Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 2, February 2015, pg.293 – 298

RESEARCH ARTICLE

Determination of Edges by Automatic Threshold Value Generation

Pardeshi Ragini Sanjay
Department of Computer Engineering (Savitribai phule Pune University),
Sir Visvesvaraya Institute Technology, Chincholi, Tal-Sinner, Dist- Nashik, India
raginipardeshi03@gmail.com

Abstract: *As the superior performance of canny algorithm it is most widely used now days. As it is based on frame level static its accuracy is higher than that of other edge algorithms. We proposed a algorithm to perform canny algorithm at block level with edge detection performance compared with original frame level canny algorithm. Directly applying the original Canny 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 avoid such problems we proposed this algorithm to detect edge based on block type and local distribution of gradients in image block. In this algorithm it uses a uniform gradient magnitude histogram to compute block based hysteresis threshold. The resulting block-based algorithm has a significantly reduced latency and can be easily integrated with other block-based image codec's*

Keywords: Canny Edge Detector, Threshold, Latency, Codec's, Hysteresis

1. INTRODUCTION

In image processing algorithms edge detection is most commonly used. Algorithms used in edge detection are image enhancement, image segmentation, tracking and image/video coding. The canny edge detection is one of the most standard algorithm in existing and also has best performance. 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 statistics. This feature of canny algorithm makes it computationally complex than 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. The original Canny algorithm computes the higher and lower thresholds for edge detection based on the entire image statistics, which prevents the processing of blocks independent of each other [4].

The Canny edge detector has remained a standard for many years and has best performance. 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 statistics. 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 [2]. Canny algorithm is still depending on a correct setting of the threshold greatly. They will miss some edges or detect some spurious edges when the threshold is not set a proper value. Therefore, this algorithm do not suit for mobile robot vision system in which all of the operation should be done by the robot controller and the environment changes constantly. Another shortcoming of the commonly used canny algorithm is that the computational cost of the algorithm is very high and it cannot be implemented in real time to meet the needs of mobile robot vision system [5].

To overcome these shortcomings of traditional canny algorithm, an adaptive threshold selection algorithm is proposed in [2] which compute the high and low threshold for each block based on the type of block and the local distribution of pixel gradients in the block. Each block can be processed simultaneously, thus reducing the latency significantly. Furthermore, this allows the block-based canny edge detector to be pipelined very easily with existing block-based codec, thereby improving the timing performance of image/video processing systems. Most importantly, conducted conformance evaluations and subjective tests show that, compared with the frame- based canny edge detector, the proposed algorithm yields better edge detection results for both clean and noisy images.

2. LITERATURE REVIEW

The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Several algorithms exists, and this worksheet focuses on a particular one developed by John F. Canny (JFC) in 1986. Even though it is quite old, it has become one of the standard edge detection methods and it is still used in research. The aim of JFC was to develop an algorithm that is optimal with regards to the following criteria:

1. Detection: The probability of detecting real edge points should be maximized while the probability of falsely detecting non-edge points should be minimized. This corresponds to maximizing the signal-to-noise ratio.

2. Localization: The detected edges should be as close as possible to the real edges.

3. Number of responses: One real edge should not result in more than one detected edge (one can argue that this is simplicitly included in the first requirement). With Canny's mathematical formulation of these criteria, Canny's Edge Detector is optimal for a certain class of edges.

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 my Sobel and Laplace edge detection tutorial, but I just wanted reemphasize the point of why you would want to detect edges. The Canny edge detection algorithm is known to many as the optimal edge detector. Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He was very successful in achieving his goal and his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection". In his paper, he 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. Based on these criteria, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

3. CANNY EDGE DETECTION METHOD

Canny developed an approach to derive an optimal edge detector based on three criteria related to the detection performance. The model was based on a step edge corrupted by additive white Gaussian noise. A block diagram of the canny edge detection algorithm is shown in Fig. The original canny algorithm [6] consists of the following steps:

1. Smoothing the input image by Gaussian mask.
2. Calculating the horizontal gradient Gxand vertical gradient Gyat each pixel location by convolving with gradient masks.
3. Computing the gradient magnitude G and direction θGat each pixel location.
4. Applying Non-Maximal Suppression (NMS) to thin edges.
5. Computing high and low thresholds based on the histogram of the gradient magnitude for the entire image.
6. Performing hysteresis Thresholding.
7. Applying morphological thinning on the resulting edge map.

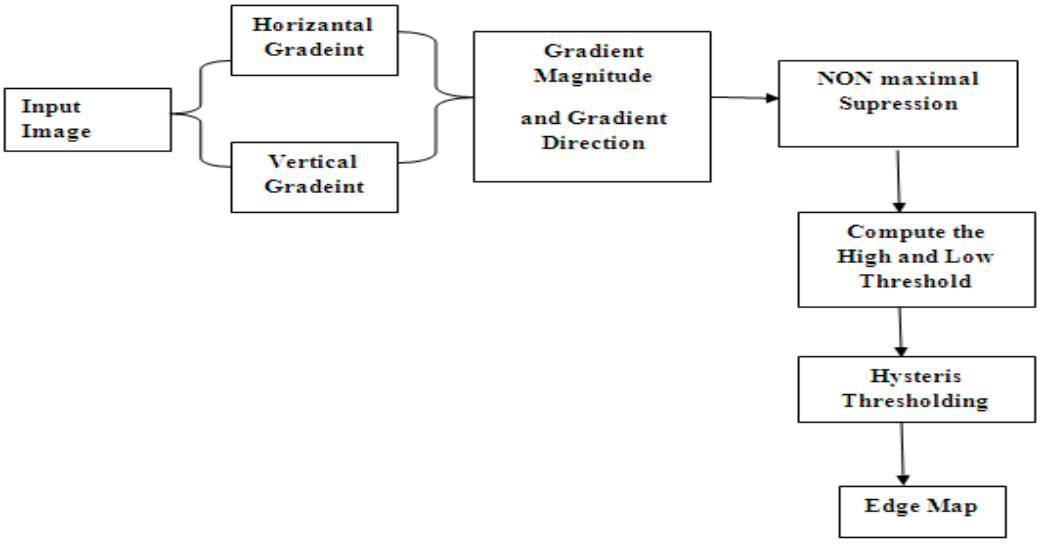

Fig1. Block Diagram of the Canny Edge Detection Algorithm

1. Smoothing: Smoothing of the image is achieved by Gaussian convolutions. Blurring of the image to remove noise.
2. Finding the Gradients: Gradient calculation is performed using Finite-impulse-Response (FIR) gradient masks designed to approximate 2D sample version of partial derivative of Gaussian Function. The size of gradient mask used by canny edge detector is function of standard deviation.

Calculation of Gx and Gy: The actual images are always discrete; we define the direction as vertical, horizontal, left-diagonal and right-diagonal of the 3x3 adjacent window of current pixel.The first-derivative of each direction is then calculated by E = ({-1,1})3 X 3 X H(i,j)3 X 34. Proposed Distributed Canny Edge Detection

The Canny edge detection algorithm operates on the whole image and has a latency that is proportional to the size of the image. While performing the original canny algorithm at the block- level would speed up the operations, it would result in loss of significant edges in high-detailed regions and excessive edges in texture regions. Natural images consist of a mix of smooth regions, texture regions and high-detailed regions and such a mix of regions may not be available locally in every block of the entire image. In [6], distributed canny edge detection algorithm is proposed, which removes the inherent dependency between the various blocks so that the image can be divided into blocks and each block can be processed in parallel.

In the proposed distributed version of the Canny algorithm, the input image is divided into m × n overlapping blocks, and the blocks are processed independent of each other. To prevent edge artifacts and loss of edges at the boundaries, adjacent blocks overlap by (L-1)/2 pixels for L× L gradient mask. However, for each block, only edges in the central n × n (where n= m-L+1) non- overlapping region are included in the final edge map. In the proposed algorithm, Steps 1 to 3 and Steps 5 to 7 are the same as in the original canny algorithm except that these are now applied at the block level. The high and low gradient threshold selection step of the original Canny (Step 4) is modified to enable block-level processing. Analysis of natural images showed that a pixel with a gradient magnitude of 4 corresponds to a psycho-visually significant edge. Also, a pixel with a gradient magnitude of 2 and 6 corresponds to blurred edges and very sharp edges, respectively. The proposed threshold selection algorithm was designed based on these observations and is as shown below:

1) Calculating the horizontal gradient Gxand vertical gradient Gyat each pixel location by convolving with gradient masks.

2) Computing the gradient magnitude G and direction θGat each pixel location.

3) Applying Non-Maximal Suppression (NMS) to thin edges.

4) Parallel block-level processing without degrading the edge detection performance.

5) Performing hysteresis thresh holding to determine the edge map.

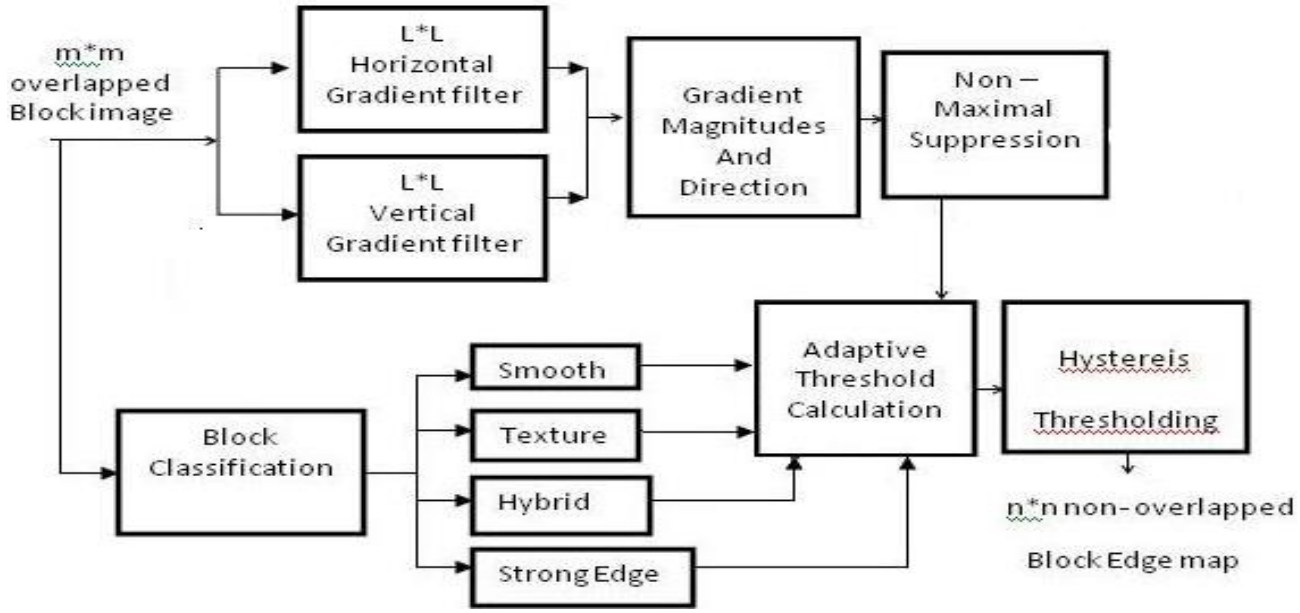

Fig2. Distributed Canny Edge Detection Algorithm Block Diagram

Natural images consist of a mix of smooth regions, texture regions and high-detailed regions and such a mix of regions may not be available locally in every block of the entire image. The input image is divided into m×m overlapping blocks. The adjacent blocks overlap by (L − 1)/2 pixels for a L × L gradient mask

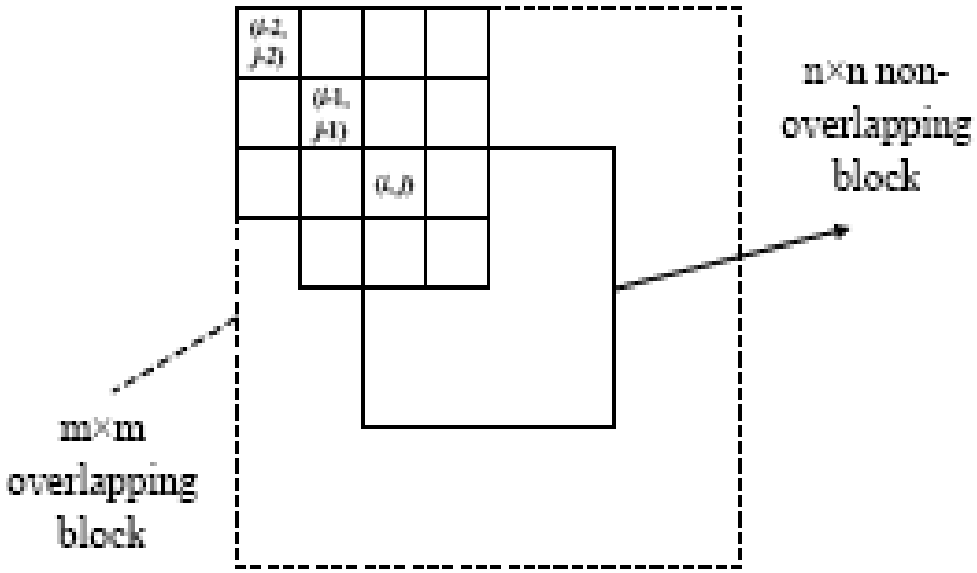

Fig3. Structure of m x m overlapping block

4. CONCLUSION

Determination by Canny Edge Detector does automatic threshold calculation; it can be implemented for day to day utilized application. Propound method presents non uniform quantized histogram calculation method in order to reduce the large latency and meet real-time requirements, we presented a novel distributed Canny edge detection algorithm which has the ability to compute edges of multiple blocks at the same time. Distributed Canny edge detection algorithm that results in a significant speed up without sacrificing the edge detection performance.

REFERENCES

[1] 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.

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

[3] 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.

[4] D. V. Rao and M. Venkatesan, "An efficient reconfigurable architecture and implementation of edge detection algorithm using handle-C," in Proc.

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

[6] Niranjan D. Narvekar and Lina J. Karam , "A No-Reference Image Blur Metric Based on the Cumulative Probability of Blur Detection (CPBD)", IEEE TransactionsOn Image Processing, Vol. 20, No. 9,September 20112678-2683.Fig4. Structure of m x m overlapping block

