spectra from the original synthetic image have been modified, making the detection of several spectrum clusters easier.

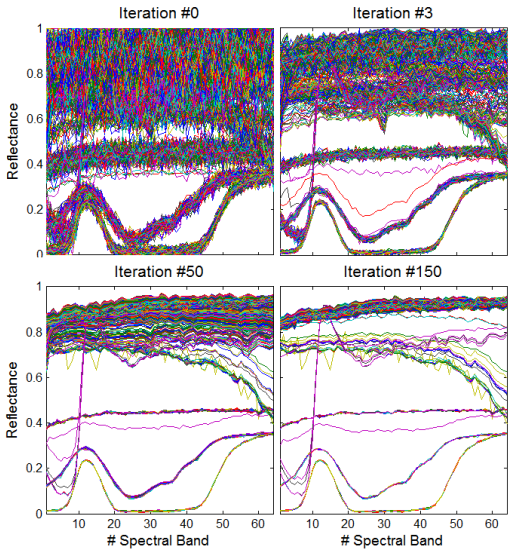

Fig. 4. Evolution of the pixel spectra in the application of CA #2 on the synthetic hyperspectral image shown in Fig. 2.

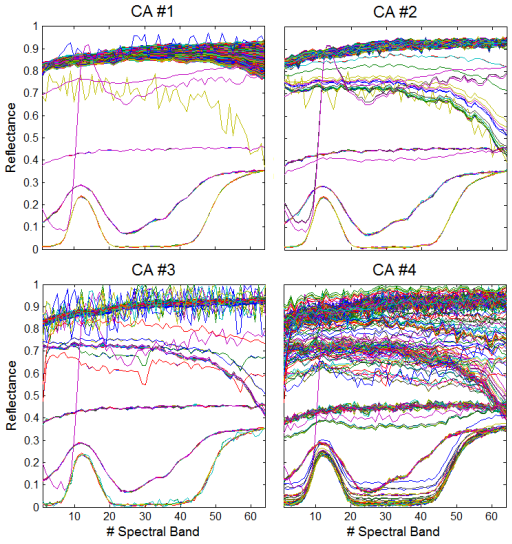

Fig. 6. Pixel spectra after 200 iterations in the application of 4 different CAs to the synthetic image shown in Fig. 2.

Secondly, the four evolved CAs considered in this section have been applied to the example synthetic image. The segmentation results after 200 iterations in the CA process are shown in Fig. 5. It can be observed that each CA solves the segmentation task differently, merging or preserving spectrum clusters, leading to segmentations with different levels of detail.

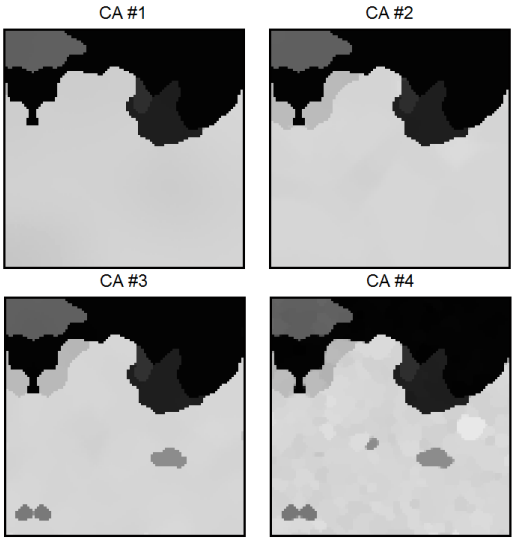

Fig. 5. 20th band of the segmented synthetic image using CA #1, CA #2, CA #3 and CA #4.

Fig. 6 shows the corresponding pixel spectra of the segmentations obtained in Fig. 5 after 200 iterations of the CAs. These graphs also exhibit the convergence of the original cell spectra to a different number of spectrum clusters depending on the specific CA that has been applied to the example image.

As it can be observed in Fig. 5, CA #1 is the CA that produces a segmentation result with a smaller number of clusters, while in contrast, CA #4 is the CA that outputs a more detailed segmentation.

The results of this section support and validate the main goal of this work, that is, to establish a technique able to generalize the segmentation process, adapting the task to particular levels of segmentation. We have shown that training the algorithm using simple synthetic RGB image datasets provides CAs that directly applied to multi-dimensional images produce satisfactory segmentations with different levels of detail as required.

Having tested the segmentation capabilities of ECAS-II, in the next subsection, the proposed algorithm is applied to a real hyperspectral image, aiming to verify its segmentation behavior over real images and to compare its performance to that of other methods extracted from the literature.

B. Application of ECAS-II to real hyperspectral images

To compare the performance of ECAS-II to that of other existing techniques, a classification task over a benchmark real hyperspectral image has been carried out. The selected image is the University of Pavia image displayed on the left image of Figure 14. 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). The left image of Figure 13 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. The left image of figure 16 shows the ground truth with the 9 classes that are considered.

The classification is performed by applying a SVM based pixel wise algorithm to the output of the ECAS-II algorithm, that is, the segmented image. The evolved CA used in this section is the one labeled as CA #2. The resulting classification provided by the algorithm is presented on the right images of Fig. 8 and 9. In Fig. 9 only the labeled areas are displayed while in Fig. 8 the

classification is shown for all the pixels of the image. The left image of Fig. 8 presents the classification obtained after applying the SVM to the original hyperspectral image. The success obtained in the classification is qualitatively confirmed by visual inspection and quantitatively in terms of class-specific accuracy (Table II), overall accuracy (OA), average accuracy (AA), and kappa coefficient (Table III).

Fig. 7 2D angular transformation of the original PAVIA image (left) and 2D angular transformation after the application of ECAS-II (right).

To compare the overall classification results obtained over this image, four reference algorithms have been selected. The first algorithm for comparison is a purely spectral algorithm based on a pixel-wise SVM [9], which corresponds to the classified pixels of the left image in Fig. 8. The second algorithm is the W-RCM algorithm, a watershed transformation-based algorithm presented by Tarabalka et al in [9]. This method consist in the application of a watershed to the result of a robust color morphological gradient (RCMG), which projects the information to one band. Then, the classification result is obtained through a majority vote (MV) process combining the spatial and spectral results. The third reference algorithm is the EMP (extended morphological profiles) algorithm, originally presented by Plaza et al. in [16]. Finally, the last reference technique is a modified V-ELM-1 [20]. This last method is based on extreme learning machines, including a regularization in the pixel-wise classifier using a connectivity of 8 neighbors and a spatial processing by watershed.

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

Area	Class	OA(%)
1	Asphalt	94,95
2	Meadows	97,26
3	Gravel	93,90
4	Tree	98,14
5	Metal sheets	99,78
6	Bare soil	98,11
7	Bitumen	99,17
8	Bricks	99,02
9	Shadows	86,48

The accuracy results for the mentioned methods are shown in Table III, and the ECAS-II technique followed by a SVM for classification is the algorithm that provides the best solution in terms of the three measures considered.

Fig. 8 SVM based pixel classification applied to the original hyperspectral image (left) and to the segmented image provided by the ECAS-II algorithm (right)

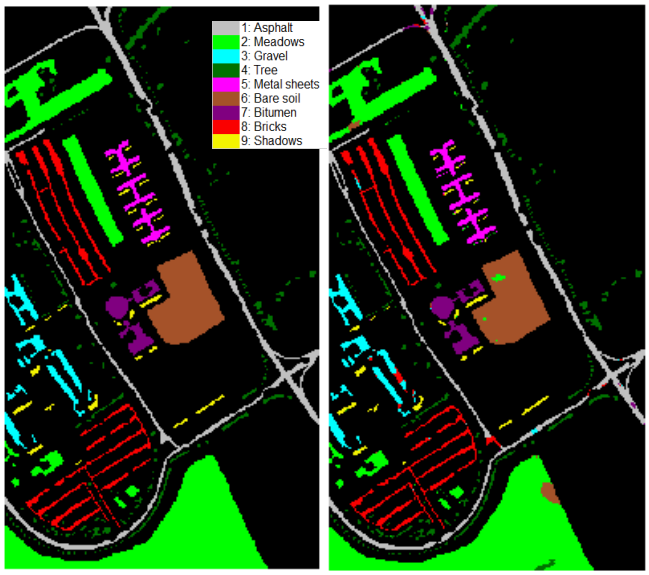

Fig. 9 Ground truth (left) and SVM based pixel classification applied to the segmented image provided by the ECAS-II algorithm showing only the labeled areas (right)

TABLE III. ACCURACY RESULTS (%) FOR DIFFERENT ALGORITHMS

	Kappa	OA	AA
ECAS-II + SVM	**95,97**	**96,95**	**96.31**
Pixel-wise SVM [9]	75,86	81,01	88,25
W-RCMG [9]	81,30	85,42	91,31
EMP [16]	80.86	85.22	85.42
V-ELM-1 [20]	95.00	96.66	95.92

IV. Conclusions

In this work we have presented an approach based on evolutionary algorithms and cellular automata that allows to efficiently segment multidimensional images. This approach has been demonstrated to be very competitive in terms of performance when compared to other proposals in the literature. In addition, it permits regulating how the segmentation is to be carried out and, more importantly, it does not require a large set of labeled multidimensional images, which is usually very difficult to obtain. In fact the algorithm is trained using synthetic RGB images, which are very easy to produce.

References

[1] 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.

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

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] B. Priego, D. Souto, F. Bellas, R. Duro, "Hyperspectral Image Segmentation through Evolved Cellular Automata", Pattern Recognition Letters, pp 1648 - 1658, 2013

[18] Storn, R. and Price, K.: Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, **11**(4) 1997, 341–359.

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

[20] Lopez-Fandino, J.; Quesada-Barriuso, P.; Heras, D.; Arguello, F., "Efficient ELM-Based Techniques for the Classification of Hyperspectral Remote Sensing Images on Commodity GPUs," Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of , vol. PP, no. 99, pp. 1-10, 2015