Image enhancement using evolved reconfigurable filter cores

Mr.P.Sivakumar[1], **Dr.S.ravi**[2]

[1] Research scholar, St.Peter's University, Chennai.
[2] Professor and Head, Department of ECE, Dr.M.G.R.University, Chennai-95.

Abstract
Nonlinear image processing is presented here as a generalization of the operation by removing the linearity constraints. It seeks the optimum mapping implemented directly in logic. Given this generalization, the optimum nonlinear solution will be either better or equivalent to the linear solution, but it should not be worse. This inequality holds regardless of the problem or the criteria, provided that the training data is sufficient. The filter is an estimator; it uses the input values to estimate an unobserved quantity. By making simple assumptions about the image statistics, we can estimate the output value at a specific point by considering only a finite window of observations centred at that point. Image processing is a very computing intensive task, because several low level (pixel level) operations are performed over an image in order to execute a certain task, like edge detection, edge linking, noise removal, dilation erosion and filtering. One common alternative is the use of dedicated DSP processors; another is the use of an ASIC. Recently, another approach starts to be used: the reconfigurable systems. However, such systems yet need some evolution to be used: some development in the reprogram ability techniques and some improvement in their compilers.

Keywords:
Image processing, Evolvable Hardware, Median filters.

1. Introduction

A digital signal or image processing operation can be viewed at its most basic level as the manipulation of a series of finite-length binary strings. Whether the operation is implemented on a processor through software or in dedicated hardware, the data and the algorithms are invariably mapped through electronic logic components, which are inherently binary in nature. In nonlinear image and signal processing, the design of operators is carried out by seeking the optimum mapping from one set of binary strings to another. This contrasts with the linear approach which formulates a solution by optimizing coefficients within a generalized multiply – accumulate context. The situation is much more complex for nonlinear systems. The task is to seek the optimal logical mapping from all possible mappings. No simple superposition properties exist, and in the most general unconstrained design case, every combination of input variables must be observed a sufficient number of times in order to estimate the conditional probabilities of the output. The linear solution should be viewed as a special case of the set of all logic-based solutions rather than as an alternative.

2. Nonlinear filters and its functional realisation

The work in this area has focused on the design of filters. Many applications are possible within this context such as noise reduction, shape, character and object recognition, enhancement, restoration, texture classification, spatial and intensity sampling and rate conversion. In order to be able to design the filter from a realistically sized training set, further constraints must be applied to the filter. For binary values, the output becomes a logical function of the input variables. If the window contains n points, there are 2nd combinations of input variables for which the relevant output must be estimated. Therefore, there are 22n possible functions and it is the objective of the design process to determine which one of these functions corresponds to the optimum. Among the 22n functions that may be applied within an n point window, there will be many subclasses of functions. We may decide to restrict the choice to a filter that is idempotent or increasing. Idempotence implies that the filter has only a one-off effect on the image such that repeated application of the filter leads to no further modification of the image. Increasing implies that the filter preserves signal ordering. It can be shown that increasing filters map to logical functions that contain no complementation of the input variables. This drastically reduces the size of the training set required and therefore makes filter design easier. This can be explained in terms of logic (since a much smaller set of functions is under consideration) or in terms of statistical estimation (since now a single training example may be used to infer information about other combinations of input variables).

3. Image processing algorithms

This section discusses the theory (in brief) of most commonly used image processing algorithms (Filtering, Morphological Operations, Convolution and Edge

Manuscript received February 5, 2012
Manuscript revised February 20, 2012

Detection) and conventional image enhancement techniques.

3.1. Filtering

3.1.1. Algorithm
A standard median operation is implemented by
Step 1 Sliding a window of odd size (eg., 3x3 window) over a chosen image is chosen.
Step 2 Sampling the values of images at each window positions and sorting them.
Step 3 The median value of samples is given as the output.
Step 4 The sample in the centre of the window is replaced with median value.

3.1.1.1Requirements
The main requirements are the computational cost (should be low) and increased time complexity (should be low).

3.1.2. Morphological Operations
The term morphological image processing refers to a class of algorithms that can extract the geometric structure of an image. Morphology can be used on binary and gray scale images and is useful in many areas of image processing, such as skeletonization, edge detection, restoration and texture analysis. A morphological operator uses a structuring element to process an image.

3.1.2.1. Algorithm
Step 1 A Scanning Window is chosen. [Can be of any size, but 3x3 and 5x5 sizes are common].
Step 2 Image is scanned over a window to extract a structural element.
The most basic building blocks for many morphological operators are erosion and dilation.

3.1.2.2. Coordinate Logic Dilation (CLD)
CLD of the images G by the structuring elements B is denoted by
$G\frac{D}{B}(g(i,j))$,or $G\frac{D}{B}$,as in equation[1]
F= $G\frac{D}{B}$=COR g(i,j)ε b
$= \sum_{k=0}^{n-1} (SK(i,j))\frac{D}{B}$ 2k i=1,2.....M, j=1,2.....N--------(1)
Where $(SK(i,j))\frac{D}{B}$ denotes the dilation operations on the binary value SK(i,j) by the structuring elements B, given by $(SK(i,j))\frac{D}{B}$ =OR (SK(i,j) ε B .

3.1.2.3. Coordinate Logic Erosion (CLE)
CLE of the image G by the structuring elements B is denoted by $G\frac{D}{B}$ and is given in eqn. [2]

F= G $\frac{E}{B}$ CAND g(i,j) ε B = $\sum_{k=0}^{n-1} (SK(i,j))\frac{E}{B}$ 2k i=1,2.....M, j=1,2.....N------(2)
Where $(SK(i,j))\frac{E}{B}$ denotes the erosion operation on the binary values by the structuring elements B, given by $(SK(i,j))\frac{E}{B}$ = AND (SK(i,j)) ε B.

3.1.3. Convolution Operation
Convolution belongs to a class of algorithms called spatial filters. Spatial filters use a wide variety of masks (Kernels) to calculate different results, depending on the desired function. Mathematically, convolution on image can be
$f(x) = \sum_{i=0}^{\text{Heigt of the image}} \sum_{j=0}^{\text{Width of the image}} h(i,j)x(m-i,n-j)$---------- (3)
represented by equation (3).
Where x is the input image, h is the filter and y is the output image

3.1.3.1. Algorithm
Step 1 A window of some finite size is scanned over an image.
Step 2 The output pixel value is the weighted sum of the input pixels within the window where the weights are the values of the filter assigned to every pixel of the window. The window with its weights is called the convolution mask.

3.1.4. Edge Detection
Edge detection consists of four major stages namely:
1. Image Smoothing
2. Vertical and Horizontal Gradient Calculation
3. Directional Non Maximum Suppression
4. Threshold

3.1.4.1. Algorithm
Step 1 First the image is smoothed by Gaussian convolution [A simple 2-D first derivative operator is applied].
Step 2 Edges translate into ridges in the gradient magnitude image.
Step 3 Then tracking is done along the top of these ridges using the following procedure.
Step 4 i) The tracking process exhibits hysteresis controlled by two thresholds: T1 and T2.
[This hysteresis helps to remove the edge fragments].
ii) With T1>T2 tracking can only begin at a point on a ridge higher than T1. Tracking then continues in both directions from that point until the height of the ridge falls below T2.
Step 5 Assign zero to all pixels that are not actually on the ridge.

3.1.4.2. Edge Extraction using CL filters

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

$f(i,j) = g(i,j)CAND[CNOT[g(i-1,j)CANDg(i = 1,jCANDgi,j+1CANDgi,j-1$ -------- (4)

4. Evolved logic functions

The function to be performed by each PE is selected from a set of evolved operators such that the evolved circuit is inherently testable without the need for a specialized data path. A total of 16 functions are selected in this work and this is given in Table 1. Another novel feature of this work is that using the primary inputs and outputs alone, the evolved circuit can be tested. As a prerequisite to this, it is assumed that none of the inputs of elements can be connected to the same data source.

Table 1Evolved Image Processing Operators

Function code	***Function description***	***Type of Notation***
0000	X>>1	Right shift 8-bit pixel by one position
0001	X	Buffer
0010	~X	Logic
0011	X & Y	Logic
0100	X|Y	Logic
0101	X^Y	Logic
0110	(X+Y)>>2	Right shift 8-bit pixel by two position
0111	(X+Y)>>1	Right shift 8-bit pixel by two position
1000	X & 0x0F	Masking
1001	X & 0xF0	Masking
1010	X |0x0F	
1011	X |0xF0	
1100	Min(X,Y)	Threshold
1101	Max(X,Y)	Threshold
1110	Y<<1	Left shift 8-bit pixel by one position
1111	X+Y	Spatial/Spectral

5. Hardware architecture

The VRC consists of 25 PEs as shown in Figure 1. Four PEs are implemented as a single stage of the pipeline. Each PE can process two 8-bit inputs and produce a single 8-bit output. The outputs of PEs are equipped with registers. The two inputs of every PE can be connected to one of the outputs from the previous l columns where l is the level back parameter.

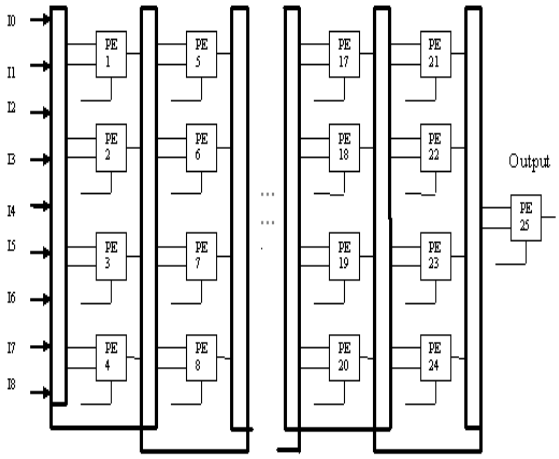

Figure 1 Architecture of Virtual reconfigurable circuit

6. Noise filter process

Starting with the noise corrupted image In, the objective is to find a filter ψ to recover image Io. In practice, this may not be possible. The design task therefore reduces to finding the optimum filter ψ pt out of all possible filters ψ that minimizes the difference between the filtered noisy image ψ (In) and the original IO. In the language of statistics, an optimal estimator is being sought. Its task is to estimate the true value of the image pixels from a noise-corrupted version.

6.1. Error criterion

Given two images $I_1(r, c)$ and $I_2(r, c)$ with the same number of R rows and C columns, their MAE is defined as

$$MAE(I1, I2) = \frac{1}{RC} \sum_{c=0}^{C-1} \sum_{r=0}^{r-1} |I1(r, c)-I2(r, c)|$$

The optimum filter is therefore defined as the one that minimizes the difference between the ideal image I_O and the filtered version of the noisy image ψ (I_n),MAE (ψ (I_n), I_O).

An error occurs only at those locations where the filter output and the ideal image differ. For each location where this occurs, the contribution to the total MAE is precisely one pixel.

7. Results and discussion

The original and distorted bitmap image (boat.bmp) is stored in the input buffer initially. Bitmap of test image is used as target image at different distortion levels for testing the performance. Additive Gaussian noise with

mean 0 and standard deviation 0.05 is added (Figure2.1b). The results obtained with Median filter and proposed EHW filter are shown in Figure 2.1.c and 2.1.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 EHW Filter performed well both in edge and continuous regions, while effectively removing Gaussian noise. Similar improved results are obtained for other standard noise models such as salt and pepper noise.

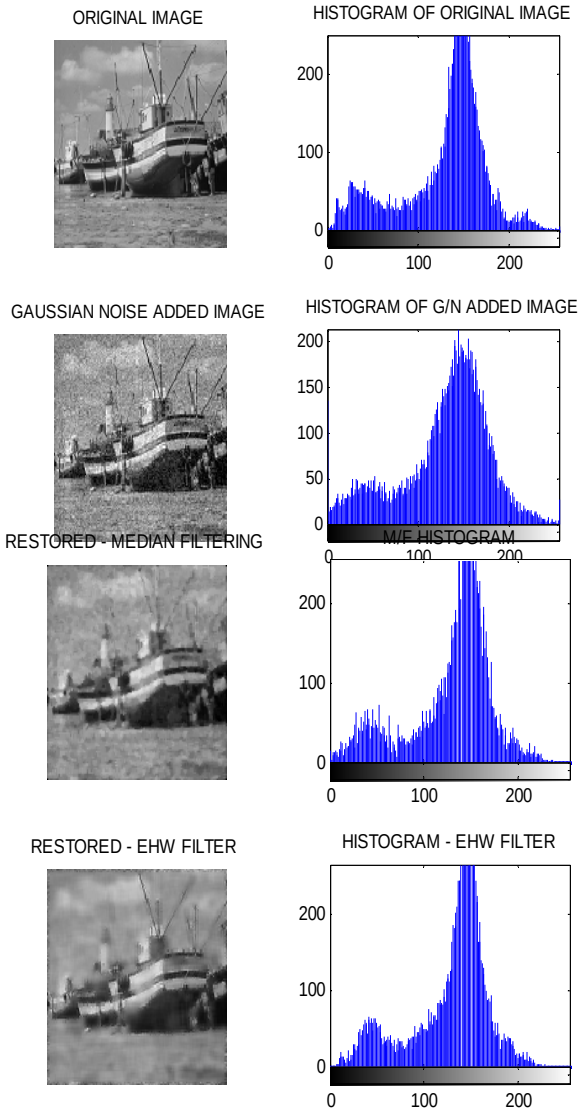

Figure2.1. a.Original Image, b.Gaussian noise added image c.Restored image using Median filter d. Restored image using EHW

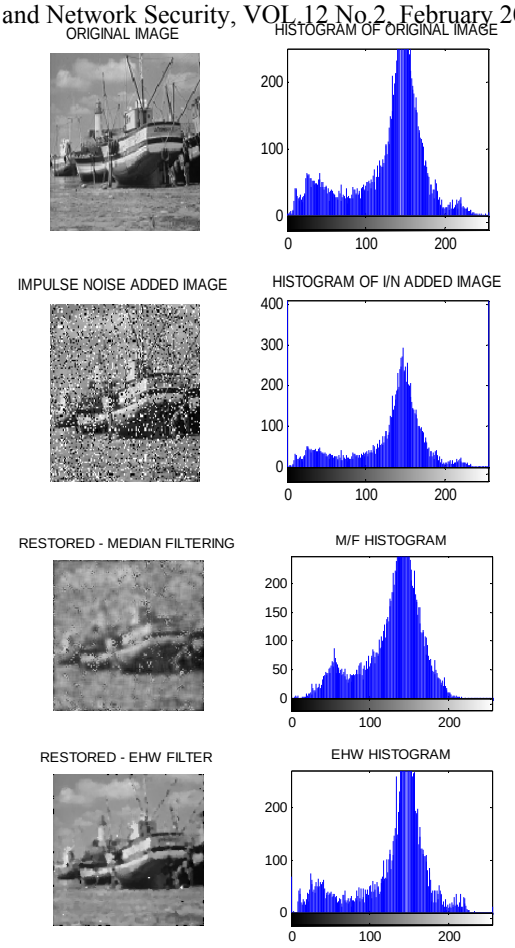

Figure2.2 a.Original Image, b.Salt and Pepper noise added image c.Restored image using Median filter d. Restored image using EHW

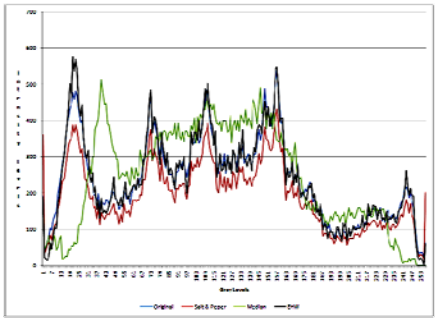

Figure 2.3 Histogram of images before and after Gaussian noise removal using different filters

From the results given in table 2, it is clear that EHW filter outperforms the other filters for both Gaussian and impulse noise. The nonlinear part of EHW 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.

The performance of the proposed EHW filter is also evaluated using histogram analysis and is shown in Figure 2.3. The histogram value of median filtered image

has a wide variation with respect to the histogram of original image. But, the variation in pixel distribution in the case of EHW is very minimum and this establishes that EHW Filter is more suited for removing the noise.

Table2 Comparison study

`	Noise added	PSNR (dB)	MSE
Median Filter	Gaussian with S.D=O.05	32.7	3.47
EHW		50.40	0.593
Median Filter	Salt & Pepper	24.17	6.32
EHW		30.56	2.30

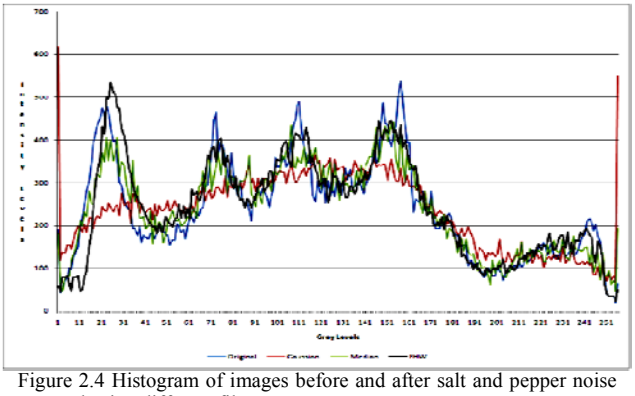

Figure 2.4 Histogram of images before and after salt and pepper noise removal using different filters

8. Conclusion

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. 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 images. The degradation and blurring of edges, which accompanies magnification, has been removed the best in the CLF approach. The system dynamically allocates the circuit function during the operation, for the time-sharing using of hardware and saving hardware resources. The future direction of research shall focus on fast algorithm implementation, high efficiency hardware realization and its testability.

References

[1] Paul Layzell, "Reducing Hardware Evolution's Dependency on FPGAs", Proc. of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems (MicroNeuro '99), IEEE 1999, pp. 171-178.

[2] L.Sekanina,"Image Filter Design with Evolvable Hardware", Applications of Evolutionary Computing Proc.of the 4th Workshop on Evolutionary Computation in Image Analysis and Signal Processing (EvoIASP'02), Springer- Verlag, Berlin, 2002, pp. 255-266.

[3] L. sekanina, "Virtual Reconfigurable Circuits for Real-World Applications of Evolvable Hardware", Evolvable Systems: From Biology to Hardware: Fifth International Conference, ICES 2003, pp. 186-198.

[4] Samhouri etal (2005),"Surface Roughness in Grinding: Off-line Identification with an Adaptive Neuro Fuzzy Inference system", Paper submitted to NAMRAC 33-2005 conference, May 24-27, Columbia.

[5] Biswajit Mishra and Peter Wilson, "Color edge detection hardware based on geometric algebra", 3rd European Conference on Visual Media Production, 2006. (CVMP 2006), pp. 115-121.

[6] Xilinx Inc. "Virtex–4 user guide," UG070,version2.2,http://www.xilinx.com/bvdocs/userguides/ug070.pdf, April 2007.

[7] Nader I. Rafla, "Evolvable Reconfigurable hardware framework for edge detection", IEEE 2007.

[8] A.Guruva Reddy, Dr.M.N.Giri Prasad, Dr.k.sri Rama Krishna, "Reconfigurable circuit design using evolvable hardware chip for illumination tolerant image enhancement", IEEE 2008.

[9] Tripti Jain, Prashant Bansod, C. B. Singh Kushwah and Mayenk Mewara, "Reconfigurable Hardware for Median Filtering for Image Processing Applications", IEEE 2010.

[10] Dr.V.Nagarajan, "Efficient Run-time Task Allocation in Reconfigurable Multiprocessor System-On-Chip with Network-On-Chip", IEEE 2011.

P.SIVAKUMAR received DCT, B.E. (ECE) & M.Tech. (Applied Electronics) degree in the year 2002, 2005 & 2007 respectively. He is having 7 Years of teaching experience & presently working in Kingston Engineering College, Katpadi. His area of interests includes digital image processing & digital signal processing. He is a member of ISTE, IAENG.

Dr.S.RAVI received DEEE, A.M.I.E (ECE), and M.E (CS) degree in the year 1989, 1991 and 1994 respectively and the Doctorate degree from Anna University, Chennai in the year 2004. He is having 18 years of teaching experience in engineering college. His areas of interests are VLSI, Embedded Systems, Image Processing and Simulation and Modelling. He has published more than 25 papers in International / National Conferences and Journals. He is a member of IEEE and ISTE and Life member of IETE and IETUK.