

Design and Implementation of Efficient Edge Detection Algorithm for Image Processing Applications

Meghana B Ganesh, Keshava. A

¹M.Tech Student, ²Assistant Professor

¹VLSI Design and Embedded Systems Bangalore Institute of Technology Bangalore, India

²Dept. of Electronics and Communication, Bangalore Institute of Technology Bangalore, India

Abstract - Image processing is essential for medical, remote sensing, microscopic imaging, etc applications. One of the main strategies for object detection is the edge detection algorithm. Digital image processing techniques such as segmentation and extraction are used to perform this edge detection. Real-time embedded system research for digital image processing is flourishing. Edge detection is essential for real-time picture processing and is extremely helpful for verifying design recognition, texture extraction, and full picture segmentation. The limitations of processor speed mostly effect how image algorithms are implemented in software coding, hence advances in VLSI Technology are employed to control this behaviour. In the current project, the sobel edge detection technique for Field Programmable Gate Array devices is implemented using the Hardware Description Language (HDL). The study demonstrates that using the hardware Sobel edge operator produces a substantially faster speed than using simulation software. The suggested approach primarily makes use of an Improved hardware architecture, and the images/pictures are sent from a PC to an FPGA computer via serial UART communication. As follows, the primary objective of this project is to enhance the speed and to efficiently scale down the hardware resources.

Keywords: Image Processing, Edge Detection, Sobel operator, Hardware Description Language, Field Programmable Gate Arrays.

I. INTRODUCTION

Image processing is the process of enhancing and extracting data from an image. Techniques for digital image processing are helpful when managing digital images with computers. Image processing is a category of signal processing where the images are being fed as an input and we extract data/information/ features/characteristics which are allied with the input image. In the present day scenario, image processing is one of the most expeditiously advancing technologies. Many research fields in Engineering and Computer Science disciplines are continuously using Image processing technique.

There are primarily two types of image processing. They are digital image processing and analog image processing, respectively. When we need to study physical copies, such photographs and paper printouts, analog image processing is typically used. When interpreting images using these visual methods, image analyst utilize several interpretational principles. By using computers and related software tools, digital image processing is useful for managing digital images. These image processing techniques consist of three main steps: pre-processing, augmentation and presentation, and information extraction. In this project, Verilog HDL and MATLAB are used

to build image processing techniques. MATLAB is used as a reference for verifying the output of HDL. The architecture for image processing algorithms mainly color conversions, Edge detection, Image binarization, pixel summation is designed implemented and verified using Verilog HDL and MATLAB. Simulation tools which are used are Modelsim and Xilinx ISIM. Image processing algorithms vary. Edge detection is crucial to real-time picture processing. It is extremely beneficial for obtaining texture extraction, verifying entire picture segmentation, and achieving design identification. Generally Edges in the images are the regions with high disparity in intensity, every pixel will have a different intensity compared to the next pixel, so the whole process of Edge Detection substantially scale down the amount of information present in the image and refine the superfluous information, but also secures the necessary structural features of the image.

II. EDGE DETECTION

Edge Detection is an essential stage in analyzing images, computer vision techniques and their processing, discovering image patterns. The Edge Detection can be defined as a process where the edges and the acute changes of image capture will eventually result in extensive changes in image. It is primarily used for extraction and Feature Detection, where discovering the co-ordinates which has sharp parity in the brightness and discontinuity is of high importance. Most of the Edge Detection Algorithms are helpful in discovering edges of object entities present in an image [1]. Edges occur between two separate areas of object entities in the image. There are many algorithms to carry out this functionality. But most of them can be grouped as derivative based in which the algorithm considers the initial or second derivative for every pixel. They can also be grouped as gradient based in which a gradient of successive pixels are placed in x, y coordinate plane. The most often employed technique for image edge processing is typically sobel edge detection algorithm. There are various other algorithms and each one is utilized based o their suitability to the applications. A clear analysis is made and the appropriate edge detector is chosen. These edge operators are very important to extract the features of the picture and to analyze them. Thus it is vital in the processing of images and the respective applications.

Sobel Edge Detection

For the application of Sobel Edge Detection, the hardware analysis of FPGA is considered. This type of operator can be integrated in a simple manner and the huge amount of time can be saved [2]. A VHDL/Verilog based Edge Detector uses a hardware where the architecture has a greater speed image processing system. Edge Detection finds sharp image differences. Pixel power changes cause the differences, that in turn figures out the perimeters of the object entity in a broad spectrum. Edges for an Image are usually considered as the prominent characteristic of the image which provides significant information. With this kind of information, the pictures are videos are identified at every prominent stage. The main aim of this process is the recognition of abrupt adjustments within the image brightness by Edge Detection Algorithm.

The Sobel based image detector is mainly equipped here to identify the edges of a desired image. There are mainly two 3x3 Kernels which are distributed in x and y co-ordinates. We use 3x3 kernel matrix, one matrix for each direction of x and y co-ordinates. The gradient which is present in the x co-ordinates has negative numericals on the left part and we have the positive numericals on the right hand side. We are securing a minor amount of centre pixels. In a similar way, the gradient which is present in the y co-ordinates has negative numericals on the bottom side and positive numericals on the top side. This can be seen in the figure 1. The Sobel Operator method can be normally implemented for Image Segmentation without any human interaction and involvement. The Sobel Operator is highly instantaneous for the implementation. It also generates the exact same result everytime it executes. Because of this, the Sobel Operator is considered as a reliable method for detecting the edge for Image Segmentation.

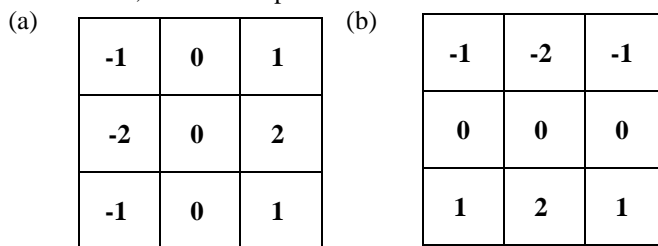


Figure 1. The 3x3 Kernels (a) X-Direction (b)Y-Direction

For the sobel operator to measure approximation of a derivative that provides edges or high-pass images, two 3x3 kernels are required, one for horizontal and the other for vertical. The kernels are instructed to respond to the pixel grid in both the vertical and horizontal directions, with each kernel taking into account one among both crosswise directions in turn. These kernels could be clearly integrated to the input of the image in order to provide distinct measures in each direction.

III. PROPOSED METHODOLOGY

This methodology takes into account the hardware analysis of the FPGA in order to create the real-time, Sobel-based picture edge detection system. The sobel operator's primary benefit is that, in comparison to other existing operators, noise has less impact and relatively few components are needed. This operator is quick to implement and saves time. The hardware architecture of a VHDL/Verilog edge detector uses a fast image processing method.

Hardware Model

The hardware pattern for the Sobel operator is shown in figure 2. The nine eight-bit pixels that are input to the Sobel module are designated by P0, P1, P2, P3, P4, P5, P6, P7, and P8, as illustrated in the diagram. The aforementioned module has shift registers, module operators, and signed subtractions.

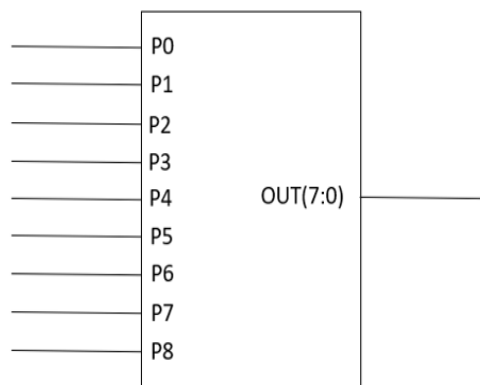


Figure 2. Hardware design

The implementation of the proposed design for the sobel edge detection algorithm involves a number of modules to be integrated. In this, the testbench gives the image data to the design code as input and the operation is carried out. The resultant is observed through GUI. The system of the proposed design is as described in figure 3.

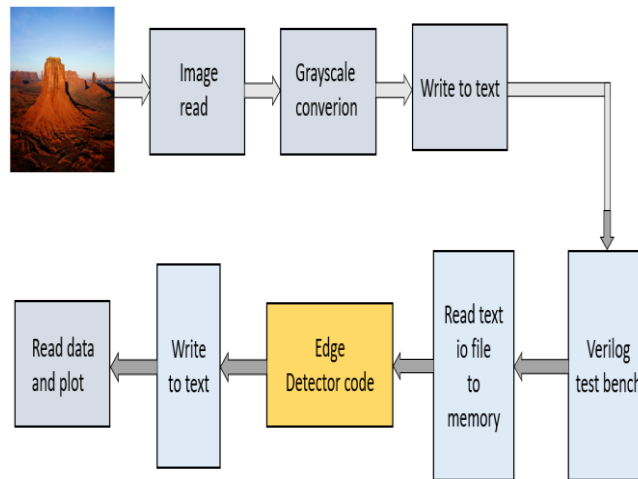


Figure 3. System design

The final adder block in the hardware design is made up of 11 bits (that is, ten bits for data, with a extreme number of 4*255, and here the eleventh bit designating a signal bit). 11 bits make up the ultimate resulting value. The output data, which consists of eight bit complete pixels for the produced picture output, that is compared with a maximum value of 255. This category is characterised primarily by its scalability, speed, and moderate cost. The implementation is as described.

A. Image Acquisition

It's the method of obtaining an image from a source typically a hardware-based source, for processing in image processing. Image is input into the MATLAB tool using the imread command. Then the RGB to Grayscale conversion flow is as described in the figure 4. The image which is in RGB format needs to be converted into gray scale format that makes the processing steps much simpler.

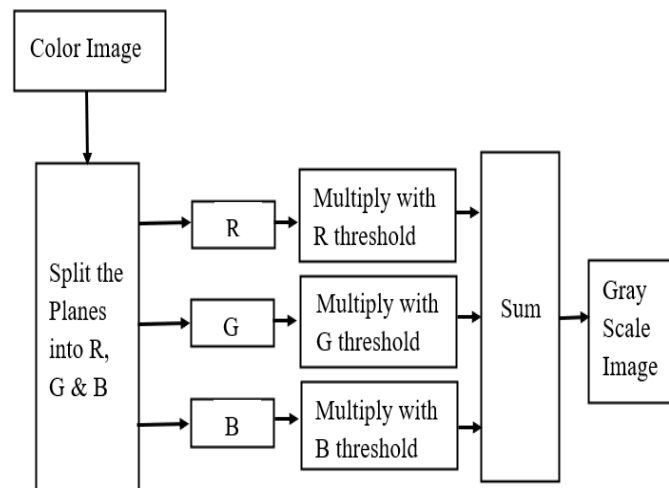


Figure 4. RGB to Grayscale conversion

Numerous procedures, including the average approach, weighted or luminosity method, decomposition method, and desaturation method, are available to convert RGB to grayscale. The luminosity approach is the most extensively used and accepted algorithm among them, and it is utilised by many image processing programmes as well as the MATLAB. The aforementioned method is based on the perception of brightness in the human eye, which is strongest for green, followed by red, and then closely followed by blue.

$$Y = 0.281 * R + 0.562 * G + 0.093 * B$$

Consequently, the luminosity algorithm, which is described by the equation, is employed in this project to convert RGB to grey.

B. Image Segmentation

Image segmentation in computer vision is the process of dividing a digital picture to segments (like sets of pixels, which are commonly referred as image objects). Image Segmentation's goal is, to make it easier and/or transform an image's representation into one that is clearer and easier to understand. Typically, Image segmentation identifies objects and boundaries (lines, curves, etc). The result of this Segmentation is either a mixture of contours drawn from the image.

C. Thresholding Operations

A structuring element will be applied via morphological operations to image that is input so as to produce a resultant image of identical size. Every pixel's value in a morphological operation is based on how well its neighbours correlate with the corresponding input image pixel. With the selection of the neighbour's shape and size, we may create a morphological operation which is responsive to certain shapes in the input image data. Typically, Separating "object" or foreground pixels from background pixels aids image processing.

D. Edge Detection

The edges of an image are those regions which correspond to object boundaries and those regions where there are abrupt changes in image brightness. It is derived from the picture function features in a pixel's vicinity and acts as a kind of attachment to each particular pixel. Verilog is the hardware description language that is utilized in the design of the Sobel edge detector.

The gradient scale is,

$$|G| = \sqrt{Gx^2 + Gy^2}$$

Typically we have,

$$|G| = |Gx| + |Gy|$$

The kernels are combined to calculate the every point's absolute gradient magnitude,

$$|G| = |(P1+2*P2+P3) - (P7+2*P8+P9)| + |(P1+2*P4+P7) - (P3+2*P6+P9)|$$

P1	P2	P3
P4	P5	P6
P7	P8	P9

Figure 5. Pseudo convolution kernels

This gradient computation is done using the verilog coding that includes the hardware components in building up of the system. The various verilog modules involved in the system design are buffer, gradient calculator, square root calculator, binary calculator, median and threshold calculator, etc.

IV. RESULTS

The simulation of the designed sobel edge detection algorithm is carried out on Xilinx ISE 14.5 platform. The integrated ModelSim SE simulator enables the analysis of the generated waveform. This sobel edge detection algorithm is evaluated with the help of MATLAB R2018b through the GUI. This enables the analysis of the color image, grayscale image and edge detected image. The GUI output of grayscale and binary conversion is as described in figure 6. The image that is input is converted to grayscale image, whose pixels values are written into a text format.



Figure 6. Grayscale conversio

These pixel values are then fed to the verilog co and the sobel edge detection method is performed. Based on this technique, the edges are detected and is indicated. The edge detection is then observed through the edge detected image through the integration of matlab software. This enables the edge detected image. The matlab user interface application is developed to test the entire system. To test the core, a co-simulation in verilog and matlab is run. The grayscale image is next subject to the edge detection technique. The edge detected image is produced by the Verilog Sobel edge detector code as illustrated in figure 7. This represents the original image, grayscale image and the edge detected image.

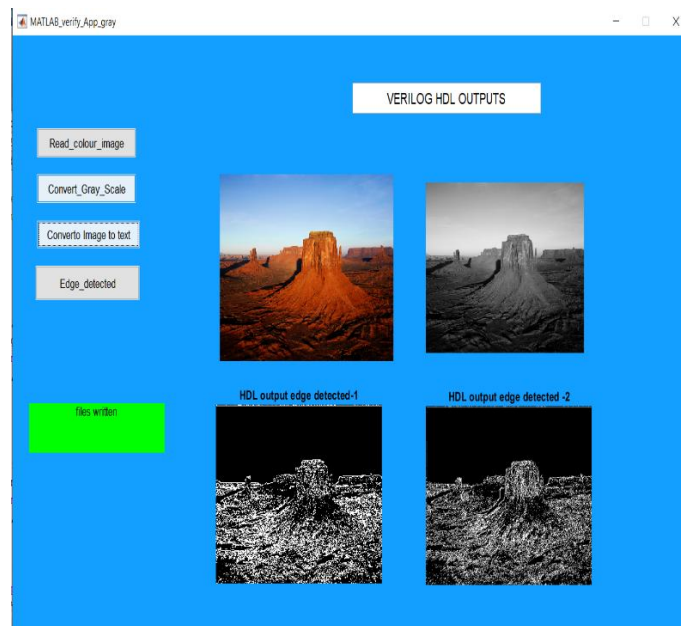


Figure 7. Hardware based sobel edge detected image

Synthesis is the transformation technique of a hardware description language design toward a gate-stage netlist, in consideration to the optimization constraints mentioned. It auto converts the verilog logic into a gate-stage realization and maps these design models to various technologies that is targeted, as per the target libraries. This synthesis process is carried out for the proposed sobel edge detector in FPGA flow for the Virtex7 family and the device utilization and timing parameters are obtained as per the architecture design.



Figure 8. RTL Schematic

The ModelSim tool is integrated with Xilinx ISE and the simulation of the design is done to observe the working of the components. The verilog design of sobel edge detection simulation results are as shown in the figure 9.

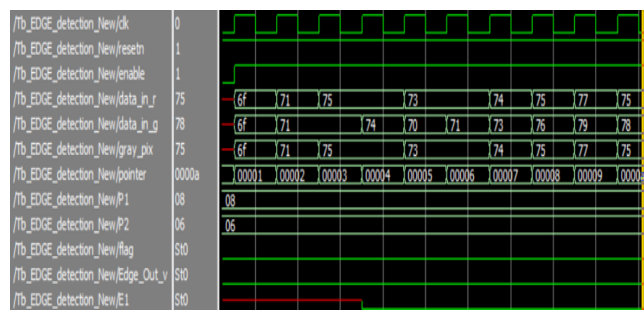


Figure 9. Simulation output

The simulation time of hardware based sobel edge detection is compared with matlab simulation and is as shown in the table 4.1. It is observed that the hardware based design is more efficient.

Table 1. Simulation times

Timing value in MATLAB	Timing value in Hardware based design
---------------------------	--

469 μ s	5.667 ns
-------------	----------

The table 2 shows that the proposed architecture improves the execution time compared to other designs and also optimizes the number of resources which in turn reduces the power.

Table 2. Performance comparison

	Existing methods	Proposed method
Delay	10 ns	5.66 ns
Frequency	100 MHz	176.46 MHz

The designed edge detection algorithm can be efficiently utilized in image processing applications. The processed images can be used in medical science, satellite imaging, robotics, etc. They are used for feature extraction and image recognition. The following shows the sobel edge detected image that can be used in medical diagnosis.

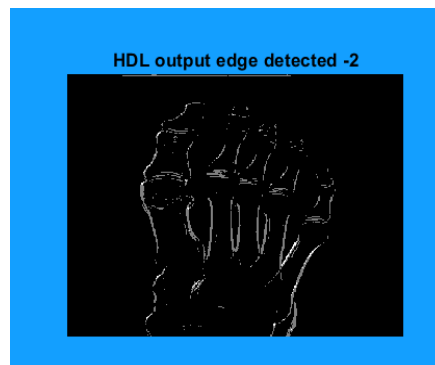


Figure 10. Sobel edge detected image

Also, the edge detection is used in license plate recognition, authentication, etc to detect something in aerial views or cracks in the buildings, and to do authentication through biometrics.

Table 3. Resource usage

Parameters	Number of available blocks	Utilization in proposed architecture
Number of Slice Registers	408000	<1%
Number of Slice LUTs	204000	<1%
Number of LUT-FF pairs	411	13%
Number of bonded IOBs	600	10%

The table 3 shows the hardware component utilization on the Virtex7 FPGA device.

V. CONCLUSION

The application of sobel edge detection algorithm on a hardware platform proves to be very time efficient. The computational delay is reduced and is architected to optimize the resource usage. The aforementioned architecture uses less hardware resources than other edge sensor techniques. Experimental evidence supports the comparison of these hardware-based image processing systems to software-based image processing in terms of speed and activity. The Sobel edge detector algorithm supports the FPGA devices in relation to the excellent FPGA technology. It is efficient, cost effective and also less time consuming. The system is scalable, fast and cost effective. The system's complexity and computational time are lowered as a result of the design.

VI. FUTURE SCOPE

The advanced work can be done in the improvement of the design with pipeline structure enabling greater efficiency. The field of edge detection is the subject of extensive research. There is lot of scope for the edge detectors in image processing, thus improvisations tend to attain greater stability. The pipelined structure can also be used in computer vision systems with less powerful hardware.

REFERENCES

1. Yang Cheng, Bingjie Li, "Image Segmentation Technology and Its Application in Digital Image Processing", IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), 2021
2. Deeba K, Janarathanan R, Menaka R. FPGA implementation of low power and high speed image edge detection algorithm[J]. Microprocessors and Microsystems 75 (2020)

3. Coronel L, Badoy W, Namoco C. Identification of an efficient filtering-segmentation technique for automated counting of fish fingerlings[J]. International Arab Journal of Information Technology, 2018
4. Tai K. The application of digital image processing technology in glass bottle crack detection system[J]. Acta Technica CSAV (Ceskoslovensk Akademie Ved), 2017
5. Raj S M A, Jose C, Supriya M H. Hardware realization of canny edge detection algorithm for underwater image segmentation using field programmable gate arrays[J]. Journal of Engineering ence and Technology, 2017
6. G. Anusha, T. JayaChandra Prasad, D. Satya Narayana., "Implementation of SOBEL Edge Detection on FPGA" International Journal of Computer Trends and Technology- volume (3) issue (3)- 2012
7. Raman Maini, Dr. Himanshu Aggarwal, "Study and Comparison of Various Image Edge Detection Techniques", International Journal of Image Processing (IJIP), Volume (3), 2009
8. D. T.Saegusa, T.Maruyama, Y.Yamaguchi, —How fast is an FPGA inn image processing?!, IEICE Technical Report, Vol.108. nNo.48, pp.83–88, 2008t
9. R.Gonzalaz, R Woods, "Digital Image Processing", New Jersey: Prentice –Hall 2002
10. Ziou and S. Tabbone, "Edge detection techniques - an overview," International Journal of Pattern Recognition and Image Analysis, vol. 8, pp. 537–559, 1998
11. S. Sarkar and K.L. Boyer. On Optimal Infinite Impulse Response Edge Detection Filters. Ieee Transactions on Pattern Analysis and Machine Intel ligenca, 13(11):1154 1171, 1991
12. P. Bouthemy, "A maximum likelihood framework for determining moving edges", IEEE Trans. Patt. Anal. Machine Intell., vol. 11, pp. 499-511, May 1989
13. J. F. Canny. "A computational approach to edge detection". IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no. 6, pp. 679-697, 1986
14. W. E. L. Grimson, "Computing stereopsis using feature point contour matching", in Techniques for 3-D Machine Perception (A. Rosenfeld, Ed.), New York: Elsevier, pp.75-111, 1986
15. T. Poggio, H. Voorhees, and A. Yuille, "A regularized solution to edge detection", Tech. Rep. MA, Rep. AIM-833, MIT Artificial Intell. Lab., May 1985
16. J. E. W. Mayhew and J. P. Frisby, "Psychological and computational studies toward a theory of human stereopsis", Artificial Intell., vol. 17, pp. 349-385, 1981