ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

Mean-Shift Algorithm: Verilog HDL Approach

Rahul V. Shah[*], Amit Jain[*],Rutul B. Bhatt[#], Pinal Engineer[$], Ekata Mehul[*]

[*]*ASIC department*
Einfochips, Ahmedabad

[$]*Electronic& Communication Department, SVNIT,Surat*

[#]*Electronic& Communication Department, SVIT, Vasad,Gujarat Technology University*

Abstract— **Object tracking algorithms, when it comes to implementing it on hardware ASIC, it becomes difficult task, due to certain limitations in hardware. This paper shows how mean-shift algorithm is implemented in HDL along with the description of ports and interfaces.**

Keywords— **Object tracking, complexity in hardware ASIC, Mean Shift algorithm, Histogram, Bhattacharya coefficient**

I. INTRODUCTION

Object tracking in its simplest form can be defined as the problem of estimating the trajectory of an object in an image plane as it moves around the scene. In other words the tracker assigns consistent labels to the tracked objects in different form of videos.

Object tracking is the important part in computer vision technology, but till time it is mainly dealt with the high grade softwares like MATLAB, ADOBE etc. But when it comes to implement it on ASIC chips it becomes difficult when many mathematical operations are mainly operated through adders and subtracters and not with multipliers and dividers. These object tracking algorithms when implemented on ASIC covers good amount of chip area as many such mathematical operations are involved in it, which are complex to implement with adders and subtracters. Here the images are considered as the matrix representation, object may be the cofactor of that image matrix.

Mean shift algorithm when implemented on HDL it become really useful to check the functions of the object tracking. The Mean shift algorithm requires Histogram, Bhattacharya coefficient, centre calculation which are implemented with small mathematical operations like multipliers, Dividers & square root etc. That makes algorithm to implement on HDL easy.

Section II deals with the overall algorithm and the pins description for whole object tracking controller, section III deals with histogram and its development, Section IV deals with Bhattacharya co-efficient and Rho calculation, finally in section V state diagram of Object tracking controller and Mean shift controller and finally section VI deals with results of the implementation on verilog HDL compared with MATLAB.

II. OBJECT TRACKING CONTROLLER

The Object Tracking algorithm is designed for tracking of objects in video or in real-time. It is based on Meanshift algorithm. Interface with external video source i.e. camera for input video, HOST interface for providing inputs, interface with external memory for storing video frames and interface with monitor for video output are some of the interfaces required for object tracking. The input video interface is used to provide video input to the algorithm. In this algorithm, we are using 24-bit RGB format for input frames. After getting video from the source, it is passed on to the processing unit as well as external memory. Host Interface provides inputs to the processing units for performing the operations on the input video. Processing unit consists of different functional sub-blocks like histogram, Kernel, Meanshift, Bhattacharyya Coefficient etc. After completion of the operations, outputs are passed on to the monitor for display.

External Memory interface is used for storing video frames depending upon the clock frequency and video speed. Since video is continuous, so it is required to save frames in memory and simultaneously perform processing on continuous flowing frames.Output interface with monitor is required to show the output video frames after doing the operations on them and verify the result. Figure 1 shows the detailed Processing Unit diagram with different sub-blocks like histogram, Meanshift, rect_box, Bhattacharyya coefficient. According to functionality, video frames are stored in the memory. The no. of frames to be stored depends on clock frequency and video speed. According to inputs **ot_centerx_i, ot_centery_i, ot_hx_i, ot_hy_i,** and **ot_bins_i** reference histogram **(ot_qu)** is calculated for first frame. For finding reference histogram **(ot_qu)** for first frame, cropping is done on the image and kernel value is calculated for that. Then depending upon the R, G, B values hist_model is calculated.

ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

Now to track object in next frames, reference histogram **(ot_qu)** and other inputs are given to Mean-Shift block. In Mean-Shift block, histogram **(ot_1pu)** is calculated for second frame using old center values **(ot_centerx_i, ot_centery_i)** and then Bhattacharyya coefficient **(ot_1rho)** is calculated using **ot_qu** and **ot_1pu**. Then weighted array **(ot_w)** and normalized row and column arrays **(ot_1pl and ot_2pl)** are calculated for second frame. Then by using weighted array and normalized row and column arrays, new center values **(ot_new_centerx and ot_new_centery)** for second frame are calculated. The reason for calculating new center values is that we assume that object has displaced from its original position as compared to frame 1. Then new histogram **(ot_2pu)** using new center values and new Bhattacharyya coefficient **(ot_2rho)** using **ot_qu and ot_2pu** are calculated. **Iteration loop** depends upon comparison results of **ot_1rho** and **ot_2rho.** According to iteration loop, final center value for 2nd frame is obtained. With final **center values**, **ot_hx_i** and **ot_hy_i,** rectangular box is created around the targeted object in 2nd frame. The iteration loop is looped 20 times, which is maximum value for this algorithm. Iteration is done because at first instant it doesn't give final center for current frame and moreover, object is displaced in next consecutive frames but to know how much it moved and in which direction it moved, iteration value is taken. For all other frames, this process is repeated in Mean-shift block and object is tracked in rest of the frames.

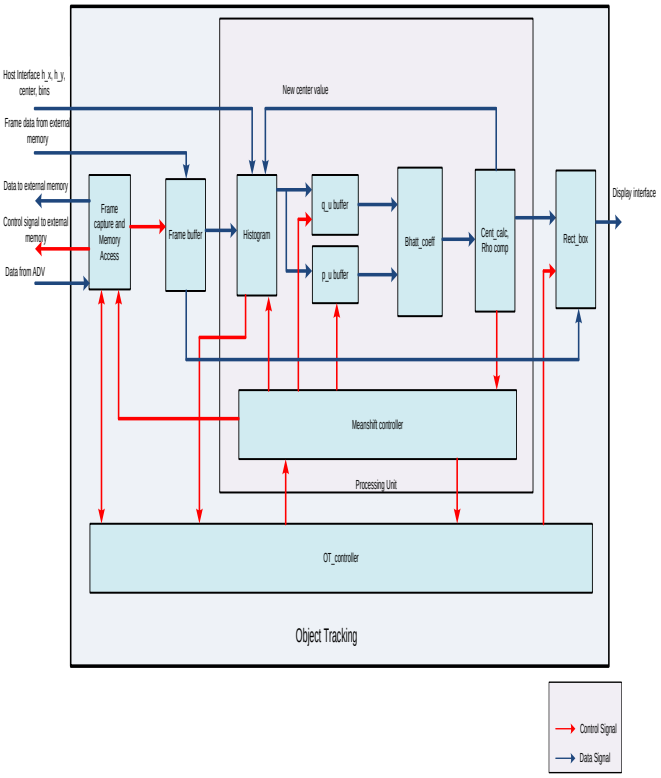

Figure:1Block diagram of algorithm

Functional Description:

1) OT_controller enables the frame capture and memory access logic (FCMA). FCMA takes data from ADV and enables external memory. Then FCMA stores data into external memory and sends signal back to OT_controller that frame is available in external memory. OT_controller then signals mean shift controller to take control.

2) Mean-shift controller enables the FCMA to read frame from external memory. FCMA enables internal frame buffer for storing frame. Frame data goes to histogram block and reference histogram is calculated.

3) Histogram signals OT_controller that reference histogram of first frame has been calculated. OT_controller signals Mean-shift controller to store reference histogram in q_u buffer. Mean-shift controller enables q_u buffer to store reference histogram. Mean shift controller sends signal to OT_controller that reference histogram is stored.

ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

4) OT_controller enables FCMA and FCMA takes second frame data and store in external memory by enabling the external memory. FCMA signals OT_controller that frame is available in external memory. Then OT_controller signals Mean shift controller and Mean shift controller enables buffer through FCMA.

5) FCMA stores second frame in frame buffer and it goes to histogram. Histogram block sends signal to OT_controller that calculations are done for second frame with present center values. OT_controller signals Mean shift controller to take control. Mean shift controller enables p_u buffer to store histogram of second frame.

6) Data from q_u buffer and p_u buffer goes to Bhattacharyya coefficient block and rho1 is calculated. Then for second frame, new center is calculated and feedback to the Histogram block. Histogram block calculates histogram for the second frame with new center value. And it will be stored in p_u buffer. Again Bhattacharyya coefficient (rho2) is calculated.

7) In center calculation and rho logic block, rho values are compared and final center value after 20 iterations is obtained. Center calculation sends signal to Mean-shift controller that final center value is calculated for second frame. Mean-shift controller sends signal to OT_controller that center for second frame is calculated.

8) OT_controller sends enable signal to rect_box. Second frame from frame buffer goes to rect_box and new center values goes to rect_box from center calculation block. Second frame is then sent to display device.

9) OT_controller takes third frame and like this process is repeated till end of the video.

III. HISTOGRAM

The histogram is an important term in this algorithm. It is a combination of R, G, B and Spatial information. The inputs of rect_crop are images, ot_centerx, ot_centery, ot_hx, ot_hy. Binwidth is calculated by dividing 256 by bins. The image_model is generated by dividing pixel values by binwidth to give output image_model in normalized format. kernel value is calculated using these inputs. Histogram array of size defined by bin value is initialized with zeros.

R, G, B values of normalized image_model are calculated using I & j logic. Then these R, G, B values are used for finding locations in histogram array where squared kernel value is added. Then this squared Kernel value is added to constant value "c". The histogram is divided by the final value of constant "c" to get normalized histogram.

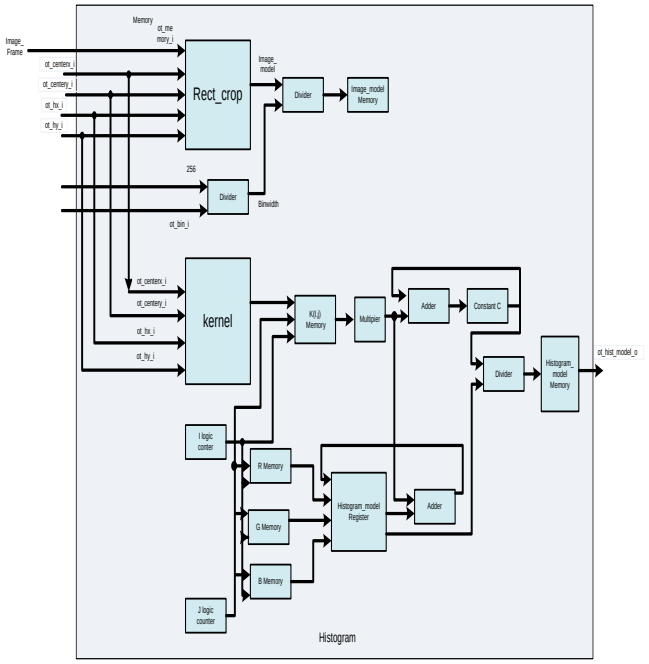

Figure 2: Histogram block diagram

Histogram operations consists of two sub-operations namely

A. Kernel

The main aim of this block is to calculate the kernel values. The inputs to kernel are ot_centerx_i, ot_centery_i, ot_image_size1_i, ot_image_size2_i, ot_hx (width), ot_hy(height). For calculating normalized row and column matrices, pixel values are obtained by subtracting ot_centery & ot_centerx from i and j values and dividing the result by ot_hy & ot_hx respectively. I and j loops for normalization depend on image size1 and image size2 respectively. Normalized row and column arrays are then cropped and stored into memory. Distance is calculated by squaring and adding all values of normalized row and column matrix. Distance value is compared, if it is greater than or equal to 1,

ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

then kernel value at I & j (ot_k(I, j)) location is 0 otherwise it is 1.

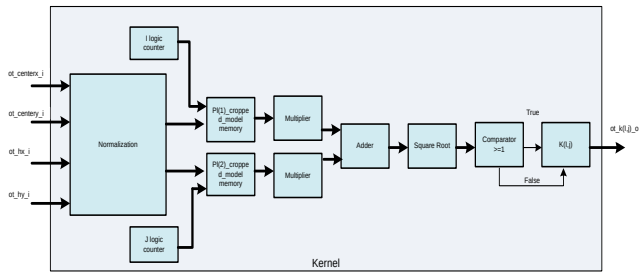

Figure 3: kernel

B. Rect_crop

Crop operation is clipping drawing views with a defined boundary. Crop operation takes rendered images and crop the image to a specified rectangular area.To perform image cropping, inputs are ot_centerx, ot_hx, ot_centery, ot_hy. Limits of image to be cropped are defined by ot_centerx, ot_hx and ot_centery, ot_hy. x1 <= j <= x2, y1 <= i <= y2, where x1 is defined by ot_centerx-ot_hx, x2 is defined by ot_centerx+ot_hx, y1 is defined by ot_centery-ot_hy, y2 is defined by ot_centery+ot_hy. The pixels which satisfy the above equation are stored in memory and the pixels which do not satisfy this condition are discarded.

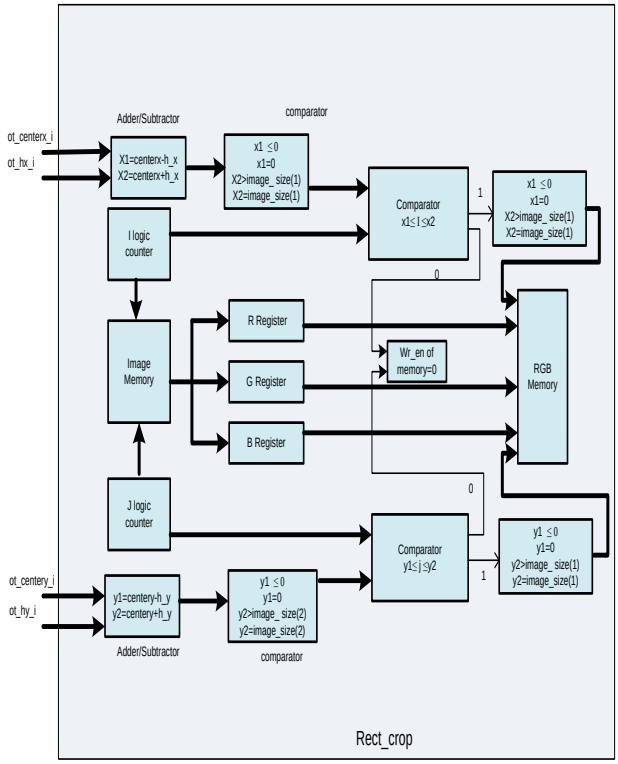

Figure 4: Rectangular crop

IV. HISTOGRAM

According to i, j, k logic reference histogram ot_qu(i,j,k) & new histogram (ot_pu(i,j,k) are multiplied. Then the square root of result is added to the initialized rho value.

Bhattacharyya coefficient measures the similarity between two histograms. To calculate Bhattacharyya Coefficient, histograms of the reference image and histogram of current frame is required.

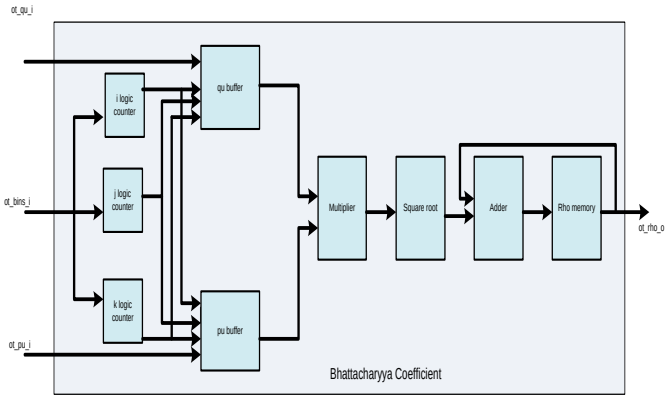

Figure 4: Histogram

V. CENTER CALCULATION & RHO LOGIC

A. Centre calculation

The inputs of Center Calculation Block are ot_image_model_size1,ot_image_model_size2,ot_old_centerx, ot_old_centery, ot_hx, ot_hy and ot_w. According to i and j values, which are dependent on ot_image_model_size1 & ot_image_model_size2, all values of normalized row and column arrays are multiplied by weighted array values and tempx and tempy variables are calculated. Weighted array values are also added to temp_t variable.

At the end of the loop, temp_x and temp_y values are divided by temp_t variable and the final result is stored in temp_x and temp_y. For calculating new center values, temp_x and temp_y are multiplied by ot_hx and ot_hy and added to ot_old_centerx and ot_old_centery respectively. Then the result is stored in ot_center1 & ot_center2.

ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

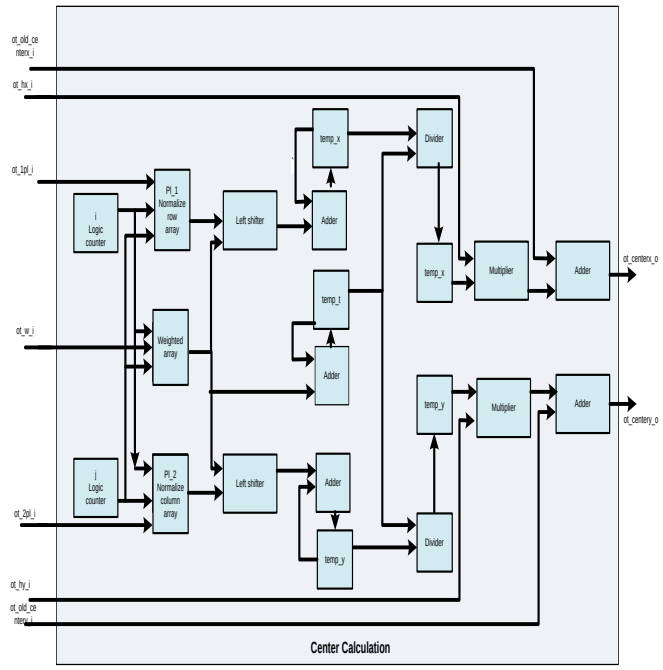

Figure 5: Center calculation

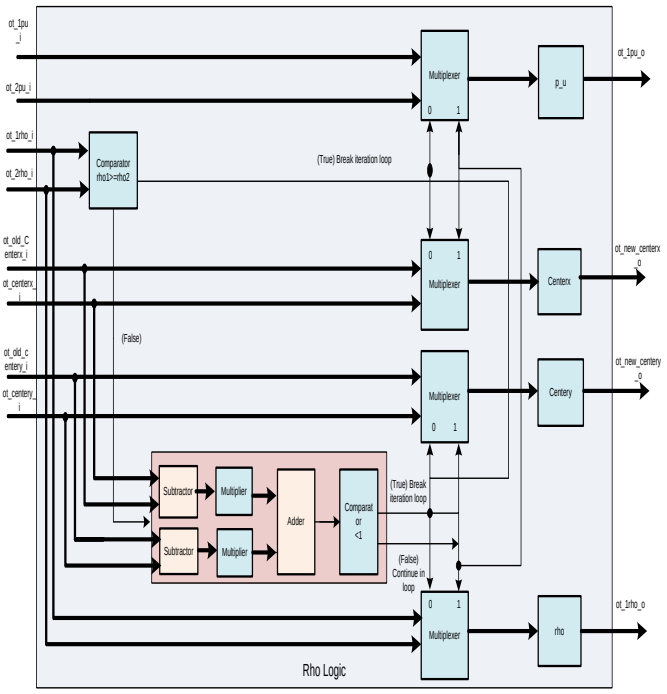

Figure 6: Rho logic

B. Rho logic

The inputs of Rho Logic Block are ot_pu1, ot_pu2, ot_rho1, o_rho2, ot_centerx, ot_centery, ot_old_centerx, ot_old_centery. Bhattacharyya coefficient of 2nd frame for old center (Ot_rho1) and Bhattacharyya Coefficient of 2nd frame for new center (ot_rho2) are compared. If ot_rho1 is greater than ot_rho2, then ot_old_center is final center for this frame and iteration loop is broken. If ot_rho2 is greater then ot_rho1, then sum of squares of ot_center1 - ot_old_center1 and ot_center2 - ot_old_center2 is calculated. If result is less then 1, then final center is center value and iteration loop is broken. Both conditions are not satisfied then ot_pu2, ot_rho2 and center (ot_centerx & ot_centery) are feedback to iteration loop.

The iteration value is looped 20 times, which is maximum value for this algorithm. Iteration is done because at first instant it didn't give final center for that frame and moreover, object is displaced in next consecutive frames but to know how much it moved and in which direction it moved, iteration value is taken.

C. Weighted array

The inputs of weighted array calculation are ot_centerx, ot_centery, width (ot_hx), height (ot_hy), ot_bins, histogram of frames (ot_pu1) and reference Histogram (ot_qu). Image model from R, G, B values is obtained by using Rect_crop. Then, binwidth is calculated by dividing 256 by bins value, which is a input value. For normalization of R, G, B values, divide them by binwidth. By doing so, R, G, B values come into limit of bins value and decrease calculation cost.

Check histogram of frames (ot_pu1) for R, G, B values. If it is zero, then Weighted array value is also zero. Otherwise, Weighted array value is calculated by taking square root of division result of reference histogram (ot_qu) by histogram of frame (ot_1_pu).

ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

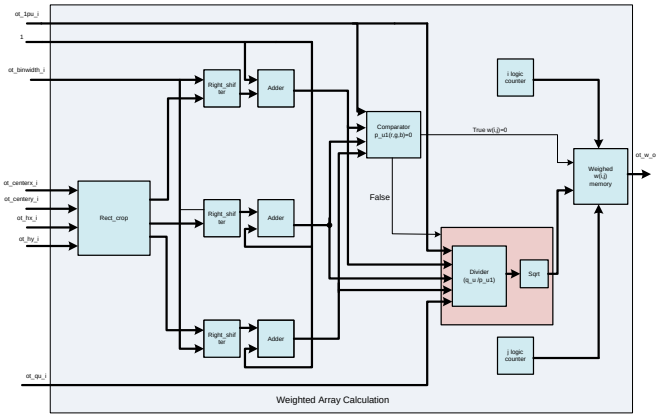

Figure 7: weighted array

VI. State Representation

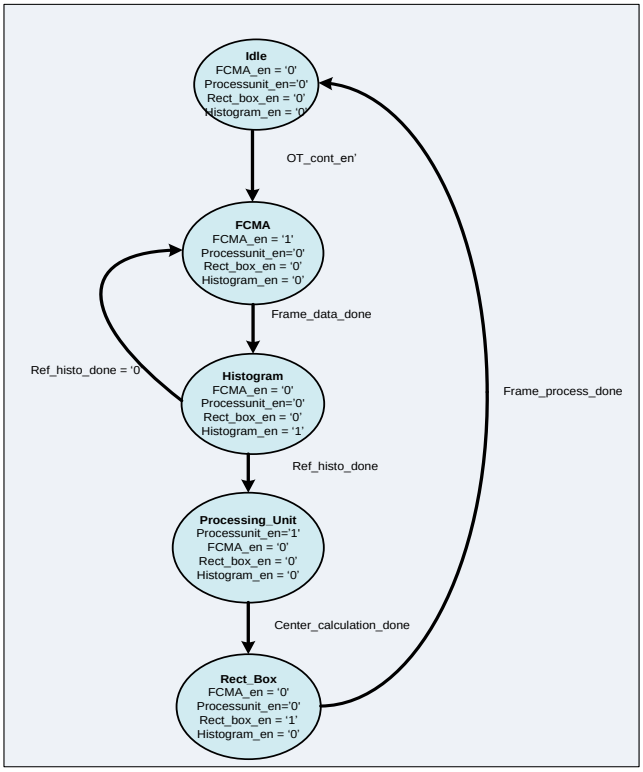

Figure 8: State diagram of OT controller

The OT controller only deals with the FCMA, histogram, Processing unit(internal logic after histogram calculation) and Rectangular box. When OT controller is enabled the it sets the FCMA and the image from the series of images is loded into it which now FCMA checks the status of histogram if enabled the histogram of the image is calculated after the calculation of histogram the FCMA sends other image and the histogram of that image is also calculated and this is compared and through processing unit and center and Rho value is calculated after itrating it to to get good output the rectangular box is made based on the kernel needed and set during the histogram calculation.

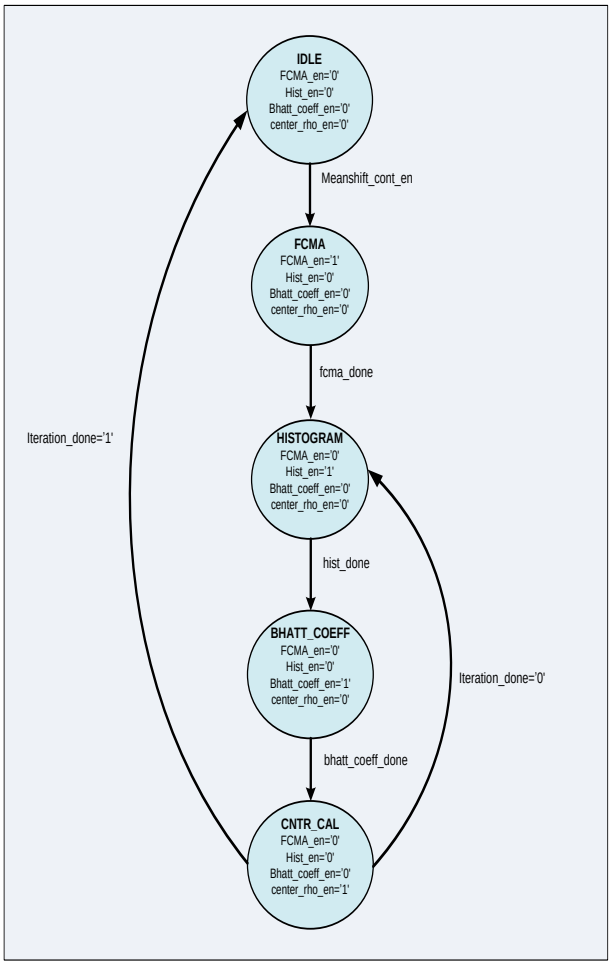

Figure 9: State diagram of Mean shift Algorithm

In mean shift algorithm state diagram the internal processing unit is explained in much detail after the calculation of histogram the the values are stored in buffer and then a type of comparison that is calculation of bhattacharya coefficient is done this bhattacharya coefficient act as the threshold for the image and based on this center is calculated. This process is iterated almost twenty times to get the perfect output of where the object is shifted and how it moves. When the comparison is done it again passes the control to the FCMA to load the next image and the comparison is done with the new image loaded.

VII. Conclusions & Results

ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

Considering all the above descriptions the Mean shift algorithm when implemented using verilog HDL the real problem was to build the HDL for multiplication, division & square-root through verilog the results which were compared through MATLAB was really proving the working of Mean-shift algorithm. The difference was only that the MATLAB output was with using floating point and verilog output was using fixed point.

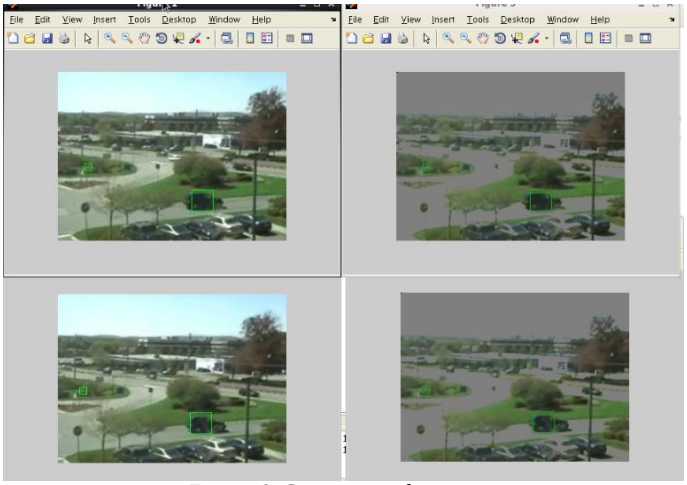

Figure 10: Comparison of outputs

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey", ACM Computing Surveys, Vol. 38, No. 4, Article 13, Publication date: December 2006.

[2] D. Ballard, and C. Brown, *Computer Vision*. Prentice-Hall 1982.

[3] D. Comanciu, V. Ramesh, and P. Meer, Kernel-based object tracking. *IEEE Trans. Patt. Analy. Mach. Intell. 25*, 564–575 2003.

[4] J. Wang, and E.Adelson, Representing moving images with layers. *IEEE Image Process. 3*, 5, 625–638, 1994.

[5] N. Xu, and N. Ahuja, Object contour tracking using graph cuts based active contours. In *IEEE International Conference on Image Processing (ICIP)*. 277–280 2002.

[6] Samir Palnitkar, Verilog HDL A Guide to Digital Design & Synthesis. © 2003 Sun Microsystems, Inc. USA.

[7] eInfochips. "Design Specification document for the Object Tracking.".

BIBILOGRAPHY

Rahul V Shah is currently working with eInfochips as Director Customer Solutions. In his five years at eInfochips he has played various roles starting with being a team member and growing all the way to be more customers focused and heading the IT infrastructure at eInfochips.

Rahul firmly believes that the growth of any business is purely based on the customer understanding and coming up with solutions to create a win-win scenario. Prior to joining eInfochips, Rahul has acquired good technical background after working in companies in United Status of America like IBM, Chameleon Systems and Sonics Incorporation. He has not only worked with multi-national companies across US and Europe but also ventured out to be the first employee of another startup in service domain. He also provides innovative solutions at eInfochips to create more differentiators and provide a good value add to customers. His recent article in Electronics Design Strategy News (EDN US) talked about the power challenges which will be faced by industry and the need to come up with a standard approach of managing the low power designs.

Rahul is also a part of Board of Studies in Nirma University and Dharamsingh Desai Institute of Technology. He is actively involved with the education system to enable the industry and education university relationship to enable the next generation with latest market trends and demands. He recently gave a lecture on "Future Careers '09" to student community supported by Computer Society of India (CSI)

Rahul holds a Bachelor of Electronics and Communication Engineering degree with a Gold Medal from Dharmsingh Desai Institute of Technology.

Few Articles

Xilinx Xcell Journel : Power Beast in Consumer Handheld
EDN US : Power Management
EETimes India Verification Management
FPGA Journal: ASIC Verification with FPGA Verification Components
SoC Central : Image Processing Applications

Amit Jain is currently working with eInfochips as ASIC Design Engineer. He obtained a B.E. degree in Electronics and Communication Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal in 2008 and M.Tech. in Communication Systems from Sardar Vallabhbhai National Institute of Technology, Surat in 2011. His fields of interest are Image Processing, Digital design with FPGA.

Rutul B. Bhatt is currently persuing his M.E. from Gujarat Technology University and he also Trainee at e-infochips Ahmedabad. He has obtained his B.E. From Veer Narmad South Gujarat University, Surat. His field of intrest is ASIC Design and Verification.

Pinal Engineer received M.E. degree from M. S. university of Baroda in 2002. He is pursuing Ph. D. from IIT Bombay. He is currently assistant professor in the Department of Electronics & Communication Engineering at SVNIT, Surat. He is having total teaching experience of more than 10 years in field of Engineering & Technology. He is actively involved in field of embedded system for more than 8 years. His research interests include embedded systems, reconfigurable computing, parallel computing, image processing. He is senior member of IETE as well as member of IEEE.

Mrs. Ekata Mehul is working as Head of Training and Research Academy with eInfochips, India named eiTRA. She is currently undergoing her Ph. D. in the area of “Wireless Sensor Networks”. She has been working as an Academician for last 14 years and has finished her M. Tech with specialization in the area of Information & Communication Technology. She is author of couple of Papers at International level and also some of the Chapters on “Security in Adhoc Networks” & “Low Power Details”. She has been a dedicated teacher always and a helping hand to the society for the field of ICT. She is also an executive member for “Computer Society of India, Ahmedabad Chapter” and a life member for IETE and other professional bodies.