



## Non Linear Image Processing with Evolvable Hardware Filter

### KEYWORDS

Evolvable Hardware filters, Edge detector, Image enhancement, configurable logic block

**Syed Jahangir Badashah**

Department of Electronics & Communication Engineering, Research Scholar of Sathyabama University, Chennai, INDIA

**Dr. P. Subbaiah**

Professor in ECE Dept, Dhanalakshmi College of Engineering, Tambaram, Chennai, INDIA

**ABSTRACT** In recent years, nonlinear image processing is increasing in applications, and are better than linear filters in removing noise without disturbing image features. Nonlinear Evolvable Hardware Filters are introduced to describe the behavior of filters in the presence of fluctuating signals, edge detection and impulsive noise. The edge detection analysis is especially informative as the degree to which a filter preserves all the angles. Evolvable Hardware Filter proposed in this work filters without priori information. The image processing is a computational intensive task with applications in various engineering fields. In several image processing applications, a significant amount of computing power is committed to image enhancement operations, basic segmentation and identification of regions of interest for further analysis. Such type of processing can be done efficiently done by Evolvable Hardware filters.

### I. INTRODUCTION

Typical non-linear image processing applications include the correction of non-linear distortion introduced by components, communication channels and compensation for non-linearity in input output devices. Slow SNR variations can typically arise due to changing noise characteristics from parameters such as weather, distance or input power. Similarly, a lack of synchronization in capturing a moving image with a static camera or vice-versa can result in rapid SNR variation effects. Traditional adaptive filter works in a rectangular window whose size varies during filter operation depending on certain conditions. The optimal size of the window is crucial and influences the computation and memory requirements. However, this optimal size selection is highly application specific and its convergence from an initial value requires experimentation with various sizes of standard filters.

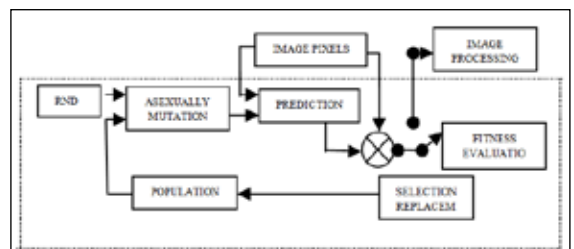
### II. RELATED WORKS

Evolvable Hardware for the Generation of Sequential Filter Circuits by Robert Thomson et al., describes a unified approach, on how Finite Impulse Response Primitive Operator Filters (FIRPOFs) can be created so that they operate sequentially, over two cycles. These filters were optimized for area and delay. Two factors with limited performance for the POF creation problem are the lesser number of components needed, and the area consumed by the accumulation block. The performance of the EA system is highly dependent upon the relative areas of the components being used. This technique could be applied to the creation of other types of sequential circuit. Sekanina has achieved evolutionary design of image filters with virtual reconfigurable circuits in extrinsic EHW environment. Digital image processing operations, such as image smoothing, edge detection, and image compression, have been carried out in an extrinsic EHW environment. This exhibits the potential of EHW in digital image operator design. This work presents complete evolvable hardware architecture, dedicated for implementing high performance digital image filters on FPGA so that the time for the evolution can be greatly reduced. Reconfigurable hardware devices offer both the flexibility of computer software, and the ability to construct custom high performance computing circuits. Thus, in many cases they make a good compromise between software and hardware solutions. The structure of a reconfigurable hardware device can be changed any number of times by downloading into the device a software bit string called configuration bits.

From the Literature review surveyed, so far, it has been observed that current evolutionary techniques has practical limitation when applied for complex real world problems. Also, the search spaces can become vast for large circuits and a greater deal of research needs to be directed at scalability. Hence, in this work, it is presented that, one can still evolve circuits with limited interactions and so can be used by traditional designers as building blocks for larger circuits. Initial research involved evolving circuits at a very high primitive gate level and results obtained using this approach showed that evolved circuits were less useful for more demanding commercial applications. Hence, to overcome this problem a function-level evolution is proposed in this work and domain knowledge is used to select high level computational units, which can be represented directly in the chromosome.

### III. CONFIGUREABLE LOGIC BLOCK ARCHITECTURE FOR IMAGE PROCESSING

The Configureable logic block (CLB) engine architecture, which can be divided into optimization module and application module at function level, which is shown in Figure 1. The optimization module and the application module are independent of each other i.e., while  $K^{\text{th}}$  image is optimized, the  $(K-1)^{\text{th}}$  optimized image can be processed. The optimization module is the most important and complex part in the system, and has the ability to give the optimal predictive parameters through learning the features of different images within the prescribed time. The optimization module consists of the population registers module, the mutation module, the fitness evaluation module and the selection module. Pipeline registers are inserted among them, so that all the modules can process the data in parallel to speed up the optimization.



**Figure 1** CLB engine architecture  
A. Evolutionary Circuit Design

The idea behind evolutionary circuit synthesis/design is to use a genetic search/optimization algorithm that operates in the space of all possible circuits and determines solution circuits with desired functional response. The genetic search is tightly coupled with a coded representation of the candidate circuits. Each circuit gets associated a "genetic code" or chromosome. The simplest representation of a chromosome is binary strings, (a succession of 0s and 1s) that encode a circuit. Synthesis is the search in the chromosome space for the solution corresponding to a circuit with a desired functional response. The genetic search follows a "generate and test" strategy. A population of candidate solutions is maintained each time; the corresponding circuits are then evaluated and the best candidates are selected and reproduced in a subsequent generation, until a performance goal is reached.

**IV. REQUIREMENTS AND MERITS OF CLB**

In CLBs based design, there is often a requirement for a fast partial reconfiguration, including the controllable granularity of configurable elements and a transparent structure of the configuration data. Ideally, a specialized reconfigurable device should be constructed for a given application in order to meet its particular requirements. However, developing an application specific integrated circuit (ASIC) is less feasible for many (mainly economic) reasons. A CLB is, in fact, an implementation of a domain-specific reconfigurable circuit on top of an ordinary programmable hardware device. The designer can construct the CLBs. Typical evolution of logic circuits, image filters, sorting networks etc. As the evolutionary algorithm (EA) can be implemented in the same hardware in case of these applications, a fast configuration interface can be established.

On the pessimistic side, the implementation of CLBs is relatively expensive in terms of gates used, since, interconnection circuits are based on area-expensive multiplexers. However, on the optimistic side, they can be designed totally independent of a target platform. Also, they can be utilized (in connection with a hardware implementation of the evolutionary algorithm) to implement soft evolvable IP cores. CLBs at the level of IP cores has become an integrated component in a variety of systems. These systems are responsible for completing tasks that are difficult for conventional hardware solutions, for instance, adaptation of functionality, adaptation of sensing, autonomous self-repairing and learning. As a result, there is an increasing interest in the use of CLBs in standalone applications (space applications) and in extreme environments exhibiting increased radiation and temperature levels.

A. Coordinate Logic Dilation (CLD) CLD of the images G by the structuring elements B is denoted by  $(g(i,j))$ , or  $\cdot$ , or is in eqn. [1]

Where  $(s_{k(i,j)})$  denotes the dilation operations on the binary value  $s_k(i,j)$  by the structuring elements B, given by  $(s_k(i,j)) \in \text{OR}(S_k(i,j)) \in B$ .

**B. Coordinate Logic Erosion (CLE)**

CLE of the image G by the structuring elements B is denoted by  $G \ominus B$  and is given in eqn. [2]

where  $(s_k(i,j))$  denotes the erosion operation on the binary values by the structuring elements B, given by  $(s_k(i,j)) \in \text{AND}(S_k(i,j)) \in B$ .

**V. IMAGE ANALYSIS AND PATTERN RECOGNIZATION APPLICATIONS**

Typical image analysis and pattern recognition applications using the CL filters are received. These applications are grouped into two classes:

- (i) Implementation of morphological operations using

CLOs, and

- (ii) Novel technique which is not based or related to morphology and are based on the exploitation of the properties, the inherent structure and the characteristics of the CLOs.

**A. Edge Extraction**

Edge extraction in an image G can be achieved with CL filters using the same approach adopted with morphological filter with the eroded image  $G \ominus B$  subtracted from the original image G, so that the edge detector is  $G - G \ominus B$ . Edges in different orientations can be obtained by using a 1D structuring element. The size of the structuring elements controls the thickness of the edge markers. Among the variety of the CL-based edge detectors, an efficient one that corresponds to the morphological  $G \ominus B - G \ominus B$  edge detector and given very similar results which is shown in the eqn. [3]

$$[(G \ominus B) \text{ CXOR } G] - (G \ominus B) \text{ CXOR } G \dots \dots \dots (3)$$

where A CXOR B represents a measurement of the difference between A and B.

A novel approach for edge extraction and enhancement is based on the direct application of CL filters to the original image, without using an arithmetic subtraction between images. The edge extraction results is given in eqn. [4]

$$f(i,j) = g(i,j) \text{ CAND } \left[ \text{CNOT} \left[ \begin{matrix} G(i-1,j) \text{ CAND } g(i=1,j) \\ \text{CAND } g(i,j+1) \text{ CAND } g(i,j-1) \end{matrix} \right] \right] \dots \dots (4)$$

**B. Noise removal – Coordinate Logic Filter**

Efficient filter for noise removal may be built using CLOs and the underlying Boolean operations. In binary signals, the majority function is a simple predicate whose output has the same value (1 or 0) as the majority of the population 'b' of the input units. If 'b' is odd, the majority is  $(b + 1) / 2$  terms or more, while if 'b' is even the majority is  $b / 2$  terms or more. If no pixel value in the specific neighborhood satisfies the majority creation, the output of the function is 0.

The majority Coordinate Logic (MCL) filter implements the majority function at each level of the signal in the specific active neighborhood by selecting the 'n' bits having the majority in each of the 'n' binary images. The MCL filter results by checking all possible combinations of the 'b' pixels of the structuring elements at all binary elements at all 'n' binary levels, which has  $\lfloor (b+1)/2 \rfloor$  at a time, where  $\lfloor x \rfloor$  denoted integer part of the positive number 'x'. Thus, the MCL filter easily and quickly implements an approximation of the exact majority function applied on an active neighborhood containing 'b' pixel, which is suitable for additive noise removal.

**VI. RESULTS AND DISCUSSION**

**A. Image Noise Filtering**

The original and distorted bitmap images are stored in the input buffer initially. Bitmap of test images are used as the target images at different distortion levels for testing the performance. All the results are compared with the results obtained from a Gaussian filter which is shown in Figure 2. The Mean Difference, MSE in dB and PSNR in dB are computed for different images at different levels of variance. The results are tabulated in Tables 2. In each case, a Gaussian noise of mean 0 and variance 0.03 is assumed to corrupt the original image. It can be clearly perceived from the results shown that the proposed CLF filter gives an improved PSNR and produced a reconstructed image with enhanced sharpness bringing out the finer details in the machined surfaces. The degradation and blurring of edges, which accompanies magnification, has been removed the best in the CLF approach.



Figure 2(a) Original Chemical Plant image 128x128



Figure 2(b) Image Distorted by Gaussian Noise of Mean 0 and Variance 0.006



Figure 2(c) Image Filtered by Gaussian Filter



Figure 2(d) Image Filtered by CLF Filter

Table 1 Comparison of Mean Square Error (dB) for various standard test images

Image	Variance	Gaussian Filter	CLF Filter
Chemical Plant	0.006	22.44	21.08
Man	0.009	23.96	23.14
Moon	0.01	24.41	20.98

**B. Edge Detection with CLF Filter**

A circuit is evolved to detect the edges of the standard test image. The conventional filter output and the evolved circuit output are shown in Figures 3.

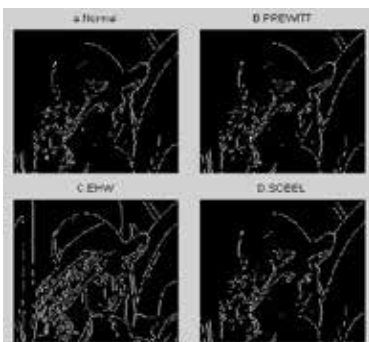


Figure 3 Edge Detection using CLF and other conventional Filters

**C.Resource Usage**

The logic resources used by the reconfigurable algorithm described in this work, is measured in terms of configurable logic block (CLB) usage. The time generation report for each of the individual operations and the resource utilization of the algorithm is shown in Table 2. It can be observed that the implemented algorithm fits within the resources of the device. The table includes the PCI bus access time. The total time was then divided by the length of the sequence to obtain time per bit. The results indicate that the proposed implementation outperforms the processor based implementation.

Table 2 Operations

Operation	Time Taken (ms)
Image Storage	0.409
Random Number generator (Population Generation)	0.1
Individual output evaluation	0.409
Single population	6.55
New population generation	0.093

**D. Implementation Results of CLF Filter**

In this section, the performance of CLF Filter in removing Gaussian noise is presented. The evolved architecture for this case is shown in Figure 4.

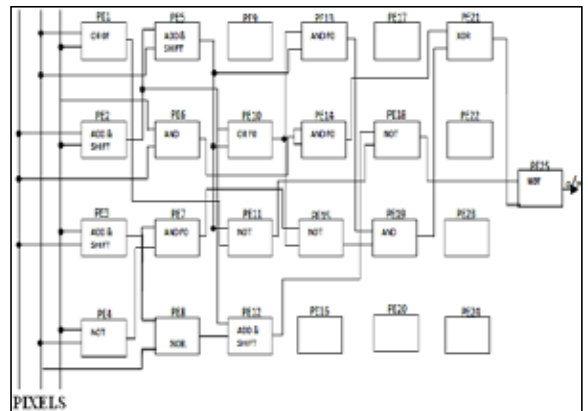


Figure 4 Architecture of VRC evolved to filter Gaussian Noise

The standard test image, Lena (Figure 5(a)) is used for study and results are shown. Additive Gaussian noise with mean 0 and standard deviation 0.05 is added (Figure 5(b)). The results obtained with Median filter and proposed CLF filter are shown in Figure 5(c) and 5(d) respectively. It is observed that Median Filter does not effectively remove Gaussian noise and performs only in edge regions, with blur effects still present in continuous regions. The CLF Filter performed well both in edge and continuous regions and effectively removes Gaussian noise.

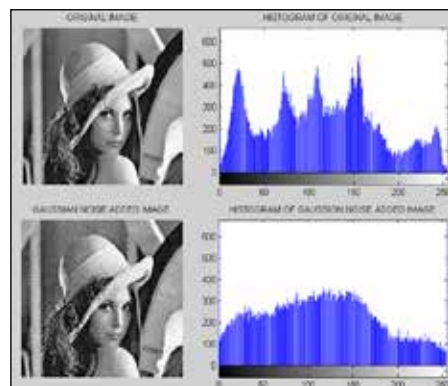
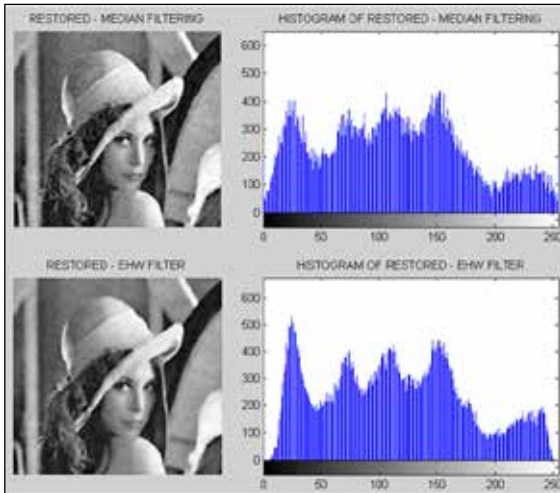
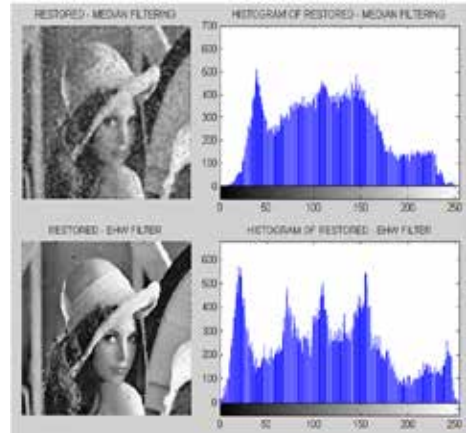


Figure 5(a) Original Image (b).Gaussian noise added image

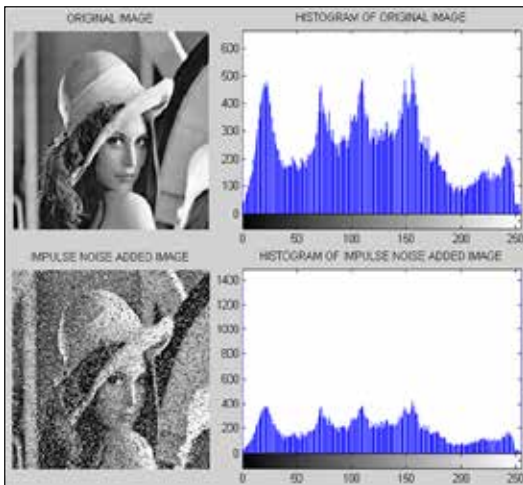


**Figure 5(c) Restored image using Median filter and (d) Restored image using CLF**  
 Similar improved results are obtained for other standard noise models such as multiplicative noise, quantization noise, impulsive noise and salt and pepper noise (Figure 6).



**Figure 6(c) Restored image using Median filter (d) Restored image using CLF**

It is cleared that EHW filter outperforms the other filters for both Gaussian and impulse noise which is shown in Table 3. The nonlinear part of CLF filter preserves the edges and removes the Gaussian noise effectively. The linear part smoothens the impulse noise and removes the spurious parts of the image.



**Figure 6(a).Original Image, (b) Salt and Pepper noise added image**

**Table 3 Comparison Study**

IMAGE	GAUSSIAN NOISE SD=0.05		SALT & PEPPER	
	MEDIAN FILTER	CLF	MEDIAN FILTER	CLF
PSNR	27.72	28.47	25.44	27.60
MDPP	188524	148652	177350	139831
MSE	24.76	20.87	25.35	20.37

The histogram value of median filtered histogram of original image. Whereas, the variation in pixel distribution in the case of CLF is very minimum and this establishes that CLF Filter is more suited for removing the noise.

**7. CONCLUSION**

In this paper EHW filter is used to remove evolve noise to produce better quality image than by using standard filters. As optimization module and the application module are independent to each other, optimized image can be processed. The optimization module is the most important and complex part in the system. In optimization module pipeline registers are inserted,all the modules are processed in parallel to speed up the optimization. From the above results, it is clear that EHW filter outperforms than other filters.

**REFERENCE**

[1] Lukas Sekanina, " Image Filter Design with Evolvable Hardware ", Brno University of Technology, Brno, 2002. | [2] Lukas Sekanina, Stepan Friedl, "AN EVOLVABLE COMBINATIONAL UNIT FOR FPGAS - DRAFT!!! ", Brno University of Technology, Brno, 7th July, 2005. | [3] Tomas Martinek, Lukas Sekanina, "An Evolvable Image Filter: Experimental Evaluation of a Complete Hardware Implementation in F P G A ", Brno University of Technology, Brno, 2005. | [4] A. Sumathi, Dr. R. S. D. Wahida Banu, "Digital Filter Design using Evolvable Hardware Chip for Image Enhancement ", International Journal of Computer Science and Network Security, May, 2006. | [5] Wei Zhang, Yuanxiang Li, Nian Liu, "An Online Evolvable Chebyshev Filter Based on Immune Genetic Algorithm ", Proceedings of the International Multi Conference of Engineers and Computer Scientists IMECS, Hong Kong, (19-21) March, 2008. | [6] Brenda Lin, Mai Irie, "Evolvable Hardware ", 2008. | [7] Uma Rajaram, Raja Paul Perinbam, Bharghava, "EHW Architecture for Design of FIR Filters for Adaptive Noise Cancellation ", Anna University, Chennai, India CVEST, International Institute of Information Technology, Hyderabad, India, January, 2009. | [8] Fan Yang, Michel Paidavoine, "A New Image Filtering Technique Combining a Wavelet Transform with a Linear Neural Network : Application to Face Recognition ", Universit' e de Bourgogne, France, The University of Texas at Dallas Program in Cognition, U.S.A., 2000. | [9] Jaime Ramirez-Angulo, " A Restrained Neighborhood Analog Median Filter For Image Processing Applications ", Alejandro Diaz-Sánchez National Institute for Astrophysics, Optics and Electronics, Tonantzintla, Puebla, Mexico, New Mexico State University. Las Cruces, New Mexico, USA. | [10] Alexandru Isar, Sorin Moga, Dorina Isar, "Image Denoising Using a Bishrink Filter with Reduced Sensitivity ", "Politehnica" University, Timisoara, Romania, Brest, France. |