ISSN: 2321-7782 (Online)
Volume 2, Issue 9, September 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

# Detecting Moving object using Background Subtraction Algorithm in FPGA

**Poonam Gujrathi**[1]
Department of E&TC Engineering
DYPCOE, Akurdi
Pune – India

**R. Arokia Priya**[2]
Prof., Department of E&TC Engineering
DYPCOE, Akurdi
Pune – India

**Dr. P. Malathi**[3]
Department of E&TC Engineering
DYPCOE, Akurdi
Pune – India

***Abstract: Image as well as video processing applications like visual surveillance, vehicle tracking, traffic monitoring etc have real time constraints to identify moving object in the given scene. Several hardware as well as software approaches are proposed which suits the real time moving object detection. However due to low cost and high performance, hardware based approaches are preferred over software based approaches. In this paper, we propose a FPGA based hardware architecture which is capable of detecting moving object using background subtraction algorithm. The steps of the process involves - (a) storing grey level image (aka background) in FPGA, (b) Applying color reduction on current and background image, (c) Subtract both filtered images using image subtraction, (d) Calculate gravity center of object and transfer it to display using RS-232 interface,(e) Identify object's edges using Sobel edge detection. Implementation of edge detection along with motion detection algorithm allows the system to be extended where in object's could be classified based on the shapes.***

***Keywords: Background Subtraction Algorithm, Erosion, FPGA Motion detection, Segmentation, Sobel edge detection.***

## I. INTRODUCTION

Real time motion detection has got a lot of attention in last decade. Several approaches based on hardware as well software are proposed. FPGA based hardware approaches suit the real time constraints. Applications like vehicle tracking, visual surveillance, autonomous navigation etc requires approaches that consume less power and are high in performance [3]. FPGA based hardware approaches serves this need.

Motion detection algorithm requires a series of images as input or video input to identify moving region in the scene. Following are three most commonly used techniques for motion detection – (a) Temporal difference, (b) Background Subtraction and (c) Optical flow [1] [6]. This paper primarily will focus on Background Subtraction Algorithm. We will also see how this algorithm could be utilized for posterior recognition.

In Background subtraction algorithm, first a reference image is identified which will be stored in memory. Then the video input is converted into individual frames/image. Each image (pixel by pixel) is then subtracted from the reference image by using image subtraction. There are two styles of background subtraction – adaptive and non adaptive [4]. In Non adaptive, the reference image is fixed. The image is loaded offline. Assume that at the time of capturing the background image if there is any illuminating or stationary object in the scene, then those objects will affect the real background to be capture. Thus this style of background subtraction is more error prone. To deal with such problems, use adaptive method. In adaptive method background image is constantly updated by simply averaging out the frames over a period of time. This approach is therefore more effective as it eliminates the problems like frequently moving object or temporarily stationary objects etc., which in general would have become the part of the background image. Because of its simplicity and less computational requirement Background subtraction algorithm is widely used technique in field of motion detection.

In Temporal difference, consecutive images are subtracted. Instead of using a reference image every current image becomes the reference image for next in line image. This dynamic update of reference image allows this approach to easily adapt to scene changes. This approach is very simple and cost effective. Computation requirement is also very less. Posterior recognition with temporal difference approach is difficult thus failing to identify moving object's shape.

In Optical flow, for each pixel a motion vector is computed. Entire image is seen as a vector field. The brightness of the pixel represents the motion vector of that pixel. Across image the region in which there is a brightness change is considered as part of the moving object. This approach produces decent performance; however the implementation of the algorithm is very complex, as more than one image needs to be stored, thus increasing the memory requirements of the system, in-turn yielding high cost.

Background Subtraction and temporal difference approaches are based on Image subtraction method [3]. Let I be the luminous intensity of the pixel in the image. Suppose X and Y are the horizontal and vertical coordinate points of the pixel at time t. So at any given point of time, the difference ΔI between reference image and current image is given as follows:

$$\Delta I = | f(x, y, t) - f\ reference(x, y) | \qquad (1)$$

Image subtraction is based on the fact that the object that was not part of the background image will cause significant change in the pixel's luminous intensity. The objects that were part of background image will result in zero or slight variation in luminous intensity. The input images might contain noise, and this may cause some variation in pixel's luminous intensity. Hence some threshold value is required to classify the difference ΔI as either the part of moving object or noise. Any change in pixel value greater than threshold is set as 1 which means part of moving object whereas any change in pixel value less than threshold is considered as noise or part of background image and its pixel value is put as zero.

In section II, we provide a high level overview of the past research. In section III we discuss how our system is implemented. Section IV describes the results obtained so far. Section V covers the conclusion followed by references.

## II. RELATED WORK

Image as well as video processing applications like visual surveillance, vehicle tracking, traffic monitoring etc have real time constraints to identify moving object in the given scene. Several hardware as well as software approaches are proposed which suits the real time moving object detection. However due to low cost and high performance, hardware based approaches are preferred over software based approaches. In paper [3], a hardware approach for identifying motion detection on transit roads was proposed. Experimental results carried in the work shows that hardware approaches that uses reconfigurable hardware are 7.5 times faster than normal software approaches.

Y Dedeoglu in his work has presented a smart visual surveillance system [1] which detects real time motion in the scene and classifies the objects based on its shape. The system is also capable of tracking object. The smart system is also able to adjust to changing illumination conditions. The proposed system works well with outdoor as well as indoor environments.

D.Markis in his work [2] suggests that the motion detection approaches can easily be extended to track object's path. Each moving object's trajectory is stored in the system. Using this object's trajectories, system is trained to identify known routes in the given scene and any unknown route would generate an alarm thus alerting about suspicious activities in the given scene.

Cost and area of the hardware implementation plays a major role in deciding with the hardware matches the need of the real time motion detection applications. A lot of research is done which basically focuses on reducing the cost and area of the hardware implementation and at the same time improves the performance of the hardware implementation. Saad et al. [4] in his work proposed an FPGA based hardware implementation which uses on temporal difference algorithm. The system is capable of processing images at very high rate (around 1100 fps) in a single FPGA chip of low cost thus suiting a wide range of the real time motion detection applications.

Similarly, in [5] a chip based design to detect moving object is proposed which uses digital camera module. The mechanism described in the paper uses one page comparison algorithm instead of two page comparison. This significantly reduces the memory and logic gates requirement of the FPGA implementation. The drawback of this mechanism is that it requires a digital camera of higher resolution. Experimental results however show that the performance of the system is satisfactory enough which complement the requirement of using high resolution camera.

In our paper, we have proposed a low cost motion detection system that works well with real time motion detection applications. The proposed system not only identifies the moving object but also identifies object's shape thus helping in posterior recognition, unlike [4] which fails to provide posterior recognition as the underlying algorithm is temporal difference. Also the proposed system is of low cost unlike [5] which requires a high cost camera with higher resolution. The proposed system could be easily extended to implement object tracking [2] and object classification [1] however the same is not implemented in current piece of work.

## III. System Implementation

In this section, we will describe the underlying hardware followed by block diagram of the system and some modules implemented in MATLAB which basically does some preprocessing on the input video.

### Hardware

The proposed system is based on Spartan-3 EDK Board (XC3S200). The specification of the board includes a 200 K gate Spartan board, 1 MB fast asynchronous SRAM, input and output devices, Flash JTAG programmable ROM. Thus this board provides a self development platform which is not only powerful but is also capable of executing designs targeted for new Spartan 3 FPGA. Software that we have used to carry out our work is Xilinx platform studio and Matlab. Background subtraction algorithm is implemented in system C and RS 232 module is implemented using VB-script.

### Video/Image Capture Module.

This module basically does the preprocessing on the input video. This module captures and converts the input video into individual frames. This module then converts each of these frames into its corresponding C style header files. This header files act as input to the background subtraction algorithm which is implemented in C. Header file basically contains the pixel intensity of each pixel in the frame. The module is divided into sub modules like (a) Video to Frame conversion (b) Frame to header file creation. Video to Frame conversion module extracts individual frames from the captured video. Each of these frames is transformed from color to grey level scale. This grey level frames are then passed to 'frame to header file conversion module' which generates the header file for its respective frame.

### Block Diagram

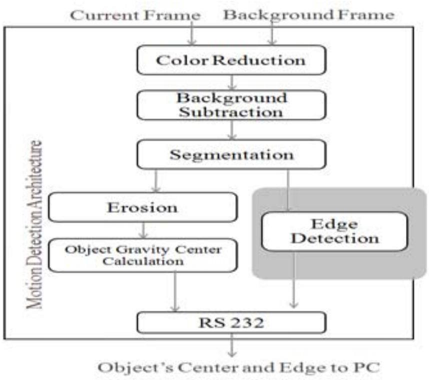



Fig1.Proposed Motion Detection Architecture

Fig.1 above shows the block diagram of the proposed system. The system is an extension to typical motion detection architecture using background subtraction algorithm [6] where the grey color denotes the module responsible for detecting object’s edge.

The proposed system could be easily extended to classify objects based on their shape. A brief overview of each block is provided below:

a) *Color reduction:* Our primary focus in this work is on gray scale image. Before image subtraction both current as well as reference image is converted into gray scale images. After color reduction, these images are converted into C style header file. These images are then loaded into the FPGA for image subtraction.

b) *The Background Subtraction and Segmentation Process:* This block is responsible for subtracting current image from reference image. This block provides one output pixel per clock cycle. Segmentation operation is then performed on the difference image. In segmentation each pixel value is compared with the predefined threshold value. Any difference in pixel value that is above threshold is considered as 1 else zero.

c) *The Erosion operation:* This block carries out the functionality of applying logic operation between the pixel of segmented image and a structuring element of 3*3. The 3*3 structuring element consists of all 1’s. If A(x,y) is the segmented image and B(i,j) is the structuring element then eroded image C(x,y) is given by

$$C(x,y) = \text{Minimum}\{A(x-i, y-j) * B(i,j)\} \qquad (2)$$

d) *The Gravity Center Calculation:* This block takes as input the eroded image and obtains the object’s center of gravity using approach defined in [6].

e) *The RS-232 Communication:* This block basically transfers the bit stream from the FPGA to PC for display.

f) *Object's edge detection:* This module implements sobel edge detection which is responsible to identify object's edge. The input to this module is segmented image.

**Sobel Edge Detection**

The Sobel edge detection algorithm takes a 3*3 convolution masks which are shown in Fig. 2 below.

| | | |
|---|---|---|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

Gx

| | | |
|---|---|---|
| +1 | +2 | +1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

Gy

Fig. 2: Sobel masks

One pair of mask Gx is used to estimates the gradient in the horizontal-direction (i.e. x direction for columns), whereas Gy is used to estimates the gradient in the vertical-direction (i.e. y direction for rows). In general, the size of the convolution mask is kept much smaller than the size of actual image. These masks are slided over the input image which manipulates one square pixel at any time. The masks are shifted one pixel to the right after every cycle. This process continues to the right until row end is reached. Once the row end is reached, the mask is then shifted to next row until all rows are parsed. This operation is applied to each frame to identify object’s edge in each frame.

## IV. Results

This section shows the output of the various blocks that we have implemented so far. We will also show the output of Video to Frame Module and Frame to header file conversion Module.

a) *Video to Frame Module:* Fig. 3(a) shows the graphical user interface on which the input video was played and Fig. 3(b) the converted frames are shown.

b) *Frame to Header conversion*: This module is responsible for generating C style header file out of the selected image. Before header file conversion color image is transformed into grey scale.

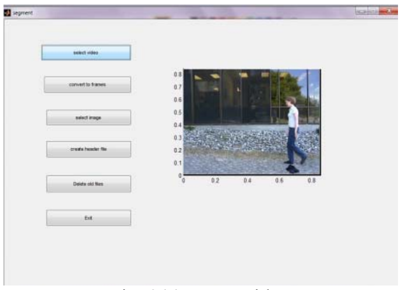


Fig. 3(a) Input Video

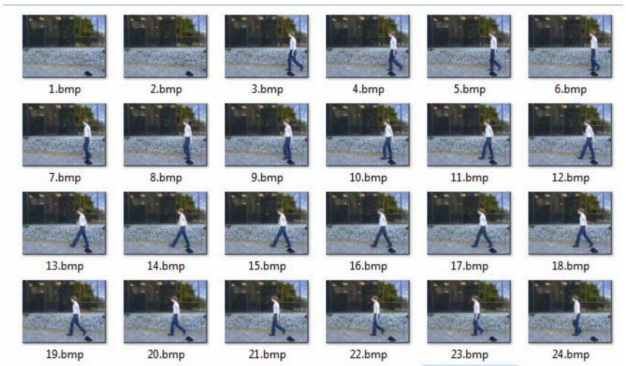
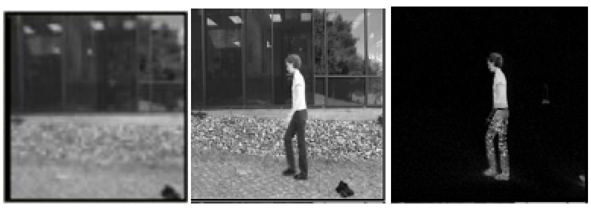



Fig. 3(b) Converted Frames

c) *Background Subtraction Algorithm:* As part of the test, we took two images. One reference image and other image of the moving object. The equivalent header files are provided as input to this module which performs the image subtraction. This module performs image subtraction i.e. subtracting array contained in current image header from the array contained in header file of reference image. The difference image obtained after image subtraction is shown in Fig. 4. The difference images are shown on PC using RS232 interface.



Background Current Difference

Fig.4. – (a) background image (b) current image (c) Difference image

d) *Segmentation:* The module takes as input the difference array. Using a predefined threshold value, segmentation operation is performed. Threshold used here is 80. Any deviation in pixel value which is greater than 80 is set as 255. 255 means white color in grey scale. Any pixel value less than 80 is put as zero. Zero means black color in grey scale. The segmented output is shown in figure 5(b).

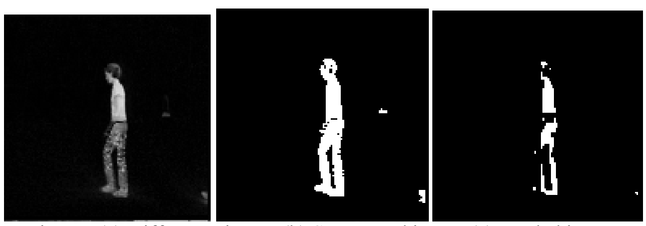


Fig. 5 – (a) Difference image (b) Segmented image (c) Eroded image

e) *Erosion:* The segmented array obtained in step d act as input to this module. This module implements formula shown in equation (2) to extract eroded image out of segmented array. The structuring element of 3*3 is used which contains all 1's. The eroded image is shown in figure 5(c). Output clearly shows that eroded image remove the edges from object boundary.

f) *Object Center of gravity Calculation* – Eroded image is used to compute the object center of gravity. Object centers are then stored in object center array. For sequence of images object centers of gravity for each image is stored and is used to plot the x, y coordinate of the center of gravity on the 2D graph.

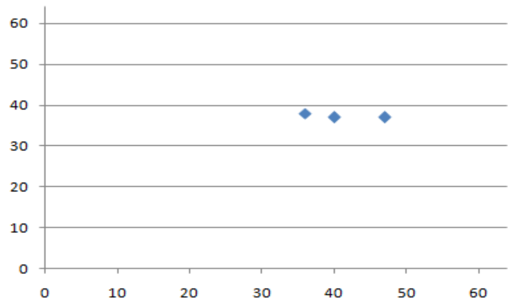



Fig.6 – Object's center of gravity plotted on 2 D graph.

Figure 6- shows the plotted graph. The graph is plotted by using object center of 4 images where first image was reference image and rest 3 images include moving object. The blue dots in the graph represent the object center of gravity for that image.

g) *Sobel edge detection:* Figure 7 (a) shows the input image of the edge detection operation. Input to the module is segmented image and as output the object's edge are detected as shown in figure 7(b). The flipped version is shown in figure 7 (c)

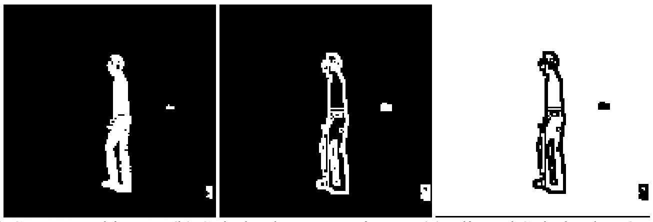


Fig. 7 – (a) Segmented image (b) Sobel Edge output image (c) Flipped Sobel Edge Output image

## V. CONCLUSION

This paper proposes modified motion detection architecture which is also capable of identify object's edge. The paper compares the background subtraction algorithm with its other counter parts algorithm. It also describes how this approach is different than that of the related work and how it incorporates the techniques from related work. In the modified architecture, object edge detection based on Sobel edge detection is added to the architecture. The proposed system could be extended to classify objects based on their shapes. Apart from this the hardware requirement of the system is very less as it doesn't involve high cost cameras and could be implemented using entry level Spartan 3 FPGA board as well. The proposed system could face challenges in hard real time systems where the delay between FPGA and PC could be a problem, however in soft real time system this approach would work well.

## References

1. Y. Dedeoglu, "Moving Object Detection, Tracking And Classifcation For Smart Video Surveillance", Bilkent University, Department of Computer Engineering and the Institute of Engineering and Science, August, 2004.
2. D Makris, "Visual Learning in Surveillance Systems", City University, Department of Electrical, Electronic and Information Engineering Machine Vision Group, London, May 2001.
3. G. S. Menezes and A. G. Silva-Filho, "Motion Detection of Vehicles Based on FPGA", Proc. IEEE VI Southern Conference on Programmable Logic (SPL), pp. 151-154, Brazil, 2010.
4. M. Saad, A. Hamdy and M. M. Abutaleb, "FPGA-Based implementation of a Low Cost and Area Real-Time Motion Detection", 15th International Conference of Mixed Design MIXDES, pp. 249-254, Pozna´n, Poland, 2008.

5. Ying-Hao Yu, Q. P. Ha, and N. M. Kwok, "Chip-based Design for Real time Moving Object Detection using a Digital Camera Module", The 2nd International Congress on Image and Signal Processing, CISP, 2009.

6. C Sanchez-Ferreira, J.Y Mori, C.H. Llanos, "Background subtraction algorithm for moving object detection in FPGA", Department of Mechanical Engineering, University of Brasilia, Brazil,IEEE,2012

## AUTHOR(S) PROFILE

**Poonam Gujrathi,** is pursuing her Master in Engineering degree in VLSI and Embedded System from D.Y.Patil College of Engineering, Pun, affiliated to University of Pune. She also holds Bachelor of Engineering in Electronics and Telecommunication from SSGMC, Shegaon. Prior to that she had worked in Embedded Microtechniques pvt Ltd. Pune for 2.5 years as hardware test engineer.