ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Canny Edge Detection using Verilog

D Narayana Reddy[1], Mohan A R[2], Subhramanya Bhat[3]
PG Student [Electronics], Dept. of ECE, Canara Engineering College, Mangalore, Karnataka, India [1]
Assistant professor, Dept. of ECE, Canara Engineering College, Mangalore, Karnataka, India [2]
Associate professor, Dept. of ECE, Canara Engineering College, Mangalore, Karnataka, India [3]
narayareddy139@gmail.com

Abstract

Edge detection is one of the key stages in image processing and objects identification. The Canny Edge Detector is one of the most widely used edge detection algorithm due to its good performance. Edge detection carries preprocessing step for many image processing algorithms such as image enhancement, image segmentation, tracking and image/video coding. Canny's edge detection algorithm that results in significantly reduced memory requirements decreased latency and increased throughput with no loss in edge detection performance as compared to the sobel algorithm. The proposed canny edge detection in verilog algorithm gives good localization. Here we are using matlab to convert image into text/ pixel value. Then we will apply verilog canny algorithm to text/pixel value, then we will get edged text/pixel value, that text will given to matlab and finally get edged output image.

Keywords: Canny edge detection, non maximal suppretion, thresholding.

Introduction

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. This was also stated in Sobel and Laplace edge detection, but I just wanted reemphasize the point of why you would want to detect edges. The Canny's edge detection algorithm is known to many as the optimal edge detector.

Here followed a list of criteria to improve current methods 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 responses 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 2 were not substantial enough to completely eliminate the possibility of multiple responses to an edge. And finally the proposed methodology gives less time consuming compared to other edge detection.

Canny Edge Detection

The popular canny edge detector uses the following steps to find contours presents in the image. The first stage is achieved using Gaussian smoothing. The resulting image is sent to the PC that sends it back to the gradient filter, but here we modified our gradient filter a bit because this time we don't only need the gradient magnitude that is given by our previous operator, but we need separately Gx and Gy. We also need the phase or orientation of our gradient which is obtained using the following formula: $\theta = \arctan(Gy/Gx)$ as we can see, this equation contains an arctan and a division. These operators are very difficult to implement using hardware. We also don't need a high precision. Arctan and the division can be eliminated by simply comparing Gx and Gy values. If they are of similar length, we will obtain a diagonal direct ion, if one is at least 2.5 times longer than the other, we will obtain a horizontal or vertical direction.

After the edge directions are known, non-maximum suppression is applied. Nonmaximum suppression is used to trace pixels along the gradient in the edge direction and compare the value s

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

perpendicular to the gradient. Two perpendicular pixel values are compared with the value in the edge direction. If their value is lower than the pixel on the edge, then they are suppressed i.e. their pixel value is changed to 0, else the higher pixel value is set as the edge and the other two are suppressed with a pixel value of 0.

Finally, hysteresis is used as a means of eliminating streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1.

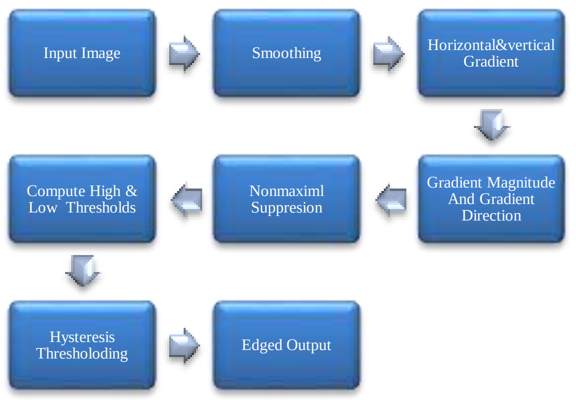
Input Image
Smoothing
Horizontal&vertical Gradient
Gradient Magnitude And Gradient Direction
Nonmaximl Suppression
Compute High & Low Thresholds
Hysteresis Thresholoding
Edged Output

Fig1: Block diagram of canny edge detection

Step1

In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased.

Image smoothing is the first stage of the canny edge detection. The pixel values of the input image are convolved with predefined operators to create an intermediate image. This process is used to reduce the noise within an image or to produce a less pixilated image. Image smoothing is performed by convolving the input image with a Gaussian filter. A Gaussian filter is a discrete version of the 2-dimensional function shown in equation (1).

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right) \qquad (1)$$

In (1), σ is the standard deviation of the Gaussian filter, which describes the narrowness of the peaked function, and x and y are spatial coordinates.

Step2

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

In this stage, the blurred image obtained from the image smoothing stage is convolved with a 3x3 Sobel operator. The Sobel operator is a discrete differential operator that generates a gradient image. Sobel operators used to calculate the horizontal and vertical gradients. The sobel operators are shown in Fig 2.

$$gx = \begin{array}{|c|c|c|}\hline -1 & 0 & +1 \\ \hline -2 & 0 & +2 \\ \hline -1 & 0 & +1 \\ \hline \end{array}$$

$$gy = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline +1 & +2 & +1 \\ \hline \end{array}$$

Fig 2: Sobel Operators.

Calculating Edge Strength

The Sobel operators from equation (2) are

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

used to obtain a gradient image. To obtain the gradient image, a smoothened image from the first stage is convolved with the horizontal and vertical Sobel operators as shown in equations (2.a) and (2.b), respectively.

$$G_x = (I * gx) \tag{2.a}$$

$$G_y = (I * gy) \tag{2.b}$$

In (2), I is the image obtained after Image smoothing; Gx and Gy are images with pixel values that are the magnitude of the horizontal and vertical gradient, respectively. Images Gx and Gy from equations (2.a) and (2.b) are used in equation (2.1)to obtain the "edge strength" of a pixel in an image. The edge strength G is

$$|G| = \sqrt{{G_x}^2 + {G_y}^2} \tag{2.1}$$

The magnitude, or edge strength, of the gradient is then approximated using the formula 2.2:

$$|G| = |G_x| + |G_y| \tag{2.2}$$

Step3:

The direction of the edge is computed using the gradient in the x and y directions. However, an error will be generated when sum is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If GY has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:

"Edge direction is defined as the direction of the tangent to the contour that the edge defines in 2-dimensions". The edge direction of each pixel in an edge direction image is determined using the arctangent

$$\theta = \arctan\left({G_y}/{G_x}\right) \tag{3}$$

Step4:

The edge strength for each pixel in an image obtained from equation (2.1) is used in non-maximum suppression stage. The edge directions obtained from equation (3) are rounded off to one of four angles 0 degree, 45 degree, 90 degree or 135 degree before using it in non-maximum suppression, as shown in Figure 3.

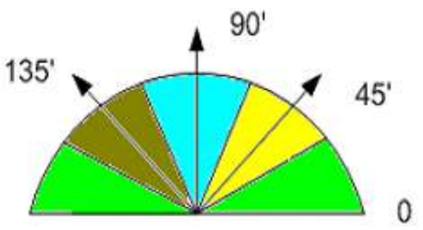

Fig 3: Edge directions are rounded off to one of the four angles.

Step5:

After the edge directions are known, non maximum suppression now has to be applied. Non maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Non-maximum suppression (NMS) is used normally in edge detection algorithms. It is a process in which all pixels whose edge strength is not maximal are marked as zero within a certain local neighborhood. This local neighborhood can be a linear window at different directions of length 5 pixels. The linear window considered is in accordance with the edge direction of the pixel under consideration for a block in an image as shown in Figure 4.

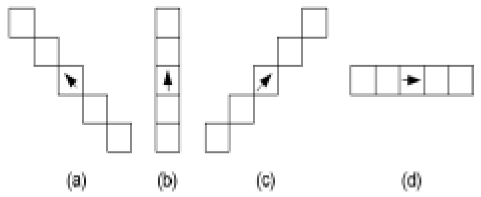

Fig 4: Linear window at the angle of (a) 135° (b) 90° (c) 45° (d) 0°.

Step 6

Thresholding with hysteresis is the last stage in canny edge detection, which is used to eliminate spurious points and non-edge pixels from the results of non-maximum suppression. The input image for thresholding with hysteresis has gone through Image smoothing, calculating edge strength and edge pixel, and the Non-maximum suppression stage to obtain thin edges in the image. Results of this stage should give us only the valid edges in the given image, which is performed by using the two threshold values, T1 (high) and T2 (low), for the edge strength of the pixel of the image. Edge strength which is greater than T1 is considered as a definite edge. Edge strength which is less than T2 is set to zero.

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

The pixel with edge strength between the thresholds T1 and T2 is considered only if there is a path from this pixel to a pixel with edge strength above T1. The path must be entirely through pixels with edge strength of at least T2. This process reduces the probability of streaking. As the edge strength is dependent on the intensity of the pixel of the image, thresholds T1 and T2 also depend on the intensity of the pixel of the image. Hence, the thresholds T1 and T2 are calculated by the canny edge detectors using adaptive algorithms. Thus all the edges in an image are detected using the canny edge detection.

Proposed Methodology:

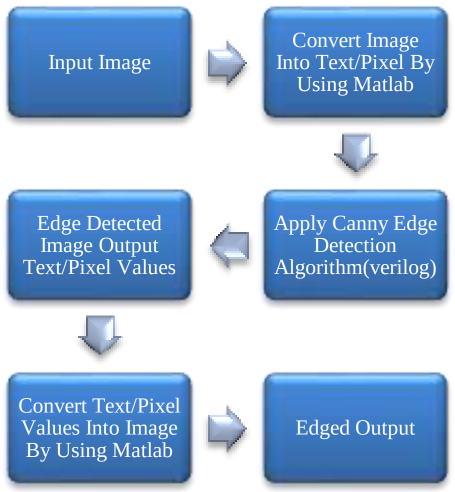
Input Image
Convert Image Into Text/Pixel By Using Matlab
Apply Canny Edge Detection Algorithm(verilog)
Edge Detected Image Output Text/Pixel Values
Convert Text/Pixel Values Into Image By Using Matlab
Edged Output

Fig 5: Proposed canny edge detection algorithm.

Algorithm of canny edge detector algorithm in Verilog is shown in Fig 5.
The canny edge detection consists of following steps:

- Converting input image into pixels by Matlab.
- Performing Canny Edge detector in verilog on the pixel.
- Get the edge detected image output in pixel format.
- Using Matlab convert pixels value in the image.
- Observe the edge detected image output.

Results

The canny edge algorithms is coded in Verilog. And the simulated wave form using model sim simulator is shown in Fig 6. The matlab is inter connected with the Verilog coding, hence the image is converted to edged image as shown in Fig 7.

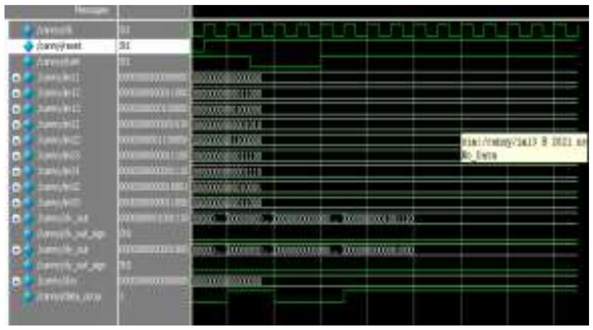

Fig 6: Simulation result using ModelSim Simulator.

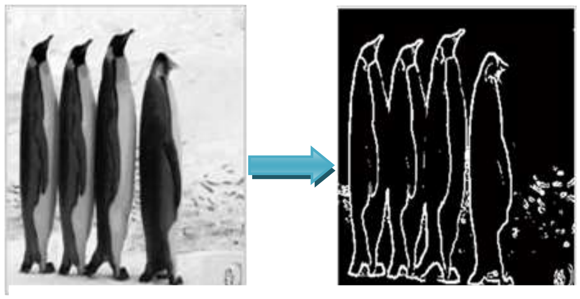

Fig 7: Results of Canny Edge Algorithm in Verilog.

This section should be typed in character size 10pt Times New Roman, Justified

Conclusion

Thus our proposed project will detect the edges efficiently with reduction in the processing speed and reduced the memory requirement. Proposed work will reduce the latency and increase the throughput.

Acknowledgements

The authors wish to thank the ECE Department of Canara Engineering College, Mangalore, Karnataka, India, for supporting this work.

References

1. L. Torres, M. Robert, E. Bourennane, and M. Paindavoine, “Implementation of a recursive real time edge detector using Retiming techniques,” VLSI, pp. 811 –816, Aug. 1995.
2. Qian Xu, Chaitali Chakrabarti and Lina J. Karam, “A Distributed Canny Edge Detector and Its Implementation On FPGA”, Tempe, AZ.
3. D. V. Rao and M. Venkatesan, “An efficient reconfigurable Architecture and implementation of

edge detection algorithm Using Handle-C," ITCC, vol. 2, pp. 843 – 847, Apr. 2004.
4. Shengxiao Niu, Jingjing Yang, Sheng Wang, Gengsheng Chen,"Improvement and Parallel Implementation of Canny Edge Detection Algorithm Based on GPU".
5. W. He and K. Yuan, "An improved Canny edge detector and its Realization on FPGA," WCICA, pp. 6561 –6564, Jun. 2008.
6. T. Rupalatha**,** Mr.C.Leelamohan, Mrs.M.Sreelakshmi, "implementation of distributed Canny edge detector on fpga". International Journal of Innovative Research in Science, Engineering and Technology, Vol. 2, Issue7, July 2013.
7. Chandrashekar N.S, Dr. K.R. Nataraj, "A Distributed Canny Edge Detector and Its Implementation on FPGA" International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue.7. Issn 2250-3005(online) November| 2012.
8. Tejaswini h.r, vidhya n, swathi r varma, santhosh b, "an implementation of real time optimal edge detector and vlsi architecture". International conference on electronics and communication engineering, 28th april-2013, bengaluru, isbn: 978-93-83060-04-7.

BABILOGRAPHY

D Narayana Reddy received his B.E degree in Instrumentation Engineering from RYM Engineering College, Bellary in 2012. Currently he is perusing M.Tech degree in Electronics at Canara Engineering College, Mangalore. His areas of interest are VLSI and Image Processing.

Mr. Mohan A.R received his M.Tech degree in VLSI Design and Embedded Systems from SJB Institute of technology, Bangalore in 2011 and B.E. degree from Coorg Institute of Technology, Ponnampet in 2007. Currently he is working as an Asst. Professor in the Department of E&C, Canara Engineering College, Mangalore. His area of interest is Digital VLSI Design.

Mr. Subramanyam Bhatt received his B.E degree from MIT, Mnglore, India. He obtained his Masters degree from Shree Jayaramachandra College of Engg. Mysore, India. Currently he is persuing Ph.d from Visweswaraya Technological University, Belgaum, India and Currently he is working as an Asst. Professor in the Department of E&C, Canara Engineering College, Mangalore. His area of interest include Signal and Image processing, power electronics.