ISSN:2249-5789
Tarek M Bittibssi et al , International Journal of Computer Science & Communication Networks,Vol 2(4), 536-542

Image Enhancement Algorithms using FPGA

Tarek M. Bittibssi [1] , Gouda I. Salama [2] , Yehia Z. Mehaseb [3] and Adel E. Henawy [4]
[1&4] Dept. of ECE, Ain Shams University, Cairo, Egypt
[2&3] Dept. of ECE, Military Technical College, Cairo, Egypt

Abstract

Image Enhancement techniques can be classified into two categories as spatial domain and frequency domain. This paper presents hardware implementation of a five image enhancement algorithms in spatial domain using FPGA technology. These algorithms are: median filter, contrast stretching, histogram equalization, negative image transformation and power-law transformation. All algorithms are studied and hardware circuits are realized for them. Then hardware is modeled using Altera System and synthesized onto Cyclon III on Development kit (DE0) FPGA chip.

Keywords: Image Enhancement, FPGA

I. INTRODUCTION

Image enhancement is the processing of images to improve their appearance to human viewers or to enhance other image processing systems performance. Methods and objectives vary with the application. When images are enhanced for human viewers, as in television, the objective may be to improve perceptual aspects: image quality, intelligibility, or visual appearance. In other applications, such as object identification by machine, an image may be pre-processed to aid machine performance. Because the objective of image enhancement is dependent on the application context, and the criteria for enhancement are often subjective or too complex to be easily converted to useful objective measures, image enhancement algorithms tend to be simple, qualitative, and disambiguation

In [1], S. Sowmya , Roy Paily addresses the implementation of image enhancement algorithms like brightness control, contrast adjustment and histogram equalization on FPGA. The minimum period to the implemented algorithms is 5 ns for an image size 100x100.

In [2], Wang Bing-jian, Liu Shang-qian, Li Qing and Zhou Hui-xin describe a simple and effective implementation of the histogram equalization algorithm, including its threshold value calculation, by using pipeline and parallel computation architecture. The proposed algorithm is used to enhance the contrast of infrared images generated from an infrared focal plane array system and image contrast is improved significantly. By using pipeline and parallel computation architecture, the system can process 25 frames of image size 128x128 with 8 bits infrared images in every second.

In [3], Nitin Sachdeva and Tarun Sachdeva give a design of real-time Histogram Equalization circuit for enhancement of images using FPGA. This design makes use of counters in conjunction with a special decoder designed to compute the histogram statistics and equalization in parallel, and the total time required to perform Histogram Equalization of an image of size 256×256 is 0.263 msec.

In [4] Gerasimos Louverdis, Ioannis Andreadis and Antonios Gasteratos design a hardware implementation of a content based median filter suitable for real-time impulse noise suppression is presented. The function of the proposed circuitry is adaptive; it detects the existence of impulse noise in an image neighborhood and applies the median filter operator only when necessary. The adaptive filter was designed and implemented in FPGA. The typical clock frequency is 55 MHz and the system is suitable for real-time imaging applications.

In [5] Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez and Juan A. Gómez-Pulido introduces a new architecture and optimizations for median filter implementation with FPGAs. The practical results show the effectiveness of our improvements allowing real-time processing and a minimum use of resources. The average execution time is indicated for the operation on 30 images of 640x480 pixels of 256 gray levels. The typical clock frequency is 16 MHz.

In [6] D.Dhanasekaran and K.Boopathy Bagan investigates a high-speed non-linear Adaptive median filter implementation is presented. Then Adaptive Median Filter solves the dual purpose of removing the impulse noise from the image and reducing distortion in the image. Adaptive Median Filtering can achieve

the filtering operation of an image corrupted with impulse noise of probability greater than 0.2.
In [7] Karan Kumar, Aditya Jain, and Atul Kumar Srivastava are designing, modeling, simulation and synthesis of four Image Enhancement techniques on FPGA like power law transformation, image negative, Gradien (Sobel) Filter and Laplacian Filter. The simulation and synthesis results were obtained and designs are successfully validated using hardware co-simulation feature of Virtex 4 kit.

This paper introduces implementation of a five image enhancement algorithms include: median filter, contrast stretching, histogram equalization, negative image transformation and power-law transformation.
The organization of this paper as follows: this section resents the related work. Section II, discusses the theory of the implemented image enhancement algorithms. Section III introduces hardware realizations for a five image enhancement techniques. Section VI reports experimental result. Finally, the conclusion of this paper presented in section V.

II. IMAGE ENHANCEMENT ALGORITHMS

This section discusses the theory of the implemented image enhancement algorithms like, 1) Median Filter 2) Contrast Stretching 3) Histogram Equalization 4) Negative image transformation 5) Power law transformation.

A. Median Filter

Median filtering is a non-linear, low-pass filtering method, which are used mainly to remove salt-and pepper noise from an image. A median filter can outperform linear low-pass filters on this type of noisy image, because it can potentially removes all the noise without affecting the "clean" pixels. Median filters remove isolated pixels, whether they are bright or dark. The steps for the implemented median filtering are shown in List 1. Common drawback of the median filtering are destroy small features in the image and computational cost. Computing a two-dimensional median for a *NxN* window, requires sorting of *NxN* elements for every image pixel Therefore, using median filtering in any real-time vision system requires a significant computational power.

1. Read the image from left to right, top to bottom pixel by pixel.
2. Initiate a 3 x 3 mask (neighborhood windows), starting from the pixel, whose value is going to change after filtering.
3. Extract all 3 x 3 mask elements and put into the 1-D element array.
4. Sort the 1-D element array in ascending order.
5. Extract the middle value of sorted element array Replace the current pixel value in the image with the medium pixel value in the 1-D element array,
6. Move to the next pixel.
7. Repeat steps 3 to 6 until end of the image.

List 1 : Median Filter Algorithm

B. Contrast Stretching

Contrast stretching attempts to improve an image by stretching the range of intensity values it contains to make full use of possible values. contrast stretching is restricted to a linear mapping of input to output values. The steps for the implemented Contrast stretching are shown in List 2.

1. Read the image pixel by pixel.
2. Determine the limits over which image intensity values will be extended. These lower and upper limits will be called a and b, respectively (for standard 8-bit Gray scale pictures, these limits are usually 0 and 255).
3. compute the value limits (min. = c, max. = d) in the unmodified picture.
4. Then for each pixel, the original value r is mapped to output value s using the function below

$$s = (r - c)\left(\frac{b - a}{d - c}\right) + a$$

List 2 : Contrast stretching Algorithm

C. Histogram Equalization

Histogram equalization is a common technique for enhancing the appearance of images. Suppose we have an image which is predominantly dark. Then its histogram would be skewed towards the lower end of the grey scale and all the image detail is compressed into the dark end of the histogram. If we could 'stretch out' the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer. Histogram equalization stretches the histogram across the entire spectrum of pixels (0 – 255) . It increases the contrast of images for the finality of human inspection and can be applied to normalize illumination variations in image understanding problems [8]. The steps for the implemented Histogram equalization are shown in List 3.

1. Read the image pixel by pixel.
2. Counts the occurrence of each pixel value in the image (256 value).
3. Compute the cumulative number of pixels (unscaled values**).**
4. Multiply each unscaled value with the scaling factor [G-1/MxN] to obtain the new scaled value. where G is the maximum grey level , M is the number of image rows, N is the number of image columns.
5. Assign a nearest available brightness values to the new scaled value**.**

List 3 : Histogram equalization Algorithm

C. Negative image transformation

The negative of an image with grey levels in the range [0, G-1] is obtained by the negative transformation which is given by the expression**, s = G - 1 – r**. This expression results in reversing of the grey level intensities of the image thereby producing a negative like image. The output of this function can be directly mapped into the grey scale look-up table consisting values from 0 to G-1.

D. Power law transformation

This transformation function is also called as gamma correction . which is given by the expression $s = cr^{\gamma}$. For various values of γ different levels of enhancements can be obtained. This technique is quite commonly called as Gamma Correction. If you notice, different display monitors display images at different intensities and clarity. That means, every monitor has built-in gamma correction in it with certain gamma ranges and so a good monitor automatically corrects all the images displayed on it for the best contrast to give user the best experience.

III. HARDWARE REALIZATIONS FOR FIVE OF THE IMAGE ENHANCEMENT TECHNIQUES

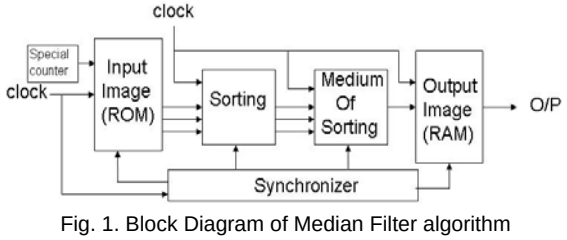

Fig. 1. Block Diagram of Median Filter algorithm

Fig 1 Shows the block diagram of a median filter algorithm, The ROM is initialized by image, special counter generate mask (neighbourhood window) moving on the image pixel. Then, mask will be mapped into 1-D array in sorting block. After that, sort the 1-D array in ascending order and Extract the middle value of sorted 1-D array by medium of sorting block. Finally put the middle value in output RAM.

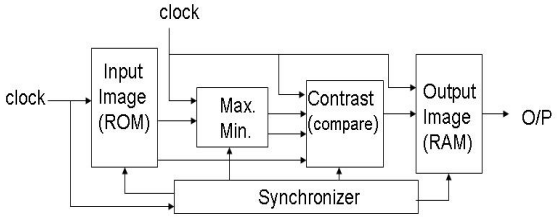

Fig. 2. Block Diagram of Contrast Stretching algorithm

Fig 2 shows the Block Diagram of Contrast stretching algorithm. The ROM is initialized by image. Max-Min Block will Counts the maximum & minimum pixel value occurrence in the image. Contrast compare block will apply "IF" condition, if image pixel value greater than maximum it will be equal 255 else if image pixel value smaller than minimum it equal to zero else if image pixel value is between maximum and minimum it will be equal (pixel value – minimum value) then stored in output RAM block.

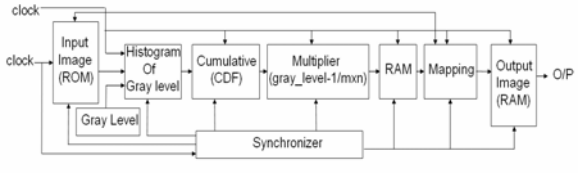

Fig. 3. Block Diagram of Histogram equalization algorithm

Fig 3 shows the Block Diagram of Histogram Equalization algorithm. The ROM is initialized by image. Histogram of gray level block will counts the occurrence of each pixel value (gray value block) in the image in 1-D array. Cumulative block will count each value in the array with previous one, multiplier block will multiply CDF array with constant value, Then mapping block will mapping each pixel in unmodified image to corresponding value in new matrix.

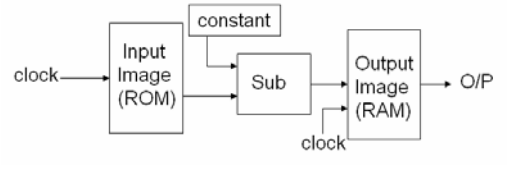

Fig. 4. Block Diagram of Negative image transformation algorithm

Fig 4 shows the block diagram of negative image transformation algorithm, The ROM is initialized by image, sub block will subtract the constant value block

(255) from image pixel value then stored the output in output RAM block.

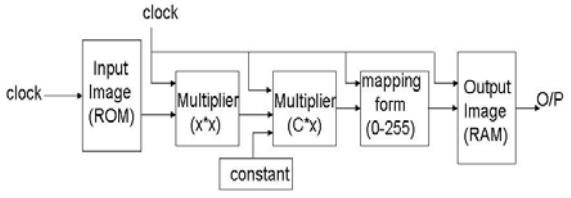

Fig.5. Block Diagram of Power law transformation algorithm

Fig 5 shows the block diagram of power law transformation algorithm, The ROM is initialized by image, Multiplier block will multiply image pixel value by it self then multiply by constant value in multiplier 2 block, mapping block will mapping image pixel value from 0 to 255 by using shift right operation then stored the output in output RAM block.

IV. EXPERIMENTAL RESULTS

The image processing algorithms discussed above were modeled in Quartus II 11.0sp1 Web Edition using Altera environment, all design algorithms was implemented on Cyclon III FPGA based hardware on a 100x100 size grayscale "Lena" image.

Fig 6 shows an example of sample noise image and the resultant enhanced image using median filter

Fig 7 shows an example of a low contrast image and resultant enhanced images after applying contrast stretching, histogram equalization, negative image transformation and power law transformation of image enhancement algorithms

(a) (b)

Fig. 6. (a) A sample noise image. (b) Image after Median filter.

(a) (b)

(c) (d) (e)

Fig. 7. (a) A low contrast Image. (b) Image after contrast stretching (c) image after histogram equalization. (d) image after negative image transformation. (e) image after power law transformation.

Table 1 shows The total power and maximum frequency of implemented algorithms on an image of size 100x100 pixel . It could be noticed that, Power law transformation is the highest Frequency 212.77 MHz while Median filter is lowest frequency 161.29 MHz.

Table 2 shows the compilation report of implemented algorithms on an image of size 100x100 pixel. It could be seen that, The maximum memory usage and logic element algorithm is histogram equalization while minimum logic element and register is negative image transformation.

Table 3 shows the comparison of implementation results of histogram equalization algorithm. It could be noticed that the proposed algorithm like other implemented algorithms with respect to image pixel size and total time of histogram equalization

Table 4 shows comparison of implementation results of median filter algorithm. Proposed algorithm process an image of size 100x100 pixel, it requires 0.186 ms when a clock of 161.29 MHz is used and hence is suitable for real time applications.

Table 5 shows comparison of implementation results of contrast stretching algorithm. Proposed algorithm has low power consumption and number of hardware resources used is lesser than compare to other implementation algorithm.

Table 6 and 7 shows comparison of implementation results of negative image transformation algorithm and power law transformation algorithm . Propose algorithm uses a number of hardware resources lesser than compare to other implementation algorithm (1 Logic Element = 0.5 Slice "2 4-LUTs, 2 flip-flops") [9].

TABLE 1
POWER AND TIMING ANALYSIS OF IMPLEMENTED ALGORITHMS

No.	Design	Maximum Frequency (MHz)	Total Power(m W)
1	Median Filter	161.29	184.27
2	Contrast Stretching	192.68	139.44
3	Histogram Equalization	200.00	321.01
4	Negative image transformation	200.00	119.71
5	Power law transformation.	212.77	158.22

TABLE 2
COMPILATION REPORT OF IMPLMENTED ALGORTIMS

No.	Design	Total logic elements	Total registers	Total memory (bit)	Embedded Multiplier	Total PLLs
1	Median Filter	21 (< 1 %)	128	160,088(31 %)	0	0
2	Contrast Stretching	45 (< 1 %)	24	80,000(16 %)	0	0
3	Histogram Equalization	291 (2 %)	212	249,98(48 %)	0	0
4	Negative image transformation	22 (< 1 %)	24	80,000(16 %)	0	0
5	Power law transformation.	84 (< 1 %)	43	80,000 (16 %)	2	0

TABLE 3
COMPARISON OF IMPLEMENTATION RESULTS OF HISTOGRAM EQUALIZATION ALGORITHM

Implementation	An FPGA Based Real-time Histogram Equalization Circuit for Image Enhancement [3]	A real-time contrast enhancement algorithm for infrared images based on plateau histogram [2]	FPGA Implementation of Image Enhancement Algorithms [1]	Histogram Equalization
Name	Nitin Sachdeva, Tarun Sachdeva.	Wang Bing-jian et. Al	S. Sowmya , Roy Paily	proposed
Device	Altera - Stratix II	Altera – EPIKl00QC208 of Acexlk	xc2vp30-7ff896	Altera ep3c16f484c6
Image size (pixel)	256x256	128x128	100x100	
Time period (ms)	0.263	1.018	--	0.108 ms
Power (mw)	**--**	--	148mW	321.01 mW
Area	--	IO Cells : 25 Block Rams : 32 x1	IO Cells : 32 Block Rams : 16	logic elements : 6,820 memory bits :172,304

TABLE 4
COMPARISON OF IMPLEMENTATION RESULTS OF MEDIAN FILTER ALGORITHM

Implementation	A NEW CONTENT BASED MEDIAN FILTER [4]	AN FPGA-BASED IMPLEMENTATION FOR MEDIAN FILTER MEETING THE REAL-TIME REQUIREMENTS OF AUTOMATED VISUAL INSPECTION SYSTEMS [5]	High Speed Pipelined Architecture for Adaptive Median Filter [6]	Image Enhancement Algorithms using FPGA
Name	GerasimosLouverdis, Ioannis Andreadis Antonios Gasteratos	Miguel Rodríguez, Sánchez Pérez, Gómez-Pulido	D.Dhanasekaran Boopathy Bagan	Propsed
Device	Altera EPF10K200SFC484	HOT2-XL - XC4062XLA-09HQ240C	--	Altera EP3c16f484c6
Clock Freq (Mhz)	55	16	100	161.29
Image Size (pixel)	260×244	640x480	256x256	100x100
Time (msec)	10.6	33.27	0.65	0.186

TABLE 5
COMPARISON OF IMPLEMENTATION RESULTS OF CONTRAST STRETCHING ALGORITHM

Implementation	FPGA Implementation of Image Enhancement Algorithms [1]	Image Enhancement Algorithms using FPGA
Name	S. Sowmya , Roy Paily	Propsed
Device	Xilinx xc2vp30-7ff896	Altera EP3c16f484c6
Image size (Pixel)	100x100	
Clock Freq (Mhz)	199.96	192.68
Total Power(mW)	170	139.44
Time (msec)	--	0.052
Compilation report	Slice Flip Flops :214 LUT : 298 Block RAM : 48	Total Logic element : 45 Total Registers :24 Total Memory : 80.000 bit Multiplier : 0

TABLE 6
COMPARISON OF IMPLEMENTATION RESULTS OF NEGATIVE IMAGE TRANSFORMATION ALGORITHM

Implementation	FPGA Implementation of Image Enhancement Techniques [7]	Image Enhancement Algorithms using FPGA
Name	Karan Kumar, Aditya Jain, and Atul Srivastava	Propsed
Device	Virtex4 xc4vsx35-10ff668	Altera EP3c16f484c6
Image size (Pixel)	--	100x100
Clock Freq (Mhz)	--	200
Total Power(mW)	--	119.71
Time (msec)	--	0.050
Compilation report	Slice Flip Flops :419 4 input LUT : 621 Occupied slice : 500	Total Logic element : 24 Total Registers :22 Total Memory : 80.000 bit Multiplier : 0

TABLE 7
COMPARISON OF IMPLEMENTATION RESULTS OF POWER LAW TRANSFORMATION ALGORITHM

Implementation	FPGA Implementation of Image Enhancement Techniques [7]	Image Enhancement Algorithms using FPGA
Name	Karan Kumar, Aditya Jain, and Atul Srivastava	Propsed
Device	Virtex4 xc4vsx35-10ff668	Altera EP3c16f484c6
Image size (Pixel)	--	100x100
Clock Freq (Mhz)	--	212.77
Total Power(mW)	--	158.22
Time (msec)	--	0.047
Compilation report	Slice Flip Flops :419 4 input LUT : 621 Occupied slice : 500	Total Logic element : 84 Total Registers :43 Total Memory : 80.000 bit Multiplier : 0

V. CONCLUSION

The main goal of this paper is to implement and improved the computational speed of different image Enhancement Techniques on FPGA.. The simulation and synthesis results were obtained and designs are successfully validated using hardware simulation feature of using Cyclone III family chip type EP3C16F484C6 on development kit type DE0. This system has the advantages of being simple, flexible development cost.

REFERENCES

[1] S. Sowmya , Roy Paily, "FPGA IMPLEMENTATION OF IMAGE ENHANCEMENT ALGORITHMS," International Conference Communications and Signal Processing (ICCSP), pp. 584 – 588, Feb. 2011.

[2] Wang Bing-jian,Liu Shang-qian,Li Qing and Zhou Hui-xin, "A REAL-TIME CONTRAST ENHANCEMENT ALGORITHM FOR INFRARED IMAGES BASED ON PLATEAU HISTOGRAM," Infrared Physics & Technology, Elsevier, pp. 77-82, 2006.

[3] Nitin Sachdeva and Tarun Sachdeva, "AN FPGA BASED REAL-TIME HISTOGRAM EQUALIZATION CIRCUIT FOR IMAGE ENHANCEMENT," IJECT Vol. 1, Issue 1, December 2010.

[4] Gerasimos Louverdis, Ioannis Andreadis and Antonios Gasteratosn, "A NEW CONTENT BASED MEDIAN FILTER," 12th European Signal Processing Conference, pp. 1337-1340, September 2004.

[5] Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez and Juan A. Gómez-Pulido, "AN FPGA-BASED IMPLEMENTATION FOR MEDIAN FILTER MEETING THE REAL-TIME REQUIREMENTS OF AUTOMATED VISUAL INSPECTION SYSTEMS," 10th Mediterranean Conference on Control and Automation - MED2002, July 2002.

[6] D.Dhanasekaran and K.Boopathy Bagan "HIGH SPEED PIPELINED ARCHITECTURE FOR ADAPTIVE MEDIAN FILTER," European Journal of Scientific Research ISSN 1450-216X Vol.29 No.4, pp. 454-460, 2004.

[7] Karan Kumar, Aditya Jain, and Atul Kumar Srivastava "FPGA IMPLEMENTATION OF IMAGE ENHANCEMENT TECHNIQUES," Proc. of SPIE Vol. 7502 750208-7, September 2011.

[8] International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 1, A Comprehensive Review of Image Enhancement Techniques , H. K. Sawant, Mahentra Deore.

[9] Guidance for Accurately Benchmarking FPGAs available at: www.altera.com.