International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.04, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

AN INSIGHT INTO CELLULAR AUTOMATA-BASED IMPULSE NOISE FILTRATION ALGORITHMS

Zubair Jeelani[1], Fasel Qadir[2]

[1]Department of Computer Science, University of Kashmir, India
[2]Department of Computer Science, University of Kashmir, India

ABSTRACT

Application of Cellular Automata (CA) to Digital Image Processing has achieved considerable attention in last several years.CA are now used for noise filtration in digital images, particularly many CA-Based Impulse Noise Filters have been proposed. Impulse noise in an image is introduced during transmission over transmission media by external noise sources like atmospheric disturbances, or due to corrupted hardware memory locations or fault in camera sensors. In this paper we present a review of different CA-based impulse noise filtration algorithms with a focus on their use of Cellular Automata. A comparison of the algorithms on the basis of CA used, type of image they filter, parameters used, and type of impulse noise they filter used is also discussed.

Keywords: Cellular Automata, Image Enhancement, Impulse Noise, Noise Filtration, Salt And Pepper Noise

I.INTRODUCTION

Noise in image processing generally refers to unwanted data that conceal or distort the original information carried in a digital image. Digital images may get corrupted by different types of noise during different stages of image processing like image acquisition due to malfunctioning of the sensors in a digital camera, or during encoding and transmission, when the images are transferred over noisy transmission lines. Different types of noise like impulse noise, Gaussian noise, speckle noise may corrupt digital images. There are two types of impulse noise, salt and pepper noise and random valued noise. For images corrupted by salt and pepper noise, the noisy pixels can take only the maximum and the minimum values in the dynamic range [1]. Random valued noise on the other hand, may assume any value in the dynamic range.

Cellular Automata (CA) are dynamic, complex space and time discrete systems originally proposed by Stanislaw Ulam and John von Neumann in the 1940s as formal models for self-reproducing organisms [2][3]. 2D-CA is a grid of cells arranged in a rectangular area. Each cell can be in one of the finite number of states defined for the automaton. The cells change their current state after discrete time steps according to some pre-defined rules. The rule (*transition function*) uses the current state of the cell under consideration as well as the states of the neighboring cells to evaluate the next state of the cell. Efficiency of CA lies in the fact that most of

the logic can be implemented directly in hardware [4][5][6]. In last few years, many CA-Based noise filtration algorithms have been proposed. Many of these algorithms use variations of Totalistic Cellular Automata (TCA) [7][8][9] while others are based on Fuzzy Cellular Automata (FCA) [1][10]. In this paper, we concentrate on noise filtration algorithms based on variations of TCA.

The rest of this paper is organized as follows: section (2) describes the 2D-CA and TCA models ; section (3) presents the general TCA-Based noise filtration model; section (4) presents the different parameters used for measuring the effectiveness of noise filtration capability of a given technique, section (5) provides the review of different CA-Based noise filtration algorithms followed by the conclusion of the review in section (6).

1.1Two Dimensional Cellular Automata (2D-CA)

A 2D-CA *C* consists of a two dimensional lattice (Z^2) of square cells. Each cell has a state at any given time, *t.* The 2D-CA evolve by changing the state of its cells at time *t+1* synchronously using a local transition function δ. A 2D-CA can thus be represented by a triple formally [11] as:

$$C = (S, N, \delta) \tag{1}$$

where, S represent the non-empty set of states, $N \subseteq Z^2$ represent the neighbourhood, and $\delta: S^N \rightarrow S$ define the local transition function of C.

1.1.1 Local Neighbourhood

The local neighbourhood N of a cell C(*i,j*) is a subset of the CA lattice Z^2. A brief review of the two most studied local neighborhoods is given in the following two sub subsections.

1.1.2 von Neumann Neighbourhood (N_{vN})

N_{vN} consists of a given cell C(*i,j*) and the cells located on its left, right, top and bottom as shown in Fig. 1(a). The cells in N_{vN} of a cell C(i,j) can be represented as follows [12]:

$$N_{vN}(i,j) = [(i',j') : |i' - i| + |j' - j| \leq 1] \tag{2}$$

1.1.3 Moore Neighbourhood (N_M)

N_M of a given cell C(*i,j*) consists of its N_{vN} and the cells located on its top left, top right, bottom left and bottom right as shown in Fig. 1(b). The cells in N_M of a cell C(i,j) can be represented as follows [12]:

$$N_M(i,j) = [(i',j') : |i' - i| \leq 1, |j' - j| \leq 1] \tag{3}$$

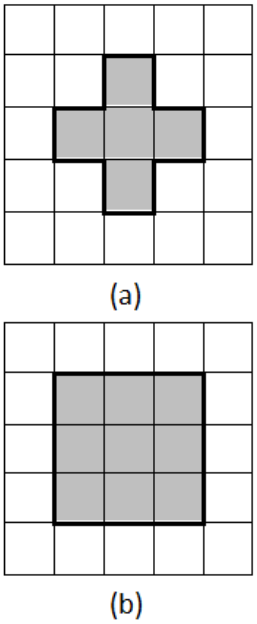

Figure 1: (a) von Neumann Neighbourhood (N_{vN}); (b) Moore Neighbourhood (N_M)

1.1.4 Boundary Conditions

The cells at the boundaries of a CA lattice don't have all the cells in its neighbourhood and the boundary conditions determine these cells by adding a boundary of cells around the CA lattice. Closed Boundary

International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.04, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Condition (BC_C) adds a boundary of cells with 0-state around the CA lattice Z^2 [13]. Periodic Boundary Condition (BC_P) makes last row adjacent to first row and last column adjacent to the first column. The resulting geometric figure can be thought of as a toroid.

1.1.5 Totalistic Cellular Automata (TCA)

TCA are 2D-CA in which the new state of cell C(*i,j*) is computed as a function (δ) of the state of the cell under observation and the states of the cells in its neighbourhood taken together. All the cells change state simultaneously. In Fig. 2, we show the conceptual model of TCA. Although Fig. 2 shows evolution of only one cell by applying local transition function (δ) but in practice all the cells change state simultaneously.

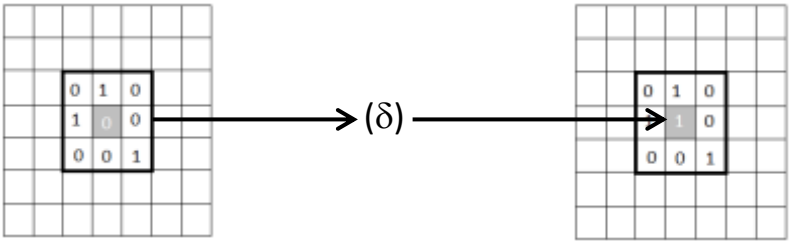

Figure 2: Local Transition Function (δ) compute next state of a cell as a function of cells in its neighbourhood (N_M)

The local transition function (δ) for TCA takes the form [a]:

$$C(i,j)^{(t+1)} = \delta\left(\sum_{i',j'} C(i',j')^{(t)}\right) \quad (4)$$

where, $C(i'j') = N_{vN}$ or N_M

1.1.1. Outer Totalistic Cellular Automata (OTCA)

OTCA is a variation of TCA in which the new state of cell (C(i,j)) is computed as a function (δ) of the state of the cell under observation and the states of the cells in its neighbourhood taken separately.

The local transition function (δ) for OTCA thus takes the form [11]:

$$C(i,j)^{(t+1)} = \delta\left(C(i,j)^{(t+1)}, \sum_{i',j'} C(i',j')^{(t)}\right) \quad (5)$$

where, $C(i'j') = N_{vN}$ or N_M, but excluding the cell C(i,j).

II .CA for noise filtration

The input image (possibly corrupted with impulse noise) is analyzed by a CA/Non-CA based module. If the image is not corrupted by noise, it is accepted. Otherwise, an initial CA configuration is initialized using the gray-scale or binary intensity values of the corrupted image. Fig. 3 shows the flow chart of the process.

The TCA or OTCA local transition function (δ) or a variation of these functions is applied to the initial CA to get the first generation of the processed image. The image is checked for noise reduction using a parameter like PSNR (Peak Signal to Noise Ratio). If the noise is reduced to acceptable levels the image is accepted, otherwise

the local transition function is applied to first generation CA to further improve the image. The local transition function is applied iteratively till the noise is reduced to acceptable levels.

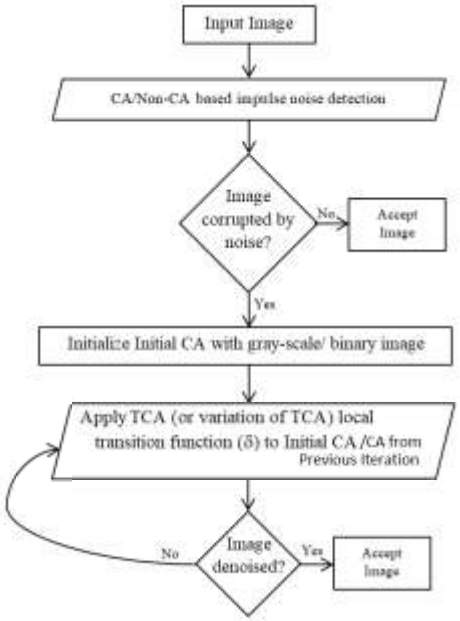

Figure 3: General TCA-Based Noise Filtration

III. NOISE FILTRATION PARAMETERS

3.1 Peak Signal to Noise Ratio (PSNR)

PSNR refers to the ratio between the maximum possible power of a signal and the power of the corrupting noise that affects the reliability of the signal. The PSNR between a gray-scale reference image *f* and a gray-scale test image *g* is defined as [14]:

$$PSNR(f,g) = 10log_{10}(255^2/MSE(f,g)) \quad (6)$$

where,

$$MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij} - g_{ij})^2 \quad (7)$$

3.2 Structural Similarity (SSIM)

The SSIM quality metric was developed by Wang *et al.*[e]. SSIM is used to measure the similarity between two images. SSIM uses three factors to model the image distortion; (i) loss of correlation, (ii) luminance distortion, and (iii) contrast distortion. The SSIM for a gray-scale reference image *f* and a gray-scale test image *g* is defined as [14]:

$$SSIM(f,g) = l(f,g)c(f,g)s(f,g) \quad (8)$$

where,

$l(f,g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2\mu_g^2 + C_1}$ represents the function to measure the closeness between the mean luminance (μ_f and μ_g) of images *f* and *g*. This factor is maximal and equal to 1 only if $\mu_f = \mu_g$.

$c(f,g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2\sigma_g^2 + C_2}$ represents the function to measure the closeness between the contrast (measured by standard deviation, σ_f and σ_g) of images *f* and *g*. This factor is maximal and equal to 1 only if $\sigma_f = \sigma_g$.

$s(f,g) = \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3}$ represents the structure comparison function and is used to measure the correlation coefficient between the two images. (σ_{fg} represents the covariance between the images *f* and *g*).

The SSIM index value ranges between 0 and 1 inclusive. A value of 0 for SSIM indicates that the two images *f* and *g* have no correlation. Whereas, SSIM value of 1 indicates the maximum possible correlation between the two images, i.e. *f* = *g*.

3.3 Hamming Distance (HD)

The hamming distance between two images *f* and *g* is calculated as follows [9]:

$$HD = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(f_{ij}^2 \oplus g_{ij}^2)}{m \times n} \quad (9)$$

where, *f* is the reference image and *g* is the test image; (m × n) is the size of images *f* and *g*.

Smaller values of HD indicate better noise filtration. Both SSIM and PSNR are considered better parameters than HD [9]. HD gives only the number of different pixels in the two images whereas PSNR computes the degree of difference between the two images and SSIM provide even better comparison by considering three important factors as discussed in above section.

IV CA BASED NOISE FILTRATION ALGORITHMS -

Liu *et al.* [7] proposed a Salt and Pepper noise filter based on Cellular Automata with Moore Neighbourhood (N_M). The algorithm is compared with the median filter and the CA-Based filter proposed in [15] and is shown to perform better than these algorithms particularly with noise density > 0.4. The use of M_N with the local transition function is shown to perform better when the density of salt and pepper noise is high.

The algorithm employs the TCA local transition function (δ) with N_M as shown in (4) above. As such, all the pixels in the Moore Neighbourhood are considered while computing the new value for the pixel under consideration.

Let $Max(C(i,j)^t, N_M)$ returns the cell/pixel with the maximum intensity value in the N_M of cell/pixel $C(i,j)$ at time *t. Similarly,* $Min(C(i,j)^t, N_M)$ returns the cell/pixel with the minimum intensity value in the N_M of cell/pixel $C(i,j)$ at time *t.* The Pseudocode for the algorithm is presented as under:

International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.04, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Algorithm 1

$C_{max} \leftarrow$ **Max**$(C(i,j)^t, N_M)$
$C_{max} \leftarrow$ **Min**$(C(i,j)^t, N_M)$
if $C_{min} < C(i,j)^t$ **and** $C(i,j)^t < C_{max}$ **then**
    $C(i,j)^{t+1} \leftarrow C(i,j)^t$
else if $C_{min} = C_{max}$ **or** $|S|$ for N_M of $C(i,j)^t = 2$ **then** ▶S is the non-empty finite set of states
    if $C_{min} \neq 0$ **then**
        $C(i,j)^{t+1} \leftarrow C_{min}$
    else if $C_{max} \neq 255$ **then**
        $C(i,j)^{t+1} \leftarrow C_{max}$
    else
        $C(i,j)^{t+1} \leftarrow C(i,j)^t$
    end
else
    $C_{mean} \leftarrow$ **Mean**$(N_M - \{C_{min}, C_{max}\})$ ▶Calculate mean of cells leaving cell C_{min} and C_{max}
    if Abs$(C(i,j)^t - C_{max}) <$ *threshold* **then**
        $C(i,j)^{t+1} \leftarrow C(i,j)^t$
    else
        $C(i,j)^{t+1} \leftarrow C_{mean}$
    end
end

The *threshold* variable is used to define the acceptable levels of noise.

The experimental image used in the experiment is classic lena.jpg image with 256 × 256 size. The denoising performance is evaluated using the Hamming Distance (HD) parameter discussed in section 4.3 above. HD values tending to 0 indicate better denoising. The first experiment uses lena.jpg image corrupted with 10% salt and pepper noise and after denoising HD value of 0.0013 is achieved against HD value of 0.0497 achieved by applying the technique proposed in [15]. The second experiment uses lena.jpg image with 10%, 20% and 40% salt and pepper noise and the results reveal that the algorithm performs better than the median filter in all cases. The computational effort, however, required by the algorithm is considerable larger than that of the median filter.

Shukla and Agarwal [8] have proposed a CA-Based noise filtration algorithm. The algorithm works with a fixed boundary, i.e. local transition rule is applied to only non-boundary cells. The local transition function uses N_M and the experiment is applied to gray-scale images. The algorithm is discussed as under:

1. A set of n sample pixels $\{I_1(t), I_2(t), \dots, I_n(t)\}$ are chosen from N_M of a pixel C(i,j) at time *t*.
2. The intensity values are arranged in non-decreasing order as $\{I_{i1}(t), I_{i2}(t), \dots, I_{in}(t)\}$.
   where, $I_{i1}(t)$ represents the minimum intensity value, and $I_{in}(t)$ represents the maximum intensity value
3. The local transition function takes the general form:

$$C(i,j)^{(t+1)} = \delta\left(\sum_{i',j'} C(i',j')^{(t)}\right) \quad (10)$$

   where, $C(i',j') \subseteq N_M$

The specific definition of δ for calculating the state of cell at time *t+1* is given as under:

$$I^{(t+1)} = \frac{1}{n-2[\alpha n]} \sum_{j=[\alpha n]+1}^{n-[\alpha n]} I_j^{(t)} \quad (11)$$

where [.] denotes the greatest integer part and $0 \leq \alpha < 0.5$.

When the value of $\alpha = 0.5$, the filter behaves like a median filter and as the α value is decreased, i.e. as it tends more towards 0, the filter behaves like a mean or averaging filter. The algorithm, however, ignores the maximum and minimum intensity values of pixels under consideration.

A sample size of 10 images is used in the experiments at different levels of salt and pepper noise. The probability density function (PDF) of 0.01, 0.05, 0.10, 0.20 and 0.50 is introduced for salt and pepper noise. The PSNR and MSSIM (Mean SSIM) parameters are used for evaluation and comparison of the algorithm. The algorithm is compared with 3×3 Median Filter, Weiner Filter, and Average Filter and is shown to perform consistently better when compared to these filters.

Dalhoum *et al.* [9] proposed a CA-Based Filter for two types of noise; salt and pepper noise and uniform noise. The algorithm is compared with the one proposed by [7] and the other by [16] and is shown to have better denoising effect. The algorithm begins by computing the histogram of the noisy image to detect the type of noise. The salt and pepper noise is if the frequency 0 and/or 255 values is high. Otherwise, the image is assumed to be corrupted by uniform noise. For a cell C(i,j), the algorithm computes the next state as shown in the Pseudocode below:

Algorithm 2

```
if uniform noise then
    Cmax ← Max(C(i,j)^t, NM)
    Cmax ← Min(C(i,j)^t, NM)
    if C(i,j)^t = Cmin or C(i,j)^t = Cmax then
        Median(NM − {Cmin, Cmax})
    end
else                                          ►salt and pepper noise
    else if C(i,j)^t = 0 or C(i,j)^t = 255 then
    if Cmin ≠ 0 then
        if NM − {C(i,j)} ≠ 0 or NM − {C(i,j)} ≠ 0 then   ►No neighboring cell has a value 0 or 255
            Median(NM − {Cmin, Cmax})
        else
            Mean(NM − {C(i,j)}
        end
    end
end
```

The algorithm is evaluated using the MSE and HD parameters. Noise percentage of 5%, 10%, 25%, 50%, 75%, 90% and 95% is used for uniform noise and salt and pepper noise. The algorithm is run for five iterations for each image and the MSE and HD is recorded. Test results have shown that the proposed algorithm perform better in terms of both MSE and HD when compared to those proposed in [7] and [16].

Table 1: Comparison of CA-Based Noise Filtration algorithms

Algorithm	Local Transition Function (δ)	Boundary Condition	Neighbor-hood	Image type	Noise Type	Paramet-ers used	Compared with
Liu *et al.* [7]	The transition function considers 256 states (0-255) and in one special case it requires to calculate the *mean* of neigbouring cells of the cell under observation.	Not mentioned	Moore (N_M)	Gray-scale	Salt and Pepper Noise	Hamming Distance (HD)	Median Filter Wang *et al.* [15]
Shukla and Agarwal [8]	The transition function behaves more like mean or median filter depending on the value of α. The function considers a subset of cells.	Fixed boundary	Moore (N_M)	Gray-scale	Salt and Pepper Noise Gaussian Noise	PSNR MSSIM	Median Filter Weiner Filter Gaussian Filter
Dalhoum *et al.*[9]	The transition function basically uses the mean of neighbourhood in some cases and median in other cases.	Not mentioned	Not mentioned [we assume N_M] here.	Gray-scale	Salt and Pepper Noise Uniform Noise	Mean Square Error (MSE) HD	Liu *et al* [7] Chang *et al.* [16]

Although, the algorithms reviewed above use the concept of Cellular Automata but they still rely on other filters like the median filter [8][9] or mean filter [7][8], as shown in Table 1, to perform effectively. Algorithms proposed in [8] and [9] make use of the mean and median filters to a great extent and as such it is difficult for these algorithms to take advantage of the inherent parallelism available with the Cellular Automata.

IV.CONCLUSION

In this paper we presented a review of different CA-Based noise filtration algorithms. Although all the reviewed noise filtration models show good filtration results on the parameters taken for evaluation but we are of the view that the extent of the use of Cellular Automata in these models is not up to the mark. As such, there is a need to relook at the application of Totalistic Cellular Automata and its variations in image noise filtration. More research needs to be done to systematically explore the huge rule space provided by the Totalistic Cellular Automata and the application of these rules in image denoising algorithms.

REFERENCES

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

[2] J. von Neumann, *Theory of self-reproducing automata* (University of Illinois Press, 1966).

[3] A. W. Burks, Ed., *Essays on cellular automata* (University of Illinois Press, 1970).

[4] K. Preston, M. Duff, *Modern cellular automata: theory and applications* (Plenum Press, New York, 1984).

[5] M. Halbach, R. Hoffman, Implementing cellular automata in FPGA logic, in 18th International Parallel And Distributed Processing Symposium, 2004, IEEE, Santa Fe, USA, 2004, 258-262.

[6] T. Toffoli, N. Margolus, *Cellular automata machines: a new environment for modeling* (MIT Press, Cambridge, Mass., 1987).

[7] S. Liu, H. Chen, and S. Yang, An Effective Filtering Algorithm for Image Salt-Pepper Noises Based on Cellular Automata, 2008 Congress on Image and Signal Processing, IEEE, Sanya, China, 2008, 294-297.

[8] A. P. Shukla and S. Agarwal, An Enhanced Cellular Automata based Scheme for Noise Filtering, International Journal of Signal Processing, Image Processing and Pattern Recognition, 7(4), 2014, 231–242.

[9] A. L. A. Dalhoum, I. Al-Dhamari, A. Ortega, and M. Alfonseca, Enhanced cellular automata for image noise removal, In Proceedings of the Asian Simulation Technology Conference, Singapore, 2011, 69-73.

[10] S. Sadeghi, A. Rezvanian, and E. Kamrani, An efficient method for impulse noise reduction from images using fuzzy cellular automata, AEU - International Journal of Electronics and Communications, 66(9), 2012, 772–779.

[11] N. H. Packard, and S. Wolfram, Two-dimensional cellular automata, Journal of Statistical Physics, 38(5–6), 1985, 901–946.

[12] A.L.A. Dalhoum, B.A. Mahafzah, A.A. Awwad, I. Aldhamari, A. Ortega, and M. Alfonseca, Digital Image Scrambling Using 2D Cellular Automata, IEEE Multimedia, 19(4), 2012, 28-36.

[13] S. Torbey, Towards a framework for intuitive programming of cellular automata, Parallel Processing Letters, 19(1), 2009, 73-83.

[14] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," 2010 20th International Conference on Pattern Recognition, IEEE, Istanbul, Turkey, 2010, 2366-2369.

[15] W. Hai-ming, G. Shi-de, Y. Dao-heng, A new CA method for image processing based on morphology and coordinate logic, Computer Application Research, 81(1), 2004, 243-245.

[16] C. Chang, J. Hsiao, and C. Hsieh, An Adaptive Median Filter for Image Denoising, Second International Symposium on Intelligent Information Technology Application, IEEE, Shanghai, China, 2008, 346-350.