A COMPARATIVE EVALUATION OF FOREGROUND / BACKGROUND SEGMENTATION ALGORITHMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUHAMMET PAKYUREK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

A COMPARATIVE EVALUATION OF FOREGROUND / BACKGROUND SEGMENTATION ALGORITHMS

Submitted by **MUHAMMET PAKYUREK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ____________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ____________________
Head of Department, **Electrical and Electronics Eng.**

Prof. Dr. Gozde Bozdagi Akar ____________________
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members

Prof. Dr. Aydın Alatan ____________________
Electrical and Electronics Eng. Dept., METU

Prof. Dr. Gözde Bozdağı Akar ____________________
Electrical and Electronics Eng. Dept., METU

Prof. Dr. Kemal Leblebicioğlu ____________________
Electrical and Electronics Eng. Dept., METU

Assoc. Prof. Dr. Ilkay Ulusoy Parnas ____________________
Electrical and Electronics Eng. Dept., METU

Halil İbrahim Cüce, M.Sc. ____________________
İLTAREN, TÜBİTAK

Date: 11.09.2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all referenced material and results that are not original to this work.

Name, Surname: Muhammet Pakyurek

Signature:

ABSTRACT

A COMPARATIVE EVALUATION OF FOREGROUND / BACKGROUND SEGMENTATION ALGORITHMS

Pakyurek, Muhammet

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Gözde Bozdağı Akar

September 2012, 77 pages

Foreground Background segmentation is a process which separates the stationary objects from the moving objects on the scene. It plays significant role in computer vision applications. In this study, several background foreground segmentation algorithms are analyzed by changing their critical parameters individually to see the sensitivity of the algorithms to some difficulties in background segmentation applications. These difficulties are illumination level, view angles of camera, noise level, and range of the objects. This study is mainly comprised of two parts. In the first part, some well-known algorithms based on pixel difference, probability, and codebook are explained and implemented by providing implementation details. The second part includes the evaluation of the performances of the algorithms which is based on the comparison between the foreground background regions indicated by the algorithms and ground truth. Therefore, some metrics including precision, recall and f-measures are defined at first. Then, the data set videos having different scenarios are run for each algorithm to compare the performances. Finally, the performances of each algorithm along with optimal values of their parameters are given based on f measure.

Keywords: Background – Foreground Segmentation, Mixture of Gaussians, Codebook Background Modeling, Precision – Recall Measures, Ground Truth Analysis

ÖZ

ARKA PLAN ÇIKARMA ALGORITMALARININ UYGULAMALARI VE PERFORMANS DEGERLENDIRMELERİ

Pakyürek, Muhammet

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Gözde Bozdağı Akar

Eylül 2012, 77 sayfa

Ön Plan – Arka Plan ayırma işlemi, uzun süre hareketsiz kalan objelerin sahnenin geri kalanından ayrıldığı bir görüntü işleme prosesidir. Bu işlem bilgisayarlı görme uygulamlarında çok önemli rol oynamaktadır. Bu çalışmada bir çok Ön Plan – Arka Plan ayırma algoritmaları, algoritmaların belli zorluklara olan hassasiyetini görmek için kendi kritik parametreleri değiştirilerek analiz edilmiştir. Bu zorluklar, ortamın ışık seviyesinin değişimi, kameranın bakış açısının değişimi, görüntülerdeki gürültü seviyeleri ve nesnelerin kameraya olan uzaklıkları gibi sıralanabilir. Tez çalışması temel olarak iki kısımdan oluşmaktadır. Birinici kısımda, iyi bilinen pixel farkları, olasılıksal ve codebook tabanlı birçok algoritma uygulama detayları da verilerek açıklanmış ve uygulanmıştır. İkinci kısımda ise, Zemin Doğruluk analizindeki ön plan - arka plan datalar ile algoritmalardaki ön plan - arka plan dataların karşılaştırılması esasına dayanan bir performance değerlendirmesi yapılmıştır. Bu yüzden Kesinlik – Hatırlama Ölçekleri, f-ölçümlerini bulunduran bazı metriklere ihtiyaç duyulmaktadır.

Algoritmaların performanslarını karşılaştırmak için değişik senaryolar içeren video data setleri bu metrikler çerçevesinde herbir algoritma için analiz edilmiştir. Son olarak, Algoritmalarin performanslari f ölçümüne göre optimal parametreleri birlikte verilmektedir.

Anahtar Kelimeler: Ön Plan – Arka Plan Çıkarımı, Gaus Dağılımlarınını Karışımı, Codebook Arka Plan Modelleme, Kesinlik – Hatırlama Ölçekleri, Zemin Doğruluk Analizi

To my family

ACKNOWLEDGEMENTS

First of all I would like to thank my thesis advisor Prof. Dr. Gözde Bozdağı Akar very much for her fruitful and endless help, guidance, encouragement and support during my graduate studies.

I am grateful to Halil İbrahim Cüce for his endless help, motivation and support.

I am thankful to my company TÜBİTAK İLTAREN Inc. for letting and supporting my thesis.

I should say that Erdinç Tatar have always supported me during my life and this thesis study. I am so lucky that I have such a brother.

I also express my gratitude to TÜBİTAK BİDEB "National Scholarship Program for MSc Students" for supporting me.

Finally, I would like to thank my family for their love, support and patience over the years. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

Background subtraction is the first and one of the most vital parts of autonomous vision system used in visual surveillance, motion detection applications and human-computer interaction systems. Basically, background subtraction procedure means the comparison of current frame with reference background model. If a pixel in the current frame is matched to the background model, it is classified as background. Otherwise, it is a foreground pixel. After this process, the mask showing only foreground objects are acquired for object analysis process.

Earlier, the background subtraction process was performed by the so primitive methods that were not able to cover the whole foreground pixels. However, as the challenges in background subtraction such as illumination changes, camera noise, non-static backgrounds, shadows and weather conditions (rain, snow) arise, these methods remain inadequate to handle these kinds of difficulties. Therefore, more complicated and effective algorithms probabilistic methods, Mixture of Gaussians and Codebook Background Modeling are designed for creating for a more robust and adaptive background model.

The performance evaluations of the background subtractions are indispensable to calculate their accuracy and to find the ideal parameters for optimal results. There are two main concepts on performance evaluation [3]. The first concept is Ground-truth (GT) which is based on manually annotations of video foreground objects [4] [5]. In this methodology, all foreground objects are defined manually frame by frame. The other concept is not-based on Ground-truth (NGT) whose annotations are created automatically when the algorithms are operating. However, due to the difficulty on defining a criterion for good subtraction performance, it has been rarely used by video-surveillance community [6] [7].

1.1 Research Objectives

The main goal of this study is to deeply understand the responses of background subtraction algorithms against their own parameter changes and the different videos which contain shadows, periodic like motion, camera noise and non-static background. The specific objectives can be listed as follows:

1. A variety of background / foreground segmentations algorithms is searched in academic literature. These algorithms are learned intensively and understood deeply. Some innovations are made to improve their performance.
2. These algorithms are evaluated by a systematic methodology such as ground truth analysis.
3. The effects of the changing parameters on the performance of the algorithms are observed.

1.2 Thesis Organization

The organization of the thesis and the contents of the following chapters are as follows:

Chapter 2 introduces background subtraction methods including primitive methods and advanced ones. Firstly, primitive methods are described to understand the concept of background subtraction process. Then, it mentions more complicated and robust background subtraction algorithms. While describing these methods, a number of details for the implementation of them are given. Finally, post processes which are used in background subtraction algorithms to get better performance are described.

Chapter 3 gives a brief information about the Ground Truth Analysis and then, video sequences which have used for this study are classified and evaluated regarding some criteria such as noise level, long or short range video, shadows of foregrounds, slowly moving foreground objects and complexity of background (multi model or not) . After the classification of the videos, precision – recall table and figures will be exhibited.

Chapter 4 firstly summarizes the whole studies. It also presents the conclusions and observations made throughout the thesis study. Tentative future works are also explained briefly.

CHAPTER 2

BACKGROUND SUBTRACTION

In this chapter, Background Subtraction algorithms are evaluated in four different groups. These groups named as Pixel difference based methods, Probability based methods, Codebook based methods and other methods. Some methods in these groups are easy to implement while others are very complicated methods. Furthermore, a number of algorithms in these groups have a capability to handle multimodal background whereas other approaches are not able to deal with multimodal background. Numerous algorithms from these groups are able to cope with camera noise, illumination changes, but others cannot handle such problems. After the algorithms, a few post-processing methods such as morphological operations and connected component labeling to improve the performance of the subtraction algorithms are mentioned.

2.1 Background Subtraction Methods in Literature

In this part, background foreground subtraction algorithms which have searched for during this thesis study are mentioned. While dividing them into four main groups, their main approaches to the segmentation problem are regarded.

2.1.1 Pixel Difference Based Methods

The main idea of such methods is that if there is a difference between consecutive two frames, it means that there is motion there. But these kinds of algorithms cannot remember the foregrounds after a few frames.

2.1.1.1 Frame Differencing

Frame differencing is the one of the most primitive background subtraction methods. The main idea behind this algorithm is to search for temporal changes in video sequences. This method has been widely used in background subtraction studies for many years since the implementation of this method is quite easy. However, it cannot remember the history of pixel. Therefore, camera noise and distortions have a huge impact on the performance of the algorithm. Today, it can be also used as a reference method to compare the performance with the other methods' performances.

$$foreground(x) = \begin{cases} 0, & I(x,y,t) - I(x,y,t-1) \leq Th \\ 1, & I(x,y,t) - I(x,y,t-1) > Th \end{cases} \qquad 2.1$$

Where $\mathrm{I(x,y,t)}$ $and\ I(x,y,t-1)$ are current and previous illumination values of the pixel in (x, y) position. Th is predefined threshold for determining the background foreground. Considering the equation 2.1, this method apply frame differencing between two consecutive frames pixel by pixel. If the subtraction is greater than predefined threshold, it is a foreground pixel. Otherwise, it is labeled as background pixel. This method has the capability of handling non-static backgrounds and instantaneous motion of foreground objects. On the other hand, Figure 2-1 shows that it cannot detect overlapped part of two foreground objects in two consecutive frames. Moreover, if the foreground object suddenly stops moving, it is not able detect this object as foreground anymore.

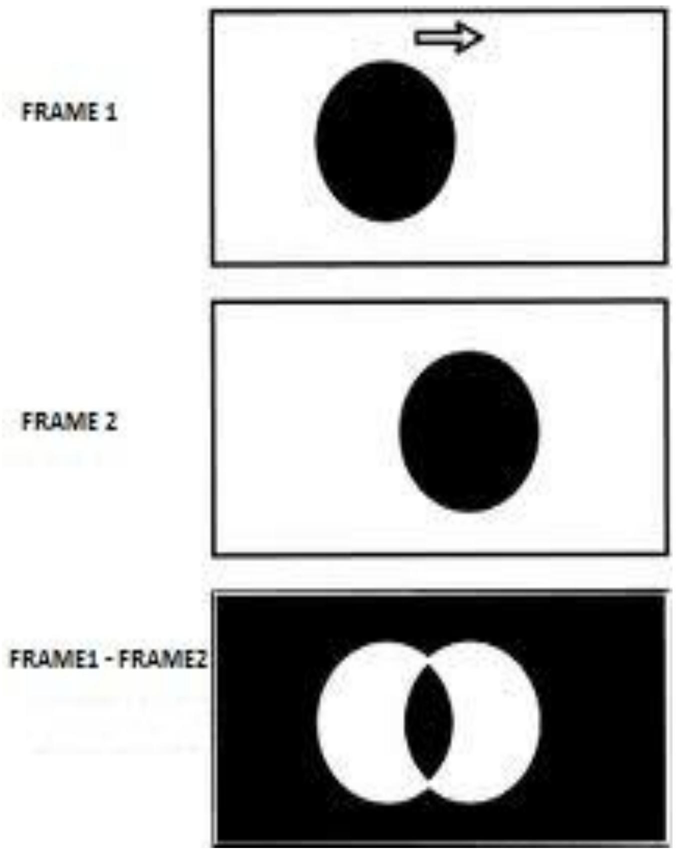


Figure 2-1: Frame Differencing between two consecutive frames and the result of this process [8]

2.1.2 Probability Based Methods

The main advantage of this type of algorithms is that they are easy to model the system since they don't need to know the mathematical model of the system. On the other hand, they are very complicated to implement. Furthermore, their time and operation cost are higher than other algorithms.

2.1.2.1 Kernel Density Estimation

In this method [29], the probability density function of each pixel is calculated thorough taking the average of the effect of a number of kernel functions (Gaussian pdf.) whose mean values are most nearly at pixel value for the last N previous frames.

$$Pr(x_t) = \frac{1}{N}\sum_{i=1}^{N} K(x_t - x_i) \qquad 2.2$$

These functions parameters are updated for each new incoming frame. A pixel is labeled as a background pixel, if the probability of the pixel developed by the corresponding kernel functions is higher than the threshold τ. Otherwise; it is named as a foreground pixel.

2.1.2.2 Running Gaussian average

Azarbayejani and his colleagues [9] present a statistical background modeling methods which has a dynamically changing threshold for each pixel. This method tries to fit a Gaussian distribution **(μ, σ)** over the histogram of each pixel so; the PDF of background model is obtained for each pixel. First of all, it is controlled that whether the current pixel value $\boldsymbol{x_t}$ can be represented by the corresponding Gaussian distribution satisfying the equation 2.3.

$$|x_t - \mu_t| \leq 2.5 * \sigma_t \qquad 2.3$$

Where μ_t, σ_t are updated mean and variation of the corresponding Gaussian distribution respectively. If $\boldsymbol{x_t}$ is matched with a Gaussian distribution, the parameters of the corresponding Gaussian distribution is updated as follows and it is labeled as background.

$$\mu_t = (1 - k) * \mu_{t-1} + k * x_t \qquad 2.4$$

$$\sigma_t^2 = (1-k) * \sigma_{t-1}^2 + k * (x_t - \mu_t)^2 \qquad 2.5$$

Where k is learning rate. If k is too high, recent pixel values have more impact on the model. If it is too low, the effect of last pixel value is too little.

If there is no match with the background model, it is labeled as a foreground pixel. This procedure is repeated for each pixel.

The running average is advantageous in terms of increasing the effect of recent changes. Furthermore, in this method, when the foregrounds stop moving, they cannot be detected as background objects since the history of background is kept in Gaussian distribution. On the other hand, when a new background model which requires a new Gaussian distribution comes, it is still detected as foreground since the number of Gaussian distribution is only one.

2.1.2.3 Mixture of Gaussians (MOG)

Stauffer and Grimson [16] present an adaptive and probabilistic methodology for background subtraction procedure. Numerous problems such as illumination changes, quasi periodic motions, varying weather conditions, camera noise cannot be solved by Running Gaussian average. Therefore, mixture of Gaussian methods is developed to cope with these problems. In this method, it is assumed that each pixel behaves independent from the pixels in neighborhood. The updates of the Gaussian distribution and its' weight is similar to Running Gaussian Average's equations 2.3, 2.4 and 2.5. This method generates a mixture of Gaussian distributions for each pixel and uses an on – line approximation to update the model. After that, the evaluation of these adaptive Gaussian distribution is performed to find out which combination of these distributions most likely represents the background model.

A number of studies have been published after this study since there are some vital problems with the algorithm such as initial weight calculation, shadow elimination. [15], [13] propose reasonable solutions to them. For the initial weight calculation, instead of after the first iteration, Gaussian Distributions' weights are normalized after the nth iteration because at the first iteration, the modeled pixel may belong to a foreground object. The weight of the corresponding Gaussian distribution gets a too much high weight score instantly after the normalization process even if it appear for only one time. Then, when the background model starts to be appeared by time, the algorithm counts this pixel as foreground until the weight of the corresponding Gaussian distribution is high enough to be included by background model. The other problem of shadow elimination is solved by Horprasert and his partners [17]. They introduce an efficient methodology in calculation. Basically, it is based on chromatic and brightness components for each pixel. These components are calculated by comparing non-background pixel with the corresponding pixel of the updated background model. If the differences in both components are within the range of predefined threshold, it is regarded as shadow pixel by the algorithm. The last problem of this algorithm is revealed by this thesis study. The problem is about the update of the weight and variation of the Gaussian distributions. For a long period of time, when constantly the same value for the same pixel comes, the Gaussian Distributions representing these background pixels is gaining too much larger weight and having lower deviation and variation. Therefore, even a new value with a too small difference cannot be modeled by these Gaussian Distributions. The algorithm has to create a new Gaussian for the new incoming pixel with small weight and high variation. However, it is not able to be included by the background model until its weight reaches a threshold limit. During this period, the pixel with this new value is not be detected as a background pixel. In order to solve this problem, when the variations of Gaussian Distributions get close to critical point, it is not be updated in descending direction anymore.

The details of the proposed modification along with Mixture of Gaussian algorithm are given in Figure 2-2. The algorithm starts with the initialization of Gaussian distributions as given in Equations 2.6, 2.7 and 2.8

$$\omega_i = 0.05 \tag{2.6}$$

$$\sigma_i = 20 \tag{2.7}$$

$$\sigma_i^2 = 400 \tag{2.8}$$

The weights of Gaussian distributions are initially set into a low value since initially high value weight make the distribution to be included by the background model immediately. The variances and standard deviations of the Gaussian Distributions are adjusted to high values so; changes in a pixel are modeled better. Otherwise, all distributions would be used just for one background model. During the experiments, initial variance and deviation values are adjusted to see the effect of them on ground truth analysis. This part of the algorithm is applied only at the beginning of the implementation. After the initialization, the weights and deviations of the Gaussian distributions are updated as in the original form except for the normalization of the weights. This part is modified by taking into consideration of first L frame. As mentioned at the beginning of the Mixture of Gaussians, normalizing weights for the first frame of a few frames is not a good approach since these distributions are instantly included by background as a result of having a very high weight.

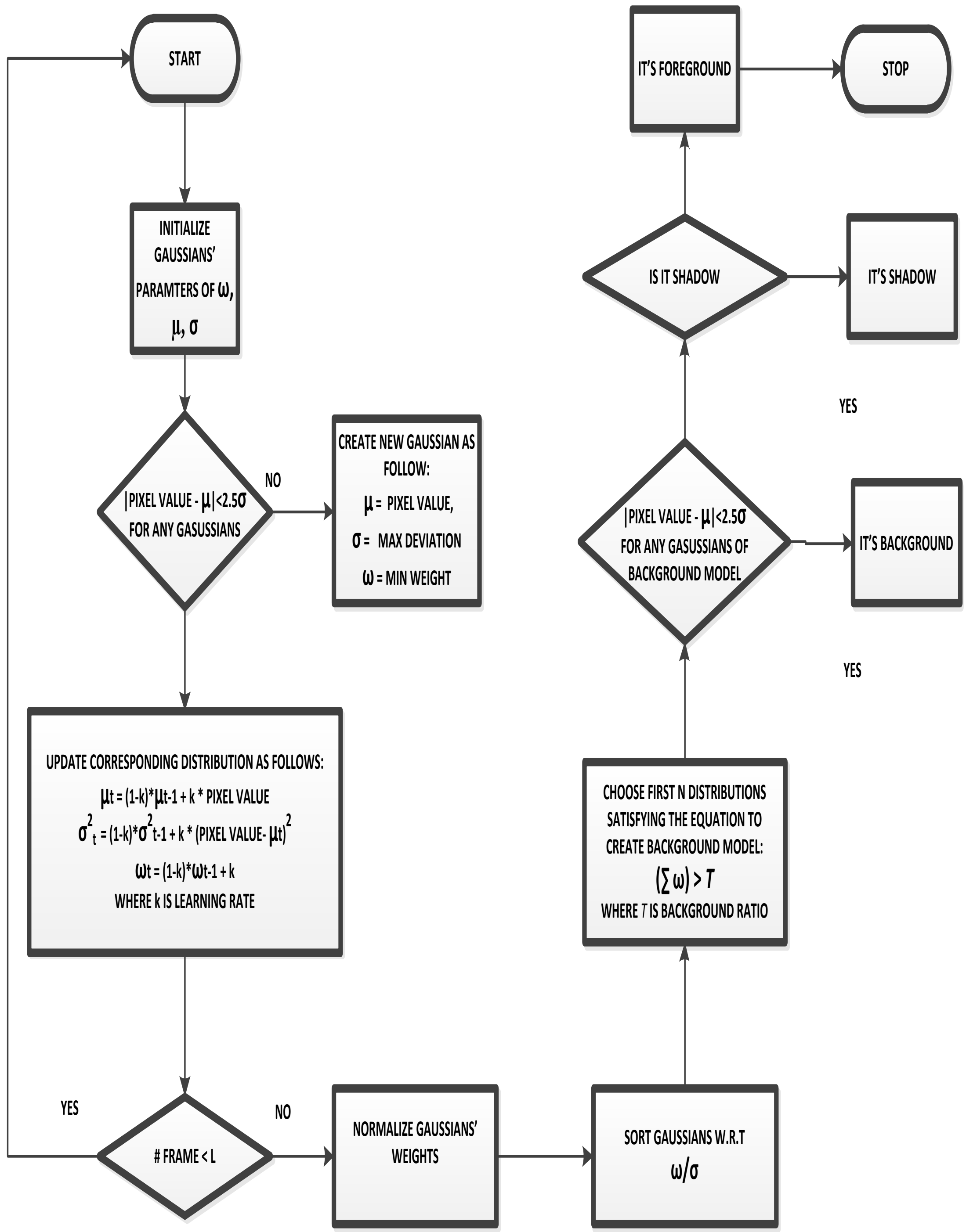


Figure 2-2: Flow Chart of Mixture of Gaussians Algorithm

When the first L frames are finished, it is checked whether the new incoming pixel is matched with any existing Gaussian distributions of the pixel. The criterion of matching with Gaussian distribution is as follows:

$$|x_{t,i} - \mu_{t,i}| \leq 2.5 * \sigma_{t,i} \qquad 2.9$$

Where $x_{t,i}$ is the value of i^{th} pixel and $\mu_{t,i}, \sigma_{t,i}$ are the mean and standard deviation of the updated Gaussian distribution for the i^{th} pixel at time t. If the new pixel value is matched, the parameters of the corresponding Gaussian distribution are updated as follows:

$$\mu_{t,i} = (1-\alpha) * \mu_{t-1,i} + \alpha * x_{t,i} \qquad 2.10$$

$$\sigma_{t,i}^2 = (1-\alpha) * \sigma_{t-1,i}^2 + \alpha * (x_{t,i} - \mu_{t,i})^2 \qquad 2.11$$

$$\omega_{t,i} = (1-\alpha) * \omega_{t-1,i} + \alpha * M_{t,i} \qquad 2.12$$

Where $\mu_{t,i}$,$\sigma_{t,i}^2$ and $\omega_{t,i}$ are the parameters of the updated Gaussian distribution of the i^{th} pixel at time t. $M_{t,i}$ becomes 1 when the corresponding Gaussian is matched, otherwise it is set to zero. If there is no match, a new Gaussian distribution is created by setting its' parameters as shown in Equations 2.6, 2.7 and 2.8.

After the update or creating a Gaussian distribution, the next step is the construction of the background model. The Gaussian distributions of a pixel are ordered with respect to $\frac{\omega}{\sigma}$ from the highest to the lowest. It means that The Gaussian with high weight and low deviation is most likely to be in the first ranks. Then, the first N distributions satisfying the Equation 2.13 are picked up to build a background model.

$$\sum_{i}^{N} \omega_{t,i} > T \qquad 2.13$$

For a new incoming pixel value, it is looked for that whether there is a matched Gaussian satisfying the Equation 2.9. If there is a match among the Gaussian distributions of the background model, it is classified as background. Otherwise, another criterion is necessary to decide that whether the pixel is shadow or foreground [15]. The equations for the criterion are as follows:

$$a = \min_{z}(I - zE)^2 \qquad 2.14$$

$$c = \|I - aE\| \qquad 2.15$$

Where E position vector of the pixel in (R, G, B) space, ‖E‖ is an expected chromaticity line. Equation 2.14 is for the calculation of brightness distortion; Equation 2.15 is for the calculation of the distortion in the chromaticity. If these distortions are lower than the thresholds, this pixel is labeled as shadow. Otherwise, it is categorized as foreground.

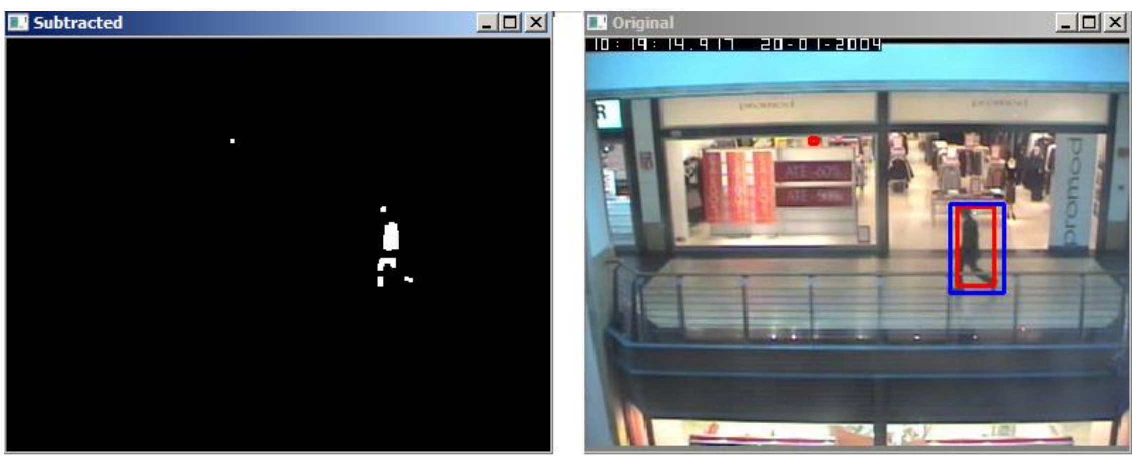


Figure 2-3: Output of the Mixture of Gaussians

2.1.3 Codebook Based Methods

2.1.3.1 Real Time Foreground – Background Segmentation using Codebook Model (Modified Codebook)

This study is conducted by Larry Davis and his research group [1]. The codebook background modeling method is kind of pixel wise background subtraction algorithm in which each pixel is quantized to find the best match of the codewords representing a compact form of background model of long image sequence. Moreover, this algorithm provides a solution to handle structural background changes such as periodic motion over a long period of time under the circumstance of limited memory.

When the algorithm is compared with other Background-Foreground subtraction algorithms, it runs faster, and it is also more efficient in memory. Furthermore, variety of problems such as moving backgrounds, illumination changes and camera noises can be handled when the parameters are set accordingly.

In addition to the basic algorithm [2], two features whose names are layered modeling detection and adaptive codebook updating are added in modified codebook algorithm to handle the background changes.

In general, the algorithm is divided into the two main parts which are background modeling and foreground detection respectively. In the background modeling, the codebook is created which includes one or more codewords for each pixel. This codewords are included if they satisfy the color distortion and brightness limit. The second part of the algorithm is the foreground detection part where moving objects are separated from the background. The detection process is performed by evaluating the difference between current image and background model in pixel wise. To be more clear, the difference means that the distortion in color space and variation in brightness. If the incoming pixel satisfies these two conditions, it is labeled as background model. Otherwise, it is counted as foreground.

In order to give more detail about the implementation of the algorithm, each part of the algorithm is explained step by step below.

The algorithm of the whole method as follows:

i. Training Part:

The first step of the algorithm is codebook construction which is given in Figure 3-3. This flow chart is only for one pixel. Therefore, it must be applied for all pixels. Firstly, the parameters of codewords which are $\boldsymbol{v_m}$, $\boldsymbol{\hat{I}}$, $\boldsymbol{\check{I}}$, $\boldsymbol{\lambda}$, $\boldsymbol{p}, \boldsymbol{q}$ are initialized by a default value. $\boldsymbol{v_m}$: is a vector which holds the mean of the R, G, and B value of the m^th^ codeword. $\hat{I}$: is the highest brightness value of a codeword. $\check{I}$: is the lowest brightness value of a codeword. λ: is maximum negative run-length of a codeword. p: holds the first access time of a codeword. q: holds the last access time of a codeword. x_t: is a vector which holds red, green, blue values of pixels. f: shows the number of occurrences of a codeword. After the initialization, brightness of the each pixel is calculated as follows:

$$I^2 = R^2 + G^2 + B^2 \qquad 2.16$$

Where R, G, B are red, green and blue values of the pixels. Thirdly, it is searched whether there is match for the pixel or not with the corresponding codewords. This matching procedure is conducted by using $colordist(\boldsymbol{x_t}, \boldsymbol{v_m})$ and $brightness\left(\boldsymbol{I}, \langle \boldsymbol{\hat{I}}, \boldsymbol{\check{I}} \rangle\right)$. $colordist(\boldsymbol{x_t}, \boldsymbol{v_m})$ and $brightness\left(\boldsymbol{I}, \langle \boldsymbol{\hat{I}}, \boldsymbol{\check{I}} \rangle\right)$ functions are defined as follows

$$\|x_t\|^2 = R^2 + G^2 + B^2 \qquad 2.17$$

$$\|v_i\|^2 = \bar{R}^2 + \bar{G}^2 + \bar{B}^2 \qquad 2.18$$

$$\langle x_t, v_i \rangle^{\mathbf{2}} = (\bar{R}R + \bar{G}G + \bar{B}B)^{\mathbf{2}} \qquad 2.19$$

Where $\overline{\mathrm{R}}$, $\overline{\mathrm{G}}$ and $\overline{\mathrm{B}}$ are mean R,G,B values of previous frames for the v_i of a codeword.

The color distortion δ is calculated by following equations

$$p^2 = \|x_t\|^2 (\cos\theta)^2 = \frac{\langle \boldsymbol{x_t}, \boldsymbol{v_i} \rangle^2}{\|\boldsymbol{v_i}\|^2} \qquad 2.20$$

$$colordist(x_t , v_m) = \delta = \sqrt{\|x_t\|^2 - p^2} \qquad 2.21$$

$$I_{low} = \propto \hat{I} \qquad 2.22$$

$$I_{hi} = min\left\{\beta\hat{I}, \frac{\check{I}}{\propto}\right\} \qquad 2.23$$

$$brightness\big(I, \langle\hat{I}, \check{I}\rangle\big) = I_{low} < I < I_{hi} \qquad 2.24$$

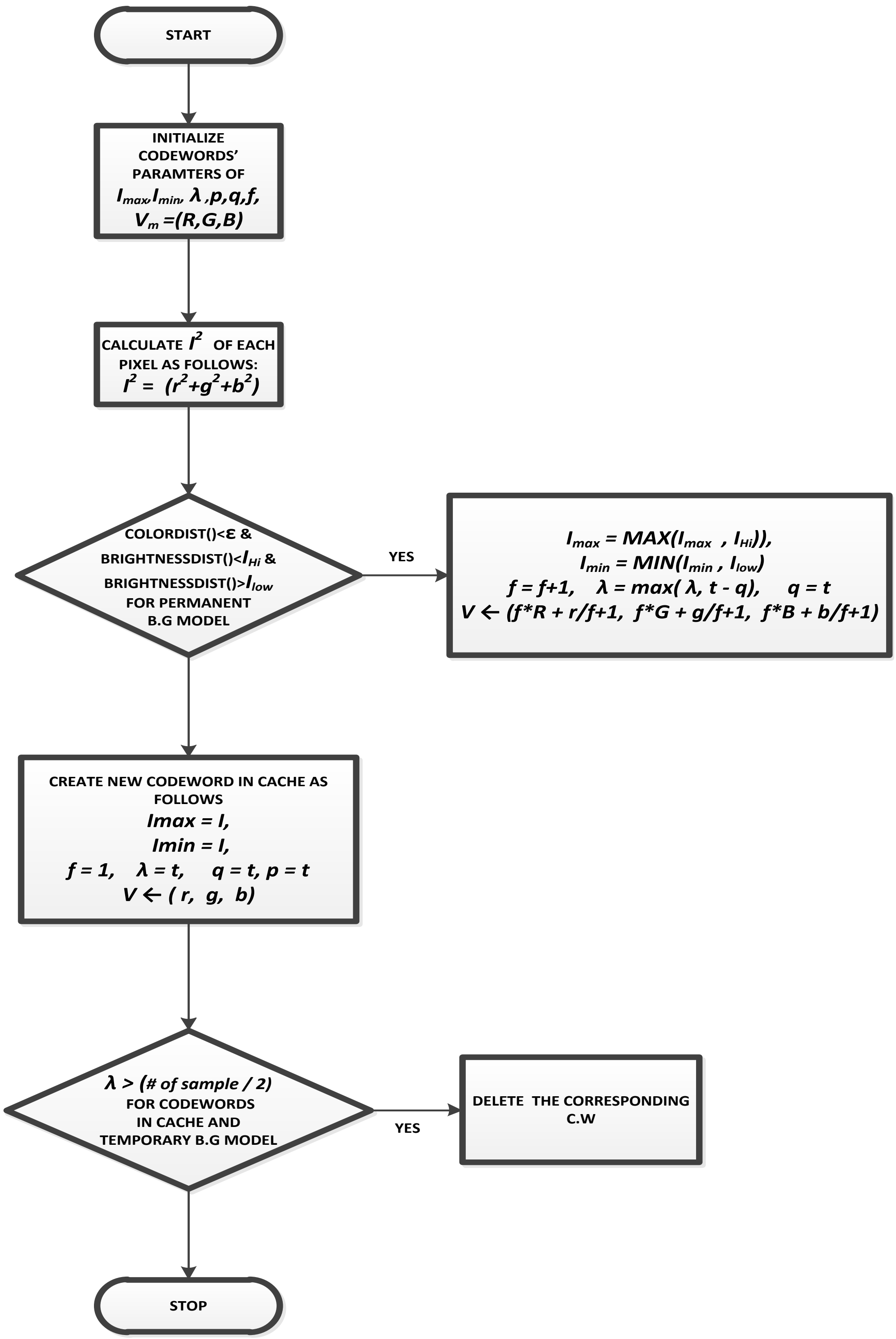


Figure 2-4: Codebook construction part

If there is a match in the corresponding codewords, updating procedure of the codeword is as follows

$$v_m \leftarrow \left\{\frac{f\bar{R} + R}{f+1}, \quad \frac{f\bar{G} + G}{f+1}, \quad \frac{f\bar{B} + B}{f+1}\right\} \tag{2.25}$$

$$\check{I} = min\{\check{I}, I\} \tag{2.26}$$

$$\hat{I} = max\{\hat{I}, I\} \tag{2.27}$$

$$f = f + 1 \tag{2.28}$$

$$\boldsymbol{\lambda} = max\{\lambda, t - q\} \tag{2.29}$$

$$q = t \tag{2.30}$$

Where t represents the time in frame domain. If there is no match with codewords, a new codeword is added into the codebook by increasing the number of codewords. The new codeword's parameters are set as follows

$$v_m \leftarrow \{R, G, B\} \tag{2.31}$$

$$\check{I} = I \tag{2.32}$$

$$\hat{I} = I \tag{2.33}$$

$$f = 1 \tag{2.34}$$

$$\boldsymbol{\lambda} = t - 1 \tag{2.35}$$

$$q = t \tag{2.36}$$

$$p = t \tag{2.37}$$

Finally, the codewords in which $\boldsymbol{\lambda} < \boldsymbol{N}/\boldsymbol{2}$ are eliminated from the codebook since they are assumed that they belong to foreground model.

ii. Background Subtraction Part:

As shown in Figure 2.3, the brightnesses of the pixels are calculated like in the construction part. Secondly, match of pixel with background codewords is searched thorough using $\mathrm{colordist}(\mathrm{x_t}\,,\ \mathrm{v_m})$ with different threshold and $\mathrm{brightness}\big(\mathrm{I},\ \langle\hat{\mathrm{I}},\ \check{\mathrm{I}}\rangle\big)$. However, in this case, the search is conducted in not only permanent background model but also temporary background model and the cache that holds the instant models. If there is a match, the matched codeword's parameters are updated similarly in the construction part. However, if this match belongs to the cache, the corresponding pixel is classified as foreground.

$$\lambda < T / 2 \qquad 2.38$$

Where T is predefined threshold for the maximum negative run-length. Like in the construction part, the codewords of temporary background model and cache are eliminated when they cannot satisfy the equation 2.38. In other words, if the maximum run length of codewords exceeds predefined T time threshold, these codewords are excluded from codebook background model. If the remaining codewords stay in the cache satisfying the Equation 2.39, they are inserted into the temporary background.

$$f < N \qquad 2.39$$

Where N is a predefined threshold for the number of occurrence,

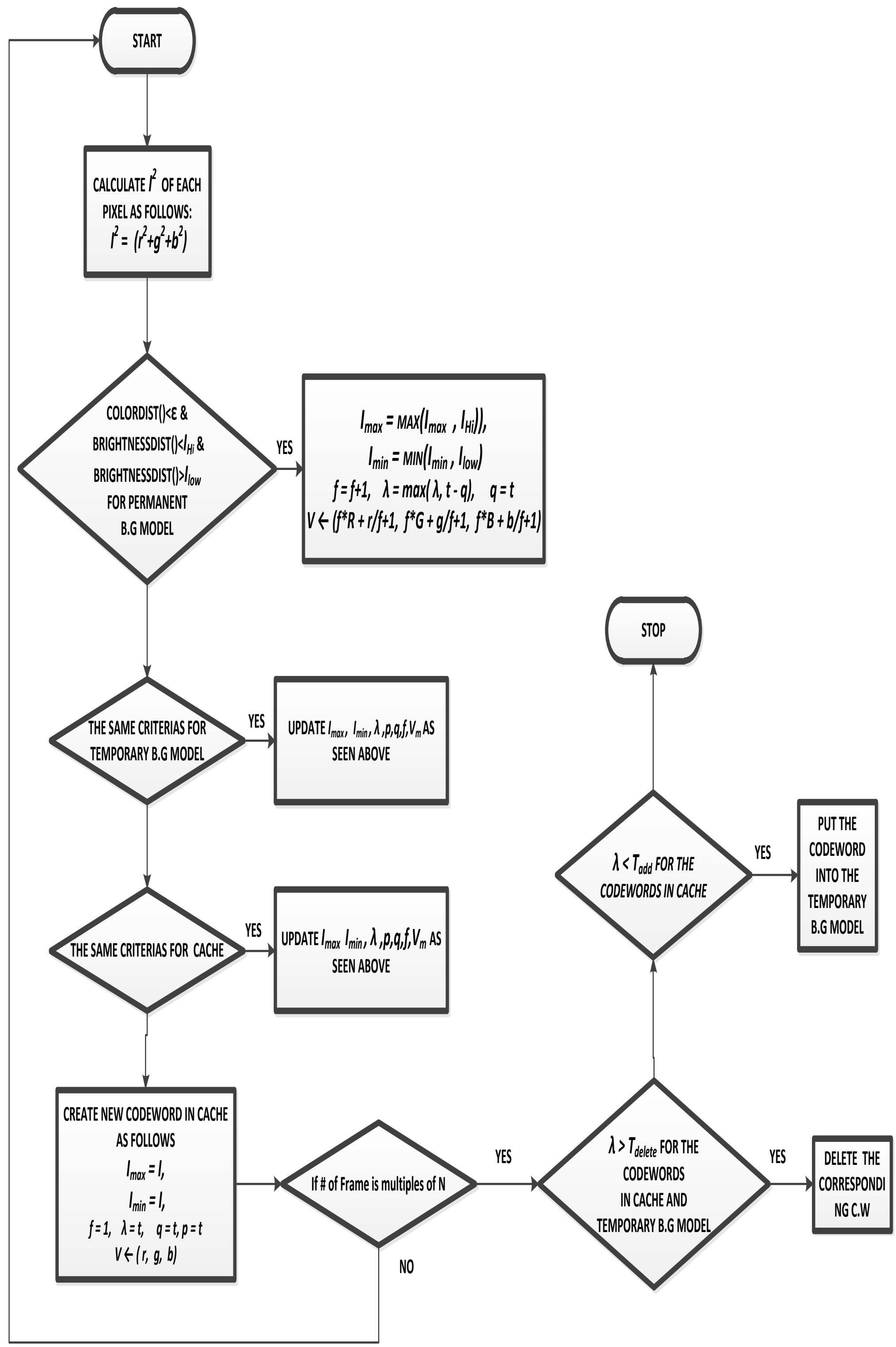


Figure 2-5: Subtraction Part of Codebook Model

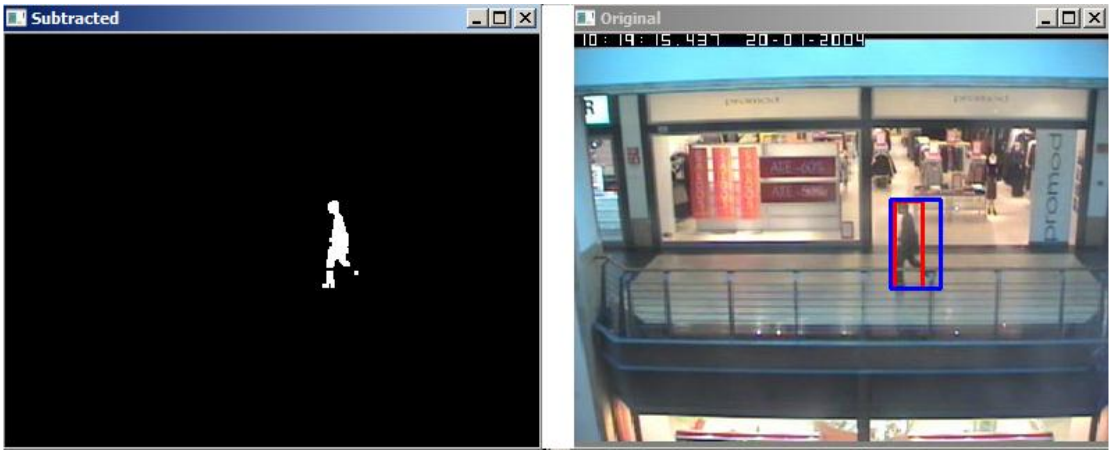


Figure 2-6: Output of Modified Codebook

2.1.4 Other Methods

2.1.4.1 Eigen Backgrounds

Eigen background is a non-pixel wise method [37]. It was proposed by Oliver *et al.* [38]. In order to model the background of the sequences an Eigen space is used. The most important advantage of this method is the ability of learning the background model from any video sequences without constraints. In fact it is able to develop background model from video containing moving objects initially. Eigen background method regards neighboring pixels while developing background model for each pixel. Therefore, it has a more comprehensive definition for background modeling.

2.2 Post - processing methods

Post processing methods is performed after the background subtraction process. The main advantage of these methods is to improve the performance of the background - foreground segmentation.

2.2.1 Morphological Operations

Serra [10] presented a number of morphological operations in his book. The most basic of these operations are named as dilation and erosion. Such operations are implemented on binary images generally. These operations are used in removing noise, segmentation of objects and complete the missing parts of objects. Furthermore, in order to find image gradients, morphological operators are also applied. During this thesis study, $\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$ (3x3) kernel is used for both the dilation and erosion processes.

2.2.1.1 Dilation

Dilation is an operation which is the result of the convolution of the images and a kernel. The shape and size of the kernel can be varied depending on the implementation. The shape is generally chosen a small solid square or disk with the anchor point which is located at the center of the kernel. In the dilation process, the whole image is scanned by this kernel. For the whole small regions of images, the kernel finds the maximal pixel in the overlapped region and then, the pixel under the anchor point is replaced with that maximum value. This process ends up with that bright regions grow within an image.

Figure 2-7: Original Binary Image (left) and Processed Binary Image by the Dilation operation (right)

As shown in Figure 2-7, the Dilation process plays more significant role especially in smaller bright regions since fewer pixels are bright in such regions. When small regions are intended to make larger, the dilation process is applied into desired regions.

2.2.1.2 Erosion

Erosion process is converse version of the dilation. On the contrary to the dilation, the kernel finds the minimum pixel value and then, this value is assigned into the pixel where anchor point hits on scanned region. The main effect of this operation is to erode away the boundary of the foreground pixel. When small region such as noises is desired to be released, this process puts on a good performance.

Figure 2-8: Original Binary Image (left) and Processed Binary Image by the Erosion operation (right)

As shown in Figure 2-8, the effects of erosion operation are seen more clearly. On the left down corner of the original image, there are a number of small bright regions but, these all regions are disappeared when the erosion operation is employed.

2.2.2 Connected Component Labeling (CCL)

Connected component labeling is a methodology to segment objects using the divisions between regions by labeling them as different objects. It is widely used in computer vision studies to find connected regions on an image. Before the further detail for the connected component labeling, some brief explanations for the Connectivity concept must be mentioned. The Connectivity plays a significant role in labeling. Connected component labeling connects the pixels in neighborhood based on the connectivity and their relative values of pixels [27]. As shown in Figure2-9, 4-connected or 8-connected connectivity are implemented in image graphs [28]. The 8-connected connectivity is chosen in this study to decide more precisely.

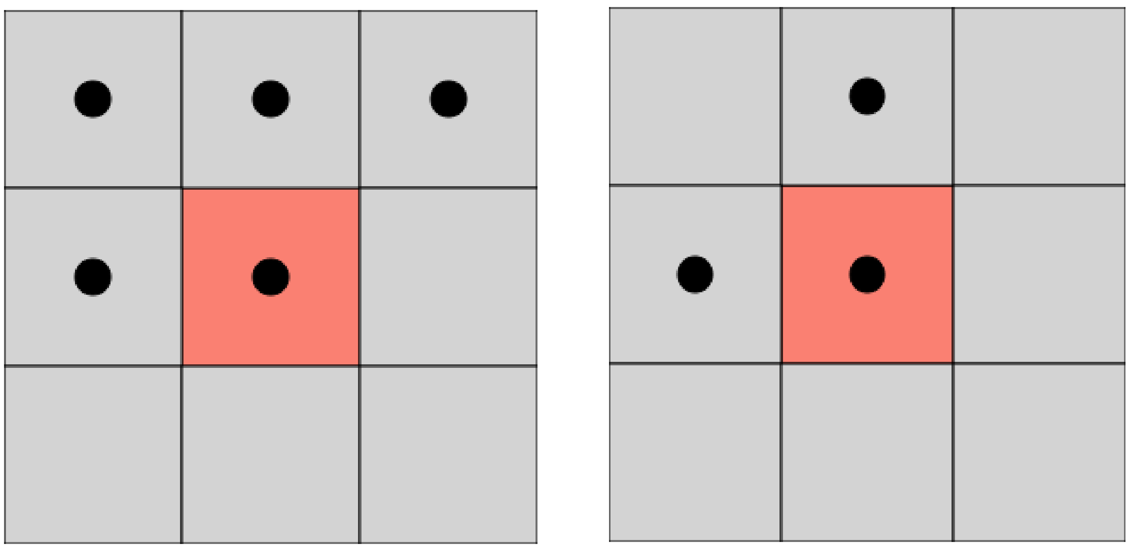


Figure2-9: 8-connected (left) and 4-connected (right) [27]

There are a variety of connected component labeling algorithms in academic literature. But they can be categorized in three main division named as two pass algorithm, sequential algorithm and one pass optimized algorithm. In the thesis, two - pass algorithm is used. Two – pass algorithm is relatively easy to understand and implement

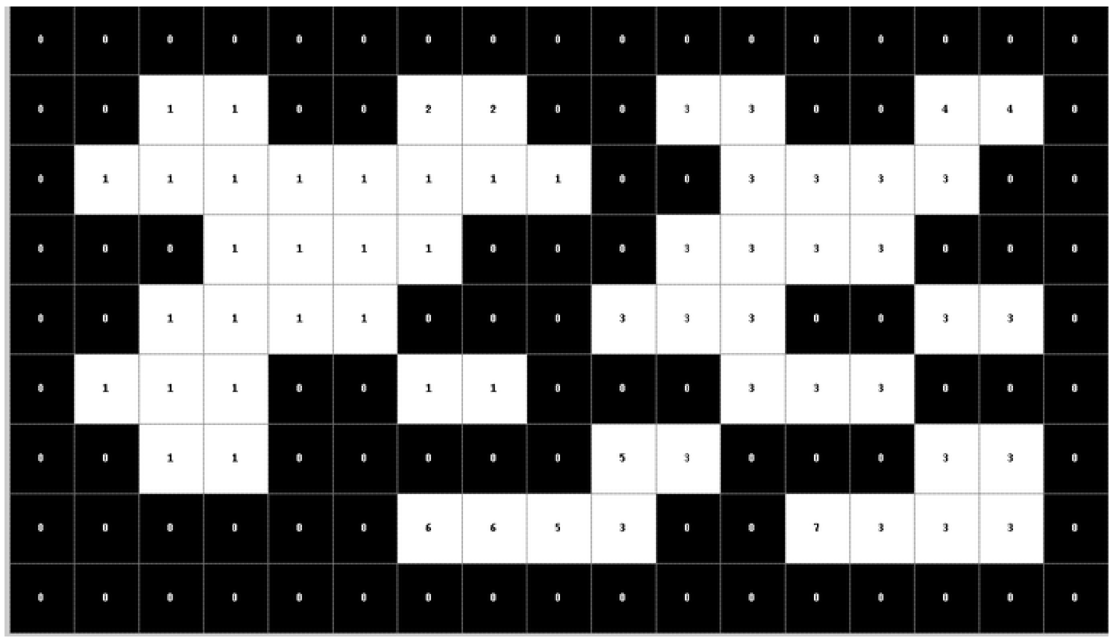


Figure 2-10: The results of the one pass part [27]

when it is compared with other algorithms [11]. Firstly, the algorithm controls the connectivity relationship within neighboring pixels. While applying this procedure, the label ID of the pixel is defined by finding the labeling group with the smallest label id.

If there is any equivalence between neighboring labels, it is stored in the equivalence table that all equivalence relations between the corresponding labels are hold. Now, it has label id to be connected with the other pixels which have the same label id. Up to this part, all procedures are called as first pass.

On the second pass, all related labels are resolved. In order to make this easy, the reflectivity and Floyd - Warshall (F - W) algorithm is applied.

a)	1	2	3	4	5	6
1		1				1
2	1					
3				1		
4			1		1	
5				1		
6	1					

b)	1	2	3	4	5	6
1	1	1				1
2	1	1				1
3			1	1	1	
4			1	1	1	
5			1	1	1	
6	1	1				1

Figure 2-11: binary equivalence matrix (left) and after the reflectivity and F –W algorithm is applied [18]

As seen in Figure 2-11, for each labels, all relations within other labels can be accessed thorough looking only one label which has the same equivalent label. After the reflectivity and F –W algorithm, the labels which have a relation are merged.

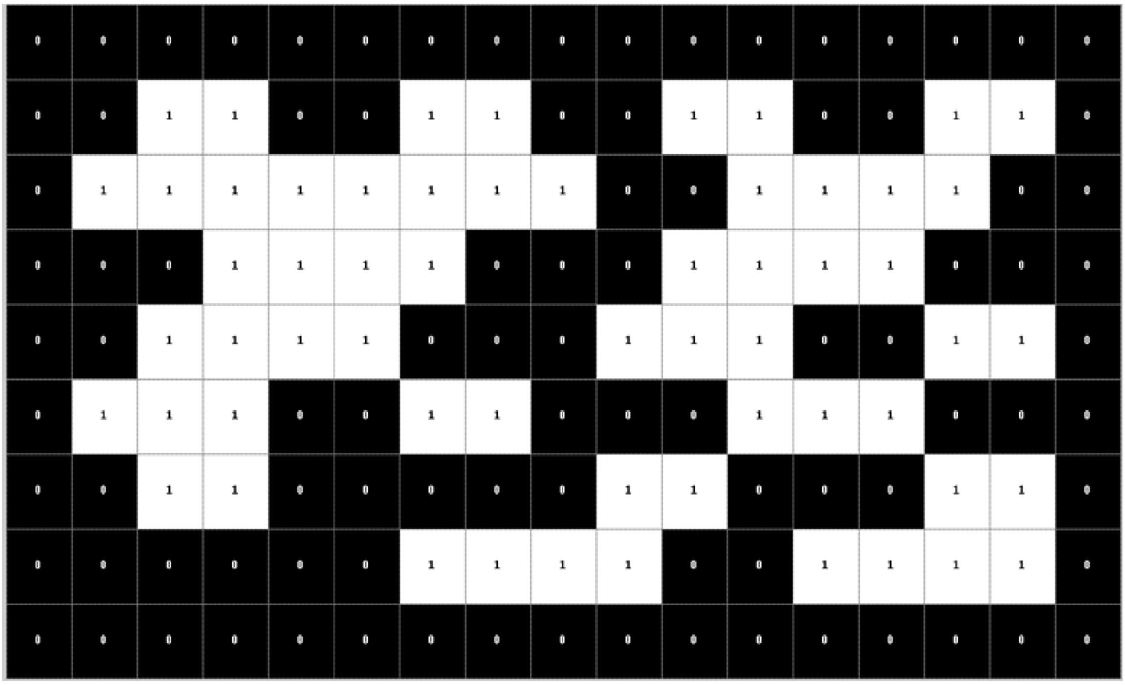


Figure 2-12: The result of the second pass part [27]

CHAPTER 3

EXPERIMENTAL RESULTS AND COMPARISONS

This chapter firstly presents the metrics used for the evaluation of the algorithms. Secondly, the results of ground truth analysis are exhibited for each algorithm. These results are explained by supporting with graphs and tables.

3.1 Definition of Metrics

All metrics that will be defined is used in the evaluation of the background subtraction algorithms. These metrics are defined as follows [31]:

- True Positive (TP): is a quantity which holds the number of correctly detected foreground pixels by the subtraction algorithms

- False Positive (FP): is associated with the number of pixels labeled as foreground objects incorrectly.

- True Negative (TN): is responsible for the number of pixels which are classified in background model by the algorithms.

- False Negative (FN): stands for the number of pixels detected as a background pixel falsely.

There are a number of analysis methods for the evaluation of the performance of the background segmentation algorithms [34]:

- Percentage correct classification
- Jaccard coefficient
- Yule coefficient

However, as mentioned above recall precision measures are employed in this study.

3.1.1 Recall

Recall is a measure of completeness that is related with how many ratios of real foreground objects are detected by the algorithms. It is defined as the number of true positives divided the total number of pixels belonging to real all foreground objects [31].

$$Recall = \frac{TP}{(TP + FN)} \quad 3.1$$

In other words, it can be written as follows:

$$Recall = \frac{number\ of\ correctly\ identified\ foregorund\ pixels}{number\ of\ foreground\ pixels\ in\ ground\ truth} \quad 3.2$$

3.1.2 Precision

Precision can be thought of a measure of accuracy and is calculated through dividing the number of correctly detected foreground items by total number of pixels categorized as foreground by the algorithms [31].

Precision calculation is performed as following formula:

$$Precision = \frac{TP}{(TP + FP)} \quad 3.3$$

$$Precision = \frac{number\ of\ correctly\ identified\ foregorund\ pixels}{number\ of\ foreground\ pixels\ detected\ by\ the\ algorithm} \quad 3.4$$

3.1.3 F measures

The F score is a kind of harmonic average which is calculated as follows [39]:

$$F_1 = 2 * \frac{Precision * Recall}{(Precision + Recall)} \quad 3.5$$

Where an F_1score reaches its best score when both recall and precision values are high and close to each other and worst when both recall and precision values are low and close to each other. Therefore, in ground truth analysis, F_1score is used to see easily the most optimal Recall – Precision pair.

3.2 Data-Set Selection and Description

In this thesis study, mainly three different types of videos are used to evaluate the algorithms. As shown in Figure 3-1, Figure 3-2 and Figure 3-3, these three types of videos are categorized as videos captured from the front view, videos captured from the corner view and videos captured from the tower view.

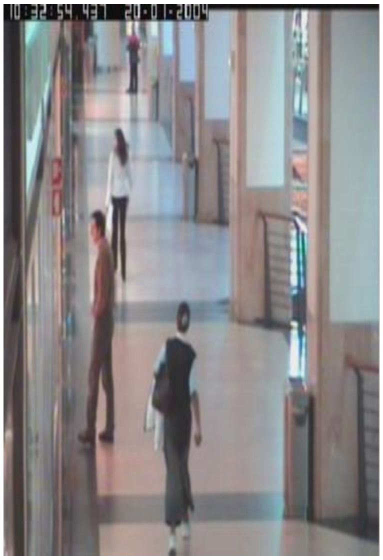


Figure 3-1: Corner view Videos

Figure 3-2: Front view Videos

Figure 3-3: Tower view videos

For each type of videos, four videos with ground truth are used. These videos are acquired from “http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1” [26].

Each video consists of 2756 frames on average. Each frame on the videos consists of 384 x 288 pixels. The frames’ height is 288 pixels and the frames’ width is 384 pixels. All of these videos are in RGB channels.

In order to capture and display frames, opencv2.3 version is used.

Even though quantitative results of the algorithms offer clear conclusions on each performance of the algorithms, a number of qualitative aspects of the videos and the algorithms make these results more reasonable. Furthermore, the effects of perspective for the videos are observed in this study

Table 3-1: Qualitative Aspects of the Videos

	Corner view	**Front view**	**Tower view**
Range	*Short*	*Long*	*Medium*
Noise Level	*Medium*	*Low*	*High*
# of Highly Illuminated Regions	*Medium*	*Low*	*High*
# of Shadowed Foreground Objects	*Medium*	*High*	*Low*
# of Non-static Background objects	*Medium*	*Low*	*High*

3.3 Experimental Results

Experiments are conducted on three different types of videos. The results on the tables are average of four different videos for each group. On the table, P1, R1 represent the precision and recall value of the foreground pixel. P0, R0 represent precision and recall value of the background pixel. While obtaining the P and R values for each parameter, other parameters take certain optimal values. By looking the tables, the individually effect of a parameter on the algorithm can be seen. In addition to table, precision recall graphs of foreground objects are given to visually understand relations between parameters and difficulties on the videos. Furthermore, precision recall graphs provide people who use these algorithms on these types of videos to choose most suitable parameters for their application by following the graphs. For each group of videos, a number of challenges on the videos are mentioned, then, tables and precision recall graphs are given. In order to understand the tables and graphs, some reasons for these results are given by explaining them in detail. Finally several snapshots from the outputs of the algorithms are shown in time domain to compare their performance visually.

3.3.1 Results of Videos Captured from the Corner Side

In this group of videos, the complication of foreground background model is high since videos are captured from a short range. Furthermore, the field of view of the camera is narrower than other groups of videos.

Figure 3-4: Corner Side Plane [26]

As seen in Figure 3-4, there is a huge perspective in Y axis. Due to the perspective in Y direction, the detection of the motion in this direction is quite difficult since great deals of the objects are overlapped on two consecutive frames. Another difficulty of this type of videos is that lack of background model. Since foreground objects have big sizes and slowly moving looks, some background pixels are released by the algorithms. When these objects leave these pixels, new incoming pixel counted as foreground even if it was labeled as background. These would be a tough problem for the algorithms which are not able to model new background and are able to model new background slowly. Another problem on these type video is that, at beginning of the videos, if such motions happen on the videos initially, the background is not modeled correctly. Generally, false alarm rate is high as result of all these mentioned above on these videos' performance evaluations. This would be a deal especially for the algorithms which require a training part to model the background such as codebook model and single Gaussian method.

Table 3-2: Recall – Precision results of MOG for the Videos captured from the corner side

MOG							
Std. Dev.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
5	**98,4225**	**74,3825**	**95,755**	**84,2875**	**97,07043**	**79,02584**	
10	**98,055**	**82,9325**	**97,2175**	**81,895**	**97,63445**	**82,41048**	
20	**98,0225**	**84,9675**	**97,4075**	**80,0325**	**97,71403**	**82,4262**	**
40	**96,065**	**88,485**	**98,69**	**59,5075**	**97,35981**	**71,1593**	
60	**91,535**	**99,1**	**99,9875**	**5,3225**	**95,57473**	**10,10242**	
80	**88,885**	**99,9725**	**100**	**0,0475**	**94,11547**	**0,094955**	
Learn. Rat.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,00001	**98,675**	**75,435**	**96,4775**	**86,79**	**97,56388**	**80,7151**	
0,00010	**98,6575**	**76,6325**	**96,55**	**85,875**	**97,59237**	**80,99092**	
0,00100	**98,39**	**81,4575**	**97,8275**	**83,5575**	**98,10794**	**82,49414**	**
0,01000	**95,045**	**94,8875**	**99,7775**	**44,61**	**97,35377**	**60,68828**	
0,10000	**92**	**97,585**	**99,975**	**9,115**	**95,82185**	**16,67268**	

Table 3-3: Recall – Precision results of Modified Codebook for the Videos captured from the corner side

Modified Codebook							
Alpha	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,10	**91,865**	**95,0275**	**99,9575**	**9,9225**	**95,74055**	**17,96875**	
0,20	**92,305**	**95,985**	**99,95**	**15,7025**	**95,9755**	**26,98967**	
0,40	**97,87**	**88,45**	**98,8725**	**79,4975**	**98,3687**	**83,73514**	
0,60	**99,0475**	**82,86**	**97,615**	**91,5**	**98,32603**	**86,96593**	**
0,80	**99,6325**	**69,675**	**94,3975**	**97,185**	**96,94438**	**81,16223**	
0,90	**99,935**	**57,0625**	**92,20917**	**99,51**	**95,91676**	**72,5324**	
Epsilon	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
10	**99,2725**	**80,87**	**97,605**	**93,5975**	**98,43169**	**86,76951**	
20	**98,91**	**84,7875**	**98,3825**	**89,835**	**98,64554**	**87,2383**	**
40	**98,6725**	**86,175**	**98,5925**	**88,035**	**98,63248**	**87,09507**	
60	**98,6525**	**86,1475**	**98,5925**	**87,585**	**98,62249**	**86,8603**	
80	**98,6125**	**86,1125**	**98,5675**	**86,895**	**98,58999**	**86,50198**	
100	**98,62**	**86,0825**	**98,535**	**86,80167**	**98,57748**	**86,44059**	

Table 3-4: Recall – Precision results of codebook construction for the Videos captured from the corner side

CodeBook Construction							
Alpha	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,10	**97,2075**	**62,0875**	**92,6575**	**78,96**	**94,87798**	**69,51458**	
0,20	**97,4925**	**62,72**	**92,15**	**84,1625**	**94,746**	**71,87612**	
0,40	**99,355**	**65,6325**	**90,2575**	**95,585**	**94,588**	**77,82632**	
0,60	**99,5625**	**65,3225**	**92,65**	**97,0025**	**95,98195**	**78,0711**	**
0,80	**99,7175**	**59,0825**	**88,755**	**98,3925**	**93,91743**	**73,83108**	
0,90	**99,8975**	**53,1863 3**	**85,39433**	**99,7575**	**92,07832**	**69,38149**	
Epsilon	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
10	**99,665**	**62,8275**	**90,9575**	**97,9**	**95,11237**	**76,53715**	
20	**99,47**	**67,0675**	**93,4525**	**96,1525**	**96,3674**	**79,0186**	
40	**99,11**	**68,6525**	**94,205**	**93,2225**	**96,59527**	**79,07284**	**
60	**99,045**	**68,6925**	**94,2275**	**92,7325**	**96,57621**	**78,92244**	
80	**99,0475**	**69,0825**	**94,38**	**92,7**	**96,65744**	**79,16737**	
100	**99,0175**	**68,86**	**94,16**	**92,4125**	**96,52768**	**78,91643**	

Table 3-5: Recall – Precision results of Single Gaussian for the Videos captured from the corner side

Single Gaussian							
Std. Dev.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
5	**99,6875**	**63,6575**	**91,505**	**97,58**	**95,42116**	**77,0503**	
10	**99,6075**	**69,0125**	**93,475**	**96,855**	**96,44386**	**80,59693**	
20	**99,1025**	**80,0525**	**96,5125**	**91,65**	**97,79035**	**85,45958**	**
40	**96,4925**	**90,7825**	**99,165**	**63,125**	**97,8105**	**74,46869**	
60	**91,82**	**97,8025**	**99,9875**	**6,755**	**95,72986**	**12,63718**	
80	**90,41167**	**98,735**	**100**	**0,2575**	**94,96442**	**0,51366**	
Learn. Rat.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,00001	**98,73**	**82,9825**	**98,2**	**87,4825**	**98,46429**	**85,1731**	
0,00010	**98,7675**	**83,2975**	**98,2325**	**87,7925**	**98,49927**	**85,48595**	
0,00100	**98,9275**	**82,76**	**98,1275**	**89,3925**	**98,52588**	**85,94849**	**
0,01000	**99,4175**	**72,46**	**95,545**	**95,165**	**97,44279**	**82,27479**	
0,10000	**99,615**	**63,8775**	**91,5875**	**97,1125**	**95,43274**	**77,06446**	

Table 3-6: Recall – Precision results of Frame Differencing for the Videos captured from the corner side

Frame Differencing							
Pixel Val. Th.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
10	**94,6375**	**94,8**	**99,8375**	**36,83**	**97,16798**	**53,04997**	**
20	**92,4125**	**97,0175**	**99,97**	**10,575**	**96,04281**	**19,07122**	
40	**91,7425**	**98,7925**	**99,9975**	**2,43175**	**95,6923**	**4,746662**	
60	**91,62**	**99,57**	**100**	**0,8575**	**95,62676**	**1,700356**	
80	**91,5625**	**99,8525**	**100**	**0,355**	**95,59543**	**0,707485**	

Table 3-7: Summary of the |Precision – Recall Graphs of the Algorithms for the Corner View Videos

Algorithm	**Parameters**	**P1**	**R1**
MOG	σ=20, k = 0,001	84,97	80,03
MODIF. CODEBOK	α = 0,6 ε = 20	84,79	89,84
CODEBOOK CONST.	α = 0,6 ε = 40	68,65	93,22
SINGLE GASUSSIAN	σ=20, k = 0,001	82,76	89,39
FRAME DIFFERENCING	Th = 10	94,8	36,83

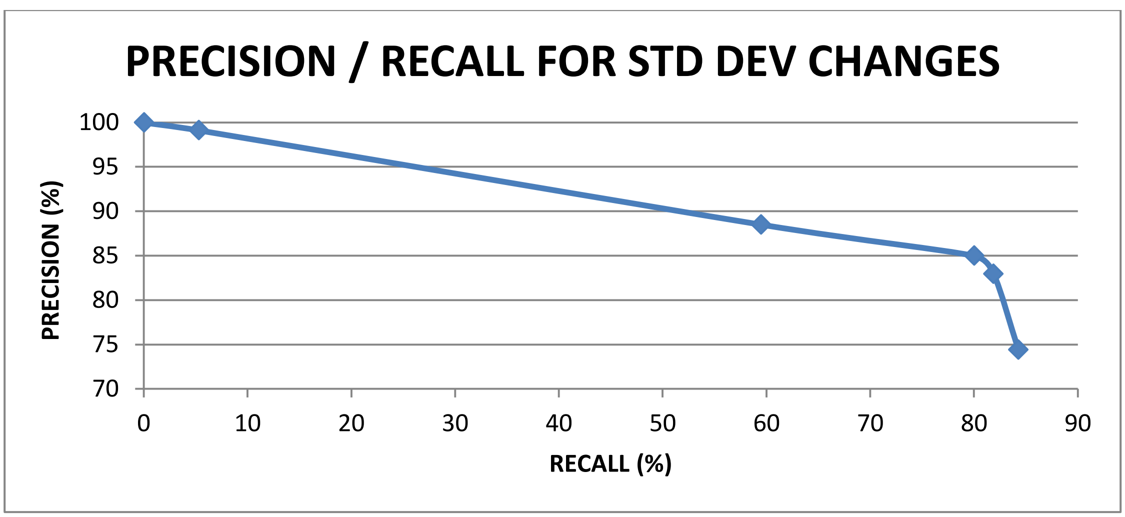


Figure 3-5: Precision - Recall Graph of MOG wrt. Standard deviation changes

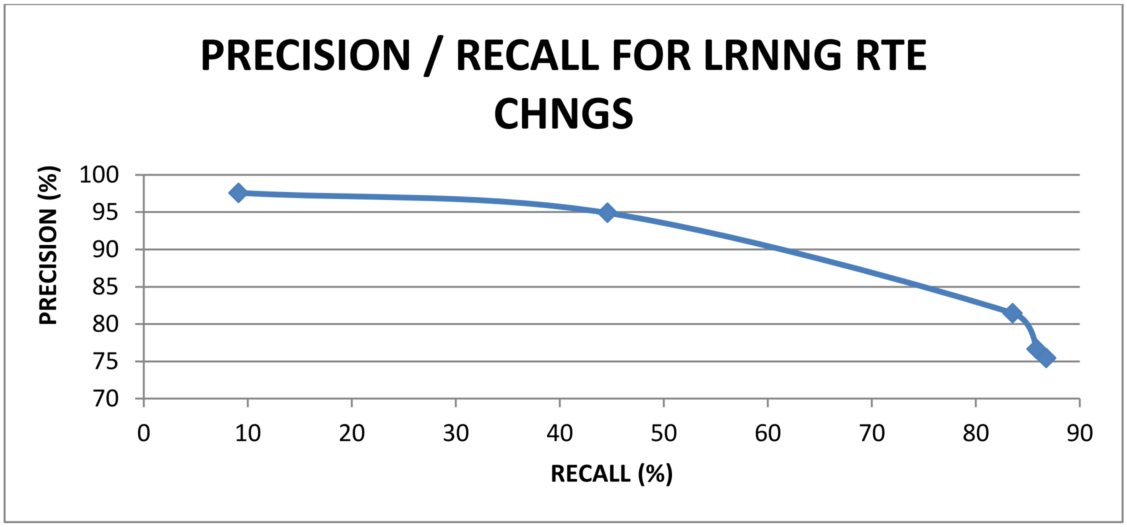


Figure 3-6: Precision - Recall Graph of MOG wrt. Learning rate changes

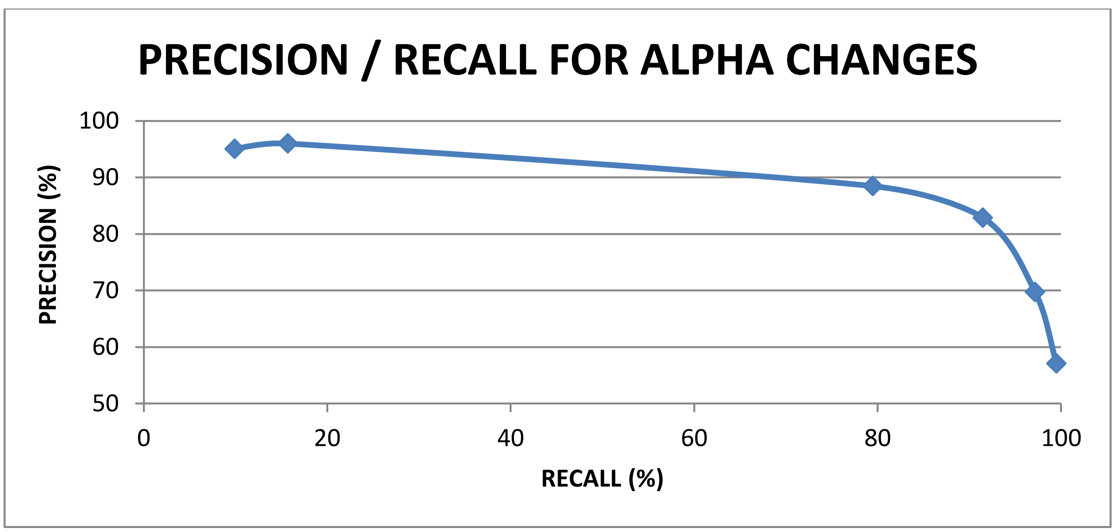


Figure 3-7**:** Precision - Recall Graph of Modified Codebook model wrt. Alpha changes

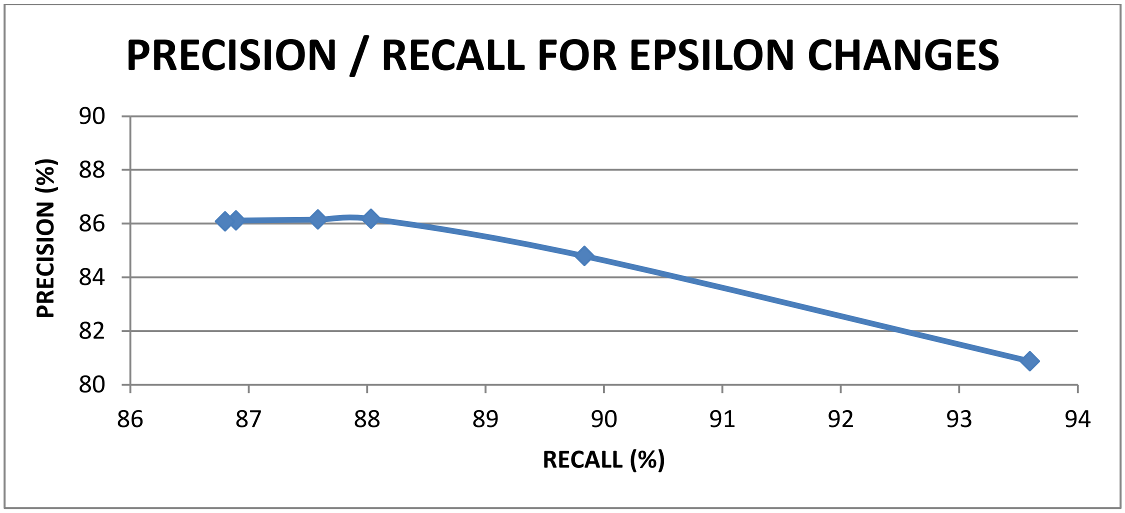


Figure 3-8: Precision - Recall Graph of Modified Codebook model wrt. Epsilon changes

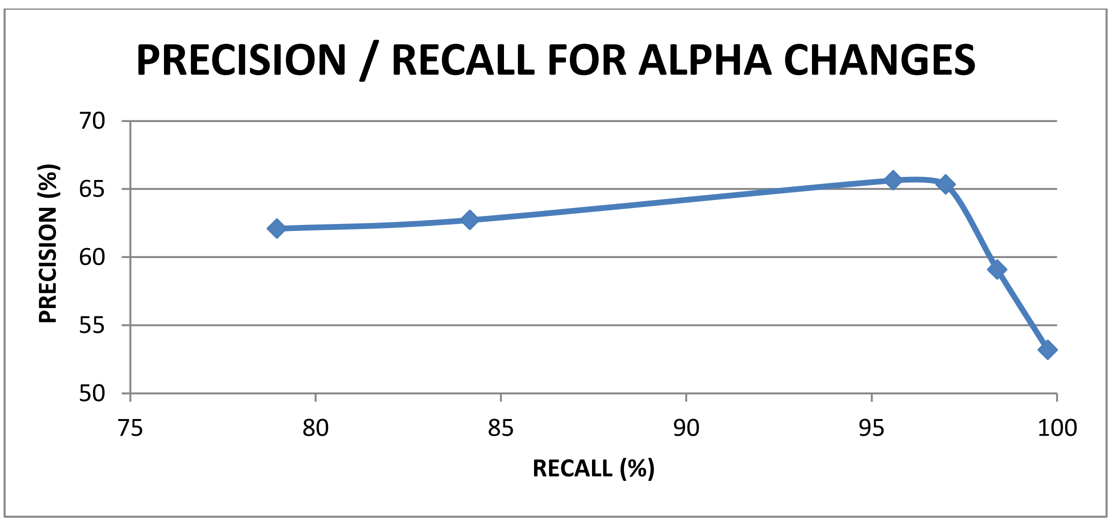


Figure 3-9: Precision - Recall Graph of Codebook Construction model wrt. Alpha changes

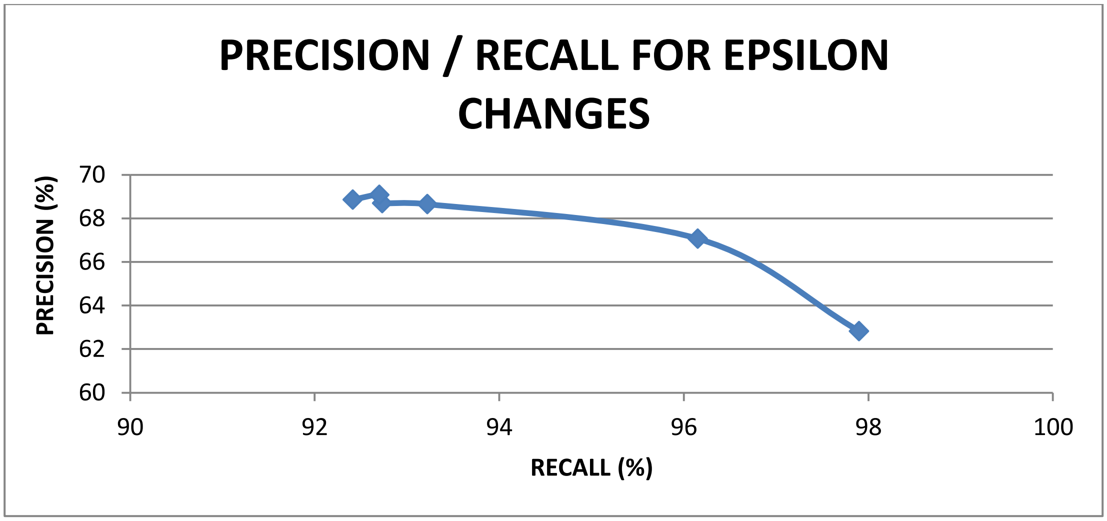


Figure 3-10: Precision - Recall Graph of Codebook Construction model wrt. Epsilon changes

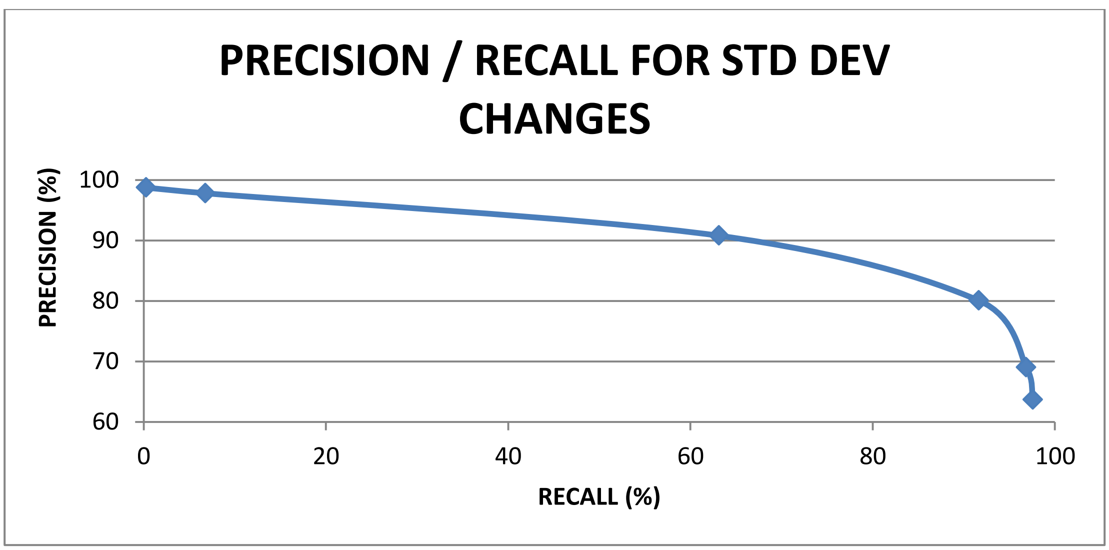


Figure 3-11: Precision - Recall Graph of Single Gaussian model wrt. Standard deviation changes

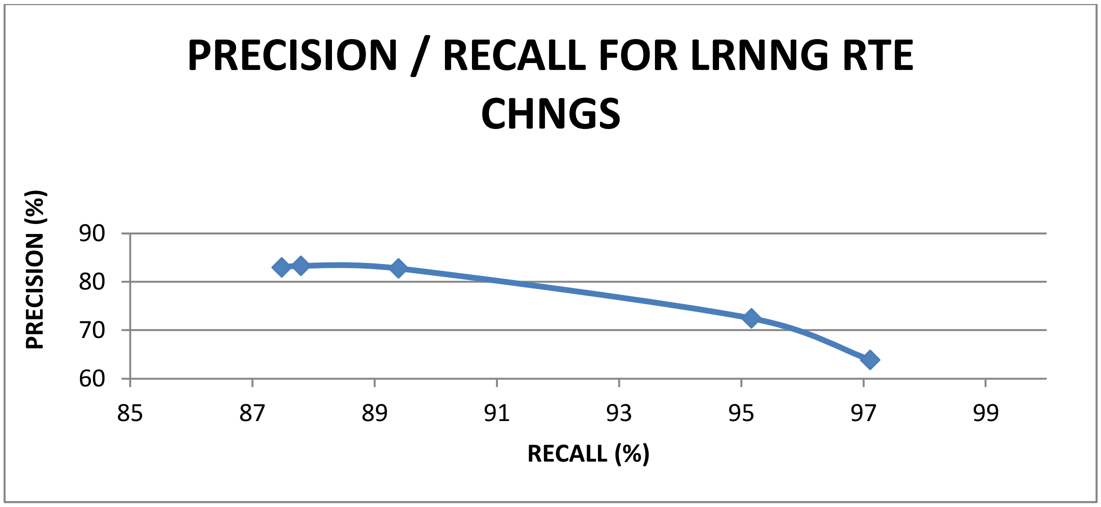


Figure 3-12: Precision - Recall Graph of Single Gaussian model wrt. Learning rate changes

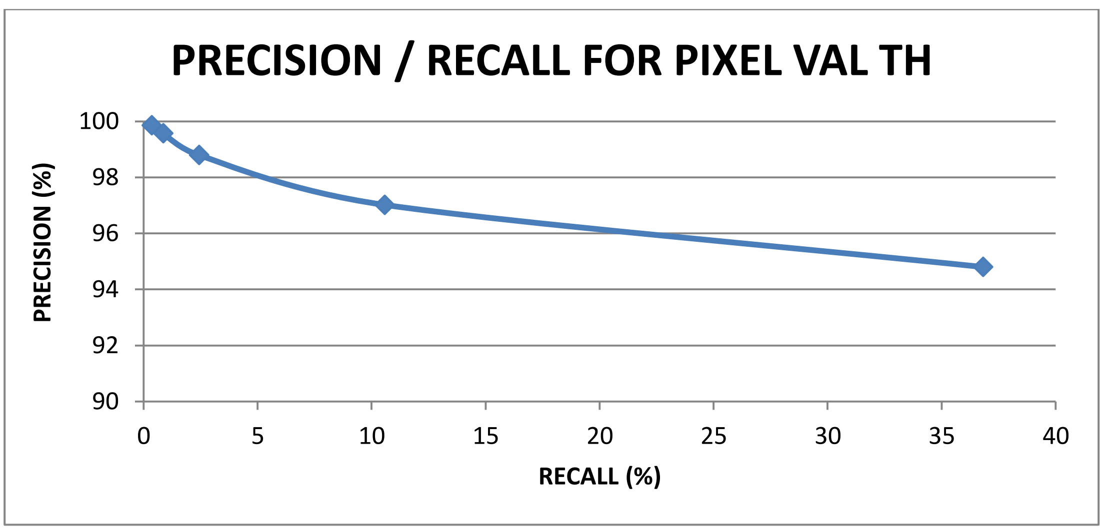
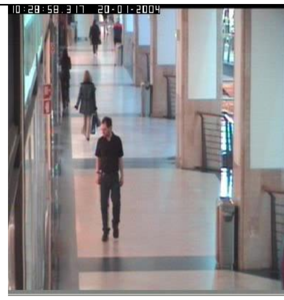
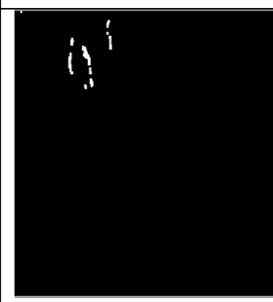
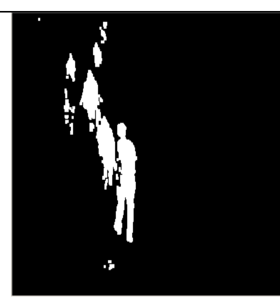
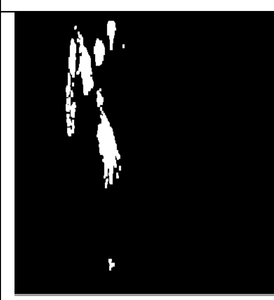
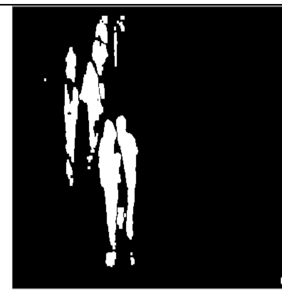
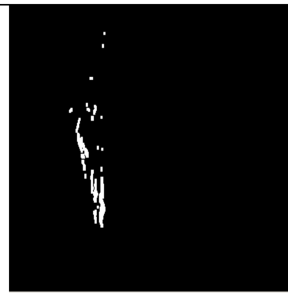
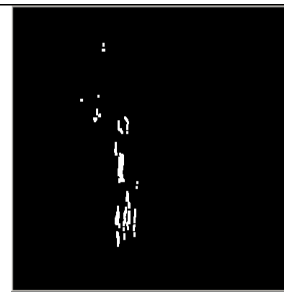
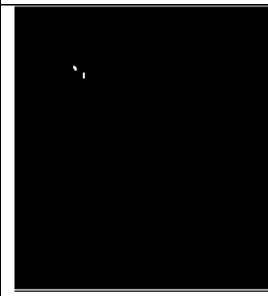


Figure 3-13: Precision - Recall Graph of Frame Differencing model wrt. Pixel value threshold changes

Table 3-8: Visually comparison of the algorithms on corner view videos

Frame Number	200	300	500
Original Frames			
Mixture of Gaussians			
Modified Codebook			
Codebook Construction			

Single Gaussian			
Frame Differencing			

The reason for why the precision of codebook construction is low is that codebook construction algorithm is not able have model for the new incoming background model. Therefore, noise and illumination changes lead the algorithms' precision value to decrease. Secondly, mixture of Gaussians and modified codebook model have low precision values since some real foreground objects are not included by the ground truth analysis data while mixture of Gaussians and modified codebook model detect these objects as foreground. Therefore, pixels for these objects are counted as false alarm for these two algorithms. Furthermore, the existence of initially moving objects at beginning cause the precision value of Mixture of Gaussian to decrease. On the other hand, frame differencing method detects only instantly moving objects. The effect of this problem doesn't affect frame differencing as much as others. Another critical conclusion, single Gaussian methods is not able to model shadows as background. Therefore, this results in decrease in precision value performance.

3.3.2 Results of Videos Captured from the Tower

There are varieties of difficulties in this type of videos. First of all, on the videos, all foreground objects are not labeled as foreground. In the red rectangle region Figure 3-14, all motion and foregrounds are missed there.

Figure 3-14: The Plane of the Tower View [26]

Another problem is that too highly illuminated regions. These regions results in a number of high false alarm detections. Even if Codebook model has an advantage for highly illumination region, it cannot handle this as expected. Moreover, camera noise plays significant role in these videos. It causes numerous false alarm detections as well. Last but not least, some new background objects in these videos are counted as foreground, even though they remain immobile over a long time. Normally, they would be labeled as background after a sufficient time. These also give rise to a decrease on both the recall of the foregrounds and the precision of the backgrounds.

Table 3-9: Recall – Precision results of MOG for the Videos captured from the tower

MOG							
Std. Dev.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
5	**99,395**	**90,5725**	**99,78**	**70,52**	**99,58713**	**79,2982**	
10	**99,475**	**92,6**	**99,845**	**69,295**	**99,65966**	**79,27011**	**
20	**99,4175**	**93,16**	**99,855**	**64,4925**	**99,63577**	**76,2198**	
40	**98,9925**	**96,5875**	**99,975**	**35,18**	**99,48132**	**51,57491**	
60	**98,635**	**99,8675**	**100**	**5,295**	**99,31281**	**10,05679**	
80	**98,5775**	**99,995**	**100**	**1,15**	**99,28365**	**2,273849**	
Learn. Rat.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,00001	**99,53**	**89,7625**	**99,755**	**72,435**	**99,64237**	**80,1732**	
0,00010	**99,53**	**90,0075**	**99,755**	**72,455**	**99,64237**	**80,28306**	
0,00100	**99,5175**	**92,0175**	**99,8175**	**71,235**	**99,66727**	**80,30341**	**
0,01000	**99,21**	**92,6025**	**99,93**	**55,3825**	**99,5687**	**69,31186**	
0,10000	**98,72**	**85,89**	**99,98**	**24,9575**	**99,34601**	**38,67655**	

Table 3-10: Recall – Precision results of Modified Codebook for the Videos captured from the tower

Modified Codebook							
Alpha	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,10	**98,9125**	**96,575**	**99,99**	**14,87**	**99,44833**	**25,77182**	
0,20	**99,195**	**96,86**	**99,975**	**36,725**	**99,58347**	**53,25723**	
0,40	**99,59**	**95,1075**	**99,92**	**62,3075**	**99,75473**	**75,29029**	
0,60	**99,825**	**90,185**	**99,7**	**78,055**	**99,76246**	**83,68272**	**
0,80	**99,8375**	**72,5975**	**99,315**	**90,355**	**99,57556**	**80,5087**	
0,90	**99,595**	**60,95**	**86,75**	**94,965**	**92,72979**	**74,24708**	
Epsilon	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
10	**99,725**	**74,9275**	**95,6**	**93**	**97,61894**	**82,99126**	
20	**99,89**	**86,5675**	**99,7**	**90,72**	**99,79491**	**88,59512**	
40	**99,825**	**88,995**	**99,7775**	**88,5175**	**99,80124**	**88,75561**	**
60	**99,805**	**89,505**	**99,8**	**87,865**	**99,8025**	**88,67742**	
80	**99,805**	**89,605**	**99,805**	**87,835**	**99,805**	**88,71117**	
100	**99,805**	**89,7225**	**99,8125**	**87,8775**	**99,80875**	**88,79042**	**

Table 3-11: Recall – Precision results of Codebook Construction for the Videos captured from the tower

CodeBook Construction							
Alpha	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,10	**99,2825**	**48,2275**	**99,545**	**35,3025**	**99,41358**	**40,76503**	
0,20	**99,5025**	**65,02**	**99,525**	**54,335**	**99,51375**	**59,19922**	
0,40	**99,8025**	**73,855**	**99,305**	**81,6225**	**99,55313**	**77,54472**	
0,60	**99,925**	**71,3625**	**98,9175**	**92,3725**	**99,4187**	**80,51953**	**
0,80	**99,9675**	**64,3975**	**98,61**	**97,645**	**99,28411**	**77,61043**	
0,90	**99,975**	**57,95**	**98,18**	**97,85**	**99,06937**	**72,79085**	
Epsilon	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
10	**99,8125**	**46,8125**	**97,6575**	**89,08**	**98,72324**	**61,37289**	
20	**99,86**	**73,63**	**99,0525**	**88,4725**	**99,45461**	**80,37174**	
40	**99,885**	**82,42**	**99,2875**	**89,6675**	**99,58535**	**85,89114**	
60	**99,8775**	**85,7775**	**99,4275**	**88,795**	**99,65199**	**87,26017**	**
80	**99,875**	**85,795**	**99,4425**	**88,5175**	**99,65828**	**87,13499**	
100	**99,8875**	**86,555**	**99,4525**	**88,125**	**99,66953**	**87,33294**	

Table 3-12: Recall – Precision results of Single Gaussian for the Videos captured from the tower

Single Gaussian							
Std. Dev.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
5	**99,93**	**63,2725**	**96,2475**	**96,355**	**98,05419**	**76,38561**	
10	**99,7425**	**78,0475**	**98,87**	**85,1**	**99,30433**	**81,42132**	**
20	**99,49**	**89,71**	**99,725**	**69,1375**	**99,60736**	**78,09157**	
40	**98,8075**	**93,955**	**99,9625**	**32,46**	**99,38164**	**48,25028**	
60	**98,425**	**98,7725**	**100**	**3,8275**	**99,20625**	**7,36943**	
80	**98,37**	**100**	**100**	**0,3025**	**99,1783**	**0,603175**	
Learn. Rat.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,00001	**99,3975**	**59,54**	**98,3675**	**82,79**	**98,87982**	**69,26602**	
0,00010	**99,6425**	**65,8575**	**98,8325**	**82,33**	**99,23585**	**73,17821**	
0,00100	**99,635**	**83,715**	**99,64**	**81,98**	**99,6375**	**82,83842**	**
0,01000	**99,665**	**79,6975**	**99,52**	**85,71**	**99,59245**	**82,59447**	
0,10000	**99,8325**	**73,9375**	**98,885**	**92,94**	**99,35649**	**82,35683**	

Table 3-13: Recall – Precision results of Frame Differencing for the Videos captured from the tower

Frame Differencing							
Pixel Val. Th.	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
10	**98,795**	**90,6675**	**99,97**	**23,2875**	**99,37903**	**37,05707**	**
20	**98,58**	**97,7275**	**99,9975**	**10,075**	**99,28369**	**18,26682**	
40	**98,5**	**98,5725**	**100**	**3,0025**	**99,24433**	**5,827496**	
60	**98,4025**	**97,445**	**100**	**0,55**	**99,19482**	**1,093826**	
80	**98,395**	**97,515**	**100**	**0,185**	**99,19101**	**0,369299**	

Table 3-14: Summary of the |Precision – Recall Graphs of the Algorithms for the Tower View Videos

Algorithm	Parameters	P1	R1
MOG	σ=10, k = 0,001	92,6	69,3
MODIF. CODEBOK	α = 0,6 ε = 40	89	88,52
CODEBOOK CONST.	α = 0,6 ε = 60	85,78	88,795
SINGLE GASUSSIAN	σ=10, k = 0,001	83,72	81,98
FRAME DIFFERENCING	Th = 10	90,67	23,29

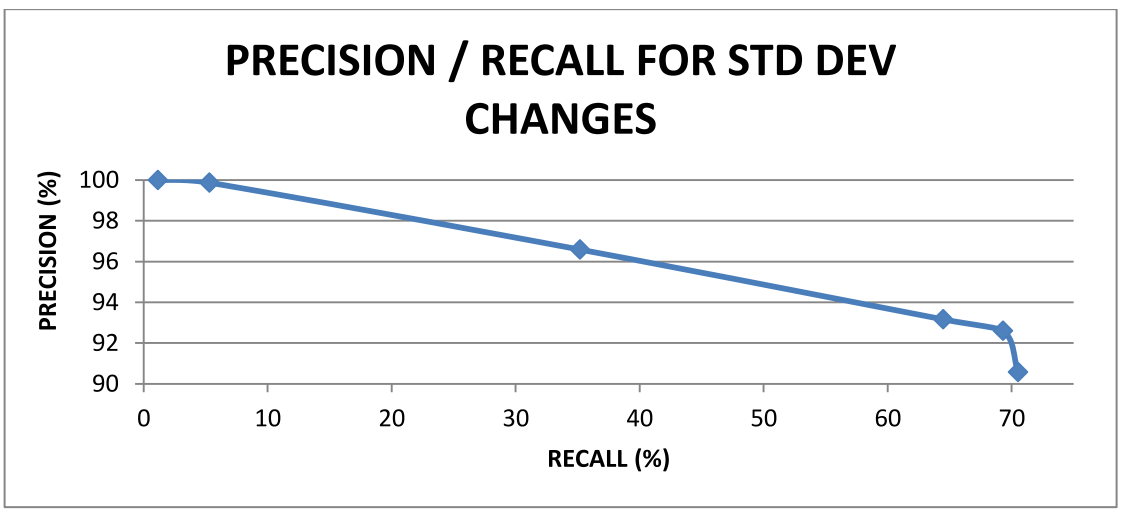


Figure 3-15: Precision - Recall Graph of MOG wrt. Standard deviation changes

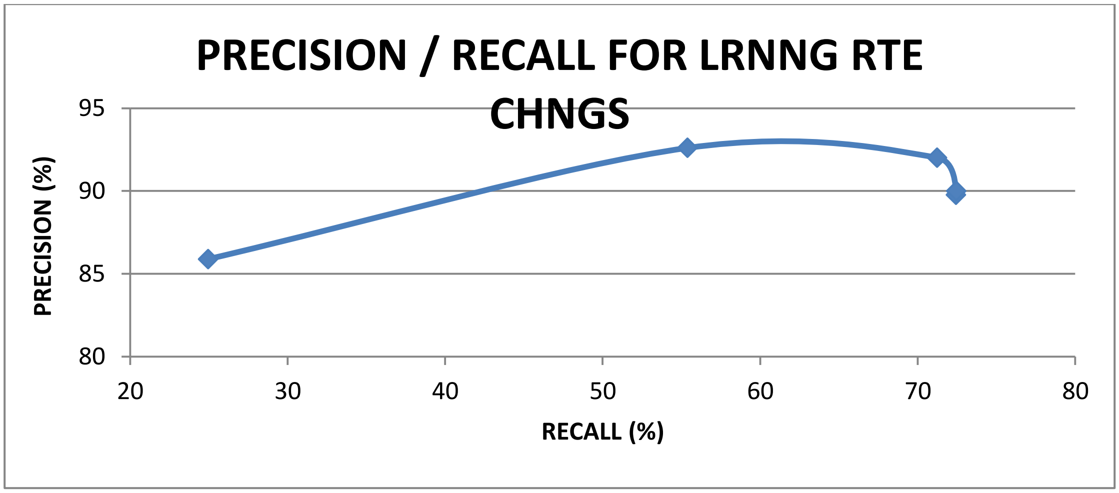


Figure 3-16: Precision - Recall Graph of MOG wrt. Learning rate changes

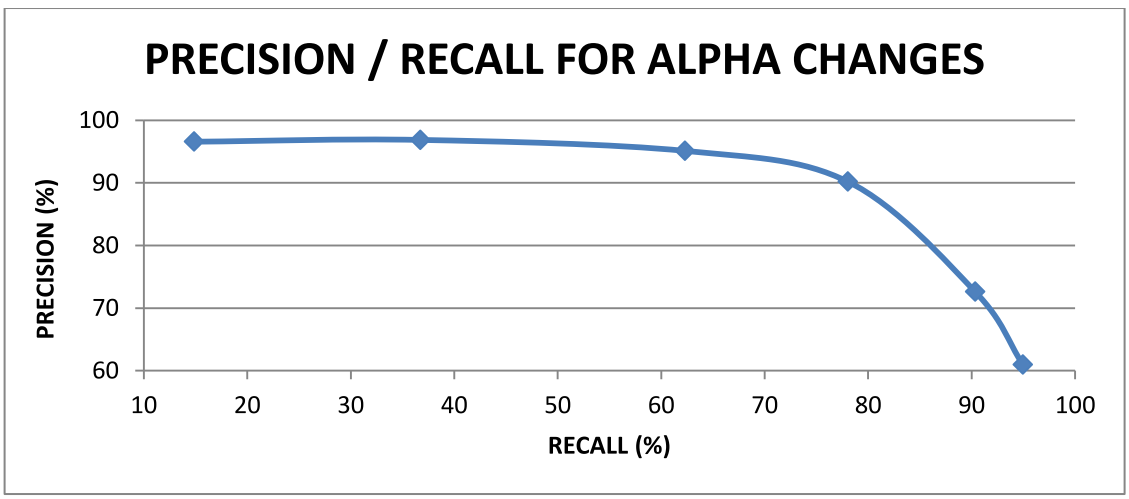


Figure 3-17: Precision - Recall Graph of Modified Codebook model wrt. Alpha changes

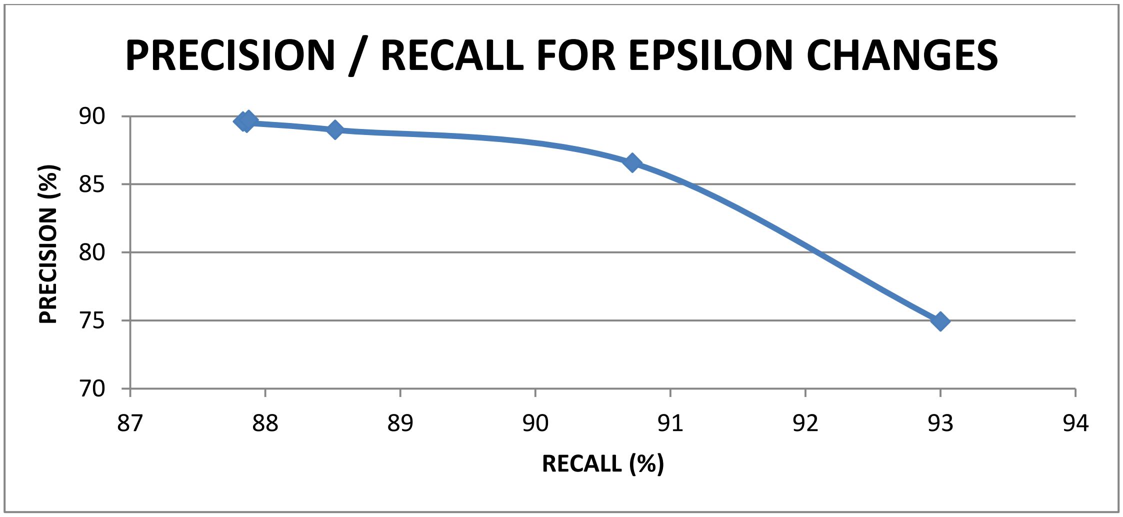


Figure 3-18: Precision - Recall Graph of Modified Codebook model wrt. Epsilon changes

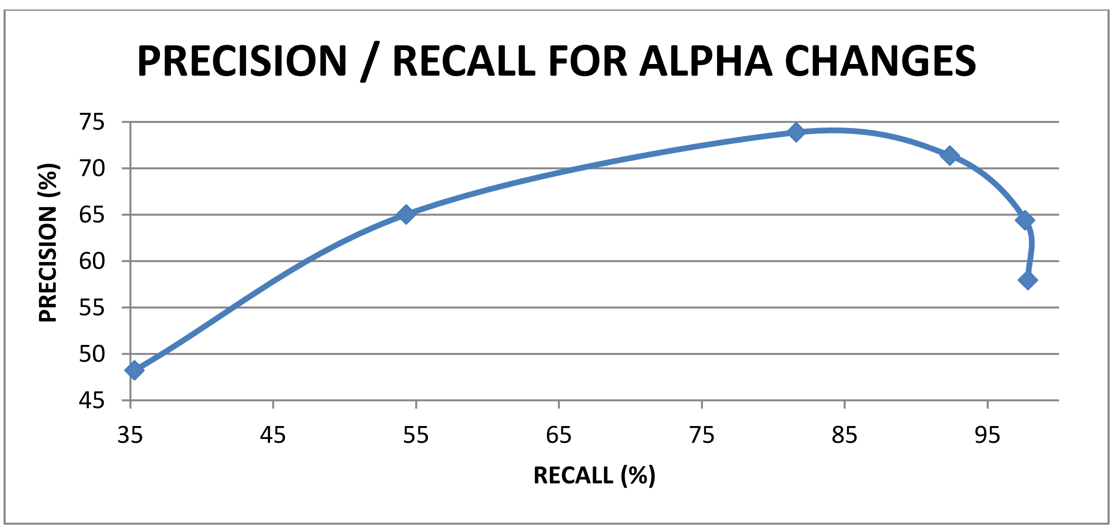


Figure 3-19: Precision - Recall Graph of Codebook Construction model wrt. Alpha changes

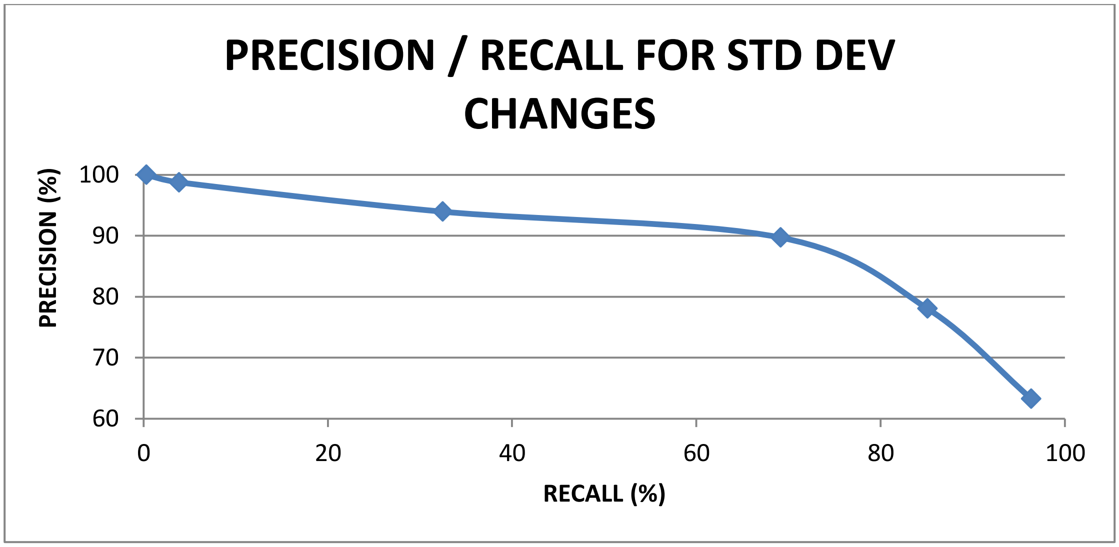


Figure 3-20: Precision - Recall Graph of Single Gaussian model wrt. Standard deviation changes

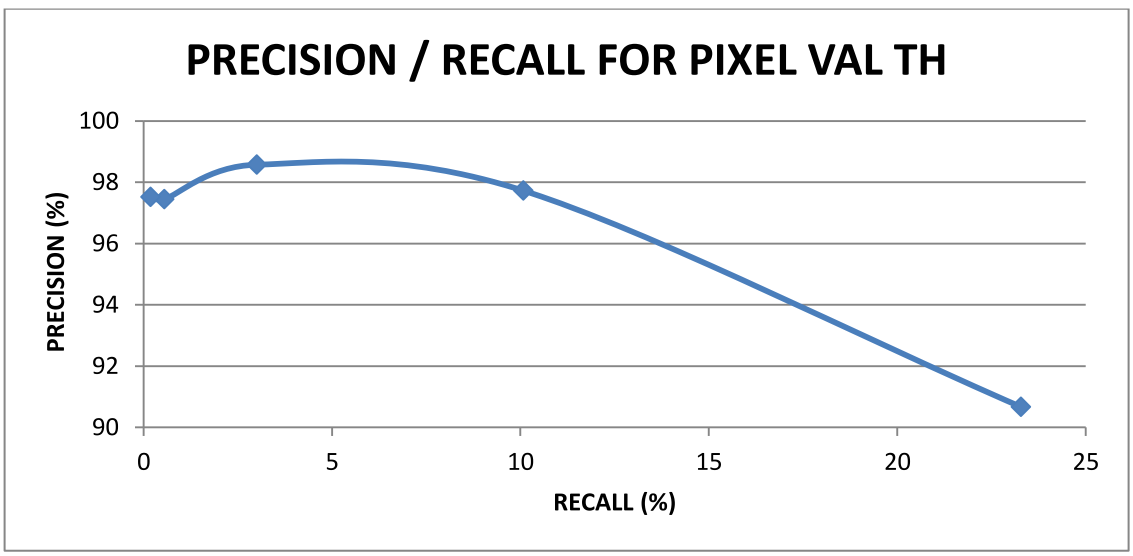
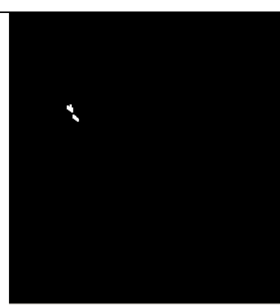
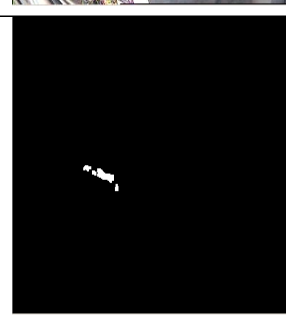
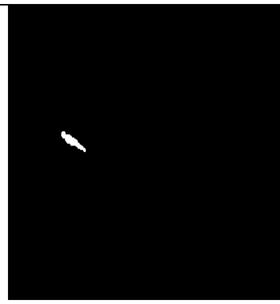
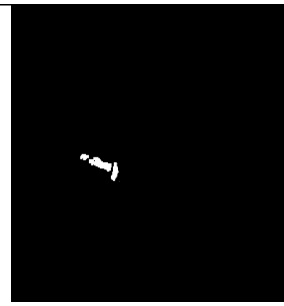
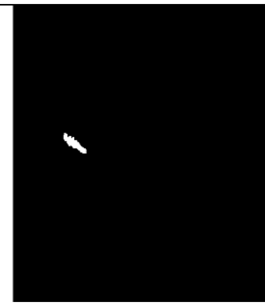
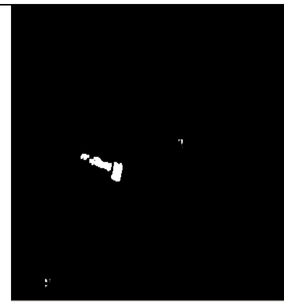
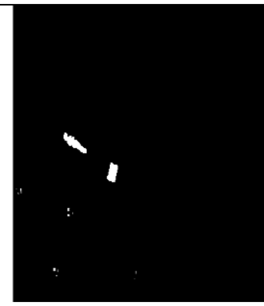
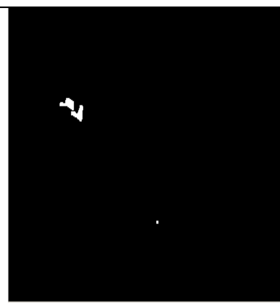
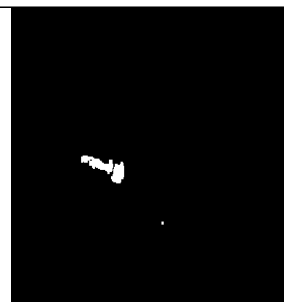
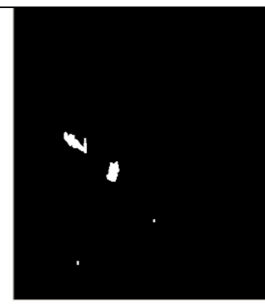
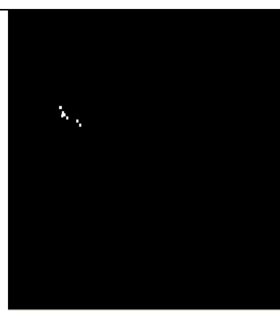
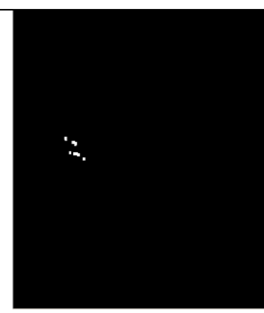


Figure 3-21: Precision - Recall Graph of Frame Differencing model wrt. Pixel value threshold changes

Due to high level noise, σ parameters of mixture of Gaussians and running average Gaussian methods had to be increased to model the high variance in noise. However, the σ parameter is decreased since majority of the scenes has low contrast regions. On the other hand, codebook models increase their ε parameter to compansate the noise with a high variance. When ε parameter increases, tolerance in colour distortion increases as well. Finally, Frame differencing cannot handle the high level noise as much as others can since it cannot remember the history of noise so it has no backgorund model for the noise.

Table 3-15: Visually comparison of the algorithms on Tower view videos

Frame Number	**450**	**600**	**750**
Original Frames			
Mixture of Gaussians			

Modified Codebook			
Codebook Construction			
Single Gaussian			
Frame Differencing			

3.3.3 Results of Videos Captured from the Front Side

In this type of videos, the number of the objects with their shadows is higher when it is compared with other types of videos so, the algorithms must be able to detect these shadows exactly. Therefore, from this perspective MOG and CodeBook models have better performances than others.

Figure 3-22: Front Side Plane [26]

Moreover, as seen in Figure 3-22, motion in x directions results in background modeling as mentioned before on corner side videos. Thus, motion in this direction may look like background. It may cause the algorithms to impair their own background models at these regions. Lastly, the performances of the algorithms are higher; when the parameters of the algorithms which are heavily related with contrast are set to the values which make algorithms more sensitive to the pixel changes. The algorithms must be more sensitive for this type of videos as a result of low illumination.

Table 3-16: Recall – Precision results of MOG for the Videos captured from the front side

MOG							
PARAMETERS	P0	P1	R0	R1	F0	F1	
5	99,3375	92,365	99,935	55,3225	99,63535	69,19831	**
10	99,3225	94,68775	99,9625	53,225	99,64147	68,14498	
20	99,3125	96,73	99,9625	50,8275	99,63644	66,63903	
40	99,0225	97,5775	99,9775	30,8725	99,49771	46,90481	
60	98,6525	99,825	100	2,1275	99,32168	4,166209	
80	98,5425	100	100	0,0275	99,2659	0,054985	
0,00001	99,1275	86,58	99,86775	58,2075	99,49625	69,61382	
0,00010	99,125	86,83	99,855	58,1925	99,48866	69,68374	
0,00100	99,3475	92,8675	99,94	55,8875	99,64287	69,78095	**
0,01000	99,0625	99,08	99,9975	32,925	99,5278	49,42554	
0,10000	98,7	98,9625	100	5,58	99,34575	10,56433	

Table 3-17: Recall – Precision results of Modified Codebook for the Videos captured from the front side

Modified Codebook							
PARAMETERS	**P0**	**P1**	**R0**	**R1**	**F0**	**F1**	
0,10	**98,835**	**94,1925**	**99,9425**	**22,47**	**99,38566**	**36,28425**	
0,20	**98,84**	**94,055**	**99,9425**	**22,68**	**99,38819**	**36,54718**	
0,40	**99,12**	**92,6225**	**99,91**	**42,845**	**99,51343**	**58,58838**	
0,60	**99,5625**	**88,65**	**99,82**	**73,6125**	**99,69108**	**80,43446**	
0,80	**99,88**	**75,1275**	**99,4375**	**93,6025**	**99,65826**	**83,35354**	**
0,90	**99,965**	**60,7775**	**98,1775**	**98,74**	**99,06319**	**75,24153**	
10	**99,85**	**78,8225**	**99,575**	**90,53**	**99,71231**	**84,27157**	
20	**99,8075**	**87,0825**	**99,79**	**86,6625**	**99,79875**	**86,87199**	**
40	**99,78**	**88,64**	**99,815**	**84,6675**	**99,7975**	**86,60822**	
60	**99,7725**	**88,6025**	**99,815**	**84,01**	**99,79375**	**86,24516**	
80	**99,7725**	**88,6925**	**99,82**	**83,5425**	**99,79624**	**86,0405**	
100	**99,7725**	**88,8775**	**99,8275**	**83,485**	**99,79999**	**86,0969**	

Table 3-18: Recall – Precision results of Codebook Construction for the Videos captured from the front side

CodeBook Construction							
PARAMETERS	P0	P1	R0	R1	F0	F1	
0,10	99,46	49,33	98,76	65,715	99,10876	56,3557	
0,20	99,4675	49,4775	98,7175	66,5325	99,09108	56,75135	
0,40	99,645	54,8625	98,765	78,535	99,20305	64,59831	
0,60	99,8175	62,81	98,97	90,0575	99,39194	74,00542	
0,80	99,9325	62,2975	97,2625	96,8825	98,57942	75,83286	**
0,90	99,9125	57,3425	97,52	97,605	98,70175	72,24272	
10	99,97	47,5175	95,3625	98,9975	97,61191	64,21341	
20	99,915	64,3225	98,2575	95,4275	99,07932	76,84677	
40	99,8775	71,9825	98,8725	92,925	99,37246	81,12395	
60	99,87	72,8	98,9475	92,5125	99,40661	81,48095	
80	99,8675	72,9175	98,9525	92,455	99,40789	81,53215	**
100	99,845	72,875	98,9725	92,3725	99,40684	81,4735	

Table 3-19: Recall – Precision results of Single Gaussian for the Videos captured from the front side

Single Gaussian							
PARAMETERS	P0	P1	R0	R1	F0	F1	
5	99,9525	68,05	98,9975	97,86	99,47271	80,27693	
10	99,8275	81,505	99,6825	89,0875	99,75495	85,12774	**
20	99,555	89,455	99,8925	68,8925	99,72346	77,83866	
40	99,1	96,2675	99,985	35,2925	99,54053	51,64975	
60	98,705	99,91	100	5,53	99,34828	10,47994	
80	98,63	99,92	100	0,695	99,31028	1,380399	
0,00001	99,77	82,1125	99,69	85,2025	99,72998	83,62897	
0,00010	99,7775	82,2875	99,6925	85,735	99,73498	83,97588	
0,00100	99,8275	81,505	99,6825	89,0875	99,75495	85,12774	**
0,01000	99,9525	69,6575	99,07	97,69	99,50929	81,32588	
0,10000	99,955	58,545	98,545	98,01	99,24499	73,30325	

Table 3-20: Recall – Precision results of Frame Differencing for the Videos captured from the front side

Frame Differencing							
PARAMETERS	P0	P1	R0	R1	F0	F1	
10	99,2025	95,875	99,9775	40,76	99,58849	57,20152	**
20	98,915	98,605	99,9975	17,8375	99,4533	30,21005	
40	98,7225	99,305	100	3,4325	99,35714	6,635638	
60	98,69	99,445	100	0,9225	99,34068	1,828042	
80	98,93	99,865	100	0,2775	99,46212	0,553462	

Table 3-21: Summary of the |Precision – Recall Graphs of the Algorithms for the Front View Videos

Algorithm	**Parameters**	**P1**	**R1**
MOG	σ=5, k = 0,001	92,86	55,89
MODIF. CODEBOK	α = 0,8 ε = 20	87,08	86,66
CODEBOOK CONST.	α = 0,8 ε = 80	72,92	92,46
SINGLE GASUSSIAN	σ=10, k = 0,001	81,51	89,09
FRAME DIFFERENCING	Th = 10	95,875	40,76

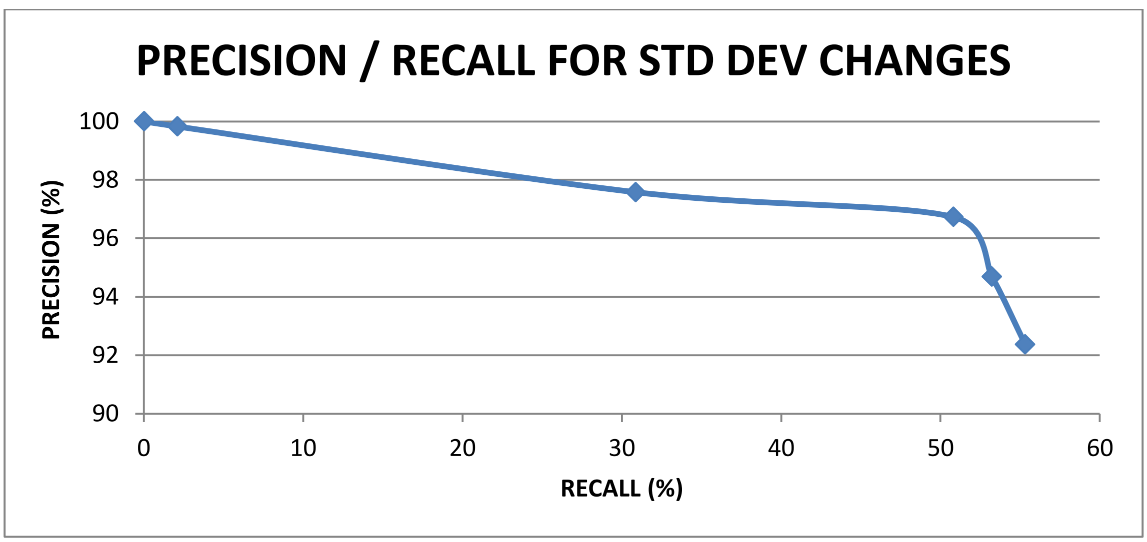


Figure 3-23: Precision - Recall Graph of MOG wrt. Standard deviation changes

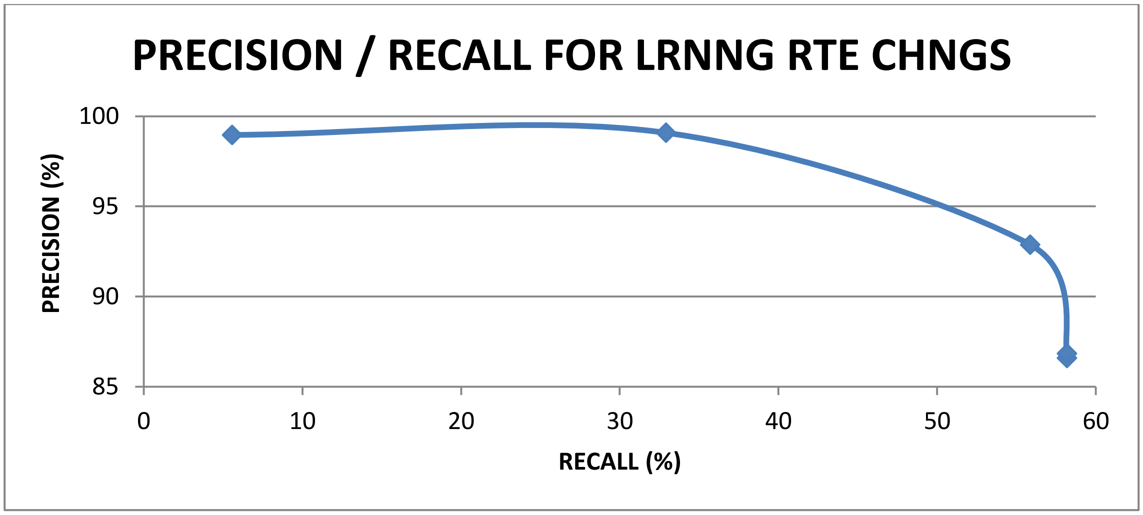


Figure 3-24: Precision - Recall Graph of MOG wrt. Learning Rate changes

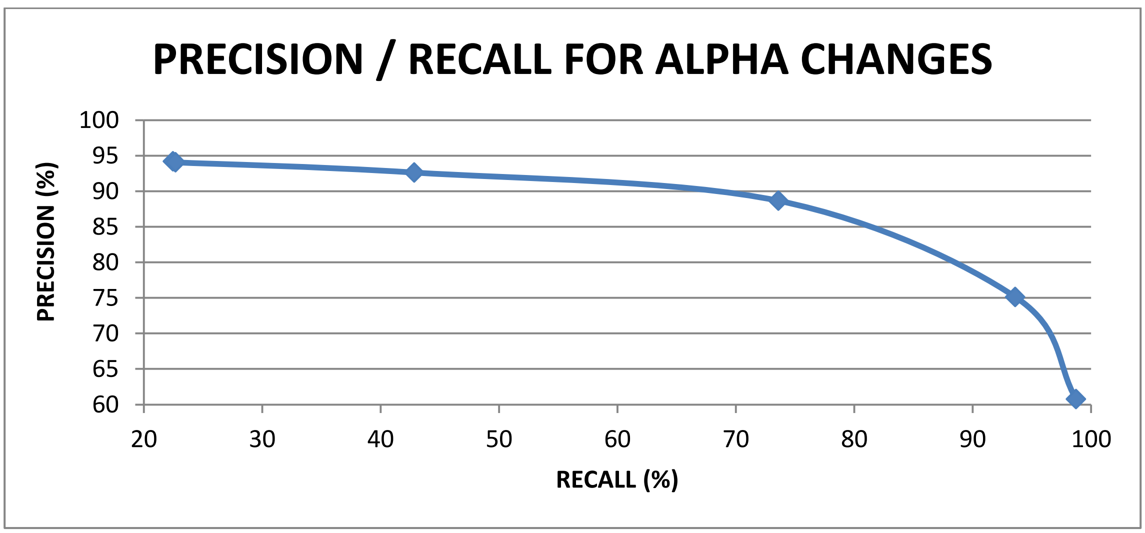


Figure 3-25: Precision - Recall Graph of Modified Codebook model wrt. Alpha changes

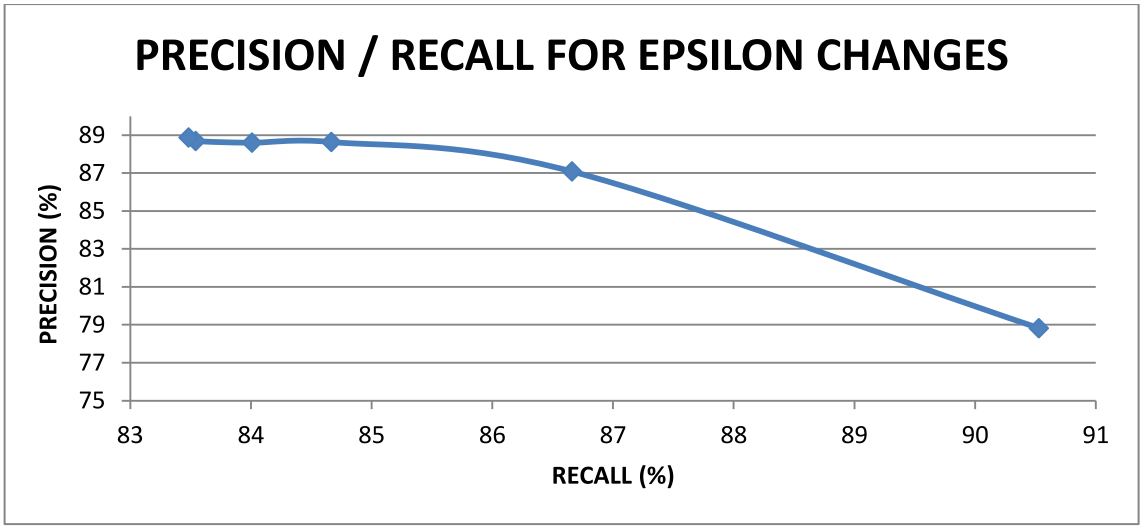


Figure 3-26: Precision - Recall Graph of Modified Codebook model wrt. Epsilon changes

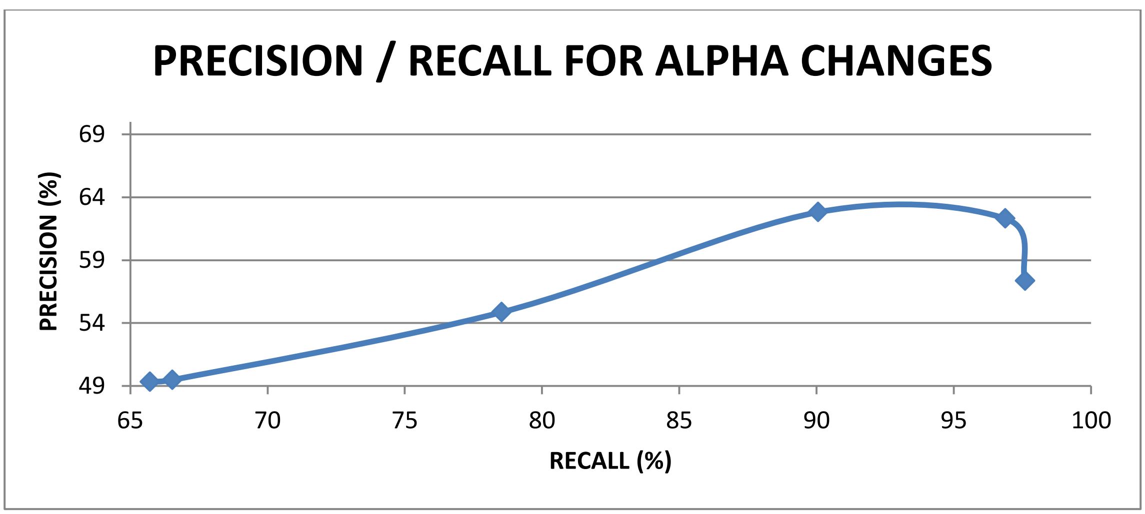


Figure 3-27: Precision - Recall Graph of Codebook Construction model wrt. Alpha changes

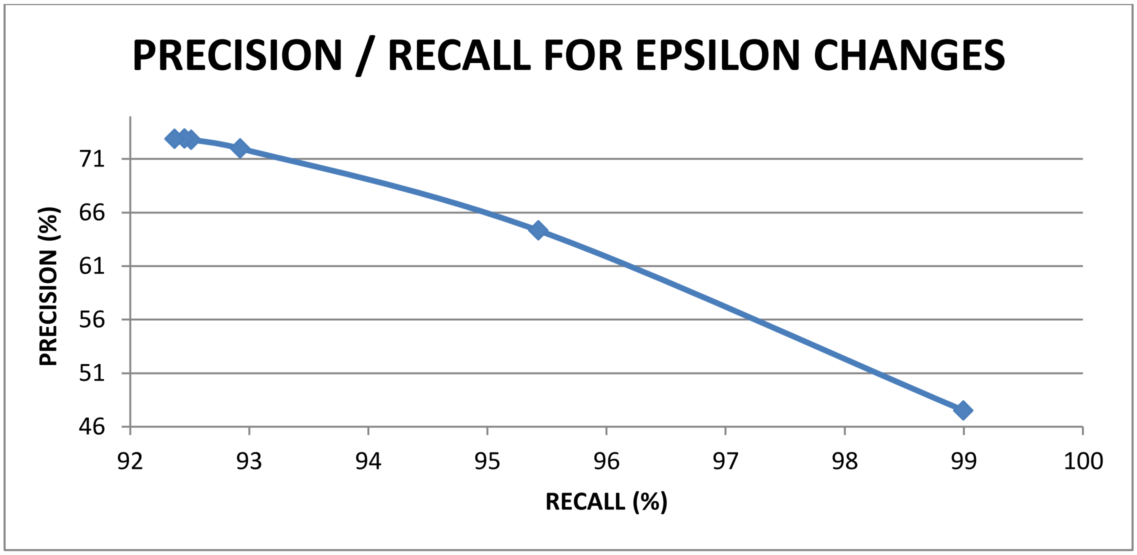


Figure 3-28: Precision - Recall Graph of Codebook Construction model wrt. Epsilon changes

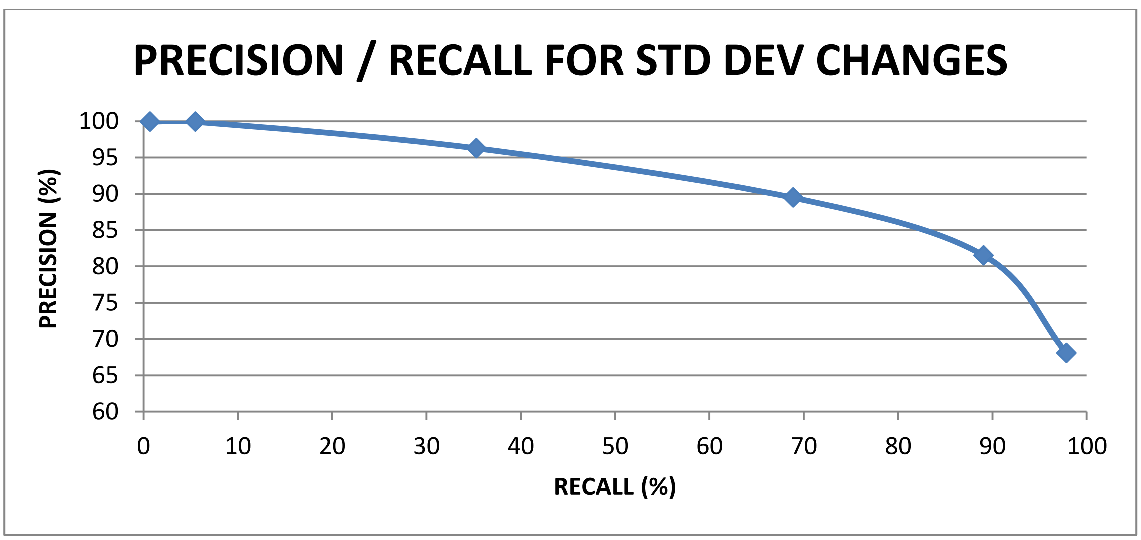


Figure 3-29: Precision - Recall Graph Single Gaussian model wrt. Standard Deviation changes

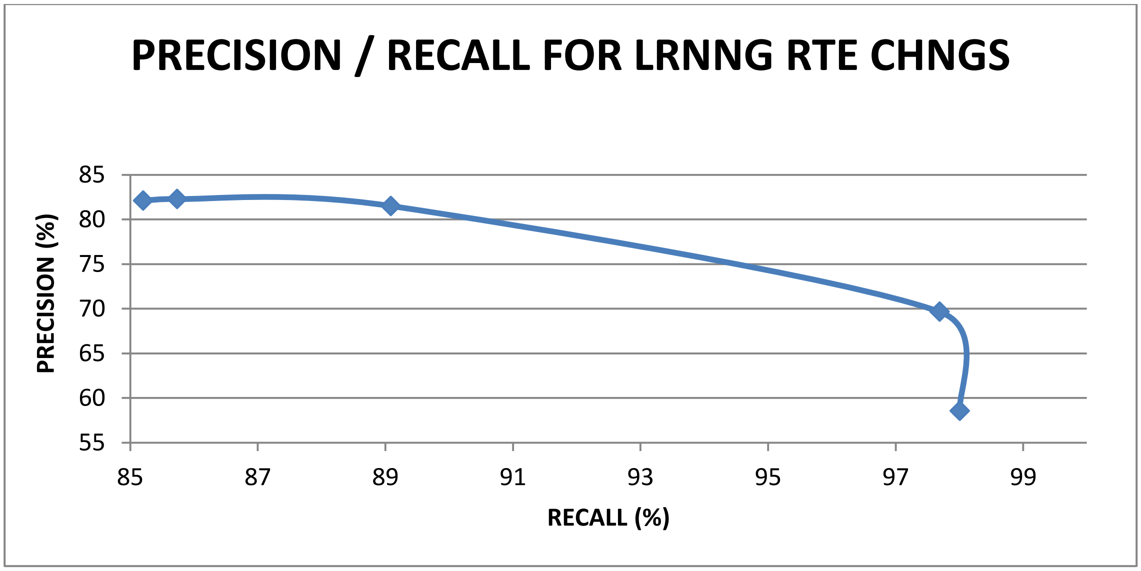


Figure 3-30: Precision - Recall Graph Single Gaussian model wrt. Learning Rate changes

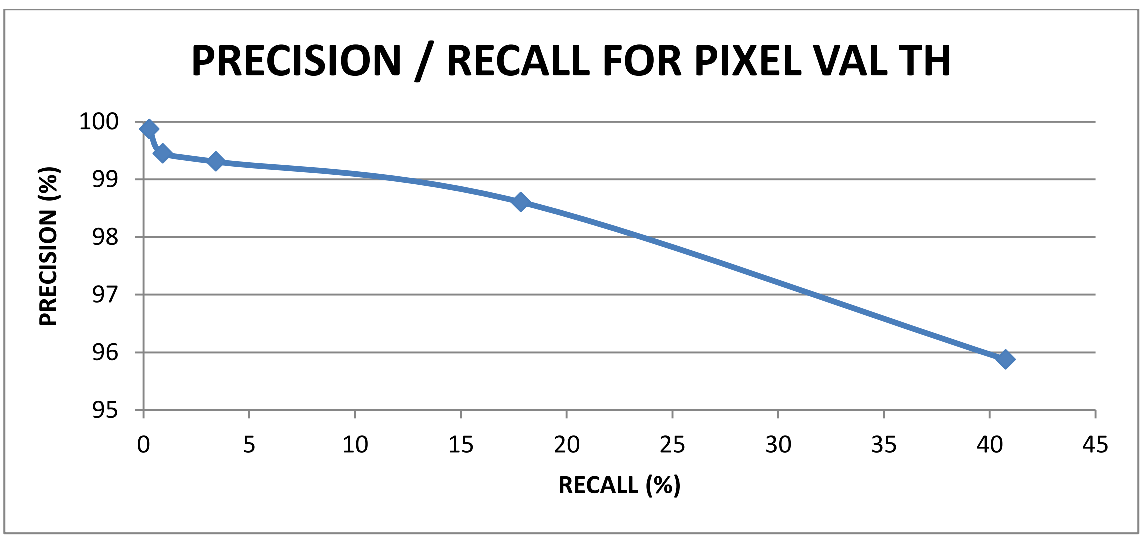
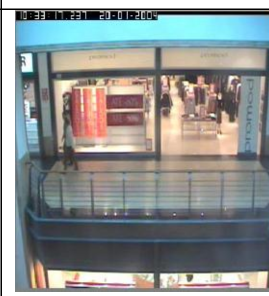
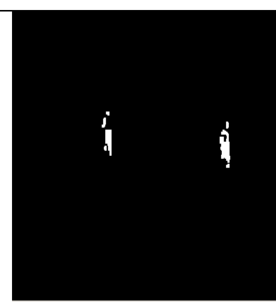
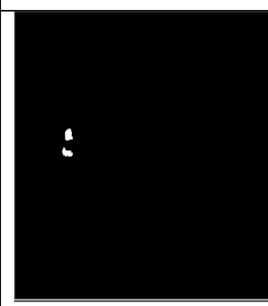
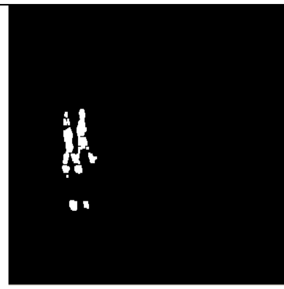
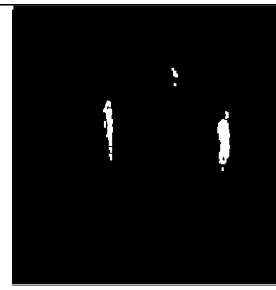
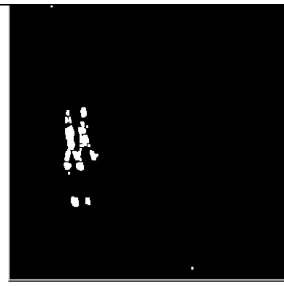
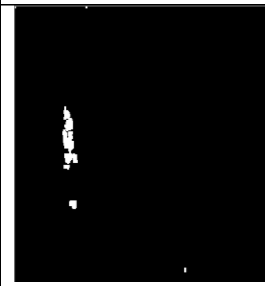
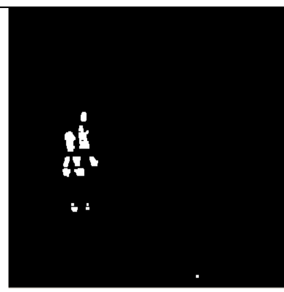
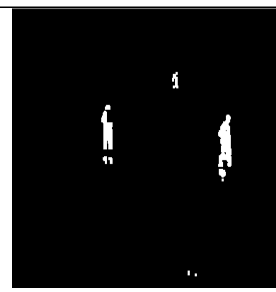
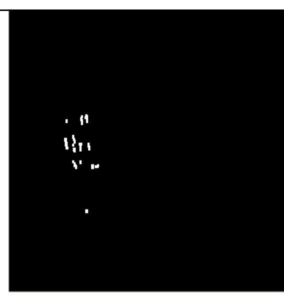


Figure 3-31: Precision - Recall Graph of Frame Differencing model wrt. Pixel value threshold changes

Due to the fact that illumination level of the scene is low, α parameter of codebook model is increased to increase the sensitivity to the brightness. On the other hand, Gaussian based methods decrease their initial standard deviation value to increase the sensitivity to contrast. On the other hand, it is observed that this action is not enough since only some certain values are applied in this study while frame differencing puts better performance for the false alarm since low level noise. Frame differencing only detects changes in pixel therefore, for moving objects only edges are detected. In fact, stationary foreground objects are not totally detected as seen Table 3-22.

Table 3-22: Visually comparison of the algorithms on front view videos

Frame Number	**400**	**800**	**1200**
Original Frames			
Mixture of Gaussians			

Modified Codebook			
Codebook Construction			
Single Gaussian			
Frame Differencing			

CHAPTER 4

CONCLUSIONS AND FUTURE WORKS

4.1 Conclusions

As a summary, this study presents comparative results of background subtraction algorithms. Firstly, the algorithms to be implemented are searched on the literature. The chosen algorithms are described. The algorithms which have been implemented during this thesis study are explained in detail. Some critical points for the implementations are mentioned. In order to evaluate the performance of algorithms, a number of numerical metrics such as precision - recall and f – measures metrics are applied. Videos used in this study are categorized by regarding background dynamics (static or non- static background), noise, the foreground objects having quasi - periodic motions, illumination changes and weather changes so; people who read this study are able to choose the most appropriate algorithm with the most suitable parameter value to get the best result.

Based on accomplishments and results of this research following conclusions can be drawn

1. Mixture of Gaussians performance decrease when the illumination level of the scene decreases. On the other hand modified codebook show better result in such scenes.
2. When the noise level of a scene is high, initial standard deviation must be set to high value.
3. Mixture of Gaussians puts on better performance with respect to the number of false alarm than modified codebook since tolerance to the disturbances is

higher in mixture of Gaussian model. On the other hand, modified codebook has better results on mis - detection alarms since it has more detailed model for the background pixel when the speed of changes in a scene increases, learning rate must be increased for good performance. For Modified Codebook, time period for updating background model must be decreased.

4. When the speed of changes in a scene increases, learning rate must be increased for good performance. For modified codebook, time period for updating background model must be decreased.
5. When α parameter of codebook increase, sensitivity to the brightness increase as well.
6. When ε parameter of codebook increase, sensitivity to the noise increase as well.
7. The videos which have dynamically changing background and foreground objects needs to be modeled by the background subtraction algorithms which have a multimodal structure and a capable of updating the models over time.
8. Algorithms such as Codebook Construction, Single Gaussian are more suitable for the background in which there is no more new background model. For these types of algorithms, foreground objects are always counted foreground even if, they remain stationary. On the other hand, Mixture of Gaussians and Modified Codebook algorithms learn foreground objects over time. These objects are not foreground any more for this type of algorithms. This results in a decrease in performance of the foreground detection for this type of background scenes.
9. Merge of foreground objects has a great impact on the performance and the generation of the foreground objects. If the threshold for the distance between two foreground detection regions is high, the number of objects decreases while the boundary of the objects is increasing. This would result in better or worse performance depending on the application.

10. Pre-processing and Post-processing such as dilation, erosion, median filtering and connected component labeling play significant role in the performance of the background subtraction algorithms.

4.2 Future Works

For more healthy performance evaluation, a design of software which enables user to define foreground precisely and easily is aimed. Furthermore, it would be very easy to relate data structures with this design so; ground truth analysis will be performed better. After the background – foreground subtraction process, some post processing is required to improve the performance. Even a primitive tracker which looks for target in tracked objects' neighborhood will be enough to keep track of foreground objects.

All the algorithms in this study are designed for static camera. Therefore, a new background foreground subtraction algorithm which is able to detect foreground objects on moving background is intended to study for the further studies. In today's technology driven world, such an algorithm helps the computer vision and robotics studies go further. It would bring a number of solutions to a variety of problems in robotics and computer vision studies.

REFERENCES

[1] K. Kyungnam, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Real time foreground–background segmentation using codebook model,” *Journal Real-Time Imaging archive,* volume 11, Issue 3, June 2005.

[2] K. Kyungnam, T. H. Chalidabhongse, D. Harwood, L. Davis, “Background Modeling and Subtraction by Codebook Construction” *Image Processing, 2004. ICIP '04, 2004 International Conference,* volume 5, pp.3061 - 3064

[3] J.C. SanMiguel and J.M. Martinez, “On the Evaluation of Background Subtraction Algorithms without Ground-truth”, *7th IEEE International Conference on Advanced Video and Signal Based Surveillance,* pp. 180-187, 2010

[4] S. Herrero and J. Bescos, “Background subtraction techniques: Systematic evaluation and comparative analysis”, *Proceeding of ACIVS,* pp. 33-42, 2009

[5] J. Nascimento and J. Marques, “Performance evaluation of object detection algorithms for video surveillance”, *IEEE Transactions on Multimedia,* 8(4): pp. 761-774, 2006

[6] P. Correia and F. Pereira, “Stand-alone objective segmentation quality evaluation”, *EURASIP Journal on Applied Signal Processing,* volume 4, pp. 389-400, 2002

[7] C. Erdem, Sankur and A. Tekalp, “Performance measures for video object segmentation and tracking”, *IEEE Transactions on Image Processing,* 13(7): pp. 937-951, 2004

[8] T. Hashimoto, Y. Izawa, H. Yokoyama, T. Kato and T. Morizumi, “A new video/computer method to measure the amount of overall movement in experimental

animals (two-dimensional object-difference method)", *Journal of Neuroscience Methods,* 91(1-2):pp.115-22, 1999

[9] W.R. Christopher, A. Azarbayejani, T. Darrell and A. Pentland. "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* volume 19, pp. 780-785, 1997

[10] J. Serra, (1983), Image Analysis and Mathematical Morphology, *New York: Academic Press*

[11] R.M Haralick and L.G. Shapiro, (1992), Computer and Robot Vision, *Addison Wesley*

[12] M. Lindstrom and M. Zind, "Model-Based Background Subtraction System Application Domain: Pedestrian Tracking", *M.Sc. Thesis,* Department of Software and Media Technology, Aalborg University, Esbjerg, Denmark, January 2004

[13] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction", *Proceedings of the 17th International Conference on (ICPR'04)*, volume 2, pp. 28 -31, 2004

[14] Z. Zivkovic and Ferdinand V.D.H., "Efficient adaptive density estimation per image pixel for the task of background subtraction", *Journal Pattern Recognition Letters,* volume 27, pp. 773 – 780, 2006

[15] P. KaewTraKulPong and R. Bowden, "An Improved Adaptive Background Mixture model for Real-Time Tracking with Shadow Detection", *2'nd European Workshop on Advanced Video Based Surveillance Systems (AVBS '01)*, September 2001

[16] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking", *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, volume 2, pp. 246-252, June 1999.

[17] T. Horprasert, D. Harwood, L.S. Davis "A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection", *IEEE (ICCV'99) FRAME-RATE WORKSHOP*, 1999

[18] J.M. Park, C. G. Looney, H.C. Chen, "Fast Connected Component Labeling Algorithm Using A Divide and Conquer Technique", *CATA 2000 Conference on (*

Computers and Their Applications), pp. 1-8, December 2000

[19] M. Karaman, L. Goldmann, D. Yu and T. Sikora, "Comparison of Static Background Segmentation Methods", *Proceedings of the SPIE,* volume 5960, pp. 2140-2151, 2005

[20] J. Gallego and M. Pardas, "Enhanced Bayesian Foreground Segmentation using Brightness and Color Distortion Region-Based Model for Shadow Removal", *17th IEEE International Conference (ICIP),* pp. 3449 – 3452, 2010

[21] R.H. Evangelio and T. Sikora, "Static Object Detection Based on a Dual Background Model and a Finite-State Machine", *EURASIP Journal on Image & Video Processing,* 2011

[22] A. Amato, M.G. Mozerov, F.X. Roca and J. Gonzalez, "Robust real-time background subtraction based on local neighborhood patterns", *EURASIP Journal on Advances in Signal Processing - Special issue on advanced image processing for defense and security applications,* 2010

[23] K. Kim, T.H. Chalidabhongse, D. Harwood, L.S. Davis, "PDR: A Performance Evaluation Method for Foreground-Background Segmentation Algorithms", *EURASIP Journal on Applied Signal Processing,* 2006

[24] A. Ilyas, M. Scuturici and S. Miguet, "Real Time Foreground-Background Segmentation Using a Modified Codebook Model", *Advanced Video and Signal Based Surveillance, AVSS '09,* pp. 454-459, 2009

[25] M.H. Sigari and M. Fathy, “Real-time Background Modeling/Subtraction using Two-Layer Codebook Model”, *Proceedings of the International Multi Conference of Engineers and Computer Scientists (IMECS)*, pp.717-720, 2008

[26] CAVIAR Test Case Scenarios, *http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1*, accessed on 24.05.2012

[27] Wikipedia, “Connected Component Labeling”, *http://en.wikipedia.org/wiki/Connected-component_labeling*, accessed on 02.08.2012

[28] R. Fisher, S. Perkins, A. Walker and E. Wolfart, "*Connected Component Labeling*”, (2003)

[29] A.M. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction”, ECCV 2000, volume 1843, pp. 751–767, (2000)

[30] A. Mittal, N. Paragios, “Motion-based background subtraction using adaptive kernel density estimation” *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, pp. 302–309, (2004)

[31] N.S. Naraghi, “A Comparative Study of Background Estimation Algorithms”, *M.Sc. Thesis*, Department of Electrical and Electronic Engineering, Eastern Mediterranean University, Gazimağusa, North Cyprus, September 2009

[32] G.V. Jaime “Foreground segmentation and tracking based on foreground and background modeling techniques”, *M.Sc. Thesis*, Departament de Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya, Barcelona, Spain, 2009

[33] D. L. Olson, and D. Delen “Advanced Data Mining Techniques”, *Springer, 1st edition* (February 1, 2008), page 138, ISBN 3-540-76916-1

[34] R. Radke, S. Andra, O. Al-Kofahi and B. Roysam, “Image Change Detection Algorithms A Symmetric Survey”, *IEEE Transactions on Image Processing*, volume 14, pp. 294-307, 2005

[35] T. Baloch, “Background Subtraction in Highly Illuminated Indoor Environment”, *M.Sc. Thesis*, Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, India, January 2010

[36]Wikipedia, “Precision and Recall”, *http://en.wikipedia.org/wiki/Precision_and_recall,* accessed on 02.08.2012

[37] Y. Benezeth, P.M. Jodoin, B. Emile , H. Laurent and C. Rosenberger, “Comparative Study of Background Subtraction Algorithms”, *Journal of Electronic Imaging*, volume 19, Issue 3, pp. 033003-033003-12 (2010)

[38] N. M. Oliver, B. Rosario, and A. P. Pentland, “A Bayesian computer vision system for modeling human interactions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* volume. 22, pp. 831–843, (2000)

[39]Wikipedia, “F1 score”, *http://en.wikipedia.org/wiki/F1_score*, accessed on 04.08.2012