Mathematical Problems in Engineering, 2002, Vol. 8, pp. 87–99

Taylor & Francis
Taylor & Francis Group

QCA Implementation of a Multichannel Filter for Image Processing

J. L. CARDENAS-BARRERA[a], K. N. PLATANIOTIS[b] and
A. N. VENETSANOPOULOS[c]

[a]Centre for Studies on Electronics and Information Technologies, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba; [b]Department of Electrical and Computer Engineering; [c]Multimedia Laboratory, The Edward S. Rogers Sr., University of Toronto, Toronto, Ontario, M5S 3G4, Canada

(Received 26 June 2001)

A novel way to implement a color image filter is introduced here. The quantum-dot cellular automata (QCA) framework is adopted due to its adequacy to implement highly parallel systems. The principle behind the new design is explained in detail, and simulation results are included to demonstrate the effectiveness of the proposed method.

Key words: Image processing, quantum-dot cellular automata, color imaging, median filters

1 INTRODUCTION

Filtering of multichannel images has recently received increased attention due to its importance in the processing of color images. It is widely accepted that color conveys information about the objects in a scene and that this information can be used to further refine the performance of an imaging system. The generation of high quality color images which are aesthetically pleasing is of great interest [1].

Noise in the image sequence may result from sensor malfunction, electronic interference, or flaws in the data transmission procedure. In considering the signal-to-noise ratio over practical mediums, such as microwave or satellite links, there would be a degradation in quality due to the low received signal. Degradation of the broadcasting quality can be also a result of processing techniques, such as aperture correction which amplifies both high frequency signals and noise [2].

The appearance of the noise and its effect is related to its characteristics. Noise signals can be either periodic in nature or random. Usually noise signals introduced during the transmission process are random in nature resulting in abrupt local changes in the image data. These noise signals cannot be adequately described in terms of the commonly used Gaussian noise models [3]. Rather, they can be characterized as 'impulsive' sequences which occur in the form of short time duration, high energy spikes attaining large amplitudes with probability higher than the probability predicted by a Gaussian density model.

There are various sources that can generate impulsive noise. Among others, man made phenomena, such as car ignition systems, industrial machines in the vicinity of the receiver,

ISSN 1024-123X print; ISSN 1563-5147 online/ © 2002 Taylor & Francis Ltd

switching transients in power lines and various unprotected electric switches. In addition, natural causes, such as lightning in the atmosphere and ice cracking in the antarctic region, also generate impulsive noise.

Impulsive noise is frequently encountered during the transmission of TV signals through UHF, VHF, terrestrial microwave links and FM satellite links. It is therefore important to develop a digital signal processing technique that can remove such image impairment in real-time and thus, guarantee the quality of service delivered to the consumers. Such a system is proposed here. A new two-stage color filter is developed. The color filter is applied on-line on the digitized image frames in order to remove image noise.

A number of digital techniques have been applied to the problem aiming to smooth out impulsive noise and restore TV images. In [4, 5] a multi-shell median filter has been introduced. The approach introduced in [5] is applicable only to gray-scale images. Since the TV signal is a color signal, such an approach can be applied only to the luminance component of the transmitted signal without any reference or association to the corresponding chrominance signals. However, there is some indication that noise correlation among the different image channels exists in real color images. Particularly, in the case of NTSC television broadcast signal, if there is any degradation of the chrominance signal that is broadcast, both the I and Q components would be affected simultaneously [6]. Therefore, noise removal operations on only one channel are not adequate and a multichannel filter is necessary to remove the noise and restore the originally transmitted signal.

The implementation of video processing techniques, such as digital video filters, is often carried out using VLSI parallel architectures or video processors [7]. The continuing development of smaller electronic devices into the nanometer regime offers great possibilities of highly parallel computing systems and then for video processing, as it allows to reduce power consumption, device sizes and to increase operating speed. As device sizes continue to shrink deeper into the nanometer regime, physical limitations of conventional electronics including power consumption, interconnect, and lithography will become increasingly difficult to overcome. Some viable electronic devices with dimensions on the order of nanometers have already being proposed [8]. Recently, Quantum-dot cellular automata (QCA) have been proposed as an alternative for nanoelectronic devices [9]. The advantage of QCA over VLSI is not only offering an improvement in size, speed and power consumption, but also offering a routing capability that VLSI lacks: the ability to cross wires in a plane. Then QCA is a new opportunity for the design of highly parallel algorithms and architectures.

Section 2 describes the operation and performance of a two stage directional marginal median filter. Section 3 discusses how this filter can be implemented using the QCA paradigm and Section 4 presents our conclusions.

2 A COLOR IMAGE FILTER FOR IMPULSIVE NOISE REDUCTION

Impulsive noise can be classified as a short duration, high energy spike which results in the alteration of the digital value of the image pixel. After the effect of the noise, the altered value of the image pixel usually differs from the corresponding values of the neighboring pixels. However, in TV signals, any kinds of scenes, pictures or images are transmitted. Thus, it is important for the filter to differentiate between impulsive noise and other image features such as, intended dots or thin lines in the image, which may resemble this kind of noise.

The class of median filters is considered the most appropriate for the removal of impulsive noise [2]. When dealing with univariate signals, the median filter operating over a window of $n = 2v + 1$ samples, selects the sample S_{v+1} from the ordered set of samples

$\{S_1, S_2, \ldots, S_{v+1}, \ldots, S_n\}$. The extension of the median definition for multivariate signals is not straightforward, since different ordering criteria can be followed [1, 10]. As a consequence, different median definitions based on those ordering can be stated. Two ordering principles are of practical interest for developing nonlinear filters for color image processing: marginal and reduced ordering [1].

In marginal ordering (M-ordering) scheme, the multivariate samples are ordered along each of the *p-dimensions* independently yielding:

$$\begin{aligned} \mathbf{x}_{1(1)} &\leq \mathbf{x}_{1(2)} \leq \cdots \leq \mathbf{x}_{1(n)} \\ \mathbf{x}_{2(1)} &\leq \mathbf{x}_{2(2)} \leq \cdots \leq \mathbf{x}_{2(n)} \\ &\cdots \qquad\qquad \cdots \\ \mathbf{x}_{p(1)} &\leq \mathbf{x}_{p(2)} \leq \cdots \leq \mathbf{x}_{p(n)} \end{aligned} \tag{1}$$

According to M-ordering principle, ordering is performed in each channel of the multichannel signal independently. The marginal median is defined as $\mathbf{x}(v+1) = [x_{1(v+1)}, x_{2(v+1)}, \ldots, x_{p(v+1)}]^\tau$ for $n = 2v + 1$, which may not correspond to any of the original multivariable samples. In contrast, in the scalar case there is an one-to-one correspondence between the original samples $\mathbf{x}_i$ and the order statistics $\mathbf{x}_{(i)}$.

In reduced ordering (R-ordering) each multivariate observation $\mathbf{x}_i$ is reduced to single, scalar value by means of some combination of the component sample values. The resulting scalar values are then amenable to univariate ordering. R-ordering is based on the generalized distance

$$R(\mathbf{x}, \bar{\mathbf{x}}, \Gamma) = (\mathbf{x} - \bar{\mathbf{x}})^\tau \Gamma^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \tag{2}$$

where $\bar{\mathbf{x}}$ is a location parameter for the data set, or underlying distribution, into consideration and Γ is a dispersion parameter with Γ^{-1} used to apply a differential weighting to the components of the multivariate observation inversely related to the population variability.

A variation of the R-ordering scheme that employs a dissimilarity (or alternatively similar) measure of the set of $\mathbf{x}_i$ has been used to obtain a vector median definition that has similar properties to those of the median operation of the scalar case, and which can also be reduced to it if the number of dimensions reduces to one [1]. If the aggregate distance of the sample $\mathbf{y}_i$ to the set of vectors $W(n) = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ is employed as a reduction function then,

$$d_i = \sum_{j=1}^{n} d(\mathbf{y}_i, \mathbf{y}_j) \tag{3}$$

where $d(\mathbf{y}_i, \mathbf{y}_j)$ represents an appropriate distance (or similarity) measure. The arrangements of d_i's in ascending order, associates the same ordering to the multivariate samples. Thus, an ordering

$$d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(n)} \tag{4}$$

implies the same ordering to the corresponding $\mathbf{y}_i$'s. The vector median value can be defined as the vector $\mathbf{y}_{VM}$ contained in the given set whose distance to all other vector is minimum.

$$\sum_{j=1}^{n} d(\mathbf{y}_{VM}, \mathbf{y}_j) \leq \sum_{j=1}^{n} d(\mathbf{y}_i, \mathbf{y}_j) \quad \forall i \in [1, 2, \ldots, n] \tag{5}$$

In the ordered sequence of (5), $\mathbf{y}_{VM} = \mathbf{y}_{(1)}$. The most commonly used measure is the L_1 norm. Even using this distance measure, the computational load is relatively high because we have to calculate $n(n-1)/2p$-*dimensional* norms and to perform $[n(n-1)/2]-1$ comparisons.

If multivariate data are strongly correlated among their different dimensions, the marginal and the vector median tend to have the same value and similar behaviour [10]. Hence, in most practical cases, the marginal median is preferred, as it has lower computational complexity.

Repeated applications of a median filter in a filtering window centered around a pixel of the image will probably remove the noise but will also reduce the resolution of the image by filtering out thin lines and details. Similarly, using a larger size of filtering window (*e.g.*, 5×5 instead of 3×3) might result in better noise removal, but will blur the fine details of the image. Thus, to filter out noise and preserve image details a different approach is necessary.

A two stage adaptive median filter is introduced. As with any other nonlinear filter, a working area (window or template) is centered around an image pixel [2, 11]. To prevent thin lines and intended spots in the image from being altered through the nonlinear filtering process, we applied directional median filters inside the processing window. In other words, instead of a combined median filter applied to the whole window, four different median filters are applied across the four main directions at 0°, 45°, 90°, 135° (Fig. 1). The pixel at the window center (pixel under consideration) belongs to all four sets. If the pixel under consideration has considerably larger or smaller values than those of the other pixels along a specific direction it will be treated as an outlier and it will be replaced by the median value across this specific direction. Otherwise the value remains unchanged during this operation.

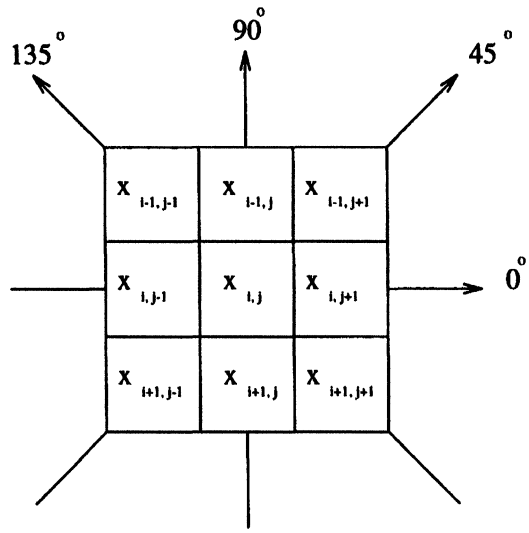


FIGURE 1 The new filter.

Thus, by employing filtering across the main directions, lines and other fine details will be preserved. In a second stage, another median operates on the four filtered results to generate the final output.

The mathematical description of the filter can be summarized as follows:

Let $y(x) : Z^l \rightarrow Z^m$, represent a multichannel signal and let $W \in Z^l$ be a window of finite size $n \times n$ (square window with filter length n^2), where n is generally an odd number. The pixel under consideration $x_{i,j}$ is at the window center. The noisy vectors (n^2 in total) inside the window W are noted as (see Fig. 1):

$$x_{i+k,j+l}, \quad k, l = 0, \pm 1, \pm 2, \ldots, \pm \frac{(n-1)}{2} \tag{6}$$

The median filter applied along the 0° direction operates on the horizontal pixels, across and including the center pixel $x_{i,j}$, noted as (see Fig. 1):

$$x_{i,j+l}, \quad l = 0, \pm 1, \pm 2, \ldots, \pm \frac{(n-1)}{2} \tag{7}$$

For simplification and clarity, let these vectors be $h_1 \ldots h_n$ (h stands for horizontal direction). Now, according to M-ordering, the corresponding ordering for the h_p's is:

$$h_{(1)} \leq h_{(2)} \leq \cdots \leq h_{(n)}, \tag{8}$$

where, $h_{(p)}$ is the pth order statistics [2]. The marginal median y_1 along the 0° direction is defined as:

$$y_1 = h_{(v+1)} \tag{9}$$

Similarly, the process is repeated for the other three directions. The vectors $f_p, p = 1, \ldots, n$ (f stands for 45° direction) representing those pixels along the 45° direction are (see Fig. 1):

$$x_{i-k,j+k}, \quad k = 0, \pm 1, \pm 2, \ldots, \pm \frac{(n-1)}{2} \tag{10}$$

The marginal median y_2 along the 45° direction is as follows:

$$y_2 = f_{(v+1)} \tag{11}$$

For the 90° direction, the corresponding vectors $v_p, p = 1, \ldots, n$ (v stands for vertical, *i.e.* 90° direction) are (see Fig. 1):

$$x_{i-k,j}, \quad k = 0, \pm 1, \pm 2, \ldots, \pm \frac{(n-1)}{2} \tag{12}$$

The marginal median y_3 along the 90° direction is given as:

$$y_3 = v_{(v+1)} \tag{13}$$

Finally, the vectors $r_p, p = 1, \ldots, n$ (r stands for reverse 45°, *i.e.* 135° direction) representing those pixels along the 135° direction are (see Fig. 1):

$$x_{i-k,j-k}, \quad k = 0, \pm 1, \pm 2, \ldots, \pm \frac{(n-1)}{2} \tag{14}$$

The marginal median y_4 along the 135° direction is defined as:

$$y_4 = r_{(v+1)} \tag{15}$$

In the second stage, a marginal median filter is applied to the four marginal median outputs y_1, y_2, y_3 and y_4 obtained in the directional filtering of the previous stage. Hence, the final output x_{DMMF} of this Directional Marginal Median Filter (DMMF) is derived as:

$$x_{\text{DMMF}} = y_{(v+1)} \tag{16}$$

where, $y_{(v+1)}$ is the $(v+1)$th order statistic of the ordered sequence of vectors $y_p, p = 1, \ldots, 4$.

This new DMMF is applied to a color image of Lenna to assess qualitatively the performance. The original image is corrupted such that some lines are partially or totally missing (Fig. 2). Then, the DMMF, with a window size of 5×5, is applied to the corrupted image and the filtered output image is displayed and compared visually with the original image. As can be seen from Figure 3, no impulsive noise is visible. In addition, all the edge information, thin lines and fine details, are well preserved.

FIGURE 2 Lenna image with missing line.

FIGURE 3 Filtered results of Figure 2.

3 QUANTUM-DOT CELLULAR AUTOMATA IMPLEMENTATION OF THE DMMF

The proposed methodology can be applied online to any image processing systems. Since it is a digital image processing technique, analog-to-digital (A/D) converters are needed if the incoming signal is analog. Once in digital form, the image is processed using the described technique, and an appropriate, parallel implementation, approach should be taken in order to achieve real-time operation.

The main structural block in this design is the marginal median filter (Fig. 4). The parallel implementation of a median filter can be addressed using a number of architectures [12]. Among them, the stack filter implementation avoids time-consuming ordering of input samples and expedites the process through the utilization of logic gates for its implementation. Due to its low latency time and ease of implementation, we decided to proceed with the stack implementation of the marginal filter.

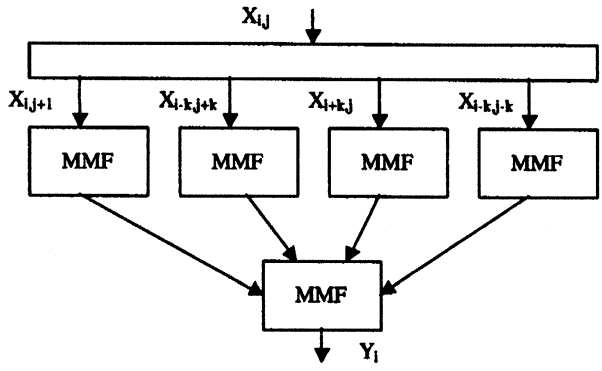


FIGURE 4 Filter implementation.

Stack filters belong to a class of nonlinear filters that satisfy the threshold decomposition and the stacking property [13]. A general expression for these filters is as follows:

$$y(k) = Sf(x_k) = \sum_{n=1}^{M} f_b(\tau_n(x_k)) \tag{17}$$

The threshold decomposition operation (τ) allows to represent integer numbers, normally represented with b bits, as 2^b bits numbers with an ordering scheme in which all 1's are group together and followed by a string of 0's.

$$\tau_m(x_i) = \begin{cases} 1, & \text{if} \quad x_i \geq m \\ 0, & \text{if} \quad x_i < m \end{cases}, \tag{18}$$

$$\tau_{m_1}(x_i) \geq \tau_{m_2}(x_i) \quad \forall m_1 \leq m_2 \tag{19}$$

This distribution of bits is called a stack. Every bit level (from 1 to 2^b) in the new representation is called a layer. The filtering operation can be performed at each layer using a Boolean filter (f_b). The final output is computed adding the results of every Boolean filter. This adding operation can be performed very efficiently if the outputs of the filters hold the stacking property [13],

$$f_b(\tau_{m_1}(x_i)) \geq f_b(\tau_{m_2}(x_i)) \quad \forall m_1 \leq m_2 \tag{20}$$

Stack implementation leads to large arrays, since the architecture size grows exponentially with the number of bits of the input samples, but the use of nanoelectronic devices opens a new opportunity to this kind of implementations [8, 14].

A Quantum-dot Cellular Automaton (QCA) is an array of cells, in which each cell is coupled to a group of neighboring cells, through classical Coulombic interaction. A typical cell (Fig. 5) in the structure consists of five quantum wells and two electrons, leading to two possible polarization states within the cell. These two cell states can be regarded as binary values and, in a structure of identical cells, the polarization of an input cell will induce a specific polarization in all the cells in the array. By combining QCA cells in a particular layout, combinational digital logic can be constructed. Actually, digital logic has already been designed and tested using coupled quantum-dot cells (Fig. 6). Logic gates and wires, more complicated logic devices, and memory suitable for general purpose computing have been proposed in [15, 16]. Due to the coupling mechanism of the structure, the interconnection problem of classical electronic devices is avoided, and energy needs only to be supplied at the edges of the array [9].

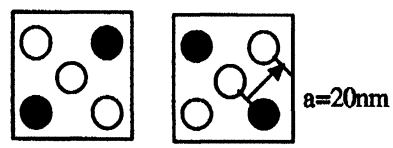


FIGURE 5 QCA cells.

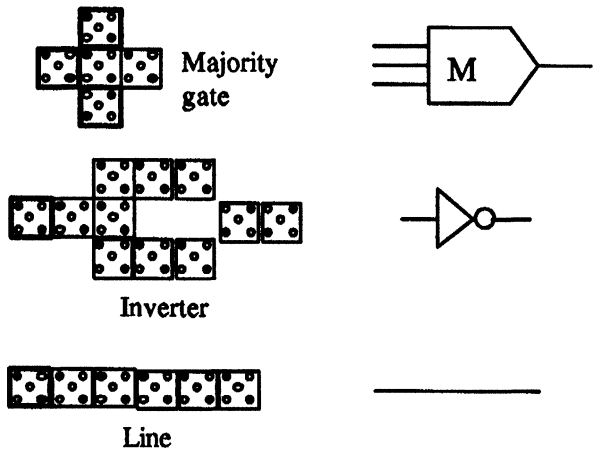


FIGURE 6 Some QCA logic devices: a majority cell, an inverter and a wire.

The fundamental QCA logical circuit is the majority gate (Fig. 6). With x_1, x_2 and x_3 being the inputs, the output of the gate forms a 3-point binary median and its corresponding Boolean function is:

$$\mathrm{MAJ}(x_1, x_2, x_3) = x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3 \tag{21}$$

Larger median filters can be implemented by combining this QCA fundamental cell [14].

The realization of the directional marginal median filter (DMMF) using QCAs can be carried out as follows. As color planes are treated separately, a marginal median filter for an RGB is nothing more than 3 scalar median filters, each operating on one plane. Let us consider one of the color planes and five samples per direction inside the operating window (Fig. 7). Within the QCA structure three main steps should be considered, namely: (*i*) threshold decomposition, (*ii*) binary median filters and (*iii*) adding. Therefore, the filtering operation over the (5 × 5) processing window can be implemented as follows:

- Define a processing direction.
- Threshold decompose an element into $M = 2^b$ bits.
- Apply the first stage binary filters at every layer $m (m = 1, 2, \ldots, M)$ using a 5-bit binary median filter implementing in this way each directional median filter.
- Apply the second stage 4-bit binary filters at every layer m using as inputs the outputs of previous first-stage binary filters.
- Obtain the filtered value as the sum of the outputs from second-stage filters.

3.1 Threshold Decomposition

Every b-bit sample on each direction within a processing window is threshold decomposed into $M = 2^b$ bits. The samples are compared to each threshold value m, $1 \leq m \leq M - 1$ and is decomposed into a set of binary values $\tau_m(x_k)$. In this work, threshold decomposition is carried out using a slight variation of the scheme that has been proposed for adding (Section 3.3), instead of using transportation networks as proposed in [14]. In this way, the operation is completed without any comparisons. Figure 8 presents the operation of the

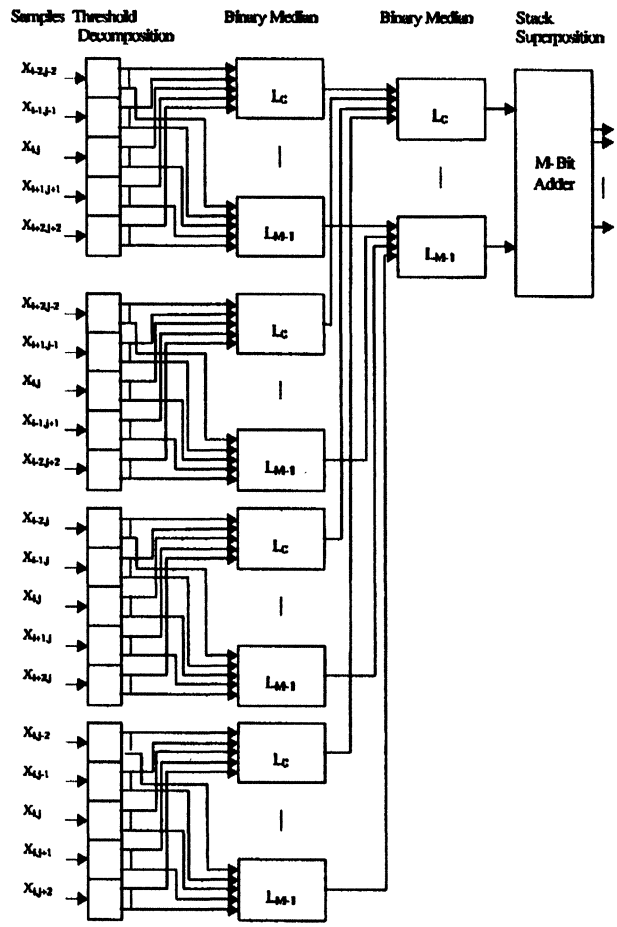


FIGURE 7 Stack filter implementation.

architecture for threshold decomposing 3-bit input samples into 8 bits, along with the QCA implementation of the switches.

3.2 Binary Filtering

The bits corresponding to layer $m(m = 1, \ldots, M-1)$ from a specific direction are filtered using a 5-bit binary median filter, implementing in this way each directional median filter. QCA devices for realizing 5-bit median filters using fundamental logic gates can be found in [14]. This operation can be expressed as:

$$\text{MEDIAN}(x_1, x_2, x_3, x_4, x_5) = \text{MAJ}(\text{MAJ}(x_1, x_2, x_3), \text{MAJ}(x_2, x_3, x_4), \text{MAJ}(x_3, x_4, x_5)) \quad (22)$$

It only uses 4 majority cells in an area of 19×15 cells. Assuming, following [17], QCA cells with a near-neighbor dot distance $a = 20\,\text{nm}$ (Fig. 5) and separated by $3a$, the array occupies an area of approximately $1.13 \times 0.9\,\mu\text{m}^2$.

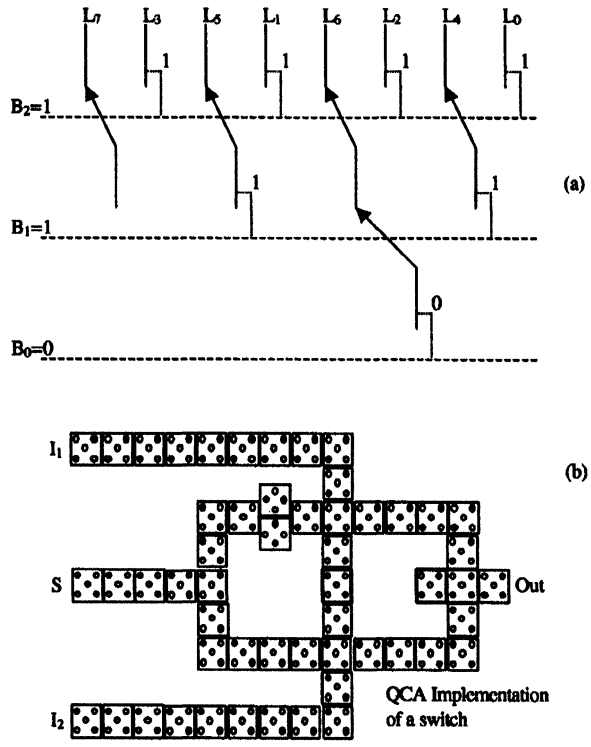


FIGURE 8 (a) Threshold decomposition implementation using switches. (b) implementation of a QCA switch.

The subsequent filtering stage is carried out through M 4-bit binary filters. The QCA structure remains the same than the one used for the previous stage, fixing one of its inputs either to 1 or 0. In this case, as there is an even number of inputs, the binary median may not always be uniquely determined. Hence, a fifth binary input has to be selected as a fixed input for this QCA structure. As can be seen from Figure 7, the cascade of filters can be implemented without getting back to the classical binary representation.

3.3 Adding

The filtered value of an operating window is obtained by adding the results of all second-stage binary median filters. The filtered output results from every binary filter on the last stage hold the stacking property, since only positive Boolean functions have been used. Then, its transformation into the standard binary-weighted representation is carried out by using an efficient adder implementation that performs the sum by detecting the 1 to 0 transition in a binary representation that holds the stacking property [13]. Figure 9 depicts this adder when $b = 3$ bits is used. Switches are implemented as in Figure 8.

3.4 QCA Operation

To perform computation in the QCA paradigm, some (input) cells are held in a fixed state and the rest of the array is allowed to relax to a ground state. During relaxation, cells are switched

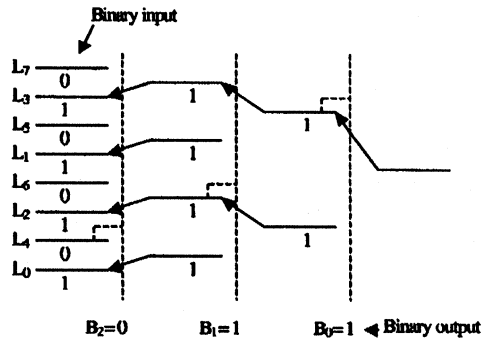


FIGURE 9 Adder implementation using switches.

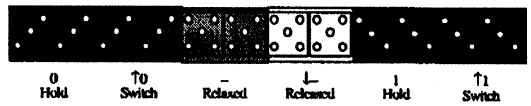


FIGURE 10 Adiabaticswitching of a line.

to whatever polarization lowers the overall system energy. Reaching the ground state may not always be possible, since the system can be stuck in a metastable state. In order to achieve proper performance of the QCA implementation, adiabatic switching [17] of the array can be used. With adiabatic switching the clock in QCA is multi-phased and cells are group together into pipelined subarrays. Four phases are used: (*i*) Switch, (*ii*) Hold, (*iii*) Release and (*iv*) Relaxed (Fig. 10). A single potential controls the inter-dot barriers of all the cells within a sub-array. Thus, a sub-array (in Switch phase) can perform its calculations based on the outputs of a previous sub-array (in Hold phase) that has been "frozen" by raising its inter-dot barriers. The successor sub-array does not influence the computations, since it is kept in an unpolarized state (Relaxed phase). The Release phase is used to prepare a subarray to undergo the Relaxed phase by lowering the inter-dot barriers and allowing the subarray to relax to an unpolarized state. Pipelining is a natural and useful consequence of this scheme, and with it no registers are needed for delay compensation.

Threshold decomposition, first stage filters, second stage filters, and adding can each pertain to different clock phases. Actually, many clock phases should be used for the design to operate at a high clock rate [16, 17]. As pipelining is achieved, the structure can be shared for filtering various pixels concurrently.

4 CONCLUSIONS

A new adaptive filter was introduced in this paper. The new filter perfectly suitable for real time implementation was used to remove impulsive noise and other impairment from color TV signals. Experimental results have been used to illustrate our discussion and to demonstrate the effectiveness of our method. In addition, we have proposed a QCA implementation which makes the proposed solution particularly attractive. With the advent of the all-digital TV system, such filters can lead to systems which would retain accurate image reproduction fidelity despite any unforeseen transmission developments.

References

[1] Venetsanopoulos, A. N. and Plataniotis, K. (2000) *Colour Image Processing and Applications*. Springer Verlag, ISBN 3-540-66953-1.

[2] Pitas, I. and Venetsanopoulos, A. N. 1990 *Nonlinear Digital Filters: Principles and Applications*. Norwell Ma: Kluwer Academic.

[3] Pitas, I. and Venetsanopoulos, A. N. (1992) Order statistics in digital image processing, *Proceedings of IEEE*, **80**(12), 1893–1923.

[4] Siu, J., Li, J. and Luthi, S. (1993) A real-time 2-D median based filter for video signals, *IEEE Trans. on Consumer Electronics*, **39**(2), 115–121.

[5] Juan, C. J. (1995) Modified 2D median filter for impulse noise suppression in a real-time system, *IEEE Trans. on Consumer Electronics*, **41**, 73–80.

[6] Grob, B. (1984) *Basic Television and Video Systems*. NY: McGraw-Hill.

[7] Implementation of an image processing library for the TMS320C80 (July 1997) Application Note no. BPRA059, Texas Instruments.

[8] Fountain, T. J., Duff, M. J. B., Crawley, D. G., Tomlinson, C. D. and Moffat, C. D. (1998) The use of nanoelectronic devices in highly parallel computing systems, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **6**(1), 31–38.

[9] Lent, C. S., Tougaw, P. D., Porod, W. and Bernstein, G. H. (1993) Quantum cellular automata, *Nanotechnology*, **4**(1), 49–57.

[10] Pitas, I. and Tsakalides, P. (1991) Multivariate ordering in color image filtering, *IEEE Transactions on Circuits and Systems for Video Technology*, **1**(3), 247–296.

[11] Venetsanopoulos, A. N. and Plataniotis, K. N. (1995) Colour Image Software, *Proceedings of the European Conference on Circuits and Systems Design*, 247–251.

[12] Chakrabarti, C. and Lucke, L. E. (2000) VLSI architectures for weighted order statistic (WOS) filters, *Signal Processing*, **80**(8), 1419–1433.

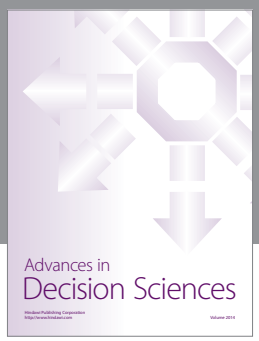
[13] Wendt, P. D., Coyle, E. J. and Gallager, N. C., Jr. (1986) Stack filters, *IEEE Transactions on Acoustic, Speech and Signal Processing*, **ASSP-34**, 898–911.

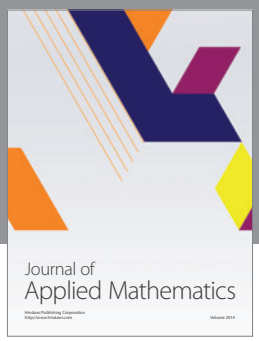
[14] Helsingius, M., Kuosmanen, P. and Astola, J. (1997) Nonlinear filters using quantum-dot cells, *Electronics Letters*, **33**(20), 1735–1736.

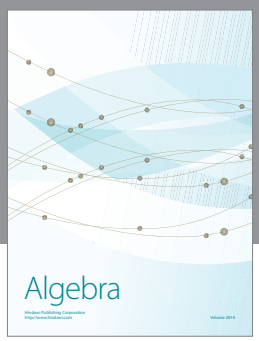
[15] Tougaw, P. D. and Lent, C. S. (1994) Logical devices implemented using quantum cellular automata, *Journal of Applied Physics*, **75**(3), 1818–1825.

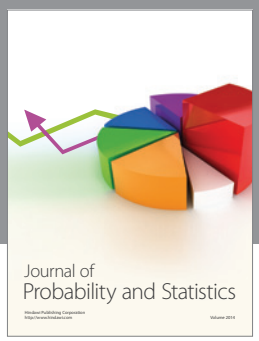
[16] Niemier, M. T. and Kogge, P. M. (1999) Logic in wire: using quantum dots to implement a microprocessor, *Proceedings of the 9th Great Lakes Symposium on VLSI*, 118–121.

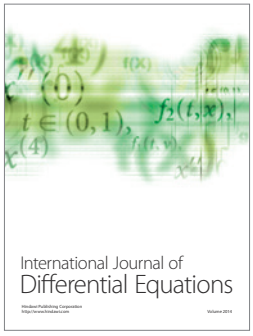
[17] Lent, C. S. and Tougaw, P. D. (1997) A device architecture for computing with quantum dots, *Proceedings of the IEEE*, **85**(4), 541–557.

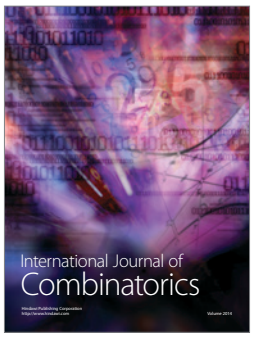


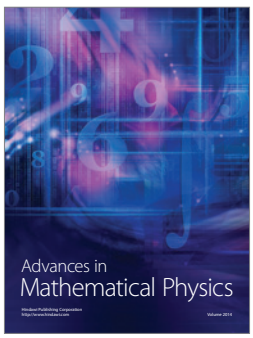


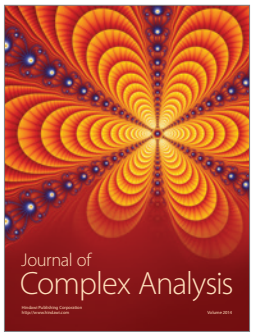


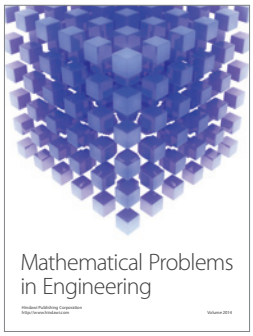


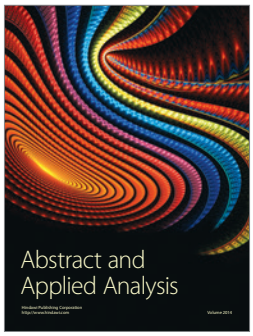


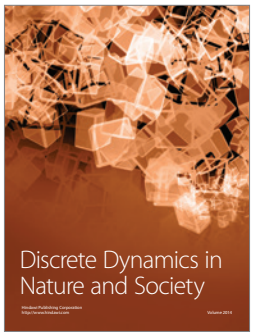


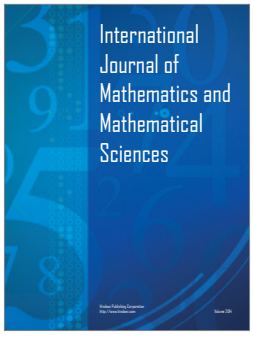


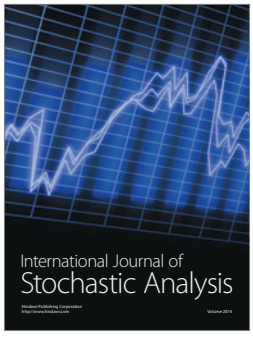












Hindawi
Submit your manuscripts at
http://www.hindawi.com