International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Distributed Canny Edge Detector: Algorithm & FPGA Implementation

Veeranagoudapatil[1], ChitraPrabhu[2]
PG student[1], Assistant professor[2]
[1,2]Department of ECE, SDIT- Mangalore, India

Abstract-The Canny edge detector is one of the most widely used edge detection algorithms due to its superior performance. Unfortunately, not only is it computationally more intensive as compared with other edge detection algorithms, but it also has a higher latency because it is based on frame-level statistics. 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-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 solve this problem, we present a distributed 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. In addition, the new algorithm uses a non-uniform gradient magnitude histogram to compute block-based hysteresis thresholds. The resulting block-based algorithm has significantly reduced latency and can be easily integrated with other block-based image codecs. It is capable of supporting fast edge detection of images and videos with high resolutions; including full-HD since the latency is now a function of the block size instead of the frame size. In addition, quantitative conformance evaluations and subjective tests show that the edge detection performance of the proposed algorithm is better than the original frame-based algorithm, especially when noise is present in the images. Finally, this algorithm is implemented using a 32 computing engine architecture and is synthesized on the Xilinx Virtex-5 FPGA. The synthesized architecture takes only 0.721 ms (including the SRAMREAD/WRITE time and the computation time) to detect edges of 512 × 512 images in the USC SIPI database when clocked at 100 MHz and is faster than existing FPGA and GPU implementations.

I. INTRODUCTION

EDGE detection is the most common pre-processing step in many image processing algorithms such as image enhancement, image segmentation, tracking and image/video coding. Among the existing edge detection algorithms, 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. Many implementations of the Canny algorithm have been proposed on a wide list of hardware platforms. There is a set of work on Deriche filters that have been derived using Canny's criteria and implemented onASIC-based platforms. The Canny-Deriche filterisa network with four transputers that detect edges in a256×256 image in 6s, far from the requirement for real-time applications. Although the design in improved the Canny-Deriche filter implementation of and was able toprocess 25 frames/s at 33 MHz, the used off-chip SRAM memories consist of Last-In First-Out (LIFO) stacks, which increased the area overhead compared. The proposed a new organization of the Canny-Deriche filter in which reduces the memory size and the computation cost by a factor of two. However, the number of clock cycles per pixel of the implementation varies with the size of the processed image, resulting in variable clock-cycles/pixel from one image size to another with increasing processing time as the image size increases. There is another set of work on mapping the Canny edge detection algorithm onto FPGA based platforms. The two FPGA implementations in translate the software design directly into VHDL or Verilog using system-level hardware design tools, which results in a decreased timing performance as shown later in of this paper.

II. EXISTING METHODOLOGY

In edge detection, the Sobel operator is used commonly. The Sobel operator is a classic first order edge detection operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is the corresponding norm of this gradient vector. The Sobel operator only considers the two orientations which are 0°and 90°convolution kernels. The operator uses the two kernels which are convolved with the original image to calculate approximations of the gradient. The two convolution kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations.

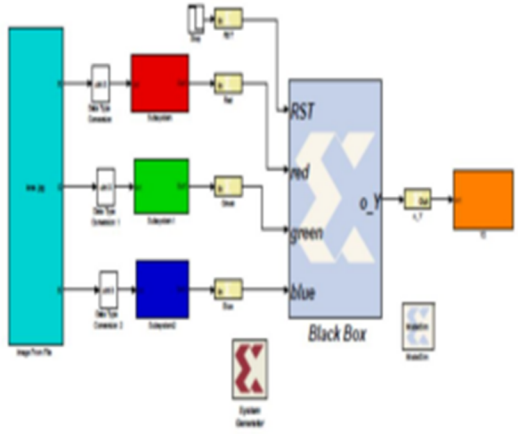

Fig 1Sobel edge detection

The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these Gx andGy).These can then be combined together to find theabsolute magnitude of the gradient at each point.

III. PROPOSED METHODOLOGY

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 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. 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. 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.

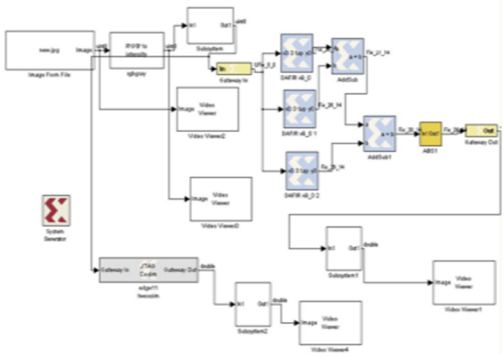

Fig 2 Canny edge detection

IV. ADVANTAGES

A. Using probability for finding error rate
B. Localization and response.
C. Improving signal to noise ratio.

D. Better detection specially in noise condition

V. APPLICATIONS

A. Suppress noise.
B. Canny edge detector is adaptable to various environments.
C. Canny edge detector has been modified in many different ways to solve specific problems.
D. Robot applications.
E. The brain MR image analysis in the applications of medicine.

VI. CONCLUSION

The original Canny algorithm relies on frame-level statistics to predict the high and low thresholds and thus has latency proportional to the frame size. 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. To support this, an adaptive threshold selection method is proposed that predicts the high and low thresholds of the entire image while only processing the pixels of an individual block. This results in three benefits: 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 addition, a low complexity non-uniform quantized histogram calculation method is proposed to compute the block hysteresis thresholds. The proposed algorithm is scalable and has very high detection performance. We show that our algorithm can detect all psycho-visually important edges in the image for various block sizes. Finally, the algorithm is mapped onto a Xilinx Virtex-5 FPGA platform and tested using Model Sim. The synthesized results show 64% slice utilization and 87% BRAM memory utilization. The proposed FPGA implementation takes only 0.721ms (including the SRAM read/write time and the computation time) to detect edges of 512 × 512 images in the USC SIPI database when clocked at 100MHz. Thus the proposed implementation is capable of supporting fast real-time edge detection of images and videos including those with full-HD content.

REFERENCES

[1] R. Deriche, “Using canny criteria to derive a recursively implementedoptimal 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. Asia South Pacific 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. IEEE Conf. ITCC, vol. 2. Apr. 2004, pp. 843–847.
[5] H. Neoh and A. Hazanchuck, “Adaptive edge detection for real-time video processing using FPGAs,” Altera Corp., San Jose, CA, USA, Application Note, 2005.