

FPGA Implementation of Modified Canny Edge Detection Algorithm

Binika Nambam

M. Tech, VLSI Design & Embedded Systems
T. John Institute of Technology, Bangalore, India

Mr. Narayana Swamy R.

Associate Professor,
Dept. of ECE, T. John Institute of Technology,
Bangalore, India

Abstract—Edge detection refers to the process of identifying as well as locating sharp discontinuities in an image. While preserving the essential structural properties in an image, edge detection process significantly reduces the amount of data and filters out useless information. Due to its superior performance, the Canny edge detector is one of the most widely used edge detection algorithms. Compared with other edge detection algorithms, it is computationally more intensive and it also has a higher latency as it is based on frame-level statistics. In this paper a mechanism is proposed to implement the Canny algorithm at the block level without any loss in edge detection performance compared with the original frame-level Canny algorithm. Applying the original Canny algorithm directly at the block-level leads to excessive edges in smooth regions as well as to loss of significant edges in high-detailed regions since the original Canny computes the high and low thresholds based on the frame-level statistics. A modified Canny edge detection algorithm is presented that adaptively computes the edge detection thresholds based on the block type. Additionally, the resulting block-based algorithm will give a significantly reduced area and increased frequency. Additionally, the edge detection performance of the proposed algorithm will be better than the existing algorithm. The proposed algorithm will give better result than the existing algorithm as it can preserve more useful edge information.

Keywords— *Image processing, Edge detection, Canny edge detector.*

I. INTRODUCTION

One of the most important information in an image is Image edge information, which can describe the target outline, the relative position within the target area, and other important information. Edge detection is one of the most important process in many image processing algorithms such as image enhancement, image segmentation, tracking and image/video coding. Canny edge detector has best performance and has remained a standard for many years. The superior performance of the Canny algorithm is due to the fact that it performs hysteresis thresholding which requires computing high and low thresholds based on the entire image statistics. But this feature makes the Canny edge detection algorithm more computationally complex as compared to other edge detection algorithms namely the Roberts and Sobel algorithms and it also necessitates additional pre-processing computations to be done on the entire image. Therefore, a direct implementation of the Canny algorithm has high latency and cannot be employed in real-time applications.

Different implementations of the Canny algorithm have been proposed on a wide list of hardware platforms. Using Canny's criteria, there is a set of work [3]–[5] on Deriche filters that have been derived and implemented on ASIC-based platforms. The Canny-Deriche filter [3] is a network with four transputers that detect edges in a 256×256 image in 6s, far from the requirement for real time applications. The design in [4] improved the Canny-Deriche filter implementation of [1] and was able to process 25 frames/s at 33 MHz, but the used off-chip SRA Mmemories consist of Last-In First-Out (LIFO) stacks, which increased the area overhead compared to [3]. A new organization of the Canny-Deriche filter in [5] was proposed by Demigny, which reduces the memory size and the computation cost by a factor of two. The number of clock cycles per pixel of the implementation [5] varies with the size of the processed image, resulting in variable clock cycles/pixel from one image size to another and as the image size increases, the processing time increases. There is another set of work [6]–[8] on mapping the Canny edge detection algorithm onto FPGA-based platforms. The two FPGA implementations in [6] and [7] translate the software design directly into VHDL or Verilog using system-level hardware design tools, which results in a decreased timing performance. The parallel implementation in [8] operates on only 4 pixels in parallel, resulting in an increase in the number of memory accesses and in the processing time as compared to the proposed algorithm. Furthermore, in order to reduce the computational complexity, all of these implementations compute the high and low thresholds off-line and use the same fixed threshold values for all the images, which result in a decreased edge detection performance. A self-adapt threshold Canny algorithm [9] is proposed for a mobile robot system where a low and high threshold values are computed for each input image and an implementation for an Altera Cyclone FPGA is presented. Recently, the General Purpose Graphic Processing Unit (GPGPU) has emerged as a powerful and accessible parallel computing platform for image processing applications [10], [11]. All of these implementations are frame-based and do not have good edge detection performance since they use the same fixed pair of high and low threshold values for all images.

In the original Canny method, the computation of the high and low threshold values depends on the statistics of the whole

input image. Most of the above existing implementations (e.g., [6]–[8]) use the same fixed pair of high and low threshold values for all input images resulting in a decreased edge detection performance. The non-parallel implementation [9] computes the low and high threshold values for each input image. This results in increased latency as compared to the existing implementations (e.g., [6]–[8]). Furthermore, the non-parallel implementations ([6]–[9]) result in a decreased throughput as compared to the parallel implementations ([8]). The issue of increased latency and decreased throughput is becoming more significant with the increasing demand for large-size high-spatial resolution visual content.

Focus of this paper is on reducing the latency, increasing the frequency and reducing the area and thus making it usable in real-time processing applications. In this paper, we propose an adaptive threshold selection algorithm. The proposed algorithm yields better edge detection results for both clean and noisy images.

The rest of the paper is organized as follows. Section 2 gives a brief overview of the original Canny algorithm. Section 3 presents the proposed distributed Canny edge detection algorithm. Finally, conclusions are presented in Section 4.

II. CANNY EDGE DETECTION ALGORITHM

John F. Canny developed an approach to derive an optimal edge detector to deal with step edges corrupted by a white Gaussian noise. The original Canny algorithm [14] consists of the following steps: 1) Calculating the horizontal gradient G_x and vertical gradient G_y at each pixel location by convolving with gradient masks. 2) Computing the gradient magnitude G and direction θ_G at each pixel location. 3) Applying Non-Maximal Suppression (NMS) to thin edges. This step involves computing the gradient direction at each pixel. 4) Computing high and low thresholds based on the histogram of the gradient magnitude for the entire image.

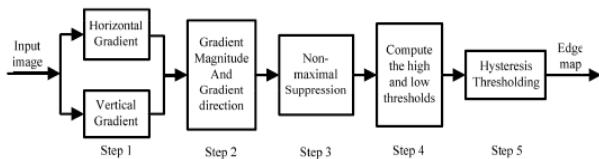


Fig. 1. Block diagram of the Canny edge detection algorithm

A block diagram of the Canny edge detection algorithm [14] is shown in Fig. 1. In this original Canny edge detection algorithm [14], the gradient calculation (Step 1) is performed by using Finite-impulse Response (FIR) gradient masks designed to approximate the following 2D sampled versions of the partial derivatives of a Gaussian function:

$$F_x(x, y) = -\frac{x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \left(-x e^{-\frac{x^2}{\sigma^2}}\right) \left(\frac{1}{\sigma^2} e^{-\frac{y^2}{\sigma^2}}\right) \quad (1)$$

$$F_y(x, y) = -\frac{y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \left(-y e^{-\frac{y^2}{\sigma^2}}\right) \left(\frac{1}{\sigma^2} e^{-\frac{x^2}{\sigma^2}}\right) \quad (2)$$

where σ is the standard deviation of the Gaussian function. The size of the gradient masks used by the Canny edge detector is usually implemented as a function of the chosen σ , with larger values of σ yielding larger masks.

The magnitude and direction of gradient are given by:

$$G = \sqrt{G_x^2 + G_y^2} \quad (3)$$

$$\theta = \tan^{-1} \frac{G_y}{G_x} \quad (4)$$

III. PROPOSED MODIFIED CANNY EDGE DETECTION ALGORITHM

Although the traditional Canny edge detection algorithm is more widely used in practical engineering, there are still aspects to improve. The conventional canny operator suffers from the problem of detecting false edges and smoothes edges with noise. Double thresholding is used which require to set high and low threshold values manually. To get appropriate threshold values for an image several experiments are required.

Canny edge algorithm is considered as optimal edge detection technique but it responds to false image in noisy environment and is not adaptive in nature. Canny algorithm leads to excessive edges in smooth regions and to loss of significant edges in high-detailed regions since it computes the high and low thresholds based on the frame-level statistics. Thresholding is important operation in edge detection. Canny operator use double thresholding concept which uses two threshold values, high and low. But it is required to set manually and lots of experiments are required to find appropriate values for an image. The received image may still contain false edge points.

The existing Canny edge detection algorithm gives unwanted edge image information. In the proposed Canny edge detection algorithm, an adaptive threshold selection algorithm will be used. With the proposed modified Canny edge detection algorithm the required edge image information will be obtained. It will give better edge detection result of image with reduced area and increased frequency. The algorithm keeps the advantages of the traditional Canny algorithm. It enhances the ability of noise suppression and keeps more edge information.

TABLE I.

| MODEL & AUTHORS | FREQUENCY |
|--|------------------|
| EXISTING MODEL (QIAN XU, SRENVAS VARADARAJAN, CHAITALI CHAKRABART, LINA J. KARAM.) | 133MHZ |
| PROPOSED MODEL | MORE THAN 133MHZ |

Table I. shows the comparison between the existing model and proposed modified Canny method in terms of frequency. The proposed model give better image edge information with increased frequency and reduced area.

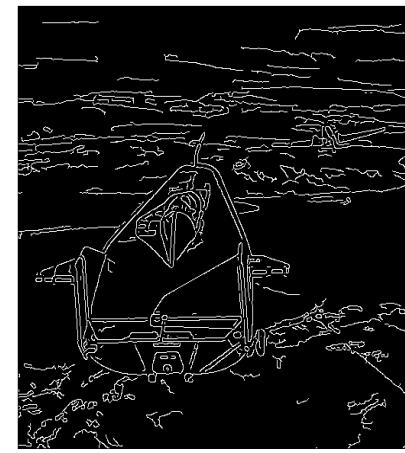


(a)



(a)

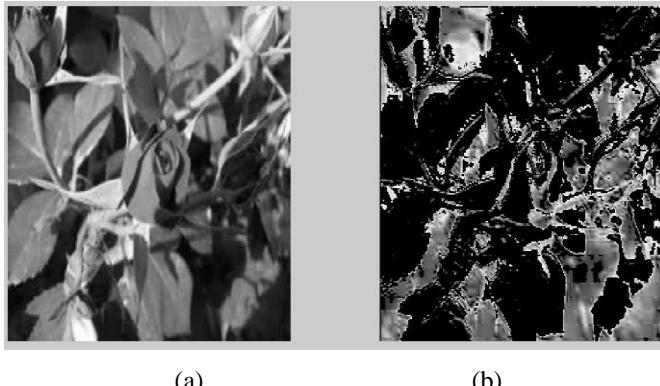
(b)



(b)

Fig. 2 (a) Original Image (b) Existing Canny edge image

Fig. 2 (a) Original Image (b) Existing Canny edge image



(a)

(b)

Fig. 3(a) Input Image (b) Existing Canny edge image

The above figures 2, 3 and 4 shows the input images with the images obtained with existing system. The images obtained with the existing system give not just the required but the unwanted edge image information. With the proposed modified Canny edge detection algorithm, better edge detection result of image will be obtained.

IV. CONCLUSION

The modified Canny edge detection algorithm will enhance the ability of noise suppression and keeps better edge information. It will give better edge detection result of image with reduced area and increased frequency.

ACKNOWLEDGMENT

We take this opportunity to express our deepest gratitude and appreciation to all those who have helped us directly or indirectly towards completing this paper.

REFERENCES

- [1] Different Techniques Of Edge Detection In Digital Image Processing Pooja Sharma, Gurpreet Singh, Amandeep Kaur / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 3, May-Jun 2013, pp.458-461
- [2] VHDL Based Canny Edge Detection Algorithm International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 - 5161 ©2014 INPRESSCO®
- [3] Implementation Of Distributed Canny Edge Detector On FPGA International Journal of Innovative Research in Science, Engineering and Technology Vol. 2, Issue7, July 2013.
- [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, Realtime 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. 91–104, Jan. 011.
- [12] J. D. Owens *et al.*, "A survey of general-purpose computation on graphics hardware," *Comput. Graph. Forum*, vol. 26, no. 1, pp. 80–113, 2007.