

# Real-Time VLSI Architecture for Detection of Moving Object Using Wronskian Determinant

R. Aguilar-Ponce, J. Tessier, C. Emmela, A. Baker, J. Das,  
J.L. TecpanecatI-Xihuitl, A. Kumar and M. Bayoumi

Center for Advance Computer Studies  
University of Louisiana at Lafayette  
PO Box 44330, Lafayette, LA 70504-4330 USA  
ruth@louisiana.edu

**Abstract**— Several computer vision applications require reliable object detection. Traditionally detection algorithms have been implemented solely in software. Object detection in upcoming wireless visual sensors has a need of hardware implementation with requirements of low power and small area. This paper introduces a hardware implementation of a real-time change detector based on Wronskian Determinant. This detection algorithm offers regularity, low complexity and accuracy as well as robustness against global illumination changes. The proposed architecture is able to process incoming frames on-the-fly, therefore requiring a small amount of memory. The maximum frame rate is 15 fps, however the implementation is flexible enough to allow analysis of less frames if required. Processing unit consist of a basic processing element implemented in pipeline fashion and adder tree to produce final results. The architecture was implemented using a XCV800 FPGA. The power consumption of the whole system is 121 mW.

## I. INTRODUCTION

Several applications such as video surveillance [1], remote sensing [2], object-based video coding [3], and smart cameras [4] require detection of object of interest. Such applications extract high level information from raw data, i.e. video stream. These systems require change detection algorithm that are accurate and robust. Detection can be performed by background subtraction.

Change detection algorithms take two digitized images as input and return the location of meaningful differences. The localized difference must contain changes due to motion of an object or addition or removal of object in the field of view. However, image difference can be caused by changes in the illumination or noise in the acquisition process. For that reason, change detection must be independent of illumination changes and robust against noise.

Several techniques have been proposed over the years, one of the most reliable techniques is Wronskian Change Detector (WCD) [5]. WCD employs the Wronskian of intensity ratios as a measure a change. A large mean or large variance of the intensity ratios increases the Wronskian

value. This method can detect object interiors and structural changes. Also, WCD is robust against illumination changes.

WCD is a suitable algorithm to be implemented in real-time due to its low complexity. Also, this technique requires only one previous frame; therefore it is appropriate for applications where resources are limited. The proposed architecture analyzes frames on-the-fly; therefore current frame is not stored. Consequently memory requirements are low. Our proposed architecture aims real-time operation while achieving low-power consumption. The architecture consists of a pipeline processing element that analyzes frames on the fly; partial results are stored on the memory unit to be reused.

The organization of the paper is as follow, section 2 discusses WCD algorithm while the proposed architecture is described in section 3. Simulation results and conclusions are presented in section 4 and 5 respectively.

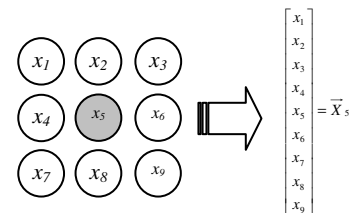


Figure 1. Region of support for pixel  $x_5$

## II. WRONSKIAN CHANGE DETECTOR

Wronskian Change Detector (WCD) is a non-recursive background subtraction technique that distinguishes changes based on intensity values. Consequently, WCD requires conversion of images into luminance values. Changes are detected based on intensity ratios variance. In order to determine if a change has occurred, a region of support is assigned to each pixel as shown in Figure 1. The size of the region of support can vary from  $3 \times 3$ ,  $5 \times 5$  and  $9 \times 9$  pixels.

WCD exploits the fact that the luminance ratios from two sources help to quantify the difference between the light

sources. In order to measure the change, Wronskian of a pixel is calculated by subtracting the sample mean of the intensity ratios inside the region of support from the squared ratios. Wronskian calculations are followed by thresholding where a change is detected if the values exceed a preset threshold. Typical thresholds are in the range of 0.6 to 0.7.

Detection is calculated by equation 1, where  $x$  and  $y$  are the intensity values of the current and previous frame respectively. The dimension of the region of support is  $n$ . In our simulations, sizes of region of support larger than 3 do not provide better results but increases the computational complexity. Therefore a fixed value of 3 is employed in our approach.

$$W\left(\frac{x}{y}\right) = \frac{1}{n} \left( \sum_{i=1}^n \frac{x_i^2}{y_i^2} - \sum_{i=1}^n \frac{x_i}{y_i} \right) = 0 \quad (1)$$

$W(x/y)$  detects changes corresponding to dark zones, while its inverse ration  $W^*(y/x)$  finds if a change has occurred in bright zones. Therefore, computing both values allows robust detection against global illumination changes.

### III. PROPOSED ARCHITECTURE

Our proposed architecture consists of three basic units: processing, memory and control units. The architecture is illustrated in Figure 2. The basic operation of the processor can be divided into two stages: processing frame and frame output. Incoming frame is processed on-the-fly by the pipeline processing element.

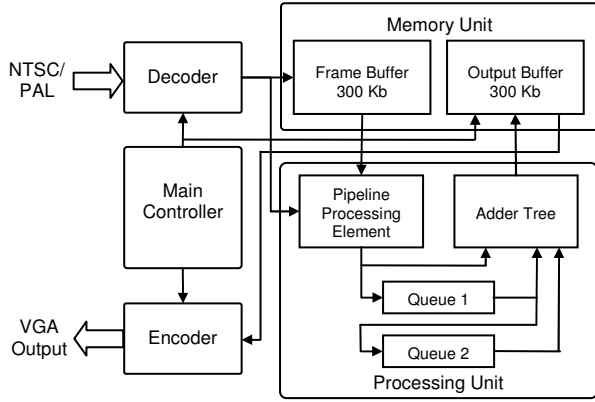


Figure 2. Proposed architecture.

#### A. Processing Element

The basic operation that must be performed on each pixel value is shown in equation 2.

$$D(x_i, y_i) = \frac{x_i}{y_i} \left( \frac{x_i}{y_i} - 1 \right) \quad (2)$$

Equation 1 can be rewritten as shown in equation 3, where  $TH$  is a threshold. Our processing element must

perform this calculation to prove what degree of linear independence exists between  $x_i$  and  $y_i$ , which will be used to determine if a change has occurred.

$$W\left(\frac{x}{y}\right) = \frac{1}{n} \sum_{i=1}^n D(x_i, y_i) \leq TH \quad (3)$$

In designing the PE there are two main concerns, the first one is how to capture the range of the function with only 8-bit unsigned arithmetic. The second concern is guaranteeing precision, considering that [5] suggests threshold values in the range of 0.6 to 0.7 to detect a change. In order to solve these problems, the PE must be designed to capture the range of  $D(x_i, y_i)$  that could indicate a change. Therefore, the equation 3 must be scaled so that an unsigned 8-bit integer threshold can be used and all overflows are saturated. Since the range of the threshold is relatively small, we can achieve near floating-point accuracy. Only the partial range of  $D(x_i, y_i)$  where  $TH_{min} \leq D(x_i, y_i) \leq nTH_{max}$  is significant, where  $TH_{min}$  and  $TH_{max}$  are the minimum and maximum threshold to be used. Consequently,  $D(x_i, y_i)$  must comply with the following equation

$$\frac{x_i^2}{y_i^2} - \frac{x_i}{y_i} - nTH_{max} = 0 \quad (4)$$

Solving the equation,

$$\frac{x_i}{y_i} = \frac{1 + \sqrt{1 - 4nTH_{max}}}{2} \quad (5)$$

Using a maximum threshold of 1.7 in a 9-dimensional vector, the 8-bit result of  $x_i/y_i$  could be a 3/5 fix-point number because any result greater than 4.443 would exceed  $nTH_{max}$  and could be saturated. Thus, an equivalent operation for 3/5 fixed point would be

$$D(x_i, y_i) = 2^{10} \left[ \frac{x_i}{y_i} \left( \frac{x_i}{y_i} - 1 \right) \right] \quad (6)$$

This solved the problem of precision, but creates results that are too large to sum  $n$  times. For that reason, the five least significant bit of the product are discarded after multiplication, and the rest of the bits are employed as the result leading to a final equation.

$$D(x_i, y_i) = 2^{10} \left[ \frac{x_i}{y_i} \left( \frac{x_i}{y_i} - 1 \right) \right] \gg 5 \quad (7)$$

Since all overflows are saturated at 255, then 254 corresponds to  $TH_{max} = 254 / (24n) = 0.8819$  and 0 corresponds to  $TH_{min} = 0.0313$ . This yields an acceptable range of

thresholds with the use of only 8-bit arithmetic and an 8-bit threshold. The final design of the PE is illustrated in Figure 3. The PE is implemented in pipeline fashion. Latches are introduced in the middle of the dividers, between divider and multiplier stage and in the middle of the multipliers. The number of stages within the PE block is 4 and the operating frequency is 25 MHz. Thus, we can process 4 pixels at a time. The processing element gets the inputs from the decoder, either directly (for the current frame) or through memory module (for the preceding frame). The decoder allows the luminance values within the range of 16 to 235.

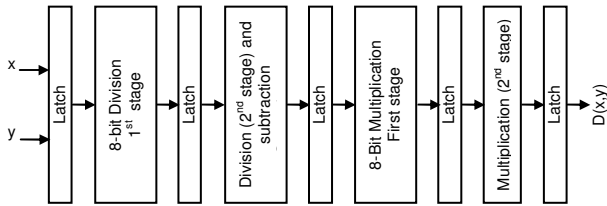


Figure 3. Processing Element Design.

### B. Processing Unit

The design uses control signals to pad the image by grounding the bus whenever it is required. The adder trees sum PE outputs to produce the final results. These results are compared to the threshold, and the change/no change bits are then stored into the output frame. The vertical padding control signal enables/disables summing the last iteration's partial sums. This allows the padding at the beginning and end of each line.

The architecture processes pixels on-the-fly. The system accepts video frames on American Video standard, National Television System Committee (NTSC). The NTSC standard displays 60 fields per second. Each field is composed by even and odd lines. The even and odd fields are displayed sequentially, thus interlacing the full frame. The proposed architecture also accepts frames on PAL standard. PAL standard is the dominant television standard in Europe. The distinction between these standards is that color is handled differently. The NTSC signal transmits the odd fields first and then the even fields. Accordingly, to apply WCD for a particular pixel, we calculate the  $D(x_i, y_i)$  values either for the neighbor pixels of the odd-fields or the even-fields. Once the NTSC signal is decoded to digital video, the pixel data will be transmitted to the PE for calculation and to the memory module, which is used to store the preceding frame data. When we have calculated values from PE for two lines, the adder tree starts operation and calculates  $W(x/y)$ . The calculated values from the adder tree are stored in a memory module. Once the calculation for all the pixels for even- or odd- fields of a particular frame is completed, the VGA encoder gives the results. The implementation uses two adder trees. One adder tree calculates the Wronskian values and the other calculates the Wronskian Conjugate values.

The principle of Wronskian change detection shows that for any change, either of the Wronskian or Wronskian conjugate can detect the changes. We have incorporated both of them into our design. A comparator compares these two values (Wronskian and Wronskian conjugate) and outputs the one that detects changes.

Our current implementation provides only the change values of the concerned frame as the output. Thus the output will be a combination of either black or white values. However, if we have memory module configuration with higher bandwidth, we can impose the changed values on the current frame for output. This will not require any significant modification in the proposed architecture. The system can process 15 frames per second

### C. Main Controller

The system is managed by the control unit. The controller has three states: process, display and idle states. In process state, the system input and calculates Wronskian values. Display state shows the output through the encoder. While in idle state the process unit does not performed any action. The controller ensures this operation flow through activation and de-activation of different components. During processing, the controller activates the PE, de-activates the output and controls the data from and to the memory module that holds the preceding frame data. For each frame, after one line is processed, the controller activates the adder tree and the memory module that holds the calculated values. Once all the data for a frame is transmitted from the decoder, the controller de-activates the PE block. The adder tree operates for one more line to calculate WCD values for the last line of the frame. After this addition is completed, the controller de-activates the adder tree, resets the output encoder and the system enters into the display phase. The architecture also allows different frame rate analysis selection. The maximum frame rate is 15 frames per second. If the system process less than 15 frames per second, the circuit will remain idle for the rest of the frames of a second. The controller will keep all the units de-activated during such idle phase.

The system can calculate any of the three values: Wronskian values, Wronskian conjugate values or both on the basis of change-detection. The system is flexible enough to select any of these three options. Accordingly, the controller will activate and de-activate corresponding units.

### D. Memory Unit

For storing the preceding frame data, we used a 300Kb memory. Another memory of same size is required to store the output values. The memory is addressed by 19 bits that includes field index for a frame, vertical addressing and horizontal addressing. The values for 2-pixels, i.e., 16 bits of data is read and stored at a time. The XSV FPGA board v. 1.1 used in our implementation contains memory modules with a single port that we have used both for reading and storing data.

#### IV. SIMULATION RESULTS

Implementation of the proposed architecture was done in VHDL using Mentor Graphic Modelsim Simulator. Synthesis was done using Synopsis Synthesis Tools targeting Xilinx XCV800 FPGA. Xilinx Virtex FPGA provides a framework for processing video and audio signals. The XSV-800 board can accept PAL, SECAM, or NTSC video with up to 9-bits of resolution on the red, green, and blue channels and can output video images through a 110 MHz, 24-bit digital to analog converter. Two independent banks of 512K x 16 SRAM are provided for local buffering of signals and data.

The implementation of the system is done with a fixed region of support size of  $3 \times 3$ . The main components of the PE are divider, adder and multiplier. Multiplication is done by Booth algorithm because it represents a good trade-off between speed and power for 8 bit fixed point arithmetic [6]. After reviewing several division algorithms, we conclude that integer restoring using 8 conditional subtractors is simple and fast enough for our application [7]. Booth multiplier and divider are implemented in two pipeline stages.

The architecture is capable of analyze frame size of  $640 \times 480$  pixels. The test images for this simulation are shown in Figure 4, while the results of the simulation on indoor and outdoor scene are shown in Figure 5. The test result was obtained using a threshold value of 0.6 that corresponds to 173 in  $3/5$  fixed point arithmetic for indoor scene. Results for outdoor scene were obtained also with threshold value of 0.6

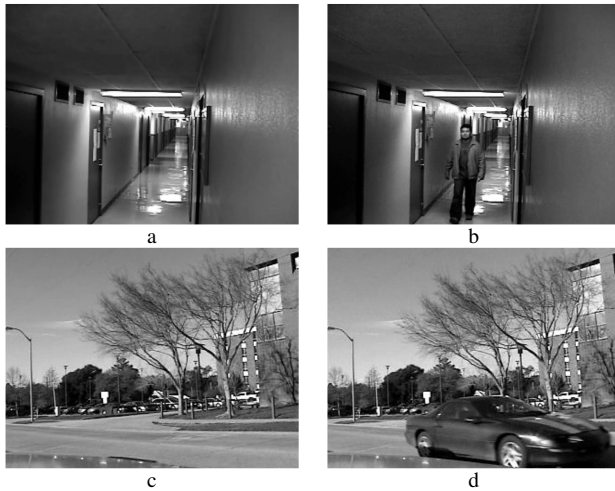


Figure 4. Test images for indoor and outdoor scene.

Most of the area and power consumption is occupied by the processing unit. The adder tree is fully asynchronous and its maximum delay of critical path is 12 ns. The PE is synchronous with four stages of asynchronous logic. One stage's maximum delay of critical path is 27 ns. The total power consumption of the system is 121 mW. The total area of the system in LUT is 2312 (7247 slices).

#### V. CONCLUSION

A background subtraction architecture using the Wronskian Change Detector algorithm has been presented for VLSI realization in wireless visual applications. The proposed architecture consists of three units: processing, memory and controller. The processing unit is composed by pipeline processing element that performs the basic operation. Partial results are stored and used on the adder tree to obtain final results. Memory unit consists of two buffers, one stored the previous frame (Frame Buffer) and the other stored partial results and output. The architecture is capable of computing Wronskian, Conjugate Wronskian and both. The maximum frame rate is 15 fps. The power dissipated by the whole system is 121 mW.



Figure 5. Results obtained using Simulink. a) Threshold value 0.6 for indoor scene, b) Threshold value 0.6 for outdoor scene.

#### ACKNOWLEDGMENT

The authors acknowledge the support of the U.S. Department of Energy (DoE), EETAPP program DE97ER12220, the Governor's Information Technology Initiative, the DoE award DE-FGO2-04ER46136 and the Louisiana Board of Regents contract DOE/LEQSF (2004-07)-ULL, and the support of National Science Foundation NSF, INF 6-001-006.

#### REFERENCES

- [1] R. Aguilar-Ponce, A. Kumar, J.L. Tecpanecatl-Xihuitl and M. Bayoumi, "An Architecture for Automated Scene Understanding", *IEEE Inter. Workshop on Comp. Arc. for Mach. Perception*, Terrasini - Palermo, July 4-6 2005
- [2] L. Bruzzone and D. F. Prieto, "An Adaptive Semiparametric and context-Based Approach to Unsupervised Change Detection in Multitemporal Remote-Sensing Images," *IEEE Trans. Image Processing*, vol. 11, no. 4, Apr. 2002, pp.452-466.
- [3] L. Bedat, O. Deforges, T. Landais and J.L. Corre, "Low bit-rate codec based on LAR method for video surveillance via internet", *Europ. Sig. Proc. Conf. EUSIPCO*, Sep 2000, Finland
- [4] W. Wolf, B. Ozer, T. Lv, "Smart Camera as Embedded Systems", *Computer*, Vol. 35, No. 9 Sept 2002, pp.48-53
- [5] E. Durucan and T. Ebrahimi, "Change Detection and Background Extraction by Linear Algebra", *Proc. IEEE*, vol. 89, no. 10, Oct. 2001, pp. 1368-1381.
- [6] I.S. Abu-Khater, A. Bellaour and M.I. Elmasry, "Circuit Techniques for CMOS Low-Power High-Performance Multipliers", *IEEE Journal of Solid-State Circuits*, Vol 31, Issue 10, Oct. 1996, pp. 1535-1546.
- [7] <http://tima-cmp.imag.fr/~guyot/Cours/Oparithm/english/Divise.htm>