International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161
©2014 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

# VHDL Based Canny Edge Detection Algorithm

Syed Sameer Rashid[Ȧ*], Swati R. Dixit[Ȧ] and A.Y.Deshmukh[Ȧ]

[Ȧ]Department of Electronics & Telecommunication Engineering, PG Student, G.H.Raisoni College of Engineering, Nagpur, India

Accepted 27 March 2014, Available online 01 April 2014, **Vol.4, No.2 (April 2014)**

## *Abstract*

*Edge is basic feature of image and it is to be detected using various methods which are used for image enhancement, image segmentations, tracking etc. In this paper, a canny edge detection algorithm which is based on VHDL is proposed. Generally images are affected by noise; to reduce the effect of noise Gaussian filtering is used. It also performs image smoothing.The aim of this paper is to develop an edge detection which automatically detects edges of digital image. The complete design of canny edge detector algorithm followed by the Gaussian filtering is done on Xilinx System Generator (XSG).The complete design combines MATLAB, Simulink and XSG. The VHDL code is generated by using Xilinx system generator (XSG). Further the generated VHDL code is synthesize in Xilinx ISE Design Suit 13.1.*

***Keywords:*** *Edge Detection, Canny Edge Detection, Gaussian filtering, FPGA, Xilinx System Generator (XSG).*

## 1. Introduction

Generally 75% information can be obtain by human vision only. The digital representation of visual sense is called digital images. Digital image processing is a field which deals with digital images. The visual perception can be static (sense contain unchanged in time i.e. images) or dynamic (sense contain changes in time i.e. video). Typically, visual sense is a static image, a “snap shot”. A digital image contains various type of information; there are lots of methods in image processing which extract useful information from image. One of the useful methods is edge detection. By the use of edge detection we can obtain some basic information such as location of objects present in the image, their size, and shape image enhancement and sharpening.

The canny edge detector is a multi-stage algorithm; it is widely used in real world due to its good edge detection performance. It is also produce computational theory which explains why this technique works. The operation of canny edge detector can be determined by three parameters, width of Gaussian kernel used in the smoothing phase, the upper and lower thresholds used by the tracker. Sobel, Prewitt, Roberts, Laplacain are affected by the noise and having very low performance in presence of noise. The canny edge detector which has been followed by Gaussian convolution to perform smoothing to all the images gives better performance in presence of noise. Gaussian smoothing operator is 2-D convolution operator that is widely used to blur images and remove detail, noise.

However lots of methods have been introduced for canny edge detection. The method uses fast multilevel fuzzy edge detection to increase the performance of edge detection (Jinbo Wu. et al, 2007). A real time implementation of edge detection which uses cyclone II FPGA pair with 1.3 megapixel CMOS camera gives good performance (Mohamed Nasir Bin Mohamed Shukor. et al, 2007). There is an implementation of distributed canny edge detector on FPGA which uses Xilinx Vertex – 5 FPGA. It reduces memory requirement and latency. For the images corrupted by Gaussian noise (QianXu. et al, 2011). Another proposed method is an improved canny edge detector which is based on predisposal method; by this proposed method better edge images were obtained. Also there is an developed architecture which is based on Xilinx System Generator for image filtering uses the Xilinx blocksets for image filtering (Alba M. Sanehez G. et al, 2007; Wang Xiao et al, 2010). A method for MRI images which is also based on Xilinx System Generator. It gives efficient FPGA implementation of MRI image filtering and tumor characterization, where as a Beamlet transform edge detection algorithm using FPGA, develop an efficient Beamlet edge detection (Dr. D. Selvathi. et al, 2013; Mrs. S. Allin Christe et al, 2011). An implementation gives an overview of MRI Brain classification using FPGA (Dr. MohdFauzi Bin Othman et al, 201O).

By using above existing methods analysis, a canny edge detector based on Xilinx System Generator which is followed by Gaussian filtering is proposed. Canny edge detector performs well in noisy environment. Proposed method uses Gaussian filtering to avoid noisy environment. It gives efficient implementation of canny edge detector algorithm based on Xilinx System Generator. It also gives edge detection in brain tumor or MRI images.

*Corresponding author: **Syed Sameer Rashid**

## 2. Proposed Method

Canny edge detection using Xilinx System Generator contains various steps. First image is needed to be filtered to remove noise present in an image. Next processing steps are to detect edges of an image, and then further steps are generation of VHDL code. The flowchart for the proposed method is shown as follows.

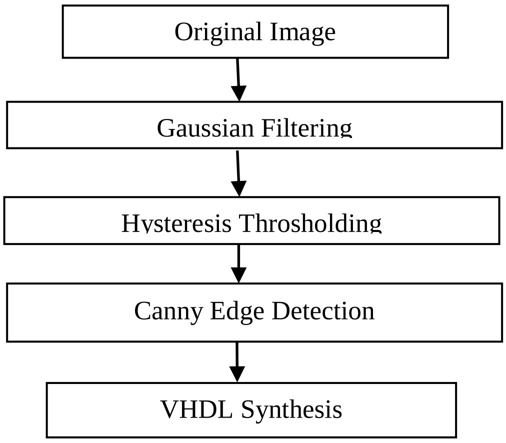


**Figure 1:** Block Diagram of Canny Edge Detection based on VHDL

### *A. Image Filtering*

The digital images obtain from the source often have noise, poor contrast, intensity functions, out-of- focus region, etc. applying edge detection to such an image will gives very poor results. Therefore edges of an image need to be enhanced using image processing methods like histogram stretching, adjustment of hue, median filtering or Gaussian filtering etc.

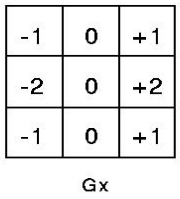
### *B. Canny Edge Detection*

The canny edge detector algorithm was developed to give optimal edge detection. The word optimal means it should provide good detection capabilities, good edge point localization and minimal response. While development of canny edge detector some list of criteria was followed to improve current methods of edge detection. The first one was to obtain low error rate i.e. it should not miss any single edge occurred in an image. The second criteria were it should localize edge points in perfect manner. The third criteria were that it should give a single response to a single edge. It was necessary because first two criteria were not enough to eliminate possibility of multiple edge response.

To implement canny edge detector some processing steps must be followed. The first step is to eliminate the noise components present in an image before finding edges. Therefore ones suitable mask is obtain Gaussian smoothing is performed using standard convolution. The result of this is a blurred image with each pixel is free form noise. Here is an example of Gaussian filter with 5×5 mask having σ = 1.4.

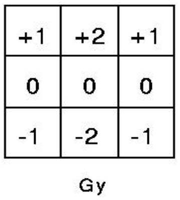
$$\mathbf{B} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

After the removal of noise next processing step is to take the gradient of the image to find the edge strength. The Sobel, Roberts, Prewitt etc. performs 2-D spatial gradient measurement. Then absolute gradient value can be found at each point. The Sobel operator uses two 3×3 convolution masks. One is used to find gradient in x-direction and other in y-direction i.e. rows and columns.



| | | |
|---|---|---|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

Gx



| | | |
|---|---|---|
| +1 | +2 | +1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

Gy

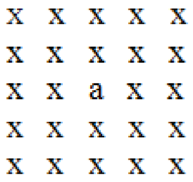
Then gradient approximated using the formula

|G| = |GX | + |GY|

Finding the edge direction is much easy when the gradients in x and y direction known. Therefore the formula for edge direction is given as

Ɵ = invtan(GY/GX)

Once the edge direction is found, the next step is to compare the edge direction to a direction that can be traced in an image. So the pixel of 5×5 image aligned as follows



x x x x x
x x x x x
x x a x x
x x x x x
x x x x x

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the neighborhood pixels in the horizontal direction (0 degrees), along the positive diagonal (45 degrees), in the vertical direction (90 degrees), or along the negative diagonal (135 degrees). So now edge orientation has to be resolved into one of these four directions depending on which direction closet to it (e.g. if the orientation angle is found to be 5 degrees, make it zero degrees).

After the edge directions are known, the edge thinning technique called non-maximum suppression has to apply. Non-maximum suppression is used to trace the edge in edge direction and suppress any pixel value that is not considered to be an edge. This will give thin line in an output image. Finally, the last step is tracing edges from the image and hysteresis throsholding. The hysteresis is used forremoving streaking from an image. Streaking is the breaking up of an edge contour it is caused due to fluctuation of operator aboveand below the threshold. Thresholding with hysteresis requires two thresholds – high and low level thresholds

.

### C. *VHDL Synthesis*

The implemented design is need to be synthesized in ISE Design Suit 13.1. It produces device utilization summary for hardware implementation.

## 3. Processing Steps

Xilinx System Generator (XSG) is the industry's leading high level tool for designing high-performance DSP systems using FPGAs. It provides an integrated design environment for FPGA within the ISE 13.1 development suit. The Simulink is used as a development environment and is presented in the form of model base design. For programming FPGA the Bit Stream (*.bit) file is necessary. The Xilinx System Generator has an integrated design flow which moves Simulink design directly into Bit Stream (*.bit) file means it provide system modeling and automatic code generation from Simulink and MATLAB. The system generator design is consist of models called "XILINX BLOCKS". These blocks consist of signs, ports, attributes and entities which are used to produce synthesis in FPGAs or HDL simulation. System generator is a part of system Editions of ISE Design Suit. With these developers having little FPGA design experience can quickly create production FPGA implementations of DSP algorithm.

### A. *Gaussian filtering using XSG*

The model base design uses Xilinx blocksets. Gaussian filtering uses 5 line buffer, a sequential stream of pixels to constructs 5 lines of output. Each line is delayed by N samples, where N is the length of line. Further it uses 5×5 filter. It is implemented by using 5n-tap MAC FIR filters. Nine different 2-D filters have been provided to filter grayscale images. The designed Gaussian filtering using Xilinx blocks is shown in Fig 2.

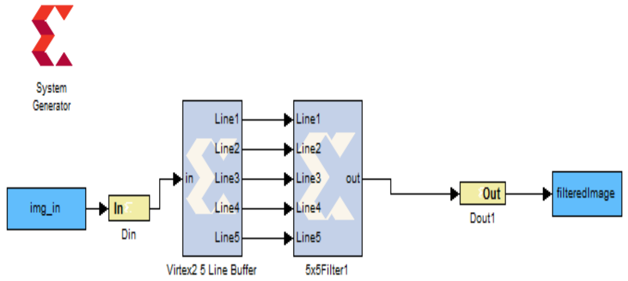


**Figure 2:** Gaussian Filtering using XSG

### B. *Canny Edge Detection using XSG*

This block is also consisting of Xilinx blocksets. Gateway In block converts inputs of type Simulink integer. Double and fix point to Xilinx fixed point type. Gateway Out block converts Xilinx fixed point input into outputs of type Simulink integer, double, or fixed point. All the Xilinx blocksets are used between these two blocks i.e. Gateway In and Gateway Out. The Canny Edge Detection based on Xilinx blockets shown in Fig 3.

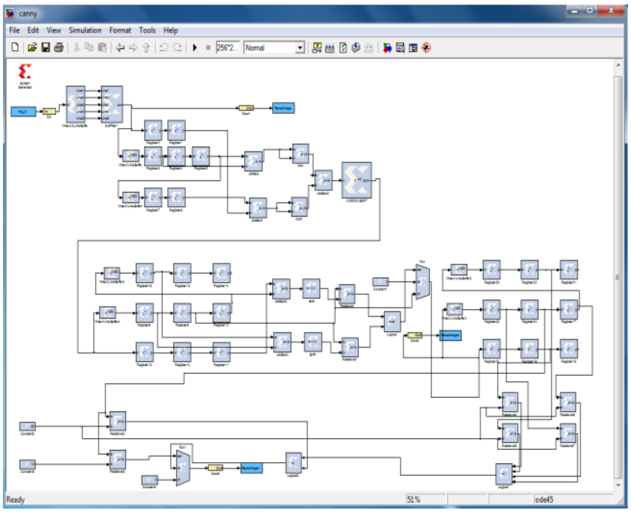
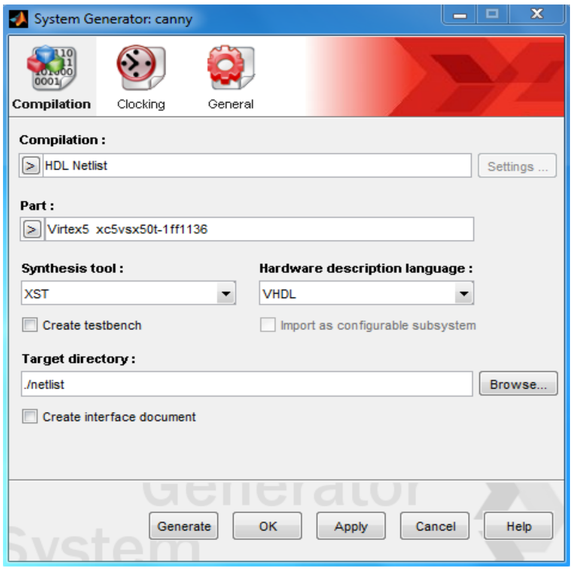


**Figure 3:** Canny Edge Detection using XSG

### C. *VHDL Code Generation*

The Xilinx block called System Generator, it is a system generator token serves as a control panel for controlling system and simulation parameters. Any Simulink design which contains ant type of Xilinx blocks must contain at least one System Generator token.



Compilation : HDL Netlist

Part : Virtex5 xc5vsx50t-1ff1136

Synthesis tool : XST

Hardware description language : VHDL

□ Create testbench

□ Import as configurable subsystem

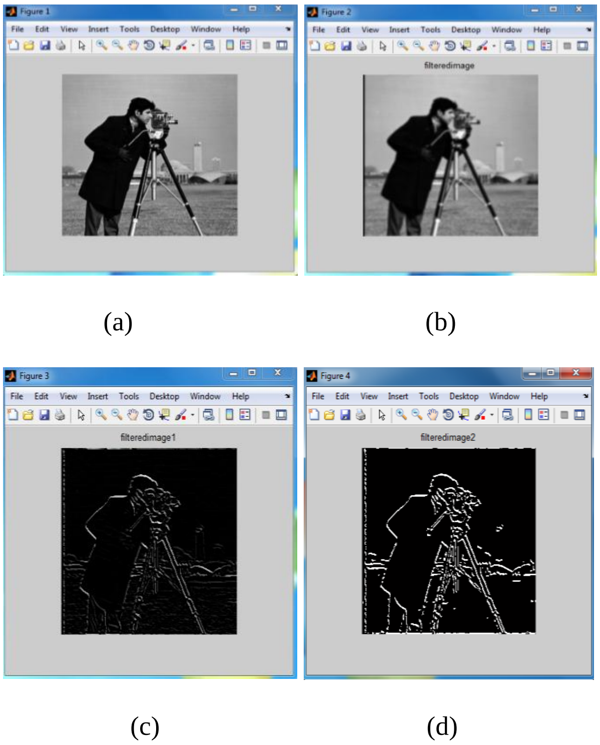
Target directory : ./netlist

□ Create interface document

**Figure 4:** System Generator Block

## 4. Results

The proposed work is implemented by using MATLAB Simulink and XSG (Xilinx System Generator) with Xilinx blocksets. The method was tested on standard test image like 'cameraman'. The output results show an image filtered from Gaussian filtering, another output is hysteresis throsholding output which removes streaking form an image. Final image is of Canny Edge Detected image. The generated system is targeted for VIRTEX 5 starter kit. Further the VHDL code is generated by using System Generator token, this code is perfectly synthesized in ISE 13.1 Design Suit. After synthesis the device resource usage summary was produced for the targeted device.



(a) (b)

(c) (d)

**Figure 5 :** (a) Original Image (c) Gaussian Filtering Result (c) Hysteresis Throsholding Result (d) Canny Edge Detection Result

The device utilization summary is estimated for the proposed design. Performance of this architecture implemented in Vertex5 (xc5vsx50t-1ff1136) as shown in Table. The proposed architecture provides lower complexity as well as improves efficiency in area. It also provides good choice in terms of low cost hardware.

## Conclusions

The presented proposed work gives an efficient and robust Canny Edge Detection algorithm. The edges are well located using Xilinx Blocksets. The Canny Edge Detector gives better results than other operators like Sobel, Laplacian, Log, etc. The results have shown better improvement as compared to other operators. The technique uses updated Xilinx System Generator within ISE 13.1 design suit. The implemented design is targeted on Vertex5 xc5vsx50t-1ff1136 starter kit. Generated VHDL code is perfectly synthesized in ISE design suit, produced device utilization summary which improves efficiency in area. This approach is very useful in medical images to diagnose the diseases.

## References

Jinbo Wu, Zhouping Yin, and YoulunXiong. (2007), The Fast Multilevel Fuzzy Edge detection of Blurry Images, *IEEE Signal Processing Letters, Vol. 14, No. 5,*

Mohamed Nasir Bin Mohamed Shukor, Lo HaiHiung, Patrick Sebastian,(2007), Implementation of Real-Time Simple Edge Detection on FPGA, *International Conference on Intelligent and Advanced Systems.*

QianXu, Chaitali Chakrabarti and Lina J. Karam, (2011), A Distributed Canny Edge Detector and Its Implementation on FPGA, IEEE.

Wang Xiao, XueHui, (2010), An Improved Canny Edge Detection Algorithm Based on Predisposal Method for Image Corrupted By Gaussian Noise, *World Automation Congress (WAC).*

Alba M. Sanehez G., Rieardo Alvarez G., Sully Sanehez G, (2007), Architecture For Filtering Ximages Using Xilinx System Generator, *International Journal of Mathematical Models and Methods in Applied Sciences, Volume 1.*

S. AllinChriste, Mr.M. Vignesh, Dr.A.Kandaswamy, ( 2011), An Efficient FPGA Implementation of MRI Image Filtering and Tumor Characterization Using Xilinx System Generator, *International Journal of VLSI Design & Communication Systems (VLSICS) Vo1.2, No.4*

D.Selvathi, J.Dharani, (2013), Realization of Beamlet Transform Edge Detection Algorithm Using FPGA, *International Conference on Signal Processing, Image Processing and Pattern Recognition [ICSIPR]*

MohdFauzi Bin Othman , Norarmalina Abdullah, NurAizudin Bin Ahmad Rusli, (201O), An Overview Of MRI Brain Classification Using FPGA Implementation, *IEEE Symposium On Industrial Electronics& Applications (ISIEA) Oet,Malaysia.*

Device Utilzation Summary

(Vertex5 xc5vsx50t-1ff1136)

| Resource | Used | Available | Device usage |
|---|---|---|---|
| Registers | 2,931 | 97,280 | 3% |
| LUTs | 2,247 | 97,280 | 2% |
| Logic | 1,914 | 97,280 | 1% |
| Memory | 302 | 26,240 | 1% |
| Number of occupied Slices | 1,042 | 24,320 | 4% |
| Number with an unused Flip Flop | 375 | 3,306 | 11% |
| Number with an unused LUT | 1,059 | 3,306 | 32% |
| fully used LUT-FF pairs | 1,872 | 3,306 | 56% |
| slice register sites lost to control set restrictions | 92 | 97,280 | 1% |
| IOBs | 83 | 640 | 12% |
| BlockRAM/FIFO | 14 | 212 | 6% |
| Total Memory used (KB) | 342 | 7,632 | 4% |
| BUFG/BUFGCTRLs | 1 | 32 | 3% |
| DSP48Es | 9 | 128 | 7% |