# Efficient Architecture for Digital Image Processing Based on EPLD

Muzhir Shaban Al-Ani
*University of Human Development, College of Science and Technology, Department of Computer Science, Sulaimani, KRG, Iraq [E-mail: muzhir.al-ani@uhd.edu.iq]*

***Abstract:*** *Digital image processing becomes a wide area of applications that introduce in each part of real life. One of the big challenge of digital image processing is how to speed up the processing speed in order to support the real time application. The main objective of this work is to design a simple method that achieve the digital image processing with in the real time video rate. The architecture of real time digital image processing approach needs high speed components to process a big amount of pixels at minimum time as possible. The implemented approach used Erasable Programmable Logic Device (EPLD) to realize the real time digital image processing. This approach combine both parallel processing and pipeline processing in the architecture. This approach leads to good architecture performance.*

***Keywords:*** *EPLD, Real Time Processing, High Speed Processing, Parallel Processing and Digital Control.*

Date of Submission: 02-11-2017 Date of acceptance: 16-11-2017

## I. Introduction

The most important concept of the design of any system is the right compensation between cost and quality [1,2]. Roughly speaking about quality of image that have big amount of pixels, on the hand this big amount of pixels need a high speed of processing to reach the real time of processing [3,4]. So most of digital image processing try to combine between software and hardware in such a way to achieve reliability and high speed of processing [5,6,7]. Increasing the size of image leads to increase the number of pixel per image that means required more processing time but on the other hand this gives good quality [8,9,10].

Programmable Logic Device (PLD) and Programmable Logic Array (PLA) have been in use since 70s [11,12,13]. Real time processing deals with the introducing more hardware in the implemented system [14,15,16]. At this point of view introducing pipeline processing and parallel processing lead to high speed of processing [17,18,19]. On the other word introducing Erasable Programmable Logic Device (EPLD) leads an acceptable approach for real time digital image processing [20,21,22].

To address this design challenge in the domain of image processing, EPLD approach have been presented that introduce an efficient hardware architecture for digital image processing. This approach try to overcome the problems arise of real time image processing.

## II. Digital Image Processing

Image processing is a direct application of two dimensional signal processing, on the other hand any processing via processors is a type of digital processing including digital image processing [23,24]. Processing filed can be categorized according input and output data into types (Fig. 1) [25,26]:

Data processing (ID/OD), in which the input is data and the output is data. This including the study of generation of data including controlling data, pattern recognition etc. [27,28]. Artificial intelligent (II/OD), in which the input is image and the output is data. This receive all pixels from image and produce some significant output data such as statistics, area, information, histogram etc. [29,30].

Computer vision (ID/OI), in which the input is data and the output is image. This including that algorithm which receive input data generated by certain devices to generated images. This also may be used to find the structure of the system [31,32].

Image processing (II/OI), in which the input is image and the output is image. This make some transformation of images to generate an output image with same size of input. These algorithms may be filter, convolution, enhancement, transformation etc. [33,34].

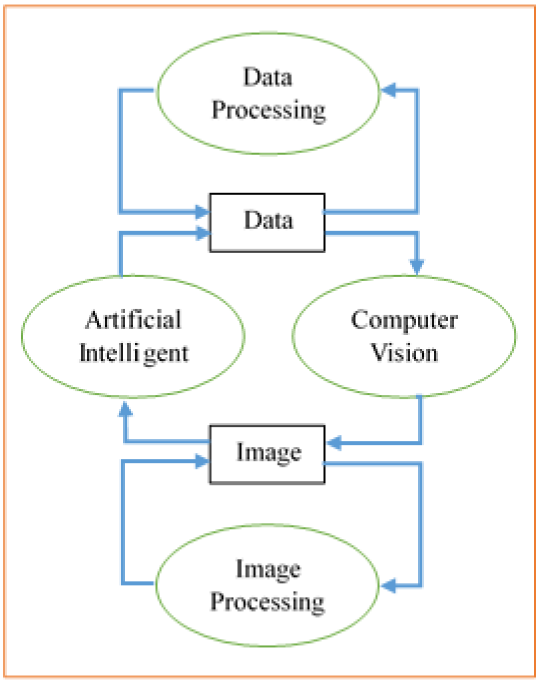

**Figure 1** fields of processing

## III. Literature Review

The field of architecture for digital image processing is widely expanded for their wide range of application and published works. This section will be focusing on the updated published papers.

S. Schulze, S. Sawitzki (2012) introduced and explained the basic methods for describing hardware in Lava to the implementation of the instruction set architecture and the structure and control of the pipelines. The results of the FPGA synthesis are presented and compared with a traditional design flow based on VHDL. The implementation of a coprocessor interface used to accelerate application-specific code is discussed with the Fast Fourier Transformation case study. For the best knowledge of the authors, this is the first attempt to describe, simulate, verify, synthesize and test a complete von Neumann machine in Lava. The project experiments are summarized, followed by an outline of the possible orientations for the improvement of the lava in particular and of the functional languages of description of the material in general [35].

Thorsten M. Buzug (2012) proposed two magnetic fields for static selection field and an oscillating drive field are created and superimposed. If super paramagnetic iron oxide (SPIO) nanoparticles are subjected to the oscillating magnetic field, the particles will react with a non-linear magnetization response, which can be measured with a suitable collection coil device. Due to the non-linearity of the magnetization of the particles, the signal received consists of the fundamental excitation frequency as well as the harmonics. After separation of the fundamental signal, the concentration of the nanoparticles can be reconstructed quantitatively as a function of the harmonics. Spatial coding is done with the static selection field that produces a free field point, which is moved in the field of view [36].

Anders Eklund (2013) described a colored digital image processing system. This system can configure a set of filters according to the requirements of a user, to adapt the system to the images to be processed. This approach performed smoothing, edge detection, histogram equalization, color normalization, and gloss normalization functions. The system has been described using the hardware description language of system Verilog and implemented in an Altera FPGA. Due to its configurable feature, this system is capable of processing color images for various applications such as agriculture and medicine [37].

Amir Gargouri, Dorra Sellami Masmoudi (2013) focused on a new digital architecture of the pulsed mode neuro-fuzzy system with a learning ability on chip. The main objective is to use the exceptional neuro-fuzzy characteristics in the function approximation and implement a reconfigurable architecture with a chip-based learning in a field-programmable gate array platform. Details of the entire design are provided with on-chip learning solutions. As an application illustrated the effectiveness and scalability of the proposed pulsed mode neuro-fuzzy system. Image degradation approach is considered that is very important step in image processing work. The experimental results show a high efficiency of the proposed method, surpassing the other decontamination techniques [38].

A. Swarnalathaa, A.P. Shanthi (2014) achieved a major acceleration by transferring the implementation to the hardware. The main problems to be dealt with in the implementation of the hardware are the scalability, the flexibility and the reduction of the calculation time. This is the first system based on full hardware evolution on the programmable chip for evolvable hardware. The architecture includes the memory and the modules necessary to perform all the operations of the algorithm. It is built entirely in the configurable logic blocks of a field programmable gate array. The encoding is performed using the Verilog hardware description language. The results show that the architecture is able to respond to the unrestricted evolution of the number of generations, accompanied by increasing in the use of resources [39].

Zhilei Chai et al. (2014) studied the acceleration of algorithms for the detection of image limits by parallelism material. The probability limit algorithm is selected as the representative algorithm for detecting high quality boundary based on the gradient. First, there are different types of parallelism in probability limit and are analyzed. Next, this approach discussed an appropriate hardware structures to accelerate probability limit as a function of these parallels. Finally, the performance, accuracy and scalability of the probability limit parallel detector accelerated by the hardware is presented. This document demonstrates a promising way to improve the real-time performance of high quality image border detection systems, especially when integrated and real time systems are considered [40].

Oliver Reiche et al. (2015) Explained two approached, one approach is to use a domain-specific language to generate code highly optimized for synthesis using high-level general purpose synthesis frameworks. Another approach is to instantiate a generic IP-Core VHDL library for local image operators. They provided a comparison of the results for both approaches, a non-expert algorithm developer can achieve. In addition, an automatic optimization process is presented to give the developer more control algorithm at the time of operations for the use of resources, which could be applied through the two approaches. To evaluate this optimization procedure, this approach compared the implementations of FPGA accelerators result with highly optimized several image filters relevant for image processing and near-sensor video with strict real-time constraints [41].

Luis Araneda, Miguel Figueroa (2015) presented a hardware architecture for real-time digital video stabilization for high performance embedded systems. The stabilization algorithm analyzed the current and past video images and obtained a motion estimation vector, which is filtered to isolate unwanted camera movements from the intentional pan. The vector is then used to correct the output video frame. The approach described the hardware architecture for motion estimation, filtering and correction and its implementation in a Xilinx Spartan-6 LX45 FPGA. The results are evaluated on several reference video sequences, both visually and using the inter-frame conversion fidelity index [42].

Garnet Wilson, Y. Premson (2016) explained the procedure to achieve real-time performance for high-definition video applications, it is necessary to design an effective hardware architecture to improve contrast to meet the needs of real-time processing. Also this approach explained a modified adaptive gamma correction algorithm with weight distribution to improve contrast. This new hardware-oriented contrast enhancement algorithm is modeled in Xilinx system generator and implemented in FPGA using Zed Board XC7Z020. This algorithm improved the material-oriented contrast allows good image quality by measuring the results of qualitative and quantitative analyzes [43].

Sambaran Hazra et al. (2016) presented a new histogram generation hardware architecture, which can develop a histogram for all types of grayscale images (256 x 256). The histogram produced from the proposed material provides exactly the same information as that obtained from a histogram plot using a simulation tool such as Matlab with 100% accuracy. The physical realization of the architecture makes it possible to generate a histogram in real time in medical and military applications, which is practically difficult in software-based implementations. The experimental results show that the operating frequency of the proposed architecture is very high for both Field Programmable Gate Array (FPGA) and programmable SoC [44].

Wei Jin et al. (2017) proposed a CMOS image signal processor of 4928 × 3264 pixels for low-complexity, high-performance digital cameras. To reduce equipment costs and maintain a high performance, new algorithm is proposed to process image signals. A joint disassembly and relocation algorithm is presented for color interpolation and the elimination of Gaussian noise. This new set algorithm achieves high performance and logs line buffers. In addition, improving the edge of the image is done together with this algorithm to save the cost of memory. Then, an automated low-complexity white-balance hardware architecture based on a histogram equalization algorithm is presented. This approach eliminates the impulse noise, via hardware implementation of high performance and low complex based on the average filter using only nine comparators [45].

Aakash Patil (2017) demonstrated the use of a low power and compact random feature extractor hardware (RFE) implementation in image recognition applications. It is shown that weight distributions with zero means are necessary for good performance in this application. This approach shows the performance in this application of a newly proposed weight reuse technique that practically expands the number of random functionalities available from the random feature extractor kernel. In addition, this approach proposed a method to reduce computation costs by pruning random neurons "unknowns" or random redundant [46].

## IV. Methodology

This approach guided a direct methodology that start a simple and effective procedure to design and perform the architecture for digital image processing via the EPLDs.

### EPLD Architecture

Integrated circuits (IC) starts from few processing elements up to Ultra High Scale Integration (UHSI) that integrate millions of processing elements via IC. For a long time EPLD started a significant field of processing via building ICs associated with the real time system design concentrated on specific hardware at low cost. Image processing algorithm is designed and implemented using EPLD circuit to perform the input and output specifications (identify all pins), this named as Transformation Units (TUs). Also other specifications of EPLD chip is determined to perform data in/data out algorithm that named Generation of Data Units (GDUs).

Depending on these specifications an image processing system is developed that shown in Fig, 2. In this system many embedded algorithms can be run at real time including the necessary circuits of TUs and GDUs that implemented via EPLD. From the implemented system if is more convenient that the pipelines (which represented the paths between TUs that can be programmed), these paths can be determined by software on a network of switches.

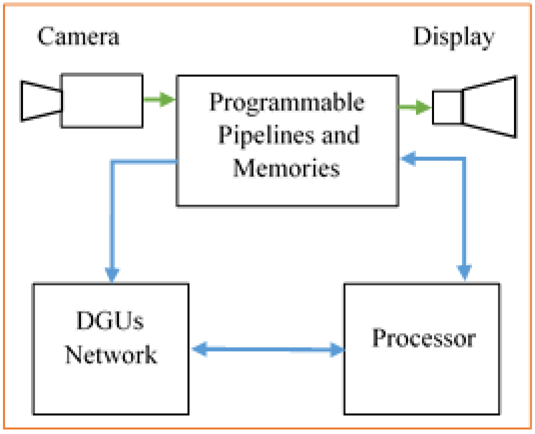

**Figure 2** EPLD architecture

### Transformation Units Architecture

Transformation units is designed to perform image processing unit pixel by pixel to generate an output image that depends on the hardware design of the TU architecture. One of the main characteristic of the implemented design is that must be able to work in real time within the video rate. TUs always required certain paths to communicate with the processor, this communication is necessary to perform different functions in the TUs. Considering that this communication dose not interrupt the processing rate of video signal. Applying this configuration leads a set of daughter board with EPLDs can be performed and many different tasks can be implemented such as refresh memory, binarization, convolution, filtering, histogram, storage, etc.

Each TU receives input image as flow of pixels from preceding TU and transmit that to the next TU. The first TU of the pipeline receives its input from a digitizer, and the last TU deliver its output to a video signal generator. As the image is processed through TUs pipeline, then the performance of the system increases greatly if the path of the image can be programmed. Consequently, it is necessary to configure the network between input and output of each TU. Fig. 3 illustrate how this network receives all outputs from TUs and generates all inputs to TUs. The input firstly converted into digital using analog to digital convertor (ADC) then the output converted into digital form using digital to analog convertor (DAC). The correspondence between inputs and outputs of the network can be programmed by the system processor. All the inputs (flow of pixels) must be accepted by the network at video rate and the computer must be generated at the same rate, but it does not matter the time delay between the input (pixels) and the output result (pixels).

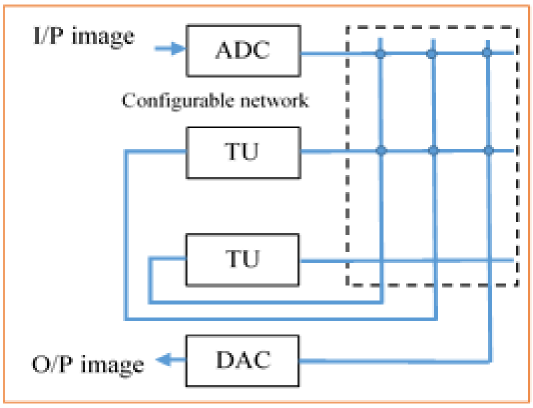

**Figure 3** configurable network architecture

**Implemented System Approach**

The architecture of the real time digital image processing system based on EPLD is implemented to perform the video rate processing. This system consist of many hardware parts such as (Fig. 4):

- ADC: analog to digital convertor that convert analog image into digital form to be processed at the specified processor. ADC is chosen to be 8 bits that perform one pixel, in addition it is of flashing type that achieve high speed of conversion.
- DAC: digital to analog convertor that convert digital image into analog form to be displayed on the screen. DAC is chosen to be 8 bits that receive one pixel of 8 bits, in addition it is of flashing type that achieve high speed of conversion.
- TU architecture: four units of TU are used and configure to perform point processing, area processing and two storage units.
- DGU architecture: this unit is implemented to perform the histogram operation in which calculate number of pixels in each gray scale.
- ALU: arithmetic logic unit is implemented to perform the required operation for the overall system.
- Processor: this is a digital signal processing (DSP) processor that perform high speed processing. The main purpose of the DSP processor is used to control both flow of data (pixels) and the addressing of data in addition it is the main control of the all units of the system.

These parts are connected and synchronized with the video rate according to the flow of pixels. The communications between these parts are established to perform the mentioned operations via the algorithms that run with the same hardware unit. Convolution is one of the important operation, so to perform the convolution operation of N1*N2 mask and M1*M2 image, this operation has the ability to change the filter coefficient which represent the mask coefficients. The TUs and DGUs are able to receive images pixel by pixel at video rate. The network receives all the outputs of TUs and generates all the inputs to the TUs and DGUs without changing. The networks are constructed from set of big multiplexers and it is easier to make it with a structure able to perform a sequential reading of input and sequential writing of output, this network is more efficient applying EPLDs.

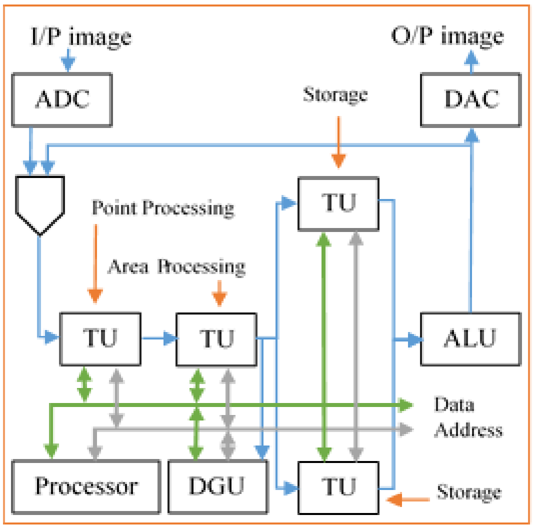

**Figure 4** implemented system approach

## V. Conclusion and recommendation

The required information concerned as an important part of the system design and implementation. The hardware architecture of the implemented system deals with high speed of processing, in addition this aspect is an important part if digital image processing. Digital image processing architecture have been developed to perform many operations of this field of processing. An analyzed of the requirements for image processing applications is the guide to establish what requirements are common for more sets of tasks. Introducing EPLD lead to develop an efficient architecture with high characteristics such as high resolution, perfect integrated circuits with specified functions, and flexibility of programmable software in addition to work with high speed processing.

## References

[1] Abdullah Mohammed Awad, Muzhir Shaban Al-Ani (2017) Efficient Image Compression Based on Modified Haar Wavelet. Journal of Theoretical and Applied Information Technology, 15th June 2017. Vol.95, No.11, 2442- 2452.

[2] Muzhir Shaban Al-Ani and Khattab M. Ali Alheeti (2017) Precision Statistical Analysis of Images Based on Brightness Distribution. Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 4, 99-104.

[3] R.J. Winder, P.J. Morrow, I.N. McRitchie, J.R. Bailie, P.M. Hart (2009) Algorithms for digital image processing in diabetic retinopathy. Computerized Medical Imaging and Graphics, Vol. 33, Issue 8, December 2009, PP. 608-622.

[4] Khader Mohammad, Sos Agaian, Fred Hudson (2010) Implementation of Digital Electronic Arithmetics and its application in image processing. Computers & Electrical Engineering, Vol. 36, Issue 3, May 2010, PP. 424-434.

[5] S. Dutta, S.K. Pal, S. Mukhopadhyay, R. Sen (2013) Application of digital image processing in tool condition monitoring: A review. CIRP Journal of Manufacturing Science and Technology, Vol. 6, Issue 3, 2013, PP. 212-232

[6] Hyuck-Joo Kwon, Jae-Ho Nah, Dinesh Manocha, Woo-Chan Park (2014) Effective traversal algorithms and hardware architecture for pyramidal inverse displacement mapping. Computers & Graphics, Vol. 38, February 2014, PP. 140-149.

[7] Zeqian Lv, Xiaohai Xu, Tianhao Yan, Yulong Cai, Qingchuan Zhang (2018) High-accuracy optical extensometer based on coordinate transform in two-dimensional digital image correlation. Optics and Lasers in Engineering, Vol. 100, January 2018, PP. 61-70.

[8] Muzhir Shaban Al-Ani (2017) Efficient Image Encryption Approach Based on Chaos Technique. IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE). Volume 12, Issue 3 Ver. II (May – June 2017), PP.54-60.

[9] Muzhir Shaban Al-Ani (2017) Hardware Implementation of a Real Time Image Compression. IOSR Journal of Computer Engineering (IOSR-JCE). Volume 19, Issue 3, Ver. V (May - June 2017), PP 06-13.

[10] Carlos González, Sergio Sánchez, Abel Paz, Javier Resano, Antonio Plaza (2013) Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing Integration, the VLSI Journal, Vol. 46, Issue 2, March 2013, PP. 89-103.

[11] Layla Horrigue, Taoufik Saidani, Refka Ghodhbani, Julien Dubois, Mohamed Atri (2014) An efficient hardware implementation of MQ decoder of the JPEG2000. Microprocessors and Microsystems, Vol. 38, Issue 7, October 2014, PP. 659-668.

[12] Tomislav Janjusic, Krishna Kavi (2014) Chapter Three: Hardware and Application Profiling Tools. Advances in Computers, Volume 92, 2014, Pages 105-160.

[13] Ali Gholami Rudi, Saeed Jalili (2015) Image preprocessing with a parallel optoelectronic processor. Computers & Electrical Engineering, Vol. 46, August 2015, PP. 554-565.

[14] J. Kotyza, V. Kasik (2016) Image Processing of Composite Video with FPGA Programmable Logic. IFAC-Papers On Line, Vol. 49, Issue 25, 2016, PP. 482-486.

[15] Haider Ismael Shahadi, Razali Jidin, Wong Hung Way (2016) Concurrent hardware architecture for dual-mode audio steganography processor-based FPGA. Computers & Electrical Engineering, Vol. 49, January 2016, PP. 95-116.

[16] Matt D. Miesner, Emily J. Reppert (2017) Diagnosis and Treatment of Hardware Disease. Veterinary Clinics of North America: Food Animal Practice, Vol. 33, Issue 3, November 2017, PP. 513-523.

[17] Vineet Kumar, Abhijit Asati, Anu Gupta (2017) Hardware implementation of a novel edge-map generation technique for pupil detection in NIR images. Engineering Science and Technology, an International Journal, Vol. 20, Issue 2, April 2017, PP. 694-704.

[18] Jef Thoné, Jeroen Verlinden, Robert Puers (2010) An efficient hardware-optimized compression algorithm for wireless capsule endoscopy image transmission. Procedia Engineering, Vol. 5, 2010, PP. 208-211.

[19] Huang You-wen (2011) An efficient hardware implementation for deblocking filter of AVS decoder. Procedia Environmental Sciences, Vol. 11, Part A, 2011, PP. 505-510.

[20] O. Fluck, C. Vetter, W. Wein, A. Kamen, R. Westermann (2011) A survey of medical image registration on graphics hardware. Computer Methods and Programs in Biomedicine, Vol. 104, Issue 3, December 2011, PP. e45-e57.

[21] P. Baptista, T.R. Cunha, C. Gama, C. Bernardes (2012) A new and practical method to obtain grain size measurements in sandy shores based on digital image acquisition and processing. Sedimentary Geology, Vol. 282, 30 December 2012, PP. 294-306.

[22] João P.F. Barbosa, Antonyus P.A. Ferreira, Rodrigo C.F. Rocha, Erika S. Albuquerque, Edna N.S. Barros (2015) A high performance hardware accelerator for dynamic texture segmentation. Journal of Systems Architecture, Vol. 61, Issue 10, November 2015, PP. 639-645.

[23] Ayman Alfalou, Christian Brosseau (2015) Chapter Two: Recent Advances in Optical Image Processing. Progress in Optics, Vol. 60, 2015, PP. 119-262.

[24] Roberto Perez-Andrade, René Cumplido, Claudia Feregrino-Uribe, Fernando Martin Del Campo (2010) A versatile hardware architecture for a constant false alarm rate processor based on a linear insertion sorter. Digital Signal Processing, Vol. 20, Issue 6, December 2010, PP. 1733-1747.

[25] Alison Barros, Euler de Vilhena Garcia, Rafael Morgado Silva (2011) Simulation of stochastic processes using graphics hardware. Computer Physics Communications, Vol. 182, Issue 4, April 2011, PP. 989-993.

[26] Z.A. Ahmad, M. Elshaikh, C.M. Nor, M.S. Mustafa, M.F. Jamlos (2012) Low Cost Parallel Processing System for Image Processing Applications. Procedia Engineering, Vol. 41, 2012, PP. 771-776.

[27] Gianpaolo Cugola, Alessandro Margara (2012) Low latency complex event processing on parallel hardware. Journal of Parallel and Distributed Computing, Vol. 72, Issue 2, February 2012, PP. 205-218.

[28] Sébastien Courroux, Stéphane Chevobbe, Mehdi Darouich, Michel Paindavoine (2013) Use of wavelet for image processing in smart cameras with low hardware resources. Journal of Systems Architecture, Vol. 59, Issue 10, Part A, November 2013, PP. 826-832.

[29] Kai Hu, Qiaocui Cheng, Bodong Li, Xieping Gao (2018) The complex data denoising in MR images based on the directional extension for the undecimated wavelet transform. Biomedical Signal Processing and Control, Vol. 39, January 2018, PP. 336-350.

[30] A. De Silvestro, K. Martini, A.S. Becker, T.D.L. Kim-Nguyen, T. Frauenfelder (2017) Postoperative imaging of orthopaedic hardware in the hand and wrist: is there an added value for tomosynthesis?. Clinical Radiology, In press, corrected proof, Available online 11 September 2017.

[31] Mackenzie G. Glaholt, Grace Sim, Philips Laou, Simon Roy (2018) Evaluation of fused imagery using eye movement-based measures of perceptual processing. Information Fusion, Vol. 39, January 2018, PP. 186-203.

[32] Sourour Karmani, Ridha Djemal, Rached Tourki (2009) Efficient hardware architecture of 2D-scan-based wavelet watermarking for image and video. Computer Standards & Interfaces, Vol. 31, Issue 4, June 2009, PP. 801-811.

[33] Giuseppe Cocorullo, Pasquale Corsonello, Fabio Frustaci, Stefania Perri (2016) An efficient hardware-oriented stereo matching algorithm. Microprocessors and Microsystems, Vol. 46, Part A, October 2016, PP. 21-33.

[34] Tom Pearson (2009) Hardware-based image processing for high-speed inspection of grains. Computers and Electronics in Agriculture, Vol. 69, Issue 1, November 2009, PP. 12-18.

[35] S. Schulze, S. Sawitzki (2012) Processor design using a functional hardware description language Microprocessors and Microsystems. Vol. 36, Issue 8, November 2012, PP. 676-694.

[36] Thorsten M. Buzug, Gael Bringout, Marlitt Erbe, Ksenija Gräfe, Matthias Graeser, Mandy Grüttner, Aleksi Halkola, Timo F. Sattel, Wiebke Tenner, Hanne Wojtczyk, Julian Haegele, Florian M. Vogt, Jörg Barkhausen, Kerstin Lüdtke-Buzug (2012) Magnetic particle imaging: Introduction to imaging and hardware realization. Zeitschrift für Medizinische Physik Vol. 22, Issue 4, December 2012, PP. 323-334.

[37] Anders Eklund, Paul Dufort, Daniel Forsberg, Stephen M. LaConte (2013) Medical image processing on the GPU – Past, present and future. Medical Image Analysis, Volume 17, Issue 8, December 2013, Pages 1073-1094.

[38] Amir Gargouri, Dorra Sellami Masmoudi (2013) New pulse mode neuro-fuzzy hardware architecture and application to image denoising. AEU - International Journal of Electronics and Communications Vol. 67, Issue 6, June 2013, PP. 513-520.

[39] A. Swarnalathaa, A.P. Shanthi (2014) Complete hardware evolution based SoPC for evolvable hardware. Applied Soft Computing. Vol. 18, May 2014, PP. 314-322.

[40] Zhilei Chai, Xinglong Shao, Yuanpu Zhang, Wenmin Yang QinWu (2014) Accelerating image boundary detection by hardware parallelism. Microprocessors and Microsystems. Vol. 38, Issue 5, July 2014, Pages 458-469.

[41] Oliver Reiche, Konrad Häublein, Marc Reichenbach, Moritz Schmid, Frank Hannig, Jürgen Teich, Dietmar Fey (2015) Synthesis and optimization of image processing accelerators using domain knowledge. Journal of Systems Architecture Vol. 61, Issue 10, November 2015, PP. 646-658.

[42] Luis Araneda, Miguel Figueroa (2015) A compact hardware architecture for digital image stabilization using integral projections. Microprocessors and Microsystems. Vol. 39, Issue 8, November 2015, PP. 987-997.

[43] Garnet Wilson, Y. Premson (2016) FPGA Implementation of Hardware Efficient Algorithm for Image Contrast Enhancement Using Xilinx System Generator. Procedia Technology. Vol. 24, 2016, Pages 1141-1148.

[44] Sambaran Hazra, Sudip Ghosh, Santi P. Maity, Hafizur Rahaman (2016) A New FPGA and Programmable SoC Based VLSI Architecture for Histogram Generation of Grayscale Images for Image Processing Applications. Procedia Computer Science Vol. 93, 2016, Pages 139-145.

[45] Wei Jin, Guanghui He, Weifeng He, Zhigang Mao (2017) A 12-bit 4928 × 3264 pixel CMOS image signal processor for digital still cameras. Integration the VLSI Journal. Vol. 59, September 2017, Pages 206-217.

[46] Aakash Patil, Shanlan Shen, Enyi Yao, Arindam Basu (2017) Hardware architecture for large parallel array of Random Feature Extractors applied to image recognition. Neurocomputing. Vol. 261, 25 October 2017, Pages 193-203.

Muzhir Shaban Al-Ani Efficient Architecture for Digital Image Processing Based on EPLD.” IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE), vol. 12, no. 6, 2017, pp. 01-07.