Foreground detection of video through the integration of novel multiple detection algorithms

A Thesis submitted for the Degree of
Doctor of Philosophy

By
Muhammad Nawaz

Brunel
UNIVERSITY
LONDON

Supervised By: **Professor John Cosmas**

Department Electronic and Computer Engineering
School of Engineering and Design
Brunel University

September 2013

ABSTRACT

The main outcomes of this research are the design of a foreground detection algorithm, which is more accurate and less time consuming than existing algorithms. By the term accuracy we mean an exact mask (which satisfies the respective ground truth value) of the foreground object(s). Motion detection being the prior component of foreground detection process can be achieved via pixel based and block based methods, both of which have their own merits and disadvantages.

Pixel based methods are efficient in terms of accuracy but a time consuming process, so cannot be recommended for real time applications. On the other hand block based motion estimation has relatively less accuracy but consumes less time and is thus ideal for real-time applications. In the first proposed algorithm, block based motion estimation technique is opted for timely execution. To overcome the issue of accuracy another morphological based technique was adopted called opening-and-closing by reconstruction, which is a pixel based operation so produces higher accuracy and requires lesser time in execution. Morphological operation opening-and-closing by reconstruction finds the maxima and minima inside the foreground object(s). Thus this novel simultaneous process compensates for the lower accuracy of block based motion estimation.

To verify the efficiency of this algorithm a complex video consisting of multiple colours, and fast and slow motions at various places was selected. Based on 11 different performance measures the proposed algorithm achieved an average accuracy of more than 24.73% than four of the well-established algorithms.

Background subtraction, being the most cited algorithm for foreground detection, encounters the major problem of proper threshold value at run time.

For effective value of the threshold at run time in background subtraction algorithm, the primary component of the foreground detection process, motion is used, in this next proposed algorithm. For the said purpose the smooth histogram peaks and valley of the motion were analyzed, which reflects the high and slow motion areas of the moving object(s) in the given frame and generates the threshold value at run time by exploiting the values of peaks and valley.

This proposed algorithm was tested using four recommended video sequences including indoor and outdoor shoots, and were compared with five high ranked algorithms.

Based on the values of standard performance measures, the proposed algorithm achieved an average of more than 12.30% higher accuracy results.

ACKNOWLEDGMENTS

I am very thankful to my supervisor, Professor John Cosmas, who did more than just supervise my research. He gave me guidance and extended his support at times I most needed it on both personal and professional level. It was his extraordinary support that enabled me complete my research. He helped me, like a friend, when I was at my lowest point due to a serious illness during my PhD.

I am grateful to my colleagues and friends who supported me in more ways than I can mention during my Ph.D research. I would especially like to thank my colleagues at the Centre for Multimedia and Communication Research (CMCR). I would also like to thank the Research Office staff at the School of Engineering and Design, Brunel University. I am grateful to the Higher Education Commission (http://hec.gov.pk/) of Pakistan and the Institute of Management Sciences (http://imsciences.edu.pk/), Peshawar for funding my Ph.D.

I am also thankful to Thierry Bouwmans, Associate Professor at MIA Lab-University La Rochelle, France; Benalia Mohcene, University of Djelfa, (Djelfa) Algeria; Soo Wan Kim, Seoul National University Korea; Pavlos Lazaridis; Zaharias D. Zaharis; Yue Zhang; Mr. Awais Adnan, Assistant Professor (Image Processing) IMSciences Peshawar, Pakistan; Obaid Fatah; Karim Ullah and Hamdullah Mohib. They helped me in conducting this research and support over the last four years in terms of technical assistance, proposed-algorithms re-verification and results verifications.

Lastly, I owe my deepest gratitude to my parents, brothers, sisters, in-laws, wife, and kids without whom I would not have made it through my PhD studies. Thank you all.

Dedicated to my mother and father

Table of Contents

List of Figures

List of Tables

LIST OF ABBREVIATION

Adaptive Rood Pattern Search	ARPS
Balance Classification Rate	BCR
Diamond search	DS
False Positive	*fp*
False Negative	*fn*
Frame per second	fps
Large Diamond Search Pattern	LDSP
Mixture of Gaussian	MoG
Morphological operations	MOs
Motion vector	MV
Positive Predictive Value	PPV
Red Green Blue	RGB
Region of Interest	ROI
Region of support	ROS
Small Diamond Search Pattern	SDSP
Three-step Search Algorithm	TSS
True Negative	*tn*
True Positive	*tp*
True Positive Rate	TPR
Yule Coefficient	YC

AUTHOR'S DECLARATION

The work described in this thesis has not been previously submitted for a degree in this or any other university and unless otherwise referenced it is the author's own work.

STATEMENT OF COPYRIGHT

Chapter 1
INTRODUCTION

1.1 OVERVIEW

This chapter provides a background to foreground detection, the motivation, aims and objectives of this research. Furthermore, the core research contributions and methodology are briefly described. Finally, the thesis structure is outlined.

1.2 BACKGROUND

Temporal differencing, optical flow and background subtraction are the three most common representative approaches used for motion segmentation.

Temporal differencing exploits the frame difference between consecutive frames in a video to detect moving regions; the method is highly adaptive to changing environments but in most cases does not perform well while capturing shapes of certain kinds of moving objects. Background Subtraction is a most widely used technique for detecting moving objects in video; the method compares the current image with a model of the background image as the reference to detect motion. Optical flow is an estimation of the local image motion. These methods along with their varying implementations are studied and explained with regards to methodology and accomplishment.

Temporal differencing was one of the approaches introduced earlier, to aid the task of motion estimation. However it generally fails to extract complete shapes of certain kinds and thus introduces the foreground aperture problem. Algorithms based on this technique require the support of additional methods to detect stationary objects. The strengths of this method are in its adaptation to dynamic changes in the background of images sequences [1]. Oral and Deniz (2007) highlighted that the weaknesses within this method lay in its failure to extract all relevant pixels whilst performing foreground segmentation [1]. This would cause the development of holes in detected foreground objects as a result. The temporal differencing method was centered on the comparison of video frames. The video frames compared are required to be separated by a constant time δt to calculate the absolute difference [2]. This reveals the changed regions in the image sequence. A limitation within this method is the unmanageability to track any objects in the event of substantial camera movement. Chang *et al* (2005) proposed a method based on temporal difference solely for the task of "change detection" [3]. The task operated within various systems including image processing and visual surveillance. They regard change detection to be extremely critical in the video surveillance field, as it possesses the means to perform foreground-background segmentation on moving objects [3].

The optical flow algorithms are primarily based on motion vectors and use the spatiotemporal derivatives of pixel values or block matching techniques [4]. This procedure is capable of detecting the person in a changing background. Thus, the method has the capability of extracting the foreground from complex outdoor scenes that contain non-stationary vegetation [5]. Since this method does not use background subtraction, it produces good results in cases where the background image is not available [5-6]. Obviously, in these cases the traditional background subtraction methods fail. The Optical flow models are based on a two-frame differential method for motion estimation. This method estimates the motion between two frames, which are taken at a time interval *t*. Optical flow methods are very useful in pattern recognition, computer vision and other image processing applications. All of the above methods consider only the changing parts of an image as the foreground. This is not always a true assumption and can further cause two types of problems respectively called *foreground aperture* and *false foreground detection*. The case of false foreground detection occurs when there are light variations in the background or very small waving of tree branches. Within optical flow there are five common methods. These are: gradient-based, block-matching-based, energy-based, phased-based and neuro-dynamic type algorithms. The limitations, that each of these methods suffers from, is the inability to avoid the negative effect of lighting on the output, which results in background noise [7]. Ridder *et al*'s (1995) investigations of the optical flow approach highlighted limitations with the method [8]. To successfully detect a foreground object it had to be continuously moving, if not it would be mistaken for the background. Also, the implementation of the method was computationally expensive and required specific hardware to deploy the method on real-time applications [9]. Kim *et al* (2012) investigated the calculation of optical flow in pixels to assist them in developing a method to detect moving objects with a moving camera using non-panoramic backgrounds [10]. They learned that the optical flow method detected changes within adjacent frames. However, like Ridder *et al* (1995) they too found shortfalls associated with the method [8]. These shortfalls were the high level of computation required and the restriction of camera movement [9].

The *background subtraction* or *frame differencing algorithms* provide one of the most convenient ways for foreground detection [11-13] due to their simple implementation and processing. The main drawback of *background subtraction* or *frame differencing algorithms* is that in real situations it is not always possible to have foreground-free images. Also, the algorithms encounter problems in several cases of background variation, e.g., cases of camera motion, backgrounds that contain shadows, wavering of plant branches or illumination changes. To overcome these problems, the Gaussian function can be applied for optimum results. There have been various proposals made for background models, however, an ideal example should satisfy the following conditions: resistance to noise, resistance to changes in illumination of the scene, flexibility to changes in the original background, resistance to any movement, production of no distortions on foreground, account as little as possible for post-processing operations, efficiency and consistency and ease of implementation [1]. From the many background model proposals made by different researchers, no model fulfills every condition stated above thus far. However, due to the popularity and relative simplicity of the background subtraction method, authors are continuously developing new ways to improve it. The background subtraction method has limitations that the early work of Grimson *et al* (1999) attempted to solve [14]. The disadvantages were related to the constant updating of the background model for each frame. Another of the common problems known to the background subtraction method was its sensitivity to active scenes; this included changes to lighting, any background movement and possible shadows [15]. In an attempt to address these problems, Grimson *et al* (1999) proposed an adaptive multi-coloured background model for real-time tracking [14]. However, they later found and highlighted the problems with the method to be slow to learn at the beginning of usage particularly in busy environments and its inability to distinguish between moving shadows and objects [15].

1.3 MOTIVATION

Foreground detection is basically the detection of motion of an object or event in a video sequence. Motion detection plays a very important role in any object(s) tracking or video surveillance algorithm.

The foreground is the more visible and prominent part of the scene in a picture or video. In contrast to the background, the foreground can be defined as that part of the scene in an image, which consists of bits closer to the viewer, while the background refers to the bits at the back or further away from the viewer. The foreground may refer to an image object relatively closer to the camera [16]. Foreground detection in a video is the identification of the Region of Interest (ROI), or the identification of the moving objects (foreground) and the static parts (background). Due to its motion, a human is considered as a foreground by the surveillance systems. Therefore, the challenge in detecting a foreground is to fully cover the shape of the moving object in various motion styles, e.g., walking, sitting or jumping [17].

The foreground detection is the prerequisite step for many video analysis systems such as intelligent video surveillance or vehicular traffic analysis, human detection and tracking, or gesture recognition in human-machine interface and video compression. So far, different algorithms have been proposed but none of them can be considered as a comprehensive solution for different situations and application scenarios. Furthermore, the level of complexity in the foreground detection may depend on the level of complexity of the videos under observation [18].

Based on the algorithm, foreground detection process can be divided into many steps, motion detection, object classification (large or small), tracking, activity understanding (setting, jumping, binding or walking), and semantic description, and every step exhibits its own hurdles and challenges for complete system design. The prior activity is the precise motion detection of an object, without which, remaining activities cannot be achieved perfectly [18].

The benefit of automated foreground detection system is thus to increase the accuracy and reduce the execution time to make it more useful for applications such as surveillance systems increase automated object detection for safety and crime control such as in transportation, industrial applications, indoor and outdoor enjoyment and military applications [18].

1.3.1 EFFICIENT-FOREGROUND-DETECTION ALGORITHM

Building an ideal foreground detection algorithm is not an easy task, since every algorithm is built for handling a specific issue and at the same time no algorithm guaranties 100 percent accuracy and ideal computational time as they have inversely proportional relationship. However research is going on to attain a perfect solution for foreground detection.

The following situations in the analysis of videos should be considered in order to construct an efficient algorithm for foreground detection. Since the videos can be of different nature due to the application scenarios involved, the designed algorithm should be efficient enough to capture the following details accordingly:

- In talk shows the background is usually static most of the time, while the foreground consists of moving objects.
- There can be situations where the background and foreground are both moving at the same time. For example, in the mobile video sequence, it is possible that the objects in the background are moving in the same direction as the ones in the foreground. Similarly, the camera movement might be involved during the capture of the video sequence.
- There might be a situation where the ROI in the video under analysis is static while the camera, which is capturing the video, is moving. This can happen in the situations of aerial surveys.
- Foreground detection in rush hours or traffic jams is another situation where the video is analysed for foreground detection, keeping in view the vehicles’ as well as the pedestrian’s relative movement.
- Light variation should be considered by the foreground detection algorithm. This is because, the video quality changes with changes in the light intensity. For example, in cases of cloudy weather the video frames might need a different approach to extract the desired foreground compared to frames in cases of sunny weather. A similar situation can

arise when a video is captured in cases where the objects are moving from a dark or semi-dark environment to high intensity light locations.

Colour and texture changes are other important features, which should be taken into account. Since, there are multiple objects in the same frame that might be of the same colour or texture; these objects need to be individually identified by the foreground detection algorithm.

1.4 AIMS AND OBJECTIVES

Computer vision is a technique used on images and video to detect, classify and track object or event in order to "understand" a real-world scene.

Foreground detection is one of the components of computer vision, which detect any object or event in motion in given a frame.

Various image processing operations are applied on image or given frame such noise removal and contrast adjustment are applied as preliminary operations. Which computer vision techniques are utilised to detect, identify, classify, recognise and track any moving object or event. The next step is of interpretation whether the moving object or event is pedestrian, bicyclist, truck, car, traffic violation or an accident.

Computer vision have so many applications in robotic vision, identification of moving object in remote area, surveillance system, thumb recognition and vehicles number plates recognition.

The aim of the research presented in this thesis is to design and implement a foreground detection algorithm, which is more accurate and less time consuming. By the term accuracy we mean to extract the exact mask (which satisfies the respective ground truth value) of the foreground object(s). The research aims and objectives are summarized as follows:

- The most important goal of the research is to devise a foreground detection algorithm which detects the *full mask* of foreground object(s).
- The proposed algorithm should be capable to detect foreground object without using *reference image* (free background image). Reference image is the prerequisite requirement of background subtraction method.
- *Less time execution* is another main objective of the proposed foreground detection algorithm. This is very important especially for real time applications such as surveillance systems.

- Being the most important component of any foreground detection, *precise motion detection* is also the aim of the research.
- To overcome one of the major limitations of most cited and widely used background subtraction algorithm [19], which is to automatically calculate the value of *threshold* at run time for precise foreground detection.
- Finally the *validation of results* using standard performance measures.

1.5 CONTRIBUTION TO KNOWLEDGE

This thesis contributes to knowledge in the research area of precise foreground detection followed by obtained results validation by standard performance measures.

Based on aforesaid discussion in the topic (1.4) under heading aims and objectives, accurate motion detection is the prior goal of any foreground detection algorithm for the successful achievements.

Precise motion detection followed by full mask of the moving object extraction. Where motion estimation and morphological operation opening-and-closing-by-reconstruction approach is utilized in order to achieve the prime goal. Operation opening-and-closing-by-reconstruction approach is pixel based which also increases the accuracy of segmentation process [17] and is not possible in block based motion segmentation. Opening-and-closing-by-reconstruction identifies the minima and maxima inside the foreground object which leads to the further enhancement of foreground detection result and plays a very important role in obtaining of full foreground object mask [17]. In order to find accurate areas of the motion the proposed algorithms utilized block matching algorithm for motion estimation, which is less time consuming as compare to pixel based motion estimation process. As our proposed motion estimation approach is block based, thus it requires less time in execution. For the omission of miss and over calculated motion areas certain morphological operations are used in a particular fashion to overcome this issue. In the proposed algorithm there is no need of reference image in advance [17].

Most of the researchers proved their results' accuracy using few performance measures such precision, recall and F-measure. However the proposed algorithms are tested and verified by 11 different performance measures to prove its credibility scientifically. Performance measures compare algorithm generated results with ground truth of the respective frames. Ground truth is prepared via frame by frame human segmented results based on motion. Ground truth is the ideal (intended) results to be obtained by proposed algorithm, which is not possible to obtain 100% accuracy, in all aspects. In other words to satisfy all performance measures.

It is clearly shown from the results obtained from various performance measures that the proposed algorithm performs much better than three well-established algorithms, on average of more than 24.74% accuracy.

The next objective is based on the challenging issue in the most popular and cited algorithm background subtraction which requires auto threshold value mechanism. In the proposed algorithm the auto threshold value is achieved by motion histogram approach.

In the background subtraction method one of the core issues is how to setup the threshold value precisely at run time, which can ultimately overcome several limitations of this approach in the foreground detection.

After studying the most cited algorithms on background subtraction [1-3] from very simple to complex, none were found to segment with 100 percent accuracy in all aspects for real time implementation due to numerous complications in real world situation.

To avoid the need of previous background learning, a robust pixel foreground classification is introduced, [19] which is a well cited algorithm. This algorithm claims that robust pixel foreground classification is possible without the need of previous background learning. For proper pixel classification joint background subtraction and frame-by-frame differencing method is used and background model is selectively updated according to above classification by [20], based on equation $B_t = (1-\alpha)B_{t-1} + \alpha F_t$, where B_t is current background at time t , B_{t-1} is previous background at time *t-1*, F_t is current foreground at time t and α dependents on pixel classification.

The satisfactory performance of this algorithm has been confirmed in [21]. However, this is at the cost of two disadvantages: This method totally fails when the foreground object stops or if its speed is low, and the second difficulty is that of setting the proper differencing threshold value [21-22]. A proper differencing threshold value affects quite critically the performance of successive steps of the algorithm. Manual setting of this value when an automated solution is possible for systems is not a good choice.

This problem of threshold value can be solved using: Clustering, entropy and object attribute based methods. Other two methods are spatial and local methods. These methods are mathematically complex and time consuming [23].

So in the proposed algorithm the key feature of any foreground detection algorithm; motion is used however obtaining the threshold value from the original motion histogram is not possible due to the large number of peaks and valleys, so for the said purpose smooth motion histogram is used in a systematic way to obtain the threshold value in way to cover both the lower and higher motion areas in the respective frame.

In the proposed algorithm the main focus is to get a better estimation of threshold so that a dynamic value is obtained from the histogram at run time, frame by frame.

In the experiments so far, this proposed algorithm did not encounter any *ghosting* and *foreground aperture* problem in the videos ranging from slow to normal and fast, plus indoor and outdoor videos.

It is very important to note that, in the proposed algorithm, opening-and-closing by reconstruction technique (already explained in chapter 4, of the thesis) is not applied to cover the precise mask of the foreground object and eradicate and cover the miscalculated background pixels and foreground area. The goal of this algorithm is to highlight the strength and weakness of the proposed algorithm only.

It is obvious that the proposed algorithm clearly outperforms the other five methods. Histogram approach [24] is the second best approach. Mahalanobis [25] distance approach was found to be the third best method, while, the Euclidean Distance algorithm[25] was found to be the fourth best algorithm, based on average of performance measurement results. Overall, the performance of the Local-self similarity [26] and GGM Zivkovic [27] techniques were found to be non-satisfactory.

It is clearly shown from the results obtained that the proposed algorithm performs much better than the second best algorithm Histogram, on average by 12.30%.

1.6 RESEARCH METHODOLOGY

The research methodology followed for the proposed research is as follows:

- In the beginning phase, an extensive literature review was conducted of present and past work done in the area of foreground detection. Research papers in the form of authentic conference and journal papers were studied in depth to understand the work done so far in the field of foreground detection to find research gaps and milestones of the research. Papers were mainly from IEEE, ACM and Springer journals.
- Various classical books on image and video processing [28-30] were studied for understanding of the well-known theoretical and practical approaches towards foreground detection and associated topics such as motion estimation and the role of morphological operations in noise reduction.
- For simulation purposes, MATLAB was used to test and verify the output of the proposed logic(s).
- In order to quantify the obtained results performance measures were carried out for various segmentation algorithms. These performance measures indicate the strength and weakness of the proposed algorithm over the other researchers' work.
- Publishing in IEEE international conferences and transactions was another platform where the proposed work was reviewed by various experts in the field. Expert feedback in the form of review is another healthy approach towards the advancement of knowledge, and a source of further improvement in the proposed work.
- Knowledge exchange with experts in the field was another experience for the source of knowledge gaining which included face to face meetings and exploiting the use of electronic means of communication.

1.7 RESEARCH OUTPUT

During this research the proposed algorithms were published and presented in top ranked International IEEE conferences, and Transactions. The list of published and tentative publications is given below:

- Nawaz, M., Cosmas, J., Adnan, A. and Ali, M. (2011) "Inter-intra frame segmentation using colour and motion for region of interest coding of video", *Broadband Multimedia Systems and Broadcasting (BMSB), 2011 IEEE International Symposium*, pp.1-4

- Nawaz, M., Fatah, O.A., Comas, J. and Aggoun, A. (2012) "Extracting foreground in video sequence using segmentation based on motion, contrast and luminance", *Broadband Multimedia Systems and Broadcasting (BMSB), 2012 IEEE International Symposium*, pp.1-3

- Fatah, O.A.; Aggoun, A.; Nawaz, M.; Cosmas, J.; Tsekleves, E.; Swash, M.R.; Alazawi, E.; , "Depth mapping of integral images using a hybrid disparity analysis algorithm," Broadband Multimedia Systems and Broadcasting (BMSB), 2012 IEEE International Symposium, pp.1-4, 27-29 June 2012

- Nawaz, M., Cosmas, J., and Adnan, A. "Foreground detection using background subtraction with histogram", IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Brunel University, London June 4th – 7th 2013.

- Nawaz, M., Cosmas,J., Lazaridis,P., Zaharis, Z, D., Mohib, H. and Y.Z., "Precise-Foreground-Detection Algorithm using Motion Estimation, Minima and Maxima inside the Foreground Object" *IEEE Transactions on Broadcasting.* Accepted on 17 September 2013 for publication.

- Nawaz, M., Cosmas, J., Lazaridis, P., Zaharis , Z, D. and Y.Z., "A novel idea of foreground detection for moving camera", *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society.* (My next publication)

1.8 THESIS STRUCTURE

This thesis consists of six chapters, initially with this introductory chapter, which provide a concise synopsis of the thesis.

Chapter-1 briefly discusses the motivation, aims and objectives, and research methodology.

Chapter-2 is about the standard and classical foreground detection algorithms discovered in the literature review to understand the present and past contributions to the field. However in the core chapters, the relevant literature reviews are also given in concise form to support the topic under discussion for understanding purpose.

Chapter-3 discusses the fundamentals of segmentation performance measures to quantify the strength and weakness of any algorithm in different ways.

Chapter-4 reports the first contribution to knowledge which is how to precisely find the foreground area in a video sequence. This chapter discusses motion estimation, noise removal process, the role of morphological operations opening-and-closing-by-reconstruction to find the minima and maxima of the foreground object and finally the technical conclusion based on various performance measures to show the strength and weakness of the proposed algorithm in comparison to the state-of–the-art algorithms such as Gaussian Mixture [31], optical flow [32], SGM-R [33] and Soo Wan Kim algorithms [34].

Chapter-5 mainly focuses on automatic threshold value for the state-of-the-art, most cited and widely used algorithm, namely: the background subtraction method. For the solution to this problem, the effective motion histogram based solution is proposed, which dynamically (at run time) calculates the value of threshold and thus enhances the accuracy and performance of background subtraction method. The value of threshold can identify the slow and fast motion areas of the respective frame. Finally the evaluation results, which are based on various performance measures, are presented to prove the effectiveness of the proposed algorithm over the other approaches.

Chapter-6 summarizes the overall findings of the entire thesis and proposes future work which may be carried out in connection with research presented in this thesis.

1.9 REFERENCES

1. Oral, M. and Deniz, U. (2007) "Centre of mass model–A novel approach to background modelling for segmentation of moving objects", *Image and Vision Computing*, vol. 25, no. 8, pp. 1365-1376.

2. Lipton, A.J., Fujiyoshi, H. and Patil, R.S. (1998) "Moving target classification and tracking from real-time video", *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop*, pp. 8.

3. Chang, C., Chia, T. and Yang, C. (2005) "Modified temporal difference method for change detection", *Optical engineering,* vol. 44, no. 2, pp. 027001-027010.

4. Horn, B.K. and Schunck, B.G. (1981) "Determining optical flow", *Artificial Intelligence,* vol. 17, no. 1, pp. 185-203.

5. Iketani, A., Nagai, A., Kuno, Y. and Shirai, Y. (1998) "Detecting persons on changing background", *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference*, pp. 74.

6. Wixson, L. (2000) "Detecting salient motion by accumulating directionally-consistent flow", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, no. 8, pp. 774-780.

7. Li, W., Wu, X., Matsumoto, K. and Zhao, H. (2010) "Foreground detection based on optical flow and background subtract", *Communications, Circuits and Systems (ICCCAS), 2010 International Conference*, pp. 359.

8. Ridder, C., Munkelt, O. and Kirchner, H. (1995) "Adaptive background estimation and foreground detection using kalman-filtering", *Proceedings of International Conference on recent Advances in Mechatronics* Citeseer, pp. 193.

9. Di, M., Joo, E.M. and Beng, L.H. (2008) "A comprehensive study of kalman filter and extended kalman filter for target tracking in wireless sensor networks", *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference*, pp. 2792.

10. Kim, S.W., Yun, K., Yi, K.M., Kim, S.J. and Choi, J.Y. (2012) "Detection of moving objects with a moving camera using non-panoramic background model", *Machine Vision and Applications,* pp. 1-14.

11. Cheng, F., Huang, S. and Ruan, S. (2011) "Illumination-sensitive background modeling approach for accurate moving object detection", *Broadcasting, IEEE Transactions on,* vol. 57, no. 4, pp. 794-801.

12. Haritaoglu, I., Harwood, D. and Davis, L.S. (2000) "W< sup> 4</sup>: real-time surveillance of people and their activities", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, no. 8, pp. 809-830.

13. Zhao, T. and Nevatia, R. (2004) "Tracking multiple humans in complex situations", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, no. 9, pp. 1208-1221.

14. Stauffer, C. and Grimson, W.E.L. (1999) "Adaptive background mixture models for real-time tracking", *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference*.

15. Maddalena, L. and Petrosino, A. (2008) "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications", *Image Processing, IEEE Transactions on,* vol. 17, no. 7, pp. 1168-1177.

16. Maddalena, L. and Petrosino, A. (2008) "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications", *Image Processing, IEEE Transactions on,* vol. 17, no. 7, pp. 1168-1177.

17. Nawaz, M., Cosmas, J., Adnan, A. and Ali, M. (2011) "Inter-intra frame segmentation using colour and motion for region of interest coding of video", *Broadband Multimedia Systems and Broadcasting (BMSB), 2011 IEEE International Symposium on* IEEE , pp. 1.

18. Pal, S.K., Petrosino, A. and Maddalena, L. (2012) *Handbook on soft computing for video surveillance,* CRC Press.

19. Migliore, D.A., Matteucci, M. and Naccari, M. (2006) "A revaluation of frame difference in fast and robust motion detection", *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks* ACM, pp. 215.

20. Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P. (1997) "Pfinder: Real-time tracking of the human body", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, no. 7, pp. 780-785.

21. Martínez-Martín, E. and del Pobil, A.P. (2012) *Robust motion detection in real-life scenarios,* Springer.

22. Sun, S., Wang, Y.F., Huang, F. and Liao, H.M. (2012) "Moving foreground object detection via robust SIFT trajectories", *Journal of Visual Communication and Image Representation,* vol. 24, issue 3, pp. 232-243.

23. Sezgin, M. (2004) "Survey over image thresholding techniques and quantitative performance evaluation", *Journal of Electronic imaging,* vol. 13, no. 1, pp. 146-168.

24. Zivkovic, Z. (2004) "Improved adaptive Gaussian mixture model for background subtraction", *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* IEEE, pp. 28.

25. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. and Rosenberger, C. (2010) "Comparative study of background subtraction algorithms", *Journal of Electronic Imaging,* vol. 19, no. 3, pp. 033003-033012.

26. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. and Rosenberger, C. (2010) "Comparative study of background subtraction algorithms", *Journal of Electronic Imaging,* vol. 19, no. 3, pp. 033003-033012.

27. Zivkovic, Z. (2004) "Improved adaptive Gaussian mixture model for background subtraction", *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, pp. 28.

28. Gonzalez, R.C., Woods, R.E. and Eddins, S.L. (2009) *Digital image processing using MATLAB,* Gatesmark Publishing Knoxville.

29. Martínez-Martín, E. and del Pobil, A.P. (2012) *Robust motion detection in real-life scenarios,* Springer.

30. Pal, S.K., Petrosino, A. and Maddalena, L. (2012) *Handbook on soft computing for video surveillance,* CRC Press.

31. Stauffer, C. and Grimson, W.E.L. (1999) "Adaptive background mixture models for real-time tracking", *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference.*

32. Barron, J.L., Fleet, D.J. and Beauchemin, S.S. (1994) "Performance of optical flow techniques", *International journal of computer vision*, vol. 12, no. 1, pp. 43-77.

33. Olson, T. and Brill, F. (1997) "Moving object detection and event recognition algorithms for smart cameras", *Proc. DARPA Image Understanding Workshop,* pp. 205.

34. Kim, S.W., Yun, K., Yi, K.M., Kim, S.J. and Choi, J.Y. (2012) "Detection of moving objects with a moving camera using non-panoramic background model", *Machine Vision and Applications,* pp. 1-14.

Chapter 2
LITERATURE REVIEW

2.1 OVERVIEW

This chapter introduces the uses, problems and methods surrounding the topic of foreground-background video segmentation. Discovery of a fast, accurate and reliable technique for motion segmentation is still a challenging research issue in the field of multimedia. Temporal differencing, optical flow and background subtraction are the three most common approaches used for motion segmentation.

Temporal differencing exploits the frame difference between consecutive frames in a video to detect moving regions; the method is highly adaptive to changing environments but in most cases does not perform well while capturing shapes of certain kinds of moving objects. Background Subtraction is most widely used technique for detecting moving objects in video; the method compares the current image with a model of the background image as the reference to detect motion. Optical flow is an estimation of the local image motion. These methods along with their varying implementations are studied and explained with regards to methodology and accomplishment.

2.2 TEMPORAL DIFFERENCING

Temporal differencing was one of the earlier approaches introduced to aid the task of motion estimation. The technique extracts moving regions from video sequence by comparing the difference between consecutive frames; the method is very adaptive to dynamically changing environments. However it generally fails to extract complete shapes of certain kinds and thus introduces the foreground aperture problem. Algorithms based on this technique require the support of additional methods to detect stopped objects.

A typical implementation of Temporal Differencing detects motion by using a threshold procedure that was used against the inter-frame difference of blocks. The output from this would then be binarised by a pre-determined threshold value that gave the capacity to determine whether the reference in question was active or not [22].

The strengths of this method were in its adaptation to dynamic changes in the background of images sequences [16]. Oral and Deniz (2007) highlighted that the weaknesses within this method lay in its failure to extract all relevant pixels whilst performing foreground segmentation. This would thus cause the development of holes in detected foreground objects as a result.

The temporal differencing method was centered on the comparison of video frames. The video frames to be compared are required to be separated by a constant time δt to calculate the absolute difference [19]. This reveals the changed regions in the image sequence. A limitation within this method is the inability to track any objects in the event of substantial camera movement. This simple variant of the temporal differencing method employs the use of a threshold function to determine change. The pixel wise difference function is defined in function (2.1):

$$\Delta_{\boldsymbol{n}} = |\boldsymbol{I_n} - \boldsymbol{I_{n-1}}| \tag{2.1}$$

Where I_n denotes the intensity of the n^{th} frame, I_{n-1} denotes the intensity of the n-1 *frame.*

A motion image M_n is extracted by thresholding from the following function (2.2):

$$\mathbf{M_n}(\mathbf{u}, \mathbf{v}) = \begin{cases} \mathbf{I_n}(\mathbf{u}, \mathbf{v}), \Delta_{\mathbf{n}}(\mathbf{u}, \mathbf{v}) \geq \mathbf{T} \\ \mathbf{0} \qquad\qquad , \Delta_{\mathbf{n}}(\mathbf{u}, \mathbf{v}) < \boldsymbol{T} \end{cases} \tag{2.2}$$

Where T, u and v denote the threshold, intensity of the current and previous frame respectively. The benefit of this algorithm is robust to varying illumination, if the time span is short. However, a very small difference between two consecutive frames may result in unchanged pixels if the moving object has some overlapping blocks.

Lipton *et al* (1998) gained motivation from a real-time system titled Pfinder (person finder) [19]. Pfinder was designed to track and read human behaviour. It used region-based features to provide real-time detection in video sequences [19]. Lipton *et al* (1998) considered the approaches of Pfinder for their system, which included corresponding areas of interest to listed target models constructed. It also involved the requirement of a high number of pixels for the target. After researching these approaches they highlighted potential complications that each presented individually. For scenarios involving outdoor surveillance systems, the requirement of a high number of pixels for the target would be unreachable in all instances. Additionally, there may be various targets of interest that could be ignored because of omitted target models.

For these reasons Lipton *et al* (1998) proposed a target classification system that made use of temporal consistency and temporal differencing techniques. Temporal consistency was used in tandem with temporal differencing to categorise and track potential targets in image sequences. It accomplished this by assigning classification metrics to targets and then proceeded to track them by utilising temporal differencing and template matching techniques [19]. The system would identify any classified target over a prolonged period of time and distance and was susceptible to interferences from the background, appearance changes and temporarily motionless targets [19].
Chang *et al* (2005) proposed a method based on temporal difference solely for the task of "change detection" [22]. The task operated within various systems including image processing and visual surveillance. They regard change detection to be extremely critical in the video surveillance field, as it possesses the means to perform foreground-background segmentation on moving objects and provide continued surveillance on that target [22].

2.2.1 KALMAN FILTERING

Lipton *et al* (1998) recognised that many tracking systems previously proposed were based on Kalman filtering [19]. The Kalman filter is a processing algorithm that is capable of generating optimal estimates from a set of measurements. Its use is within the field of target tracking in the presence of unimodal Gaussian density noise. However, Lipton *et al* (1998) revealed a flaw with this method. Due to the Kalman filtering method being limited to unimodal Gaussian densities, its usefulness is also limited. This is because noise in most real systems generally takes the form of other distributions. Thus a Kalman filter fails to support simultaneous alternative movement [19].

Ridder *et al* (1995) used this recursive technique to solve issues surrounding video sequences that contained objects that were not continuously moving [13]. They also used Kalman filtering to address an object's change of speed during a sequence, which is accomplished by suppressing the foreground adaptation [13].

2.3 OPTICAL FLOW

The optical flow algorithms are primarily based on motion vectors and use the spatiotemporal derivatives of pixel values or block matching techniques [23-24]. This procedure is capable of detecting the person in a changing background. Thus, the method has the capability of extracting the foreground from complex outdoor scenes that contain non-stationary vegetation [24]. Since this method does not use background subtraction, it produces good results in cases where the background image is not available [24-25]. Obviously, in these cases the traditional background methods fail. The Optical flow models are based on a two-frame differential method for motion estimation. This method estimates the motion between two frames, which are taken at a time interval *t*. Optical flow methods are very useful in pattern recognition, computer vision and other image processing applications.

All of the above methods only consider the changing parts of an image as the foreground. This is not always a true assumption and can further cause two types of problems respectively called *foreground aperture* and *false foreground detection*. The case of foreground aperture occurs when the foreground (moving part) is much bigger compared to the background and thus is assumed as the background of the video sequence. This situation might happen because the object in the frame is temporarily still or because it shares the same texture or colour with other objects, and thus the motion is only detected in the borders. The case of false foreground detection occurs when there are light variations in the background or very small waving of tree branches.

Optical flow is used to transform one image into another. It monitors and uses the changes in pixel intensity to determine pixel movement in image sequences [17].

Assuming the image sequence intensity is given by:

$$I(x, y, t)$$

Where I represents the intensity and x, y denotes the location coordinates and t represents time. By expanding the equation, the following calculation is derived [17]:

$$\boldsymbol{I(x + dx, y + dy, t + dt) \approx I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt} \tag{2.3}$$

Where (dx, dy) denotes a change in space and dt is the change with time, and ∂ is a partial derivative used in differential equations.

After dividing the equation (2.3) with dt the derived optical flow is:

$$\boldsymbol{\nabla I \cdot V + \frac{\partial I}{\partial t} = 0} \tag{2.4}$$

Where $V = (V_x, V_y) = \left(\frac{dx}{dt}, \frac{dy}{dt}\right)$ denotes the motion vector and $\frac{\partial I}{\partial t}$ represents how fast the image intensity changes with time [17].

Within optical flow, there are five common methods, namely: gradient-based, block-matching-based, energy-based, phased-based and neuro-dynamic type algorithms. The limitation that each of these methods suffers from is the inability to avoid the negative effect of lighting on the output, which results in background noise [10]. Ridder *et al*'s (1995) investigations of the optical flow approach highlighted limitations with the method [13]. To successfully detect a foreground object it had to be continuously moving, if not it would be mistaken for the background. Also, the implementation of the method was computationally expensive and required specific hardware to deploy the method on real-time applications [21].

Kim *et al* (2012) investigated the calculation of optical flow in pixels to assist them in developing a method to detect moving objects with a moving camera using non-panoramic backgrounds [9]. They learned that the optical flow method detected changes within adjacent frames. However, like Ridder *et al* (1995) they too found shortfalls associated with the method [13]. These shortfalls were the high level of computation required and the restriction of camera movement [9]. As an alternative, they proposed a solution that would not require a panoramic background model but was able to detect moving objects with a moving camera. Their method also boasts the feature of allowing real-time and online computation. To achieve this they opted to integrate the functions of a background subtraction method [9].

2.4 BACKGROUND SUBTRACTION

The *background subtraction* or *frame differencing algorithms* provide one of the most convenient ways for foreground detection [26-28] due to their simple implementation and processing. In these algorithms, the frame under analysis is compared to (i.e. subtracted from) foreground-free frames, as given below in the Equation (2.5):

$$Foreground = R_i - R_j \tag{2.5}$$

Where R_i is a foreground-free (reference) image and R_j is the image taken when the foreground object is present. Thus, a simple subtraction between the two images results in the foreground object. In these methods, each video frame is continuously compared to the reference image (background model). The situation, where pixels of the current frame deviate significantly from respective pixels of the reference image, points to a moving object in the current image. Furthermore, the subtraction algorithms are used to manipulate the obtained foreground pixels for object location and tracking. The main drawback of these algorithms is that in real situations it is not always possible to have foreground-free images. Also, the algorithms encounter problems in several cases of background variation, e.g., cases of camera motion, background that contains shadows, wavering of plant branches or illumination changes. To overcome these problems, the Gaussian function can be applied for optimum results. The Gaussian function describes the distribution of colour in the stable background of an object. This process is performed on each pixel of the object of interest [29, 30]. To follow the changes in the background of the video, the Gaussian model parameters are recursively updated.

Background subtraction methods are the most utilised of the three main approaches to foreground detection. The method employs the use of a background model that acts as a reference image for the background. This allows the method to detect foreground objects that remain still for some part of the sequence as well as objects that are continuously in motion [16]. There have been various proposals for background models made, however, an ideal example should satisfy the following conditions: resistance to noise, resistance to changes in illumination of the scene, flexibility to changes in the original background, resistance to any movement, produce no distortions on foreground, account as little as possible for post-processing operations, be efficient and consistent and ease of implementation [16].

From the many background model proposals made by different authors, no model fulfills every condition stated above thus far. However, due to the popularity and relative simplicity of the background subtraction method, authors are continuously developing new ways to improve it.

The simple background subtracting method is the most popular of all methods that perform foreground-background video segmentation. It is the foundation for all background subtraction methods and presents the least difficulty to implement. It performs the task of detecting moving regions in an image using pixel intensities to find the difference of the reference image and an image from the sequence [16]. The simple background subtracting is computed as follows [16]:

$$\boldsymbol{D}(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \mathbf{1} & |\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{B}(\boldsymbol{x},\boldsymbol{y})| > \boldsymbol{\tau} \\ \mathbf{0} & \textbf{otherwise} \end{cases} \tag{2.6}$$

Where $D(x, y)$ denotes the binary difference image. $B(x, y)$ denotes the reference image which is the static background. $I(x, y)$ denotes current image from a sequence, which may contain foreground objects and the background. x and y are the pixels locations and τ is selected as noise threshold to discriminate between foreground and background pixels.

If in case that the difference between $B(x, y)$ and $I(x, y)$ exceeds τ, it is assumed that the pixels contain motion. This results in a binary image (x, y), where active pixels are marked with 1 and non-active with 0.

Of all background subtraction approaches, this method suffers the most as it is affected by noise and moving backgrounds. It also suffers from illumination changes, which describes a change in brightness. For example, in an outdoor scene, moving clouds may obscure sunlight; this would temporarily create a change in brightness for a particular time period in a sequence. Consequently, researchers began to re-develop background subtraction methods by introducing adaptive background models. Temporal averaging is an example of this. This method improves upon the simple background subtraction algorithm by significantly reducing the effects of illumination changes in the background. An adaptive background model is computed as follows [16]:

$$\boldsymbol{D}(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \mathbf{1} \ |\boldsymbol{I}_{\boldsymbol{k}}(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{B}_{\boldsymbol{k}}(\boldsymbol{x},\boldsymbol{y})| > \boldsymbol{\tau} \\ \mathbf{0 \ otherwise} \end{cases} \tag{2.7}$$

Where $D(x,y)$ denotes the binary difference image, $B_k(x,y)$ denotes the adaptive background frame, $I_k(x,y)$ denotes current frame from a sequence and $\boldsymbol{k}$ is the respective frame number. If the difference is larger than the defined threshold τ, it is considered that the pixel exhibits motion. Active pixels are marked with 1 and non-active ones with 0.

To improve the management of dynamic backgrounds when performing background subtraction the background model needs to be updated on frequent basis, to ensure consistent reliable motion detection. The background model is updated by integrating incoming information to the current background image using the following filter [16],

$$\boldsymbol{B}_{(\boldsymbol{k}+\mathbf{1})}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{\alpha}\boldsymbol{I}_{\boldsymbol{k}}(\boldsymbol{x},\boldsymbol{y}) + (\mathbf{1}-\boldsymbol{\alpha})\,\boldsymbol{B}_{\boldsymbol{k}}(\boldsymbol{x},\boldsymbol{y}) \tag{2.8}$$

Where α is an adaptation coefficient chosen arbitrarily. The greater the value of α is, the faster the integration of changes to the background image. However, α cannot be too big as it may cause artificial trails to form behind the moving objects [16]. Adaptive background models are still not faultless and struggle to extract all relating pixels for foreground objects. It also struggles to deal with backgrounds that present sudden illumination changes.

Thus, statistical approaches were introduced to combat these weaknesses. Statistical approaches perform motion segmentation by comparing background models and pixels or blocks pixels statistics. This method of background subtraction improve the adaptive background models as it is more robust to noise, shadow and changes in illumination. The method also facilitates the construction of more dynamically updated background models by using the characteristics of pixels or block pixels. This is in complete contrast to the adaptive modeling approach discussed earlier.

Mean μ_{xy} and standard deviation σ_{xy} are computed from the frames in $[t_0, t_{k-1}]$ time interval for each of the image elements in the background image.

As mentioned earlier, statistical approaches perform motion segmentation by comparing background model with the pixels or blocks pixels statistics, which results in the absolute difference between an incoming frame I_k and the means of background pixels. The pixel is a part of a foreground object if the difference is greater than λ times the standard deviation else it belongs to background. A simple statistical approach is computed using the following equation:

$$\boldsymbol{D}(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \mathbf{1} \; |\boldsymbol{I}_{\boldsymbol{k}}(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{\mu}_{\boldsymbol{xy}}| > \boldsymbol{\lambda}\boldsymbol{\sigma}_{\boldsymbol{xy}} \\ \mathbf{0} \textbf{ otherwise} \end{cases} \tag{2.9}$$

Statistical methods improve upon the traditional background subtraction methods. Kim *et al* (2012) considered an extension to the background subtraction method [9]. This approach required the implementation of panoramic background models, which would be used in conjunction with camera motion matrices. This revealed an issue with image registration that would cause errors. Upon addressing this issue with the development of a panoramic background mosaic, Kim *et al* (2012) realised that the method still suffered from difficulties regarding background adaptation, slow initialisation and large computation memory [9]. Not being deterred, they looked to take a different approach. Instead of developing a large background model like many authors proposed previously, they opted to design a small background model that utilised spatial and temporal information. Additionally, they used an image registration method to find the overlapped region and the most recent covered region in the current frame [9, 31,32].

2.4.1 LIMITATIONS OF BACKGROUND SUBTRACTION

The background subtraction method has limitations that the early work of Grimson *et al* (1999) attempted to solve [3]. The disadvantages were related to the constant updating of the background model for each frame. This process presented some degree of difficulty as sections of the frame are required to be absent of objects before updating can commence [2]. Another of the common problems known to surround the background subtraction method was its sensitivity to active scenes; this included changes to lighting, any background movement and possible shadows [11].

In an attempt to address these problems, Grimson *et al* (1999) proposed an adaptive multi-coloured background model for real-time tracking [3]. Several authors also looked to improve the background subtraction technique and eliminate any disadvantages attached to the method. It was Kaewtrakulpong and Bowden's (2001) main objective to analyse and improve on this proposed method of Grimson *et al* (1999) [5]. Initially, they considered Grimson *et al*'s (1999) proposed solution to be successful in solving the many problems concerned with the background subtraction method [3]. However, they later found and highlighted the problems with the method to be: slow to learn at the beginning of usage particularly in busy environments and its inability to distinguish between moving shadows and objects [5]. Kaewtrakulpong and Bowden (2001) believed that to achieve their objective they would need to develop a solution that would learn faster, improve on accuracy and respond better to changing environments in comparison to Grimson *et al*'s solution [5].

With this objective in mind, they explored many different proposals and chose to utilise methods from each to formulate their own. They noted how Grimson *et al* (1999) used the Gaussian mixture model and the universal use and importance of the expectation maximisation algorithm used in conjunction with it [3]. Kaewtrakulpong and Bowden (2001) also took a keen interest in McKenna *et al*'s (1999) development of a most recent window that represented a procedure, which without alerting the previous model structure update the previous estimate with new data

[12]. For a good estimate of mixture method from the beginning before all samples were processed [5], they used it to form the basis of theirs and introduced the use of expected sufficient statistics update equations. This method helped them to speed up the learning in busy environments and improve accuracy.

2.4.2 SHADOW DETECTION

Kaewtrakulpong and Bowden (2001) developed their model further by incorporating a shadow detection feature. This separates further the performance gap between their model and Grimson *et al*'s (1999) [5]. To achieve this they considered separated chromatic and brightness component color model, which makes the most of their mixture model. Their method for calculating which part of a given image was an object's shadow is by comparing the non-background pixel with the current background [5]. The resulting values of the difference between the chromatic and brightness components define whether or not the related pixels are considered a shadow.

2.4.3 IMPROVEMENT METHODS TO BACKGROUND SUBTRACTION

Maddalena and Petrosino (2008) had similar motivation to Kaewtrakulpong and Bowden (2001) with respect to wanting to improve the background subtraction method [11]. However, Maddalena and Petrosino (2008) took a different approach. They pointed to the fact that visual surveillance systems were one of the main uses for foreground-background segmentation. Consequently they decided to focus their efforts on the detection phase of such systems. They proposed an algorithm that could automatically generate a background model and detect moving objects [11]. They went about this by designing their algorithm to adopt a novel neural network mapping method that learn patterns, both motion and motionless, then update the background model. The selected network is organized as a 2-D flat grid of neurons that facilitates the production of training samples. It upholds the values gained from input patterns that are similar to the output patterns. The network structure also allows for a more efficient learning process [11].

Manzanera and Richefeu (2006) proposed a $\sum - \Delta$ filter to be used in conjunction with a spatio-temporal regularisation algorithm for the task of motion detection [18]. Their objective was to successfully tackle and overcome problems posed by highly dynamic background scenes, whilst still maintaining accuracy and a computational efficient method. They also wanted to focus their efforts on working with long autonomous applications such as video surveillance.

To begin they highlighted the specific areas involved in video surveillance applications that they wanted to address. They also acknowledged the growth in the number of autonomous operations within video surveillance systems. So any background scenes related to these unmanned surveillance cameras would need to be updated someway. Henceforth, Manzanera and Richefeu (2006) looked to incorporate a temporarily adaptive background model that utilised a local estimation.

2.4.4 HISTOGRAM-BASED GRAPH CUT ALGORITHM

Authors have begun to combine different foreground detection approaches to achieve greater improvements to existing methods. Each approach has advantages and so combining them has seen methods advance. The work of Kim and Paik (2012) sees the combination of background subtraction and optical flow approaches [8]. They incorporated a histogram-based graph cut algorithm with automatically generated label maps. This method was made up of four significant steps to give an end result of a segmented image. These were: pre-processing by over segmentation, initial label map generation, updating of the label map and object segmentation respectively.

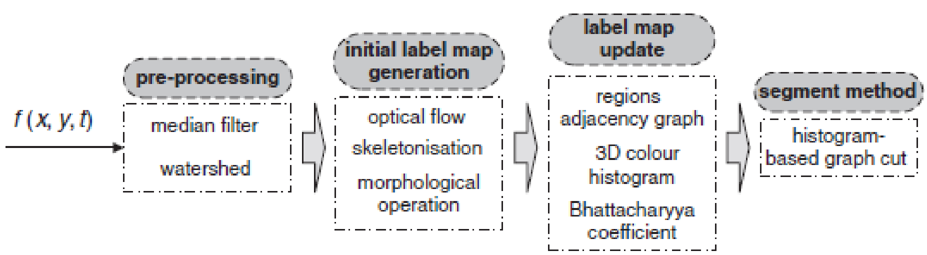


Figure 2.1 Proposed histogram-based graph cut algorithm using label maps

The pre-processing step involved the over segmentation of the input sequence using the median filter-based watershed algorithm. Optical flow and morphological skeletonisation processes facilitate the generation of the label map. Under the notion that any moving area in a video sequence is of importance, the automatic generation of an initial label map using optical flow assigns the following function:

$$\boldsymbol{V} = [\boldsymbol{V_{x,}}\ \boldsymbol{V_y}]^{\boldsymbol{T}} \qquad (2.10)$$

Where $\boldsymbol{V}$, is the velocity vector, x and y represents the image co-ordinates and T represents time. Therefore the optical flow equation is expressed as:

$$(\partial f/\partial x)\boldsymbol{V_x} + (\partial f/\partial y)\boldsymbol{V_y} + (\partial f/\partial t) = \boldsymbol{0} \qquad (2.11)$$

Where (x, y) represents the image coordinate and t represents time. ∂f and ∂y are partial derivatives used to represent the differentiation of the velocity vector coordinates.

The moving area R_M is expressed as:

$$\boldsymbol{R_M} = \left\{(\boldsymbol{x}, \boldsymbol{y}) | \sqrt{\boldsymbol{V_x^2} + \boldsymbol{V_y^2}} > \boldsymbol{\theta_M}\right\} \qquad (2.12)$$

Where θ_M is a pre-specified threshold.

This is updated by the use of the Bhattacharyya coefficient and colour histogram. The final stage of the process sees the histogram-based graph method used to perform the foreground segmentation [8]. The advantage of this method is the automatic process of performing foreground segmentation without any user interaction. This assisted in speeding up the whole process, making it faster than existing foreground-background segmentation methods.

Manzanera and Richefeu (2006) considered adopting the histogram approach into their proposal as they too found strengths with the method. They considered its use for foreground detection. This process functions by analysing the histogram values taken by each pixel within a fixed number of past frames. Then the resulting averages of the histogram could be chosen to set the whole background value. This enables the foreground to be detected after comparing the difference in frames using the background histogram variance. However, weaknesses were found which halted any further consideration of the method. This was the requirement for a lot of memory. As a result of this they abandoned the histogram method [18].

2.4.5 SPATIO-TEMPORAL SALIENCY

The combination of foreground detection approaches allowed for the disadvantages of one approach to be eliminated by the other and vice versa. Xia *et al* (2013) studied two different approaches individually for their proposal of a moving foreground detection method [15]. They noted past and current methods were disappointing when it came to handling objects that were moving under different speeds and exposed to diverse lighting. From investigating visual saliency methods they noticed such methods increased detection capability because of spatial information capacity [15]. However, from experiments taken by Itti *et al* (2003), they discovered that the method mistook a lot of background regions for the foreground [4]. They then decided to incorporate background subtraction method thus gaining its qualities, whilst incorporating the spatial information obtained by visual saliency to improve the foreground detection performance. After all considerations, they developed a novel object detection algorithm that featured a Spatio-temporal saliency approach. It operated by modifying the pixel-wise learning rate adaptability to give an improved detection capability [4]. Thus making foreground detection more robust when objects move at different speeds and under illumination changes.

Table (2.1) shows the brief description and disadvantages of all aforesaid cited algorithms.

Table 2.1 Algorithms comparison

Method	Disadvantages
Temporal differencing	• Cannot extract complete shapes of certain kinds of moving objects. • Development of holes in detected foreground objects. • Inability to track any objects in the event of substantial camera movement
Background subtraction	• The main drawback of these algorithms is that in real situations it is not always possible to have foreground-free images in advance. • Problems in cases of camera motion, background that contains shadows, wavering of plant branches or illumination changes and raining. • Constant updating of the background model for each frame. • Setting Threshold value
Optical Flow	• Foreground aperture and false foreground detection. • Inability to avoid the negative effect of lighting on the output, which results in background noise [10]. • To successfully detect a foreground object it had to be continuously moving, if not it would be mistaken for the background. • Also, the implementation of the method was computationally complex and required specific hardware to deploy the method on real-time applications [21] • Restriction of camera movement [9]
Histogram-based graph cut algorithm	As using histogram values of each pixel within fixed past frames • Time consuming • High memory requirement
Spatio-temporal saliency	This method increase accuracy due to pixel wise learning: • Requires high memory • High execution time

2.5 CONCLUSION

In this chapter the discussion was mainly on three popular foreground detection approaches and its variants: temporal differencing being the earliest one, optical flow and the most popular and widely use method background subtraction.

The main drawbacks of temporal differencing are: it cannot extract complete shapes, development of holes in detected foreground objects and inability to track any objects in the event of substantial camera movement. The Optical flow method encounters the problems of *foreground aperture* and *false foreground* detection, inability to avoid the negative effect of lighting on the output, which results in background noise and this method also required specific hardware to deploy the method for real-time applications. The main drawback of *background subtraction* or *frame differencing algorithms* is that it requires reference image in advance, which is not possible in the real world (e.g. live) videos. Also, the algorithms encounter problems in several cases of background variation, e.g., in case of camera motion, background that contains shadows, wavering of plant branches or illumination changes.

2.6 References

1. Collins, R.T., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O. and Burt, P. (2000) *A system for video surveillance and monitoring,* Carnegie Mellon University, the Robotics Institute Pittsburg.

2. Donatello, C., Pasquale, F., Gennaro, P., Francesco, T. and Mario, V. (2010) "An experimental evaluation of foreground detection algorithms in real scenes", *EURASIP Journal on Advances in Signal Processing,* vol. 2010.

3. Stauffer, C. and Grimson, W.E.L. (1999) "Adaptive background mixture models for real-time tracking", *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference*.

4. Itti, L., Dhavale, N. and Pighin, F. (2004) "Realistic avatar eye and head animation using a neurobiological model of visual attention", *Optical Science and Technology, SPIE's 48th Annual Meeting*International Society for Optics and Photonics, pp. 64.

5. KaewTraKulPong, P. and Bowden, R. (2002) "An improved adaptive background mixture model for real-time tracking with shadow detection" in *Video-Based Surveillance Systems* Springer, pp. 135-144.

6. Kim, K., Chalidabhongse, T.H., Harwood, D. and Davis, L. (2005) "Real-time foreground–background segmentation using codebook model", *Real-Time Imaging,* vol. 11, no. 3, pp. 172-185.

7. Kim, K., Chalidabhongse, T.H., Harwood, D. and Davis, L. (2006) "PDR: Performance Evaluation Method for Foreground-Background Segmentation Algorithms", *EURASIP Journal on Applied Signal Processing*.

8. Kim, D. and Paik, J. (2012) "Automatic moving object segmentation using histogram-based graph cut and label maps", *Electronics Letters,* vol. 48, no. 19, pp. 1198-1199.

9. Kim, S.W., Yun, K., Yi, K.M., Kim, S.J. and Choi, J.Y. (2012) "Detection of moving objects with a moving camera using non-panoramic background model", *Machine Vision and Applications,* pp. 1-14.

10. Li, W., Wu, X., Matsumoto, K. and Zhao, H. (2010) "Foreground detection based on optical flow and background subtract", *Communications, Circuits and Systems (ICCCAS), 2010 International Conference on* IEEE, pp. 359.

11. Maddalena, L. and Petrosino, A. (2008) "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications", *Image Processing, IEEE Transactions on,* vol. 17, no. 7, pp. 1168-1177.

12. McKenna, S.J., Raja, Y. and Gong, S. (1999) "Tracking colour objects using adaptive mixture models", *Image and Vision Computing,* vol. 17, no. 3, pp. 225-231.

13. Ridder, C., Munkelt, O. and Kirchner, H. (1995) "Adaptive background estimation and foreground detection using kalman-filtering", *Proceedings of International Conference on recent Advances in Mechatronics Citeseer*, pp. 193.

14. Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P. (1997) "Pfinder: Real-time tracking of the human body", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, no. 7, pp. 780-785.

15. Xia, Y., Hu, R., Wang, Z. and Lu, T. (2013) "Moving Foreground Detection Based On Spatio-temporal Saliency", *IJCSI International Journal of Computer Science Issues,* vol. 10, issue 1, pp. 79-84.

16. Oral, M. and Deniz, U. (2007) "Centre of mass model–A novel approach to background modelling for segmentation of moving objects", *Image and Vision Computing,* vol. 25, no. 8, pp. 1365-1376.

17. Doshi, A. and Bors, A.G. (2010) "Smoothing of optical flow using robustified diffusion kernels", *Image and Vision Computing,* vol. 28, no. 12, pp. 1575-1589.

18. Manzanera, A. and Richefeu, J.C. (2007) "A new motion detection algorithm based on< i> Σ</i>–< i> Δ</i> background estimation", *Pattern Recognition Letters,* vol. 28, no. 3, pp. 320-328.

19. Lipton, A.J., Fujiyoshi, H. and Patil, R.S. (1998) "Moving target classification and tracking from real-time video", *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on* IEEE, pp. 8.

20. McFarlane, N.J. and Schofield, C.P. (1995) "Segmentation and tracking of piglets in images", *Machine Vision and Applications,* vol. 8, no. 3, pp. 187-193.

21. Di, M., Joo, E.M. and Beng, L.H. (2008) "A comprehensive study of kalman filter and extended kalman filter for target tracking in wireless sensor networks", *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on* IEEE, pp. 2792.

22. Chang, C., Chia, T. and Yang, C. (2005) "Modified temporal difference method for change detection", *Optical engineering,* vol. 44, no. 2, pp. 027001-027010.

23. Horn, B.K. and Schunck, B.G. (1981) "Determining optical flow", *Artificial Intelligence,* vol. 17, no. 1, pp. 185-203.

24. Iketani, A., Nagai, A., Kuno, Y. and Shirai, Y. (1998) "Detecting persons on changing background", *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on* IEEE, pp. 74.

25. Wixson, L. (2000) "Detecting salient motion by accumulating directionally-consistent flow", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, no. 8, pp. 774-780.

26. Cheng, F., Huang, S. and Ruan, S. (2011) "Illumination-sensitive background modeling approach for accurate moving object detection", *Broadcasting, IEEE Transactions on,* vol. 57, no. 4, pp. 794-801.

27. Haritaoglu, I., Harwood, D. and Davis, L.S. (2000) "W< sup> 4</sup>: real-time surveillance of people and their activities", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, no. 8, pp. 809-830.

28. Zhao, T. and Nevatia, R. (2004) "Tracking multiple humans in complex situations", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, no. 9, pp. 1208-1221.

29. Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P. (1997) "Pfinder: Real-time tracking of the human body", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 780-785.

30. Boult, T., Micheals, R., Gao, X., Lewis, P., Power, C., Yin, W. and Erkan, A. (1999) “Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets”, *Visual Surveillance, 1999. Second IEEE Workshop on, (VS’99)* IEEE, pp. 48.

31. Li, L., Huang, W., Gu, I. Y. H., & Tian, Q. (2004) “Statistical modeling of complex backgrounds for foreground object detection”, *Image Processing, IEEE Transactions on*, vol. 13, pp.1459-1472

32. Cucchiara, R., Grana, C., Piccardi, M., & Prati, A. (2003). “Detecting moving objects, ghosts, and shadows in video streams” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, 1337-1342.

Chapter 3
SEGMENTATION PERFORMANCE MEASURES

3.1 OVERVIEW

In this chapter, there is a discussion about various techniques to judge the quality of any foreground detection algorithm, since in most of the papers researchers have used only a few of the performance measures for the quality evaluation of their respective algorithm e.g., precision, recall, area under the curve and f-measure. However, for the fair evaluation in the proposed work 11 different performance measures are taken for consideration of performance.

3.2 PERFORMANCE MEASURES

There are a significant number of foreground-background segmentation algorithms being developed for private or public use. Each method differs from the other with the introduction of new techniques or features aimed for functional improvement. It is important to be able to quantitatively assess and evaluate these algorithms to determine how effective they are. It is believed that foreground detection is an essential component of many video analysis systems, yet there was not a clear method readily available to be adopted by all. Following this initial assessment, various methods were devised to analyse and test different methods under various conditions to highlight the proposed method's strengths and weaknesses.

The performance of any foreground detection algorithm can be judged via qualitative or quantitative methods. The qualitative method is applied by a human who judges the visual quality of results based on human visual perception. However, most of the researchers opt for the quantitative method as an accurate tool for performance measurement. In terms of generating valid ground truth, quantitative evaluation is a difficult and time consuming job, [1, 2], since the ground truth which is the correct representation that is expected from the proposed algorithm, is required to be obtained from every image of a video sequence. A second issue is that, if ground truth is generated by humans, each human observer can segment differently for the same data at different timings. Another issue is to describe the relative importance of the different types of errors as there are various quantitative methods to compare ground truths with respect to the candidate binary mask. There are different standard procedures for comparing the ground truth to a candidate binary change mask. In general, the following parameters are involved while calculating different performance measures:

3.2.1 PARAMETERS OF PERFORMANCE MEASURES

- True Positive (*tp*) refers to the number of foreground pixels correctly detected.
- False Positive (*fp*) refers to the number of background pixels incorrectly detected as foreground or, in other words, the average of false alarms per frame.
- False Negative (*fn*) refers to the number of foreground pixels incorrectly detected as background, or we could say, the average of false misses.
- True Negative (*tn*) refers to the number of background pixels correctly detected.

The above parameters can be seen in illustrated form in the Figure (3.1), describing *tp, fp, fn,* and *tn,* respectively. In this figure, the detected foreground means the result obtained from the proposed algorithm and ground truth foreground is considered to be the perfect result based on human segmented result.

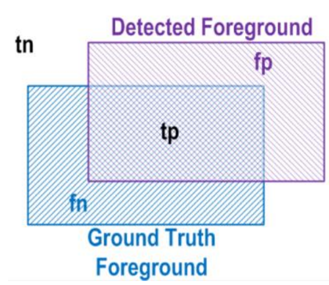


Figure 3.1 Confusion metric variables

In Table 3.1, C1 represents first column elements *tp* and *fp* and C2 represents second column elements *fn* and *tn* of the confusion matrix.

Table 3.1 Confusion metric binary values

Resultant		Ground Truth	Resultant image
C1	*tp*	0	0
	fp	1	0
C2	*fn*	0	1
	tn	1	1

Based on the values of *tp*, *fp*, *fn* and *tn*, as given in Table (3.1), confusion matrix for the binary classification and its corresponding array representation can be seen in Table (3.2).

Table 3.2 Confusion metric classifiers

Data Class	Classified as *positive/ detected*	Classified as *negative/not detected*
positive *(pos)/actual object*	true positive (*tp*)	false negative (*fn*)
negative *(neg)/non-object*	false positive (*fp*)	true negative (*tn*)

From Table (3.2), we can derive its mathematical form as given in Equation (3.2).

$$Confusion\,matrix = [C1 \quad C2] \tag{3.1}$$

or

$$Confusion\,matrix = \begin{bmatrix} tp & fn \\ fp & tn \end{bmatrix} \tag{3.2}$$

To quantitatively compare the proposed method, the desired pixels for the foreground objects in the test images were manually labeled and taken as the ground truth. Then the true positive rate (*tpr*) and false positive rate (*fpr*) pixels were computed for the segmentation results. The *tpr* is defined as the ratio of the number of correctly classified object pixels to the number of total object pixels in the ground truth. The *fpr* is defined as the ratio of the number of background pixels but classified as object pixels to the number of background pixels in the ground truth. Obviously, the higher the *tpr* and the lower the *fpr*, the better is the proposed method performance [3]. It is to be noted that in the performance measurements described below all values can be converted into percentages for more clarity.

3.3 Different Performance Measures

There are 11 different performance measurements: precision, recall, F-score, specificity, area under the curve, BER%, accuracy, geometric mean of sensitivity and specificity, similarity and false positive rate.

3.3.1 Precision

Precision is used to quantify how well the proposed algorithm matches the ground truth. Some researchers use precision and recall [4, 5]. Precision is also known as Positive Predictive Value (PPV). Precision is defined by Equation (3.3), and is the measure of how well we have identified the ground truth foreground without misidentifying the background. fp is the area of miscalculated foreground in the resultant segmented image. The lower its value the greater is the value of precision.

$$P = \frac{tp}{tp + fp} \qquad (3.3)$$

3.3.2 Recall or Sensitivity or True Positive Rate (*TPR*)

As stated earlier, Recall is another measure used to quantify how the proposed algorithm matches the ground truth. Recall, or Sensitivity, or equivalently True Positive Rate (TPR) is defined by Equation (3.4) and is a measure of how well we have identified the ground truth foreground without misidentifying the foreground [6]. The ideal % value of Recall is 100.

$$R_S = \frac{tp}{tp + fn} \qquad (3.4)$$

3.3.3 F-SCORE OF PRECISION AND RECALL

F-score is the weighted percentage average of precision and recall. F-score of Precision and Recall (i.e., harmonic mean) is defined in Equation (3.5). F-score measures the proposed methods accuracy. The ideal % value of F-score is 100.

$$\%F_{scorePR} = \frac{100 \times 2 \times (recall \times precision)}{recall + precision} \tag{3.5}$$

3.3.4 SPECIFICITY OR TRUE NEGATIVE RATE

This measure describes the ratio of detected foreground pixels that are true positives. If the value of specificity is 100%, this shows that the segmentation process recognizes all actual negatives, or in other words, 100% specificity shows no positives are incorrectly tagged. Specificity is defined by Equation (3.6), and is a measure of how well we have been able to identify the ground truth background without misidentifying the ground truth foreground. It is the opposite of precision; the lower the value of fp, the greater the value of specificity. The ideal % value of specificity is 100.

$$Spec = \frac{tn}{tn + fp} \tag{3.6}$$

3.3.5 BALANCE CLASSIFICATION RATE OR AREA UNDER THE CURVE

This statistical tool is also called *Yule Coefficient* (YC). Balance Classification Rate (BCR) or Area Under the Curve is defined by Equation (3.7), and is the overall measure of how well we have been able to identify the ground truth foreground and background. The greater the area under the curve, the better is the performance. The ideal % value of BCR or area under the curve is 100.

$$BCR \text{ } or \text{ } AUC = \frac{1}{2}\left(\frac{tp}{tp+fn} + \frac{tn}{tn+fp}\right) \tag{3.7}$$

or

$$BCR\ or\ AUC = \frac{1}{2}(Recall + \ Specificity)$$

3.3.6 Geometric Mean of Sensitivity and Specificity

Geometric mean of sensitivity and specificity is defined by Equation (3.8), and is an overall measure of how well we have been able to identify the ground truth foregrounds and backgrounds. The ideal % value of the geometric mean of sensitivity and specificity is 100.

$$G = \sqrt{recall \times specificity} \tag{3.8}$$

3.3.7 F-Score of Sensitivity and Specificity

F-score of sensitivity and specificity (i.e., harmonic mean) is defined by Equation (3.9) and is an overall measure of how well we have been able to identify the ground truth foregrounds and backgrounds. The ideal % value of F-score of sensitivity and specificity is 100.

$$\%F_{scoreSS} = \frac{100 \times 2 \times (recall \times specificity)}{recall + specificity} \tag{3.9}$$

3.3.8 %Balance Error Rate

Percentage Balance Error Rate is defined by Equation (3.10), and is the overall measure of how much we have misidentified the ground truth foreground and background. The ideal value of %Balance Error Rate is 0.

$$\text{BER}\ \% = \ 100 \times \left[1 - \frac{1}{2} \times \left\{\left(\frac{\text{tp}}{\text{tp} + \text{fn}} + \ \frac{\text{tn}}{\text{tn} + \text{fp}}\right)\right\}\right] \tag{3.10}$$

Or

$$\mathrm{BER}\,\% = 100 \times \left[1 - \frac{1}{2} \times \{(\mathrm{Recall} + \mathrm{Specificity})\}\right]$$

3.3.9 Similarity

Similarity is defined by Equation (3.11), also called Jaccard coefficient, which is a statistical tool used for comparing the similarity and diversity of sample sets. It is a measure of how similar the segmented foreground is to the ground truth foreground with 1 being most similar and anything less than 1 being increasingly less similar. The lower the value of $(fn + fp)$, the greater is the value of similarity. The ideal % value of similarity is 100.

$$Sim = \frac{tp}{tp + fn + fp} \tag{3.11}$$

3.3.10 Accuracy

Accuracy is also known as percentage correct classification. This statistical measure describes how well the proposed segmentation process excludes or identifies foreground pixels. 100% accuracy means that the values obtained from the proposed algorithm are exactly the same as the values in the ground truth. Accuracy is defined by Equation (3.12), and is a measure of how well we have identified the foreground and background ground truths without misidentifying the foregrounds and backgrounds. The ideal % value of accuracy is 100.

$$A = \frac{tp + tn}{tp + fn + fp + tn} \tag{3.12}$$

3.3.11 False Positive Rate

This measure is used to calculate the background pixels misclassified as foreground. False Positive Rate is defined by Equation (3.13), and is the fraction of the ground truth background that has been misidentified as foreground. The greater the value of tn, the lesser the value of the false positive rate. The ideal percentage value of false positive rate is 0.

$$FPR = \frac{fp}{fp+tn} \tag{3.13}$$

Table (3.3), shows the aforesaid performance measures and their mathematical expression.

Table 3.3 Performance measures

No	Performance Measure	Mathematical Expression
1	Precision	$P = \frac{tp}{tp + fp}$
2	Recall or Sensitivity or True Positive Rate (*tpr*)	$R_S = \frac{tp}{tp + fn}$
3	F-score of Precision and Recall	$\%F_{scorePR} = \frac{100 \times 2 \times (recall \times precision)}{recall + precision}$
4	Specificity or True Negative Rate	$Spec = \frac{tn}{tn + fp}$
5	Balance Classification Rate or Area Under the Curve	$BCR\ or\ AUC = \frac{1}{2}\left(\frac{tp}{tp + fn} + \frac{tn}{tn + fp}\right)$
6	Geometric Mean of Sensitivity and Specificity	$G = \sqrt{recall \times specificity}$
7	F-Score of Sensitivity and Specificity	$\%F_{scoress} = \frac{100 \times 2 \times (recall \times specificity)}{recall + specificity}$
8	%Balance Error Rate	$\mathrm{BER}\ \% = 100 \times \left[1 - \frac{1}{2} \times \left\{\left(\frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}} + \frac{\mathrm{tn}}{\mathrm{tn} + \mathrm{fp}}\right)\right\}\right]$
9	Similarity	$Sim = \frac{tp}{tp + fn + fp}$
10	Accuracy	$A = \frac{tp + tn}{tp + fn + fp + tn}$
11	False Positive Rate	$FPR = \frac{fp}{fp + tn}$

3.4 CONCLUSION

In this chapter there was a discussion about 11 performance measures. These measures compare the ground truth with the obtained results from the respective output of algorithm. These measures provide a statistical description of the object detection algorithm measuring each type of error. In this way it is possible to perform a rational evaluation among different algorithms. Furthermore it evaluates their strengths and weaknesses and allows the user to perform a reliable choice of the best method for a specific application.

3.5 REFERENCES

1. Hu, J., Kashi, R., Lopresti, D., Nagy, G. and Wilfong, G. (2001) "Why table ground-truthing is hard", *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on* IEEE, pp. 129.

2. Martínez-Martín, E. and del Pobil, A.P. (2012) *Robust motion detection in real-life scenarios,* Springer.

3. Collins, R.T., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O. and Burt, P. (2000) *A system for video surveillance and monitoring,* Carnegie Mellon University, the Robotics Institute Pittsburg.

4. Sen-Ching, S.C. and Kamath, C. (2004) "Robust techniques for background subtraction in urban traffic video", *Electronic Imaging 2004*International Society for Optics and Photonics, pp. 881.

5. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. and Rosenberger, C. (2008) "Review and evaluation of commonly-implemented background subtraction algorithms", *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* IEEE, pp. 1.

6. Gao, X., Boult, T.E., Coetzee, F. and Ramesh, V. (2000) "Error analysis of background adaption", *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on* IEEE, pp. 503.

7. Donatello, C., Pasquale, F., Gennaro, P., Francesco, T. and Mario, V. (2010) "An experimental evaluation of foreground detection algorithms in real scenes", *EURASIP Journal on Advances in Signal Processing,* vol. 2010.

Chapter 4
PRECISE-FOREGROUND-DETECTION ALGORITHM USING MOTION ESTIMATION, MINIMA AND MAXIMA INSIDE THE FOREGROUND OBJECT

4.1 INTRODUCTION

In this chapter the precise foreground mask is obtained in a complex environment by applying simple and effective methods on a video sequence consisting of multi-colour and multiple foreground object environment. To detect moving objects we use a simple algorithm based on block based motion estimation, which requires less computational time as pixel based motion estimation. To obtain a full and improved mask of the moving object, we use an opening-and-closing-by-reconstruction mechanism to identify the minima and maxima inside the foreground object by applying a set of morphological operations. This further enhances the outlines of foreground objects at various stages of image processing. Therefore, the algorithm does not require the knowledge of the background image. That is why it can be used in real world video sequences to detect the foreground in cases where we do not have a background model in advance. The comparative performance results are not only confined to a few conventional performance measures such as precision, recall and area under the curve but also to F-score of precision, specificity, geometric mean of sensitivity and specificity, F-score of sensitivity and specificity, %balance error, similarity, accuracy and false positive rate, and they finally demonstrate the effectiveness of the proposed algorithm.

4.2 WHY FOREGROUND DETECTION IS A CHALLENGING ISSUE?

It is well understood that image segmentation in terms of foreground and background separation is among one of the most interesting but demanding areas, from the implementation point of view, in the image processing field. However foreground detection is the prerequisite process for many image processing procedures. The present state of the art in foreground detection algorithms does not produce the same good quality results for different types of images [7-12] due to the varying nature of images and end user requirements. For these reasons, the segmentation process is much more difficult when dealing with videos, having numerous frames, a range of luminance, contrast, texture, color and a varying number of moving objects (ranging from low to high speed). The aforementioned difficulties appear also in the selection of video for the implementation and testing of the proposed foreground detection algorithms. In the existing research on foreground detection, researchers have selected simple videos with a limited number of foreground objects and movement with static background. Moreover, the number of frames selected is always very small, [13-15]. On the contrary, this research is conducted on multi-featured videos in order to test the performance of our algorithms for various types of videos.

4.3 EXPERIMENTAL APPROACH

The objective in any foreground detection algorithm is to find areas of the video sequence where motion exists. The next task is to identify sufficiently the mask of the moving object. This second goal is more challenging than the first one.

The translation motion of objects is estimated by a block-matching algorithm, which finds matching blocks in a video sequence. Such algorithms are the cross search, full search algorithm, spiral search, full exhaustive search, three step search, new three step search, simple and effective search, four step search, two dimensional logarithmic search, binary search, orthogonal search, hierarchical search, and diamond search [1, 2].

4.3.1 LIMITATIONS OF THE AFORESAID ALGORITHMS

The limitations of all these motion estimator algorithms are that they do not deal with:

- **Objects rotation** – objects in the scene may rotate which makes objects that are viewed by a camera have a different aspect of the image.
- **Objects change of shape** – objects in a scene may change shape. An example of this may be clouds or human walking.
- **Camera rotation and tilt** – The camera may rotate or tilt. This cannot be modelled as translation motion and thus will incur prediction error from a predictor that only uses

translation.

- **Objects Occlusion** – When one object moves in front of second object the second object will have part of it occluded. When an object rotates then part of the object will go out of view or come into view due to occlusion.
- **Camera positive and negative zoom** – When a camera zooms, this scales the image up or down and brings out or in other part of a scene.
- **Ambient lighting conditions** – When the ambient lighting conditions change, due to a light being switched on/off or the sun going behind clouds, then the luminance of the image changes and cannot be modelled with just translation.
- **Scene cuts** – When there is a scene cut then the image completely changes with no relation between successive images
- **Object or camera absolute motion** – Cameras or objects do not move to the nearest pixel. They might move to a fraction of a pixel.
- **Object Resolution** – Motion is estimated using 8x8 or 16x16 blocks where there may be more than one object represented within each block having a different motion characteristic.

Each algorithm has its own merits and disadvantages but their performance is measured by their accuracy and computation time [3].

4.3.2 ADAPTIVE ROOD PATTERN SEARCH

In the present study, we adopt the Adaptive Rood Pattern Search (ARPS) [2], which is based on the fact that motion in a frame is generally coherent, i.e., if the macro-blocks around a given macro-block moves in a certain direction, this macro-block is highly probable to have a similar motion vector (MV). In the ARPS, each macro-block benefits from the MV of its adjacent left one to guess its own MV.

The ARPS estimates the four endpoints of the four-armed rood pattern of its diamond (Small Diamond Search Pattern (SDSP) or Large Diamond Search Pattern (LDSP)) along with the predicted point (from the neighboring block) of the motion vector (MV) to measure the current block motion tendency as given in the Figure (4.1). At the primary step a minimum SAD (sum of absolute difference) is found and it becomes the center for unit sized rood pattern. The four endpoints of the four-armed rood pattern (in both cases i.e., that of SDSP or LDSP) are then calculated and compared with the SAD to find a new minimum SAD. This is repeated in order to find the minimum SAD at the rood center.

The search pattern has a very important role in searching algorithms and its size has its own significance. A small search pattern is useful primarily for small motion detection and will result in false estimates while probing a large motion vector (MV). In such a case a large search pattern is suitable. Consequently, search pattern size and magnitude of motion vector should be adaptable to the various situations.

In the prediction of accurate MV of the current block the region of support (ROS) and the algorithm to predict the motion vector are very important. The current block motion vector is predicted from the MVs of the ROS, i.e. the neighboring blocks. The other solution is to focus on few but the most important blocks around the current block, above, above right, above left and left blocks as given in the Figure (4.1). The MV of these blocks is used as a reference. Further details can be seen in [2].

The next step is the search pattern, where initially adaptive search is performed and then fixed pattern is chosen for local search. The four search points located at the four vertices as depicted in the figure 4.1 the rood pattern symmetry.

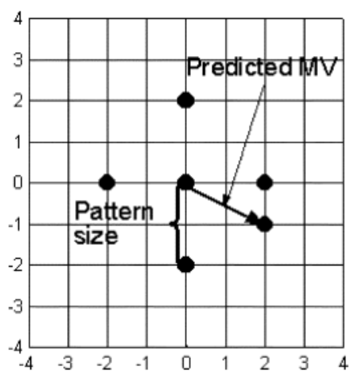


Figure 4.1 Adaptive rood pattern (ARP)

The size of the rood shape is referred to the distance between the center and any vertex point. It has been noticed that the MV distribution in horizontal and vertical directions are higher than that in other directions, [4]. The search can fast detect the motion in the horizontal or vertical directions as these are the most probable motions of cameras. Also, a MV is possible to be decomposed into its horizontal and vertical components. The rood shape can detect the main tendency of motion which is the purpose of the initial search. Summarizing, the adaptive pattern has a rood-shaped pattern (with four vertex points) and a search point, which is specified by the predicted MV.

The initial adaptive rood search leads to the final step of local search, avoiding the extra intermediary searches. There are many searching algorithms that can be used e.g. SDSP in Diamond search (DS), [5]. The advantage of these algorithms over DS is that if the predicted motion vector is at point (0, 0), it does not waste computational time in LDSP, and it rather directly starts using SDSP. Furthermore, if the predicted motion vector is far away from the center, then again ARPS is saving on computations by directly jumping to that vicinity and using SDSP, whereas DS is wasting time doing LDSP, [6].
Based on the above discussion it is clear that the Adaptive Rood Pattern Search algorithm is relatively precise in terms of motion estimation and requires less time in execution [2].

4.3.3 MOTION AND STATIC PART OF THE FRAME

In Equation (4.1) below, M_{part} represents the Motion part and S_{part} is the static part of the foreground object, which is the eventual objective of any foreground detectable algorithm. As a result sufficient mask of only foreground object is obtained.

$$Foreground = M_{part} + S_{part} : \; S_{part} \in M_{part} \qquad (4.1)$$

In the Figure (4.2), $\boldsymbol{u}$ is the universal set that contains all the elements being considered in a particular image.

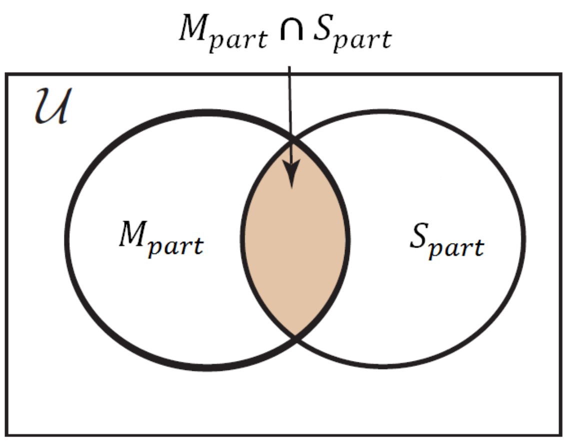


Figure 4.2 Static and motion part of a frame

The Foreground area expressed Equation (4.2), can be easily detectible by our motion estimation technique, with few miscalculated or over calculated areas of motion that will be assumed to be noise. To a greater extent this noise can be reduced by using certain morphological operations.

$$M_{part} \cap S_{part} = \{x : x \in M_{part} \text{ and } x \in S_{part}\} \tag{4.2}$$

Equation (4.3) represents a perfect foreground segmentation, which is a challenging task, and which ultimately covers the full mask of foreground object. For the solution of Equation (4.3), we determine the minima and maxima of the foreground object. The minima and maxima of the foreground object can be determined by morphological operation of opening-and-closing by reconstruction. By minima and maxima of the foreground object we mean the area inside the foreground where the values of intensities are low and high respectively. However this does not cover the background area.

$$Perfect_{foreground} = \{S_{part} \cup (M_{part} \cap S_{part})\} \tag{4.3}$$

4.3.4 Role of Morphological operations

Morphological operations are applied on binary images to eliminate noise or irrelevant detail. In general, dilation expands, while erosion shrinks the pixel areas with the defined radii or structuring element in the given image respectively. Mathematically, dilation of a binary image α by factor β is defined as in Equation (4.4).

The process of dilation enlarges/expands a region by turning foreground pixels that were originally background [23, 24].

$$\alpha \oplus \beta = \{x | (\beta')_x \cap \alpha \neq \emptyset\} \tag{4.4}$$

OR

$$\alpha \oplus \beta = \bigcup_{b \in \beta} \beta_b$$

Dilation has the effect of increasing the size of an object. Erosion of the image α by a factor β is defined mathematically as in Equation (4.5), where α is the image and β is the structuring element and $\boldsymbol{\alpha^c}$ is the complement of $\boldsymbol{\alpha}$.

Erosion removes or shrinks pixels from an image or, equivalently, turns background pixels that were originally foreground [23, 24].

$$\alpha \ominus \beta = \{x | (\beta)_x \cap \alpha^c \neq \emptyset\} \tag{4.5}$$

Closing process is an operation of dilation followed by erosion. This operation closes up the narrow gap between portions of a feature or in other words missing pixels within a feature, by filling in places where isolated pixels were classified as background [23, 24]. Mathematically, the closing of image α by structuring element β is in Equation (4.6).

$$\alpha \cdot \beta = (\alpha \oplus \beta) \ominus \beta \tag{4.6}$$

On the other hand, opening is the reverse operation of closing i.e., erosion followed by dilation This operation opens up spaces between just-touching features, and hence is used for removing noisy pixels from binary images as a common tool. Mathematically opening of a binary image α by structuring element β can be defined as in Equation (4.7).

$$\alpha \circ \beta = (\alpha \ominus \beta) \oplus \beta \tag{4.6}$$

The proposed work first computes motion estimation and then the minima and maxima of the foreground object in the video sequence are determined frame by frame. The motion estimation process is block based, whereas the second noise removal process is pixel based. The objective for both processes is to compensate for the missing areas of foreground object. Noise is removed from the original frames using MO, opening-and-closing-by-reconstruction.

In order to obtain pixel based foreground, regional minima and maxima were used. For this purpose, MOs are applied to the segmented image for different intensity values, where the lowest and the highest intensities are used to determine the foreground maxima and minima respectively inside each frame. This maxima and minima inside the foreground object are those objects which are nearer to the camera. Both minima and maxima are added to obtain a sufficient mask of the object in Figure 4.2(b) and (c). The resultant binary mask is combined by an OR logical operator with block-based motion estimation mask to generate the final binary mask as shown in the Figure 4.2(d).

Figures 4.3 (a) demonstrate the block-matching estimation result with miss and over-calculated blocks. Figure 4.3 (b) and (c) are the foreground object minima, maxima masks obtained after the opening-and-closing-by-reconstruction process. To obtain the full mask of the foreground objects all three previous results are added using logical OR operator as shown in the Figure 4.3 (d).

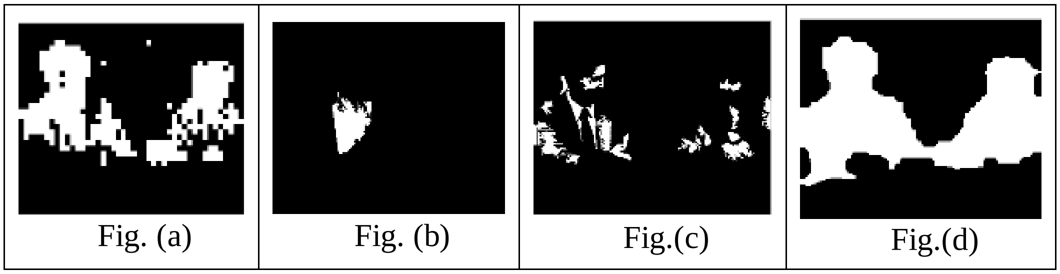


Figure 4.3 Motion estimation, minima, maxima and full foreground mask

4.3.5 NOISE REMOVAL

In Figure 4.4(a), the result of the block-based motion estimation on the video sequence obtains segmentation of the foreground objects from background with a large amount of noise. For this purpose, a couple of MOs such as clean, bridge, dilation and erosion are applied to remove the isolated pixels, bridge them if unconnected, expand, and shrink pixels respectively. As a result, a sufficiently noiseless block-based motion estimated foreground is obtained as shown in the Figure 4.4(b) as compared to Figure 4.3(a), but still with some missing areas of foreground objects.

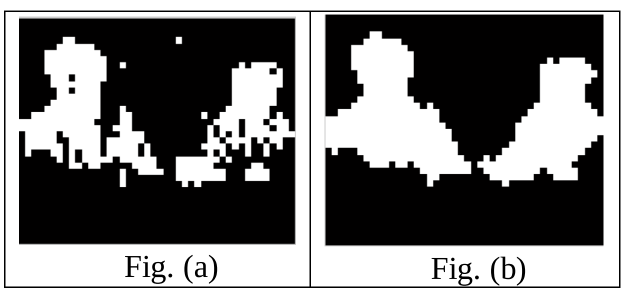


Figure 4.4 Motion estimation and noise removal

Figure 4.5, depicts the overall layout of the proposed algorithm, where both motion estimation and opening-and-closing-by-reconstruction operations are applied on the same frames simultaneously. A sufficient mask of the foreground is eventually obtained for the frames under observation as shown in Figure 4.3(d).

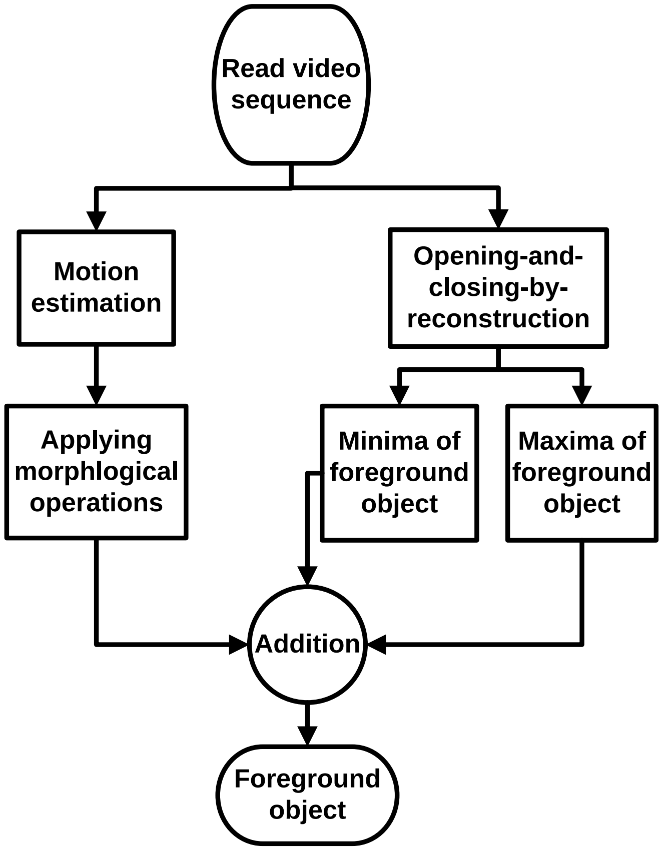


Figure 4.5 Motion estimation and noise removal

4.3.6 Algorithm to find maxima of the foreground object:

Step 1: Define structuring element (β),

Step 2: Apply MO opening on (α)

Step 3: Apply MO closing on the resultant of step 2

Step 4: MO Reconstruct results from step 2 and 3

Step 5: Apply closing operation on resultant of step 2

Step 6: Dilate reconstructed resultant from step 4

Step 7: Reconstruct complemented results from step 4 and 6

Step 8: Complement resultant of step 7

Step 9: Apply regional maxima operation on step 8

4.3.7 ALGORITHM TO FIND MINIMA OF THE FOREGROUND OBJECT:

Figure (4.6) demonstrates the step by step algorithm for computing the minima of foreground object. Finally minima result is added with maxima of the same frame using OR logical operator to obtain full mask of the foreground object.

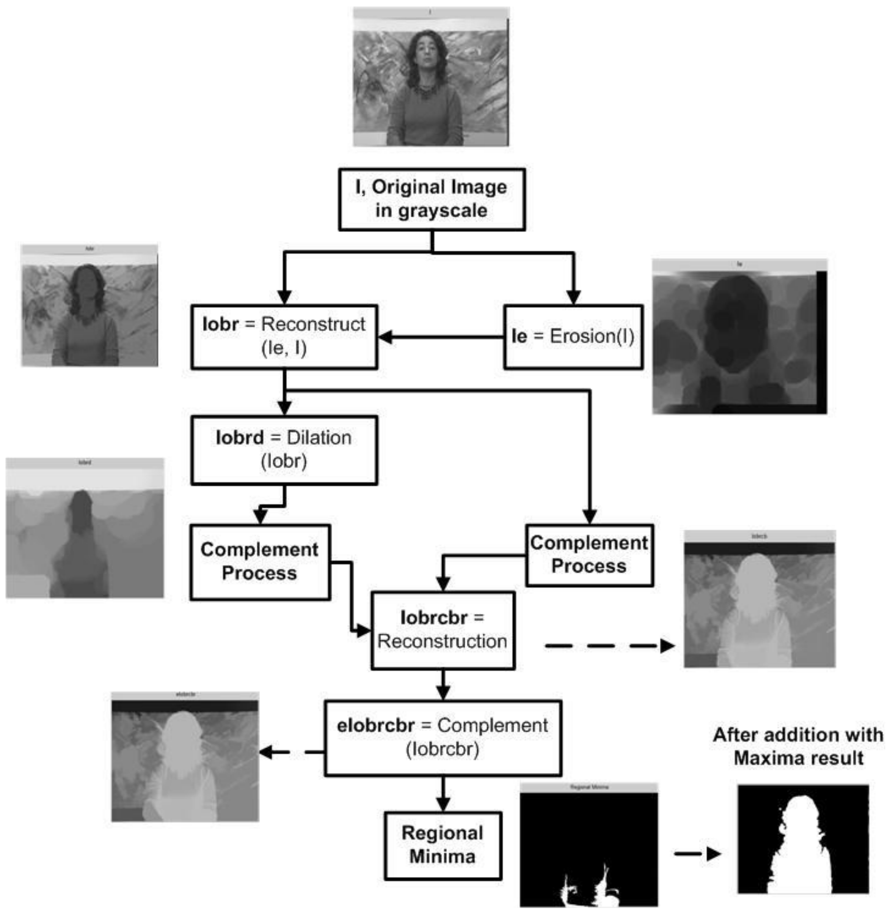


Figure 4.6 Algorithm for Minima extraction and addition with maxima

4.3.8 SUBJECTIVE RESULTS

Figure (4.7) are the original frames of the video sequence, Figure (4.8) shows the ground truth for respective frames and Figure (4.9) to Figure (4.12) demonstrate respective frames' foreground detection results by various state of the art algorithms [9-12].

Results of the proposed algorithm for foreground detection are given in the Figure (4.13).

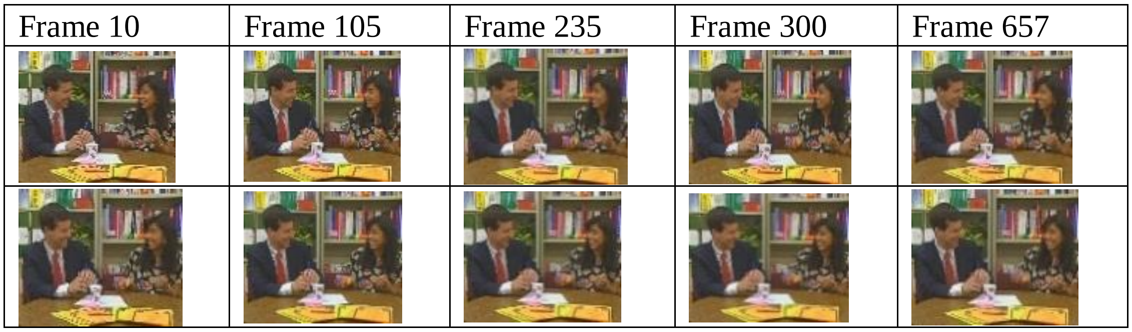


Figure 4.7 Original video sequence

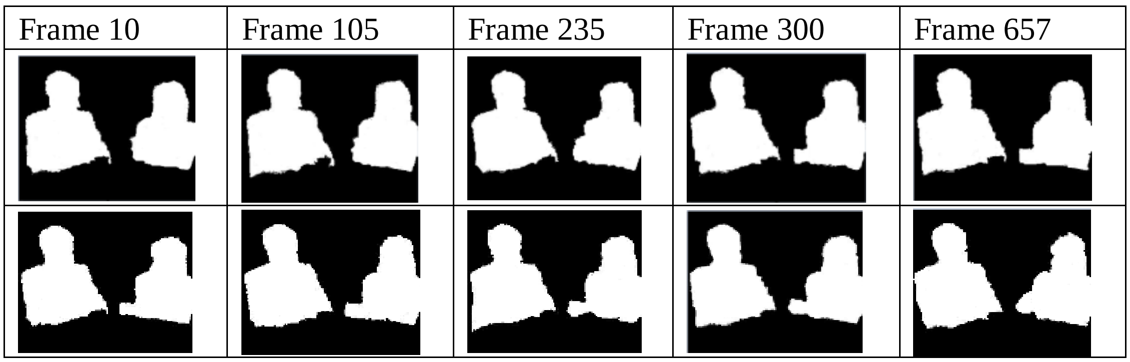


Figure 4.8 Ground truth

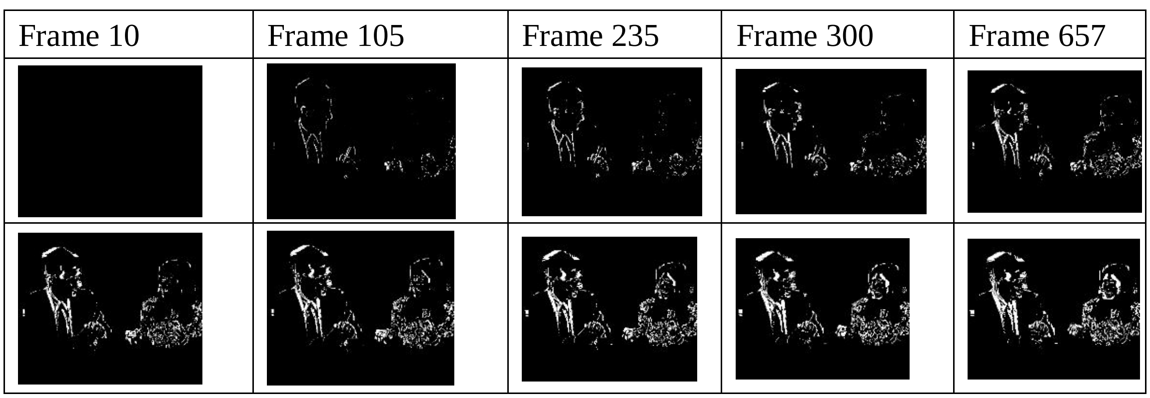


Figure 4.9 Mixture of Gaussian [9]

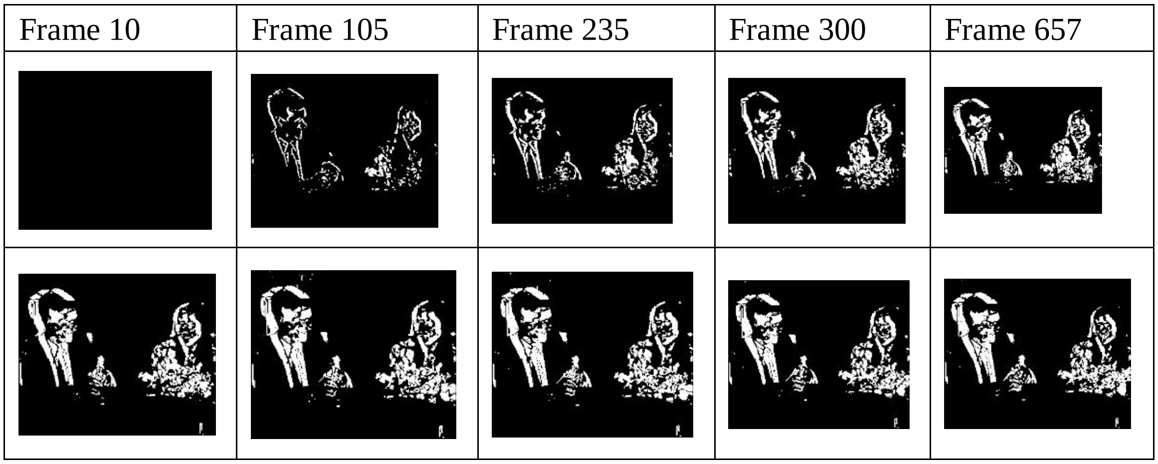


Figure 4.10 SGMR [11]

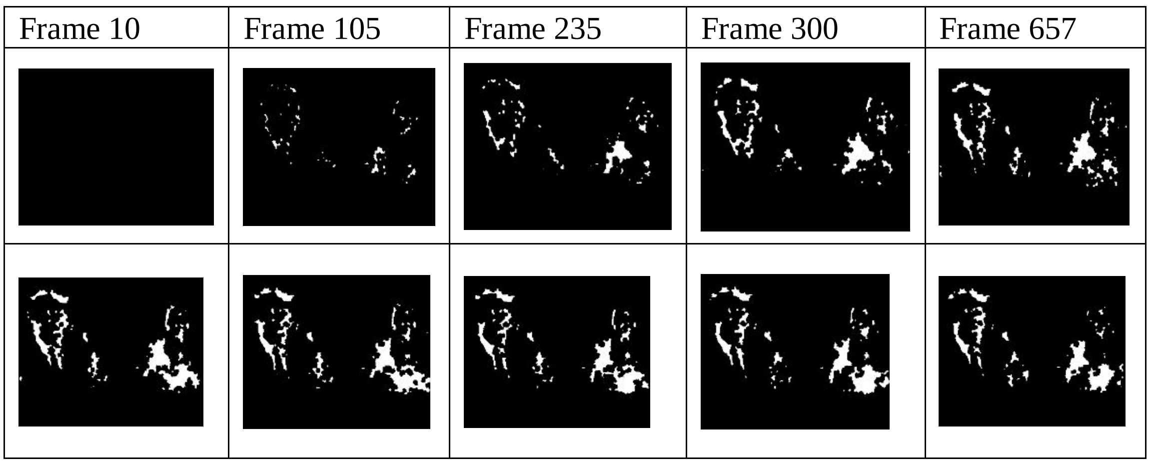


Figure 4.11 Soo Wan Kim algorithm [12]

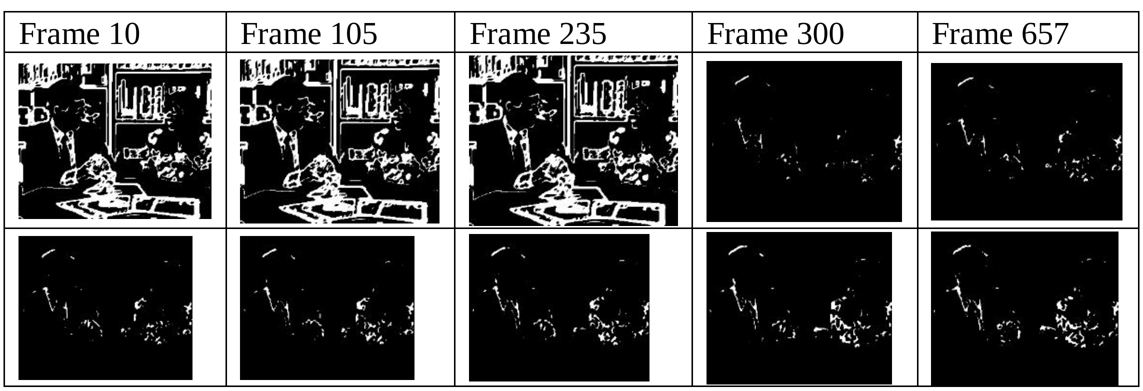


Figure 4.12 Optical flow [10]

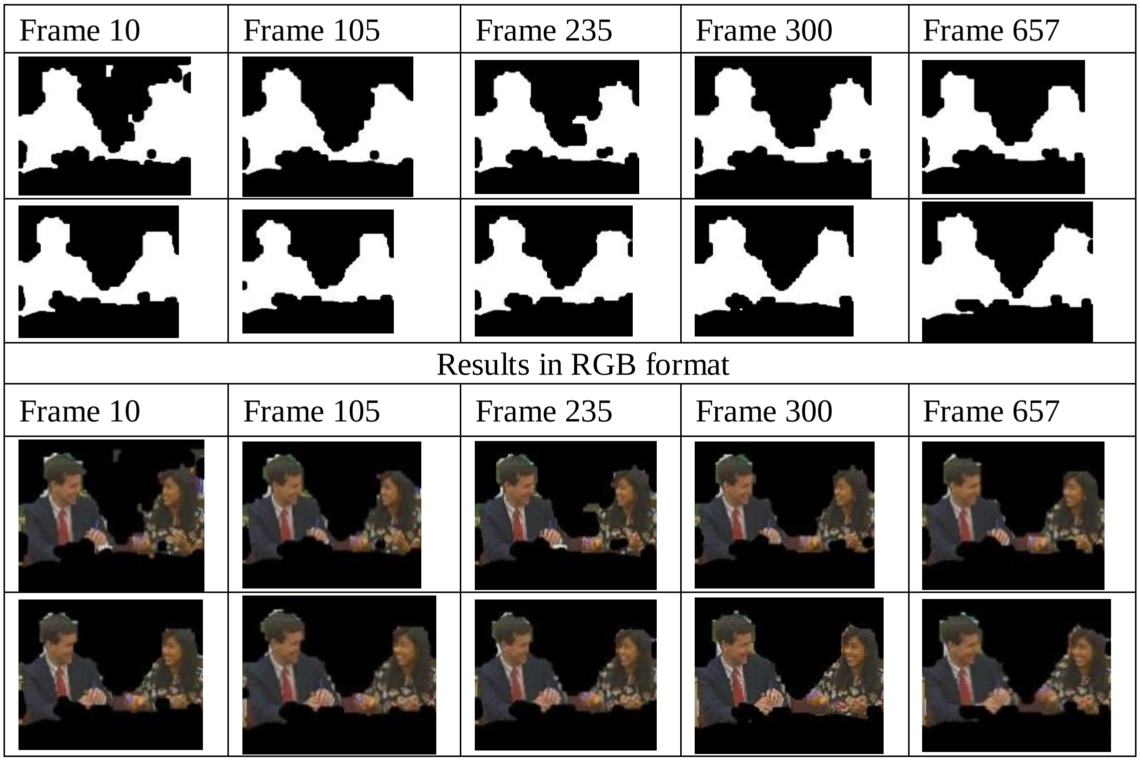


Figure 4.13 Proposed algorthm

4.4 Performance Measures and Results Comparison

There are 11 different performance measurements that were used: precision, recall, F-score, specificity, area under the curve, BER%, accuracy, geometric mean of sensitivity and specificity, similarity and false positive rate. With the help of these measures we will also compare our results with well-established algorithms such as: optical flow [10], Soo Wan Kim approach [12], Mixture of Gaussian (MoG) [9], and the SGM-R algorithm, [11].

4.4.1 Precision

Our precision value is 93.60% as shown in Table (4.1), meaning that we have been able to identify more of the ground truth (intended region foreground) than other techniques, while the ideal value of precision is 100.

The second highest value was obtained by the SGM-R algorithm, which is 73.51%, while the Optical Flow method performs poorly with a value of only 65.75%.

4.4.2 RECALL OR SENSITIVITY OR TRUE POSITIVE RATE (TPR)

As shown in Figure (4.14) and in Table (4.1), there was as much false identification of regions with the proposed method as with the other techniques. The ideal value of Recall is 100. The proposed algorithm has achieved 93.44%. The overall highest value was obtained by the Soo Wan Kim algorithm, which is 97.86%, while the Optical Flow method performs worse with a value of 90.81%.

4.4.3 F-SCORE OF PRECISION AND RECALL

The proposed algorithm has achieved 93.46%, which is the highest value among the other four algorithms. The second highest value was obtained by the SGM-R algorithm, which is 82.65%, while the Optical Flow method performs worse with a value of 75.88%.

4.4.4 SPECIFICITY OR TRUE NEGATIVE RATE

The ideal value of specificity is 100%, and the proposed algorithm has achieved 88.23%, which is the highest value among the other four algorithms.

The second highest value was obtained by the SGM-R algorithm, which is 39.24%, while the Optical Flow method performs worse with a value of 17.68%.

4.4.5 BALANCE CLASSIFICATION RATE OR AREA UNDER THE CURVE

The proposed algorithm has achieved 90.84% which is the highest value among the other four algorithms. The second highest value was obtained by the SGM-R algorithm which is 66.84%, while the Optical Flow method performs worse with a value of 54.25%.

4.4.6 GEOMETRIC MEAN OF SENSITIVITY AND SPECIFICITY

The proposed algorithm has achieved 90.65% which is the highest value among the other four algorithms. The second highest value was obtained by SGM-R which is 60.67%, while Optical Flow performs worse with a value of 38.79%.

4.4.7 F-SCORE OF SENSITIVITY AND SPECIFICITY

The proposed algorithm has achieved 90.48%. The second highest value was obtained by SGM-R, which is 55.16%, while Optical Flow performs worse with a value of 28.35%.

4.4.8 %Balance Error Rate

The ideal value of %Balance Error Rate is 0, and the proposed algorithm has achieved 9.16% which is the best value. The second best value was obtained by SGM-R, which is 33.16%, while Optical Flow performs poorly with the value of 45.75%.

4.4.9 Similarity

The proposed algorithm has achieved 87.78% which is the highest value. The second highest value was obtained by SGM-R which is 70.44%, while Optical Flow performs poorly with the value of 61.91%.

4.4.10 Accuracy

The proposed algorithm has achieved 91.58% which is the highest value from other four algorithms. The second highest value was obtained by SGM-R, which is 74.59%, while Optical Flow performs poorly with the value of 64.51%.

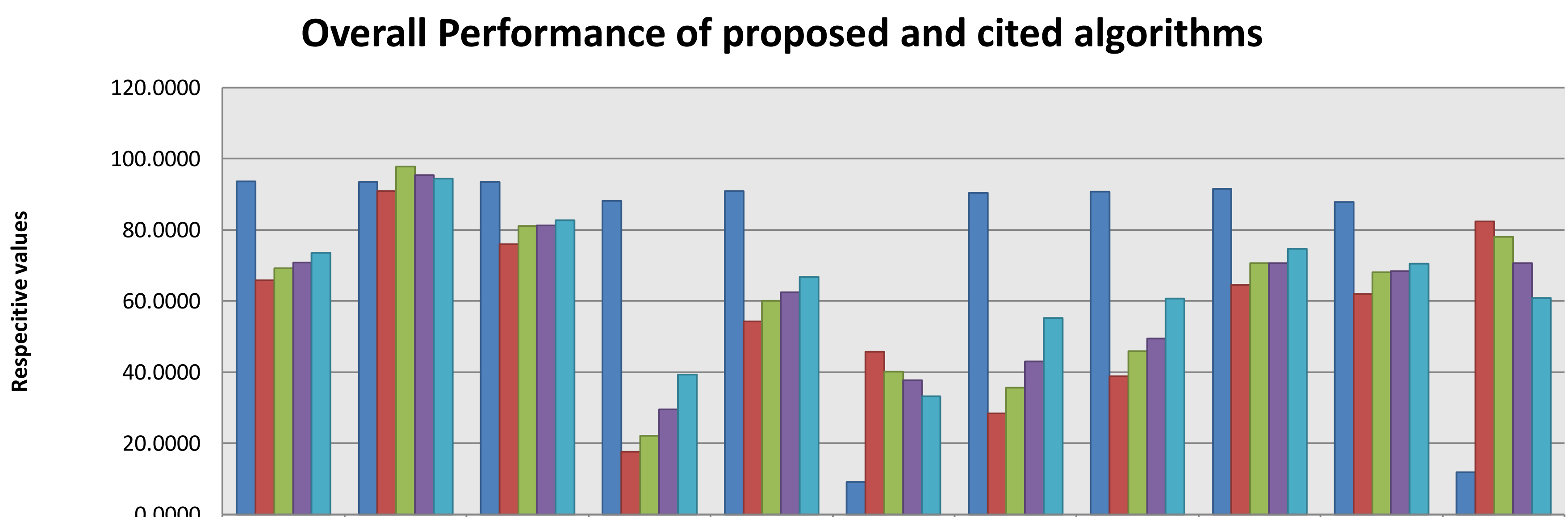
4.4.11 False Positive Rate

The proposed algorithm has achieved 11.76% which is the best value from other four algorithms. The second highest value is that of SGM-R which is 60.76%, while Optical Flow performs poorly with the value of 82.32%.

Table 4.1 Rank of proposed algorithm with the well-established algorithms

Results												
Algorithm No	Ideal or Perfect match value	100	100	100	100	100	0.0	100	100	100	100	0.0
	Frames	Precision	Recall or Sensitivity or True Positive Rate	%F-score of Precision and Recall	Specificity	AUC/BCR (Balanced Classification Rate)	BER (%)	%F-score of Sensitivity and Specificity	Geometric mean of sensitivity and specificity	Accuracy: mean of sensitivity and specificity	Similarity	False Positive Rate
1	Proposed algorithm	**93.5979**	93.4452	**93.4652**	**88.2348**	**90.8399**	**9.1601**	**90.4771**	**90.6552**	**91.5770**	**87.7849**	**11.7652**
2	Optical Flow	65.7502	90.8131	75.8838	17.6818	54.2474	45.7526	28.3530	38.7906	64.5063	61.9064	82.3182
3	Soo Wan Kim algo	69.1276	**97.8641**	81.0086	22.0438	59.9540	40.0460	35.6124	45.9462	70.5913	68.0895	77.9562
4	MoG	70.8468	95.3105	81.1824	29.4351	62.3728	37.6273	43.0487	49.5021	70.6347	68.3454	70.5649
5	SGM-R	73.5124	94.4399	82.6509	39.2389	66.8393	33.1607	55.1559	60.6727	74.5871	70.4394	60.7611
Diff between No 1st & 5th *		20.0855	-4.4189	10.8143	48.9959	24.0006	24.0006	35.3212	29.9825	16.9899	17.3455	48.9959

* In case of Recall the difference is between algorithm 1st and 3rd as our obtained value is lesser than the Soo Wan Kim approach



	Precision	Recall or Sensitivity or True Positive Rate	%F-score of Precision and Recall	Specificity	AUC/BCR (Balanced Classification Rate)	BER (%)	%F-score of Sensitivity and Specificity	Geometric mean of sensitivity and specificity	Accuracy: mean of sensitivity and specificity	Similarity	False Positive Rate
Our Approach	93.5979	93.4452	93.4652	88.2348	90.8399	9.1601	90.4771	90.6552	91.5770	87.7849	11.7652
Optical Flow	65.7502	90.8131	75.8838	17.6818	54.2474	45.7526	28.3530	38.7906	64.5063	61.9064	82.3182
Soo Wan Kim algo	69.1276	97.8641	81.0086	22.0438	59.9540	40.0460	35.6124	45.9462	70.5913	68.0895	77.9562
MoG	70.8468	95.3105	81.1824	29.4351	62.3728	37.6273	43.0487	49.5021	70.6347	68.3454	70.5649
SGM-R	73.5124	94.4399	82.6509	39.2389	66.8393	33.1607	55.1559	60.6727	74.5871	70.4394	60.7611

Figure 4.14 Overall Performances of proposed and cited algorithms

4.5 TECHNICAL EVALUATION

One of the main reasons of this big difference in results is that apart from optical flow algorithm, all other methods use background subtraction and requires a reference image, which is free of foreground object(s). Furthermore in real world videos like the one used in the proposed algorithm it is not possible to have reference image in advance which is free from the foreground.

Soo Wan Kim, MoG and SGMR use Mixture of Gaussian, which are among most recent methods that have been proposed for foreground detection. These methods produce good results in outdoor scenes. In Mixture of Gaussian, the colours of the background objects' pixels are represented by multiple Gaussian distributions. Many researchers have reported that more than two Gaussians can badly degrade the foreground object extraction [16, 17]. The main disadvantage of Mixture of Gaussian algorithm is that it is a computationally complex method and the fact that its variables require careful setting. Thus, the method requires more time in processing. Also, Mixture of Gaussian is very sensitive to sudden changes in global illumination and thus can sometimes produce inaccurate results for this reason. Consequently, when the scene is still for a long time, a rapid change in global illumination may turn the whole frame into foreground [17, 18].

The comparison of results is shown in the Table (4.3) and Figure (4.14). It is obvious that the proposed algorithm clearly outperforms the other four methods. SGM-R is the second best approach. MoG being the quite similar technique to SGM-R was found the third best method, while, the Soo Wan Kim algorithm was found the fourth best algorithm, based on performance measure results. Overall, the performance of the Optical Flow technique was found to be non-satisfactory.

The recall value of the proposed method is lower than Soo Wan Kim algorithm by 4.42%. The recall or true positive rate (*trp*) and precision quantify how well an algorithm matches the ground truth [19, 20], but the proposed algorithm outperforms in precision and %F-score of precision and recall over the rest of the four methods by 20.09% and 10.81%, respectively. It is also important to know that only recall is not sufficient to compare different methods and is generally used in conjunction with precision, that gives the percentage of detected true positive areas, as compared to the total number of items detected [21]. It is clearly shown from the results obtained, that the proposed algorithm performs much better than the second best algorithm SGM-R, on average by 24.74%.

The proposed algorithm is very simple with low complexity. During execution time of the proposed algorithm it was found satisfactory and can be adopted for real world applications. The accuracy is higher because of simultaneous execution of two processes: block motion estimation and calculating minima and maxima inside the foreground object. The reason for lesser time of execution is that in the proposed algorithm block motion estimation was adopted rather than pixel motion estimation procedure which requires higher time of execution.

4.6 CONCLUSION

This chapter presents a simple and effective algorithm to obtain sufficient precise foreground from background using motion estimation, maxima and minima inside the foreground object. The previous works [7, 22, 9-15] on foreground detection shows that our final result has produced better foreground mask based in terms of quantitatively and qualitatively. For quick and accurate execution of block motion estimation we have used Adaptive Rood Pattern Search algorithm. In order to obtain precise mask of the foreground we used opening-and-closing operation. From the performance measures it is shown that our algorithm is relatively more accurate in terms of precision, %F-score of precision, recall, sensitivity, specificity, area under the curve, accuracy and similarity.

4.7 REFERENCES

1. Barjatya, A. (2004) "Block matching algorithms for motion estimation", *IEEE Transactions Evolution Computation,* vol. 8, no. 3, pp. 225-239.

2. Nie, Y. and Ma, K. (2002) "Adaptive rood pattern search for fast block-matching motion estimation", *Image Processing, IEEE Transactions on,* vol. 11, no. 12, pp. 1442-1449.

3. John Cosmas “ee2605: Multimedia Analysis and Content Delivery” Brunel University, Computer Systems Engineering course notes, October 2013.

4. Zhu, S. and Ma, K. (1998) "A new star search algorithm for fast block-matching motion estimation", *Proc. Workshop on Very Low Bitrate Coding (VLBV),* pp. 173.

5. Zhu, S. and Ma, K. (2000) "A new diamond search algorithm for fast block-matching motion estimation", *Image Processing, IEEE Transactions on,* vol. 9, no. 2, pp. 287-290.

6. Barjatya, A. (2004) "Block matching algorithms for motion estimation", *IEEE Transactions Evolution Computation,* vol. 8, no. 3, pp. 225-239.

7. Kelly, P., Ó Conaire, C., Monaghan, D., Kuklyte, J., Connaghan, D., Pérez-Moneo Agapito, J.D. and Daras, P. (2010) "Performance analysis and visualisation in tennis using a low-cost camera network" .

8. KaewTraKulPong, P. and Bowden, R. (2002) "An improved adaptive background mixture model for real-time tracking with shadow detection" in *Video-Based Surveillance Systems* Springer, pp. 135-144.

9. Stauffer, C. and Grimson, W.E.L. (1999) "Adaptive background mixture models for real-time tracking", *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* vol. 2, pp. 252.

10. Barron, J.L., Fleet, D.J. and Beauchemin, S.S. (1994) "Performance of optical flow techniques", *International journal of computer vision,* vol. 12, no. 1, pp. 43-77.

11. Olson, T. and Brill, F. (1997) "Moving object detection and event recognition algorithms for smart cameras", *Proc. DARPA Image Understanding Workshop,* pp. 205.

12. Kim, S.W., Yun, K., Yi, K.M., Kim, S.J. and Choi, J.Y. (2012) "Detection of moving objects with a moving camera using non-panoramic background model", *Machine Vision and Applications,* pp. 1-14.

13. Kaup, A. and Aach, T. (1994) "Efficient prediction of uncovered background in interframe coding using spatial extrapolation", *Acoustics, Speech, and Signal*

*Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*IEEE, , pp. V/501.

14. Patwardhan, K.A., Sapiro, G. and Morellas, V. (2008) "Robust foreground detection in video using pixel layers", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 30, no. 4, pp. 746-751.

15. Elgammal, A., Duraiswami, R., Harwood, D. and Davis, L.S. (2002) "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance", *Proceedings of the IEEE,* vol. 90, no. 7, pp. 1151-1163.

16. Gao, X., Boult, T.E., Coetzee, F. and Ramesh, V. (2000) "Error analysis of background adaption", *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on* IEEE, pp. 503.

17. Sen-Ching, S.C. and Kamath, C. (2004) "Robust techniques for background subtraction in urban traffic video", *Electronic Imaging 2004* International Society for Optics and Photonics, pp. 881.

18. Bouwmans, T., El Baf, F. and Vachon, B. (2008) "Background modeling using mixture of gaussians for foreground detection-a survey", *Recent Patents on Computer Science,* vol. 1, no. 3, pp. 219-237.

19. Sen-Ching, S.C. and Kamath, C. (2004) "Robust techniques for background subtraction in urban traffic video", *Electronic Imaging 2004*International Society for Optics and Photonics, pp. 881.

20. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. and Rosenberger, C. (2008) "Review and evaluation of commonly-implemented background subtraction algorithms", *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* IEEE, pp. 1.

21. Maddalena, L. and Petrosino, A. (2008) "A self-organizing approach to background subtraction for visual surveillance applications", *Image Processing, IEEE Transactions on,* vol. 17, no. 7, pp. 1168-1177.

22. Nawaz, M., Cosmas, J., Adnan, A. and Ali, M. (2011) "Inter-intra frame segmentation using colour and motion for region of interest coding of video", *Broadband Multimedia Systems and Broadcasting (BMSB), 2011 IEEE International Symposium on* IEEE, pp. 1.

23. Shapiro, L.G. and Stockman, G.C. (2001) "Computer vision”, pp. 279–325.

24. Martínez-Martín, E. and del Pobil, Á.P. (2012) “Computer Vision Concepts” in *Robust Motion Detection in Real-Life Scenarios* Springer, pp. 99-108.

Chapter 5
FOREGROUND DETECTION USING BACKGROUND SUBTRACTION WITH HISTOGRAM

5.1 INTRODUCTION

One of the core challenges for the background subtraction algorithm is how to setup its threshold value precisely at run time, in order to obtain the foreground detection more precisely. In proposed algorithm the key feature of any foreground detection algorithm is motion, which is used to obtain a histogram of the foreground regions in order to subsequently detect those moving regions more extensively. However getting the threshold value from the histogram of foreground regions detected by the original motion algorithm is not possible due to large number of peaks and valleys. In order to facilitate the detection of threshold values motion histogram smoothing is used in a systematic way to help obtain the threshold values.

In the proposed algorithm the main focus is to obtain better estimation of the threshold value of foreground regions by obtaining it dynamically from the histogram at run time. If the proposed algorithm is used intelligently by combining motion magnitude and motion direction it could potentially distinguish more accurately between background and foreground in the presence of moving camera. But the motion direction utilization is not implemented in current proposed algorithm and dedicated to future work under the topic (6.3.2) in the chapter-6 of thesis.

In the experiments so far, the proposed algorithm did not encounter any *ghosting* or *foreground aperture* problem in the videos ranging from slow to normal and fast.

5.2 BACKGROUND SUBTRACTION

Motion plays a very important role in any foreground detection algorithm. The level of preciseness of motion detection directly affects the efficiency of performance measures and subjective quality of detected foreground. Motion detection approaches are different for static and dynamic backgrounds. The algorithm presented in this chapter is more suitable for static backgrounds but can be extended to dynamic background by using motion vectors along with motion magnitude which is dedicated for future work.

Any pixels for which motion is detected using motion estimation are considered as a part of the foreground object. Based on temporal information between frames implied in the difference, two different approaches are recommended: *background (-frame) subtraction* and *techniques based on temporally adjacent frames*.

Background (-frame) subtraction uses the first frame as the reference image/frame and if there is significant motion (in pixels) in the subsequent frame, then it shall be considered as part of the foreground object as given in the Equation (5.1). This is a very simple and useful solution that can be used in the following two situations:

Ideal situation: This occurs when there is no foreground object in the reference image so the result of detecting the foreground using motion produces good results as shown in Figure (5.1), since there is no motion for example in tree or its branches of the background scene.

$$
\begin{gathered}
Reference_{image} - Frame_1 = Foreground_1 \\
Reference_{image} - Frame_2 = Foreground_2 \\
Reference_{image} - Frame_3 = Foreground_3 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots = \ldots\ldots\ldots\ldots\ldots\ldots \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots = \ldots\ldots\ldots\ldots\ldots\ldots \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots = \ldots\ldots\ldots\ldots\ldots\ldots \\
Reference_{image} - Frame_n = Foreground_n
\end{gathered}
\tag{5.1}
$$

However in the *real world general situations* there is a possibility that there is a foreground object in the reference image. When the current image appears with a new foreground object, the new and old foreground objects are both detected i.e., the foreground object in the reference image appears and this is also known as *ghosting*.

Event	Reference frame	Current frame	Background subtraction result
Ideal			 White mask is that of current frame
General			 Yellow and blue foreground masks are of current and refrence images respectively

Figure 5.1 Background Subtraction

Time-differencing is a technique *based on temporally adjacent frames,* which suggests that a pixel is in motion if and only if its intensity has significantly changed between the previous and the current frame. In Equation (5.2), x is the pixel position intensity that belongs to a moving object, I_t represents current frame at time *t*, I_{t-1} is the previous frame at time *t-1*, *and* τ is the threshold value.

$$|I_t(x) - I_{t-1}(x)| < \tau \tag{5.2}$$

There is no doubt that this is an easy approach but it will only work if the object speed and frame rate are known in advance. Otherwise this technique leads to two types of problems [1]: *foreground aperture* and *ghosting*.

Here the object speed is a relative term (i.e., slow and fast motion) and threshold value is adjusted manually by hit and trial. Below is an example of MATLAB code segment.

```
for f = SF: EF   %Frame from SF  To  EF
    B = A;
    img = read(MV,f);
    A  = double(rgb2gray(img));
```

```
%Threshold value
thresh=11;
fr_diff = abs(A-B);
%fr_diff = abs(imsubtract(A,B));

for j = 1:width
    for k = 1:height
    if (fr_diff(k,j)>thresh)
        fg(k,j) = A(k,j);
    else
                fg(k,j) = 0;
    end
    end
end
```

Where variable `SF` is the starting frame, `EF` is the ending frame and `MV` reads the video sequence. `A` and `B` are the current and previous frames. Whereas `thresh` is the threshold value variable, in this code it was manually adjusted for video sequence Paris by hit and trial as `11`. Variable `fr_diff` is the absolute difference between current and previous frame.

Frame rate or frame frequency is normally expressed as frame per second (fps).

Thus the value of threshold is dependent on both: object speed and frame rate and it is set manually in the equation (5.1).

But in our proposed solution there is no need of such manual setting of threshold value.

Foreground aperture problem is created when the object speed in the scene is very slow or remains static for some time, so it is considered as background rather the foreground.

The solution of this problem is known as the *double-difference* image [2] and the flow diagram of this algorithm is given in the Figure (5.2).

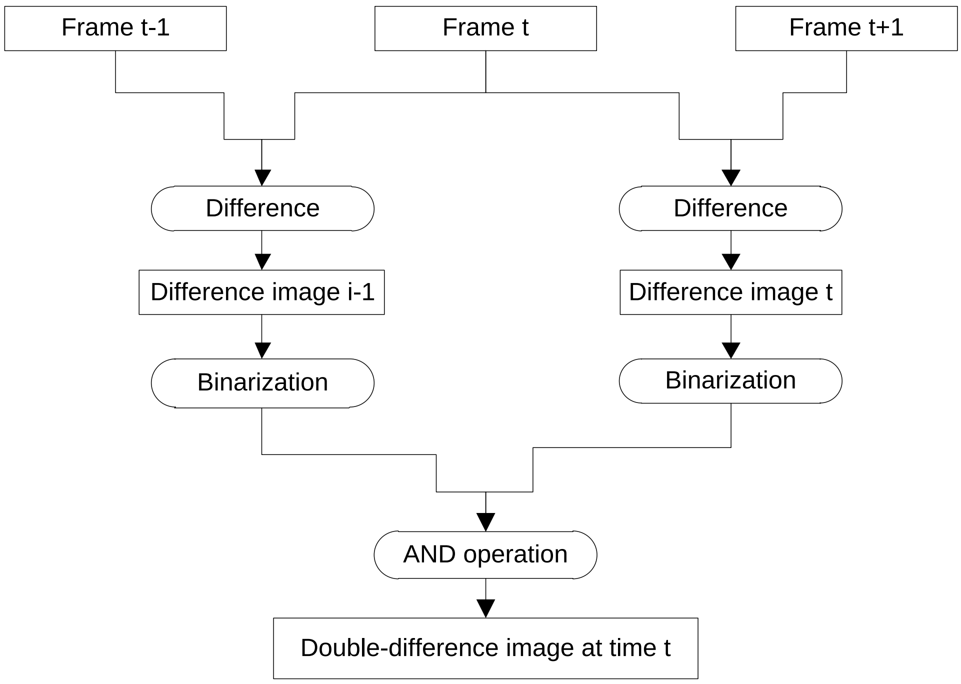


Figure 5.2 Double-Difference Image Algorithm

This approach requires a threshold difference between frame at time *t-1* and *t*, and between frames at *t* and *t+1*. Finally they are combined by a logical AND operator. However, this approach has the drawback that it cannot find the precise position of an object in real time. Similarly, accurate motion detection becomes a problem, if there is not enough texture.

Another algorithm that has been proposed is known as the hybrid algorithm [1] for motion detection. This algorithm is based on *three-frame differencing* methods as depicted in the Figure(5.3), which finds the difference between frames at time t and $t-1$ and the difference at t and $t-2$ in order to determine the regions of reasonable motion and overcome the *ghosting* problem. Adaptive background (*-frame*) subtraction [3] was applied to overcome the issue of the *foreground aperture*. This background update procedure has a few problems and fails [7] when the foreground object begins or ends motion and in case of luminance variation. It produces a problem at variable depth shots (zoom in and out of particular objects) as this is mainly used in outdoor environment shooting with low depth of field images.

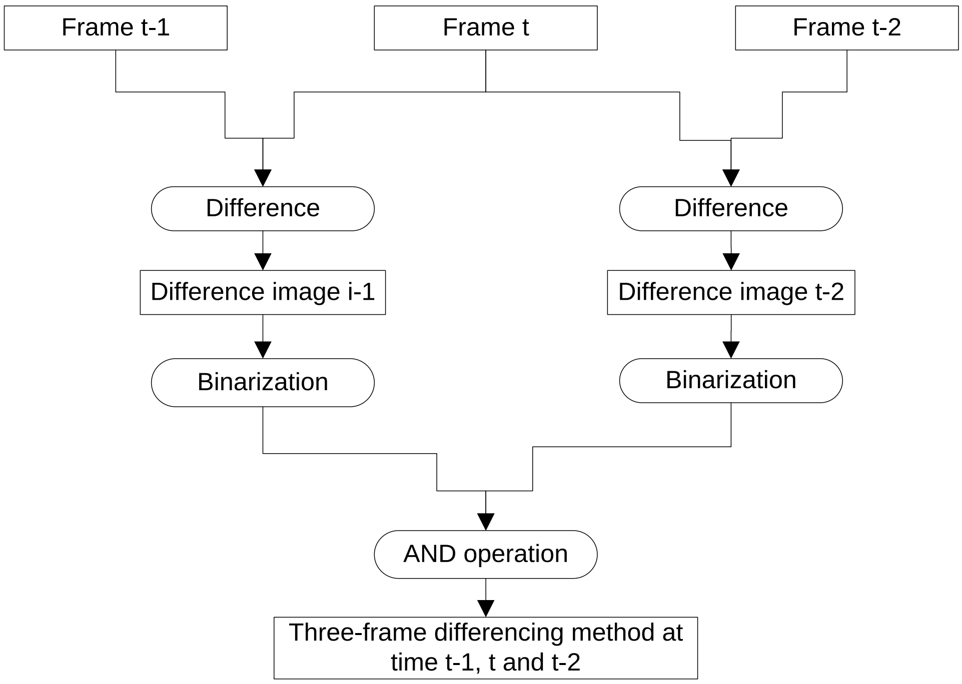


Figure 5.3 three-frame differencing method

To avoid the need of previous background learning, robust pixel foreground classification is introduced, [4] which is a well cited algorithm. This algorithm claims that robust pixel foreground classification is possible without the need of previous background learning. For proper pixel classification joint background subtraction and frame-by-frame differencing method is used and background model is selectively updated according to above classification by [5], based on equation (5.2) , where B_t is background at time *t* , B_{t-1} is background at time *t-1*, F_t is foreground at time *t* and α depend on pixel classification.

$$B_t = (1-\alpha)B_{t-1} + \alpha F_t \tag{5.3}$$

Algorithm 5.1

$if\ \big((|F_t(x) - B_{t-1}(x)| > \tau_b)$ AND $(|F_t(x) - F_{t-1}(x)| > \tau_a)\big)\ then$

Foreground Pixel;

$else\ if\ \big((|F_t(x) - B_{t-1}(x)| > \tau_b)$ AND $(|F_t(x) - F_{t-1}(x)| < \tau_a)\big)\ then$

Collect pixels in blobs;

$if\ (\#Foreground\ Pixels \geq \big(\gamma * (\#Total\ Pixels)\big)\ then$

Foreground Pixel; // *foreground aperture* problem solution

else

Background pixel; // background object suddenly starts moving at time *t*

end if

$else\ if\ \big((|F_t(x) - B_{t-1}(x)| < \tau_b)$ AND $(|F_t(x) - F_{t-1}(x)| > \tau_a)\big)\ then$

Background Pixel; //*ghosting* problem solution

else

$//\ \big((|F_t(x) - B_{t-1}(x)| < \tau_b)$ AND $(|F_t(x) - F_{t-1}(x)| < \tau_a)\big)\ then$

Background Pixel;

end if

The satisfactory performance of this algorithm has been confirmed in [7]. However, this is at the cost of two disadvantages: this method totally fails when the foreground object stops or if its speed is low, and the second difficulty is that of setting the proper threshold value to detect precise foreground in the Algorithm (5.1) [7-8].

This problem can be solved using: clustering, entropy and object attribute based methods. Other two methods that can provide a solution are spatial and local methods. These methods are mathematically complex and time consuming.

The proposed algorithm provides the solution to this problem using a *histogram based method* that detects the histogram peaks and valleys of the smoothed histogram.

The reasons for selecting a *histogram based method* are that it obtains higher value of accuracy, precision, specificity, similarity, and false positive as compared to other methods of foreground detection [10-14]. Furthermore the execution time is relatively less due to the block size (4×4) used in the proposed algorithm.

The issue in original (non-normalized) histogram is depicted in Figure 5.4, which shows the original histogram of the reference image and current frame. Both have so many peaks and valleys that decision making using threshold value is not possible.

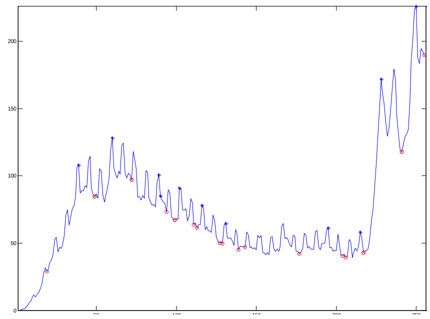
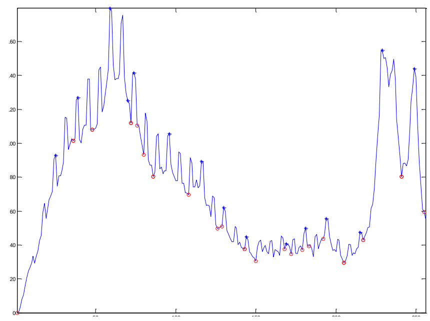
	Frames	Respective non normalized Histogram
Reference Image		
Current Frame		

Figure 5.4 Original Histogram

5.3 Proposed Algorithm explanation

The main idea of this algorithm is to discover the maximum area of motion. This method is easy to understand, requires less computation and produces good results. However, the emphasis is not on the full foreground object mask recovery process as discussed earlier in chapter-4.

5.3.1 Video frames extraction

Video is extracted frame by frame in RGB colour space as given in Equation (5.4) and then converted to grey-level as given in Equation (5.5). Total 256 intensity levels (8-bits) are used to represent the brightness of the image where 0 represents black (dark) and 255 represents white (bright) level. Equation (5.4) is used for this conversion, where f represents frame, M is full movie, n is frame number and movie is N frames of length

$$M = f_1, f_2, f_3, \dots f_{n-1}, f_n, f_{n+1}, \dots\ f_N \qquad (5.4)$$

Where each frame (f_i) is of size 240x180

5.3.2 FRAMES CONVERSION TO GRAY LEVEL

Each frame is then converted into grey level scale using following function

$$G_n = Grey(f_n)$$

Where $Grey(f_n)$ is a function used to convert RGB image into grey level image using following standard equation as defined by (International Telecommunication Union) ITU CCR 601 [16].

.

$$I_{Gray} = 0.299 \times \mathrm{R} + 0.587 \times \mathrm{G} + 0.114 \times \mathrm{B} \tag{5.5}$$

Where ***R***, ***G***, and ***B*** represent Red, Green, and Blue component of the pixels respectively in RGB colour space.

Frame of size 240×180 is used because it maintain the standard video aspect ratio of 4×3 [17] and divisible of 4 by 4 block.

Macro blocks calculation.

$$\frac{frame\ width}{block\ size} = x$$

Therefore

$$x = 60$$

$$\frac{frame\ height}{block\ size} = y$$

Therefore

$$y = 45$$

$$macoblock\ size = x \times y$$

Therefore

$$60 \times 45 = 2700$$

Thus grey level frame is divided into 2700 macro blocks each of size 4×4.

$$G_n = \bigcup_{x=1,y=1}^{x=60,y=45} m_n(x,y) \qquad (5.6)$$

Equation (5.6) means there are total of 60 macro blocks along the width and 45 along the height of the frame and m_n represents macro blocks

$$\bigcap \big(m_n(i,j), \qquad m_n(k,l)\big) = \varnothing \qquad (5.7)$$

For $i \neq k$ *and* $j \neq l$

In the above Equation (5.7), m_n represents macro-blocks and overall equation shows no macro block overlaps with each other.

The vector size is set as 4×4 =16; the reason is that using a smaller sized macro blocks increases the execution time, whilst larger sized blocks decreases the accuracy of motion detection and quality of edges.

5.3.3 CALCULATION OF MOTION VECTORS FOR EACH MACRO BLOCK

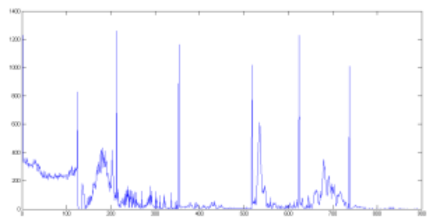
The motion vectors for each macro block are calculated from where magnitude of the motion is used to separate background and foreground.
While with the addition of motion direction (from motion vectors) we can find the camera motion (if it exists) and then on the basis of that handle that situation, we can use another algorithm (such as the one proposed which is described in the chapter-4) to cover foreground object motion only. However this work is dedicated to future work at this stage.

This is the core and the complex step of proposed method that requires maximum of the computation time. Here the motion vector using two consecutive frames is calculated for each block using “Three-step Search (TSS) Algorithm for Block-Matching Motion Estimation Method” [9]. This process can be summarized in Equation (5.8).

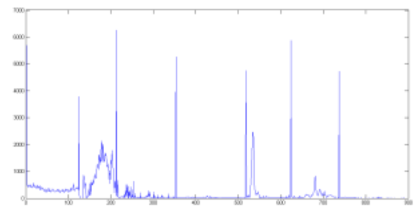
$$v_n(i,j) \leftarrow MV(m_n(\mathrm{i,j}), m_{n+1}(\mathrm{i,j}), \mathrm{s}) \qquad (5.8)$$

Where $v_n(i,j)$ is motion vector for block *i, j* in frames *n* and *n+1*, *s* (search distance) is the sensitivity level used to estimate motion, which is set as 5 in the proposed algorithm. It sets the sensitivity and possible values of motion, for example if it has the value of 2, then the maximum motion will be around 2 and if it is 10, motion vectors will be calculated in 10 values. During testing different values were tried such as 5, 10 and 15 but results were almost the same while 5 is faster in terms of execution time, than the other two values. Furthermore search size 5 was also found suitable for 32 bins.

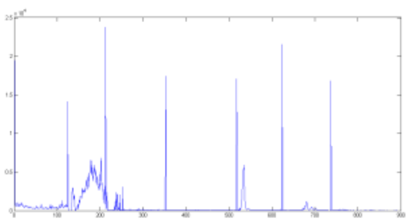
To calculate the motion-vector, different sensitivity values are used from 5 to 15 in equation (5.8) but experimental results show that all these give approximately similar results. Figure (5.5) show results for sensitivity values s = 5, 10 and 15, while last figure (5.5-d) shows a combined graph for the three parameters where resemblance among the three can be seen.



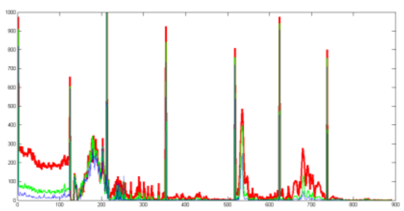
a. Using P = 5



b. Using P = 10



c. Using P = 15



d. Combing 5, 10 and 15

Figure 5.5 Continuation function using P = 5, 10, 15

5.3.4 GETTING MOTION INFORMATION OF PREVIOUS 5 FRAMES

The reason for taking window size of 5 is that, if a large window size is used, it will ignore short motions. The motion information in single (one) frame is not enough to guide us about the next (upcoming) frame motion information. So during the experimental process it was observed that taking the motion information of 5 consecutive frames is a good choice for the prediction of the next frame.

That is why to get sufficient motion information the average on the previous 5 frames was used, while on the other hand if only one previous frame is used this does not have enough information to be utilised effectively.

5.3.5 MOTION VECTOR CONVERSION TO 32 VALUES

The motion vector is converted into 32 values to represent/plot data on x-axis of the histogram, which represents the motion vectors magnitude. For the decision making smooth values of the histogram are considered as shown in the figure (5.7) and figure (5.8).

5.3.6 X-AXIS AND Y-AXIS OF THE HISTOGRAM

The overall histogram of the proposed algorithm is created so that x-axis has motion vectors magnitude (in 32 values) and y-axis has the number of blocks of size (4×4) having motion in that range (total bins are 32), which is used for decision making of a threshold value as shown in the Figure (5.6) to Figure (5.8).
The search distance value is set as 5 because it is faster terms of execution. Therefor the choice of 32 bins was found suitable for search distance value 5.

5.3.7 THRESHOLD

In the proposed algorithm, x-axis of the histogram is divided in two equal portions as given in the Equation (5.9) and Equation (5.10).

1st half represents relatively slow motion and 2nd half represents higher motion.

$$1st\ half = 0 > and\ < 16 \tag{5.9}$$

OR

$$0 < 1st\ half\ < 16$$

$$2nd\ half = 16 \geq and \leq 32 \tag{5.10}$$

OR

$$16 \leq 2nd\ half\ \leq 32$$

5.4 Threshold Values

From histogram there are three different situations as explained below:

5.4.1 Situation-I:

If there is one largest peak and that lies in the 1st bin as shown in the figure (5.6), this shows that maximum numbers of blocks are stationary so it is considered as background.

The largest peak in the 1st bin belongs to stationary part of the frame. It means that most of the blocks are stationary hence this forms stationary background and anything else, if any, is foreground.

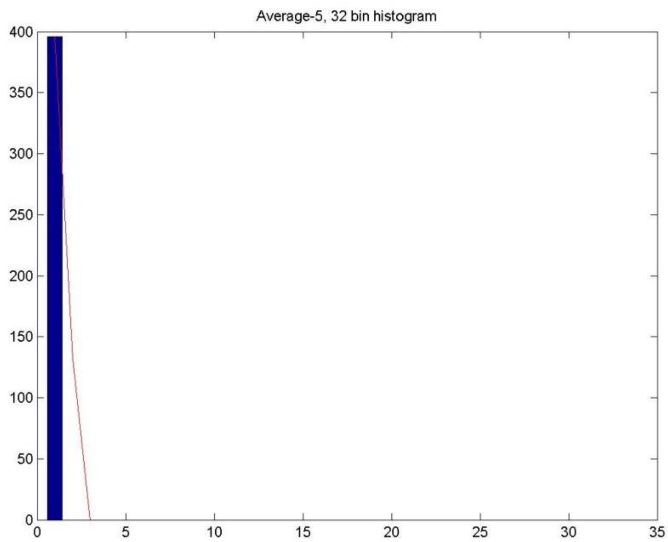


Figure 5.6 Case-I

5.4.2 SITUATION-II:

The second case might be that largest peak is in the 1st half of the histogram as shown in the figure (5.7), which shows relatively the slow motion. In this case, another peak is found away (in the second half) from this first peak to determine the valley between these two peaks for a threshold value. All objects above the threshold is considered as foreground and below this threshold as background, as given in the Equation (5.11) and Equation (5.12).

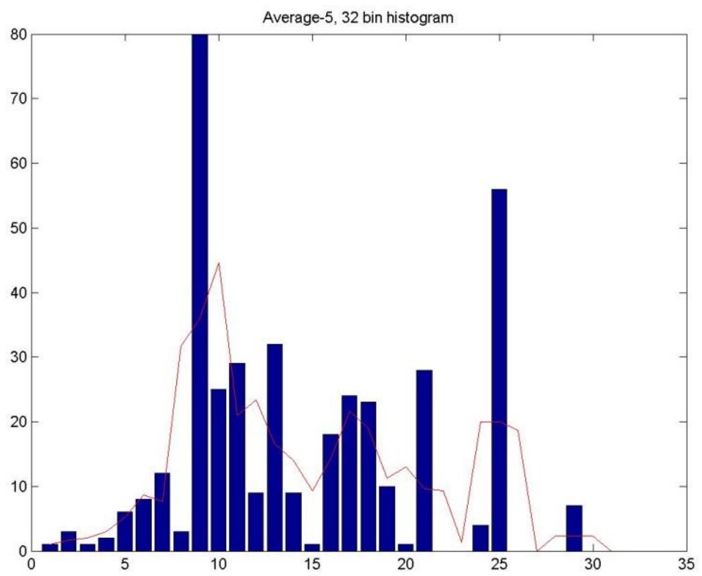


Figure 5.7 Case-II

5.4.3 Situation-III:

When the largest peak is in the 2nd half of the histogram that belongs to the fast motion part of the image as shown in the figure (5.8), which indicates that some major object is moving in the frame which represents the foreground. To determine the threshold another peak is found in the first half of the histogram. A lowest value (valley) in the histogram between the two peaks is the threshold that separates fast moving objects from slow moving or stationary background. This situation occurs when camera is moving very fast or if some large object is moving in the scene.

All objects above the threshold is considered as foreground and below this threshold as background, as given in the Equation (5.11) and Equation (5.12).

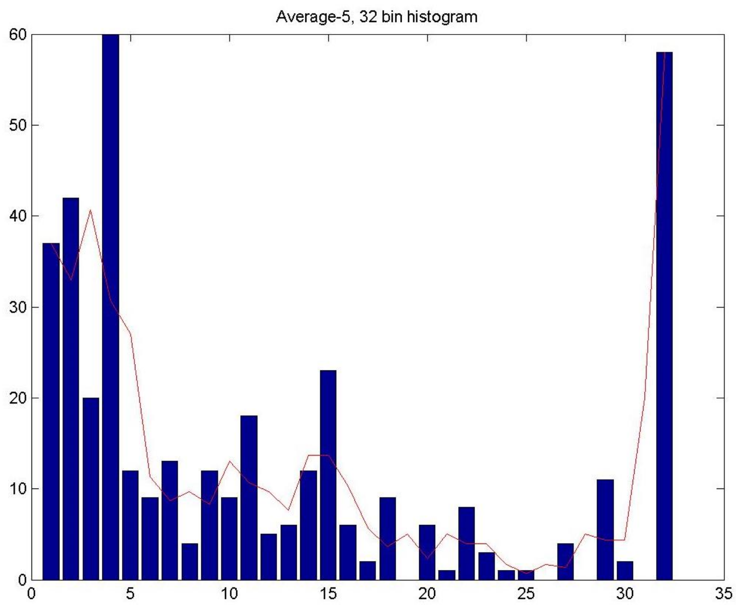


Figure 5.8 Case-III

To summarize the discussion in point 5.4.1 to 5.4.3 to calculate threshold, from histogram of motion values two largest peaks are selected and the valley is calculated, which is located in between these two peaks in the histogram, subject to the condition that if first peak is in first half and lies in first bin, where motion is zero and considered as background. It is worth to mention that at this point the camera is still and background part of the frame is shown. Similarly if camera is in motion, then the first peak of the histogram will not be in the first bin, as the first bin shows that most of the blocks have zero motion, which is not possible if the camera is moving.

From experiments it is observed that if the camera is still and the objects in the foreground are moving then the largest peak of histogram lies in the 1st half of histogram, while if camera is moving very fast or large number of objects are moving then the largest peak of the histogram lies in the 2nd half of the histogram, which represents maximum motion of objects.

The reason for selection of largest peaks from both halves is to obtain a better valley and cover the maximum motion area (one peak in first half and second peak in second half or vice versa for better motion estimation). Furthermore during the searching for the second peak in the range from 255 to 1 on y-axis, this peak represents another major object motion in the image.

The reason for picking a valley as the threshold point, although it points to a minimum value between two peaks is because a minimum value (is unlikely to represent any cluster of a moving object. Furthermore it is known that on both sides there are two peaks so this means there are most likely two major motion of moving objects (as peaks means that many blocks have these motions). This produces segments on both sides of the threshold value so this is the best point for the threshold, as it is the best boundary between two major values.

Valley point having minimum value, is a point dividing the feature vector into two halves which have at-least one peak on each half and so by taking this as a threshold and values less than or equal to threshold belongs to background part while values above than this belong to foreground part of the respective frame, as expressed in the Equation (5.11) and Equation (5.12).

$$Background \leq Threshold \tag{5.11}$$

$$Foreground > Threshold \tag{5.12}$$

Figure (5.9) to figure (5.12), shows the foreground results using four standard video sequences [15] known as the change detection benchmark dataset, along with the respective original frames, ground truth and corresponding results of the proposed algorithm and other five state of the art algorithms[10-14].

In the proposed algorithm results, the black area represents the background and RGB represents detected foreground area. It is very important to note that in the proposed algorithm opening-and-closing by reconstruction technique (already explained in chapter 4, of the thesis) is not applied to cover the precise mask of the foreground object and eradicate and cover the miscalculated background pixels and foreground area. The reason is that the goal of this chapter is to highlight the strength and weakness of the current proposed algorithm only.

5.5 SUBJECTIVE RESULTS OF CITED AND PROPOSED ALGORITHMS

Figure (5.9) to Figure (5.12), shows the subjective results of four video sequences, including indoor and outdoor.

Video	Description
Highway video	This video sequence as shown in the figure (5.9) consists of 1700 frames, in this sequence number of different colour of vehicle are moving, sun is falling on the opposite side of the vehicles so shadow is visible with each vehicle. This is an outdoor shot video with a fixed camera. However the motion of trees branches is very slow as compared to the vehicles and this movement detection is considered as erroneous result [15]. Among all cited algorithms only the proposed algorithms did not cover the waving of tree leaves, as this is very slow. The worst results were produced by Local-self similarity.
Office video	This video sequence as shown in the figure (5.10) consists of 2050 frames. This is an indoor video with fixed camera. In this video the foreground detection algorithm is required to detect the man's movement i.e., entrance, standing and exit along with the book in the hand movement. The worst result is produced by GMM Zivkovic and Local-self similarity.
Pedestrians video	This video sequence as shown in the figure (5.11) consists of 1099 frames. The foreground algorithm is required to detect the movement of pedestrians inside the park where the camera is fixed. The shadow effect is very clear. The worst results were produced by local-self similarity.
PETS2006 video	This video sequence as shown in the figure (5.12) consists of 1200 frames. This is an indoor video and foreground detection algorithm is required to detect the moving people

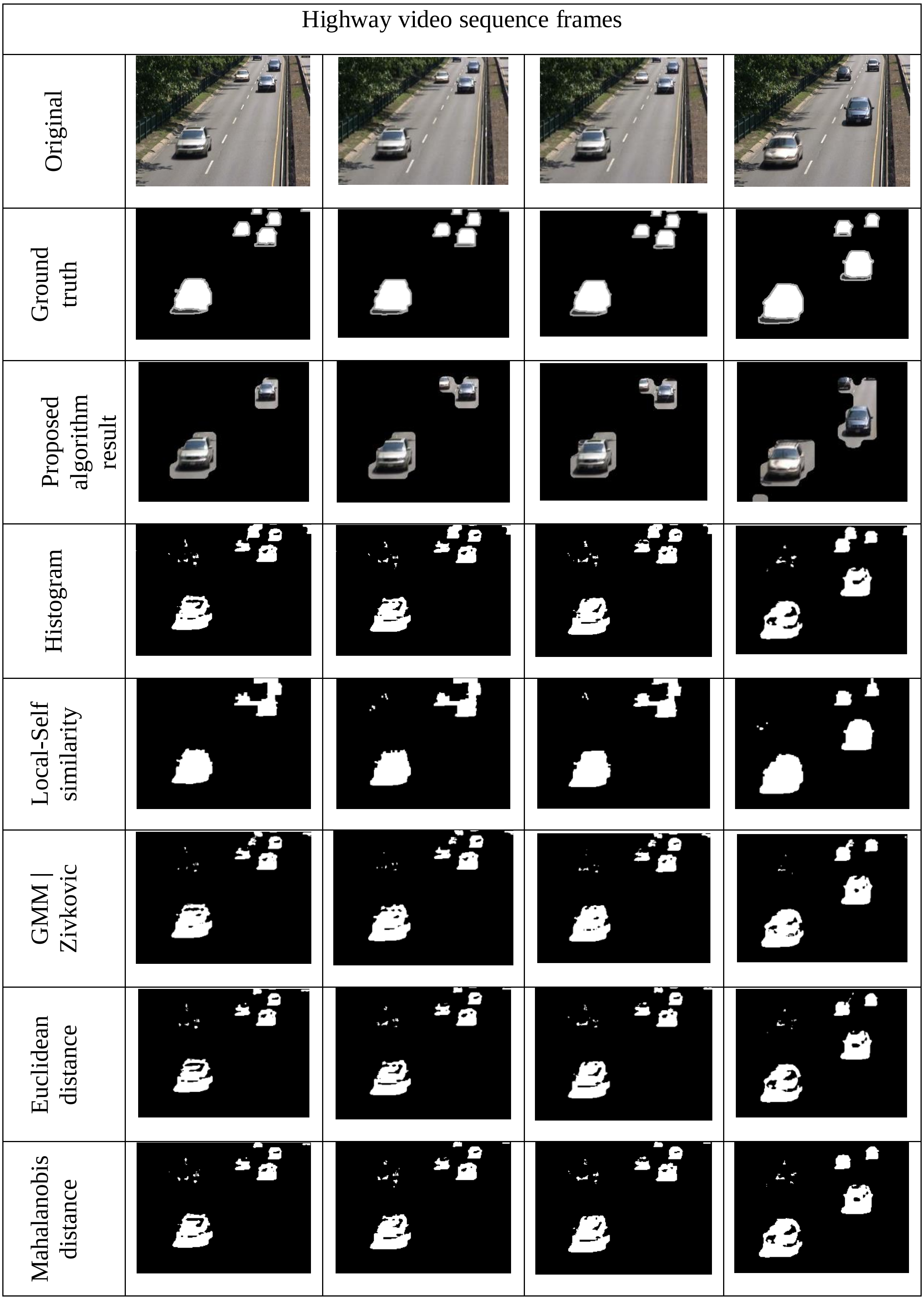


Figure 5.9 Highway video sequence experimental results

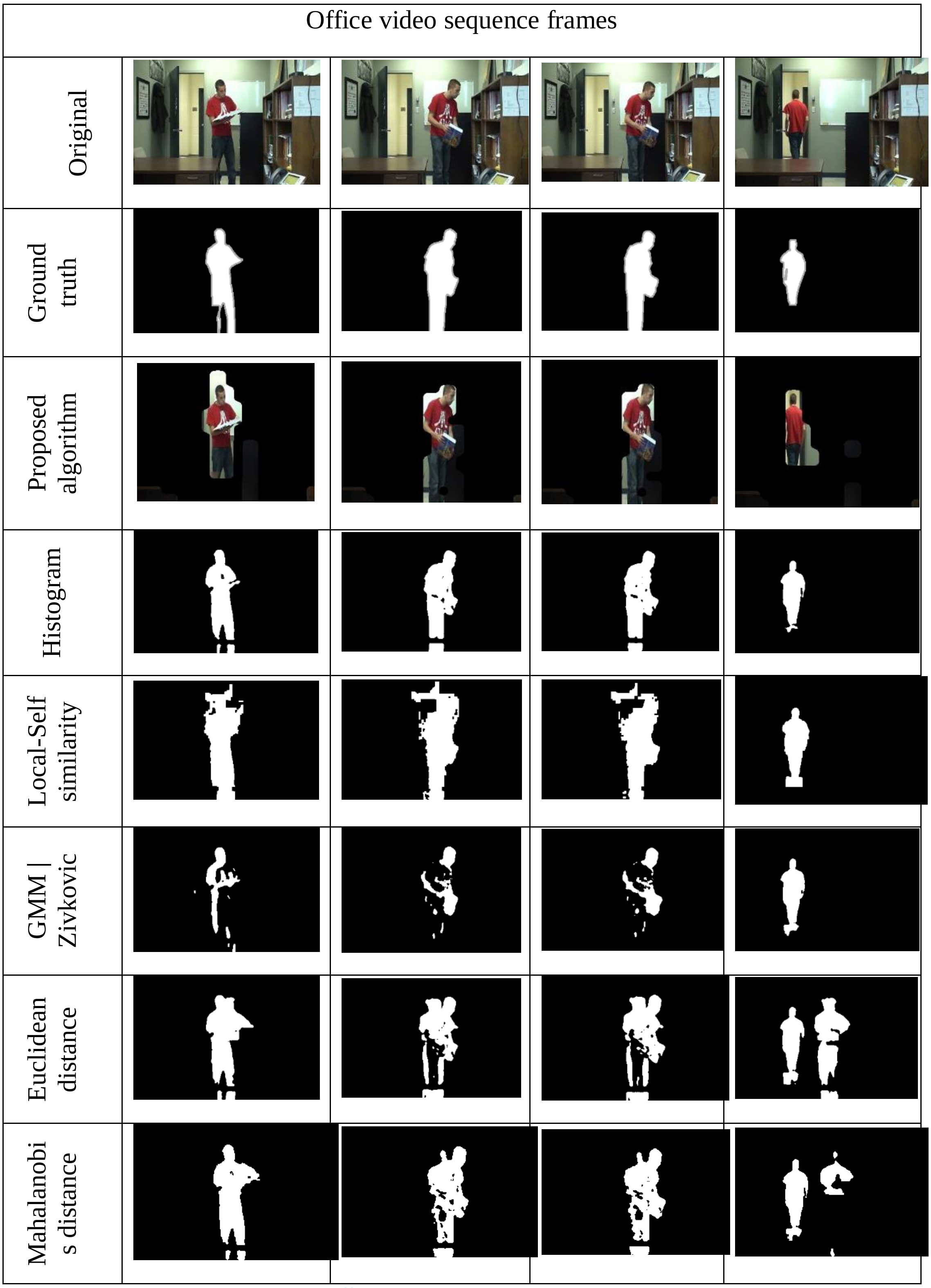


Figure 5.10 office video sequence experimental results

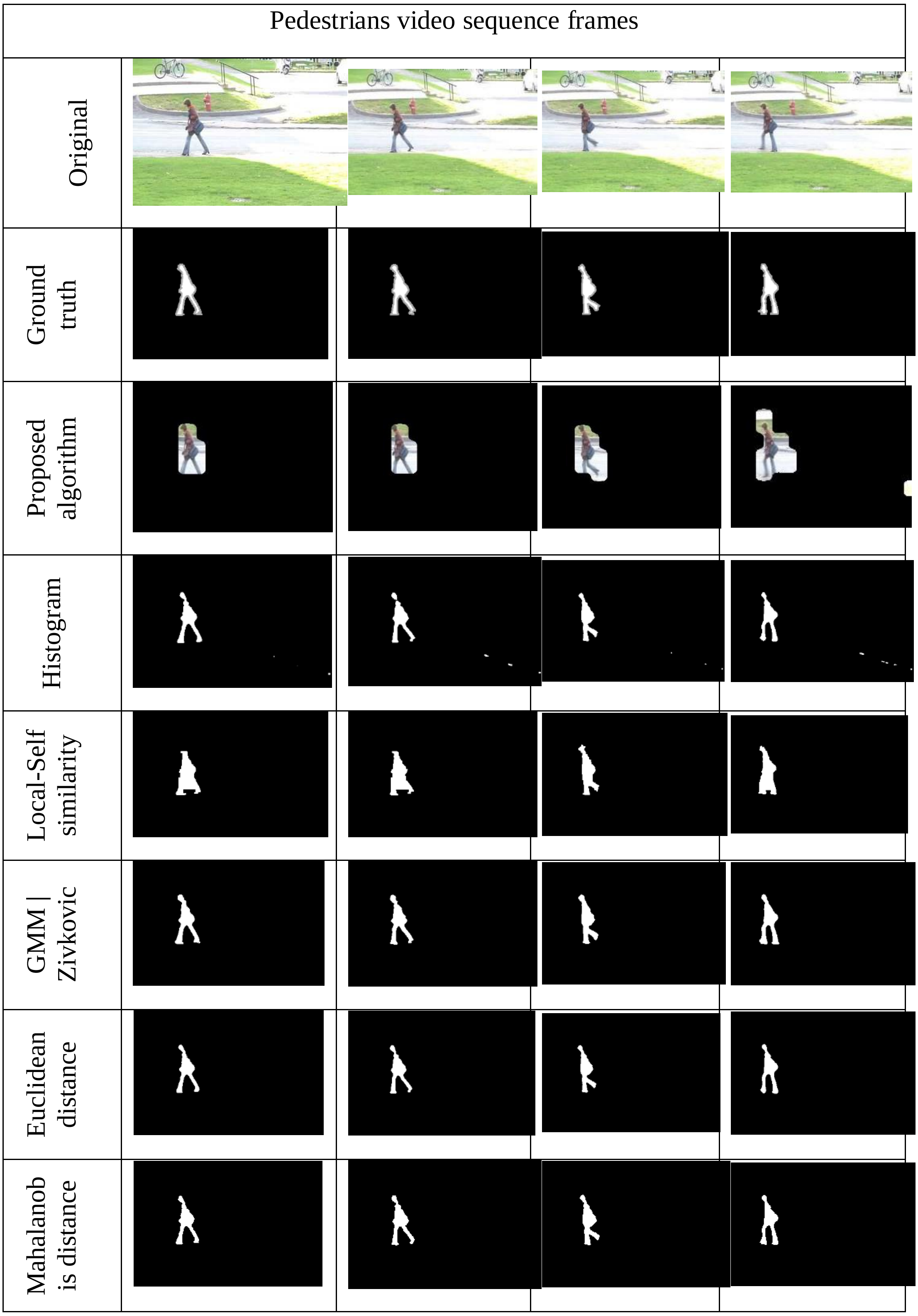


Figure 5.11 Pedestrians video sequence experimental results

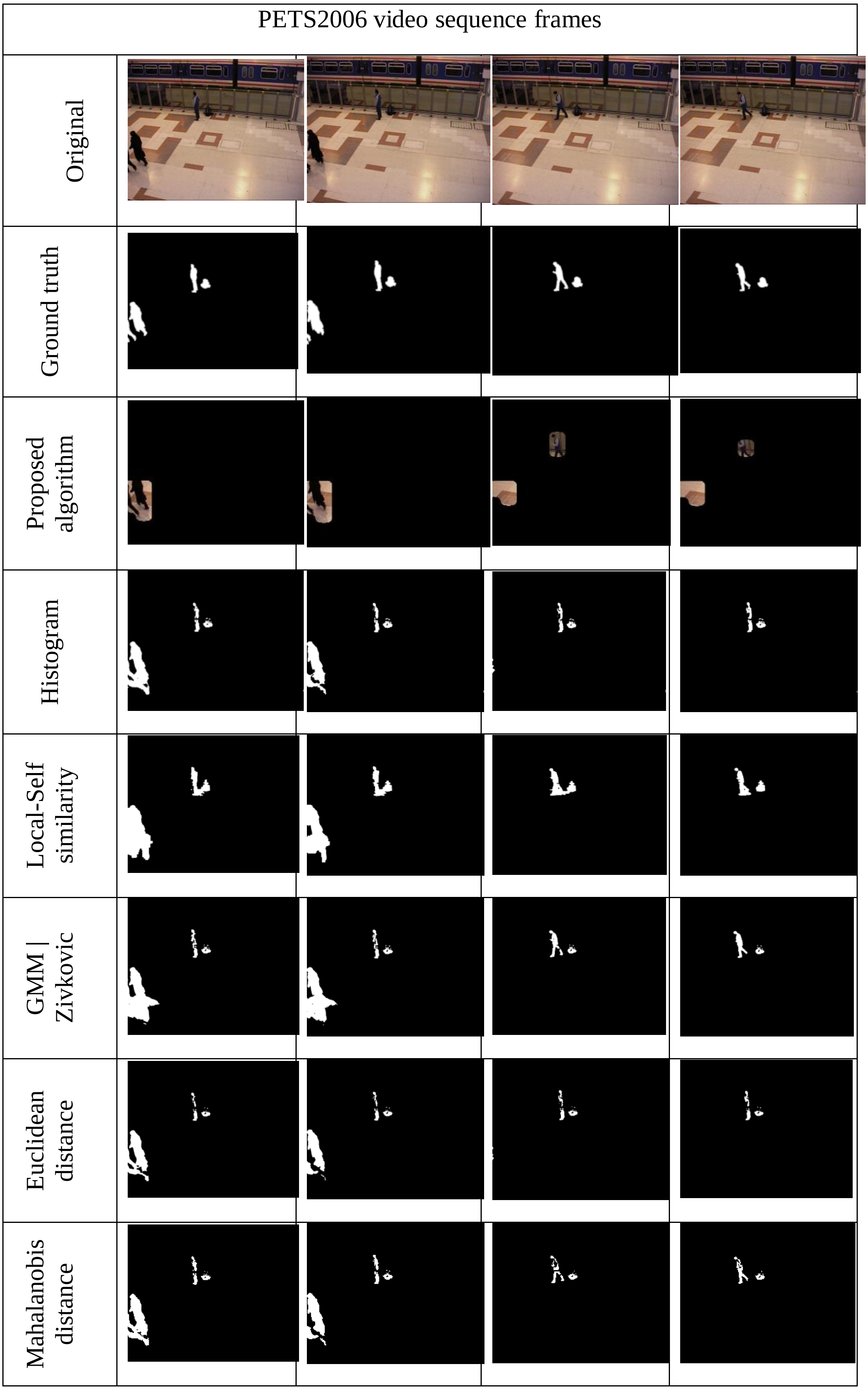


Figure 5.12 PETS2006 video sequence experimental results

5.6 INDIVIDUAL EVALUATION OF CITED AND PROPOSED ALGORITHMS BASED ON OBJECTIVE RESULTS OBTAINED

The proposed algorithm was tested on four different standard video sequences namely pedestrians, PETS2006, highway and office. To check the strength and weakness of the proposed algorithm the respective results were compared with sate of the art published foreground algorithm: Histogram [10], Local-Self similarity [11], GMM | Zivkovic [12], Euclidean distance [13] and Mahalanobis distance [14].

Based on the overall objective results of various performance measures in the Figure (5.19) the proposed method obtained the highest score, the Histogram approach obtained the second highest score and Mahalanobis distance the third. While Local-self similarity approach obtained very low scores on a range of different performance measures.

5.6.1 HISTOGRAM APPROACH

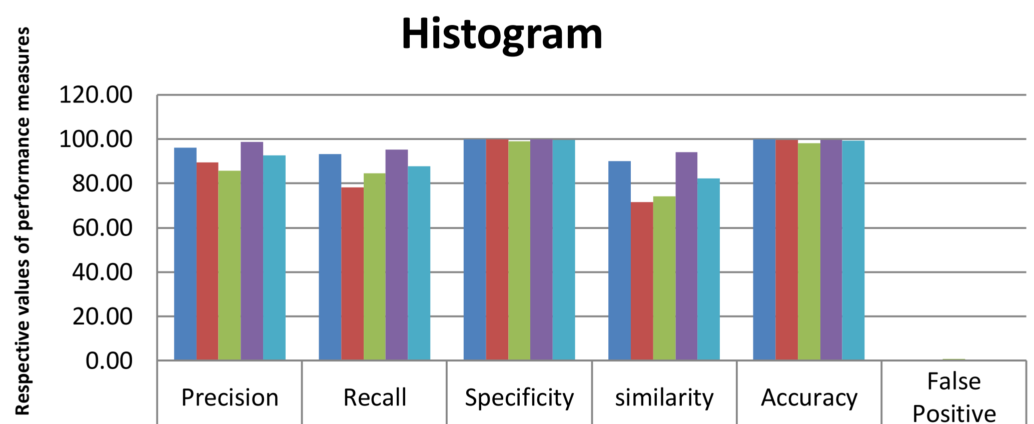
Figure (5.19), presents the overall performance measures results of Histogram approach. This method performs well in precision, specificity, similarity, accuracy and false positive.

This means that ratio of *tp* and *tn* is greater than *fp*. Which reflects the strength of this method as compared to Local-self similarity, Euclidean distance, GMM Zivkovic and Mahalanobis distance methods.

However the average value of specificity and false positive is equal to that of Local-Self similarity approach.

The main weakness of this method is the lower value of recall which is 87.77. This shows that ratio of *fn* is highest, and ranked 4th on recall value.

Figure (5.13) shows the individual and overall performance measure for four video sequences.



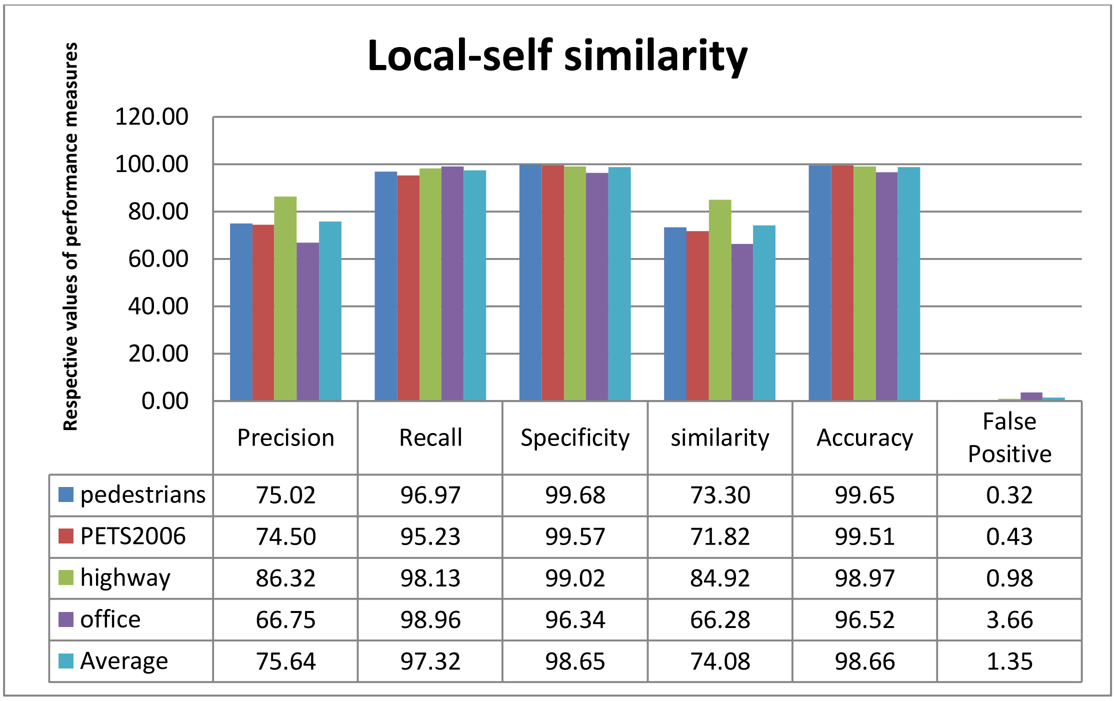
	Precision	Recall	Specificity	similarity	Accuracy	False Positive
pedestrians	96.16	93.31	99.96	89.96	99.90	0.04
PETS2006	89.45	78.13	99.88	71.53	99.60	0.12
highway	85.78	84.51	99.12	74.13	98.25	0.88
office	98.78	95.13	99.91	94.02	99.58	0.09
Average	92.54	87.77	99.72	82.41	99.33	0.28

Figure 5.13 Performance measures of Histogram Approach

5.6.2 LOCAL-SELF SIMILARITY

Figure (5.19), presents the overall performance measures of Local-Self similarity approach, this method performs very well only in recall (ranked as 1^{st}) while for the other performance measures it performs very low. This clearly shows that the average ratio of *fp*, *fn* is greater than *tp* and *tn*. In other words, this method did not detect the ground truth correctly among the five different video sequences.

Figure (5.14) shows the individual and overall performance measure for four video sequences.



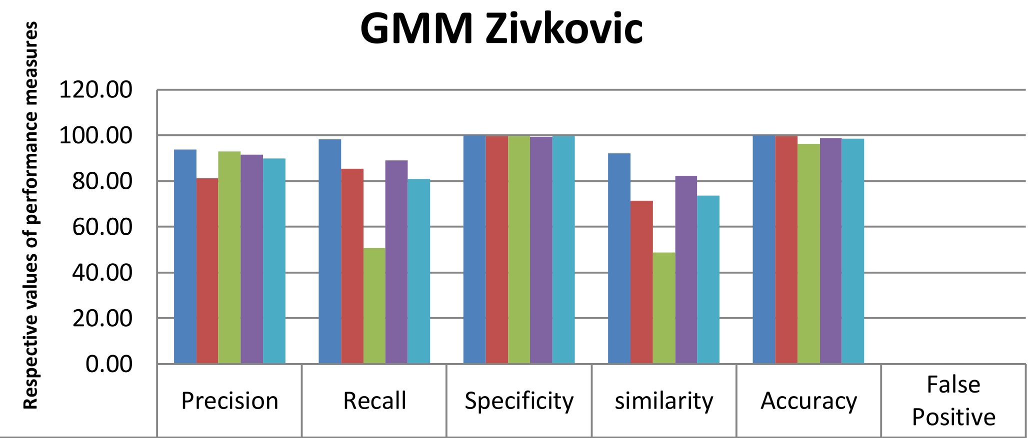
	Precision	Recall	Specificity	similarity	Accuracy	False Positive
pedestrians	75.02	96.97	99.68	73.30	99.65	0.32
PETS2006	74.50	95.23	99.57	71.82	99.51	0.43
highway	86.32	98.13	99.02	84.92	98.97	0.98
office	66.75	98.96	96.34	66.28	96.52	3.66
Average	75.64	97.32	98.65	74.08	98.66	1.35

Figure 5.14 Performance measures of Local-Self similarity

5.6.3 GMM | ZIVKOVIC

Figure (5.19), presents the overall performance measures of GMM Zivkovic method. This approach performs equally well in specificity and false positive as that of Histogram method, while on the other performance measures it performs very low compared to the other four algorithms.

Figure (5.15) shows the individual and overall performance measure for four video sequences.



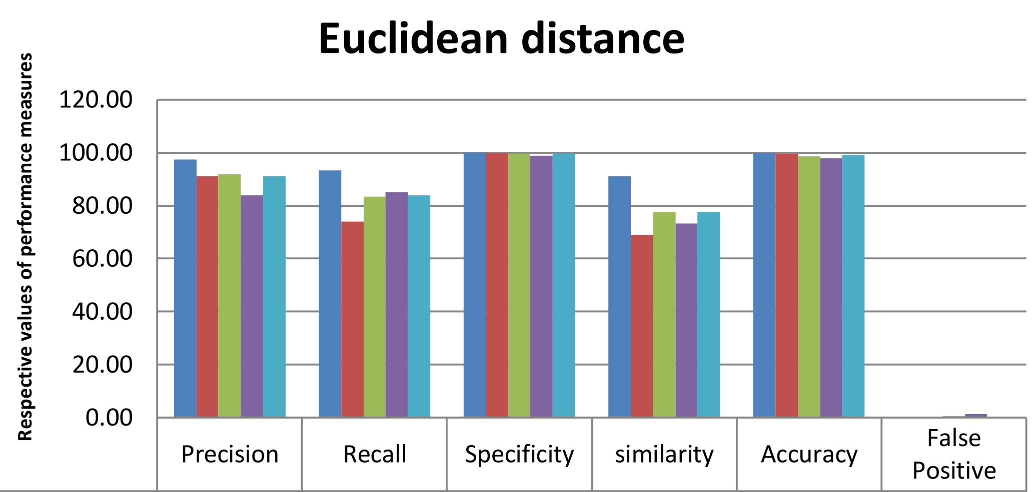
	Precision	Recall	Specificity	similarity	Accuracy	False Positive
pedestrians	93.86	98.20	99.94	92.27	99.92	0.06
PETS2006	81.35	85.28	99.74	71.33	99.55	0.26
highway	92.90	50.75	99.71	48.85	96.33	0.29
office	91.63	89.16	99.49	82.44	98.87	0.51
Average	89.93	80.85	99.72	73.72	98.67	0.28

Figure 5.15 Performance measures of GMM| Zivkovic

5.6.4 EUCLIDEAN DISTANCE

Figure (5.19), presents the overall performance measures of Euclidean distance algorithm. This method performs well only in precision (ranked as on number 3rd) as compared to other three other methods namely Mahalanobis distance, GMM Zivkovic and Local-self similarity. Based on the value of precision this clearly indicates that the ratio of *tp* is greater than *fp*.

Figure (5.16) shows the individual and overall performance measure for four video sequences.



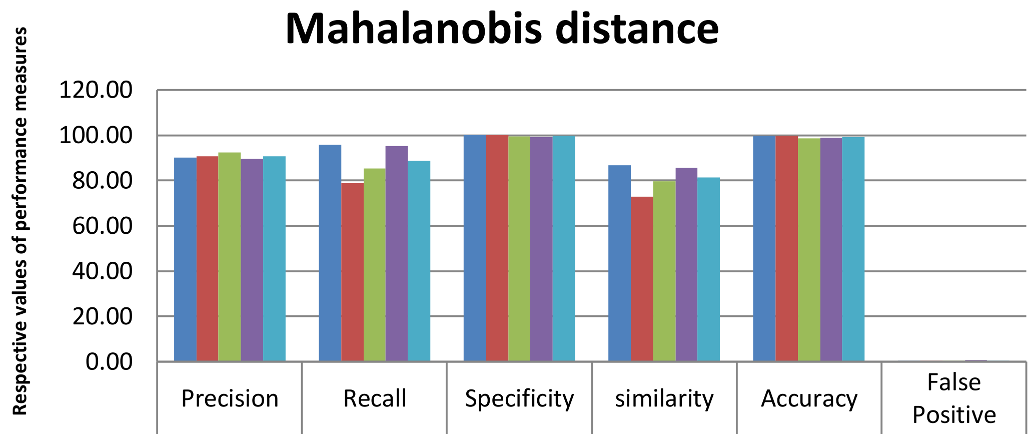
	Precision	Recall	Specificity	similarity	Accuracy	False Positive
pedestrians	97.50	93.22	99.98	91.04	99.91	0.02
PETS2006	91.22	73.85	99.91	68.95	99.57	0.09
highway	91.87	83.33	99.54	77.60	98.58	0.46
office	83.96	85.00	98.80	73.13	97.84	1.20
Average	91.14	83.85	99.55	77.68	98.97	0.45

Figure 5.16 Performance measures of Euclidean distance

5.6.5 MAHALANOBIS DISTANCE

Figure (5.19), presents the overall performance measures of Mahalanobis distance algorithm. Overall this method performs well enough (ranked 3rd) in recall, specificity, similarity, accuracy and false positive. This indicates that the ratio of tp and tn is higher than fp and fn. While the value of precision is lower than proposed method, Histogram and Euclidean distance. Based on this result it is shown that the ratio of *tp* is lower than *fp*.

Figure (5.17) shows the individual and overall performance measure for four video sequences.



	Precision	Recall	Specificity	similarity	Accuracy	False Positive
pedestrians	90.20	95.67	99.90	86.66	99.86	0.10
PETS2006	90.76	78.75	99.89	72.90	99.62	0.11
highway	92.32	85.40	99.55	79.73	98.71	0.45
office	89.57	95.08	99.18	85.60	98.90	0.82
Average	90.71	88.72	99.63	81.22	99.27	0.37

Figure 5.17 Performance measures of Mahalanobis distance

5.6.6 PROPOSED ALGORITHM

Figure (5.19), shows the overall performance measures of the proposed algorithm. Based on average values of performance measures for the four videos, the proposed algorithm performs best in precision, specificity, similarity, accuracy and false positive. This reflects that algorithm detected maximum foreground mask as compared to other approaches. The value of recall is lower than one cited approach i.e., Local-self similarity, which has the highest value of 97.32 while that of the proposed algorithm obtained 91.20. The reason of this low rank in recall is that proposed method obtains higher values of *fn* for the three videos: pedestrians, PETS2006 and Highway than that of Local-self similarity method as shown in the Table (5.1) and Table (5.2). However due to overall high values of *tp* and *tn* and lower the values of *fp* and *fn* of the proposed method has detected sufficient foreground and correct background.

Figure (5.18) shows the individual and overall performance measure for four video sequences.

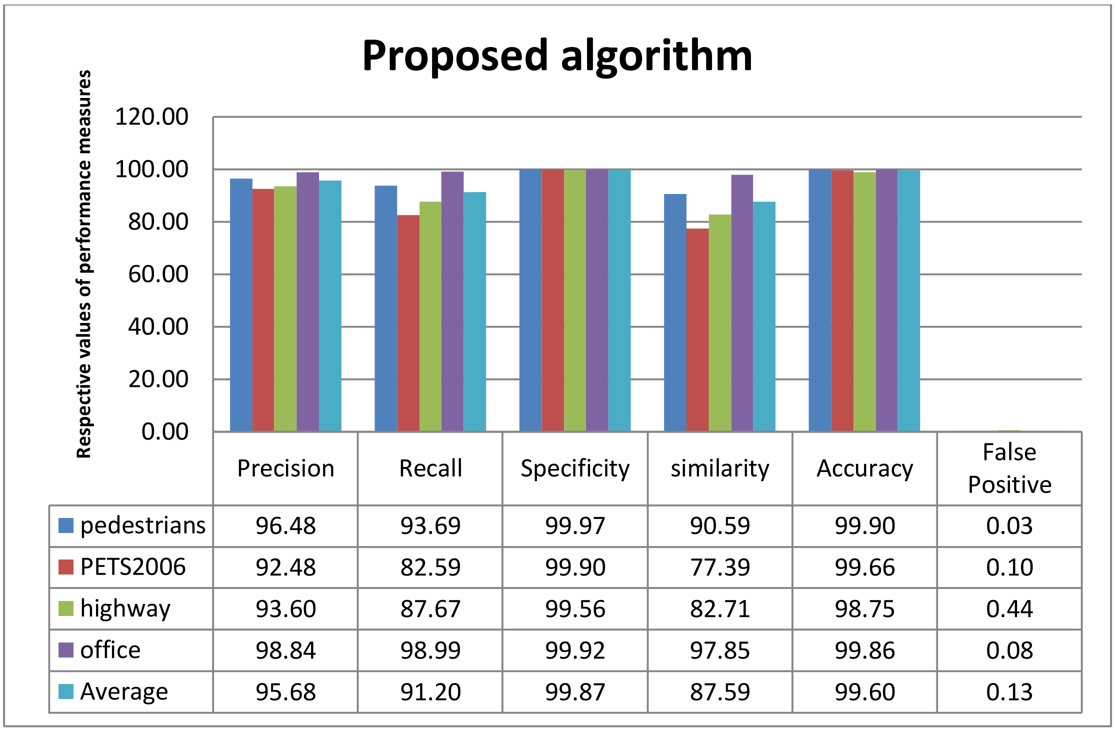


Figure 5.18 Performance measures of proposed algorithm

Table 5.1 Proposed algorithm

Proposed Algorithm				
Video sequences	TP	FP	FN	TN
Pedestrians	654844	23876	44125	68472151
PETS2006	4464520	363120	941280	377258274
Highway	5723546	391465	805326	88890111
Office	8619411	100859	88083	125789544
Average	**4865580**	**219830**	**469704**	**165102520**

Table 5.2 Local-self similarity

Local-self similarity Algorithm				
Video Sequences	TP	FP	FN	TN
Pedestrians	650557	216616	20305	67352491
PETS2006	4597989	1574118	230201	365139186
Highway	5355956	849083	102033	85805386
Office	8552271	4260755	89823	112277931
Average	**4789193**	**1725143**	**110591**	**157643749**

5.7 Discussion on individual performance measure for cited and proposed algorithms

There are 6 basic performance measurements that were used: precision, recall, specificity, accuracy, similarity and false positive rate. With the help of these measures the proposed algorithm is compared with five state of the art algorithms [10-14] on four change detection benchmark datasets [15], which also contains its ground truth values. All performance measures results (%average) for the respective method are shown in the figure (5.19).

5.7.1 Precision

Our precision value is 95.68% as shown in figure (5.19), meaning that we have been able to identify more of the ground truth (intended region of foreground) than other techniques, while the ideal value of precision is 100.

The second highest value is that of the Histogram approach, which obtained 92.54%, while the Local-self similarity method performs poorly with a value of only 75.64%.

5.7.2 Recall or Sensitivity or True Positive Rate(*TPR*)

As shown in Figure (5.19), there was as much false identification of regions with the proposed method as with the other techniques.

The ideal value of Recall is 100. The proposed algorithm has achieved 91.20%, and ranked as 2nd.

The overall highest value that was obtained is that of the Local-self similarity algorithm which is 97.32%, while the GMM Zivkovic method performs worse with a value of 80.85%.

5.7.3 Specificity or True Negative Rate

The ideal value of specificity is 100%, and the proposed algorithm has achieved 99.87% in the figure (5.19), which is the highest value among the other five algorithms.

The second highest values that was obtained are that of the Histogram and GMM Zivkovic algorithms, which is 99.72%, while the Local-self similarity method performs lowest with a value of 98.65%.

5.7.4 Similarity

The proposed algorithm has achieved 87.59% in the figure (5.19), which is the highest value of similarity. The second highest value that was obtained is that of Histogram which is 82.41%, while GMM Zivkovic similarity performs poorly with the value of 73.72%.

5.7.5 Accuracy

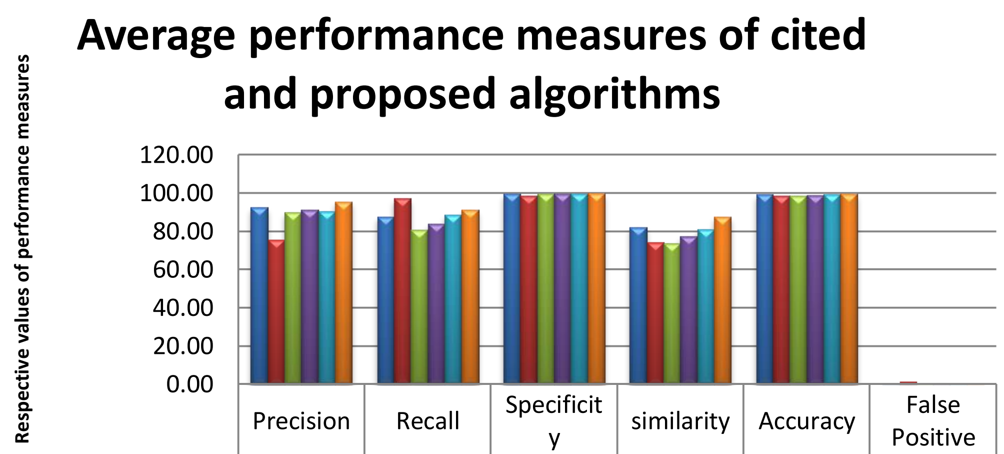
The proposed algorithm has achieved 99.60% in the figure (5.19) which is the highest value compared to the other five algorithms. The second highest value that was obtained is that of Histogram approach which is 99.33%, while Local-self similarity performs lowest with the value of 98.66%.

5.7.6 False Positive Rate

The proposed algorithm has achieved 0.13% in the figure (5.19) which is the best value compared to the other five algorithms. The second highest values are that of Histogram and GMM Zivkovic approaches which is 0.28%, while Local-self similarity perform with the lowest of value of 1.35%.

The comparison of results is shown in the Figure (5.19). It is obvious that the proposed algorithm clearly outperforms the other five methods. Histogram approach is the second best approach. Mahalanobis distance approach being was found to perform the third best method, while, the Euclidean Distance algorithm was found the fourth best algorithm, based on performance measurement results. Overall, the performance of the Local-self similarity and GGM Zivkovic techniques were found to be non-satisfactory.

It is clearly shown from the results obtained in the figure (5.20), that the proposed algorithm performs much better than the second best algorithm Histogram, on average by 12.30%.



	Precision	Recall	Specificity	similarity	Accuracy	False Positive
Histogram	92.54	87.77	99.72	82.41	99.33	0.28
Local-Self similarity	75.64	97.32	98.65	74.08	98.66	1.35
GMM Zivkovic	89.93	80.85	99.72	73.72	98.67	0.28
Euclidean Distance	91.14	83.85	99.55	77.68	98.97	0.45
Mahalanobis distance	90.71	88.72	99.63	81.22	99.27	0.37
Proposed Method	95.68	91.20	99.87	87.59	99.60	0.13

Figure 5.19 Performance measures average of all algorithms

Method	%Precision	%Recall	%Specificity	%Similarity	%Accuracy	%False Positive	Sum
Histogram	92.54	87.77	99.72	82.41	99.33	0.28	462.05
Local-Self similarity	75.64	**97.32**	98.65	74.08	98.66	1.35	
GMM Zivkovic	89.93	80.85	99.72	73.72	98.67	0.28	
Euclidean Distance	91.14	83.85	99.55	77.68	98.97	0.45	
Mahalanobis distance	90.71	88.72	99.63	81.22	99.27	0.37	
Proposed Method	**95.68**	91.20	**99.87**	**87.59**	**99.60**	**0.13**	474.06
Difference	3.13	-6.12	0.15	5.18	0.26	0.15	12.30

Figure 5.20 Differences from proposed algorithm to 2nd highest

5.8 THE EFFECT OF HIGH CAMERA MOTION ON FOREGROUND DETECTION RESULTS

During the experimental process it was observed that the proposed algorithm does not produce good enough results where there is a high camera motion and foreground objects motion simultaneously. Figure (5.21), shows the results of video sequence wild life, where camera is moving with high speed to capture the movements of running horses. The respective histogram shows that the highest peak lies in the second half of the histogram which indicates that there is high motion. This high motion is because of two (double) motions: foreground objects and camera motion, so approximately the whole scene is considered as foreground object. As a result erroneous foreground is detected. In this figure (5.21), second column shows the foreground results where black mask represents background and RGB area represents detected foreground.

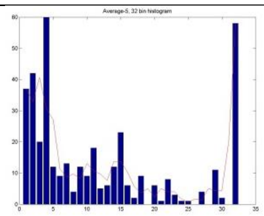
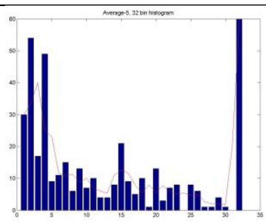
Original Frame	Respective Foreground	Respective histogram
		
		

Figure 5.21 High camera motion results

5.9 CONCLUSION

In this chapter, we presented a simple technique to obtain the threshold value for background subtraction using a method that first smoothes the motion histogram and then separates foreground (motion area) and background (static). The proposed algorithm is tested on four different video sequences of various natures and produced satisfactory results subjectively and objectively. From the experimental results it is clear that the proposed algorithm performs very well when camera and background are fixed and only the foreground object(s) is/are in motion; however when camera and foreground objects are moving very fast simultaneously the proposed algorithm may produce erroneous results.

During the experimentation process hence it is concluded that the best feature of this algorithm is that it covers the sufficient mask of the moving object(s), and *foreground aperture* and *ghosting* problem is not detected in the proposed method.

During the experimentation process hence it is concluded that the best feature of this algorithm is that it covers the sufficient mask of the moving object(s), and *foreground aperture* and *ghosting* problem is not detected.

5.10 REFERENCES:

1. Collins, R.T., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O. and Burt, P. (2000) *A system for video surveillance and monitoring,* Carnegie Mellon University, the Robotics Institute Pittsburg.

2. Kameda, Y. and Minoh, M. (1996) "A human motion estimation method using 3-successive video frames", *International conference on virtual systems and multimedia*, pp. 135.

3. Kanade, T., Collins, R., Lipton, A., Burt, P. and Wixson, L. (1998) "Advances in cooperative multi-sensor video surveillance", *Proceedings of DARPA Image Understanding Workshop*Citeseer, pp. 2.

4. Migliore, D.A., Matteucci, M. and Naccari, M. (2006) "A revaluation of frame difference in fast and robust motion detection", *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks* ACM , pp. 215.

5. Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P. (1997) "Pfinder: Real-time tracking of the human body", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, no. 7, pp. 780-785.

6. Hati, K.K., Sa, P.K. and Majhi, B. (2012) "LOBS: Local background subtracter for video surveillance", *Microelectronics and Electronics (PrimeAsia), 2012 Asia Pacific Conference on Postgraduate Research in*IEEE, pp. 29.

7. Martínez-Martín, E. and del Pobil, A.P. (2012) *Robust motion detection in real-life scenarios,* Springer.

8. Sun, S., Wang, Y.F., Huang, F. and Liao, H.M. (2012) "Moving foreground object detection via robust SIFT trajectories", *Journal of Visual Communication and Image Representation*.

9. Barjatya, A. (2004) "Block matching algorithms for motion estimation", *IEEE Transactions Evolution Computation,* vol. 8, no. 3, pp. 225-239.

10. Zheng, J., Wang, Y., Nihan, N.L. and Hallenbeck, M.E. (2006) "Extracting roadway background image: Mode-based approach", *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1944, no. 1, pp. 82-88.

11. Jodoin, J., Bilodeau, G. and Saunier, N. (2012) "Background subtraction based on local shape", *arXiv preprint arXiv:1204.6326,* .

12. Zivkovic, Z. (2004) "Improved adaptive Gaussian mixture model for background subtraction", *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* IEEE, pp. 28.

13. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. and Rosenberger, C. (2010) "Comparative study of background subtraction algorithms", *Journal of Electronic Imaging*, vol. 19, no. 3, pp. 033003-033003-12.

14. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. and Rosenberger, C. (2010) "Comparative study of background subtraction algorithms", *Journal of Electronic Imaging*, vol. 19, no. 3, pp. 033003-033003-12.

15. Goyette, N., Jodoin, P., Porikli, F., Konrad, J. and Ishwar, P. (2012) "Changedetection. net: A new change detection benchmark dataset", *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on* IEEE, pp. 1.

16. Velho, L. (2009) *Image processing for computer graphics and vision*, Springer.

17. Media College (n.d) *Aspect ratios.* Available at: http://www.mediacollege.com/video/aspect-ratio/ (Accessed: 20 June, 2013).

Chapter 6
CONCLUSION

6.1 INTRODUCTION

This chapter presents the main conclusion and summarises the major contributions of this thesis. The future work section highlights those research areas where the findings of this research can further be investigated for new research directions.

6.2 CONCLUSION

This thesis has investigated the issues of precise foreground detection which are compared on the base of performance measures with existing research in the field of foreground detection.

Precise motion detection followed by full mask of the moving object extraction is our prime goal. For the said purpose motion estimation and morphological operation: opening-and-closing-by-reconstruction approaches were utilized in order to achieve the main goal of precise motion detection followed by full mask of moving object extraction. Operation opening-and-closing-by-reconstruction approach is pixel based which also increases the accuracy of foreground detection process [1] and is not possible in block based motion segmentation. Opening-and-closing-by-reconstruction identifies the minima and maxima inside the foreground object which leads in the further enhancement of foreground detection result and played a very important role in obtaining of full foreground object mask [1]. In order to find accurate areas of the motion the proposed algorithms utilized block matching algorithm for motion estimation which is less time consuming as compared to pixel based motion estimation process. Since our proposed motion estimation approach is block based, it requires less time for execution. For the omission of miss and over calculated motion areas certain morphological operations are used in a particular fashion to overcome this issue. In the proposed algorithm there is no need of reference image in advance [1]. As required in the Background subtraction method.

6.2.1 PROPOSED ALGORITHM-I

The proposed algorithm is tested and verified by 11 different performance measures to measure its performance scientifically. It is clearly shown from the results obtained from various performance measures that the proposed algorithm performs much better than four well-established algorithms [2-5], on average by more than 24.74%.

The proposed method, precision value is 93.60% meaning that we have been able to identify more of the ground truth (intended region foreground) than other techniques. The second highest value that was obtained is that of the SGM-R algorithm, which is 73.51%, while the Optical Flow method performs poorly with a value of only 65.75%.

There was as much false identification of regions with the proposed method as with the other techniques. The proposed algorithm has achieved 93.44% on recall. The overall highest value that was obtained is that of the Soo Wan Kim algorithm, which is 97.86%, while the Optical Flow method performs worse with a value of 90.81%.

The proposed algorithm has achieved 93.46% F-score of Precision and Recall, which is the highest value among the other four algorithms. The second highest value that was obtained is that of the SGM-R algorithm, which is 82.65%, while the Optical Flow method performs worse with a value of 75.88%.

The proposed algorithm has attained 88.23% specificity, which is the highest value among the other four algorithms. The second highest value is that of the SGM-R algorithm, which is 39.24%, while the Optical Flow method performs worse with a value of 17.68%.

The proposed algorithm has accomplished 90.84% for Balance Classification Rate or Area Under the Curve which is the highest value among the other four algorithms. The second highest value that was obtained is that of the SGM-R algorithm which is 66.84%, while the Optical Flow method performs worse with a value of 54.25%.

The proposed algorithm has achieved 90.65% for Geometric Mean of Sensitivity and Specificity, which is the highest value among the other four algorithms. The second highest value that was obtained is that of SGM-R which is 60.67%, while Optical Flow performs worse with a value of 38.79%.

The proposed algorithm has achieved 90.48% on F-Score of Sensitivity and Specificity. The second highest value that was obtained is that of SGM-R which is 55.16%, while Optical Flow performs worse with a value of 28.35%.

The proposed algorithm has succeeded with value of 9.16% on %Balance Error Rate, which is the best value. The second best value that was obtained is that of SGM-R which is 33.16%, while Optical Flow performs poorly with the value of 45.75%.

The proposed algorithm has obtained 87.78% similarity which is the highest value. The second highest value that was obtained is that of SGM-R which is 70.44%, while Optical Flow performs poorly with the value of 61.91%.

The proposed algorithm has achieved 91.58% accuracy, which is the highest value from other four algorithms. The second highest value that was obtained is that of SGM-R which is 74.59%, while Optical Flow performs poorly with the value of 64.51%.

The proposed algorithm has achieved 11.76% for the false positive performance measure, which is the best value from other four algorithms. The second highest value that was obtained is that of SGM-R which is 60.76%, while Optical Flow performs poorly with the value of 82.32%.

To summarise the above discussion, it is obvious that the proposed algorithm clearly outperforms the other four methods. SGM-R is the second best approach. MoG being the most similar technique to SGM-R was found the third best method, while, the Soo Wan Kim algorithm was found to perform the fourth best algorithm, based on performance measurement results. Overall, the performance of the Optical Flow technique was found to be the most non-satisfactory.

The recall value of the proposed method is lower than Soo Wan Kim algorithm by 4.42%. The recall or true positive rate (*trp*) and precision quantifies how well an algorithm matches the ground truth [6, 7], but the proposed algorithm outperforms in precision and %F-score of precision and recall over the rest of the four methods by 20.09% and 10.81%, respectively. It is also important to know that only recall is not sufficient to compare different methods and is generally used in conjunction with precision that provides the percentage of detected true positive as compared to the total number of items detected [8]. It is clearly shown from the results obtained, that the proposed algorithm performs much better than the second best algorithm SGM-R, on average by 24.74%.

This proposed contribution presented a simple and effective algorithm to obtain sufficient precise foreground from background using motion estimation, maxima and minima inside the foreground object. The previous works [2-5,10] which are based on foreground detection shows that our final result has produced better foreground mask when compared quantitatively and qualitatively (subjectively). For quick and accurate execution of block motion estimation we have used Adaptive Rood Pattern Search algorithm.

In order to obtain the precise mask of the foreground we used opening-and-closing operation. From the performance measures it is shown that our algorithm is relatively more accurate in terms of precision, %F-score of precision, recall, sensitivity, and specificity, specificity, area under the curve, accuracy and similarity.

6.2.2 PROPOSED ALGORITHM-II

In the experiments so far, this proposed algorithm did not encounter any *ghosting* or *foreground aperture* problem in the videos ranging from slow to normal and fast plus indoor and outdoor videos.

It is obvious that the proposed algorithm clearly outperforms all the other five methods. Histogram approach [14] is the second best approach. Mahalanobis [15] distance approach was found to be the third best method, while the Euclidean

Distance algorithm [16] was found the fourth best algorithm, based on average of performance measure results. Overall, the performance of the Local-self similarity [16] and GGM Zivkovic [17] techniques were found to be non-satisfactory. It is clearly shown from the results obtained that the proposed algorithm performs much better than the second best algorithm Histogram, on average by 12.30%.

The overall performance measures of the proposed algorithm. Based on average values of all the classic performance measures for four videos, the proposed algorithm performs best for precision, specificity, similarity, accuracy and false positive. This reflects that algorithm detected the maximum foreground mask as compared to other approaches. The value of recall is lower than the one cited approach i.e., Local-self similarity, which has the highest value of 97.32 while that of the proposed algorithm obtained 91.20. The reason of this low rank in recall is that proposed method has higher values of *fn* for three videos: pedestrians, PETS2006 and highway than that of Local-self similarity method. However the overall high values of *tp* and *tn* and lower the values of *fp* and *fn* of the proposed method suggests that it has detected sufficient foreground and correct background.

The value of proposed algorithm, precision is 95.68%, meaning that we have been able to identify more of the ground truth (intended region of foreground) than other techniques. The second highest value that was obtained is that of the Histogram approach, which is 92.54%, while the Local-self similarity method performs poorly with a value of only 75.64%.

The value for recall for proposed algorithm was as much false identification of regions with the proposed method as with the other techniques. The proposed algorithm has achieved 91.20% for recall. The overall highest value is that of the Local-self-similarity algorithm which is 97.32%, while the GMM Zivkovic method performs worse with a value of 80.85%.

The proposed algorithm has achieved 99.87% on specificity, which is the highest value among the other five algorithms. The second highest values are that of the Histogram and GMM Zivkovic algorithms, which obtained 99.72%, while the Local-self similarity method performs lowest with a value of 98.65%.

The proposed algorithm has achieved 87.59% for similarity, which is the highest value. The second highest value that was obtained is that of Histogram which is 82.41%, while GMM Zivkovic similarity performs poorly with the value of 73.72%.

The proposed algorithm has achieved 99.60% for accuracy which is the highest value from other five algorithms. The second highest value that was obtained is that of Histogram approach which is 99.33%, while Local-self similarity performs lowest with the value of 98.66%.

The proposed algorithm has achieved 0.13% on the value of false positive, which is the best value compared to all the other five algorithms. The second highest values are that of Histogram and GMM Zivkovic approaches which is 0.28%, while Local-self similarity perform with the lowest of value of 1.35%.

It is obvious that the proposed algorithm clearly outperforms the other five methods. Histogram approach is the second best approach. Mahalanobis distance approach was found to perform the third best, while, the Euclidean Distance algorithm was found to perform the fourth best algorithm, based on performance measure results. Overall, the performance of the Local-self similarity and GGM Zivkovic techniques were found to be non-satisfactory. It is clearly shown from the results obtained, that the proposed algorithm performs much better than the second best algorithm Histogram, on average by 12.30%.

In this proposed approach, we presented a simple technique to obtain the threshold value for background subtraction using a method that first smoothes the motion histogram and then separates foreground (motion area) and background (static). The proposed algorithm is tested on four different video sequences of various natures and produced satisfactory results subjectively and objectively. From the experimental results it is clear that the proposed algorithm performs very well.

During the experimentation process it is concluded that the best feature of this algorithm is that it covers the sufficient mask of the moving object(s), and *foreground aperture* and *ghosting* problems were not detected.

6.3 Future Work

There are several recommendations which can be used for future research direction in the area of foreground detection.

6.3.1 Recall value

As it was noticed in the proposed solutions that the performance measure , recall is lower than the other selected algorithms, so it requires systematic investigation to identify why this is occurring in the proposed algorithms, although as stated earlier that only recall is not a sufficient measure to judge the performance of an algorithm, its alternative F-score of precision and recall is relatively good choice for judgment where the proposed algorithms produce good enough results as compared to other algorithms.

6.3.2 Foreground detection when there is high camera and foreground objects motion is involved

The second proposed algorithm has the capacity to find the direction of various objects in the frame. Presently this utilizes the motion vectors to find the motion magnitude and the motion histogram is generated to find the area between two peaks to detect the foreground and background. If in the same manner motion direction is obtained, this can be used to predict the foreground object direction. On the further analysis one can also differentiate the camera motion.

A brief idea can be extracted from Figure (6.1) to Figure (6.4). Figure (6.1), is the original frame of video sequence, wild life where there is high motion of foreground objects, horses and in the background water waves can be seen. There is also very high motion of camera simultaneously.

Original Image

Figure 6.1 Original frame

Figure (6.2) shows the two things: blue bars represent the number of blocks in motion and red line is the smooth histogram of motion. It is to be noted that the first highest peaks lies in the second half of the histogram which indicates there is high motion. The reason of this high motion is due to two combined motion: high camera motion and foreground objects motion.

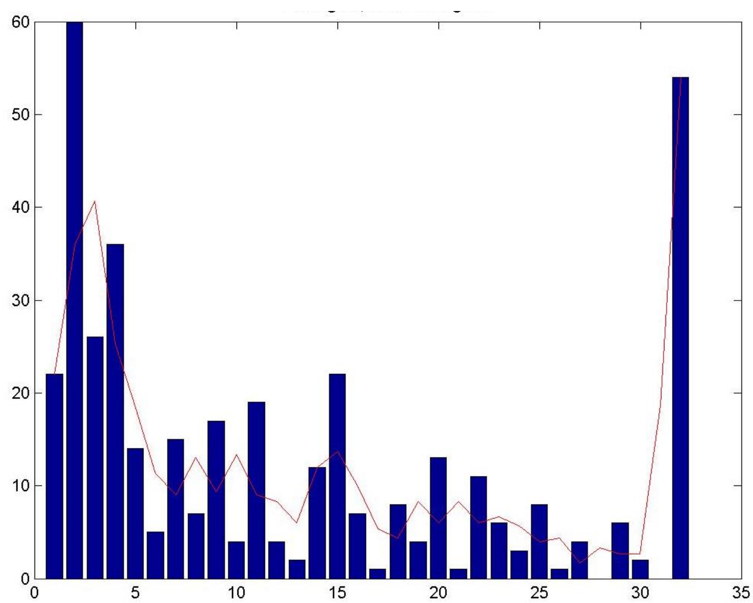


Figure 6.2 Smooth motion histogram

Figure (6. 3), shows motion vectors in respective frame, where dots represent still objects and various directed arrows are of foreground objects motion, as well as of camera motion, which is moving from left to right or in the same direction of foreground objects.

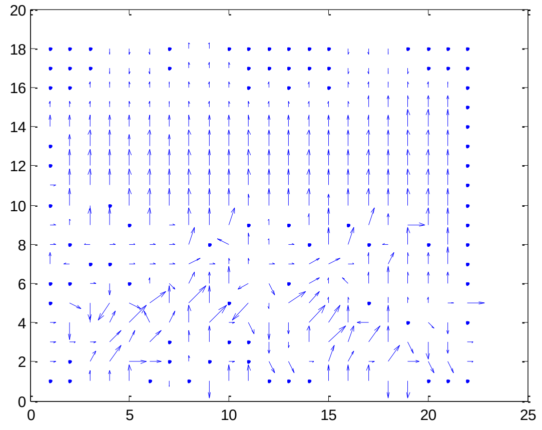


Figure 6.3 Motion vectors

Figure (6. 4) is very important and known as compass. In this figure various directed arrows shows the number of foreground objects motion in particular direction ranging from 0 to 360 degrees.

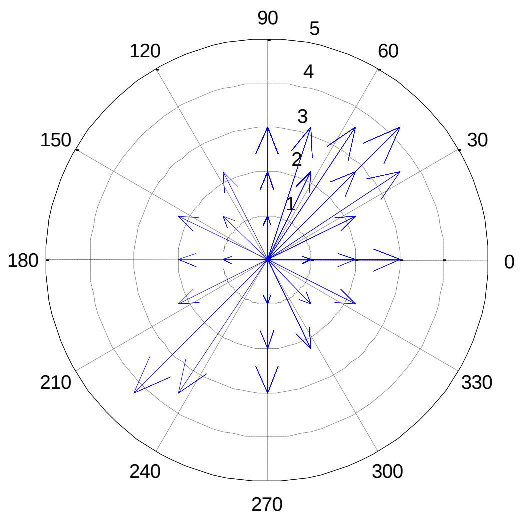


Figure 6.4 Respective Compass

If the results of compass are properly analyzed so proper direction of foreground objects can be determined, similarly camera motion can be determined.

If needed one can eliminate the camera motion, as that produce erroneous result and badly affect the foreground object motion.

6.3.3 MORE ACCURATE FOREGROUND

The idea of opening-and-closing-by-reconstruction which finds the minima and maxima inside the foreground object and ultimately leads to more precise foreground mask (as explained in chapter-4, of this thesis) can be incorporated in our second proposed algorithm (as explained in chapter-5, of this thesis). Thus as a result a more accurate algorithm can be designed by this integration.

6.3.4 PRECISE THRESHOLD

Our second proposed algorithm (as explained in chapter-5, of this thesis) estimates threshold from two peak of the motion histogram. So as a future work it is suggested to rather than calculating threshold value from two peaks only, why not to consider next two higher peaks also and generate the value of threshold. Finally the results of this algorithm can be compared with the existing algorithm.

6.3.5 EXECUTION TIME

It is also suggested for the future work to find the execution time of various algorithms on different standard foreground algorithms and then compared with the proposed algorithms execution time. However in general execution time is inversely proportional to the accuracy of algorithm as normally highly accurate algorithms are complex in nature and ultimately requires higher execution time.

So it is recommended that during this comparison both factors should be considered.

6.3.6 PERFORMANCE IN VARIOUS SITUATIONS

It is also suggested for the future work to examine the performance of proposed algorithms i.e., both in the wider range of situations such as semi dark, raining and cloudy weather etc. However to some extent the solution is not very difficult as already researchers have worked and are working on these issues.

6.3.7 PERFORMANCE ON LIVE VIDEO

As a very important component for future task, the proposed algorithm should be examined on live video obtained from surveillance systems. Although there will be very slight difference for second proposed algorithm as already tested on outdoor and indoor video sequences.

6.4 References

1. Nawaz, M., Fatah, O.A., Comas, J. and Aggoun, A. (2012) "Extracting foreground in video sequence using segmentation based on motion, contrast and luminance", *Broadband Multimedia Systems and Broadcasting (BMSB), 2012 IEEE International Symposium on*IEEE, , pp. 1.

2. Stauffer, C. and Grimson, W.E.L. (1999) "Adaptive background mixture models for real-time tracking", *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*IEEE, .

3. Barron, J.L., Fleet, D.J. and Beauchemin, S.S. (1994) "Performance of optical flow techniques", *International journal of computer vision*, vol. 12, no. 1, pp. 43-77.

4. Olson, T. and Brill, F. (1997) "Moving object detection and event recognition algorithms for smart cameras", *Proc. DARPA Image Understanding Workshop*, pp. 205.

5. Kim, S.W., Yun, K., Yi, K.M., Kim, S.J. and Choi, J.Y. (2012) "Detection of moving objects with a moving camera using non-panoramic background model", *Machine Vision and Applications,* pp. 1-14.

6. Sen-Ching, S.C. and Kamath, C. (2004) "Robust techniques for background subtraction in urban traffic video", *Electronic Imaging 2004*International Society for Optics and Photonics, pp. 881.

7. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. and Rosenberger, C. (2008) "Review and evaluation of commonly-implemented background subtraction algorithms", *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* IEEE, , pp. 1.

8. Maddalena, L. and Petrosino, A. (2008) "A self-organizing approach to background subtraction for visual surveillance applications", *Image Processing, IEEE Transactions on,* vol. 17, no. 7, pp. 1168-1177.

9. Kelly, P., Ó Conaire, C., Monaghan, D., Kuklyte, J., Connaghan, D., Pérez-Moneo Agapito, J.D. and Daras, P. (2010) "Performance analysis and visualisation in tennis using a low-cost camera network", .

10. Migliore, D.A., Matteucci, M. and Naccari, M. (2006) "A revaluation of frame difference in fast and robust motion detection", *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*ACM, , pp. 215.

11. Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P. (1997) "Pfinder: Real-time tracking of the human body", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, no. 7, pp. 780-785.

12. Martínez-Martín, E. and del Pobil, A.P. (2012) *Robust motion detection in real-life scenarios,* Springer.

13. Sun, S., Wang, Y.F., Huang, F. and Liao, H.M. (2012) "Moving foreground object detection via robust SIFT trajectories", *Journal of Visual Communication and Image Representation*.

14. Zheng, J., Wang, Y., Nihan, N.L. and Hallenbeck, M.E. (2006) "Extracting roadway background image: Mode-based approach", *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1944, no. 1, pp. 82-88.

15. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. and Rosenberger, C. (2010) "Comparative study of background subtraction algorithms", *Journal of Electronic Imaging,* vol. 19, no. 3, pp. 033003-033003-12.

16. Jodoin, J., Bilodeau, G. and Saunier, N. (2012) "Background subtraction based on local shape", *arXiv preprint arXiv:1204.6326,* .

17. Zivkovic, Z. (2004) "Improved adaptive Gaussian mixture model for background subtraction", *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*IEEE, , pp. 28.