Implementation of moving object tracking using EDK

Kota Solomon Raju[1], Gargi Baruah[2], Manipati Rajesham[1], Palash Phukan[2] , Manoj Pandey[1]

[1] DSG, Council of Scientific and Industrial research (CSIR) -Central Electronics Engineering Research Institute (CEERI)
CSIR-CEERI, Pilani 333031[1],

[2] Dept of Electronics & Communication Engineering, Tezpur (Central) University
Tezpur-784028[2]

Abstract

The Xilinx Spartan family provides the ability to perform partial reconfiguration. This paper concentrates on how to track a moving object using three different components- fixed hardware, reconfigurable hardware and software. This flow can be considered to be a part of a general methodology to implement mean shift algorithm for tracking moving object.

Keywords: *FPGA, Xilinx, EDK, XPS, Bhattacharya Coefficient, pixel values, tracking, mean shift.*

1. Introduction

Visual Object tracking is an important topic in multimedia technologies. The aim of an object tracker is to generate the trajectory of an over time by locating its position in every frame of the video. The efficient tracking of visual features in complex environment is a challenging task. Real time applications such as surveillance and monitoring [1], perceptual user interfaces [1], smart rooms [1] and video compression[1] all require the ability to track moving objects. Tracking algorithms can be classified into two major groups, namely state-space approach and kernel based approach. State-space approaches are based largely on probability, stochastic processes and estimation theory, which, when combined with systems theory and combinatorial optimization, lead to a plethora of approaches, such as Kalman filter, Extended Kalman Filter (EKF) [2,3], Unscented Kalman Filter (UKF) [4], Particle Filter (PF) [5]. The ability to recover from lost tracks makes State space approach one of the most used tracking algorithms.

The Mean Shift (MS) algorithm is a non-parametric method which belongs to the second group. MS is an iterative kernel-based deterministic procedure which converges to a local maximum of the measurement function under certain assumptions about the kernel behaviors [6]. Mean Shift algorithm has recently gained significant attention as an efficient and robust method for visual tracking. A number of attempts have been made to achieve robust, high performance target tracking [7][8].

Tracking objects also require complex computational processing throughput which seems challenging in terms of processing as well as cost. An FPGA can give high efficiency, flexibility, greater processing ability and can reduce costs with various verification techniques such as behavioral simulation and post route simulation. Also, 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. Now every time we track an object, the algorithm we applied needs to be improved or changed or another algorithm added with it to get a better tracking system. Thus certain system components can be replaced while the remaining unaffected parts can remain fully operational. A developer could use one chip for different tasks and switch between them during runtime. The Xilinx Spartan family provides for this partial reconfigurability.

Thus this project aims at designing a real time video capture and tracking an object on Spartan 6 Industrial Video Processing Kit(Spartan 6 LX150T FPGA) using the Mean Shift algorithm.

Also in our implementations we have used RGB color space. In many implementations either RGB is converted to grayscale[8] or to Luv[9] or they use just one of the 3 RGB channels. Here we have used all the 3 channels.

2. RGB Histogram

The choice of histogram function is also very important. If a bin is used for all possible colors for a 24 bit frame, then there will be 256*256*256= 16 million bins. So it is set up to 3 colors, 16 bin per model i.e. the feature space is quantized to 16*16*16=4096 bin values[10].

The range of three components of RGB color space is [0,255], so the level of object's feature space N_3 can range in $[0,255^3]$. In this way the color information of object can be described adequately, but the level is too large, resulting in huge computation scale and poor real-time performance. Based on the N-Bin histogram [5, 8] algorithm, the problem of level quantization is solved by aggravating the weight of red components and weakening that of blue components. So from literature [9] we first define a quantitative constant Q, where Q= [0，255], and then the level range N_3 of feature space in color histogram is quantized into $[0,(255/Q)^3]$ is computed as follows:

$\Delta(x, y) = N2 * R'(x, y) + N1 * G'(x, y) + N0 * B'(x, y)$

In (2), fix (*) means to round to the integer. Here we set the value of Q to be 16, so the quantitative level will lower down to $4096(16^3)$.

This paper is divided into 2 sections. The first section describes the Mean shift algorithm that is used for tracking and the second section describes the hardware implementation of the tracking algorithm in Spartan 6 LX150T.

3. Mean Shift Algorithm

Mean shift is a nonparametric density gradient statistical method which considers feature space as an empirical probability density function. If the input is a set of points, then mean shift considers them as sampled from the underlying density function.

3.1 Target representation

A feature space is first chosen to characterize the target represented by its pdf 'q' in the feature space centred at a spatial location 0. In the subsequent frame, a target candidate is defined at a location 'y' with pdf p(y). Thus,

From the literature [2] **object model** pdf is given by

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

Assuming size of the model to be normalized with kernel radius h=1. Here C is the normalization constant.

$$C=[\sum_{i=1}^{n} k(\left||xi|\right|)^2]^{-1} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.... (2)$$

Kernel profile k weights contribution by distance to centroid and ∂ is the Kronecker delta function

$\partial(a) = \{ {}^{1\ ,if\ a=0}_{0\ \ otherwise}$ i.e. $(k\left||xi|\right|)^2$ to Qu only if b(xi)=u. ………………………………… (3)

Target model for target centred at y and y_i=1,2..nh are the pixel locations, Ch is the normalization constant.

$$p(y)=Ch\sum_{i=1}^{nh} \left|(|\left|\frac{y-yi}{h}\right||^2\right)\partial(b(yi)-u). \ldots\ldots\ldots\ldots.(4)$$

From [2] the Bhattacharyya coefficient is given by

$$\rho(\vec{y}) = \rho[\vec{p}(y),\vec{q}] = \sum_{u=1}^{m} \sqrt{p_u(y), q_u} \ldots\ldots\ldots\ldots\ldots....(5)$$

Now the similarity function defines a distance among target model and candidates and we define that distance between two discrete distributions as

$$d(y)=\sqrt{1-\rho[\vec{p}(y),\vec{q}]} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots...(6)$$

Thus, the steps for **Mean Shift Tracking Algorithm** are given as follows:

Given {qu} of model and location y of target in previous frame:

1. Initialize location of target in current frame as y.

2. Compute {pu(y)}, u = 1, . . . ,m, and ρ(p(y), q) from eq. (5)

3. Compute weights

$$w_i = \sum_{u=1}^{m} \sqrt{\frac{q_u}{p_u\ (y)}}\ [\partial(\mathrm{b(yi)-u})] \ldots\ldots\ldots\ldots\ldots\ldots.(7)$$

.4 Apply **mean shift**: Compute new location Z as

$$z = \frac{\sum_{i=1}^{nh}(wig|\left|\frac{y-yi}{h}\right||^2)yi}{\sum_{i=1}^{nh}(wig|\left|\frac{y-yi}{h}\right||^2)} \ldots\ldots\ldots\ldots\ldots(8)$$

Where $g(x) = -k'(x)$

5. While $\rho[p(y),q] < \rho[p(z),q]$

Do $y1 \leftarrow 1/2(y+z)$

Evaluate $\rho[p(y1),q\}$

6. If $\|z - y\| <$ € Stop.

Otherwise set $y \leftarrow z$ and go to Step 2.

4. Software flow compilation.

The flowchart for the algorithm is shown below

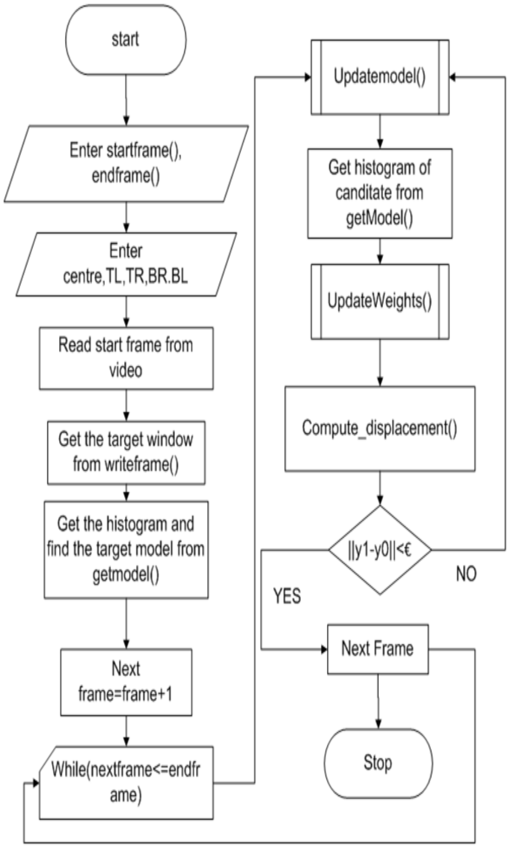


Fig. 1 Flowchart for mean shift

This Flowchart is programmed in C using EDK. The Input files → *.c, *.h, libc.a, libXil.a, libm.a. The EDK goes through 4 stages.

- Pre-processor: Replaces all macros with definitions as defined in the .c or .h files
- Machine-specific and language-specific compiler: Compiles C/C++ code
- Assembler: Converts code to machine language and generates the object file
- Linker: Links all the object files using user-defined or default linker script.

The output file is then obtained as **executable.elf.**

5. Hardware description

The board we have used is Xilinx Spartan-6 XC6SLX150T-3FGG676C FPGA. I/O Connectors are two FMC LPC general-purpose I/O expansion, and a memory of 128 MB DDR3 SDRAM. For communication we used RS-232 serial port ,USB 2.0 USB-RS232 Port and for configuration — XCF32 and XCF08 Platform Flash Configuration Flash, Xilinx Parallel Cable IV or Platform USB Cable support for JTAG Programming/ Configuration. The figure below shows the board details.

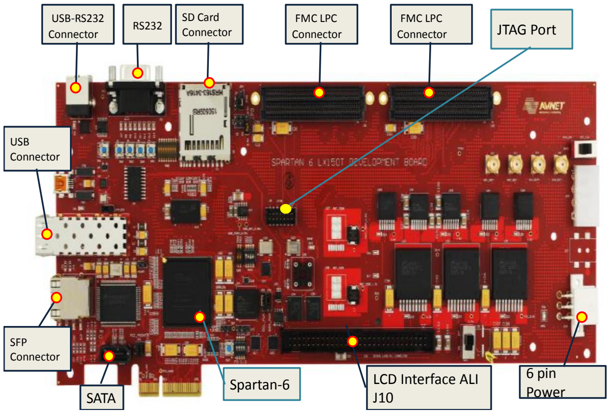


Fig. 2 Spartan 6 board call out diagram

5.1. Embedded development Kit

The Embedded Development Kit is the Xilinx software suite for designing complete embedded programmable systems. It enables the integration of both hardware and software components of an embedded system. In it Xilinx Platform Studio (XPS) is a graphical Integrated Design Environment (IDE) that incorporates all the Embedded System Tools for seamless creation of hardware and software components and, optionally, a verification component.

All of the EDK designs are built on a Base Platform which provides a common base and building blocks. Each of the EDK reference designs included with the IVK is built from the base platform. The Base Platform is not a separate design that is delivered with this kit, rather it is the starting point from which all the other designs were built.

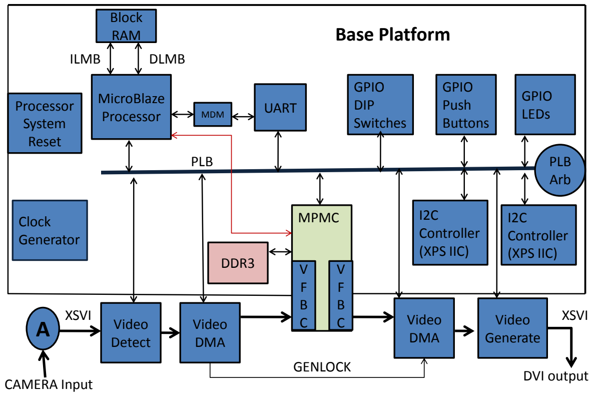


Fig.3 Base Platform with the added IPs

The section A is described as follows and fig 4 shows its complete description.

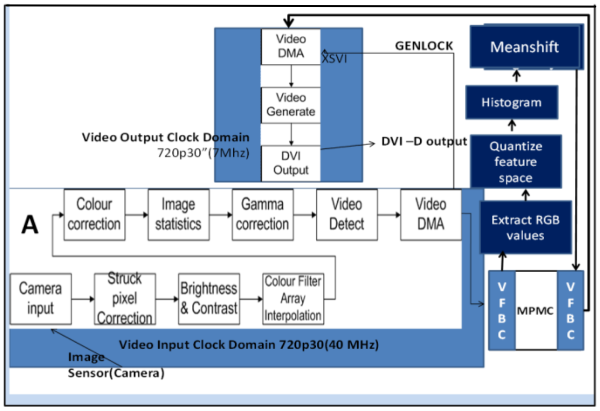


Fig. 4 Camera Frame Buffer – Video Pipeline

The image sensor video input source enters the Camera Input PCORE [11]. This PCORE decodes the BT656 codes to generate synchronization signals and formats the video as an XSVI bus interface. The Video Detect PCORE does not alter the video, but monitors the VSYNC and ACTIVE_VIDEO signals to determine the dimensions of the active video streaming through the FPGA. It also generates Video DMA compatible bus interface used to write video data to external memory. The Video DMA PCOREs, in collaboration with the Video Frame Buffer Controller (VFBC)[11].

Interfaces on the Multi-Port Memory Controller (MPMC), perform the actual transfers to/from external memory. These cores are extremely flexible and are configured via the Micro Blaze processor. The GENLOCK port indicates where the first Video DMA has written the incoming frames. The second Video DMA reads video frames from memory based on the GENLOCK [11] information. After that the histogram calculation IP and later the mean shift block gets the pixel data and takes the RGB values, each of 8 bit. It takes the precalculated kernel values and finds out the histogram of the target and the candidate model and they compute the displacement in the mean shift block of the target object in each frame.

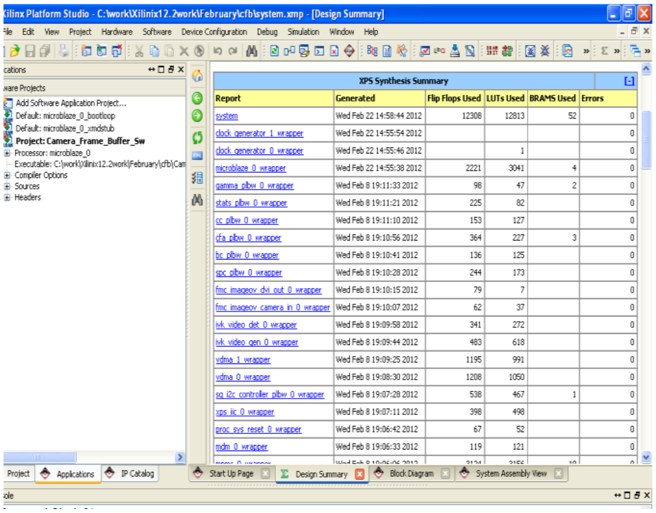
Since the output frame rate is higher than the input frame rate, frames are duplicated when necessary. The Video Generate PCORE, under control of the MicroBlaze, generates video timing for the output. It also generates a Video DMA compatible bus interface used to read video data from external memory.

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.

The video capture is at 1280x720P @ 30Hz and video playback at 1280x720P @ 60Hz. These resolutions are configured by the embedded processor (MicroBlaze) and can be modified to support other resolutions (limited by the image sensor used).

6. Experimental results

The resources, TP and the addresses obtained from the EDK tool and the elf file generated are shown below:

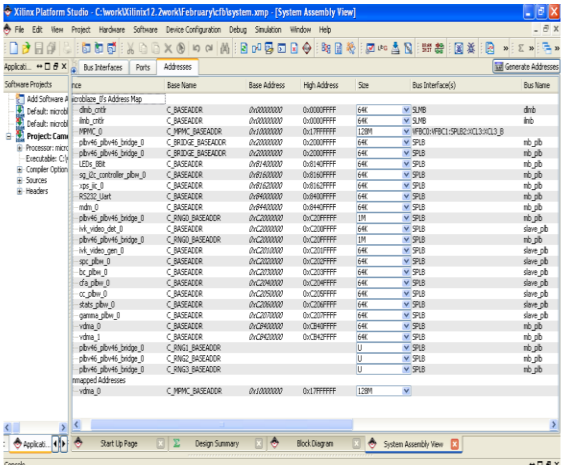


Add Software Application Project...
Default: microblaze_0_bootloop
Default: microblaze_0_xmdstub
Project: Camera_Frame_Buffer_Sw
Processor: microblaze_0
Executable: C:\work\Xilinix12.2work\February\cfb\Cam
Compiler Options
Sources
Headers

XPS Synthesis Summary

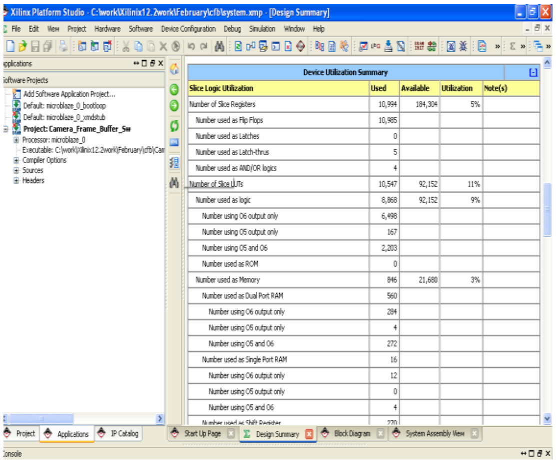
Report	Generated	Flip Flops Used	LUTs Used	BRAMS Used	Errors
system	Wed Feb 22 14:58:44 2012	12308	12813	52	0
clock_generator_1_wrapper	Wed Feb 22 14:55:54 2012				0
clock_generator_0_wrapper	Wed Feb 22 14:55:46 2012		1		0
microblaze_0_wrapper	Wed Feb 22 14:55:38 2012	2221	3041	4	0
gamma_plbw_0_wrapper	Wed Feb 8 19:11:33 2012	98	47	2	0
stats_plbw_0_wrapper	Wed Feb 8 19:11:21 2012	225	82		0
cc_plbw_0_wrapper	Wed Feb 8 19:11:10 2012	153	127		0
cfa_plbw_0_wrapper	Wed Feb 8 19:10:56 2012	364	227	3	0
bc_plbw_0_wrapper	Wed Feb 8 19:10:41 2012	136	125		0
spc_plbw_0_wrapper	Wed Feb 8 19:10:28 2012	244	173		0
fmc_imageov_dvi_out_0_wrapper	Wed Feb 8 19:10:15 2012	79	7		0
fmc_imageov_camera_in_0_wrapper	Wed Feb 8 19:10:07 2012	62	37		0
ivk_video_det_0_wrapper	Wed Feb 8 19:09:58 2012	341	272		0
ivk_video_gen_0_wrapper	Wed Feb 8 19:09:44 2012	483	618		0
vdma_1_wrapper	Wed Feb 8 19:09:25 2012	1195	991		0
vdma_0_wrapper	Wed Feb 8 19:08:30 2012	1208	1050		0
sg_i2c_controller_plbw_0_wrapper	Wed Feb 8 19:07:28 2012	538	467	1	0
xps_iic_0_wrapper	Wed Feb 8 19:07:11 2012	398	498		0
proc_sys_reset_0_wrapper	Wed Feb 8 19:06:42 2012	67	52		0
mdm_0_wrapper	Wed Feb 8 19:06:33 2012	119	121		0

Fig. 5 XPS Synthesis summary



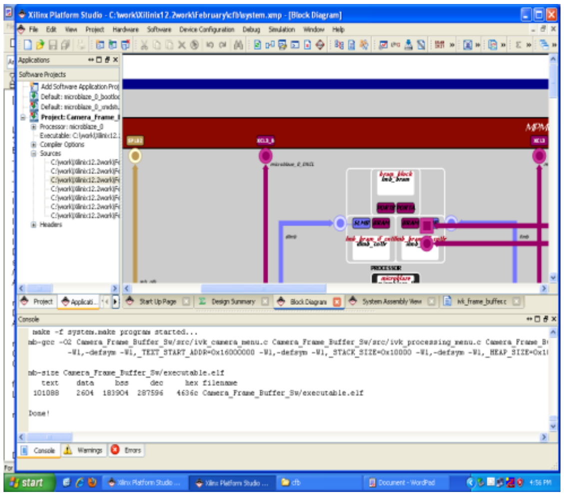
nce	Base Name	Base Address	High Address	Size	Bus Interface(s)	Bus Name
icroblaze_0's Address Map						
dlmb_cntlr	C_BASEADDR	0x00000000	0x0000FFFF	64K	SLMB	dlmb
ilmb_cntlr	C_BASEADDR	0x00000000	0x0000FFFF	64K	SLMB	ilmb
MPMC_0	C_MPMC_BASEADDR	0x10000000	0x17FFFFFF	128M	VFBC0:VFBC1:SPLB2:XCL3:XCL3_B	
plbv46_plbv46_bridge_0	C_BRIDGE_BASEADDR	0x20000000	0x2000FFFF	64K	SPLB	mb_plb
plbv46_plbv46_bridge_0	C_BRIDGE_BASEADDR	0x20000000	0x2000FFFF	64K	SPLB	mb_plb
LEDs_8Bit	C_BASEADDR	0x81400000	0x8140FFFF	64K	SPLB	mb_plb
sg_i2c_controller_plbw_0	C_BASEADDR	0x81600000	0x8160FFFF	64K	SPLB	mb_plb
xps_iic_0	C_BASEADDR	0x81620000	0x8162FFFF	64K	SPLB	mb_plb
RS232_Uart	C_BASEADDR	0x84000000	0x8400FFFF	64K	SPLB	mb_plb
mdm_0	C_BASEADDR	0x84400000	0x8440FFFF	64K	SPLB	mb_plb
plbv46_plbv46_bridge_0	C_RNG0_BASEADDR	0xC2000000	0xC20FFFFF	1M	SPLB	mb_plb
ivk_video_det_0	C_BASEADDR	0xC2000000	0xC200FFFF	64K	SPLB	slave_plb
plbv46_plbv46_bridge_0	C_RNG0_BASEADDR	0xC2000000	0xC20FFFFF	1M	SPLB	mb_plb
ivk_video_gen_0	C_BASEADDR	0xC2010000	0xC201FFFF	64K	SPLB	slave_plb
spc_plbw_0	C_BASEADDR	0xC2020000	0xC202FFFF	64K	SPLB	slave_plb
bc_plbw_0	C_BASEADDR	0xC2030000	0xC203FFFF	64K	SPLB	slave_plb
cfa_plbw_0	C_BASEADDR	0xC2040000	0xC204FFFF	64K	SPLB	slave_plb
cc_plbw_0	C_BASEADDR	0xC2050000	0xC205FFFF	64K	SPLB	slave_plb
stats_plbw_0	C_BASEADDR	0xC2060000	0xC206FFFF	64K	SPLB	slave_plb
gamma_plbw_0	C_BASEADDR	0xC2070000	0xC207FFFF	64K	SPLB	slave_plb
vdma_0	C_BASEADDR	0xCB400000	0xCB40FFFF	64K	SPLB	mb_plb
vdma_1	C_BASEADDR	0xCB420000	0xCB42FFFF	64K	SPLB	mb_plb
plbv46_plbv46_bridge_0	C_RNG1_BASEADDR			U	SPLB	mb_plb
plbv46_plbv46_bridge_0	C_RNG2_BASEADDR			U	SPLB	mb_plb
plbv46_plbv46_bridge_0	C_RNG3_BASEADDR			U	SPLB	mb_plb
nmapped Addresses						
vdma_0	C_MPMC_BASEADDR	0x10000000	0x17FFFFFF	128M		

Fig. 6 IP addresses



Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	10,994	184,304	5%	
Number used as Flip Flops	10,985			
Number used as Latches	0			
Number used as Latch-thrus	5			
Number used as AND/OR logics	4			
Number of Slice LUTs	10,547	92,152	11%	
Number used as logic	8,868	92,152	9%	
Number using O6 output only	6,498			
Number using O5 output only	167			
Number using O5 and O6	2,203			
Number used as ROM	0			
Number used as Memory	846	21,680	3%	
Number used as Dual Port RAM	560			
Number using O6 output only	284			
Number using O5 output only	4			
Number using O5 and O6	272			
Number used as Single Port RAM	16			
Number using O6 output only	12			
Number using O5 output only	0			
Number using O5 and O6	4			
Number used as Shift Register	270			

Fig. 7 Device utilization summary

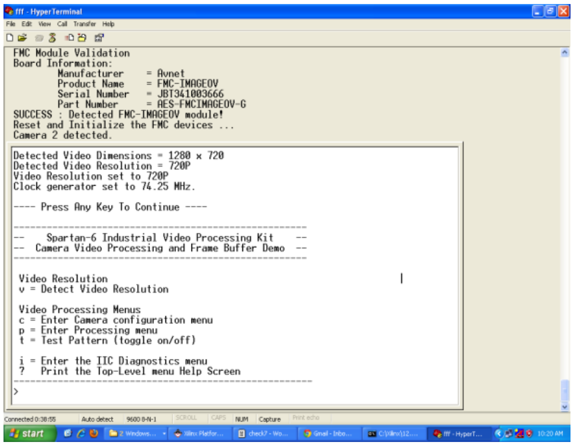


```
 make -f system.make program started...
mb-gcc -O2 Camera_Frame_Buffer_Sw/src/ivk_camera_menu.c Camera_Frame_Buffer_Sw/src/ivk_processing_menu.c Camera_Frame_B
        -Wl,-defsym -Wl,_TEXT_START_ADDR=0x16000000 -Wl,-defsym -Wl,_STACK_SIZE=0x10000 -Wl,-defsym -Wl,_HEAP_SIZE=0x1

mb-size Camera_Frame_Buffer_Sw/executable.elf
   text    data     bss     dec     hex filename
 101088    2604  183904  287596   4636c Camera_Frame_Buffer_Sw/executable.elf

Done!
```

Fig. 8 The elf file (Camera_Frame_Buffer_Sw/executable.elf)



```
FMC Module Validation
Board Information:
        Manufacturer    = Avnet
        Product Name    = FMC-IMAGEOV
        Serial Number   = JBT341003666
        Part Number     = AES-FMCIMAGEOV-G
SUCCESS : Detected FMC-IMAGEOV module!
Reset and Initialize the FMC devices ...
Camera 2 detected.
Detected Video Dimensions = 1280 x 720
Detected Video Resolution = 720P
Video Resolution set to 720P
Clock generator set to 74.25 MHz.

---- Press Any Key To Continue ----

------------------------------------------------------
--    Spartan-6 Industrial Video Processing Kit     --
--  Camera Video Processing and Frame Buffer Demo   --
------------------------------------------------------

 Video Resolution
 v = Detect Video Resolution

 Video Processing Menus
 c = Enter Camera configuration menu
 p = Enter Processing menu
 t = Test Pattern (toggle on/off)

 i = Enter the IIC Diagnostics menu
 ?   Print the Top-Level menu Help Screen
-------------------------------------------------------
>
```

Fig. 9 The Hyper terminal window

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:

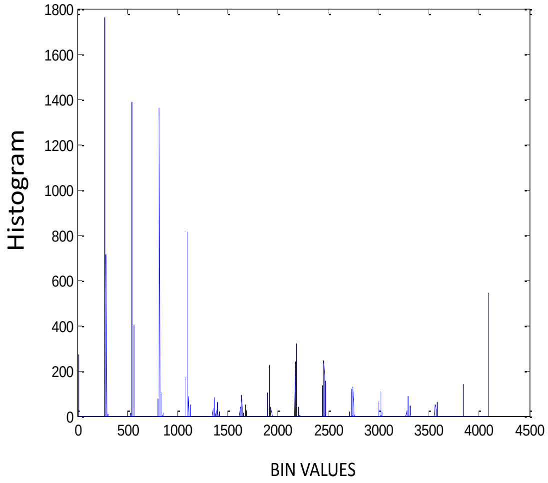


Fig. 10 Histogram of a 160 X 80 target window of 1280X720 frame(1st frame)

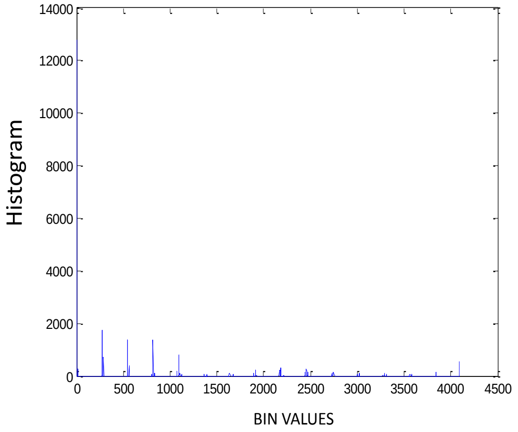


Fig. 11 Histogram of a 160 X 80 target window of 1280X720 frame (2nd frame)

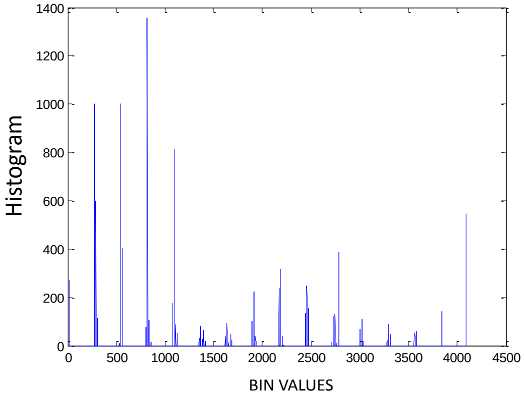


Fig. 12 Histogram of a 160 X 80 target window of 1280X720 frame (3^{rd} frame)

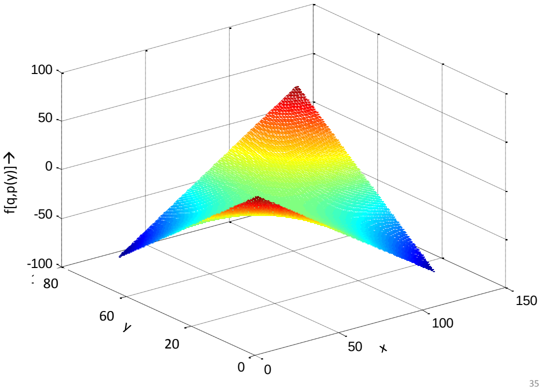


Fig.13 Similarity function f [q,p(y)]

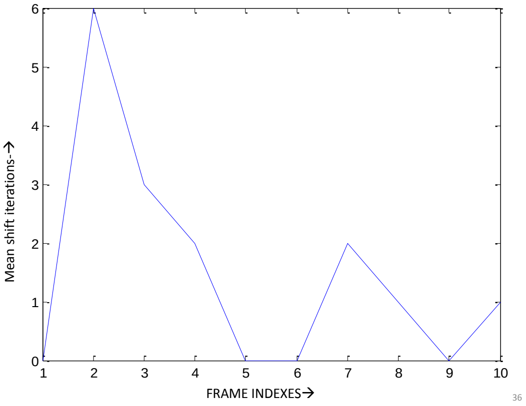


Fig.14 Mean shift iterations per frame index

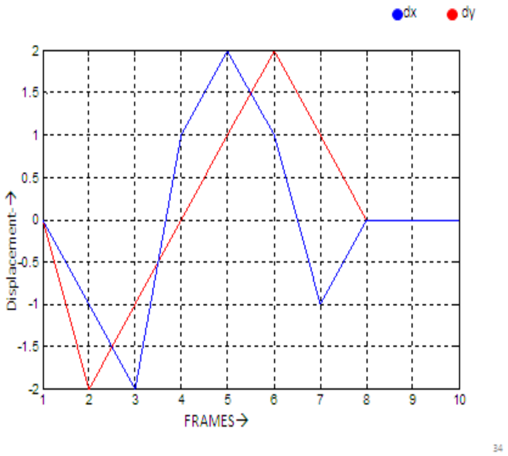


Fig. 15 Displacement of the target object in 10 consecutive frames

6. Conclusion And Future Work

In this paper we have explored the use of variable kernels to enhance a weighted histogram and then used Mean shift to determine the average shift and thus 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.

Future work is currently underway to extend out test bed platform for tracking of objects in real time by

1. developing scaling parameter

2. Complementing our motion-tracking algorithm by adding further improved calculations.

Acknowledgment

The work reported in this paper is a part funded by CSIR-Supra Institutional Project. For this project we would like to thank Mr. Vaibhawa Mishra for his 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,vol. 40, No. 7,pp 234-246, 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] K. Fukunaga, L.D. Hostetler, "The Estimation of the Gradient of a Density Function, with applications in Pattern Recognition", IEEE Transactions on Information Theory, vol. 21, pp. 32-40. January 1975.

[5] D. Comaniciu, P. Meer, "Mean shift analysis and applications," In IEEE Int. Conf. on Computer Vision, vol. 2, pp. 1197-1203, March 1999

[6] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-based object tracking," IEEE Trans. On Pattern Analysis and Machine Intelligence, pp. 564-575, Dec 2003.

[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,Dec 2002.

[8] 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

[9] Benjamin Gorry, Zezhi Chen, Kevin Hammond, Andy Wallace, and Greg Michaelson, "Using Mean-Shift Tracking Algorithms for Real-Time Tracking of Moving Images on an Autonomous Vehicle Testbed Platform" ,International Conference on Intelligent Robotics and Manufacturing Automation, Venice, Italy, 2007 , World Academy of Science, Engineering and Technology (PWASET) , pp-45-48, November 23-25, 2007

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

[11] Spartan-6 Industrial Video Processing Kit – EDK Reference Design Tutorial

Kota Solomon Raju has been working as Principal Scientist in Digital Systems Group, CSIR -Central Electronics Research Institute (CSIR - CEERI), Pilani, Rajasthan, India. He received the Bachelor of Engineering degree in 1997 from Andhra University, Master of Engineering in 2003 from Birla Institute of Technology and Science (BITS), Pilani and Ph.D. in 2008 from department of Electronics and Computer Engineering, IIT Roorkee, India. Dr. Solomon is an advanced electronic systems design engineer. His research work focused on reconfigurable computing systems (RCS), advanced embedded systems design and wireless sensor network (WSN) based embedded systems design and included hardware/software codesign, parallelizing applications, customized computing, and high-level synthesis, ad hoc networking, Zigbee based networking, and other sensor based embedded systems, protocols design and CAD tools for electronic systems design. He established state-of-the-art RCS Lab & WSN Systems Lab in Digital Systems Group. Apart from above R&D he also teaches /gives lectures in System Modeling & Design languages as well as in Real -time systems design courses for the post graduate students. He has delivered fifteen invited talks in international / national conferences /seminars/workshops. He is author and co-author of more than 35 scientific papers, published in peer-reviewed international journals and conferences, guided around 40 M.Tech dissertations so far and many B.Tech, M.Sc and MCA thesis. He has been guiding two Ph.D. students. He is active in several technical programs and delivered more than a dozen invited talks in international and national conferences and workshops. He is a life member of the IETE, New Delhi.

Gargi Baruah is presently doing her M. Tech in Electronics Design and Technology from Tezpur (Central) University, Assam, India. In 2009 she received her B.E degree in ECE from Anna University, Chennai, India She is currently working as a project trainee in CSIR-CEERI, Pilani for the completion of her Master's degree. She has published approximately four research papers in various international and national conferences. Her research interests are in tracking multiple moving objects and image and video processing using FPGAs as embedded systems.

M. Rajesham has received his M.Sc.Tech from NIT-Waranga in 2009 and **B.Sc from** Nizam College in 2006. He has got 1.9 years experience in development of Image Processing Algorithms and Applications on FPGAs. He has published approximately three research papers in international conferences. His interest lies in the field of image processing algorithms in FPGAs.

Palash Phukan is presently doing his M.Tech in Electronics Design and Technology from Tezpur (Central) University, Assam, India. In 2010 he received his B.E degree in ECE from Girijananda Chowdhury Institute of Management and Technology Gauhati University, Guwahati India. He is currently working as a project trainee in CSIR-CEERI, Pilani for the completion of his Master's degree. He has published approximately five research papers in various international and national conferences. His research interests are in designing median and various other filters to reduce noise in images using FPGAs as embedded systems.

Manoj Pandey is working as a Lecturer at B K Birla Institute of Engineering and Technology Pilani pursuing PhD at Sant Longowal Institute of Engineering & Technology, Longowal Punjab in area of Image Processing applications using FPGA based Reconfigurable Techniques. He has obtained his M. Sc. in Electronics from Deen Dayal Upadhyay Gorakhpur University, Gorakhpur and M.Tech in Electronics Design and Technology from Tezpur University (Central), Tezpur in year 2004 and 2007 respectively.