FPGA based Computing Displacement of Moving Object in a Real Time Video

Kota Solomon Raju
DSG, CSIR- Central Electronics Research Institute (CSIR-CEERI), Pilani, Rajasthan, India -333031
+91-9460842188
Kota_solomonraju@yahoo.uk

Gargi Baruah
Dept. of Electronics & Communication Engg. Tezpur University Naapam, Assam, India-784028
+91-9509774812
gargi369@gmail.com

Manipati Rajesham
DSG, CSIR- Central Electronics Research Institute (CSIR-CEERI), Pilani, Rajasthan, India -333031
+91-7891400797
rajash.manipati@gmail.com

Palash Phukan
Dept. of Electronics & Communication Engg. Tezpur University Naapam, Assam, India-784028
+91-9435358807
phukan.palash@gmail.com

ABSTRACT

This paper describes and focuses on the implementation of computation of displacement of a moving object in a real time video using EDK on FPGA platform. This will be used to track the object where the object will be in the consecutive frames. The entire displacement algorithm has been implemented in the Avnet video processing board based on Spartan 6 device S6LX150T. The Xilinx Spartan family has the ability for partial reconfigurability and thus can be used in real time video processing.

Keywords: FPGA, Xilinx, EDK, XPS, Bhattacharya Coefficient, pixel values, tracking, real time video.

1. INTRODUCTION

The integration of real-time calculation of the displacement of a moving object is a challenging task. This displacement calculation serves a crucial point in tracking moving objects. To date, many tracking methods [1] have been proposed such as Kalman Filter[1], Extended Kalman Filter (EKF) [2], Particle Filter [1] with varying characteristics, such as tolerance to illumination and geometric variations and computational complexity. These techniques [3], however, still assume the availability of modest processing power, memory, floating-point capabilities, etc., rendering them inappropriate for constrained real-time implementations.

In fact image processing is difficult to achieve on a serial processor.[3] because large parallel data set required to represent the image and the complex operations that need to be performed on the image[4]. If we consider video rates of 30 frames per second, a single operation performed on every pixel of a 1280 by 720 color image (HDMI) equates to 663 million operations per second. Many image processing applications require that several operations be performed on each pixel in the image resulting in an even large number of operations per second. Thus the perfect alternative is to make use of an FPGA [4]. Continual growth in the size and functionality of FPGAs over recent years has resulted in an increasing interest in their use for image processing application.

The main advantage of using FPGAs for the implementation of image processing applications is because their structure is able to exploit spatial and temporal parallelism [5]. FPGA implementations have the potential to be parallel using a mixture of these two forms. For example, the FPGA could be configured to partition the image and distribute the resulting sections to multiple pipelines all of which could process data concurrently. Such parallelization is subject to the processing mode and hardware constraints of the system. There is also the presence of Xilinx Embedded Development Kit (EDK) tools can make it possible to implement a complete digital system on a single FPGA using hardware/ software design methods.. A developer could use one chip for different tasks and switch between them during runtime. The Xilinx Spartan family is very suitable for all these operations.

In this paper we propose a low cost FPGA system for displacement calculation of a moving object in a real time video. First, we present a high performance calculation method using the Mean shift tracking algorithm [6] which, while being robust to illumination and geometric variations, entails a low computational load. This makes it suitable for simple microprocessor as well as custom hardware implementations. We then present an FPGA realization of this method. In recent years, FPGAs have proven to be invaluable in image processing applications [4] because they combine the reconfigurability advantage of general purpose processors with the parallel processing and speed advantages of custom hardware. Our proposed FPGA based displacement calculation combines very low logic and memory costs with high processing rates. Here we have used Spartan 6 LX150T for implementing our proposed method.

ICACCI '12, August 03 - 05 2012, CHENNAI, India

2. DISPLACEMENT CALCUTAION METHOD

The problem of displacement calculation for moving objects in various frames has been an active area of study for some time. For it numerous methodologies exist with the help of tracking methods such as point based, kernel based and silhouette based [1]. For our computation we have used kernel based mean shift method and the feature space is the colour values i.e. RGB [7] The advantage of colour-based computation methods is that, while computationally efficient, they are generally robust to geometric transformations, such scale, orientation and viewpoint changes, since such transformations do not affect the colour, as well as to complex backgrounds and illumination variations. In our previous work, the detailed simulation has been shown in [9]

The first step is the quantization of the RGB space. If a bin is used for all possible colours for a 24 bit frame, then there will be 256*256*256= 16 million bins [8,9]. Thus the feature space is quantized to 16*16*16=4096 bin values. We have then chosen a target window of 160x80 of the 1280x720 frame and used the calculations described below

From the literature [6], [7],[9] **object model** Probability Density Function is given by

$$Qu=C\sum_{i=1}^{n} C|k(||xi||^2)\partial(b(xi)-u) \quad (1)$$

Target model [6],[7],[9] for target centred at y is given by

$$p(y)=Ch\sum_{i=1}^{nh}\left|(|\left|\frac{y-yi}{h}\right||^2\right)\partial(b(yi)-u) \quad (2)$$

The new position y1 is given by applying Mean Shift which goes as

$$y_1=\frac{\sum_{i=1}^{m} x_i * w_i * g(\left\|\frac{y_0-x_i}{h}\right\|^2)}{\sum_{i=1}^{m} w_i * g(\left\|\frac{y_0-x_i}{h}\right\|^2)} \quad (3)$$

Now the similarity function [7] defines a distance among target model and candidates. When similarity increases Bhattacharya coefficient increases.

The displacement [8] between two discrete distributions as

$$d(y)=\sqrt{1-\rho[\vec{p}(y),\vec{q}]} \quad (4)$$

But this displacement decreases when similarity increases. For further reference refer literature [9].

Figure 1 shows the flowchart for the of the software code implemented in EDK for displacement calculation. The full flow of this displacement flowchart can be obtained from Ref.[9].

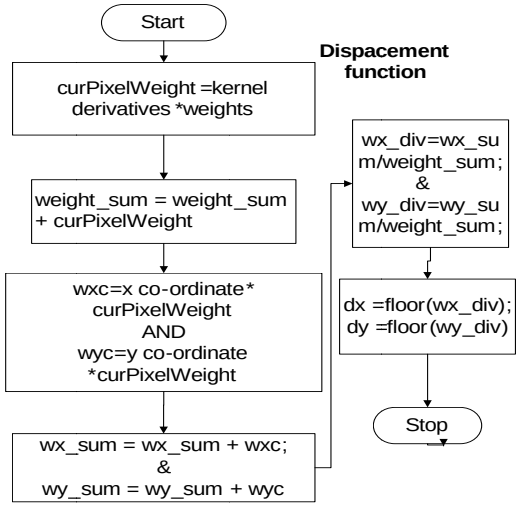

Figure 1. Flowchart for displacement calculation

3. FPGA-BASED SYSTEM

The top level organization of the FPGA system which implements the displacement calculation method described in section 2 is shown in Figure 2 and 3. Thus, at the top level, the proposed FPGA-based system is a system consisting of six stages, namely video capturing, memory partitioning, writing the video in VFBC, processing this active data, transferring this processed data to the read VFBC memory and finally displaying it.

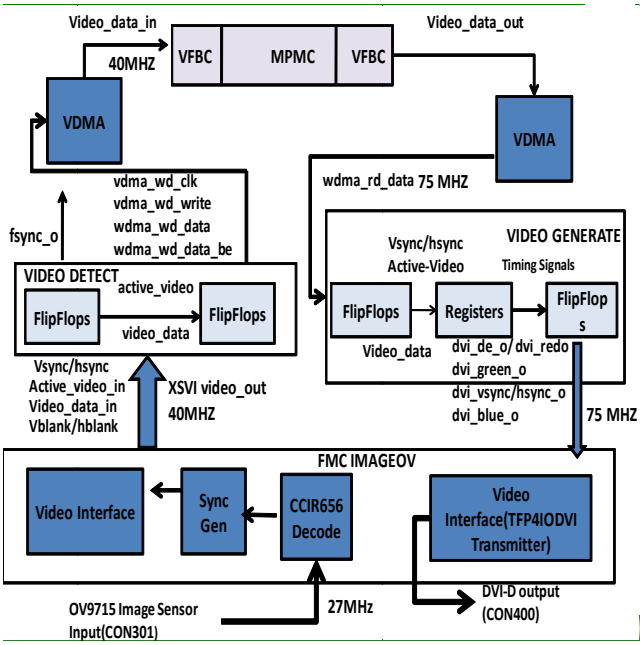

Figure 2 Video Signal Detection & Generation block diagram

The video input from image sensor OV9715 source enters the Camera Input PCORE [10]. This PCORE decodes the BT656 codes with the help of CCIR656 to generate synchronization signals and formats the video as an XSVI bus interface. The Video Detect PCORE monitors the VSYNC and ACTIVE_VIDEO signals to determine the dimensions of the active video streaming through the FPGA. The Video DMA PCOREs with the Video Frame Buffer Controller (VFBC) [10] interfaces on the Multi-Port Memory Controller (MPMC) and perform the actual transfer of active video data to/from external memory.

In order to process the video data from the data it is very important for the read and the write frame buffer memory address to be different. Hence memory partitioning is done and after the extraction of RGB values, the video data is then passed on to the read frame buffer memory address in DDR.

From the Read VFBC the video data goes to the Video Generate PCORE which again generates timing signals required for an XSVI bus. It is then forwarded to the DVI output PCORE. The DVI Output PCORE takes an XSVI bus interface as input and optionally drives the pins of the DVI output interface. This output to the FMC connector will only be driven once the FMCIMAGEOV module has properly been identified.

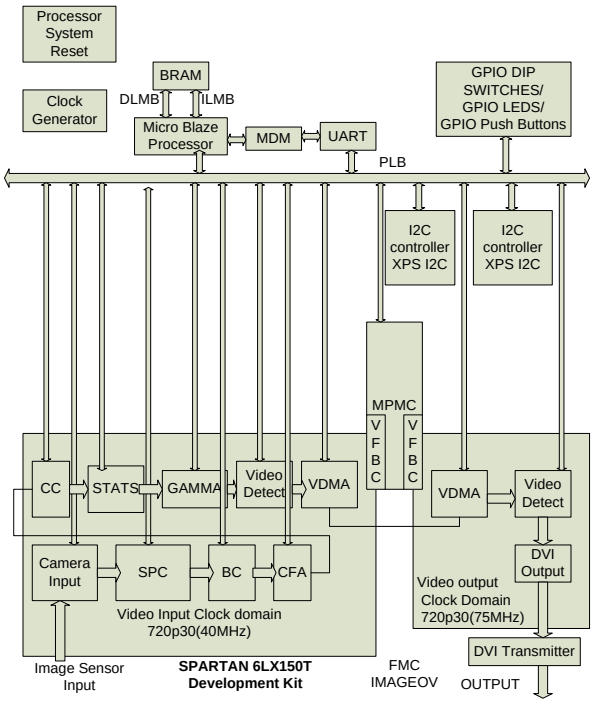

Figure 3 Camera Frame Buffer – Video Pipeline

The video resolution used is 1280x720P @ 30Hz .These resolutions are configured by the embedded processor (Micro Blaze) and can be modified to support other resolutions (limited by the image sensor used).

Thus the software and the hardware part is merged using EDK, and bit stream is generated which is dumped into the FPGA .Spartan 6 LX110T with speed grade -3. Figure 4 shows the board and the terminals used for this process.

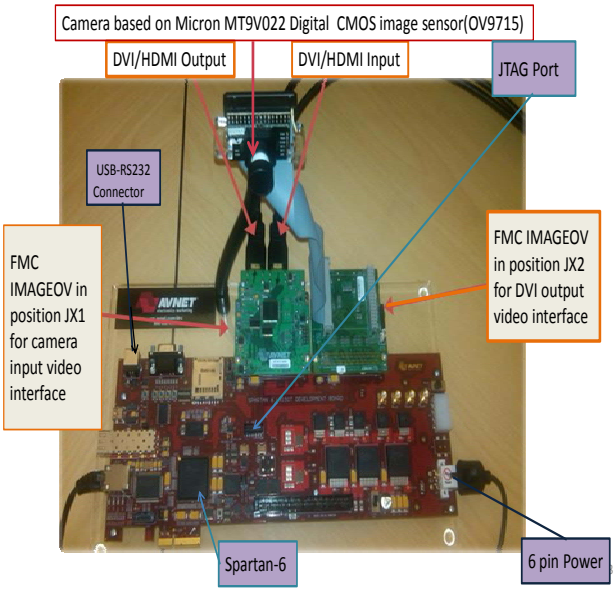

Figure 4: Call out diagram of Spartan 6 IVK board

4. EXPERIMENTAL RESULTS

The architecture of the displacement computation method is shown in Figure 4

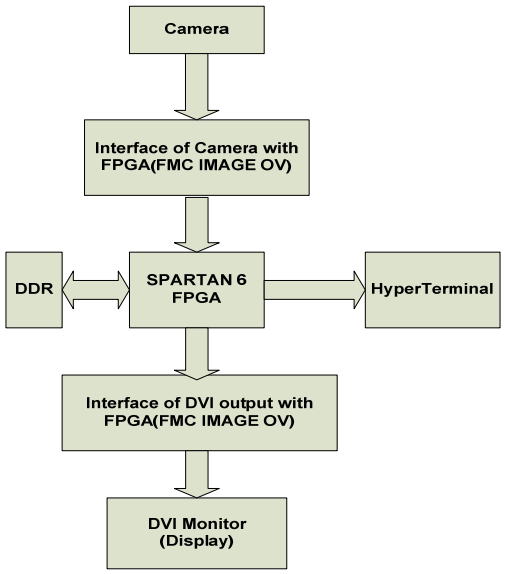

Figure 5: Architecture of the Displacement Computation Block

After the base platform is created and the IPs are added (according to the architecture shown in Figure 4,)using EDK 12.2, a block diagram of the embedded system is generated in EDK which is shown in the figure 5**.**

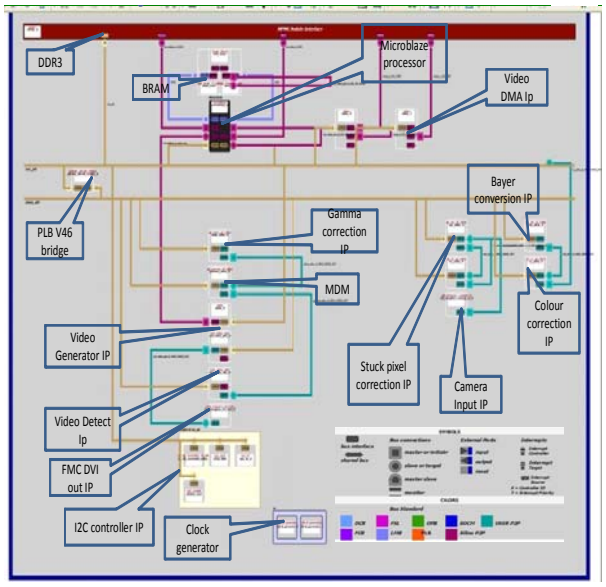

Figure 6: Generated Block diagram after implementation.

The data that was obtained from the output in HyperTerminal window using the EDK tool were used in Matlab 7.8.0 (R2009a) after which the following graphs were obtained.

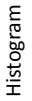

Figure 7: Histogram of a 160 X 80 target window of 1280X720 frame (1st frame)

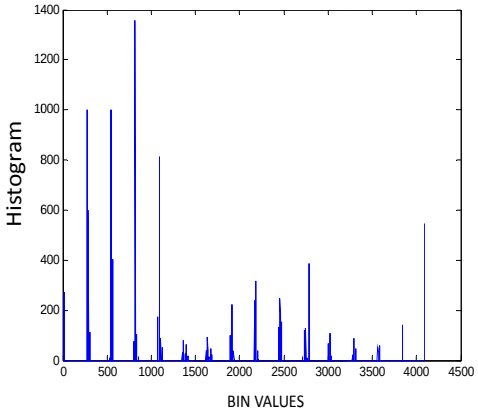

Figure 8: Histogram of a 160 X 80 target window of 1280X 720 frame (3rd frame)

Figure 7 and 8 shows the histogram plot of the 160x90 target window of the first 3 consecutive 1280x720 frames os a real time video. The data was obtained in real time.

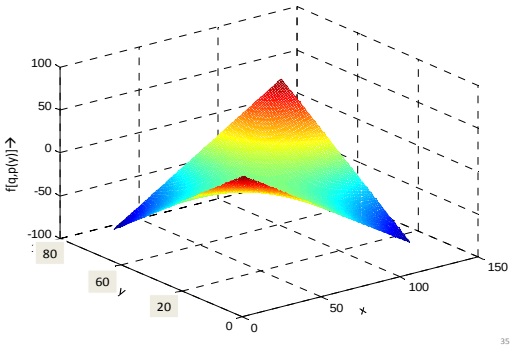

Figure 9: Plot of the similarity function,

The data obtained was calculated for histogram and then calculated for the similarity function for the equation.(5)

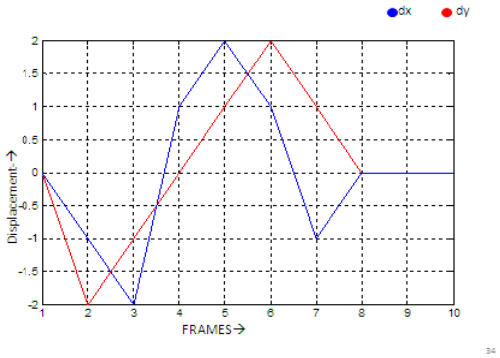

Figure 10: Displacement of the target object in 10 Consecutive frames

Finally, the displacement is calculated from the data obtained in the HyperTerminal window and the equation (6) and is shown in figure 10.

Table: 1 The table shows the amount of resources used and its percentage utilization.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	11,437	184,304	6%
Number of Slice LUTs	10980	92,152	11%
Number used as memory	846	21,680	3%
Number of bonded IOBs	95	386	23%
Number of BUFG/BUFGCTRLs	10	32	31%
Number of DSP48Es	18	128	4%

The earlier work for real time displacement calculation was done in MATLAB [11],[12]. But here we have developed a code in EDK in C for developing an embedded system. This work will further lead to make a reconfigurable embedded system by applying the concepts of hardware-software co-design, reconfiguration of the hardware modules. Decision for which IPs should be used for parallel processing (hardware blocks) and which should be left as sequential (software flow). The latency factor will decide this software/ hardware partition. We have implemented this code using all the RGB channels whereas in previous works they have used either one colour channel or converted RGB images into 8 bit grayscale images. Also tracking in real time is a tedious process in software. Here our target is to develop an embedded system for real time moving objects.

6. CONCLUSION AND FUTURE WORK

In this paper we have explored the use of variable kernels to enhance a weighted histogram and then compute the displacement of an object in the video frames which can be used for various tracking and other video processing algorithms. The main advantage is that a system has been developed which is not only accurate but its computation is very high compared to other software platforms since in EDK 100% bit stream is generated. Also the use of resources is very less.

Future work is currently underway to extend our testbed platform for tracking of objects in real time by developing new hardware for various image processing algorithms, complementing our motion-tracking algorithm by adding further improved calculations and developing custom IPs for parallel processing.

ACKNOWLEDGMENT

The work reported in this paper is a part funded by CSIR-Supra Institutional Project.(SIP 21). We would like to thank Dr. Chandra Shekhar, Director CSIR-CEERI, Pilani, Dr. P. Bhanu Prasad, Chief Scientist, DSG Group CSIR-CEERI,Pilani, Mr. Manoj Pandey PhD Scholar and Mr. Vaibhawa Mishra, QHF for their help and support.

REFERENCES

[1] Alper Yilmaz, Omar Javed, Mubarak Shah," **Object Tracking- A Survey",** ACM Computing Surveys, Vol. 38, No. 4, Article 13, pp.145-190, December 2006

[2] Welch & Bishop**, "An Introduction to the Kalman Filter"** UNC-Chapel Hill, TR 95-041, July 24, 2006

[3] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim,"**A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking"** IEEE transactions on signal processing, vol. 50, no. 2, pp 55-70, February 2002.

[4] Chi-Jeng Chang, Pei-Yung Hsiao, Zen-Yi Huang '**Integrated Operation of Image Capturing and Processing in FPGA'**, IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.1A, pp 173-179. (2006).

[5] Crookes D., Benkrid K., Bouridane A., Alotaibi K., and BenkridA.(2000), '**Design and implementation of a high level programming environment for FPGA-based image processing'**, Vision, Image and Signal Processing, IEE Proceedings, vol. 147, Issue: 4 pp. 377 - 384, Aug, 2000.

[6] D. Comaniciu, V. Ramesh, P. Meer, "**Kernel-basedobject tracking**," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, pp. 564-575, May 2003K. Fukunaga, L.D. Hostetler, "**The Estimation of the Gradient of a Density Function, with applications in Pattern Recognition**", *IEEETransactions on Information Theory*, Vol. 21, pp. 32-40. January 1975

[7] D. Comaniciu and P. Meer, "**Mean shift: A robust approach toward feature space analysis**," *IEEE Trans. Pattern Anal. Machine Intel.*, vol. 24, no. 5, pp. 603–619, 2002.

[8] Kota Solomon Raju, Gargi Baruah, Manipati Rajesham and Palash Phukan, "**Computing Displacement of Moving Object of a Real time Video Using EDK**" *Proc International Conference on Computing, Communications, Systems And Applications(ICCCSA) Hyderaba*d, ,pp 76-79; ISBN: 978-81-921580-8-2;30th-31st March 2012

[9] Kota Solomon Raju, Gargi Baruah, Manipati Rajesham and Palash Phukan, "**Histogram Calculation of Real time video using EDK**," *Proc. International Conference on Electronics and Communication Engineering(ICECE) Chandigarh,,* pp.45-48; IPM PVT. LTD, ISBN:978-93-81693-46-9. 17th - 18th March

[10] Spartan-6 Industrial Video Processing Kit – EDK Reference Design Tutorial, www.em.avnet.com/spartan6video

[11] Madhurima, Madhulika, " **Object tracking in a video sequence using Mean-Shift Based Approach: An Implementation using MATLAB7**" ,*IJCEM International Journal of Computational Engineering & Management,* Vol. 11, January 2011

[12] D. Comaniciu, V. Ramesh, and P. Meer, "**Real-time tracking of non-rigid objects using mean shift**," *inProc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp 407-418, January 2003.*Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp 407-418, January 2003