FPGA-based Smart Camera System for Real-time Automated Video Surveillance

Sanjay Singh*, Sumeet Saurav, Ravi Saini, Atanendu S. Mandal, Santanu Chaudhury

[1] CSIR-Central Electronics Engineering Research Institute (CSIR-CEERI)
Pilani – 333031, Rajasthan, India.
sanjay@ceeri.res.in

Abstract. Automated video surveillance is a rapidly evolving area and has been gaining importance in the research community in recent years due to its capabilities of performing more efficient and effective surveillance by employing smart cameras. In this article, we present the design and implementation of an FPGA-based smart camera system for automated video surveillance. The complete system is prototyped on Xilinx ML510 FPGA platform and meets the real-time requirements of video surveillance applications while aiming at FPGA resource reduction. The implemented smart camera system is capable of automatically performing real-time motion detection, real-time video history generation, real-time focused region extraction, real-time filtering of frames of interest, and real-time object tracking of identified target with automatic purposive camera movement. The system is designed to work in real-time for live color video streams of standard PAL (720x576) resolution, which is the most commonly used video resolution for current generation surveillance systems. The implemented smart camera system is also capable of processing HD resolution video streams in real-time.

Keywords: Smart Camera System, Automated Video Surveillance System, FPGA Implementation.

1 Introduction

The last decade saw an explosion of video surveillance activities in most cities and countries. Today, these systems are ubiquitous and can be found at airports, railway and bus terminals, banks and ATMs, departmental stores, offices, etc. In order for these systems to be effective, the cost and difficulty of deployment must be reduced. Continuous monitoring of multiple video streams by a human operator and manual browsing of thousands of video frames for crime scene and forensic analysis are neither reliable nor scalable [1]. This has generated enormous interest in research activities related to automation of these systems among the researchers of integrated systems and computer vision communities. The advantage of automated video surveillance systems over conventional closed-circuit television (CCTV) based surveillance systems lies in the fact that these are self-contained systems capable of performing automatic analysis of a scene by using smart cameras without intervention of human operators [2]. The employed smart cameras can intelligently detect relevant motion in a live video stream, decide what to store for further processing, what to communicate, and what to track. The smart cameras not only potentially cut the cost of human resources observing the output of the cameras but also reduce errors caused due to manual operation. However, crucial to these systems are the real-time requirements.

To meet the hard real-time requirements of smart camera systems, very different technologies and design methodologies have been used in literature. These range from use of General Purpose Processors (GPPs) or special purpose Digital Signal Processors (DSPs) or Graphics Processing Units (GPUs) to Application Specific Integrated Circuits (ASICs) or Applications Specific Instruction Set Processors (ASIPs) or even programmable logic devices like Field Programmable Gate Arrays (FPGAs).

Due to rapid advancements in fabrication technology (adoption of finer chip geometries down to 28 nm and higher levels of integrations), there has been stunning growth in the size, functionality, and performance of field programmable gate arrays (FPGAs) in recent years. The size and speed of current generation FPGAs are comparable to ASICs (though ASICs are typically faster, low power, and occupy less area as compared to FPGAs, given the same manufacturing technology). On the other hand, FPGAs limit the extensive design work required for ASICs, shorten the development cycle (results in reduced cost), and admit the possibility of performing algorithmic changes in later stages of system development as well. Furthermore, FPGA structure is able to exploit spatial and temporal parallelism inherent in image processing, computer vision, and pattern recognition applications. Even if one is following the ASIC development route, the functional verification on an FPGA before manufacturing an ASIC is always advised since many errors can be detected and corrected that way by running the FPGA emulation of the ASIC over extended periods and data sets.

With expanding resolutions and evolving algorithms of a smart camera system, there is a need for high performance while keeping the architecture flexible to allow quick and easy upgradability. FPGAs provide real-time performance that is hard to achieve with GPPs/DSPs, limit the extensive design work, time, and cost required for ASICs, and provide the possibility of performing algorithmic changes in later stages of system development. These features make FPGAs a suitable choice for implementing / prototyping smart camera systems.

Developing a complete FPGA-based smart camera system has many open unresolved design and research challenges such as designing efficient hardware architectures and memory controllers, implementing camera and display device interfaces for real-time video capturing and viewing, automatically controlling camera pan-tilt-zoom features in real-time, and so on. These challenges can be handled at different abstraction levels and by various techniques. In this paper, to be able to solve these issues, different area/memory efficient real-time hardware accelerators and input/output interfaces are designed, implemented, and integrated into a complete working system prototype. The overall goal of this paper is to present the complete designed and implemented FPGA-based smart camera system prototype which can cater to various real world surveillance applications like real-time motion detection, video history generation, focused region extraction, and object tracking with automatic purposive camera pan-tilt capabilities. The paper describes the implemented smart camera system level architecture, its functionality, synthesis results, and visual results.

2 Developed Smart Camera System

For developing the FPGA-based smart camera system, we have used the Xilinx ML510 (Virtex-5 FX130T) FPGA development platform. As there is no input video interface available with this board for connecting the video input, therefore, we have developed a custom video camera interface by designing the interfaces PCB for connecting the Digilent VDEC1 video decoder board with Xilinx ML510 (Virtex-5 FX130T) FPGA development platform through high speed input/output ports available with this FPGA platform. The reason for selecting this particular FPGA board and details of the physical camera interface design are out of the scope of this article and their details are available in [3]. The smart camera system described in this article is designed and implemented to meet the following requirements and functionalities:

- Video resolution ≥ Standard PAL (720x576) Size
- Frame rate ≥ 50 frames per second
- Real-time capturing of live video stream and result display
- Real-time motion detection
- Real-time focused region extraction
- Real-time filtering of frames of interest based on motion detection only in focused regions
- Real-time object tracking for detected targets
- Real-time object history generation
- Real-time automatic purposive camera movement (pan-tilt) to follow target object
- Standalone operation mode
- All proposed, designed, and implemented architectural modules deliver real-time performance and at the same time are area and memory efficient (implemented with limited resources on a single FPGA development platform).

A complete system level architecture of the functional decomposition of a smart camera system addressed, designed, and implemented in this article is depicted in Fig. 1. Starting at the input, the camera feeds the system with a real-time video stream (a sequence of images) of a scene through *Camera Interface* module which extracts the image pixel data from the incoming video stream. After receiving the pixel data from the *Camera Interface* module, the *Motion Detection* module detects the relevant motion in the incoming video stream. Parallelly, focused regions in the incoming video stream are extracted by the *Focused Region Extraction* module. The outputs of these two modules are given to the *Motion Detection in Focused Regions* module for finding motion in only focused areas in a scene. The output of this block can be used to generate the alarm signal for security personnel in case any relevant motion is detected in restricted areas. The output of *Motion Detection in Focused Regions* module is also used for automatic detection of a moving target in the incoming live video stream, and, the detected moving target is then tracked by *Object Tracking* module in subsequent frames. The moving target object for *Object Tracking* module can be selected either manually by moving the camera on the target object or automatically based on the output from *Motion Detection in Focused Regions* module. While in automatic moving target detection and tracking mode, the implemented system starts detecting motion in focused regions at power-on. The size of relevant motion detection region varies, depending on the size of moving object present in the scene. However, the implemented *Object Tracking* module is designed to track a target of maximum 100x100 pixel size. For this region, a control logic has been implemented to select the upper most part of size 100x100 pixels of moving object. After selecting the upper most part of size 100x100 pixels of the moving object, it draws the rectangle on selected region and passes the co-ordinate information to the *Object Tracking* module. In subsequent frames, the *Object Tracking* module tracks the detected 100x100 pixel size moving target with purposive camera movement to follow the tracked target object.

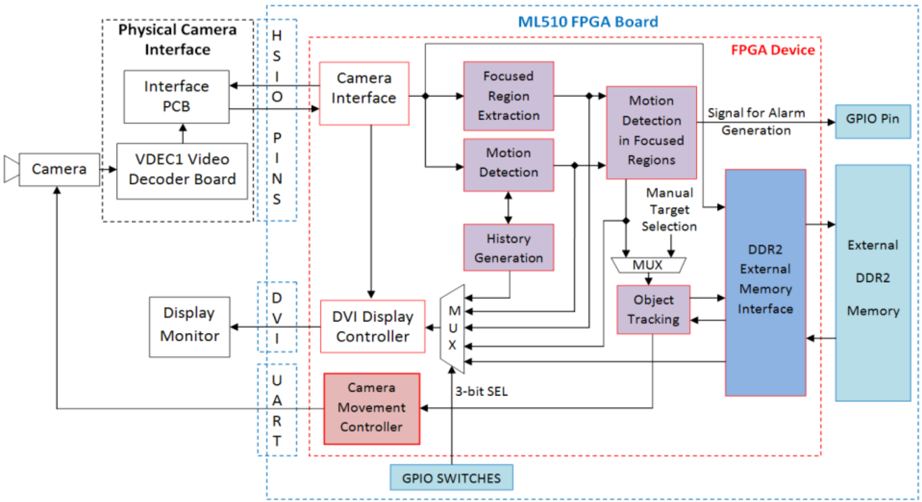


Fig. 1. Complete System Level Architecture of Designed and Implemented Smart Camera System.

As object tracking requires information from multiple video frames, the frame-wise pixel data output by the Camera Interface module is stored in *External DDR2 Memory* through *DDR2 External Memory Interface* controller. *Object Tracking* module also accesses the stored frames/video pixel data through this interface. The output of the *Object Tracking* module is written back to *External DDR2 Memory*. The *Object Tracking* module extracts and updates certain useful parameters internally, such as co-ordinates of new location of tracked object in the current frame. These parameters are used by the *Camera Movement Controller* and the *Object Tracking* module itself in the next frame. Based on the values of new co-ordinates of the tracked object in current frame, the *Camera Movement Controller* generates the necessary commands for purposive camera movement (pan-tilt) in required directions so that it may follow the tracked object. PTZ (Pan-Tilt-Zoom) *Camera* is used to cover a large field of view and follow the tracked object. The output of *Motion Detection* module is further modified by *History Generation* module to generate the history of moving objects in a video stream. The outputs of *Motion Detection* and *Motion Detection in Focused Regions* blocks also enable the filtering of frames of interest based on relevant motion in the video scene which in turn reduces further communication and processing overheads. The output of every individual module is connected to a *Display Monitor* through a multiplexer (*MUX*) and a *DVI Display Controller*. At any point of time, the output of any module can be viewed on the monitor by appropriately setting the multiplexer control lines. The displayed results are not just images but objects at higher levels of abstraction (e.g. extracted regions of interest, tracked object, filtered frames of interest, history of moving objects, etc.). At the heart of the system are five different efficient hardware units designed for executing intelligent algorithms.

The implemented prototype smart camera system is capable of automatically performing real-time motion detection, real-time video history generation, real-time focused region extraction, real-time motion detection only in focused regions (filtering of frames of interest), real-time object tracking of a manually selected target with automatic purposive camera movement, and real-time automatic moving target detection and tracking with purposive camera movement. Output of any of the five processing modules can be displayed by selecting the output of the associated module for display and sending it to *DVI Display Controller*. This output selection is done by a multiplexer (*MUX*). The output of a particular block can be selected for display based on the value of the 3-bit control signal to the multiplexer. The control signal is provided using *GPIO* (General Purpose Input Output) switches available on the Xilinx ML510 (Virtex-5 FX130T) FPGA development platform.

For developing this complete FPGA based smart camera system, we have designed and implemented four input/output interfaces [3], namely – *Camera Interface*, *DDR2 External Memory Interface*, *Camera Movement Controller*, and *DVI Display Controller* and five dedicated VLSI/hardware architectures, namely - *Motion Detection* [4]-[6], *Focused Region Extraction* [7], *History Generation* [8], *Motion Detection in Focused Regions*, and *Object Tracking* [9]-[10]. For details related to the design and implementation of these individual blocks, refer to [3]-[10].

3 Synthesis Results

A top level design module was created which invoked all the five hardware architectures and all the four input/output interfaces. A User Constraint File (UCF) was created to map the input/output ports of the design on the actual pins of the FPGA. All the above mentioned modules of the implemented smart camera system were coded in VHDL and simulated using ModelSim. This top level design was synthesized using Xilinx ISE (Version 12.1) tool chain. The resulting configuration (.bit) file was stored in the Flash Memory to enable automatic configuration of the FPGA at power-on. A standalone complete prototype of real-time smart camera system is built and is shown in Fig. 2. The components of this system are Xilinx ML510 (Virtex-5 FX130T) FPGA platform, Sony EVI D-70P Camera, and display monitor.

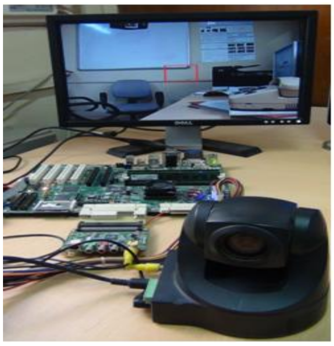


Fig. 2. Complete System Hardware Setup.

Table 1. FPGA Resources Utilized by Implemented Smart Camera System.

Resources	Resources Utilized	Total Available Resources	Percentage of Utilization
Slice Registers	28838	81920	35.20%
Slice LUTs	39916	81920	48.72%
Route-thrus	9729	163840	5.93%
Occupied Slices	13588	20840	65.20%
BRAMS 36 K	216	298	72.48%
Memory (Kb)	7776	10728	72.48%
DSP Slices	3	320	0.94%
IOs	292	840	34.76%
DCMs	2	12	16.67%

FPGA resources utilized by the complete smart camera prototype system are given in Table 1. Maximum operating frequency of the complete integrated system is 125.8 MHz, and, maximum possible frame rate for PAL (720x576) size color video is 244 frames per second. Synthesis results reveal that the complete implemented prototype smart camera system utilized approximately 66% FPGA slices and 73% Block RAMs (on-chip memory) on Xilinx ML510 (Virtex-5 FX130T) FPGA development platform. This is because, the implemented hardware accelerators were designed to meet the real-time requirements of video surveillance applications while aiming at FPGA resource reduction. Thus, there is further scope of implementing a few addition features of a smart camera system on the same FPGA development platform.

The implemented smart camera system works in real-time for a standard PAL (720x576) resolution live color video stream. All architectures of the implemented smart camera system are adaptable and scalable for other video resolutions also. The system is also capable of processing HD resolution videos in real-time. The seven configuration (.bit) files – one each corresponding to motion detection, history generation, focused region extraction, motion detection in focused regions (filtering of frames of interest), object tracking with purposive camera movement for manually selected target, automatic moving target detection and tracking with purposive camera movement, and complete integrated system are stored in the Compact Flash memory. With the help of three configuration switches available on the Xilinx ML510 (Virtex-5 FX130T) FPGA development platform, the desired configuration (.bit) file can be downloaded onto the FPGA at power-on and the system can be used for that purpose.

4 Visual Results and Discussions

The implemented prototype smart camera system can cater to various real world video surveillance applications such as motion detection, focused region extraction, video history generation, and automatic moving target detection and tracking with purposive camera movement. Its real-time test results for these applications are presented in this section. The prototyped system is tested for live colored video streams directly coming from the camera at 25 fps (frames per second) frame rate. The color video resolution is of standard PAL (720x576) size.

4.1 Motion Detection

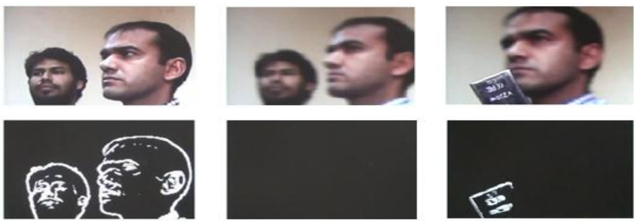
The implemented smart camera system for motion detection was tested for different real-world scenarios (both indoor and outdoor), which are broadly classified into two categories *i.e.*, static background situations and pseudo-stationary background situations. Fig. 3 shows examples of real-world situation of static background scenario captured by the camera. In Fig. 3, the background is static and the moving objects are present in the scene. Motion 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 motion detection system. Fig. 4 shows the scenario of pseudo-stationary background with moving foreground objects. In this case, there are moving leaves of the trees in the background. Despite these pseudo-stationary movements in background, only moving objects in the foreground have detected and the movements of leaves of trees in the background (irrelevant motion) have been eliminated. Results of the tests show that the implemented smart camera system is robust enough to detect only the relevant motion in a live video scene and eliminates the continuous unwanted movements in the background itself. All the color frames are of PAL (720 × 576) size and are extracted from live video streams (at 25 fps) produced by the implemented system.

Fig. 3. Moving Objects in Video Scene and Corresponding Motion Detected Outputs for Static Background Scenarios.

Fig. 4. Moving Objects in Video Scene and Corresponding Motion Detected Outputs for Pseudo-stationary Background Scenarios.

4.2 Focused Region Extraction

Three different possible situations are considered for testing focused region extraction feature of the implemented smart camera system (Fig. 5). The first row shows the input frames. The extracted focused edge pixels results are shown just below the respective input frame. Fig. 5(a) shows the situation where complete scene (both faces are focused) is in focus. The extracted focused edge pixels are shown just below the input frame in Fig. 5(a). The situation of completely out of focus scene is shown in Fig. 5(b). In this case the resulting image (shown just below the input frame in Fig. 5(b)) is blank as no focused edge pixel exists in input image. An example of focused (metallic device) and non-focused (face) objects in same scene is considered in Fig. 5(c). In this case, in the output image (shown just below the input frame in Fig. 5(c)), only focused edge pixels (metallic device) are extracted. The implementation yields good results and works in real-time. All the color frames are of PAL (720 × 576) size and are extracted from live video streams (at 25 fps) produced by the implemented system



(a) (b) (c)

Fig. 5. Non-focused and focused objects present in the scenes and only focused pixels are extracted.

4.3 Video History Generation

Results of video history generation are shown in Fig. 6. Fig. 6(a) shows the original extracted video frames from two different live video sequences. Corresponding results produced by the implemented motion detection system for each frame are shown in Fig. 6(b). These results show only the motion blocks in each frame. The results produced by implemented video history generation system for the two video sequences are shown in Fig. 6(c). For each input video sequences, video history frame is generated for 60 frames. The output video history frame for each video sequence shows more meaningful information about the scene and details about the moving object and the trajectory of the moving object. As a single output video history frame is generated for 60 input video frames, the output video frame rate is also reduced, thereby reducing the further communication and processing overheads.

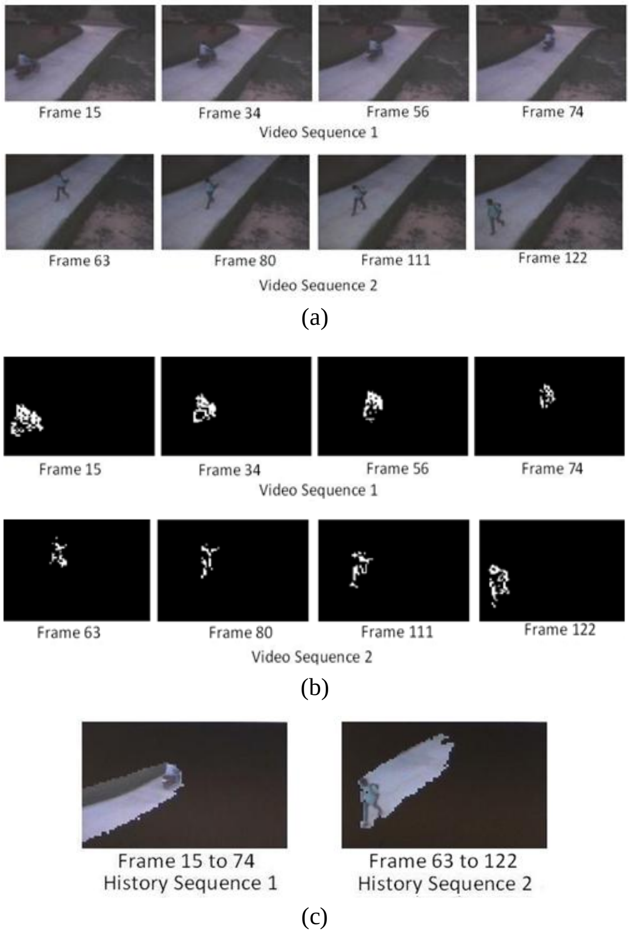


Fig. 6. (a) Original Video Frames; (b) Motion Detection Results; (c) Video History Generation Results.

4.4 Object Detection and Object Tracking

Results of automatic moving target detection and tracking for live video sequences are shown in Fig. 7. All images corresponding to two video sequences are extracted from live output results produced by the implemented smart camera system. In both the video sequences there exists no moving object in the first image. In the second image of both the video sequences, the moving object enters into the scene and system for motion detection in focused regions detects the moving object (results of motion detection in focused regions are not shown as motion detection system is working internally and results are displayed for tracking only). The moving object/person size in both the video sequences is larger than 100x100 pixel size and the object tracking system implemented is designed to track a target of maximum 100x100 pixel size. Therefore, the upper 100x100 pixel size part (i.e. person's face in this case) of the moving object is selected and red color rectangle is drawn by the implemented control logic. This is shown in the third image of each video sequence. In the remaining three images of each video sequence, the selected 100x100 pixel size target (person's face in this case) is tracked despite the presence of other moving objects/persons in the video scene. The background also changes due to purposive camera movement to follow the tracked target and to keep the tracked target in the middle of the frame.

Fig. 7. Automatic Moving Target Detection and Tracking with Purposive Camera Movement.

4.5 Filtering of Frames of Interest

The results of filtering of the frames of interest based on motion detection only in focused regions are shown in Fig. 8. Fig. 8(a) shows four different images extracted from different live video streams. Corresponding results produced by focused region extraction system for each image are shown in Fig. 8(b). In Fig. 8(a), Image 1 and Image 3 are focused images and Image 2 and Image 4 are defocused/blurred images. Therefore, results produced by the implemented focused region extraction system for Image 2 and Image 4 are black images, as in both the images there are no focused regions. The results produced by the focused region extraction system for Image 1 and Image 3 show the extracted focused edge regions. Motion detection results for the four images are shown in Fig. 8(c). Moving objects are present in all the four images. Therefore, results produced by the implemented motion detection system show the motion blocks for all the four images, irrespective of the fact that they are focused images or defocused images. The module for motion detection in focused regions detects motion in focused regions only and filters the frames accordingly (i.e. filters the frames which show the motion in focused regions only). Frames filtered by the implemented motion detection in focused regions system are shown in Fig. 8(d). Output frames are filtered for Image 1 and Image 3 as these are focused images. For defocused Image 2 and Image 4, the output frames are not filtered and therefore, the results show black images.

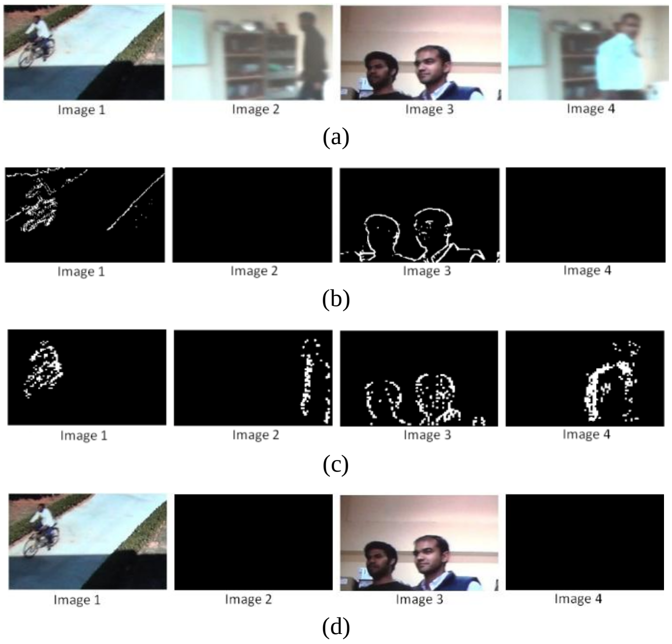


Fig. 8. (a) Original Video Frames; (b) Focused Region Extraction Results; (c) Motion Detection Results; (d) Filtered Frames of Interest based on Motion Detection only in Focused Regions.

5 Conclusions

In this article, we have described the system level architecture, functionality, synthesis results, and visual results of complete smart camera system developed on Xilinx ML510 (Virtex-5 FX130T) FPGA Board. The implemented system is capable of automatically performing real-time motion detection, real-time video history generation, real-time focused region extraction, real-time filtering of frames of interest based on motion detection only in focused regions, real-time object tracking of manually selected target with automatic purposive camera movement, and real-time automatic moving target detection and tracking with purposive camera movement. The system is designed to work in real-time for live color video streams of standard PAL (720x576) resolution, which is the most commonly used video resolution for current generation surveillance systems. The system is capable of processing HD resolution video streams in real-time and adaptable to different video resolutions. The developed standalone prototype smart camera system can be effectively used for surveillance applications.

Acknowledgments

This work was carried out under a project supported by Department of Electronics & Information Technology (DeitY) / Ministry of Communications and Information Technology (MCIT), Govt. of India.

References

1. A. Kandhalu, A. Rowe, R. Rajkumar, C. Huang, and C.C. Yeh, Real-time Video Surveillance over IEEE 802.11 Mesh Network, In Proceedings: 15th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 205-214, 2009.
2. H. Jiang, H. Ardo, and V. Owall, A Hardware Architecture for Real-time Video Segmentation Utilizing Memory Reduction Techniques, IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 2, pp. 226-236, 2009.
3. S. Singh, S. Saurav, C. Shekhar, and A. Vohra, Prototyping an Automated Video Surveillance System using FPGAs, International Journal of Image, Graphics and Signal Processing (IJIGSP), MECS Press, Vol. 8, No. 8, pp. 37-46, 2016.
4. S. Singh, S. M. Dunga, R. Saini, A.S. Mandal, C. Shekhar, A. Vohra, and S. Chaudhary, Hardware Accelerator Design for Change Detection in Smart Camera, SPIE Proceedings 8285: International Conference on Graphics and Image Processing (ICGIP 2010), Cairo, Egypt, Vol. 8285, Article ID 82852T, pp. 1-5, October 1-2, 2011.
5. S. Singh, A.S. Mandal, C. Shekhar, and A. Vohra, Real-time Implementation of Change Detection for Automated Video Surveillance System, ISRN Electronics, Hindawi, USA, Vol. 2013, Article ID 691930, pp. 1-5, 2013
6. S. Singh, C. Shekhar, and A. Vohra, FPGA-based Real-time Motion Detection for Automated Video Surveillance Systems, Electronics, MDPI, Switzerland, Vol. 5, No. 1, Article No. 10, pp. 1-18, 2016.
7. S. Singh, S. Saurav, R. Saini, A.K. Saini, C. Shekhar, and A. Vohra, Automatic Real-time Extraction of Focused Regions in a Live Video Stream using Edge Width Information, IEEE Proceedings: 18th International Symposium on VLSI Design and Test (VDAT-2014), PSG College of Technology, Coimbatore, India, pp. 1-2, July 16-18, 2014.
8. S. Singh, S.M. Dunga, A.S. Mandal, C. Shekhar, and S. Chaudhary, FPGA based Embedded Implementation of Video Summary Generation Scheme in Smart Camera Systems, Advanced Materials Research, Trans Tech Publications, Switzerland, Vol. 403-408, pp. 516-521, 2011.
9. S. Singh, S.M. Dunga, R. Saini, A.S. Mandal, C. Shekhar, and A. Vohra, Hardware Accelerator Design for Tracking in Smart Camera, SPIE Proceedings 8285: International Conference on Graphics and Image Processing (ICGIP 2010), Cairo, Egypt, Vol. 8285, Article ID 82852S, pp. 1-5, October 1-2, 2011.
10. S. Singh, R. Saini, S. Saurav, and A.K. Saini, Real-time Object Tracking with Active PTZ Camera using Hardware Acceleration Approach. International Journal of Image, Graphics and Signal Processing (IJIGSP), MECS Press, Vol. 9, No. 2, pp. 55-62, 2017.