Research Article

Real-Time Implementation of Change Detection for Automated Video Surveillance System

Sanjay Singh,¹ A. S. Mandal,¹ Chandra Shekhar,¹ and Anil Vohra²

¹ CSIR-Central Electronics Engineering Research Institute (CSIR-CEERI), Pilani, Rajasthan 333031, India

² Electronic Science Department, Kurukshetra University, Kurukshetra, Haryana 136119, India

Correspondence should be addressed to Sanjay Singh; sanjay.csirceeri@gmail.com

Received 10 April 2013; Accepted 9 May 2013

Academic Editors: M. Hopkinson, M. Rawski, and M. Terauchi

Copyright © 2013 Sanjay Singh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Change detection is one of the several important problems in the design of any automated video surveillance system. Appropriate selection of frames of significant changes can minimize the communication and processing overheads for such systems. This research presents the design of a VLSI architecture for change detection in a video sequence and its implementation on Virtex-IIPro FPGA platform. Clustering-based scheme is used for change detection. The proposed system is designed to meet the real-time requirements of video surveillance applications. It robustly detects the changes in a video stream in real time at 25 frames per second (fps) in gray scale CIF size video.

1. Introduction

Automated video surveillance systems are gaining importance due to their capability of performing automatic scene analysis. These systems can automatically select the frames of interest in a video stream by using scene change detection algorithms. The main challenge for designing these systems lies in their real-time performance requirements [1]. The software implementation of change detection algorithm running on general purpose processors could not achieve the frame rates required for real-time processing of a video stream for surveillance applications. Due to this reason, one needs to explore the different technologies possible for implementing real-time change detection system. One solution is to use special purpose digital signal processors, which are designed to execute some low level image processing operations very fast [2]. But the application like change detection uses a large range of low, medium, and high level operations. Therefore, this is not a good solution for achieving real-time performance. Alternative solutions are the design of application-specific integrated circuit (ASIC) or application-specific instruction set processor (ASIP) for change detection algorithm or use of field programmable logic devices. The main advantages of FPGAs (field programmable gate arrays)

over ASIC and ASIP are their ability to allow modification in the design in later stages of system development, architectural efficiency in terms of parallelism, shorter design time, cost efficiency, and real-time performance (high throughput). These features make FPGAs a suitable choice for implementing change detection in real time.

This paper presents the design of a VLSI architecture for real-time change detection in a video stream and its FPGA implementation. Different algorithms have been proposed in the literature for change detection. A review can be found in [3]. These algorithms use different parameters for change detection and give different responses and performances for the same input image. We have used a clustering-based scheme for change detection developed by Chutani and Chaudhury [4]. Their proposed approach significantly reduced the complexity and is much simpler as compared to other techniques. A dedicated VLSI architecture is developed for change detection using clustering-based scheme. Designed architecture is integrated with camera interface design and VGA display logic. Complete design is implemented on Xilinx Virtex-IIPro FPGA platform. Implemented system robustly detects the changes in real time in a video stream coming from camera.

FIGURE 1: Block diagram of change detection system.

FIGURE 2: Proposed implementation data flow diagram.

The rest of the paper is organized in the following way: Section 2 briefly describes the clustering-based change detection scheme. In Section 3, we show the proposed architecture, its implementation, and working. Section 4 discusses the synthesis results and evaluates the practical tests which have proved the successful implementation and effectiveness of our implementation. Finally, we conclude the paper in Section 5.

2. Change Detection Algorithm

The clustering-based change detection scheme used for change detection is briefly described in this section; for a more thorough description we refer to [4]. The main steps involved in the algorithm are given as follows.

- (1) Each incoming gray frame is partitioned into 4×4 blocks.
- (2) Each block is represented by a group of clusters, and each cluster contains centroid value and the frame number which updated the cluster last.
- (3) Initially, the cluster set of each block is initialized with a cluster having its centroid set to average color value of the corresponding block of the first frame.
- (4) Each block has 3 cluster nodes.
- (5) For every new frame, each block is compared with the corresponding cluster group. The difference is computed by taking Manhattan distance between average color value of the block and its centroid.

(6) If the difference is below a threshold value, it is considered as a matching cluster.

(7) For a matching cluster, the frame number and centroid associated with the cluster node of the corresponding block are updated.

For a given block, if no matching cluster is found, then a new cluster node is created by replacing the existing cluster node which has not been updated for the longest period of the time.

3. Proposed Architecture

This section discusses the design and implementation of proposed architecture for clustering-based change detection scheme. This research paper is the extension of our earlier published work in conference [5].

The main components of a complete change detection system are analog camera, video decoder card for analog to digital video conversion, a platform for processing or running change detection algorithm, and display device. Figure 1 shows the basic block diagram of the system.

In this implementation, captured video is digitized by Digilent VDEC1 video decoder board. Xilinx Virtex-IIPro Field programmable gate array (FPGA) board is used as processing platform. The change detected output is displayed on monitor via VGA connector on FPGA board. The video decoder board is connected to FPGA board through Hirose FX2 high-speed 100-pin connector. There are mainly five more blocks/modules running on FPGA in addition to change detection logic. These are camera interface logic, RGB to gray conversion, input memory buffer, output memory buffer, and VGA display controller as shown in Figure 2.

The camera interface module receives the digital signals from video decoder card and extracts the 24-bit RGB data and video timing signals. The data is coming from the camera row by row. The 24-bit RGB color data is converted into 8-bit gray data before storing it in input memory buffer. As clustering-based approach is based on the processing of 4×4 image blocks, the streaming video processing cannot be used. The four rows of image data are stored in input buffer memory before processing. The change detection architecture takes the inputs from buffer memory and processes the data. The change detection module output information is stored in output buffer memory. The VGA controller sends the change detected information to VGA port for display on monitor. It takes data from output memory buffer and timing information from camera interface module.

The proposed VLSI architecture for clustering-based change detection scheme is shown in Figure 3. The first module is *Block Centroid Computation Block*. It computes the average centroid for 4×4 image block by taking gray pixel data from input memory. This is done by adding 16 pixel values of current block and then dividing the sum by 16. Division by 16 is done by shift operation. Computed block centroid value is stored in pipeline register. As mentioned in the algorithm section, clustering-based scheme stores centroid and frame number information for each 4×4 image block; therefore we need to assign unique identity for each block.

FIGURE 3: Proposed VLSI architecture of clustering-based change detection.

FIGURE 4: Person movement with static background and detected changes.

FIGURE 5: Person movement with pseudo-stationary background and detected changes.

FIGURE 6: Multiple object movement and detected changes.

This is done by using row and column addresses generated by *Row and Column Address Generation* unit. This unit takes the control signals and video timing signals from camera interface module and generates the addresses. Generated addresses are stored in pipeline register. The frame number coming from camera interface module is also stored in pipeline register for its use in next pipeline stage.

There are two memory modules (*Node Centroid Memory* and *Node Frame No. Memory*) in change detection logic for storing background-related information of all cluster nodes for each block. *Node Centroid Memory* stores the centroid-related information of all nodes for each block. *Node Frame No. Memory* stores the frame number information of all nodes for each block.

In second stage, the difference of different node centroid values with current block centroid is computed. This is done by reading the centroid values for all cluster nodes of current 4×4 image block from node centroid memory and subtracting these values from current block centroid value. After this, the minimum difference value is selected and compared with a user defined threshold. This whole task is done by *Min* (minimum difference comparison) block. It generates the 1-bit *Select* signal which is high if the difference is greater than threshold and low if the difference is less than threshold.

For low value of *Select* signal, *Update Centroid and Frame No.* block becomes active. It computes the updated centroid value and frame number for corresponding node of the block. These computed values are used for updating the information in two memories.

For high value of *Select* signal, frame number values of all the nodes of current block are read from *Node Frame No. Memory*, and the node having minimum frame number is selected. The selected node's centroid is replaced by current block centroid value, and the frame number is replaced by current frame number. This task is performed by *Replace Centroid and Frame No.* module. Multiplexers are used for making the selection between values coming from *Update Centroid and Frame No.* block and *Replace Centroid and Frame No.* block. *Select* signal decides selection of centroid value and frame number value which will update the corresponding node information of current block in two memories. Also if *Select* signal is high, the corresponding block is marked as change block and this information is stored in output memory. These above-mentioned tasks are performed for each 4×4 block of input image.

4. Results

All design modules of the proposed architecture are coded in VHDL and simulated using ModelSim.

Synthesis is carried out using Xilinx ISE tool chain. The whole system is prototyped on Xilinx XUP Virtex-IIPro FPGA board (target device xc2vp30ff896-7). It utilizes 5% of the total available SLICES and 61% of the total available Block RAMs. The implementation is tested for scene change detection for CIF (352×288) resolution in gray scale. Three different situations are considered for testing. Figure 4 shows the situation where single object is entered in the scene and background is fixed. The detected changes are shown on the right side. The situation of pseudo-stationary background with single object movement with corresponding detected changes is shown in Figure 5. Multiple object movement example is shown in Figure 6. The implementation yields good results and works in real time (25 fps for CIF size video in gray scale). CIF size images are the most commonly used video formats for video surveillance cameras. Therefore, the implemented system can easily detect changes in real time for surveillance applications.

5. Conclusion

This paper has presented the design and implementation of a VLSI architecture for change detection, aiming to meet the real time constraints of video surveillance systems. Clustering-based scheme was used for detecting changes in a video stream coming from camera. The proposed system robustly detected the changes in real time. It can be used for selecting frames of interest based on scene change analysis. The future work will look at the extension of this work for standard PAL size color video for higher frame rate applications and optimizations in terms of memory requirements.

Acknowledgments

This work was carried out under a project supported by the Ministry of Communications and Information Technology (MCIT)/Department of Information Technology (DIT), Government of India. Sanjay Singh would like to thank Mr. Raj Singh, Group Leader, IC Design Group, for his constant guidance, suggestions, support, and encouragement.

References

- [1] C. T. Johnston, K. T. Gribbon, and D. G. Bailey, "Implementing image processing algorithms on FPGAs," in *Proceedings of the 11th Electronics New Zealand Conference (ENGCon '04)*, pp. 118–123, 2004.
- [2] H. Meng, N. E. Pears, and C. Bailey, "FPGA based video processing system for ubiquitous applications," in *Perspectives in Pervasive Computing*, pp. 57–63, October 2005.
- [3] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294–307, 2005.
- [4] E. R. Chutani and S. Chaudhury, "Video trans-coding in smart camera for ubiquitous multimedia environment," in *Proceedings of the International Symposium on Ubiquitous Multimedia Computing (UMC '08)*, pp. 185–189, Hobart, Australia, October 2008.
- [5] S. Singh, S. M. Dunga, R. Saini, A. S. Mandal, C. Shekhar, and A. Vohra, "Hardware accelerator design for tracking in smart camera," in *Proceedings of the International Conference on Graphic and Image Processing (ICGIP '11)*, October 2011.