I.J. Image, Graphics and Signal Processing, 2017, 2, 55-62
Published Online February 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2017.02.07

Real-time Object Tracking with Active PTZ Camera using Hardware Acceleration Approach

Sanjay Singh, Ravi Saini, Sumeet Saurav, Anil K Saini
CSIR - Central Electronics Engineering Research Institute (CSIR-CEERI)
Pilani - 333031, Rajasthan, India.
Email: sanjay.csirceeri@gmail.com

***Abstract*—**This paper presents the design and implementation of a dedicated hardware (VLSI) architecture for real-time object tracking. In order to realize the complete system, the designed VLSI architecture has been integrated with different input/output video interfaces. These video interfaces along with the designed object tracking VLSI architecture have been coded using VHDL, simulated using ModelSim, and synthesized using Xilinx ISE tool chain. A working prototype of complete object tracking system has been implemented on Xilinx ML510 (Virtex-5 FX130T) FPGA board. The implemented system is capable of tracking the moving target object in real-time in PAL (720x576) resolution live video stream directly coming from the camera. Additionally, the implemented system also provides the real-time desired camera movement to follow the tracked object over a larger area.

***Index Terms*—**VLSI Architecture, Real-time Object Tracking, FPGA Implementation.

I. INTRODUCTION

Visual object tracking is one of the exigent missions in computer vision community and has a wide range of real-world applications, including surveillance and monitoring of human activities in residential areas, parking lots, and banks [1]-[2], smart rooms [3]-[4], traffic flow monitoring [5], mobile video conferencing [6], and video compression [7]-[8]. Different issues arising in development of a real-time object tracking system for such applications are because of camera movement, complex object motion, presence of other moving objects in video scene, and real-time processing requirements. These issues may cause degradation in tracking procedure, and, even failures. To deal with these issues, numerous algorithms for object tracking have been proposed in the literature. Detailed surveys of these algorithms have been presented by Yilmaz et al. [9], Li et al. [10], and Porikli [11]. Most of these algorithms either simplified their domains by imposing constraints on the class of objects they can track [12]-[14], or assumed that the camera is stationary and exploited background subtraction/segmentation [15]-[20]. In case of all background segmentation based tracking approaches, if the background changes due to camera movement, the whole system must be initialized again for capturing background information. Therefore, these approaches work only for fixed backgrounds - thus limiting their applications for stationary camera systems.

Several techniques have been proposed for object tracking which did not assume static background. The object trackers based on condensation [21], covariance matrix matching [22], normalized correlation [23], active contours [24]-[25], appearance modeling [26], incremental principal component analysis [27], kernel scheme [28]-[31], and particle filter [32]-[34] can track the object of interest even in the presence of camera motion. But most of these algorithms discussed above for object tracking in dynamic backgrounds are computationally too expensive to be deployed on a moderate computing machine for a practical real-time tracking surveillance application. Recently, some researchers have combined color histogram with particle filtering framework to design computationally less expensive object trackers for dynamic background situations and achieved real-time performance by designing their hardware architectures [35]-[37]. In their implementations the maximum size considered for object of interest was 20x20 pixels [35] and the maximum number of particles considered were 31 [35]. As the number of particles considered directly effects the robustness of the tracker, their robustness is relatively low. Several other researchers tried to implement the object tracking system using hardware/VLSI approach [38]-[46], but these implementations fell short of a complete system design and none of them discusses or reports the details of input/output video interface design, real-time results, frame resolutions, and frame rates.

For an object tracking system, to be useful in current real-world surveillance scenarios, should have at least real-time video input interface for capturing live video streams, output interface for displaying results, external memory (DDR) interface for storing multiple intermediate video frames, and camera movement control interface for desired automatic movement of the camera. Unfortunately, none of the implementations, discussed above, addresses all these issues together.

In this paper, we present the design and development of a real-time object tracking system with automatic purposive camera movement capabilities by combining simplified particle filtering framework with color histogram computation that can handle camera movement and associated background change issues. For achieving better robustness, we have used 121 particles as compared to 31 particles by [35] and the maximum tracked object size of 100x100 pixels as compared to 20x20 pixels by [35]. For algorithmic verification, the simplified particle filtering and color histogram based object tracking scheme has been initially implemented in C/C++ programming language on 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). The Open Computer Vision (OpenCV) libraries have been used in the code for reading video streams (either stored or coming from camera) and displaying object tracking results. To achieve real-time performance for PAL (720x576) resolution videos, we have designed a dedicated hardware architecture for the above mentioned scheme and implemented it on the FPGA development platform. In addition to the dedicated object tracking hardware (VLSI) architecture, we have also developed four input/output interfaces, namely input camera interface for capturing live video streams, output display interface for displaying the object tracking results, the DDR2 external memory interface to store the large amount of video data needed during object tracking, and the camera movement feedback interface to provide real-time purposive camera movement to follow the tracked object and keep it in the field of view of the camera as long as possible. By integrating the designed dedicated object tracking VLSI architecture with all these four interfaces, a FPGA-based working prototype of complete object tracking system with automatic purposive camera movement capabilities has been developed on Xilinx ML510 (Virtex-5 FX130T) FPGA platform.

The rest of the paper is organized as follows: in the next section, we present the details of the design and working of complete system level architecture for implemented object tracking system on Xilinx ML510 FPGA board. Synthesis and experimental results are reported in section three and four respectively. Finally, we conclude this paper with a short summary.

II. Proposed and Developed Object Tracking System

The complete system level architecture of the proposed, designed, and implemented FPGA-based real-time object tracking system using Xilinx ML510 (Virtex-5 FX130T) FPGA platform is shown in Fig. 1. The different components of the complete FPGA-based standalone object tracking system are: analog PTZ Camera, VDEC1 Video Decoder Board for analog to digital video conversion, custom designed Interface PCB, Xilinx ML510 (Virtex-5 FX130T) FPGA platform, and display monitor.

Digilent VDEC1 Video Decoder Board digitizes the input video captured by a Sony Analog Camera and corresponding digital signals are transferred to the FPGA platform using high speed I/O ports (HSIO PINS) available on Xilinx ML510 (Virtex-5 FX130T) FPGA Board using custom designed Interface PCB. The components inside dark dashed blue line are available on Xilinx ML510 (Virtex-5 FX130T) FPGA Board. These include FPGA Device, DDR2 Memory, High Speed Input Output Interface (HSIO PINS), UART port, and DVI Interface port. In order to access the DDR2 memory, the four FIFO (first-in-first-out registers) interfaces (Input FIFO1, Input FIFO2, Output FIFO1, and Output FIFO2) are designed. This is because the clock rate at which the DDR2 memory operates is different than the clock rates at which the different FPGA modules work (which accesses the DDR2 memory).

The Camera Interface module receives the digitized video signals and extracts the RGB pixel data. The extracted RGB pixel data is written to the DDR2 memory through input interface FIFO1. The DVI display interface reads the processed image data, stored in DDR2 memory, through output FIFO2 and sends it to the display monitor through DVI connector available on FPGA development platform. The object tracking module (shown inside dashed gray line) reads pixel data from DDR2 memory through output FIFO1 interface and writes back the processed pixel data to the DDR2 memory through input FIFO2 interface. The camera feedback module takes inputs from the object tracking module running on FPGA and generates the commands for automatic purposive movement of the camera. The generated commands are transferred to the camera through UART port using RS232 controller. The Controller module receives the signals (video timing signals) from the camera interface module and generates the necessary control signals for all modules of the object tracking system to ensure proper functioning and synchronization among different modules of the architecture. The functionalities of object tracking module and camera feedback controller module are described in details in the following sub-section.

A. Object Tracking Architecture

The object tracking architecture reads the stored image pixel data from the DDR2 memory through output FIFO1 interface. During the initialization process (performed during the initial frames), the reference image of the object to be tracked is read from DDR2 memory and stored in the reference image buffer memory and is used during the tracking procedure. The size of the object to be tracked is considered as 100x100 pixels and each RGB color pixel is of 24 bits. Therefore, reference image buffer memory is of size 100x100x24 bits. Once the initial reference image of the object to be tracked is written in reference image buffer memory, color histogram of the 100x100 pixel reference image is computed by the histogram computation module.

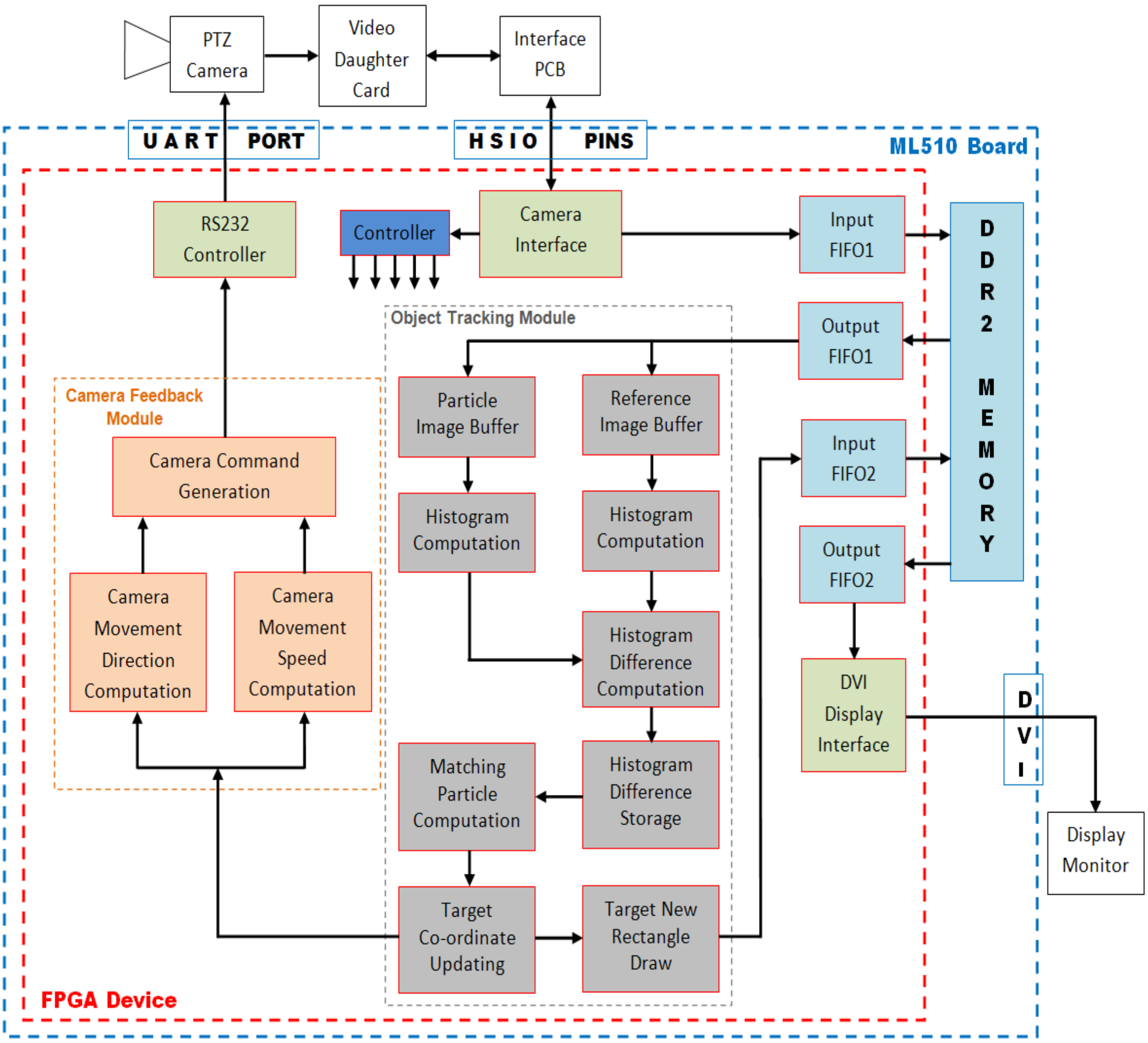

Fig.1. Proposed and Developed Object Tracking System Architecture.

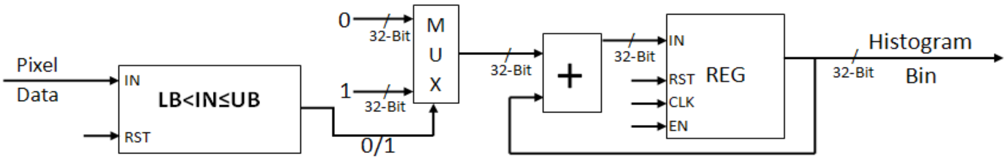

Fig.2. Histogram Bin Computation Architecture.

After initialization process, for all the subsequent frames, the objective is to track the object of interest in incoming video frames. This is done by computing the histograms for the 121 particles in the current frame. Computation of histograms for the 121 particles requires the reading of 121 images of 100x100 pixel size in each frame. For this reason, for all subsequent incoming frames, the particle images are read from DDR2 memory through output FIFO1 and are stored in particle image memory. The corresponding histogram computation module computes the histograms for each particle. Color histograms (100x100 pixel size color images) are computed separately for each of the R, G, and B color channels. For each color channel's histogram, we have considered 26 bins for getting enhanced accuracy. These bins are: B1 for data values from 0 to 9, B2 for data values from 10 to 19, B3 for data values from 20 to 29, B4 for data values from 30 to 39, B5 for data values from 40 to 49, ………., B25 for data values from 240 to 249, and B26 for data values from 250 to 255. The output of histogram computation module contains the 26 bin values separately. The basic architecture of histogram computation for single bin is shown in Fig. 2. The incoming pixel data is compared with the boundary values of the bin (i.e. with lower boundary value LB and upper boundary value UB). If the incoming pixel value lies in range then it outputs 1 otherwise it outputs 0. Based on this output, the bin counter value either gets incremented or remains same. We have used 26 such modules in parallel, each corresponding to a bin, for computing the histogram of one color channel.

After computing the histogram of the particle image, the next step is to find the matching index of the particle image histogram with the reference image histogram. This is done by computing the absolute histogram difference value between the particle image histogram and the reference image histogram by histogram difference computation module. This histogram difference value is stored by histogram difference storage module. This process will continue till the histogram difference values for all 121 particles images with reference image get computed and stored. In order to speed up the tracking procedure, we have processed three particles at time by exploiting the parallelism concept. It takes a total of 209100 clock cycles to compute these 121 histogram difference values. Once the all 121 histogram difference values (which contains the similarity information between particle images and reference image of the object to be tracked) of all 121 particle images are stored in the histogram difference storage module, the next step is to find the best matching particle among 121 particles. This is done by matching particle computation module by finding the minimum histogram difference value among 121 particles. The particle with minimum histogram difference value is the particle having maximum similarity with reference image of the object to be tracked. The target co-ordinate updating module finds the co-ordinates of this best matched particle and update the co-ordinates of the tracked object as these are the new co-ordinates of tracked object in the current frame. The updated co-ordinate information is passed on to target new rectangle draw module, which writes the updated rectangle pixel data to the DDR2 memory through input FIFO2 interface. The updated co-ordinate values are also used by camera feedback controller module. The updated co-ordinate values of the tracked object in the current frame act as the X, Y co-ordinates of the tracked object for the incoming next frame. In the next frame, all the 121 particles will be generated around these updated co-ordinate values and computations will be performed on them.

B. *Camera Movement Controller*

Sony EVI-D70P camera, used in our implementation, has pan, tilt, and zoom features. In order to automatically control these pan-tilt features in real-time, a dedicated camera feedback controller module has been designed. It takes the updated location co-ordinates as inputs from the object tracking module and generates the necessary commands for purposive movement of the camera. It has three main modules as shown in Fig. 1 (modules with light orange color background inside light orange color dashed line). Camera movement direction and camera movement speed are computed by Camera Movement Direction Computation and Camera Movement Speed Computation blocks, respectively. Camera Command Generation module uses speed and direction information of these blocks to generate necessary commands for controlling the desired camera movement in the XY plane. The generated commands are sent to RS232 Controller which sends these commands to the camera. The camera receives the generated movement control commands through a cable which connects the VISCA RS232 IN port of the camera to the UART PORT of the Xilinx ML510 (Virtex-5 FX130T) FPGA board.

III. Synthesis Results

The proposed architectural modules of the object tracking system are coded in VHDL. The simulation is carried out using ModelSim. A top level design module is created which invokes all the design modules. The input/output ports of the design are mapped on actual pins of the FPGA by creating a User Constraint File (UCF). Synthesis of the complete design is carried out using Xilinx ISE (Version 12.1) tool chain. The resulting configuration (.bit) file is stored in the Flash Memory to enable automatic configuration of the FPGA at power-on. Thus, a complete standalone prototype system for real-time object tracking is developed which is shown in Fig. 3. The three main components of the system are Xilinx ML510 (Virtex-5 FX130T) FPGA platform, Sony EVI D-70P Camera, and the display monitor.

Table 1 shows the FPGA resources (post-place and route results) utilized by the proposed and implemented complete object tracking hardware architecture. The maximum operating frequency is 48.116 MHz and the maximum possible frame rate for PAL (720x576) size color video is 116 frames per second. Synthesis results reveal that the implemented object tracking system utilizes approximately 53% FPGA slices and 35% Block RAMs (on-chip memory) on Xilinx ML510 (Virtex-5 FX130T) FPGA development platform.

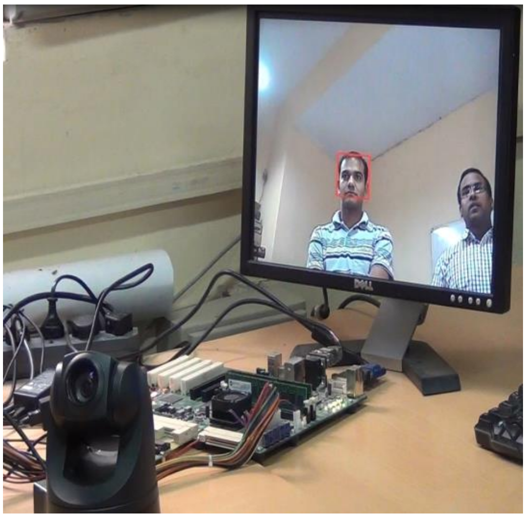

Fig.3. Complete Object Tracking System Hardware Setup.

Table 1. FPGA Resource Utilization by Complete Object Tracking Implementation.

Resources	Resources Utilized	Total Available Resources	Percentage of Utilization
Slice Registers	24417	81920	29.81%
Slice LUTs	33604	81920	41.02%
Occupied Slices	11075	20840	53.14%
BRAMs 36K	103	298	34.56%
Memory (Kb)	3708	10728	34.56%
DSP Slices	3	320	0.94%
IOs	292	840	34.76%
DCMs	1	12	8.33%

Table 2. Comparison with Existing Object Tracking Implementations.

Implementation	FPGA Device	Object Size	No. of Particles	Video Resolution	Frame Rate (fps)
Our Implementation	Virtex5 (xc5fx130t)	100x100	121	PAL (720x576)	116
[35]	Virtex5 (xc5lx110t)	20x20	31	136x136	Not Available
[36]	Virtex4 (xc4vlx200)	15x15	Not Available	VGA (640x480)	81
[37]	Virtex4 (xc4vlx200)	15x15	Not Available	VGA (640x480)	81

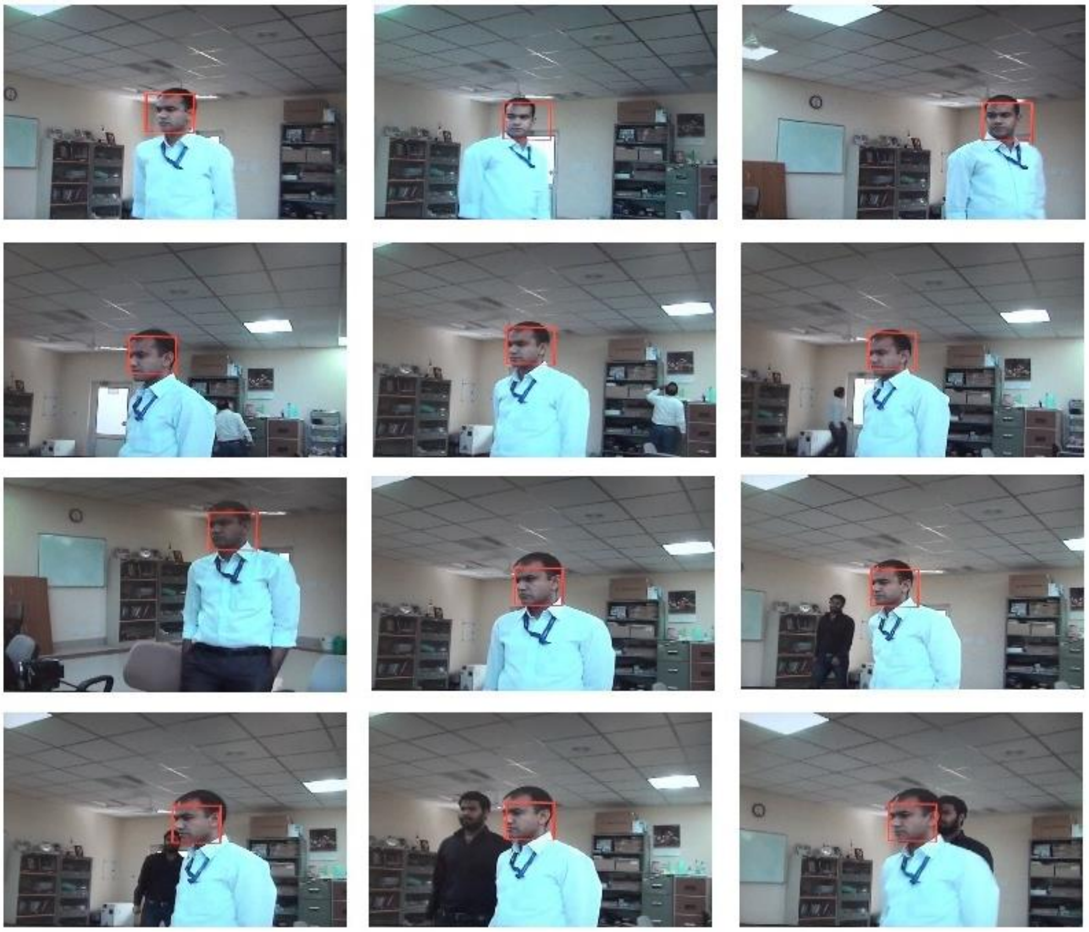

Fig.4. Frame Sequence Showing Object Tracking in Scene that changes due to Camera Movement and Presence of Other Moving Objects in the Scene.

We have compared our proposed and implemented object tracking architecture with existing color histogram and/or particle filtering based object tracking implementations [35]-[37]. The comparison is shown in Table 2. For comparison we have considered four parameters i.e. Frame Size / Video Resolution, Tracked Object Size, No of Particles Considered (more number of particles leads to robust tracking), and Frame Rate. It is thus clear that the architecture for object tracking proposed and implemented by us in this paper outperforms the existing architectural implementations in terms of processing performance (frame rate), video resolutions, tracked object size, and number of particles. The proposed and implemented object tracking system architecture is adaptable and scalable for different video sizes and target object sizes. Another point to note is that automatic purposive camera movement (pan-tilt) mechanism had not been designed in any of the hardware implementations considered for comparison. Although, there exist object trackers in literature which were designed using PTZ cameras [47]-[51], but all these were implemented on PCs/servers using Matlab or C/C++ programming languages.

IV. Object Tracking Results

We have tested our implemented system by running it for real-world scenarios for tracking of an object of interest in live color video streams directly coming from the camera. Camera used for capturing the live video stream is Sony EVI D70P and is of PAL (720x576) resolution. The tracking results for the captured real-world scenario are shown in Fig. 4. In different frames of the video sequence of Fig. 4, the scene is changing due to camera movement and the presence of other multiple moving objects in the background. The background in each frame is different because of the camera movement. Despite these changes in the background due to camera movement and presence of other moving objects, the object of interest initially (in first frame) present in 100x100 pixel box is tracked robustly. The system provides the real-time feedback to the camera by generating the commands to automatically move the camera in the direction required to follow the tracked object and keep it in the field of view of the camera as long as possible. All the frames shown in Fig. 4 are extracted from the results produced and displayed by the object tracking system (designed and developed by us in this paper) on the monitor for live PAL (720x576) resolution color video streams coming directly from the PTZ camera. These results demonstrate that the implemented system can robustly track an object of interest in real-time for video surveillance applications.

V. Conclusions

This paper has presented the design and implementation of our proposed standalone system for real-time object tracking with automatic camera movement capabilities. The system has been implemented by designing a dedicated hardware architecture for object tracking and associated input/output video interfaces by using VLSI design methodologies. The complete system, implemented on Xilinx ML510 (Virtex-5 FX130T) FPGA Board, robustly tracks the target object present in the scene in real-time for standard PAL (720x576) resolution color video streams directly coming from the camera and at the same time provides the automatic camera movement to follow the tracked target object. The automatic camera movement capabilities allow the system to track the target objects over a larger area. The implemented system can be effectively used as a standalone system for video surveillance applications to track an object of interest in real-time in a live video stream coming directly from a camera.

Acknowledgment

The financial support of Department of Electronics & Information Technology (DeitY) / Ministry of Communications and Information Technology (MCIT), Govt. of India is gratefully acknowledged.

References

[1] I. Haritaoglu, D. Harwood, and L.S. Davis, W4: Real-time Surveillance of People and Their Activities, IEEE Transsaction on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 809-830, 2000.

[2] X. Chen and J. Yang, Towards Monitoring Human Activities Using an Omni-directional Camera, In Proceedings: Fourth IEEE International Conference on Multimodal Interfaces, pp. 423-428, 2002.

[3] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, Pfinder: Real-Time Tracking of the Human Body, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 780-785, 1997.

[4] S.S. Intille, J.W. Davis, and A.E. Bobick, Real-Time Closed-World Tracking, In Proceedings: IEEE Conference on Computer Vision and Pattern Recognition, pp. 697-703, 1997.

[5] B. Coifmana, D. Beymerb, P. McLauchlanb, and J. Malikb, A Real-Time Computer Vision System for Vehicle Tracking and Traffic Surveillance, Transportation Research Part C: Emerging Technologies, Vol.6, No. 4, pp. 271–288, 1998.

[6] S. Paschalakis, and M. Bober, Real-Time Face Detection and Tracking for Mobile Videoconferencing, Real-Time Imaging, Vol. 10, No. 2, pp. 81–94, 2004

[7] T. Sikora, The MPEG-4 Video Standard Verification Model, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 1, pp. 19-31, 1997.

[8] A. Eleftheriadis and A. Jacquinb, Automatic Face Location Detection snd Tracking for Model-Assisted Coding of Video Teleconferencing Sequences at Low Bit-Rates, Signal Processing: Image Communication, Vol. 7, No. 3, pp. 231–248, 1995.

[9] Yilmaz, O. Javed, and M. Shah, Object Tracking: A Survey, ACM Computing Surveys, Vol. 38, No. 4, Article 13, pp. 1-45, 2006.

[10] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and V.D. Hengel, “A Survey of Appearance Models in Visual Object Tracking, ACM Transactions on Intelligent

Systems and Technology, Vol. 4, No. 4, Article 58, pp. 1-48, 2013.

[11] F. Porikli, Achieving Real-time Object Detection and Tracking under Extreme Condition, Journal of Real-time Image Processing, Vol. 1, No. 1, pp. 33-40, 2006.

[12] A. Doulamis, N. Doulamis, K. Ntalianis, and S. Kollias, An Efficient Fully Unsupervised Video Object Segmentation Scheme Using an Adaptive Neural-Network Classifier Architecture, IEEE Transactions on Neural Networks, Vol. 14, No. 3, pp. 616-630, 2003.

[13] J. Ahmed, M.N. Jafri, J. Ahmad, and M.I. Khan, Design and Implementation of a Neural Network for Real-Time Object Tracking, International Journal of Computer, Information, Systems and Control Engineering, Vol. 1, No. 6, pp. 1825-1828, 2007.

[14] J. Ahmed, M.N. Jafri, and J. Ahmad, Target Tracking in an Image Sequence Using Wavelet Features and a Neural Network, In Proceedings: IEEE Region 10 TENCON 2005 Conference, pp. 1-6, 2005.

[15] C. Stauffer and W. Grimson, Learning Patterns of Activity using Real-time Tracking, IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 747-757, 2000.

[16] C. Kim and J. N. Hwang, Fast and Automatic Video Object Segmentation and Tracking for Content-Based Applications, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 2, pp. 122-129, 2002.

[17] T. Gevers, Robust Segmentation and Tracking of Colored Objects in Video, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 6, pp. 776-781, 2004.

[18] C.E. Erdem, Video Object Segmentation and Tracking using Region-based Statistics, Signal Processing: Image Communication, Vol. 22, No. 10, pp. 891-905, 2007.

[19] K.E. Papoutsakis and A.A. Argyros, Object Tracking and Segmentation in a Closed Loop, Advances in Visual Computing: Lecture Notes in Computer Science, Vol. 6453, pp. 405-416, 2010.

[20] F. Kristensen, H. Hedberg, H. Jiang, P. Nilsson, and V.O. Wall, An Embedded Real-Time Surveillance System: Implementation and Evaluation, Journal of Signal Processing Systems, Vol. 52, No. 1, pp. 75-94, 2008.

[21] M. Isard and A. Blake, CONDENSATION – Conditional Density Propagation for Visual Tracking, International Journal of Computer Vision, Vol. 29, No. 1, pp. 5 - 28, 1998.

[22] F. Porikli, O. Tuzel, and P. Meer, Covariance Tracking using Model Update Based on Lie Algebra, In Proceedings: IEEE Conference on Computer Vision and Pattern Recognition, pp. 728-735, 2006.

[23] S. Wong, Advanced Correlation Tracking of Objects in Cluttered Imagery: In Proceeding: SPIE, Vol. 5810, pp. 1-12, 2005.

[24] A. Yilmaz, X. Li, and M. Shah, Contour-based Object Tracking with Occlusion Handling in Video Acquired using Mobile Cameras, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 11, pp. 1531-1536, 2004.

[25] M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models, International Journal of Computer Vision, Vol. 1, No. 4, pp 321-331, 1988.

[26] M. J. Black and A. D. Jepson, EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-based Representation, International Journal of Computer Vision, vol. 26, no. 1, pp. 63 - 84, 1998.

[27] C.M. Li, Y.S. Li, Q.D. Zhuang, Q.M. Li, R.H. Wu, and Y. Li, Moving Object Segmentation and Tracking In Video, In Proceedings: Fourth International Conference on Machine Learning and Cybernetics, pp. 4957-4960, 2005.

[28] D. Comaniciu, R. Visvanathan, and P. Meer, Kernel based Object Tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, pp. 564-577, 2003.

[29] D. Comaniciu, V. Ramesh, P. Meer, Real-time Tracking of Non-Rigid Objects Using Mean Shift, In Proceedings: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 142–149, 2000.

[30] V.P. Namboodiri, A. Ghorawat, and S. Chaudhuri, Improved Kernel-Based Object Tracking Under Occluded Scenarios, Computer Vision, Graphics and Image Processing: Lecture Notes in Computer Science, Vol. 4338, pp. 504-515, 2006.

[31] A. Dargazany, A. Soleimani, and A. Ahmadyfard, Multibandwidth Kernel-Based Object Tracking, Advances in Artificial Intelligence, Vol. 2010, Article ID 175603, pp. 1-15, 2010.

[32] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking, IEEE Transactions on Signal Processing, Vol. 50, No. 2, pp. 174-188, 2002.

[33] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.J. Nordlund, Particle Filters for Positioning, Navigation, and Tracking, IEEE Transactions on Signal Processing, Vol. 50, No. 2, pp. 425-237, 2002.

[34] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, Color-Based Probabilistic Tracking, Computer Vision: Lecture Notes in Computer Science, Vol. 2350, pp. 661-675, 2002.

[35] S. Agrawal, P. Engineer, R. Velmurugan, and S. Patkar, FPGA Implementation of Particle Filter based Object Tracking in Video, In Proceedings: International Symposium on Electronic System Design, pp. 82-86, 2012.

[36] J.U. Cho, S.H. Jin, X.D. Pham, D. Kim, and J.W. Jeon, A Real-Time Color Feature Tracking System Using Color Histograms, In Proceedings: International Conference on Control, Automation and Systems, pp. 1163-1167, 2007.

[37] J.U. Cho, S.H. Jin, X.D. Pham, D. Kim, and J.W. Jeon, FPGA-Based Real-Time Visual Tracking System Using Adaptive Color Histograms, In Proceedings: IEEE International Conference on Robotics and Biomimetics, pp. 172-177, 2007.

[38] J. Xu, Y. Dou, J. Li, X. Zhou, and Q. Dou, FPGA Accelerating Algorithms of Active Shape Model in People Tracking Applications, In Proceedings: 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, pp. 432-435, 2007.

[39] M. Shahzada and S. Zahidb, Image Coprocessor: A Real-time Approach towards Object Tracking, In Proceedings: International Conference on Digital Image Processing, pp. 220-224, 2009.

[40] K.S. Raju, G. Baruah, M. Rajesham, P. Phukan, and M. Pandey, Implementation of Moving Object Tracking using EDK, International Journal of Computer Science Issues, Vol. 9, No. 3, pp. 43-50, 2012.

[41] K.S. Raju, D. Borgohain, and M. Pandey, A Hardware Implementation to Compute Displacement of Moving Object in a Real Time Video, International Journal of Computer Applications Vol. 69, No. 18, pp. 41-44, 2013.

[42] M. McErlean, An FPGA Implementation of Hierarchical Motion Estimation for Embedded Object Tracking, In Proceedings: IEEE International Symposium on Signal Processing and Information Technology, pp. 242-247, 2006.

[43] Y.P. Hsu, H.C. Miao, and C.C. Tsai, FPGA Implementation of a Real-Time Image Tracking System, In Proceedings: SICE Annual Conference, pp. 2878 - 2884, 2010.
[44] L.N. Elkhatib, F.A. Hussin, L. Xia, and P. Sebastian, An Optimal Design of Moving Object Tracking Algorithm on FPGA, In Proceedings: International Conference on Intelligent and Advanced Systems, pp. 745 - 749, 2012.
[45] S. Wong and J. Collins, A Proposed FPGA Architecture for Real-Time Object Tracking using Commodity Sensors, In Proceedings: 19th International Conference on Mechatronics and Machine Vision in Practice, pp. 156-161, 2012.
[46] D. Popescu and D. Patarniche, FPGA Implementation of Video Processing-Based Algorithm for Object Tracking, U.P.B. Sci. Bull., Series C, Vol. 72, No. 3, pp. 121-130, 2010.
[47] S. Kang, J.K. Paik, A. Koschan, B.R. Abidi, M.A. andAbidi, Real-time video tracking using PTZ cameras, In Proceedings: SPIE, Volume 5132, pp. 103-111, 2003.
[48] T. Dinh, Q. Yu, and G. Medioni, Real Time Tracking using an Active Pan-Tilt-Zoom Network Camera, In Proceedings: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3786-3793, 2009.
[49] P.D.Z. Varcheie and G.A. Bilodea, Active People Tracking by a PTZ Camera in IP Surveillance System, In Proceedings: IEEE International Workshop on Robotic and Sensors Environments, pp. 98 – 103, 2009.
[50] M.A. Haj, A.D. Bagdanov, J. Gonzalez, and F.X. Roca, Reactive object tracking with a single PTZ camera, In Proceedings: International Conference on Pattern Recognition, pp. 1690-1693, 2010.
[51] S.G. Lee and R. Batkhishig, Implementation of a Real-Time Image Object Tracking System for PTZ Cameras, Convergence and Hybrid Information Technology: Communications in Computer and Information Science, Vol. 206, pp 121-128, 2011.

Authors' Profiles

Sanjay Singh is working as Scientist in CSIR-Central Electronics Engineering Research Institute, Pilani, India - A Government of India Research Laboratory. He is Member of IEEE - USA, IACSIT - Singapore, IAENG - Hong Kong, and SSI - India. He is involved in various projects sponsored by Indian Government on Computer Vision and VLSI Design. His research interests are VLSI Architectures for Computer Vision, FPGA Prototyping, and VLSI Design. He received his Ph.D. in VLSI Design for Computer Vision, M.Tech. in Microelectronics & VLSI Design, M.Sc. in Electronics, and B.Sc. in Electronics & Computer Science from Kurukshetra University, Kurukshetra, Haryana, India. He earned Gold Medal (First Position in University) during his M.Tech. and M.Sc. He topped college during B.Sc. He received more than 20 Merit Certificates and Scholarships during his academic career.

Ravi Saini is working as Scientist in CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan, India. He is Member of IEEE – USA. His research interests include VLSI Architectures, ASIC and ASIP Design, HDLs, and FPGA Prototyping. He received his Ph.D. in VLSI Design from Kurukshetra University, Kurukshetra, Haryana, India, M.Tech. in Microelectronics from Punjab University, India and M.Sc. in Electronics from DAVV, Indore, India.

Sumeet Saurav is working as Scientist in CSIR-Central Electronics Engineering Research Institute, Pilani, India - A Government of India Research Laboratory. His research interests are Computer Vision and VLSI Design. He received his M.Tech. in Advanced Semiconductor Devices from Academy of Scientific & Innovative Research (AcSIR), India.

Anil Kumar Saini is working as Scientist in CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan, India. He is Member of IEEE – USA and IACSIT – Singapore. His research interests include Analog and Mixed Signal Design, Embedded System Design, and CMOS RF IC design. Prior to joining this research lab, he worked in Cadence, India. He received his Master in Technology and Master in Science in 2003 and 2000 respectively from IIT Roorkee, India.

How to cite this paper: Sanjay Singh, Ravi Saini, Sumeet Saurav, Anil K Saini,"Real-time Object Tracking with Active PTZ Camera using Hardware Acceleration Approach", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.9, No.2, pp.55-62, 2017.DOI: 10.5815/ijigsp.2017.02.07