Training Cellular Automata for Salt and Pepper Noise Filtering

Anand Prakash Shukla
Department of Computer Science
and Engineering
Motilal Nehru National Institute of Technology,
Allahabad,India.
Email:anandskla@gmail.com

Suneeta Agarwal
Department of Computer Science
and Engineering
Motilal Nehru National Institute of Technology,
Allahabad,India.
Email: sunita@mnnit.ac.in

***Abstract*—Cellular Automata is significantly applying to image processing operations. The description about the use of training of cellular automata for filtering the salt and pepper noise in binary images is exemplified in this paper. The selection of the best rule set from large search space has been performed on the basis of sequential floating forward search method.The peak signal to noise ratio values between original and filtered image is used as the objective function. The proposed method is also compared with some standard methods and found to perform better in respect to restoration of the image.**

I. Introduction

Cellular Automata is called Systems of Finite Automata, i.e. Deterministic Finite Automata (DFA) arranged in an infinite, regular lattice structure[1]. 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.

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[2] and Von Neuman [3] in 1950's and also discussed in the book of Wolfram 'A New Kind of Science'[4] with the purpose of obtaining models of biological self-reproduction.Cellular automata make up a very important class of completely discrete dynamical systems Now a day's Cellular Automata became very popular because of its diverse function and utility as a discrete model for many processes. Cellular Automata also provided a concept for computational automata.

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).

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

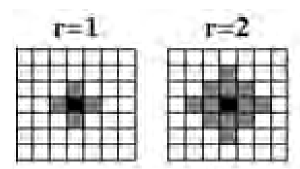

Fig. 1. von Neuman Neighborhood

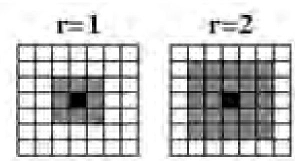

Fig. 2. Moore Neighborhood

1) Von Neumann Neighborhood: 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 of range r is defined by the equation 1.

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

The von Neumann neighborhood is illustrated in figure I-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 the equation 2

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

Moore neighborhoods is illustrated in figure I-2

A. Application of Cellular Automata

Some applications of cellular automata in complex systems' modeling, analyzing and controlling are: The Games of Life[5], biological systems[6], Cellular automata in environmental system and ecological system [7], [8], CA in edge deduction[9], [10], [11], Cellular automata in the traffic system[12], CA in image processing[13], Cellular automata in machine learning and control[14] and CA in cryptography[15]. Digital image processing plays an important role in real life applications such as satellite television, computer tomography and magnetic resonance imaging as well as in areas of research and technology such as biological information systems and astrophysics[16].CA is used in various image processing tasks such in Image Filtering in better way than some existed filters in denoising process[17], [18], Border Detection in Digital Images that provide boundaries of images[19], Connected Set Morphology (applied on more than one image at a time), Thinning and Thickening of images[17], Image Segmentation which is an integral part of image processing applications like medical image analysis and photo editing[20], [19] and in Image Enhancement because of its dynamic behaviour[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. This paper concentrates on training of CA for image's thinning and thickening process with their applications in different fields.

II. Training of Cellular Automata

Cellular automata training is basically used to acquire knowledge, skill to improve capability, capacity and performance of cellular automata in image processing like- noise filtering, Thinning, Convex Hull, calculating distance features, Template Matching, Image Sharpening, Simple Object Recognition using rule sets which provide specific operations to their states at each step of time.

A. Related work

To find out the desired rule set, feature selection is acquiescent and can be performed using branch and bound algorithms[22]. In automatically learning of rules most of researchers focus in the density classification problem[20] that is recognized as a standard for exploring cellular automata rules with universal properties further applied standard genetic algorithm(GA) for learning rules[20][23].A standard genetic programming scaffold is used in learning and training of cellular automata[24]. There are various feature selection methods that were introduced also consider GA and SFFS (sequential floating forward search)[24], [25], [17], where SFFS have better characteristics over the GA approach. Rule set automatically generated by evolutionary algorithms but probably this procedure was applied to artificial problems like- Majority Problem or for boundary detection in binary images[26], [27]. It depends on a rule set produced exact target output by increasing the size of neighborhood[28]. A form of hill climbing and backtracking algorithms also used to identify the rules. To train the cellular automata for image's thinning and thickening process Paul Rosin has been used RMS criterion and Hausdorff distance error measures with SFFS procedure as objective function but did not produce satisfactory results and produced some lines in fragmented way. Fragmented problem overcame by an algorithm and two cycle cellular automata to produce better results in learning of rules[17].

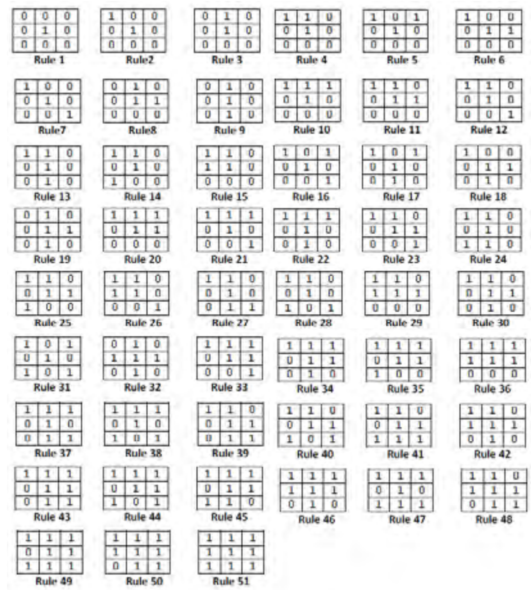

Fig. 3. Rule Set

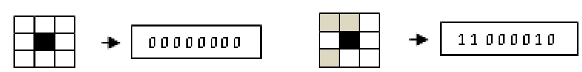

Fig. 4. A sample for converting rule set into string

III. Methodology

In the current experiments, binary images are considered which means cells have two states i.e. white or black and Moore neighborhood is considered and transition rules are only applied to non boundary cells. The initial cell values are considered as the pixel values of the input image. As the Moore neighbourhood is considered there are 2^8 rules are possible which are reduced to 51 by taking into account 45^o rotational symmetry and bilateral reflection [17] and the reduced rule set is shown in figure 3.The neighborhood pattern is encoded in form of string by concatenating the neighbors on 3x3 blocks as shown in figure 4 and the same string is compared with the encoded neighbors of the input image. If both are the same then the central pixel is inverted. Each rule among 51 rules is applied at one time and the PSNR value of resulting filtered image is calculated. Next, all rules are applied on noisy images in combination and find out the PSNR value as the objective function and select the rules at that PSNR which produce maximum value during iteration. In case of salt and pepper noise, rules are applied when it get black pixels at white surface or vice-versa by removing symmetries with rotation $(90^o, 180^o and 270^o)$ of rule set.

In each step each image pixel should be processed in parallel. However, since it is used as sequential operation, repeated until the maximum number of pixels matched to the standard method, the processed pixels are stored in a secondary image C. At the end of every iteration it is copied back to original image in step 7. In Step 6, this cycle is repeated until the maximum number of pixels matched to the standard method bwmorph(). The same procedure with minute change can be used for the thickening operation. The only change is required is to change the value of central black pixel in step 3 if it does not match with the encoded rule set. All other steps are same.

To find the optimum set of the rules the sequential floating forward search(SFFS) method is used. The SFFS is a deterministic search method which provide the effectiveness equivalent to the genetic algorithm(GA)[23].The 0 shows the SFFS algorithm

Let Y_k denote the rule set at iteration and its score be $J(Z_k)$. Here, It is defined by the result of the 0 by applying the CA rule set to the input image and computing the fitness function. Step 1 shows that the initial set is empty. In step 2 at each iteration, all rules are considered for addition to the rule set and only the rule giving the maximum score is added the resulting rule set. This process is repeated until no improvements in score are gained by adding rules.In step 3, each rule in rule set found in step 2 is removed to find the rule whose removal provides the resulting rule set with the improved value of objective function. As shown in step 4 if removal of the rule cause the better score of the objective function then it is discarded from the rule set and again next rule is tried for the deletion and process go to step 3. Otherwise, the process go to step 2 for the addition of new rule to the rule set.

IV. EXPERIMENTAL RESULTS

Following experiments are performed by using MATLAB R2014a.Ten sample images are considered and salt and pepper noise with different intensity levels has been introduced .The images are then filtered with the help of 51 rules as described in previous section. The peak signal to noise ratio (PSNR) value of filtered image is computed for each rule and the rules having best PSNR have been selected. In this section only a part of result is shown. The figure 5 shows the original sample binary images lena, cameraman and rice respectively. Figure 6 shows the images affected by salt and pepper noise with density 0.01. Figure 7 shows the images filtered by the rule set having hte best PSNR value. Figure 8 shows the images filtered by the median filter. Similarly, figure 9 to 14 illustrate the same for the noise density 0.05 and 0.1 respectively.

(a) Lena (b) Cameraman (c) Rice

Fig. 5. Sample input images.

To obtain the best rules set, rules are applied on images iteratively until the difference between the PSNR values in two successive iteration is less than 0.0001 that comes in three or four iterations or sometimes on or two more iterations are required. After performing this experiment it as found that cellular automata provides the better noise filtering in

(a) Lena (b) Cameraman (c) Rice

Fig. 6. Sample images affected by salt and paper noise with pdf 0.01.

(a) Lena (b) Cameraman (c) Rice

Fig. 7. Images filtered by rule set.

(a) Lena (b) Cameraman (c) Rice

Fig. 8. Images filtered by median filter.

(a) Lena (b) Cameraman (c) Rice

Fig. 9. Sample images affected by salt and paper noise with pdf 0.05.

(a) Lena (b) Cameraman (c) Rice

Fig. 10. Images filtered by rule set.

(a) Lena (b) Cameraman (c) Rice

Fig. 11. Images filtered by median filter.

(a) Lena (b) Cameraman (c) Rice

Fig. 12. Sample images affected by salt and paper noise with pdf 0.1.

(a) Lena (b) Cameraman (c) Rice

Fig. 13. Images filtered by rule set.

Algorithm 1 Sequential Floating Forward Search

procedure SFFS
Step 1:
  $Y \leftarrow \{\phi\}$
Step 2:
  Select the best feature
  $x^+ \leftarrow argmax J(Z_k + x) | x \notin Z_k$
  $Z_k \leftarrow Z_k + x^+$
  k=k+1
Step 3:
  Select the worst feature
  $x^- \leftarrow argmax J(Z_k - x) | x \in Z_k$
Step 4:
  if $J(Z_k - x^-) > J(Z_k)$ **then**
    $Z_k \leftarrow Z_k - x^-$
    $k = k + 1$
    Go to Step 3
  else
    Go to Step 2

(a) Lena (b) Cameraman (c) Rice

Fig. 14. Images filtered by median filter.

some cases than median filter and maintain the original image structure in resulting images while the median filter introduces some little bit distortion in the original images structure that are clearly seen in the median filtered images in figure 8, **??** and 14 When the noise density is at low level such as 0.01 the learned rules set is [0 0 0 0 0 0 0 0; 1 0 1 0 1 0 1 0;0 1 0 1 0 1 0 1;] in cameraman image , in rice image rules set is [0 0 0 0 0 0 0 0 ; 1 0 0 0 0 0 0 0 ;1 0 1 0 0 0 0 0;] , in lena image rules set is [0 0 0 0 0 0 0 0 ;0 1 0 1 0 1 0 1;] when the noise density is increased to level 0.1 then cameraman, rice and lena images produce same rule set [0 0 0 0 0 0 0 0;1 0 0 0 0 0 0 0;0 1 0 0 0 0 0 0;]. At every iteration different rules set learned after performing iterations in training procedure of cellular automata, up to convergence level which occurs after three or four iterations or some time takes one or two more. Then it is analyzed that in most of the situations at different density of noise in images it has provided an average range of rules. For testing the training data set is compared to the standard filter (median filter). So that sufficiently delegate to a good rule set (1[0 0 0 0 0 0 0 0], 2[0 0 0 0 0 0 0], 3[0 1 0 0 0 0 0 0]) was learned.

V. Conclusion and Future Work

It is found that the results for the salt and pepper noise filtering through cellular automata rules are encouraging. The resulting rule sets provides better filtering as compared with the median filter. It is also required to mention that the aim to use the cellular automata is to provide simplicity to solve the complex processes which is fulfilled in our experiment and clearly depicted in the results section. Implementation and use of cellular automata rules is easier and straight forward as compare to other methods. As the work presented in the paper is limited in salt and pepper noise with the use of sequential search method, the above algorithm can be enhanced by changing for some gray scale images and other searching methods to improve the results and also the same can be used to train the cellular automata for other type of noises.

References

[1] M. Sipper, “The evolution of parallel cellular machines: Toward evolware,” *BioSystems*, vol. 42, no. 1, pp. 29–43, 1997.

[2] S. Ulam, “Some ideas and prospects in biomathematics,” *Annual review of biophysics and bioengineering*, vol. 1, no. 1, pp. 277–292, 1972.

[3] J. Von Neumann, A. W. Burks *et al.*, “Theory of self-reproducing automata,” 1966.

[4] S. Wolfram, *A new kind of science*. Wolfram media Champaign, 2002, vol. 5.

[5] M. Gardner, “Mathematical games: The fantastic combinations of john conways new solitaire game life,” *Scientific American*, vol. 223, no. 4, pp. 120–123, 1970.

[6] 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),” in *Towards evolvable hardware*. Springer, 1996, pp. 76–98.

[7] S. Aassine and M. C. El Jaı, “Vegetation dynamics modelling: a method for coupling local and space dynamics,” *Ecological modelling*, vol. 154, no. 3, pp. 237–249, 2002.

[8] 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.

[9] C.-l. Chang, Y.-j. Zhang, and Y.-Y. Gdong, “Cellular automata for edge detection of images,” in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, vol. 6. IEEE, 2004, pp. 3830–3834.

[10] 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.

[11] 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.

[12] 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, pp. 1868–1870, 1996.

[13] A. Popovici and D. Popovici, "Cellular automata in image processing," in *Fifteenth International Symposium on Mathematical Theory of Networks and Systems*, vol. 1, 2002.

[14] F. M. Marchese, "A directional diffusion algorithm on cellular automata for robot path-planning," *Future Generation Computer Systems*, vol. 18, no. 7, pp. 983–994, 2002.

[15] S. Nandi, B. Kar, and P. Pal Chaudhuri, "Theory and applications of cellular automata in cryptography," *Computers, IEEE Transactions on*, vol. 43, no. 12, pp. 1346–1357, 1994.

[16] L. S. Davis, "A survey of edge detection techniques," *Computer graphics and image processing*, vol. 4, no. 3, pp. 248–270, 1975.

[17] P. L. Rosin, "Training cellular automata for image processing," *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2076–2087, 2006.

[18] A. P. 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.

[19] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 105–112.

[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. Somol, P. Pudil, and J. Kittler, "Fast branch & bound algorithms for optimal feature selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 7, pp. 900–912, 2004.

[23] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 2, pp. 153–158, 1997.

[24] D. Andre, F. H. Bennett III, and J. R. Koza, "Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem," in *Proceedings of the First Annual Conference on Genetic Programming*. MIT Press, 1996, pp. 3–11.

[25] H. Hao, C.-L. Liu, and H. Sako, "Comparison of genetic algorithm and sequential search methods for classifier subset selection." in *ICDAR*. Citeseer, 2003, pp. 765–769.

[26] A. Adamatzky, "Automatic programming of cellular automata: identification approach," *Kybernetes*, vol. 26, no. 2, pp. 126–135, 1997.

[27] M. Mitchell, P. Hraber, and J. P. Crutchfield, "Revisiting the edge of chaos: Evolving cellular automata to perform computations," *arXiv preprint adap-org/9303003*, 1993.

[28] B. Straatman, R. White, and G. Engelen, "Towards an automatic calibration procedure for constrained cellular automata," *Computers, Environment and Urban Systems*, vol. 28, no. 1, pp. 149–170, 2004.