Moving Object and Face Detection Using FPGA

Hrishikesh R[1], Hemanth P[2], Gajaraj B Patil[3], Mahesh Kumar M[4], Anuradha V Rao[5]
[1,2,3,4] *UG, Department of Electronics & Communication Engineering, B.N.M. Institute of Technology, Bengaluru-560070*
[5]*Assistant Professor, Department of Electronics & Communication Engineering, B.N.M. Institute of Technology, Bengaluru-560070*

Abstract **- Surveillance plays an important role in the present digital world. The advent of new technologies is making the computational process much faster than the existing ones. FPGA implementation of the moving object and face detection enhances the surveillance system by providing the results using the concept of adaptive threshold. The concept of adaptive threshold is used, so that the input comparison is done many times to produce an efficient and clear output. FPGA works at a greater speed (high clock speed) compared to other technologies for implementation. The input images are passed through the Gaussian filter. The HAAR Transform is applied on Gaussian filter's outputs and considered only LL band for further processing to detect object/face. The modified background subtraction algorithm is applied on LL bands of input images. The adaptive threshold is calculated using LL-band of reference image and object is detected through modified background subtraction algorithm. The detected object is again passed through the second Gaussian filter to get final good quality object. The face detection can be performed using an additional matching unit along with object detection unit. The reference image is replaced by face database images in the face detection. Moving object and face detection system is designed using Xilinx Spartan 6 LX45 FPGA kit. Using FPGA for object and face detection increases the speed of computation and makes the system more efficient. It has applications in various fields such as traffic systems, hospitals, banks etc.**

Index Terms **- HAAR, LL band, VHDL, adaptive threshold, system generator.**

I.INTRODUCTION

In the advanced surveillance systems make wider use of visual information, which gives the fullest opportunity to assess the situation in locations monitored by the control system. Such a system usually consists of multiple cameras (analog or digital), each of which generates a data stream. Information from the cameras is transmitted to the computer center, where the fundamental phase of image processing and analysis takes place and also all video data is stored. Because of the computational complexity of algorithms, high-powered computing units, among others supercomputers and additionally, hardware accelerators, e.g., GPU are used. The type and nature of the calculations on the visual data enables the utilization of reconfigurable computers supported FPGA devices.

Moving object and Face detection using FPGA enhances the surveillance system by providing the results using the concept of adaptive threshold. Adaptive thresholding is a method where the edge value is calculated for smaller regions and thus, there'll vary threshold values for various regions. Face detection principles find its applications in various fields such as video surveillance, criminalities, teleconferencing, robotics etc.

The biometric system in general, has three sections to recognize a person viz., pre-processing, feature extraction and matching section. In pre-processing section, the images are resized, color conversion, noise removal etc., are performed to reinforce quality of images. The features like mean, variance, principal component analysis and standard deviation are extracted in spatial domain by directly manipulating enhanced image. The transform domain features are extracted from Discrete Wavelet Transform (DWT). Features also are extracted by fusing spatial and transform domain features for better identification.

II. LITERATURE SURVEY

[1] Arvind Kumar and Sartaj Singh Sodhi, proposed a Gaussian filter, a Median Filter and a denoising auto encoder for noise removal. Gaussian filter is a linear type of filter which is based on Gaussian function.

[2] Vrushali R. Pagire and Anuradha C. Phadke, proposed a system which is implemented to detect foreground object in any environmental condition and the model is implemented on FPGA board. Real time foreground object detection is a very complex task. Proposed system detects object by background subtraction algorithm based on dynamic thresholding.

[3] Akshatha Venkatesh, Priyanka Karanth M N and Kaveri Talawar, proposed an efficient FPGA based reconfigurable architecture of moving object detection using Adaptive threshold method is proposed.

[4] Vidya P Korakoppa, Mohana, H. V. Ravish Aradhya proposed a more area efficient architecture for moving object and face detection using adaptive threshold. Here instead of detecting the moving body, only face is considered for detection.

[5] Sateesh Kumar H.C, Sayantam Sarkar, Satish S Bhairannawar, Raja K.B. and Venugopal K.R. proposed FPGA implementation of object and face detection using adaptive threshold. Input image and reference images are passed through Gaussian filters to remove noise and DWT is applied to generate LL band coefficients. The modified background subtraction is used along with adaptive threshold on two LL bands to detect an object. The final object is obtained after passing output of modified background subtraction through Gaussian filter.

III. IMPLEMENTATION

In this section we will discuss about the tools, mathematical formulae required for implementation and the board required.

- Software tools: Xilinx 14.5, Matlab 2012a, System generator tool of Xilinx 14.5 which is configured with Matlab 2012a.
- Development Board: Spartan 6 LX45 (Digilent Atlys Development Board)

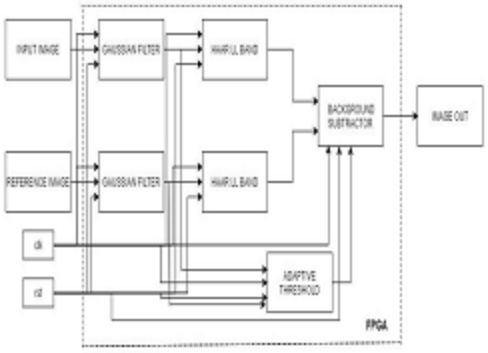

Figure 1: Architecture of Moving Object Detection

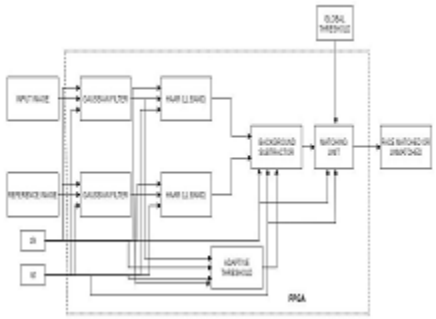

Figure 2: Architecture of Face Detection

The proposed architecture for Moving Object Detection and Face Detection is shown in Figure 1 and Figure 2 respectively.

A. Preprocessing

The Inputs to the preprocessing stage in this project are the video frames given as images to the preprocessing stage. They include the reference image or the background image and Input image. This project is applicable only for non-changing backgrounds only. The preprocessing stage includes various processes such as RGB to Grey scale conversion, resizing (Converting the input image to 256*256), Transpose (performing the transpose of the 256*256 image matrix obtained after resizing), conversion of 2D image to 1D and a buffering for storing of the pixel values. These are done using the standard Simulink blocks available in the system Generator Block set. In general, the random noise (Gaussian noise) is encountered in real time images. In order to eliminate this noise a Gaussian filter has been used.

B. Gaussian Filter

In Figure 1 and Figure 2 Gaussian filters are used to remove the random noise and the frequency edges which are because of variations in the light intensities

and natural calamities such as fog, rain etc. The impulse response of the Gaussian filter is shown in Figure 3 and for a two-dimensional filter is given by Equation (1).

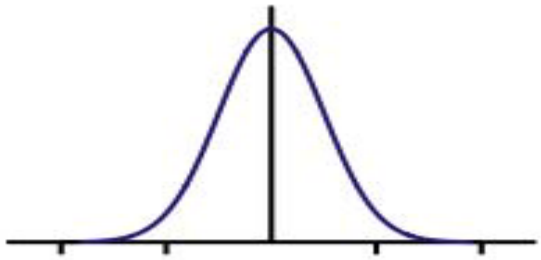

Figure 3: Impulse response of Gaussian filter

$$f(x,y) = \frac{1}{2\pi\sigma^2}.e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (1)$$

Where, x is the distance from origin along the horizontal axis

Y is the distance from origin along the vertical axis and σ is the standard deviation of Gaussian distribution and is unity.

From Equation 1, the standard Gaussian mask can be derived by taking various values for x and y, where x, y ∈ (-1,1) for a standard 3*3 matrix. The obtained matrix is given in Equation (2).

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \quad (2)$$

Since the order of the Gaussian mask considered is 3*3, the same size pixel matrix of the image is needed in order to filter the image. This can be obtained with the help of Moving window architecture as shown in Figure 4.

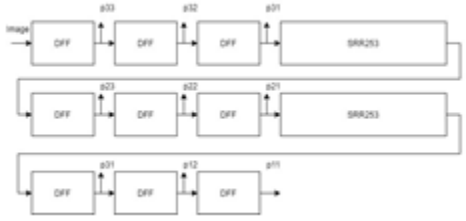

Figure 4: Moving Window Architecture

The moving window consists of two shift right registers (SRR253), which works on principle of FIFO (First in First out) and nine D-Flip-flops, which are used to obtain the 3*3-pixel matrix of the image. The SRR253 is in turn made up of arrays of D-Flip-flops as shown in Figure 5. If a single input is given, then it will appear at the output after 253 clock cycles.

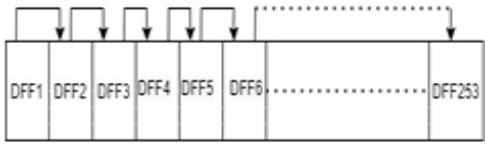

Figure 5: Array of D-Flip-flops constituting SRR253.

The input image is of size 256*256. During the first 256 clock cycles the entire first row of the image matrix is stored in the first row of the moving window. After next 256 clock cycles the entire second row of the image matrix is stored in the first row of the moving window and the previously stored vales in the first row of the moving window is moved to the second row if the moving window. Again after 3 clock cycles the first three elements of the third row of the image matrix is sent into the first three D-flip-flops and after corresponding shifting the first 3*3 value of the image matrix is obtained.

So, if a single input is given to the moving window it appears at the output after 515 clock cycles (256 clock cycles of the first row + 256 clock cycles of the second row + 3 clock cycles of the DFF's). Finally, the 3*3 image matrix is obtained and is represented in Equation (3)

$$\begin{bmatrix}p33 & p32 & p31\\ p23 & p22 & p21\\ p13 & p12 & p11\end{bmatrix} \quad (3)$$

The Gaussian filtering is carried out by multiplying the Gaussian mask of Equation 2 with the obtained image matrix from moving window of Equation (3). The multiplication is shown below

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} * \begin{bmatrix}p33 & p32 & p31\\ p23 & p22 & p21\\ p13 & p12 & p11\end{bmatrix} \quad (4)$$

Equation (4) represents filtered image pixel values. The division by 16 is performed by right shift operation four times. Any random noises due to variation in the light intensity is overcome by use of Gaussian filter.

C. HAAR (DWT)

The HAAR Transform is applied on to the Gaussian filter output. The purpose of using HAAR transform is to obtained the LL Band coefficients of the image. The LL Band is used for calculation of adaptive threshold value because major characteristics of the image are located in LL Band. Further it is used to compress image to reduce the computational complexity and memory size. It also provides an effective wave to perform both lossy and lossless image compression.

The architecture of HAAR Transform is shown in Figure 6.

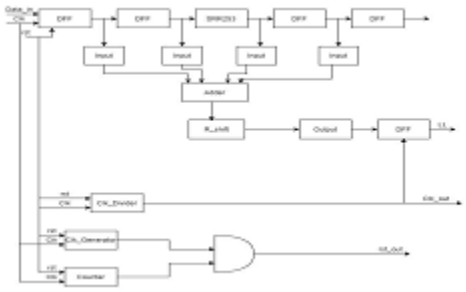

Figure 6: Architecture of HAAR

The size of the input matrix is decreased from 3*3 to 2*2. Four D-Flipflops and one SRR253 is used in order to obtain the 2*2 matrix as given in Equation (5) and the LL Band coefficients are computed using the Equation (6)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (5)$$

$$LL = \frac{1}{4}[(a+b)+(c+d)] \quad (6)$$

Figure 6 consists of various blocks such as Input, Output, Right shift, adder and a controller unit. The input block is a data increment block in which the input is incremented into 12 bits. Corresponding pixel values are added using the adder. Then the division by 4 with respect to the Equation 6 is carried out with the help of Right shift operation by twice (>>2). The output is the data decrement block in which the data is converted back to 8 Bit. The controller consists of a clock divider, clock generation and a counter which together functions as a controller unit and is used for clocking of the D Flip-flop.

D. Background Subtraction

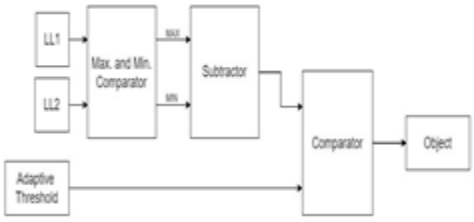

Figure 7: Architecture of Background Subtraction

After the calculation of adaptive threshold. The output of HAAR block which are the LL band co-efficients of input image and the background image are given as input to the background subtraction block shown in Figure 7. This block calculates the difference of two images using pixel values of both the images respectively and continues for all the pixels in the frame. Similarly, all the remaining frames of the video are processed. The mathematical expression for Background Subtraction is given in Equations (7) & (8). Once the adaptive threshold (ATi) is calculated, it is compared with the background subtracted value and if the background subtracted value is greater than the adaptive threshold value, then the value is recognized as foreground pixel and it advances to the output.

$$LL_j = \left|[(LL2)_j - (LL1)_j]\right| \quad (7)$$

LL2 is the LL band co-efficients of the input image

LL1 is the LL band co-efficients of the reference image

$$BS = \begin{cases} LL_j \ ; \ if \ LL_j \geq AT_i \\ 0 \ ; \ Otherwise \end{cases} \quad (8)$$

E. Adaptive Threshold

The adaptive threshold block is the core of this project and the architecture is shown in Figure 8. It is used to calculate the threshold value which is primarily used to detect the object/image efficiently. When the input pixel value is greater than the threshold value then the corresponding values are sent to the output else the output will be zero. The difference of pixel values (S) between the reference and input image is given in Equation (9) and (10).

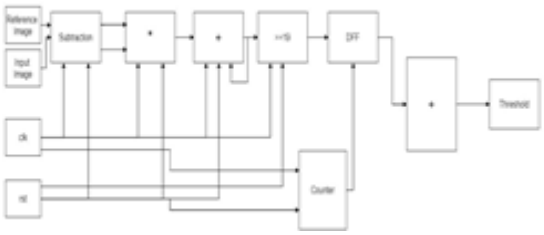

Figure 8: Architecture of Adaptive Threshold

$$S = \frac{[A_1 - B_1]^2 + [A_2 - B_2]^2 + \cdots + [A_N - B_N]^2}{8N} \quad (9)$$

$$S = \sum_{i=1}^{N} \frac{[A_i - B_i]^2}{8N} \quad (10)$$

Where, N is the dimension of input image pixel values (N=256x256).

A1, A2…AN are the actual input image pixel intensity values after filtering.

B1, B2...BN are the reference image pixel intensity values after filtering.

The Adaptive Threshold value (ATi) is calculated using LL coefficient values of reference

image (LL2) and S are given in Equation (11) and (12).

$$AT_i = S + (LL\ Coefficients\ of\ Reference\ Image)_i \quad (11)$$

$$AT_i = S + (LL2)_i \quad (12)$$

Where, i= 1 to 16384 (i.e. 128*128)

F. Matching Unit

Matching unit is one of the important blocks in face detection process. It consists of the following structure as shown in Figure 9.

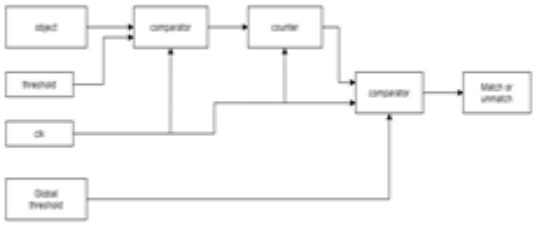

Figure 9: Architecture of Matching Unit

The test face image is subtracted from the input face image and the similar portions are cancelled out by the background subtraction module. A subtracted pixel value is compared with the chosen threshold value and if that input pixel value is lower than that of threshold value, then the counter is incremented. If the value of the counter exceeds the set global threshold value, the face is claimed to be matched, else the face is claimed to be unmatched.

IV. RESULTS

The HDL (VHDL) code for all the blocks i.e., Gaussian filter, matching unit, HAAR, adaptive threshold and background subtraction blocks were developed in Xilinx 14.5 and the dumped on the Xilinx Spartan 6 LX 45 FPGA board. The Figure 10 and Figure 11 shows the system generator model for moving object detection and face detection respectively.

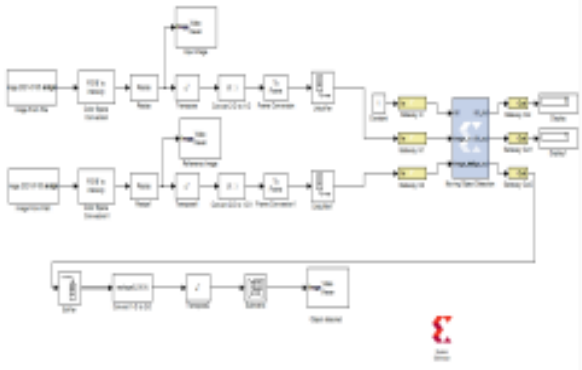

Figure 10: System Generator Model for Moving Object Detection

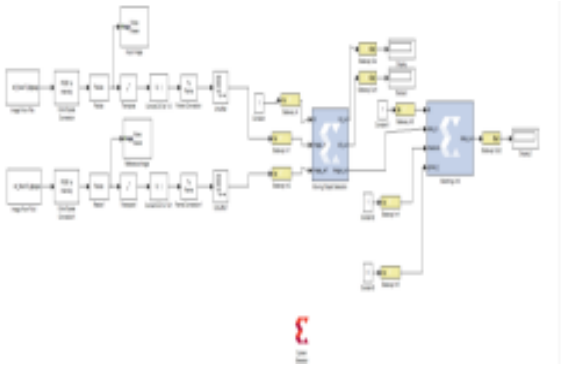

Figure 11: System Generator Model for Face Detection

A. Moving Object Detection Results

Since this projected is applicable for only static backgrounds, the reference image is the video frame having a static background and the input image is the actually video frame. The pre-processing is done with the Simulink blocks that are available in Simulink library. Both these images are converted to grey scale images, resized to 256*256. The obtained pixel values after resizing are taken transpose. The image matrix is then converted from 2D to 1D followed by frame conversion and finally given as serial input to the black box with the help of buffers. The black box in Figure 10 contains the HDL code for moving object detection. After processing through the black box the obtained values are sent to the Simulink block that performs almost opposite processes as that of the stated above processes. Figure 12 shows the video frame with static background, Figure 13 shows the actual input frame and Figure 14 shows the detected object image.

Figure 12. Static Background Image

Figure 13: Actual Input Image

Figure 14: Detected Object Image

B. Face Detection Results

The face detection model includes an additional unit known as the matching unit as shown in Figure 11. The matching unit is described such that if both input images are the face images of the same person, then the output will be '1' indicating the match case else the output will be '0' indication the unmatched case. The threshold value was some positive integer 25 for the test face as shown in Figure 15 and Figure 16 which are face images of the same person with slight differences.

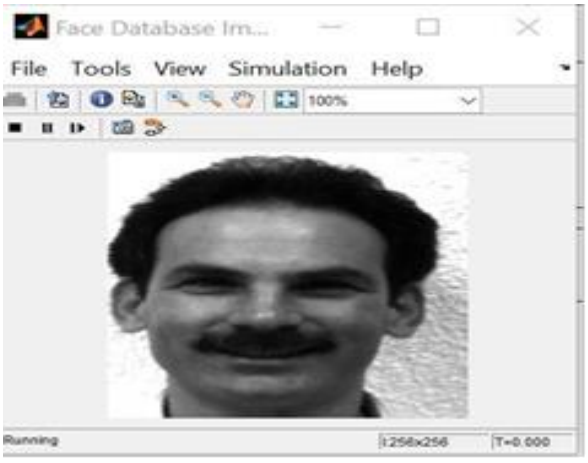

Figure 15: Face image in Database

Figure 16: Input Face Image

C. Hardware Results

The hardware setup involves connecting FPGA Spartan-6 LX45 to the system through JTAG as shown in the Figure 17. The JTAG cable assists in dumping the VHDL code written in Xilinx ISE 14.5 into the FPGA kit.

Figure 17: Hardware connection for the FPGA Spartan 6 LX45

After finishing the hardware setup, that is connecting the FPGA to the computer thought the JTAG cable as shown in the above Figure 17. After dumping the code on the FPGA, a green light glows indicating successful dumping of the code. Then the hardware co-simulation is done by the hardware co-simulation model as shown in the Figure 18.

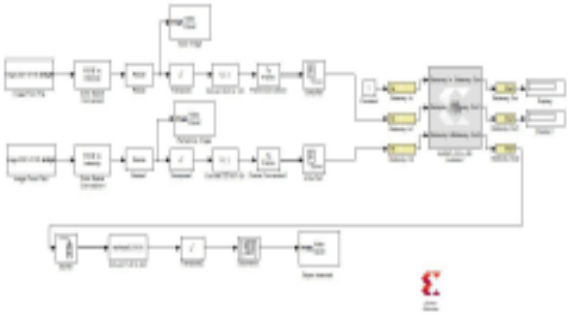

Figure 18: Hardware co-simulation model for moving object detection

The result obtained have utilized less number of slice registers when compared to the earlier work as shown in Table 1 and Table 2.

TABLE 1. FPGA logic utilization comparison for moving object detection

FPGA Logic utilization	[5]	[4]	Proposed
Number of slice registers	365	320	203

TABLE 2. FPGA logic utilization comparison for face detection

FPGA Logic utilization	[5]	[4]	Proposed
Number of slice registers	861	537	324

V. CONCLUSION

This project based on video/image processing using HDL can be used in many surveillance applications. Hardware co-simulation using FPGA will decrease the computational time, thereby the efficiency of the model to detect moving object and the face detection is increased. The FPGA logic utilization i.e. LUT's has been reduced to 203 compared to earlier works. Thus, decrease in the slice registers will result in improved area efficiency and a faster hardware accelerated model is obtained.

REFERENCES

[1] Mukul Yadav, Ritik Koul, and Kriti Suneja " 7th International Conference on Signal Processing and Integrated Networks (SPIN)",2020.

[2] Sharath S, Rangaraju H G " Efficient FPGA Implementation of Human Detection from Video Sequences" International Journal of Innovative Technology and Exploring Engineering, Volume-9 Issue-6, April 2020.

[3] Arvind Kumar and Sartaj Singh Sodhi " Comparative Analysis of Gaussian Filter, Median Filter and Denoise Autoencoder", 7th International Conference on Computing for Sustainable Global Development, 2020.

[4] Vidya P Korakoppa, Mohana, H. V. Ravish Aradhya "An Area Efficient FPGA Implementation of Moving Object Detection and Face Detection using Adaptive Threshold Method" 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017.

[5] Sateesh Kumar H.C, Sayantam Sarkar, Satish S Bhairannawar, Raja K.B. and Venugopal K.R. "Fpga Implementation of Moving Object and Face Detection Using Adaptive Threshold", International Journal of VLSI design & Communication Systems (VLSICS) Vol.6, No.5, October 2015

[6] Murad Qasaimeh, Joseph Zambreno and Phillip H. Jones " A Modified Sliding Window Architecture for Efficient BRAM Resource Utilization", IEEE International Parallel and Distributed Processing Symposium Workshops, 2017.

[7] Rajesh Rohilla, Aman Raj, Saransh Kejriwal, Rajiv Kapoor "FPGA Accelerated Abandoned Object Detection" International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), 2016.

[8] Divya Meena and Ravi Sharan " An Approach to Face Detection and Recognition", IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAI), 2016.

[9] Sanjay Singh, Atanendu Sekhar Mandal, Chandra Shekhar and Anil Vohra "Memory Efficient VLSI Implementation of Real-Time Motion Detection System Using FPGA Platform", Journel of Imaging, 2017.

[10] Sameh A. Napoleon, M. Abd Elnaby, Mohamed El- Darieby, Mohamed Maher Ata "Traffic Video Enhancement based Vehicle Correct Tracked Methodology" Using modified background subtraction for detecting vehicles in Videos", International Journal of Advanced Intelligence Paradigms, March, 2017.