Implementation of Visual Object Tracking by Adaptive Background Subtraction on FPGA

Denna Paul
Electronics and Communication Department, VJCET
Mahathma Ghandhi University

Ranjini Surendran
Electronics and Communication Department, VJCET
Mahathma Ghandhi University

Abstract— **This paper introduces an improvement to available background subtraction method to detect moving object, based on adaptive background subtraction. First of all, a reliable background updating model based on statistical data is established followed by a dynamic optimization threshold method to obtain a more complete moving object. This adaptive background model along with dynamic threshold gives better noise immunity. Next a morphological filtering helps solve the background disturbance problems by eliminating the noise . At last, the moving objects are accurately and reliably detected across different frames of input video. Both background subtraction model and adaptive background model were simulated using MATLAB and implemented on Xilinx Spartan-III XC3S200 FPGA, with Microblaze processor using Xilinx EDK.The methods were compared and the experiment results confirm the accuracy of the proposed method in real-time scenarios.**

***Index Terms*—Background modeling, background subtraction, video segmentation, video surveillance, Image Processing.**

I. Introduction

Object tracking is the process of locating a moving object in time using a camera. The algorithm analyses the video frames and gives the location of moving targets within the video frame. In video surveillance systems, background subtraction is considered the first processing stage where objects in a particular scene is determined. It stands for a process which aims to separate foreground objects from a relatively stationary background. It should be processed in real time. Here pixel-by pixel difference between the current frame and the image of the background is found out followed by an automatic threshold. This type of whole body tracking has many applications such as video surveillance, military reconnaissance, mobile robot navigation, collision avoidance, video compression, path planning, among others. Most of these applications demand low power consumption, compact and lightweight design, and high speed computation platform for processing image data in real time. So we need to implement the algorithm on FPGA.

II. Algorithm

Three ways for detecting motion in image sequences: (a) background subtraction, (b) temporal difference and (c) optical flow. The most used algorithm is the background subtraction, due to the fact that it is not a computationally expensive algorithm and also presents high performance.

A. Background Subtraction

The idea of background subtraction is to subtract the current frame image from a reference image that models the background scene. Steps of the algorithm include Background modeling, Threshold selection and Subtraction. Background modeling step gives a reference image which represent the background. Threshold selection determines most appropriate threshold values to be used in the subtraction operation for a desired detection rate. Subtraction operation or pixel classification classifies the type of a given pixel, i.e., the pixel is the part of background or it is a moving object.

In the background training process, the reference background image and some parameters associated with normalization are computed over a number of static background frames. A statistical model of the background can be drawn on pixel-pixel basis.

$$d(x, y, t) = \begin{cases} 1 \text{ if } |f(x, y, t) - B(x, y)| > Td \\ 0 \text{ otherwise} \end{cases}$$

Mathematically, the background subtraction algorithm can be defined by (1) [1], where Td is a predetermined threshold, $f(x, y, t)$ is an image taken at time t and $B(x, y)$ is the reference image (or background). In the dynamic image analysis, all pixels in the motion image $d(x, y, t)$ with value ―1‖ are considered as moving objects in the scene [1].

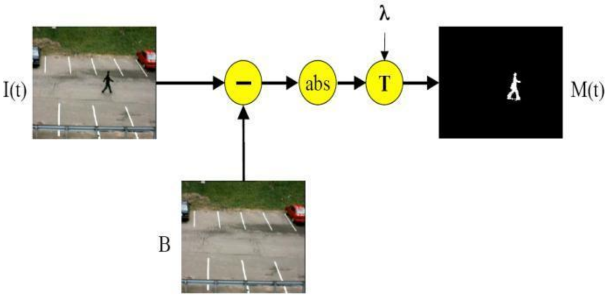


Fig. 1 Background Subtraction using fixed background

So far, we have discussed about a fixed background which has many drawbacks. Objects that enter the scene and stop continue to be detected, making it difficult to detect new objects that pass in front of them. Also if B moves both the object and its negative ghost are detected. This method is highly sensitive to changing illumination and unimportant movement of the background.

B. Adaptive Background Subtraction

For the background model can better adapt to light changes, the background needs to be updated in real time, so as to accurately extract the moving object. The update algorithm is as follows:

$$B(x, y, t) =\alpha f(x, y, t) – (1-\alpha)B(x, y,t-1)$$

where α ϵ **(0,1)** is update coefficient, in this paper α = 0.5. *f(x,y,t)* is the pixel gray value in the current frame. *B(x,y,t)* and *B(x,y,t-1)* are respectively the background value of the current frame and the previous frame. As the camera is fixed, the background model can remain relatively stable in the long period of time. Using this method can effectively avoid the unexpected phenomenon of the Background, such as the sudden appearance of something in the background which is not included in the original background. Moreover by the update of pixel gray value of the background, the impact brought by light, weather and other changes in the external environment can be effectively adapted. In detection of the moving object, the pixels judged as belonging to the moving object maintain the original background gray values, not be updated.

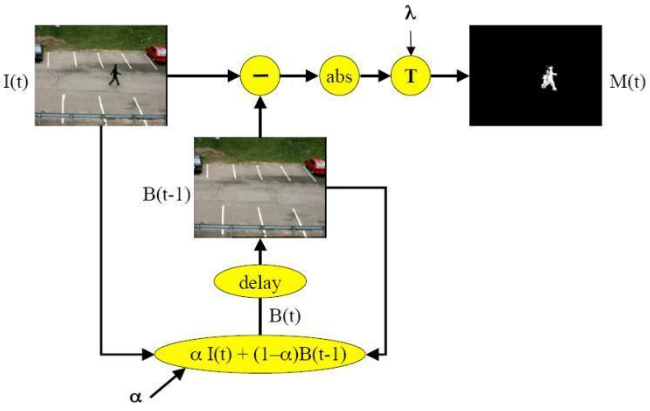


Fig. 2 Background Subtraction using adaptive background

C. Segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments . The aiml of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze in images.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. There are many different ways to perform image segmentation, including: Thresholding methods, such as Otsu‘s method ,Clustering methods, such as K-means and principle components analysis Transform methods, such as watershed , Texture methods, such as texture filters .

D. Simple and Dynamic Thresholding

After the background image B(x, y) is obtained, subtract the background Image B(x,y) from the current frame F (x, y). If the pixel difference is greater than the set threshold T , then it determines that the pixels appear in the moving object, otherwise, as the background pixels. The moving object can be detected after threshold operation. Its expression is as follows:

$$d(x, y, t) = \begin{cases} 1 \text{ if } |f(x, y, t) - B(x, y)| > Td \\ 0 \text{ otherwise} \end{cases}$$

where *d(x, y ,t)* is the binary image of differential results. *T* is gray-scale threshold; its size determines the accuracy of object identification. As in the algorithm *T* is a fixed value, only for an ideal situation, is not suitable for complex environment with lighting changes. In proposed dynamic threshold method, we dynamically change the threshold value according to the lighting changes of the two images obtained. This dynamic thresholding technique is clean, straightforward, easy to code, and produces the same output independently of how the image is processed. Instead of computing a running average of the last s pixels seen, we compute the average of an s x s window of pixels centered on each pixel. This is a better average for comparison since it considers neighboring pixels on all sides.

```
Procedure DynamicThreshold(in,out,w,h)
for i = 0 to w do
    sum = 0
    for j = 0 to h do
        sum= sum+in[i, j]
        if i = 0 then
            intImg[i, j]= sum
        else
            intImg[i, j] =intImg[i−1, j]+sum
        end if
    end for
end for
for i = 0 to w do
    for j = 0 to h do
        x1= i−s/2 {border checking is not shown}
        x2= i+s/2
        y1= j−s/2
        y2= j+s/2
        count= (x2−x1)×(y2−y1)
        sum=intImg[x2,y2]−intImg[x2,y1−1]−intImg[x1-1,y2] +intImg[x1−1,y1−1]
        if (in[i, j]×count) ≤ (sum×(100−t)/100) then
            out[i, j] =0
        else
            out[i, j]= 255
        end if
    end for
end for
```

Fig.3 Algorithm for dynamic thresholding

The average computation is accomplished in linear time by using the integral image. We calculate the integral image in the first pass through the input image. In a second pass, we compute the s x s average using the integral image for each pixel in constant time and then perform the comparison. If the value of the current pixel is t percent less than this average then it is set to black, otherwise it is set to white.The pseudocode demonstrates our technique for input image in, output binary image out, image width w and image height h.

III. SIMULATION RESULTS

This Visual Object Tracking system has been simulated using MATLAB 12.0 and both background subtraction using fixed and adaptive background were implemented on Spartan 3 XC3S200 FPGA board, with Microblaze processor using Xilinx EDK.

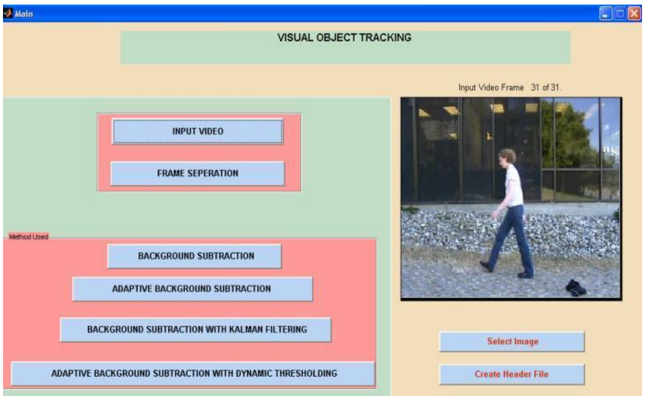

Main
VISUAL OBJECT TRACKING
Input Video Frame 31 of 31.
INPUT VIDEO
FRAME SEPERATION
Method Used
BACKGROUND SUBTRACTION
ADAPTIVE BACKGROUND SUBTRACTION
BACKGROUND SUBTRACTION WITH KALMAN FILTERING
ADAPTIVE BACKGROUND SUBTRACTION WITH DYNAMIC THRESHOLDING
Select Image
Create Header File

Fig. 4 Main GUI showing selected video frame

A. Fixed Background Model

This background subtraction model using fixed background has many advantages. It is easy to implement, uses low cost algorithm, gives extract shape of an object and has less noise sensitivity. There are certain disadvantages also. Objects that enter the scene and stop continue to be detected, making it difficult to detect new objects that pass in front of them. Also if the background moves both the object and its negative ghost are detected. This model is sensitive to changing illumination and unimportant movement of the background

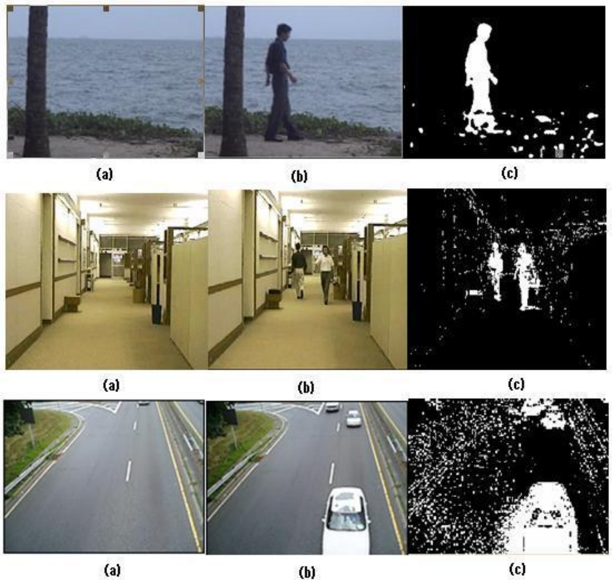


Fig.5 Matlab simulation results for fixed background a) current Frame b) background c) Background Subtracted image

B. Adaptive Background Model

In this method, the current image is “blended” into the background model with parameter α. α = 0 yields simple background subtraction, α = 1 yields frame differencing.

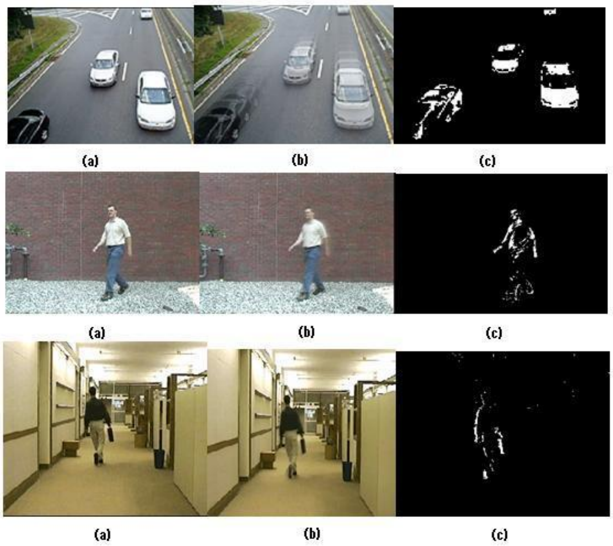


Fig. 6 Matlab simulation results for fixed background a) current Frame b) Adaptive background c) Adaptive Background Subtracted image.

Background models are not constant; they change over time in adaptive background subtraction. So it is more responsive to changes in illumination and camera motion and has less Noise sensitivity. Ghosts left behind by objects that start motion, gradually fade into the background. We used only one global threshold *Th*. *Th* is not a function of time t. Hence it will not give a good result if objects are moving fast or the frame rate is slow.

C. Dynamic Threshold or Adaptive Threshold

For different regions in the image different threshold is used. The initial threshold value is set by considering the mean or median value.

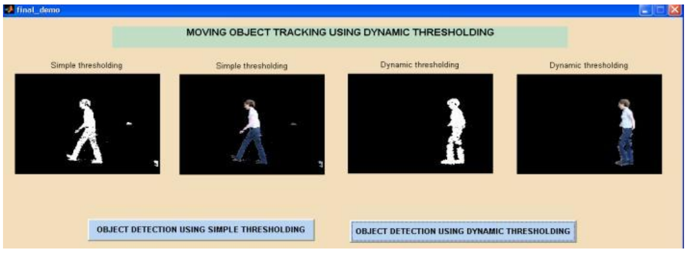
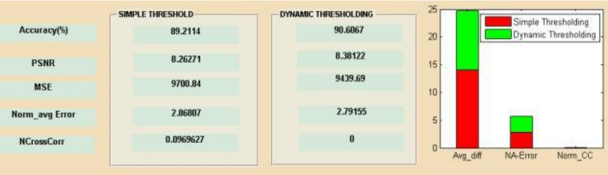

final_demo
MOVING OBJECT TRACKING USING DYNAMIC THRESHOLDING
Simple thresholding
Simple thresholding
Dynamic thresholding
Dynamic thresholding
OBJECT DETECTION USING SIMPLE THRESHOLDING
OBJECT DETECTION USING DYNAMIC THRESHOLDING

Fig.7 Matlab simulation results for simple and dynamic threshold shown in binary and color image formats.

Percentage of correct classification is used as the metric for comparison, and is defined as,

PCC=(TP+TN)/TPF

where TP is true positive that represents the number of correctly detected foreground pixels and TN is true negative representing the number of correctly detected background pixels.TPF represents the total number of pixels in the frame and are measured from a predefined ground truth.



	SIMPLE THRESHOLD	DYNAMIC THRESHOLDING
Accuracy(%)	89.2114	90.6067
PSNR	8.26271	8.38122
MSE	9700.84	9439.69
Norm_avg Error	2.86807	2.79155
NCrossCorr	0.0969627	0

Fig. 9 Matlab analysis results for simple and dynamic threshold for sample video SV-1

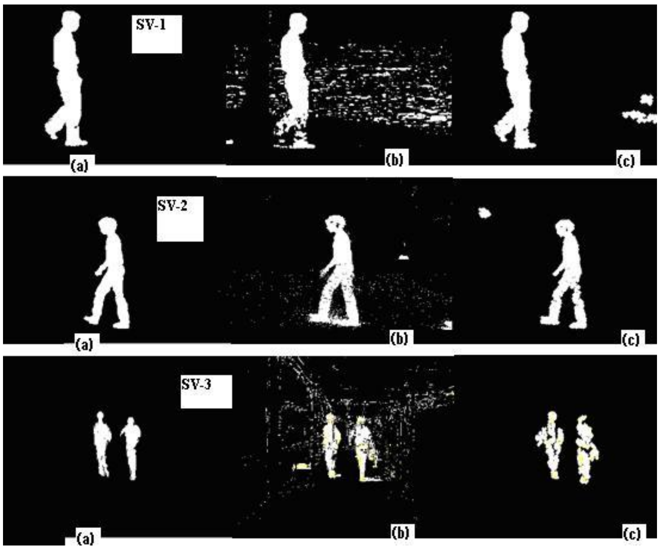


Fig.8 Matlab comparison results for three sample videos SV-1,SV-2,SV-3 showing a) Ground truth b) Simple threshold Image c) Dynamic Threshold image

IV. IMPLEMENTATION RESULTS

Both background subtraction model and adaptive background model were implemented on Xilinx Spartan- III XC3S200 FPGA, with Microblaze processor using Xilinx EDK.

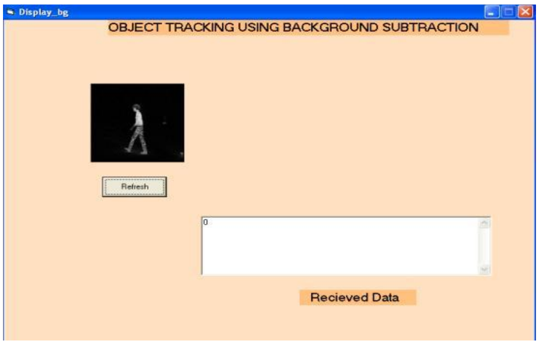


Fig.9 VB interface for receiving data from FPGA and the received data is displayed as image

The pixels generated through the FPGA are viewed using VB. The recieved pixel values are displayed in a seperate text box after converting to hex . The interface can be established by appropriately setting the baud rate and COM port.

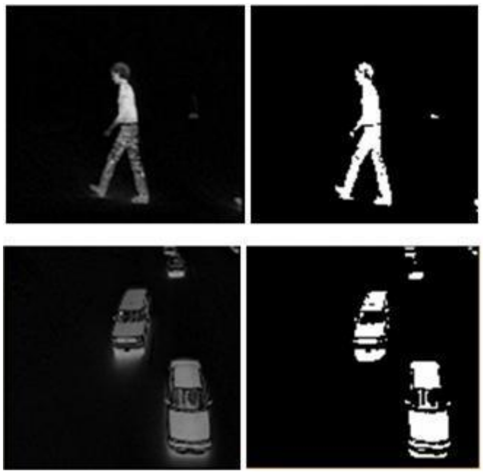


Fig.10 FPGA output shown using VB interface for background subtracted image and threshold applied image.

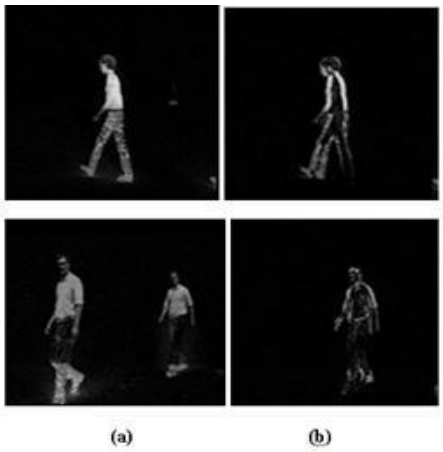


Fig.11 FPGA output shown using VB interface for a)background subtracted Image and b) adaptive background subtracted image

TABLE I
DEVICE UTILIZATION SUMMARY FOR SELECTED DEVICE : 3S200TQ144-4

Number of Slices:	1877	out of	1920	97%
Number of Slice Flip Flops:	2109	out of	3840	54%
Number of 4 input LUTs:	2991	out of	3840	77%
Number used as logic:	2438			
Number used as Shift registers:	297			
Number used as RAMs:	256			
Number of IOs:	62			
Number of bonded IOBs:	62	out of	97	63%
IOB Flip Flops:	64			
Number of BRAMs:	4	out of	12	33%
Number of MULT18X18s:	3	out of	12	25%
Number of GCLKs:	4	out of	8	50%
Number of DCMs:	1	out of	4	25%

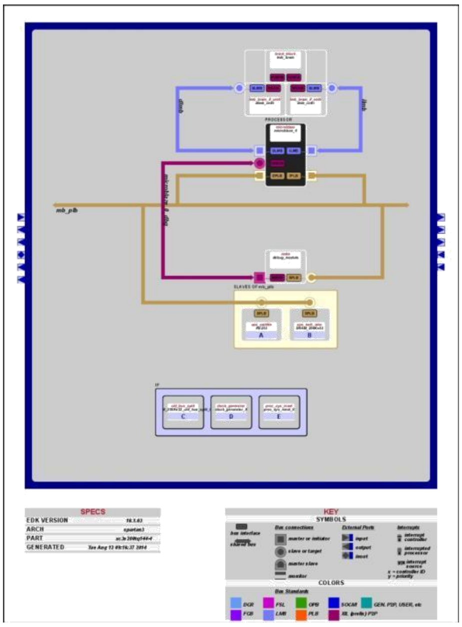


Fig.12 Synthesized block diagram using Xilinx EDK

V. CONCLUSIONS

The Visual Object Tracking system was simulated in Xilinx and MATLAB and implemented in Spartan 3 FPGA. For implementation, Impulse C language was used and was converted to VHDL with help of Xilinx EDK. As a modification to the existing system, I have used adaptive background subtraction along with adaptive threshold algorithm. With this method, higher noise immunity was achieved. In addition, most of the disadvantages of existing system were overcome.

As the image itself will occupy large memory we are sending the background image and current frame first to SRAM and is used in further processing steps. When we compare the outputs obtained from Matlab and FPGA, we find the outputs obtained using the Spartan 3 kit is computationally efficient. The pixel values are scaled and the outputs are comparable to the ones obtained using Matlab.

As for future enhancement, real time image capture and processing using Altera/higher boards and also storing of video in SRAM instead of image can make the full implementation on hardware possible with few extra processing steps on FPGA board.

ACKNOWLEDGMENT

I express my sincere thanks to Dr.Francis C Peter, Principal, Viswajyothi College of Engineering and Technology for his support and encouragement. I express my heartfelt gratitude to the Head of the Department of Electronics and Communication Engineering, Prof. Jose P Varghese for his support and guidance. I specially acknowledge my tutor Mrs. Rose Mary Kuruvithadam, Asst. Professor of ECE Department and my Guide, Mrs. Ranjini Surendran**,** Asst. Professor of ECE Dept. for their guidance and support through the entire project.

And finally, I wish to express my special regards to my parents and friends for their moral support and cooperation, without whom, I could never have completed this venture.

REFERENCES

[1] Background Subtraction Algorithm for Moving Object Detection on FPGA, C.S´anchezFerreira ,J.Y.Mori, C.H.Llanos,Department of Mechanical Engineering, University of Brasilia Programmable Logic (SPL), 2012 VIII Southern Conference on 20-23 March 2012.

[2] A Directional-Edge-Based Real-Time Object Tracking System Employing Multiple Candidate-Location Generation IEEE Transactions On Circuits And Systems For Video Technology, Vol. 23, No. 3, March 2013

[3] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, Dec. 2006.

[4] Intensity Range Based Background Subtraction for Effective Object Detection Kalyan Kumar Hati, Pankaj Kumar Sa, and Banshidhar Majhi IEEE Signal Processing Letters, Vol. 20, No. 8, August 2013

[5] C.Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Realtime tracking of the human body," *IEEE Trans. Patt. Anal. Mach. Intell.*,vol. 19, no. 7, pp. 780–785, Jul. 1997.

[6] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive object tracking based on an effective appearance filter," *IEEE Trans. Patt. Anal. Mach.Intell., vol. 29, no. 9, pp. 1661–1667, Sep. 2007.*

[7] Centroid weighted Kalman filter for visual object tracking Zhaoxia Fu , Yan Han

[8] Adaptive Thresholding Using the Integral Image,Derek Bradley_Carleton University, Canada and Gerhard Roth, National Research Council of Canada

[9] B. Han, Y. Zhu, D. Comaniciu, and L. Davis, "Visual tracking by continuous density propagation in sequential Bayesian filtering framework," *IEEE Trans. Patt. Anal. Mach. Intell., vol. 31, no. 5, pp. 919–930, May* 2009.

[10] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans.Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[11] W. Kim and C. Kim, "Background subtraction for dynamic texture scenes using fuzzy color histograms," *IEEE Signal Process. Lett.*, vol.19, no. 3, pp. 127–130, Mar. 2012.

[12] Y.-J. Yeh and C.-T. Hsu, "Online selection of tracking features using AdaBoost," *IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 3*, pp. 442–446, Mar. 2009.

[13] Q. Chen, Q.-S. Sun, P. A. Heng, and D.-S. Xia, "Two-stage object tracking method based on kernel and active contour," *IEEE Trans.Circuits Syst. Video Technol., vol. 20, no. 4, pp. 605–609, Apr.*2010.

[14] R1. Joshi, Kinjal A., and Darshak G. Thakore, 2012. "A Survey on Moving Object Detection and Tracking in Video Surveillance System, "International Journal of Soft Computing and Engineering (IJSCE) ISSN", 2231-2307.

[15] Wallflower: Principles and Practice of Background Maintenance ,Kentaro Toyama, John Krumm, Barry Brumitt, Brian Meyers ,Microsoft Research Redmond, WA *International Conference on Computer Vision, September 1999, Corfu, Greece*

[16] Dheeraj Agrawal, Nitin Meena, 2013. "Performance Comparison of Moving Object Detection Techniques in Video Surveillance System" The International Journal of Engineering And Science (IJES), Volume. 2, Issue. 01, 240-242.

[17] Visual Object Tracking Based on Local Steering Kernels and Color Histograms IEEE transactions on circuits and systems for video technology, vol. 23, no. 5, may 2013

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

[19] Christopher T. Johnston, Kim T Gribbon, Donald G. Bailey (2005)'FPGA based Remote Object Tracking for Real-time Control',

1st International Conference on Sensing Technology November 21-23, 2005 Palmerston North, New Zealand.

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

[21] Hong C.S,. Chung S.M, Lee J.S. and Hong K.S. (1997), 'A Vision-Guided Object Tracking and Prediction Algorithm for Soccer Robots',IEEE Robotics and Automation Society, Vol. 1 pp: 346-351.

[22] L. Baumela and D. Maravall, "Real-time target tracking," Aerospace and Electronic Systems Magazine, IEEE, vol. 10, no. 7, pp. 4-7, 1995.

[23] L. Kilmartin, M. O Conghaile (1999), 'Real Time Image Processing Object Detection and Tracking Algorithms', Proceedings of the IrishSignals and Systems Conference, NUI, Galway, June 1999, pp. 207-214.

[24] B.P.L. Lo and S.A. Velastin, "Automatic congestion detection system for underground platforms," *Proc. of 2001 Int. Symp. on Intell. Multimedia*, Video and Speech Processing, pp. 158-161, 2000.

[25] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shadows in video streams", *IEEE Trans. on Patt. Anal. and Machine Intell.*, vol. 25, no. 10, Oct. 2003, pp. 1337-1342.

[26] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder:Real-time Tracking of the Human Body," IEEE Trans. on Patt. Anal. and Machine Intell., vol. 19, no. 7, pp. 780-785, 1997.

[27] C. Stauffer, W.E.L. Grimson, "Adaptive background mixture modelsfor real-time tracking", *Proc. of CVPR 1999*, pp. 246-252.

[28] A. Elgammal, D. Harwood, and L.S. Davis, "Non-parametric Model for Background Subtraction", *Proc. of ICCV '99 FRAME-RATE Workshop*, 1999.

[29] J. Mike McHugh, Janusz Konrad, Venkatesh Saligrama, and Pierre-Marc Jodoin, "Foreground-Adaptive Background Subtraction", *IEEE Signal Process. Lett*, 2009 IEEE.

[30] P.-M. Jodoin, M. Mignotte, and J. Konrad, "Statistical background subtraction using spatial cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, pp. 1758–1763, Dec. 2007.

[31] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density for visual surveillance", *Proc. IEEE*, vol. 90, pp. 1151.1163, 2002.

[32] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 11, pp. 1778.1792, 2005.

[33] M. Seki, T. Wada, H. Fujiwara, K. Sumi, "Background detection based on the cooccurrence of image variations", *Proc. of CVPR 2003*, vol. 2, pp. 65-72.

[34] T. Aach and A. Kaup, "Bayesian algorithms for adaptive change detection in image sequences using Markov random fields", *Signal Process., Image Commun*., vol. 7, pp. 147.160, 1995.