

Novel Hardware Unit For Edge Detection With Comparative Analysis Of Different Edge Detection Approaches

Shinde S.G.

Professor Electronics and Telecommunication Engineering Department

TPCT'S College of Engineering, Osmanabad, (M.S.) India

Kalpana N.Hajgude

Electronics and Telecommunication Engineering Department

TPCT'S College of Engineering, Osmanabad, (M.S.) India

Abstract- An edge in an image is a contour across which the brightness of the image changes abruptly. In image processing, an edge is often interpreted as one class of singularities. Edge detection is an important task in image processing. It is a main tool in pattern recognition, image segmentation, and scene analysis. An edge detector is basically a high pass filter that can be applied to extract the edge points in an image. This topic has attracted many researchers and many achievements have been made.. Many researchers provided different approaches based on mathematical calculations which some of them are either robust or cost effective. A new algorithm will be proposed to detect the edges of image with increased robustness and throughput. Using this algorithm we will reduce the time complexity problem which is faced by previous algorithm. We will also propose hardware unit for proposed algorithm which will reduce the area, power and speed problem. We will compare our proposed algorithm with previous approach. For image quality measurement we will use some scientific parameters those are PSNR, SSIM, FSIM. Implementation of proposed algorithm will be done by Matlab and hardware implementation will be done by using of Verilog on Xilinx 14.1 simulator. Verification will be done on Model sim.

Keywords: edge, robustness, time complexity, area, power, speed, matlab

1. Introduction:

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. Edges can be created by shadows, texture, geometry, and so forth. Edge points are to be associated with the boundaries of objects and other kinds of changes. Edges within an image generally occur at various resolutions or scales and represent transitions of different degree, or gradient levels. There are many ways to perform edge detection. However, most of them may be grouped into two categories, namely, gradient based edge detection and Laplacian-based edge detection. In the gradient based edge detection, we calculate an estimate of the gradient magnitude using the smoothing filter and use the calculated estimate to determine the position of the edges. In other words the gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. In the Laplacian method we calculate the second derivative of the signal and the derivative magnitude is maximum when second derivative

is zero. In short, Laplacian method searches for zero crossings in the second derivative of the image to find edges.

Various edge detection algorithms have been developed in the process of finding the perfect edge detector. Some of the edge detection operators that are discussed in this thesis are Robert, Prewitt, Sobel, FreiChen and Laplacian Of Gaussian (LOG) operators. Prewitt, Sobel and FreiChen are 3x3 masks operators. The Prewitt masks are simpler to implement than the Sobel masks, but the later have slightly superior noise suppression characteristics. LOG is a more complicated edge detector than the previous mentioned operators. In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also

are possible. An image defined in the "real world" is considered to be a function of two real variables, for example, $a(x,y)$ with a as the amplitude (e.g. brightness) of the image at the real coordinate position (x,y) .

Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers. The goal of this manipulation can be divided into three categories:

1. Image Processing (image in -> image out)
 2. Image Analysis (image in -> measurements out)
- Image Understanding (image in -> high-level description out)

An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve color rendition. Sequence of image processing.

Most usually, image processing systems require that the images be available in digitized form, that is, arrays of finite length binary words. For digitization, the given image is sampled on a discrete grid and each sample or pixel is quantized using a finite number of bits. The digitized image is processed by a computer. To display a digital image, it is first converted into analog signal, which is scanned onto a display.

Closely related to image processing are computer graphics and computer vision. In computer graphics, images are manually made from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from natural scenes, as in most animated movies. Computer vision, on the other hand, is often considered high-level image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans).

In modern sciences and technologies, images also gain much broader scopes due to the ever growing importance of scientific visualization (of often large-scale complex scientific/experimental data). Examples include microarray data in genetic research, or real-time multi-asset portfolio trading in finance. Before going to processing an image, it is converted into a digital form. Digitization includes sampling of image and quantization of sampled values. After converting the image into bit information, processing

is performed. This processing technique may be Image enhancement, Image restoration, and Image compression.

Image enhancement:

It refers to accentuation, or sharpening, of image features such as boundaries, or contrast to make a graphic display more useful for display & analysis. This process does not increase the inherent information content in data. It includes gray level & contrast manipulation, noise reduction, edge christenings and sharpening, filtering, interpolation and magnification, pseudo coloring, and so on.

Image restoration:

It is concerned with filtering the observed image to minimize the effect of degradations. Effectiveness of image restoration depends on the extent and accuracy of the knowledge of degradation process as well as on filter design. Image restoration differs from image enhancement in that the latter is concerned with more extraction or accentuation of image features.

Image compression:

It is concerned with minimizing the number of bits required to represent an image. Application of compression are in broadcast TV, remote sensing via satellite, military communication via aircraft, radar, teleconferencing, facsimile transmission, for educational & business documents, medical images that arise in computer tomography, magnetic resonance imaging and digital radiology, motion, pictures, satellite images, weather maps, geological surveys and so on.

Image processing is defined as the manipulation of image representation stored on a computer. Operations on images that are considered a form of image processing include zooming, converting to gray scale, increasing or decreasing image brightness, red-eye reduction in photographs, edge and shape detection of an object and analysis of object properties such as size and color. These operations typically involve iteration over all individual pixels in an image.

In image processing system, an image is usually processed in the following phases.

1. Edge Detection
2. Shape Detection
3. Object (shape) analysis by calculating different physical properties represented by the shape.

1.1. Introduction to edge detection:

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same

problem of finding discontinuities in 1D signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection examples of operators such as Canny, Sobel etc and feature extraction.

Points in an image where brightness changes abruptly are called edges or edge points. There are different types of sharp changing points in an image. Edges can be created by shadows, texture, geometry, and so forth. Edges can also be defined as discontinuities in the image intensity due to changes in image structure. These discontinuities originate from different features in an image. Edge points are to be associated with the boundaries of objects and other kinds of changes. Edges within an image generally occur at various resolutions or scales and represent transitions of different degree, or gradient levels. Edge detection refers to the process of identifying and locating sharp discontinuities in an image. There are many ways to perform edge detection. However, most of them may be grouped into two categories, namely, gradient based edge detection and Laplacian-based edge detection. In the gradient based edge detection, we calculate an estimate of the gradient magnitude using the smoothing filter and use the calculated estimate to determine the position of the edges. In other words the gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. In the Laplacian method we calculate the second derivative of the signal and the derivative magnitude is maximum when second derivative is zero. In short, Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge map detected from its original image contains major information, which only needs a relatively small amount of memory space to store. The original image can be easily restored from its edge map.

1.2. Effects of noise on edge detection:

Edge detection is susceptible to noise. This is due to the fact that the edge detectors algorithms are designed to respond to sharp changes, which can be caused by noisy pixels. Noise may occur in digital images for a number of reasons. The most commonly studied noises are white noise, "salt & pepper" noise and speckle noise. The effect of noise on edge detection

1.3. Algorithms used for Edge Detection

Edge detection refers to the process of identifying and locating sharp discontinuities in an image. Edge detection technique is usually applied on gray-scale image. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of objects in a scene. Classical

methods of edge detection involve convolving the image with an operator (a 2-D filter), which is constructed to be sensitive to large gradients in the image while returning values of zero in uniform regions. There are an extremely large number of edge detection operators available, each designed to be sensitive to certain types of edges. Variables involved in the selection of an edge detection operator include:

Edge orientation: The geometry of the operator determines a characteristic direction in which it is most sensitive to edges. Operators can be optimized to look for horizontal, vertical, or diagonal edges.

Noise environment: Edge detection is difficult in noisy images, since both the noise and the edges contain high-frequency content. Attempts to reduce the noise result in blurred and distorted edges. This results in less accurate localization of the detected edges. Operators used on noisy images are typically larger in scope, so they can average enough data to discount localized noisy pixels.

Edge structure: Not all edges involve a step change in intensity. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity. The operator needs to be chosen to be responsive to such a gradual change in those cases. Newer wavelet-based techniques actually characterize the nature of the transition for each edge in order to distinguish, for example, edges associated with hair from edges associated with a face.

In the discussions that follow, the word derivative will refer to a spatial derivative of image pixel color value, unless otherwise specialized. There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories:

Gradient: The gradient method detects the edges by looking for the maxima and minima in the first spatial derivative of the image. Mathematically, the gradient of a two-variable function (here the image intensity function) at each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction. Spatial derivative for a 7x7 matrix of pixel intensity values is shown in the Table 2.1. Partial derivative towards x and y direction of the pixels (intensity values) are calculated first by applying an operator. Then gradient valued $f(x; y)$ is calculated from these partial derivatives. Clearly, the derivative shows a maximum

located at the center of the edge in the original signal. This method of locating an edge is characteristic of the 'gradient filter' family of edge detection filters and includes the Sobel method. A pixel location is declared an edge location if the value of the gradient exceeds some threshold. The original image can be easily restored from its edge map.

Various edge detection algorithms have been developed in the process of finding the perfect edge detector. Some of the edge detection operators that are discussed in this thesis are Robert, Prewitt, Sobel, FreiChen and Laplacian Of Gaussian (LOG) operators. Prewitt, Sobel and FreiChen are 3x3 masks operators. The Prewitt masks are simpler to implement than the Sobel masks, but the later have slightly superior noise suppression characteristics. LOG is a more complicated edge detector than the previous mentioned operators.

Let us see these operators:

Sobel Operator:

Sobel operator is a pair of 3x3 convolution kernels as shown in Fig1. The second kernel is obtained by rotating the first by 90; the two kernels are orthogonal to each other. These kernel values are designed for maximum response to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. One can apply kernels separately to the input image, in order to produce separate measurements of the gradient component in each orientation (known as Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by Equation . The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by,

$$\theta = \tanh^{-1} \frac{G_x}{G_y}$$

The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Although typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

which is much faster to compute. And the angle of orientation is given by:

$$\theta = \arctan \left(\frac{G_y}{G_x} \right) - 3\pi/4$$

In this case, orientation 1 is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anticlockwise from this.

Prewitt Operator:

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images. At each point in the image, the result of the Prewitt operator is the corresponding gradient vector. As can be seen, the result of all three (Robert's Cross, Sobel and Prewitt) operators at an image point which is in a region of constant image intensity is a zero vector and at a point on an edge is a vector which points across the edge, from darker to brighter values.

Laplacian of Gaussian Filter

The Laplace operator, also known as Laplacian, is a 2-D isotropic (i.e. does not depend on the direction) measure of the second spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed in order to reduce noise. The operator normally takes a single gray level image as input and produces another gray level image as output.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}; \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

The Laplacian L(x; y) of an image with pixel intensity values I(x; y) is given by:

$$L(x, y) = \frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2}$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. Three commonly used small kernels are shown in Figure. Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step. In fact, since the convolution operation is associative, we can convolve

the Gaussian smoothing filter with the Laplacian filter first and then convolve this hybrid filter with the image to achieve the required result. This approach has advantages: Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations. The LoG ('Laplacian of Gaussian') kernel can be precalculated in advance so only one convolution needs to be performed at run-time on the image.

Canny's Edge Detection Algorithm

The Canny edge detection algorithm was developed to improve the existing method of edge detection. The first and most obvious is low error rate: it is important that edges occurring in images should not be missed and that there be no response to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first two were not substantial enough to completely eliminate the possibility of multiple responses to an edge. Based on these criteria, the Canny edge detector first smooths the image to eliminate noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis, shown in the fig. is used to track along the remaining pixels that have not been suppressed. Canny edge follows two threshold values. If the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the two thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above the lower threshold value. Canny edge detection has been improved by using type-2 fuzzy sets. The two significant features of this method are improvement of NMS (non-maximum suppression) and double thresholding of the gradient image. Under poor illumination, the region boundaries in an image may become vague, creating uncertainties in the gradient image.

Type-2 fuzzy based Canny edge detection technique handles uncertainties and automatically selects the threshold values needed to segment the gradient image

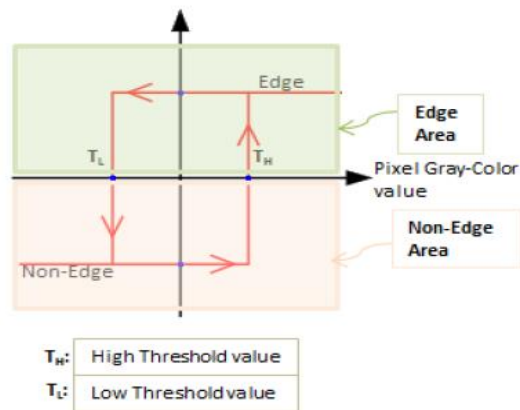


fig.1.: Hysteresis representing two threshold values

The results show that the algorithm works significantly well on different benchmark images as well as medical images (hand radiography images). The current image edge detection methods are mainly differential operator technique and high-pass filtration. Among these methods, the most primitive of the differential and gradient edge detection methods are complex and the effects are not satisfactory. The widely used operators such as Sobel, Prewitt, Roberts and Laplacian are sensitive to noises and their anti-noise performances are poor. The Log and Canny edge detection operators which have been proposed use Gaussian function to smooth or do convolution to the original image, but the computations are very large. We mainly used the Sobel operator method to do edge detection processing on the images which have been disturbed by white Gaussian noises. It has been proved that the effect by using this method to do edge detection is very good and its anti-noise performance is very strong too accuracy

2. Literature Review:

2.1 Pseudo codes for Sobel Edge Detection

Input: A Sample Image.

Output: Detected Edges.

Step 1: Accept the input image.

Step 2: Apply mask G_x, G_y to the input image.

Step 3: Apply Sobel edge detection algorithm and the gradient.

Step 4: Masks manipulation of G_x, G_y separately on the input image.

Step 5: Results combined to find the absolute magnitude of the gradient.

Step 6: The absolute magnitude is the output edges.

2.2How It Works:

In theory at least, the operator consists of a pair of 3x3convolution kernels as shown in Figure 2. One kernel is simply the other rotated by 90°. This is very similar to the Roberts Cross operator.

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

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

Gx

Gy

fig.2.: Sobel convolution kernels

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these Gx andGy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

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

Typically, an approximate magnitude is computed using:

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

which is much faster to compute.

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

$$\theta = \arctan(Gy/Gx)$$

P ₁	P ₂	P ₃
P ₄	P ₅	P ₆
P ₇	P ₈	P ₉

fig.3.Pseudo-convolution kernels used to quickly compute approximate gradient magnitude

Using this kernel the approximate magnitude is given by:

$$|G| = |(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times P_8 + P_9)| + |(P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4 + P_7)|$$

a	b	c
d	e	f
g	h	i

the directional derivative estimate vector G was defined such as density difference /distance to neighbor. This vector is determined such that the direction of G will be given by the unit vector to the approximate neighbor. Note that, the neighbors group into antipodal pairs: (a,i), (b,h), (c,g), (f,d). The vector sum for this gradient estimate:

$$G = \frac{(c-g)}{R} \cdot \frac{[1,1]}{R} + \frac{(a-i)}{R} \cdot \frac{[-1,1]}{R} + (b-h) \cdot [0,1] + (f-d) \cdot [1,0]$$

where, $R = \sqrt{2}$. This vector is obtained as

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

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

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

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

1	0	1
-2	0	2
-1	0	1

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

These masks are used by the edge detection function in the following section. Each direction of Sobel masks is applied to an image, and then two new images are created. One image shows the vertical response and the other shows the horizontal response. Two images combined into a single image. The purpose is to determine the existence and location of edges in a picture. So lets have a look on literature review :

1.This image combination is explained that the square of created masks pixel estimate coincidence each other as coordinate are summed. Thus new image on which edge pixels are located obtained the value which is the squared of the above summation. The value of threshold in this above process is used to detect edge pixels [10]. An algorithm is developed to find edges using the new matrices and then, a matlab function, which is called as Sobel5x5.m, is implemented in matlab. This matlab function requires a grayscale intensity image, two-dimensional array. The result which is returned by this function is the final image in which the edge pixels are denoted by white

color. According to [1],[3],[8] author presents hardware structure of the sobel smooth filter. According to [1] the proposed Sobel edge detection operator is model using of Finite State Machine (FSM) which executes a matrix area gradient operation to determine the level of variance through different of pixels In this paper author used the old Sobel edge detection concept. Drawback of this paper is that paper I present old existing approach in hardware form there is no any novelty as algorithm form. Due to using of old approach there is large hardware unit is require.

2. According to [3] author presents a kind of parallel processing construction of Sobel edge detection enhancement algorithm. In this paper author used the old Sobel edge detection concept. In this work author use in the place of square unit he used absolute technique which will reduce some amount of timing and hardware complexity. Drawback of these approach is also present modified old existing approach in hardware. Due to using of modified old approach there is still large hardware unit is require. In [8] author proposed a new approach for detection of edge on noisy image. In this work they combine sobel edge detection and wavelet threshold de noising approach. But drawback of these approach is due to combination of two approach there is large hardware unit with height time complexity is required.

3.Implementation Details:

In these section we present implementation of some existing edge detection approach like Prewitt Operator [20], Canny edge Detection [11],[20], sobel Edge detection [3],[20].

Implementation of prewitt operation[20]: This approach is implemented by using of matlab. According to that approach we will calculate horizontal and vertical gradient using horizontal and vertical prewitt mask.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$

Implementation of Canny edge detection[11],[20]: This approach is implemented by using of mat lab. According to that we have to follow these five steps:

1. Apply Gaussian filter to smooth the image
2. Find the intensity gradients of the image

3. Apply non-maximum suppression.
4. Apply double threshold.
5. Track edge by hysteresis

Implementation of Sobel Edge detection[3],[20]: This approach is implemented by using of matlab. According to these we have to calculate horizontal and vertical gradient by using of sobel mask. At last step we have to calculate modulus by using of square root approach, But in [3] these modulus approach is replaced by using of absolute approach which is present in [3].

Implementation of Improved Sobel Edge detection[11]: This approach is implemented by using of matlab. According to these we have to reduce noise by using of de-noising algorithm which is known as 2D and WAVELET THRESHOLD DE-NOISING than we have to calculate horizontal and vertical gradient by using of sobel edge detection approach. This approach will generate a good quality output image.

Hardware Implementation: Proposed algorithm will also propose an novel hardware unit with using of Verilog HDL.

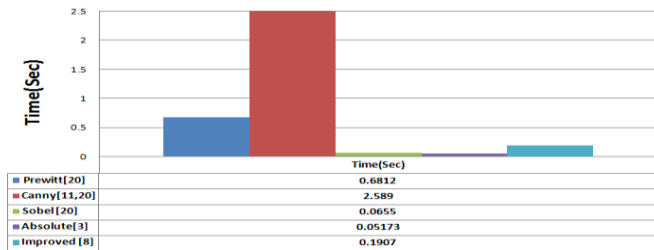
4. Result & Analysis:

A new algorithm will be proposed and that algorithm will be implemented by using of MATLAB. We will also propose hardware unit that will make justice with SPAA metrics. As compared to previous approach proposed algorithm will reduce hardware complexity and implementation will be done by using of Verilog on Xilinx 14.1 simulator. Verification will be done on Model sim. Most important is that this proposed approach will reduce timing complexity:

Time Complexity Analysis:

parameter	Prewitt	Canny	Sobel	Absolute	Improved
Time(sec)	0.6812	2.589	0.0655	0.05173	0.1907

Comparative Timing Analysis:



5. Conclusion:

The key contribution of this work is to develop a fast Edge detection algorithm. Using this work we will develop a SPAA aware error tolerant Edge detection Unit. This proposed edge detection unit will require less area, power and speed. In this approach we will propose a new approach of approximation which will reduce some amount of accuracy. In proposed approach we will use only 4 pixel. Using this approach we will expect that it will reduce the timing complexity and hardware complexity with 30-40%.

References:

1. Yasri, I.; Hamid, N. H.; Yap, V. V., "Performance analysis of FPGA based Sobel edge detection operator," *Electronic Design*, 2008. ICED 2008. International Conference on , vol., no., pp.1,4, 1-3 Dec. 2008
2. Nita, I.; Costachioiu, T.; Lazarescu, V.; Seceleanu, T., "Multiprocessor real time edge detection using FPGA IP cores," *Intelligent Computer Communication and Processing (ICCP)*, 2011 IEEE International Conference on , vol., no., pp.331,334, 25-27 Aug. 2011
3. Zhengyang Guo; Wenbo Xu; ZhiLei Chai, "Image Edge Detection Based on FPGA," *Distributed Computing and Applications to Business Engineering and Science (DCABES)*, 2010 Ninth International Symposium on , vol., no., pp.169,171, 10-12 Aug. 2010
4. 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
5. Pradabpet, C.; Ravinu, N.; Chivapreecha, S.; Knobnob, B.; Dejhan, K., "An efficient filter structure for multiplierless Sobel edge detection," *Innovative Technologies in Intelligent Systems and Industrial Applications*, 2009. CITISIA 2009 , vol., no., pp.40,44, 25-26 July 2009
6. 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., pp.31,35, 11-12 April 2009
7. 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
8. Wenshuo Gao; Xiaoguang Zhang; Lei Yang; Huizhong Liu, "An improved Sobel edge detection," *Computer Science and Information Technology (ICCSIT)*, 2010 3rd IEEE International Conference on , vol.5, no.,
9. Hongyan Sun; Shuxue Tian, "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
10. 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
11. 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
12. 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
13. 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
14. 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

15. 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

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

16. Lin Zhang; Zhang, D.; Xuanqin Mou; 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

17. 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

18. O. Golan, S. Kiro, and I. Horvitz, "Method and System for Edge Detection," U.S. Patent 20 120 243 793, September, 2012

19. http://en.wikipedia.org/wiki/Edge_detection

IJSER