

Edge Detection Based on Extrinsic Evolvable Hardware

M. Hosseini and S. Mirzakuchaki

Abstract—Edge detection, the process of locating the edge of an image, is a basic and important subject in computer vision and image processing. There are different techniques for detecting the edges of images among which the evolvable hardware has drawn the attention of researchers for the case in the recent years. This paper is to describe the application of Evolvable Hardware (EHW) to detect the edges based on Sobel operator, the primary edge detection method. In this regard, the smoothing filter is applied to the images which are contaminated by Gaussian noise with the mean of 0 and different variance levels. Evolutionary design of edge detector performs Evolution of spatial masks and threshold. EHW filter is used to conduct the purpose of this paper and the relevant obtained results are compared with the ones gained using only Sobel operator which has constant threshold and spatial masks. According to the results, the edge detection is improved because of using EHW filter.

Index Terms—Edge detection, Evolvable hardware, Genetic algorithm, Sobel operator.

I. INTRODUCTION

Evolvable Hardware (EHW) [1] refers to self-reconfiguration hardware design, where the configuration is under control of an evolutionary algorithm (EA) [2]. In other words, evolvable hardware is the combination of evolutionary algorithms and the software reconfigurable devices. Recently, EHW technique has attracted increasingly the researchers' attentions and offered them the promise to be employed as an alternative to conventional specification-based electronic circuit design method.

EHW was first used by Hugo de Garis in the early 90s. It was proposed as an opponent to conventional methods for hardware design. Since then, and especially after introducing the Programmable Logic Devices such as FPGAs, EHW has been investigated thoroughly by different researchers, Adrian Thompson [3], Julian Miller [4] and Higuchi [5] are just a few names to be mentioned. This technique appears to be successful and promising, as it could automatically design digital circuits by using Evolutionary Algorithms. However, researchers in this field are still facing some real challenges such as scalability and generalisation issues, etc [6], [7].

Manuscript received August 29, 2010

Maryam Hosseini is with the Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran. (maryam.hosseini79@gmail.com)

Sattar Mirzakuchaki is with the Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran. (m_kuchaki@iust.ac.ir)

Regarding conventional hardware, the structure is irreversibly fixed in the design process; however, evolvable hardware is designed in such a way to be adapted to the changes taking place in the task requirements or in the environment. That is this device has the ability to reconfigure its own hardware structure dynamically and autonomously.

A recent application of EHW is in computer vision and image processing. Edge detection is among many algorithms used in the mentioned areas. Detection of edges in an image is a very important step towards understanding image features. Edges have meaningful features and can offer significant information.

Many efforts have been concentrated on finding better edge detection algorithms and also their implementations. In this regard evolutionary algorithms have been used as one of the effective devices. This application is seen in a broad and disparate range particularly in the field of digital image processing, seen in the literature. The two notable examples, mentioned here, confirm the above efforts: "Edge Detection of Image Based on Improved Sobel Operator and Genetic Algorithms" [8]; "Performance analysis of FPGA based sobel edge detection operator" [9].

Recently, like increasing the application of evolutionary algorithm, EHW has simultaneously been in the center of attention and used increasingly in signal and image processing. Here, several efforts, found in the literature are listed as follows:

"Evolvable Reconfigurable Hardware Framework for Edge Detection" [10]; "Digital image filter design using evolvable hardware" [11]; "Digital circuit design using intrinsic evolvable hardware" [12].

In this paper, the EHW and Sobel operator are used at the same time for detecting the edges of images. Here, Sobel edge detection operator is selected as it incorporates both edge detection and smoothing operator to provide good edge detection capability in the noisy conditions.

The present paper is organized as follows:

- section 1: Introduction;
- section 2: Evolutionary and genetic algorithms;
- section 3: EHW;
- section 4: Edge detection and Sobel operator;
- section 5: Proposed evolvable hardware filter;
- section 6: Results;
- section 7: Conclusion.

II. EVOLUTIONARY ALGORITHM

Evolutionary algorithms are stochastic search algorithms inspired by Darwin's theory of evolution (Bäck, 1996; Holland, 1975). These algorithms operate with the

population of candidate solutions (individuals) other than working with one solution at a time. Each new population is formed by the operators inspired genetically (like crossover and mutation) and through a selection pressure, which guides the evolution towards better areas of the search space. This selection is taken place in the evolutionary algorithms by evaluating the candidate solution through its fitness which is calculated by the fitness function. In other words, the EA progression quality is verified by this function.

The EAs are used in different areas, of which some are applied in the engineering optimization problems, such as, neural networks, machine learning and hardware design.

Here, the four known EAs are mentioned: Genetic Algorithms (GAs) [13], Genetic Programming (GP) [14], Evolutionary Strategy (ES) [15] and Evolutionary Programming [16]. The first two ones are of these four widely used in the evolution of digital circuits.

A. Genetic algorithms

GAs are a subset of evolutionary algorithms, presented by an American scientist named John Holland in the 1960's [14]. Such algorithms, known as the most powerful search procedures, determine how the hardware structure should be reconfigured whenever a new hardware structure is needed for a better performance. The canonical GA has a population of chromosomes, each of them is obtained by encoding a point in the search space. Usually, they are represented by the strings of binary or the vectors of double characters.

The performances of GAs are significantly affected by genetic operators due to mimicking the process of heredity in creating new offspring at each generation.

The main genetic operator is crossover which may taken place in three forms: single-point, double-point and uniform [17]. Crossover operates on two parents (chromosomes) at the same time and combine their features to generate new offspring, see Fig. 1.a.

Another operator for GAs, the important and complicated one, is mutation. In this process the genes of a selected individual are changed. The number of individuals that should suffer from mutation is called mutation rate and the number of genes to be altered within an individual is mutation degree. Several mutation operators such as swap, inversion, insertion, etc, have been proposed for permutation representation, [18], [19]. However, a simple way to achieve mutation would be to alter one or more genes, see Fig. 1.b.

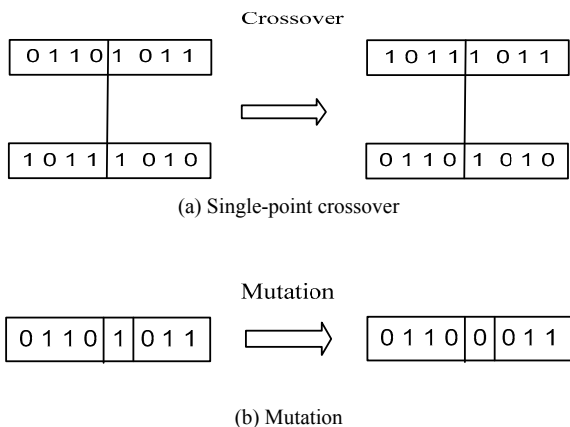


Fig. 1. Genetic algorithm operators

As mentioned earlier, selection is another phase in the evolutionary algorithms. This selection intended to improve the quality of population by giving a better chance to the high-quality chromosomes, having higher fitness, to get copied into the next generation. In the selection process the focus is on exploring the promising regions in the solution space.

The mechanisms of GAs, adapted to all Evolutionary algorithms, are represented schematically in Fig. 2.

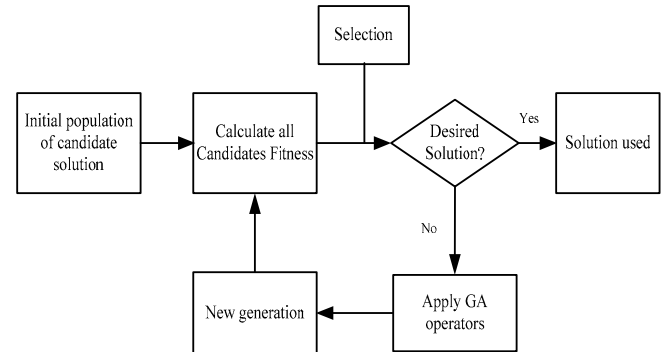


Fig. 2. Schematic diagram of GAs

In GAs, first all the chromosomes are randomly initialized; then the fitness value of each individual is computed and finally, the fittest individual is selected. The evolution process stops where the fitness value of a chromosome is 100% or the number of generations has reached its maximum, set by the user for the particular experiment. In case of not meeting the above conditions, a new population will be generated by crossovering and mutating the best chromosomes (selected at third step). In the next cycle, all new created chromosomes are evaluated and operated in the same way. The evolution process continues until the fitness value of chromosome is 100% or the maximum number of generations, set by the user, is obtained.

The main advantages of GAs are its robustness and superior search performance in different type of problems without needing any prior knowledge. Implementing GAs in software has time constraint due to the fitness evaluation. In this regard the evaluation can be executed more quickly by certain hardware device. Therefore, GAs can be used more effectively as its most serious problem is solved in this way. Evolvable Hardware is based on the above concept and uses the robust capability of GAs by reducing its computational cost.

III. EVOLVABLE HARDWARE

Evolvable hardware is a technique to automatically design circuits, where the circuit configuration is under the control of evolution algorithms.

Evolvable hardware has four major steps: 1) Specify coding schemes; 2) Determine the target properties and fitness evaluation mode of circuit; 3) Select evolutionary algorithm and arithmetic operators; 4) Implement of hardware evolution.

The hardware evolution process can generally be done in several ways. In the case of extrinsic evolution [20], [21], both the circuit evolution and fitness evaluation are done by software and only the final correct circuit is implemented on the reconfigurable hardware. In the intrinsic evolution [2],

[21], on the other hand, each evolved circuits are downloaded onto the hardware and the fitness is evaluated on the device itself, as shown in Fig. 3. In the third technique, complete hardware evolution, the evolutionary algorithm is implemented in the same hardware [22].

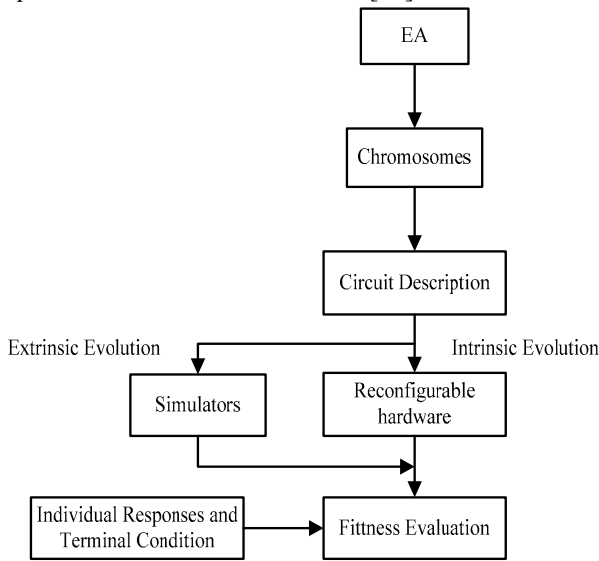


Fig. 3. Schematic diagram of EHW

The evolution can be taken place in the gate level [23] or function level [24], shown in Fig. 4 and 5, respectively. In the gate level evolution, the hardware evolution is based on the primitive gates such as AND gates and OR gates. The circuits, generated by gate level evolution, is not very large because the mentioned above EA execution takes a long time to evolve large circuits. Therefore, difficult to use gate level EHW to produce hardware functions which are useful for practical applications. In the function level evolution, hardware synthesis involves higher level hardware functions in comparison with the primitive gates of gate level evolution. In opposing to the gate level evolution, in the function level evolution, more useful hardware functions can be synthesized and larger hardware circuits can be designed.

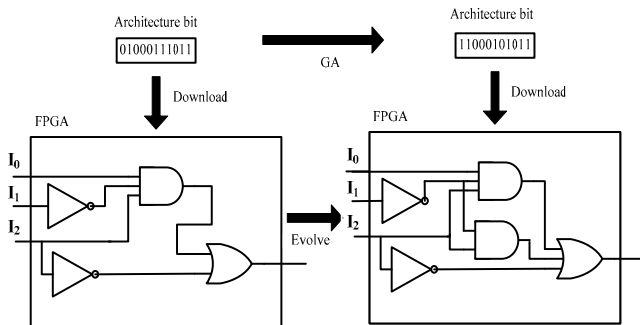


Fig. 4. Gate level evolution

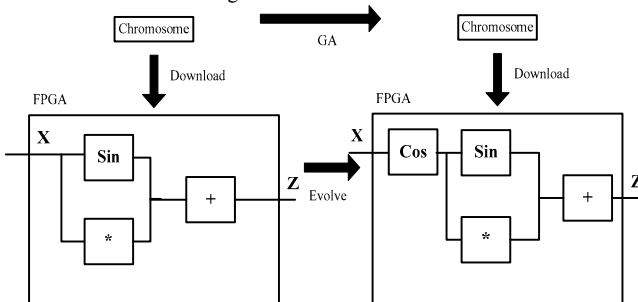


Fig. 5. Function level evolution

IV. EDGE DETECTION

A very important goal of computer image analysis and processing is to generate particular images that are more appropriate, concerning people or machines, for observation or identification. Edge detection operation is an essential part in the field of image processing. Edges are considered as the most important specification on an image providing valuable information for human image perception. While human eyes can detect the edges easily, performing this task is difficult for machines. This process becomes much more complex where the images are contaminated by different noise levels. A number of operators are defined to solve the problem of edge detection which may behave properly in one application but poorly in the other.

The primary edge detection methods such as Prewitt and Sobel operators [25], [26], used local operators to approximately compute the first derivative of gray level gradient of an image in the spatial domain. The position of local maximum of the first derivative is considered as the edge points. The Laplacian of Gaussian (LOG) operator for edge detection has been proposed by Marr and Hildreth [27]. In this method a Gaussian function is used for image smoothing, then the second derivative is computed and the edge points are displayed using zero crossing points. Alternatively, in Canny operator method an optimal edge detector is used for 2 dimensional images. This operator can give both intensity and direction information of the edge [28].

Recently, the implementation of edge detection algorithms has been on the table due to increasing the speed of responding, applicability, reducing the circuit size, etc. Evolvable hardware, the main subject of this paper, is one of the methods used in increasing the capability of some algorithms in confronting different images.

V. SOBEL EDGE DETECTION OPERATOR

The Sobel operator is widely used in image processing, particularly within edge detection algorithms, which is a differential based approach to edge detection. All gradient based algorithms have masks that calculate the slopes' strengths in the directions (ordinarily vertical and horizontal) which are orthogonal to each other. Then, the contributions of different components of the slopes are combined to give the total value of edge strength. The final step of this operation is to filter out the edge pixels from non edge pixels through comparing the value of edge strength with a predetermined threshold. This operation is shown in Fig.6.

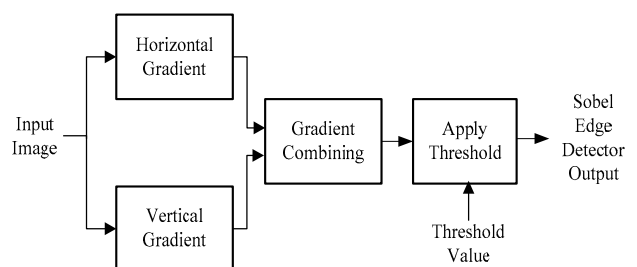


Fig. 6. Sobel edge detector

In the following, the equations of Sobel edge detection operator are expressed.

The templates (masks) of Sobel operator are shown in Fig.7.



Fig. 7. Sobel operator

The pixel number in a 3×3 image region is as follows:

$$\begin{matrix} f(x-1, y-1) & f(x-1, y) & f(x-1, y+1) \\ f(x, y-1) & f(x, y) & f(x, y+1) \\ f(x+1, y-1) & f(x+1, y) & f(x+1, y+1) \end{matrix}$$

Fig. 8. 3×3 sub-domain of image

where, x and y are the coordinates of pixel of the image and $f(x, y)$ is the gray level of pixel in the image.

Horizontal gradient and vertical gradient, at the point (x, y) in the image is obtained by convolving the sub-domain of image shown in Fig. 8 with Sobel operator is presented in Fig.7. In convolving, the corresponding elements are multiplied and then the results are added altogether. The relevant formulas are as follows:

$$G_x = [f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)] \quad (1)$$

$$G_y = [f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)] \quad (2)$$

where, G_x is horizontal gradient; G_y is vertical gradient.

The gradient of each pixel is calculated by:

$$G(x, y) = |G_x| + |G_y| \quad (3)$$

Therefore, the threshold value (t) is selected and $G(x, y) > t$, will be the edge point.

Sobel operator is used as it is not sensitive to noise and has relatively small masks in comparison with other operators like Robert one. When Sobel operator is implemented in the evolutionary hardware, the search process is done quickly and efficiently reducing the computation cost which is the major problem with its software-based counterpart.

VI. PROPOSED EVOLVABLE HARDWARE FILTER

In this paper extrinsic EHW is employed by Modelsim SE 6.2b (software simulator) and Matlab R2008a (GA solver). The evolutionary design of edge detector evolves the spatial masks and threshold. The experimental results in section VI reveal that this evolutionary design of edge detector

outperforms conventional approaches.

VII. EVOLUTIONARY ALGORITHM

In this research, GA, having elitism and fixed population, is employed with selection, crossover and mutation operators.

The elements of spatial masks and threshold are considered as the genes in the chromosomes. As mentioned in the previous paragraph, the spatial masks and threshold are evolved by the evolutionary design of edge detector.

Here the type of population is Double vector and the initial population (30 chromosomes) is generated randomly according to the coding rules. Then the fitness of initial population is evaluated and the parent chromosomes are selected using roulette wheel selection. The crossovers are then performed between selected parents at a rate of 85%. The next operation is mutation and carried out at a rate of 10%. In this phase a member of current population is randomly selected and replaced with the new one obtained by a random process. The elitism is of 3 Chromosomes with the highest fitness which are retained for the subsequent generation. These chromosomes and other 27 new ones (new Chromosomes are considered as Childs) provide the new population for next generation. The maximum number of generations is 30 which determines the stopping of GA solver.

VIII. FITNESS FUNCTION

The only evolutionary objective in our experiment is to minimize the difference between the detected edge of original image and that of noisy image. To measure the quality of detected edge of image, the output of EHW filter, the mean difference per pixel (MDPP) is used based on fitness function. The relevant formulas are presented as follows:

$$M = \left(\sum_{i=1}^{N-2} \sum_{j=1}^{N-2} \text{xor}(O(i, j), C(i, j)) \right) \quad (4)$$

$$\text{Fitness} = M / ((N-2) \times (N-2))$$

where, i and j are the coordinates of pixel; $O(i, j)$ is the gray level of pixel in the detected edge of original image; $C(i, j)$ is the gray level of pixel in the detected edge of noisy image; $N \times N$ is the size of original image; xor is a logical function for obtaining the differences between O and C pixels; M is the sum of above obtained differences for considered images.

It should be mentioned that the original image size is $N \times N$ ($N = 512$); however, only a sub-area image of $(N-2) \times (N-2)$ pixels is considered because the pixels at the image borders are untouchable for the 3×3 masks and remain unfiltered.

The fitness function shows the mean difference per pixel in two images. Therefore, the lower the fitness function value is, the lower the difference between two images will be, and accordingly the images are more similar.

IX. RESULTS

A. Simulation Results

In this research, the simulations are conducted by a 3GHZ Intel- based system with 4 GB memory and 6MB cache. The test image (bitmap of Lena) has 512x512 pixels with 256 gray levels and is distorted by Gaussian noise. This image is contaminated by Gaussian noise with mean 0 and variance of 0.02 and 0.04. Pre-processing step should be used to decrease the noise effect. Fig. 9.a is the original Lena bitmap of 512x512. The resulting edge image of Lena bitmap detected by Sobel operator, is presented in Fig. 9.b. Lena bitmap is distorted by Gaussian noise with mean 0 and variance 0.02, Fig. 10.a. Then this image is filtered by Gaussian, Fig. 10.b.. The edge image of noisy image detected by Sobel operator, is shown in Fig. 10.c. The edge image detected by the proposed filter, is presented in Fig. 10.d. The fitness is 0.04143 for the filtered image by Sobel operator and 0.03963 filtered image by evolvable hardware filter.

Lena bitmap is distorted by Gaussian noise with mean 0 and variance 0.04, Fig. 11.a. Then this image is filtered by Gaussian, Fig. 11.b. The edge image of noisy image detected by Sobel operator is shown in Fig. 11.c. The edge image detected by the proposed filter is presented in Fig. 11.d. The fitness is 0.04696 for the filtered image by Sobel operator and 0.04453 filtered image by evolvable hardware filter.

As mentioned above, the lower the fitness value is, the more similar the images are. In this research, the results obtained by EHW filter are of lower values in comparison with those by Sobel operator.



Fig. 9.(a) Original Lena image 512x512, (b) Edge image of Lena detected by Sobel operator.

B. Synthesis Results

In this paper, hardware evolution process is conducted in the form of extrinsic method. Therefore, the best structure of filter should be implemented on the hardware.

The inputs to filter are: nine eight-bit pixels values coming through nine bytes bus; one byte for threshold value; one byte for controlling operation. The Output is one bit for the detected edge pixel value and another one bit for output controlling.

In this research, the evolvable image filter is described in VHDL. After simulating the filters with Modelsim and terminating the GA, the configuration of best filter is synthesized by Xilinx ISE 6.3 for Virtex 5 xc5v1x30 FPGA. The results, obtained from the synthesis of evolvable image filter, are summarized in Table 1.

According to the synthesis reports, the maximum FPGA clock frequency, attained for the evolved image filter, is 67.895MHz.

TABLE 1. SYNTHESIS RESULTS OBTAINED IN VIRTEX5 XC5VLX30 FPGA

	Usage	Total	Percentage FPGA
Number of Slice Registers	422	19200	2%
Number of Slice LUTs	580	19200	3%
Number of fully used Bit Slices	391	14455	3%
Number of BUFG/BUFGCTRLs	4	32	13%

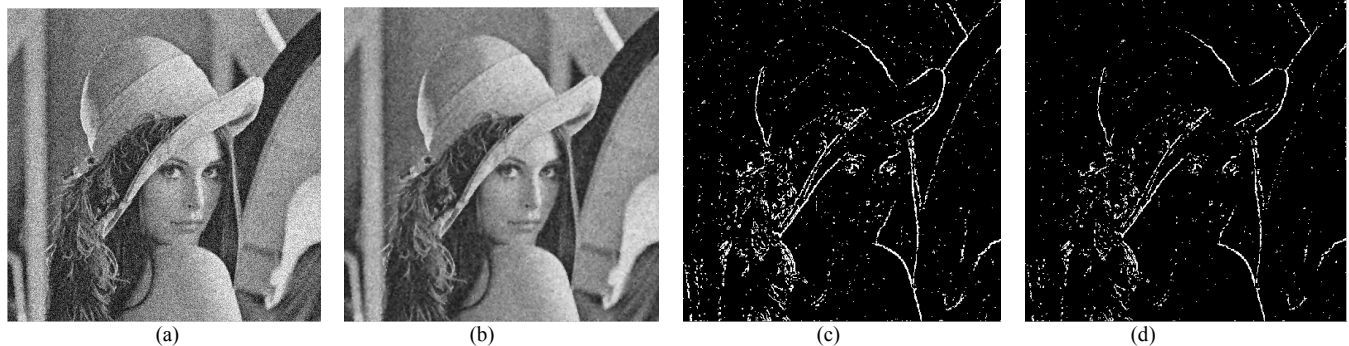


Fig. 10. (a) Image distorted by Gaussian noise of mean 0 and variance 0.02, (b) Image filtered by Gaussian filter, (c) Edge image of noisy image detected by Sobel operator, (d) Edge image of noisy image detected by EHW filter.

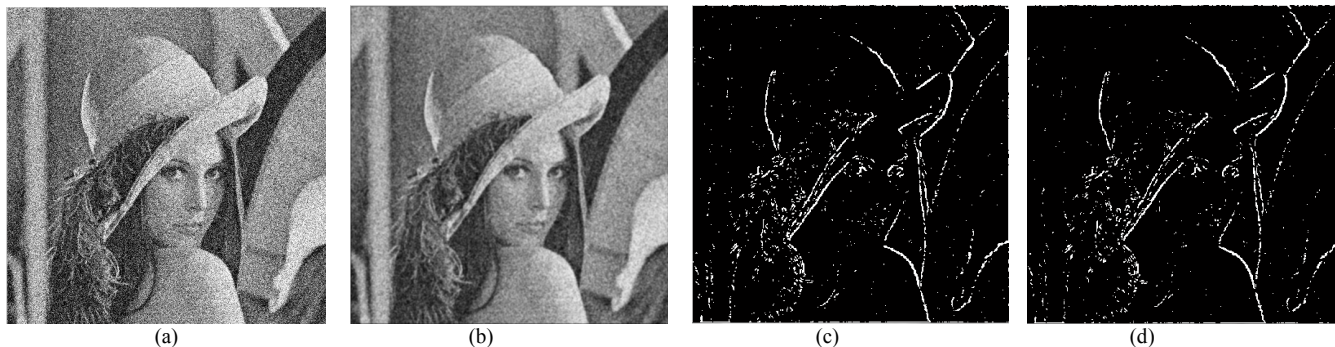


Fig. 11. (a) Image distorted by Gaussian noise of mean 0 and variance 0.04, (b) Image filtered by Gaussian filter, (c) Edge image of noisy image detected by Sobel operator, (d) Edge image of noisy image detected by EHW filter

X. CONCLUSION

In this research a compound of Sobel operator and EHW is used for detecting the edge of image. For this purpose, first the Sobel operator is applied by itself and then is mixed with EHW. Comparing the obtained images, it shows that the EHW has significant effect on the improvement of edge detection, explained in section [VI].

REFERENCES

- [1] X. Yao, T. Higuchi. "Promises and challenges of evolvable hardware", IEEE Trans. Systems, Man and Cybernetics, PartC, vol. 29, pp. 87 - 97, Feb. 1999.
- [2] Garrison W. Greenwood and Andrew M. Tyrrell, Introduction To Evolvable Hardware: A Practical Guide for Designing Self-Adapting Systems. Wiley-IEEE Press, November 2006.
- [3] A. Thompson, Silicon Evolution. In: J. R. Koza, et al. Proceedings of Genetic Programming, 1996 (GP96), pp. 444-452. MIT press, Cambridge, 1996.
- [4] J. F. Miller, P. Thompson, and T. C. Fogarty, Designing Electronics Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study, In Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, Ch. 6, Wiley Publisher, 1997.
- [5] T. Higuchi, et al. "Evolvable Hardware with Genetic Learning," In Simulation of Adaptive Behaviour, MIT Press, 1992.
- [6] E. Benkhelifa, A. Pipe, M. Nibouche, G. Dragffy, "Evolvable Hardware: a Problem of Generalisation. Which Works Best: Large Population Size and Small Number of Generations or vice versa,". NICSO, Italy. Nov 2007. Studies in Computational Intelligence, Springer Verlag.
- [7] Vesselin K. Vassilev and Julian F. Miller "Scalability problems of digital circuit evolution: Evolvability and efficient designs," In J. Lohn, A. Stoica, and D. Keymeulen, editors, The Second NASA/DoD workshop on Evolvable Hardware, pp. 55-64, Palo Alto, California, 13-15 July 2000, IEEE Computer Society.
- [8] Zhang Jin-Yu, Chen Yan and Huang Xian-Xiang, "Edge detection of images based on improved sobel operator and genetic algorithms," IEEE Trans. On Image Analysis and Signal Processing, 2009. IASP 2009. International Conference on, pp. 31-35.
- [9] I. Yasri, N. H. Hamid and V. V. Yap, "Performance analysis of FPGA based sobel edge detection operator," IEEE Trans. On Electronic Design, 2008. ICED 2008. International Conference on, pp. 1-4.
- [10] Nader I. Rafla, "Evolvable Reconfigurable Hardware Framework for Edge Detection," IEEE Trans. On Circuits and Systems, August 2007. MWCAS 2007. 50th Midwest Symposium on.
- [11] P. Nirmal Kumar, S. Suresh and Prof. J. Raja Paul Perinbam, "Digital image filter design using evolvable hardware", IEEE Trans. On Computer and Information Science, 2005. Fourth Annual ACIS International Conference on, pp. 483 - 488.
- [12] Yang Zhang, Stephan L. Smith and Andy M. Tyrrell, "Digital Circuit Design using Intrinsic Evolvable Hardware", Proc. of the 2004 NASA/DoD Conference on Evolution Hardware (EH'04), IEEE Computer Society, 2004.
- [13] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Kluwer Academic Publishers, Boston, MA, 1989.
- [14] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introduction Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, MA, 1992.
- [15] I. Rechenberg, Evolution Strategy. In: J. Zurada, , R. Marks, C. J. Robinson, Computational Intelligence Imitating Life, IEEE Press: Piscataway, NJ, 1992, pp. 147-159.
- [16] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural selection. MIT Press, Cambridge 1992.
- [17] J. F. Miller, P. Thompson, and T. C. Fogarty, Designing Electronics Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study, In Genetic Algorithms and Evolution Strategies in Engineering and Computer Science, pp. 105-131, Wiley Publisher, 1997.
- [18] L. Lin and M. Gen, "Bicriteria network design problem using interactive adaptive-weight GA and priority-based encoding method".
- [19] A. Kumar, M. Pathak and Y. P. Gupta, "Genetic-algorithm based reliability optimization for computer network expansion," IEEE Trans. On Reliability, Mar. 1995, Vol. 44, Issue 1, pp. 63-72.
- [20] T. Kalganova, R. Poli, W. Banzhaf, W. B. Langdon, J. Miller, P. Nordin, and T. C. Fogarty, Eds., "An extrinsic function-level evolvable hardware approach," in Proc. 3rd Eur. Conf. Genetic Program., 2000, vol. 1802, pp. 60-75.
- [21] L. Sekanina, T. Martinek, Z. Gajda, "Extrinsic and Intrinsic Evolution of Multifunctional Combinational Modules," IEEE Congo Evolutionary Computation, 16-21 July 2006, pp. 2771-2778.
- [22] Tuft G. and Haddow P., "Prototyping a GA Pipeline for Complete Hardware Evolution," Proceedings of the 1st NASA/DoD Workshop on EHW (EH'99), Pasadena, CA, USA, July 1999, pp. 18-25.
- [23] Higuchi et al. "Evolvable hardware: A First step towards building a Darwin machine," In 2nd International Conference on Simulated Behaviour, pp. 417-424. MIT Press, 1993.
- [24] T. Higuchi, M. Murakawa, M. Iwata, I. Kajitani, W. Liu, and M. Salami, "Evolvable hardware at function level," In Proc. IEEE Int. Conf. Evol. Comput., 1997, pp. 187-192.
- [25] A. Rosenfel, "Computer vision: a source of models for biological visual process," IEEE Transactions on Biomedical Engineering 36 (1), 1989, pp. 83-94.
- [26] I. Sobel, "Neighbourhood coding of binary images fast contour following and general array binary processing," Computer Graphics and Image Processing, Aug. 1978, Vol. 8, Issue 1, pp. 127-135.
- [27] D. Marr, E. C. Hildreth, "Theory of edge detection," Proceedings of the Royal Society, 1980 Vol. 207, No. 1167, pp. 187-217.
- [28] J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern analysis and Machine Intelligence 8 (6), 1986, pp. 679-698.