Cellular Automata for Image Noise Filtering

P. Jebaraj Selvapeter
Senior Engineer – DSP
Jasmin Infotech
Chennai, India
jebarajsp@gmail.com

Wim Hordijk
Dept. of Statistics
University of Oxford
Oxford, UK
wim@santafe.edu

Abstract

This paper presents an image noise filter based on cellular automata (CA), which can remove impulse noise from a noise corrupted image. Uniform cellular automata rules are constructed 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. The performance of this CA approach is compared with the classical median filter and different switching filters in terms of peak signal to noise ratio. This comparison shows that a filter based on cellular automata provides significant improvements over the standard filtering methods.

1. Introduction

Digital image processing plays an important role in daily life applications such as satellite television, magnetic resonance imaging, and computer tomography, as well as in areas of research and technology such as geographical information systems and astronomy. However, data sets collected by image sensors are generally contaminated by noise. Imperfect instruments, problems with the data acquisition process, and interfering natural phenomena can all degrade the data of interest. Furthermore, noise can be introduced by data transmission errors and/or data compression. Thus, noise filtering is often a necessary first step before the image data can be analyzed. However, image noise filtering still remains a difficult challenge because noise removal introduces artifacts and causes blurring of images.

One particular kind of noise, impulse noise, is often caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel. The median filter was once the most popular non-linear filter for removing impulse, or "salt and pepper" noise, because of its good "denoising" power and computational efficiency. However, when the noise level is over 30%, this filter smears some details and edges of the original image. Different remedies of the median filter have been proposed, e.g. the rank conditioned median filter [1], adaptive median filter [2], progressive switching median filter [3], or the median filter based on homogeneity information [4]. These so-called "decision based" or "switching" filters first identify possible noisy pixels and then replace them by using the median filter or its variants, while leaving all other pixels unchanged. These filters are good at detecting noise even at a high noise level. But their main drawback is that the noisy pixels are replaced by some median value in their vicinity without taking into account local features such as the possible presence of edges. Hence details and edges are not recovered satisfactorily, especially when the noise level is high.

In this paper, we present a filter based on cellular automata [5], [6], which is used to remove impulse noise from noise-corrupted images. Our cellular automaton algorithm is applicable to both binary and gray level images, whereas in [5], [6] only binary images were considered. Furthermore, we investigate several different variants of the standard CA rule to improve the performance. Our results using a cellular automaton based filter indeed show significant improvements over the performance of the various standard median filters.

2. Cellular Automata

Cellular automata (CA) were first introduced by John von Neumann (after a suggestion by Stanislaw Ulam) in the late 1940's [7], [8]. But only in the late 1960's, when John Horton Conway developed the Game of Life [9], did cellular automata become more well-known and popular [10].

The essential property of a CA is a regular d-dimensional lattice of cells (d is in most cases only one or two), where each "cell" of this lattice has a discrete state, which is updated at discrete time steps according to a deterministic update rule. This rule determines the state of a cell at the next time step, depending on the state of the cell itself and that of other cells in its local neighborhood. This local neighborhood is often simply the adjacent cells (left and right in 1-D CAs, or left, right, up, and down in 2-D CAs), or some extension of that.

A CA update rule can be expressed as a lookup table that lists for each possible local neighborhood configuration ("nbh") the state which is taken on by the central cell at the next time step ("state"). Figure 1 illustrates a 1-D binary state nearest neighbor cellular automaton. The lattice configuration (10 cells wide) is shown at two successive time steps. For example, the local neighborhood configuration of the third cell at time step $t = 0$ is "001" (the current values of the second, third, and fourth cells), and the lookup table states that this cell will be in state "1" at the next time step $t = 1$. All cells in the lattice are updated in a similar way and simultaneously. Note that in this example periodic boundary conditions are used, i.e., the lattice is viewed as a circle with the leftmost cell being the right neighbor of the rightmost cell and vice versa.

Rule table

nbh:	000	001	010	011	100	101	110	111
state:	0	1	0	1	0	0	1	0

CA lattice

$t = 0:$	1	0	0	1	1	0	1	0	1	1
$t = 1:$	1	0	1	1	1	0	0	0	1	0

Figure 1. Illustration of a simple 1-D CA.

3. CA for Noise Filtering

A digital image is a two dimensional array of $n \times m$ pixels, each with a particular gray value or color. An image can thus also be considered as the lattice configuration of a 2-D cellular automaton where each cell corresponds to an image pixel, and the possible states are the different gray values or colors. In our experiments, binary (black and white) images and 256-level gray scale images are considered. We used a Moore neighborhood (the eight neighboring cells surrounding a cell) on binary images, and a Von Neumann neighborhood (the four neighboring cells up, down, left, and right) on both binary and gray scale images. Fixed value boundary conditions are applied, i.e., the update rule is only applied to non-boundary cells. An impulse noise corrupted image is taken as the initial CA lattice configuration.

To remove this noise from the images, we used a "majority" CA update rule. This rule is stated as follows: if the center pixel (cell) gray level is 0 or 255 (i.e., black or white), then the gray level that is the majority in the local neighborhood replaces the center pixel's value. If none of the gray levels in the local neighborhood is a majority, then there is a tie. This can be dealt with either deterministically or randomly. In the deterministic rule, the center pixel is replaced by the gray level which is in a fixed position in its local neighborhood (e.g., the pixel directly above it). Obviously this choice of the fixed position of the replacement pixel is arbitrary (it could also be the pixel directly below, or to the left, etc.). In general, we do not expect this choice to make a difference in the filtering performance, unless there is a clear "orientation" or directional color gradient present in the image. In the random majority rule, the center pixel's value is replaced by the gray level of a randomly chosen pixel in its local neighborhood. In this case, the replacement pixel is chosen independently (at random) for each occurrence of a tie.

The CA noise filtering method is evaluated and compared with standard filtering techniques in terms of the peak signal to noise ratio (PSNR). We used one binary image ("Cameraman") and two gray scale images ("Fishing boat" and "Lena") as test cases. We considered impulse, or "salt and pepper" noise, which means: 1) only a (random) proportion of the image pixels are corrupted, and 2) a noisy pixel takes either a very large value as a positive impulse (gray scale value 255) or a very small value (0) as a negative impulse. The noise ratio is used to represent how much an image is corrupted. For example, if an image is corrupted by 30% impulse noise, then 15% of the pixels in the image are corrupted by positive impulses and 15% of the pixels by negative impulses (randomly).

Finally, note that the CA filtering algorithm does not require any more computational time or effort than the standard filtering methods, and is therefore at least as efficient. The running time is linear in the number of pixels in the image.

4. Experimental Results

Table 1 shows the peak signal to noise ratio (PSNR) for different filters and noise ratios for the "Cameraman" image. These results show that the performance of a median filter with window size 3×3, switching-I scheme [11], and adaptive median filter are the same. The performance of a switching-II filter [11] is comparable with that of the CA filter with a Von Neumann neighborhood. But at higher noise levels the CA filter outperforms the switching-II filter. The reconstructed images in Figure 2 show that when the window size increases the median filter performance degrades, as it produces a blurred image. It is clear form the results presented in table 1 that the performance of the CA filter with a Moore neighborhood is better than that of all other filters.

Table 2 shows the results for the 256-level gray scale "Fishing boat" image. A deterministic CA rule results in noise propagation, which is solved by using a random rule for breaking majority ties, as explained above. The random CA rule improves the performance by 1 to 3 dB (depending on the noise level) compared with the deterministic CA filter, and both clearly outperform the standard filtering methods. Figure 5 shows three CA filtered "Fishing boat" images,

Noise ratio:	1%	10%	20%
Median 3×3	12.77	12.44	12.05
Median 5×5	11.31	11.15	11.07
Switching-I	12.77	12.44	12.05
Switching-II	13.44	13.10	12.66
Adaptive median	12.77	12.44	12.05
CA (von Neumann)	13.82	13.34	12.83
CA (Moore)	23.75	16.12	14.00

Table 1. PSNR values for different filters and noise ratios for the "Cameraman" image.

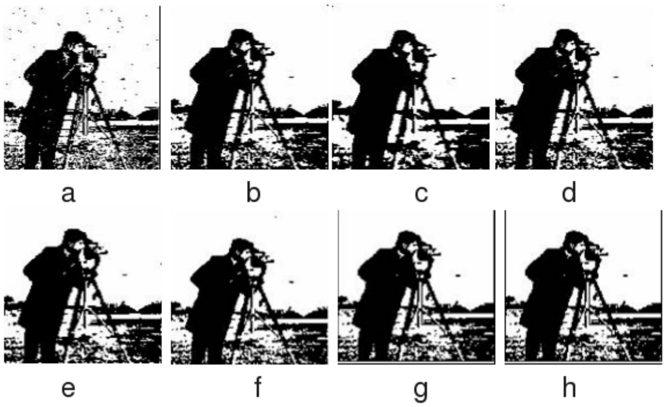

Figure 2. Reconstructed "Cameraman" image with 10% noise ratio. a) Noisy image b) Median 3×3 c) Median 5×5 d) Switching-I e) Switching-II f) Adaptive median g) CA (Von Neumann) h) CA (Moore).

using a deterministic and random CA (figures a and b) and a mirrored CA (figure c; explained below), respectively.

Noise ratio:	1%	10%	20%
Median 3×3	28.49	27.74	25.70
Median 5×5	25.04	24.86	24.48
Switching-I	31.81	30.11	26.79
Switching-II	34.25	30.98	27.08
PSM filter	32.07	29.84	27.86
Adaptive median	32.94	31.91	29.10
Deterministic CA	43.98	33.37	29.66
Random CA	46.46	34.55	30.64

Table 2. PSNR values for different filters and noise ratios for the "Fishing boat" image.

Figure 3 shows the reconstructed gray scale "Lena" image with a noise ratio of 20%. Here too, the median filter results in a blurring effect. Switching-I, switching-II and PSM impulse detectors are not able to detect the impulse noise located in areas where gray levels are comparable with impulse noise levels. The performance of the adaptive median filter is comparable with other switching filters at low noise levels. At higher noise levels, the performance of the adaptive median filter is better than that of other switching filters.

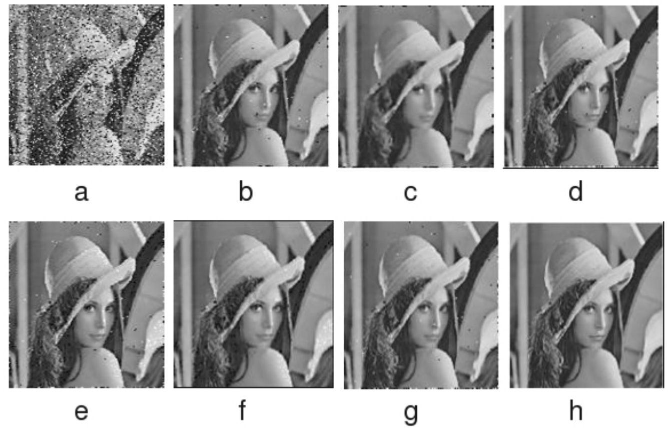

Figure 3. Reconstructed "Lena" image with 20% noise ratio. a) Noisy image b) Median 3×3 c) Median 5×5 d) Switching-I e) Switching-II f) PSM filter g) Adaptive median h) Deterministic CA.

Figure 4 shows the PSNR in dB for the different filters and noise ratios for the "Lena" image. At low noise levels the performance of the progressive switching median filter is comparable with that of the CA filter, but at higher noise levels the CA filter is better than the PSM filter.

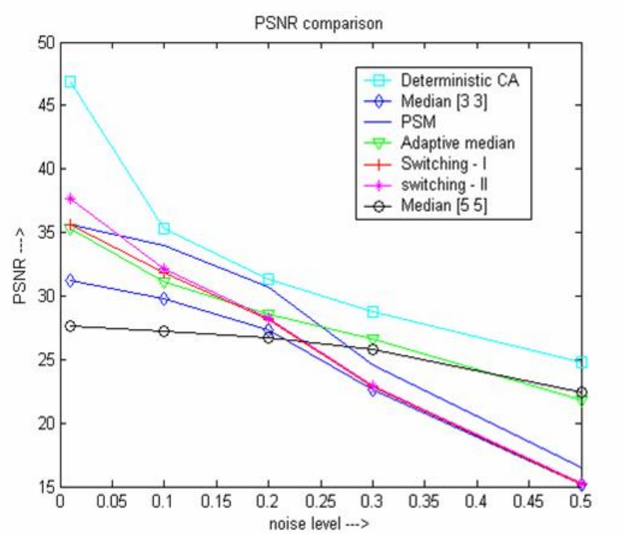

Figure 4. PSNR values for different filters and noise ratios for the "Lena" image.

In the fixed boundary CA as used so far, the update rule is not applied to the cells at the boundary of the lattice, so noisy pixels at the boundary of an image can never be restored. In fact, it can introduce a border in the image, as shown in figures 5a and b (full image) and figures 6a and b (image detail). However, a mirrored CA solves this problem.

In a mirrored CA, an existing neighbor of a boundary cell is “mirrored” into an otherwise nonexistent neighboring cell, so the update rule can be applied as usual. So, for example, to create a “left neighbor” for a cell in the leftmost column of an image, its “right neighbor” is mirrored. Table 3 shows the image restoration results for a fixed boundary CA (deterministic and random) and a mirrored CA. As the table clearly shows, a mirrored CA improves the performance by 8 to 23 dB (depending on the noise level) compared with the fixed boundary CA filters. Figure 6 shows the boundary effects of fixed boundary and mirrored CA rules in a detail of the top-left corner of the image.

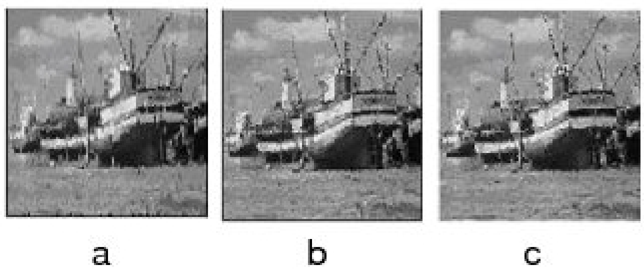

Figure 5. Reconstructed “Fishing boat” image with 50% noise ratio. a) Deterministic CA, b) Random CA, and c) Mirrored CA.

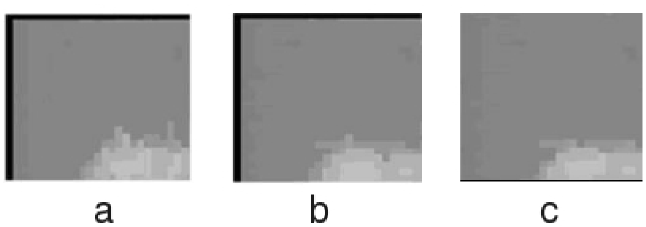

Figure 6. Boundary effect of a) Deterministic CA, b) Random CA, and c) Mirrored CA (detail of image corner) with a 50% noise ratio.

Noise ratio:	1%	10%	20%
Deterministic CA	22.70	22.36	21.84
Random CA	22.71	22.41	22.05
Mirrored CA	45.03	34.16	30.71

Table 3. PSNR values for fixed boundary and mirrored CA filters for different noise ratios.

In summary, the presented results clearly show that it is possible to construct simple cellular automata filtering rules that outperform standard noise filtering methods in several important ways. Furthermore, it is possible to make variations to the CA rules to deal with additional issues, such as blurring and boundary effects.

5. Conclusions

Our initial experiments with CAs for noise filtering are encouraging. They show that it is possible to construct good rule sets to perform common image processing tasks. A 2-D cellular automaton with a very simple update rule can be used as an efficient impulse noise filter in digital images. In particular, for filtering salt and pepper noise in a binary image, the CA based on a Moore neighborhood performed better than the standard median filter, adaptive median filter, and switching filters. For gray level images, compared with a deterministic CA, a random CA performs better for a detailed image. A mirrored CA solves the fixed boundary problem. Our results are an extension of and improvement over previous methods [5], [6] in that we also considered gray scale images and not only binary images, and tried several variants of the standard CA rule for improved performance.

To improve performance even further, there are several areas to investigate. For example, most CAs use identical rules for all the cells in the lattice. An extension is to use non-uniform CAs, that is, different rules can be applied at different locations and also at different time steps depending upon local conditions of the image. We can also use alternative neighborhood definitions, or, to get possibly even better CAs, we could use an evolutionary algorithm to search for good filtering rules. Finally, a very important feature of the proposed method is its intrinsic parallelism, since it is implemented as a cellular automaton where the individual cells update in a synchronous manner. This provides the potential (when implemented appropriately) to make the proposed impulse noise filter method faster than other typical filter algorithms.

References

[1] L. Alparone, S. Baronti, and R. Carla, “Two-dimensional rank-conditioned median filter,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 2, pp. 130–132, 1995.

[2] H. Hwang and R. A. Haddad, “Adaptive median filters: new algorithms and results,” *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 499–502, 1995.

[3] Z. Wang and D. Zhang, “Progressive switching median filter for the removal of impulse noise from highly corrupted images,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 1, pp. 78–80, 1999.

[4] G. Pok, J. C. Liu, and A. S. Nair, “Selective removal of impulse noise based on homogeneity level information,” *IEEE Transactions on Image Processing*, vol. 12, no. 1, pp. 85–92, 2003.

[5] A. Popovici and D. Popovici, “Cellular automata in image processing,” in *Proceedings of the 15th International Symposium on the Mathematical Theory of Networks and Systems*, D. S. Gilliam and J. Rosenthal, Eds., 2002, electronic proceedings.

[6] P. L. Rosin, “Training cellular automata for image processing,” in *Proceedings of the 14th Scandinavian Conference on Image Analysis*, H. Kalviainen, J. Parkkinen, and A. Kaarna, Eds., 2005, pp. 195–204.

[7] J. von Neumann, *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966, (edited and completed by A. W.Burks).

[8] A. W. Burks, Ed., *Essays on Cellular Automata*. University of Illinois Press, 1970.

[9] M. Gardner, “The fantastic combinations of John Conway’s new solitaire game ”Life”,” *Scientific American*, vol. 223, no. 120, p. 123, 1970.

[10] N. Ganguly, B. K. Sikdar, A. Deutsch, G. Canright, and P. P. Chaudhuri, “A survey on cellular automata,” 2003, technical Report, Centre for High Performance Computing, Dresden University of Technology.

[11] T. Sun and Y. Neuvo, “Detail-preserving median based filters in image processing,” *Pattern Recognition Letters*, vol. 15, no. 4, pp. 341–347, 1994.