JOURNAL OF TECHNICAL REPORTS IN ENGINEERING AND APPLIED SCIENCE

Contents available at: www.jotres.com

# An Architecture of RGB Edge Detection Technique For Fault Acceptance Applications

Arpita Upadhyay[1], Nitin Jain[2]
*1. M.Tech. Scholar, Dept. of Electronics Engg., CEC Bilaspur.*
*2. Assistant Professor, Dept.of Electronics Engg., CEC Bilaspur.*

**ARTICLE INFO** | **ABSTRACT** | **REVIEW ARTICLE**

**Article History**
**Received: August 2016**
**Accepted: Sept' 2016**
**Keywords:**
Sobel,
RGB,
SPAA,
Architecture,
FPGA.

An edge is a contour across which the brightness of the image changes in an abrupt manner. In image processing, an edge is generally presented as one class of singularities. In a function, singularities can be characterized easily as discontinuities where the gradient approaches infinity. Because image data is discrete, so edges in an image are often referred as the local maxima of the gradient. In image processing, edge detection is an important task. It is a major tool in applications like, image segmentation, pattern recognition and scene analysis. An edge detector can be called as a high pass filter. In an image, it can be applied to extract the edge points. Many researchers have been attracted by this topic and many contributions have been made. The efficacy of many image processing depends on the perfection of detecting meaningful edges. It is one of the techniques for the detection of discontinuities of intensity in a digital image. Many researchers provided various approaches based on mathematical calculations in which some of them are cost effective or robust. A new algorithm will be proposed for detection of the edges of noisy RGB image with increased robustness and throughput. Using this algorithm the time complexity problem will be reduced which was faced by previous algorithms. A hardware unit will also be created for the proposed algorithm which will reduce the problems of area, power and speed.



## Introduction

The points at which brightness of the image changes sharply are organized into a set of curved line segments referred as edges. There are various types of sharp changing points in an image. Edges can be created by texture, geometry, shadows and so forth. Edges can also be explained as discontinuities in the intensity of the image due to the changes in the structure of the image. These discontinuities emerge from different features in an image. Edge points are associated with the boundaries of the objects and other kinds of changes. Edges within an image generally occur at different resolutions and represent transitions of different gradient levels. Edge detection is a basic tool in, image processing, computer vision and machine vision, specifically in the areas of feature detection, feature extraction. [16, 17].

### 1.2 Noise in edge detection

Edge detection is easily influenced by noise. This is because the edge detection algorithms are designed to respond to sharp changes that can be caused by noisy pixels.

Salt & pepper noise, white noise, and speckle noise are the examples of the most common types of noise. Edges are located by gradient of image intensity. The noise effect can be minimized by filtration of the image by Gaussian function or by using smoothing function. Linear edge detectors such as LOG, sobel, canny smooths the image before detecting the edges.

Edge detection can be performed in various ways. There are many ways in which edge detection can be performed. However, most of them are grouped into two categories: First one is the gradient based edge detection and the second one is Laplacian-based edge detection. In the gradient method, an estimate of the gradient magnitude is found using the smoothing filter and then this estimate is used to determine the position of the edges. In other words, this method detects the edges by looking for the minima and maxima in the first derivative of the image. In Laplacian method we calculate the second derivative of the signal. When second derivative is zero, the derivative magnitude is maximum. In short, this method searches for zero crossing for finding the edges in the second derivative of the image .

**Sobel Operator**

It uses two 3x3 kernels. These are convolved with the original image. The second kernel is obtained by rotation of the first kernel by 90°.These are orthogonal. The kernel values are designed so as to get maximum response to edges running in vertical direction and horizontal direction related to the pixel grid. The kernels are used separately by the input image in order to produce separate measurement and Gy ( gradient component).Then these can be grouped together so that absolute magnitude and orientation of that gradient could be obtained.

**Literature Review**

Sobel operators for a 3×3 neighborhood is vector sum of a pair of orthogonal vectors [2,6,7,20].The orthogonal vector is referred as a directional derivative estimate multiplied by a unit vector which specifies the direction of the derivative. The directional derivative estimate vector *G* has been defined as density difference distance to neighbor in [7].This vector is determined so as the direction of *G* will be given by the unit vector to the approximate neighbour.The neighbors are grouped into antipodal pairs: (a,i), (b,h), (c,g), (f,d). One mask is the other one rotated by 90°.These masks are designed in such a way so as to respond maximally to edges running in vertical direction and horizontal horizontal direction relative to the pixel grid. The definition of the gradient can be used for 5×5 neighborhood [8].Here, twelve directional gradient should be determined instead of four gradients. The resultant vector *G’* for 5×5 that is similar to the determination of 3×3 Sobel method is given as follows:

**Pseudo-codes for sobel edge detection**

Input: A sample image is the input.

Output: Detected edges are the output.

Step 1: Firstly, the input image is accepted.

Step 2: Then mask Gx, Gy to the input image is applied.

Step 3:Further, Sobel edge detection algorithm and the gradient is applied.

Step 4:Masks manipulation of Gx, Gy is done separately on the input image.

Step 5:The Results are combined so as to find the absolute magnitude of the gradient.

Step 6: Finally, the absolute magnitude is the output edge.

Now let us see how it works.

We know that the sobel operator has a pair of 3×3convolution kernels. One kernel is simply rotated by 90°.

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

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

Fig-1: Sobel convolution kernels

These kernels are designed in such a manner so as to respond maximally to edges running in vertical direction and horizontal direction relative to the pixel grid. For the separate measurement of the gradient component in each orientation i.e *Gx* and *Gy* ,the kernels are applied separately to the input image .Further these can be combined for finding the gradient's absolute magnitude at each point and its orientation.

The gradient magnitude is given by:

(1)

An approximate magnitude is calculated using:

(2)

The orientation angle of the edge (relative to pixel grid) that give rise to the spatial gradient is given by:

$\theta = \varepsilon$ (3)

| | | |
|---|---|---|
| $P_1$ | $P_2$ | $P_3$ |
| $P_4$ | $P_5$ | $P_6$ |
| $P_7$ | $P_8$ | $P_9$ |

Fig-2:Pseudo-convolution kernels for quickly computing approximate gradient magnitude

Using the kernel the approximate magnitude is given by:

$|G|=|(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times)\quad | + |(P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4$
(4)

| | | |
|---|---|---|
| a | b | c |
| d | e | f |
| g | h | i |

Fig -3: The neighbors grouped into antipodal pairs

In [1], the directional derivative estimate vector *G* was defined as density difference distance to the neighbor. This vector is determined in such a way that the direction of *G* will be given by the unit vector to the approximate neighbor. The neighbors are grouped into antipodal pairs: (a,i), (b,h), (c,g), (f,d).For this gradient the vector sum is:

G= + · +(b-h)·[0,1]+(f-d)·[1,0] (5)

Where, *R* = . This vector is obtained as

$$G = [(c - g - a + i)/2 + f - d, (c - g + a - i)/2 + b -]$$

(6)

Here, this vector is multiplied by 2 because of replacing the divide by 2. The resultant formula is given as follows

G'= =[(c-g-a+i)+2·(f-d),(c-g+a-i)+2·(b-h)] (7)

The following weighting functions for x and y components were obtained by using the above vector.

Now, it is explained that the dimensions of the matrices are tended by using [1]. The definition of the gradient could be used for 5×5 neighborhood[8].In this case, twelve directional gradients must be determined instead of four gradients.

The resultant vector *G'* (similar to Sobel 3×3 determination method) for 5×5 is given as below:

G'=[20(n-l)+10(i-r-g+t+o-k)+5(e-v-a+z)+4(d-w-b+y)+8(j-p-f+u),20(h-s)+10(i-r+g-t)+5.(e-v+a-z)+4(j-p+f-u)+8(d-w+b-y). (8)

The horizontal and vertical masks are obtained by using the coefficients in this equation such as

| | | | | |
|---|---|---|---|---|
| -5 | -4 | 0 | 4 | 5 |
| -8 | -10 | 0 | 10 | 8 |
| -10 | -20 | 0 | 20 | 10 |
| -8 | -10 | 0 | 10 | 8 |
| -5 | -4 | 0 | 4 | 5 |

| | | | | |
|---|---|---|---|---|
| 5 | 8 | 10 | 8 | 5 |
| 4 | 10 | 20 | 10 | 4 |
| 0 | 0 | 0 | 0 | 0 |
| -4 | -10 | -20 | -10 | -4 |
| -5 | -8 | -10 | -8 | -5 |

.

Each of the direction of the Sobel mask is applied to an image, and then two new images are created. One image will show the vertical response and the other one will show the horizontal response. These two images combined into a single image. The purpose of combination of two images into a single image is for determination of the existence and location of the edges. The value of threshold in the above process is used for detection of edge pixels [10].By the use of new matrices an algorithm was developed for finding the edges and then, a MATLAB function, which is called as Sobel 5×5.m, was implemented in MATLAB. This MATLAB function required a grayscale intensity image, two-dimensional array. The result returned by the function include the edge pixels denoted by white color. According to [5,17] author presented a new approach for detection of edges in a noisy image. In this work wavelet threshold denoising approach and sobel edge detection are combined together. But drawback of these approaches is due to combination of two approaches. Here large hardware unit with high time complexity is required. Similarly according to [17] a novel colour edge detection algorithm is described. An improved Kuwahara filter was used in this algorithm.

Previous existing edge detection approaches faced many problems. Some methods faced the quality issue. The detected edges were not fine enough. The detected edges were not very clear and tiny but meaningful changes were left undetected. Some algorithms faced the problem of time complexity. The time taken by the algorithms for computation of the edges were very large. Some of the approaches faced the issue of area complexity and power consumption. The power consumption by some of them was large..Some approaches were based only on gray scale images. Some were based on RGB images.

`

**Proposed Methodology**

In this paper we will develop a fast algorithm and architecture for RGB color image based noisy edge detection. At initial stage we will design RGB Edge detection algorithm using matrix reduction technique. The proposed method will be implemented on MATLAB to investigate the required time thoroughly to detect edges within an object and compare output image with various parameters. Here we will use RGB image and we will find sobel edge detection with improved filter based technique. For filtering approximate 2D Gaussian filter will be used which is proposed in [19]. After that we will check the efficiency of our proposed algorithm by the use of image parameters. After that we will design the architecture of our proposed algorithm using Verilog in Xilinx 14.2 ise. In this work we are trying to reduce the time complexity issue with reduction in area and energy consumption. There are some image quality parameters which we will use for quality analysis.

**Quality Parameter comparison:**

Various parameters are used to evaluate the proposed algorithm at both levels. The various parameters are:

PSNR (Peak signal-to-noise ratio)

SSIM (Structural-Similarity-based Image Quality Assessment) [16]

FSIM (Feature Similarity Index for Image Quality Assessment) [15].

**Proposed 3X3 Sobel Edge Detection Technique:**

In this technique at initial stage we will pass input image from the 3X3 Gaussian smooth filter. Then we will apply the generated image to the proposed edge detection technique. As we know edge detection is very useful in graphics world. As we have already seen that there are different kinds of problem faced by previous edge detection approaches. So for reduction of those problems here we will propose a new Sobel edge detection technique. According to this technique we will follow the old sobel approach but we will change the vertical mask and horizontal mask. Here we will propose a plus mask logic. According to this approach, for horizontal we will use only two pixel values and for vertical also we will use only two pixels.

The main assumption of masking is made by the consideration of the concept of interpixel correlation. The pixel values of the image are too close to each other and the variation is nearly equal to one. Instead of processing the entire pixel in 3*3 kernels, a suitable mask is applied as a filter which passes horizontal and vertical pixels as shown in below figure.

Proposed Vertical Mask:-

$$Gy = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Proposed Horizontal Mask:-

$$Gx = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Overall Proposed mask is following Diamond Structure:-

$$Gx + Gy = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Gx1=Absolute (Gx)
Gy1=Absolute (Gy)
Sobel Edge Pixel = Gx1+Gy1

Here we are using cross logic which will reduce the time complexity issue with good output image quality which is acceptable by human eye. The generated filter mask consists of few values that are to be processed. This results in fast computation and low area and power consumption at architectural level. The new filter mask has negative and positive values. By applying absolute on the result values and summing up them generates the same conventional function which reduces complexity.

**Implementation Details**

Here we will implement the proposed architecture using verilog and all simulation will be performed in xilinx 14.2. In this approach we will propose a multiplier less architecture. Comparing to previous methods our design needs only four pixels.


Fig-4: Top module of approximate 3X3Gaussian filter architecture

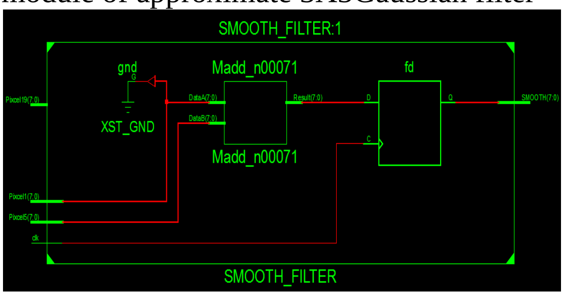


Fig-5:LUT block module of approximate 3X3 Gaussian filter architecture

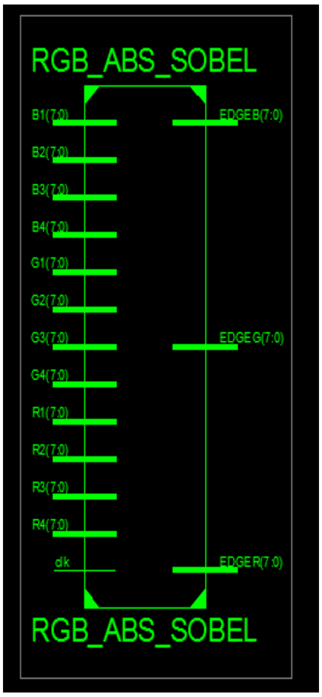


Fig-6:Top module of proposed edge detection architecture

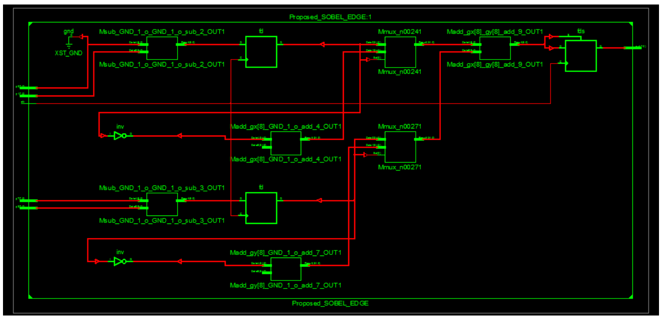

Fig-7:LUT block module of proposed edge detection algorithm

## RESULT & ANALYSIS

A new algorithm will be proposed and that algorithm will be implemented using MATLAB. For image quality measurement some scientific parameters like PSNR, SSIM, FSIM are used. We will also propose hardware unit for the proposed algorithm which will reduce the area, power and speed problem. We will compare the proposed algorithm with the previous approaches. Hardware implementation will be done using Verilog on Xilinx 14.2 simulator. The verification will be done on Model sim.

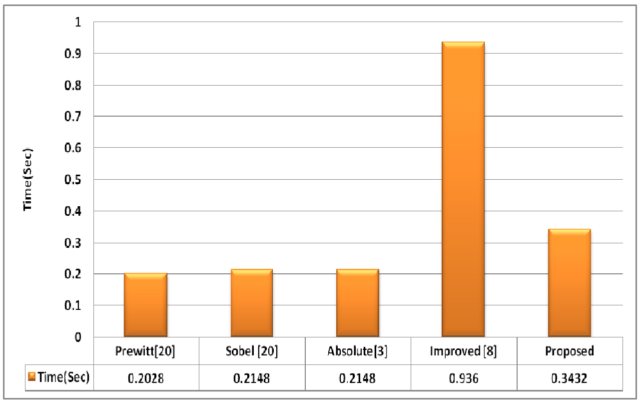


Chart-1: Time complexity analysis

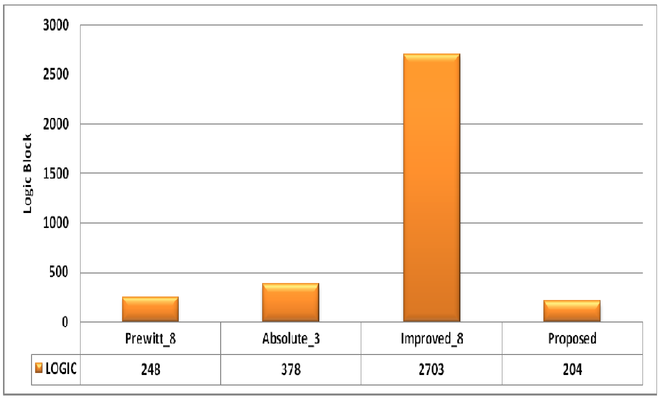


Chart-2: Comparative analysis of Logic blocks of various methods

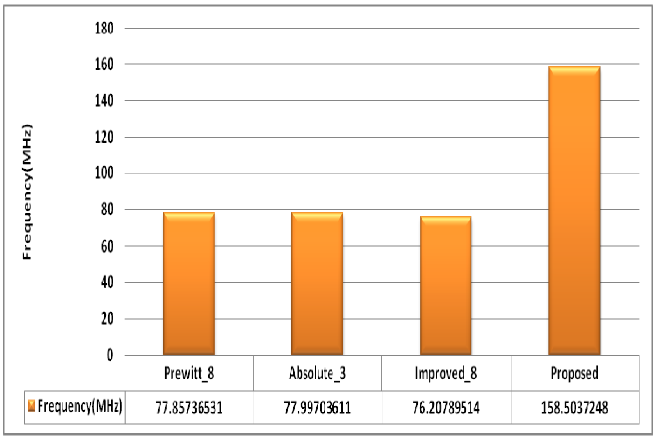


Chart-3: Comparative analysis of Frequency of various methods

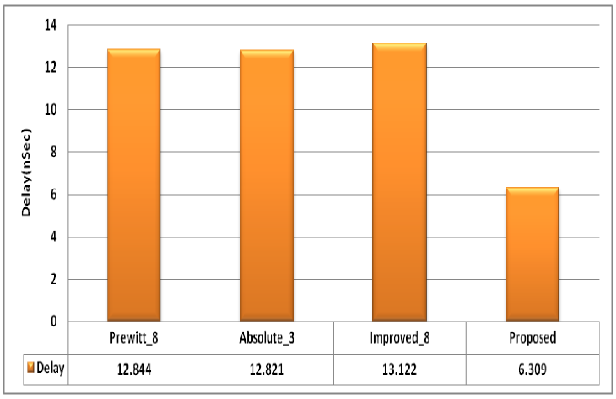


Chart-4: Comparative analysis of delay of various methods

From above results we can say that the time complexity of proposed method is less. The quality of image is maintained. The hardware requirement is less compared to other methods. Also this method works for a noisy RGB image where several other approaches fail. Thus the proposed method reduces the time , power consumped and area required and maintains the quality of the image at the same time.

**Applications**

Edge detection is a basic tool for scene analysis, image segmentation ,pattern recognition, etc. It has an important feature for image analysis. These features are used in advanced computer vision algorithm. Edge detection is used for object detection that serves different applications like medical image processing biometrics. Some of the applications of edge detection include face detection, object detection,navigation,2D to 3D conversion etc. It can be used in detection or recognition of faces, leafs, objects etc.

**Conclusions**

In this paper, basically we present RGB based edge detection calculation for RGB noisy images. According to this, we will resolve the previous existing problems which include latency, power, area. The key contribution of this work is developing a fast Edge detection algorithm. A SPAA aware error tolerant edge detection unit will be developed using this work. The proposed edge detection unit will require less area, power and speed. Here a new approach of approximation is used which will reduce some amount of accuracy. In proposed approach we will use only 4 pixel. Here we are using cross logic where we will take 2 pixel from horizontal and 2 pixel from vertical mask means basically we will propose a new mask. Using this approach we will reduce the time complexity and hardware complexity with 20-30%.

**References**

[1] Mukherjee, S.; Chatterjee, A.; Dhar, M., "Image parsing: An advanced approach for image identification," Computing and Communication Systems (NCCCS), 2012 National Conference on , vol., no., pp.1,6, 21-22 Nov. 2012.

[2] Pradabpet, C.; Ravinu, N.; Chivapreecha, S.; Knobnob, B.; Dejhan, K., "An efficient filter structure for multiplierlessSobel edge detection," Innovative Technologies in Intelligent Systems and Industrial Applications, 2009. CITISIA 2009 , vol., no., pp.40,44, 25-26 July 2009.

[3] Zhang Jin-Yu; Chen Yan; Huang Xian-xiang, "Edge detection of images based on improved Sobel operator and genetic algorithms," Image Analysis and Signal Processing, 2009. IASP 2009. International Conference on , vol., no., p.31,35, 11-12 April 2009.

[4]Zhao Chunjiang; Deng Yong, "A Modified Sobel Edge Detection Using Dempster-Shafer Theory," Image and Signal Processing, 2009. CISP '09. 2nd International Congress on , vol., no., pp.1,4, 17-19 Oct. 2009.

[5] WenshuoGao; Xiaoguang Zhang; Lei Yang; Huizhong Liu, "An improved Sobel edge detection," Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference.

[6] Hongyan Sun; ShuxueTian, "Image retrieval based on blocked histogram and Sobel edge detection algorithm," Computer Science and Service System (CSSS), 2011 International Conference on , vol., no., pp.3277,3281, 27-29 June 2011.

[7] Zhong Zhang; Geng Zhao, "Butterworth filter and Sobel edge detection to image," Multimedia Technology (ICMT), 2011 International Conference on , vol., no., pp.254,256, 26-28 July 2011.

[8] Osman, Z.E.M.; Hussin, F.A.; Ali, N.B.Z., "Optimization of Processor Architecture for Image Edge Detection Filter," Computer Modelling and Simulation (UKSim), 2010 12th International Conference on , vol., no., pp.648,652, 24-26 March 2010.

[9] Caixia Deng; Weifeng Ma; Yin Yin, "An edge detection approach of image fusion based on improved Sobel operator," Image and Signal Processing (CISP), 2011 4th International Congress on , vol.3, no., pp.1189,1193, 15-17 Oct. 2011.

[10] Gupta, K.G.; Agrawal, N.; Maity, S.K., "Performance analysis between aparapi (a parallel API) and JAVA by implementing sobel edge detection Algorithm," Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on , vol., no., pp.1,5, 21-23 Feb. 2013.

[11] El-Khamy, S.E.; Lotfy, M.; El-Yamany, N., "A modified fuzzy Sobel edge detector," Radio Science Conference, 2000. 17th NRSC '2000. Seventeenth National , vol., no., pp.C32/1,C32/9, 2000.

[12] Naqash, T.; Shafi, I., "Edge sharpening in grayscale images using modified Sobel technique," Multitopic Conference (INMIC), 2011 IEEE 14th International , vol., no., pp.132,136, 22-24 Dec. 2011.

[13] R. Biswas and J. Sil, "An Improved Canny Edge Detection Algorithm Based on Type-2 Fuzzy Sets," Procedia Technology, vol. 4, no. 0, pp. 820 – 824, 2012

[14] Lin Zhang; Zhang, D.; XuanqinMou; Zhang, D., "FSIM: A Feature Similarity Index for Image Quality Assessment," Image Processing, IEEE Transactions on , vol.20, no.8, pp.2378,2386, Aug. 2011

[15] Zhou Wang; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P., "Image quality assessment: from error visibility to structural similarity," Image Processing, IEEE Transactions on , vol.13, no.4, pp.600-612, April 2004.

[16] O. Golan, S. Kiro, and I. Horovitz, "Method and System for Edge Detection,"U.S. Patent 20 120 243 793, September, 2012.

[17] Chen, Xin, and Houjin Chen. "A novel color edge detection algorithm in RGB color space." Signal Processing (ICSP), 2010 IEEE 10th International Conference on. IEEE, 2010.

[18]http://en.wikipedia.org/wiki/Edge_detection.

[19] Jaiswal, Ankur,et al."SPAA-Aware 2D Gaussian Smoothing Filter Design Using Efficient Approximation Techniques." VLSI Design (VLSID), 2015 28th International Conference on. IEEE, 2015.

[20] Sidhu, RavneetKaur. "Improved canny edge detector in various color spaces."Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2014 3rd International Conference on IEEE 2014.