International Journal of Research Studies in Science, Engineering and Technology
Volume 2, Issue 10, October 2015, PP 107-115
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

ASIC- FPGA Implementation of Real Time Video Segmentation for Surveillance System

Siva Nagi Reddy Kalli [1], Dr. Bhanu Murthy Bhaskara[2]

[1] Research Scholar, Hyderabad, Telangana, India
sivanagireddykalli@gmail.com
[2] Professor and Vice Dean, Faculty of Integrative Sciences and Technology,
Quest International University, Perak, MALAYSIA
bhanu.b@gmail.com

Abstract: *In recent times, Detection and tracking of moving objects is important in the analysis of video data and higher level security assessment for both a civilian and a military purpose, e.g. traffic control, security monitoring and antiterrorism. Image registration plays a key role in designing video surveillance system. The purpose of image segmentation in surveillance application such as feature detection, or motion sensing. But the real-time video segmentation has suffered from the low computational power of the last generation machines. Segmentation of single frames resulted too slow or with an acceptable frame rate, but subjected to excessive restrictions. To this aim, the usage of adaptive Gaussian mixture models had become a standard in the recent years, due to their theoretical foundations and analytical representation. Use of high performance FPGA with specific application like ASIC-FPGA that can be dynamically reconfigured for video surveillance applications, together with camera interface can offer improved performance. The Main aim of paper focuses Identification of the back ground in video using the Gaussian Mixture Model (GMM) Background Subtraction implemented on Xilinx Kintex-7KC705 (xc7k325tffg900-2) using Vivado HLS (2014.2) software. . It allows to process a 720 * 576 pixels and 20 frames per second video stream in real-time and reduces the memory bandwidth of more than 80%.*

Keywords: *Xilinx, GMM, ASIC-FPGA, Background Subtraction, Kintex-7, BRAM, Match*

1. INTRODUCTION

Foreground/Background (F/B) segmentation has received many efforts in the last decade, due to their less computational burdensome requirements and it is often used in different applications to model the background and then detect the moving objects in the scene like in video surveillance [1]-[3], optical motion capture [4]-[6] and multimedia [7]-[10]. The simplest way to model the background is to acquire a background image which doesn't include any moving object. In some environments, the background isn't available and can always be changed under critical situations like illumination changes, objects being introduced or removed from the scene. To take into account these problems of robustness and adaptation, many background modeling methods have been developed and the most recent surveys can be found in [11].

Background modeling can be clustered into two main groups: non-statistical [12-14] and statistical approaches [15, 16]. The non-statistical background modeling presented in [5], considers each pixel in a frame to be either as part of the moving object (simply the object) or the background. In this approach, the first frame is considered as the background scene and the sequence of frames are subtracted from the background scene. Then the pixels with a value higher than a threshold are considered as the objects. In this approach, the background is updated along the frame sequences. In the second group of background modeling approaches, the statistical based approaches, and the probability density functions of the background pixels is estimated; then, in each video frame, the log-likelihood that a pixel belongs to the background is computed. The statistical based approaches have a better performance compared to the non-statistical based approaches for modeling background of the outer scenes. They may require more memory and processing time and hence statistical based approaches are slower than the non-statistical based approaches.

One of the important statistical-based approaches to model the background image is the Gaussian mixture model. This approach uses mixture of models (multi- models) to represent the statistics of the pixels in the scene. The multimodal background modeling [17] can be very useful in removing

repetitive motion, for examples shining water, leaves on a branch. This model estimates the probability distribution function of observing a specified value for a pixel in its previous values obtained from older frame sequences. [18–20] papers presented Temporal difference techniques; allow the detection of the Active parts of a scene foreground by comparing two consecutive frames.

For Real-time video Segmentation with large frame size, dedicated hardware architectures are required. A list of Hardware processors have been proposed in [21]–[27]. Papers [21-22] present GPU implementations based on GMM algorithm. In the Papers [23]– [27] proposed FPGA implementations Video segmentation Architectures that are most suited to embedded and low power systems. The Video segmentation circuit of FPGA [23] is improved in [24-25]. The Reference [26] proposed a Open CV GMM algorithm implemented on Virtex5 FPGA and the circuit of [26] performance is improved in [27].

In this paper first analyzed foreground detection method for sequences of Frames with GMM process using Matlab and which is implemented in ASIC-FPGA. The proposed system allows real-time processing for a video stream with resolution of 720 × 576 pixels (10-20 fps) and improves the memory throughput utilizing a memory reduction scheme.

The paper is organized as follows. Sections II discuss the original algorithm. The hardware architecture is presented in Section III, Brief Introduction to ASIC-FPGA and design scenarios. Section IV explains the GMM Background subtraction Implementation in ASIC - FPGA Steps in detail. Finally, the results and conclusions are covered in Sections V and VI.

2. BACKGROUND MODELING USING MIXTURE OF GAUSSIANS

Adaptive GMM approach can handle challenging situations such as sudden or gradual illumination changes, slow lighting changes, long-term scene changes periodic motions from a cluttered background and repetitive motions in the clutter, etc. In this method a different threshold is selected for each pixel. These pixel-wise thresholds are adapting by time. Objects are allowed to become part of the background without destroying the existing background model. In order to meet the needs for modeling multi-modal pixel process, a mixture of Gaussian distributions are assigned to each one of the pixels in the image sequences. The whole procedure is formulated as follows. A pixel process, defined as an collection of most recent measurements I (x, y, t), where I is the image sequence, can be viewed as a mixture of several independent noise processes and thus modeled using the sum of Gaussian distributions with weighting factors. The probability of observing the current pixel value is

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \; \eta\left(\mathrm{X_t}\middle|\mu_{\mathrm{i,t}}, \Sigma_{\mathrm{i,t}}\right) \tag{1}$$

Where $\omega_{i,t}$ is the i[th] Gaussian mixture weight and $\eta\,(\mathrm{X_t}, \mu_{\mathrm{i,t}}, \Sigma_{\mathrm{i,t}})$ the component Gaussian densities stated by

$$\eta\left(\mathrm{X_t}\middle|\mu_{\mathrm{i,t}}, \Sigma_{\mathrm{i,t}}\right) = \Phi\left(x_i\middle|\Theta_j\right) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_j|^{\frac{1}{2}}}\, exp\left\{-\frac{1}{2}\left(x_i-\mu_j\right)^T\Sigma_i^{-1}\left(x_i-\mu_j\right)\right\} \tag{2}$$

Where K is the number of Gaussian distributions and ω is the weighting factor. The mixtures of Gaussian distributions are ordered according to ω/σ in decreasing order. The portion of Gaussian distributions considered as background process is defined by

$$B = arg\,\min_b \sum_{k=1}^{b} \omega_k > H \tag{3}$$

Where H is the predefined parameter determining how much proportion of the mixture distributions for the background process. When H parameter is small, the pixel process becomes a single adaptive Gaussian distribution. In the process each new pixel value is verified with the Gaussian mixture to update the parameters of each Gaussian,. When a match is found, the weight, mean and variance values of the matched distributions are updated as follows:

$$\omega_{k,t} = (1-\alpha)\omega_{k,t} + \alpha$$

$$\mu_{i,t+1} = (1-\rho)\,\mu_{i,\,t} + \rho X_{t+1}$$

$$\sigma^2{}_{\mathrm{i},t+1} = (1-\rho)\sigma^2{}_{\mathrm{i},t} + \rho(X_{t+1}-\mu_{i,t+1})^T(X_{t+1}-\mu_{i,t+1}) \tag{4}$$

Where α, ρ is the learning factor $\rho = \eta\,(X_t|\,\mu_{i,t}, \Sigma_{i,t})$. A match is defined as a pixel value within 2.5 standard deviations of a distribution. For those unmatched, the weight is updated according to

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} + \alpha (M_{i,t+1}) \quad (5)$$

Where $(M_{i,t+1})=1$ for the Matching Gaussian and 0 for all others.

If a pixel $I_{(i,t,x)}$ does not matches with any one of the back- ground component, then the pixel is marked as foreground. The GMM suffers from slow learning, if a small value of learning rate is chosen, especially in busy and fast changing environments. In addition, a moving shadow region may be wrongly classified as foreground. The problem aggravates further in presence of a quick illumination change such as light switched on/off and can increase the number of falsely detected foreground pixels. Consequently this may give erroneous results in tracking and classification. If the position of the background object is changed or any new object is brought into the background. The pixel value of the new background will not match with the estimated K Gaussian and will be classified as foreground. These small changes are of little interest and should be made part of the background. Otherwise a large percentage of image will be classified as foreground. The higher the value of H in (3), the higher is the probability of a multi - modal background.

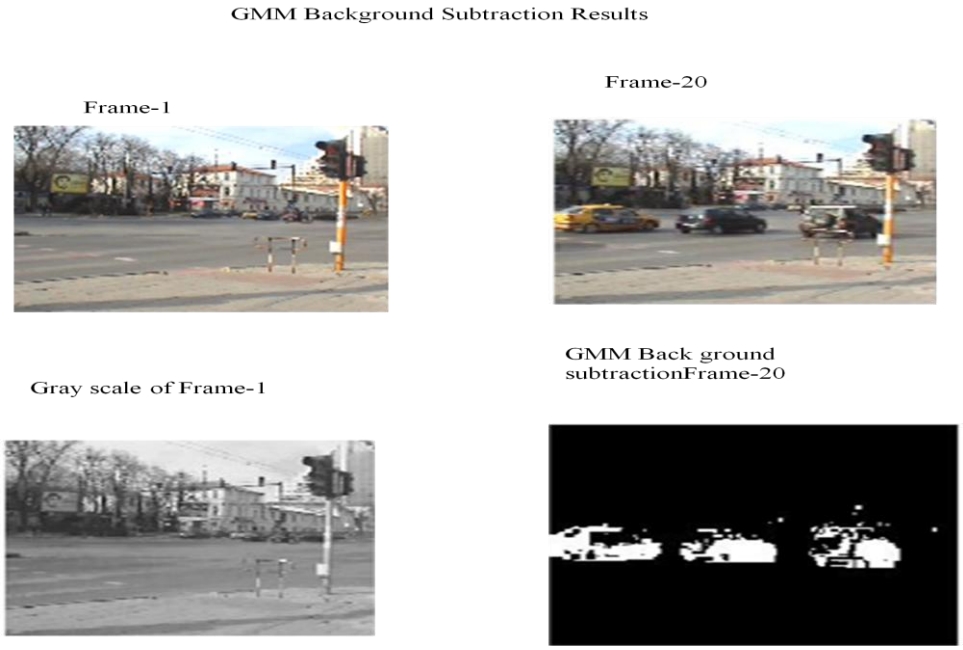

Figure1. *GMM Background Subtraction output*

2.1. Proposed System Architecture Design and Segmentation Architecture

The Fig. 2 shows a overview of a Background Subtraction system. A CMOS sensor captures each frame of the video sequence and a CMOS interface gives the pixel values to the Background circuit. The Background identification circuit processes the luminance value of the input "Pixel," and the "Statistical Model" of the pixel for the given frame. The output data are the "Updated Statistical Model" and the "Foreground /Background" mentioned as FG/BG. For each pixel, the Gaussian parameters are read from an external memory and the updated parameters are stored in the memory. For each frame of the input video sequence, the Foreground /Background produce a binary image that can be displayed on a monitor.

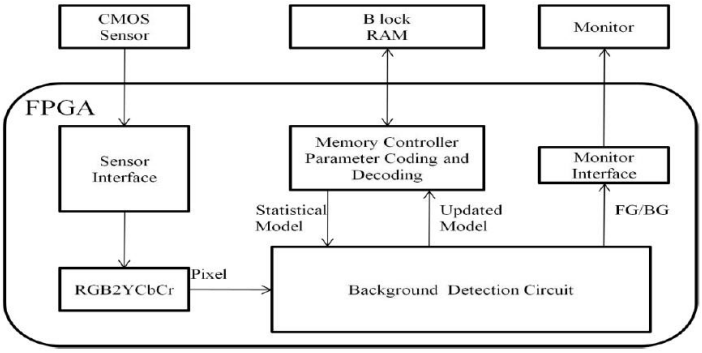

Figure2. *System Architecture for Background Detection circuit*

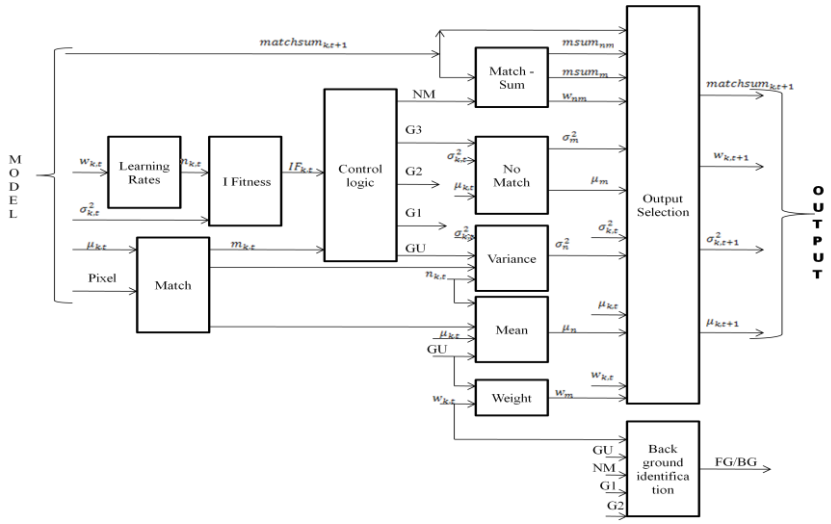

Figure3. *Segmentation Architecture*

The GMM algorithm reveals that the signals have limited dynamics range. Mean, variance, and weight range in [0,255], [0,127], and [0, 1], respectively. When dealing with limited dynamics signals, a fixed point representation is used instead of a floating point to improve performances while reducing hardware complexity and the bandwidth toward the memory. However, a reduced number of bits per signal could involve a poor quality of the processed images. In this paper, a fixed point representation of the numbers is used and, in order to accurately determine the representation of the signals, the weights of the MSB have been fixed in accordance to the signals dynamics while the weights of the LSB have been chosen by using a word length reduction procedure directed to optimize the tradeoff between accuracy and hardware complexity.

3. ASIC-FPGA DESIGN FLOW

FPGAs have undergone significant architectural changes in the last few years. Beginning with hard blocks such as RAM and DSP, FPGA's now also include transceivers and hard IP blocks, such as Ethernet and PCI Express. With these new functional blocks, FPGA designers can now create complex designs. However, these designs can sometimes push the cost, power and performance specification requirements of the targeted FPGA device. In consumer and mobile applications, for example, low power and low cost are essential, so designers may have to:

- Explore different tool settings; for example, extra effort settings for routing or placement to try to improve performance or reduce area.
- Make design changes, such as selecting Block RAM over distributed RAM for performance.
- Change the architecture of the design, for example, selecting a parallel, rather than serial, protocol. These design approaches are typically explored sequentially. As a result, overall compile time is increased, which can create issues with the schedule.

3.1. Traditional Design Flow Issues

Using the traditional design flow, FPGA designers often struggle to meet targeted design specifications without affecting the schedule. This is because the traditional design flow was conceived to be fundamentally sequential: each time the user makes a change to the design, it needs to be recompiled assess its impact. This process is repeated until the specifications are met. This approach can extend the schedule and a solution is needed that will reduce overall compile time within the design flow. FPGA design software introduced a tool called '**settings explorer**', which allows the designer to select optimization settings, such as retiming or fan-out control, for the design and to then archive the results of the entire space; all runs done by exploring the different settings or to save only the run with the best results. The designer can let the tool select the settings automatically based on some high level goals, such as 'design for lower power' or 'reduce area of the design' and explore the entire space. While this feature improved the ability to meet targets, it did not alleviate the schedule pressure completely, since any change to the design could cause the user to launch the tool again and cause a lengthy compile time.

Another innovation allowed FPGA design software to use any number of cores to reduce compile time. However, changes to the design still required another compile, which added to the total elapsed time, even if the compile time of one design iteration had been reduced.

3.2. Incremental Design Flow

FPGA design software has borrowed from ASIC design methodology and introduced an incremental design flow .In this flow, users can partition their designs based on logical hierarchies for runtime reduction and timing preservation. Using this approach, users can create a partition on the logical hierarchies where design changes could occur and would need to be recompiled. This approach can help to reduce overall compile time while preserving the performance of the rest of the design. While a major step in the right direction; this approach does not address the sequential nature of the design flow. Users can have only one implementation of the design active at a given time and need to wait to for two compiles to be run sequentially in order to compare the results. What is needed is a more radical change in the FPGA design flow, so users can compile multiple implementations in parallel and compare the results of two implementations after one compile, rather than after two sequential compiles. Users then could reject or accept the changes quickly and with limited impact on their schedule.

3.3. Design Scenarios

Consider an example where the user has to change the design implementation from using block ram to distributed ram. Block RAM is ideal for storing operations on coefficients in DSP centric designs because it provides faster throughput. If the user wants to make such a change, they need to complete two runs sequentially before the impact of this major change can be assessed. However, with multiple implementations run in parallel, the impact can be seen more rapidly. Another example where running multiple implementations in parallel is valuable is when the user changes the architecture of the design. A typical case is in high speed mobile applications, where data traffic management is changed from a serial to a parallel implementation. With such a fundamental change to the FPGA design flow, users can speed their schedule or, at least, reduce schedule pressures and improve productivity.

In a world of fast changing applications, meeting time to market schedules is critical and designers are always under pressure to deliver their designs faster. In turn, designers are asking for faster compile times from their FPGA design software tools. Run Manager can provide them with that competitive advantage, leveraging the computer architecture and features in Lattice Diamond. Multiple implementations with an RTL file for the same design allows designers to compare their results quickly and improve productivity.

3.4. FPGA-KINTEX-KC705 Block Diagram and Description

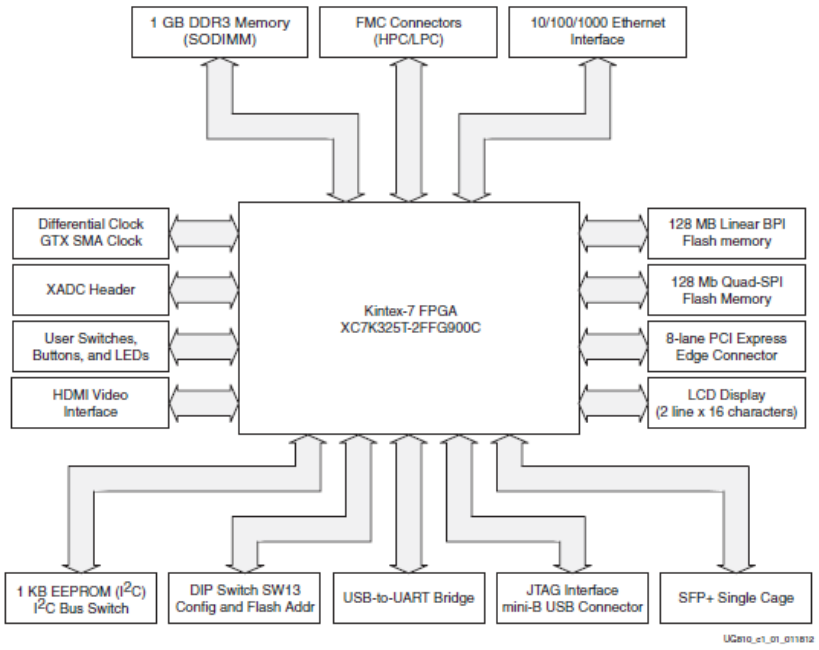

Figure4. *Kintex-7KC705 Block Diagram*

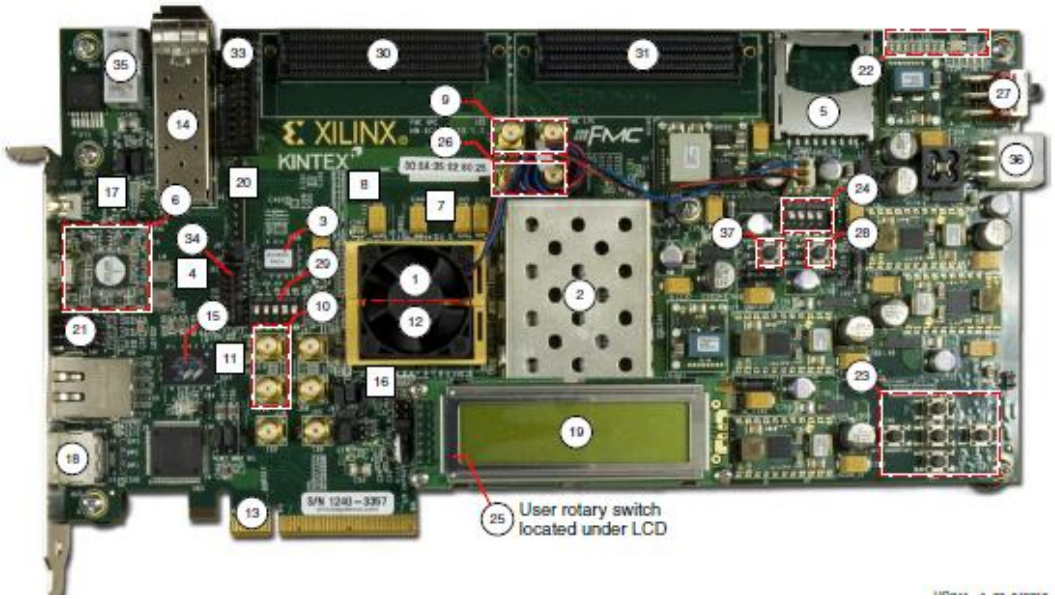

Figure5. *Kintex-7 Board View*

In terms of fabric speed, Kintex7 and Virtex7 are the same and slightly faster than Artix7. The main difference between Kintex7 and Virtex7 is that the latter have additional and faster Serdes blocks. Any of these chips should be significantly faster than a Virtex4. Kintex-7 FPGAs are built to provide optimal price performance at the lowest power to meet requirements for key applications. At the same time, the Kintex-7 FPGA family leverages the unified architecture shared across the 7 series, 28nm device families to enable customers to begin FPGA development now for designs that may ultimately migrate to Artix-7 and Virtex-7 FPGAs.

The Kintex-7 devices are offered in conjunction with the Xilinx ISE Design Suite 13, AMBA 4 Advanced Extensible Interface (AXI) bus protocol-compliant IP, and targeted reference designs. All of these Targeted Design Platform components run on the new Kintex-7 FPGA KC705 evaluation board currently being demonstrated for customers so that designers can evaluate the power consumption, performance, and capabilities of the new Kintex-7 K325T devices. Xilinx delivered the new devices in less than 90 days from tape-out by leveraging TSMC's 28nm High-Performance, Low-Power (HPL) process that relies on proven design and manufacturing methodologies. You can see a video demonstration of the device

4. Implementation Procedure GMM Background Subtraction on ASIC-FPGA and Performance Evaluation

Hardware Implementation Steps:

1. Analysis of the GMM Background subtraction of Video Segmentation in Matlab 2015a
2. GMM with EM Converted into C-code Imported the GMM background subtraction of C-code into Vivado HLS (2014.2)
3. By using Vivado HLS (2014.2) target to Kintex-7KC705 (xc7k325tffg900-2)
4. By comparing the C-libraries with HLS libraries, synthesized the C- code in Kintex-KC705 (xc7k325tffg900-2)
5. Analyzed the synthesis report and generated the report.

The proposed FPGA implementation of the GMM algorithm is synthesized and implemented on Kintex-7KC705 (xc7k325tffg900-2) Xilinx ASIC-FPGA. In order to compare the proposed circuit with previous art, the circuit has also been implemented on Virtex-II Pro (xc2pro30). The synthesis for to Kintex-7KC705 (xc7k325tffg900-2) has been carried out by using Vivado HLS(2014.2).The performances analysis for both circuits have been conducted, including input and output registers that synchronize the circuits and provide timing performances that are not dependent on the I/O pads. The circuit of Fig. 3, implemented on Kintex-7KC705 (xc7k325tffg900-2) without pipeline levels, is able to process 20 fps. The proposed FPGA implementation, as shown in Table I and discussed in the following, outperforms the previous art.

Table1. *Performances evaluation Proposed* GMM *background subtraction implemented on ASIC-FPGA and comparison with previous papers Results.*

TARGET FPGA	*circuit*	*Pipe-line Levels*	*LUT*	*Flip Flop*	*Slice*	*BRAM*	*Frequency (Mhz)*	*Frames per sec*	*Energy per pixel nJ/pixel*
KINETX-7	**Pro-posed**	**N.A**	**18/ 203800**	**13470/407600**	**431/50950**	**4 /4000**	**0.123798**	**20**	**20**
VIRTEX6 XC6VLX75T	Previous [28]	0	692/46560	0/93120	279/11640	0/156	85.7	41	1.01
VERTEX5 XC6VLX50	[28]	0	742/28800	0/28800	303/7200	0/48	79.5	38	1.03
VERTEX5 XC6VLX50	[27]	0	1066/28800	0/28800	346/7200	0/48	50.5	24	4.60
VERTEX5 XC6VLX50	[26]	0	1572/28800	0/28800	N.A	0/48	47.0	22	5.87
VIRTEXII XC2PRO30	[28]	0	1305/27392	0/27392	747/13696	0/136	44.7	21	26.72
VIRTEXII XC2PRO30	[25]	N.A	N.A	N.A	1743/13696	N.A	N.A	N.A	N.A
VIRTEXII XC2PRO30	[24]	N.A	3397/27932	N.A	N.A	13/136	83.0	40	N.A

In reference [26] proposed an implementation of the Open-CV GMM algorithm on Virtex5 xc5vlx50 FPGA and The circuit of [26] is improved in [27]. The circuit proposed in [27] modifies the GMM updating equations and resulting image quality is not reduced. With respect to [28], the proposed ASIC- FPGA of Model Kintex-7KC705(xc7k325tffg900-2) using Vivado HLS(2014.2) implementation reduces the bandwidth toward the memory, implemented without pipeline levels, increases the processing capability, reduces the area utilization, and reduces the memory bandwidth above 80% . Papers [24] and [25] proposed FPGA implementations of the GMM with the Open-CV GMM algorithm. In [24], the design of a digital surveillance system running in real-time on an embedded platform is presented. The segmentation unit of the circuit proposed in is able to run at 83 MHz on Virtex-II xc2pro30 using 3397 of the 27932 available LUT and 13 of the 136 available BRAM and the implementation of represents the Gaussian parameters using 140 bit per pixel, resulting in a higher memory bandwidth (Table I). The circuit of [24] is improved in [25]. In [25], the number of used Slices for the segmentation unit is 1743 while the proposed circuit, as shown in Table I, uses 431 slices when implemented without pipeline. With reference to the bandwidth toward the memory, as in [24] and [25] uses 140 bit for each pixel but the applied compression technique is able to reduce the bandwidth up to 75%.

5. CONCLUSION

In this work we presented a real-time video segmentation algorithm based GMM background subtraction. The main feature of our approach is reduces the bottle necks of hardware implementation with memory utilization is by 83% with capable of 20 fps at VGA Resolution. The reduction of Memory utilization can be achieved by the Xilinx ASIC-FPGA Kintex-7KC705 (xc7k325tffg900-2) platform with Vivado HLS (2014.2).

REFERENCES

[1] M. Genovese and E. Napoli, "An Improved Mixture-of-Gaussians Background Model with Frame Difference and Blob Tracking in Video Stream". Hindawi Publishing Corporation, The Scientific World Journal Volume 2014, Article ID 424050, 9 pages http:// dx. doi. org/ 10. 1155/ 2014/424050

[2] C. S. Kamath and C. Robust, "Background subtraction with foreground validation for Urban Traffic Video," *J. Appl. Signal Proc. Special Issue on Advances in Intelligent Vision Systems: Methods and Applications (EURASIP 2005)*, New York, USA, vol. 14, pp. 2330-2340, 2005.

[3] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," *Int Conf on Computer Vision, (ICCV 1999)*, Corfu, Greece, pp. 255-261, September 1999.

[4] J. Carranza, C. Theobalt, M. Magnor, and H. Seidel, "Free-Viewpoint Video of Human Actors," *ACM Trans on Graphics*, vol. 22, no. 3, pp. 569-577, 2003.

[5] T. Horprasert, I. Haritaoglu, C. Wren, D. Harwood, L. Davis, and A.Entland, "Real-time 3D motion capture," *Workshop on Perceptual User Interfaces (PUI 1998)*, San Francisco, California, pp. 87-90,November 1998.

[6] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman, "Human body model acquisition and tracking using voxel data," *Int J Comp Vision (IJCV 2003)*, pp. 199 -223, July 2003.

[7] El Baf F., T. Bouwmans, and B. Vachon, "Comparison of background subtraction methods for a multimedia learning space," *Int Conf. on Signal Processing and Multimedia (SIGMAP 2007)*, Barcelona, Spain, July 2007.

[8] A. Pande, A. Verma, and A. Mittal, "Network aware optimal resource allocation for e-learning Videos," *The 6th Int Conf. on mobile Learning*, Melbourne Australia, October 2007.

[9] J. Warren, "Unencumbered full body interaction in video games," thesis, MFA design and technology parsons school of design, New York, USA, April 2003.

[10] D. Semani, T. Bouwmans, C. Frélicot, and P. Courtellemont, "Automatic fish recognition in interactive live videos," in *Proc. of the IVRCIA 2002*, Orlando, Florida, USA, vol. 14, pp. 94-99, July 2002.

[11] M. Piccardi, "Background subtraction techniques: A review," in *Proc of the Int Conf on Systems, Man and Cybernetics (SMC 2004)*, The Hague, The Netherlands, pp. 3199-3104, October 2004.

[12] P. Blauensteiiner, H.Wildenauer, A.Hanbury and M. Kampel "On color Spaces for change Detection and shadow suppression ," proceedings of the 11th Computer Winter Vision Workshop (CVWWS6), 6-8 February 2006, pp. 117-123.

[13] Erum A. Khan and E. Reinhard, "A Survey of Color Spaces for Shadow Identification," APGV '04: Proceed-ings of the 1st symposium on Applied perception in graphics and visualization, New York, 2004, p.160.

[14] M. Piccardi, "Background Subtraction Techniques: A Review," IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, 2004, pp. 3099-3104.

[15] A. M. Elgammal, D. Harwood and L. S. Davis, "Non- Parametric Model for Background Subtraction," Proceedings of ECCV 2000, 2000, pp. 751-767.

[16] T. Horprasert, D. Harwood and L. S. Davis. "A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection," Proceedings of IEEE ICCV'99 FRAME-RATE Workshop, 1999, pp. 1-19.

[17] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao and S. Russell, "Towards Robust Automatic Traffic Scene Analysis in Real-Time," Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing, Vol. 1, 1994, pp. 126-131.

[18] Z. Chaohui, D. Xiaohui, X. Shuoyu, S. Zheng, and L. Min, "An improved moving object detection algorithm based on frame difference and edge detection," in *Proc. 4th Int. Conf. Image Graph.*, Aug. 2007, pp. 519–523.

[19] R. Mingwu and S. Han "A practical method for moving target detection under complex background," *Comput. Eng.*, vol. 31, no 20, pp. 33–34,Oct. 2005.

[20] X. Benxian, L. Cheng, C. Hao, Y. Yanfeng, and C. Rongbao, "Moving object detection and recognition based on the frame difference algorithm and moment invariant features," in *Proc. 27th Control Conf.*, Jul. 2008, pp. 578–581.

[21] P. Carr, "GPU accelerated multimodal background subtraction," in *Proc. Digital Image Comput., Tech. Appl.*, Dec. 2008, pp. 279–286.

[22] V. Pham, P. Vo, V. Thanh Hung, and L. H. Bac, "GPU implementation of extended gaussian mixture model for background subtraction," in *Proc. IEEE Int. Conf. Comput. Commun. Technol. Res. Innov. Vis. Future*, Nov. 2010, pp. 1–4.

[23] H. Jang, H. Ardö, and V. Öwall, "Hardware accelerator design for video segmentation with multi-modal background modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2. May 2005, pp. 1142–1145.

[24] F. Kristensen, H. Hedberg, H. Jiang, P. Nilsson, and V. Öwall, "An embedded real-time surveillance system: Implementation and evaluation," *J. Signal Process. Syst.*, vol. 52, no. 1, pp. 75–94, Jul. 2008.

[25] H. Jiang, H. Ardö, and V. Öwall, "A hardware architecture for real-time video segmentation utilizing memory reduction techniques," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 19, no. 2, pp. 226–236, Feb. 2009.

[26] M. Genovese, E. Napoli, and N. Petra, "Open-CV compatible real time processor for background foreground identification," in *Proc. Int. Conf. Micro electron.*, Dec. 2010, pp. 467–470.

[27] M. Genovese and E. Napoli, "FPGA-based architecture for real time segmentation and denoising of HD video," *J. Real Time Image Process.*, pp. 1–13, 2012.

[28] Mariangela Genovese and Ettore 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 (VLSI) systems*, vol. 22, no. 3, march 2014.,pp.537-547.