A Comparative Study of Background Estimation Algorithms

Nima Seif Naraghi

Submitted to the
Institute of Graduate Studies and Research
in partial fulfilment of the requirements for the Degree of

Master of Science
in
Electrical and Electronic Engineering

Eastern Mediterranean University
September 2009,
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director (a)

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Electrical and Electronic Engineering.

Assoc. Prof. Dr. Aykut Hocanın
Chair, Department of Electrical and Electronic Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Electrical and Electronic Engineering.

Assoc. Prof. Dr. Erhan İnce
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Hasan Demirel _____
2. Assoc. Prof. Dr. Erhan İnce _____
3. Asst. Prof. Dr. Önsen Toygar _____

ABSTRACT

A COMPARATIVE STUDY OF BACKGROUND ESTIMATION ALGORITHMS

Segmenting out mobile objects present in frames of a recorded video sequence is a fundamental step for many video based surveillance applications. A number of these applications can be listed as: detection and recognition, indoor/outdoor object classification, traffic flow monitoring, lane fullness analysis, accident detection etc. To achieve robust tracking of objects in the scene systems are required to have reliable and effective background estimation and subtraction units. There are many challenges in developing an all round good background subtraction algorithm. Firstly the method(s) chosen must be robust against illumination changes. Second then should avoid detection of non-stationary backgrounds (swaying grass, leaves, rain, snow etc.) and shadows cast by objects blocking sun light. Finally they should be quick in adapting to stop and start of vehicles in urban traffic. Therefore high precision and computational complexity issues are very important while trying to choose an algorithm for a particular environment.

In this thesis we have focused on five different background subtraction algorithms. The methods which attracted considerable interest in the literature and seemed to have fairly good characteristics were selected and implemented. These were namely, approximated median filtering, mixture of Gaussians model, progressive background estimation method and histogram/group-based histogram approaches. These techniques were tested under different environments (using test sequences) and also compared in a quantitative way using some synthetic video.

Also the work entailed an effective shadow removal technique which is used to avoid detection of shadow pixels as part of the foreground mask.

The results show some critical tradeoffs between precision and speed of the process. For instance, although approximated median filtering seems to be a suitable approach due to its simplicity in computation, it fails to detect foreground objects accurately when the background scene contains movements, in addition it is slow in the case of adapting to frame changes which makes this algorithm impractical for many outdoor applications.

The results of progressive method indicate that the algorithm is able to handle the adaptation or deal more effectively than approximated median filtering with even better accuracy for foreground extracting in expense of slightly losing the performance speed. However, the background movement problem (shaking leaves, flag in the wind, flickering, etc) still stands.

Mixture of Gaussians based results was promising in both adaptation and precision however the method's sensitivity to transient stops and its heavier computational complexity were its main drawbacks. Finally although the group based histogram was still too sensitive to fluctuation of light it led to acceptable results introducing itself as a reliable background-foreground segmentation method for its ability to deal with transient stops.

Keywords: Temporal Median Filtering,Background estimation,Mixture of Gaussians background estimation, Median filtering, Histogram, Precision and recall, Shadow removal

ÖZET

ARKA PLAN KESTİRİM ALGORİTMALARI ÜZERİNE KARŞILAŞTIRMALI BİR ÇALIŞMA

Bir video dizinini oluşturan çerçevelerdeki hareketli nesnelerin bölütlenmesi birçok video tabanlı sistem için temel bir adım teşkil eder. Bu uygulamalardan bazıları aşagıdaki gibi sıralanabilir: kestirim ve tanıma, bina içi veya dışı ortamlarda nesne sınıflandırması, trafik akış hesaplaması, şerit doluluk analizi, kaza algılama vb. İzlenen alandaki nesnelerin sağlıklı takibi için güvenilir ve etkili arkaplan tahmin ve ayrıştırma üniteleri gerekmektedir**.** Bütün yönleri ile iyi bir algoritma geliştirmek hemen hemen imkansızı istemek gibidir. İlk olarak seçilen yöntemler aydınlatmada meydana gelebilecek değişikliklere karşı dayanıklı olmalıdır. Daha sonra algoritmalar sabitliği devamlı değişen nesneleri (sallanan ot ve yapraklar, yağmur ve kar gibi) arka planın bir parçası olarak almamalıdırlar. Ayrıca algoritmalar güneş ışığının bloke edilmesinden oluşan hareketli gölgeleri de arka plandan ayırabilmelidirler. Son olarak şehir içi trafiğinde sıkça karşılaşılan durma ve hareket etmelere karşı arka planı hızlı bir şekilde adapte edebilmelidirler. Bu yüzden yüksek doğruluk ve hesaplama karmaşıklığının gerçek zamanlı çalışacak kadar az olması önemli noktaları teşkil etmektedir. Bu tezde dört ayrı arkaplan çıkarma algoritmasına (background subtraction algorithms) odaklanılmıştır. Literatürde en çok referans almış ve iyi benzetim sonuçları veren yöntemler seçilmiş ve gerçekleştirilmiştir. Bu beş yöntem sırası ile yaklaşık ortanca süzgeçleme yöntemi, Gauss fonksiyonları karışım modeli, aşamalı arka plan kestirim yöntemi ve histogram/grup-tabanlı histogram yöntemleridir. Bu teknikler farklı ortamlar için değişik test video dizinleri

kullanarak değerlendirilmiş ve ayrıca sentetik video dizinleri kullanılarak kıyaslamalı olarak karşılaştırılmıştır. Ayrıca, etkili bir gölge kaldırma tekniği tanıtılıp tahmini önplanlara uygulanmıştır. Sonuçlar işlemin kesinliği ve hızı arasında bazı kritik ödünleşimler göstermiştir. Örneğin approximated median filtering hesaplamadaki kolaylığı sebebiyle uygun bir yaklaşım olarak görülse de geri plandaki mekan hareket içerdiği taktirde önplandaki nesneleri doğru olarak tespit edememektedir. Ayrıca bu yöntem, çerçeve değişimlerine uyumu açısından da yavaştır ki bu durum sozkonusu algoritmayı birçok dış uygulama için kullanışsız kılmaktadır. Aşamalı arkaplan kestirim algoritmasıyla elde edilen sonuçlar göstermektedir ki bu algoritmanın adapte olma becerisi yaklaşık ortanca süzgeçli yönteme göre daha etkilidir. Çok az hız kaybına rağmen önplan çıkartması daha kesin bir biçimde yapılabilmektedir. Buna rağmen geri plan hareket problemi hala (sallanan yapraklar, dalgalanan bayrak, titreme, vb) devam etmektedir.

Gauss fonksiyonları karışımlı arkaplan kestirim yöntemi keskinlik ve adaptasyonda iyi olmasına rağmen geçici duraklama ve kalkışlara hassas ve işlem zamanı açısından daha uzun bir zaman aralığı gerektiren bir yöntemdir. Son olarak, grup temelli histogram yöntemi ışık dalgalanmalarına karşı çok hassas olmasına karşın duraklama ve kalkmalara karşı başarılı olması nedeni ile güvenilir ve başarılı bir önplan-arkaplan bölütleme yöntemi olarak kabul edilebilir.

Anahtar kelimeler: zamansal ortanca süzgeçleme, aşamalı arkaplan kestirimi, Gauss fonksiyonları karışımlı arkaplan kestirimi, keskinlik ve hatırlama ölçekleri, gölge belirleme ve kaldırma

ACKNOWLEDGEMENTS

Words fail me to express my gratitude to Dr. Erhan Ince for his Supervision, advice, and guidance from the very early stage of this research as well as his extraordinary patience throughout the work. Above all and the most needed, he provided me unflinching encouragement and support in various ways. His truly scientist intuition has made him as a constant oasis of ideas and passions in science, which exceptionally inspire and enrich my growth as a student, a researcher and a scientist want to be. I am indebted to him more than he knows.

I gratefully acknowledge the head of the department Assoc.Prof.Aykut Hocanin for providing me an opportunity of studying in the department of Electrical and Electronic Engineering as a research assistant.

I would like to extend my thanks to all of my instructors in the Electrical and Electronic Engineering department who helped me so much for increasing my knowledge.

It is a pleasure to pay tribute also to the sample collaborators. To Meysam Dehghan, Saameh G.Ebrahimi, Amir YavariAbdi, Mehdi Davoudi, Talayeh Farshim and all the people who were important to successful realization of thesis.

Last but not least, my deepest thank goes to my family for their support and encouragement for which I am indebted forever.

TABLE OF CONTENTS

ABSTRACT iii

ÖZET v

ACKNOWLEDGEMENTS vii

LIST OF TABLES xi

LIST OF FIGURES xii

LIST OF SYMBOLS xiv

LIST OF ABBREVIATIONS xv

CHAPTER 1 1

INTRODUCTION 1

1.1 Literature Review 3

1.1.1 Non-Recursive Techniques 5

1.1.2 Recursive Techniques 11

1.2 Thesis Review 14

1.3 Previous Departmental Works and Thesis Related Publications 15

CHAPTER 2 16

BACKGROUND ESTIMATION ALGORITHMS 16

2.1 Temporal and Approximated Median Filtering: 16

2.2 Mixture of Gaussians Model 19

2.2.1 Current State Estimating 24

2.2.2 Approximating Posterior Probabilities 25

2.2.3 Estimating Parameters 26

2.2.4 Online Updating 27

2.2.5 Foreground Segmentation 28

2.3 Progressive Background Estimation Method 30

2.3.1 Partial backgrounds 31
2.3.2 Histogram of Pixels 33
2.3.3 Histogram Table 35
2.4 Group-Based Histogram 37
2.4.1 Window Size Selection 39
2.4.2 Mean Estimation 40
2.4.3 Variance Estimation 41
2.4.4 Foreground Segmentation 42
CHAPTER 3 44
SHADOW REMOVAL 44
3.1 Shadow Removal Algorithm 47
3.2 Simulation Results 49
CHAPTER 4 50
SIMULATION RESULTS AND PERFORMANCE ANALYSIS 50
4.1 Ground Truth 50
4.2 Classification of Pixels 52
4.3 Recall 53
4.4 Precision 53
4.5 Data Analysis 53
CHAPTER 5 59
CONCLUSION AND FUTURE WORK 59
5.1 Conclusion 59
5.2 Future Work 60

REFERENCES 61

APPENDICES 68

Appendix A: Novel Traffic Lights Signaling Technique Based on Lane Occupancy Rates 69

Appendix B: Traffic Analysis of Avenues and Intersections Based on Video Surveillance from Fixed Video Cameras 74

LIST OF TABLES

Table 1: Estimation of error rate of Gaussian mean using histogram and *GBH*..... 40

Table 2: Recall and precision results for background estimation algorithms. 54

Table 3: Performance comparison of algorithm with respect to time..... 57

Table 4: Required number of frames to generate acceptable foreground masks..... 58

LIST OF FIGURES

Figure 1: Foreground-Background detection using temporal median filtering 17

Figure 2: Foreground-Background detection using *AMF* 19

Figure 3: (*R*,*G*) scatter plots of red and green values of a single pixel 21

Figure 4: The 1D pixel value probability 23

Figure 5: The posterior probabilities plotted as functions of *X* 24

Figure 6: Background estimation using *MoG* Model with *K*=5, *T*=0.85 30

Figure 7: Generation of Partial Backgrounds 33

Figure 8 : The partial backgrounds and histograms 33

Figure 9: The counts value for a certain intensity index *k* of a pixel 35

Figure 10: Estimated background using progressive method. 36

Figure 11: Extracting foreground objects using progressive method 37

Figure 12: Statistic analysis of pixel intensity. 42

Figure 13: Estimated Background using *GBH* method 43

Figure 14: Extracting foreground objects using *GBH* method 43

Figure 15: Object merging due to shadows 45

Figure 16: *HSV* color space 46

Figure 17: The correct identification of objects after shadow removal 48

Figure 18: Custom video recorded at Yeni-İzmir Junction 49

Figure 19: Video sequence Highway II 49

Figure 20 : Typical frame of synthetic video-2 51

Figure 21: Comparing the adverse effect of transient stops 56

Figure 22: Adverse effect of late background generation, using *AMF* 56

Figure 23: Visual comparison between algorithms in handling multi-modal scenes. 57

LIST OF SYMBOLS

X_t	Intensity observed at time *t*
$\theta_{\boldsymbol{k}}$	Parameters of the k^{th} distribution
φ	Total set of parameters
$f_X(X\|\varphi)$	Combined distribution of *X*
$w_{\boldsymbol{k}}$	Weight of the k^{th} distribution
$\mu_{\boldsymbol{k}}$	Mean of the k^{th} distribution
$\sigma_{\boldsymbol{k}}$	Standard deviation of the k^{th} distribution
$P(k\|x,\varphi)$	Posterior Probability
$d_{k,t}$	Distance from k^{th} distribution at time *t*
$M_{k,t}$	Match indicator
α_t	Learning rate
$S(t)$	Image sequence
$I(t)$	Input frame at time *t*
$B_i(t)$	Partial Background at time *t*
$h_p(t)$	Histogram of intensities for pixel *p*
v	Counts for each intensity
w	Averaging filter window width
$S_k(x,y)$	Luminance of shadowed pixel (*x*,*y*)
$E_k(x,y)$	Irradiance of location (*x*,*y*) at instant *k*
$\sum_k$	Covariance matrix

LIST OF ABBREVIATIONS

TMF	Temporal Median Filtering
AMF	Approximated Median Filtering
MoG	Mixture of Gaussians
RGB	Red Green Blue color space
HSI	Hue Saturation Intensity color space
YUV	Luminance Chrominance color space
HSV	Hue Saturation Value color space
EM	Expectation Maximization Algorithm
PM	Progressive estimation method
GBH	Group-based histogram method
SP	Shadow pixels
TP	True positive
FP	False positive
TN	True negative
FN	False negative

CHAPTER 1

INTRODUCTION

Video based surveillance systems (VBSS) employ machine vision technologies to automatically analyze traffic data collected by wired CCTV cameras and/or wireless IP camera systems. VBSS can be used to monitor highway conditions, intersections, and arterials for detection of accidents, it can be used to compute traffic flow, and for vehicle classification and/or identification. VBSS systems are of three different types:

1) Tripwire Systems,

2) Tracking Systems,

3) Spatial Analysis based systems.

In Tripwire systems the camera is used to simulate usage of a conventional detector by using small localized regions of the image as detector sites. Such a system can be used to detect the state of a traffic light (red, yellow, green) or check if a reserved section has been violated or not. Tracking systems detect and track individual vehicles moving through the camera scene. They provide a description of vehicle movements (east bound, west bound, etc.) which can also reveal new events such as sudden lane changes and help detect vehicles travelling in the wrong direction. Tracking systems can also compute trajectories and conclude on accidents

when different trajectories cross each other and then motion stops. Spatial analysis based systems on the other hand concentrate on analyzing the two-dimensional information that video images provide. Instead of considering traffic on a vehicle-to-vehicle basis, they attempt to measure how the visible road surface is being utilized.

Conventional approaches of traffic surveillance include manual counting of vehicles, or counting vehicles using magnetic loops on the road. The main drawback of these methods, besides the fact that they are costly is that these systems can only count but they cannot differentiate or classify.

Major part of the existing research and applications on traffic monitoring is dedicated to monitoring vehicles on highways which carry heavy traffic volumes and are incident prone. However, successful and efficient traffic monitoring at cross-sections of the roads in crowded urban areas is also an important issue for road engineers who are to develop new roads that will ease up the traffic load of the city. Furthermore the traffic flow in the city can be displayed at a traffic control center by combining information from various video streams and this information can be exploited for re-directing flow of traffic intelligently.

Background subtraction is a common approach for identifying the moving objects (foreground objects) in a video sequence. Each video frame from the sequence is compared against a reference or background model. Once the reference is computed (often called a *background model*), then it will be updated with each newly arriving frame by exploiting different algorithms. Current frame pixels with considerable deviation from the background model are accounted to be moving objects.

Although many background subtraction methods are listed in the literature, foreground detecting specially for outdoor scenes is still a very challenging problem.

The performance of VBSS will vary based on several environmental changes like the ones listed below:

- Variable lighting conditions, during sunset and sunrise
- Camera angle, height and position
- Adverse weather conditions such as fog, rain, snow, etc
- Presence of camera vibration due to wind and heavy vehicles

Another important consideration while trying to choose an appropriate background estimation method is the time required for processing a frame. If a system has to run in real-time, its computational complexity should not be too high. The background modeling approach must also be robust against the transient stops of moving foreground objects and yet maintain a good accuracy.

Eliminating the cast shadows as undesired parts of the detected foreground mask has become a standard pre-processing step in many applications since moving shadows would affect the detection and identification processes in a negative manner. In this work only the HSV color space based shadow removal algorithm will be mentioned as an example.

1.1 Literature Review

In the literature there are many proposed background modeling algorithms. This is mainly because no single algorithm is able to cope with all the challenges in this area. There are several problems that a good background subtraction algorithm must resolve. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background objects such as swaying leaves, grass, rain, snow, and shadows cast by moving objects. Finally, the background

model should be developed such that it should react quickly to changes in background such as starting and stopping of vehicles.

Background modeling techniques could be classified into two broad categories as: 1) Non-Predictive Modeling, and 2) Predictive Modeling. The former tries to model the scene as a time series and creates a dynamic model at each pixel to consider the incoming input using the past observations and utilizes the magnitude of deviation between the actual observation and the predicted value to categorize pixels as part of the foreground or background. However, the latter one neglects the order of the input observations and develops a statistical (probabilistic) model such as PDF at each pixel.

According to Cheung and Kamath [2], background adaptation techniques could also be categorized as: 1) non-recursive and 2) recursive. A non-recursive technique estimates the background based on a sliding-window approach. The L observed video frames are stored in a buffer, considering the existing pixel variations in the buffer the background image will be estimated. Since in practice the buffer size is fixed as time passes and more video frames come along the initial frames of the buffer are discarded which makes these techniques adaptive to scene changes depending on their buffer size. However, in the case of adapting to slow moving objects or coping with transient stops of certain objects in the scene the non-recursive techniques require large amount of memory for storing the appropriate buffer. With a fixed buffer size this problem can partially be solved by reducing the frame rate as they are stored.

On the contrary the recursive techniques instead of maintaining a buffer to estimate the background they try to update the background model recursively using either a single or multiple model(s) as each input frame is observed. Therefore, even

the very first input frames are capable to leave an effect on new input video frames which makes the algorithm adapt with periodical motions such as flickering, shaking leaves, etc. Recursive methods need less storage in comparison with non-recursive methods but possible errors stay visible for longer time in the background model. The majority of schemes use exponential weighting or forgetting factors to determine the proportion of contribution of past observations.

In this thesis we tried to neglect the methods which require a long period of initialization such as the ones described in [3] which is characterized by eigen-images and [4] using temporal maximum-minimum filtering along with maximum inter-frame differencing for entire background model, and focused more on adaptable background models.

1.1.1 Non-Recursive Techniques

The sub-sections below give a brief summary of some non-recursive techniques.

1.1.1.1 Frame Differencing

This technique is probably one of the simplest among the background subtraction algorithms. In the literature it is also referred to as the temporal differencing approach. Simply, the previous frame is considered as the estimate for the background at each time interval and foreground objects are detected by taking the difference of the current input frame and the current reference. Since this method uses only one frame to estimate the background it is quite sensitive to transient stops [5,6], and can easily be affected by camera noise and illumination changes[7]. This method also fails in correctly segmenting foreground objects if the size of the object is large and its color is uniformly distributed. In the literature this problem is referred

to as the aperture problem.

1.1.1.2 Average Filtering

Average filtering approach creates the background model by averaging the input frames over time. This is based on the assumption that since the foreground is moving its presence is transient, therefore after averaging, the proportion of object in the estimated background will become small. If one considers intensity of a certain pixel over time and assumes that the object intensity is visible for just a specific period of time (for instance 3 video frames) then the effective object intensity in the background model based on that pixel will be 3/n, where n is the total number of averaged frames.

Hence if the objects are large in size or if they move slowly their contribution becomes more and more significant. Also shadows in same position(s) where the object was detected in the previous frame(s) will appear in the background model. They are generally referred to as ghosts in the literature. Furthermore, average filtering is also known to show poor performance in the crowded scenes where lots of moving objects or bi-modal backgrounds (flickering, shaking leaves, flag in the wind, etc) has to be dealt with [8].

Koller *et al.* [15] has tried to improve the robustness to illumination changes by means of implementing a moving-window average algorithm along with an exponential forgetting factor. This trick may be helpful in suppressing some errors due to illumination changes but it will obviously fail in the case of slow moving objects and other shortcomings which were mentioned in prior to this method, since background is updated using both the information from the previous background and foreground.

Keeping these drawbacks in mind, indoor applications with little illumination changes and fast moving objects with limited sizes will be the most suitable environments for applying the average filtering method. The last step to modify this algorithm is to exclude identified foreground pixels based on our estimated background model in the updating procedure.

1.1.1.3 Median Filtering

Median filtering is widely used in many applications and has been extensively discussed in the literature [9],[10],[15],[17]. In this approach, the background estimate is computed as the median of all the pixel values stored in a buffer at each pixel location. Here an assumption is made based on the fact that the pixels belonging to the background scene are going to be sighted more than half of the length of the entire video frames in the buffer which will result in slow updating procedure due to the fact that if a static object is added to the scene it takes time at least half of the entire stored frames to become part of the background.

Replacing median by its color counterpart "medoid" can lead to color background estimation [10]. In spite of average filtering the median filtering is capable of saving boundaries and existing edges in the frame without any blurring, therefore gives a sharper background in comparison to the previous method.

1.1.1.4 Minimum-Maximum Filter

This method uses three different values to decide whether a certain pixel belongs to background or not. These three values are minimum intensity of each pixel during a specific time period while assuming no foreground objects are available in the scene (training sequence), the maximum intensity of each pixel and the maximum possible change based on the maximum intensity difference between every two consecutive frames [13].

1.1.1.5 Linear Predictive Filter

Toyama *et al.* [14] estimates the background model through applying linear predictive filters to predict the values corresponding to the background based on the available *k* pixel-samples stored in a buffer. Wiener filter is one of the most commonly used filters in such algorithms. If the accumulated pixel errors exceed the predicted value too much (several times) those pixels will then be considered as part of the foreground. The coefficients of the filter are computed at each frame time due to covariance of the samples, therefore this algorithm is not applicable in real-time procedures. Linear prediction using the Kalman filter was also used in [15], [16], [17].

Monnet *et al.* [18] has used an autoregressive form of filtering for predicting the newly added input frame. In [18] two different steps have been used to create and preserve the background model. One of the steps was responsible to update the states incrementally and the other one replaced the states of variation by means of the latest observation map. Other methods can also be considered for prediction. For instance, principal component analysis [19], [20] refers to a linear transformation of variables that keeps from *n* operators the most significant magnitude of variation among the training data in hand. Computing the basis vectors from the available data set is done using singular value decomposition concept.

Unfortunately evaluation of these basis components for vectors containing many data values is very time consuming computation. One solution to this problem is by downsizing the procedure to block level and perform the computations on each block of the image independently.

1.1.1.6 Non-Parametric (NP) Modeling

In NP modeling, the main interest is focused on estimating the corresponding probability density function (*pdf*) at each pixel. Nonparametric methods compute the density function directly from the observed data and there is no prior assumption or knowledge regarding the underlying distribution. Therefore unlike its other counterparts, there will be no model selection and distribution parameter estimation.

$$f(I_t = u) = \frac{1}{L} \sum_{i=t-L}^{t-1} K(u - I_i) \qquad (1.1.1.6.1)$$

In the above equation K(.) is the kernel estimator which most of the time is assumed to be Gaussian. The pixels from the newly input video frame named I_t is considered as foreground related pixels when the probability of such occurrence $f(I_t)$ is below a specified threshold. It has been shown by [21] and [22] that Kernel density estimators are able to converge asymptotically to practically any *pdf*. In fact, [18] explains that all other existing non-parametric density estimation techniques can be shown to be a variants of the kernel method. For example histogram based algorithms which will be detailed in this thesis also are some of these techniques.

As mentioned before kernel density estimator algorithm does not include any assumption for the general shape of the underlying distribution and it owns the flexibility to reach any type of distribution for as long as it is fed with enough data samples. Theoretical proof of this issue can be found in [21].

Flexibility to converge to almost any *pdf* makes this method appropriate to estimate the areas containing color-distributions. Unlike the Gaussian Mixture Model which is a parametric model which tries to fit Gaussian distribution(s) to each pixel, the kernel density estimation is a more general technique with no fixed parameters.

In addition, the adaptation is performed by only observing the newly added data instead of going through complex computation procedures hence it is simpler and less time consuming. However, it should also be mentioned here that while implementing kernel density estimation method, special care should be taken in selecting appropriate kernel bandwidth (scale). The choice of kernel bandwidth is a very critical task. If the bandwidth is chosen too small it will lead to rough or even misleading density estimation, while if the kernel is chosen too wide it will result in an over-smoothed density estimate [21].

Since different pixels have different intensity variations over time it's not practical to implement a single window for all pixels. A different kernel should be used for each pixel. Even different kernel bandwidths are required for separate color channels. Although wide range of kernel functions have been implemented in the literature, the majority of the algorithms use Gaussian kernel due to its specific characteristics such as continuity, differentiability, and locality. In practice selecting a kernel shape (function) has nothing to do with fitting a distribution and kernel Gaussian is only responsible to weight the data samples according to its shape.

Computational cost is one of the most notable shortcomings of the Kernel density estimation algorithm. Also, it has serious challenges when the training sequences are disturbed by the presence of foreground objects and takes quite long for algorithm to estimate the real background. [23]

In [24], Elgammal explained that for a given new pixel, background model updating process can be performed in two different ways; either by selective updating or blind updating. In the former technique, the observed sample from the input frame is added to the model if and only if it belongs to the estimated background. However, in the latter one, simply every new sample is added regardless

of its assigned category. Both of these approaches have their advantages and disadvantages.

The selective updating method raises the ability of algorithm in detecting the foreground objects more accurately, due to the fact that object related pixels are excluded from the updating procedure. However in the case of any wrong decisions, it will lead to persistent errors in future decisions. This undesired situation in the literature is referred to as the deadlock situation.

The blind updating approach is not affected by such a problem because it does not differentiate between samples as it updates the background model however this will result in poor detection of the targets (more false negatives). This problem can partially be solved by including less proportion of foreground-object related pixels through increasing the time window of sampling process [24]. When the time window is made wider, the adaptation process will be slowed down and therefore more false positives will be visible in foreground representation.

1.1.2 Recursive Techniques

What follows below is a summary of the recursive techniques that can be used for background estimation and subtraction.

1.1.2.1 Approximated Median Filter

Shortly after the non-recursive median filtering became popular among the background subtraction algorithms, McFarlane and Schofield presented in [25] a simple *recursive* filter for estimating the median of each pixel over time. This method has been adopted by some for background subtraction for urban traffic monitoring due to its considerable speed. This method is explained in the following chapter and will be examined along with the other selected methods for evaluation of

its pros and cons.

1.1.2.2 Single Gaussian

As mentioned earlier, calculating the average image of a sequence of frames and then subtracting each new input frame and checking the difference values against a predefined threshold is one of the simplest background removal techniques. In [26] Wren presents an algorithm to assign a normal distribution with a certain mean and standard deviation to each estimated background pixel using a color space named YUV color space.

This algorithm requires *t* frames to estimate the mean μ and the standard deviation σ in each color component separately:

$$\mu(x,y,t) = \sum_{i=1}^{t} \frac{p(x,y,i)}{t} \qquad (1.1.2.2.1)$$

$$\sigma(x,y,t) = sqrt\left(\sum_{i=1}^{t} \frac{p^2(x,y,i)}{t} - \mu^2(x,y,t)\right) \qquad (1.1.2.2.2)$$

Here, *p(x,y,t)* is the pixel's current intensity value at the location (*x*,*y*) at a given time *t*. After computing the parameters, a pixel is considered as a part of the foreground object based on the following formula:

$$|\mu(x,y,t) - p(x,y,t)| > c.\sigma\,(x,y,t) \qquad (1.1.2.2.3)$$

where c is a constant. Even though this method is capable of adapting to indoor environments with gradual illumination changes, it's not able to handle moving background objects like trees, flags, etc.

1.1.2.3 Kalman Filtering

This technique is one the most well known recursive methods specifically for situations where noise is known to be Gaussian. If we assume the intensity values of the pixels in the image follow a normal distribution such as $N(\mu, \sigma^2)$, where simple adaptive filters are responsible for updating the mean and variance of the background model to compensate for the illumination changes and include objects with long stops in the background model. Background estimation using Kalman filtering has been explained in [25] and [27].

Various algorithms can be found in literature that uses Kalman filtering. The main difference between them is the state space they use for tracking. The simplest ones are those which are based only on the luminance [26],[28],[29],[30].

In [31] Kalman and von Brandt added information achieved by temporal derivatives to intensity values to get better results. The following is a summary of this procedure demonstrating the general steps that should be taken to implement this method.

The internal state of the system is shown by B_t the background intensity while $B_t^{'}$, is temporal derivative. Updates are done recursively through:

$$\begin{bmatrix} B_t \\ B_t^{'} \end{bmatrix} = A \,.\, \begin{bmatrix} B_{t-1} \\ B_{t-1}^{'} \end{bmatrix} + \, K_t \,.\, I_t \, - \, H \,.\, A \,.\, \begin{bmatrix} B_{t-1} \\ B_{t-1}^{'} \end{bmatrix} \qquad (1.1.2.3.1)$$

Matrix A describes the background dynamics and H is the measurement matrix. The particular values used in [31] are as follows:

$$A = \begin{bmatrix} 1 & 0.7 \\ 0 & 0.7 \end{bmatrix} \quad , \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad (1.1.2.3.2)$$

The Kalman gain matrix K_t fluctuates between a slow adaptation rate α_1 and a fast adaptation rate $\alpha_2 > \alpha_1$. K_t will be assigned according to whether I_{t-1} is related to foreground or not, based on the following formula:

$$\begin{cases} K_t = \begin{bmatrix} \alpha_1 \\ \alpha_1 \end{bmatrix} & \text{if } I_{t-1} \text{ is foreground} \\ K_t = \begin{bmatrix} \alpha_2 \\ \alpha_2 \end{bmatrix} & \text{otherwise} \end{cases} \qquad (1.1.2.3.3)$$

1.1.2.4 Hidden Markov Models

All of the previously mentioned models are able to adapt to gradual changes in lighting. However, if considerable amount of intensity changes occur, they all encounter serious challenges. Another approach which is capable of modeling a wide range of variations in the pixel intensity is known as Markov Model and it tries to model these variations as discrete states based on modes of the environment, for instance **l**ights on/off or cloudy/sunny skies etc. In [32], a three-state HMM has been represented for modeling the intensity of a pixel in traffic-monitoring applications. In [33], as the algorithm is trying to estimate the background model, the topology of the HMM regarding global image intensity is learned.

The main problem in implementing HMMs in real world applications is twofold: the processing is not real-time since it requires long training periods, and the topology modification to address non-stationary is also computationally intense.

1.2 Thesis Review

In chapter 2, five different algorithms for background modeling will be discussed in detail. These techniques are chosen from the two major classes of background modeling; recursive and non-recursive techniques. Approximated Median filtering and Mixture of Gaussians model are selected from the former group while the progressive background generation, Temporal Median Filtering and group-based histogram approaches belong to the latter group.

Although, two out of three techniques from non-recursive algorithms are based on histograms, there are significant differences between them in data storage and updating procedures. Chapter 3 is dedicated to shadow removal algorithm which is based on *HSV* color space. The simulation results of applying these background estimation methods on different video sequences, which are mostly outdoor traffic scenes, have been provided in chapter 4. The same sets of video sequences were used while testing each individual method in order to understand the advantages and disadvantages of each method. Two quantitative scales called recall and precision have been used to compare the performance of each algorithm. In addition, the performances of algorithms in time domain are compared with respect to each other. Finally the last chapter includes conclusion and future works.

1.3 Previous Departmental Works and Thesis Related Publications

As a result of the work carried out under this thesis two conference publications were made; one in SIU 2009 and the other in ISCIS 2009. A copy of these papers can be found in appendix A.

Earlier works done by H. Kusetoullari which was about speed measurements using surveillance camera would create the reference frame by averaging 10 consecutive frames of the video sequence when there were no vehicles or moving objects in the scene. However, in this thesis five different state of art background estimation techniques have been implemented to obtain the reference frame. In addition in this work the *HSV* color space has been used to detect and remove shadows that constitute part of the foreground image.

CHAPTER 2

BACKGROUND ESTIMATION ALGORITHMS

In this chapter the structure and implementation details of five different background model estimators are presented. The first two are based on the median operator and are statistical approaches, the third method which is also known as mixture of Gaussians model (*MoG*) tries to combine a number of Normal distributions to model the 3-tuple pixel vectors and the last two methods use histogram analysis techniques for background modeling.

2.1 Temporal and Approximated Median Filtering:

As it has been mentioned earlier there are two types of background-foreground segmentation algorithms which use median operator:

1. Temporal Median Filtering (*TMF*)
2. Approximated Median Filtering (*AMF*)

Both of these methods are based on the assumption that pixels related to the background scene would be present in more than half the frames of the entire video sequence. This is true in most of the situations unless in case of heavy traffic flow during the rush hours.

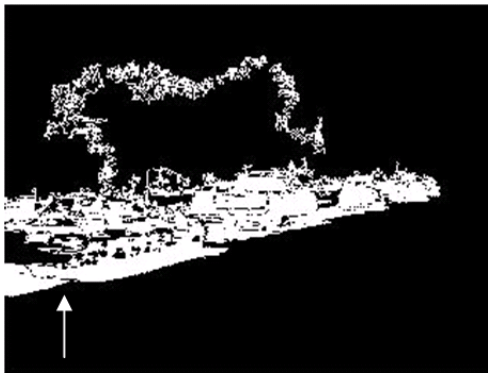
TMF computes the median intensity for each pixel from all the stored frames in the buffer. Considering the computation complexity and storage limitations it is

not practical to store all the incoming video frames and make the decision accordingly. Hence the frames are stored in a limited size buffer. Admittedly the estimated background model will be closer to the real background scene as we grow the size of the buffer. However, speed of the process will reduce and also higher capacity storage devices will be required.

In some cases the number of stored frames is not large enough (buffer limitations), therefore the basic assumption will be violated and the median will estimate a false value which has nothing to do with the real background model. An example where temporal median filtering algorithm fails to extract a proper foreground mask is shown in figure 1 below:

(a) Original frame

(b) Estimated background



(c) The mask of extracted foreground

Figure 1: Foreground-Background detection using temporal median filtering [46].

As can be seen from figure 1, the detected foreground is not acceptable. This problem is partly due to the poor background estimation since the median is not correctly detected from the frames in the buffer and partly the incapability to handle the multi-modal scenes (shaking leaves are incorrectly detected as foreground).

AMF was first introduced by McFaralane and Schofield [25] which uses a simple recursive filter to estimate the median. This filter acts as a running estimate of the median of intensities coming to the view of each pixel.

AMF apply the filtering procedure by simply incrementing the background model intensity by one, if the incoming intensity value (in the new input frame) is larger than the previous existing intensity in the background model. The reverse is also true, meaning that when the intensity of the new input is smaller than background model the corresponding intensity will be decreased by one. It has been proved by [25] that this trend will converge to the median of the observed intensities over time. Therefore unlike *TMF*, this approach does not require storing any frames in a buffer and tries to update the estimated background model online. Hence it is extremely fast and suitable for real time applications.

The background estimate and the corresponding foreground mask shown in figure 2 have been obtained by applying *AMF* to the same video sequence used while testing the TMF technique.

(a) Estimated background (b) The mask of extracted foreground

Figure 2: Foreground-Background detection using *AMF* [46].

It can be seen that foreground mask generated by *AMF* has improved (note the nearest car) since our background quality has become much better, but still the problem related to non-stationary backgrounds remained. In fact this approach is most suitable for indoor applications.

2.2 Mixture of Gaussians Model

The Mixture of Gaussians technique was first introduced by Stauffer and Grimson in [8]. It sets out to represent each pixel of the scene by using a mixture of normal distributions so that the algorithm will be ready to handle multimodal background scenes.

In this thesis, we tried to present and implement the latest version of this technique taking advantage of the available modified versions in the literature. However, the main structure is still the *MoG* model presented in [8].

The *MoG* model is designed such that the foreground segmentation is done by modeling the background and subtracting it out of the current input frame, and not by any operations performed directly on the foreground objects (i.e. directly modeling the texture, color or edges). Second the processing is done pixel by pixel rather than

by region based computations, and finally the background modeling decisions are made based on each frame itself instead of benefiting from tracking information or other feedbacks from previous steps.

In the mixture model each pixel is modeled as a mixture of K Normal distributions. Typically values for K varies from 3 to 7. For $K < 3$, the mixture model is not so helpful since it cannot adapt to multimodal environments and if K is selected a value over 5, often the disadvantage of processing speed reduction (not able to be performed in real time) outweighs the improvement in quality of background model. At any time t, K Gaussian distributions are fitted to the intensities seen by each pixel up to the current time t.

If each pixel intensity would result from specific lighting or from single mode background intensities then it would be feasible to represent the pixel value samples over time with a single distribution but unfortunately in real situation often multiple surfaces along with different illumination conditions appear in the pixel view.

Hence if it's desired to model the background using Gaussian distributions there should be mixture of distributions assigned to each pixel instead of a single one. To illustrate the occurrence of bimodal distributions, (R,G) scatter plots of single pixel at the same location in all frames over time have been shown in figure 3:

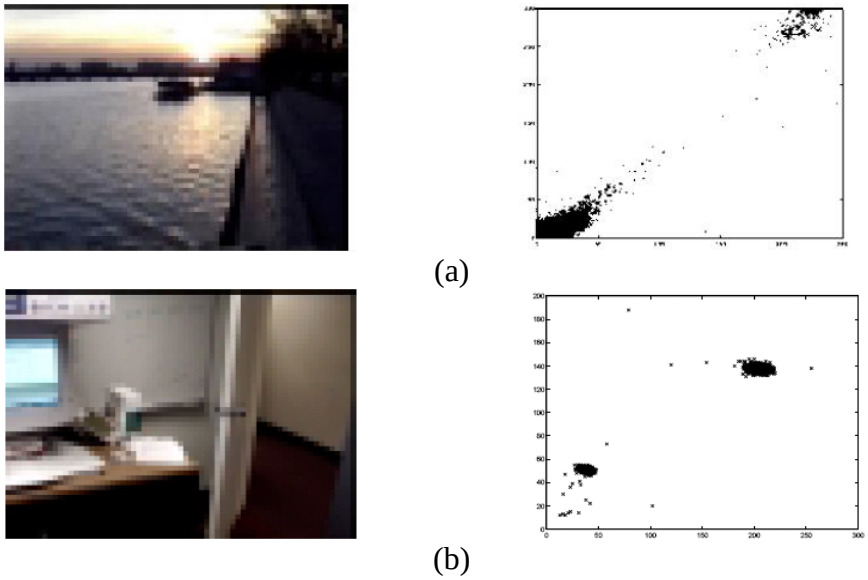


Figure 3: (*R*,*G*) scatter plots of red and green values of a single pixel[8].

The values of a certain pixel over time are called "pixel process". If the gray scale intensities are used for background modeling then pixel process is going to have 1D values (only a series of scalars between 0-255), 2D is also possible while using normalized color spaces or intensity-plus-range and in the case of standard color spaces (*RGB*, *HSI*, *YUV*, etc) triple vectors are going to form our per pixel history. Pixel process can be mathematically described as:

$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i): 1 \leq i \leq t\} \qquad (2.2.1)$$

Where (x_0, y_0) indicates the location of the pixel in the image at any time *t*, *I* represents the image sequence and *X* 's are the intensities of each pixel over time. Therefore there would be scalars in gray-scale or triple vectors in color spaces.

The algorithm should perform in a way that if a foreground object stops for a long period of time consider it as a part of background or while the pixels intensities of the scene under study are affected by illumination changes be able to adapt to the new situation .These requirements indicate that more recent observations may be

more vital for background subtraction hence, the distributions assigned to the pixels should not be weighted equally.

Therefore the observed data samples which are more likely to be a part of background are weighted more than the less probable distributions.

A pixel process *X* is assumed to be modeled by a mixture of *K* Gaussian distributions with parameters set θ_k, one for each distribution as states in equation 2.2.2.

$$f_{(X|k)}(X|k,\theta_k) = \frac{1}{(2\pi)^{n/2}|\sum_k|^{1/2}} e^{-\frac{1}{2}(X-\mu_k)^T|\sum_k|^{-1}(X-\mu_k)} \tag{2.2.2}$$

Where μ_k representing the mean of k^{th}distribution and $\Sigma_k$indicates the covariance of the k^{th} density.

In the *MoG* model theory, two assumptions have been made. Firstly it has been assumed that dimensions of *X* are considered independent. This constraint forces the covariance matrix to be diagonal (hence more easily invertible) having σ_k^2 as its variance along its diagonal components.

The second assumption is that the variances of each channel of the color space, are identical. It should be noted here that single σ_k^2 may be reasonable in linear color spaces as *RGB* but in non-linear cases, such as *HSV*, special care should be taken since this excessive simplification may not work.

Due to the fact that the *K* occurring events are disjoint, if we want to formulate the combined distribution of *X*, we can simply sum up the members of the Gaussian mixtures. Therefore the general formula would be:

$$f_X(X|\varphi) = \sum_{k=1}^{K} P(k) f_{X|k}(X|k,\theta_k) \tag{2.2.3}$$

Here, the density parameter set is $\theta_k = \{\mu_k, \sigma_k\}$ for a given *k* and the total set of parameters is $\varphi = \{w_1, \dots, w_k, \theta_1, \dots, \theta_k\}$. $P(k)$ is the probability of occurrence

for the k^{th} distribution and it represents the amount of contribution by that distribution in the mixture model. Hence $P(k)$ is the weight assigned to that distribution ($P(k) = w(k)$).

Figure 4 below provides an example for a mixture model with three distributions where w_k={0.2,0.2,0.6}, μ_k= {80,100,200} and σ_k = {20,5,10}:

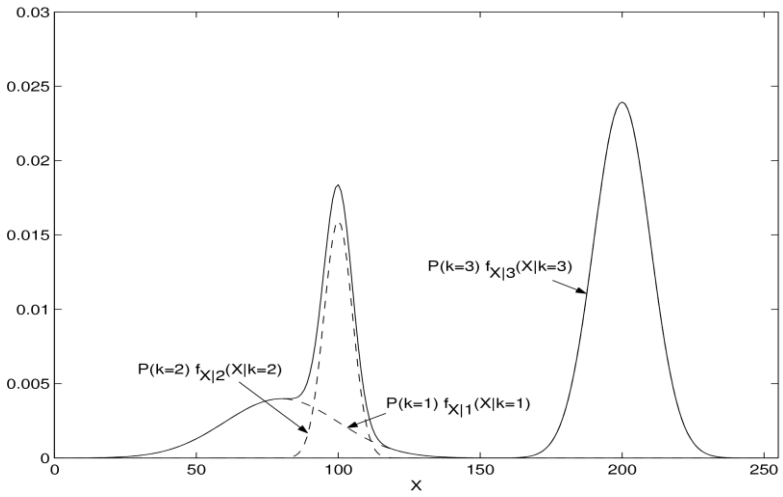


Figure 4: The 1D pixel value probability $\boldsymbol{f_X(X|\varphi)}$ [36].

During the processing, the *MoG* model has to estimate both the parameters and the hidden (unknown) state *k* given the observation *X*. This estimation problem which is referred to as the "maximum likelihood parameter estimation from incomplete data" can be solved by the use of an expectation maximization (*EM*) algorithm [34]. The *EM* algorithm works iteratively and has two main steps:

1. E-step which is responsible for finding the expected value with respect to the complete data in hand (observed data and current estimation of parameters).

2. M-step which is the calculation of maximum likelihood values for parameters based on the available observations.

2.2.1 Current State Estimating

Firstly the model has to distinguish which of the *K* distributions is more likely to describe the new data; that is, it has to estimate the distribution from which *X*, has most probably come from.

Comparing the posterior probabilities $P(\boldsymbol{k}|\boldsymbol{x},\boldsymbol{\varphi})$ which indicate likelihood of the current sample *X* belonging to the $\boldsymbol{k^{th}}$ distribution, will lead us to achieve this goal. A plot of posterior probabilities obtained using equation 2.2.4 and Bayes theorem has been provided in figure 5:

$$P(k|x,\varphi) = \frac{p(k) f_{(x|\mathrm{k})}\,(\mathrm{x}|\mathrm{k},\theta_\mathrm{k})}{f_x(x|\varphi)} \qquad (2.2.4)$$

Here the value of k which maximizes $P(k|x,\varphi)$ will determine the correct distribution from which *X* had come from.

$$\hat{k} = \arg\max_k P(k|x,\varphi) = \arg\max_k w_k\, f_{(X|k)}\,(X|k,\theta_k) \qquad (2.2.5)$$

The preceding equation is true as long as the current input has been generated by one of the distributions in the mixture.

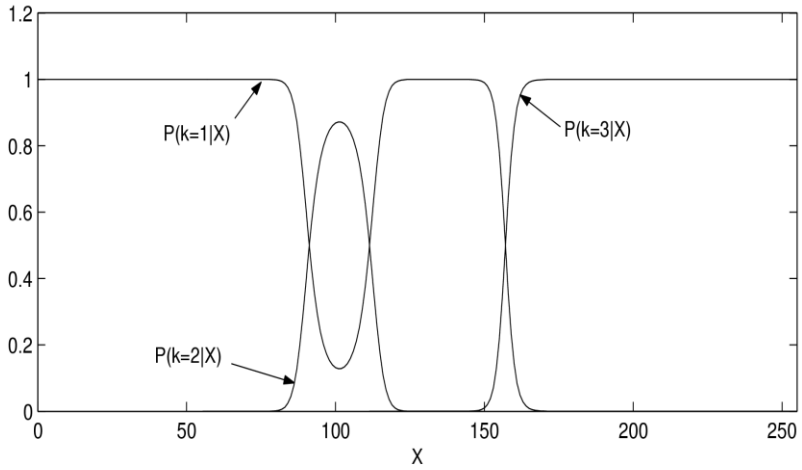


Figure 5: The posterior probabilities $\boldsymbol{P}(\boldsymbol{k}|\boldsymbol{x},\boldsymbol{\varphi})$ plotted as functions of X for each *k*=1, 2, and 3 using the same parameters as in figure 4[36].

Obviously there may be certain points (intensities), which are not covered by any of the existing distributions. For instance if we consider that the new input intensity is X=150 after computing the posterior probabilities depicted in figure 5 the algorithm considers first distribution (k=1) to be fully (almost 100%) responsible for generating the observed value. However it is clear from figure 4 that the value 150 does not belong to any of the three different distributions. This is only due to the fact that only three distributions are considered to cover the whole range of intensities (0-255). This type of challenge would be faced when a previously unseen foreground object steps in the scene. The solution lies in adding an extra distribution with weight w_{k+1} , considering current pixel value as its mean and assigning a high variance to this newly added distribution.

2.2.2 Approximating Posterior Probabilities

As mentioned before the *EM* algorithm needs much iteration to reach the final result, hence implementing an exact *EM* algorithm on each pixel of every frame would be a complicated and time costly procedure. In [8], Stauffer and Grimson developed a method to approximate the posterior probability in a fast and more sensible way through defining matching criteria.

A match is defined as a pixel value falling within 2.5 times the standard deviation of the distribution's mean. To compute the distance (d) from the mean (μ_k) of a certain distribution at time t, the following formulas are applied [36]:

$$d_{k,t} = (\sigma_{k,t} I)^{-1}\left(X_t - \mu_{k,t}\right) \tag{2.2.6}$$

$$d_{k,t}^T d_{k,t} < \lambda^2 \tag{2.2.7}$$

The parameter $M_{k,t}$ in equation 2.2.8 has been chosen to show if a match is found:

$$M_{k,t} = \begin{cases} 1 & match \\ 0 & otherwise \end{cases} \cong P\,(k|X_t, \varphi) \qquad (2.2.8)$$

This is based on the assumption that $P(k|X_t, \varphi)$ is 0 or 1 for most of the X_t values and also it is almost one for only one choice of *k* at a time, since distributions are far enough from each other (refer to figure 4). In other words, when $P(k|X_t, \varphi)$ has a value of one at time *t* for one distribution, the probabilities for other *K*-1 remaining distributions are zero.

In cases when an observed value is located in a position such that it is close to more than one distribution, more than one match may be detected. In this case, the distribution with the highest rank would be selected (Details of rank information can be found in section 2.2.5).

2.2.3 Estimating Parameters

If samples have been observed then the complete data likelihood function is calculated as:

$$P(X_1, X_2, \dots, X_N, k|\varphi) = \prod_{t=1}^{N} w_k \, f_{(X|k)}\,(X_t|k, \theta_k) \qquad (2.2.9)$$

Parameters of φ defined in equation 2.2.3 can be updated by maximizing the expected value of the previous formula with respect to *k*. The details of derivation of such a procedure are too long and complicated but it can be found in [35].

If we assume that processes are stationary and the number of observations (*N*) is fixed, then we have:

$$\widehat{w_k} = \frac{1}{N} \sum_{t=1}^{N} P(k|X_t, \varphi) \qquad (2.2.10)$$

$$\widehat{\mu_k} = \frac{\sum_{t=1}^{N} X_t \, P(k|X_t, \varphi)}{\sum_{t=1}^{N} P(k|X_t, \varphi)} \tag{2.2.11}$$

$$\widehat{\sigma_k}^2 = \frac{\sum_{t=1}^{N} ((X_t - \widehat{\mu_k}) \circ (X_t - \widehat{\mu_k})) P(k|X_t, \varphi)}{\sum_{t=1}^{N} P(k|X_t, \varphi)} \tag{2.2.12}$$

where in equation (2.2.12), ° indicates Hadamard (element by element) multiplication.

2.2.4 Online Updating

The equations (2.2.10) to (2.2.12) are weighted averages of observations by $P(k|X_t, \varphi)$, however, if we want to update our estimated parameters as the program is executed and new samples (inputs) step in, we should convert these averages to an on-line cumulative average by defining a time varying gain $\alpha_t = {}^{1}/_{t}$ and update the algorithm as follows:

$$\widehat{w_{k,t}} = (1 - \alpha_t)\widehat{w_{k,t-1}} + \alpha_t \, P(k|X_t, \varphi) \quad for\ k = 1,2,..K\ , t = 1,2, \dots, t \tag{2.2.13}$$

Note that for each *K*, at any time *t*, $w_{k,t}$ would be a scalar variable.

Considering that the method should be capable to adapt to the recent changes of the scene such as illumination variations, the latest observations should be emphasized more. Therefore just using the equation (2.2.13) will cause problems due to the fact that while the time is passing, *t* is increasing and consequently α_t will decrease. The depletion of α leads to canceling the contribution of $P(k|X_t, \varphi)$ which is related to the current time t. Hence the process is getting more and more insensitive to recent scene variations.

One practical solution is to define a lower bound $\alpha_t = \alpha$ to make the procedure leaky and as soon as the lower bound is reached, the accumulator would start to compute the new values with an exponentially decreasing emphasis [36]. This

part of the algorithm differs from what was presented by Stauffer and Grimson in [8], since they had assumed a fixed α for all time [37].

Also the mean and variance values could be updated using the equations provided below:

$$\widehat{\mu_k} = (1 - \rho_{k,t})\mu_{k,t} + \rho_{k,t}\, X_t \tag{2.2.14}$$

$$\widehat{\sigma_{k,t}}^2 = (1 - \rho_{k,t})\sigma_{k,t}{}^2 + \rho_{k,t}\left((X_t - \widehat{\mu_{k,t}})^{\circ}(X_t - \widehat{\mu_{k,t}})\right) \tag{2.2.15}$$

$$\rho_{k,t} = \frac{\alpha_t P(k|X_t, \varphi)}{\widehat{w_{k,t}}} \tag{2.2.16}$$

Here the newly introduced $\rho_{k,t}$ [36] is also different from the one defined in [8] by a factor of $f_X(X_t|k, \theta_k)$ which results in impractical values for $\rho_{k,t}$ if it is going to be implemented directly.

In [8], full computational benefit of the approximation is not obtained since, $P(k|X_t, \varphi)$ is not used in computing $\rho_{k,t}$ which affects the estimation of $\mu_{k,t}$ and $\sigma_{k,t}$.

In rare situations when there is a surface with low probability of occurrence $w_{k,t} \leq \alpha_t$ the value of $\rho_{k,t}$ may exceed one. There are other techniques available to evade such a problem. For instance by setting $\rho_{k,t} = \alpha_t$, and also keeping the latest matching X_t for each distribution and then updating the parameters using the stored X_t[36].

2.2.5 Foreground Segmentation

The mixture model contains both the distributions of the background model and the foreground model. That's why the minimum logical value for the number of distributions is 3, so that 2 of them can be assigned to handle bimodal background scenes and leave one for describing the foreground.

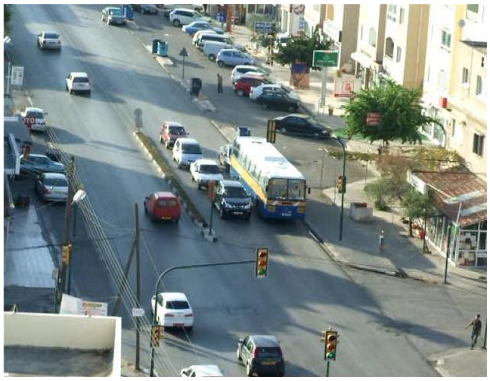
Once the current state *k* is estimated, a scale should be defined to separate the distributions belonging to the background model from the ones that represent the foreground. The distributions which are likely to be a part of the background are the ones with high weights, and also low variances.

To combine these two factors for each pixel, all the existing distributions are ranked by a criterion $\frac{w_k}{\sigma_k}$. This factor reaches its peak while w_k is large and on the contrary σ_k is small. Therefore higher ranked components are the ones with low variances (intensities do not vary much) and high occurrence probabilities. After the distributions are ranked based on the factor w_k / σ_k, the weights of the corresponding distributions are summed up and the result is checked against a predefined threshold:

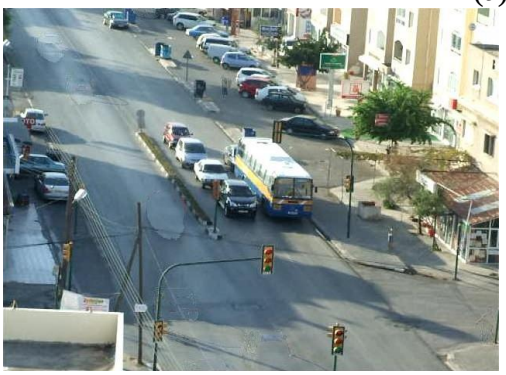
$$b = \arg min_b \left(\sum_{k=1}^{B} w_k > T \right) \tag{2.2.17}$$

Here b indicates the minimum number of distributions which belong to the background among the *K* available distributions at each pixel.

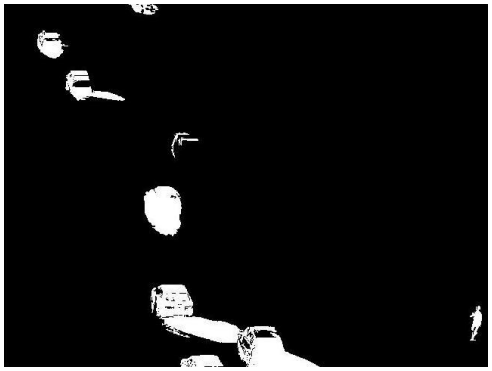
Figure 6 provides an example of the above described steps being applied to a custom video sequence taken at Yeni İzmir Junction of Famagusta in order to estimate the background in the scene. During the simulations the value for *K* and *T* were taken as 5 and 0.85 respectively.



(a) Original Frame



(b) Background estimation



(c) Foreground mask of the left lane

Figure 6: Background estimation using *MoG* Model with K=5, T=0.85

2.3 Progressive Background Estimation Method

This method was first introduced by Y.Chung in [42]. A progressive background image is generated by utilizing the histogram to record the changes in intensity for each pixel of the image, however, unlike its other histogram based background generator counterparts, progressive method does not directly use the input frames to create the histogram. The progressive method constructs the histograms from the preprocessed images also referred to as the partial backgrounds. Each partial background is obtained using two consecutive input frames (for details see section 2.3.1). This method is applicable to both gray scale and color images and

is capable of generating background in rather short period of time and does not need large space for storing the image sequences.

2.3.1 Partial backgrounds

In order to generate the partial backgrounds, the progressive method follows the following steps. First, the current frame *I(t)* at time t of an input video sequence *S(t)* is captured into the system and this image is compared with the previous frame image, *I(t-1)* to generate a current partial background *B(t)*. Each pixel at location *i* at time *t* of the corresponding partial background is called $b_i(t)$ and is computed using equation below [42]:

$$b_i(t) = \begin{cases} bg & |p_i(t) - b_i(t-1)| < \varepsilon \\ non - bg & otherwise \end{cases} \quad (2.3.1)$$

As can be seen from equation (2.3.1), the partial background pixels are divided into two categories. bg stands for pixels related to the background image whose intensity value difference from the previous partial background $b_i(t-1)$ does not exceed a small predefined threshold ε.

If the incoming intensity varies from the partial background more than the selected threshold, the corresponding pixel will be classified as $non - bg$. There are several possible ways to assign value to bg pixels; one is to take the minimum intensity between the new $b_i(t)$ and $b_i(t-1)$, another way is to average these two values and yet another is by simply taking the new value as $b_i(t)$. In this thesis we have chosen the last approach since it needs less computation and is more suitable for real time application.

For $non - bg$ pixels a specific value should be assigned, so that it will be possible to distinguish them since we are not interested in them. To separate them from bg pixels, usually they are assigned 0 or -1. After all the pixels have been

classified and the numbers are assigned to them, the whole partial background at time *t* is created as [42]:

$$B_i(t) = \bigcup_{i \in I(t)} b_i(t) \tag{2.3.2}$$

By creating the partial background images, the moving objects are discarded due to their intensity differences from the background and only the pixels which are more likely to be a part of background will be kept.

However, in some cases slow moving objects or similarity among foreground objects and background scene may cause some parts of moving objects to be misclassified as background related pixels. One solution to such a problem is to add color information in our decision making. Then equation (2.3.1) will turn to [42]:

$$b_i(t) = \begin{cases} bg & \bigcap_c |p_i^c(t) - b_i^c(t)| < \varepsilon^c \\ non - bg & otherwise \end{cases} \tag{2.3.3}$$

where *c* is the different components of the *RGB*. In other words the classification is done separately for each color channel and then their intersection is obtained in order to set aside the pixels that vary in all channels in comparison to previous partial background.

It is worth mentioning that usage of partial backgrounds instead of the original video frames directly has two advantages. Firstly foreground objects cannot interfere with background values since they are removed in partial backgrounds creation. Secondly it helps overcome the problems caused by camera vibrations that may occur due to heavy vehicles passing by or strong wind.

An example for partial background generation is shown in figure 7 below:

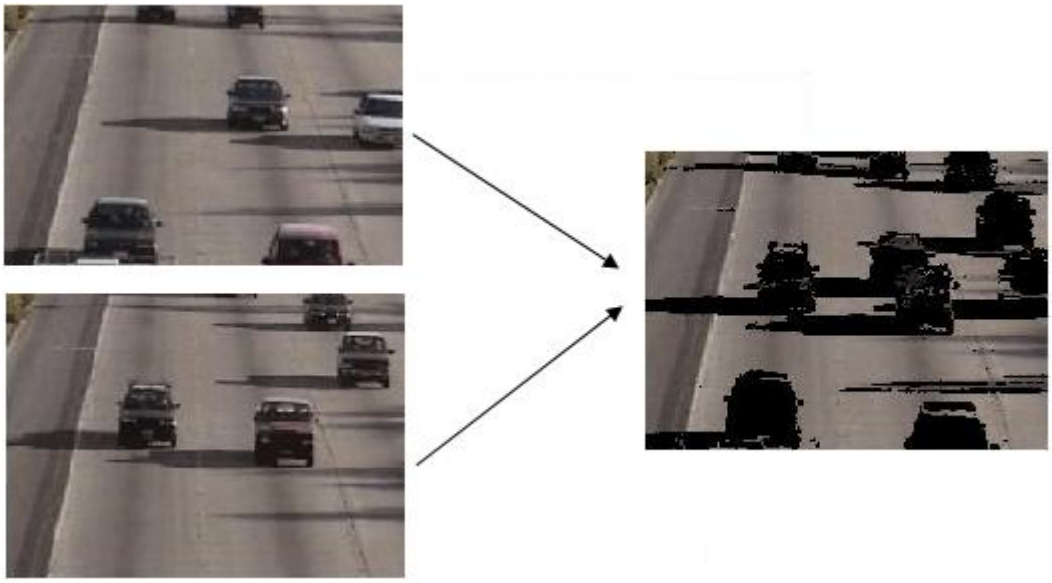
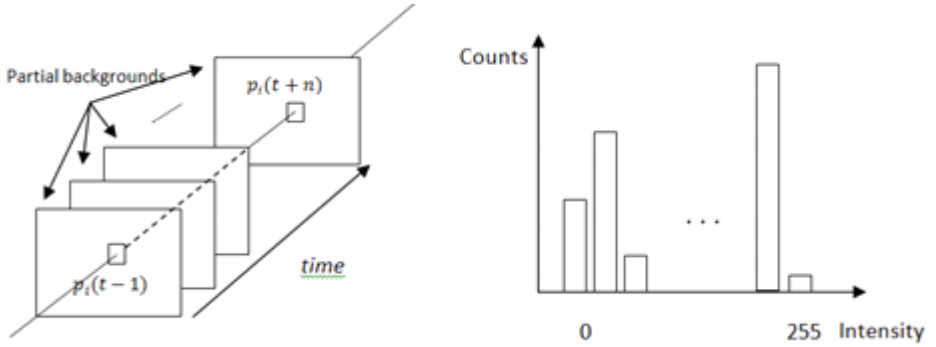


Figure 7: Generation of Partial Backgrounds

2.3.2 Histogram of Pixels

The next step of the progressive background estimation method would be generating a histogram called $h_p(t)$ using the partial backgrounds obtained from the previous step. The index p indicates that there is a histogram for every pixel of the image and t stands for time. For each pixel at time t a certain number of generated partial background depending on the size of our buffer are processed and then the histograms are created per pixel location in time.



(a) Partial background sequence for $p_i(t)$ (b) Histogram for a typical pixel p_i

Figure 8 : The partial backgrounds and histograms

2.3.2.1 Histogram Updating

The updating procedure is done simultaneously with the generation of histograms. For each pixel the incoming intensity from partial background is checked by the algorithm to discover whether the new intensity is within the local neighborhood of the previous background intensities or not. If the mentioned condition is satisfied (the intensity belongs to the neighborhood) then the frequency of that intensity is incremented by a constant factor, unlike conventional histograms this factor is more than one (flexible in general). If the constraint is violated and the newly gained intensity is located outside the boundaries of our neighborhood domain, the recorded frequency for corresponding pixel in the histogram will be decreased by a factor less than mentioned incrementing factor. The preceding discussion can be summarized by the following equations:

$$v = v + A\,\delta(b_i(t), a) - D \tag{2.3.4}$$

where, v is the count (frequency) of the intensity index a, in the histogram. A represents the rising factor while on the contrary D is the descending factor and in general $D < A$. The δ function in equation 2.3.4 can further be defined as:

$$\delta(l, r) = \begin{cases} 1 & |l - r| < \lambda \\ 0 & otherwise \end{cases} \tag{2.3.5}$$

When the newly seen intensity $(b_i(t))$ is a member of local neighborhood of $(|(b_i(t) - a| < \lambda)$ then $\delta(b_i(t), a)$ will become one and frequency of that intensity will be incremented by *A-D* (keep in mind that $D < A$) and on the other hand for the reverse case counts will be decremented by D since in this case delta function would be zero.

Because the updating process is accumulative, to avoid large numbers and to be able to cope with changes in the environment the method defines an upper bound to limit the max value the frequency of each pixel could attain [42].

Hence the histogram values will be raised if they have not already reached to a certain threshold. For a typical pixel location (x, y) the curve of frequency value for certain intensity over time would be as depicted in figure 9.

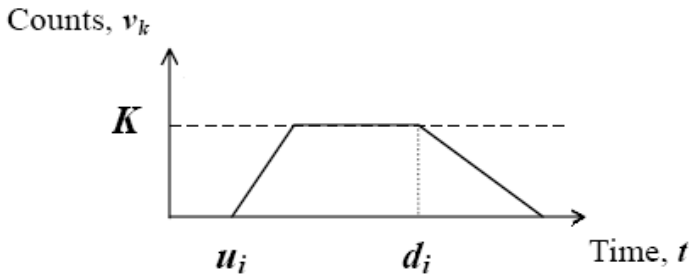


Figure 9: The counts value for a certain intensity index k of a pixel, p_i [42].

When observed for many frames, if the observed samples at the same location belongs to the local neighborhood of the previous background intensities, its frequency will be incremented till an upper limit *K* is reached. After that if the same intensity keeps coming to the view of the considered pixel, the frequency will not grow anymore but stay at this saturated value *K*. The situation would remain the same until at time d_i, for a certain reason; a new intensity starts to come to the pixel view. Therefore the frequency will be decremented by factor D for as long as this newly value is observed.

2.3.3 Histogram Table

After the histograms are generated and updated, the maximum frequency of each histogram along with its corresponding intensity for each pixel in the image are recorded in a table. The histogram table can be utilized as a reference for intensities

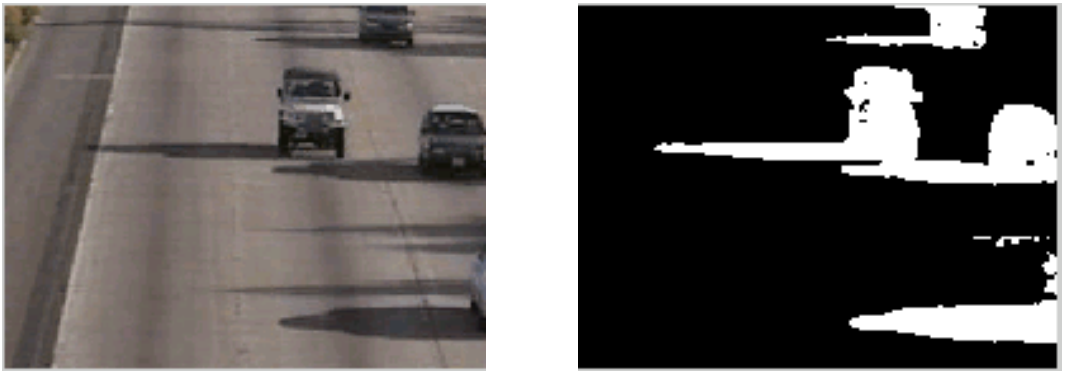
which are responsible for background generation at any time. Whenever the background image is required, the recently updated intensity values in the table are used to generate the desired background.

However, at the beginning of the process some cells of the table may not have a value and hence the background image contains leakages (undesired black dots). This problem occurs because the histograms are built over partial backgrounds which include black parts in the position of moving objects but as time passes, intensities related to the background image come to the pixel view more and more. Therefore this leakage effect will be gradually removed.

As stated in [42], a stable background image would be possible when the counts recorded in the histogram table are approximately 75-80% of a pre-determined upper limit. The higher the frequency values, the better the image quality will become. Figure 10 depicts an example where leakage problem is resolved after 5 frames of the video sequence. Also figure 11 provides a sample frame from Highway-I sequence of *VISOR* and the corresponding foreground mask obtained after background subtraction process.

(a) Existence of leakage (b) Leakages removed after 5 frames

Figure 10: Estimated background using Progressive method.



(a) Original Frame (b) Extracted foreground mask

Figure 11: Extracting foreground objects using progressive method

2.4 Group-Based Histogram

The group-based histogram (*GBH*) algorithm constructs background models by using histogram of intensities come to the view of each pixel on the image. However, unlike the other histogram based methods, group based histogram is forced to follow a Gaussian shaped trend which as it will be demonstrated later, this technique will improve the quality of the background-foreground segmentation [38].

In a video sequence taken by a fixed (static) camera the intensity of the pixels related to background scene is the most frequently recorded intensity at each pixel location (x, y). Hence many histogram approaches have been presented in the literature [39], [40], [41].

Since in histogram approaches at each pixel location the most frequent intensity is proportional (by a factor of *N*) to its occurrence probability, the maximum frequency from the histogram is considered as background model intensity in that location.

The background intensities can therefore be determined by analyzing the intensities of histogram at each pixel. However, sensing variation and noise from image acquisition devices or pixels having complex distributions may result in

erroneous estimates. This may cause a foreground object to have the maximum intensity frequency in the histogram.

Since the maximum count (amplitude) of the histogram is much greater in comparison to frequencies of intensities related to the moving objects , there will not be any effects of slow moving objects or transient stops in the detected foreground.

However, the maximum peak of the conventional histogram of each pixel will not necessarily locate the intensity of background model at that specific pixel. In some cases this maximum may not be unique so further processing may be required to compensate this loss which will affect the real time tracking.

Although the histogram approach is robust to transient stops of moving foreground objects, the estimation is still less accurate than *MoG* model in the case of non-static backgrounds (i.e. swaying grass, shaking leaves, rain, etc). Note that the frequency or probability of conventional histogram is updated by using a single intensity; while the probability of GMM is constructed from a group of intensities. Thus the GMM possesses more admirable capabilities than simple histogram to represent intensity distribution of the background image [38].

In group based histogram, each of the individual intensities is considered along with its neighboring intensity levels and forms an accumulative frequency. The frequency of coming intensity is summed up with its neighboring frequency to create a Gaussian shape histogram.

The accumulation can be done by using an average filter of width $2w+1$ where w stands for half width of the window. The output $n^*_{u,v}(l)$ of the average filter at level l can be expressed as:

$$n^*{}_{u,v}(l) = \sum_{r=-w}^{w} n_{u,v}(l+r) \quad 0 \le l + r \le (L-1) \qquad (2.4.1)$$

Here $n_{u,v}(l+r)$ is the count of the pixel having the intensity $l+r$ at the location (u, v), and L is the total number of possible intensity levels. The maximum probability density ${p^*}_{u,v}$ of a pixel can be computed through a simple division of the occurrence for a pixel by the total frequency of the *GBH* (N^*).

$$ {p^*}_{u,v} = \frac{max_{0 \le l \le L-1}\{{n^*}_{u,v}(l)\}}{N^*} \tag{2.4.2} $$

Since the filter smoothens the histogram curve, if the width of the averaging window is chosen to be less than a preset value, the location of the maximum will be closer to the center of the Gaussian model (which corresponds to background value) than the normal histograms (more details are given in the following section). Therefore the mean intensity of the background model will be:

$$ \mu_{u,v} = \arg max_l\{{n^*}_{u,v}(l)\} \tag{2.4.3} $$

Choice of the window size is a critical task since a smaller window width can save the processing time (due to fewer computations), while a larger window will lead to smoother *GBH* and therefore more accurate estimation of the real value of the pixel related to the background model.

2.4.1 Window Size Selection

To describe the determination of the window width more clearly, an example has been shown here [38]. In this case 13 Gaussian densities have been generated randomly. The mean was chosen to be 205 and standard deviations varying from 3 to15. From the generated data, histograms are created then from each of them the corresponding *GBH* are constructed using different window sizes from 3 to 7.

Table 1: Estimation of error rate of Gaussian mean using histogram and *GBH* [38].

Standard deviation	3	4	5	6	7	8	9	10	11	12	13	14	15
Hist.	−1.5%	−1.5%	**−2.0%**	**−2.4%**	**−2.4%**	**−2.9%**	**−3.4%**	**−2.9%**	**−2.9%**	**−4.4%**	**−2.9%**	**−4.9%**	**−4.4%**
Width *w*	Estimation result of *GBH*												
1	0.5%	1.0%	0.0%	−0.5%	−0.5%	**2.0%**	**−3.4%**	**−2.4%**	**−2.9%**	−1.5%	**−3.4%**	0.5%	1.0%
2	0.5%	0.0%	0.0%	−1.5%	−1.0%	1.0%	**−2.4%**	**−2.0%**	**−2.4%**	**−2.0%**	**−3.9%**	1.0%	1.5%
3	0.0%	0.0%	−0.5%	−1.0%	0.5%	−1.5%	**−2.4%**	**2.0%**	**2.0%**	**−2.4%**	−1.5%	1.5%	**-2.9%**
4	0.5%	0.5%	0.0%	−0.5%	−0.5%	0.5%	−1.5%	−1.0%	−1.5%	**−2.4%**	−1.0%	**2.4%**	**-2.4%**
5	0.5%	0.5%	−0.5%	0.0%	0.0%	−0.5%	−1.0%	−0.5%	−0.5%	**−2.0%**	−1.5%	0.5%	0.5%
6	0.5%	0.0%	−0.5%	−0.5%	−0.5%	−1.0%	−0.5%	0.0%	−0.5%	−1.5%	−1.5%	0.0%	−1.5%
7	0.5%	0.5%	−0.5%	0.5%	0.0%	−0.5%	0.0%	0.0%	0.0%	−0.5%	−1.5%	1.0%	−1.0%

The results prove the superiority of implementing *GBH* method to conventional histograms. Considering the results, it can be concluded that a greater width of average filter will be required for high-accuracy performance as the standard deviation increases. Keeping the error rate of mean estimation within ± 2%, and according to the simulation result, the width can be determined as follows [38]:

$$w = \begin{cases} 3 & 3 \le \sigma_i \le 7 \\ 5 & 8 \le \sigma_i \le 10 \\ 7 & 10 \le \sigma_i \le 15 \end{cases} \tag{2.4.4}$$

where, σ_i represents the standard deviation of the original Gaussian.

2.4.2 Mean Estimation

As mentioned before the mean intensity can be computed by selecting the maximum frequency of the smoothened histogram. When a new intensity *l* is captured, the algorithm does not process all the possible intensities, just the new one and its adjacent intensities which fall in the selected window will be affected.

The steps of the mean estimation procedure include: first recording the current intensity *l* of the pixel. Second step contains incrementing the frequency of occurrence of observed intensity (*l*) and all the neighboring intensities from *l*-*w* to

l+w by unity. Final step is checking whether the new achieved numbers (frequencies) are greater than the previously estimated maximum of counts or not. If the condition is satisfied then replacement of the former mean with the new one is done and then the algorithm will return to the first step.

2.4.3 Variance Estimation

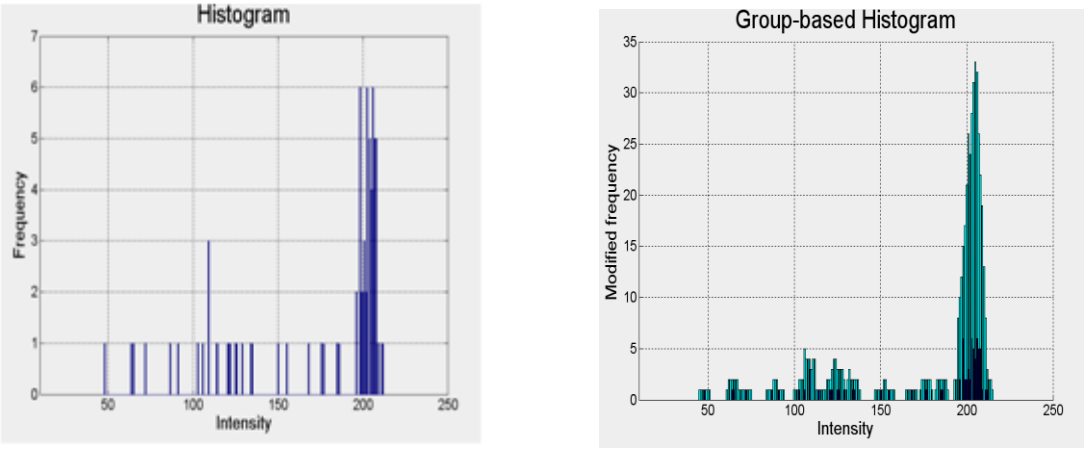
After computing the mean intensity of the Gaussian shaped histogram the variance could be estimated using the following expression:

$$\sigma_{u,v}^2 = \frac{1}{\sum_{x=\mu_{u,v}-3\sigma'}^{x=\mu_{u,v}+3\sigma'} n_{u,v}(x)} \times \sum_{x=\mu_{u,v}-3\sigma'}^{x=\mu_{u,v}+3\sigma'} (x-\mu_{u,v})^2 n_{u,v}(x) \qquad (2.4.5)$$

where, σ' is the maximum standard deviation of the Gaussians.

Figure 12 demonstrates the histogram smoothing after the implementation of average filtering window for a certain traffic sequence. From Figure 2.1(a) one can conclude that it would be possible to model the results with a Gaussian distribution technique over a histogram of a certain pixel in a video sequence. If it is desired to fit a Gaussian distribution model to the data in hand, the center of the Gaussian would be 203.65 with a standard deviation of 3.88 [43].

However, since several peaks with similar frequencies are in the histogram, selecting the mean is not straightforward. By applying the windowing technique proposed in *GBH,* the histogram will be smoothed and this multiple peak problem will be resolved. In figure 12 the estimated mean and standard deviation are 205 and 4 respectively, which indicates acceptable error rates of 0.67% and 3.17% for mean estimation and standard deviation respectively.



(a) Histogram (b) *GBH*

Figure 12: Statistic analysis of pixel intensity [38].

To cope with illumination changes of the environment, the histogram can be re-built every 15 minutes.

2.4.4 Foreground Segmentation

A Gaussian distribution is fitted to smoothed histogram of each pixel in the image. Based on tolerance intervals in statistical issues pixel intensity is considered as a part of foreground while its intensity is outside $\pm 3\sigma$ the mean of the background Gaussian distribution.

If the current pixel intensity is represented by *I(u,v)* where (*u*,*v*) corresponds to the location of pixel on the image, then foreground objects are extracted by using equation 2.4.6:

$$F(u,v) = \begin{cases} 1, & (moving\ objects) \quad if|I(u,v) - \mu(u,v) > 3\sigma(u,v)| \\ 0, & otherwise \end{cases} \tag{2.4.6}$$

where, $\mu(u,v)$, $\sigma(u,v)$ represent mean and standard deviation of the background model at location (*u*,*v*).

Figure 13 provides an example for background estimation by applying the *GBH* approach on a video sequence at a junction. The segmented foreground objects are

vehicles and pedestrians with their corresponding cast shadows. On segmented foreground objects shadow removal algorithms are applied in order to get vehicles without cast shadows.

(a) Original Frame

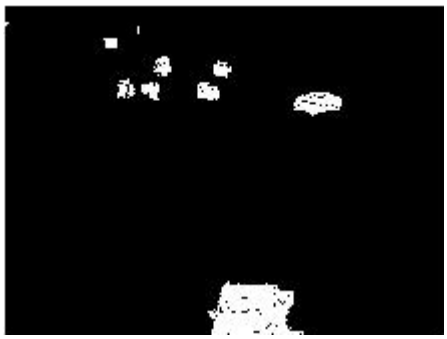
(b) Extracted Foreground

Figure 13: Estimated Background using *GBH* method

Figure 14 gives another example where the video sequence is recorded from one of the streets of Famagusta.

(a) Original frame

(b) Estimated background



(c) Extracted foreground mask

Figure 14: Extracting foreground objects using *GBH* method

CHAPTER 3

SHADOW REMOVAL

As it was mentioned in previous chapters video-surveillance and traffic analysis systems can be heavily improved using vision-based techniques that could detect objects such as vehicles, people, etc., monitoring the trajectory of foreground items in the scene. However, although extracting foreground objects out of frames of a video sequence is an essential task and in fact is the basic step in almost all of the related applications, in some cases the execution of background subtraction won't be enough by itself.

In this chapter one of the algorithms for removing the undesired shadows which are often misclassified by foreground segmentation algorithms is presented. This unwanted phenomenon should be removed as much as possible due to its adverse effects on quality of detected background model.

Incorrect detection of shadows as foreground objects will cause serious problems in many applications. Some of these applications are listed below:

1. Classifying segmented objects

2. Computing the area occupied by an object on the road (lane fullness analysis)

3. Recognition procedures

4. Evaluating the centroid of specified items or motion variation of foreground objects (tracking).

In general there are two types of shadows present in a scene while a video sequence is being recorded. First group is static shadows which do not move with the displacement of moving objects while the other type of shadows is referred to as the cast shadows. The second group is generated due to occlusion of sun light by moving objects. The resultant shadow is the projected area on the scene which moves along side of the moving object, therefore has the same trajectory. An example of incorrectly detected shadows is shown in figure 15 [14]:

Figure 15: Object merging due to shadows

It is apparent from the figure that some of the marked blobs contain more than a single moving object due to the existence of cast shadows. Hence it is impossible to detect the number of objects or perform any classification.

The intensities of pixels related to the cast shadows are significantly different from the corresponding pixels in the background model. Also, since they appear in the recorded frames as frequently as the foreground objects, the background estimation algorithms cannot differentiate them from real moving objects. Therefore

these pixels will be misclassified as foreground objects. This problem is referred to as "*under-segmentation*" in the literature [14].

When a shadow occurs, the intensities of the surface (pixels) which shadow is projected on becomes significantly less, however, the color information of that surface is preserved. This feature is the key factor of the algorithm presented in [14].

Human visual system is able to distinguish the colors of objects located in shaded areas. Therefore to remove the cast shadows the Hue-Saturation-Value (*HSV*) color space has been used. The *HSV* color space corresponds closely to the human perception of color, and it has been proven to be more accurate in distinguishing shadows in comparison to the *RGB* space [45].

In HSV color space Hue varies between zero and one representing the color (from red through yellow, green, cyan, blue, magenta, and back to red) Saturation indicates the purity of the color. In other words S shows how much that color is diluted by white. When $S = 1$ the color is 100% pure and no white is mixed with it. The reverse is also true while $S = 0$. Finally the V component is a measure for brightness (intensity). H and S are used to describe chrominance information while V represents luminance. The following figure shows the same discussion graphically:

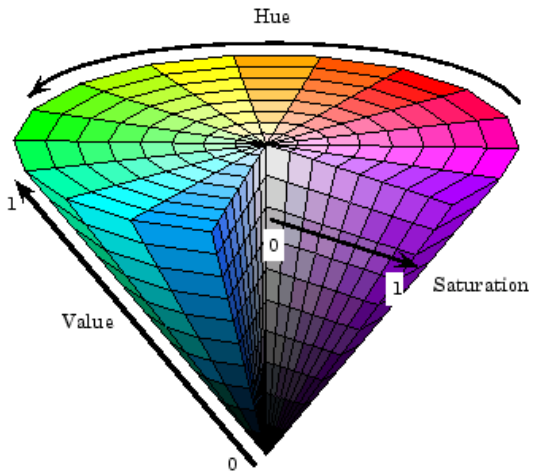


Figure 16: HSV color space

3.1 Shadow Removal Algorithm

The luminance of a point at location (x, y) which belongs to cast shadow at instant *k* can be described as [45]:

$$S_k(x,y) = E_k(x,y)\,\rho_k(x,y) \tag{3.1.1}$$

where $\rho_k(x,y)$ represents reflection of the surface, $E_k(x,y)$ indicates irradiance and can be formulated as:

$$E_k(x,y) = \begin{cases} C_A + C_P \cos(N(x,y), L) & \text{illuminated} \\ C_A & \text{shadowed} \end{cases} \tag{3.1.2}$$

where C_A and C_P are the intensity of the ambient light and of the light source, respectively, L is the direction of the light source and N(x,y) the object surface normal [14].

If a static background point is covered by a shadow, then we have:

$$R_k(x,y) = \frac{C_A}{C_A + C_P \cos(N(x,y), L)} \tag{3.1.3}$$

Since the angle between $N(x,y)$ and L varies from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ for shadow points the denominator is greater than numerator. Hence $R_k(x,y)$ would be less than one. Taking advantage of equation (3.1.3) and considering the key feature mentioned above, the following constraint can be used to classify the shadow points [45]:

$$SP_k(x,y) = \begin{cases} 1 & if\ \alpha \le \dfrac{I_k^V(x,y)}{B_k^V(x,y)} \le \beta \\ & \wedge \left(I_k^S(x,y) - B_k^S(x,y)\right) \le \tau_S \\ & \wedge |I_k^H(x,y) - B_k^H(x,y)| \le \tau_H \\ 0 & otherwise \end{cases} \tag{3.1.4}$$

The first condition considers the variation of the luminance (the *V*-component). Some background points which are affected by noise may have not exactly the same value. Hence when the luminance ratio is computed the result will

be less than one. To compensate this loss, an upper bound β(less than one) is used to avoid the incorrect identification of the regular pixels as shaded ones. The lower bound α is defined to take strength of the light source into account, (i.e. stronger and higher the sun the lower will be that ratio, and lower value of α must be chosen).

Since H and S components are responsible for chrominance information, the variation of these values should not exceed predefined thresholds (τ_H , τ_S). However, the choice of the parameters τ_H and τ_S is less straightforward and is done empirically with the assumption that the chrominance of shadowed and non-shadowed points even if could vary, does not vary too much [14].

Figure 17 shows the same scene in figure 15, however, this time the shadows are correctly detected and removed. One can easily notice how shadow suppression allows the correct identification of all the objects in the scene.

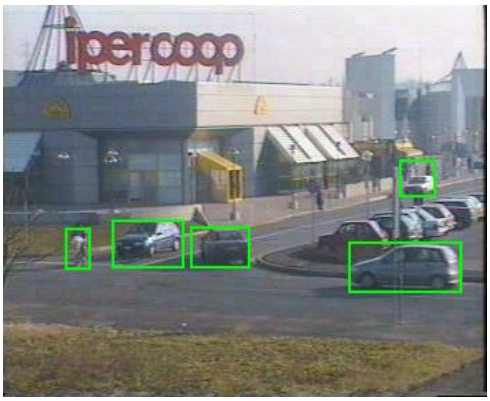


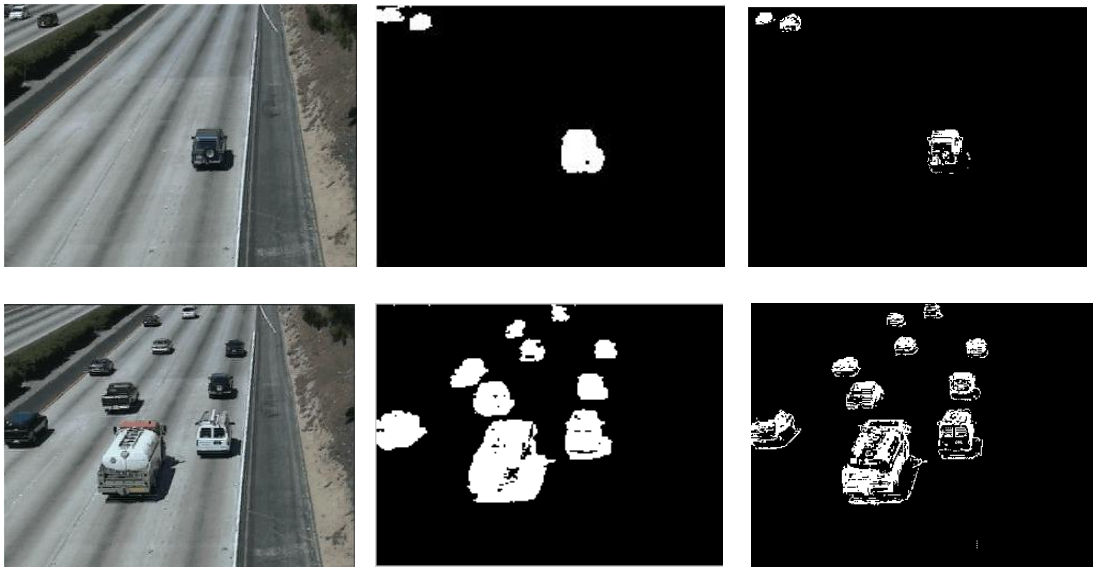
Figure 17: The correct identification of objects after shadow removal [14].

3.2 Simulation Results

We have also applied the *HSV* color space based shadow removal technique to some custom recorded and standard video sequences. In the figures below same sample frames are given to show that the algorithm would perform fairly well on all the different test sequences used.

(a) Original frame (b) Extracted foreground (c) After shadow removal

Figure 18: Custom video recorded at Yeni-İzmir Junction



(a) Original frame (b) Foreground mask (c) After shadow removal

Figure 19: Video sequence Highway II

CHAPTER 4

SIMULATION RESULTS AND PERFORMANCE ANALYSIS

Up to this point several algorithms have been selected from literature and are implemented. It has been tried here to mention the algorithms which are fundamentally different from each other but most of the existing methods suffer from a common problem. As they strive to deal with multi-modal scenes they become more and more sensitive to slow moving objects and transient stops which are often the case in intersections due to the traffic signals. Therefore these algorithms become less likely to be implemented in vision-based traffic monitoring systems (VTMS).

Vision based judgment is one of the common measures for comparison purposes since most of the failures of the algorithms lead to visible defects in the final detected background model. However, for a fairer comparison here a quantitative scale is used additionally.

4.1 Ground Truth

This concept is used as base for the quantitative comparisons. Ground truths are special kind of video sequences which contain only the desired moving objects of the scene (ideal foreground detection).

Here two video sequences are used along with their corresponding ground truths. One of them include indoor scenes and the other one is recorded from outdoor environment. These videos are recorded just from a scene without any foreground objects and then animated moving objects are superimposed manually on the recorded background scenes. Therefore the exact location of the pixels related to foreground items are known, in other words the ground-truths of these sequences are available.

Another advantage of using this kind of sequences is that since the super imposed objects do not contain shadows, we can only focus on the performance of background detection instead of dealing with shadow removal algorithms which at this point, we are not interested in them.

For more clearance a typical frame and its corresponding ground-truth are shown in figure (4.1):

(a) Original frame (b) Corresponding Ground-truth

Figure 20 : Typical frame of synthetic video-2 [47].

The created sequences are fed to the applied background estimation methods and the extracted foreground of each is recorded frame by frame.

The next step would be taking advantage of practical scales in order to help us compare our achieved results with the ground-truths. One of the most well known measures is called “Recall-Precision” scale.

4.2 Classification of Pixels

Prior to the details of recall and precision definitions, certain concepts should be explained. These concepts include classifying pixels in 4 different groups:

1. True Positive (TP): which represents the number of foreground pixels correctly detected by the algorithm.

2. False Positive (FP): is responsible for the number of pixels which are incorrectly classified as foreground objects.

3. True Negative (TN): indicating the number of background pixels which are correctly detected as background scene by the algorithm.

4. False Negative (FN): stands for the number of pixels corresponding to foreground objects which are misclassified as part of background image (also referred as misses) [44].

There are several other methods for quantifying a classifier’s performance (background estimators) [44]:

1. Percentage correct classification

2. Jaccard coefficient

3. Yule coefficient

However, in this thesis the pre mentioned recall and precision measures are applied.

4.3 Recall

Recall is measure of completeness and is defined as number of true positives divided by the total number of elements that actually belong to the foreground objects. (i.e some of both true positives and false negatives).

$$Recall = \frac{TP}{TP + FN} \tag{4.3.1}$$

In other words it can be rewritten as:

$$Recall = \frac{number\ of\ correctly\ identifiedforeground\ pixels}{number\ of\ foreground\ pixels\ in\ ground\ truth} \tag{4.3.2}$$

4.4 Precision

Precision can be considered as a measure of exactness or fidelity and is evaluated through dividing the number of items (foreground objects) correctly detected by the total number of pixels classified as foreground by algorithm.

In fact we are evaluating if the algorithm shows that a certain pixel is foreground and how reliable that statement would be.

$$Precision = \frac{TP}{TP + FP} \tag{4.4.1}$$

$$Precision = \frac{number\ ofcorrectly\ identified\ foreground\ pixels}{number\ of\ foreground\ pixels\ detected\ by\ algorithm} \tag{4.4.2}$$

4.5 Data Analysis

We have applied our implemented methods to two mentioned videos in section (4.1). The outdoor sequence includes shaking leaves along with passing of various objects from a small cat up to vehicle.

It should be noted here that to keep the condition of the experiments almost the same (real time performance) except the approximated median filtering method (which is fast enough even while performing on colorful images) other algorithms have been executed in gray-scale mode.

The results in both of the measures will increase while the color information is added. The results are summarized in the table shown below:

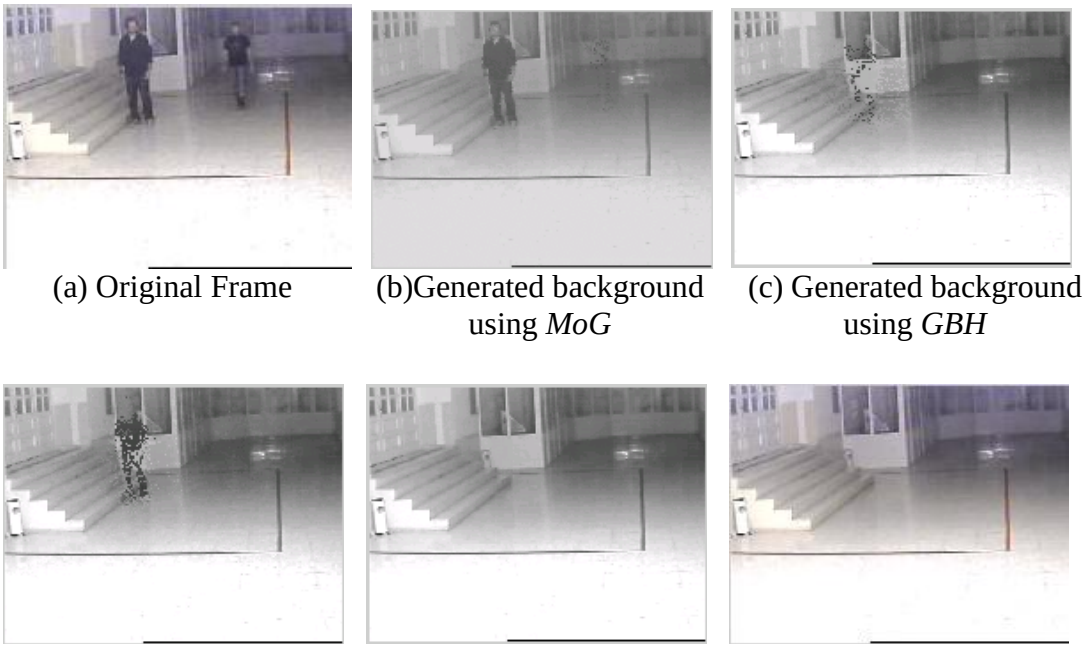
Table 2: Average recall and precision results for five background estimation algorithms.

VIDEO 2				VIDEO 7		
Estimation Method	**Recall**	**Precision**		**Estimation Method**	**Recall**	**Precision**
Group-based histogram (*GBH*)	99.25	93.19		Group-based histogram (*GBH*)	86.18	74.42
Progressive estimation (*PM*)	90.58	99.21		Progressive estimation (*PM*)	72.30	60.92
Mixture of Gaussians (*MoG*)	81.84	91.22		Mixture of Gaussians (*MoG*)	85.38	77.96
Approximated Median Filtering(*AMF*)	92.26	91.5		Approximated Median Filtering(*AMF*)	82.34	58.19
Temporal Median Filtering(*TMF*)	84.01	99.99		Temporal Median Filtering(*TMF*)	77.88	49.65

The simulation results prove that dealing with outdoor environments (video 7) is a more challenging task. In the case of indoor scenes with real static background (without any undesired movements) most of the algorithms have acceptable performance (over 85% in both scales).

As it was mentioned before, vehicles often stop transiently due to traffic signals. Hence, in this part we have compared the performance of algorithms in the

case of transient stops as well. The video sequence, which is used here, contains an indoor scene showing students walk through the corridor, stop for a while, then continue walking again. The simulation results suggest that the *MoG* Model is not robust against these kind of stops in comparison with other tested methods. The figures below demonstrate this problem more clearly:



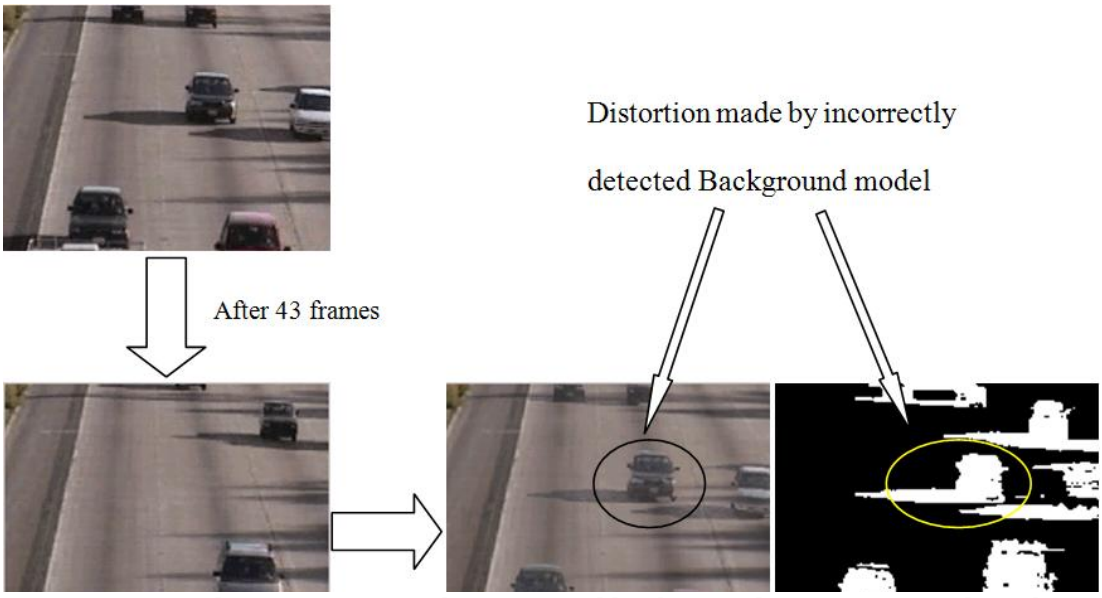
(a) Original Frame (b)Generated background using *MoG* (c) Generated background using *GBH*

(d) Generated background using *PM* (e) Generated background using *TMF* (f) Generated background using *AMF*

Figure 21: Comparing the adverse effect of transient stops

To test which one of the algorithms generate the background model faster, we have applied them to a video sequence, which does not start with any empty frames.

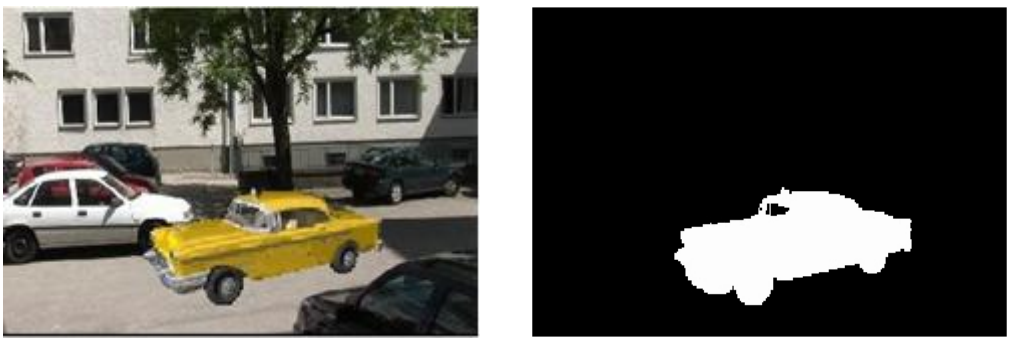
Although *AMF*, processes each frame faster than other tested algorithms(less than half a second per each frame), it takes longer time to create background model. It is obvious from the following figure that a ghost (faded) effect of the present objects in the initial frame is still visible after passing of 44 frames.



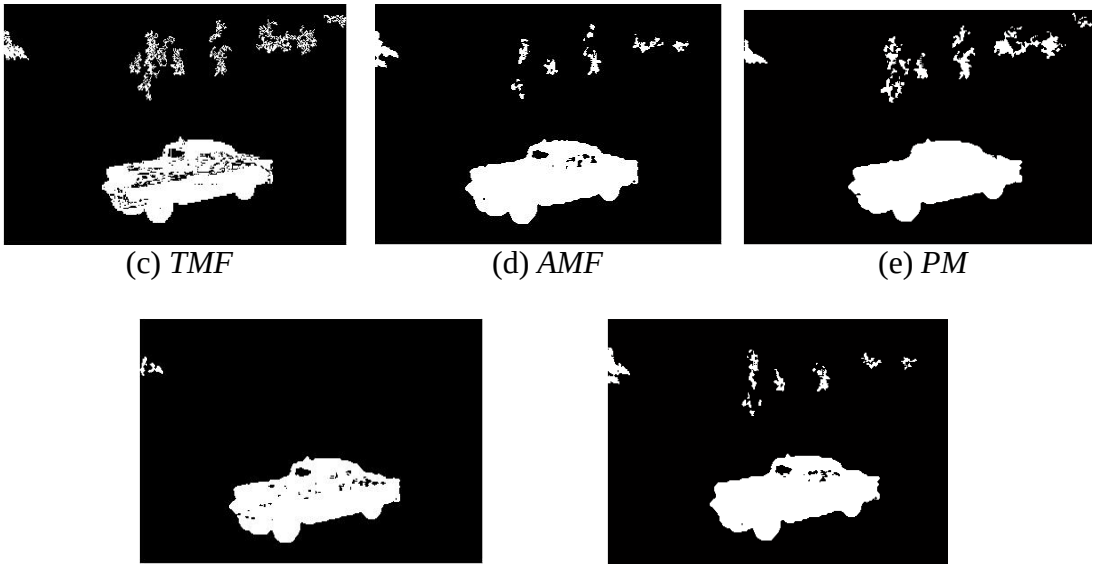
(a) Frames1and 44 (b) Estimated background (c) Foreground mask

Figure 22: Adverse effect of late background generation, using *AMF*

Although the percentages from the table contain comparison information about the ability of algorithms in handling multi-modal background scenes for the sake of more clarity, the 380th frame is selected from synthetic video 7. This frame includes bi-modal background scene (notice the shaking leaves of the trees in the background). The simulation results are demonstrated along with the corresponding foreground mask from the ground-truth sequence in the following figures:



(a) Frame 380 from video 7 (b) Ground-truth of frame 380



(c) *TMF* (d) *AMF* (e) *PM*

(f) *MoG* (g) *GBH*

Figure 23: Visual comparison between algorithms in handling multi-modal scenes.

As it is obvious from the preceding figure *MoG* algorithm is able to suppress the multi-modal background more effectively. The speed of algorithm in processing each frame is an important issue for real-time applications; hence here these algorithms are applied to a sample video sequence and average processing times are computed for each of the methods.

Table 3: Performance comparison of algorithm with respect to time

Methods	Average Processing Time(frame/sec)
Group-based histogram (*GBH*)	4.0214
Progressive estimation (*PM*)	2.7422
Mixture of Gaussians (*MoG*)	1.4152
Approximated Median Filtering (*AMF*)	0.0490
Temporal Median Filtering (*TMF*)	1.8570

The algorithms are also compared from initialization point of view to see which algorithm creates the background scene faster. To achieve this goal, a video sequence recorded from a rather crowded highway which does not contain any empty frames (foreground objects are always present in the scene) is used. The number of frames required to pass in order to enhance an acceptable foreground mask is recorded in the table below.

Table 4: Required number of frames to generate acceptable foreground masks

Methods	Number of required frames
Group-based histogram (*GBH*)	6
Progressive estimation (*PM*)	10
Mixture of Gaussians (*MoG*)	9
Approximated Median Filtering (*AMF*)	44
Temporal Median Filtering (*TMF*)	12

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The implementation results indicate that critical tradeoffs are always present between the accuracy of estimated background model and the real time performance of the method. The choice of algorithm for background modeling should be made according to the desired application. For instance if it is desired to monitor an indoor scene environment, one of the most suitable choices would be the Approximated Median Filtering, however, the same algorithm (as shown in chapter 2) is not a proper choice when it comes to outdoor scene surveillance. Due to the fact that it cannot deal with multi-modal background scenes or cope with weather condition changes. Mixture of Gaussian Model is one the most reliable background estimation methods in the literature which is capable of handling multi-modal background scenes. The application results from chapter 2 proved the same utter. However, these kind of algorithms fail in the case of illumination changes and transient stops of moving objects (in locations such as intersections).

Histogram based approaches seemed to be robust to transient stops but they are still too sensitive to illumination changes and required larger storage spaces.

5.2 Future Work

Most of the tested implemented algorithms suffer from variation of weather conditions. It is intended to develop a new algorithm which combines the updating procedure in progressive method with windowing technique in *GBH* method. Also it is needed to add an illumination tracking process to the algorithm in order to make the foreground segmentation part adaptive to the illumination changes.

REFERENCES

[1] A. Mittal, N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," *in Proc. of the Int. Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 302-309, 2004.

[2] S.C. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Video Communications and Image Processing, SPIE Electronic Imaging, UCRL Conf. San Jose,* vol.200706, Jan 2004.

[3] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence* ,vol.22, pp. 831-843, Aug 2000.

[4] I. Haritaoglu, D. Harwood, and L. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.22, pp. 809-830, Aug 2000.

[5] AR. Francois, GG. Medioni, "Adaptive color background modeling for real-time segmentation of video streams," *in Proceedings of the Wireless Sensor Networks Recent Patents on Computer Science,* USA; vol.1, pp.227-232, 2008.

[6] S. Huwer, H. Niemann , "Adaptive Change Detection for Real-Time Surveillance Applications," *Third IEEE Int. Workshop on Visual Surveillance*; pp. 37-45, 2000.

[7] RJ. Radke, S. Andra, O. Al-Kofahi, B. Roysam, "Image Change Detection Algorithms: A systematic survey," *image processing, IEEE Transaction*, vol.14, pp. 294-307, March 2005.

[8] C. Stauffer, WEL. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition CVPR,*; vol.2, pp. 246-252, 1999.

[9] R. Cutler and L. Davis, "View-based detection," *in Proceedings Fourteenth International Conference on Pattern Recognition*, vol.1, pp. 495-500, (Brisbane, Australia), Aug 1998.

[10] R. Cucchiara, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.25, pp. 1337-1342, Oct 2003.

[11] B. Gloyer, H. Aghajan, K.-Y. Siu, and T. Kailath, "Video-based freeway monitoring system using recursive vehicle tracking," *in Proceedings of SPIE*, vol. 2421, pp. 173-180, Feb 1995.

[12] Q. Zhou and J. Aggarwal, "Tracking and classifying moving objects from videos," *in Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.

[13] I. Haritaoglu, D. Harwood, LS. Davis, "W4: Who? when? where? what? a real time system for detecting and tracking people," *In Third Face and Gesture Recognition Conf*; pp. 222-227, Apr 1998.

[14] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Statistical and knowledge-based moving object detection in traffic scene," *in Processings of IEEE Int'l Conference on Intelligent Transportation Systems*, pp. 27-32, 2000.

[15] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, S. Russell, "Toward robust automatic traffic scene analysis in realtime," *in Proc. Int. Conf. Pattern Recognition,*; pp. 126-131,1994.

[16] K-P Karmann, A. Brandt, "Moving object Recognition using and adaptive background memory," in Time-Varying Image Processing and Moving Object Recognition. V. Cappellini, Ed. vol.2, pp. 289-307, Elsevier Science Publishers B.V. 1990.

[17] K-P. Karmann, AV. Brandt, R. Gerl, "Moving object segmentation based on adaptive reference images". *in Signal Processing V: Theories and Application.* Amsterdam. The Netherlands: Elsevier, 1990.

[18] A. Monnet, A. Mittal, N. Paragios, V. Ramesh, "Background modeling and subtraction of dynamic scenes," *in Proc. Int. Conf. Computer Vision*, Nice, France,; vol.2, pp.1305-1312, 2003.

[19] IT. Jolliffe "Principal Component Analysis". Springler-Verlag, 1986.

[20] F. Torre, MJ. Black, "Robust principal component analysis for Compvision". *In ICCV*, Vancouver, Canada; vol.1, pp. 362-369 July 2001.

[21] DW.Scott "Mulivariate Density Estimation," New York: Wiley- Inter science, 1992.

[22] RO. Duda, DG. Stork, PE. Hart, "Pattern Classification," New York: Wiley, 2000.

[23] C. Lambert, S. Harrington, C. Harvey, A. Glodjo, "Efficient on-line nonparametric kernel density estimation," Aorithmica; vol.25, pp. 37-57, 1999.

[24] A. Elgammal, D. Harwood, L. Davis, "Non-parametric Model for Background subtraction," *in Proceedings of the 6th European Conf. on Compter Vision-Part II*; vol.2, pp. 751-767, 2000.

[25] N .McFarlane, C.Schofield , "Segmentation and tracking of piglets in images," *Machine Vision Application*; vol. 83, pp. 187-193, 1995.

[26] C. Wren, A. Azabayejani, T. Darrel, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19, pp. 780-785, July 1997.

[27] S. Jabri, Z. Duric, H. Wechsler, A. Rosenfeld, "Detection and Location of

People in Video Images Using Adaptive Fusion of Color and Edge Information," *ICPR, 15th Int. Conf. on Pattern Recognition*; vol.4, pp. 4627-4630, 2000.

[28] J. Heikkila and O. Silven, "A real-time system or monitoring of cyclists and pedestrians," in Second IEEE Workshop on Visual Surveillance, pp. 246-252, (Fort Collins, Colorado), Jun 1999.

[29] G. Halevy and D. Weinshall, "Motion of disturbances: detection and tracking of multi-body non-rigid motion," Maching Vision and Applications, vol.11, pp. 122-137, 1999.

[30] T. Boult et al., "Frame-rate omni-directional surveillance and tracking of camuaged and occluded targets," in Proceedings Second IEEE Workshop on Visual Surveillance, pp. 48-55, (Fort Collins, CO), June 1999.

[31] K.P. Karmann and A. Brandt, "Moving object recognition using and adaptive background memory," *in Time-Varying Image Processing and Moving Object Recognition*, V. Cappellini, ed.,vol. 2, pp. 289-307, Elsevier Science Publishers B.V., 1990.

[32] J. Rittscher, J. Kato, S. Joga, A. Blake, "A probabilistic background model for tracking," *In Proc. 6th Eur. Conf. Computer Vision*, vol.2, pp. 336-350, 2000.

[33] C. Montacie, M-J. Caraty, C. Barras, "Mixture Splitting Technique and Temporal Control in a HMM-Based Recognition System," *in Proc. Intl. Conf. on Spoken Language Processing (ICSLP)*; vol.2, pp. 977-980, 1996.

[34] A. Dempster, N. Laird , and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B;* vol.39, pp. l–38, 1977.

[35] J. Bilmes, “A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models,” Tech. Rep. ICSI-TR-97-021, University of California Berkeley, 1998.

[36] P. W. Power , J. A. Schoonees, “Understanding background mixture models for foreground segmentation,” *In Proceedings Image and Vision Computing*, New Zealand(Auckland), pp. 267-271, Nov 2002.

[37] P.Kaewtrakulpong and R.Bowden, “An improved adaptive background mixture model for real time tracking with shadow dectection,” *in Proc. 2nd European Workshop on Advanced Video-Based Surveillance Systems*, 2001.

[38] K.Song and J.Tai, “Real-Time Background Estimation of Traffic Imagery Using Group-Based Histogram,” *Journal of Information Science and Engineering*, vol. 24, pp. 411-423, 2008.

[39] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, “Traffic monitoring and accident detection at intersections,” *IEEE Transactions on Intelligent transportation Systems*, Vol. 1, pp. 108-118, 2000.

[40] R. A. Johnson and G. K. Bhatacharyya, *Statistics: Principles and Methods*, John Wiley & Sons, New York, 2001.

[41] P. Kumar, S. Ranganath, W. Huang, and K. Sengupta, "Framework for real-time behavior interpretation from traffic video," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6, pp. 43-53, 2005.

[42] Y.Chung, J.Wang and S.Chen, "Progressive Background Images Generation," *In Proc. 15th IPPR. Conf. Computer Vision*, 2002.

[43] J. T. McClave, T. Sincich, and W. Mendenhall, *Statistics*, 8th ed., Prentice Hall, New Jersey, 1999.

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

[45] N. Herodotou, K.N. Plataniotis, and .N.Venetsanopoulos, "A color segmentation scheme for object-based video coding," *in Proceedings of the IEEE Symposium on Advances in Digital Filtering and Signal Processing*, pp. 25–29, 1998.

APPENDICES

Appendix A: Novel Traffic Lights Signaling Technique Based on Lane Occupancy Rates

Nima Seifnaraghi, Saameh G. Ebrahimi and Erhan A. Ince[*]

Electrical and Electronic Eng. Dept.,
Eastern Mediterranean University
Famagusta, North Cyprus, via Mersin 10 Turkey.
*e-mail: erhan.ince@emu.edu.tr

Abstract—**In a conventional traffic lights controller, the lights either change at constant cycle times or at times proportional to the length of each leg of the intersection. Such approaches clearly are not perfect for optimizing traffic flow. Waiting times proportional to lane length may work well for a single-lane road but when roads with multiple lanes are considered the solution would not be optimal. The authors believe that an adaptive signaling based on fullness of each leg of the intersection would be a better approach. This paper presents the segmentation of foreground objects from frames of the surveillance video using an adaptive K-Gaussian mixture model and describes an approach for determining the lane occupancy rates for the north leg of the intersections. To give an accurate fullness measure the cast shadows that might be present in the segmented foregrounds are removed using a combined probability map called the shadow confidence score. Simulation results are provided for two standard and one custom recorded sequence.**

Keywords-component; gaussian mixture model,cast shadow removal, convex hull fitting, convex hull mask, lane occupancy rates

I. INTRODUCTION

In visual surveillance applications, a common approach for differentiating moving objects from the static part of the video frames is detection by background subtraction. Background subtraction involves calculating a reference image, subtracting this reference from each new frame and then thresholding the result. The key issue in background subtraction is how to model the background and update the model in order to adapt to the changes of the background. The changes include variations in the intensity, inclination of the incident light(s) and physical changes such as the small motions of background objects (swaying tree branches, moving clouds, rain or snow). Over the years various statistical models have been proposed. For example in [1], Ridder modelled each pixel with a Kalman filter and presented a more illumination robust system. In [2] and [3] it was assumed that the series of intensity values of a pixel can be modelled by a single unimodal distribution. However in time it was shown that a single-mode model will not handle multiple backgrounds well. For modelling of complex and non-stationary backgrounds [4, 5 and 6] suggest the use of generalized mixture of Gaussians (MoG) model. During modelling the pixel distributions can be initialized randomly, using the K-means approximation and the expectation-maximisation (EM) algorithm [7]. Random initialization is known to result in slow learning and at times would even result in instability. Initialization with the K-means or the EM algorithm would give significantly better results. The EM algorithm is computationally intensive and takes the initialization process off-line. In this study since we would be dealing with real time video from a busy plaza (many moving humans and vehicles) the on-line K-means initialization was adopted. In [8] Kim reported that for backgrounds with fast variations, the MoG with multiple distributions will not be accurate enough and suggested that for such situations non-parametric techniques [9] are used. However in this study since the observed location is an intersection no such fast variations will be encountered (generally people slow down as they approach the traffic lights, and when the light turns green they start from stationary position and gradually speed up).

Various results presented in the literature point out that neither motion segmentation nor change-detection algorithms can distinguish between moving objects and moving shadows. In order to guarantee accurate segmentation shadow suppression must be administered. Shadows occur when objects partially or fully block direct light coming from a source. They are composed of self-shadow and cast shadow. The former is due to the fact that a part of the object is not illuminated directly by the light source and the latter is the region projected by the object in the direction of light. Many shadow removal algorithms exploiting the variations of the brightness and chrominance distortion metrics (BD, SBD, HBD, α_i, CD_i), the YUV or HSV colour spaces, and texture difference between foreground and background exist. In this paper for the detection of cast shadows the total shadow confidence score proposed by Fung in [11] will be used.

II. ADAPTIVE K-GAUSSIAN MIXTURE MODEL

As described in [4] and [5] each 3-tuple pixel vector in the current scene is modeled by a separate mixture of K Gaussians:

$$P(X_{i,t}) = \sum_{i=1}^{K} w_{i,t} \cdot \eta(X_{i,t}, \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

In the adaptive *K-MoG* model $X_{i,t}$ is the current pixel value vector which consists of *R*ed, *G*reen and *B*lue components, $w_{i,t}$ is an estimate of the weight of the i^{th} Gaussian in the mixture at time t, $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean value and the covariance matrix of the i^{th} Gaussian in the mixture. $P(X_{i,t})$ denotes the probability of observing the current pixel value vector given the mixture of K Gaussian distributions and $\eta(X_{i,t}, \mu_{i,t}, \Sigma_{i,t})$ is a Gaussian probability density function.

$$X_{i,t} = (x_{i,t}^R, x_{i,t}^G, x_{i,t}^B)$$
$$\mu_{i,t} = (\mu_{i,t}^R, \mu_{i,t}^G, \mu_{i,t}^B)$$
$$\Sigma_{i,t} = \begin{pmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_G^2 & 0 \\ 0 & 0 & \sigma_B^2 \end{pmatrix} \quad (2)$$
$$\eta(X_{i,t}, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(X_{i,t}-\mu_{i,t})^T \Sigma^{-1}(X_{i,t}-\mu_{i,t})}$$

Background/foreground separation consists of two independent steps: 1) estimating the parameters of K distributions; and 2) evaluating the likelihood of each distribution to represent the background.

A. Parameter Updating

Since at the start of modelling all the Gaussians have an equal probability for representing the background the weights $w_{i,t}$, i∈ (1...K), are all set to the value $1/K$ and the variances are set randomly to high values. Then every new pixel value vector $X_{i,t}$ is checked against the existing K Gaussian distributions until a match is found (a match is defined as a pixel value vector whose Euclidean distance is within 1.5 standard deviations of a distribution). The parameters of the matched component are then updated using the recursive equations below:

$$\mu_{i,t} = (1-\rho)\cdot \mu_{i,t-1} + \rho X_{i,t}$$
$$\Sigma_{i,t} = (1-\rho)\cdot \Sigma_{i,t-1} + \rho \cdot diag\{(X_{i,t}-\mu_{i,t})^T (X_{i,t}-\mu_{i,t})\} \quad (3)$$
$$\rho = \alpha \cdot \eta(X_{i,t} | \mu_{i,t-1}, \Sigma_{i,t-1})$$

In equation (3) α represents the user-defined learning rate and has a value in the range $0 \le \alpha \le 1$. ρ on the other hand is a learning rate for the parameters.

For the case when there are no matches the Gaussian distribution with the least weight is replaced by a new component with a mean equal to the current pixel vector. The variance for this new distribution is set high and the weight is set to a low prior value. Finally, the weight of all the K Gaussians ($G_i, i \in 1...K$) at time t are updated and normalized using equation (4)

$$w_{i,t} = (1-\alpha)\cdot w_{i,t-1} + \alpha \cdot M_{i,t}$$
$$w_{i,t} = \frac{w_{i,t}}{\sum_{m=1}^{K} w_{m,t}} \quad (4)$$

When there is a match $M_{i,t}$ is assumes as 1 and 0 otherwise.

B. Background Estimation

Once the parameters for all the Gaussian distributions are updated the ones that are most likely produced by background processes are determined. First, the K Gaussians are sorted in descending order by the value of $w_{i,t}/\Sigma_{i,t}$ and then the first B distributions are chosen to be in the background model using the value of B as given by (5)

$$B = \arg\min_b \left\{ \sum_{k=1}^{b} w_k > T \right\} \quad (5)$$

Here, T assumes a value between 0.5 and 1. Generally the segmented foreground would contain some noise. It is possible to get rid of this noise by making use of morphological operations and connected component analysis [13].

III. CAST SHADOW DETECTION AND REMOVAL

Ideally, background subtraction should detect real moving objects with high accuracy. However in practice the detection of cast shadows as foreground objects is very common. The cast shadows that are projected on the road surface can change in size based on the elevation of the illuminating light source. When cast shadows stretch, two or more independent objects can appear to be connected together. Unavoidably, the accuracy of segmentation and estimation of how full the intersection legs are would be affected negatively. In order to alleviate these problems the paper adopts a combined probability map also known as the shadow confidence score (*SCS*) [10,11]. The characteristics of the cast shadow in the luminance, chrominance and gradient density domain dictates that:

- Luminance values of the cast shadow pixels are lower than those of the corresponding pixels in the background image.
- The chrominance values of the cast shadow pixels are identical or only slightly different from those of the corresponding pixels in the background
- The difference in gradient density values of the cast shadow pixels and the corresponding background pixels is relatively low. The difference in gradient density values between the vehicle pixels and the corresponding background pixels is relatively high

Based on these observations, the luminance, the chrominance and the gradient density scores for each blob in the foreground mask can be computed using the equation defined in [11] and a total shadow confidence score (*SCS*) can be obtained by combining these three scores.

Figure 2 depicts the effect of *SCS* based shadow removal for frame #1590 of the custom video. From the *SCS* depicted in part (d) it can be seen that the pixels representing the foreground objects in comparison to the cast shadows have lower intensities. To distinguish between the two a threshold may be applied and as depicted in Figure 2(e) sometimes parts of the objects can be misclassified as shadows (incorrect decisions led to undesired erosion on the foreground mask). To fix this problem a convex hull can be fitted to the remaining shadow free foreground mask and then inside the hull is filled to create a new more complete foreground mask. Finally the convex hull based new mask can be used to segment the foreground objects from the input frame.

A. *Convex Hull Fitting*

Generating a polygon that completely and closely surrounds a given set of points in 2D is called convex hull fitting. In the literature there are many algorithms for convex hull generation. Some well-known ones include incremental, gift wrapping, divide and conquer and quick hull algorithms. In this paper we describe the incremental algorithm. The processing starts with a single point and then using two more points a triangle is created. Next a new point is selected. If the new point is inside the hull there is nothing to do. Otherwise one must delete all the edges that the new point can see and add two new edges to connect the new point to the remainder of the old hull. This process is then repeated for all the remaining new points.

B. *Convex Hull Mask*

As mentioned earlier while trying to separate objects from their cast shadows some pixels belonging to the actual vehicles can be misclassified as shadow and this would cause partial erosion or holes to appear on the foreground objects mask. Creating a new mask by using the set of points included in the convex-hull would bring a solution for this problem. As can be seen from Figure 1 every two points in the convex hull (red stars) can define a line and when all the lines are considered we have a closed polygon. The new mask will be composed of all the points that fall inside this polygon.

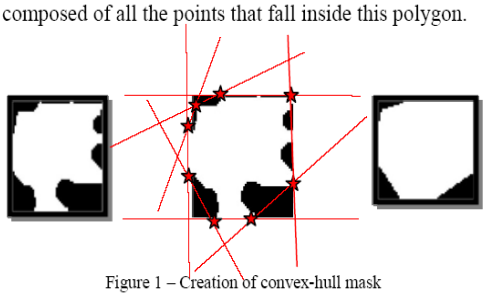


Figure 1 – Creation of convex-hull mask

Figure 2(g) and 2(h) shows the newly created convex hull mask and the segmented *RGB* foreground objects based on this new mask.

IV. TEST SEQUENCES

The background estimation and shadow removal algorithms discussed above were tested using two standard sequences (*Highway-I* and *Highway-II)* and one custom recorded video sequence. *Highway-I* sequence has a frame rate of 25 samples per second and a resolution of 352 × 288. *Highway-II* which is obtained from VISOR image lab [14] has a frame rate of 15 samples per second and a resolution of 320 × 240. The custom sequence was recorded by a Fuji Film Fine Pix S6000fd camera. The recording speed was set at 30 frames per second and the resolution was 680 × 480.

V. SIMULATION RESULTS

In this study the adaptive *K*-Gaussian mixture modeling was not applied to every single pixel in the input frame. Instead we looked at the difference between the current frame and a previous reference frame and if the difference was less than a previously defined threshold (35 in this study) the tested pixel was assumed to be part of the background. The adaptive model was only applied for the locations where the difference was more than the selected threshold. It was found that this would greatly speed up the processing.

For all video sequences K=7, α=0.05, β=1.5, and T =0.85 were used in the adaptive *K-MoG* model. The thresholds used by the *SCS* calculator have also been summarized in Table 1. While for the higher resolution custom video a separate set of threshold values were required for the standard sequences same set of values was sufficient.

TABLE I. SHADOW DETECTION ALGORITHM PARAMETERS

Yeni-Izmir Junction	TL=180	TC1=9.5	TC2=19	TG1 = 0.3	TG2 = 0.6
Highway-I	TL=200	TC1=7.5	TC2=15	TG1 = 0.5	TG2 = 1.0
Highway-II	TL=200	TC1=7.5	TC2=15	TG1 = 0.5	TG2 = 1.0

As pointed out in section 3 the first set of results were obtained using the *Yeni-İzmir* custom sequence. These results are presented in Figure 2. The second set of simulations were carried out using the standard *Highway-I* sequence. Figure 3 shows the results of background estimation, subtraction, and shadow removal steps as applied to frame #1230. Figure 3(c) depicts the extracted foreground with some shadow, (e) is the computed shadow confidence score, (f) shows the remaining foreground after shadow is removed, (g) is the convex hull based new mask and (h) is the foreground objects with minimal shadow.

In [12], an edge-based moving shadow removal algorithm composed of many steps (high computational complexity) had been proposed and the authors had claimed that their approach would give better results than the ones obtained in the *HSV* domain or than by using the *SCS* based map. This statement would only be true if the foreground segmentation is done

right after SCS is thresholded. If after the application of the threshold a convex hull is fitted to the partly deformed foreground and a new mask based on the points defining the convex hull is created then the segmented foregrounds using this new mask would be as good as the ones obtained in [12]. Simulation results shown in Figures 2, 3 and 4 all indicate that segmentation with a convex hull based mask works quite well.

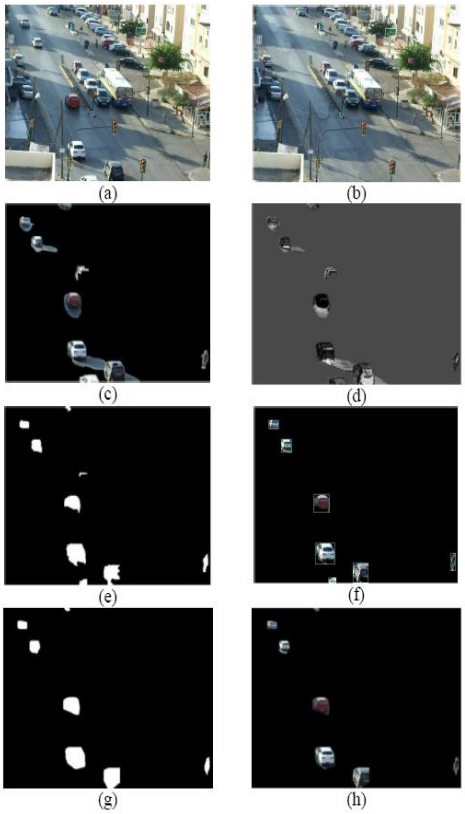


Figure 2 – Foreground segmentation and shadow removal for Yeni-İzmir Junction

(a) input frame (b) estimated background (c) RGB foreground with shadows (d) total SCS (e) shadow free foreground mask (f)bounding box cropped FG (g) new convex hull mask (f) FG objects cropped by new convex hull mask.

A. Analysis of Lane Fullness

Assuming that in real life each leg of an intersection is being monitored simultaneously by fixed surveillance cameras this section suggests a way for computing the fullness of a single leg of an intersection. In systems using fuzzy logic each leg houses two sensors behind traffic lights separated by a distance *D*. The sensor at distance *D* from the light counts the number cars coming to the intersection and the second counts the cars passing the traffic light. The amount of cars between the sensors is determined by the difference of the readings. However, this approach can not differentiate between a truck, a bus or a car. Hence determining what percent of the road is full based on size becomes fairly difficult.

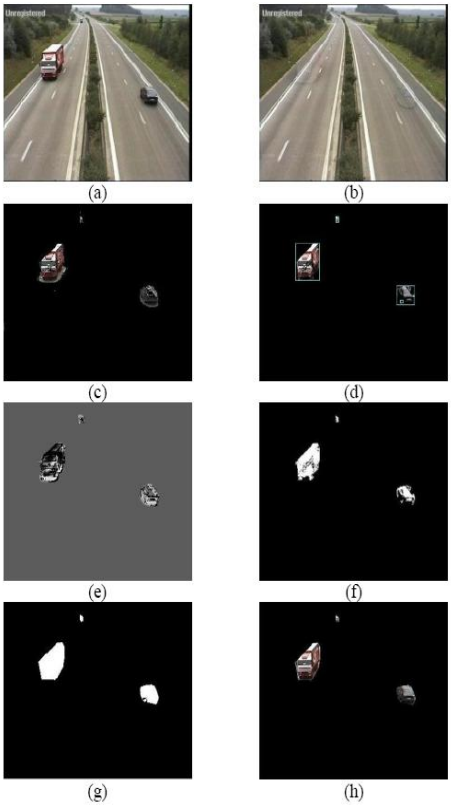


Figure 3– Segmentation and shadow removal for Highway-I

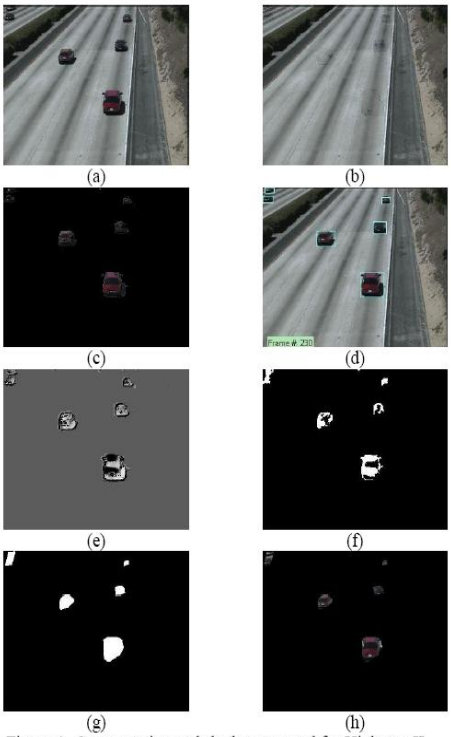


Figure 4– Segmentation and shadow removal for Highway-II

A better approach that would not require any information on the type of cars present behind the traffic lights would be the use of the foreground mask(with shadows removed) together with two lane masks for determining how much each lane and the detected foreground overlap outside a designated region *A* (ref to Figure 5(e)) . Afterwards we test to see if any of the foreground objects fall in this designated region. If region *A* contains no moving objects it is assumed 100% full. Otherwise the overlap between the extracted FG over region A and the ground truth mask of region A is computed. The application of the fullness analysis to the north leg of the intersection for frame #1890 is depicted in Figure 5.

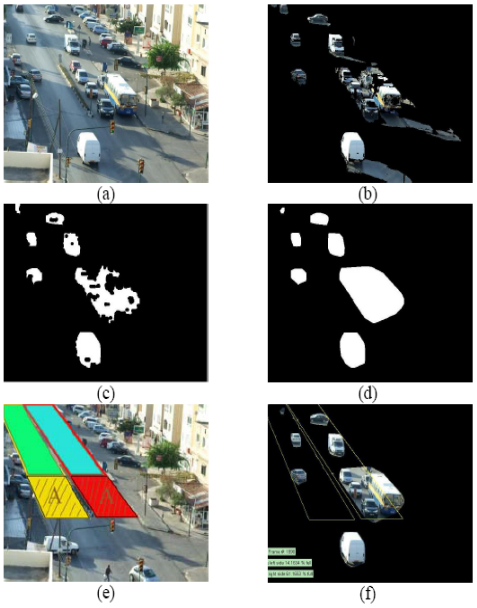


Figure 5– Lane masks and fullness analysis

VI. CONCLUSIONS

The paper proposed an adaptive signaling technique based on fullness analysis of the different legs of an intersection and gave an example for the north leg of *Yeni-İzmir* Junction of Famagusta. It also introduced the concept of applying a convex hull on the output obtained from shadow removal routines operating in *HSV* domain or using a combined probability map known as *SCS*. Various examples were provided using standard and custom sequences to show the advantage of applying the convex hull before segmentation. Future work will include the collection of all the fullness measures for the four different legs and signaling of the control device to switch the lights based on the analysis of the collected data.

ACKNOWLEDGEMENTS

This work is supported by a grant from the Ministry of Education and Culture of T.R.N.C. under the contract number BAP-290/115.

REFERENCES

[1] [1] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using Kalman-filtering," Proc. of inter. conf. on recent advances in mechatronics, ICRAM'95, pp.193-199, 1995.

[2] [2] C.R. Wren, A. Azearbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of human body," IEEE Trans. on PAMI, Vol.19 No.7, pp. 780-785, 1997.

[3] [3] T. Horprasert, D. Harwood, and L.S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," IEEE Frame-rate applications workshop, Kerkya-Greece, 1999.

[4] [4] C. Stauffer, and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," CVPR'99, Vol. 2, pp. 246-252, June 1999.

[5] [5] S-C S. Cheung, and C. Kamath, "Robust techniques for background subtraction in urban traffic video," Proc. of the SPIE, Volume 5308, pp. 881-892, 2004.

[6] [6] M. Harville, G. Gordon, and J. Woodfill, "Foreground segmentation using adaptive mixture models in color and depth," Proc. of the IEEE workshop on detection and recognition of events in video, 2001.

[7] [7] G. Stijinman, and R. Boomgaard, "Background extraction of colour image sequences using gaussian mixture mode," ISIS TR series, Vol. 10, 2000.

[8] [8] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modelling and subtraction by codebook construction," IEEE Int. conf. on image processing, 2004.

[9] [9] A. Elgammal, D. Harwood, and L.S. Davis, "Nan-parametric model for background subtraction," European conf. on computer vision, Vol. 2, pp. 751-767, 2000.

[10] [10] C. Jeong-Hoon, K. Tae-Gyun, J. Dae-Geun, and H. Chan-Sik, "Moving cast shadow detection and removal for visual traffic surveillance, "LNAI 3809, pp. 746-755, 2005.

[11] [11] S.K. F. George, H. C. Y. Nelson, K. H. P. Grantham, and H.S.L. Andrew, "Effective moving cast shadow detection for monocular colour traffic image sequences," Opt. Eng. 41(6), pp. 1425-1440, June 2002.

[12] [12] M. Xiao, C-Z Han, and L. Zhang, "Moving shadow detection and removal for traffic sequences," Inter. jour. of automation and computing, pp. 38-46, Jan 2007.

[13] [13] Gonzalez, R.C., and R.E. Woods, Digital Image Processing, Prentice Hall, New Jersey, 2002.

[14] [14]Video surveillance online repository at URL: http://www.openvisor.org.

Appendix B: Traffic Analysis of Avenues and Intersections Based on Video Surveillance from Fixed Video Cameras

Saameh G. Ebrahimi[1], *Nima Seifnaraghi*[1] ,*Erhan A. İnce*[1]

Elektrik ve Elektronik Mühendisliği Bölümü
Doğu Akdeniz Üniversitesi
golzadeh.s@gmail.com, nimaseif@yahoo.com, erhan.ince@emu.edu.tr

Özetçe

Bu makalede sabit bir görsel gözetim sistemi tarafından gözetlenmekte olan herhangi bir cadde, kavşak veya seçilmiş bölgedeki hareketli nesnelerin uyarlanır Gauss Fonksiyonları Karışımı (GFK) yöntemi kullanılarak arkaplandan ayrıştırılması ve kavşak kollarındaki doluluk oranı analizleri sunulmaktadır. Kavşak kollarında veya caddelerdeki doluluk oranlarını doğru kestirebilmek için öncelikle ön-plandaki gölgelerin mümkün olduğunca elimine edilmesi gerekmektedir. Bu çalışmada gölge kestirimi ve silinmesi renk özü, doygunluğu ve yeğinliği uzayı (HSV) kullanılarak gerçekleştirilmiştir. Benzetimler PETS 2001 Camera 1 dizini ve KKTC-Mağusa şehrinde kaydedilen bir dizinin kullanımıyla gerçekleştirilmiştir. Kavşak bacaklarının sağ ve sol şeritlenin yüzde cinsinden doluluk hesaplaması için yeni bir yöntem önerilmiş ve bu oranlar seçilmiş çerçeve üzerine işlenmiştir.

Abstract

Based on adaptive Gaussian mixture modelling this article presents the separation of foreground objects from frames of surveillance video taken at avenues and/or intersections. The paper also describes an approach for determining the lane fullness of a dedicated leg of an intersection. In order to give an accurate fullness measure the cast shadows that might be present in the segmented foregrounds must be eliminated. In this study the detection and removal of shadows have been carried out using the HSV color space. The simulations were carried out using the Camera 1 sequence from PETS 2001 database and a custom sequence recorded in TRNC-Famagusta. A new method for computing right and left lane fullness in each leg of the intersection has been proposed and values computed have been recorded on the bottom left corner of the frame under study.

1. Giriş

Görsel gözetlemede hareketli nesneleni arka plandan ayırmanın yaygın yolu arkaplan kestirimi ve mütakiben arkaplan çıkarımıdır. Bazı yöntemler bir pikselin zamana yayılmış seri halindeki yeğinlik değerlerinin tek doruklu bir dağılım fonksiyonu kullanılarak modellenebileceğini varsaymaktadır [1], [2]. Bununla birlikte tek doruklu bir model sallanan ağaç dalları veya savrulan kar zerreciklerinin neden vereceği çoklu arkaplanlarla başa çıkamamaktadır. Genellikle, karmaşık ve durağan olmayan arkaplanların modellenmesi işinde genelleştirilmiş Gauss Fonksiyonları Karışımı (*GFK*) kullanılmaktadır [3],[4],[5]. Modelleme esnasında ilklendirme ve parametre güncelleme beklenti enbüyütme yöntemi (*EM*) [6] veya *K*-ortalama yöntemi ile gerçekleştirilebilir. *EM* ilklendirmesi kullanan yöntemlerin daha iyi sonuç verdiği bilinse de bu yaklaşımın karmaşıklığı ve başlangıçta çevrimdışı olmasından dolayı bu çalışmada K-ortalama ilklendirmesi kullanılmıştır. Modelleme esnasında zeminde hızlı değişimler oluyorsa *GFK*'nın bile 3-5 adet Gauss fonksiyonu ile yeterince doğru bir sonuç vermeyeceği [7] de belirtilmektedir. Bu problemleri çözecek parametrik olmayan ve çekirdek yoğunluk kestirimi kullanan bir yöntem [8] de sunulmuştur. Bu çalışmada görsel sistemin kaydettigi çerçevelerdeki zeminde hızlı ve ani değişiklikler olmadığından parametrik olmayan yöntemlerin kullanımına ihtiyaç olmamıştır.

Bildiri düzeni aşağıdaki gibidir. İkinci kısım Gauss Fonksiyonları Karışım Modeli ve bu modeldeki parametre güncelleme detaylarını anlatmaktadır. Daha sonra 3. kısımda önplan/arkaplan ayrıştırma işlemi sonunda ön planın bir parçası olarak bulunan gölgelerin *HSV* uzayında kestirimi ve silinmesi anlatılmıştır. Dördüncü bölüm kullanılan video test dizinlerini ve bu dizinlerin özelliklerini belirtmiş, beşinci kısımda ise benzetim sonuçları sunulmuştur.

2. Gauss Fonksiyonları Karışım Modeli

Uyarlanır Gauss fonksiyonları karışım modeli [3] ve [4] de belirtildiği üzere her çokuzlu piksel vektörünü *K*-adet Gauss dağılımı karışımından oluşacak şekilde modellemektedir. Çokuzlu piksel vektörü X_t kırmızı, yeşil ve mavi bileşenlerden oluşan bir yeğinlik değerler vektörünü, $w_{i,t}$ belirli bir zamanda karışımdaki her Gauss dağılım fonksiyonu için kestirilmiş bir katsayıyı, $\mu_{i,t}$ ve $\Sigma_{i,t}$

ise karışımdaki her dağılım fonksiyonunun avaraj değer ve ortak değişinti matrisini temsil etmektedir.

Ön/arka plan ayrıştırma işlemi esasen iki bağımsız problem olarak görülebilir. Bunlardan ilki K elemanlı karışımdaki Gauss fonksiyonları ile ilgili parametrelerin kestirimi, ikincisi ise her dağılımın arkaplanı temsil etme olasılığının değerlendirilmesidir.

2.1 Parametre Güncellenmesi

Başlangıçta tüm Gauss fonksiyonları eşit olasılıklı olduğu için tüm katsayılar, $w_{i,t}$, $1/K$ değerine eşitlenmekte ve değişinti değerleri de rastgele yüksek değerler olarak alınmaktadır. Daha sonra her çokuzlu piksel vektörü eldeki K-adet Gauss fonksiyonu ile karşılıştırılmaktadır. Bir çakışma durumunda (çokuzlu piksel vektörü X_t'nin herhangi bir dağılımdan 2.5 standart sapmadan daha yakın olma durumu) ilgili dağılım ve/veya dağılımların parametreleri güncellenmektedir.

Tarama sonucunda bir çakışma bulunmaması durumunda katsayısı en düşük olan Gauss dağılımı bir yenisi ile değiştirilir. Bu yeni dağılımın katsayısı düşük, değişintisi yüksek ve avaraj değer vektörü ise X_t ye eşit tutulur. Gauss dağılımına ait katsayılar ise her gözlemleme zamanı için değiştirilmektedir. Çakışma var ise $M_{i,t}$ 1 diğer durumlar için 0 olarak kabul edilmektedir.

2.2 Arkaplan Kestirimi

Her çokuzlu piksel vektörü için parametre güncellemesi yapıldıktan sonra dağılımlar w_k/σ_k değerlerine göre sıralanmakta ve üst tarafta kalan B adet dağılımın zemini en iyi temsil eden dağılımlar olduğu kabul edilmektedir.

$$B = \arg\min_{b}\left\{\sum_{k=1}^{b} w_k > T\right\} \qquad (1)$$

Denklem (1)'deki T eşik değeri 0.5 ile 1 arasında değişebilmektedir.

2.3 Morfolojik İşlemler

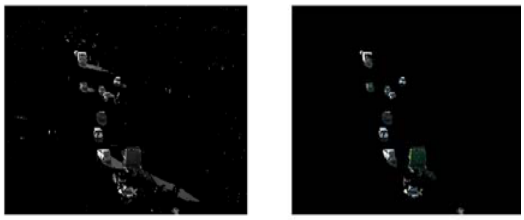
Üzerinde çalışılan çerçeve ön ve arkaplan olarak ayrıştırıldıktan sonra genelde ön-planda hareketli nesneler ile birlikte bazı gürültüler de bulunur. Bu gürültüleri mümkün olduğunca azaltmak için bazı morfolojik operatörlerden yararlanılmaktadır. Şekil 1 de ayrıştırma sonrası elde edilen bir ön plan görüntüsündeki gürültünün nasıl minimize edildiğini gösterilmiştir. Gürültüden kurtulmak için hem kapama ve açma işlemleri hem de bağlantılı bileşen analizi uygulanmış ve piksel sayısı toplamı belli bir eşik değerin altında olan bileşenler gürültü kabul edilip ön plan çerçeveden silinmiştir.

3. Gölgesizleştirme

Günün değişik saatlerinde cadde üzerinde hareketli haldeki nesneler belli açıdan vuran ışığı bloke edeceği için yol üzerinde gölgeler oluşacaktır. Bu gölgeler ışık kaynağının yüksekliğine göre uzayabilmekte veya daralabilmektedir. Gölgelerin uzaması durumunda birbirinden bağımsız iki nesne birleşebilmekte ve bu hem araç sayımını hem de şerit doluluk oranı hesaplarını olumsuz yönde etkilemektedir. Bu yüzden çalışmamızda [10] da belirtilen renk özü, doygunluğu ve yeğinliği uzayını kullanan bir gölge nokta maskesi (*GNM*) ve deneysel olarak elde edilen bazı eşik değerleri gölgesizleştirme amaçlı kullanılmıştır. Bu çalışmada gölge sezim ünitesince 'ht'ya. duyulan α, β, τ_S ve τ_H değerleri sırası ile 0.48, 0.95, 0.4 ve 0.7 olarak alınmışlardır.

$$GNM_k(x,y) = \begin{cases} 1 & \left\{\alpha \le \dfrac{I^V{}_k(x,y)}{B^V{}_k(x,y)} \le \beta \right\} \cap \\ & \left\{\left(I^S{}_k(x,y) - B^S{}_k(x,y)\right) \le \tau_s \right\} \cap \\ & \left\{\left|I^H{}_k(x,y) - B^H{}_k(x,y)\right| \le \tau_H \right\} \\ 0 & \textit{aksi takdirde} \end{cases} \qquad (2)$$

Denklem (2) de $I_k(x,y)$ ve $B_k(x,y)$ giriş video dizini ve arkaplan modellerinin k'inci çerçevesindeki (x,y) koordinatlı piksel değerlerini temsil etmektedir.



(a) (b)

Şekil 1: Magosa_Yeni_İzmir_Kavşağı, çerçeve 2100 de gürültüsüzleştirme ve gölgesizleştirme.

4. Video Test Dizinleri

Yapılan çalışmaların performans değerlendirmesi için PETS 2001 [11] veri tabanından Kamera 1 ve Kamera 2 dizinleri ile Magosa Yeni İzmir kavşağında AVI olarak kaydedilmiş "Magosa_Yeni_İzmir_Kuzey" dizini kullanılmıştır. Fuji S6500 marka kamera ile kaydedilen dizin 640×480 çözünürlük ve saniyede 30 çerçeve içermektedir. Kamera kayıt yaparken hareketli *JPEG* sıkıştırması uyguladığından çerçevelerin MATLAB ortamına okunabilmesi için sıkıştırmayı açacak bir kod çözücüsü yüklenmiştir.

5. Benzetim Sonuçları

Bu çalışmada video dizinlerindeki her çerçeveye Gauss fonksiyonları karışım modeli uygulanmadan önce bir

eşikleme uygulanmıştır. Güncel çerçevedeki (x,y) koordinatlarındaki piksel geçmişle kıyaslandığında oldukça uzun bir süre değişmemişse bu piksel sabit varsayılmaktadır. Bir başka deyişle güncel çerçeve ile önceki bir referans çerçevesindeki değer farkı deneysel olarak seçilmiş bir eşik değerinin (minDiff) altında ise bu piksel arkaplanın bir parçası olarak kabul edilmekte ve uyarlanır süreç atlanmaktadır. Bu yaklaşım sadece kısıtlı sayıdaki koordinatta uyarlanır süreci kullanacağı için hem çerçeve hem de dizinin çok daha hızlı işlenmesine neden vermektedir.

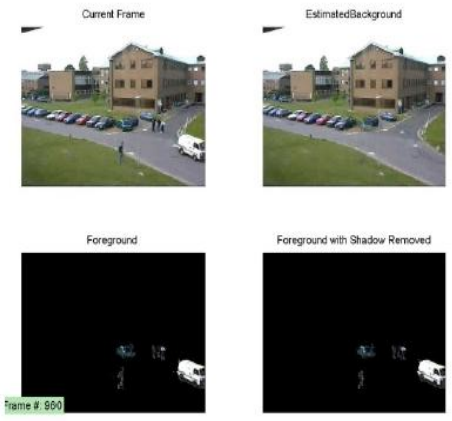
Önceki bölümde belirtilen test video dizinleri için bu çalışmada seçilen eşik değeri 35dir. 5-bileşenli *GFK* modelinde $\alpha = 0.05$, $\beta = 2.6$ ve $T = 0.85$ olarak alınmıştır. İlk deneme PETS-2001 veritabanından Kamera 1 test dizinine uygulanmış ve geliştirilen algoritma arkaplan kestirimi ve ayrıştırmayı başarıyla gerçekleştirmiştir. Şekil 2 bu dizinin 960'ıncı çerçevesi için ayrıştırma, gölgesizleştirme ve hareketli nesnelerin belirlenmesi işlerini göstermektedir. Hareketli nesnelerin birbiriyle fiziki olarak örtüştüğü durumlarda nesneleri ayırma işlemine gidilmemiştir.

İkinci deneme Magosa şehrinde Yeni İzmir kavşağında çekilen bir video dizini kullanılarak gerçekleştirilmiştir. Bu dizin yakın çekimle kavşağın kuzey bacağını görüntülemektedir. Güneşli bir günde ve öğle vaktinden önce çekilen bu dizinde hareketli cisimlerin ışığı bloke etmesiyle yol üzerinde oluşmuş değişik boylarda gölgeler mevcuttur. Bu gölgeler hem nesnelerin ön planda birleşmesine hem de hareketli nesneleri belirlerken yanlış kabullere neden verebilmektedir. Gölge sezim ünitesinde α, β, τ_S ve τ_H değerleri sırası ile 0.48, 0.95, 0.4 ve 0.7 olarak alındığında Şekil 3 deki sonuçlar elde edilmiştir. Çalışma yanlış kabulleri minimize etmek amaçlı olarak hareketli nesne bulunduğu varsayılan bölge ile arkaplan resmindeki ayni bölge arasındaki ilinti değerlerini de hesaplamış ve bu değerin yüksek olduğu durumlar için o bölgeyi seçmekten kaçınmıştır.

5.1 Şerit Doluluk Analizi

Geleneksel trafik ışık sistemleri ya eşit ya da her şeridin uzunluğu ile orantılı bekleme peryotları gerektirmektedir. Bu tür sistemler ve uyguladıkları mantık ile trafik akışının optimize edilmesi yetersiz kalmaktadır. Dörtyolların değişik bacaklarında bekleyen araçlara uyarlanır işaretleşmeye dayalı yol vermenin (ihtiyaca ve doluluk oranına bağlı olarak) tüm yönlerdeki trafik akışını optimize edeceği düşünülmektedir.

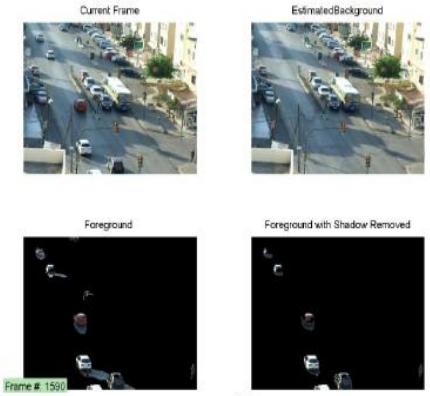
Denetim altındaki kavşağın tüm bacaklarının tek bir güvenlik kamerası ile izlenmesi mümkün olmadığından bu çalışmada sadece kavşağın kuzey bacağı için doluluk oranı



(a)

(b)

Şekil 2: PETS 2001 Kamera 1 dizininin 960'ıncı çerçevesi



(a)

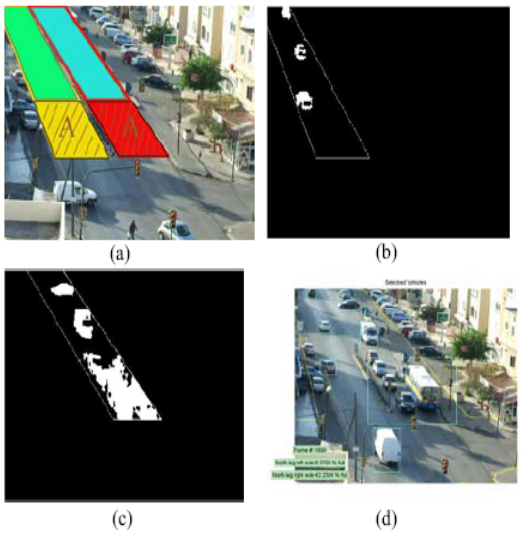
(b)

Şekil 3: Magosa_YeniİzmirKuzey dizinindeki 1590'ıncı çerçevede önplan ayrıştırma , gölgesizleştirme ve hareketli nesne tesbiti.

hesaplamaları verilmiştir. Şekil 4(a) ′daki şerit maskeleri ile gölgesizleştirilmiş önplandaki hareketli nesnelerin yüzde kaç örtüştüğünü (şerit doluluk oranı) tespit edilerek her şeridin yüzde doluluk oranı bulunabilmektedir. Trafik ışıklarına yakın ön bölgede sınırları belli bir alan seçilmekte (alan-*A*), ve ön/arka plan ayrıştırmasından sonra hareketli nesnelerden herhangi birinin bu alan içinde olup olmadığı sorgulanmaktadır. Seçilmiş bölge içerisine herhangi bir hareketli nesne düşmezse bu bölge 100 % dolu kabul edilmekte geriye kalan yol yüzeyinin doluluk oranı ise önplan ve yol maskelerinin örtüşme oranına göre hesaplanmaktadır.

İnceleme altındaki bacağın her iki şeridinin de doluluk oranı 1890′ıncı çerçeve için Şekil 4 de verilmiştir. Tüm yönlerdeki trafik akışını bu mantıkla kontrol edebilmek için dörtyolun her bacağında ayni analiz eş zamanlı olarak yapılmalı ve kıyaslamalar sonunda kontrol cihazına bir sinyal gönderilmesi gerekmektedir. Eş zamanlı analiz her bacağın ayrı bir *IP*-kamerasıyla izlenmesini şart koşacaktır.

Şerit doluluk analizi yaparken trafik ışıklarına yakın olan, ön bölgede belirlenen alan-*A*′nın derinliği, bu çalışmada sabit tutulmuştur. Fakat video dizininin her 5 dakikalık aralığı taranıp trafik akış hızı belirlenirse bu derinliğin uyarlamalı şekilde değiştirilmesi mümkün olacaktır. Bu da günün farklı zamanlarında yapılan doluluk tahminlerinin daha gerçekçi olmasını sağlayacaktır.



Şekil 4: Yeni_İzmir kavşağının kuzey bacağındaki sağ ve sol şeritlerdeki doluluk oranı analizi.

6. Vargı ve İleriki Çalışmalar

Bu çalışmada ana caddeler veya herhangi bir kavşak üzerindeki nesnelerden hareketli olanların arkaplan kestirimi ve ön/arka plan ayrıştırması, gürültüsüzleştirme ve gölgesizleştirme işlemleri sonrası belirlenmesi ve trafik ışıklarının akıllı kontrolu için kavşağın değişik bacaklarındaki doluluk analizinin nasıl yapılabileceği konuları işlenmiştir. Seçilmiş bazı video dizinlerinye yapılan benzetimler sonrası yöntemlerin başarıyla uygulandığını göstermek amacıyla makalede örnekler sunulmuştur. Gölgesizleştirme işlemi hiçbir zaman 100 % doğru olmadığı için ön plan görüntüsünde hareketli nesne diye belirlenen bazı bölgeler yanlış kabul çıkmaktadır. Bu yüzden şerit doluluk oranlarında 5% lik bir hata payı olabileceğini kabul etmek gerekir. Projenin devamında ayrıştırılan ön plandaki nesnelerin özellik çıkarımı, nesne sınıflandırma ve kırmızı ışık ihlalleri üzerinde araştırmalar yürütülecektir.

7. Teşekkürler

Bu makalede sunulan sonuçlar KKTC Milli Egitim ve Kültür Bakanlığı tarafından finanse edilen BAP-290/115 nolu proje çerçevesinde yürütülen çalışmalarla elde edilmiştir.

8. Kaynakça

[1] Wren, C.R., Azearbayejani, A., Darrell T., and Pentland, A., "Pfinder: Real-time tracking of human body", *IEEE Trans. On PAMI, Vol.19 No.7, pp. 780-785,1997.*

[2] Horprasert, T., Harwood, D., and Davis, L.S., "A statistical approach for real-time robust background subtraction and shadow detection", *IEEE Frame-Rate Applications Workshop, Kerkya-Greece, 1999.*

[3] Stauffer, C. and Grimson, W.E.L., "Adaptive background mixture models for real-time tracking", *CVPR, Vol. 2, pp. 246-252, June 1999.*

[4] Cheung S-C S. and Kamath C., "Robust techniques for background subtraction in urban traffic video", *Proceedings of the SPIE, Volume 5308, pp. 881-892, 2004.*

[5] Harville, M., Gordon, G., and Woodfill J., "Foreground segmentation using adaptive mixture models in color and depth", *Proceedings of the IEEE Workshop on Detection and Recognition of Events in Video, 2001.*

[6] Stijinman, G., and Boomgaard, R., "Background extraction of colour image sequences using gaussian mixture model", *ISIS TR series, Vol. 10, 2000.*

[7] Kim, K., Chalidabhongse, T.H., Harwood, D., and Davis, L., "Background modeling and subtraction by codebook construction", IEEE International Conference on Image Processing (ICIP), 2004.

[8] Elgammal, A. and Harwood, D., and Davis, L.S., "Nan-parametric model for background subtraction", *European Conf.on Computer Vision, Vol. 2, pp. 751-767, 2000.*

[9] Ergezer, H. and Leblebicioğlu, K., "Visual detection and tracking of moving objects", *Signal Processing and Communications Applications, SIU 2007.*

[10] Cucchiara, R., Grana, C., Piccardi, M., Prati, A., and Sirotti, S.,"Improving shadow suppression in moving object detection with HSV color information", *IEEE Intelligent Transportation Systems Conference, pp. 334-339, Aug 2001.*

[11] PETS2001, http://ftp.pets.rdg.ac.uk/PETS2001/