



# INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

## REVIEW PAPER ON DESIGN OF MODIFIED CANNY EDGE DETECTOR BASED IN VHDL FOR PORTABLE DEVICE

PALLAVI RAMGUNDEWAR<sup>1</sup>, DR. S.P. HINGWAY<sup>2</sup>, PROF. K. MANKAR<sup>2</sup>

1. Electronics and Telecommunication Engineering, GHRIETW, RTMN University, Nagpur, India.
2. Electronics and communication Engineering, GHRIETW, RTMN University, Nagpur, India.

Accepted Date: 05/03/2015; Published Date: 01/05/2015

**Abstract:** Edge detection is one of the key stages in image processing and object reorganization. The Canny Edge Detector is one of the most widely used edge detection algorithm due to its superior performance. In this paper, we propose a mechanism to implement the Canny algorithm at the block level without any loss in edge detection performance compared with the original frame-based Canny algorithm. Directly applying the original Canny Edge detection algorithm at the block-level leads to excessive edges in smooth regions and 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. To solve this problem, we present a modified Canny edge detection algorithm that adaptively computes the edge detection thresholds based on the block type and the local distribution of the gradients in the image block. Here we propose the design of modified Canny Edge detection algorithm that results in significantly reduced memory requirement, decrease in latency, increase throughput, with no loss in edge detection performance as compare to original Canny Detector Algorithm. Here we are using matlab to convert image into text/ pixel value.

**Keywords:** Distributed Processing, Canny Edge Detector, High Throughput, Parallel Processing, FPGA.

Corresponding Author: MS. PALLAVI RAMGUNDEWAR



PAPER-QR CODE

Access Online On:

[www.ijpret.com](http://www.ijpret.com)

How to Cite This Article:

Pallavi Ramgundewar, IJPRET, 2015; Volume 3 (9): 915-921

## INTRODUCTION

EDGE detection is the most common preprocessing step in many image processing algorithms such as image enhancement, image segmentation, tracking and video coding. In 1986 John Canny describes a computational approach to edge detection which is known as canny edge detector. Typically, edge detection algorithms are implemented using software. With advances in Very Large Scale Integration (VLSI) technology, their hardware implementation has become an attractive alternative, for real-time applications. Its superior performance is due to the fact that the Canny algorithm performs hysteresis thresholding which requires computing high and low thresholds based on the entire image. Unfortunately, this feature makes the Canny edge detection algorithm not only more computationally complex as compared to other edge detection algorithms, such as the Roberts and Sobel algorithms, but also necessitates additional pre-processing computations to be done on the entire image. As a result, a direct implementation of the Canny algorithm has high latency and cannot be employed in real-time applications. This places heavy requirements on memory and results in large latency hindering real-time implementation of the Canny edge detection algorithm.

In order to reduce memory requirements, decrease latency and increase throughput, a distributed Canny edge detection algorithm is proposed. The hysteresis threshold calculation is a key element that greatly affects the edge detection output. However, the original Canny algorithm computes the high and low thresholds for edge detection based on the entire image statistics, which prevents the processing of individual blocks independently. I will propose a new threshold selection algorithm based on the distribution of pixel gradients in a block of pixels to overcome the dependency between the blocks. However, the hysteresis thresholds calculation is based on a very finely and uniformly quantized 64-bin gradient magnitude histogram, which is computationally expensive and, thereby, hinders the real-time implementation. In this paper we are proposing compression of image and then applying distributed canny edge detector algorithm in VHDL.

### A. CANNY EDGE DETECTION ALGORITHM

The Canny edge detection algorithm is known to many as the optimal edge detector. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there are no responses to non-edges. The second criterion is that the edge points be well localized. A third criterion is to have only one response to a single edge. A block diagram of the Canny edge detection algorithm is shown in Fig. 1. The Canny Edge Detector is carried out in following four steps:

1. Smooth the image with a 2 dimensional Gaussian. In most cases the computation of a 2 D Gaussian is expensive, therefore it's approximated by 2 one dimensional Gaussians, one within the x direction and also the alternative within the y direction.
2. Take the gradient of the image. This shows changes in intensity, that indicates the presence of edges. This truly provides 2 results, the gradient within the x direction and also the gradient within the y direction.
3. Non-maximal suppression. Edges can occur at points the wherever the gradient is at a most. Therefore, all points not at a most should be suppressed. So as to try and do this, the magnitude and direction of the gradient is computed at every picture element. Then for every picture element check if the magnitude of the gradient is larger at one pixel's distance away in either the positive or the negative direction perpendicular to the gradient.
4. Edge Thresholding. The strategy of thresholding utilized by the cagey Edge Detector is referred to as "hysteresis". It makes use of each a high threshold and an occasional threshold. If a picture element includes a price on top of the high threshold, it's set as a position picture element. If a picture element includes a price on top of the low threshold and is that the neighbour of a position picture element, it's set as a position picture element further. If a picture element includes a price on top of the low threshold however isn't the neighbour of a position picture element, it's not set as a position picture element. If a picture element includes a price below the low threshold, it's ne'er set as a position picture element.

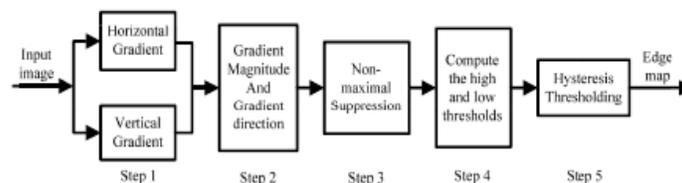


Fig. 1. Block diagram of the Canny edge detection algorithm.

## B. DISTRIBUTED CANNY EDGE DETECTION ALGORITHM

In order to improve the performance of the edge detection at the block level and achieve the same performance as the original frame-based Canny edge detector when this latter one is applied to the entire image, a distributed Canny edge detection algorithm is propose. A diagram of the propose algorithm is shown in Fig. 2. In the propose distributed version of the Canny algorithm, the input image is divided into  $m \times m$  overlapping blocks and the blocks are processed independent of each other. For an  $L \times L$  gradient mask, the  $m \times m$  overlapping

blocks are obtained by first dividing the input image into  $n \times n$  non-overlapping blocks and then extending each block by  $(L + 1)/2$  pixels along the left, right, top, and bottom boundaries, respectively. Steps 1 to 3 and Step 5 of the distributed Canny algorithm are the same as in the original Canny algorithm except that these are now applied at the block level. Step 4, which is the hysteresis high and low thresholds calculation, is modified to enable parallel block-level processing without degrading the edge detection performance.

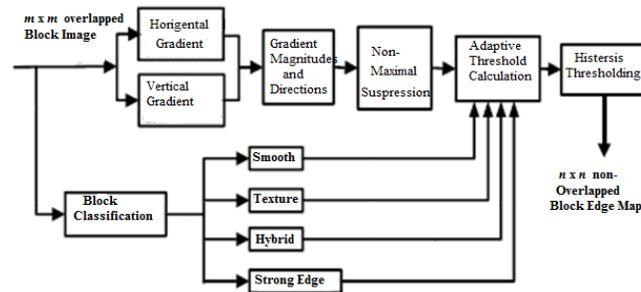


Fig. distributed canny edge detection algorithm

As discussed above, exploiting the local block statistics can significantly improve the edge detection performance for the distributed Canny edge detector. For this purpose, in order to learn the statistics of local blocks, we use a training database consisting of natural images. For each image in the database, we first divide the image into  $n \times n$  non-overlapping blocks and classify the blocks into six types, uniform, uniform/texture, texture, edge/texture, medium edge, and strong edge block, by adopting the block classification method. This classification method utilized the local variance of each pixel using a  $3 \times 3$  window that is centered around the considered pixel in order to label it as of type edge, texture, or uniform pixel. Then, each block is classified based on the total number of edge, texture, and uniform pixels in the considered block.

## II. LITERATURE SURVEY

The systems proposed earlier are explained in short here Paulo Possa, Naim Harb [1], proposed the design of a Multi-Resolution for Real-Time Edge and Corner Detection using FPGA. In this paper, we have presented a new flexible architecture for Canny and Harris feature detectors. This new architecture has a reduced latency and memory requirement supporting images with variable resolutions.

Kyu Park, N. Singhal, and C. W. Kim [2], proposed the General Purpose Graphic Processing Unit (GPGPU) has emerged as a powerful and accessible parallel computing platform for image processing applications. Studies of GPGPU accelerated Canny edge detection have been presented. 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.

N. D. Narvekar and L. J. Karam, [3] For this purpose, the psycho-visual sharpness metric of [3] is used, which primarily depends on the edge map to estimate the cumulative probability of detecting blur (CPBD) in an image. This metric was shown in [3] to achieve the best performance among existing sharpness/blur metrics. It calculates the probability of detecting blur at each edge in an image and then obtains a final quality score over the entire image by evaluating the cumulative probability of blur detection. The effect of replacing the frame-based Canny edge detection with the proposed distributed Canny edge detection on the sharpness metric of [3], for masks of different sizes and for different block sizes. Note that a  $512 \times 512$  block corresponds to the original frame-based Canny edge detector. The PCC is robust to changes in block size. In conclusion, the proposed distributed Canny algorithm can detect all the psycho-visually important edges for images with moderate noise and blur levels, similar to the original frame-based Canny algorithm.

C. Gentsos, S. Nikolaidis, and N. Vassiliadis [4], proposed the design of a new parallel Canny edge detector FPGA implementation. The parallel implementation in [4] 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.

W. He and K. Yuan [5], proposed design of an optimal edge detector and has been widely used for edge detection. This paper presents a self-adapt threshold Canny algorithm 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. This results in increased latency as compared to the existing implementations. Furthermore, the non-parallel implementations result in a decreased throughput as compared to the parallel implementations. The issue of increased latency and decreased throughput is becoming more significant with the increasing demand for large-size high-spatial resolution visual content.

D. V. Rao and M. Venkatesan [6], proposed the design of an efficient reconfigurable architecture of edge detection algorithm using handle-C, the Canny edge detection algorithm ontoFPGA-based platforms. The two FPGA implementations translate the software design directly into VHDL or Verilog using system-level hardware design tools, which results in a decreased timing performance.

So we saw that the systems proposed earlier contained the original Canny algorithm relies on frame-based statistics to predict the high and low thresholds and thus has latency proportional to the frame size. In addition, the new algorithm uses a nonuniform gradient magnitude histogram to compute block-based hysteresis thresholds. The resulting block-based algorithm has a significantly reduced latency and can be easily integrated with other block-based image codecs.

### III. PROPOSED WORK

#### MATLAB TO CONVERT IMAGE INTO TEXT/PIXEL VALUE

The image of which the edges is to be detected will be the input to the propose work. The Matlab is used to convert image into text/pixels value, here we get the representation of image into its corresponding pixels. We will apply the compression to the output of Matlab and then applying distributed canny edge detector algorithm in VHDL. At the output of algorithm we will get edge detection of applied input image into text /pixels format. To convert text/pixel into image we will use Matlab and the output of the Matlab we will get edge detected image as a output.

### IV. CONCLUSION

Thus, our propose project will detect the edges efficiently with reduction in the processing speed and reduced the memory requirement. In order to reduce the large latency and meet real-time requirements, we present a distributed Canny edge detection algorithm which has the ability to compute edges of multiple blocks at the same time and also increase the throughput. The propose design will benefits in: 1) a significant reduction in the latency; 2) better edge detection performance; 3) the possibility of pipelining the Canny edge detector with other block-based image codecs. In this paper we are proposing compression of image and then applying distributed canny edge detector algorithm in VHDL.This shows that the processing speed and memory requirement is reduced. Finally, the algorithm is mapped onto a Xilinx FPGA platform.

## **V. REFERENCES**

1. Paulo Ricardo Possa, Sidi Ahmed Mahmoudi, Naim Harb, Carlos Valderrama and Pierre Manneback, "A Multi-Resolution FPGA-Based Architecture for Real-Time Edge and Corner Detection", IEEE TRANSACTIONS ON COMPUTERS, VOL. XX, NO. XX, JANUARY 2013.
2. 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. 2011.
3. N. D. Narvekar and L. J. Karam, "A no-reference image blur metric based on the cumulative probability of blur detection (CPBD)," IEEE Trans. Image Process., vol. 20, no. 9, pp. 2678–2683, Sep. 2011..
4. 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.
5. 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.
6. 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.
7. S. Varadarajan, C. Chakrabarti, L. J. Karam, and J. M. Bauza, "A distributed psycho-visually motivated Canny edge detector," IEEE ICASSP, pp. 822 –825, Mar. 2010.