International Journal of Oceans and Oceanography
ISSN 0973-2667 Volume 10, Number 1 (2016), pp. 73-80
© Research India Publications
http://www.ripublication.com

Hardware Co-simulation of Underwater Moving Object Detection using Xilinx System Generator

Alex Raj S. M. and Supriya M. H.
Department of Electronics,
Cochin University of Science & Technology,
Kochi, Kerala, India.

Abstract

In this paper, implementation of moving object detection in field programmable gate array (FPGA) is presented. The proposed system monitors movement by animals, humans or vehicles across a desired area. The system can be used for automatic under water vision system for monitoring moving objects to avoid potential human errors. FPGA implementations are faster than digital signal processors (DSP) and general purpose processors (GPP) for algorithms which exploits large amount of parallelism. Hardware co simulation was carried out using Xilinx system generator tool. Proposed algorithm makes use of differential of consecutive image frames after edge detection and then applying thresholds to obtain the result. Experimental results shows that the hardware co-simulation was successful with virtex 4 Xilinx FPGA and detects moving objects from a video with stationary background.

Keywords: FPGA, image processing, moving object, under water.

INTRODUCTION

Video surveillance has long been in use to monitor security sensitive areas such as banks, department stores, highways, crowded public places and borders. The advance in computing power, availability of large-capacity storage devices and high speed network infrastructure paved the way for cheaper, multi sensor video surveillance systems. Traditionally, the video outputs are processed online by human operators and are usually saved to storage devices for later use only after a forensic event. The increase in the number of cameras in ordinary surveillance systems overloaded both the human operators and the storage devices with high volumes of data and made it infeasible to ensure proper monitoring of sensitive areas for long times. Assisting the human operators with identification of important events in video by the use of video surveillance systems has become a critical requirement. The making of such video surveillance systems requires fast, reliable and robust algorithms for moving object detection, classification, tracking etc. [1].

Automation of underwater image or video processing is highly appreciated to avoid the necessity of human operators in real time opeartions. Fast and reliable real time systems for under water operations can be achieved by implementing those highly complicated parallel computations in an FPGA. Underwater detection and identification is a difficult and frustrating task as the identification must be carried out by visual means in waters that are by visual means that are generally turbid [2]. Fast and reliable real time systems for under water operations can be achieved by implementing those highly complicated parallel computations in an FPGA.

FPGAs are widely employed as co-processors in personal computers such as GPUs or as accelerators in specific purpose devices as high-capacity network systems [3] or high-performance computing [4]. Nowadays, it is possible to embed full systems on a single FPGA. Regarding Computer Vision, FPGAs are widely used both in industry and research. They offer a high degree of flexibility and performance to handle many different applications. Most compute-intensive algorithms were migrated to FPGAs: stereo vision [5], geometric algebra [6], optical flow [7], object recognition [8] or video surveillance [9, 10] to name some examples.

Most of the above FPGA-based implementations are typically programmed with low-level hardware description languages (HDL). FPGA implementations of Parallel multidimensional filtering algorithms have been traditionally programmed utilizing hardware description languages (HDLs), like VHDL and Verilog. These implementation methodologies require thousands of code lines which must be manually coded, debugged, verified, refined and re-entered line-by-line. Currently, FPGAs become a multimillions gates open chip to be configured for any efficient implementation of parallel multi-dimensional algorithms which demands highly computational complex operations at maximum sampling frequency. These requirements go beyond hand code HDLs programming methodologies [11].

An FPGA is superior, over a GPU, for algorithms requiring large numbers of regular memory accesses, while the GPU is superior for algorithms with variable data reuse. In the presence of data dependence, the implementation of a customized data path in an FPGA exceeds GPU performance by up to eight times [12].

The FPGA implementation is structurally based on the Xilinx System Generator (XSG) development tool of the ISE development suite, to detect moving objects targeting Virtex-4 FPGA board.

MOVING OBJECT DETECTION ALGORITHM

Different moving object detections schemes like adaptive back ground subtraction scheme [13], temporal differencing [14], adaptive background mixture models[15] etc. In adaptive background subtraction method, a reference background is ini-tialized at the start of the system with the first few frames of video and updated to adapt to short and long term dynamic scene changes during the operational period. At each new frame, foreground pixels are detected by subtracting the intensity values from the background and filtering the absolute value of the differences with a dynamic threshold per pixel. Background subtraction is particularly a commonly used technique for motion segmentation in static scenes [16]. It attempts to detect moving regions by subtracting the current image pixel-by-pixel from a reference background image that is created by averaging images over time in an initialization period. Temporal differencing attempts to detect moving regions by making use of the pixel-by-pixel difference of consecutive frames (two or three) in a video sequence. This method is highly adaptive to dynamic scene changes.

The advantage of frame difference method is its small calculation, and the disadvantage is that it is sensitive to the noise. If the objects do not move but the brightness of the background changes, the results of frame difference methods may be not accurate enough. Since the edge has no relation with the brightness, moving edge method can overcome the disadvantage of frame difference method. First of all, it detects the edges of each two continuous frames by Sobel edge detector and gets the difference between the two edge images. And then, it decides if they are moving areas by comparing the number of non-zero pixels to a threshold.

In this work, we have adopted edge difference method for detecting moving object. In [17] Heikkila and Silven uses the simple version of temporal differential scheme where a pixel at location (x, y) in the current image It is marked as foreground if

$$|It(x, y) - Bt(x, y)| > t$$

is satisfied where t is a predefined threshold. The background Bt is updated with the current pixel value. In this work, pixel value is taken after edge detection. The size of threshold decides the motion detection sensitivity. Low threshold cannot effectively inhibit the image noise, too high threshold will inhibit the useful change in the image.

HARDWARE CO-SIMULATION

System Generator is a DSP design tool from Xilinx that enables the use of the Mathworks model-based design environment Simulink for FPGA design. It is a system-level modeling tool in which designs are captured in the DSP friendly Simulink modeling environment using a Xilinx specific Blockset. System Generator provides many features such as System Resource Estimation to take full advantage of the FPGA resources, Hardware Co-Simulation and accelerated simulation through hardware in the loop co-simulation which give many orders of simulation performance increase. It also provides a system integration platform for the design of DSP FPGAs that allows the RTL, Simulink, MATLAB and C/C++ components of a DSP system to come together in a single simulation and implementation environment.

Hardware Co-Simulation (HWCOSIM) is a feature in System Generator (SysGen) that allows users to run the full or part of a SysGen design on the FPGA and increase the simulation speed dramatically. SysGen already includes HWCOSIM plugins for commonly used DSP demo boards. Users can easily use the SysGen Board

Description Builder (SBDBuilder) to create new HWCOSIM plugins for unsupported boards or unsupported features (for example non-memory mapped or NMM ports) on existing boards

Hardware co simulation was carried out by selecting Xilinx Virtex 4, XC4fx668-10 board. Combining Advanced Silicon Modular Block architecture with a wide variety of flexible features, the Virtex-4 family from Xilinx enhances programmable logic design capabilities, making it a powerful alternative to ASIC technology. Virtex-4 FPGAs comprise three platform families, LX, FX, and SX offering multiple feature choices and combinations to address all complex applications. The wide array of Virtex-4 FPGA hard-IP core blocks includes the PowerPC processors (with a new APU interface), tri-mode Ethernet MACs, 622 Mb/s to 6. 5 Gb/s serial transceivers, dedicated DSP slices, high-speed clock management circuitry, and source-synchronous interface blocks.

IMPLEMENTATION

The data to be entered to the FPGA through system generator is one dimensional data and the two dimensional image has to be converted to one dimensional data. In a similar fashion, the one dimensional output from the system generator is to be converted back to two dimensional data for proper viewing. One dimensional data is properly stored in a shared RAM which works in a first in first out fashion. Then using a 5x5 Sobel mask, edge detection was carried out and is properly delayed so as to subtract from the current frame pixel. Matlab function block (Mcode) is programmed so as to separate pixel values beyond a threshold value.

The above process occurs simultaneously in the R, G, B colour codes. The output from the MCodes are then moved to a three input MCode where the perfect detection of the video occurs, ie, the object in motion appears as white and the background as black. The ouput of three input Mcode, which is one-dimensional, is then converted back to three-dimensional Thus the target movement can be detected. The movement can be tracked by comparing the pixel values of this video to the original video. The regions where they differ can be tracked and bound. Median filtering is used to remove the noises in the video frames. Blob analysis is used to track the detected video.

The proposed wok is carried as shown in the following blocks. Threshold setting and comparison were carried out using Mcode block of system generator. Figure 4 shows HWCOSIM implementation.

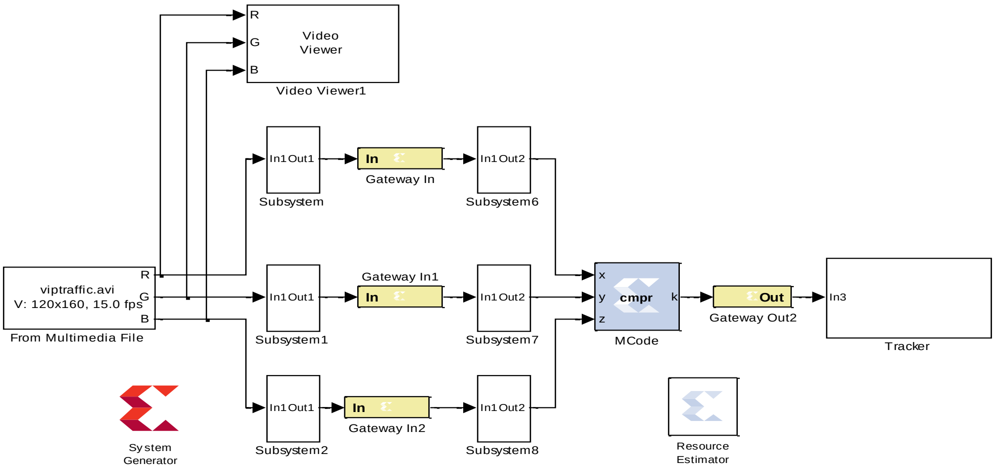

Figure 1: System generator model

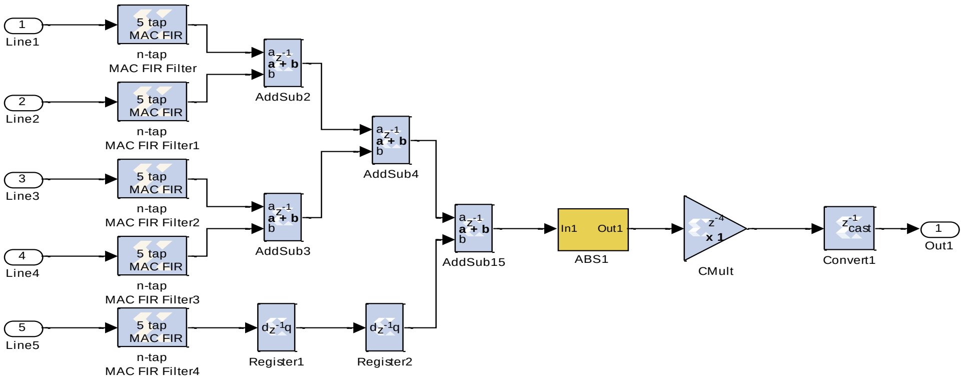

Figure 2: Edge detection filter model

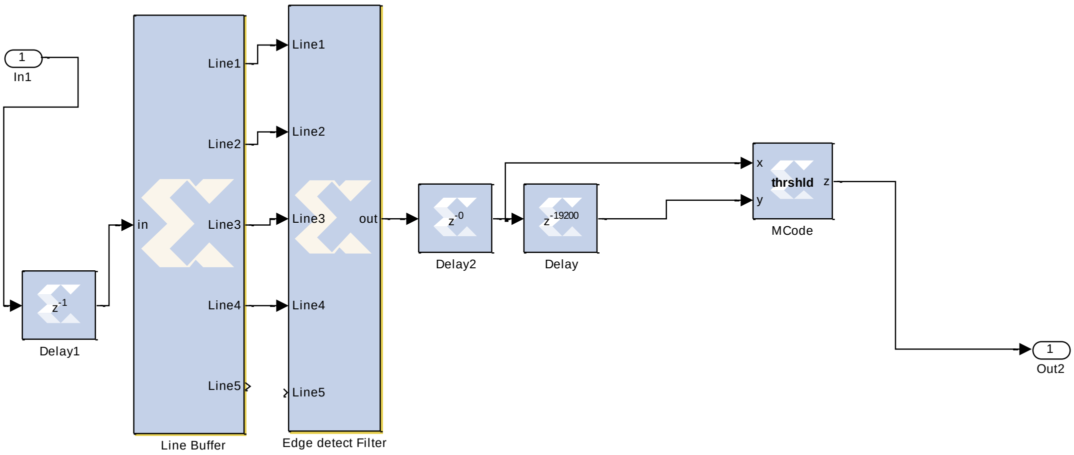

Figure 3: Edge differential and threshold model

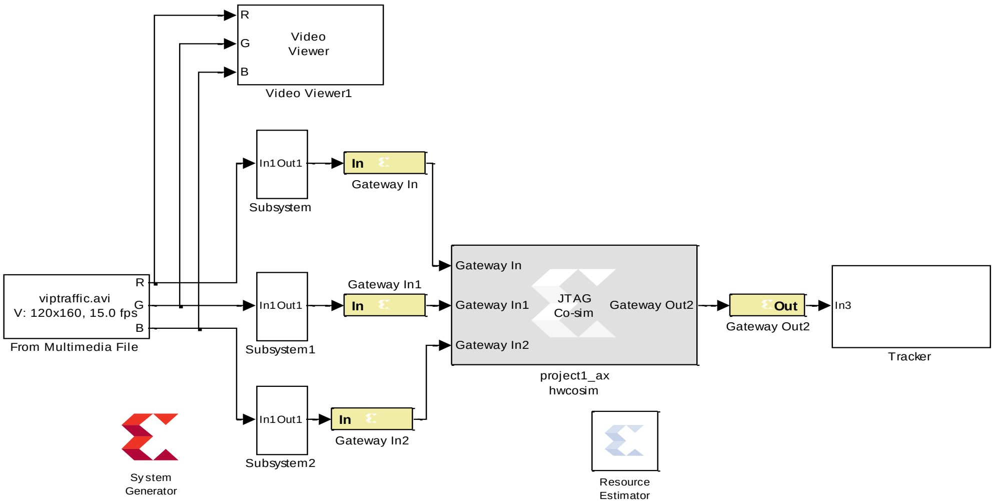

Figure 4: Hardware Co-Simulation model

RESULT

The goal of this paper is a new FPGA implementation method that provides fast FPGA prototyping for high performance computation of parallel image processing algorithms**.** The original video and detected video are shown in figure (5, 6) and figure (7, 8) respectively. The resource utilization is shown in Table 1. Threshold value plays an important role in detecting a moving object and it depends on the video also.

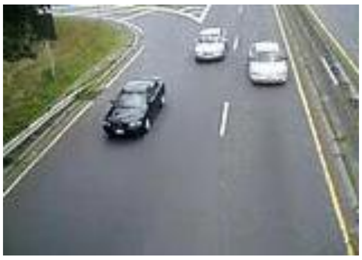

Figure 5: Original video

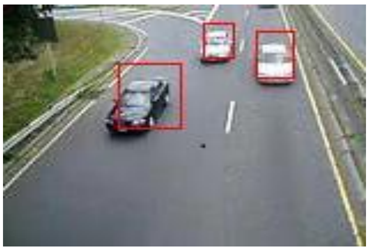

Figure 6: Object detection

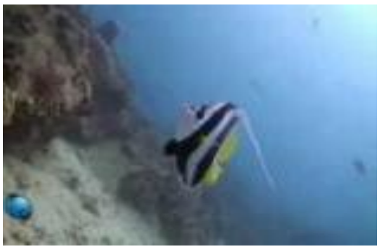

Figure 7: Original underwater video

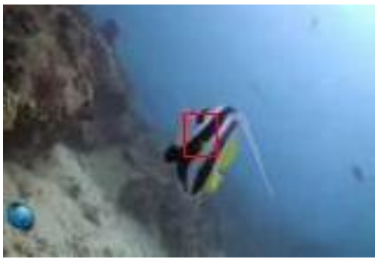

Figure 8: Object detection

Table 1: Resources Utililization

Resources	Number
Slices	14685
Flipflops	28818
BRAMs	27
Look up Tables	28347
IOBs	56
DSP48s	15

CONCLUSION

This paper demonstrated the implementation of moving object detection in an FPGA which requires high parallel computation. The implementation is behaviorally targeted to Xilinx Virtex 4 FPGA board. Underwater automatic video surveillance system can be achieved by reconfigurable hardware logic such as FPGA. Parallel processing algorithms such as image processing will take less computational time and less power in an FPGA compared to conventional DSPs.

REFERENCES

[1] A. Cavallaro and F. Ziliani. Image Analysis for Advanced Video Surveillance, chapter 2. 3, pages 57-67. Multimedia Video-Based Surveillance Systems. Kluwer Academic Publishers, Boston, 2000

[2] Alan Weidemann, George R. Fourier, L. Forand and P. Mathieu" In harbor underwater threat detection/identification using active imaging" Naval Research laboratory, Oceanographic Division, Canada, 2005

[3] Djordjevic, I., Arabaci, M. & Minkov, L. (2009). Next generation fec for high-capacity communication in optical transport networks, Journal of Lightwave Technology 27(16): 3518-3530.

[4] Craven, S. & Athanas, P. (2007). Examining the viability of fpga supercomputing, EURASIP Journal on Embedded systems 2007(1): 13-13

[5] Jin, S., Cho, J., Dai Pham, X., Lee, K., Park, S., Kim, M. & Jeon, J. (2010). Fpga design and implementation of a real-time stereo vision system, Circuits and Systems for Video Technology, IEEE Transactions on 20(1): 15-26.

[6] Franchini, S., Gentile, A., Sorbello, F., Vassallo, G. & Vitabile, S. (2009). An embedded, fpga-based computer graphics coprocessor with native geometric algebra support, Integration, the VLSI Journal 42(3): 346-355.

[7] Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors, IEEE transactions on pattern analysis and machine intelligence pp. 1615-1630

[8] Meng, H., Appiah, K., Hunter, A. & Dickinson, P. (2011). Fpga implementation of naïve bayes classifier for visual object recognition, IEEE Computer Vision and Pattern Recognition

[9] Nair, V., Laprise, P. & Clark, J. (2005). An fpga-based people detection system, EURASIP journal on applied signal

[10] Salem, M., Klaus, K., Winkler, F. & Meffert, B. Resolution mosaic-based smart camera for video surveillance, Distributed Smart Cameras,. ICDSC 2009. Third ACM/IEEE International Conference on, IEEE, pp. 1-7.

[11] B. Cope, P. Cheung, W. Luk, and L. Howes, “Performance Comparison of Graphics Processors to Reconfigurable Logic: A Case Study, ” IEEE Trans. Computers, vol. 59, no. 4, pp. 433-448, Apr. 2010.

[12] Hasan, Sami, Alex Yakovlev, and Said Boussakta. "Performance efficient FPGA implementation of parallel 2-D MRI image filtering algorithms using Xilinx system generator. " *CSNDSP, 2010 7th International Symposium on*. IEEE, 2010.

[13] R. T. Collins et al. A system for video surveillance and monitoring: VSAM final report. Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000.

[14] T. Brodsky et al. Visual Surveillance in Retail Stores and in the Home, chapter 4, pages 51-61. Video-Based Surveillance Systems. Kluwer Academic Publishers, Boston, 2002.

[15] C. Stauffer and W. Grimson. Adaptive background mixture models for realtime tracking. In Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, page 246252, 1999.

[16] M. McIvor. Background subtraction techniques. In Proc. of Image and Vision Computing, Auckland, New Zealand, 2000

[17] J. Heikkila and O. Silven. A real-time system for monitoring of cyclists and pedestrians. In Proc. of Second IEEE Workshop on Visual Surveillance, pages 74-81, Fort Collins, Colorado, June 1999.