

ENHANCED OPTIMIZATION OF EDGE DETECTION FOR HIGH RESOLUTION IMAGES USING VERILOG HARDWARE DESCRIPTION LANGUAGE WITH LOW POWER CONSUMPTION AND LESS HARDWARE TECHNOLOGY

T.Manikyala Rao¹, Praveen Chakravarthy BH², R. Sushma³, G.Parameswara Rao⁴
Assistant Professor, Electronics and Communication Engineering, SSCE¹²³⁴

ABSTRACT:

Edge Detection plays a crucial role in Image Processing and Segmentation where a set of algorithms aims to identify various portions of a digital image at which a sharpened image is observed in the output or more formally has discontinuities. The contour of Edge Detection also helps in Object Detection and Recognition. Image edges can be detected by using two attributes such as Gradient and Laplacian. In our Paper, we proposed a system which utilizes Canny and Sobel Operators for Edge Detection which is a Gradient First order derivative function for edge detection by using Verilog Hardware Description Language and in turn compared with the results of the previous paper in Matlab. The process of edge detection in Verilog significantly reduces the processing time and filters out unneeded information, while preserving the important structural properties of an image. This edge detection can be used to detect vehicles in Traffic Jam, Medical imaging system for analysing MRI, x-rays by using Xilinx ISE Design Suite 14.2.

KEYWORDS:

Sobel, Canny, Verilog, ISE Design Suite 14.2

1. INTRODUCTION:

Images are the Pictorial representation of an Object or a Person in a two dimensional format. If the labels of the co-ordinate functions (X, Y) represent the Intensity level or Gray level then the function “f”, it represents the amplitude of the function of an image. And when those co-ordinate functions represent discrete values then that image can be represented as a Digital Image. That Digital image can be easily processed by many computer algorithms as a sub-category field in digital signal processing.

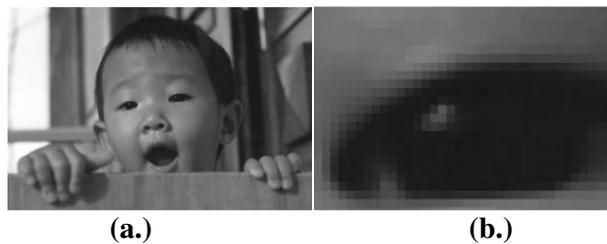


Fig1 (a.) Represents the Image of a boy (b.) Represents Gray-Square of a Image

The above figure (1) represents that each point of an original image has an 8-bit gray coded level and each pixel as a gray square.

There are no clear cut boundaries in the continuous series or a whole, no part of an image is noticeably different from different parts, although extremes of it are very different from each other for image processing at one end to the counter vision to the other. However, one useful paradigm is to consider three types of computerized processes in the continuum: Low-Mid & High level processes.

Low level processes involve primitive operations such as image pre-processing to reduce noise, contrast enhancement and image sharpening. Mid Level Processing of images involves tasks such as segmentation (Partitioning and Recognition) of individual objects. The output of the Mid-Level Processing is the attributes extracted from the image (e.g. Edges, Contours, and identity of individual objects). Finally, higher level processing involves making sense of an ensemble of recognized objects as in image analysis, at the far end of the continuum, performing the cognitive functions normally associated with the vision.

In the existing method, we preferred Mid-Level Processing for edge detection which has image as an input and the attribute of the image as output.

2. EDGE DETECTION:

Edges are the outlines of an image which serves as a boundary between background and the forefront part of an image. So Detection of edges plays a crucial role in image segmentation and object recognition. The main goal of our paper is to use different operators for edge detection and compare the results between them by looking at the less hardware technology and the power comparison between them under variety of conditions to determine which operator is useful or not.

The fundamental and most commonly used important task in image processing is Image Segmentation which determines the accuracy of the computational analysis which are determined by two primitive properties such as, image intensity values that is finding out the edges of an image and partitioning of image into different segments based on primitive factor such as thresholding, region growing and splitting.

As in the first Criteria, detection of discontinuities in a digital image, our basic approach leads to Edge detection instead of Point and Line Detection.

The two most serious errors in Edge Detections are anisotropic edge detection and wrong estimation of direction of the edges.

While optimizing the Edge Detection in Image Processing, Properties of the edges has to be considered where averaging filters suppresses structures with high wave numbers. Edge detection can be done by using three different techniques. Such as,

1. Gradient-Based Edge Detection
2. Edge Detection by Zero Crossings
3. Regularized Edge Detection

In our existing work, the main principle of Edge detection is done by using Regularized edge detection using two operators such as CANNY and SOBEL. Whereas, the other two techniques such as Gradient and Zero-Crossed Edge Detection technique has poor performance mostly in noisy images because of their small mask sizes. And as the high noise images had high wave numbers with greater mask size least performance would be achieved by using these two techniques.

Thus the regularized edge detection is used in high wave numbers to achieve maximum signal to noise ratio.

And also we need to derive some standard filters to perform a derivation in one direction and smoothening in all directions. Smoothing has no negative coefficients in higher dimensional signals as smoothening in all directions perpendicular to the gradient doesn't blur the image. The derivative filters used after the edge detection is called as the "**Regularized Edge Detective Filters**" and they provide estimating derivatives even from the ill-posed problems of discrete signals.

The Sobel operator is a discrete differential operator with smallest differences with odd number of coefficients averages the images in the direction perpendicular to the differentiation. It utilizes two 3*3 Matrices where one utilizes gradients in x-direction while the other utilizes gradients in y-directions.

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

Gx

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

Gy

Fig2. Sobel Operator using 3*3 matrices

The matrices here are multiplied with a 1/8 factor to get the right gradient value. Smoothing can be used after the Sobel operation to remove the noise after the edge detected image. The image is convolved finally with the two gradients horizontal and vertical and at each point the magnitude of the gradient can be approximated as

However it is faster to compute the magnitude of the Gradient with

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

The Canny Edge Detector can be done by selecting the threshold either for thick or thinning edges. This operator is widely known as Optimum operator and it has three steps to implement the edge detection. Such as, first, a Gaussian filter to smooth the image and to remove the noise and the second is to compute the horizontal and vertical gradients (Gx) and (Gy) similar to the Sobel operator. And the third one is non-maximal suppression which is the algorithm to remove unwanted pixels in the image.

The first step in the canny operator is done by using hysteresis thresholding along the edges. The two types of hysteresis thresholding techniques are

1. Upper threshold (which selects the pixels as an edge having higher gradients)
2. Lower threshold (Which discards the pixels having higher gradients)

And if the pixel gradient is in between the two thresholding points then the pixel which has highest gradient than threshold will be selected as an edge of the image.



Fig3 (a.) Original Image, (b.) Edge detected image using Sobel operator

Table1. Comparison of Performances of Sobel and Canny Operators

S.No	Sobel Operator	Canny Operator
1	Algorithm is Simple and easy to operate within less time	Complexity in algorithm and takes large time response
2	Inaccurate and Sensitive to Noise	Utilizes Smoothing effect to remove noise
3	Good localization and response and immune to noise environment	Localization will be compromised for higher wave numbers

3. PROPOSED WORK

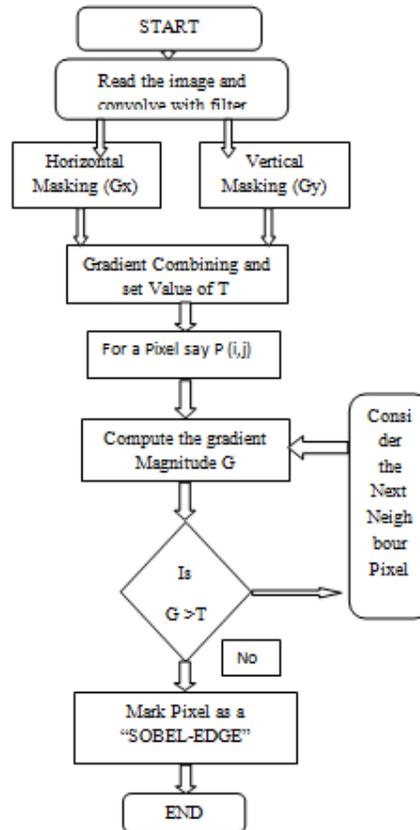
In the Proposed work, the complexity of programming the algorithmic steps of Sobel and Canny operators using Matlab is resolved by implemented the edge detection using Verilog Programming language. Thus Complexity in algorithm is greatly reduced which in turn reduces the cost and complexity of the system with less optimized power.

In the Sobel operator, the horizontal and vertical masking is used to calculate the horizontal and vertical gradients (Horizontal and Vertical edges of the image). And then we calculate the overall gradient by using the overall square root for the summation of the square of both the horizontal and vertical gradients (G_x and G_y).

Steps for Sobel operator:

1. Read the Pixels of the image and convolve with the filter.
2. After Horizontal and vertical masking is done separately find out the Overall Gradient of the image.

3. Then Consider the First Pixel, Let's say "P"
 4. For "P", If G is greater than "T" then move to next Pixel otherwise mark the Pixel as an Edge.
 5. Continue the process for overall gradients of all the pixels.
- The Flow Chart for Sobel Operator is shown below:



Sobel Edge detector is compromised in such a way to use a Smoothing filter to increase the noise immunity under certain conditions but it hurts the localization factor.

The implementation of Sobel Edge detection algorithm is as follows; Consider we have the image $G(c, r)$, Then G_x is given by

$$G_x = [G(c+1, r-1) + 2*G(c+1, r) + G(c+1, r+1) - G(c-1, r+1) - G(c-1, r) - G(c-1, r-1)];$$

Also G_y is given by

$$G_y = [G(c-1, r-1) + G(c, r-1) + G(c+1, r-1) - G(c-1, r+1) - 2*G(c, r+1) - G(c+1, r+1)];$$

Once you have the G_x and G_y its sum is calculated to obtain the gradient magnitude and is compared with the threshold value. If the compared value is higher than threshold then the pixel is replaced by a one otherwise with a zero. The same is done with all the pixels of the image to find the edges.

Using Canny Operator:

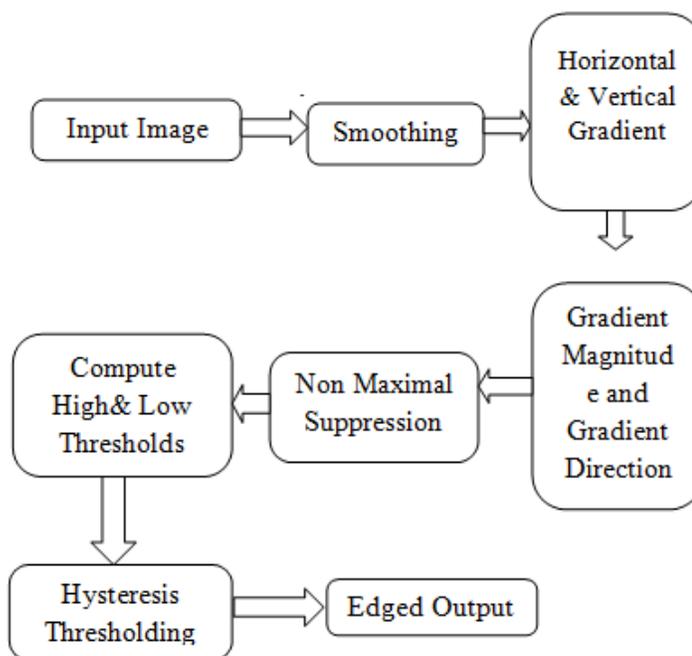
Canny operator is the best optimal operator which uses three architecture types such as:

1. Low error rate (i.e. no edges has to be missed and no non-edges has to be considered)
2. Edge points must be confined with less distance among them
3. Only one reaction to a single edge.

In the Canny operator, Smoothing is computed first and then the gradients are calculated individually with highest masks. The one with the least will be marked as zero otherwise it is set as one. Hence it has higher immunity to Gaussian noise than Sobel and other operators.

In this operator the gradient arrays are removed by using Hysteresis. The one with highest hysteresis will mask as one otherwise the pixel is masked as zero. If the gradient mask is in between the two verges then also it is masked as zero (i.e. Non-edge)

Block diagram of Canny Operator:



The Canny Edge Detection Algorithm:

Step1:

Smooth the image with two Gaussian Operators for both horizontal and vertical gradients and remove the noise part in the image.

Step2:

Consider the Horizontal and Vertical gradients of the image along x and y axes.

Step3:

If there exists an error, in the complete resultant gradient then it is set to either 0 or 90 degrees. This can be done by setting y-axis to 0 or complete x-axis to 0 by using the formula shown below:

Is set to 0 if denominator is 0 otherwise it is set to 90 degrees.

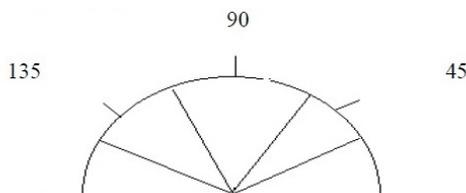
Step4:

Once the edge direction is recognized, the coming step is to agnate the edge direction to a direction that can be marked in an image. So if the pixels of a 5x5 image are arranged as follows

```
  x xxxx
  x xxxx
x x a x x
  x xxxx
  x xxxx
```

It can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (adjacent the actual diagonal), 90 degrees (in the vertical direction), or 135 degrees (adjacent the negative diagonal).

So now the edge position has to be adjudicate into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, adjust it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions.



Therefore, any edge direction falling within the range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction lies in the range (22.5 to 67.5 degrees) is equal to 45 degrees. Any edge direction lies in the range (67.5 to 112.5 degrees) is equal to 90 degrees. And lastly, any edge direction lies within the range (112.5 to 157.5 degrees) is set to 135 degrees

Step5:

Suppress the non-edges and give a thin line to the edges.

Step6:

Hysteresis thresholding is used to eliminate streaking. It makes use of both a high threshold and a low verge. If a pixel has a value above the high verge, it is equal to an edge pixel. If a pixel has a value above the low threshold and is the neighbour of an edge pixel, it is equal to an edge pixel as well. If a pixel has a value above the low verge but is not the neighbour of an edge pixel, it is not equal to an edge pixel. If a pixel has a value below the low verge, it is never equal to an edge pixel.

Canny local operator gives Smoothing to the noisy images and also supports localization but consumes more time because of its complex algorithm.

Implementation of Verilog for any operator

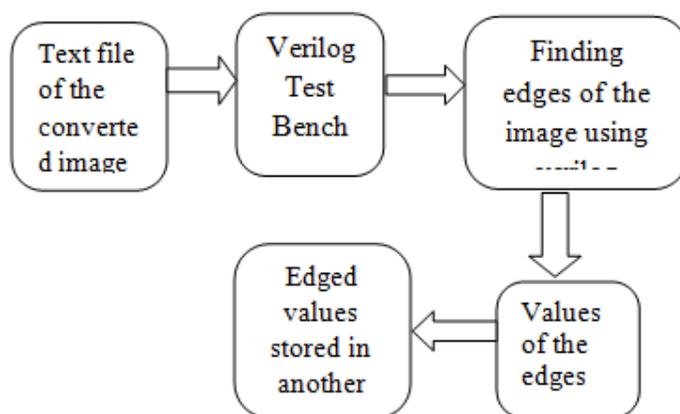


Fig 4. Block Diagram for Any Operator In Verilog.

4. RESULTS:

Canny operator:

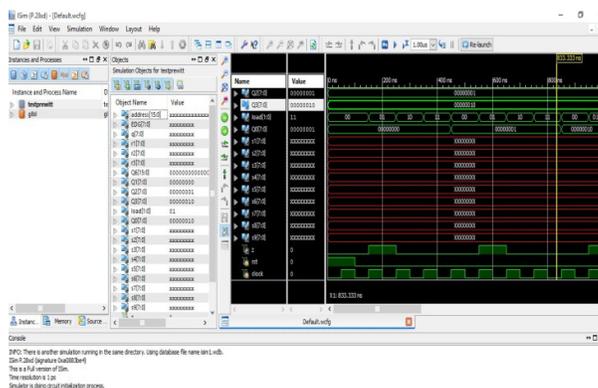


Fig5. Timing Diagram of Canny Operator

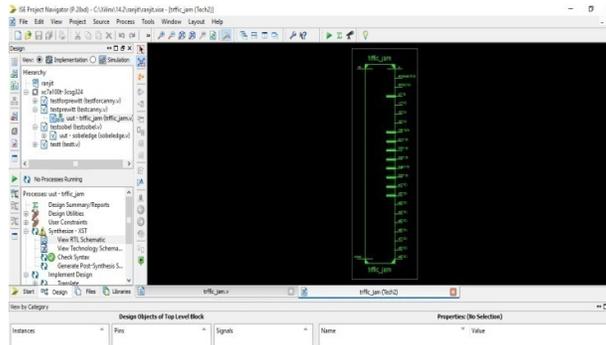


Fig6. RTL Schematic of Canny operator

Logic utilization	Used	Available	Utilization
Number of Slice Registers	11443	149760	7%
Number of Slice LUTs	26223	149760	17%
Number of fully used LUT-FF pairs	3440	34226	10%
Number of bonded IOBs	23	680	3%

Fig7. Device utilization summary of canny operator (previous work)

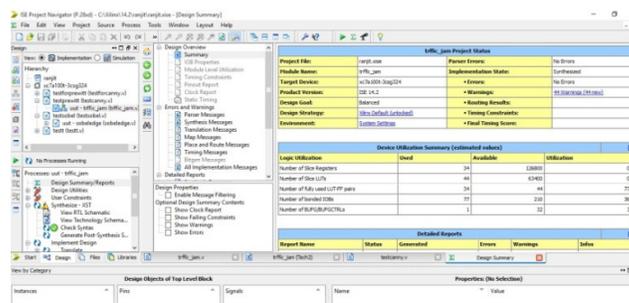


Fig8. Device Utilization summary of canny operator (Proposed work)

From the above figure 6 represents the timing waveforms, figure 7 represents the RTL schematic diagram,

Figure 8 represents device utilization summary of canny operator in previous work figure 9 represents device utilization summary of our present work, when compared to the previous work number of slice registers, slice LUTs, LUT flip flop pairs used are decreases, hence our system is better than the previous one.

Sobel Operator:

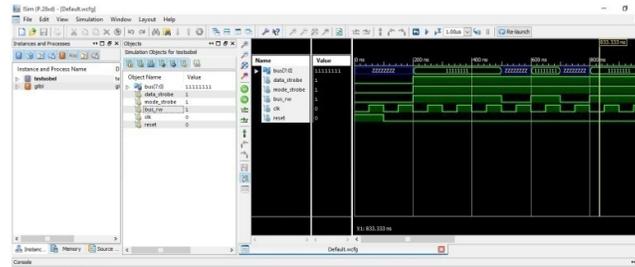


Fig9. Timing Waveform of Sobel operator

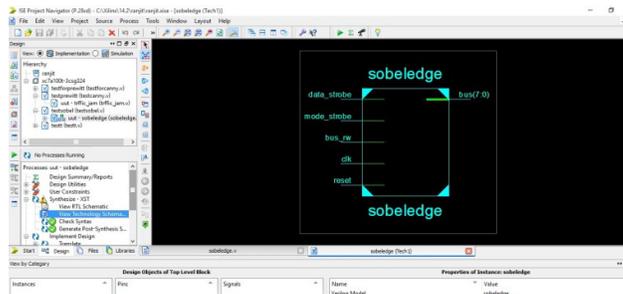


Fig10. RTL Schematic Diagram of Sobel Operator

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2890	3584	80%
Number of Slice Flip Flops	845	7168	11%
Number of 4 input LUTs	4969	7168	69%
Number of bonded IOBs	22	97	22%
Number of GCLKs	1	8	12%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed Apr 29 10:49:48 2009	0	3 Warnings (3 new, 0 filtered)	5 Infos (5 new, 0 filtered)
Translation Report					
Map Report					
Place and Route Report					
Static Timing Report					
Bilgen Report					

Fig11. Device utilization summary of Sobel operator (Existing work)

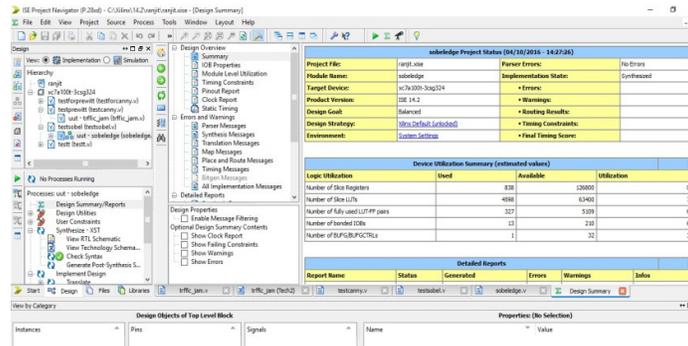


Fig.12. Device utilization summary of Sobel operator (Proposed work)

From the above figure 10 represents the timing waveforms, figure 11 represents the RTL schematic diagram, figure 12 represents the design utilization summary of Sobel operator.

Figure 12 represents device utilization summary of Sobel operator in previous work figure 13 represents device utilization summary of our present work, when compared to the previous work number of slice registers, slice LUTs, LUT flip flop pairs used are decreases, hence our system is better than the previous one.

5. CONCLUSION AND FUTURE SCOPE

In our paper, edge detection is done by using three techniques using Verilog module. They are Sobel, canny and prewitt. By using these techniques high end images can also be edge detected in desired fashion by taking Pixel intensity and separate quantization levels. This procedure is very easy when compared to the other modules implemented using mat lab and vhdl and requires less number of flip-flops, logic gates with low power consumption.

Future scope:

This paper can be implemented on FPGA and CPLD kits and high end images can be edge detected and viewed by using different display devices, which could be a most powerful tool in image processing and visualization techniques.

REFERENCES:

- [1] D. Ziou and S. Tabbone, "Edge detection techniques – an overview," International Journal of Pattern Recognition and Image Analysis, vol. 8, pp. 537–559, 1998.
- [2] Chandrashekar N.S, Dr. K.R. Nataraj, "A Distributed Canny Edge Detection and Its Implementation on FPGA" International
- [3] Tejaswini H.R, Vidhya N, Swathi R Varma, Santhosh B, "An Implementation of Real Time Optimal Edge Detection and VLSI Architecture". International conference on electronics and communication engineering, 28th april-2013, bengaluru, isbn: 978-93-83060-04-7.