VOL. 14, NO. 18, SEPTEMBER 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A NOVEL ALGORITHM BASED ON CELLULAR AUTOMATA TO ELIMINATE NOISE IN DIGITAL IMAGES

Karen V.Angulo-Sogamoso[1], Danilo G. Gil-Sierra[1] and Helbert E. Espitia-Cuchango[2]
[1]Ingeniería de sistemas, Facultad de Ingeniería, Universidad Distrital Francisco Joséde Caldas, Bogotá, Colombia
[2]Facultad de Ingeniería, Universidad Distrital Francisco Joséde Caldas, Bogotá, Colombia
E-Mail: kvangulos@correo.udistrital.edu.co

ABSTRACT

A cellular automaton is a processing system that makes decisions based on the information of its neighbors. Implementing a cellular automaton can have a direct relationship with the form of digital images representation, whereby it is possible to perform image processing applications using this concept. Due to this, this document presents the proposal of an algorithm based on cellular automata for the elimination of impulsive noise in digital images, which makes use of the adaptation mechanism to adjust to the environment conditions (in this case, the image) at the moment that the information acquired by the cellular automaton is insufficient to make a decision about the pixel under evaluation. In addition, thanks to the consideration of related works, an evaluation method can be established to observe the performance of the proposed algorithm against two algorithms presented by other authors. Tests are carried out with four images that have different characteristics. In evaluating the images exposed to different noise levels, the results obtained show that the proposed algorithm presents better according to the Structural Similarity Index (SSIM), since at noise levels between 10% and 90% improvements in noise reduction range between 15% and 68%.

Keywords: cellularautomaton, digitalimages, processing, noise.

1. INTRODUCTION

For several years, digital images have played an important role into the research environment, thanks to the fact that, like any type of data, it can reveal valuable information for any field [1]. With the arrival of information systems and continuous technological advances, it became necessary to capture an image in a computational way and, at the same time, create systematized techniques or applications that will help extract information from it. From this need, the processing of digital images arises. In the field of image processing, there is great variety of techniques that fulfil different purposes and, in turn, influence different forms on an image, all aiming the improvement of the especial conditions that this presents. This is the case of techniques focused on the elimination of noise, that seek a corrupt image (image that is affected by some external factors which generate noise on it, making it impossible to read it properly [1]) exposes its best shape, by varying or changing the value of noisy pixels by one that is more in line with its surroundings.

When discussing noise elimination, it is important to mention that within the field of noise typifications of image processing, there is great variety, however, among the most common are: Gaussian noise and impulsive noise, since work and research are frequently presented focused on solving this problem and, at the same time in many real life cases, often the images are affected by impulsive noise, which can be caused by different factors such as errors or interferences in the transmission sections (that is, by transmitting the information bits), defective memory locations in the hardware, errors associated with the equipment used to capture images or malfunction of the pixels in the camera sensors [2]. However, within the classification of impulsive noise there is a special and well known case called "salt and pepper", which is observed in a contaminated image through pixels that can only take the maximum or minimum values; that is, they usually appear as black and white dots on the image [1], [2].

Taking into account that the presence of noise can significantly affect the quality of an image, various techniques have been proposed for the restoration of images that have been affected by impulsive noise, which have made solid contributions to this field, through the creation of new methods from the variation of conventional techniques or the formulation of new types of filters, which have been continuously improved in order to obtain better results in less time. Among the most used methods to suppress this type of noise is the non-linear filter called "median filter", because its strength lies in efficiently solving the effects on images with low noise densities [2]. However, when applied in images with high noise densities, a slight deformation of the image details (lines, corners, edges) is generated [3], since the operation is carried out indiscriminately over the whole image and does not discriminate if the pixel is noisy or not. Based on the above, and observing the problems that still occur in the basic filtering of salt and pepper noise, several researchers have developed new techniques and approaches to obtain improvements in this topic.

Particularly, Garnett, Huegerich, Chui and He [4], proposed a filter which through statistical values obtained from the quantification of the differences between pixel intensities with respect to their more similar neighbors. When applied to an image, it can be seen that this filter is extended in such a way that it is possible to eliminate both Gaussian and Impulsive noise, the results were outstanding at a quantitative and qualitative level. However, the filter has some faults when the noise level is too high, because if most of its neighbors are noisy pixels, the calculated statistical value is not optimal.

In [5] Echeverri, Rudas, Toscano and Ballesteros, through the use of interpolation through radial base functions, managed to generate a method for the elimination of impulsive noise in color images. For this method, their results were compared with classical non-linear algorithms such as filtering by median, mean and "outlier", thus demonstrating that this algorithm is more efficient and robust against images with a high percentage of noise, while with low noise density images it did not have the same efficacy. At the same time, in the work developed in [6] by Esakkirajan, Veerakumar, Subramanyam and Prem a decision-based median asymmetric filter algorithm was proposed, which was applied in the restoration of grayscale and color images affected by impulsive noise. At a qualitative level, the algorithm presented better results than the Standard Median Filter (MF), Decision Based Algorithm (DBA), the Modified Decision Based Algorithm (MDBA) and the Progressive Switched Median Filter (PSMF).

Sahin *et al.* [7] proposed an algorithm that combines the concepts of a 2D cellular automaton and fuzzy logic for the restoration of digital images affected by impulsive noise. The algorithm uses a local fuzzy transition rule that assigns a membership value to the neighborhood of corrupt pixels and sets the next status value to the pixel under evaluation, eliminating noise by up to 90%. On the other hand, although the method of elimination of noise divided into two phases elaborated in [8] shows similarity with the algorithm proposed by Sahin, in [7]; regarding the percentage of noise eliminated in the image, this algorithm has proved to be very effective for high noise densities; however, the size of the window increases the processing time considerably. In the first phase, this algorithm applies an adaptive median filter which identifies the pixels that are probably contaminated by noise (called noise candidates), and in the second phase the restoration of the image is done through a regularization specialized method, which is applied to pixels that have been selected as noisy candidates.

There is also another group of algorithms that seek to suppress the noise present in an image through different techniques, one of them is based on the application of fuzzy logic in order to create a multi-step filter to eliminate impulsive noise as shown in [1].

As can be seen, although sometimes the results offered by the filters that work in the frequency domain are satisfactory (based on the Fourier transform), and in the space domain (directly working on the pixels of the image) [9], they generally offer generic results that do not fully cover the most relevant characteristics of the images, either because of their design or implementation. Likewise, when reviewing in retrospect the potential of the filters applied in images with a high number of pixels, it is highlighted that these are not very efficient and their performance in the optimization of resources is flawed, since several steps are required to reach the final result (e.g. filters in the frequency domain require the execution of the Fourier transform of the original image, a sub-filter frequency attenuator and the inverse transform to achieve its objective), and on the other hand, when applied in images with high noise levels, the result does not show an acceptable restoration and on the contrary quality is lost, which generates visual errors and in some cases, shortcomings that represent a high cost in resources and execution time.

This is why at present, there are multiple applications in the field of image processing that make use of the concept of cellular automaton (CA), developed by John von Neumann [10], [11], to efficiently solve various problems that arise since over the years. It has been able to competently establish a simile between a digital image and the design of an CA or in other words, it has been thought that digital image can be manipulated in an appropriate way by means of an AC, which has facilitated the resolution of such problems. Through a CA it is possible to simulate complex behaviors in a finite set of elements, [12], [13], [14]. In the same way, cellular automata, when considered as bio-inspired systems [15], [16], [17], can carry out complex tasks by defining a series of rules that are easy to incorporate, remove or modify according to the need, which implies a better global behavior by means of the interactions of a matrix with its environment, thus optimizing the resources used [18]. Also, thanks to the versatility of the cellular automata, it is possible to implement different activities simultaneously without greatly affecting their performance.

For the above, and considering the strong relationship that exists between the implementation of a cellular automaton and a digital image, in this document the proposal of an adaptive algorithm based on cellular automata to eliminate impulsive noise in digital images is made, which by its adaptive characteristics allows to obtain more information of the image at the time of modifying the corrupt pixels that in situations with high levels of noise are really efficient, since the quality of the restored image is considerably improved, and thanks to the rules established in the automaton it is not necessary to operate on the whole image, considerably reducing the computational time used by the algorithm, thus generating an innovative contribution for this field.

2. MATERIALS AND METHODS

2.1 Cellular Automaton

Cellular automaton constitutes a particular class of discrete dynamic systems [10], which through a simple definition, can model highly complex systems [19], [20], [21]. These can be defined as a set of cells with a D dimension, that can take the form of $D = 1$, $D = 2$ or $D = 3$, as showed in Figure-1, depending on the utility that is given. Currently, the most commonly used case in the various applications of the investigative sciences, particularly in image processing, is that which simulates a two-dimensional space.[22], [23], [24].

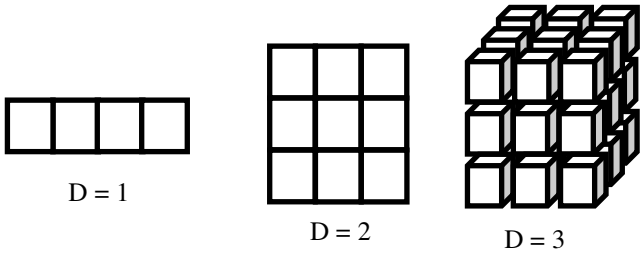


Figure-1. Cellular automaton dimensions.

Regardless of the chosen dimensions, every cellular automaton has a certain basic structure, which is composed by three (3) parts: a) a set of cells or lattice, b) a set of adjacent neighbors or neighborhood, and c) A rule set for local transitions [22].

- Set of cells or lattice, which has a finite number of possible states or values for each cells, in the case of binary images 1 or 0, and for grayscale images from 0 to 255.
- A set of adjacent neighbors or neighborhood, such as the neighborhood of Von Neumann (four (4) neighbors) or the neighborhood of Moore (with eight (8) neighbors).
- A ruleset for local transitions (local transition functions, transition of the state of the cells or transitions table of the states of the cells). In synthesis, within the groups of rules there are two (2) particularities to be highlighted: i) the communication between the central cell and its neighbors is local, uniform, and synchronous and ii) the global evolution of the system through a step of time discreet is deterministic [22].

In view of the above, regarding the operation of a cellular automaton, it is clear that the state of the cell under evaluation, at time $t + 1$, will be determined by the current state of the surrounding neighbors and their current state, in other words, this will be updated synchronously in discrete time intervals [10], [11]. By defining it from a more formal perspective, it can be said that a cellular automaton d-dimensional (o d-CA) can be seen as a quadruple$A = (Z^d, S, N, \delta)$, such that, $\mathbb{Z}^d$: is a d-dimensional space of integers, S: is a finite group that corresponds to the states of A, $N = \left\{\frac{n_i}{n_j} = (x_{ij}, \dots, x_{dj}), j \in \{1, \dots, n\}\right\}$: is an ordered finite subset of $\mathbb{Z}^d$, called neighborhood of$A$ and δ: corresponds to a local transition function, or local A rule, such that $\delta: S^{n+1} \rightarrow S$.

On the other hand, the neighborhood of a deterministic cellular automaton, is represented by contiguous or surrounding cells with respect to a central cell, by extrapolating the concept to an image, one could say that these are directly associated with a pixel. This set of cells has a particular training according to the situation that is presented, since more or less information may be required [22], [23]. Essentially, the neighborhoods can be defined in terms of symmetry: symmetric and asymmetric neighborhoods. The symmetric neighborhoods have a base (an integer), where the value assigned to it corresponds to the number of neighbors in each direction, although by default the value of the base corresponds to 1 [22], [23], [24]. , which means that from the central cell there is a neighbor to the right, one to the left, one at the top and one at the bottom, for the particular case of the Neumann neighborhood. On the other hand, asymmetric neighborhoods, as the name implies, can vary the number of cells from the central cell. Within the multiple variations existing in a neighborhood, there are two well-known neighborhood types, those defined by Neumann and by Moore. The first of these at its most elementary level has 4 neighbors from the central pixel as showed in Figure-2.

	(x, y-1)	
(x-1, y)	(x, y)	(x+1, y)
	(x, y+1)	

Figure-2. von Neumann neighborhood.

In many of the cases this neighborhood can be extended n number of times, looking for that by means of the previously established rules a greater quantity of information can be handled as it is showed in Figure-3.

		2		
	2	1	2	
2	1	*P*	1	2
	2	1	2	
		2		

Figure-3. Von Neumann neighborhood extended one position ($n = 2$).

The Moore neighborhood in its essential form, 8 neighbors are considered from the base cell, which facilitates the handling of edges and corners; in addition, the information that can be managed with this neighborhood is greater than the Neumann neighborhood, four additional cells are considered, as showed in Figure-4.

(x-1, y-1)	(x, y-1)	(x+1, y-1)
(x-1, y)	(x, y)	(x+1, y)
(x-1, y+1)	(x, y+1)	(x+1, y+1)

Figure-4. Moore neighborhood.

In the same way as the Neumann configuration, this neighborhood also has extended versions of n pixels, which obey the formula$(2n + 1)^2 + 1$, an example of this neighborhood can be seen in Figure 5.

2	2	2	2	2
2	1	1	1	2
2	1	*P*	1	2
2	1	1	1	2
2	2	2	2	2

Figure-5. Moore neighborhood extended one position ($n = 2$).

2.2 Digital filters

On the other hand, digital filters can be defined as the computational process or algorithm by which, a digital signal (sequence of samples) is transformed into another output sequence [25]. That is, it corresponds to a process that takes a succession of numbers (input signal) and transforms it into another (output signal) in order to meet certain specifications. A digital filter can be represented by the block diagram of Figure-6.

$$y(nT) = R\{x(nT)\}$$

Figure-6. Digital filter representation [25].

Where $x(nT)$ is the input sequence (the filter excitation) and $y(nT)$ is the response of the filter to excitement $x(nT)$. Considering the above, an image can be filtered in the domain of space, working directly on the pixels of the image, or in the frequency domain, where operations are carried out by means of the Fourier transform of the image [26]. For the filtering process in the space domain, a set or neighborhood of nearby pixels must be related to the target pixel, in order to obtain useful information that allows the filter to operate on the specific pixel in which the process is being performed and thus obtain improvements or useful data that can be subsequently used on it; this applies to each and every one of the pixels of the image [27]. These filters can be categorized into linear filters (filters based on Kernels or convolution masks) and non-linear filters.

Considering the above, currently within the common techniques or filters, there is a wide variety of them that seek to efficiently restore an image corrupted by salt and pepper noise, which is defined by the probability density function that is presented as follows:

$$p(z) = \begin{cases} P_a & for\ z = a \\ P_b & for\ z = b \\ 0 & for\ other\ z \end{cases} \quad (1)$$

In the equation (1), if $> a$, the intensity of b will be reflected as a white dot on the image and a as a black dot. Until now, for the images filtering the nonlinear technique called median filter has been one of the common ones to eliminate this type of noise due to the simplicity and ease of implementation when smoothing images (in other words, reduce the amount of intensity variations between neighboring pixels) [2]. However, because it assigns to each pixel the value of the local median (samples around each value of the signal $f(x) = y_{med}$), loss of image details is generated (fine dots and lines), rounding the corners of the objects and moving the edges; in addition, it highlights the low effectiveness of this filter to eliminate noise at high densities [2], [28].

Therefore, a method is proposed to combine filtering in the space domain and the principle of operation of cellular automata together with that of an adaptive algorithm, aiming the elimination of salt and pepper noise only by modifying those pixels that are classified as corrupt, in order to improve the filtering process and, consequently, reduce the machine time and the quality of the restored image.

2.3 Structural Similarity Index (SSIM)

It is a quality indicator of a digital image. This is used to measure the similarity of two images, that is, to determine how different the original image is from the distorted one, by calculating three indexes that compare the luminance terms (l), contrast (c) and structure (s) of a couple of images. The combination of the three terms is defined by the Equation (2).

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \quad (2)$$

Where, $l(x,y) = \frac{2\mu_x 2\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$, $c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$ and $s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$. In these expressions μ denotes the average value, σ the standard deviation and σ_{xy} the covariance. In the same way, the variables x and y represent the original and distorted images respectively. Constants C_i values are applied to avoid division by zero, parameters α, β, and γ have the default value 1, by multiplying the three indices the following equation is obtained:

$$SSSIM(x,y) = \frac{(2\mu_x 2\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3)$$

Figure-7 shows the schema for calculating the SSIM index, which is done automatically.

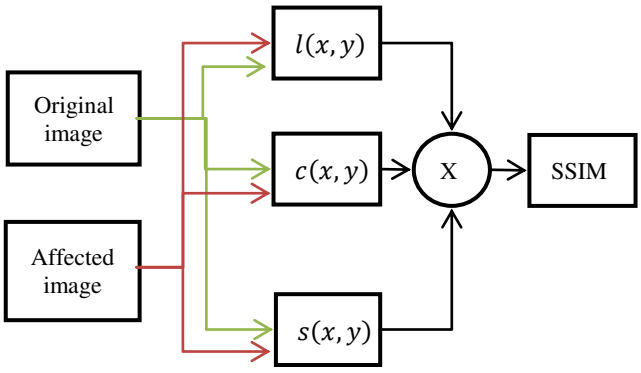


Figure-7.SSIM index functioning.

3. CELLULAR AUTOMATA DESIGN

In this work, a method based on cellular automata is proposed, which seeks to remove impulsive noise in digital images represented in gray scale, taking as a reference the method proposed by Tourtounis, Mitianaudis and Sirakaulis in [2], which goes through the image pixel by pixel and making use of the concept of neighborhood proposed by Moore, generating a$3\,x\,3$ matrix, having as center the pixel in evaluation and depending on the state of it and its neighbors, makes a decision which may or may not modify it. The image in two dimensions (2D) can be represented as a $m\,x\,n$ matrix, where each position corresponds to a pixel inside the image and the values assigned to these positions refer to the intensity of the image at that particular point. Also, it is considered that each cell of the cellular automaton is a pixel inside the image; therefore, the dimensions of the cellular automaton is $r = 1$ and the number of its neighbors equals 8, which indicates that both the width and the length are equal, that is to say in size $l_1 = l_2$. For the case, in particular those of border conditions $C(i_0, j_0)$, the first and last row and column are extended, so that when the automaton reaches the limits of the image, the final result of the processing is not affected.

Taking into account the previously described concepts, through the definition of Moore's neighborhood (M) of a cellular automaton (named N) in the cell $C(i_0, j_0)$, the fixed size 3×3 is defined, by the following equation:

$$N(i_0, j_0)^M = \{(i,j): |i - i_0| \leq r, |j - j_0| \leq r\} \quad (4)$$

Where r, is the range. On the other hand, it should be noted that in a grayscale image, each pixel can take discrete values between 0 and 255, which, for this case are used to establish the state values of the cellular automaton and, in turn, the extreme values are part of the initial denomination of salt and pepper noise, which define the minimum and maximum value of the states as $minimum\ state = 0$ and $maximum\ state = 255$.

From the above, it is proposed a local two-dimensional rule (2D) that defines the initial behavior of the cellular automaton, which achieves the location and classification of the pixels that are determined as noisy or not, as showed below:

$$C^{t+1}(i,j) = \begin{cases} C^t(i,j), & \text{if } 0 < C^t(i,j) < 255 \\ Cn^t(i,j), & \text{if } C^t(i,j) = 0 \text{ or } C^t(i,j) = 255 \end{cases} \quad (5)$$

As mentioned, the image upon being affected by salt and pepper type noise takes maximum or minimum discrete values in each affected pixel. This means that when applying the equation (5), only those pixels where their value is a maximum will be affected (255 or white) or a minimum (0 or black). In short, the value $C^{t+1}(i,j)$ is directly related to $C^t(i,j)$ or the value of the pixel under evaluation, which depending on the case may take a different value. For the case where the assigned value is $Cn^t(i,j)$ or the new value, a calculation is made based on the sub-rule described below:

$$Cn^t(i,j) = \begin{cases} C(2n+1)^2 + 1, if condition A \\ avg\left(C^t(i,j)\right), if condition B \end{cases} \quad (6)$$

$$\begin{cases} A: \forall\, C^t(i,j) \in N, C^t(i,j) \neq 0\ or C^t(i,j) \neq 255\ and N = 0 \\ B:\ si\ \forall\, C^t(i,j) \in N, \exists C^t(i,j) \neq 0\ or \exists C^t(i,j) \neq 255 \end{cases}$$

When a pixel is considered as noisy, the sub-rule described in the equation (6) is applied, in the case in which Moore's neighborhood only has maximum or minimum values in its structure, a position will be adapted and extended, $N = 24$ as showed in Figure-4 and the state of the cell will be the average with the additional information excluding the maximum and minimum values. Otherwise, all the values 0 and 255 are deleted and an average is applied on them to obtain the new value of the cell. On the other hand, it should be clarified that, when the state of the cell of the cellular automaton is not equal to any of the values considered as thresholds, this cell will not be considered as a noisy cell and, consequently, will maintain its initial value. Because the noise is not eliminated in the first pass of the cellular automaton, the whole cellular automaton is evaluated a finite number of times depending on the value of the applied noise, that is, if the noise level is equal to n, the cellular automaton will iterate $\left(\frac{n}{10}\right) + 1$.

In general, the pseudocode code of the proposed algorithm can be seen below:

Proposed Algorithm: MERIG
Step 1: Read the original image$I(x,y)$.
Step 2: If $I(x,y)$ it's not grayscale, convert it to this type of image.
Step 3: Normalize $I(x,y)$.
Step 4: From $I(x,y)$ create a $3\,x\,3$ matrix, which will simulate Moore's neighborhood.
Step 5: The central pixel of the previous matrix will represent the$C_{(i,j)}$.
Step 6: Create a vector V of $8\,x\,1$ dimensions. This will contain the values of the pixels inside the matrix excluding the central pixel.
Step 7: If $0 < C_{(i,j)} < 255$, $C_{(i,j)}$ is not a noisy pixel, so it keeps its value.
Step 8: If $C_{(i,j)}$ is a noisy pixel ($C_{(i,j)} = 0 \vee C_{(i,j)} = 255$, then:
Case 1: if all the elements of$V = 0 \vee V = 255$, then
Expand the neighborhood and$V_{new}\, C_{(i,j)} = mean\ (V_{new})$ without $V_{new_i} = 0 \wedge V_{new_i} = 255$
End if
Case 2: all the elements of $V \neq 0 \vee V \neq 255$, then
$C_{(i,j)} = mean(V)$ sin $V_i = 0 \wedge V_i = 255$
End if
End if
Step 9: Repeat steps 7 and 8 for all pixels within$I(x,y)$ until n (noise level) be equal to$\left(\frac{n}{10}\right) + 1$.

4. RESULTS

According to the different applications, methods and algorithms presented in the introductory section, and taking into account that in order to make a base standard and objective comparison, at least some analogous characteristics must be obtained. As a base of comparison were used those methods that despite their different peculiarities in some exposed characteristics were similar to the model proposed in this work. However, it is important to note that the research replication work was carried out through the information provided by the authors in their articles and it is possible that not all the variables taken into account in these cases have been published, which makes it impossible a reliable copy of them and, consequently, it is possible that the results obtained at the time of replication may present some differences with respect to the results presented in their final works.

Starting from the above, and taking into account that with these results we seek to measure the performance of the system when performing a correct restoration of different gray scale images affected by salt and pepper noise. We proceeded to configure four (4) images exposed to different noise levels (between 10% and 90%, making 10% increments), so that these were the basis of the comparison at the end of the restoration process exposed by each chosen algorithm. Subsequently, by means of the SSIM index, the similarity of the result image was verified against the original.

To evaluate the performance of the Proposed Algorithm (or MERIG), the algorithms described in the works of Dalhoum, Al-Dhamari, Ortega and Alfonseca, hereinafter referred to as Abdel Algorithm., (see Figure-8) and Tourtounis, Mitianoudis, and Sirakoulis, here in after Dimitrios Algorithm (see Figure-9), which also implement an algorithm based on cellular automata to eliminate noise in grayscale images. This implementation was carried out in MatlabR2017a.

Abdel Algorithm.
Check the pixel value $X_{i,j}$ and the values of their neighbors
If the noise is uniform
$mx = max(neighbors)$;
$mn = min(neighbors)$;
If $X_{i,j} == mx$ or $X_{i,j} == mn$
//Take the value of the median of the neighbors.
$y = SM_{i,j}$
End
Else // Salt and pepper noise
If $(X_{i,j} == 0)$ or $(X_{i,j} == 255)$
If there are neighbors that are not 0 o 255
$X_{i,j}$ = the median step 10
Else
$X_{i,j} = mean(neighbors)$
End
End
End

Figure-8. Pseudocode of the algorithm proposed in [12].

Dimitrios Algorithm.
Read the original image $I(x, y)$
If $I(x, y)$ is on a RBG scale, passes it to grayscale, or works independently on each color channel
Take a $2-D$ window of $3x3$ size which scans the image $I(x, y)$
$C_{i,j}$ represents the central pixel of a 2D Moore's neighborhood in the cellular automaton
Create the vector B, which has $8x1$ dimensions. The values of the pixel inside the window, excluding the central pixel, are sorted in ascending order
B_{min} y B_{max} represent the minimum and maximum values of the pixel
If $0 < C_{i,j} < 255$, $C_{i,j}$ is a non-corrupt pixel and does not change
If $C_{i,j}$ is a noisy pixel (e.g. $C_{i,j} = 0 \vee C_{i,j} = 255$) then
Case 1: If $B_{min} = 0 \wedge B_{min} = 255$ then
$C_{i,j} = mean\ (B)$ without $B_{min} = 0$ and $B_{max} = 255$
End
Case 2: If (all the elements of $B = 0 \vee B = 255$) then
$C_{i,j} = 255$
End
Case 3: If $B_{min} > 0 \wedge B_{min} < 255$
$C_{i,j} = mean\ (B)$
End
Repeat Steps (6) to (8) for all the pixels in the image $I(x, y)$ during $\frac{n}{10} + 1$ iterations (n corresponds to the noise percentage)

Figure-9. Pseudocode of the algorithm proposed in [2].

To observe in more detail the performance of the algorithms, a comparison of the original images versus the resulting images was carried out with each of the algorithms, and through the SSIM index, a series of values ranging from 0 to 1 was established, where zero (0) equals

the total loss of similarity and one (1) is absolute similarity. The values are presented in the tables from 1 to 4. Again, it is important to emphasize that the results with which the values obtained in this research are compared, were obtained through the replication of the work done by the authors named above, through the information provided by them (pseudo-codes, mathematical models or flow charts).

Table-1. SSIM results for different noise levels Baboon (512x512).

Baboon (512x512)			
Noise %	**Abdel Algorithm**	**Dimitrios Algorithm**	**MERIG**
10	0,9265	0,9767	0,9772
20	0,7973	0,9450	0,9529
30	0,6567	0,8948	0,9237
40	0,5287	0,8102	0,8886
50	0,4180	0,6968	0,8467
60	0,3161	0,5574	0,7927
70	0,2333	0,4082	0,7232
80	0,1560	0,2715	0,6266
90	0,0908	0,1469	0,4740

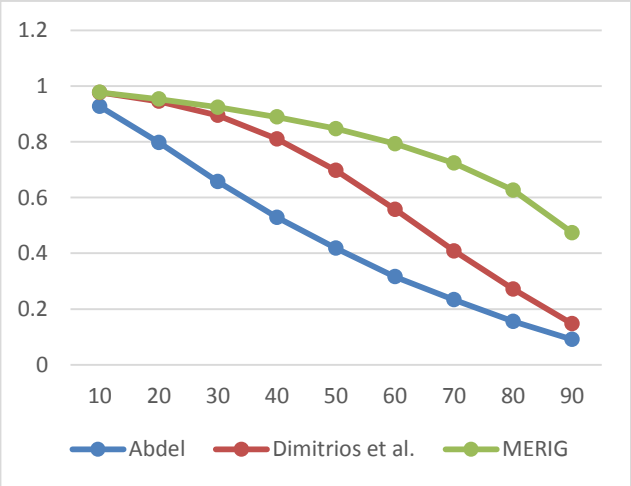


Figure-10. Percentage of noise vs. SSIM for the image of the Baboon (512x512).

As can be seen in Figure-10, the algorithms of Abdel and Dimitrios, have an approximation to 0 as the percentage of noise increases, which indicates that these algorithms present difficulties in eliminating noise when it is very high. On the other hand, the Proposed Algorithm (MERIG) is affected to a lesser degree when the noise level increases, although Figure 10 shows a steeper drop when noise levels exceed 50%, however, it still presents the best noise restoration in terms of image quality.

According to the results obtained in Tables 1, 2, 3, and 4, it can be seen that the proposed algorithm presents a better performance with respect to the algorithms taken as a reference, in most of the different noise levels considered since in any point of comparison the Proposed Algorithm was exceeded.

It should be noted that the difference in the 10% noise images is much smaller than the difference when the noise percentage exceeds a level of 50%, which can be seen in more detail in Figures 10, 11, 12 and 13.

In Figure-14, the graphic results are found after the application of the different algorithms. Which are applied on two images with dimensions of 512x512 pixels that correspond to the image of baboon (box 1) and Lenna (box 2). Also, these were applied to images with dimensions of 256x256 pixels, which are Lenna (box 3) and camera man (box 4).

Table-2. SSIM results for different noise levels Lenna (512x512).

Lenna (512x512)			
Noise %	**Abdel Algorithm**	**Dimitrios Algorithm**	**MERIG**
10	0,9532	0,9875	0,9889
20	0,7766	0,9615	0,9765
30	0,5507	0,9007	0,9632
40	0,3641	0,7830	0,9469
50	0,2468	0,6325	0,9270
60	0,1657	0,4693	0,8998
70	0,1132	0,3214	0,8634
80	0,0772	0,2024	0,8057
90	0,0467	0,1104	0,7063

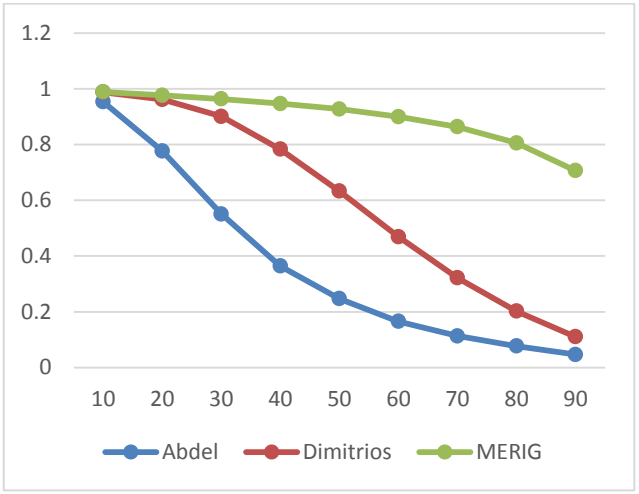


Figure-11. Percentage of noise vs. SSIM for the image of Lenna (512x512).

Table-3. SSIM results for different noise levels Lenna (256x256).

Lenna (256x256)			
Noise %	**Abdel Algorithm**	**Dimitrios Algorithm**	**MERIG**
10	0,9614	0,9875	0,9897
20	0,8084	0,9642	0,9784
30	0,6007	0,9062	0,9659
40	0,4351	0,8000	0,9499
50	0,3040	0,6671	0,9318
60	0,2162	0,5252	0,9010
70	0,1512	0,3828	0,8592
80	0,1020	0,2501	0,7943
90	0,0610	0,1417	0,6628

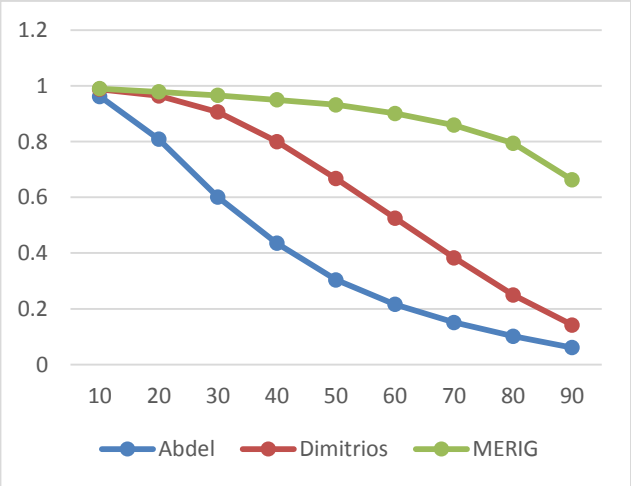


Figure-12. Percentage of noise vs. SSIM for the image of Lenna (256x256).

Table-4. SSIM results for different noise levels Cameraman (256x256).

Cameraman (256x256)			
Noise %	**Abdel Algorithm**	**Dimitrios Algorithm**	**MERIG**
10	0,9438	0,9767	0,9782
20	0,7748	0,9391	0,9582
30	0,5580	0,8741	0,9314
40	0,3994	0,7412	0,9062
50	0,2762	0,5867	0,8753
60	0,2087	0,4383	0,8337
70	0,1426	0,2990	0,7799
80	0,0965	0,1994	0,7100
90	0,0533	0,1129	0,6197

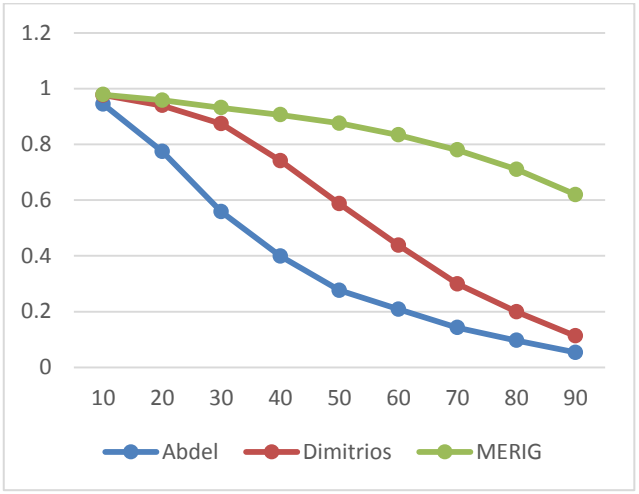


Figure-13. Percentage of noise vs. SSIM for the image of the Cameraman (256x256).

On the other hand, a percentage comparison was established based on the results obtained for each of the images using the SSIM index.

Table-5. Comparison SSIM values with 50% Noise.

	SSIM with 50% Noise			**Percentage difference (MERIG)**	
	Abdel	**Dimi-trios**	**MERIG**	**Abdel**	**Dimi-trios**
Image 1	0,4180	0,6968	0,8467	43%	15%
Image 2	0,2468	0,6325	0,9270	68%	29%
Image 3	0,0610	0,1417	0,6628	60%	52%
Image 4	0,2762	0,5867	0,8753	60%	29%

According to Table-5, it is possible to verify that the results obtained are significantly better than the algorithms compared. Specifically, and bearing in mind that images 1 and 2 correspond to matrices of 512x512, it can be seen that there is a difference between 43% and 68% with respect to Abdel Algorithm, and on the other hand, for Dimitrios Algorithm there is a variation between 15% and 29%.

In the case of images 3 and 4 with 256x256 dimensions, the existing variation with Abdel Algorithm. tended to stay at 60%. However, for the Dimitrios Algorithm. The range of variation is between 29% and 52%. Indicating that the restoration of images that this type present better quality in what is expressed by the Proposed Algorithm.

From Table-6, it can be seen that for the images with 512x512 dimensions the improvement is between 38% and 66% compared to the algorithm of Abdel, And between 33% and 60% in correspondence with the algorithm of Dimitrios. Also, in the case of smaller

images, the variation with respect to the algorithm of Abdel is found with a minimum of 57% and a maximum of 60%. Likewise, in relation to the algorithm of Dimitrios, the Proposed Algorithm (MERIG) improved the quality of the image by 51% and 52%.

Table-6. Comparison SSIM values with 90% Noise.

	SSIM with 90% Noise			**Percentage difference (MERIG)**	
	Abdel	**Dimi-trios**	**MERIG**	**Abdel**	**Dimi-trios**
Image 1	0,0908	0,1469	0,4740	38%	33%
Image 2	0,0467	0,1104	0,7063	66%	60%
Image 3	0,3040	0,6671	0,9318	60%	52%
Image 4	0,0533	0,1129	0,6197	57%	51%

From Table-6, it can be seen that for the images with 512x512 dimensions the improvement is between 38% and 66% compared to the algorithm of Abdel, And between 33% and 60% in correspondence with the algorithm of Dimitrios. Also, in the case of smaller images, the variation with respect to the algorithm of Abdel is found with a minimum of 57% and a maximum of 60%. Likewise, in relation to the algorithm of Dimitrios, the Proposed Algorithm (MERIG) improved the quality of the image by 51% and 52%.

In general terms, when the Proposed Algorithm tried to eliminate impulsive noise on a scale of 90%, it can be seen that despite having a level of restoration around 61%, the resulting image has an acceptable quality, keeping in mind that the amount of corrupted pixels is higher than normal pixels in a ratio of 9 to 1.

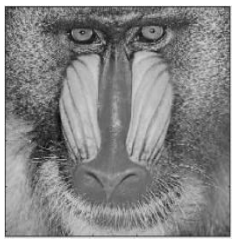
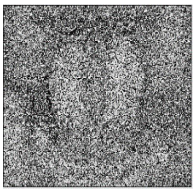
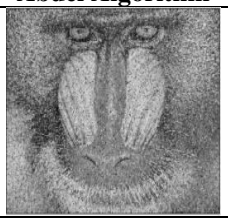
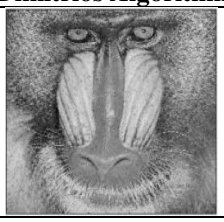
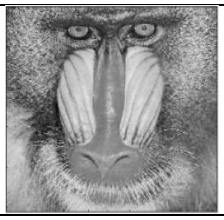
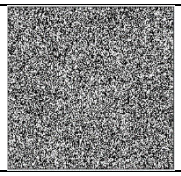
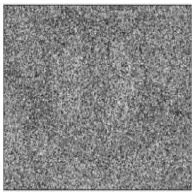
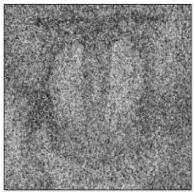
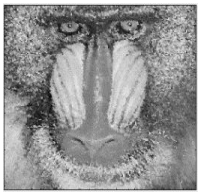
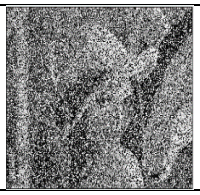
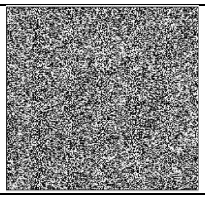
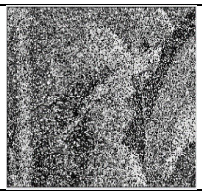
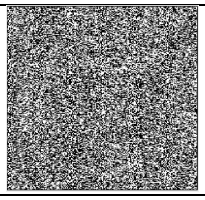
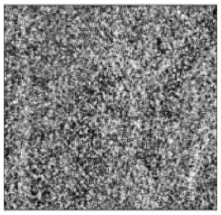
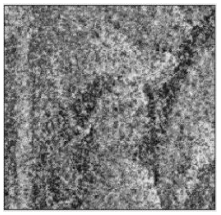
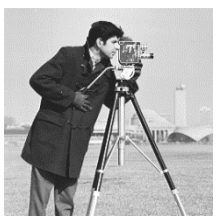
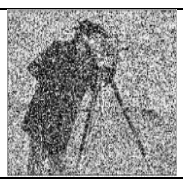
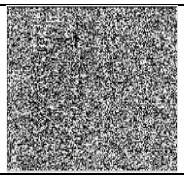
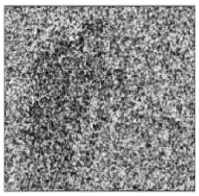
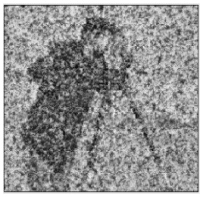
Original Image	Noise 50%	Abdel Algorithm	Dimitrios Algorithm	MERIG
 512x512				
	Noise 90%			
				
 512x512	Noise 50%			
				
	Noise 90%			
				
 256x256	Noise 50%			
				
	Noise 90%			
				
 256x256	Noise 50%			
				
	Noise 90%			
				

Figure-14. Resulting images after applying 50% and 90% noise.

5. CONCLUSIONS AND FUTURE WORKS

The use of bio-inspired systems allows to obtain efficient solutions to different situations, in the case of cellular automata it is possible to carry out tasks of considerable complexity through a simple and adequate development in computational terms, also, the implementation of an adaptive schema gives a better perception of the environment, which implies that the results both qualitatively and quantitatively are better because they have more information to reconstruct the images.

In general terms, the different developed algorithms present greater difficulty to operate in images with noise higher than 50%, however, the MERIG algorithm is more stable even with higher noise levels. Comparatively, this algorithm is superior in a range between 15% and 68% contemplating both the 256x256 and the 512x512 pixel images.

The restoration of the quality of the image with a noise level of 90% is one of the high points of everything that includes the elimination of noise, since there is very little useful in the environment. However, the MERIG Algorithm, by means of its adaptive behavior, exceeded the algorithms of Abdel between 38% and 66% and that of Dimitrios between 33% and 60%.

REFERENCES

[1] Betancourt A. Mujica and H. Tapias. 2003. Procesamiento difuso de imágenes: filtro difuso para eliminar el ruido impulsivo. Revista Ingeniería. 8(2): 40-46.

[2] D. Tourtounis, N. Mitianaudis and G. Sirakaulis. 2018. Salt-n-pepper noise filtering using Cellular Automata. Journal of Cellular Automata. 13(1-2): 81-101.

[3] I. Pitas and A. N. Venetsanopoulos. 1990. Nonlinear Digital Filters, Principles and Applications. New York: Springer US.

[4] R. Garnett, T. Huegerich, C. Chui and W. He. 2005. A Universal Noise Removal Algorithm with an Impulse Detector. IEEE Transactions on Image Processing. 14(11): 1747-1754.

[5] J. A. Echeverri, J. E. Rudas, R. Toscano and R. Ballesteros. 2011. Eliminación de ruido impulsivo en imágenes a color, utilizando interpolación con funciones de base radial. Revista Ingeniería. 16(1): 27-35.

[6] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam and C. H. Prem Chand. 2011. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Processing Letters. 18(5): 287-290.

[7] U. Sahin, S. Uguz and F. Sahin. 2014. Salt and pepper noise filtering with fuzzy- cellular automata. Computers & Electrical Engineering. 40(1): 59-69.

[8] R. H. Chan, C. W. Ho and M. Nikolova. 2005.b. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. IEEE Transactions on Image Processing. 14(10): 1479-1485.

[9] G. Bueno, J. Dorado. 2007. Gestión, procesado y análisis de imágenes biomédicas. España: Universidad de Castilla–La Mancha.

[10] J. V. Neumann. 1952. Theory of automata. Urbana: University of Illinois Press.

[11] N. Gómez. 2013. Vida artificial: ciencia e ingeniería de sistemas complejos. Bogotá: Editorial Universidad del Rosario.

[12] A. Dalhoum, I. Al-Dhamari, A. Ortega and M. Alfonseca. 2011. Enhanced Cellular Automata for Image Noise Removal. in Proceedings Asian Simulation Technology Conference (ASTEC 2011). pp. 69-73.

[13] Dalhoum, B. A. Mahafzah, A. Awwad, I. Al-Dhamari, A. Ortega and M. Alfonseca. 2012. Digital image scrambling method based on two dimensional cellular automata: a test of the lambda value. IEEE Multimedia. 19(4): 28-36.

[14] D. R. Nayak, R. Dash and B. Majhi. 2015. Salt and Pepper Noise Reduction Schemes Using Cellular Automata. in Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, New Delhi, India.

[15] P. Anand, and S. Agarwal. 2014. Training cellular automata for salt and pepper noise filtering. in Innovative Applications of Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH), Networking and Informatics, New Ghaziabad, India.

[16] P. Anand and S. Agarwal. 2014. Training two dimensional cellular automata for some morphological operations. in Innovative Applications of Computational Intelligence on Power, Energy and

Controls with their impact on Humanity (CIPECH), Networking and Informatics, New Ghaziabad, India.

[17] P. Anand, S. Chauhan and S. Agarwal. 2013. Training cellular automata for image thinning and thickening. in Confluence 2013: The Next Generation Information Technology Summit (4th International Conference), Noida, India.

[18] P. L. Rosin. 2006. Training Cellular Automata for Image Processing. IEEE Transactions on Image Processing. 15(7).

[19] W. Banzhaf. 2009. Self-organizing Systems. New York: Encyclopedia of Complexity and Systems Science. Springer.

[20] A. Rueda, and W. Torres. 2013. Autómatas celulares para la segmentación y clasificación de imágenes multiespectrales. in V Jornadas Nacionales de Geomática y Percepción Remota, Caracas, Venezuela.

[21] G. J. Martínez, A. Adamatzky, K. Morita and M. Margenstern. Computation with Competing Patterns in Life-Like Automaton. Game of Life Cellular Automata, Springer, London.pp. 547-572.

[22] P. Z. Pan, X. T. Feng and Zhou H. 2009. Solid cellular automaton method for the solution of physical field. in World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA.

[23] A. Nooraliei, and R. Iraji. 2010. Noise Reduction and Image Sharpening Using IJA Cellular Learning Automaton. in Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia.

[24] K. Mangalam and K. S. Venkatesh. 2017. Bitwise operations of cellular automaton on gray-scale images. in 28th Irish Signals and Systems Conference (ISSC), Killarney, Ireland.

[25] R. C. González and R. E Woods. 2006. Digital Image Processing. Upper Saddle River, NJ: Prentice-Hall, Inc.

[26] C. García and J. S. Viteri. 2009. Análisis e implementación de algoritmos para distorsionar imágenes con distintos tipos de ruido y aplicación de filtros en dos dimensiones para restaurarlas. M.S. thesis, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador.

[27] R. Castillo, J. M. Hernández, E. Inzunza and J. P. Torres. 2013. Procesamiento digital de imágenes empleando filtros espaciales. in Décima Segunda Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI 2013), Orlando, FL, USA.

[28] 2013. Procesamiento y Análisis de Señales e Imágenes. class notes for CCADET-UNAM, Posgrados de Ciencias de la Computación, Ingeniería Eléctrica y Física, Instituto de Ciencias Aplicadas y Tecnología, UNAM.