ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE)
Vol 3, Issue 7, July 2016

FPGA Implementation of Distributed Canny Edge Detector for Low Clarity Images

[1] Rinjo A J, [2] Remya K P
Department of Electronics and Communication Engineering,
Nehru College of Engineering and Research Centre,
Pampady, University of Calicut, Kerala, India

Abstract: **Edge detection is the most common preprocessing step in many image processing algorithms. Edge detection is the method to find 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 purpose of edge detection is to reduce the data in an image. Canny detector has high latency, because it's a frame level processing. So in order to avoid that problem new canny edge detection developed. It's a block level processing. The entire image divides into block. In the original canny compute the high and low threshold values based on the frame level.. In the new distributed canny detector have a histogram, which helps to give more clarity to the image. The synthesis tool used here is Xilinx ISE 14.2. Using hardware description language (Verilog) the system can implement on Spartan 6 FPGA.**

Keywords: **Histogram, canny edge, Xilinx.**

I. INTRODUCTION

An edge may be defined as the two disjoint regions in animage. Edge detection is basically, a method of segmenting an image into regions of discontinuity. Edge detection has an important role in image processing. Most of the edge detection grouped into two, gradient and Palladian. The gradient method detects the edges by calculating the maximum and minimum in the first derivative of the image. The Palladian method searches for zero crossings in the second derivative of the image to find edges.In the original canny method, the computation of the high and low threshold values depends on the statistics of the whole input image. However, most of the above existing implementations ([2]-[4], [5]-[7]) use the same fixed pair of high and low threshold values for all input images. This new canny edge detector works on the VLSI platform. There are lots of advantages while doing this canny on the VLSI. In the VLSI, it can be easily change the algorithm at any stage, but in mat lab it is impossible. It is power efficient, less latency, good efficiency etc.

Histogram equalization is use for removing the noise and get clear image; however it is very useful for scientific images like thermal, satellite or x-ray images etc. Also histogram equalization can produce undesirable effects (like visible image gradient) when applied to images with low color and noise image

II EDGE DETECTION TECHNIQUES

Robert, Sobel, Prewitt are the major edge detection techniques. These are very easy and high sensitive to noise. This is called first order edge detection or gradientbasededgeoperator.

1, Sobel operator:This operator consists of 3×3 convolution matrix. This matrix applied separately to the

input image, to get the gradient component in each orientation (call Gx and Gy).

$$KGx = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$KGy= \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

These can then be combined together to find the magnitude of the gradient at each point of the image. The gradient magnitude and direction is given by

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

International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE)
Vol 3, Issue 7, July 2016

or

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

2.Roberts cross operator: The Roberts Cross operator performs a simple, quick to compute, 2-D spatialgradient measurement on an image. The operator consists of 2x2 convolution matrix.

$$\underset{KG_x}{\begin{pmatrix} +1 & 0 \\ 0 & -1 \end{pmatrix}} \quad \underset{KG_y}{\begin{pmatrix} 0 & +1 \\ -1 & 0 \end{pmatrix}}$$

These can then be combined together to find the magnitude of the gradient at each point of the image. The gradient magnitude and direction isgiven by

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

or

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

3. Prewitt operator:Prewitt operator is similar to the Sobel operator and is used for detectingvertical and horizontal edges in images

$$\underset{KG_X}{\begin{pmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{pmatrix}} \quad \underset{KG_Y}{\begin{pmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}}$$

III CANNY EDGE DETECTION ALGORITHAM

The Canny edge detector is a technique that uses to detect a wide range of edges in images. It consists of following steps [8]:

- ❖ Smoothening- It is inevitable that all images taken from a camera will contain some amount of noise. Toprevent that noise is mistaken for edges, noise must be reduced. Therefore the image is firstsmoothed by applying a Gaussian filter.
- ❖ Calculating the horizontal gradient Gx and vertical gradient Gy at each pixel location.
- ❖ Computing the gradient magnitude G and direction ø at each pixel location.
- ❖ Applying Non-Maximal Suppression (NMS) to thin edges.
- ❖ Computing high and low thresholds. The high threshold is calculated by $(1-P_1)$ and the low threshold value is 40% of the high threshold value.
- ❖ Performing hysteresis thresholding to determine the edge map. The Gradient magnitude above the high threshold value is strong edge and the gradient value below the low threshold value is weak edge.

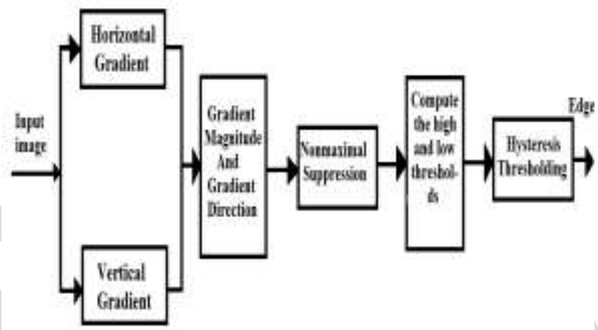

Fig1. Block diagram of the canny edge detection algorithm

IV PROPOSED DISTRIBUTED CANNY EDGE DETECTION ALGORITHM

The original canny edge detector is a frame level processing. The new detector is a block level processing. Which means the entire image divides into blocks, and applies separate processing for each block. So it gives more clarity to the edges. In this proposed system a histogram is applied. The histogram is used for giving clarity to the image. Due to lack of light, the clarity of image will low. So in order to overcome that problem histogram equalization is used. The histogram equalization is applied at the beginning of the processing.

The given image divides into mom non overlapping blocks. These m × m blocks will feed into the histogram. The histogram will count the data in an organized way. It will count the data from 0-255. After getting the result, it will predict the image color distribution

1. Histogram Equalization: It is a contrast adjustment technique. Consider a discrete gray scale image $\{x\}$. The probability of an occurrence of a pixel of level i in the image is

IFERP

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE)

Vol 3, Issue 7, July 2016

$$P_x(i) = P(x=i) = \frac{ni}{n},\ 0 \le i < L$$

Where n_i be the gray level occurrence I, L be the total number of gray level. (Generally 256), n be total number of pixel and $P_X(i)$ being image's histogram for pixel value i, generally [0, 1].

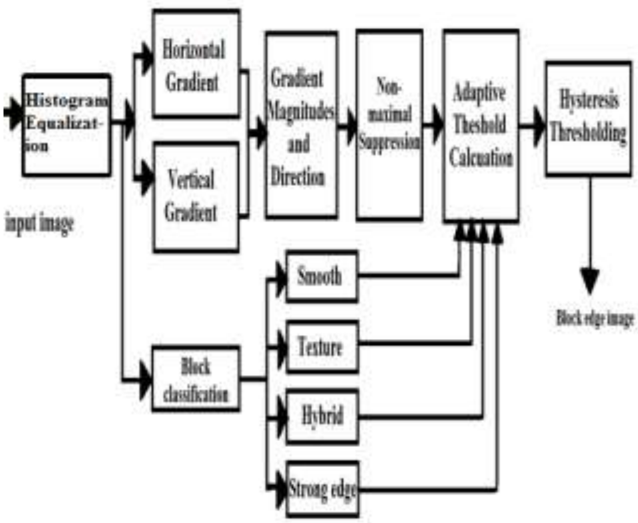

Fig. 2: Proposed distributed canny edge detection algorithm

Cumulative distributive function of P_x is given by

$$Cdf_x(i) = \sum_{j=0}^{i} Px(j)$$

The cdf must be normalized to [0,255]. The general histogram equalization formula is:

$$h(v) = \text{round}\left(\frac{cdf(v) - cdfmin}{(M \times N) - cdfmin} \times (L-1)\right)$$

Wherecdfmin is the minimum non-zero value of the cdf,M × N gives total pixel of the imageand L be the number of grey levels used (generally 256).

2. *Block classification*: The block classification unit consists of two stages; Stage 1 performs pixel classification while stage 2 performs block classification. In pixel classification, the pixel distribution calculated according to the variance of the pixel. Variance is calculated as follows:

$$var = \frac{1}{8}\sum_{i=1}^{8} (x_i - \bar{x})^2$$

Where $x_i$is the mean value is the pixel intensity and $\bar{x}$is the mean value.

Step 1: Pixel classification

$$\text{Pixel type} = \begin{cases} uniform, & var(x,y) \le Tu \\ texture, & Tu < var(x,y) \le Te \\ edge & Te < var(x,y \end{cases}$$

Step 2: Block classification

Block type	No. of pixels of pixel type	
	$N_{uniform}$	N_{edge}
Smooth	≥0.3 Total_Pixel	0
Texture	<0.3 Total_Pixel	0
Edge	<0.65(Total_Pixel-$N_{edge)}$	(>0)&(<0.3Total_Pixel)
Medium edge	≥0.65(Total_Pixel-$N_{edge)}$	(>0)&(<0.3Total_Pixel)
Strong edge	≤ 0.7Total_Pixel	≥0.3 Total Pixel

Var(x, y): the local (3×3) variance at pixel (x, y);
Tu and *Te*: two thresholds, *Tu* =100; *Te*=900;
Total Pixel: the total number of pixels in the block;
$N_{uniform}$: the total number of uniform pixels in the block; $N_{edge:}$ the total number of edge pixels in the block;

Let P1 be the percentage of pixels, in a block, that would be classified as strong edges.

Step 1: If smooth block type
P1 = 0; /* No edges*/
else if texture block type
P1 = 0.03; /* Few edges*/
else if texture/edge block type
P1 = 0.1; /* some edges*/
else if medium edge block type
P1 = 0.2; /* Medium edges*/
else
P1 = 0.4; /* Many edges*/

Step 2: Compute the 8-bin non-uniform gradient magnitude Histogram and the corresponding cumulative distribution function F (G).

Step 3: Compute High threshold as

F (High threshold)= 1-P1

Step 4: Compute Low threshold = 0.4*High threshold

The pixel classified into 3, uniform, texture and edge. The uniform and edge pixel will count using the counters. The result is the $N_{uniform}$ andN_{edge} . According to the $N_{uniform}$ andN_{edge} the blocks will classify. For each type block there will be a P1 value, which is used for calculating the threshold values.

3. Gradient and Magnitude Calculation: To find the Gradient and Magnitude the entire image divides into 3×3 overlapping block. To find the x and y direction gradient these 3×3 blocks will multiplied with the matrix KGx and KGy [1].

$$KGx = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$KGy = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

The gradient magnitudes (also known as the edge strengths) can then be determined as aEuclidean distance measure by applying the law of Pythagoras Equation.

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

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

Where:Gx and Gy are the gradients in the x- and y-directions respectively, G is the magnitude.

4. Directional Non Maximum Suppression(NMS): The horizontal and vertical gradient and the gradient magnitude are fetched from local memory 2, 3 and 1, respectively; and used as input to the Arithmetic unit. According to horizontal gradient G_x and the vertical gradient G_y, two intermediate gradients M1 and M2 will calculate. The M1 and M2 values will give to the Arithmetic unit.This arithmetic unit consists of one divider, two multipliers and one adder.

$$M_1\,(x,y) = (1\text{-}d).M(x+1,y\text{-}1)+d.M(x+1,y);$$

$$M_2\,(x,y) = (1\text{-}d).M(x\text{-}1,y\text{-}1)+d.M(x\text{-}1,y);$$

Where: $d=G_{y(x,y)}\,/G_{x(x,y)}$

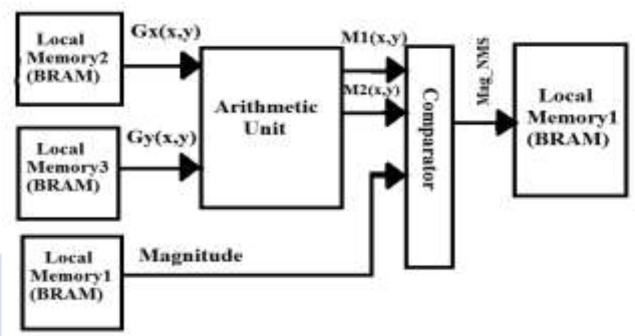

Fig 3: Directional Non Maximum Suppression Unit

Finally, the output of the arithmetic unit is compared with the gradient magnitude of the center pixel.The gradient of the pixel that does not correspond to a local maximum gradient magnitude is set to zero.The latency between the first input and the first output is 20 clock cycles and the total execution time is $m\times m$ +20.

5. Calculation of Thresholds: NMS is used for finding the threshold values, this unit can be pipelined with the directional NMS unit. Besides, the P1value, which is determined by the block classification unit, the mag_max, and mag_min, which are determined by the gradient and magnitude calculation unit, are the inputs for this unit.

The arithmetical unit can compute the corresponding R_i which is the high threshold T_H. Finally the low threshold $T_L$is 40% of the high threshold.

IFERP
ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE)
Vol 3, Issue 7, July 2016

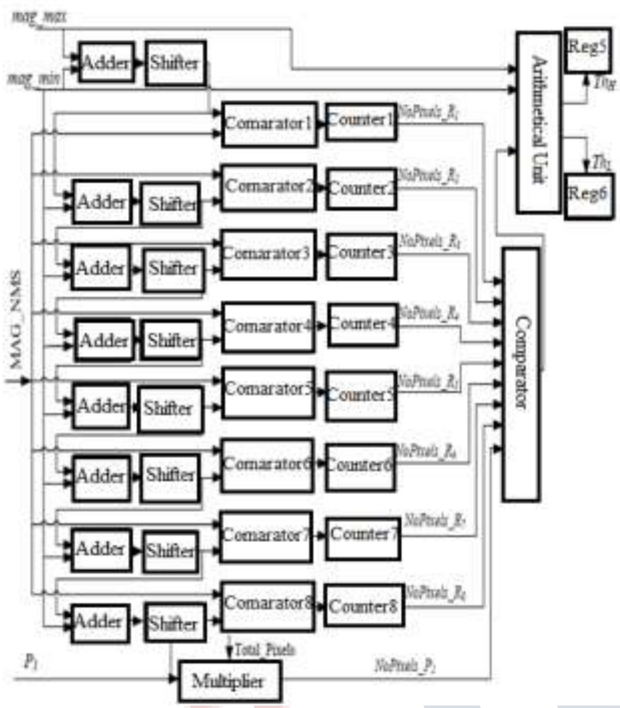

Fig 4: The architecture of thresholds calculation unit.

6. Thresholding with Hysteresis: The NMS value will compare with the high and low threshold values T_H and T_L. The NMS value below the T_H will change to 0 and the value above the T_L value wills change to 0. The remaining values will change to the maximum pixel value, typically 255.

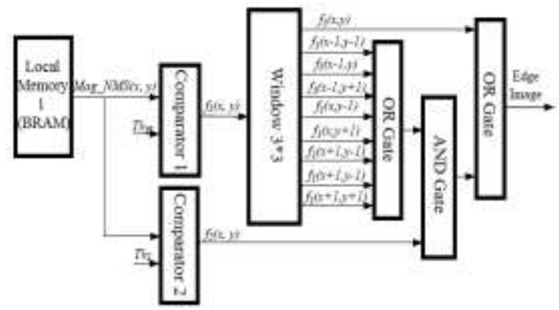

Fig 5: Pipelined architecture of thresholding with hysteresis

a ***b*** ***c***

Fig 6: (a) image with noise, (b) histogram output, (c)edge images

V SIMULATION OUTPUT

Fig 7: Output waveform of Histogram equalization

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE)
Vol 3, Issue 7, July 2016

Fig 8: Vertical gradient magnitude

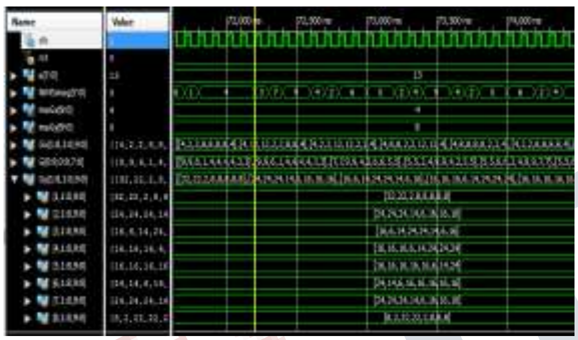

Fig 9: Horizontal gradient magnitude

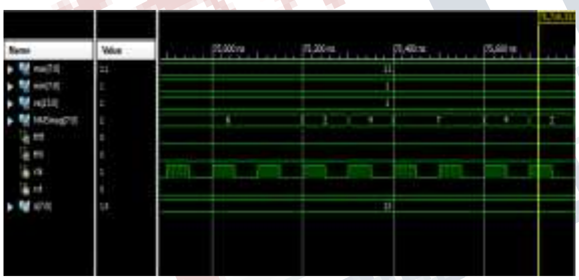

Fig 10: Output waveform of Directional NMS unit

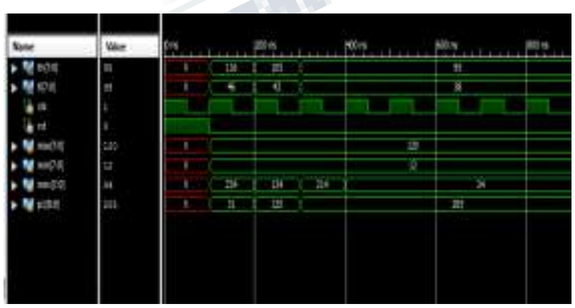

Fig 10: Output waveform of Threshold unit

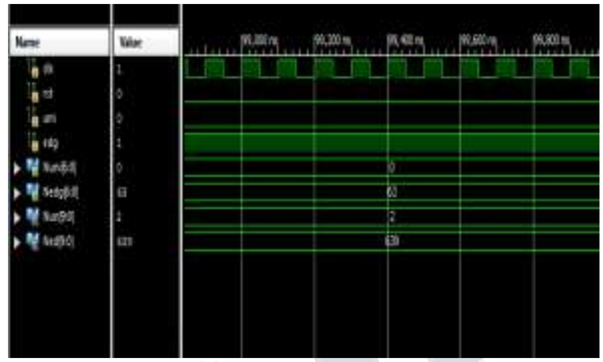

Fig 11: Output waveform of Block classification

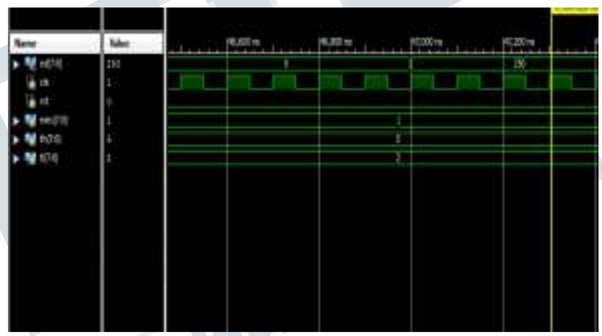

Fig 12: Output waveform of Hysterias unit

(a) (b) (c)

Fig 13(a) Low clarity input image (b) histogram output (c) Edge of the input image

VI CONCLUSION

The original canny algorithm is a frame level processing to detect the high and low threshold value. In the proposed canny edge detection algorithm, it's a block level processing. In this Histogram equalization is applied for getting good clarity image. The histogram equalization will remove the noise in the image. In the future the shake image edges can be detected. The proposed system can applied for finger print detection, face reorganization etc.

REFERENCES

[1]. QianXu, SrenivasVaradarajan, ChaitaliChakrabarti Distributed Canny Edge Detector: Algorithm and FPGA Implementation,*Fellow, IEEE*, and Lina J. Karam, *Fellow, IEEE*

[2]. D. V. Rao and M. Venkatesan, "An Efficient Reconfigurable Architecture and Implementation of Edge Detection Algorithm using Handle-C," *IEEE Conference on Information Technology: Coding and Computing* (*ITCC*), vol. 2, pp. 843 – 847, Apr. 2004.

[3]. H. Neoh, A. Hazanchuck, "Adaptive Edge Detection for Real-Time Video Processing using FPGAs," *Application notes*, Altera Corporation, 2005. Online at http://www.altera.com/

[4]. C. Gentsos, C. Sotiropoulou, S. Nikolaidis, N. Vassiliadis, "Real-Time Canny Edge Detection Parallel Implementation for FPGAs," *IEEE International Conference on Electronics, Circuits and Systems* (*ICECS*), pp. 499-502, Dec. 2010

[5]. Y. Luo and R. Duraiswami, "Canny edge detection on nvidiacuda," *Computer Vision and Pattern Recognition Workshop*, vol. 0, pp. 1–8, 2008.

[6]. R. Palomar, J. M. Palomares, J. M. Castillo, J. Olivares, and J. G´omez- Luna, "Parallelizing and optimizing lip-canny using nvidiacuda," *ser. IEA/AIE'10*, Berlin, Heidelberg: Springer-Verlag, pp. 389–398, 2010.

[7]. L.H.A. Lourenco, "Efficient Implementation of Canny Edge Detection Filter for ITK Using CUDA," *13th Symposium on Computer Systems*, pp 33-40, 2012.

[8]. J.F. Canny, "A Computation Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no 6, pp. 769-798, November 1986.

[9]. J.K. Su and R.M. Mersereau, "Post-processing for Artifact Reduction in JPEG-Compressed Images," *IEEE International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP*), vol. 3, pp. 2363-2366, 1995.

Rinjo A J received the B.Tech in Electronics and Communication Engineering Degree from Thejus Engineering College, Vellarakkad, under the University of Calicut, Kerala, India in 2013, and doing M.Tech degree (2014-16) in VLSI Design Department of ECE, from Nehru College Of Engineering and Research Centre, Pampady, under the University of Calicut, Kerala, India

Co-author:

Remya K P Assistant Professor Nehru College Of Engineering and Research Centre, Pampady, Thrissur, Kerala, India. B.Tech in CUSAT. M.Tech in college of engineering Chegannur