

# An Efficient Approach for Edge Detection Hardware Accelerator for Real Time Video Segmentation using Laplacian Operator

T. P. Lavanya  
M. E VLSI

Adhiparasakthi College of Engineering,  
Kalavai, India.

S. Premkumar  
Asst. Professor

Adhiparasakthi College of Engineering,  
Kalavai, India.

**Abstract**—Video segmentation is a very important one in video image processing application that deployed by video surveillance system. The high computation speed is required to support real time performance. This paper presents the implementation of VLSI based hardware accelerator design for real time video segmentation system. The algorithm of Laplacian edge detection operator is used to develop this hardware accelerator. The NTSC standard definition video is digitized at 720x480 with a video rate of 30 frames per second. To develop hardware accelerator datapath architecture the management of memory access is deployed and architecture based pipeline are made with the potential improvements in acceleration to the read data pixel from memory. In addition, a finite state machine is used to ensure the hardware accelerator controls the sequence of derivative computation, the write and read operations. The hardware accelerator design is implemented on Altera Stratix III DSP development board and enables application of co-processor without requiring new application specific digital signal processor. The implementation result shows a field programmable gate arrays (FPGAs) acting as coprocessor platforms for user defined co-processor, with real time performance at a frame rate of 30 fps with a resolution of 720 x 480. The parallel and pipeline technique are utilized in memory access, resulting more than 80% memory bandwidth reduction.

**Keywords**— Edge Detection, Hardware Accelerator, Real Time, VLSI, FPGAs

## I. INTRODUCTION

Video Surveillance technology becomes more attractive and plays important role nowadays. This is in part due to the feature of real time performance in displaying of object image which captured by scattered surveillance cameras. The real time performance makes security guard in the control room able to give fast respond to the critical situation. In order to differentiate between critical situation and ordinary event which are showed by video surveillance the video image processing feature need to embedded. This feature reduces attention to ordinary events which are not security relevant on the video feeds. Many feature of video image processing are available.

The most common embedded video image processing feature in video surveillance is video segmentation and object recognition. One of the key algorithms used in video segmentation and object recognition is edge detection. Edge detection is operated by mathematic algorithm. Early edge detection methods used local operators to approximately

compute the first derivative of grey level gradient of an image in the spatial domain. The position of local maximum of the first derivative is considered to be the edge point. Examples of gradientbased edge detectors are Prewit and Sobel operators [2, 3]. The Laplacian of Gaussian (LOG) operator for edge detection has been proposed by Marr and Hildreth [4] which uses a Gaussian function for image smoothing, then computes the second derivative and the edge points are displayed using zero crossing points. An alternative method is an optimal edge detector like the Canny operator, for 2 dimensional images. This operator can give the edge information of both intensity and direction [5].

With the advance of computation speed of application specific integrated circuit (ASIC) and dedicated coprocessor has lead to research study on the optimization of hardware architecture for faster speeds. This paper presents VLSI based hardware accelerator to expedite the edge detection computation using the technique of memory bandwidth reduction for real time video segmentation. The rest of paper is organized as follows. The edge detection algorithm optimization and an efficient edge detection computation are discussed in section II and section III, respectively. Section IV and V discuss VLSI based hardware accelerator architecture and implementation result, respectively. Last but not least, conclusion of this paper is discussed in section VI.

## II. EDGE DETECTION ALGORITHM OPTIMIZATION

It is assumed that Laplacian edge detection algorithm will handle monochrome images of 720x480 pixels, each having 8 bits, row upon row of pixels stored in the memory sequentially, left to right in a row, at the sequence address. The values of pixels are represented by unsigned integers where 0 is the lowest value representing the black color and 255 as the highest value representing the white color. The video frame rate assumption is 30 frames per second which fulfills a real time performance requirement, the National Television System Committee (NTSC) standard. Due to the huge amounts of pixel data to be handled in a limited time, Laplacian edge detection algorithm must pass the process of optimization in order to accommodate these assumptions. This process is also required to minimize hardware complexity and support portable design. In general, the optimized algorithm is achieved by reducing the complexity

of the algorithm or simplification of the algorithm [1, 5, 6]. The process of simplification of the algorithm aims to shorten the program flow and indirectly leads to a core function of the program. This process will reduce the complexity of the algorithm, trigger low usage of memory and result in decreased power consumption. The Laplacian edge detection algorithm works by a second order derivatives computation for pixel intensity in the direction of x and y and searches for maxima and minima values through derivatives. The derivatives formula and its simplification are given in Eq.(1)

Let pixel value of a Part of an image is  $f(x,y)$

$$f(x,y) = \begin{bmatrix} z1 & z2 & z3 \\ z4 & z5 & z6 \\ z7 & z8 & z9 \end{bmatrix}$$

0	-1	0
-1	4	-1
0	-1	0

Fig 1: Laplacian Operator

$$|D| = -Z2-Z4+4Z5-Z6-Z8$$

Where  $|D|$  is the magnitude of the derivative image pixel. The operation of Laplacian that searches for the approximate value of derivative pixel is called the convolution process. This process involves an accumulation of pixel with its eight surrounding neighbours and each multiplied by a coefficient. The operation of derivatives is operated by a couple of kernels as shown in Fig.1 Each of the pixel grids will be operated by the kernels, vertically and horizontally, in order to obtain the maximum value of edges. The value of  $D$  is utilized to find the absolute magnitude value of edge on each of the points.

The optimization process on this edge detection algorithm will be more focused in achieving computation efficiency and reducing memory access bottleneck. In order to realize these two achievements, technical analysis is involved and described in the following section.

### III. AN EFFICIENT EDGE DETECTION COMPUTATION

The efficient edge detection computation is achieved through several approaches e.g. computation element elimination, effective computation organization, etc [7,10]. The Laplacian edge detection algorithm consists of derivatives computation with a 3 x 3 convolution kernels as

the operator of the second order derivative, therefore technical analysis is performed on this derivative and its elements in order to eliminate the element of computation and enhance effective computation.

As the 3 x 3 convolution kernel is the basic element of the derivative computation, there is a possibility to conduct element elimination on this convolution kernels. Based on Fig.1, it is known that there are four zero values and understood that each of the values at this figure is a coefficient value for second order derivative computation. As the value of zero did not contribute to the result, these zero values are deleted from the derivative computation. Eliminating these zero values contributes to overall derivative computation efficiency through a reduction in the computation element. This achieves 44.4% of computation efficiency since there are 4 zero values out of 9 of total values in a 3 x 3 convolution kernels. Further analysis is required to solve the issue of memory access bottleneck before come to the final design of efficient edge detection computation. The original and second order derivative images are stored in the memory 32 bits wide and each having 8-bit byte occupying an individual address. Each video frame pixel is stored in one byte. A row of pixels within a frame enters the memory from the left to the right at consecutive addresses and rows occupy the memory from top to bottom. Read or Write operations consume 20ns for memory access with 2 cycles of 110 MHz system clock. In the earlier assumption it was mentioned that the proposed design is allocated for monochrome images of 720x480 pixels with 30 frames per second of video frames. Therefore the total pixels that need to be processed per second are 720x480x30 10.368.000 pixels. Based on these assumptions, technical analysis is carried out to accommodate the system requirement which focuses on the stages of the computation requiring memory access. There are 3 stages of computation that require memory access e.g. write video input, read data pixel and write computation result. Implementing an aggregate four pixels in the write video input operation will reduce allocated memory bandwidth to one twenty. By introducing this technique, the memory access has improved 4 times compared to the normal approach. The same approach is implemented for writing the computation result to the memory. While, read data pixel operation is finalized by computation area storage approach. The computation process is performed by moving 3 pixels in the same column from the right to the left and it only needs 3 reads for each of the 4 adjacent pixels in the computing process. This approach manages to achieve 0.15 of the memory bandwidth access performance. The complete a proposed efficient edge detection computation design is shown in Fig 2.

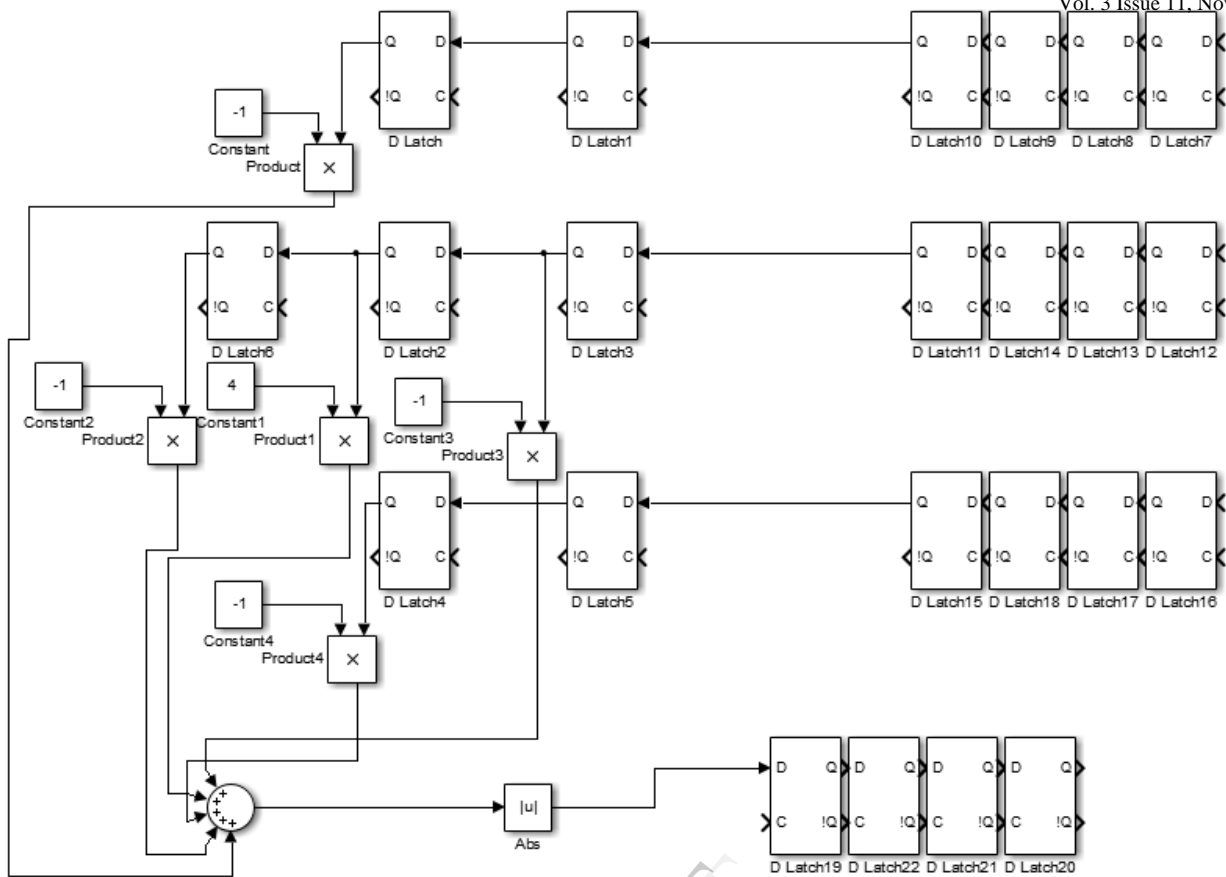


Fig. 2: An efficient edge detection computation design

#### IV. VLSI BASED HARDWARE ACCELERATOR ARCHITECTURE

There are 8 blocks functioning inside this hardware accelerator architecture, such as a clock divider controller which converts block from 110MHz clock input into 54 MHz with 18.5ns pulse width. This pulse is needed to support the operational of the accelerator control block and control address block. The second block is the accelerator control. This block performs Laplacian edge detection computation process and also acts as a controller for the flow of image data from and to the memory unit. The output of the accelerator control block will utilize the input of the following blocks: address counter, memory before access controller, and acknowledge generator. The next block is the address counter. The address location for the original image data and derivative image data is assigned by the address counter block. The assignment of the specific address is performed by increment on the base address for the respective image data pixel. The other block is the memory access controller to control the changing format of data when accessing from memory, from parallel to serial or vice versa. It is also in charge of memory control access register block to receive data from memory or to transmit the data to memory. One more block is memory access register also connected to the supporting function for memory access. The data is temporarily saved in this block being sent to or received from memory. Once the packet of data is received r sent completely this block will trigger acknowledge generator to produce an

acknowledge signal. Another block is the memory unit where data pixel, derivative address, 1st row rawpix address, 2nd rawpix address, and 3rd rawpix address are stored. The address decoder block determines the address to be accessed in the memory unit..

#### V. IMPLEMENTATION RESULT

A proposed VLSI based hardware accelerator architecture was implemented in VHDL language. Even though Stratix III based DSP board utilized as the target of design implementation, a optimized Matlab-to-VHDL converted was not used for algorithm implementation. The problem in utilizing Matlab-to-VHDL converted is required some library module support from third party and also takes longer time in performing design simulation. The VHDL language based algorithm implementation is not facing such this problem. The implementation result is represented as images. The functional test is performed when a good result is displayed by the timing verification and the test is supported by software programming to show the functionality of design according to specification. The timing verification has verified the proposed edge detection hardware accelerator architecture including all its sub-module architecture and the verification result shows this architecture able to accommodate the function of edge detection computation with capability to process 720 x 480 image pixels.



Fig.3. Input Image

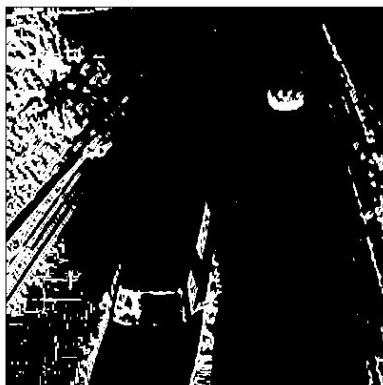


Fig. 4. Output Using Sobel Operator



Fig. 5. Output Using Laplacian Operator

## VI. CONCLUSION AND FUTURE WORK

VLSI based hardware accelerator edge detection design with improved memory access was implemented on a ALTERA Cyclone III, FPGA development board. The improved memory access has shown it significantly assisted real time performance in video segmentation. Its application is recommended for a video surveillance system and it represents a much needed enhancement of current technology.

## REFERENCES

- [1] I.Yasri, N.H. Hamid, and N. B. Zain Ali, VLSI Based Edge Detection Hardware Accelerator for Real Time Video Segmentation System , International Conference on Intelligent and Advanced Systems (ICIAS2012)
- [2] A. Rosenfel, Computer vision: a source of models for biological visual process, IEEE Transactions on Biomedical Engineering 36 (1) (1989) 83-94.
- [3] I. Sobel, Neighbourhood coding of binary images fast contour following and general array binary processing, Computer Graphics and Image Processing 8 (1978) 127-135.
- [4] JD. Marr, E.C. Hildreth, Theory of edge detection, Proceedings of the Royal Society 207B (1980) 187-217.
- [5] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern analysis and Machine Intelligence 8 (6) (1986) 679-698.
- [6] A. J., Real-Time Signal Processing: Design and Implementation of Signal Processing Systems, Englewood Cliffs, NJ: Prentice-Hall,1999.
- [7] Q. S., Embedded Image Processing on the TMS320C6000 DSP Examples in code Composer Studio and MATLAB, Berlin: Springer Verlag, 2005.
- [8] N. Kehtarnavaz and M. Gamadia, Real-Time Image and Video Processing: From Research to Reality, Morgan & Claypool Publishers, 2006
- [9] .I.Yasri, N.H.Hamid, V.V.Yap, "Implementation of an FPGA based Sobel Edge Detection Operator", IGCES, 2008.
- [10] M.D. Ciletti, Modelling, Synthesis and Rapid Prototyping, Prentice Hall, 1999
- [11] P.J. Ashenden, Digital Design An Embedded System Approach Using VHDL, Morgan Kaufmann, 2008
- [12] S.Sarangi and N.PRath, Performance Analysis of Fuzzy-based Canny Edge Detector, IEEE Computer Society, 2007.
- [13] R.Maini and J.S.Sohal, Performance Evaluation of Prewitt Edge Detector for Noisy Images, GVIP Journal, Vol. 6, Issue 3, December, 2006.
- [14] C.Perra, F.Massida and D.D.Giusto, Image Blockiness Evaluation Based Sobel Operator, IEEE, 2005
- [15] M.Boo, E.Antelo and J.D.Bruguera, VLSI Implementation of an Edge Detector Based on Sobel Operator, IEEE, 1994.
- [16] M.Budagavi, Real Time Image and Video Processing in Portable and Mobile Devices, Journal Real Time Image Processing, 1:3-7, 2006.
- [17] Video and Image Processing Design Using FPGAs, White Paper, Altera Corp., March 2007.
- [18] Avalon Interface Manual, Altera Corp, 2008.