ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 5, May 2016

Canny edge detection algorithm

AMRUTA L KABADE
II.Year, M.Tech, DSCE, Bangalore.

Dr.V.G. Sangam
DSCE, Bangalore

Abstract -In today's modern life, there is increased demand of edge detection of the image for medical and defence applications. So, it is required to study the edge detection algorithm. The various edge detection algorithms such as Prewitt, Robert, Sobel, Canny etc. are failed to meet the low area and reduced delay. Block Level Canny Edge Detection Algorithm gives simple edge detection operation which reduces the time and memory consumption.The Block Level Canny edge detection algorithm is the special algorithm to carry out the edge detection of an image. In this project work, the image frames are divided into blocks then the canny edge detector is applied. The five thresholding values are considered for the edge detection in Block Level Canny edge detector, in order to consider all the significant edges and to avoid the false edge detection in the smooth region. In the design work, the Block level canny edge detection is carried out by modifying the canny edge detection algorithm in order to meet the VLSI requirement such low area and reduced delay etc. The Block level canny edge detector is designed in Xilinx. The Xilinx is used because the design is easily synthesizable without the need of any external compiler. If the code is written in MATLAB it requires conversion of the code into Verilog HDL before synthesis and then only it is translated into a hardware device like FPGA. In this design the overlapping blocks are inputted to the edge detector using shift operation in order to reduce the number of memory accesses, so that the area and delay are reduced. The different edge detection algorithms are applied on the image using MATLAB. The obtained edge map of the various edge detectors is compared with the Block level canny edge detector based on the perceptual analysis.

I. INTRODUCTION

In this chapter the requirement of edge detection and its application and also introduction to the types of edge detection algorithm is mentioned. Edge detection is one of the significant sections of the image processing algorithms which have many applications like image morphing, pattern recognition, image segmentation and image extraction etc. As the edge is one of the major information contributors to any image, hence the edge detection is a very important step in many of the image processing algorithms. It represents the contour of the image which could be helpful to recognize the image as an object with its detected edges. In the ideal case, by applying the edge detector to an image gives the different edges that are connected to form the outline of the object. Important property of the edge detection is the detection of the exact edges along with the good orientation of the object in the image. And the memory required to store the edges of an image is less compared to the whole image even though it contains all information of the shape and orientation of the object.

Edge is portrayed from various perspectives as the inventors differentiate, one of them is "It is change either in the brightness or the colour of an image ". It is discovered that edges are basically depicted into four sorts in any image. They are step edge, ramp edge, roof edge and line edge. Many edge detection algorithms are proposed by many researchers and they are mainly classified into two types based on the order of derivative used [1]

1.Gradient Based

2.Laplacian Based

Canny edge detector is a standard edge detection algorithm for many years among the present edge detection algorithms. Image information is the main target for image processing and it is more popular from the past 30 years. An image consists of different information of a scene such as the size, colour, orientation of different objects present in that scene. An interesting point is that first the object is separated from background then all the edges has to be detected to get the outline of the object. This is the reason, the edge detection become important in computer.Vision and image processing. In the "Canny Edge Detection Algorithm" it uses only two thresholds for all the images

To get better edge map the canny edge detector is applied to the each block of an image but, it leads to the detection of false edges in the smooth region

and fails to detect some of the true edges. In order to defeat this, block level canny edge detector is proposed which could give better performance at each block of the image.

II. METHODOLOGY AND IMPLEMENTATION

In order to understand the process of edge detection some basic terms and explanation regarding an image, edge, frame, etc. has to be known, thus to fulfil this requirement this section is introduced. Then the "Canny Edge Detection Algorithm" is discussed. Due to the limitation of canny edge detection, Block Level Canny Edge detection is proposed.

Fundamentals of Image Processing:

Image processing has many applications in the field of medical and defence; hence there is more focus on image processing.The basic terms in image processing are as follows:

Image: It is the rectangular arrangement which consists of column and row wise pixel value representation. Image is two dimensional representation of the three dimensional view. The Digital image is represented as a square matrix consists pixel values. The representation is shown in figure 1.

41	40	42	43	44
40	70	72	69	68
50	34	35	76	23
67	44	60	29	33
44	254	76	65	49

Pixel value

Fig 1: Image Representation and pixel indication

The RGB image means it has three planes; they are red, blue and green. Each pixel represents the single colour. Thus if each plane pixel value ranges from 0 to 255, then whole image range up to 2553. The Gray scale image represents the pixel values in the range of 0 to 255.As it represents as shade of gray, thus it is ranging between 0 to (28-1). The pixel valuesrepresented using 8 bit binary numbers. The indexed image consists of a matrix of pixel values along with the colour map. The binary image has the pixel values as either '0' or '1'.If the bright or whiteness in the image is represented as '1' then the dark is represented as '0' or else vice versa. Depending on the application the colour maps are chosen. An image is taken as an example to illustrate different colour maps and is shown in Fig.2.For analysis and pattern recognition of image the basic step is image segmentation. Image segmentation is used to group the pixels. And the grouping is made such that adjacent groups of pixels are not identical. The image segmentation techniques are of two types. They are:

(i) Local Segmentation and
(ii) Global Segmentation.

In the Local Segmentation, the part of the image is segmented and it is economical compared to the global segmentation for the pixel data. The Global segmentation deals with the segmentation of the entire image. This segmentation contains more number of pixels compared to the local segmentation.

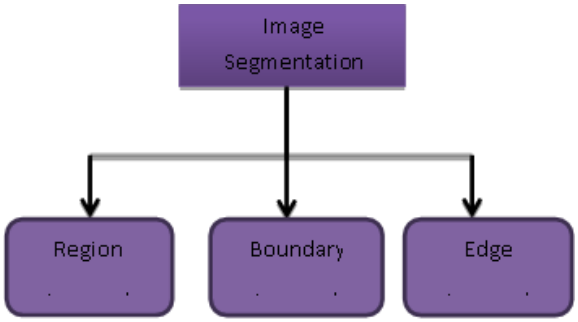

Fig 2: Classification of Image Segmentation

III.Canny Edge Detection Algorithm

From the literature survey we can observe that the "Canny Edge Detection Algorithm" is the optimal algorithm among the edge detection algorithms. The three main criteria's of the canny edge detection are as follows:

1. Low error rate: It is important that edge occurring in image should not be missed and there should be no response for non-edge.
2. Good Localization: The distance between the edge pixels as found by the detector and the actual edge is to be minimum.
3. Single Response: To have one response to a single edge. The algorithm mainly has five steps in it. They are:

Step 1: Computing the horizontal (Gx) and vertical (Gy) gradient of each pixel in an image.

Step 2: Using the above information the magnitude (G) and direction (of the each pixel in the image is calculated.

Step 3: In this step all non-maxima's are made as zero that is suppression the non- maxima's thus the step is called Non-Maximal Suppression.

Step 4: The high and low thresholds are measured using the histogram of the gradient magnitude of the image.

Step5: To get the proper edge map hysteresis thresholding is employed which will link between the weak and strong edges. The weak edges are taken into consideration if and only if it is connected to one of the strong edges or else it is eliminated from the edge map. The strong edge is the one whose pixel is greater than the high threshold and weak edge is one whose pixel value lays between high and low threshold.

Then using that, the pixel strength and orientation of that gradient is computed. In the next step it finds the all maxima's present the image then it keeps them as it and removes the other non-maxima's. The process is called as Non-Maximal Suppression. In the step 4 it makes the pixel is either the edge or non-edge, depending on the high and low thresholds set.The block diagram of the "canny edge detector" is shown in the fig. 3.

If the pixel has higher value than the high threshold then it is treated as edge. If pixel value is less than the low threshold treated as non-edge. If pixel value is in between the high and low thresholds then it is a weak edge. Thus to detect edges in the image it considers two thresholds as high and low. Then finally the hysteresis thresholding is applied which can make a decision of the detected weak edge to be considered or left. The pixel is compared with the neighbouring pixels and if the weak edge is connected to that strong edge then it is considered as edge else it is removed from the edge map. The threshold is same for all the images. Due to this it has some limitations, when applied to the block level of the image. It gives some false edges in the plain region and fails to detect the some significant edges.In order to overcome from the above limitation an adaptive thresholding block and the block classification blocks are added along with the above blocks. And the threshold is set different based on the block. Thus the performance of the proposed block level canny edge detector is improved.

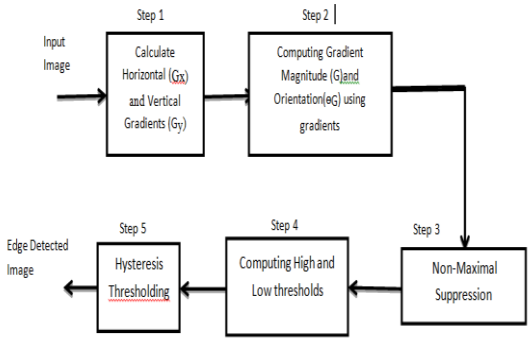

Fig 3: Block diagram of canny edge Detector

For the Edge detection, image is given as input and the edge map is obtained as output. The project uses the Verilog coding for the design. But the Xilinx doesn't allow the image data as input, as the Xilinx can work on any number systems like decimal, hexadecimal, etc. So, the image has to be first converted to the any of the number system then it can be fed to the Xilinx as input. The conversion of image to hexadecimal is done using the matlab. Image consists of a number of pixels; each pixel value indicates the intensity at that point. The matlab can accept the gray images only for processing purpose. Thus first the given input RGB image has to be converted to gray image and then image is resized, a text file is created. The pixel values are stored in that text files as equivalent hexadecimal number, in a single column for easy processing. The main idea behind the edge detection of an image is, the intensity of the pixel in an image is considered between 0 to 255. The dark portion of the image is treated as '0' and the bright portion is treated as '255'. The above block diagram indicates the steps included in the proposed design. First the obtained pixel values from the conversion of input image to the text is given as input to the shifter which will divide the obtained number of bits into 8-bit pixels and is fed to the horizontal filter that computes both horizontal gradient(Gx) and vertical gradients(Gy) are produced. Before the process of shifting first the smoothing operation is employed this will be done as shown in figure 5.

IV. RESULTS

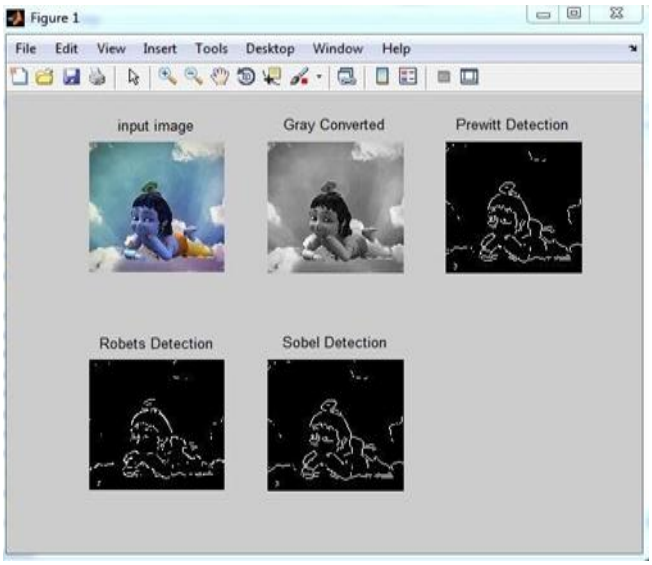

Fig 4: Different edge detection output of Krishna image

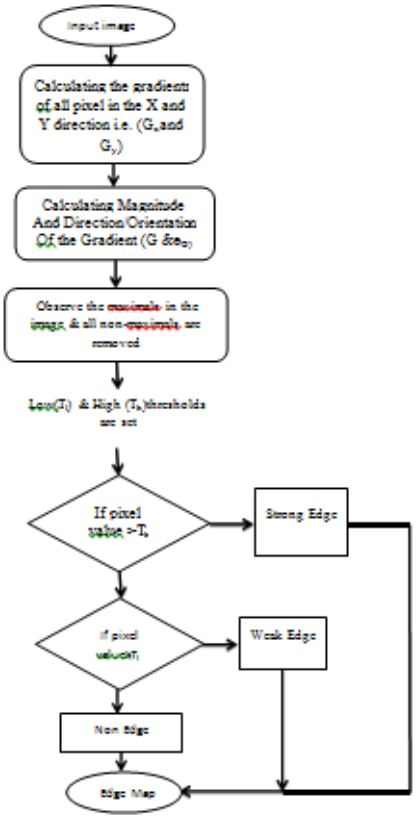

Fig5: Canny edge detector flowchart

There is a discussion about the types of the edge detection algorithms. In this section each image is taken as input and the various edge detection algorithms like Prewitt edge detector, Roberts Edge detector and Sobel edge detector are applied. The obtained edge map each algorithm is shown in the fig.4 Thus by observing the figure the improvement in the detection of the image can berealized.

REFERENCES

[1] Y. Luo and R. Duraiswami, "Can[1] James Clerk Maxwell, 1868 Digital Image Processing Mathematical and Computational Methods.

[2] BeantKaur , Anil Garg," Mathematical Morphological Edge Detection For Remote Sensing Images",*IEEE,2011*,pp: 324-327.

[3] R .Gonzalez and R. Woods, Digital Image Processing, Addison Wesley, 1992, pp 414 - 428.

[4] 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.

[5] L. Torres, M. Robert, E. Bourennane, and M. Paindavoine, "Implementation of a recursive real time edge detector using retiming technique,"in*Proc. Asia South Pacific IFIP Int. Conf. Very Large Scale Integr.*, 1995, pp. 811–816.

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

[7] D. V. Rao and M. Venkatesan, "An efficient reconfigurable architecture and implementation of edge detection algorithm using handle-C," *in Proc. IEEE Conf. ITCC*, vol. 2. Apr. 2004, pp. 843–847.

[8] H. Neoh and A. Hazanchuck, "Adaptive edge detection for real-time video processing using FPGAs," *Altera Corp., San Jose, CA, USA, Application Note*, 2005.

[9] C. Gentsos, C. Sotiropoulou, S. Nikolaidis, and N. Vassiliadis, "Real time canny edge detection parallel implementation for FPGAs," *in Proc. IEEE ICECS*, Dec. 2010, pp. 499–502.

[10] W. He and K. Yuan, "An improved canny edge detector and its realization on FPGA," in *Proc. IEEE 7th WCICA, Jun.* 2008, pp. 6561–6564.

[11] I. K. Park, N. Singhal, M. H. Lee, S. Cho, and C. W. Kim, "Design and performance evaluation of image processing algorithms on GPUs," *IEEE Trans. Parallel Distrib.Syst.*, vol. 22, no. 1, pp. 91–104ny edge detection on NVIDIA CUDA," in Proc. IEEE CVPRW, Jun. 2008, pp. 1–8.

[12] R. Palomar, J. M. Palomares, J. M. Castillo, J. Olivares, and J. Gómez-Luna, "Parallelizing and optimizing lip-canny using NVIDIA CUDA," in *Proc. IEA/AIE,Berlin, Germany*, 2010, pp. 389–398

[13] L. H. A. Lourenco, "Efficient implementation of canny edge detection filter for ITK using CUDA," in *Proc. 13th Symp. Comput. Syst.*, 2012, pp. 33–40.

[14] Q. Xu, C. Chakrabarti, and L. J. Karam, "A distributed Canny edge detector and its implementation on FPGA," *in Proc. DSP/SPE*), Jan. 2011, pp. 500–505.