FPGA Based Bone Fracture Detector

Hiba-Allah Tariq Abdullah¹ Basma MK. Younis²

¹²Northern Technical University, Engineering Technical College of Mosul, Al-Majmou’a Al-Thaqafiyya Street, Mosul, Iraq

Abstract. Medical imaging was existed as a new domain of research for the treatment of images that can be easily improved, cut and exposed. The proposed technique aims at detecting the objects’ boundaries of noisy images by means of the information on the images of X-ray. Since medical images are made up of noise, and since the edges of the bone make it difficult to detect, edge detection technology has been a key tool in discovering and improving these edges. The edges are changes in the intensity of the lighting of the image. Among several methods for edge detection, the Sobel method is used in this research paper as it is the most famous operator used in the algorithms of detection edges. The algorithms of image processing have been confined to the implementation of slow software due to the limited speed of the processor. As such, a dedicated processor for edge detection was designed and implemented. The results were compared with those arrived at by MATLAB. They were found out to be the same as those of implementation on FPGA which showed the way that Sobel edge detection can assist in determining the location of the fractures in X-ray images. A speedup of approximately 153.8586 can be obtained by Spartan3E FPGA XC3S500E kit system over the software implementation.

1. Introduction

Processing medical images is carried out through the integration of various fields, namely computer, medical, biological and data sciences. It represents one field of science, plays an important role in diagnosing diseases and helps medical practitioners throughout the decision-making process as far as the type of treatment is concerned. A bone X-ray shows the images of various bones in the human body, including the bones of the hand, wrist, foot, knee, shoulder, arm, ankle, leg (shin), elbow, thigh, pelvis or spine and the hip[1, 2]. Fractures as typical bone ailment occur when the bone cannot withstand twisting injuries, falls and outside forces like direct blows. They are chaps in bones and are defined as medical conditions in which a break in the continuance of the bone occurs. It is necessary to detect and treat the fractures in a correct way. In medical applications, edge detection plays a very vital role in terms of image analysis. It represents the basic finding and the detection of sharp changes in brightness or discontinuities in images [3]. There are several operators in edge detection, namely Robert, Laplacian, Prewitt, Sobel and Canny. The suggested work is designed by means of using the Sobel operator. The image is convolved by the Sobel operator by an integer value filter both vertically and horizontally. Hence it becomes easier regarding the computations. The Sobel edge detection operator is utilized in edge definition without the use of any sigma parameter. Accordingly, the image obtained can be more enhanced [4]. This method is implemented by using MATLAB software first followed by VHDL language for comparison purpose. Afterwards, the synthesized design is implemented on FPGA. There are three main reasons that urge researchers to implement their algorithms in the FPGA. Firstly, the nature of FPGA hardware provides greater speed compared to the software-based approaches as it uses different modules of hardware which could be run in a parallel...
manner. Secondly, the reconfigurable nature of FPGA permits the implementation of different algorithms for several times. This gives more flexibility compared to ASICs. Thirdly and finally, all the logics inside FPGA can be rewired, or reconfigured with a different design in accordance with the demand of the designers [5]. Mohammad and Brendan in [6] presents a fast FPGA architecture of first order derivative edge detection methods with a fast pipeline-based architectures that are able to perform edge detection by the use of parallelism on different levels in order to accelerate the operation. The acceleration involves the application of parallelism over convolution masks, modules of edge detection and image intensity values. Added to that, two architectures with different edge detection were proposed, namely one-way and two-way parallel methods, which use Verilog HDL for Cyclone IV FPGA. The results show the speed-up of 460 and 920 for the two parallel architectures respectively. Mahendran and Santhosh in [7] proposed a fusion-classification technique for the purpose of automatic detection of the long bones, and the leg bones in particular (Tibia bones). The proposed system involves four steps, namely preprocessing, segmentation, feature extraction and bone detecting ion. The technique uses an amalgamation of image processing techniques to detect fractures successfully. Three classifiers, viz. Feed Forward Back Propagation Neural Networks (BPNN), Classifiers of the Support Vector Machine (SVM) and the Classifiers of Naïve Bayes (NB) have been used throughout the fusion classification. The results of the various experiments have proved that the proposed system has shown significant improvement concerning the rate of detection and classification speed.

Wint et al. in [8] developed the recognition of lower leg bone (Tibia) fracture types using various image processing techniques. They aimed at detecting fracture or non-fracture and classifying the types of fracture of the lower leg bone (tibia) in X-ray image. The tibia bone fracture detection system was developed by three main steps, namely pre-processing, feature extraction and classification to classify the types of fracture and locate fracture locations. In pre-processing, Unshrap Masking (USM), which is the sharpening technique, was applied to enhance the image and highlight its edges. The system produced 82% accuracy for classification fracture types.

Dhanabal et al. in [9] presented a project implementing a co-processor for the purpose of image processing. The coprocessor was modelled for images edge detection. The Edge detection algorithm was implemented on an FPGA, where the inherent parallelism provided better performance. The architecture was like the ARM processor, which acted as the master that had the images to be processed. The image was transferred by ARM to the FPGA for the purpose of processing. After processing, it would be displayed by the FPGA through a VGA display. The image which was sent by ARM was stored in an instantiated memory in FPGA. Edge detection core was implemented in FPGA. Afterwards, the image was read from memory, processed and then the processed image was stored in the memory. The VGA controller was designed to read the processed image from the memory and display it. Sobel Edge detection algorithm has been used for the purpose of edge detection for images and it is efficient in getting smooth edges in addition to being less sensitive to noise.

2. Sobel edge detection technique

Sobel Edge Detection algorithm has been proved to be ideal for edge identification [1]. It uses an algorithm of two stages to identify a wide spectrum of edges. As an input, it takes a grey scale image (X-ray image), then processes it and finally produces the output, which shows intensity discontinuities. The operator uses two $3 \times 3$ filters that are convolved with the image to get the edge or high passed image by calculating the approximations of the derivatives; one for horizontal, and one for vertical. This $3 \times 3$ spatial kernel for Sobel Operator is shown in Fig. 1 [1].

![Masks used by Sobel Operator](image)
The edge value is determined by the following regulation:

\[
\text{Edge Magnitude} = \sqrt{G_x^2 + G_y^2}
\]  

(1)

where \(G_x\) represents the row mask whereas \(G_y\) is the column mask.

3. Practical System

The proposed system, as described in the block diagram of Fig. 2, consists of several steps. Firstly, a JPG image file is read from the hard disk via the MATLAB software and then the colored image is converted to a gray level using simple specific statement. After that and with a simple action, the resulting image is saved into a COE file. At this point, a dual port BRAM is generated using VHDL. First port is for saving the gray image COE file of size (128 × 128 × 8) which needs a memory of 16384 byte. Second port BRAM is of the same capacity, but the input and output data lines are 6-bit (16348 × 6), depending on the number of bits for each VGA channel. The next steps starts by applying Sobel filter on the input image by convolving the Sobel masks with a window of 3 × 3 from the image to calculate \(G_x\) and \(G_y\) values. To simplify, the hardware absolute value for the edges magnitude is used instead of that of Eq. (1).

\[
\text{Edge Magnitude} = |G_x| + |G_y|
\]  

(2)

Fig. 2: Block Diagram for the Proposed Bone Fracture Detector

To display the original and the resultant image on the monitor in the specific location, a VGA controller is used and programmed to work with 25MHz to obtain an ideal result. The project was carried out using the SPARTAN 3E-XC3S500, which includes the VGA port of the graphics display on the VGA monitor. The latter consists of 3 channels of color (RGB). Each channel contains only 1-bit, where any image is displayed in only eight colors. This feature is unsatisfactory and insufficient to display the images of the project as it needs to display the original image before applying the Sobel filter. Since the image Gray level is impossible to display high accuracy according to the characteristics of the above port, it should build a special port to display graphics with high accuracy as each A 6-bit channel means that the total is 18-bit. In this case, it is possible to display any image with a high resolution. It is also very acceptable with a color level (216) or (256 K color / pixel) for the color image. This is so because in case a grey level image was used, the resolution of this image would be (26) or 64 colors grey level. Such accuracy is sufficient to display the original image and image as shown in Fig. 10 to Fig.12. Fig. 3 illustrates schematic diagram of the new VGA port and Fig. 4
illustrates (6-bit, 6-bit, 6-bit) RGB VGA port, while the part of the UCF file related to the additional VGA port is shown in Fig. 5.

**Fig. 3:** Schematic diagram of the new VGA port

**Fig. 4:** (6-bit, 6-bit, 6-bit) RGB VGA port
The output of equation (2) is calibrated to be suitable for the 6-bit VGA port. This result is saved at the corresponding location into VRAM. Fig. 6 illustrates the System flowchart.

Fig. 5: Part of the UCF file related with the additional VGA port

Fig. 6: System Flowchart

The implementation time of Sobel edge detection algorithm that is detailed in the following steps,
is accurately adopted:

1. **Sobel step**: In this step, the starting address of the first window of the image is selected for the purpose of the candidate through I and J where the first window starts with the address (0, 0) and ends with the address (7DH, 7DH). The length of the image is 127 and 127 pixels wide. So it must be ended with (max_I-2, max_j-2). The sum variables (s1, s2) are initialized and the length of each variable is set to 18-bit before moving to the second step (Init_0).

2. **Init_0 step**: The second step relates to the initialization of the first element of the filter and repeats 9 times any size of the filter. It also converts the pixel address from the one-directional array to the real location in the image in a two- dimensional array and sends the address to (Image block RAM: 16384). The moving to the third step is (Init_1).

3. **Init_1 step**: Reads the pixel value from the image after specifying its address by the previous step and converts the value to "Signed NO." It also converts the filter values to "Signed NO." The values of the window consisting of 9 positive values are multiplied with the candidate values kx, ky); 9 positive and negative values for each filter and uses 9-BIT * 9-BIT embedded multiplier to obtain the first multiplication output for both filters. The process is repeated nine times (size of the filter and the window). The output is saved in 18-bit accumulators and is back to the previous step to complete the multiplication of the values of the rest of the candidates. It then moves to step four.

4. **Init_2 step**: In this step, the absolute values of the accumulators (s1, s2) are added to obtain the final result. On the completion of all the multiplication operations, the outputs are calibrated to be from (0 to 255) and converted from 18-bit to 6-bit by taking (B7 to B2) from the result in accordance with the block RAM of the image components. After the filter is made and formed of 6-bit data bus, all the previous steps are repeated so that the filter can be done on all the components of the image which is (126 x 126), which is equal to image row size-2 by image col size -2 ((128-2) x (128-2)) depends on the size of filter (3x3).

4. **Results and discussion**

Throughout the experimental part of this work, X-ray images are given as an input to MATLAB. Samples of these images and resultant images with a high resolution are shown in Fig. 7.

![Fig. 7: Sample from MATLAB results for the proposed bone fracture detector](image_url)

As illustrated in the process flow chart of Fig. 6, the image is first converted from RGB to gray. This reduces the amount of data in the image. Then a COE file is generated using MATLAB and saved on FPGA block-RAM. The code for Sobel edge detection is written in VHDL, synthesized and simulated using Xilinx ISE 10.1. Some important signals are shown in the simulation result of edge detection, see Fig. 8.
5. Execution time using MATLAB

In MATLAB programming, the execution time of Sobel filter on 128*128 pixels image is (0.671361) seconds as shown in Fig. 9 using 2.7 GHz frequency Core i7 CPU 4th generation and 4G-byte RAM.

Meanwhile, the execution time of the same filter using VHDL is calculated by using the following equation:

\[(1): \text{Sobel} + ((1):\text{Init}_0 + (1):\text{Init}_1)*(9): \text{filter size} + (1):\text{Init}_2 + (1): \text{else}*(126*126): \text{Windows number}+1: \text{else}=333397 \text{ clock cycles}\]

\[F=76.406 \text{ MHz} \text{ (For implementation process, while display process works in 25MHz). Execution time}=333397 \text{ Clock}/76.406 \text{ MHz}=0.004363 \text{ second}\]

The implementation of the Sobel filter using VHDL is faster than using MATLAB in (153.8586) times. Yet, the output results are the same in the two previous implementations as they did not require any complex computations in the floating point or the fixed point; and all computations
were on integer values. The utilization of FPGA hardware resources and the maximum operating frequency is shown in Table 1.

Table 1. Summary table.

| Device type                  | Percentage area occupied |
|------------------------------|--------------------------|
| Number of Slices             | 5%                       |
| Number of slice Flip-Flop    | 2%                       |
| Number of 4 input LUTs       | 5%                       |
| Number of bonded IOBs        | 14%                      |
| Number of BRAMs              | 70%                      |
| Number of MULT18 x 18SIOs    | 10%                      |
| Number of GCLKs              | 8%                       |
| Maximum frequency (MHz)      | 76.406                   |

Fig. 10 to Fig. 12 represent the implementation of Sobel filter on three images using Spartan 3E-XC3S500 FPGA. Fig. 13 illustrates the 3E-XC3S500 FPGA kit.

Fig. 10: Edge detected bone_1 displayed with VGA Controller

Fig. 11: Edge detected bone_2 displayed with VGA Controller
6. Conclusions
Sobel filter is an important filter for digital images. It used to detect edges in images, especially in high frequency. In this research, the system of edge detection using the Sobel filter was built using the Spartan3E-XC3S500 FPGA. Since image-filtering process does not require complex calculations such as multiplication and division of the floating point or fixed point values, the resultant image of the system is exactly the same as that produced by any program written in high-level language. The color image must be converted to a gray scale image in hexadecimal format, saved in a COE file, read by an ISE environment and then processed using VHDL. The results were as expected and achieved a speedup of about 161 over those obtained by software, which can used with the real time systems that needs a much more speed.

7. References
[1] Gonzalez, Rafael C, 2008 Digital Image Processing Pearson Education, Inc., publishing as PrenticeHall
[2] Tamisiea D.F. 2008 Radiologic aspects of orthopedic diseases", Mercier LR, ed. Practical Orthopedics, 6th ed. Philadelphia, Pa: Mosby Elsevier; Chap 16
[3] Meenal Bhatia, Swati.R.Dixit 2016 Edge Detection for X-Ray Image using LabVIEW International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 3
[4] Nancy J. and Nathan S. ”Bone Fracture Detection Using Edge Detection Technique”2017 Springer, Singaporem, © Springer Nature Singapore Pte Ltd. 2018, vol-2
[5] Mohammad Shokrolah Shirazi, Brendan Morris 2015 Fast Edge Detection Architecture using Different Levels of Parallelism on a FPGA International Journal of Computer Applications, Volume 113 - No. 13
[6] Sheetal D. Bhyar, Tarnnum Pathan, Amit D. Landge 2016 *Design and Implementation of Sobel Edge Detection technique using VHDL* International Journal of Advanced Research in Science, Engineering and Technology, Vol. 3, Issue 5

[7] S.K. Mahendran, S. Santhosh Baboo 2011 *An Enhanced Tibia Fracture Detection Tool Using Image Processing and Classification Fusion Techniques in X-Ray Images* Global Journal of Computer Science and Technology, Volume 11 Issue 14 Version 1.0 August 2011

[8] Wint Wah Myint, Khin Sandar Tun, Hla Myo Tun 2018 *Analysis on Leg Bone Fracture Detection and Classification Using X-ray Images* Machine Learning Research, Journal of Machine Learning Research, Vol. 3, No. 3, 2018, pp. 49-59. doi: 10.11648/j.mlr.20180303.11

[9] Dhanabal R, Bharath Vi, S. Kartika 2013 *Digital Image Processing Using Sobel Edge Detection Algorithm In FPGA* Journal of theoretical and applied information technology, 10th December 2013. Vol. 58 No.1 © 2005 - 2013 JATIT & LLS. All rights reserved.