Real Time Implementation of Ede Detection Technique for Angiogram Images on FPGA

SAJANA UMMER[1], SREEKUMAR R.[2]

[1]*Mtech Student, Dept. Of Ece, Gec Idukki*
[2] *Asst. Professor, Dept. Of Ece,Gec Idukki, Kerala,India*

---***---

Abstract - *An X-ray test that takes pictures of the blood flow in blood vessels is termed as angiography. An angiogram can look at the arteries near the heart, lungs, brain, head, arms, legs and the aorta. An angiogram is prepared to detect problems with veins and arteries that affect blood flow. Examples of such problems include a tear in a blood vessel, pattern of blood flow to a tumor, check how bad atherosclerosis is in the coronary arteries, or check for blocks in coronary arteries etc. Analysis of the angiogram images uses different image processing techniques such as image enhancement, segmentation etc. Also such a high end analysis requires high speed and a large amount of accuracy. This paper detects edges of vessels in an angiogram image, by means of a new algorithm on FPGA and compares the results with existing classical image processing algorithms. Steps involved are Histogram equalization technique, a new algorithm for Canny edge Detector. Exploiting the parallelism in algorithms yields considerable speedup in running times. The proposed design is coded in Verilog HDL and synthesized using ISE design suite 14.2 and implemented on ARTIX-7 FPGA.*

***Key Words*: Angiography, Image segmentation, Image enhancement, Canny edge detector, Histogram Equalization, FPGA**

1. INTRODUCTION

Detection and analysis of blood vessels in an angiogram image depends upon segmentation. One of the important image segmentation processes is the edge detection. An image processing technique for finding the boundaries of objects within images is termed as edge detection. It works by detecting discontinuities in brightness. Edge detection has prime importance in digital image processing. Edge detection removes worthless data, noise, and frequencies while extracting vascular regions from other background regions in an image. It is a process of partitioning an image into several non- overlapping regions. Based on the results obtained, surfaces of vasculatures can be extracted, and used in the detection of vascular diseases. Developing a consistent and accurate image segmentation method is therefore important for making angiography useful and proficient.

Various edge detection techniques Are available for image extraction etc. Each technique is constructed to be shrewd for firm types of edges. Among them Prewitt, Sobel, Robert, Canny are the foremost techniques. Canny edge detector is one which is commonly used due to its superior performance and accurate edge detection. Even though the accuracy of conventional canny edge detector is high, it is computationally very complex and very susceptible to noise. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may, therefore, be substantially simplified. Our aim is to reduce the latency and computational complexity without sacrificing the edge detection performance. Existing Canny algorithms fails to detect accurate edges for a less contrast images [1,9]. In such situations the proposed algorithm has a wide application. The new algorithm has reduced computational complexity and reduced delay and a good edge detection performance

2. Literature Survey

Many implementations of the Canny algorithm was proposed on a wide list of hardware platforms. There is a set of work on Deriche filters that have been derived using Canny's criteria and implemented on ASIC-based platforms. A highly efficient algorithm which uses a recursive filter for edge detection is presented [2].

The cost is still too high for real time implementation on FPGA circuits. A design which is optimized both algorithmic and architectural aspects of the original Deriche filter. A new organization of the filter is proposed at the 2D and 1D levels which reduce the memory size and the computation cost by a factor of two for both software and hardware implementations [3].

The canny edge detection is able to significantly outperform existing edge detection techniques due to its superior performance. Unfortunately, the implementation of the systems in real time is computationally complex, high hardware cost with increased latency. The algorithm in this paper uses approximation methods to replace the complex operations; the pipelining is employed to reduce the latency [4].

In this paper edge detection on a Real Time Image Capture is implemented. The video is captured and then interfaced with FPGA kit and the edge detected image is shown in VGA display. In addition, Canny Edge Detection algorithm is used to obtain the edge detected version of the RGB image captured by the interfaced camera [5].

An edge detection method for vascular angiogram tree, which is an important step in computer aided diagnosis on vascular diseases. It combines classical edge detection algorithms except canny to detect edge points and uses the connection for edge points based on the shortest path [6].

We propose a completely computerized method to enhance the angiogram images and detect the arteries. The method contains three steps which are Hessian filter enhancement, feature extraction and vessel region detection [7].

An image processing technique to detect calcified plaques hampering to diagnose vessel lumen from CCTA (Coronary Computed Tomography Angiography) image using MSCT (Multi-Slice Computed Tomography) is presented [8].

3. Proposed System

The beginning stages of image processing identify the features in images that are significant in estimating the structure and properties of objects. Edges are important features for analyzing images. It occurs on the boundary between two different regions in an image. Edge pixels are defined as the pixels at which the intensity profile in an image function changes abruptly. Edges are the set of connected edge pixels. Edge detection is probably the first step in recovering information from images. The methodology used in this paper as follows:

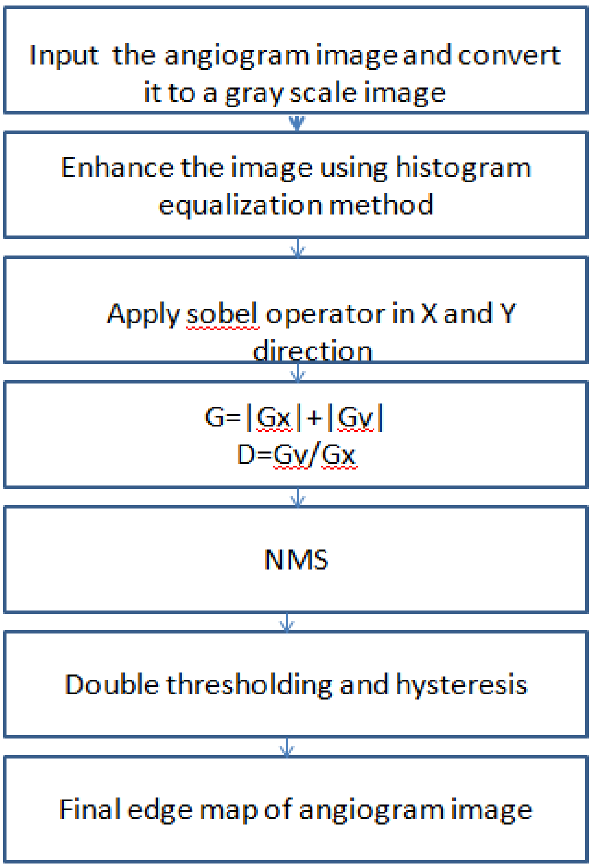


Fig -1: flow chart of propose system

Proposed algorithm:

Step 1: Read the given angiogram image, and convert it into gray scale image.

Step2: Apply HE to enhance the image.

Step3: Apply Sobel operator to find gradients in X and Y direction and calculate gradient magnitude and direction.

Step4: Apply Non maximum suppression to preserve the local maxima and discard the local minima.

Step5: Apply double thresholding to find out which are all the edge points in the final output .

Step6: Obtain the hysteresis to get final edge map of the angiogram image.

Fig -1 shows the flow chart of the propose system. First the image is converted to gray scale the apply pre processing steps and finally apply the edge detection algorithm to find the edge map.

3.1 Histogram Equalization

The principal objective of image enhancement is to process a given image so that the result is more suitable than the original image for a specific application. It accentuates image features such as edges, boundaries, or contrast to make a graphic display more helpful for display and analysis. The enhancement doesn't increase the inherent information content of the data, but it increases the dynamic range of the selected features so that they can be detected easily. HE is a contrast adjustment technique. A histogram is the estimation of the probability distribution of a particular type of data.

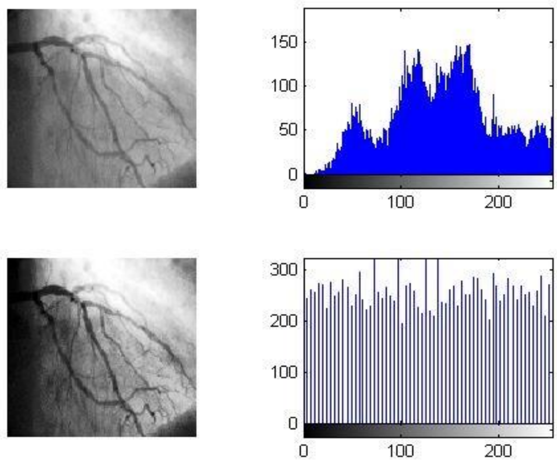


Fig-2 histogram equalization

3.1 Edge Detection

Edge detection algorithm relies on block level statistics instead of frame level statistics. The direct application of original Canny to block leads to a wrong detection because the statistics of blocks are different from that of the entire image. The proposed edge detection algorithm has mainly five steps like conventional Canny's algorithm. The difference is that the former one uses an adaptive threshold selections scheme for threshold calculation.

(i)Pixel and block classification

First, the mean and variance are calculated. According to the value of variance, each pixel is classified as uniform, texture or edge pixels. There are two threshold values Tu (for uniform pixels) and Te (for edge pixels). Here Tu=900 and Te=100.

$$pixel\ type = \begin{cases} uniform & ,var \leq Tu \\ texture & ,Tu < var \leq Te \\ edge & ,var < Te \end{cases}$$

Table -1 Block classification

Block type	No. of pixels of pixel type		Type and value of p (percentage of pixels in the block that would be classified as strong edges)
	N_uniform	N_edge	
Smooth	>=0.3*total_pixel	0	P=0 //no edges
texture	<0.3*total_pixel	0	P=0.03 //few edges
Edge/texture	<0.65*(total_pixel-N_edge)	(>0)&(<0.3*total_pixel)	P=0.1//some edges
Medium edge	>=0.65*(total_pixel-N_edge)	(>0)&(<0.3*total_pixel)	P=0.2 //medium edges
Strong edge	<=0.7*total_pixel	>=0.3*total_pixel	P=0.4 //many edges

(ii) Gradient magnitude and direction calculation

We have defined edges as abrupt changes in intensity, to detect edges all we need is to find edge pixels; the physical derivative gives rate change functions. So, derivative can be used to detect edge pixels. The first derivate sometimes called the gradient of the pixels is calculated. We use Sobel operator for gradient calculation of each pixel in X and Y directions.

$$Kgx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$Kgy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The gradient magnitudes (also known as the edge strengths) and direction can then be determined as:

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

$$d = \tan\left(\frac{Gy}{GX}\right)$$

(iii) Directional Non Maximum Suppression

The edges are made slender by converting the blurred edges of the image gradients into pointed edges. This can be accomplished by suppressing the local minima and preserving the local maxima in the gradient image. Horizontal and vertical gradient stored in the memory are the input to the NMS unit. Depending upon the direction and magnitude of four neighbourhood pixels, two intermediate magnitude values ($M_1(x,y)$, $M_2(x,y)$) are calculated according to the equation.

$$M_1(x,y) = (1-d)*M(x+1,y-1)+d*M(x+1,y-1)$$

$$M_2(x,y) = (1-d)*M(x-1,y-1)+d*M(x-1,y)$$

iv)Threshold calculation:

Threshold calculation is the decisive steps in the edge detection algorithm. The performance of Canny edge detector depend upon threshold values and so these values should be calculated with high accuracy. The input to the threshold calculation unit is the output of NMS unit. The median value of the gradient magnitude helps to find the value of thresholds and is based on the histogram of the gradient magnitude. Th=0.66*median value and low threshold Tl=42% of Th.

(v)Thresholding with Hysteresis

Non-maximum suppression technique finds the physically powerful edges and they may be affected by noise and lighting conditions. If the gradient magnitude is greater than the high threshold, then it is the strong edge pixel and those edges are conserved. If gradient magnitude is lower than the low threshold, those edges are discarded. If the gradient magnitude value lies between low and high threshold then further investigation is done. The current pixel is considered as a strong edge pixel if any of the neighbours of the current pixel is strong edge pixel; otherwise, it is suppressed ie, those who lie between these two thresholds are classified edges or non-edges based on their connectivity.

This algorithm is computationally less complex but gives accurate results if the image is corrupted by noise and for less contrast images. Also this has reduced latency and increased speed by the incorporating pipelining and parallelism in the algorithm.

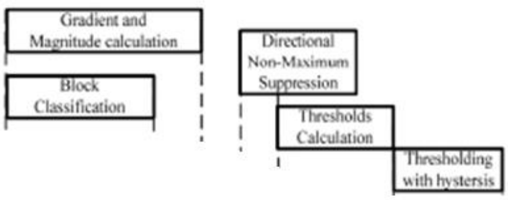
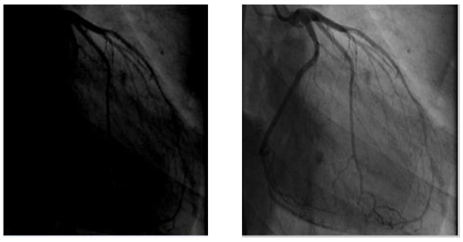


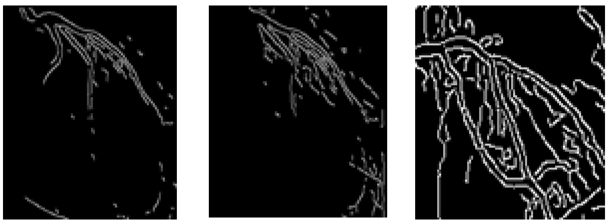
Fig-3: Parallel execution

4. Experimental Results

The proposed FPGA-based architecture can maintain multiple image sizes and block sizes. To show the performance of the proposed system, a Xilinx Artix-7 FPGA was used to process grayscale images with a block size of 64 × 64. The angiogram image is considered as the input for applying the edge detection techniques. Following figures shows the experimental results. Fig 5 shows the RTL schematic of the system. The proposed FPGA implementation takes only 0.562ms for execution.



(a) (b)



(c) (d) (e)

Fig-4: Results obtained (a) original image (b) Enhanced image(c) canny edge detection (d) distributed canny (e) proposed system.

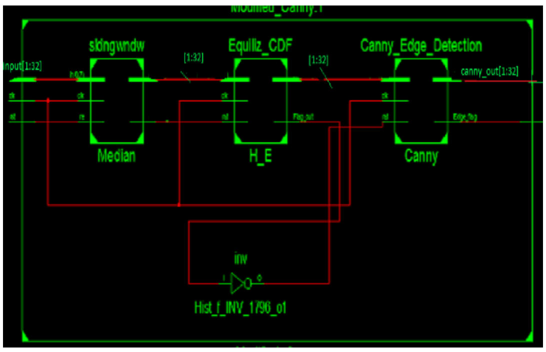


Fig-5: RTL schematic of the system

5. CONCLUSIONS

The proposed algorithm detects edges of blood vessels from an angiogram image by image processing techniques. It is observed that edges thus segmented are both precise and lucid. Steps involved in this algorithm are easy and straightforward for implementation. Results prove that detection of algorithms is proficient and useful in determining edges. The work has compared different edge detecting techniques. Also, the proposed FPGA implementation takes only 0.562ms for execution.

REFERENCES

[1] Qian Xu, Srenivas Varadarajan, Chaitali Chakrabarti and Lina J. Karam, (JULY 2014), "A Distributed Canny Edge Detector: Algorithm and FPGA Implementation", IEEE Transactions On Image Processing, VOL. 23, NO. 7, pp. 2944-2960.

[2] R. Deriche, "Using canny criteria to derive a recursively implemented optimal edge detector," Int. J. Comput. Vis., vol. 1, no. 2, pp. 167–187, 1987.

[3] F. G. Lorca, L. Kessal, and D. Demigny, (Oct. 1997), "Efficient ASIC and FPGA implementation of IIR filters for real time edge detection," Proc. IEEE ICIP, vol. 2, pp. 406–409.

[4] Sangeetha, D. and Deepa, P., (2016, January) "An Efficient Hardware Implementation of Canny Edge Detection Algorithm" VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID) IEEE, pp. 457-462.

[5] Jeyakumar, R., Prakash, M., Sivanantham, S. and Sivasankaran, K, (2015, November) "FPGA implementation of edge detection using Canny algorithm" Green Engineering and Technologies (IC-GET) IEEE, pp. 1-4.

[6] Hu, Haijiang, et al. "Simple edge detection method for vascular tree angiogram." Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2015 IEEE. IEEE, 2015.

[7] Fazlali, H. R., et al. "Vessel region detection in coronary X-ray angiograms." Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015.

[8] Yoshida, Yuki, et al. "Semi-automatic detection of calcified plaque in coronary CT angiograms with 320-MSCT." Signal Processing Conference (EUSIPCO), 2016 24th European. IEEE, 2016.

[9] **J. F. Canny,** (Nov. 1986), "A computation approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 8, no. 6, pp. 769–798.