ISSN 2348 – 9928
IJAICT Volume 2, Issue 6, June 2015 Doi:01.0401/ijaict.2014.03.03 Published Online 05 (6) 2015

DEVELOPMENT OF POST PROCESSING IP CORE FOR ULTRASOUND IMAGING

Ms. C. Arunima
PG Scholar,
Vivekanandha College of Engineering for Women,
Namakkal, Tamilnadu, India.

Mr. Jayaraj. U. Kidavu
Scientist,
National Institute of Electronics & Information Technology, Calicut, India.

Mr. Rony Mathew
M.Tech Embedded Systems,
National Institute of Electronics & Information Technology, Calicut, India.

Abstract-***Edge detection is one of the most fundamental algorithms in digital image processing. The Canny edge detector is the most common edge detection algorithm because of its ability to detect edges even in images that are highly contaminated by noise. But, this is a time consuming algorithm and hence its implementations are difficult to reach real time response speeds. Especially,in the present days where the demand for high resolution real time image processing is constantly increasing, the need for efficient and fast edge detector implementations is very necessary. In order to obtain maximum efficiency a modified canny algorithm is used in this project which explores the parallel implementation possibilities of the algorithm. A highly parallel and pipelinedarchitecture is used from which real time responses are possible. It is implemented in Field Programmable Gate Array (FPGA).***

Keywords — Canny Edge Detection, FPGA, Virtex-6

I INTRODUCTION

Among the currently available medical imaging modalities, ultrasound imaging is considered to be non-invasive, practically harmless to the human body, portable, accurate, and cost efficient. These properties have made the ultrasound imaging the most prevalent diagnostic tool around the world. Ultrasound imaging is regularly used in cardiology, gynaecology, abdominal imaging, etc. Unfortunately, the quality of medical ultrasound (as defined by image resolution and contrast) is generally limited due to a number of factors, which arise both from physical phenomena underlying the image acquisition and imperfections of the imaging system design.These undesirable physical effects should be compensated by using efficient signal processing tools.Among some of the techniques that allow to study in detail about the image acquired by the ultrasound technique is the edge detection, which is described in this paper. It is clear, from both biological and computational evidence that some form of data compression occurs at an early stage in image processing. Moreover, there is much evidence pointing that one form of this compression involves finding edges and other information high features in images. Edges often appear at points where there is a large variation in the luminance values in an image pixel, and consequently they indicate the edges, or boundaries, of the objects. However, large luminance variations may also correspond to surface markings on objects. Points of discontinuities in the luminance signal (rather than simple discontinuity) can also indicate an object boundary in the scene.The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images [1]. It was developed by John in 1986.

However, this is a time consuming algorithm and therefore its implementations are difficult to reach real time response speeds. Especially in the present day where the demand for high resolution image processing is constantly increasing, the need for faster and more efficient edge detector implementations is ever so present.In order to obtain maximum efficiency a modified canny algorithm is used in this project which explores the parallel implementation possibilities of the algorithm. A highly parallel and pipelined architecture is used from which real time responses are possiblethe parallel architecture is then used to form an efficient hardware design of the algorithm [2]. It is implemented in Field Programmable Gate Array (FPGA).

As of now, it is imperative to explore hardware implement - ations for real time response. Recent advances in FPGA technology not only provides significant increase in the logic real states, but also furnishes relatively significant amount of flexible internal RAM modules[4]. The motivation in designing the hardware modules of canny edge detector was to reduce its complexity, enhance the performance and to make it suitable for development on a reconfigurable logic block.

© 2015 IJAICT **(www.ijaict.com)**

II CANNY ALGORITHM

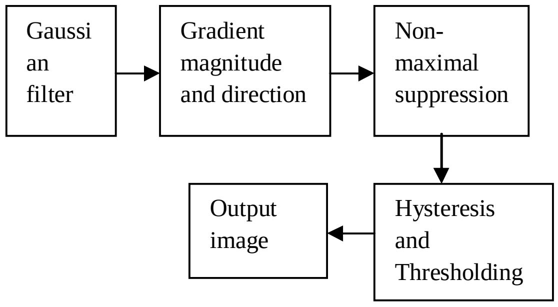


Fig 1 : Canny algorithm

It is inevitable that all images taken from a camera will contain some amount of noise. To prevent that the noise is mistaken for edges, noise must be reduced. Because the Canny edge detector is susceptible to noise present in raw unprocessed image data, it uses a filter based on a Gaussian (bell curve), where the original image is convolved with a Gaussian filter. The kernel of a 5X5 Gaussian filter with a standard deviation of σ = 1.4.

After smoothing the next step is the calculation of the gradient of the image. The gradient calculation leads to the detection of the possible edge strength operator. The most commonly used gradient operators are Sobel and Prewitt operators. The Sobel operator uses a pair of 3x3 convolution masks, one for estimating the gradient in the x-direction and the other for estimating the gradient in the y-direction[5].

An edge in an image may point in a variety of directions, so new implementation uses the mask [-1 0 1] to compute first differences in four directions: H(horizontal), V(vertical), D_1 and D_2 (diagonal).The X and Y components of the gradients are computed by projecting the diagonal differences onto the axes.

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

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

Fig 2 : Sobel masks

The gradient magnitudes (also known as the edge strengths) can then be determined as a Euclidean distance measure by applying the law of Pythagoras as shown in Equation. Gradient magnitude and orientation are measured using the following equations:

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

An image of the gradient magnitudes often indicates the edges quite clearly. However, the edges appear typically broad and thus doing not indicate exactly where the edges are. In order to make it possible to determine this, the direction of the edges is determined and stored to use for further processing as shown in Equation below.

$$\theta = \arctan(G_y | G_x)$$

The next step, the non-maximal suppression.The purpose of this step is to convert the “blurred” edges in the image of the gradient magnitudes to “sharp” ones. This can be done by preserving all local maxima in the gradient image, and deleting everything other than that. The algorithm is for each pixel in the gradient image[6]. Given the estimates of the image gradients, a search is then done to determine if the gradient magnitude assumes a local maximum in the gradient direction. The non-maximum suppression scheme can described as follows. The gradient magnitude is then non-maximum suppressed in the gradient direction.

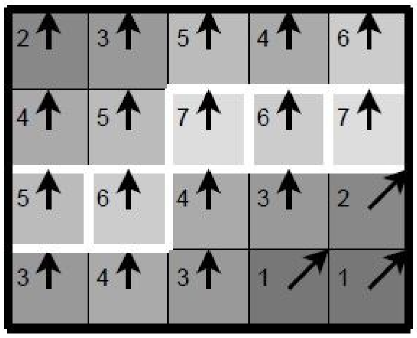


Fig 3 : Example of Non-maximal Suppression

A simple example of non-maximum suppression is shown in Figure 7. The scheme which compares the point $P_{x,y}$ with two of its neighbours only. The edge strengths are indicated as colors and numbers, while gradient directions are indicated as arrows. Almost all pixels have gradient directions pointing north. Therefore they are compared with those pixels above and below. The pixels which turn out to be maximal in this comparison are marked with white borders. All other pixels will be suppressed. The resulting edge pixels are marked with white borders.

Once this process is complete the result is a binary image where each pixel is marked as either an edge pixel or a non-edge pixel. From the complementary output from the edge tracing step, the binary edge map acquired in this way may also be treated as a set of edge curves, which after further processing may be represented as polygons in the image domain.

III MODIFIED CANNY OPERATOR

New implementation methods of canny operator are proposed with the introduction of parallel breakpoints detection and edge tracing without recursive operations. By using GPU's parallel processing characteristics; processing speed has been significantly raised.

3.1 Improved Canny Operator

As mentioned above, what we paid attention is to avoid the recursion algorithm which is completely serial as much as possible. Thus, breakpoints detection and edge tracing are introduced the last step of the canny edge detection algorithm above as two improvements which avoid the recursive operations.

3.2 Breakpoint Detection

The more appropriate method of the edge tracing should be begun with the breakpoints, which is more meaningful to the following processing. Therefore, three kinds of breakpoints are considered from the image after the proceeding of non-maximum suppression.

3.4 Linear Breakpoint

As shown in Fig 4, B represents the breakpoint pixel; PI and P2 represent the adjacent pixels of the edge from straight and diagonal directions respectively. As can be seen from Fig.2-(3), if only one pixel which confined as a part of the edge is located in PI positions, B will be confined as a breakpoint directly. On the other hand, if there exists only one pixel in P2 positions, to prevent the edge inflection points from being considered as a breakpoint, B will be considered as a breakpoint only if P3 positions are not all occupied by edge pixels.

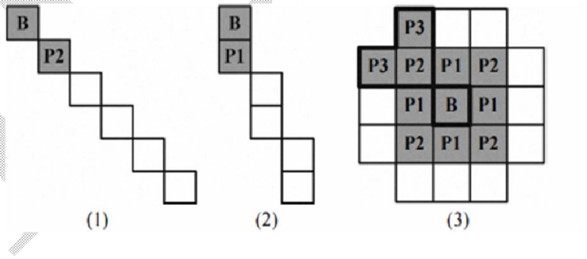


Fig 4 : Schematic diagram of Linear Breakpoint

3.5 Triangle Breakpoint

As shown in Figure 5, B represents the breakpoint pixel and constitutes a triangle shape with PI and P2 adjacent to it.

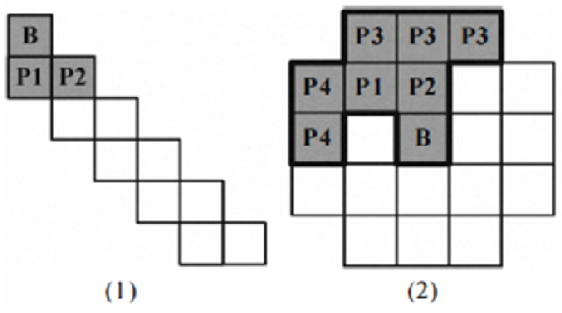


Fig 5 : Schematic diagram of Triangle Breakpoint

© 2015 IJAICT **(www.ijaict.com)**

3.6 Square Breakpoint

As shown in Fig 6, B represents the breakpoint pixel and constitutes a square shape with PI, P2 and P3 adjacent to it. Similar to the condition of triangle breakpoints, B is acknowledged as a breakpoint if one of the groups represented by P4 or P5 does not contain any edge pixels at least.

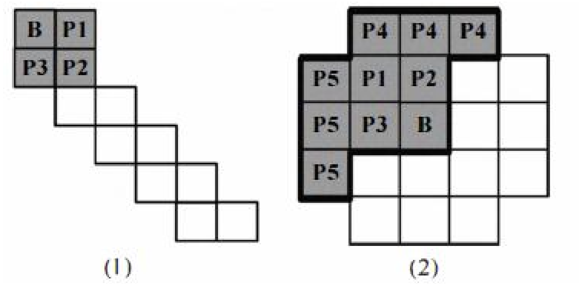


Fig 6 : Schematic of Square Breakpoint

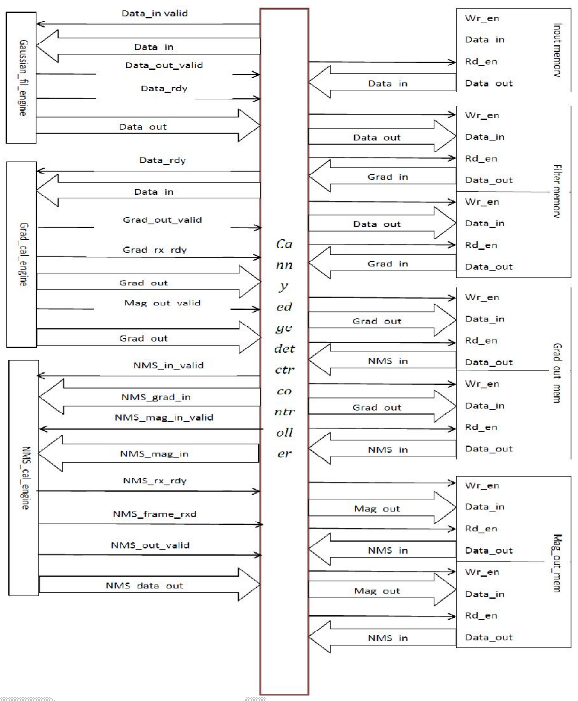
3.7 Edge Tracing

All pixels in the image can be detected in parallel. After all breakpoints in the image have been determined, edge tracing algorithm is used to connect these breakpoints belonging to the same edge. Here, the tracing process begins with those breakpoints in parallel and stops when another edge pixel is met or no adjacent pixels whose gradient magnitude is greater than the low threshold value can be found. Hence, branch tracing, the result of the recursive operations will not occur.

IV. MODIFIED CANNY ARCHITECTURE

Fig 7 shows the hardware architecture for the canny edge detection system..It consists of different processing blocks, input, output and intermediate memories and a controller to control the data transfer between different processing blocks and memories.

The main processing blocks in the canny edge detection system architecture are: Gaussian engine which read input image from the FIFO Input_img_mem and, perform Gaussian filtering with a mask of size 3x3 for removing the high frequency noise content and place the result in the Img_filtered_mem. Grad_cal_engine calculate gradient magnitude and gradient orientation corresponding to each pixel for the content in the Img_filtered_mem and place the result gradient magnitude and orientationin Gradient out_mem and which is a FIFO.



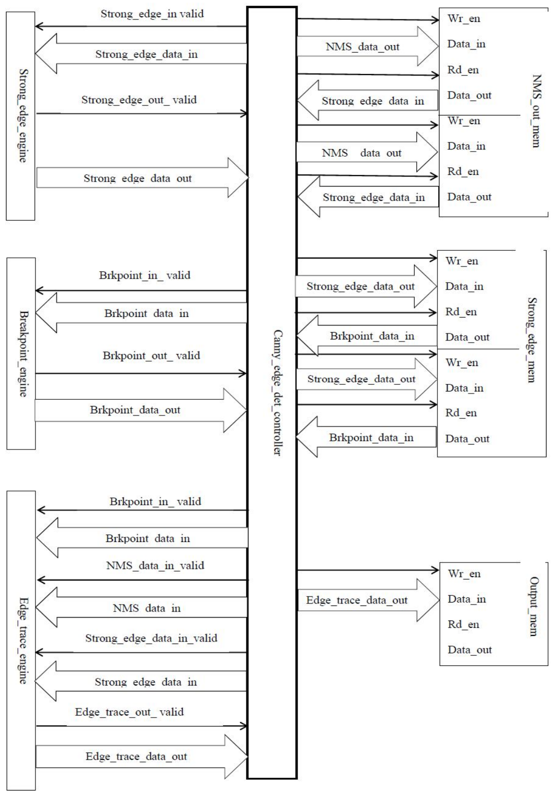


Fig 7 : Modified Canny architecture

© 2015 IJAICT **(www.ijaict.com)**

transmitter issues Physical Layer Packets (PLPs) which are terminated at the physical layer of the receiver, such PLPs are used during the Link Training and Initialization process. In this process the link is automatically configured and initialized fornormal operation; no software is involved. During this process the following features are defined: link width, data rate of link, polarity inversion, lane reversal, bit/symbol lock per lane, and lane-to-lane deskew (in case of multi-lane link) [2].

V RESULT

Obtained the results of canny edge detection algorithm in MATLAB contains 4 steps which are Smoothing, find gradients, on maximum suppression and hysteresis thresholding.

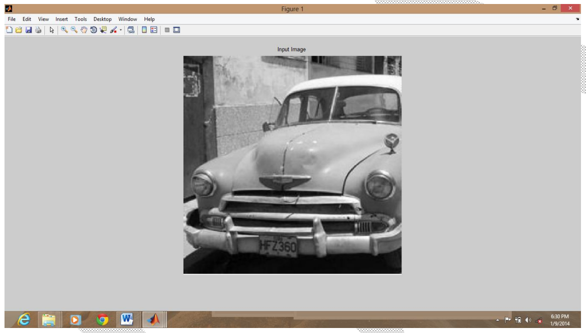
Algorithm

Step 1: Read input image

Step 2: Convolve a given image with a Gaussian of scale 'sigma'

Step 3: Location of the edges with non-maximal suppression

Step 4: Design the thresholding for edges with the hysteresis to eliminate spurious responses



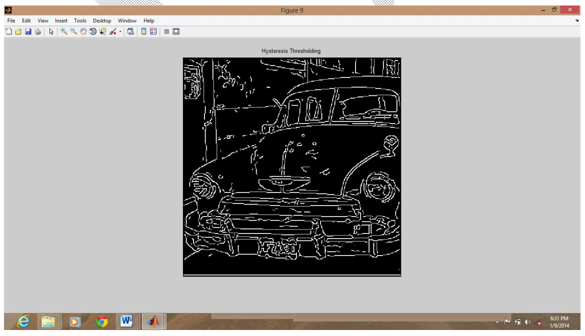


Fig 8: Output after hysteresis and thresholding

VI CONCLUSION

The purpose of edge detection in general is to significantly reduce the amount of data in any given image, while preserving the structural properties to be used for further image processing. Several algorithms exist. Here in this project am using the canny edge detection algorithm, and also going to make this algorithm as the future works.

References

[1] Chandrashekar.N.S, Department of Ece, Don Bosco Institure of Technology, Bangalore, Dr.K.R.Nataraj, Departmetn of Ece, SJB Insitirue of Technology, Bangalore, "A Distributed Canny Edge Detector and Its Implementation on FPGA" ,2012.

[2] Shengxiao Niu, Jingjing Yang, Sheng Wang, Gengsheng Chen "Improvement and Parallel Implementation of Canny Edge Detection Algorithm Based on GPU",2011 IEEE 9th International Conference on ASIC (ASICON),

[3] Hui-Ii Zhao, Guo-feng Qin and Xing-jian Wang, "Improvement of Canny Algorithm Based on Pavement Edge Detection", 2010 3rd International Congress on Image and Signal Processing, voL2, p.964-967 (2010).

[4] Ogawa, K., Ito, Y. and Nakano, K., "Efficient Canny Edge Detection Using a GPU", 2010 First International Conference on Networking and Computing, p.279-280 (2010).

[5] L. Masek, "Recognition of Human Iris Patterns for Biometric identification", Thesis Report, The University of Western Australia, 2003.

[6] Pong P Chu "FPGA prototyping by verilog examples"John Wiley.

© 2015 IJAICT **(www.ijaict.com)**