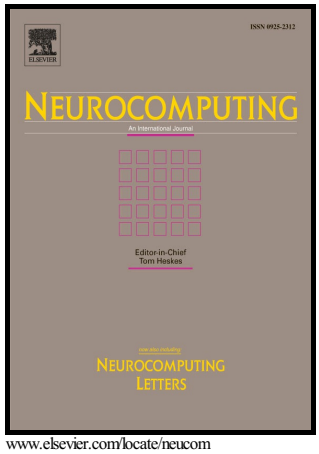
An Edge Detection Method using Outer Totalistic Cellular Automata

Sebastian Amrogowicz, Yitian Zhao, Yifan Zhao



Neurocomputing, Volume 214, 19 November 2016, pp643-653

www.elsevier.com/locate/neucom

PII: S0925-2312(16)30684-1
DOI: http://dx.doi.org/10.1016/j.neucom.2016.05.092
Reference: NEUCOM17296

To appear in: *Neurocomputing*

Received date: 15 October 2015
Revised date: 12 May 2016
Accepted date: 20 May 2016

Cite this article as: Sebastian Amrogowicz, Yitian Zhao and Yifan Zhao, An Edge Detection Method using Outer Totalistic Cellular Automata *Neurocomputing,* http://dx.doi.org/10.1016/j.neucom.2016.05.092

Neurocomputing, Volume 214, 19 November 2016, pp. 643-653
DOI:10.1016/j.neucom.2016.05.092

An Edge Detection Method using Outer Totalistic Cellular Automata

Sebastian Amrogowicz[a], Yitian Zhao[b], Yifan Zhao[c,*]

[a]*School of Aerospace, Transport and Manufacturing, Cranfield University, UK, MK43 0AL*
[b]*Beijing Engineering Research Center of Mixed Reality and Advanced Display, School of Optics and Electronics, Beijing Institute of Technology, China*
[c]*Through-life Engineering Services Institute, School of Aerospace, Transport and Manufacturing, Cranfield University, UK, MK43 0AL*

Abstract

A number of Cellular Automata (CA)-based edge detectors have been developed recently due to the simplicity of the model and the potential for simultaneous removal of different types of noise in the process of detection. This paper introduced a novel edge detector using Outer Totalistic Cellular Automata. Its performance has been compared with other recently developed CA-based edge detectors, in addition to some classic methods, through testing images from a public library. Visual and quantitative measurement of similarity with manually marked correct edges confirmed the superiority of the proposed method over conventional and state-of-the-art CA-based edge detectors.

Keywords: Cellular Automata, Edge detection, Figure of merit, Totalistic Cellular Automata

1. Introduction

Edge detection is one of the fundamental image processing tasks that has been widely investigated since technology allowed people to digitally process visual data. Information about edges is the basis of many computer vision systems such as object recognition, pattern classification, robotic vision and medical diagnosis. The quality of detected edges has a direct and high influence on the performance of mentioned systems.

As edges correspond to abrupt changes in intensity values, their presence can be detected using derivatives. The most commonly used directional masks based on the first order derivative were proposed by Roberts [1], Sobel [2] and Prewitt [3]. Marr and Hildreth [4] argued that edges are not invariant of scale, and good edge detector should be able to work at different

*Corresponding author: Tel.: +44(0)123475011 Ext 2283; fax: +44(0)1234758292;
Email address: yifan.zhao@cranfield.ac.uk (Yifan Zhao)

scales. They proposed to use the Gaussian function for smoothing and Laplacian operator for calculating derivative, resulting in Laplacian of Gaussian operator. Canny [5] is considered as a state-of-the-art edge detector. It produces thin and continuous edges with good localization. However, it requires the user to determine the size of the blurring window, high and low values of the threshold. Given the importance of edge detection, many different methods were introduced to overcome different problems that occur in classical methods. The SUSAN (Smallest Universal Segment Assimilating Nucleus) is an algorithm that allows for noise filtering, edge and corner detection [6]. It uses a circular mask and compares the difference in brightness to the central pixel in order to determine edge strength information. In order to perform non-maxima suppression edge direction is calculated based on the centre of mass. Some statistical methods [7, 8] were introduced recently to overcome the spurious edges caused by the presence of textures. Chen [9, 10] proposed a method to use feature dictionary to guide identification of the edges with varying shapes and sizes.

This study focuses on edge detection based on Cellular Automata. Cellular Automata (CA), introduced by John von Neumann [11] in the 1950s, is a spatially and temporally discrete dynamical system composed of cells arranged in a lattice. Each cell can be in one of a finite number of states. The transition between states is dependent on the cell value, the state of its neighbourhood and the transition rule. The most characteristic feature of CA is that using simple rules interacting with a local neighbourhood it can produce very complex behaviour. Many researchers have investigated the possibility of using Cellular Automata for image processing [12, 13], and a few focused on edge detection with either binary, greyscale or colour images as inputs. The linear set of rules applied to binary images were recently investigated by several authors. Quadir and Khan [14] divided all 512 rules for Moore neighbourhood into three groups depending on their ability to detect edges. However, they did not cover the different behaviour of rules or compared them. Uguz et. al. [15] focused on the benefits of implementation of the transition function in the form of matrix multiplication. They have presented the results for four rules but focused more on the speed benefits. Nayak et. al. [16] presented the results of using a larger, extended Moore neighbourhood. They showed the results for 6 out of 33554432 possible rules. Diwakar et. al. [17] presented an application of Totalistic rules with Moore neighbourhood for edge detection. Djemame and Batouche [18] used Particle Swarm Optimization heuristic to determine the best rules without enumerating the complete search space. Wongthanavasu and Sadananda [19] proposed a Weighted Cellular Automata method (WCA) based on von Neumann neighbourhood that can deal with both binary and greyscale images and can be

implemented efficiently and it does not require selection of rules or any user input. Djemame et. al. [20] presented a method using a Continuous Cellular Automata for edge detection. Chang et. al. [21] proposed a method, where an Orientation Information Measure is used to process a greyscale image into binary, and then a Cellular Automata with semi-neighbourhood is used to detect edges. Chen and Yan [22] presented a method that combines the diffusion model with CA. Many other variations of CA-based edge detector can be found in [23, 24, 25, 26]. Recently a lot of attentions were attracted to the work of Rosin [27]. He proposed to use a Sequential Floating Forward Search (SFFS), which is a deterministic algorithm, to search for the best set of rules that would allow performing different tasks like denoising, thinning and finding the convex hull. Later he proposed an extension of his method [28] to tackle edge detection. This method can generate edge intensity images with simultaneous removal of impulse noise. However, this method is relatively time-consuming since processing has to be done for a set of 255 images. In opposite to deterministic SFFS, a heuristic can be applied, with most interest presented in Genetic Algorithms. An example of searching for an optimal packet of rules is presented in work of Batouche et. al. [29] and Slatnia et. al. [30]. Similarly to Rosin's method, they searched for a set of rules that would change the central pixel state, but they did not restrict them to central white pixel. Their publications claim that great results can be obtained using only a single rule. Apart from simple cellular automata, Fuzzy Cellular Automata-based edge detector has also been studied because it incorporates fuzzy logic into transition rules, which results in a good performance when used for greyscale images [31, 32]. Patel and More incorporated fuzzy logic and cellular learning automata [33]. Sinaie et. al. presented a method for enhancement of edges acquired by fuzzy edge detector [34]. Some others CA-based methods have been developed to focus on grey and color image [35, 36, 37].

As a generalisation of the Totalistic Cellular Automaton (TCA) [38], Outer Totalistic Cellular Automaton (OTCA) has been attracting more and more investigations due to its higher complexity [39, 40]. The famous example is Conway's Game of Life. However, the literature review shows that no one has applied OTCA into edge detection. To fill this research gap, this paper introduces a novel edge detection method based on Outer Totalistic Cellular Automata. The results of testing some public data set are compared with conventional and state-of-the-art CA-based methods in terms of different types and levels of noise.

2. Method

2.1. Model

For the purpose of image processing, the proposed method uses a rectangular, two-dimensional grid L, where each cell corresponds to one pixel in the image. Every cell can be in one of finite number of discrete states $S = \{0, 1, \dots k-1\}$. Initial values of grid correspond to values from image $s_0 \in S$. Every cell updates its state simultaneously in discrete time steps depending on its local neighbourhood N according to a transition rule $f : S^2 \to S$. A Cellular Automata can be defined as quintuplet:

$$C = \{L, N, S, f, s_0\}$$

To fully specify CA it is essential to define the number of possible states, type and size of neighbourhood and transition function. For grey-scale images, k typically is 256 and for binary images, k is 2. The neighbourhood of the cell (i, j) can be described by

$$N_{(i,j)} = \{(x, y) \in L \mid x - i \mid + \mid y - j \mid \leq r\} \tag{1}$$

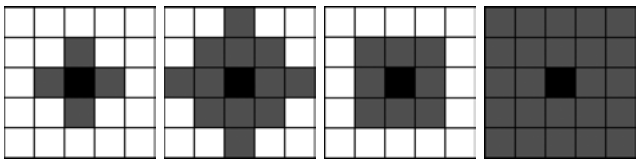
or

$$N_{(i,j)} = \{(x, y) \in L \mid x - i \mid \leq r \wedge \mid y - j \mid \leq r\} \tag{2}$$

where r defines the size of the neighbourhood; (x, y) are the coordinate of the neighbour. The most popular neighbourhood types are von Neumann (Eq. 1) and Moore (Eq. 2), illustrated by Figure 1. The majority of existing methods are based on small neighbourhoods ($r = 1$) as search space grows rapidly with the increase of r. The transition function can be defined either as a mapping from each neighbourhood state to the next cell state or in the form of a function. The total number of possible rules is determined by k and the number of neighbours. If Moore neighbourhood with $r = 1$ is considered for a grey-scale image for example, there are 256^9 possible patterns. This number can be reduced by removing rotations and symmetries, obtaining 2×10^{18} patterns. It is still impossible to enumerate all possible patterns in order to obtain the optimal solution. One common way to reduce search space is to use binary images, for which there is a total number of 2^9 patterns or only 51 invariant patterns, as shown in Figure 2.

There are a number of classes of CA, such as Uniform CA (UCA), Linear CA (LCA), Totalistic CA (TCA) and Fuzzy CA etc. This paper focuses on TCA only.

With Totalistic CA the next state of a cell is determined by the sum of its neighbourhood including the central cell. Rule number is created by defining the next cell state depending on



(a) (b) (c) (d)

Figure 1: Four typical neighbourhoods, where the black denotes the central cell and the grey cells denote its neighbours. (a) von Neumann (r=1); (b) von Neumann (r=2); (c) Moore (r=1); (d) Moore (r=2)

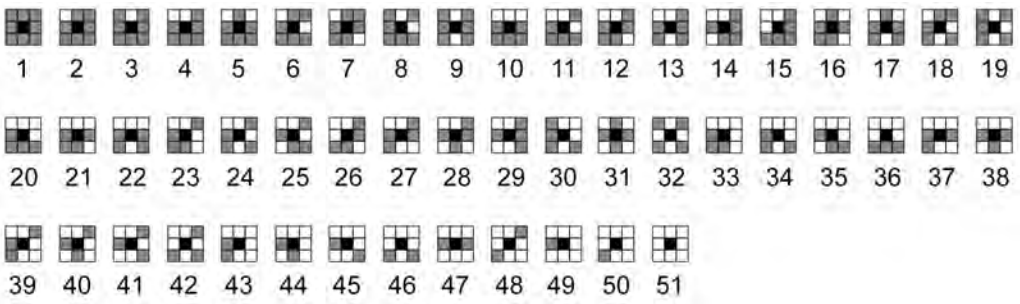


Figure 2: Invariant set of binary rules, where the black cell denotes the central cell, the grey cells denote the neighbours with value 1 and the white cells denotes the neighbours with value 0.

the sum of cells in the neighbourhood in a form of a binary string. For a Moore neighbourhood with the central pixel included, the sum can take a value between 0 and 9, which gives 1024 possible rules. Figure 3 presents the binary string for the Totalistic rule 56, for which the next state of the central cell will become 1 when the sum of neighbourhood takes values between 3 and 5.

In Outer Totalistic CA (OTCA) the next cell state depends on the sum of cells in the neighbourhood only (without counting the central cell), and the value of the central cell. In contrast to conventional TCA, the central cell state has a strong influence on the next state. Since for every sum of neighbourhood two transitions depending on the central pixel have to be defined, the search space becomes much larger. For a Moore neighbourhood, the sum can take a value

Neighbourhood sum	9	8	7	6	5	4	3	2	1	0
Next state	0	0	0	0	1	1	1	0	0	0

Figure 3: Totalistic rule 56

Neighbourhood sum	8		7		6		5		4		3		2		1		0	
Central pixel value	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Next state	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0

Figure 4: Outer Totalistic rule 832

between 0 and 8, which gives 18 distinct patterns, resulting in $2^{18} = 262144$ possible rules. As an example, OTCA rule 832 can be illustrated by Figure 4. Next state of the cell becomes '1' in the case where: the sum is 3 and central pixel is '0' and the sum is 4 regardless of the central pixel state.

2.2. Quality Metrics

Taking into account the influence of edge quality on the performance of all algorithms that require edge information, a well-defined metric for quantitative evaluation needs to be chosen. Despite the clear advantages of a unified quantitative approach, no common solution was agreed and many different metrics have been proposed.

For any metric to characterise the performance, a set of reference edges are required for the comparison with the results from the detectors. Those reference edge maps are called Ground Truth (GT) and need to be manually generated for every testing image. In order to decrease the subjective preferences, a set of GT images can be defined by different people and combined to provide the best edge map. From the many metrics proposed to quantify the similarity of the resulting edges with the GT, the most widely used one is Pratt's Figure of Merit (FoM) [41], which can be written as

$$FoM = \frac{1}{max\{I_I, I_A\}} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha {d_i}^2} \tag{3}$$

where:

- I_A: the number of detected edge points
- I_I: the number of ideal edge points (Ground Truth)
- d_i: the pixel miss distance of the ith edge detected
- α: calibration constant

It takes into account both the distance from the detected edge to the ideal edge and the number of incorrectly marked points. It is also normalised to take values in (0,1], where 1 represents

exact accuracy, and the decrease in value corresponds to higher dissimilarity. The calibration constant α allows us to control the sensitivity of the distance between edge pixels and target pixels. In this paper, α was chosen to be $\frac{1}{9}$ as suggested in [41].

The value of FoM is influenced by the discrepancy in distance and number of marked pixels, providing correct results even in the case where there is a shift or deformation between compared images. The main drawback of this method is the inability to define the reason for error, which is not in the scope of this study.

2.3. Rule Selection

The additional influence of the central pixel on the next state motivates this study to investigate if a rule or rules based on OTCA works better than previously proposed CA-based methods. The increase in performance can be evidenced by either a stronger resistance to noise, more continuous or thinner lines, or decrease in spurious edges.

With the relatively large search space, a manual selection of rule would be difficult and can result in omission of possible important rules. A complete search of rule space is, therefore, necessary in order to obtain an optimal solution. Consider a binary image with the Moore Neighbourhood. Although the total number of rules for OTCA can be completely iterated with the calculation of the quality metric in affordable time, it can be easily noticed that some rules can be removed beforehand. First of all, the first and last bit in rule string represent the situation in which neighbourhood is uniform, which corresponds to no edge. Secondly, the second bit and one before last bit represent the situation where a noise pixel is present in the centre. By fixing value for the four presented situations the number of possible rules is reduced from 262144 to 16384, which saves 93.75% of searching time. Further decrease of two bits can be similarly justified as patterns corresponding to noisy areas, however, the decision was made to keep them in search space for this study.

With the total of 16384 rules remaining it is impossible to find the best rule by visual inspection, especially through evaluating the performance on a set of images. The values of FoM were calculated for all 60 testing images. From the obtained results, for every image, a set of best rules were extracted by selecting the top 5%. A histogram was built for all rules obtained in this way and 100 most common rules were selected for visual inspection.

During the visual inspection, rules providing best results in terms of edge continuity, thinness and correct classification were selected. Further refining was performed by investigating the influence of different levels of noise. In this paper, four rules were selected as the representatives:

832, 960, 1856 and 1984. Many different rules may have better performance for either a specific type of image or a specific type or level of noise, however, this paper aims to present the ones that provide the best results over a wide range of variations.

For all but Wongthanavasu's method, either a single rule or a packet of rules are selected. It is expected that different rules present different behaviours depending on image content and presented noise. Since the number of rules presenting edge detection capabilities is very large, it is necessary to narrow down to only a few rules for every method. The most straight forward way is to consider the rules proposed in publications. For the Linear Cellular Automata, a wide range of rules shows very similar edge detection properties. In order to maintain only a few representatives, rules 68, 75, 113 and 120 with Moore neighbourhood proposed by Uguz et. al. [15], and 1025, 1040, 131073 with extended Moore neighbourhood proposed by Nayak et. al. [16] were selected. The decision to discard two rules with the extended neighbourhood is due to their high similarity with other rules. For the Totalistic Cellular Automata rules 56, 112 and 120, proposed in [18], were also selected. The totalistic rule proposed by Diwakar et. al. [17] was not considered, as it was found to result in thick edges with poor noise resistance. The method described to select the best rules for OTCA was also used to search those for TCA rules, although the three already mentioned proved to be optimal. In the case of Uniform Cellular Automata, two packets proposed by Rosin were considered [27].

The transition of cells values was performed simultaneously and is independent of each other. Although the implementation of the proposed method is created in an iterative manner, it presents the possibility of parallelisation. Since the neighbourhood encoding requires only a local value, Cellular Automata presents the type of operation that is refereed as "trivially parallel". An efficient implementation using any type of multi-CPU platform should not present any difficulties. It can benefit from both multi-core systems to General Programming Graphical Processing Units (GPGPU) containing hundreds of cores. With its intrinsic parallelism, Cellular Automata can be a great choice in any concurrent processing pipeline.

2.4. Evaluation Dataset

As a dataset for testing, a collection of images with manually specified Ground Truth edges available from the University of South Florida (USF) [42] was used. This database contains a set of 50 real-life and 10 aerial images. The collection contains pictures of a single object located in the centre, both in indoor and outdoor scenes. The complexity of edges within those images represents a wide variety of edge type, allowing us to draw valuable conclusions from the obtained

results.

Authors of this dataset had specified three different types of pixels: edge - represented by black, no-edge - represented by grey and "don't care" represented by white. For the purpose of calculating the FoM only information about edge location is important. The complete collection of images with their corresponding GT is publicly available on-line on authors' website.

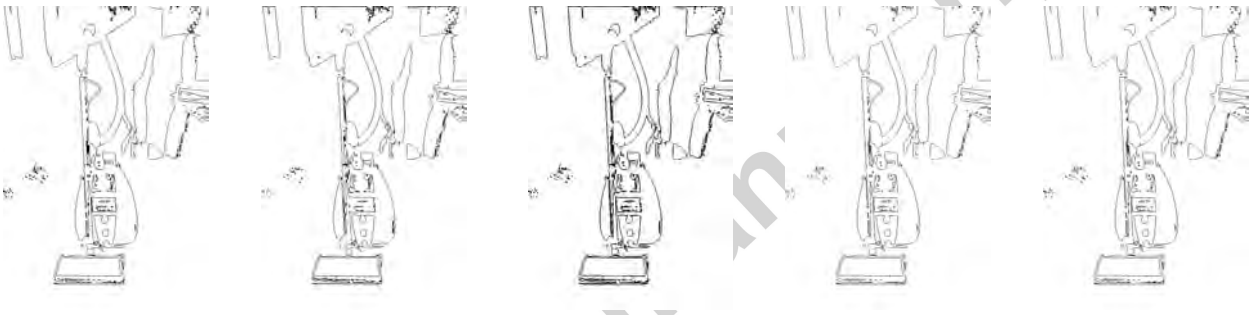
3. Results

3.1. Noise Free Performance

The image 110 from USF representing a vacuum cleaner with a couch in a background, shown in Figure 5.(a), was selected as an example image. The process of binarisation using Otsu's method [43] resulted in well-extracted objects with some noise from the textures on the couch and carpet, shown in Figure 5.(b). Figure 5.(c) shows the GT of edges for the considered image. The results for all chosen Cellular Automata methods, illustrated by Figure 5.(d)-(t), clearly show their abilities to perform edge detection task for an image representing a real scene.

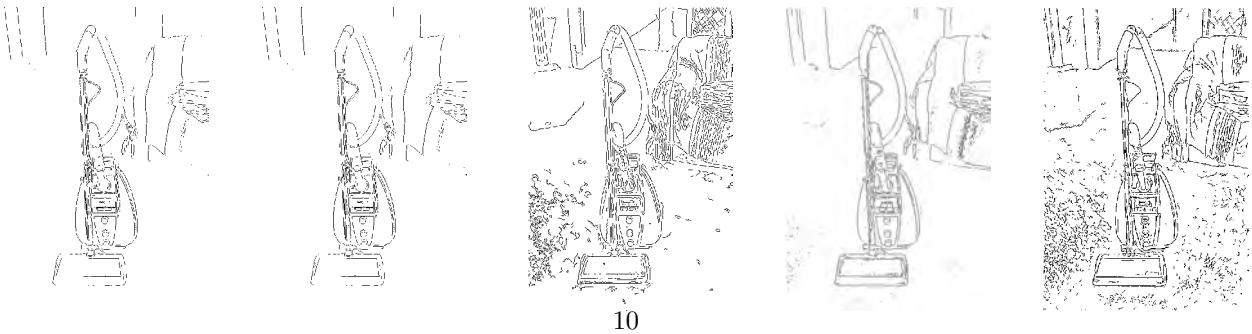
(a) Image 110 (b) Binary (c) Ground Truth (d) LCA 68 (e) LCA 75

(f) LCA 113 (g) LCA 120 (h) LCA 1025 (i) LCA 1040 (j) LCA 131073



(k) TCA 56 (l) TCA 112 (m) TCA 120 (n) OTCA 832 (o) OTCA 960

(p) OTCA 1856 (q) OTCA 1984 (r) WCA (s) UCA One (t) UCA Packet



(u) Sobel (v) Prewitt (w) Canny (x) SUSAN (y) Kovesi

Figure 5: Comparison of selected Cellular Automata methods with Sobel, Prewitt, Canny, SUSAN and Kovesi's algorithms.

The CAs are able to produce both very thin and thick lines, depending on the selected rules. In most cases the creation of thin edges introduces short discontinuities and less smoothness in resulting lines. From the selected linear rules thinnest edges are created by rule 113 and 120 which present the highest visual quality. Although all remaining rules present different levels of thickness, they provide a much greater level of continuity and smoothness. For all linear methods that present the edge detection abilities, the effect of propagation of all defects related to texture binarisation in the form of noised areas and isolated pixels has been noticed.

The result from WCA creates smooth, one pixel wide edges with good locality. No effect of smearing or connecting close edges was noticed. The influence of noise will be discussed in more details in the next section.

A uniform CA with just a single rule produced edges with quality comparable to all other methods, shown in Figure 5.(s). The resulting lines have one pixel wide. As a drawback, there are a lot of single spurious pixels connected to resulting lines. When the packet of rules is applied, the result of which shown in Figure 5.(t), where less spurious points are marked, the quality of resulting edges is better. However, occasional discontinuities can be noticed, but they are not presented when the only single rule is used.

All Totalistic methods produced edges with better quality than already mentioned methods. For all investigated rules, edges are thin, continuous and relatively accurate. It has been noticed that edges from noised areas are more straight and less spurious pixels have been found. Rule 56 and 112 create one pixel wide edges, however, they differ in localisation. Rule 120 tends to create little thicker lines. Both salt and peppers pixels are still present but in much-attenuated fashion. A shortcoming of this method is a tendency to connect lines that are very close to each other resulting in larger blocks of pixels, especially in the case of rule 120.

For all of proposed OTCA rules, resulting edges are thin and continuous. They present the same smoothing capabilities as TCA, resulting in clearer edges on noised borders. Occasional discontinuities can be noticed for all but rule 1984. All rules present resistance to bonding close edge line that could be noticed in TCA. Both salt and pepper noise pixels were attenuated, providing better filtering probabilities than other CA-based methods.

The results for both Prewitt and Sobel are very similar. To observe the differences an examination under higher magnification has been performed. Edges are continuous and more details were found than those produced by the CA-based methods. Additionally, there are no noisy areas. On the other side, many spurious edges were detected, especially in the tube. There are also discontinuities in edges, easily seen on the couch and wall. Canny with an automatic

threshold selection resulted in detecting edges not only for vacuum and couch but also with the textures on the coach and wall. A large number of excessive edges, combined with a lot of short spurious edges on the rug do not present clear advantage. Results from Kovesi's methods are very similar as for Canny, resulting in still increased amount of spurious edges. The result from SUSAN displays a clear improvement over other classical methods, but still generates a lot of small false edges which can be easily noted on the tube or couch.

3.2. Performance Under Salt & Pepper Noise

Salt & Pepper (S&P) noise representing a random white or black pixel, is commonly presented due to a fault in camera sensors. Despite the fact that it becomes less common than a few years ago, it can still be presented in images. Since S&P noise is a single abrupt change in intensity it can result in high gradient value when processed by classical edge detection methods resulting in a large decrease in quality. A salt and pepper type of noise can also be introduced during binarisation process from texture areas. The possibility of simultaneously removing S&P noise during detection of edges would present a great advantage.

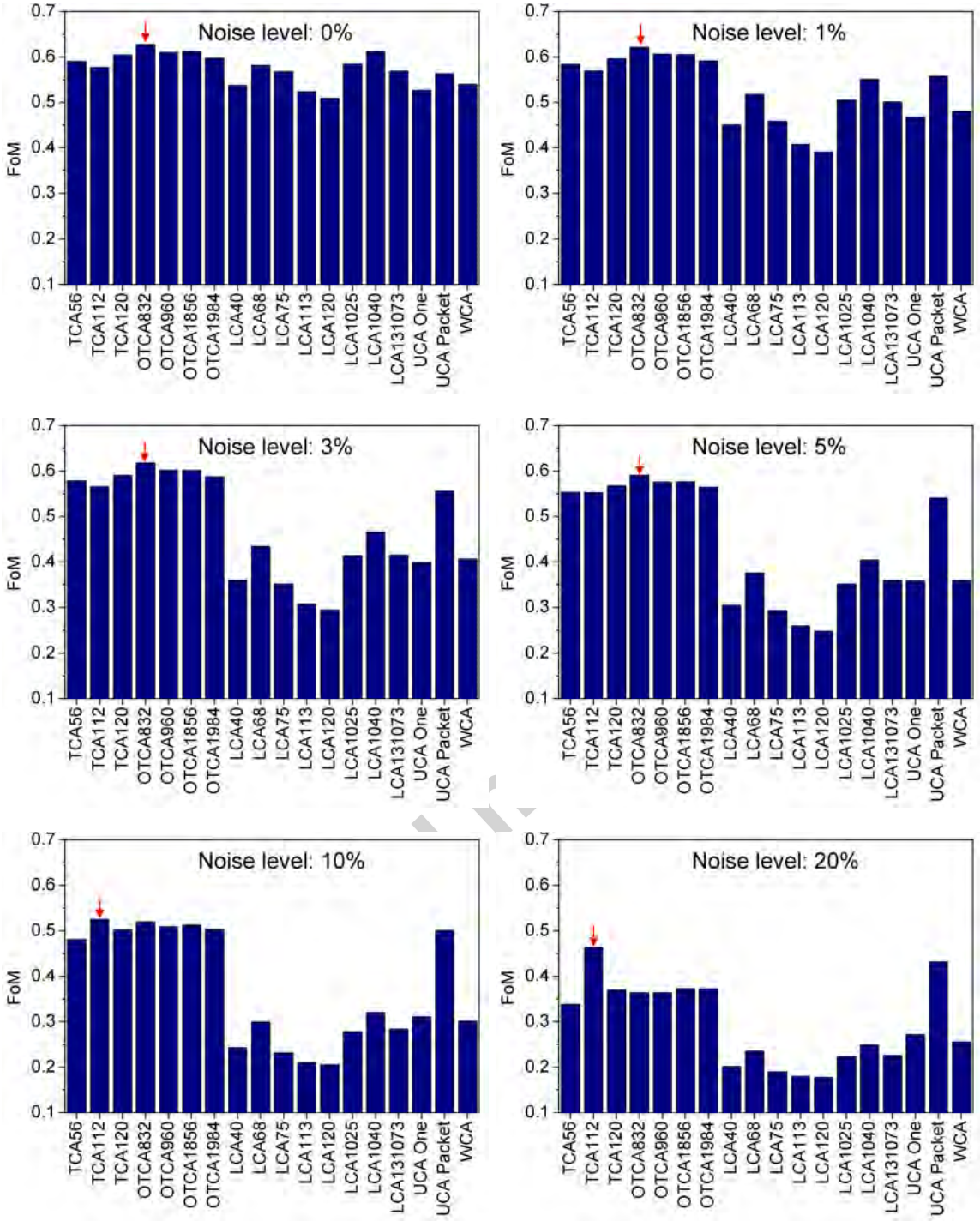
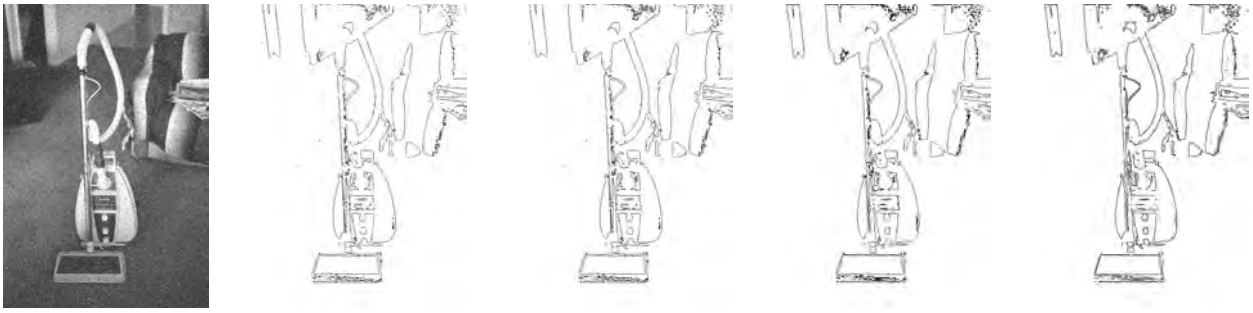


Figure 6: Figure of Merit for different levels of variation of Salt & Pepper noise, where the red arrow highlights the maximum value. (Image 110 - vacuum)

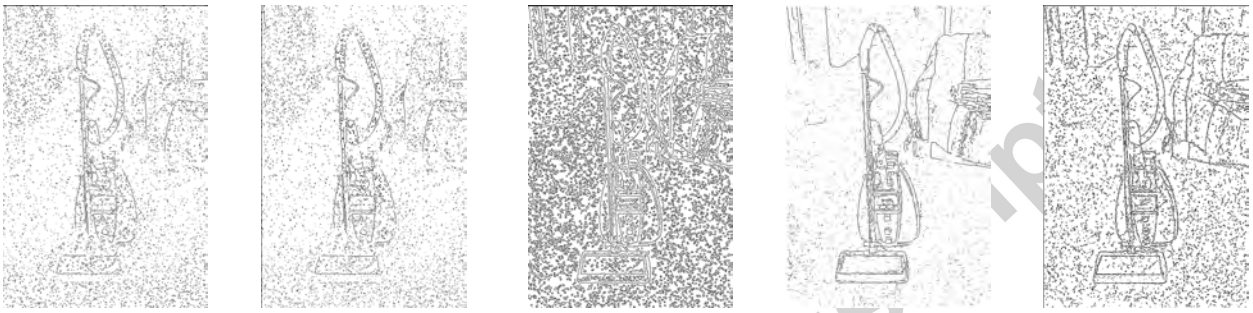
Figure 6 presents the Figure of Merit values under the influence of different levels of S&P noise. The noise was added before the binarisation in order to affect also the process of selecting the threshold value. Results show a rapid decrease in quality value for all linear rules, as well as for WCA and UCA with a single rule. All linear rules tend to carry the salt noise into resulting images, often in an enhanced manner. With the use of extended Moore neighbourhood the influence of noise decreases, but it is still presented. Both WCA and UCA with a single rule present different behaviours for S&P noise. When processing image with salt noise, they carry the noise in a slightly attenuated fashion. Impulse noise, however, is strongly enhanced by both methods. The WCA creates a white border within a von Neumann neighbourhood and UCA within a Moore neighbourhood of a pepper pixel.

It has been observed from Figure 6 that, among the considered CA methods, the proposed method produced the highest quality value up to 5% impulse noise. With the increase of noise level, TCA begins to take advantage over OTCA. It is shown that TCA 112 produced the best result for the noise level 10% and 20%. The optimal packet proposed by Rosin, whose result shown in Figure 7.(e), despite not leading under small noise levels, demonstrates relatively good resistance to higher noise levels. Results obtained from the complete dataset are highly consistent with the presented example. As with the increase of noise level, for the TCA method, more isolated pixels were removed meanwhile remaining connected and accurate edges. With more noise presented, the OTCA method tends to pass more and more isolated pixels and begins to introduce discontinuities in edges, although it presents the ability to attenuate the areas with cumulated noise. The UCA with the packet of rules shows a much better ability to resist noise. As the noise level increases, more discontinuities were introduced and spuriously connected pixels were introduced, as shown in Figure 7.(o).

With high-level noise, most of the classical methods failed to produce quality images, shown in Figure 7.(p)-(t). Up until 10% noise level, SUSAN edge detector managed to extract edges although a lot of noise pixels are presented in resulting image. Both Sobel and Prewitt resulted in discontinuous edges and filled the boundaries of the object with noise. The results for Canny edge detector present the image covered with short spurious edge lines. An attempt to manually select threshold values showed that when most of the false edges are removed the resulting image does not contain a lot of correct edges as well.

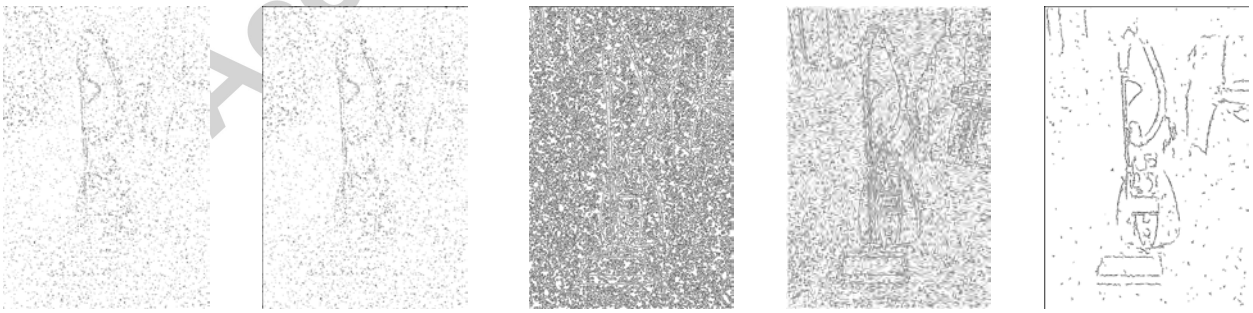


(a) 5% Noise (b) OTCA 832 (c) OTCA 1856 (d) TCA 112 (e) UCA Packet



(f) Sobel (g) Prewitt (h) Canny (i) SUSAN (j) Kovesi

(k) 20% Noise (l) OTCA 832 (m) OTCA 1856 (n) TCA 112 (o) UCA Packet



(p) Sobel (q) Prewitt (r) Canny (s) SUSAN (t) Kovesi

Figure 7: Influence of high levels of Salt & Pepper noise on TCA, OTCA and UCA compared to Sobel, Prewitt, Canny, SUSAN and Kovesi's methods.

3.3. Performance Under Gaussian Noise

Gaussian noise represents an added noise to the pixels, where the noise value is normally distributed. With the increase in variance, the resulting binary image presents more smeared edges and introduces clustered noise areas.

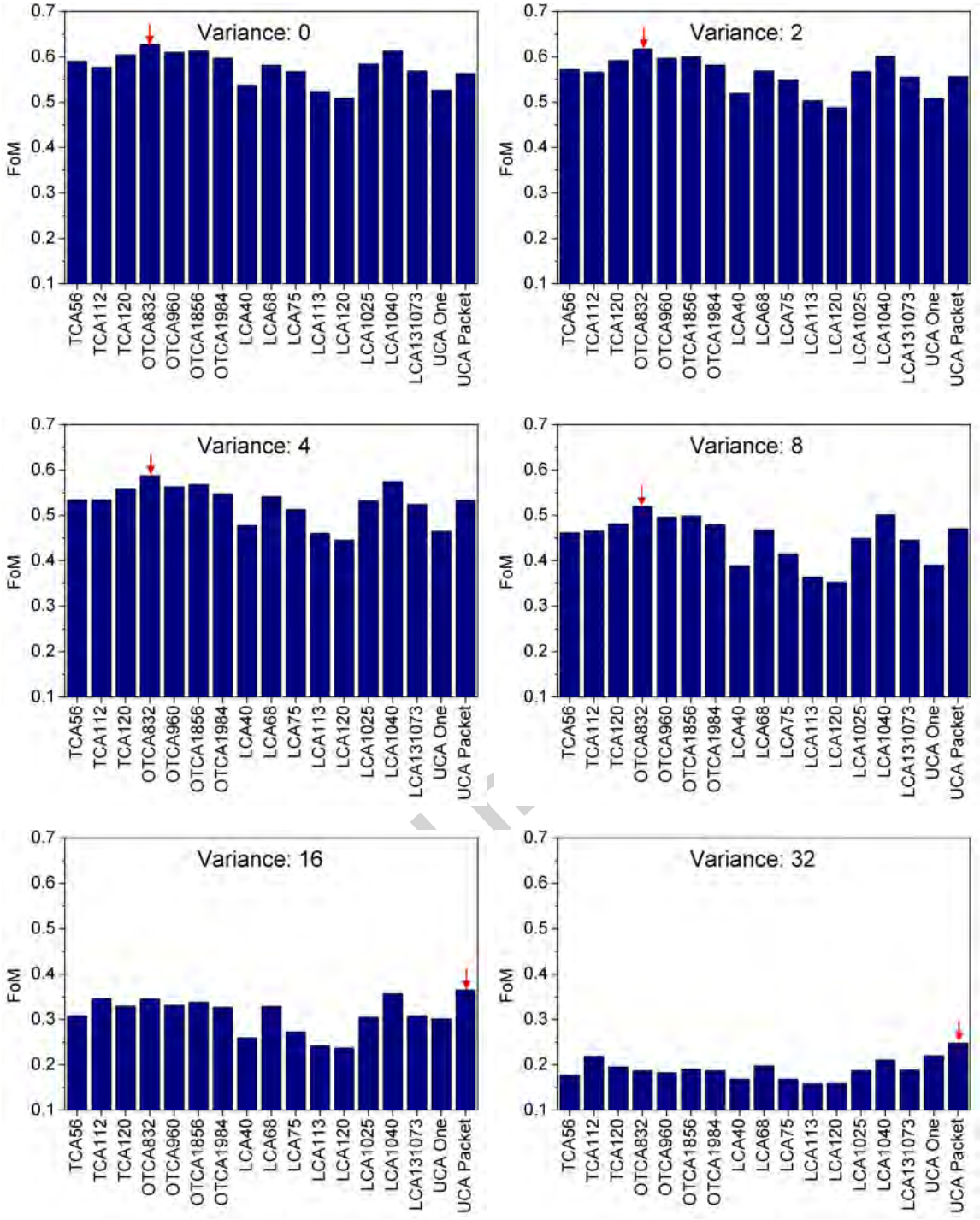
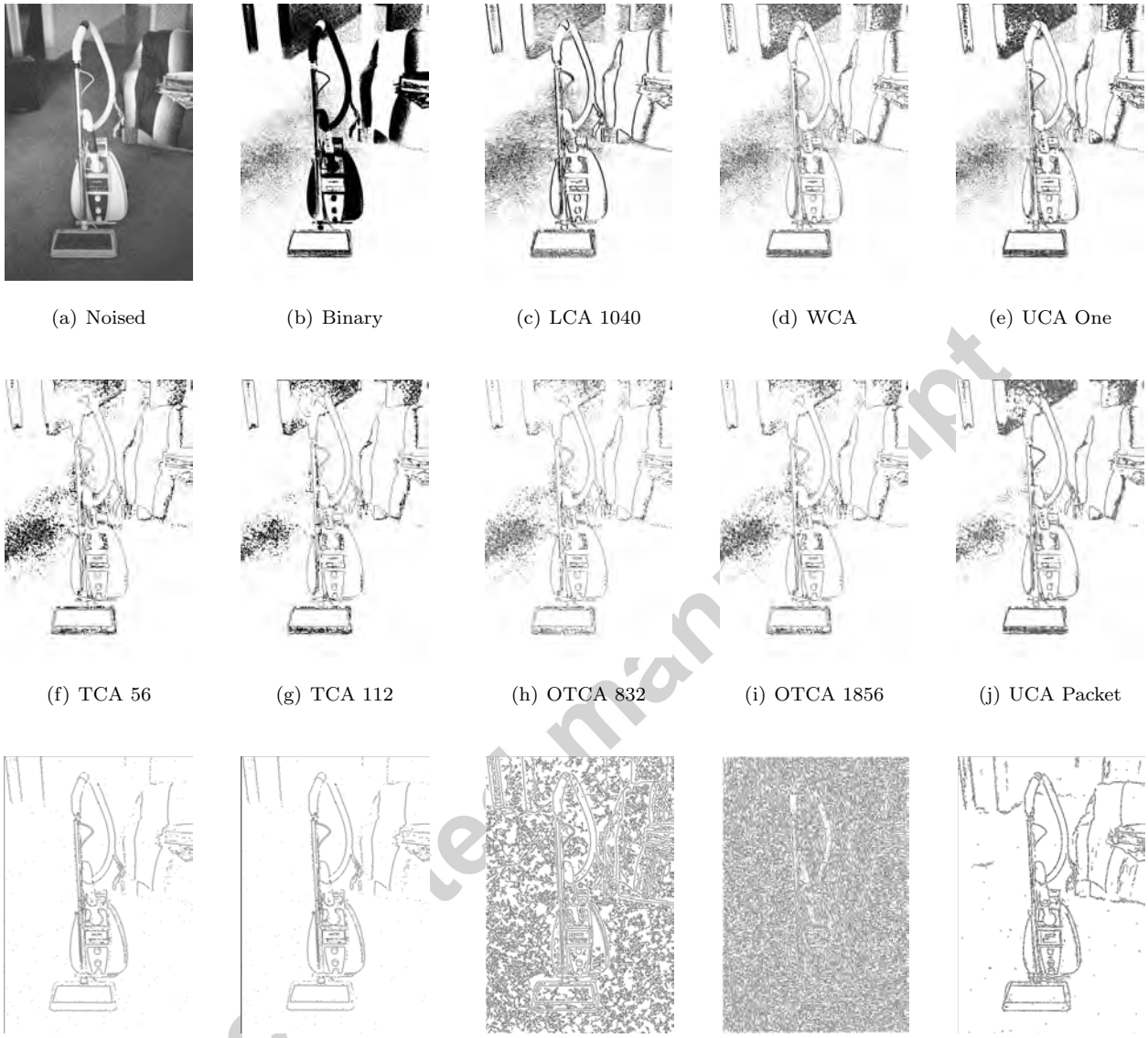


Figure 8: Figure of Merit for different levels of variations of Gaussian noise, where the red arrow highlights the maximum value. (Image 110 - vacuum)

From the comparison of Figure of Merit values against variance of added Gaussian noise, illustrated by Figure 8, it is noticed that similar with Salt & Pepper noise, the highest similarity for smaller variance levels was observed for OTCA. In the case of the two largest values (16, 32) the UCA with packet of rules presented the highest quality value. Unlike with S&P noise, the same discrepancy in results between different CA methods was not noticed. Figure 9 shows the influence of Gaussian noise with variance 16 on the results of selected methods. It has been observed that, as the variance level increases and larger noise areas are created, all linear methods tend to not only carry them into resulting edge images, but also enhance them. Resulting edges become more smeared and discontinuous, especially for rules that do not produce thick lines. As shown in Figure 9.(d) and (e), for both WCA and UCA with one rule the behaviour is identical as with Salt & Pepper noise. Resulting lines are thin and continuous, but noise is carried in attenuated fashion and enhanced, which results in almost filled areas. For TCA and OTCA rules the visual quality of resulting edges is similar, shown in Figure 9.(f)-(i), and are the best of all CA methods. With the increase of noise, TCA presents better connectivity of edges at the cost of connecting noise pixels creating clustered areas. The OTCA method results in more isolated pixels and less clustered areas.



(a) Noised (b) Binary (c) LCA 1040 (d) WCA (e) UCA One

(f) TCA 56 (g) TCA 112 (h) OTCA 832 (i) OTCA 1856 (j) UCA Packet

(k) Sobel (l) Prewitt (m) Canny (n) Susan (o) Kovesi

Figure 9: Influence of Gaussian noise with variance 16 on the result of selected CA rules and Sobel, Prewitt, Canny, SUSAN and Kovesi's methods.

3.4. Computational complexity

LCA [11]	UCA [11]	WCA [19]	TCA [18]	OTCA
1.1 ms	5.5 ms	5.6 ms	2.2 ms	2.4 ms

Table 1: Running times of all Cellular Automata methods. These values were calculated by averaging run times for 100 images.

The results of computation time always carry a lot of uncertainty, as too many factors influence the measurement that cannot be fully described. Different implementations, compilers, operating systems and workload carried by the system during testing can all impact the results. However, it is important to present an approximation of computational complexity described by running time of the proposed algorithm. All test based on Cellular Automata methods were performed on a laptop with Intel i5-3230 CPU and Windows 8.1 and compiled with Visual C++ 2013. The averaged running times for 100 testing images with the resolution of 481×711 are presented in Table 1. The implementation is not highly optimised, but all methods obviously present the ability to incorporate into real-time applications. All implementations were using a pre-calculated Look-Up Table to perform the calculation, thus invariance on rule number.

4. Conclusions

Although many methods based on CA have been developed and successfully applied to perform edge detection, a wide literature review shows that no literature has compared the relative performance among them, and also very few people have discussed the influence of noise to edge detection performance. This is a ground paper that firstly comprehensively compared the performance of other developed CA-based edge detectors as well as the proposed method in terms of different types and levels of noise.

Results conclude that the best accuracy has been achieved by TCA and OTCA among all CA-based methods regardless of the type or level of noise. The OCTA method has the best performance for all considered methods when the added S&P noise is less than 5%. With the increase of noise level, TCA takes advantage over OTCA. For all of proposed OTCA rules, resulting edges are thin and continuous. They present the same smoothing capabilities as TCA, resulting in clearer edges on noised borders. Moreover, the results clearly demonstrate that the proposed method has stronger resistance to both types of noise.

For the future work, an extension of proposed method making it possible to use grey-scale images as input might be considered. During the binarisation process, a lot of useful information

is lost. Since the rule space for grey-scale images is too large, an extensive study to determine the method of reduction of the number of possible rules should be performed. Additionally considering a larger neighbourhood might provide further improvement in performance.

Acknowledgment

This work was supported by the EPSRC Centre for Innovative Manufacturing in Through-life Engineering Services (Grant number EP/I033246/1).

References

[1] L. G. Roberts, Machine Perception of 3-D Solids, Optical and Electro-Optical Information Processing.

[2] E. Sobel, Camera Models and Machine Perceptions, Ph.D. thesis, Stanford University (1970).

[3] J. M. S. Prewitt, Object Enhancement and Extraction, Picture processing and Psychopictorics.

[4] D. Marr, E. Hildreth, Theory of Edge Detection, Proceedings of the Royal Society of London. Series B, Biological Sciences 207 (1980) 187–217.

[5] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (6) (1986) 679–698.

[6] S. M. Smith, J. M. Brady, SUSAN - A New Approach to Low Level Image Processing, International Journal of Computer Vision 23 (1) (1997) 45–78.

[7] J. S. Huang, D. H. Tseng, Statistical Theory of Edge Detection, Computer Vision, Graphics and Image Processing 43 (3) (1988) 337–346.

[8] E. Chuang, D. Sher, χ^2 Test for Feature Detection, Pattern Recognition 26 (11) (1993) 1673–1681.

[9] Y. Chen, Z. Yang, Y. Hu, G. Yang, L. Luo, W. Chen, C. Toumoulin, Thoracic low-dose ct image processing using an artifact suppressed large-scale nonlocal means, Physics in Medicine and Biology 57 (2012) 2667–2688.

[10] Y. Chen, L. Shi, Q. Feng, J. Yang, H. Shu, L. Luo, J. Coatrieux, W. Chen, Artifact suppressed dictionary learning for low-dose ct image processing, IEEE transactions on medical imaging 33 (12) (2014) 2271–2292.

[11] A. W. Burks (Ed.), Theory of Self-Reproducing Automata, University of Illinois Press, Champaign, IL, USA, 1966.

[12] N. R. Silva, P. Ween, B. D. Baets, O. M. Bruno, Improved texture image classification through the use of a corrosion-inspired cellular automaton, Neurocomputing 149, Part C (2015) 1560–1572.

[13] Y.-G. Yang, J. Tian, H. Lei, Y.-H. Zhou, W.-M. Shi, Novel quantum image encryption using one-dimensional quantum cellular automata, Information Sciences 345 (2016) 257–270.

[14] F. Qadir, K. A. Khan, Investigations of Cellular Automata Linear Rules for Edge Detection, International Journal of Computer Network and Information Security 4 (3) (2012) 47–53.

[15] S. Uguz, U. Sahin, F. Sahin, Uniform Cellular Automata Linear Rules for Edge Detection, in: 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 2945–2950.

[16] D. R. Nayak, S. K. Sahu, J. Mohammed, A Cellular Automata Based Optimal Edge Detection Technique using Twenty-Five Neighborhood Model, International Journal of Computer Applications 84 (10) (2013) 27–33.

[17] M. Diwakar, P. K. Pate, K. Gupta, Cellular Automata Based Edge Detection for Brain Tumor, in: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 53–59.

[18] S. Djemame, M. Batouche, Combining Cellular Automata and Particle Swarm Optimization for Edge Detection, International Journal of Computer Applications 57 (14) (2012) 16–22.

[19] S. Wongthanavasu, R. Sadananda, A CA-Based Edge Operator and its Performance Evaluation, Journal of Visual Communication and Image Representation 14 (2) (2003) 83–96.

[20] D. Safia, D. Oussama, B. Chawki, Image Segmentation using Continuous Cellular Automata, in: Programming and Systems (ISPS), 10th International Symposium on, 2011, pp. 94–99.

[21] C.-L. Chang, Y.-J. Zhang, Y.-Y. Gdong, Cellular Automata for Edge Detection of Images, Proccedings of the Third International Conference on Machine Learning and Cybernetics (2004) 3830–3834.

[22] Y. Chen, Z. Yan, A Cellular Automatic Method for the Edge Detection of Images, in: D.-S. Huang, I. Wunsch, DonaldC., D. Levine, K.-H. Jo (Eds.), Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Vol. 5227 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 935–942.

[23] P. Sharma, M. Diwakar, N. Lal, Edge Detection using Moore Neighborhood, International Journal of Computer Applications 61 (3) (2013) 26–30.

[24] P. K. Dhillon, A Novel Framework to Image Edge Detection using Cellular Automata, IJCA Special Issue on Confluence 2012 - The Next Generation Information Technology Summit CONFLUENCE (1) (2012) 1–5.

[25] T. Kumar, G. Sahoo, A Novel Method of Edge Detection using Cellular Automata, International Journal of Computer Applications 9 (4) (2010) 38–44.

[26] S. Wongthanavasu, Cellular Automata for Medical Image Processing, Cellular Automata - Innovative Modelling for Science and Engineering.

[27] P. L. Rosin, Training Cellular Automata for Image Processing, IEEE Transactions on Image Processing 15 (7) (2006) 2076–2087.

[28] P. L. Rosin, Image Processing using 3-State Cellular Automata, Computer Vision and Image Understanding 114 (7) (2010) 790–802.

[29] M. Batouche, S. Meshoul, A. Abbassene, On Solving Edge Detection by Emergence, in: M. Ali, R. Dapoigny (Eds.), Advances in Applied Artificial Intelligence, Vol. 4031 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 800–808.

[30] S. Slatnia, M. Batouche, K. E. Melkemi, Evolutionary Cellular Automata Based-Approach for Edge Detection, in: F. Masulli, S. Mitra, G. Pasi (Eds.), Applications of Fuzzy Sets Theory, Vol. 4578 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 404–411.

[31] M. Mraz, N. Zimic, I. Lapanja, I. Bajec, Fuzzy Cellular Automata: from Theory to Applications, in: 2th IEEE International Conference on Tools with Artificial Intelligence, 2000, pp. 320–323.

[32] K. Zhang, Z. Li, X. O. Zhao, Edge Detection of Images based on Fuzzy Cellular Automata, in: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007., Vol. 2, 2007, pp. 289–294.

[33] D. K. Patel, S. A. More, Edge Ddetection Technique by Fuzzy Logic and Cellular Learning Automata using Fuzzy Image Processing, in: 2013 International Conference on Computer Communication and Informatics (ICCCI), 2013, pp. 1–6.

[34] S. Sinaie, A. Ghanizadeh, E. M. Majd, S. M. Shamsuddin, A Hybrid Edge Detection Method Based on Fuzzy Set Theory and Cellular Learning Automata, in: Computational Science and Its Applications, 2009. ICCSA '09. International Conference on, 2009, pp. 208–214.

[35] P. Dollár, C. L. Zitnick, Structured Forests for Fast Edge Detection, in: ICCV, International Conference on Computer Vision, 2013.

[36] M. H. Mofrad, S. Sadeghi, A. Rezvanian, M. R. Meybodi, Cellular edge detection: Combining cellular automata and cellular learning automata, {AEU} - International Journal of Electronics and Communications 69 (9) (2015) 1282–1290.

[37] M. Han, X. Yang, E. Jiang, An extreme learning machine based on cellular automata of edge detection for remote sensing images, Neurocomputing (2016) In Press.

[38] S. Wolfram, A New Kind of Science, Wolfram Media, Champaign, IL, 2002.

[39] C. Marr, M. T. Hutt, Outer-totalistic cellular automata on graphs, Physics Letters A 37 (5) (2009) 546–549.

[40] R. Dogaru, I. Dogaru, M. Glesner, A smart sensor architecture based on emergent computation in an array of outer-totalistic cells, in: Bioengineered and Bioinspired Systems II, Vol. 5839, 2005, p. 254.

[41] I. E. Abdou, W. Pratt, Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors, Proceedings of the IEEE 67 (5) (1979) 753–763.

[42] K. Bowyer, C. Kranenburg, S. Dougherty, Edge Detector Evaluation using Empirical ROC Curves, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 1999, pp. 354–359.

[43] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man and Cybernetics 9 (1) (1979) 62–66.

An edge detection method using outer totalistic cellular automata

Amrogowicz, Sebastian

2016-05-20

Sebastian Amrogowicz, Yitian Zhao, Yifan Zhao, An edge detection method using Outer totalistic cellular automata, Neurocomputing, Vol 214, 19 November 2016, pp.643-653
http://dx.doi.org/10.1016/j.neucom.2016.05.092