International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.4 (2014), pp.231-242
http://dx.doi.org/10.14257/ijsip.2014.7.4.23

An Enhanced Cellular Automata based Scheme for Noise Filtering

Anand Prakash Shukla[1] and Suneeta Agarwal[2]

[1]Krishna Institute of Enginering and Technoloty, Ghaziabad, India
[2]Motilal National Institute of Technology, Allahabad, India.
anandskla@gmail.com, suneeta@mnnit.ac.in

Abstract

Cellular Automata is a computational model used to describe the complex system through simple rules. It has been significantly applied to image processing operations. It appear as a natural tool for image processing because of the simplicity of mapping a digital image to a cellular automata and the ability of applying different image processing operations in real time. Noise filtering is considered to be an important operation of image processing. In this paper cellular automata based noise filter has been proposed for different levels of noise. The filter is also compared with some standard filters in terms of peak signal to noise ratio and structured symmetry index measure and it is found that the proposed model shows consistently better performance in terms of both the parameter.

Keywords-*Cellular automata, Noise filter, Gaussian noise, Salt & pepper noise, Moore Neighborhood, PSNR, MSSIM*

1. Introduction

Due to simple structure of cellular automata (CA) to model complex behavior system, it has attracted various researchers from different areas. Cellular automata primarily announced by Ulam[1] and Von Neumann [2] in 1950's and also discussed in the book of Wolfram 'A New Kind of Science'[3] with the purpose of obtaining models of biological self-reproduction. Nowadays cellular automata became very popular because of its diverse function and utility as a discrete model for many processes. Cellular automata also provide a concept for computational automata. Cellular automata also called Systems of Finite Automata *i.e.*, Deterministic Finite Automata (DFA) arranged in an infinite, regular lattice structure[4]. In cellular automata state of a cell at the next time step is determined by the current states of a surrounding neighborhood of cells along with its own state and is updated synchronously in discrete time steps.

Cellular automaton is a discrete dynamical system. Space, time, and the states of the system are discrete. Each point in a regular spatial lattice, called a cell, can have any one of a finite number of states. The states of the cells in the lattice are updated according to a local rule. That is, the state of a cell at a given time depends only on its own state one time step previously, and the states of its nearby neighbors at the previous time step. All cells on the lattice are updated synchronously. Thus the state of the entire lattice advances in discrete time steps. Formally, a (bi-directional, deterministic) cellular automaton is a triplet

$$A = (S;N;\delta),$$

Where, S is an non-empty state set, N is the neighborhood system, and $\delta: S^N \rightarrow S$ is the local transition function (rule).This function defines the rule of calculating the cell's state at t +1 time step, given the states of the neighborhood cells at previous time step t.

ISSN: 2005-4254 IJSIP

Commonly used neighborhood systems N are the von Neumann and Moore neighborhoods.

1.1. Von Neumann Neighborhood

The von Neumann neighborhood of range r is defined by the following equation

$$N(x_0, y_0) = \{(x, y) : |x - x0| + |y - y0| \leq r\} \qquad (1)$$

A diamond shaped neighborhood that can be used to define a set of cells surrounding a given cell (x_0,y_0) that may affect the evolution of a two dimensional cellular automaton on a square grid.

The von Neumann neighborhood is illustrated in figure 1(a)

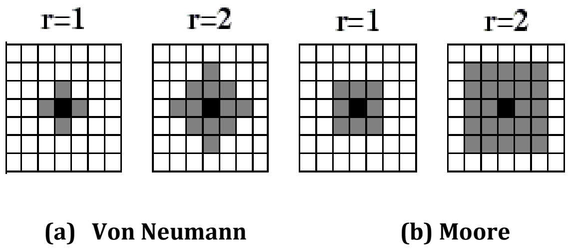

Figure 1. Neighborhood

1.2. Moore Neighborhood

A square-shaped neighborhood that can be used to define a set of cells surrounding a given cell (x_0,y_0) that may affect the evolution of a two dimensional cellular automaton on a square grid. The Moore neighborhood of range r is defined by

$$N(x_0, y_0) = \{(x, y) : |x - x_0| \leq r, |y - y_0| \leq r\} \qquad (2)$$

Moore neighborhood is illustrated in figure 1(b)

An image is a picture, photograph or any other form of two dimensional representation of any scene. Digital Images are electronic snapshots taken of a scene or scanned from documents.

Image noise is random variation of brightness or color information in images, an undesirable by product of image capturing that adds spurious and extraneous information. It can be produced by the sensor and circuitry of a scanner or digital camera. There are various types of noise such as Gaussian noise, Salt and pepper noise, Uniform noise, Shot noise, Anisotropic noise etc.

2. Related Work

There are various applications of Cellular Automata such as in complex systems' modeling, analyzing and controlling are: The Games of Life [5], cellular automata in environmental system and ecological system [6, 7], biological systems [8], cellular automata in the traffic system [9], cellular automata in machine learning and control [10] and CA in cryptography [11].

Digital image processing plays an important role in real life applications such as satellite television, computer tomography and magnetic resonance imaging. It is also used in areas of research and technology such as biological information systems and astrophysics [12]. CA is used in various image processing tasks such in Image Filtering in better way than some existing filters in denoising process [13-15], Border Detection in Digital Images that provide boundaries of images [16],CA in edge deduction [17, 18],

[19], Connected Set Morphology (applied on more than one image at a time), Thinning and Thickening of images[14], Image Segmentation which is an integral part of image processing applications like medical image analysis and photo editing[20], [16] and in Image Enhancement because of its dynamic behavior[21].

The advantage of cellular automata is that though each cell has an extremely limited view of the system (just its immediate neighbors) and each cell generally contains a few simple rules, the combination of a matrix of cells with their local interaction leads to more sophisticated emergent global behavior system *i.e*. The CA provides simplicity in complexity.

Popovici and Popovici [13] proposed the cellular automata based filter in which the Von Neumann neighborhood has been considered and the centre pixel changes its state on the basis of the majority of pixels present in the neighborhood. On comparison with the Gaussian filter, this method performs better image enhancement using CA. Also the performance of the Gaussian filter depends on the threshold value which is not true for the filter based on a cellular automaton.

Selvapeter and Hordijk [22] use Uniform cellular automata rules to filter impulse noise from both binary and gray scale images. Several modifications to the standard CA formulation are then applied to improve the filtering performance. For example, a random CA rule solves the noise propagation present in deterministic CA filters. A mirrored CA is used to solve the fixed boundary problem. Authors show that a two dimensional cellular automaton with a simple rules can be used as an efficient salt and pepper noise filter and a CA based on a Moore neighborhood performs better than the standard median filter, adaptive median filter, and switching filters for binary images. For gray level images, compared with a deterministic CA, a random CA performs better for a detailed image.

Rosin [14] trained the cellular automata rules and find the rules suitable for the noise filtering. In this work the rule set are first reduced on the basis of symmetry and mirror image, then the training has been performed by sequential floating forward search method, and the best rule set is obtained for noise filtering from the reduced rule set. In his next work, Rosin [23] uses 3 state automata for noise filtering. This introduces an alternative approach to reducing the number of cell states to enable more efficient training and application of CA to gray scale images. It is based on the texture unit texture spectrum (TUTS) method of texture analysis. This involves using a pixels value as a threshold for its neighbors. It is apparent that the TU/2 CA copes very well with all types of impulse noise, and in the extreme case of the stripe noise does very much better than the non-Cellular Automata approaches. On the other hand, for the Gaussian noise the cellular automata nonlinear filtering is less of an advantage. The loss of most of the image intensity information causes TU/2 CA to have slightly higher errors than TD/2 CA. Nevertheless, both TD/2 and TU/2 CA perform reasonably, even for Gaussian noise, outperforming the median, adaptive median and shock filters.

This paper concentrates on the new method of image noise filtering based on cellular automata and its effect on the various types of noises.

3. Proposed Work

A digital image can be assumed as a two dimensional array of m x n pixels, each with a particular gray value or color. An image can also be considered as the lattice configuration of a two dimensional cellular automata where each cell corresponds to an image pixel, and the possible states of the cells are the gray values or colors of that pixel. Following this philosophy, in the proposed work, the input image is considered as the two dimensional lattice of cells. The initial state of cells is the pixel value of that cell in the input image. Each cell interacts with its neighboring pixels and change its state according to the transition function defined in equation (3). In the current work fixed value boundary conditions are applied which means the update rule is only applied to non boundary cells.

The neighborhood selected is Moore neighborhood and the gray scale images are considered.

Let $\{I_1(t), I_2(t), I_n(t)\}$ be a set of n sample pixel values observed in a Moore neighborhood of the pixel (current cell of automata) at time t. If these values are arranged in ascending order of their intensity, the order statistics result is

$$\{I_{i1}(t), I_{i2}(t), ..., I_{in}(t)\}$$

where, $I_{i1}(t)$ is the minimum and $I_{in}(t)$ is the maximum. The central pixel changes its value at time t+1 as

$$I(t+1) = \frac{1}{n-2[\alpha n]} \sum_{j=[\alpha n]+1}^{n-[\alpha n]} I_j(t) \quad (3)$$

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

All the methods described in the section II of this paper consider all the pixel values on neighborhood for designing the transition function of cellular automata based filters. Popovici and Popovici [13] has considered all pixels in Von Neumann neighborhood and replaced the central pixel on the basis of majority pixels in the neighborhood. But Bednar and Watt [25] and Rytsar and Ivasenko [26] have shown that reducing the set of pixels under consideration improves the performance of the filter. Adopting the same philosophy, the filter shown in equation (3) has been designed. As it is easily seen from the equation that it indicates the percentage of the reduced samples from the Moore neighborhood. This filter performs like finding the median of all the cells in the neighborhood when α is close to 0.5, and moving towards averaging the neighbor pixels when α is close to 0. For simplicity and demonstration of the effect of the filter only the highest and the lowest pixel values are neglected and results are shown on the basis of same.

The main challenge is to simulate the behavior of cellular automata (parallel processing of pixels) in the system with single CPU (Central Processing Unit) i.e. in each step each image pixel should be processed in parallel. In order to simulate this, a temporary image of the same size as of input image has been taken and the processed pixels are stored in this temporary image. At the end of each iteration this temporary image is copied back to original image. The detailed algorithm is shown in Algorithm 1.

In the algorithm 1, A is the input image which has to be filtered by the proposed algorithm. First, creates the temporary image B, which is of the same size as input image A. Then for every pixel in the image A (except the boundary pixels) the pixels present in the Moore neighborhood of the pixel under consideration has been selected and arranged in order of their increasing intensity. Next, the highest and the lowest intensity values i.e. the first and the last pixels in the arranged list has been eliminated. Then the average of the remaining pixels intensity values has been calculated and at the end this value has been copied to the temporary image B. When all the pixels got processed then the image B is copied back to image A.

Algorithm 1 Cellular Automata based Noise Filter

Procedure CAFILTER

Create an image B of size as A

for i=2 to row - 1 and j= 2 to column-1 do

Take the pixels in Moore neighborhood of A(i,j)

Arrange neighbor pixels in increasing order

Eliminate the first and the last value

Find average of remaining pixels.

Put the calculated average in B(i,j)

end for

$A \leftarrow B.$

end procedure

4. Experiments and Results

The proposed algorithm has been implemented in MATLAB2014a. The algorithm is run for 10 sample gray scale images and tested for different level of salt and pepper noise and the Gaussian noise. First, in the image selected noise, with probability density function (PDF) 0.01, 0.05, 0.1, 0.2 and 0.5 respectively, has been introduced for both types of noises. Then the corrupted images are filtered by the proposed cellular automata based filter. The resulting (filtered) image is compared with the original image and the parameters peak signal to noise ratio(PSNR) and Mean Structured Similarity Index Measure(MSSIM)[24] have been calculated. The mean structural similarity index (MSSIM) is a method for measuring the similarity between two images. It is a full reference metric i.e. the measuring of image quality based on an initial noise free image as reference. The MSSIM differs with respect to other techniques such as mean square error measure(MSE) or PSNR is that these approaches estimate perceived errors; on the other hand, MSSIM considers image degradation as perceived change in structural information.

The proposed method is also compared with some standard filters. In the current experiment the Median filter, 3x3 Weiner filter, 5x5 Weiner filter and Average filter have been considered and results are found in the similar manner as for the proposed method. In the current paper the results of two images, Cameraman and Coins has been shown. The figure 2 and 3 show the effect of various filters on the cameraman image for filtering the Salt and Pepper noise with PDF = 0.01 and 0.5 respectively.

The PSNR and MSSIM values for the cameraman image are shown in Table 1 and 2 respectively.

Table 1. Table Showing the Effect of Filters in the Image Cameraman in Terms of PSNR Values for Salt & Pepper Noise

	Salt & Pepper Noise				
	PSNR				
	0.01	0.05	0.1	0.2	0.5
Median Filter	75.29	74.77	74.02	71.94	62.39
3x3 Weiner Filter	73.84	68.73	67.05	65.08	62.13
5x5 Weiner Filter	75.16	70.90	69.23	66.90	63.54
Gaussian Filter	75.97	69.69	66.81	63.61	59.43
Average Filter	72.79	71.14	69.60	67.06	62.82
Proposed CA filter	74.89	74.14	72.77	70.62	62.79

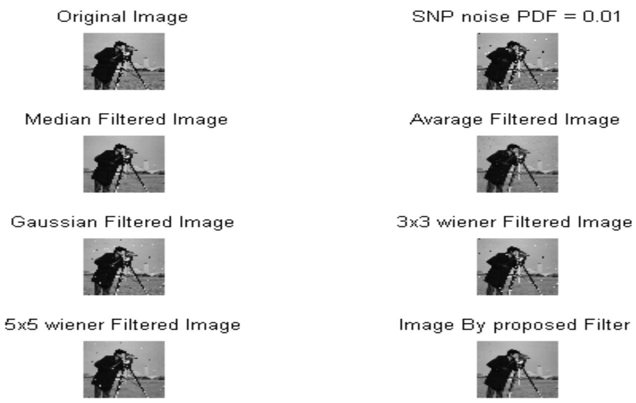

Figure 2. Effects of Filters on Image Corrupted by Salt Pepper Noise with PDF 0.01

Table 2. Table Showing the Effect of Filters on Cameraman in Terms of MSSIM Values for Salt & Pepper Noise

	Salt & Pepper Noise				
	MSSIM				
	0.01	0.05	0.1	0.2	0.5
Median Filter	0.87	0.86	0.86	0.81	0.24
3x3 Weiner Filter	0.74	0.45	0.34	0.24	0.12
5x5 Weiner Filter	0.76	0.57	0.48	0.35	0.20
Gaussian Filter	0.82	0.48	0.32	0.19	0.07
Average Filter	0.77	0.56	0.43	0.29	0.14
Proposed CA filter	0.85	0.81	0.75	0.49	0.23

The similar experiments has been performed for the Gaussian noise with PDF 0.01,0.05,0.1,0.2,0.5 for the same set of filters and PSNR and MSSIM are measured. The results of the effects of filters on Gaussian noise with PDF 0.01 and 0.5 for Cameraman image are shown in Figures 4 and 5 respectively.

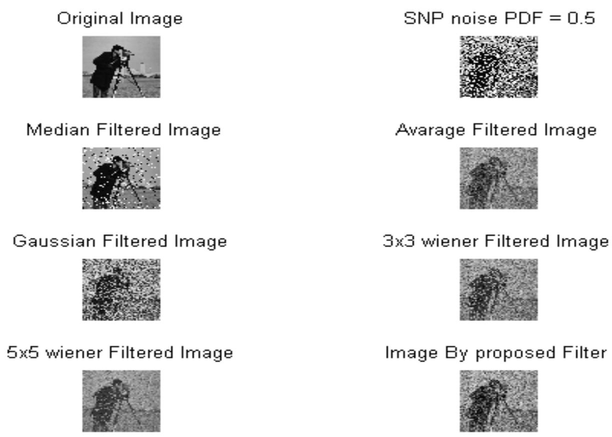

Figure 3. Effects of Filters on Image Corrupted by Salt Pepper Noise with PDF 0.5

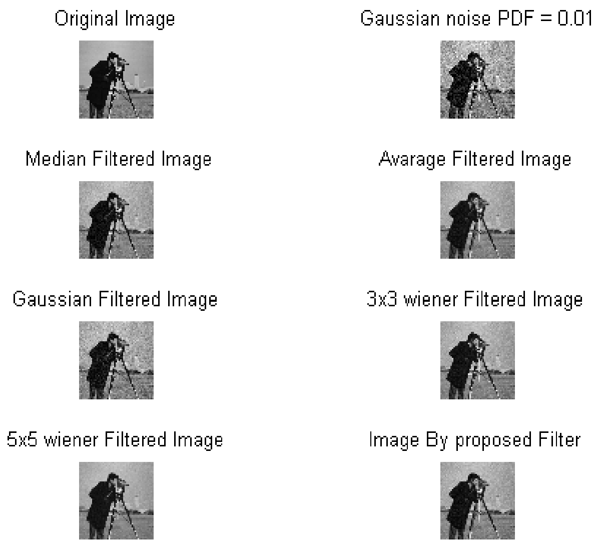

Figure 4. Effects of Filters on Image Corrupted by Gaussian Noise with PDF 0.01

Table 3. Table Showing the Effect of Filters on Cameraman in Terms of PSNR Values for Gaussian Noise

	Gaussian Noise				
	PSNR				
	0.01	0.05	0.1	0.2	0.5
Median Filter	72.17	70.29	66.80	61.81	55.44
3x3 Weiner Filter	74.05	71.15	67.20	62.01	55.64
5x5 Weiner Filter	74.00	71.19	67.23	62.02	55.64
Gaussian Filter	71.71	69.78	66.58	61.85	55.65
Average Filter	71.81	69.97	66.74	61.93	55.67
Proposed CA filter	72.12	70.21	66.83	61.89	55.58

The PSNR value and the MSSIM values for the cameraman image for Gaussian noise is shown in the Tables 3 and 4 respectively. Similarly, the results of the filters for the same parameters, as used in the cameraman image, for image Coins are shown in Figure 6, 7, 8 and 9 and in Tables 5, 6, 7 and 8.

The PSNR value and the MSSIM values for the cameraman image for Gaussian noise is shown in the Tables 3 and 4 respectively. Similarly, the results of the filters for the same parameters, as used in the cameraman image, for image Coins are shown in Figure 6, 7, 8 and 9 and in Tables 5, 6, 7 and 8.

Table 4. Table Showing the Effect of Filters on Cameraman in Terms of MSSIM Values for Gaussian Noise

	Gaussian Noise				
	MSSIM				
	0.01	0.05	0.1	0.2	0.5
Median Filter	0.54	0.52	0.49	0.45	0.52
3x3 Weiner Filter	0.63	0.60	0.57	0.53	0.53
5x5 Weiner Filter	0.71	0.68	0.65	0.60	0.52
Gaussian Filter	0.45	0.44	0.41	0.39	0.52
Average Filter	0.60	0.57	0.54	0.50	0.51
Proposed CA filter	0.60	0.57	0.54	0.50	0.52

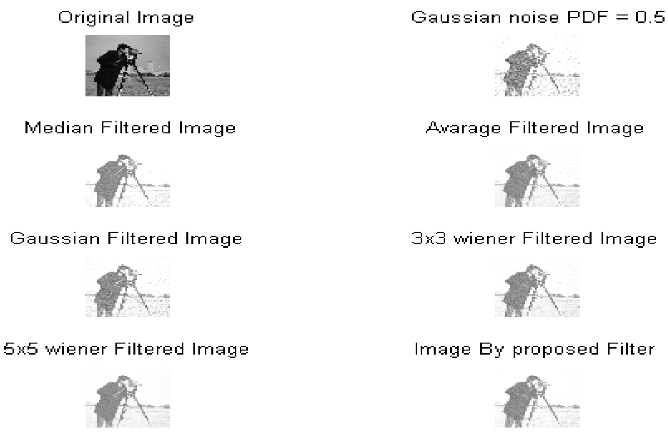

Figure 5. Effects of Filters on Image Corrupted by Gaussian Noise with PDF 0.5

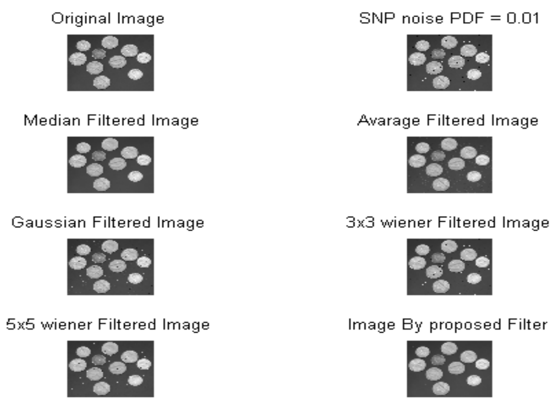

Figure 6. Effects of Filters on Image Corrupted by Salt & Pepper Noise PDF 0.01

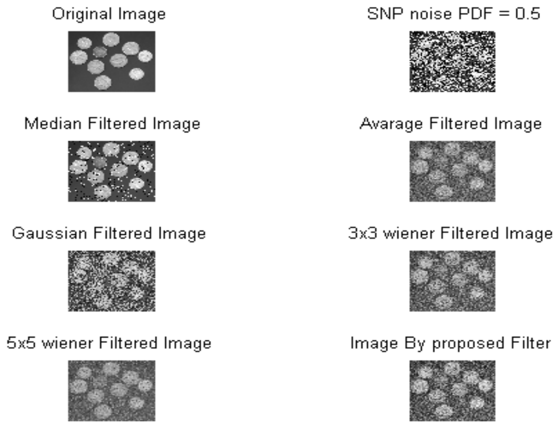

Figure 7. Effects of Filters on Image Corrupted by Salt & Pepper Noise PDF 0.5

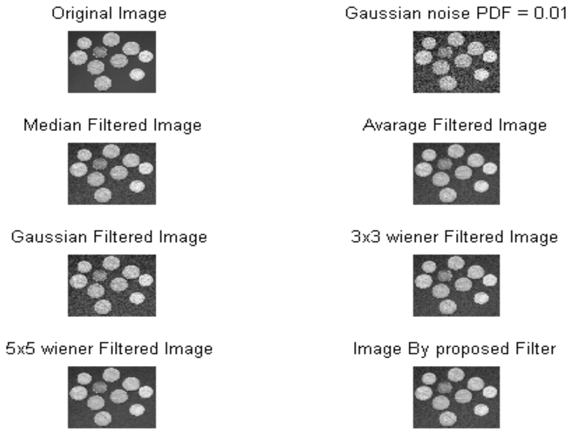

Figure 8. Effects of Filters on Image Corrupted by Gaussian Noise with PDF 0.01

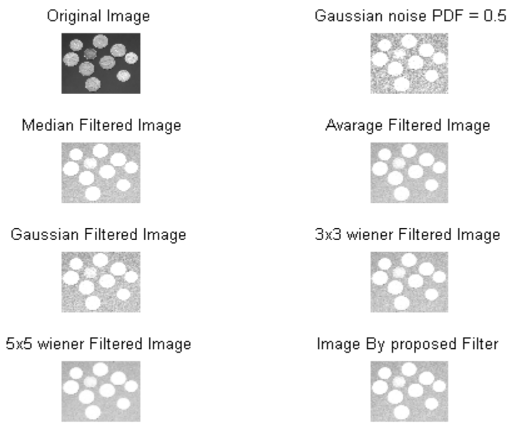

Figure 9. Effects of Filters on Image Corrupted by Gaussian Noise PDF 0.5

Table 5. Table Showing the Effect of Filters in the Image Coins in Terms of PSNR Values for Salt & Pepper Noise

	Salt & Pepper Noise				
	PSNR				
	0.01	0.05	0.1	0.2	0.5
Median Filter	82.42	81.32	80.24	76.29	63.19
3x3 Weiner Filter	73.90	68.58	66.88	65.63	62.58
5x5 Weiner Filter	75.18	71.55	69.79	68.15	64.39
Gaussian Filter	76.68	70.07	66.88	63.85	59.58
Average Filter	77.21	73.81	71.22	68.24	63.31
Proposed CA filter	79.84	77.89	75.82	73.34	63.80

Table 6. Table Showing the Effect of Filters in the Image Coins in Terms of MSSIM Values for Salt & Pepper Noise

	Salt & Pepper Noise				
	MSSIM				
	0.01	0.05	0.1	0.2	0.5
Median Filter	0.95	0.95	0.94	0.89	0.25
3x3 Weiner Filter	0.76	0.43	0.29	0.23	0.13
5x5 Weiner Filter	0.76	0.58	0.46	0.36	0.22
Gaussian Filter	0.81	0.46	0.28	0.16	0.06
Average Filter	0.83	0.60	0.45	0.31	0.15
Proposed CA filter	0.93	0.86	0.79	0.62	0.24

Table 7. Table Showing the Effect of Filters in the Image Coins in Terms of PSNR Values for Gaussian Noise

	Gaussian Noise				
	PSNR				
	0.01	0.05	0.1	0.2	0.5
Median Filter	74.24	71.46	67.32	62.03	55.08
3x3 Weiner Filter	74.86	71.73	67.46	62.17	55.10
5x5 Weiner Filter	75.94	72.27	67.67	62.24	55.12
Gaussian Filter	71.82	69.98	66.75	61.98	55.09
Average Filter	74.94	71.82	67.54	62.24	55.16
Proposed CA filter	75.02	71.85	67.51	62.16	55.14

Table 8. Table Showing the Effect of Filters in the Image Coins in Terms of MSSIM Values for Gaussian Noise

	Gaussian Noise				
	MSSIM				
	0.01	0.05	0.1	0.2	0.5
Median Filter	0.56	0.56	0.54	0.50	0.33
3x3 Weiner Filter	0.60	0.59	0.57	0.52	0.33
5x5 Weiner Filter	0.75	0.74	0.72	0.65	0.43
Gaussian Filter	0.40	0.40	0.39	0.37	0.26
Average Filter	0.62	0.62	0.60	0.55	0.37
Proposed CA filter	0.62	0.62	0.60	0.55	0.37

From the Table 1, 2, 5 and 6 it has been observed that for low level of salt and pepper noise the proposed filter performs better than other filters except the median filter and Weiner filters in terms of PSNR value. But as the noise level increases the performance of the proposed filter also increases with respect to other filters. In terms of structural similarity it maintains a good track; the only filter which is superior in terms of MSSIM is the median filter.

When the proposed filter is compared in Gaussian noise, whose results are shown in Tables 3, 4, 7 and 8, it has been observed that the proposed filter performs better than other filters in terms of both PSNR and MSSIM, even better than the median filter. The only filter which slightly performs better than the proposed one is the Weiner filter.

It is clear from the above discussion that the proposed filter performs well for both Salt and pepper noise and Gaussian noise at low to high level of noise. Some filters like median filter perform well with the salt and pepper noise but its performance deteriorates with the Gaussian noise on the other hand some filters like Weiner filter perform well with Gaussian noise but not with salt and pepper noise. The proposed filter shows the consistent performance with both types of noises as well as with respect to both parameters viz. PSNR and MSSIM.

5. Conclusion and Future Works

In this paper, a cellular automata based image noise filter has been proposed. The proposed filter is compared with some standard filters and the performance has been measured in terms of the PSNR and MSSIM values, for Salt and pepper noise as well as Gaussian noise. The careful analysis of results shows that the proposed filter performs consistently for all tested conditions *i.e.*, different noise with different level of PDF. Although the proposed filter is motivated from subsequently different design principle (CA based) but its performance is encouraging. In future some more cellular automata based filters may be designed which contains more complicated neighborhood patterns. Also the current experiment is based on Moore neighborhood for r=1, the filters with higher range may be considered. Also designing the transition function (rules for changing the state of CA) is also a big challenge as future prospective of this work.

References

[1] S. Ulam, "Some ideas and prospects in biomathematics," Annual review of biophysics and bioengineering, vol. 1, no. 1, **(1972)**, pp. 277–292.
[2] I. J. Von Neumannn and A. W. Burks, "Theory of self-reproducing automata", **(1966)**.
[3] S. Wolfram, "A new kind of science", Wolfram media Champaign, vol. 5, **(2002)**.
[4] M. Sipper, "The evolution of parallel cellular machines: Toward evolware", BioSystems, vol. 42, no. 1, **(1997)**, pp. 29–43.
[5] M. Gardner, "Mathematical games: The fantastic combinations of john conways new solitaire game life", Scientific American, vol. 223, no. 4, **(1970)**, pp. 120–123.
[6] S. Aassine and M. C. El Jaı, "Vegetation dynamics modelling: a method for coupling local and space dynamics", Ecological modelling, vol. 154, no. 3, **(2002)**, pp. 237–249.
[7] R. Smith, "The application of cellular automata to the erosion of landforms", Earth Surface Processes and Landforms, vol. 16, no. 3, pp. 273–281, **(1991)**.
[8] H. De Garis, "Cam-brain the evolutionary engineering of a billion neuron artificial brain by 2001 which grows/evolves at electronic speeds inside a cellular automata machine (cam)", Towards evolvable hardware, Springer, **(1996)**, pp. 76-98.
[9] M. Fukui and Y. Ishibashi, "Traffic flow in 1d cellular automaton model including cars moving with high speed", Journal of the Physical Society of Japan, vol. 65, no. 6, **(1996)**, pp. 1868–1870.
[10] F. M. Marchese, "A directional diffusion algorithm on cellular automata for robot path-planning", Future Generation Computer Systems, vol. 18, no. 7, **(2002)**, pp. 983–994.
[11] S. Nandi, B. Kar and P. Pal Chaudhuri, "Theory and applications of cellular automata in cryptography", Computers, IEEE Transactions on, vol. 43, no. 12, **(1994)**, pp. 1346–1357.
[12] L. S. Davis, "A survey of edge detection techniques," Computer graphics and image processing, vol. 4, no. 3, pp. 248–270, (**1975).**
[13] A. Popovici and D. Popovici, "Cellular automata in image processing," in Fifteenth International Symposium on Mathematical Theory of Net works and Systems, vol. 1, **(2002).**
[14] P. L. Rosin , "Training cellular automata for image processing," Image Processing, IEEE**,** Transactions on, vol. 15, no. 7, pp. 2076–2087, **(2006).**
[15] A. Shukla, S. Chauhan, and S. Agarwal, "Training of cellular automata for image filtering," in Proc. Second International Conference on Advances in Computer Science and Application – CSA 2013,, pp. 86–95, **(2013).**
[16] Y. Y. Boykov and M.P. Jolly, "Interactive graph cuts for optimal boundary & region Segmentation of objects in ND images," in Computer Vision, ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol. 1. IEEE, pp. 105–112, **(2001).**
[17] C. L. Chang, Y.j. Zhang, and Y.Y. Gdong, "Cellular automata for edge detection of images," in Machine Learning and Cybernetics, Proceedings of 2004 International Conference on, vol. 6. IEEE, pp. 3830–3834, **(2004).**
[18] T. Kumar and G. Sahoo, "A novel method of edge detection using cellular automata," International Journal of Computer Applications, vol. 9, no. 4, pp. 0975–8887**,(2010).**
[19] S. Wongthanavasu and R. Sadananda, "Pixel-level edge detection using a cellular automata-based model," Advances in Intelligent Systems: Theory and Applications, vol. 59, pp. 343–351, **(2000).**
[20] M. Mitchell, J. P. Crutchfield, R. Das et al, "Evolving cellular automata with genetic algorithms: A review of recent work," in Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA96), **(1996).**
[21] G. Sahoo, T. Kumar, B. Raina, and C. Bhatia, "Text extraction and enhancement of binary images using cellular automata," International Journal of Automation and Computing, vol. 6, no. 3, pp. 254–260, **(2009).**
[22] P. J. Selvapeter and W. Hordijk, "Cellular automata for image noise filtering", in Nature & Biologically Inspired Computing, 2009. NaBIC World Congress on. IEEE, 2009, pp. 193–197, **(2009)**.
[23] P. L. Rosin, "Image processing using 3-state cellular automata," Computer vision and image understanding, vol. 114, no. 7, pp. 790–802**, (2010).**
[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," Image Processing, IEEE Transactions on, vol. 13, no. 4, pp. 600–612, **(2004).**
[25] Bednar, J., and T. Watt, "Alpha-trimmed means and their relationship to median filters." Acoustics, Speech and Signal Processing, IEEE Transactions on 32.1 145-153, **(1984)**.
[26] Y B Rytsar, and I B Ivasenko, "Application of (alpha, beta)-trimmed mean filtering for removal of additive noise from images", Optoelectronic and Hybrid Optical/Digital Systems for Image Processing pp 45-52**,(1997)**.

