

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 5, May 2018

## Edge Detection using Distributed Canny Edge Detector

Bhagya Baburaj<sup>1</sup>, Jibby Peter Dacruz<sup>2</sup>

M Tech Student, Department of ECE, Viswajyothi College of Engineering & Technology, Vazhakulam, Kerala, India<sup>1</sup>

Asst. Professor, Department of ECE, Viswajyothi College of Engineering & Technology, Vazhakulam, Kerala, India<sup>2</sup>

**ABSTRACT:** Edges of an image represents significant local changes of intensity. Edges mostly occur on the boundary between two different regions in an image. Edge detection is the process of segmenting images based on abrupt changes in intensity. Important features can be extracted from the edges of an image. Canny edge detector is a type of edge detector which is used to extract useful structural information from images and dramatically reduces the amount of data to be processed. When compared with other edge detection algorithms, Canny edge detector is computationally more intensive and also has higher latency. To solve this problem, we present Distributed Canny Edge Detection Algorithm. The entire image is classified into different blocks and adaptively computes the edge detection thresholds based on block type. The resulting distributed algorithm has significantly reduced latency and also has better edge detection performance than original Canny edge detector. All this implementation is done using Matlab.

**KEYWORDS:** Canny Edge Detector, Distributed Image Processing, Edge Detection.

### I. INTRODUCTION

Edge detection process simplifies the analysis of images by reducing the amount of data to be processed and also without causing any changes in the useful structural information about the object boundaries. It is the most common pre-processing step in many image processing algorithms such as image enhancement, image segmentation and tracking.

Among the different edge detection algorithms that are existing now, the Canny edge detector has remained a standard for many years and has best performance. The superior performance of Canny edge detection algorithm is due to the fact that it performs hysteresis thresholding. High and low thresholds are to be calculated entire image statistics, to perform hysteresis thresholding. This step makes Canny Edge Detector more computationally complex when compared with other edge detection algorithms. As a result of increased complexity, the direct implementation of Canny algorithm leads to high latency and hence it is not commonly employed in real-time applications.

Several implementations of Canny Edge Detection algorithm have been proposed on a wide list of platforms in order to reduce the computational complexity. But all these implementations resulted in decreased edge detection performance because all these implementations compute the high and low thresholds offline and use the same threshold values for all the images.

The main focus of the proposed algorithm is to reduce latency and thereby improve performance of Canny algorithm. Initially, the input image is partitioned into different blocks. Depending upon the type of block, an adaptive threshold selection algorithm computes the high and low thresholds based on the type of block and local distribution of pixel gradients in the block. Each block can be processed in parallel which will reduce the latency significantly. Apart from reduction in latency, the proposed algorithm also gives better edge detection results for both noisy as well as clean images.

The rest of the paper is organized as follows. Section II gives a brief overview of the related work. Section III gives an overview of the original Canny Edge Detection Algorithm. The proposed Distributed Canny Edge Detection algorithm, computes edge detection thresholds adaptively based on block type is described in Section IV. Section V presents experimental results. Finally, Section VI presents conclusion.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 5, May 2018

## II. RELATED WORK

Several implementations of the Canny algorithm have been done on a wide list of platforms. Deriche filters have been derived using Canny’s criteria and implemented on ASIC-based platforms in the set of works [2], [4] and [5]. The edge detector [2] is a network which detect edges in a 256×256 image. It is an efficient recursive algorithm for edge detection. But this approach detects edges in a 256×256 image in 6s, which is far from the requirement for real-time applications.

The design in [5] is an improved version of implementation of [2], but it was able to process 25 frames/s at 33MHz , which increased the area overhead when compared to [1]. A new organisation of Canny-Deriche filter in [4] is proposed by Demigny, which reduced the memory size and computational cost by a factor of two. The set of work [7]-[9] mapped the Canny edge detection algorithm onto FPGA based platforms. The implementations in [8] and [9] translate the software design directly into VHDL or Verilog using system level hardware design tools. This resulted in decreased timing performance. But all these implementations resulted in decreased edge detection performance because all these implementations compute the high and low thresholds offline and use the same threshold values for all the images.

A self adapt threshold Canny algorithm [7] is proposed for a mobile robot system where a low and high threshold values are computed for each input image but this will result in increased latency. The problem of increased latency and decreased throughput is becoming more significant with the increasing demand for large-size high-spatial resolution visual content. So, the main focus of proposed edge detector is to reduce latency and thereby increase the edge detection performance.

## III. ORIGINAL CANNY EDGE DETECTION ALGORITHM

John Canny has developed an approach to derive an optimal edge detector to deal with step edges corrupted by a white Gaussian noise.

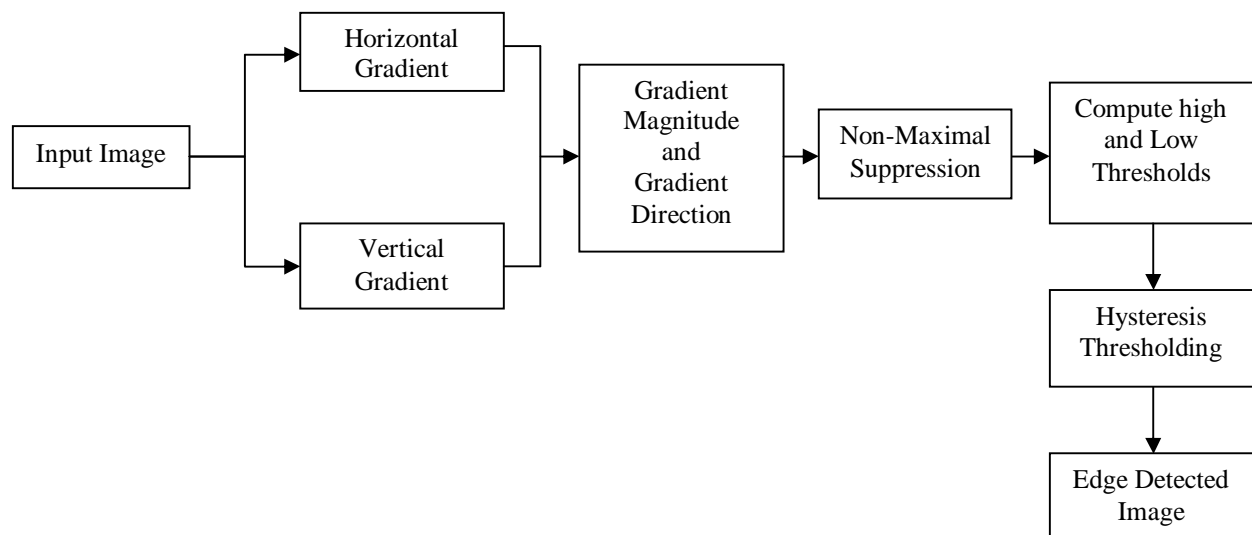


Fig.1. Block diagram of original Canny edge detection algorithm

Block diagram of Original canny edge detection algorithm [6] is shown in Fig.1. This algorithm consists of the following steps: calculating horizontal and vertical gradients, computing gradient magnitude and direction, applying Non-Maximal Suppression to thin edges, computing high and low thresholds and performing hysteresis thresholding. The three main criterias of the Canny edge detection algorithm [3] are as follows:

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 5, May 2018

1. **Low error rate of detection:** It is important to find all the edges present in the image and there should be no response for non-edge.
2. **Localization of edges:** The distance between actual edges present in the image and the edges found by the edge detection algorithm should be minimized
3. **Single Response:** To have a single response to single edge. That means, the algorithm should not return multiple edge pixels when only a single edge exists.

The original Canny edge detection algorithm [6] consists of the following steps:

1. The first step is to smoothen the input image using Gaussian mask. The high frequency components can be eliminated from the image by the process of smoothening.
2. Horizontal gradient  $G_x$  and vertical gradient  $G_y$ , at each pixel location is calculated by convolving the image with gradient masks.
3. Gradient magnitude  $G$  and gradient direction  $\theta_G$  at each pixel location is computed.
4. Non-Maximal Suppression (NMS) is applied to thin edges. If the pixel's gradient direction is one of 8 possible main directions, the gradient magnitude of this pixel is compared with two of its immediate neighbours along the gradient direction and magnitude is set to zero if it does not correspond to a local maximum. An interpolation is done to compute the neighbouring gradients, for the gradient directions that do not coincide with one of the 8 possible main directions.
5. High and low thresholds are calculated based of gradient magnitude for the image.
6. Hysteresis thresholding is performed to determine the edge detected image.

A pixel is considered as strong edge if the gradient magnitude of the pixel is greater than the high threshold. The pixel is labelled as weak edge if the gradient magnitude of the pixel is between high and low threshold.

## IV. DISTRIBUTED CANNY EDGE DETECTION ALGORITHM

Original Canny edge detector computes high and low thresholds based on frame level statistics. In the case of distributed Canny edge detector, the original Canny algorithm is implemented at block level without any degradation in edge detection performance. Thus, directly applying the original Canny algorithm to a block of the image would fail to detect the desired edges since the statistics of a block may differ significantly from the statistics of the entire natural image.

For example, if we consider two types of blocks, namely smooth block and edge block, extracted from the Lena image and House image, despite the fact that the smooth block does not contain any significant edges, applying the original Canny algorithm to this block will result in the detection of an excessive number of edges. This is due to the fact that the Canny algorithm assumes that a percentage  $PI$  of the total pixels in the block are strong edges.

On the other hand, for an edge block, the majority of pixels are edges. In order not to cause degradation in edge detection performance of the original frame-based Canny algorithm when it is applied to a block, the high and low thresholds when computed at the block level must match with the thresholds that would have been obtained for the whole image.

To compute the high and low hysteresis thresholds, a finely quantized gradient magnitude histogram is usually needed. Here, we employ the non-uniform quantizer, which has been proposed by us in [10], to obtain the gradient magnitude histogram for each block such that the high threshold can be precisely calculated.

As in [7] and [10], for the edge/texture, medium edge and strong edge block, it was observed that the largest peak in the gradient magnitude histograms of the Gaussian-smoothed images occurs near the origin and corresponds to low frequency content, while edge pixels form a series of smaller peaks, where each peak corresponds to a class of edges having similar gradient magnitudes. Consequently, the high threshold should be selected between the largest peak and the second largest edge peak.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 5, May 2018

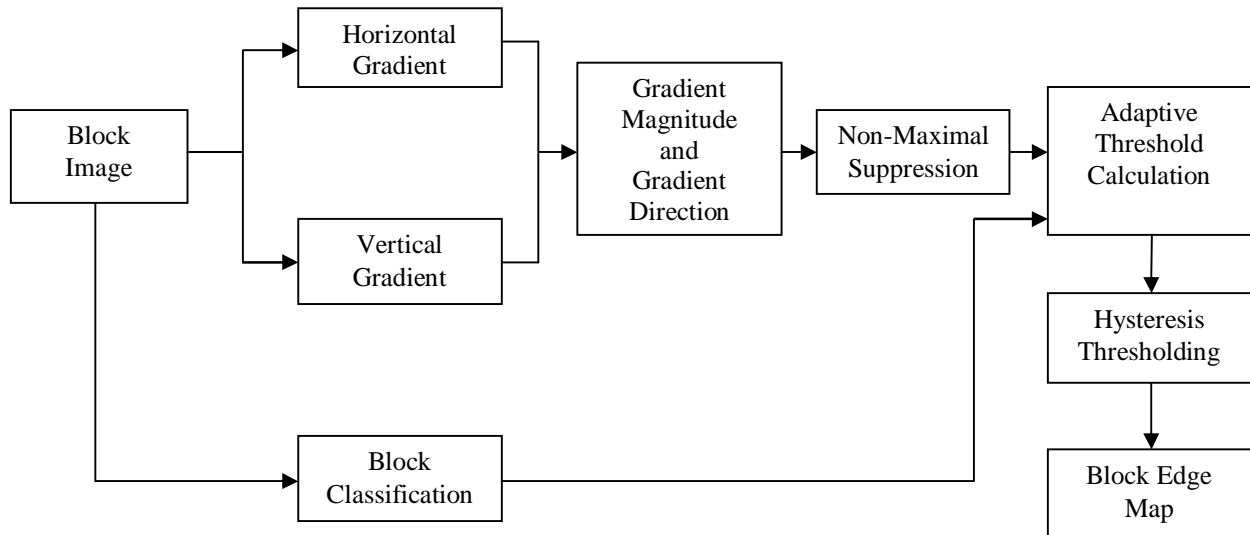


Fig.2. Block Diagram of Distributed Canny Edge Detector

The block diagram of distributed Canny edge detector is shown in Fig.2. In this algorithm, the input image is divided into different blocks and the blocks are processed independent of each other. Thus the edge detection performance can be improved compared to the original Canny edge detection algorithm.

## V. EXPERIMENTAL RESULTS

The edge detection result performance of distributed Canny edge detection approach is analyzed by comparing the significance of its resulting edge map with that of the one produced by the original Canny edge detection algorithm.

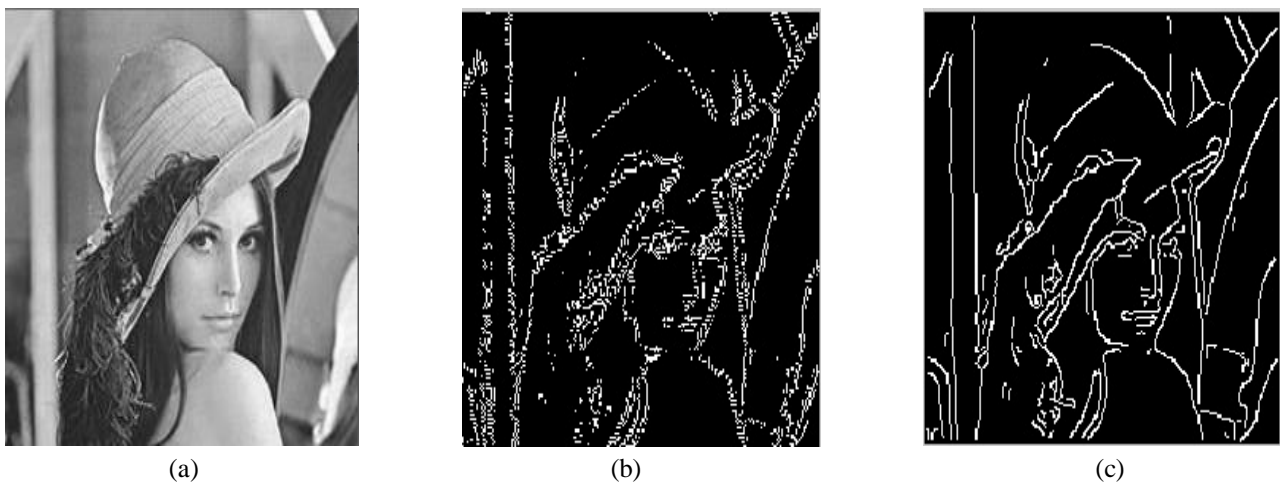


Fig.3. (a) Original "Lena" image (b) Edge maps using Original Canny Edge detector (c) Edge map using Distributed Canny Edge Detector

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 5, May 2018

Fig.3. (a) shows the original Lena image. The edge map obtained when original Canny edge detection algorithm is applied to Lena image is shown in Fig.3. (b). The edge map obtained when distributed Canny edge detection algorithm is applied to Lena image is shown in Fig.3. (c). From these results we can see that distributed Canny edge detector gives better results than original Canny edge detector.

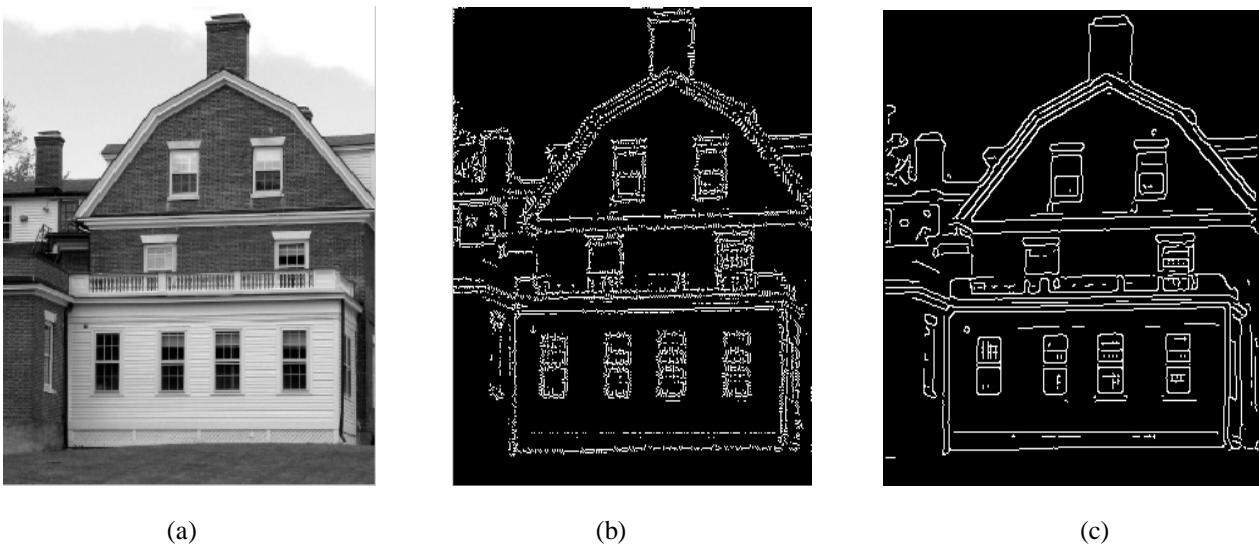


Fig. 4. (a) Original “House” image (b) Edge maps using Original Canny Edge detector (c) Edge map using Distributed Canny Edge Detector

Fig.4. (a) shows the original House image. The edge map obtained when original Canny edge detection algorithm is applied to House image is shown in Fig.4. (b). The edge map obtained when distributed Canny edge detection algorithm is applied to House image is shown in Fig.4. (c).

The performance of the proposed algorithm is affected by two main parameters, which includes the mask size and the block size. Size of the gradient mask is a function of the standard deviation of Gaussian filter and the best choice of  $\sigma$  is based on the image characteristics. For the original Canny edge detection algorithm as well as the proposed algorithm, the standard deviation is a parameter that is typically set by the user based on the knowledge of sensor noise characteristics. It can also be set by a separate application that estimates the noise present in the image.

A smaller mask size which corresponds to a lower  $\sigma$ , is better for detecting detailed textures and fine edges. Figures shows the Matlab implementation software result for the Lena image and House image using the original Canny edge detector and proposed distributed Canny edge detector.

From the results, we can see that, when compared to the original frame-based Canny edge detector, the proposed distributed Canny algorithm can detect edges in low contrast regions since it use adaptive thresholding method and hence is able to adapt to the local statistical distribution of gradient values in a block.

## VI. CONCLUSION

The classical Canny edge detection algorithm uses frame-level statistics to determine the high and low thresholds and hence it has latency proportional to the frame size. Distributed Canny edge detection algorithm has the ability to detect edges of multiple blocks at the same time. This will speed up the edge detection process and thus meet real-time requirements. Adaptive threshold selection technique is used in this algorithm to determine thresholds based on the type of block. The resulting block based algorithm has reduced latency and can be easily integrated to block based

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Issue 5, May 2018

image codecs. Thus distributed Canny edge detection algorithm has better edge detection performance and also there is significant reduction in latency.

## REFERENCES

- [1] Qian Xu , Srenivas Varadarajan , Chaitali Chakrabarti and Lina J. Karam , “ A Distributed Canny Edge Detector : Algorithm and FPGA Implementation ” , IEEE Transactions on Image Processing , Vol.23, no.7, pp.2944-2960 ,2014.
- [2] R. Deriche, “Using canny criteria to derive a recursively implemented optimal edge detector,” International Journal of Computer Vision, Vol.1, no.2, pp. 167–187, 1987.
- [3] P. Bao, L. Zhang, and X. Wu , “Canny edge detection enhancement by scale multiplication ,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, no. 9, pp. 1485–1490 ,2005.
- [4] F. G. Lorca, L. Kessal, and D. Demigny, “Efficient ASIC and FPGA implementation of IIR filters for real time edge detection,” IEEE ICIP, vol. 2. pp. 406–409, 1997
- [5] L. Torres, M. Robert, E. Bourennane, and M. Paindavoine, “Implementation of a recursive real time edge detector using retiming technique,” Asia and South Pacific IFIP International Conference on Very Large Scale Integration , pp. 811–816,1995.
- [6] J. F. Canny, “A Computation Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 769–798, 1986.
- [7] W. He and K. Yuan, “An improved canny edge detector and its realization on FPGA,” in Proc. IEEE 7th WCICA, pp. 6561–6564, Jun. 2008.
- [8] D. V. Rao and M. Venkatesan, “An Efficient Reconfigurable Architecture and Implementation of Edge Detection Algorithm using Handle-C,” IEEE Conference on Information Technology: Coding and Computing (ITCC), vol. 2. pp. 843–847, 2004.
- [9] H. Neoh and A. Hazanchuck, “Adaptive edge detection for real-time video processing using FPGAs,” Altera Corporation., San Jose, CA, USA, Application Note, 2005.
- [10] C. Gentsos, C. Sotiropoulou , S. Nikolaidis , and N. Vassiliadis , “Real-Time Canny Edge Detection Parallel Implementation for FPGAs,” IEEE International Conference on Electronics ,Circuits and Systems (ICECS), pp. 499–502, 2010.
- [11] Q. Xu, C. Chakrabarti, and L. J. Karam, “A Distributed Canny Edge Detector and its implementation on FPGA,” Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), pp. 500–505, 2011.

## BIOGRAPHY



**Bhagya Baburaj** completed Bachelor of Technology (B.Tech) in Electronics and Communication Engineering, from Viswajyothi College of Engineering and Technology (2016). She is doing M.Tech in VLSI and Embedded System at Viswajyothi College of Engineering & Technology, Vazhakulam. Her fields of interest include VLSI, Digital Electronics and Embedded System.



**Jibby Peter Dacruz** received Bachelor of Technology (B.Tech) from Sree Narayana Mangalam Institute of Management and Technology in 2008. From year 2008-2010, he did Master of Engineering (ME) in Communication Systems, from Bannari Amman Institute of Technology. He is working as Assistant Professor in Viswajyothi College of Engineering and Technology for the last 8 years. His fields of interest include Analog communication, Electronic devices and circuits, Electronic instrumentation.