ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Hardware Implementation of Edge Detection Algorithm-A Review

Kavitkar S. G.[*1], Paikrao P. L.[2]

[*1,2]Electronics and Telecommunication Engineering, Amravati University, Government College of Engineering, Amravati (M.S.), India

sgkavitkar@gmail.com

Abstract

Edge is one of the most fundamental and significant feature of image. It helps us to analyze, infer and take decision in various applications. Several edge detection algorithms are available; each one has significance based on subjective application. In this paper edge detection algorithm is proposed to be implemented over Field Programmable Gate Array (FPGA) for real time applications. FPGA has large internal memory, embedded multipliers and it offers parallelism. Hence, provides platform for processing real time algorithm with substantially higher performance than programmable Digital Signal Processor (DSPs) and microprocessor. Hardware implementation yields dedicated system which can serve for many real time image processing applications. For implementation 'Xilinx System Generator' (XSG) tool of Matlab may be used that generates net-list and bit-stream file by simplified approach.

Keywords: Edge detection algorithm, FPGA, XSG, Matlab.

Introduction

One of the main objectives of image processing is to improve the quality of the images for human interpretation. Low level image processing operations like image segmentation using edge detection and feature extraction helps us to analyze, infer and take decision in various applications. It focuses on processing an image pixel by pixel and over its neighborhood. Image processing operations can be applied to the whole image or its parts. Feature extraction algorithms are essential for many computer vision applications such as object recognition, real time flaw detection, meteorological applications, etc. It is important basis for the field of image analysis such as image target area identification, extraction and region formation. Edge detection outlines the points in an images corresponding to sharp intensity variations or discontinuities. It is widely used in image segmentation, image recognition, texture analysis, etc.

Several edge detection algorithms are available; it should not only find abrupt variations in intensity values but also should identify their exact location. Among all edge detection algorithms, Canny edge detection algorithm is most widely used feature extraction algorithm because of its reliable performance with noisy images. The discontinuities are abrupt variations in pixels intensity which characterize boundaries of objects in a scene structure. This process significantly reduces the image file, while preserving its most important structural features. Edge detection is considered as the ideal algorithm for images that are tainted with white noise subjected to features extraction.

In recent years FPGAs (Field Programmable Gate Array) have become superior platform, with high-speed parallel computing capacity. FPGA is a 'fine grained device'. FPGA is rich source of high speed multipliers, adder, and memory; in the design process they can be directly called. Subsequently it is easy to implement complex convolution over this platform. FPGA based design offers an advantage of short Turn Around Time (TAT) and small Non Recurrent Expense (NRE). While implementing an algorithm over hardware platform, in the traditional approach, user needs to emulate the floating point algorithm in HDL code. Further, it can be tested and verified by repeated functional simulations, post simulation processes and finally generate bit stream file. Xilinx introduces the system modeling tool 'Xilinx System Generator'.

It has simplified the development process. User only needs to follow the design requirements to generate executable bit stream file, test files, etc. It removes the complicated simulation comparison and verification process, speed up the FPGA based system. ISE is integrated FPGA development software of Xilinx. The platform integrates design entry, logic synthesis, place and route. System Generator tool can be used in feature extraction of images, it is available as a module in Matlab Xilinx. Edge detection algorithm will be designed and simulated in Matlab (Simulink) for generation of top-level file in ISE environment. Xilinx ISE software is

ISSN: 2277-9655
Impact Factor: 1.852

configured with Matlab, so that simulink tool boxes for Xilinx are made available in Matlab (simulink) library. Hence, design can be created in simulink environment which needs to be simulated functionally to generate HDL netist[1].

Related Work

An extensive work is done in the field feature extraction for real time image processing. Edge detection being fundamental step for any image processing operation, it provokes great interest for the scientific community. Edge detection alters the image for human interpretation and information extraction; in various fields such as in biomedical application, satellite imaging, traffic control, land acquisition, etc.

Literature reveals that hardware implementation of edge detection algorithm is necessary to meet real time and speed constraints. In [2] high throughput rate is achieved for FPGA based implementation of canny edge detection algorithm exploiting the FPGA resource at its peak. FPGA based object detection utilizing edge information in [3] proposed to offer high speed, energy efficient design. Simplified approach for hardware implementation using 'Xilinx System Generator' is proposed in [4] for vehicle edge detection for traffic analysis. Comparative analysis for software and hardware based video image edge detection is proposed [5] hardware implementation tends to yield faster results. Machine implementation and analysis of pattern is proposed using canny edge detector algorithm in [6] satisfactory results are obtained. Sobel edge detection algorithm is implemented [7] over hardware platform for optimized volume. Hardware co-simulation for video edge detection is implemented over Xilinx Virtex 5 board in [8] using XSG. Real time FPGA based tracking and counting system for people is proposed in [9] Spartan 3E is used as hardware platform.

Edge Detection

The edge is characterized by its length, slope angle and coordinate of the slope midpoint. Edges are caused by variety of factors such as surface normal discontinuity, depth discontinuity, surface color discontinuity, illumination discontinuity. Basically there are two types of edges a ramp edge, where the intensity values change slowly and a step edge or an ideal edge, where the intensity values change abruptly. Edge detection can be achieved by several approach, the majority may be grouped into two categories, Gradient (Approximation of derivative) and Laplacian (Zero crossing detectors) based approach. Edge detection operator based on gradient approach determines the level of variance between neighboring pixels. It is premeditated by forming a mask over center pixel whose properties needs to be altered. Pixel is classified as an edge if an estimated area of matrix overcomes the threshold value specified. Examples of gradient based edge detectors are Prewitt, Roberts and Sobel operators. Laplacian operators are the second order derivative operators. Canny edge detection is optimal edge detection method. In the gradient based approach strength of the kernel is computed in horizontal and vertical direction [1]. Derivative operator detects abrupt variation in intensities of an image. The abrupt variation points correspond to the edges in an image.

First derivation is computed using gradient operator. The second order derivative is very sensitive to noise present in the image. Hence, second order derivative operators are not usually used for edge detection. The second order derivative operators are used for extraction of some secondary information, such as precursor of second order derivative determines location of point along darker side or brighter side of the image. Several edge detection algorithms are available based on derivative approach. Derivative approach determines the local gradient values. Computation of derivative helps us in edges detection and localization. For edge detection original image is to be convolved with coefficient of gradient obtained along **x** and **y** direction. Results are added together which yields edges in an images. This is represented by mathematical equation.

$$|G|=|Gx|+|Gy| \quad (1)$$

Mathematically gradient can be computed with equation.

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

Edge detection algorithms are classified on the basis of arrangement of coefficient of gradient operators. Literature reviles that several edge detection algorithms are available each of them has own pros and cons based upon subjective application. Canny edge detection algorithm is most widely used algorithm since it offers superior performance for images tented with noise. Edge detection algorithms can be enlisted as follow

Edge Detection Algorithms

Several edge detection algorithms are available broadly they are classified into first order derivative (gradient) and second order derivative (laplacian) based approach [10][11].

A.First order derivative (Gradient based approach)

A. Robert operator for edge detection

Robert edge detection operator first order derivative operator. Coefficient of Robert operator can be obtained by approximating the gradient of an image by discrete differentiation of an image. Discrete differentiation is achieved by computing the addition of squares of the differences between diagonal pixels in matrix. This operator has shortest support and more vulnerable to

 ISSN: 2277-9655
Impact Factor: 1.852

output noise. This operator provides faster computational speed. Kernel can be designed on the basis of mathematical equations computing gradient values is given below.

$$|\nabla Z| = |I_5 - I_8| + |I_5 - I_6| \quad (3)$$

Where, I is the coefficient gradient operator.

B. Prewitts edge detection algorithm

Robert mask used for edge is even size mask hence; it has limitation when used for implementation of algorithm over image. This problem is eliminated in Prewitts operator, named after inventor. In this algorithm while approximating the first derivative similar weights are assigned to all the neighbors of the candidate pixel whose edge strength is calculated. Prewitts operator has faster computational speed. Kernel can be designed based on mathematical equation.

$$|\nabla Z| = |(I_7 + I_8 + I_9) - (I_1 + I_2 + I_3)| + |(I_3 + I_6 + I_9) - (I_1 + I_4 + I_7)| \quad (4)$$

Where, I is coefficient of gradient operator.

C. Sobel edge detection algorithm

It addresses limitation of Robert edge detection algorithm. In sobel algorithm higher weights are assigned to the pixels close to the candidate pixel. The Sobel operator is again first order edge detection operator, It computes gradient of the image as intensity function. The Sobel operator only considers the two orientations which are 0 and 90 degrees as convolution kernels. Absolute magnitude of the gradient at each point of image can be obtained by merging Kernels together. Kernels can be designed on the basis of mathematical equations computing gradient values is given below.

$$|\nabla Z| = |(I_7 + 2I_8 + I_9) - (I_1 + 2I_2 + I_3)| + |(I_3 + 2I_6 + I_9) - (I_1 + 2I_4 + I_7)| \quad (5)$$

Where, I is coefficient of gradient operator.

D. Compass operator for edge detection

It is an extension of Sobel and Prewitts operator which posses an inherent feature of image enhancement along horizontal and vertical direction. In compass operator it is feasible to enhance edges in an image along any direction. This can be achieved by rotating the Prewitts or Sobel operator. Compass operators are very much useful for detecting weak edges in an image.

E. Canny edge detection algorithm

This is most widely used first derivative operator named after inventor J. K. Canny. This operator is capable detecting edges even in noisy image; Gaussian function is used for smoothening of the noise in an image. This is optimal edge detector. This is probably the most widely used edge detector in computer vision applications. Canny edge is a multi-stage algorithm since several steps are involved when compared to other algorithm.

Step1. Gaussian smoothing

The first step is achieved by smoothing the input image by use of Gaussian mask. Smoothing suppress noise in an image, without destroying the true edges.

Step2. Gradient calculation

The gradient is a vector quantity thus it has specific magnitude and direction. Strength of an edge can be estimated on the basis of magnitude of gradient. Gradient is computed in horizontal and vertical direction at location of each pixel.

Step3. Non-maximum suppression

The edge magnitude image may contain wide ridges around the local maxima which are visualized as thick edges. Non-maximal suppression (NMS) produces thin edge removing the non-maximal pixel along the normal direction preserving the connectivity of the contours.

Step4. Hysteresis thresholding

The output of non-maxima suppression still contains the local maxima created by noise. The problem in using single threshold is that if the threshold is set low, some noisy maxima will be accepted too. If a high threshold is set, true maxima might be missed (the value of true maxima will fluctuate above and below the threshold, fragmenting the edge).A more effective scheme is to use two thresholds. Produce two thresholded image and link the edges into the contours, this is referred as double-thresholding.

3. Second order derivative approach

A. Laplacian Operator

Second order derivative approach confers another significant category of edge-detection. Laplacian is sum of second order derivatives in horizontal and vertical direction; it is mathematically expressed as

$$\nabla^2 E = \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial x^2} \quad (6)$$

$\nabla^2 E -$ *a Laplacian operator*

$\frac{\partial^2}{\partial x^2} -$ *Second order derivative*

This is referred as laplacian operator. The laplacian has the advantage over first derivative methods is that it is an isotropic filter [5]; this means it is invariant under rotation. Rotation of an image after application of laplacian would yields same results as obtained when laplacian is applied first and then it is rotated. This is visible that it makes this class of filters excellent for edge detection. Contrarily second derivative filters are very noise sensitive. Fig.1 shows the representation of edge in an image based on derivative approach.

B. Zero crossing evaluation

An effective use of Laplacian operator for edge detection is computing zero crossing. Edge in an image is specified by location of pixels that acquires zero values while filter is moved. In general, these are places where the result of the filter changes signs. Zero crossing can be anticipated in filtered image where the pixels satisfy either of the following

(a) Pixels having negative grey value and are next to (by four-adjacency) a pixel with positive gray value.

(b) Pixels having zero grey values and lying between negative and positive grey range. Actually these pixels possess negative grey value and offered after the pixels with positive grey value. Fig.1 gives edge representation in an image. First and second order derivative is represented as shown.

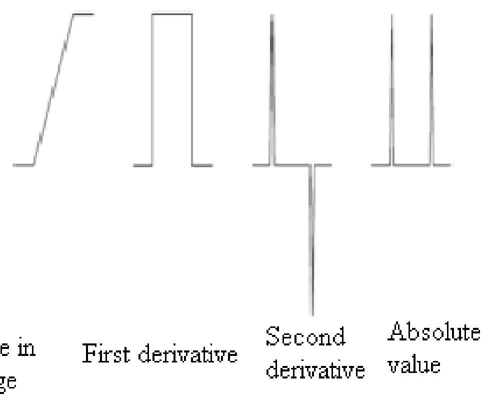

Fig.1 Edge representation by derivative approach

C. Hough transform

Edge determined by above methods if found to be inadequate it must be because of detach points in an image than a immediate line or curvature. To resolve the boundaries in an image, It is necessary to fit a line to those points in an image. It suffers from disadvantages that it consumes more time and it becomes inefficient computationally, especially if edge points are more in number. One of the approaches for judgment of boundary lines is by use of the Hough transform method. The Hough transform is basically intended to locate lines in images, but it can be easily varied to find other shapes existing in an image. Mathematically Hough transform can be modeled for any shapes. It is one of the high efficient transform for edge detection.

Proposed Work

The goal of our work is to implement an image processing algorithm in Xilinx FPGA using 'Xilinx System Generator' for still images as show in proposed block diagram fig.2. More emphasis is on achieving overall high performance, low cost and short development time. Among numerous edge detection algorithms available, we will prefer algorithm which is most reliable and produces an efficient output even for noisy images. FPGA is fine grained device it offers parallelism, in build high speed multipliers which significantly reduces the processing time. FPGAs are increasingly used in modern imaging applications such as image filtering, medical imaging, image compression and wireless transmission [12].

Significant drawback of most of the techniques available for hardware implementation is that they use a high level language for coding in the traditional approach; user needs to emulate the floating point algorithm in HDL code. Further, it can be tested and verified by repeated functional simulations, post simulation processes and finally generate bit stream file. Xilinx introduced the system modeling tool (Xilinx System Generator) that has simplified the development process [13].

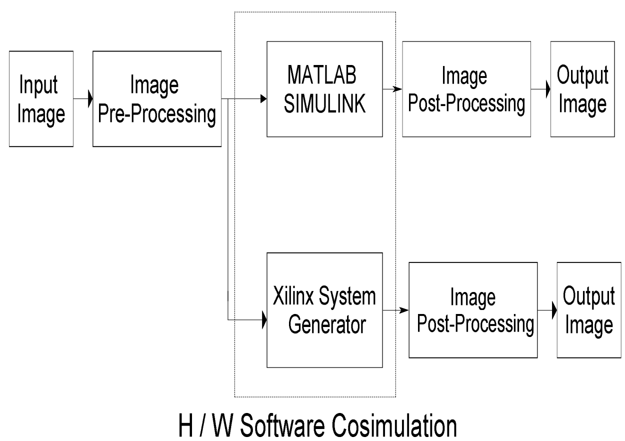

Fig.2 Proposed block diagram for system design.

Image pre-processing is required since input image can be color or grayscale. A color space converter converts RGB to greyscale intensities. Further, two dimensional data is converted to single dimensional array for further processing. Reverses processing is done at the output side. Flow of Xilinx system generator is shown in fig.3 which finally generates HDL netlist and bit stream (.*bit) file.

ISSN: 2277-9655
Impact Factor: 1.852

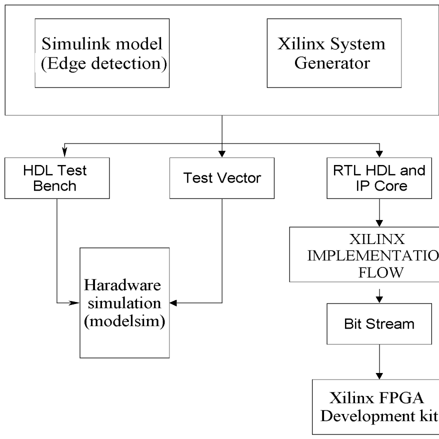

Fig.3 Xilinx system generator design flow

Conclusion

In this paper implementation of edge detection algorithm over FPGA platform is proposed. Edge is significant feature for image processing and many computer vision applications. Several edge detection algorithms are available each one has subjective pro and nuisance. Canny edge detection algorithm is best suited for real time applications because of excellent performance with noisy images. Real time applications demands implementation of edge detection algorithm over hardware platform. FPGA offers high speed, power efficient, small size and cost optimized system. It has high speed multipliers, parallel architecture which makes them superior over their DSP counterparts. FPGA based design offers an advantage of short ‘Turn Around Time’ (TAT) and small ‘Non Recurrent Expense’ (NRE). ‘Xilinx System Generator’ tool of Matlab provides an efficient and simplified approach of FPGA programming.

Acknowledgment

I express my sense of gratitude and sincere regards to my guide P.L.Paikrao and Dr. P.N. Chatur Head of Department. I thank him for extending necessary help, providing facilities and time to time guidance. I would like to thanks all the staff members of Electronics and Telecommunication Department and the people who directly or indirectly helped us in completing this task successfully.

References

[1] R. Harinarayan, R. Pannereselvam, M. Mubarak Ali, D. Tripathi, Feature Extraction of Digital Aerial Images by FPGA based implementation of edge detection algorithms, Proc. of ICETECT, 2011

[2] A.Amaricai,O.Boncalo,M.Iordate,B.Marinescu A Moving Window Architecture for HW/SW codesign Based Canny Edge Detection for FPGA, International Conference On Microelectronics (MIEL 2012)

[3] Christos Kykou,Christos Ttoffis and Theocharis Theocharides,FPGA-ACCELERATED OBJECT DETECTION USING EDGE INFORMATION, International Conference on Field Programmable Logic and Applications,2011

[4] Zhang Shanshan Wang Xiaohong, Vehicle Image Edge Detection Algorithm Hardware Implementation On FPGA, International Conference on computer Application and System Modeling (ICCASM 2010)

[5] Jincheng Wu, Jingrui Sun, Weying Liu, Design and Implementation of Video Image edge Detection System Based on FPGA,International conference on Image and signal Processsing (CISP 2010)

[6] J.Canny, A computational approach to edge detection, IEEE transaction on pattern analysis and machine intelligence, vol 8, no.6, pp,679-698,Nov 1986.

[7] Kazakova, N. Margala, M. Durdle, N.G. Mitra, Sobel edge detection processor for a real-time volume rendering system, Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04. Volume: 2 Publication Year: 2004 , Page(s): II - 913-16.

[8] Yahid Said, Taoufik Saidani, Fethi Smach and Mohamed

[9] Atri, Real Time Hardware Co-simulation of Edge Detection for Video Processing System , ICACSIS 2011. Alfredo Gardel Vicente, Ignacio Bravo Munoz, Embedded Vision Modules for Tracking and Counting People, IEEE transaction on instrumentation and measurement, vol.58 , no 9, september 2009

[10] Alasdair McAndrew, An Introduction to Digital Image Processing with Matlab, vol.

[11] Spartan-3E Starter kit Board User’s Guide,www.Xilinx.com

[12] Xilinx System Generator User's Guide, www.xilinx.com.

[13] L. Yuancheng, R. Duraiswami, Canny edge detection on NVIDIA CUDA – Proc. 2008 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2008, pp 1-8