Advances in Knowledge-Based and Intelligent Information and Engineering Systems
M. Graña et al. (Eds.)
IOS Press, 2012

doi: 10.3233/978-1-61499-105-2-2160

Unsupervised Segmentation of Hyperspectral Images through Evolved Cellular Automata

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

Grupo Integrado de Ingeniería,
Universidade da Coruña, Spain
{blanca.priego, dsouto, fran, richard}@udc.es

Abstract. The problem of segmenting multidimensional images, in particular hyperspectral images, is still an open subject. The main issue is related to preserving the multidimensional character of the signals throughout the segmentation process avoiding an early projection onto a 2D plane with the consequent loss of the wealth of information these images provide. The approach followed here is based on the use of cellular automata (CA) and their emergent behavior over the hyperspectral cube in order to achieve this objective. Using cellular automata for segmentation in hyperspectral images is not new, but most approaches to this problem involve hand designing the rules for the automata. Additionally, most references found are just extensions of one or three-dimensional methods to multidimensional images, and, as a consequence, average out the spectral information present. The main contributions 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. The procedure has been tested over synthetic and real hyperspectral images.

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

1 Introduction

In the last decade, Hyperspectral imaging has become a very important source of remote sensing and industrial processing information through the acquisition of imaging data with a very high level of spectral detail. Instead using three values to represent the color of each pixel (corresponding to the levels of red, green and blue), hyperspectral images provide hundreds of values that correspond to the intensity of a large number of bands within the visible and near infrared spectrum. As a consequence, the definition of color they provide can be very accurate and information intensive, thus facilitating discrimination of different elements within the images. A hyperspectral image is 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.

The original applications of hyperspectrometers usually fell within the realm of remote sensing with images that were obtained from high flying planes [1]. Therefore, most of the analysis methods developed were aimed at providing the ratio of endmembers present in every pixel when analyzing different types of covers. 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, especially in ground based applications [2][3]. In this realm, the images are taken close enough to the subject to obtain a relatively detailed view and the geometric layout or morphologies of the pixel sets become necessary in order to provide an adequate identification or classification of the elements present. In other words, it is becoming necessary to perform combined spatial-spectral processing in order to identify morphological features and this means segmenting objects from the background.

Hyperspectral images have always been characterized by the fact that they are very large data sets due to the spectral detail they include. This is even more pronounced in current applications where the spatial resolution is also quite large. Processing the amounts of data that result, especially in real time, requires the development of specialized techniques and the use of very high levels of parallelism. It is in this context that the use of extremely distributed techniques, such as Cellular Automata, become very interesting and can provide the level of locality in computations that allows for the use of the current state of the art FPGA [4] or GPU [5] architectures.

The concept of Cellular automata (CA) was first proposed by Von Neumann and Ulam [6] as a biologically inspired decentralized computing paradigm which can be made a universal computing machine. They are a spatially extended decentralized system in which a set of cells are distributed in space and communicate only with their neighbors. Each cell is basically an automaton (usually a finite state automaton) and the state of a cell each instant of time depends on the state of its neighbors and on its own previous state through a set of transition rules. It is by adequately choosing these rules and by iterating the state transition process in time that the full computational capabilities of the whole system are achieved.

The main difficulty when using CAs, especially when complex functions need to be achieved, is to determine the rules for the automata. That is, knowing how the whole CA system needs to behave, it is necessary to determine what rules implemented in each cell will produce the desired behavior. This is what is called the inverse problem and it is a very difficult problem to solve. Different approaches have been taken by the authors within the CA community in order to attempt to solve the inverse problem (see, for example,[7]). However, the most popular approach has been to use evolutionary techniques in order to evolve the set of rules [8][9][10][11].

Here we are interested in a particular application, that is, segmentation of hyperspectral images. This type of segmentation has been addressed by other authors using a number of approaches. They range from simple extensions of those considered in regular image processing such as watershed algorithms [12] to other more advanced techniques that try to take into account the information provided by the spectral detail of this type of images such as in the work by [13] [14] or [15] or even resort to lattice computing techniques as in [16]. Most of these techniques require complex processing stages that are very hard to implement in limited computing resources and are thus not very appropriate when contemplating real time processing in the field.

Recently, authors such as [17] have addressed hyperspectral image processing problems, which in their case was an edge detection problem, using CAs in order to improve on the efficiency of the processing. Others have addressed the segmentation problem using CAs (see [18] [19] [20]). However, the CAs they propose have been hand created ad hoc and, even though some do consider multidimensional images, they are still far from the dimensions of hyperspectral images and are usually projected onto a lower dimension during the segmentation process.

In this paper we are going to explore the possibilities of evolving CAs for hyperspectral segmentation using spectrally relevant information. To this end we consider a particular type of CA, that is, the eight cell neighborhood CA (this is usually called a Moore neighborhood). This is a system that, with appropriate rules, has also been shown to be capable of universal computation [21]. We will show how using an evolutionary approach, the automata can be tuned to perform the segmentation processes.

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 presents some experiments and results and, finally, in Section 4 we present some conclusions and indications of future avenues of work.

2 Evolution of Cellular Automata

When applying the cellular automata in this paper we take each pixel as a cell that contains an N-band spectrum. This spectrum corresponds to the state of the cell or pixel and, as a consequence, the state space is continuous and corresponds to the positive vector space $\mathbb{R}^N$.

The cellular automata are iteratively applied over the hyperspectral cube modifying in each iteration the state (spectrum) of each one of the pixels in the cube. This modification depends, on the information of the spectra of the 8 closest neighboring pixels, on the set of rules that control the automaton and, finally, on the state of the cell over which the automaton is applied.

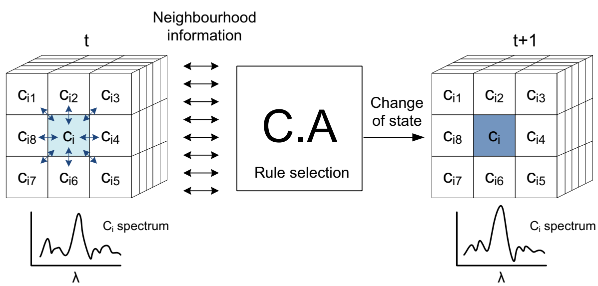

Fig. 1. Schematic of the operation of the cellular automaton over a hyperspectral cube.

For the calculations within the automata we consider a distance function between the spectra of the neighboring pixels and the pixel being processed. In this case we have chosen a very simple distance measure in the form of the spectral angle. This value for a cell *i* is obtained with respect to the eight neighboring cells *j* as:

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

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

where α_{ij} is the spectral angle (normalized between 0 and 1). A_i is the vector made up by the eight angles with respect to the eight neighboring cells.
The cellular automaton itself is made up by two angles, ε and θ, and a set of M rules, each one of them made up of 9 parameters:

$$CA = (\varepsilon, \theta, r_{11}, r_{12}, ..., r_{19}, r_{21}, r_{22}, ..., r_{29}, ..., r_{M1}, r_{M2}, ..., r_{M9})$$

The first eight r_{xj} parameters of each rule are related to the information on the spectral angle of the neighbors with respect to the pixel being processed. These eight angles can take values from {-1, 0, 1}. The value is calculated as:

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

The ninth parameter is related with the change of state of the cell and points towards one of the eight neighbors. When the automaton is executed over a given pixel, it compares the angles of the pixel to those of all of its neighbors constructing vector *A* and transforms the angles α_{ij} into α_{cij} as stated above using the first two values of the automaton, ε and θ. It then finds the rule whose first eight *r* values are most similar to *A* and applies it. Basically, when a given rule is applied the spectrum of the pixel being processed is moved towards the spectrum of the pixel towards which the rule points (in the case of the examples in this paper they are averaged). Thus, the change of state of each cell corresponds to a variation of the spectrum of the cell. In order to make rule selection invariant to rotations and reflections four possible rotation and the vertical and horizontal flips of each rotation are taken into account.

The cellular automaton is applied to all the pixels of the hyperspectral image each instant of time producing each instant a hyperspectral cube that has been modified with respect to the one corresponding to the previous instant.

The selection of the set of rules that make up the cellular automaton is a complex problem that is difficult to solve manually. In this work we address the problem in an automatic manner through the application of an evolutionary algorithm (EA). The automata 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 thresholds and rules of the automata. In addition, to produce a cellular automaton that works for the range of cases that is desired, it is necessary to provide a set of examples over which the prospective automata can be evaluated during evolution. To this end we have used synthetic images with the variability range sought. Each one is generated at run time for the evaluation of an individual. Basically the prospective cellular automaton is run over the image for a given number of iterations and the result is compared to a desired ground truth. It is important to note that a different image is generated for each evaluation.

Out of the different possibilities to measure the quality of the segmentation obtained after applying the automaton in order to provide a fitness value for the automaton, we have chosen a measure that provides a balance between the homogeneity of the segmented regions and the discrimination of the different regions within the hyperspectral image. It is given by:

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

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

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; α_{th} is a threshold angle value.

The particular evolutionary algorithm chosen for obtaining the cellular automata in this work was a standard Real Valued Genetic Algorithm (RVGA). A summary of the parameters for this RVGA are displayed in Table I. It can be seen that we are using a population of 182 individuals and a maximum of 50 generations of evolution per run.

TABLE I. PARAMETERS OF THE EVOLUTIONARY ALGORITHM

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 criteria	Generations

3 Experiments and Results

In this section we present some results of using evolution to obtain cellular automata for segmenting hyperspectral images in an unsupervised manner. The automata were obtained using the algorithm described in the previous section and were then tested over two different types of images: a set of synthetic images, and the well-known AVIRIS Salinas image for comparison purposes.

To train the genetic algorithm we generated at runtime synthetic images similar to those shown in figure 2. Every candidate automaton in the population was evaluated using a different image, although all of them had similar types of features corrupted by noise and artifacts.

Fig. 2. Left: Training RGB image. Center: Ground truth. Right: Spectral angle image with respect to an arbitrary fixed vector.

Regarding the details of the automata that are evolved in this section, we have used automata such as those described in the previous section with a total 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.

Figure 3 displays the evolution of the fitness of the best individual and the average fitness of the population for one of the evolution processes. 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. However, the global trend followed by the curves is clear.

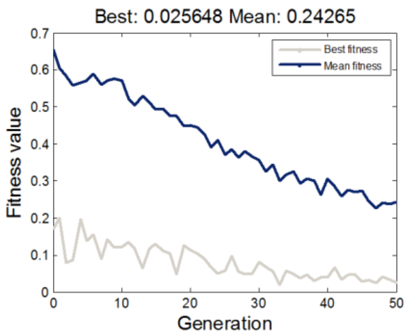

Fig. 3. Fitness function value (minimized) during one of the evolutionary processes.

The evolved CAs were first tested over synthetic images that were generated much in the same way as those used for evolution but the shapes of the areas to be segmented were completely different from anything seen during training. Figure 4 displays the result of applying the resulting CA over one of these synthetic hyperspectral images constructed using four 64 band base spectra that were spatially corrupted by noise and artifacts as represented on the left side of the figure (as a 2D projection of the spectrum as spectral angles with respect to a fixed reference). The central part of the figure displays the segmentation obtained and the right side the ground truth. This CA, after iterating 100 times over the image, obtains an OA (Overall Accuracy) of 99.9%.

Fig. 4. Left: 2D transformation of a synthetic 64 band hyperspectral image. Each pixel displays the angle between the spectrum of the pixel and a reference spectrum with all of its bands at the maximum value. Center: 2D transformation of the cube resulting from the segmentation obtained by the CA. Left: Labeling of each area.

After testing the automata over synthetic images, we tested it over a set of real images. In particular, here we are going to present the results corresponding to tests performed with one of the automata over the AVIRIS Salinas hyperspectral image. The results of the segmentation are displayed in the rightmost part of figure 5.

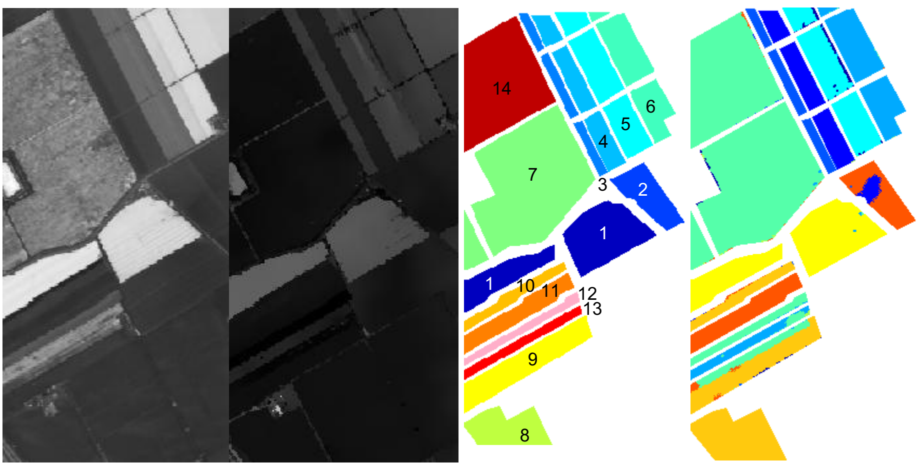

Fig. 5. From left to right: 2D angular transformation of the AVIRIS scene, 2D angular transformation of the hyperspectral cube resulting from iterating the CA over the image 100 times; ground truth (the content of each area is indicated after Table II): result of the unsupervised CA based segmentation process (the colors do not correspond to labels, they just indicate homogeneous regions or segments as determined by the CA).

These results are visually quite acceptable and show that the evolved CA has obtained a set of rules that allow it to clearly segment the image into the relevant areas except for a slight oversegmentation in areas 9 and 2, for each one of which the

CA delimits two categories, and some noisy pixels in the borders of some areas. It is difficult to really know if the aforementioned over-segmentation represents some relevant feature of the image as the only information we have on the area is the ground truth image shown in figure 5. However we can say that the areas do present different spectra in the original image and, consequently, from a strict segmentation point of view, they are correctly segmented. On the other hand, the noisy pixels in the borders of some areas have to do with imprecise delimitation of the areas in the ground truth (borders of fields that do not contain plants and are taken as paths). However, to really be able to quantify the goodness of the segmentation we have resorted again to the Overall Accuracy index, in this case for each one of the areas with the same label in the ground truth image. These OA values are presented in table II and they lead to a global OA of 94.1%.

TABLE II. OVERALL ACCURACY (%)

Area	**Cover**	**OA(%)**
1	Brocoli	99,7
2	Fallow	82,4
3	Fallow rough plow	100
4	Fallow smooth	98,4
5	Stubble	93,8
6	Celery	99,5
7	Grapes	98,7
8	Soil vineyard develop	100
9	Corn senesced green weeds	70,3
10	Lettuce romaine 4wk	94,8
11	Lettuce romaine 5wk	100
12	Lettuce romaine 6wk	97,8
13	Lettuce romaine 7wk	83,4
14	Vineyard	99,3

It is important to note here that the unsupervised segmentation procedure presented here seeks to provide a single label in terms of a single spectrum to represent each area; however, we are talking about segmentation, not classification. As a consequence the representative spectrum may not be the one corresponding to the ground truth label and when two separate areas in an image that contain the same type of cover are segmented they may be assigned different labels (much in the same way as in the case watershed based segmentation algorithms). That is, for instance, in an area where there are very small plants each surrounded by a lot of soil, the representative spectrum obtained in an unsupervised manner may be very close to soil and if one takes it at face value, without a posterior labeling or classification process, this may lead to confusion between areas that present this characteristic. Again, we are segmenting and if a posterior classification process is desired, this can be achieved selecting over the original hyperspectral cube a representative spectrum for each of the areas the segmentation algorithm has defined (maybe the average spectrum of the area) and use another process to decide what that spectrum corresponds to or whether it is a composition of several endmembers. However, this is beyond the scope of the present algorithm and paper.

4 Conclusions

Segmenting hyperspectral images is one of the basic tasks that must be performed when processing this type of data. Consequently, this task should be as efficient as possible so as to be able to perform it in quasi-real-time with adequate reliability when in the field. In this line, Cellular Automata (CA) based structures for performing spatial-spectral operations over hyperspectral images is quite a promising approach as these structures are very well adapted to their implementation in very efficient high performance processing hardware such as GPUs. Nonetheless, this approach poses the problem of generating the rule set for the CA to perform the task it has been assigned. In this paper we have shown that evolving these structures is quite efficient. The automata obtained this way are competitive and, more importantly, adapt better to changes in the way the user wants the segmentation to be performed than other more traditional approaches.

The results presented in this paper were produced using a distance metric based on the spectral angle, which, even though it does consider the spectral information present in the images, it does not do it in a very detailed fashion, that is, it considers large classes of equivalence that may not be suitable for some applications. As a consequence, an improvement in performance could be expected if more detailed spectral information is considered in the distance metric. We are now working on CA based approaches that take into account the detailed information provided by the spectra directly in the distance metric in order to benefit from this wealth of information.

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] 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.

[3] 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.

[4] 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.

[5] 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.
[6] V. N. J and B. Aw, “Theory of self-reproducing automata. Urbana: University of,” *Illinois Press*, 1966.
[7] N. Ganguly, B. K. Sikdar, A. Deutsch, G. Canright, and P. P. Chaudhuri, “A survey on cellular automata,” *Engineering*, pp. 1-30, 2003.
[8] 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.
[9] 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.
[10] R. Subrata and A. Y. Zomaya, *Evolving cellular automata for location management in mobile computing networks*, vol. 14, no. 1. 2003, pp. 13-26.
[11] W. Elmenreich and I. Fehérvári, “Evolving Self-organizing Cellular Automata Based on Neural Network Genotypes,” *Network*, vol. LNCS 6557, pp. 16-25, 2011.
[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] 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.
[14] T. Veracini, S. Matteoli, M. Diani, and G. Corsini, “Robust Hyperspectral Image Segmentation Based on a Non-Gaussian Model,” *Distribution*, pp. 192-197, 2010.
[15] 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.
[16] 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.
[17] 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.
[18] 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.
[19] 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.
[20] J. Gallego, C. Hernandez, and M. Grana, “A morphological cellular automata based on morphological independence,” *Logic Journal of IGPL*, Feb. 2011.
[21] 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.