

FPGA Implementation of GA based ICA Image Fusion Method

A. M. El Ejaily
Libyan Armed Force

F. Eltohamy
Egyptian Armed Force

M. S. Hamid
Egyptian Armed Force

G. Ismail
Egyptian Armed Force

Abstract— Remote sensing image fusion is basically performed to enhance the image resolution for better image interpretation. Nowadays, remote sensing applications require quick system response especially for critical satellite missions such as hazard and disaster estimation. The on-board embedded processing has opened new areas for real time applications and reduced the bandwidth of the downlink. FPGA devices show a good solution for the implementation of image processing algorithms for image interpretation and decision making because of their reprogram ability, concurrency, on-chip embedded resources, and low cost. In this paper, FPGA architecture for the implementation of ICA using GA is presented. The design is used for the fusion of Quickbird satellite image data. The whole design is created using VHDL and verified through comparing the results with that obtained from Matlab tools. The evolved hardware shows similar performance compared to Matlab software. The design runs at a frequency of 36.7 MHz on Spartan-3A DSP 1800A Xilinx FPGA. Synthesis results demonstrate good utilization of device resources while maintaining high system clock.

Keywords— *Image fusion; multispectral image MS; Panchromatic image PAN; Independent Component Analysis ICA; FPGA ;Genetic Algorithm GA.*

I. INTRODUCTION

Fusion of panchromatic (Pan) images having high spatial and low spectral resolutions with multispectral (MS) images having low spatial and high spectral resolutions improves the quality of remotely sensed images for better classification and feature extraction and consequently increasing satellite image data usage [1]. The importance of image fusion in the field of remote sensing motivates researchers to develop image fusion techniques that enhance the quality of the fused images.

The most common form of image fusion methods are based on intensity-hue-saturation (IHS) [2], principal component analysis (PCA) [3], discrete wavelet transform (DWT) [4], complex wavelet transform (CWT) [5], and independent component analysis (ICA) [6]. ICA based image fusion has an advantage over other image fusion techniques because of its ability of using higher-order statistical properties in eliminating the redundancy between different image data. The estimation of the data model of independent component analysis is usually performed by formulating an objective function and then minimizing or maximizing it. Therefore, the properties of the ICA method depend on both the objective function and the optimization algorithm[7].

Authors of [8] used genetic algorithm (GA) [9-11] for the optimization of the ICA by maximizing non-gaussianity through the maximization of kurtosis of the obtained components. Due to the simplicity of the calculation of the kurtosis function, this paper proposes a hardware implementation of the GA based ICA in [8] using Spartan-3A DSP 1800A Xilinx FPGA. The FPGA consists of programmable logic cells and interconnections that can be programmed by the user to perform a desired function[12,13]. The proposed system is described in VHDL and simulated using Xilinx ISE design suite 12.1. FPGAs maintain the performance of ASICs (application specific ICs) while avoiding their high development cost and inability to accommodate design modifications after production. The proposed hardware system is used to search the optimum mixing matrix which is used to decompose the MS image of the Quickbird satellite image data into its independent components. The obtained components are then used to fuse the MS and PAN images.

The advantages of the hardware implementation is to reduce the inherent long time convergence of the software implementation of the GA. moreover the hardware implementation opens the way for the creation of embedded systems to perform image fusion tasks in real time that serve to interpret satellite images and making decision while reducing the bandwidth of the downlink. This can be accomplished using soft and hard processor cores [13].

II. FPGA IMPLEMENTATION OF THE ICA

A. Selecting a Template (Heading 2)

ICA is a computational method for separating multivariate signals into additive components supposing the mutual statistical independence of the non-Gaussian source signals [7]. The components should be statistically independent. This means that the value of any one of the components gives no information about the values of the other components. If a data vector $X = (X_1, X_2, \dots, X_N)^T$ is an observed vector such

$$S = AX \quad (1)$$

where A is an unknown scalar matrix which is called mixing matrix.

The goal of the ICA is to find the right linear transformation of the correlative input that makes the outputs as independent as possible.

Evolutionary algorithms are suitable for finding [9-11] solutions of large searching space and multiple conflicting objectives. GA is a subarea of the more general topic of evolutionary algorithms. GA uses evolution as an optimization tool for engineering problems in a manner similar to the biological evolution. In the natural world there is a population of organisms, the life cycle of these organisms represent the fitness function, so the nature itself will select the best organism for the next generation. GA evolves a population of candidate solutions to a given problem, using operators inspired by natural genetic variation and natural selection.

The function of the genetic algorithm is to search for the desired solution of engineering problem among a collection of candidate solutions. The set of candidate solutions represent the population of organism, each solution represent a chromosome. To evolve good solutions and to implement natural selection, we need a measure with a set of parameters involved for selecting good solutions from available solutions. This measure is so-called fitness function and the parameters are encoded to vectors, each vector represents a chromosome (possible solution). Once the problem is encoded in a chromosomal manner and a fitness measure has been chosen, GA starts to evolve the chromosomes of the search problem using a population of initial solutions through crossover and mutation operations until a predefined stopping criterion is reached.

In this paper, the candidate solution is used to generate the independent components of satellite image data according to equation (1) and the GA is used to evolve the best solution (mixing matrix) by maximizing the fitness function. The fitness of the candidate solution is calculated by maximizing the kurtosis [ica book] of the independent components according to the following function:

$$\text{Fitness} = E(y^4) - 3(E(y^2))^2 \quad (2)$$

A maximum value of the fitness function indicates that the obtained components have a nongaussian distribution, so the components are independent. The best solution is used to generate the independent components of the MS image. The third independent component is replaced with the PAN image and then the new combination is multiplied by the inverse of the mixing matrix to obtain the fused image.

FPGA is used in the implementation of the proposed hardware system. Figure 1 shows the top level of the proposed hardware system. It comprises three counters, a 32-bit pseudo random number generator (SRG) [14], and GA finite state machine (GA_FSM). The GA_FSM is the core of the designed system and is responsible for the execution of the GA evolution cycle. Image data is entered to the design line by line or precisely pixel by pixel then the GA processing is performed on these pixels.

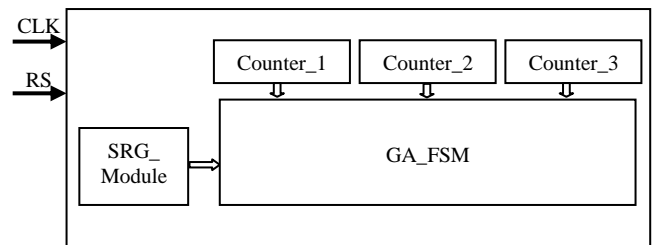


Fig. 1. The Top-level of the proposed design

The GA finite state machine is designed using VHDL in a pipelined fashion. This increases the maximum clock frequency, reduces the time to synthesis the code, and increases the throughput of the system. The counters are used for indexing the individuals of the population and selecting the genes of the chromosomes. The SRG is set to run in the background, in parallel with the GA operations. In fact, running the SRG independently in parallel with the main task gives an additional randomization to the design.

Unlike deterministic search algorithms, GAs use probabilistic transition rules to guide their search. Hence the implementation of the random number generator is required for processing the following tasks:

- Creation of the initial population.
- Selection of the surviving chromosomes for the next generation.
- Choosing the chromosomes for crossover.
- Selection of genes for mutation.

III. SIMULATION AND SYNTHESIS RESULTS

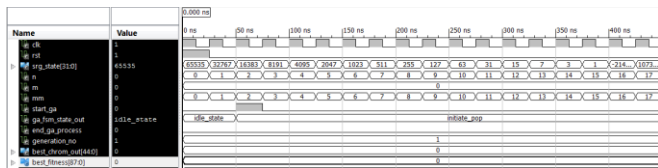
The proposed FPGA architecture for the implementation of the ICA is designed using the Spartan-3A DSP 1800A Xilinx FPGA. This is because Spartan-3A DSP FPGA device has embedded DSP modules and the fitness function of the GA requires many addition and multiplication operations. The whole design is created using VHDL and simulated using Xilinx ISE design suite 12.1. The performance of the developed FPGA implementation of the ICA algorithm is verified through comparing the hardware results with that obtained from Matlab.

Fig. 2 demonstrates the simulation results of the proposed FPGA architecture at the beginning, middle, and end of the GA evolution process through different generations. At the beginning of the GA, as shown in Fig. 2.a, the GA_FSM starts when Start_Ga signal is '1' to evolve the best solution (mixing matrix) that gives the independent components of the MS image with a maximum independency. The GA_FSM uses counter1, counter2, and the SRG for the creation of the initial population of the GA evolution cycle. It is clear from Fig.2 that the SRG is running in parallel with the other design modules and its state changes in each clock cycle. When the GA reaches generation number 100 it stops its process with the best obtained mixing matrix that is corresponding to maximum fitness as shown in Fig. 2.c.

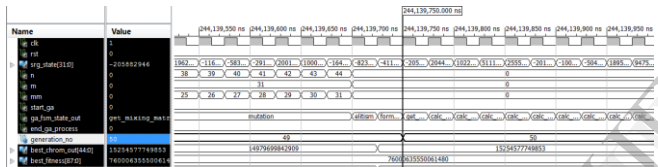
The performance of the developed FPGA implementation of the ICA algorithm is tested by transforming the MS image of the Quickbird satellite image data to its independent components using the best obtained mixing matrix using the

proposed FPGA design and Matlab. The obtained components are then used to fuse the MS and PAN images of the Quickbird satellite image data. The best obtained mixing matrices using the proposed FPGA design and Matlab are shown in Fig. 3. The GA evolution process through different generations are also demonstrated in Fig. 3. It is clear that the hardware evolves the mixing matrix the same way as Matlab. The visual quality of the obtained fused images, shown in Fig. 3, reveals a similar performance of the proposed FPGA design and Matlab.

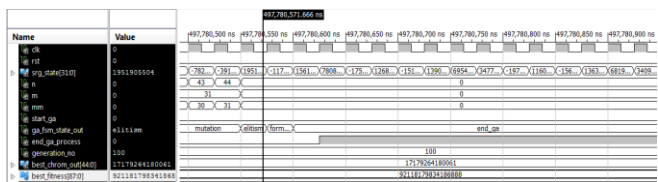
Table I demonstrates synthesis results of the proposed FPGA design. The results show that the design uses 70% of the slices and 50% of the DSP modules. The design also operates at a frequency of 36.7 MHz. Hence, the synthesis results of the proposed FPGA design on the Spartan-3A DSP 1800A Xilinx FPGA show good utilization of the device resources while maintaining high system clock.



(a)



(b)



(c)

Fig. 2. Samples of functional simulation waveforms of the designed system (a) Start of the GA process (b) Middle of the GA process (c) End of the GA process

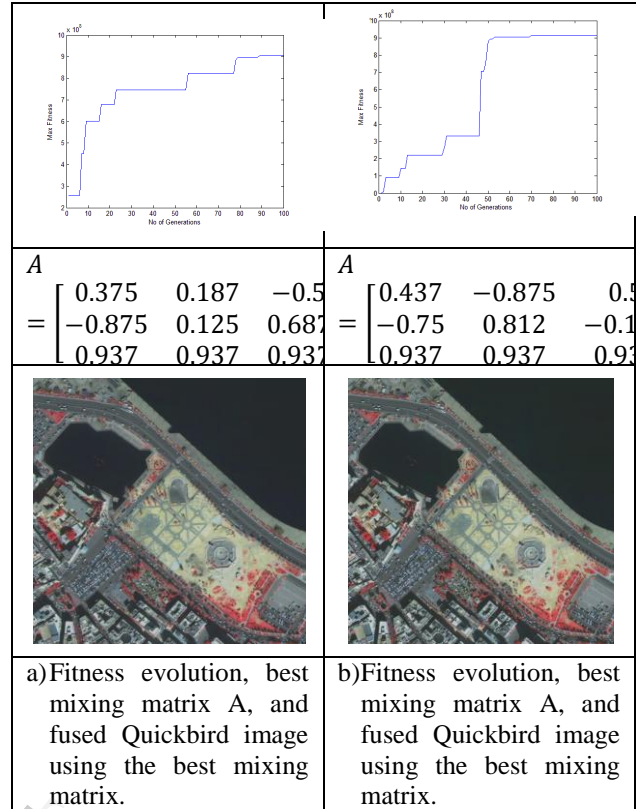


Fig. 3. GA evolution results using (a) Matlab and (b) The proposed FPGA design of functional simulation waveforms of the designed

TABLE I. SYNTHESIS RESULTS OF THE PROPOSED DESIGN

Logic Utilization	used	available	Utilization
Number of Slices	11722	16640	70%
Number of Slice Flip Flops	7022	33280	21%
Number of 4-input LUTs	22442	33280	67%
Number of GCLKs	1	24	4%
Number of DSP48S	42	84	50%
Maximum Clock Frequency	36.7 MHz		

IV. CONCLUSION

In this paper, FPGA architecture of the ICA for the fusion of remote sensing images is presented. The design includes state machine for the implementation of the GA evolution cycle, three counters, and 32-bit RNG for the selection of the chromosomes and the implementation of the GA operations. The whole design is created using VHDL and verified through comparing the results with that obtained from Matlab tools. The design is tested with Quickbird satellite images. The evolved hardware shows similar performance compared to Matlab software. The design runs at a frequency of 36.7 MHz on Spartan-3A DSP 1800A Xilinx FPGA. Synthesis results show good utilization of the device resources while maintaining high system clock.

REFERENCES

- [1] C. Pohl, and V. Genderen, "Multisensory image fusion in remote sensing: concepts, methods and application," *Int journal of remote sensing*, vol. 19, pp.823- 854. 1998.
- [2] M. Choi, "A new Intensity-Hue-Saturation Fusion Approach to Image Fusion with A trade-off Parameter," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 44, pp.1672-1682. 2006.
- [3] U. Patil, U. Mudengudi, "Image Fusion Using Hierarchical PCA," *IEEE International Conference on Image Information Processing (ICIIP)*. 2011.
- [4] H. Ma, C. Jia, S. Liu, "Wavelet-Based PAN and Multispectral Image Fusion," *Proceeding of the 6th World Congress on Intelligent Control and Automation*, vol. 2, pp. 9613 - 9617. 2006.
- [5] A. M. El Ejaily, F. Eltohamy, M. S. Hamid, and G. Ismail
"An Image Fusion Method Using DT-CWT and Average Gradient"
International Journal of Computer Science and Mobile Computing, Vol. 3 Issue.1, January- 2014
- [6] Z. Wang, X. Yu, W. Yu, and D. Sha, "A remote Sensing Image Fusion Algorithm Based on Nonnegative Ordinal Independent Component Analysis by Using Lagrange Algorithm" *IEEE International Conference*
- [7] A. Hyvarinen, J. Karhunen, E. Oja, *Independent Component analysis*, John Wiley & Sons, Inc. 2001
- [8] A. M. El Ejaily, F. Eltohamy, M. S. Hamid, and G. Ismail "Satellite Image Fusion Using Maximization of Non-Gaussianity" *International Journal of Computer Science and Mobile Computing*, vol.3, Issue. 4, pp.401-411, 2014.
- [9] J. Holland. "Adaptation in natural and artificial systems", MIT Press, Cambridge, Mass., 2nd edition, 1992.
- [10] K.S. Tang, K.F. Man, S. Kwong and Q. He " Genetic Algorithms and Their Applications", *IEEE signal processing magazine*, vol. 13, pp. 22-37, 1996.
- [11] K. F. Man, T. S. Tang, and S. Kwong, "Genetic Algorithms: Concepts and Design," Springer Verlag London, 1999..
- [12] Ion Grout "Digital System Design with FPGAs and CPLDs " Elsevier Ltd, 2008.
- [13] Steve Kilts " AdvancedFPGA Design Architecture, Implementation and Optimization" John Wiley & Sons. Inc, 2007
- [14] F. Jessie M. Williams, and N J. A. Sloane " Pseudo-Random Sequences and Arrays" Proceedings of the IEEE, December 1976.

IJERT