High Frame Rate Real-time Scene Change Detection System

Sanjay Singh*[1], Ravi Saini[1], Sumeet Saurav[1], Pramod Tanwar[1], Kota S Raju[1], Anil K Saini[1], Santanu Chaudhury[1], Idaku Ishii[2]

[1]CSIR-Central Electronics Engineering Research Institute (CSIR-CEERI)
Pilani – 333031, Rajasthan, India.
[2]Robotics Laboratory, Hiroshima University
Hiroshima, 739-8527, Japan
*sanjay@ceeri.res.in

Abstract- Scene change detection, one of the fundamental and most important problems of computer vision, plays a very important role in the realization of a complete industrial vision system as well as automated video surveillance system - for automatic scene analysis, monitoring, and generation of alerts based on relevant changes in a video stream. Therefore, in addition to being accurate and robust, a successful scene change detection system must also be of very high frame rate in order to detect scene changes which goes off within a glimpse of the eye and often goes unnoticeable by the conventional frame rate cameras. Keeping this high frame rate processing as main focus, a very high frame rate real-time scene change detection system is developed by leveraging VLSI design to achieve high performance. This is accomplished by proposing, designing, and implementing an area-efficient scene change detection VLSI architecture on FPGA-based IDP Express platform. The complete real-time scene change detection system is capable of processing 2000 frames per second for 512x512 video resolution and the developed prototype is tested for the same. The proposed and implemented system architecture is adaptable and scalable for different video resolutions and frame rates.

Keywords: High Speed Scene Change Detection; VLSI Architecture; FPGA Implementation; Automated Video Surveillance System

1 Introduction

The long-term monitoring of high-speed phenomena using high frame rate cameras requires a large amount of storage space and high communication bandwidth over a network. Real-time high frame rate scene change detection allows more efficient hard disk storage by only archiving video frames where actual change in scene has occurred and also reduces the communication and further processing overheads in a remote video surveillance scenario by selecting the frames of relevant scene changes. However, the most of the existing implementations of scene change detection algo-

rithms are done for conventional video camera (25 fps to 30 fps), and, therefore, these are not capable of automatically detecting the high speed scene change phenomena.

In this work, we have addressed this problem by developing a real-time very high frame rate system for automatic scene change detection in live incoming video streams. The high performance was achieved by leveraging VLSI design based approach. A dedicated VLSI architecture has been proposed and designed for computationally efficient clustering based scene change detection scheme [1]. The designed architecture is coded in VHDL, simulated using ModelSim, and synthesized using Xilinx ISE 10.1 design tool chain. The architecture has been integrated with real-time video input / output interfaces. The complete final design has been implemented on FPGA-based IDP Express platform. The implemented scene change detection system is capable of processing 2000 frames per second for 512x512 resolution in live incoming video streams. It has been tested for different real-world scenarios for scene change detection and the system shows good results.

The rest of the paper is organized as follows: in the next section, we present a literature review of existing real-time implementations of scene change detection schemes. A brief of clustering based scene change detection scheme is presented in section three. The details of complete scene change detection system and associated VLSI architecture are presented in section four. The synthesis results and scene change detection results are presented in section five. Finally, we conclude this paper with a short summary.

2 Literature Review

The importance of scene change detection for designing industrial vision systems and automated video surveillance systems can be gauged from the availability of a large number of robust and complex algorithms and their implementations that have been developed to-date, and the even larger number of articles that have been published on this topic so far. The simplest approach to change detection is the frame differencing method in which change detection can be achieved by finding the difference of the pixels between two adjacent frames. If the difference is higher than a threshold, the pixel is identified as foreground otherwise background. The threshold is chosen empirically. Different methods and criteria for choosing the threshold have been surveyed and their comparative results have been reported in the literature [2]-[4]. The simplicity of frame differencing based approaches comes at the cost of change detection quality. For a chosen threshold, simple differencing based approaches are unlikely to outperform the more advanced algorithms proposed for real-world surveillance applications. A comprehensive description and comparative analysis of these methods has been presented by Radke et al. [5].

The practical real-world video surveillance applications demand a continuous updating of the background frame to incorporate any permanent scene change i.e. if a pixel has remained stationary for a sufficient number of frames, it must be copied into the background frame such as , for example, light intensity changes in day time must be a part of the background. For this purpose, several researchers [6]-[8] have de-

scribed adaptive background subtraction techniques for change detection. They have used single Gaussian Density Function to model the background. These algorithms succeed in learning and refining the single background model. They are capable of handling illumination changes in a scene and are well suited for stationary background scenarios.

Due to pseudo-stationary nature of the background in real-world scenes, assuming that background is perfectly stationary for surveillance applications is a serious flaw. For example, in a real-world video scene, there may be swaying branches of trees, moving tree leaves in windows of rooms, moving clouds, the ripples of water on a lake, or moving fan in the room. These are small repetitive changes (typically not important) and so should be incorporated into background. The single background model based approaches mentioned above are incapable of correctly modeling such pseudo-stationary backgrounds. Stauffer and Grimson [9] recognized that these kinds of pseudo-stationary backgrounds are inherently multi-model and hence they developed the technique of an Adaptive Background Mixture Models, which models each pixel by a mixture of Gaussians. However, maintaining these mixtures for every pixel is an enormous computational burden and results in low frame rates when compared to previous approaches. Butler et al. [10] proposed a new approach, similar to that of Stauffer and Grimson [9], but with a reduced computational complexity. The processing, in this approach, is performed on YC_rC_b video data format, but it still requires many floating point computations and needs large amounts of memory for storing background models. In order to address this problem of reducing the computational complexity, Chutani and Chaudhury [1] proposed a block-based clustering scheme with a very low complexity for change detection. On one hand this scheme is robust enough for handling pseudo-stationary nature of background, and on the other it significantly lowers the computational complexity and is well suited for designing standalone systems for real-time applications. For this reason we have selected the clustering based change detection scheme for designing the real-time standalone high frame rate change detection system which features real-time processing and is capable of discarding irrelevant changes using adaptive background model.

3 Clustering based Change Detection Algorithm

Clustering based change detection [1] uses a block-based similarity computation scheme. At the heart of this scheme is a very low complexity method for maintaining some limited but important information about the history of each pixel. To start with, each incoming video frame is partitioned into 4x4 pixel blocks. Each 4x4 pixel block is modeled by a group of four clusters where each cluster consists of a block centroid (in RGB) and a frame number which updated the cluster most recently. Optionally, for each block there may be a change flag field. The group of four clusters is necessary to correctly model the pseudo-stationary background, as a single cluster is incapable of modeling multiple modes that can be present in pseudo-stationary backgrounds. The group size is selected as four because it has been reported by Chutani and Chaudhury [1] that four clusters per group yield a good balance between accuracy

and computational complexity. The basic computational scheme is shown in Figure 2. The sequence of steps for change detection using clustering-based scheme is Block Centroid Computation, Cluster Group Initialization, Cluster Matching, Cluster Update, Cluster Replace, and classifications.

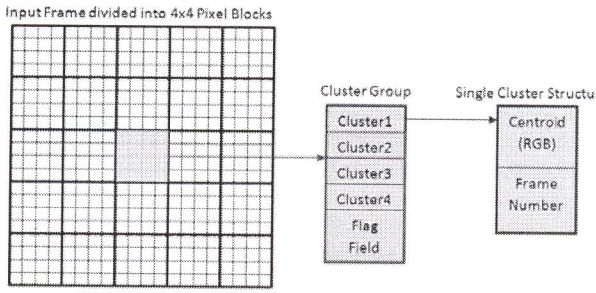

Fig. 1. Clustering-based Change Detection Scheme.

The above mentioned clustering-based change detection scheme has been implemented by us in C/C++ programming language. For running the code, a Dell Precision T3400 workstation (with Windows XP operating system, quad-core Intel® Core™2 Duo Processor with 2.93 GHz Operating Frequency, and 4GB RAM) was used. The Open Computer Vision (OpenCV) libraries have been used in the code for reading video streams (either stored or coming from camera) and displaying change detection results. The frame rate of this software-based implementation was found to be 6 frames per second (fps) for 512x512 resolution videos.

4 High Frame Rate Scene Change Detection System

In order to achieve high frame rates, as required in many industrial and surveillance applications, we have developed a dedicated hardware architecture for clustering-based scene change detection scheme and its implementation as a prototype system using the FPGA-based IDP Express Platform [11].

A simplified conceptual block diagram of the proposed and developed FPGA-based scene change detection system is shown in Figure 2 to illustrate the data flow within the system. The main components of a complete high frame rate scene change detection system are high frame rate *Camera* and *IDP Express Platform*. The image data captured by high speed camera is transferred to *IDP Express Platform*. The three main components of *IDP Express Platform* are: *Serial to Parallel Conversion*, *Virtex-5(XC5VLX50T) Xilinx FPGA*, and *Spartan-3 (XC3S5000) Xilinx FPGA*. The serial data received from camera is converted to parallel data by *Serial to Parallel Conversion* module. *Virtex-5(XC5VLX50T) Xilinx FPGA* is used for implementations of high speed video data interface and PCI bus interface. *Spartan-3 (XC3S5000) Xilinx FPGA* is used for implementations of high frame rate scene change detection VLSI architecture and interface FIFOs. *Data Format Conversion* module of *Virtex-5(XC5VLX50T)*

Xilinx FPGA provides eight pixels in parallel. The *Scene Change Detection* module, implemented on *Spartan-3 (XC3S5000) Xilinx FPGA*, receives these eight pixels, stores them, and process. The scene change detected data is given back to *Interface Logic* module implemented on *Virtex-5(XC5VLX50T) Xilinx FPGA*. Finally the data is transferred to *PCI Express Bus* of the system through *PCI Express Interface* module. Video timing signals (Pixel Clock, Hsync, Vsync, and Blank) are used for synchronization among different modules of the complete system.

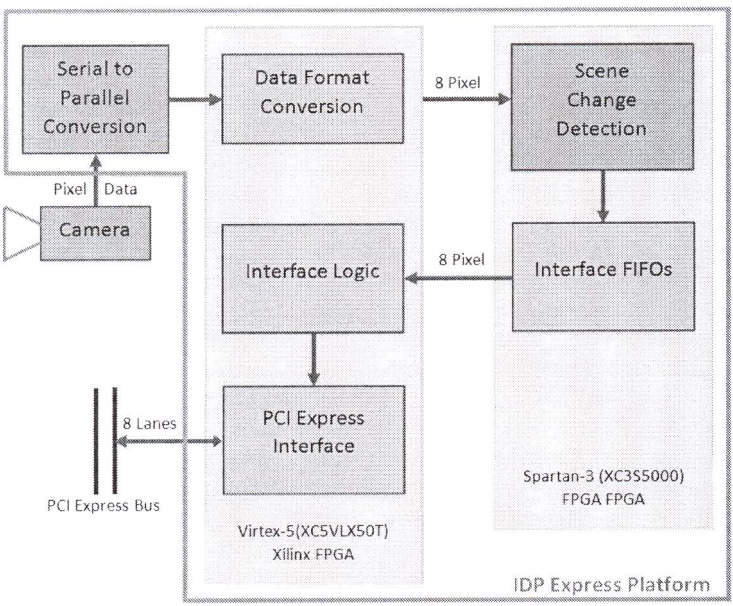

Fig. 2. Dataflow Diagram of the Developed Scene Change Detection System.

The detailed proposed VLSI architecture of *Scene Change Detection* module is shown in Figure 3. Pixel Data arrives from the *Data Format Conversion* module (implemented on *Virtex-5(XC5VLX50T) Xilinx FPGA*) row by row. As scene change detection scheme is based on the processing of 4x4 image blocks, therefore, streaming video processing cannot be used for clustering based scene change detection scheme. For this reason, the four rows of image data are buffered in input memory (INPUT MEM) before processing begins. The output from INPUT MEM is four pixel data coming out in parallel. BLCENT COMPUTATION UNIT computes the average centroid for 4x4 image block by taking pixel data from the input memory (four clock cycles are required for reading 16 pixels from the input memory buffer). This computation is done by adding 16 pixel values of current block and then dividing the sum by 16. The read address, write address, and write enable signals for input memory are generated by the corresponding INPUT-MEM RADD-WRADD WEN module.

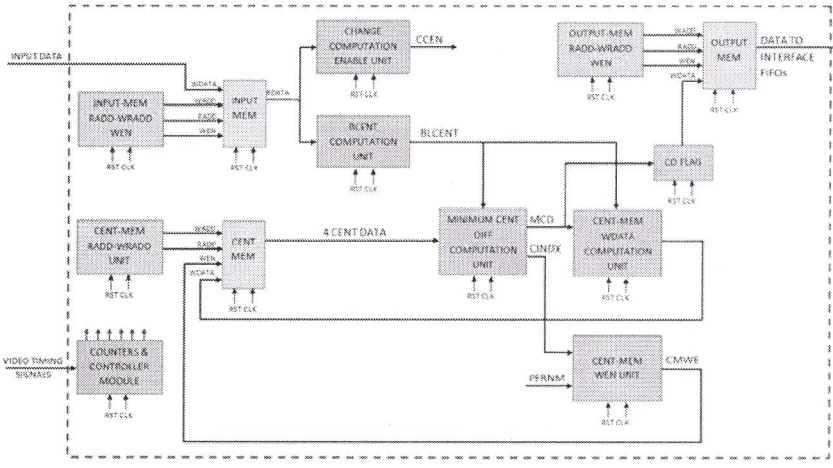

Fig. 3. Proposed VLSI Architecture for the Scene Change Detection Algorithm.

The change computation enable (CCEN) signal is generated by CHANGE COMPUTATION ENABLE UNIT. This signal is used as the enable signal in different modules of the designed scene change detection architecture.

As mentioned in algorithm section, clustering-based scheme stores associated centroid information for each 4x4 image block. It, therefore, requires the assignment of a unique identity (or address) to each block. This is done by using row and column counters generated by COUNTERS & CONTROLLER MODULE. This unit takes the video timing signals from camera interface module and generates the different counter values (row counter, column counter, frame number counter) and control signals required for the proper functioning of complete system.

For storing pixel related information in the clustering-based algorithm, a centroid memory (CENT MEM) is used. Each Centroid Memory location contains four Centroid values (corresponding to four clusters) which contain the background color and intensity related information. The read address and write address signals for CENT-MEM are generated by the corresponding CENT-MEM RADD-WRADD UNIT.

During initialization phase, control signals are generated in such a way that the four clusters are initialized. For all subsequent frames, the generated control signals enable the change detection process. After initialization, the matching cluster is searched within the cluster group of four clusters. For this purpose, first difference between cluster Centroid value (CENT DATA) and incoming current block Centroid value (BLCENT) is computed for all four clusters by reading the cluster Centroid values (4 CENT DATA) from CENT MEM corresponding to current 4x4 image block and taking absolute sum of differences with current block Centroid value (BLCENT). From the four difference values, minimum Centroid difference value is selected. This complete task is carried out by the MINIMUM CENT DIFF COMPUTATION UNIT. It outputs MCD (minimum centroid difference value) and CINDX (Centroid Index).

CINDX gives the cluster number corresponding to MCD (minimum centroid difference value).

The MCD and BLCENT values are used by CENT-MEM WDATA COMPUTATION UNIT to compute the write data for Centroid Memory (CENT MEM). MCD is compared with a user defined threshold. For MCD less than or equal to the threshold (i.e. matching cluster is found), the write data for CENT MEM is the average value of current block Centroid value (BLCENT) and matching cluster Centroid value (matching cluster number is given be CINDX). For MCD greater than threshold (i.e. no matching cluster is found), the write data for CENT MEM is current block Centroid value (BLCENT) and in this case the cluster number for which value is replaced is determined by CENT-MEM WEN UNIT which generates the write enable signal for CENT MEM.

The CINDX and FINDX values are used by CENT-MEM FRNM-MEM WEN UNIT for generating the write enable signals for CENT MEM and FRNM MEM. The write enable signals help for selecting the cluster for which the centroid value and the frame number value is to be updated or replaced.

The change detection flag generation module (CD FLAG) takes MCD as input and compares it with a user defined threshold. A 1-bit Flag signal is generated which is low if difference is less than the threshold (i.e. current block matches with background model and therefore, no change is detected) and high if the difference is greater than the threshold (i.e. current block is change detected block). This change information data of 4x4 pixel block is written to the output memory (OUTPUT MEM) and corresponding addresses for this memory are generated by OUTPUT-MEM RADD-WRADD WEN module. The processed output data is stored in output memory (OUTPUT MEM) for synchronization purpose before sending it to *Interface FIFOs* implemented on *Spartan-3 (XC3S5000) Xilinx FPGA*. This is because the output data of *Scene Change Detection* architecture is for 4x4 pixel block while the data is sent to *Interface Logic* (implemented on *Virtex-5(XC5VLX50T) Xilinx FPGA*) is row by row. Finally, the scene change detected data is read from this output buffer memory (OUTPUT MEM) and sent to *PCI Express Bus* through *Interface Logic* module and *PCI Express Interface* (*Virtex-5(XC5VLX50T) Xilinx FPGA*) module.

All design modules of proposed architecture for clustering based change detection scheme have been coded in VHDL, simulated using ModelSim, and synthesized using Xilinx ISE tool chain. Frame rate achieved through the hardware implementation is far higher than that required for real-time processing and is far higher than that obtained for software implementation on a workstation.

5 Synthesis Results

All the modules of the proposed architecture for clustering based change detection scheme were coded in VHDL and simulated using ModelSim. A top level design module was created which invoked all the designed modules of scene change detection architecture, input camera interface, and PCI express interface. A User Constraint File (UCF) was created to map the input/output ports of the design on actual pins of

the FPGA. This top level design was synthesized using Xilinx ISE tool chain. The resulting configuration file was loaded into user FPGA on IDP Express platform. Thus a complete prototype system for real-time high frame rate automatic scene change detection system was developed and is shown in Figure 4. The FPGA resource utilization results for post-place and route are shown in Table 1.

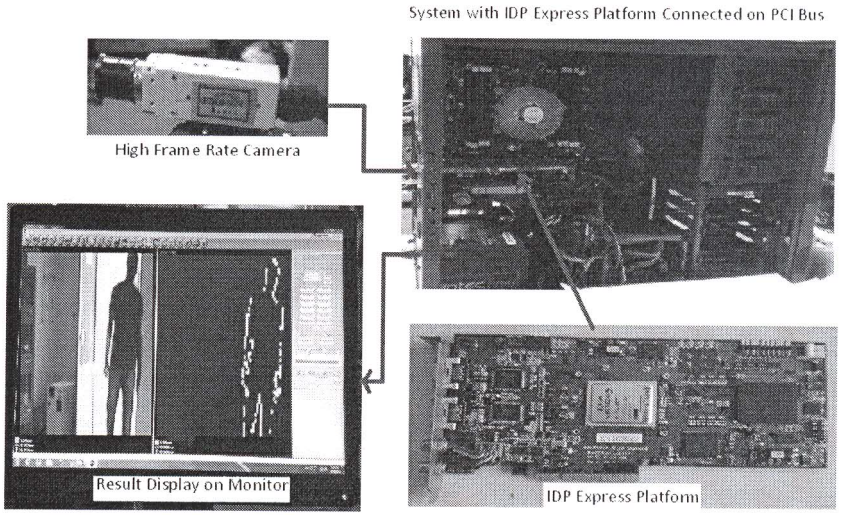

Fig. 4. Complete High Frame Rate Scene Change Detection System Setup.

Table 1. FPGA Resource Utilization by Complete High Frame Rate Scene Change Detection System.

Resources	Resources Utilized	Available Resources	Percentage of Utilization
Slice Registers	1012	66560	1.52%
Slice LUTs	1242	81920	1.87%
Occupied Slices	1265	32280	3.92%
BRAMs	72	104	69.23%
Memory (Kb)	1296	1872	69.23%
IOs	193	633	30.49%

Performance of the proposed, designed, and implemented system is compared with some recently published scene change detection implementations. The performance comparison is shown in Table 2. Kristensen et al. [12] and Jiang et al. [13] have presented the design of a digital surveillance system running in real-time on an embedded platform. The change detection/segmentation unit of the circuit proposed

by these researchers is able to run at 83 MHz on Virtex-IIPro (xc2pro30-5ff1152) FPGA. The change detection/segmentation architectures presented by Genovese et al. [14] and Genovese and Napoli [15] were designed for OpenCV GMM algorithm and were implemented on Virtex5 (xc5vlx50-2ff1153) FPGA. The scene change / motion segmentation circuit proposed by Genovese and Napoli in [16] has been implemented on both Virtex6 (xc6vlx75t-3ff784) and Virtex5 (xc5vlx50-2ff1153) FPGAs and is an improved version of the work presented by them in [15] and [14]. The architecture proposed in this paper for scene change detection scheme outperforms existing implementations in terms of processing speed while occupying less FPGA resources.

Table 2. Performance Comparison with Existing Scene Change Detection Implementations.

Target FPGA Device	Implementation	Video Resolution	Frame Rate (fps)
IDP Express Platform Spartan3 (xc3s5000-4fg900)	Our Implemented Prototype System	512x512	2000
Virtex5 (xc5vlx50-2ff1153)	[16]	720x576	315
	[15]	720x576	121
	[14]	720x576	113
Virtex-IIPro (xc2pro30-5ff1152)	[13], [12]	720x576	200

The system architecture for high frame rate scene change detection, proposed, designed, and implemented in this paper is adaptable and scalable for different video sizes and frame rates.

6 Scene Change Detection Results

For robust and real-time testing of the implemented system, the system was run for different real-world scenarios (both indoor and outdoor) under different lightening conditions. Different real-world scenarios are broadly classified into two categories i.e. static background situations and pseudo-stationary background situations. Figure 5 shows examples of real-world situations of static background scenarios captured by the camera. In all the cases of Figure 5, the background is static and the moving objects are present in the scene. Change detected by our implementation in different frames is shown just below the respective images. It can be clearly seen that only moving objects have been detected by the implemented scene change detection system.

Scenarios of pseudo-stationary background with moving foreground objects are considered in Figure 6. Despite the moving fan and movements of leaves of trees in the background, only moving objects in the foreground are detected. Results of the tests show that the system is robust enough to detect only the relevant scene changes in a live video scene and eliminates the continuous unwanted movements in the background itself. All the above video frames are of 512x512 resolution. These frames and

their results are extracted from live video streams produced by the implemented system.

If a change threshold is not crossed in a video frame, the frame is rejected at this stage and if a change threshold is crossed in a video frame, the frame is selected for further processing. This filtering of frames of interest reduces the communication and processing overheads of an automated video surveillance system for remote video surveillance scenarios.

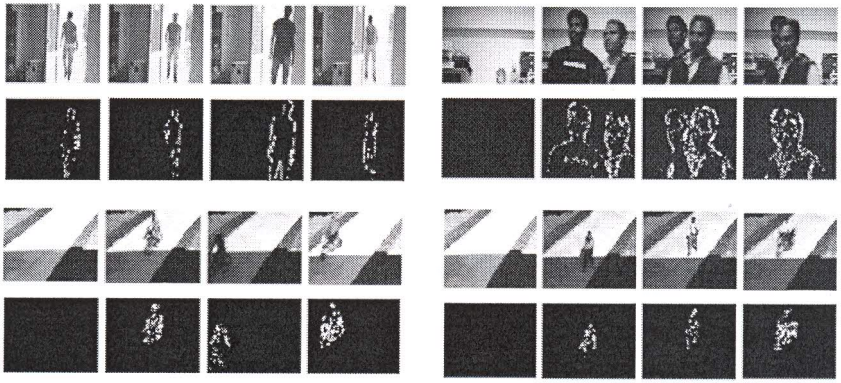

Figure 5: Moving Objects in Video Scene and Corresponding Change Detected Outputs for Static Background Scenarios.

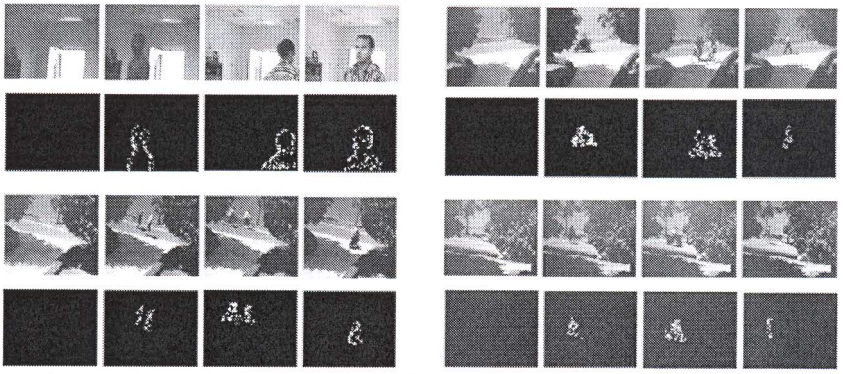

Figure 6: Moving Objects in Video Scene and Corresponding Change Detected Outputs for Pseudo-stationary Background Scenarios.

7 Conclusions

In this paper, we have proposed and described the design and implementation of a real-time high frame rate prototype system for scene change detection. To address the issue of high performance requirement, a dedicated VLSI architecture has been proposed, designed, and integrated with video interfaces. The complete system has been

prototyped on FPGA-based IDP Express board. The developed system prototype was tested for live incoming video streams and it robustly and automatically detects scene changes in real-time at 2000 fps for 512x512 resolution video streams. The implemented system can be effectively used as a standalone system for high frame rate industrial and surveillance applications (for detection of high-speed phenomena).

References

1. E.R. Chutani and S. Chaudhury, Video Trans-Coding in Smart Camera for Ubiquitous Multimedia Environment, In Proceedings: International Symposium on Ubiquitous Multimedia Computing, pp.185–189, 2008.
2. P.L. Rosin, Thresholding for Change Detection, In Proceedings: Sixth International Conference on Computer Vision, pp. 274-279, 1998.
3. P.L. Rosin and E. Ioannidis, Evaluation of Global Image Thresholding for Change Detection, Pattern Recognition Letters, Vol. 24, No. 14, pp. 2345-2356, 2003.
4. P.C. Smits and A. Annoni, Toward Specification-Driven Change Detection, IEEE Transaction on Geosciences and Remote Sensing, Vol. 38, No. 3, pp. 1484-1488, 2000.
5. R.J. Radke, S. Andra, O.A. Kofahi, and B. Roysam, Image Change Detection Algorithms: A Systematic Survey, IEEE Transactions on Image Processing, Vol. 14, No. 3, pp. 294-307, 2005.
6. A. Cavallaro and T. Ebrahimi, Video Object Extraction based on Adaptive Background and Statistical Change Detection, In Proceedings: SPIE Visual Communications and Image Processing, pp. 465-475, 2001.
7. S. Huwer and H. Niemann, Adaptive Change Detection for Real-Time Surveillance Applications, In Proceedings: Third IEEE International Workshop on Visual Surveillance, pp. 37-46, 2000.
8. T. Kanade, R.T. Collins, A.J. Lipton, P. Burt, and L. Wixson, Advances in Cooperative Multi-Sensor Video Surveillance, In Proceedings: DARPA Image Understanding Workshop, pp. 3-24, 1998.
9. C. Stauffer and W.E.L. Grimson, Learning Patterns of Activity using Real-Time Tracking, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 747-757, 2000.
10. D.E. Butler, V.M. Bove, and S. Sridharan, Real-Time Adaptive Foreground/Background Segmentation, EURASIP Journal on Applied Signal Processing, Vol. 2005, pp. 2292-2304, 2005.
11. IDP Express Platform (http://www.photonics.com/Product.aspx?PRID=46288).
12. F. Kristensen, H. Hedberg, H. Jiang, P. Nilsson, and V. Öwall, An Embedded Real-Time Surveillance System: Implementation and Evaluation, Journal of Signal Processing Systems, Vol. 52, No. 1, pp. 75-94, 2008.
13. H. Jiang, H. Ardö, and V. Öwall, A Hardware Architecture for Real-time Video Segmentation Utilizing Memory Reduction Techniques, IEEE Transaction on Circuits and Systems for Video Technology, Vol. 19, No. 2, pp. 226-236, 2009.
14. M. Genovese, E. Napoli, and N. Petra, OpenCV Compatible Real Time Processor for Background Foreground Identification, In Proceedings: International Conference on Microelectronics, pp 467-470, 2010.
15. M. Genovese and E. Napoli, FPGA-Based Architecture for Real Time Segmentation and Denoising of HD Video, Journal of Real Time Image Processing, Vol. 8, No. 4, pp. 389-401, 2013.

16. M. Genovese and E. Napoli, ASIC and FPGA Implementation of the Gaussian Mixture Model Algorithm for Real-time Segmentation of High Definition Video, IEEE Transactions on Very Large Scale Integration, Vol. 22, No. 3, pp. 537-547, 2014.